CH32FV2x_V3x Reference Manual

V1.7

http://wch.cn

Overview

The CH32F2x is a general microcontroller based on the ARM®Cortex™-M3 core, and is compatible with most ARM tools and software. It provides abundant communication interfaces and control units and is applicable to most embedded fields of control, connection, and integration, etc.
The CH32V2x and CH32V3x are industrial general microcontrollers based on the 32-bit RISC-V instruction set and architecture. And they are based on different cores according to product performance and resource differences. CH32V203x adopts QingKe V4B core, which supports hardware interrupt stack and improves interrupt response efficiency. CH32V208x adopts QingKe V4C core, which further speeds up hardware division operation and adds memory protection function. CH32V303x/305x/307x adopts QingKe V4F core, which further supports hardware floating point operations. This series of products is equipped with abundant peripheral interfaces and functional modules. Its internal organizational structure supports low-cost and low-power embedded application scenarios.
This reference manual targets application developers, it provides detailed information on how to use CH32F2x series, CH32V2x series and CH32V3x series products. It is applicable to products with different memory capacities, functional resources, and packages in the series. If there have differences, special instructions will be given in the corresponding function chapters.

Please refer to the following datasheet for the device characteristics of this series.
CH32F20x_D6: *CH32F203DS0*
CH32F20x_D8, CH32F20x_D8C:*CH32F207DS0.*
CH32F20x_D8W:*CH32F208DS0.*
CH32V20x_D6, CH32V20x_D8: *CH32V203DS0.*
CH32V20x_D8W:*CH32V208DS0.*
CH32V30x_D8, CH32V30x_D8C: *CH32V307DS0.*

For information about the ARM®Cortex™-M3 core, please refer to the *ARM® Cortex®-M3 Processor Technical Reference Manual Revision r2p1*, which can be downloaded from the ARM corporate website.
For information about the RISC-V core, please refer to the ChingKeV4 Microprocessor Manual: *ChingKeV4_Processor_Manual.*

CH32F2x serial products overview

Low-and-medium-density general (F203)		High-density general (F203)		Connectivity (F205)	Interconnectivity (F207)	Wireless (F208)
32K Flash	64K Flash	128K Flash	256K Flash	128K Flash	256K Flash	128K Flash
10K SRAM	20K SRAM	32K SRAM	64K SRAM	32K SRAM	64K SRAM	64K SRAM
2*ADC(TKey) ADTM 2*GPTM 2*USART SPI I2C USBD USBFS CAN RTC 2*WDG 2*OPA	2*ADC(TKey) ADTM 3*GPTM 4*U(S)ART 2*SPI 2*I2C USBD USBFS CAN RTC 2*WDG 2*OPA	2*ADC(TKey) 2*DAC ADTM 3*GPTM 3*USART 2*SPI 2*I2C USBD CAN RTC 2*WDG	2*ADC(TKey) 2*DAC 4*ADTM 4*GPTM 2*BCTM 8*U(S)ART 3*SPI(2*I2S) 2*I2C USBD CAN RTC 2*WDG 4*OPA RNG SDIO FSMC	2*ADC(TKey) 2*DAC 4*ADTM 4*GPTM 2*BCTM 5*U(S)ART 3*SPI(2*I2S) 2*I2C OTG_FS USBHS(+PHY) 2*CAN RTC 2*WDG 4*OPA RNG SDIO	2*ADC(TKey) 2*DAC 4*ADTM 4*GPTM 2*BCTM 8*U(S)ART 3*SPI(2*I2S) 2*I2C OTG_FS USBHS(+PHY) 2*CAN RTC 2*WDG 4*OPA RNG SDIO FSMC DVP ETH-1000MAC 10M-PHY	ADC(TKey) ADTM 3*GPTM GPTM(32) 4*U(S)ART 2*SPI 2*I2C USBD USBFS CAN RTC 2*WDG 2*OPA ETH-10M(+PHY) BLE5.3

Note: Please confirm the product package when selecting, as the number of some peripherals and function of the same series product may be limited by the package.

CH32V2x and CH32V3x serial products overview

Low- and medium-density general (V203)		High-density general (V303)		Connectivity (V305)	Interconnectivity (V307)	Wireless (V208)
QingKe V4B core		QingKe V4F core				QingKe V4C core
32K Flash	64K Flash	128K Flash	256K Flash	128K Flash	256K Flash	128K Flash
10K SRAM	20K SRAM	32K SRAM	64K SRAM	32K SRAM	64K SRAM	64K SRAM
2*ADC(TKey) ADTM 3*GPTM 2*USART SPI I2C	2*ADC(TKey) ADTM 3*GPTM 4*U(S)ART 2*SPI 2*I2C	2*ADC(TKey) 2*DAC ADTM 3*GPTM 3*USART 2*SPI	2*ADC(TKey) 2*DAC 4*ADTM 4*GPTM 2*BCTM 8*U(S)ART	2*ADC(TKey) 2*DAC 4*ADTM 4*GPTM 2*BCTM 5*U(S)ART	2*ADC(TKey) 2*DAC 4*ADTM 4*GPTM 2*BCTM 8*U(S)ART	ADC(TKey) ADTM 3*GPTM GPTM(32) 4*U(S)ART 2*SPI

USBD USBFS CAN RTC 2*WDG 2*OPA	USBD USBFS CAN RTC 2*WDG 2*OPA	2*I2C USBFS CAN RTC 2*WDG 4*OPA	3*SPI(2*I2S) 2*I2C USBFS CAN RTC 2*WDG 4*OPA RNG SDIO FSMC	3*SPI(2*I2S) 2*I2C OTG_FS USBHS(+PHY) 2*CAN RTC 2*WDG 4*OPA RNG SDIO	3*SPI(2*I2S) 2*I2C OTG_FS USBHS(+PHY) 2*CAN RTC 2*WDG 4*OPA RNG SDIO FSMC DVP ETH-1000MAC 10M-PHY	2*I2C USBD USBFS CAN RTC 2*WDG 2*OPA ETH- 10M(+PHY) BLE5.3

Note: Please confirm the product package when selecting, as the number of some peripherals and function of the same series product may be limited by the package.

RISC-V cores version comparison overview

Features / Core	Instruction set	Hardware stack	Interrupt nested	Fast interrupt channel	Integer division cycle	Vector table mode	Expand instruction	Memory Protection
QingKe V4B	IMAC	2	2	4	9	Address/Command	Supported	None
QingKe V4C	IMAC	2	2	4	5	Address/Command	Supported	Standard
QingKe V4F	IMAFC	3	8	4	5	Address/Command	Supported	Standard

Abbreviated description of the bit attribute in the register:

Register bit properties	Property Description
RF	Read-only property, reads a fixed value.
RO	Read-only property, changed by hardware.
RZ	Read-only property, auto bit clear 0 after read operation.
WO	Write only property (not readable, read value uncertain)
WA	Write only property, writable in safe mode.
WZ	Write only property, auto bit clear 0 after write operation
RW	Readable and writable.
RWA	Read only property, writable in safe mode.
RW1	Readable, write 1 is valid, write 0 is invalid.
RW0	Readable, write 0 is valid, write 1 is invalid.
RW1T	Readable, write 0 invalid, write 1 flipped.
RW1Z	Readable, write 1 to clear this bit, write 0 is invalid.
SC	Automatically cleared.

Classification description of CH32 MCUs:

Classification abbreviation	Size of FLASH	Product type
D6	32KB or 64KB	Low-and-medium-density general
D8	128KB or 256KB	High-density general
D8C	128KB or 256KB	Connectivity or interconnectivity
D8W	128KB or 256KB	Wireless

Note: D (Density), 6 (2^6), 8 (2^8)

Specific classification abbreviations:

CH32F20x_D6: CH32F203K8, CH32F203C6 and CH32F203C8.

CH32F20x_D8: CH32F203CB, CH32F203RC, CH32F203VC and CH32F203RB.

CH32F20x_D8C: CH32F205RB and CH32F207VC.

CH32F20x_D8W: CH32F208RB and CH32F208WB.

CH32V20x_D6: CH32V203F6, CH32V203G6, CH32V203K6, CH32V203F8, CH32V203G8, CH32V203K8, CH32V203C6 and CH32V203C8.

CH32V20x_D8: CH32V203RB.

CH32V20x_D8W: CH32V208GB, CH32V208CB, CH32V208RB and CH32V208WB.

CH32V30x_D8: CH32V303CB, CH32V303RB, CH32V303RC and CH32V303VC.

CH32V30x_D8C: CH32V305FB, CH32V305RB, CH32V307RC, CH32V307WC and CH32V307VC.

Chapter 1 Memory and Bus Architecture

1.1 Bus architecture

The CH32F2x is a microcontroller designed based on the ARM®Cortex™-M3 core. The core, arbitration unit, DMA module, and SRAM memory, etc. in the framework interact through multiple sets of buses.
The CH32V2x and CH32V3x are general microcontrollers designed based on the RISC-V instruction set. The core, arbitration unit, DMA module and SRAM memory, etc. of the architecture interact through multiple sets of buses. The system architecture is as shown in Figure 1-2.

Figure 1-1 CH32F203(small-and-medium capacity general-purpose) system architecture

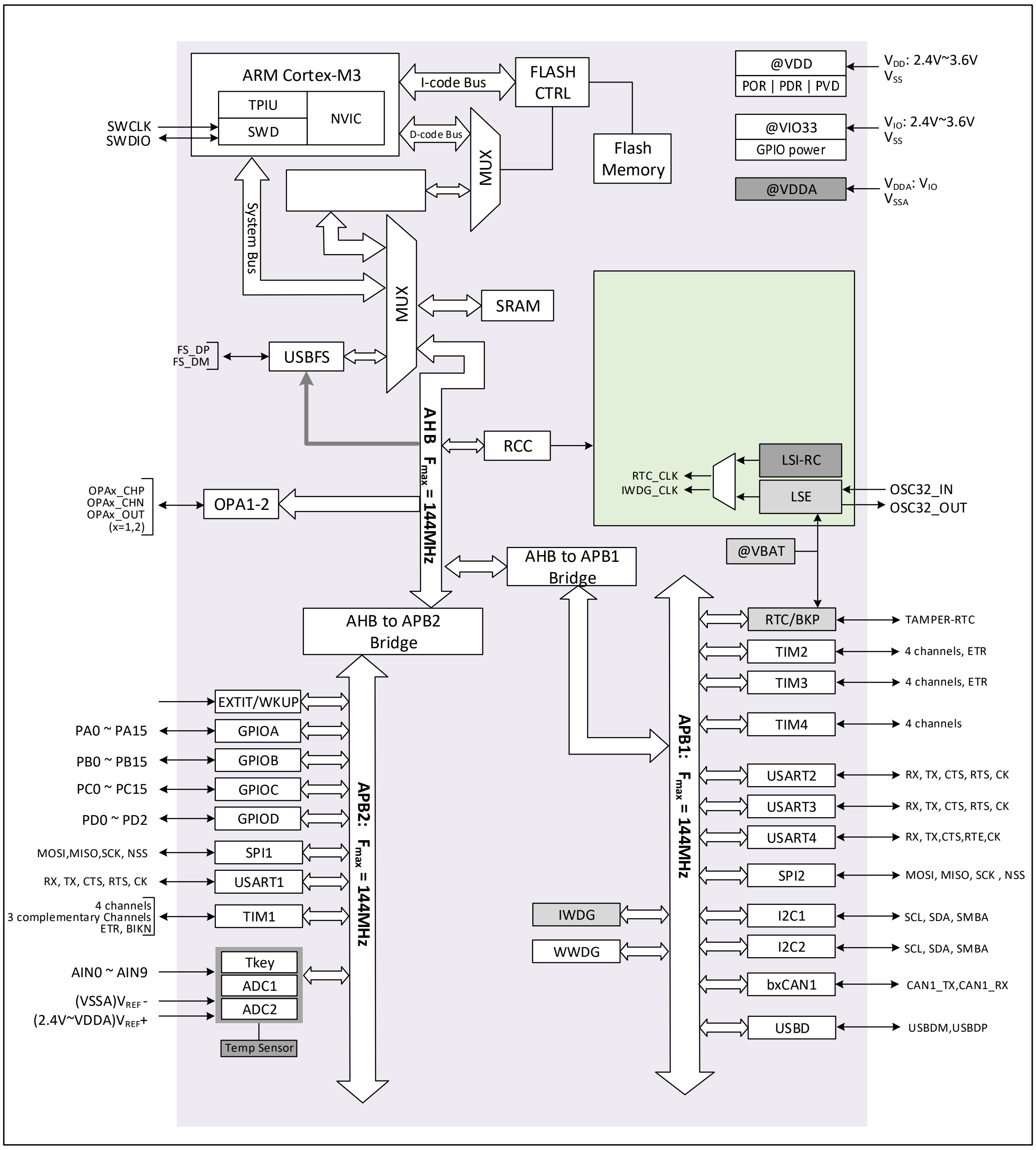


Figure 1-2 CH32V2x (connection/interconnection/high capacity general-purpose) system architecture

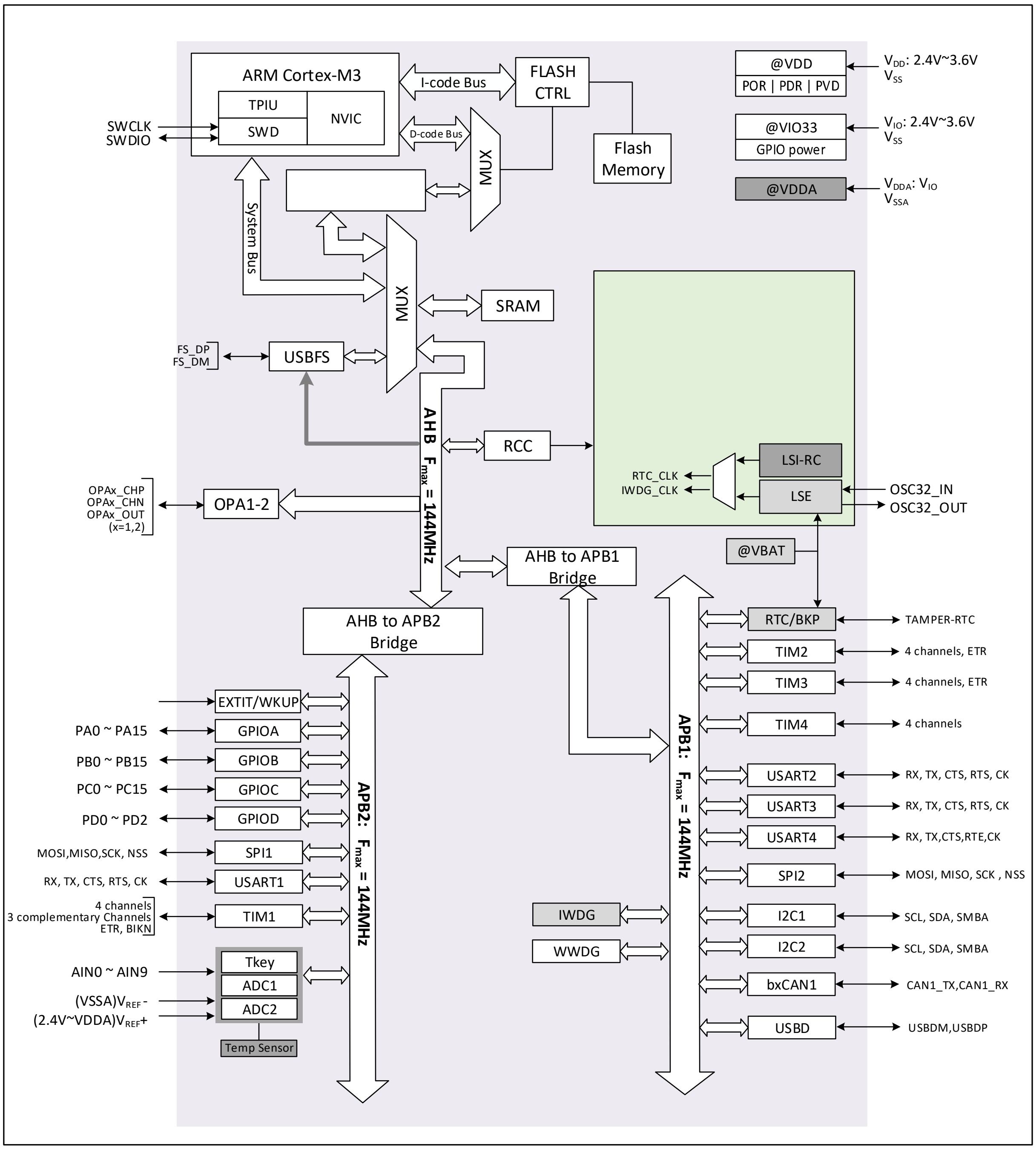


Figure 1-3 CH32F208 (wireless) system architecture

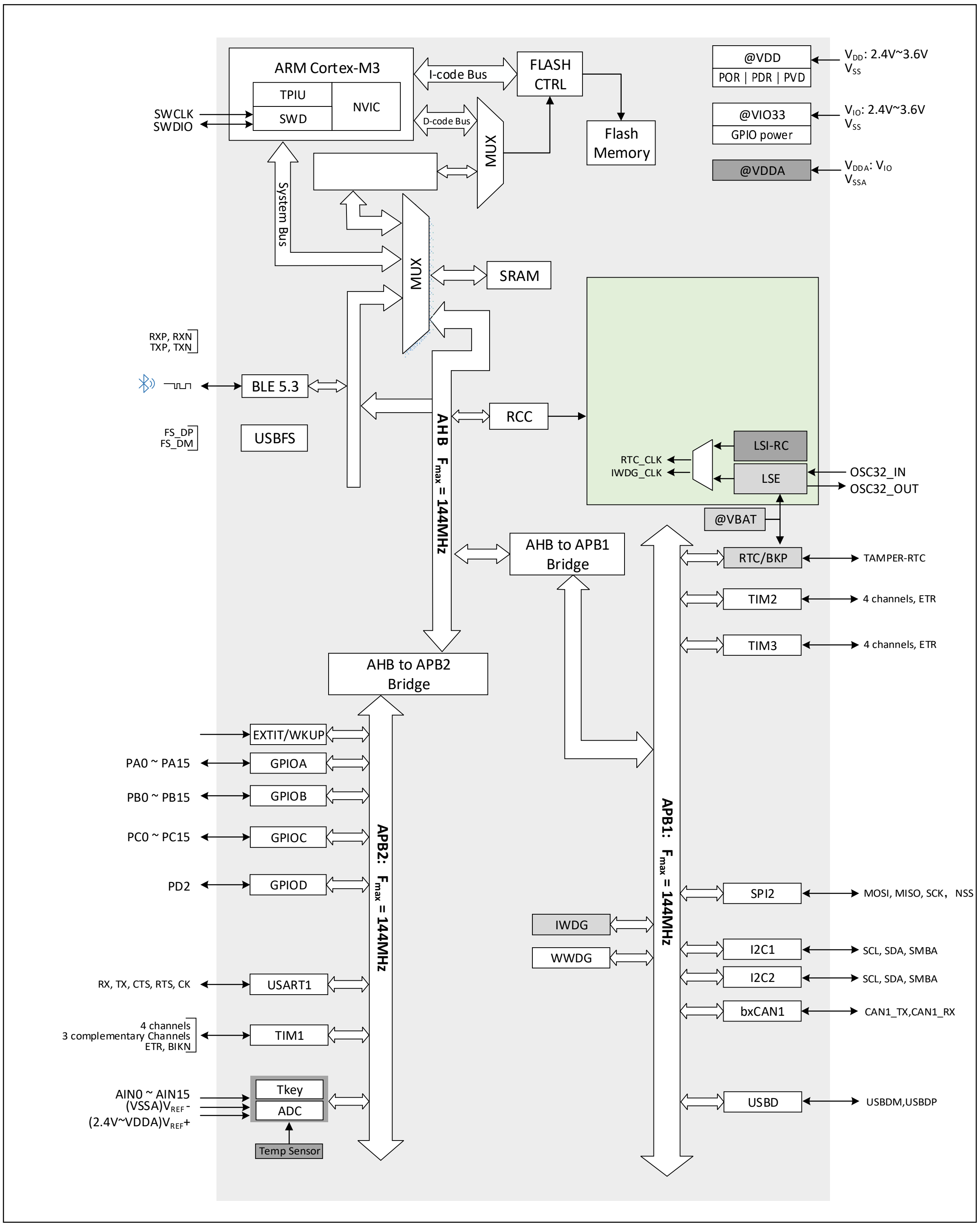


Figure 1-4 CH32V203 system architecture

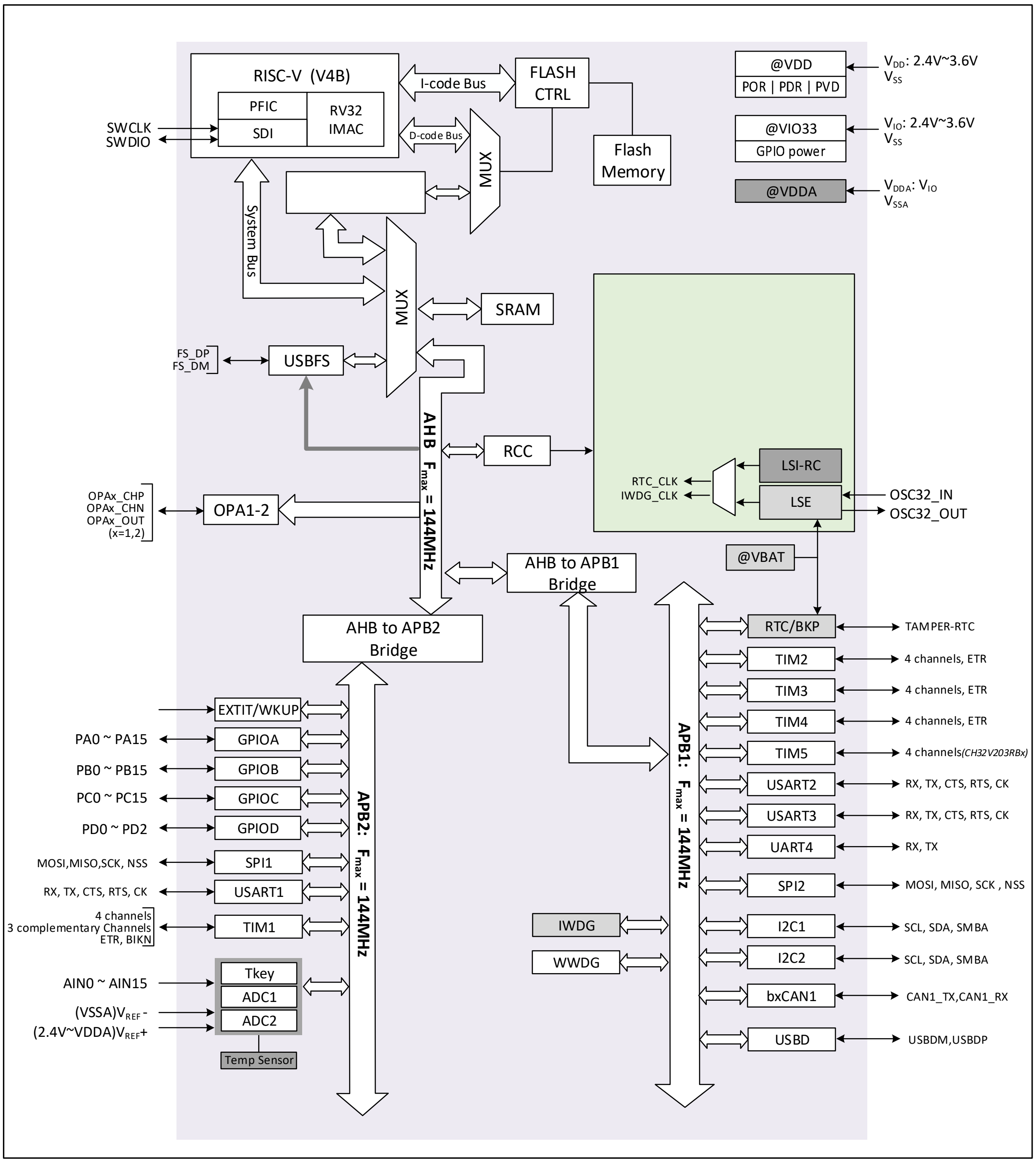


Figure 1-5 CH32V208 system architecture

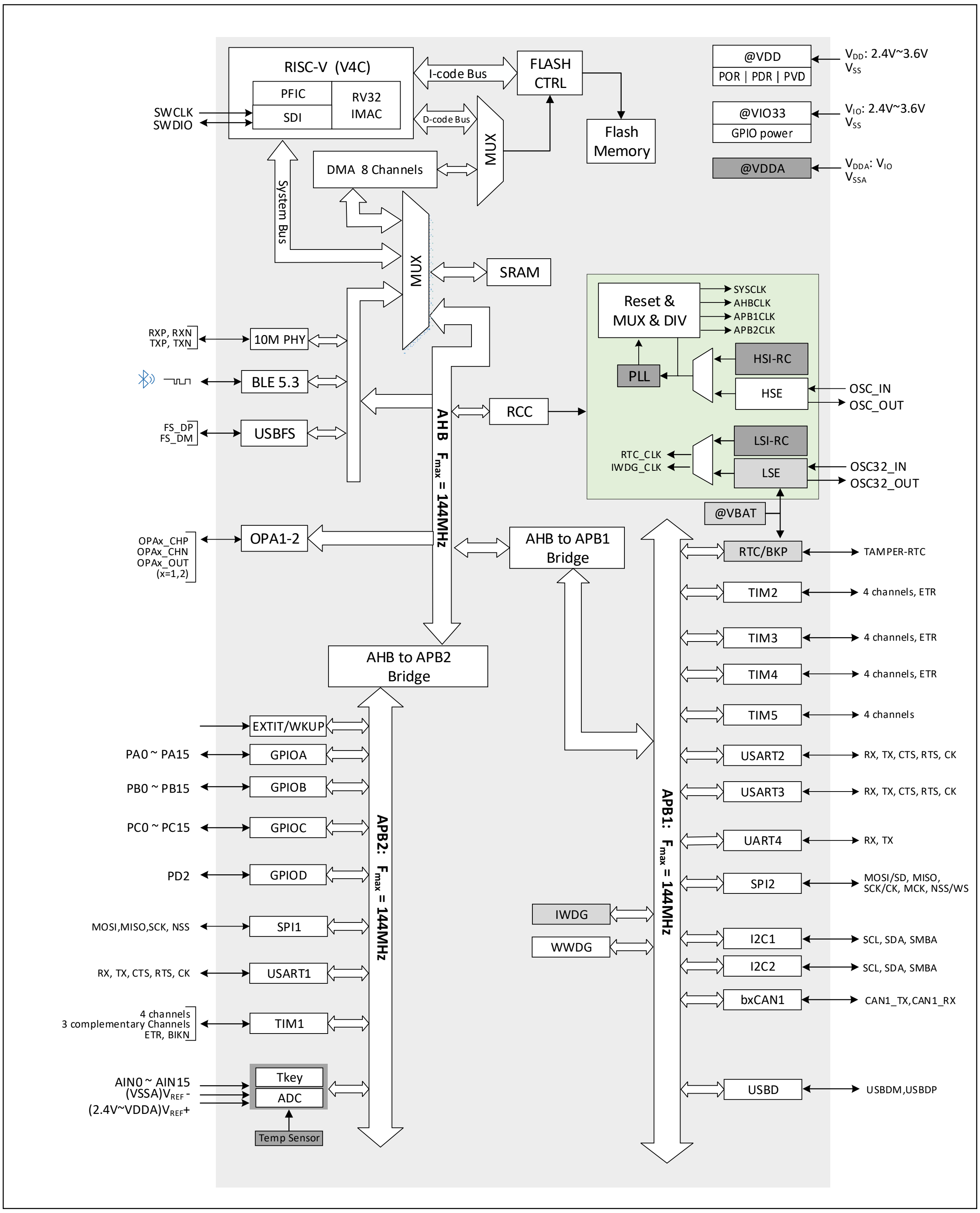
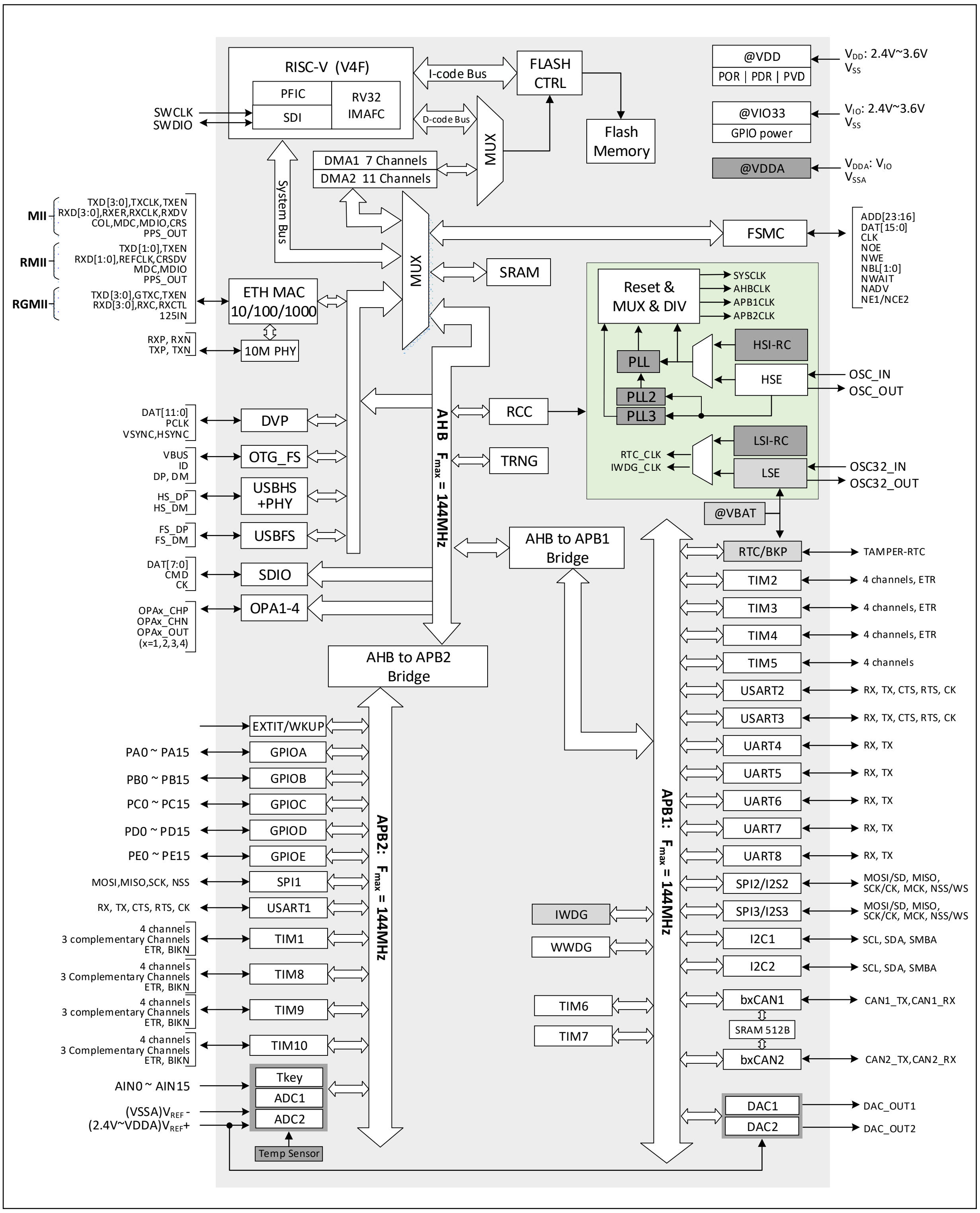


Figure 1-6 CH32V3x system architecture

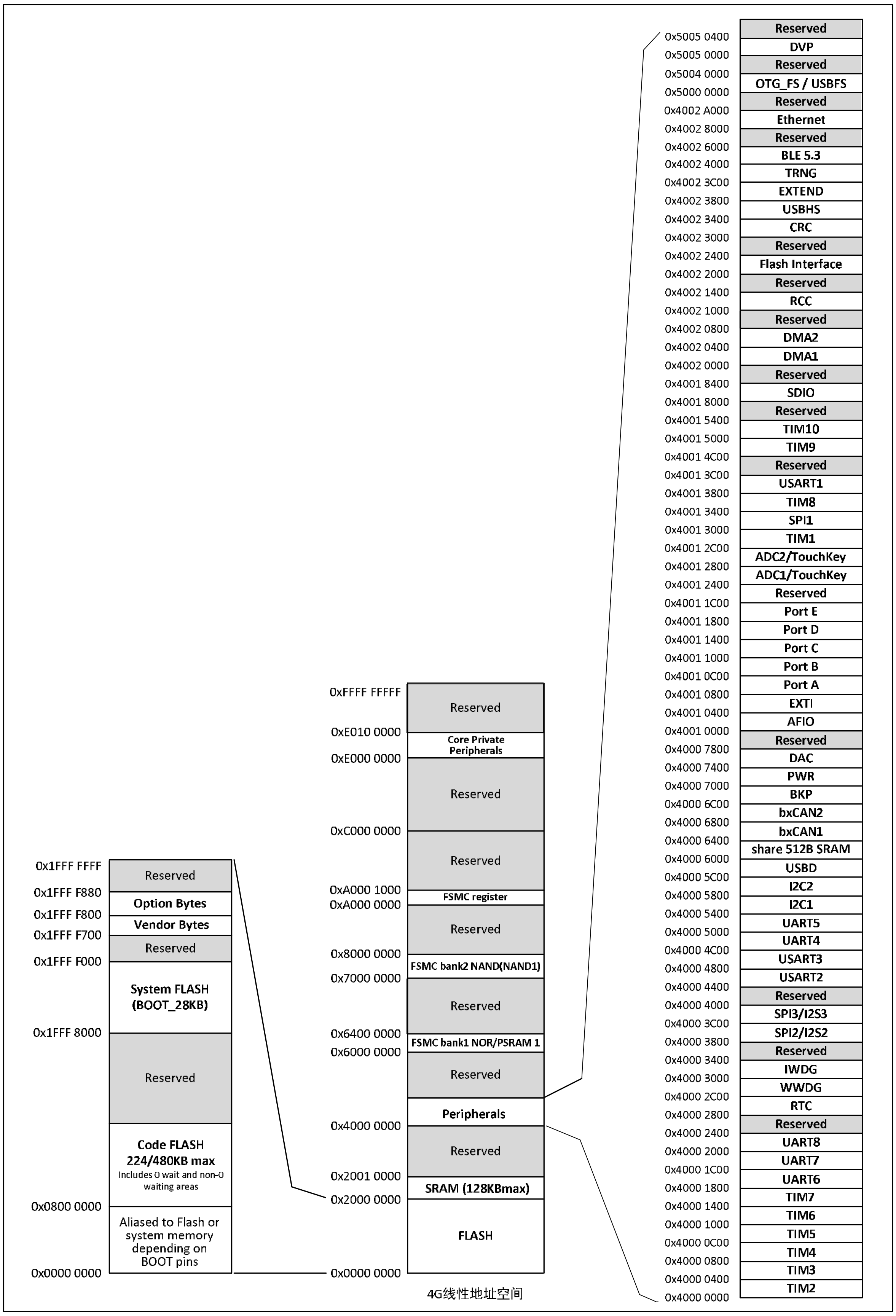


The system is equipped with: Flash access prefetch mechanism to speed up code execution; general DMA controller to reduce CPU burden and improve efficiency; clock tree hierarchical management to reduce the total operating power consumption of peripherals, while being also provided with actions such as data protection mechanism and clock security system protection mechanism to increase system stability.

- The command bus (I-Code) connects the core and the FLASH command interface, and the prefetch is completed on this bus.

- The data bus (D-Code) connects the core and the FLASH data interface for constant load and debug.
- The system bus connects the core and the bus matrix for coordinating the access of the core, DMA, SRAM and peripherals.
- The DMA bus connects the DMA AHB master control interface and the bus matrix, and the bus access objects include FLASH data, SRAM and peripherals.
- The bus matrix is used for the access coordination among the system bus, data bus, DMA bus, SRAM and AHB/APB bridge.
- The AHB/APB bridge provides a synchronous connection for the AHB bus and 2 APB buses. Different peripherals are connected to different APB buses, and different bus clocks can be configured according to actual needs to optimize performance.

Figure 1-3 Storage mapping



1.2 Memory map

The CH32F2x, CH32V2x and CH32V3x all have program memory, data memory, core register, peripheral register, etc., all of which are addressed in a 4GB linear space.

The system memory stores data in little-endian format, i.e., the low bytes are stored in the low address, and the high bytes are stored in the high address.

1.2.1 Bit segment access

Bit manipulation means the operation of reading/writing a bit independently. The CH32F2x provides bit operation read and write to the contents of peripheral register and SRAM area through the mapping processing method. Specific methods:

1) Read the 32-bit data in the mapped address area, the read value is 0 or non-zero, and the target bit value is 0 or 1;
2) Write the 32-bit data in the mapped address area, write 0 or 1, and modify the target bit value to 0 or 1.

Address mapping:
Target bit field: Base address (BEaddr) + offset address (Ofaddr) + bit number (BitN)
Mapping address: Mapaddr

Mapaddr = BEaddr +0x2000000 + (Ofaddr×32) + (BitN×4)

Example 1: Operate the bit3 target bit field in the 0x20000100 address byte of the SRAM area:

Mapaddr= 0x20000000+0x2000000+(0x100*32)+(3*4)= 0x2200200C

Read the 4-byte data content of the 0x2200200C address to know whether bit3 in the 0x20000100 address byte is 0 or 1; write 0 or 1 to the 0x2200200C address, you can modify the bit3 in the 0x20000100 address byte to 0 or 1.

Example 2: Operate bit24 in the 0x40021000 address of the peripheral area:

Mapaddr = 0x20000000+0x2000000+(0x21000*32)+(24*4)= 0x22420060

Read the 4-byte data content of the 0x22420060 address to confirm whether the bit24 in the 0x40021000 peripheral address is 0 or 1; write 0 or 1 to the 0x22420060 address; you can modify the bit24 in the 0x40021000 peripheral address to 0 or 1.

Note: The CH32V2x and CH32V3x do not support bit segment mapping access mode.

1.2.2 Memory allocation

Built-in maximum 128 Kbytes of SRAM, start address 0x20000000, support byte, half word (2 bytes), full word (4 bytes) access.
Built-in maximum 480 Kbytes of Flash program memory, for storing user applications.
Built-in 28K bytes of system memory (bootloader), is selected as the boot space (manufacturer's solidified bootloading program).
Built-in 128-byte used to store the manufacturer's configuration word, which is solidified before delivery out of the factory and cannot be modified by the user.
Built-in 128-byte space is used for user-selected word storage.
Note: Memory allocation varies by model, refer to the corresponding datasheet of the chip for details.

1.3 Boot configuration

The system can select 3 different boot modes through the BOOT0 and BOOT1 pins.

Table 1-1 Boot modes

BOOT0	BOOT1	Boot mode
0	X	Boot from program flash memory
1	0	Boot from system memory
1	1	Boot from internal SRAM

The user selects the startup mode after reset by setting the status value of BOOT pins. After the system is reset or the power is reset, the value of the BOOT pin will be latched again.

The program flash memory, system memory and internal SRAM have different access methods in different startup modes:

- When it is started up from the program flash memory, the program flash memory address will be mapped to the 0x00000000 address area and can also be accessed in the start address area 0x08000000.
- When it is started up from the system memory, the system memory address will be mapped to the address area 0x00000000 and can also be accessed in the original address area 0x1FFFF000.
- When it is started up from the internal SRAM, it can be only accessed from 0x20000000 address area. When the CH32F2x is started up in this area, it is necessary to set the vector table offset register through the NVIC controller to remap the vector table to SRAM. For CH32V2x and CH32V3x, such action is not required.

Chapter 2 Power Control (PWR)

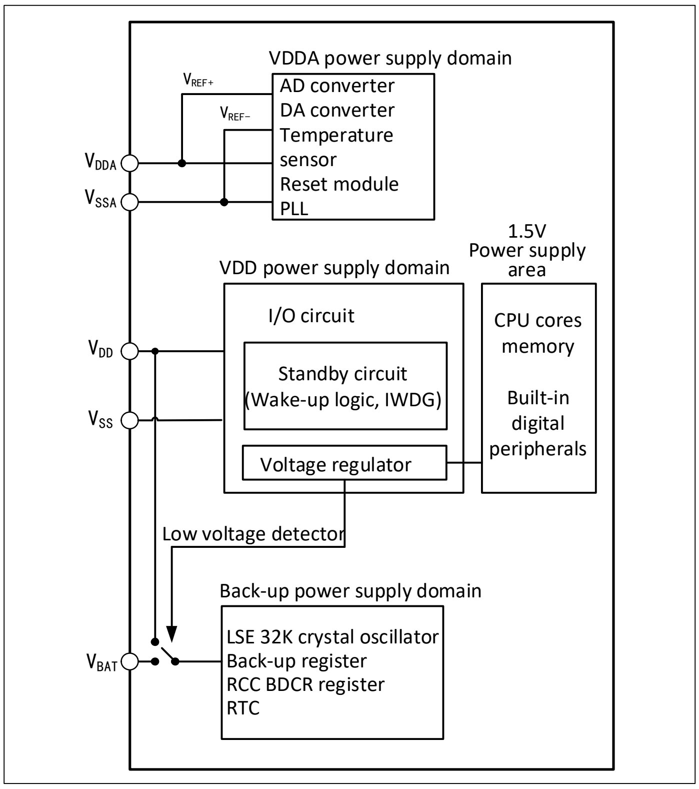
This chapter applies to the whole family of CH32F2x, CH32V2x and CH32V3x.

2.1 Overview

The system operating voltage (V_{DD}) ranges from 2.4 to 3.6V, and the built-in voltage regulator provides the 1.5V power required by the core. When the main power (V_{DD}) is off, backup power supply such as battery can supply power to the real-time clock (RTC) and backup registers through the V_{BAT} pin. If the backup power supply is not required, it is recommended to connect V_{DD} directly to the V_{BAT} pin.

The V_{DDA} and V_{SSA} pins are dedicated to supply power to the analog related circuits in the system, including ADC, DAC, temperature sensors, etc. As reference points of some analog circuits, V_{REF+} and V_{REF-} are equal to V_{DDA} and V_{SSA} inside the chip. In actual applications, V_{DDA} and V_{SSA} must be connected to V_{DD} and V_{SS}, respectively.

Figure 2-1 Power supply overview



After the main power (V_{DD}) is off, the analog switch will be turned to V_{BAT}, and the backup area will be powered by the V_{BAT} pin. At this time, the PC13 to PC15 IOs cannot be used as GPIOs, and only the following functions are available:

- PC13 can be used as TAMPER pin, RTC alarm or second output.
- PC14 and PC15 can be only used as LSE pins.

When the main power (V_{DD}) is stable, the system will automatically switch the backup area powered by V_{DD}, and the PC13 to PC15 IOs can be used as GPIOs.

When the PC13 to PC15 pins are used as GPIO output, the speed must be limited below 2MHz, the maximum load capacitance is 30pF, and it is forbidden to use it in the occasions of continuous output and draw current, such as LED drive.

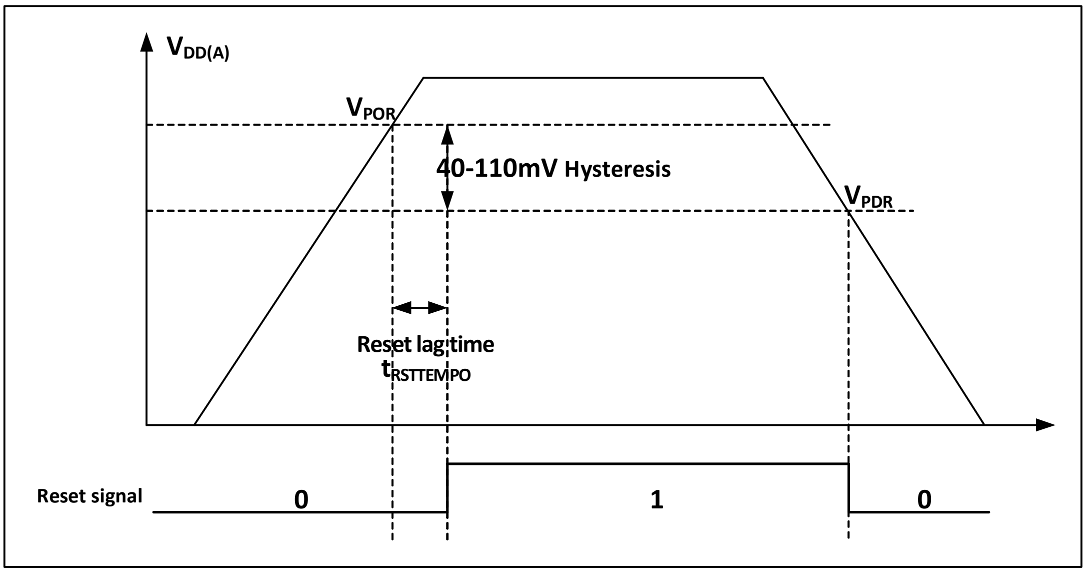
Note: During the restoration of the main power supply (V_{DD}), the internal V_{BAT} power supply is still connected to the external backup power supply through the corresponding V_{BAT} pin. If VDD is less than the reset delay time $t_{RSTTEMPO}$, it will be stabilized and be higher than the value of V_{BAT} by more than 0.6V, and the current may be injected into V_{BAT} through the diode between V_{DD} and V_{BAT} in a very short moment. Then, the backup power supply such as the battery will be injected through the V_{BAT} pin. If the backup power supply cannot withstand such instantaneously injected current, it is then recommended to add a positive on low-dropout diode between the backup power supply and V_{BAT} pin.

2.2 Power supply supervisor

2.2.1 Power-on reset and power-down reset

The power-on reset (POR) and power-down reset (PDR) circuits are integrated inside the system. When V_{DD}/V_{DDA} is below the corresponding threshold, the system will be reset by the relevant circuits, without the need for an external reset circuit. Please refer to the corresponding data sheet for more details concerning the power-on threshold (V_{POR}) and the power-down threshold (V_{PDR}).

Figure 2-2 POR/PDR waveform



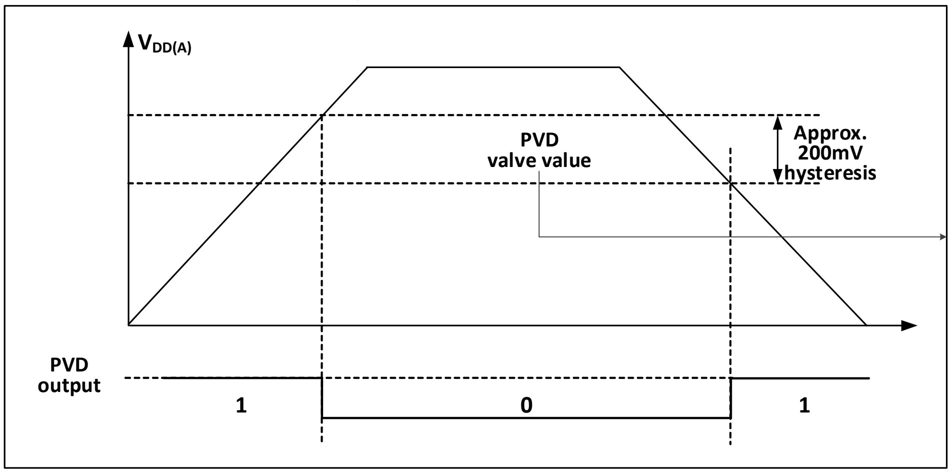
2.2.2 Programmable voltage detector

The programmable voltage detector (PVD) is mainly used to monitor the main power of the system and compares it to the threshold selected by the PLS [2:0] bits in the power control register (PWR_CTLR). Coordinated with the external interrupt register (EXTI) setting, it can generate related interrupt to notify the system in time to perform operations before power failure such as data storage.

Configuration procedure:

1) Set the PLS [2:0] bits in the PWR_CTLR register and select the voltage threshold to be monitored.
2) Optional interrupt processing. The PVD function is internally connected to the line16 of the EXTI module to trigger the setting of rising/falling edges. Enable this interrupt (with EXTI configured). When V_{DD} drops below the PVD threshold or rises above the PVD threshold, a PVD interrupt can be generated.
3) Set the PVDE bit in the PWR_CTLR register to enable PVD.
4) Read the PVD0 bit in the PWR_CSR register to obtain the relationship between the main power supply of the current system and the threshold set by PLS [2:0] bits, and perform the corresponding soft processing.

Figure 2-3 PVD thresholds



2.3 Low-power modes

After the system reset, the microcontroller is at a normal working status (Run mode). At this time, the system power can be saved by reducing the system clock frequency or disabling the peripheral clock or reducing the working peripheral clock. If the system does not need to work, the user can set the system to enter low-power mode, and let the system jump out of this status through specific events.

The microcontroller currently features 3 low-power modes, which are divided into the following modes according to the working differences of processors, peripherals and voltage regulators, etc.:

- Sleep mode: The core stops running, and all peripherals (including the core private peripherals) are still running.
- Stop mode: Stop all clocks, and the system will continue to run after awakening.
- Standby mode: Stop all clocks, and reset the microcontroller after awakening (power reset).

Table 2-1 List of low-power modes

Mode	Entry	Wake-up source	Effect on clock	Voltage regulator
Sleep	WFI	Any interrupt	Core clock OFF, no effect on other clocks	ON
	WFE	Wake-up event		
Stop	Set SLEEPDEEP to 1 Clear PDDS to 0 WFI or WFE	Any external interrupt/event (set in the external interrupt register), rising edge of WKUP pin	HSE, HSI, PLL and peripheral clock OFF	ON: LPDS=0 or Low-power: LPDS=1
Standby	Set SLEEPDEEP to 1 Set PDDS to 1 WFI or WFE	WKUP pin rising edge, RTC alarm event, NRST pin reset, IWDG reset. *Note: Any event can also wake up the system, but the system will not be reset after wake-up.*	HSE, HSI, PLL and peripheral clock OFF	OFF

Note: The SLEEPDEEP bit is the core private peripheral control bit. For CH32F2x, refer to the Cortex-M3 core manual. For CH32V2x and CH32V3x, refer to the PFIC_SCTLR register.

2.3.1 Low-power configuration

- WFI and WFE

WFI: The microcontroller is woken up by an interrupt source that has an interrupt controller response. After

the system wakes up, the interrupt service function is executed firstly (except for the microcontroller reset).
WFE: When a wake-up event triggers the microcontroller, it exits the low-power mode. Wake-up events include:

1) Configure an external or internal EXTI line as event mode. At this time, an interrupt controller does not need to be configured;
2) Or configure an interrupt source, which is equivalent to WFI wake-up, and the system will execute the interrupt service function first;
3) Or configure the SLEEPONPEN bit to enable the peripheral interrupt, but do not enable the interrupt in the interrupt controller, and the interrupt pending bit needs to be cleared after the system is wakened up.

- SLEEPONEXIT

Enable: After executing the WFI or WFE instructions, the microcontroller enters the low-power mode when it ensures that it exits all interrupt services to be processed.
Disable: After executing the WFI or WFE instructions, the microcontroller immediately enters the low-power mode.

- SEVONPEND

Enable: All interrupts or wake-up events can wake it up from the low power mode that is entered by executing WFE.
Disable: Only the interrupt or wake-up event enabled in the interrupt controller can wake it up from the low power mode that is entered by executing WFE.

2.3.2 Sleep mode

In this mode, all I/O pins keep the same state as in the Run mode, and all peripheral clocks are normal, so disable useless peripheral clocks to reduce power consumption before entering sleep mode. This mode offers the lowest wake-up time.

Enter: Configure the core register control bit SLEEPDEEP=0, and power control register PDDS=0. LPDS determines the status of the internal voltage regulator. Execute WFI or WFE, and select SEVONPEND or SLEEPONEXIT.

Exit: Any interrupt or wakeup events.

2.3.3 Stop mode

The stop mode is based on the core deep-sleep mode (SLEEPDEEP) combined with peripheral clock gating. The voltage regulator can be configured in a lower-power mode. In this mode, the high-frequency clocks (HSE/HSI/PLL) in the domain are stopped, SRAM and register contents are preserved, and the I/O pins keep the same state. The system can continue to run after being woken up from this mode, and HSI is the default system clock.

If Flash memory programming or an access to the APB domain is ongoing, the Stop mode entry is delayed until the memory or APB access has completed.

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

Enter: Configure the core register control bit SLEEPDEEP=1, and power control register PDDS=0. LPDS bit is optional. Execute WFI or WFE, and SEVONPEND and SLEEPONEXIT are optional.

Exit: Any external interrupt/event (set in the external interrupt register).

In stop mode, the LPDS bit can be configured. The voltage regulator works in normal mode when LPDS=0, and it is in low-power mode when LPDS=1. In low-power mode, configure RAMLV=1 in the PWR_CTRL register, to enable RAM low voltage mode and achieve the lowest power consumption.

2.3.4 Standby mode

The only difference between the standby mode and the stop mode is that the microcontroller will be reset and a power reset will be performed after exiting under certain specified wake-up conditions.

Modules that can work in standby mode: Independent watchdog (IWDG), real-time clock (RTC), low-frequency clock (LSI/LSE).

Enter: Configure the core register control bit SLEEPDEEP=1, and power control register PDDS=1. Execute WFI or WFE, and SEVONPEND and SLEEPONEXIT are optional.

Exit: 1) Any events (set in the external interrupt register), and the wake-up equivalent stop mode exits.
2) The rising edge of the WKUP pin, the rising edge of the RTC alarm event, the external reset and the IWDG reset on the NRST pin. After this wake-up, the microcontroller will perform a power reset.

In standby mode: when it is supplied normally, configure R2KSTY=1 in the PWR_CTLR register, to keep the 2K-byte RAM on, and configure R30KSTY=1, to keep the 30K-byte RAM on. When it is supplied by VBAT, configure R2KVBAT=1 in the PWR_CTLR register, to keep the 2K-byte RAM on, and configure R32KVBAT=1 to keep the 30K-byte RAM on. On this basis, configure RAMLV=1 in the PWR_CTRL register, to enable RAM low-voltage mode and achieve the lowest power consumption.

Note: In debug mode, if the microprocessor enters the stop or standby mode, the debug connection is lost.
R2KSTY=1: control 2-Kbyte RAM with addresses ranging from 0x20000000 to 0x20000000+2K.
R30KSTY=1: control 30-Kbyte RAM with addresses ranging from 0x20000000+2K to 0x20000000+2K+30K.

2.3.5 RTC auto-wakeup

RTC can be used to automatically wake up the MCU without depending on an external interrupt. By programming the time base, it can wake up from stop or standby mode at regular intervals.

An external low-power 32.768 KHz crystal oscillator (LSE) can be selected as the RTC clock source, or an internal oscillator (LSI) can be selected as the RTC clock source. The accuracy and power consumption indicator of LSI are worse than those of LSE.

The RTC alarm event can wake up the MCU from the stop mode. In order to achieve it, the external interrupt line 17 needs to be configured as a rising edge interrupt, and the RTC can be configured to generate an alarm event. To wake up from the standby mode, it is only needed to configure the RTC to generate an alarm event.

2.4 Register description

Table 2-2 PWR registers

Name	Access address	Description	Reset value
R32_PWR_CTLR	0x40007000	Power control register	0x00000000
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											RAMLV	R30KVBAT	R2KVBAT	R30KSTY	R2KSTY

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							DBP	PLS[2:0]			PVDE	CSBF	CWUF	PDDS	LPDS

Bit	Name	Access	Description	Reset value
[31:21]	Reserved	RO	Reserved	0
20	RAMLV	RW	RAM low voltage mode enable (lower power consumption): 1: Enable. 0: Disable. *Note: Valid when the LPDS bit of PWR_CTLR register is 1.*	0
19	R30KVBAT	RW	30K RAM power enable in Standby mode when V_{BAT} supplies power: 1: Power enabled. 0: Power disabled. *Note: Applied for CH32F20x_D8, CH32F20x_D8C, CH32V30x_D8, CH32V30x_D8C, CH32V20x_D8, CH32V20x_D8W and CH32F20x_D8.*	0
18	R2KVBAT	RW	2K RAM power enable in Standby mode when V_{BAT} supplies power: 1: Power enabled. 0: Power disabled.	0
17	R30KSTY	RW	30K RAM power enable in Standby mode: 1: Power enabled. 0: Power disabled. *Note: Applied for CH32F20x_D8, CH32F20x_D8C, CH32V30x_D8, CH32V30x_D8C, CH32V20x_D8, CH32V20x_D8W and CH32F20x_D8.*	0
16	R2KSTY	RW	2K RAM power enable in Standby mode: 1: Power enabled. 0: Power disabled.	0
[15:9]	Reserved	RO	Reserved	0
8	DBP	RW	Backup domain write enable. When the clock of RTC is 128 frequency division of the external clock, this bit must set to 1. 1: Access to RTC and backup registers enabled; 0: Access to RTC and backup registers disabled.	0
[7:5]	PLS[2:0]	RW	PVD level selection. For details, refer to the electrical characteristics of the manual. 000: 2.37V at the rising edge/2.29V at the falling edge; 001: 2.55V at the rising edge /2.46V at the falling edge; 010: 2.63V at the rising edge /2.55V at the falling edge; 011: 2.76V at the rising edge /2.67V at the falling edge; 100: 2.87V at the rising edge /2.78V at the falling edge;	0

			101: 3.03V at the rising edge /2.93V at the falling edge; 110: 3.18V at the rising edge /3.06V at the falling edge; 111: 3.29V at the rising edge /3.19V at the falling edge.	
4	PVDE	RW	Power voltage detector enable bit 1: Power voltage detector enabled; 0: Power voltage detector disabled.	0
3	CSBF	RW1	Clear standby flag bit. This bit is always read as 0. 1: Clear the SBF Standby Flag bit; 0: No effect.	0
2	CWUF	RW1	Clear wakeup flag bit. This bit is always read as 0. 1: Clear the WUF after 2 system clock cycles; 0: No effect.	0
1	PDDS	RW	Standby/stop mode selection bit when power-down and deepsleep. 1: Enter standby mode. 0: Enter stop mode. The regulator status depends on the LPDS bit;	0
0	LPDS	RW	Voltage regulator working mode selection bit in stop mode. It works when PDDS=0. 1: The voltage regulator works in low-power mode; 0: The voltage regulator works in the normal mode.	0

Note: This register is reset when woken up from Standby mode.

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							EWUP	Reserved					PVDO	SBF	WUF

Bit	Name	Access	Description	Reset value
[31:9]	Reserved	RO	Reserved.	0
8	EWUP	RW	WKUP pin enable bit 1: WKUP is forced in input pull-down configuration, used to wake up the MCU from standby mode; 0: WKUP pin is used for general purpose I/O, and it is not used to wake up the MCU from standby mode.	0
[7:3]	Reserved	RO	Reserved.	0
2	PVDO	RO	PVD output bit. It is valid only when PVD is enabled by the PVDE in the PWR_CTLR register. 1: V_{DD}/V_{DDA} is lower than the PVD threshold set by the PLS[2:0] bits; 0: V_{DD}/V_{DDA} is higher than the PVD threshold set by the PLS[2:0] bits.	0

1	SBF	RO	Standby flag bit, cleared by setting the CSBF bit to 1. 1: MCU enters standby mode; 0: MCU is not in standby mode.	0
0	WUF	RO	Wake-up flag bit, cleared by setting the CWUF bit to 1. 1: A wake-up event or RTC alarm event is detected on the WKUP pin; 0: No wake-up event occurred.	0

Note: This register remains unchanged after woken up from Standby mode.

Chapter 3 Reset and Clock Control (RCC)

This chapter applies to the whole family of CH32F2x, CH32V2x and CH32V3x.

The controller provides different types of reset and a configurable clock tree structure according to the division of power regions and taking into account the peripheral power consumption management in the application. This chapter describes the action scope of each clock in the system.

3.1 Main features

- Several types of reset
- Several clock sources, bus clock management
- Built-in external crystal oscillation monitor and clock security system
- Independent management of each peripheral clock: reset, enable, disable
- Internal clock output

3.2 Reset

There are 3 types of reset: power Reset, system Reset and backup domain Reset.

3.2.1 Power reset

When power reset occurs, all registers are reset except the Backup domain (the Backup domain is powered by V_{BAT}).

A power reset is generated when one of the following events occurs:

- Power-on/power-down reset (POR/PDR)
- Wake up from standby mode

3.2.2 System reset

When a system reset occurs, all registers are reset except the reset flags and backup domain in the control/status register (RCC_RSTSCKR). Identify the source of the reset event by checking the reset status flag bit in the RCC_RSTSCKR register.

A system reset is generated when one of the following events occurs:

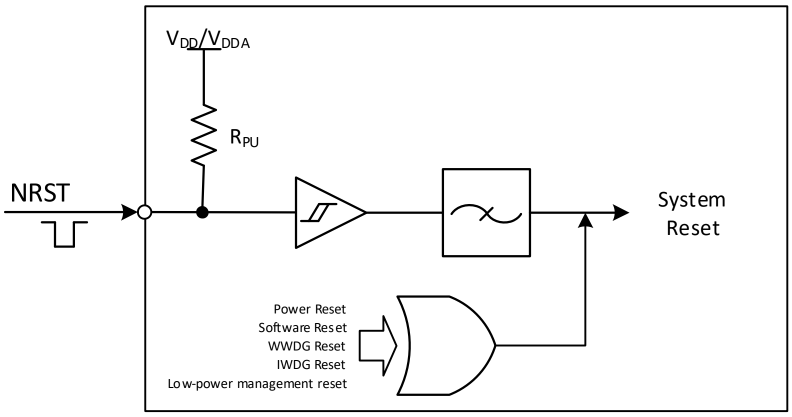
- Low level on NRST pin (external reset)
- Window watchdog end of count condition (WWDG reset)
- Independent watchdog end of count condition (IWDG reset)
- Software reset (SW reset)
- Low-power management reset

Window/independent watchdog reset: Triggered by the counting cycle overflow of the peripheral timer of the window/independent watchdog. For detailed description, please refer to the corresponding chapters.

Software reset: CH32F2x resets the system by setting bit2 in the core register (AIRCR) to 1. For specific operations, please refer to the Cortex-M3 Core Manual. The CH32V2x and CH32V3x reset the system by setting the SYSRESET bit in the interrupt configuration register (PFIC_CFGR) to 1, or by setting the SYSRESET bit in the PFIC_SCTLR register to 1. Refer to the corresponding chapters for details.

Low-power management reset: By resetting the nRST_STDBY bit in the user option bytes, the standby mode reset is enabled. After the process of entering the standby mode is executed at this time, a system reset will be performed instead of entering standby mode. By setting the nRST_STOP bit in User Option Bytes to 0, the stop mode reset is enabled. After the process of entering the stop mode is executed, a system reset will be executed instead of entering stop mode.

Figure 3-1System reset structure



3.2.3 Backup domain reset

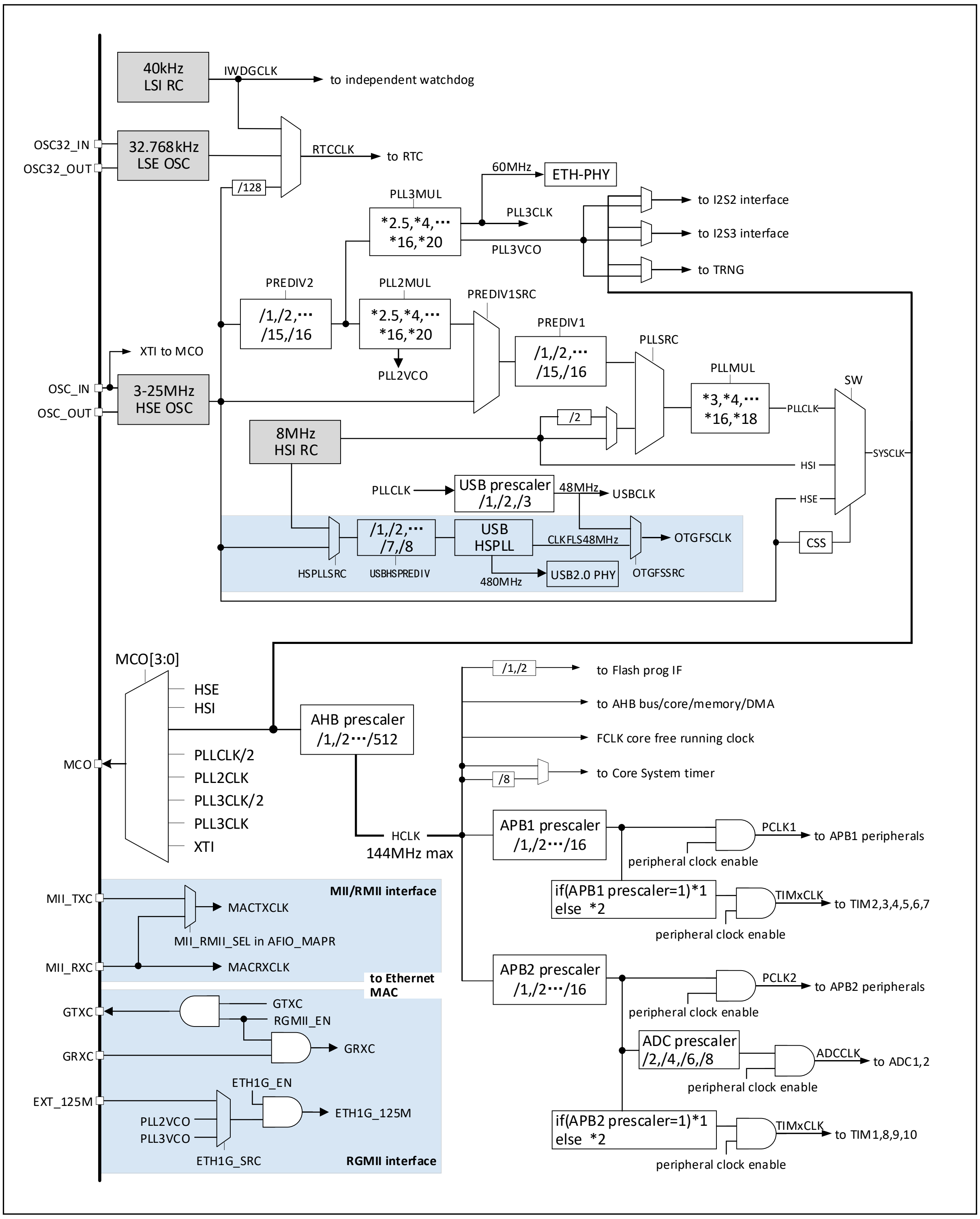
When the backup domain reset occurs, only the backup domain register will be reset, including the backup register, RCC_BDCTLR register (RTC enable and LSE oscillator). A backup domain reset is generated when one of the following events occurs:

- On the premise that both V_{DD} and V_{BAT} are powered off, it is caused by power-on of V_{DD} or V_{BAT}
- Set the BDRST bit in the RCC_BDCTLR register to 1
- Set the BKPRST bit in the RCC_APB1PRSTR register to 1

3.3 Clock

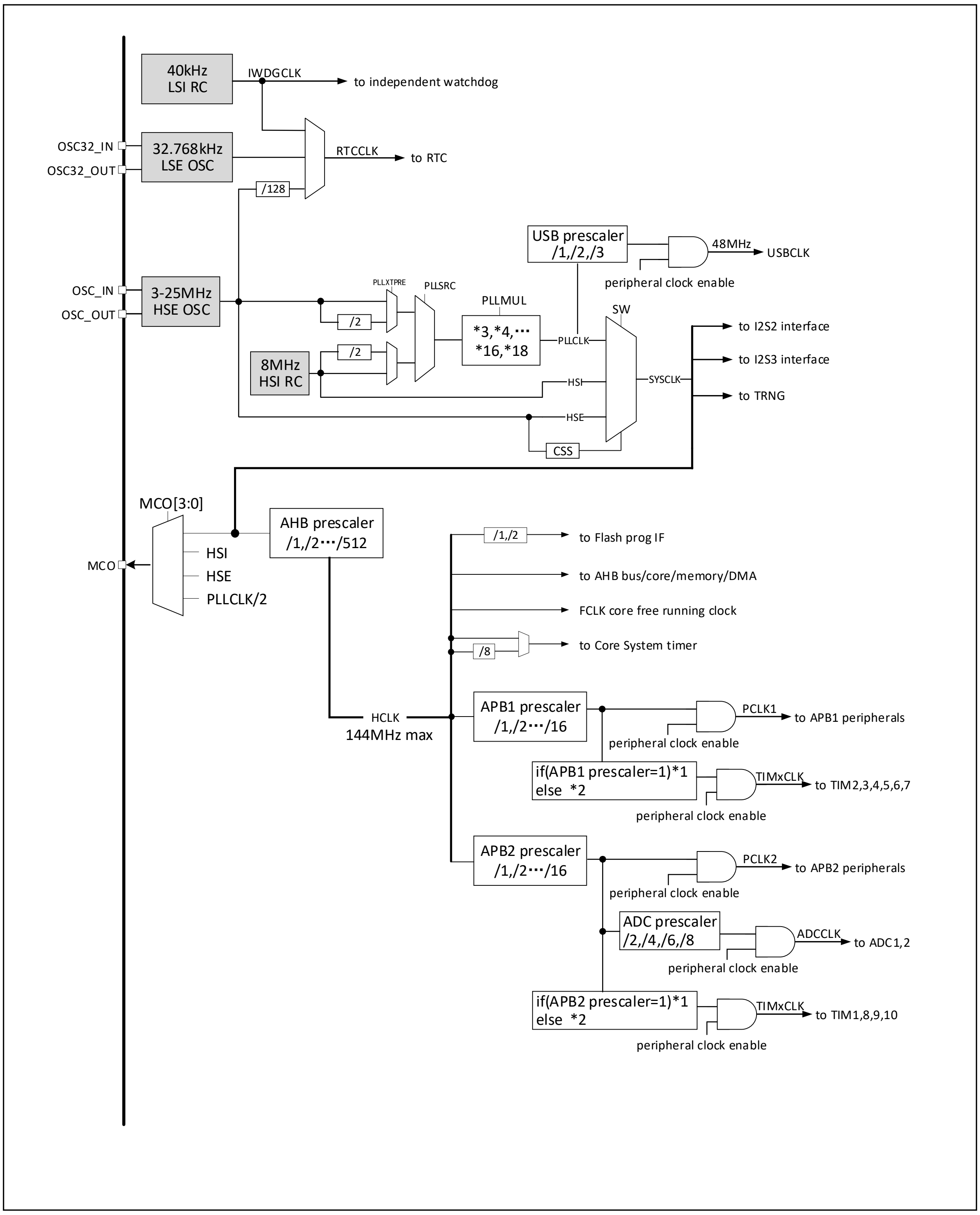
3.3.1 System clock structure

Figure 3-2 CH32V305/307 and CH32F205/207 clock tree structure



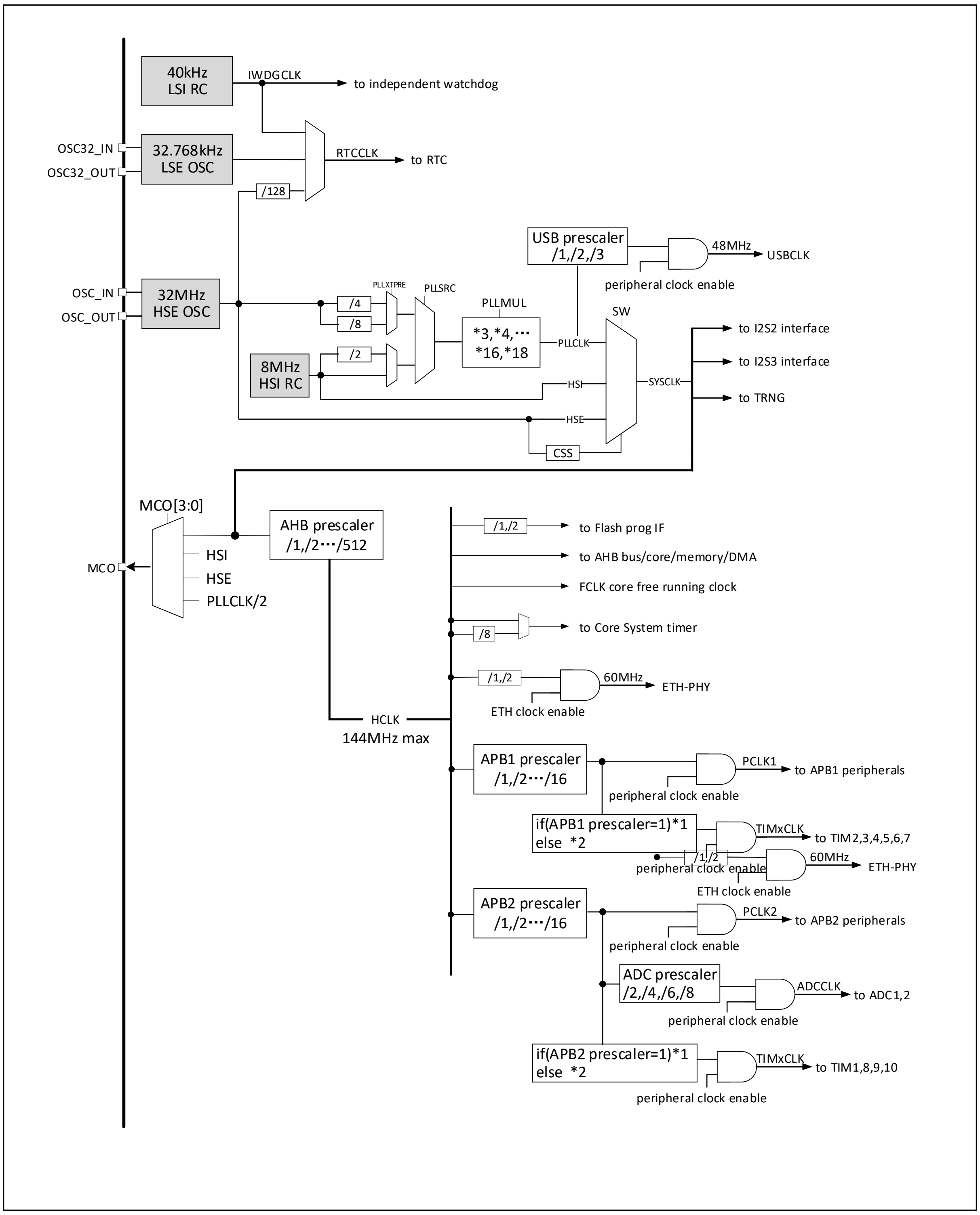
Note: This clock tree structure is applied for CH32F20x_D8C and CH32V30x_D8C. When the USB interface is used, the frequency of CPU must be 48MHz or 96MHz or 144MHz. When USBHD high speed function is used, the clock source of USBHSPLL can only be HSE. When system wakes up from sleep state, HIS is automatically switched as system clock frequency.

Figure 3-3 CH32FV203/V303 clock tree structure



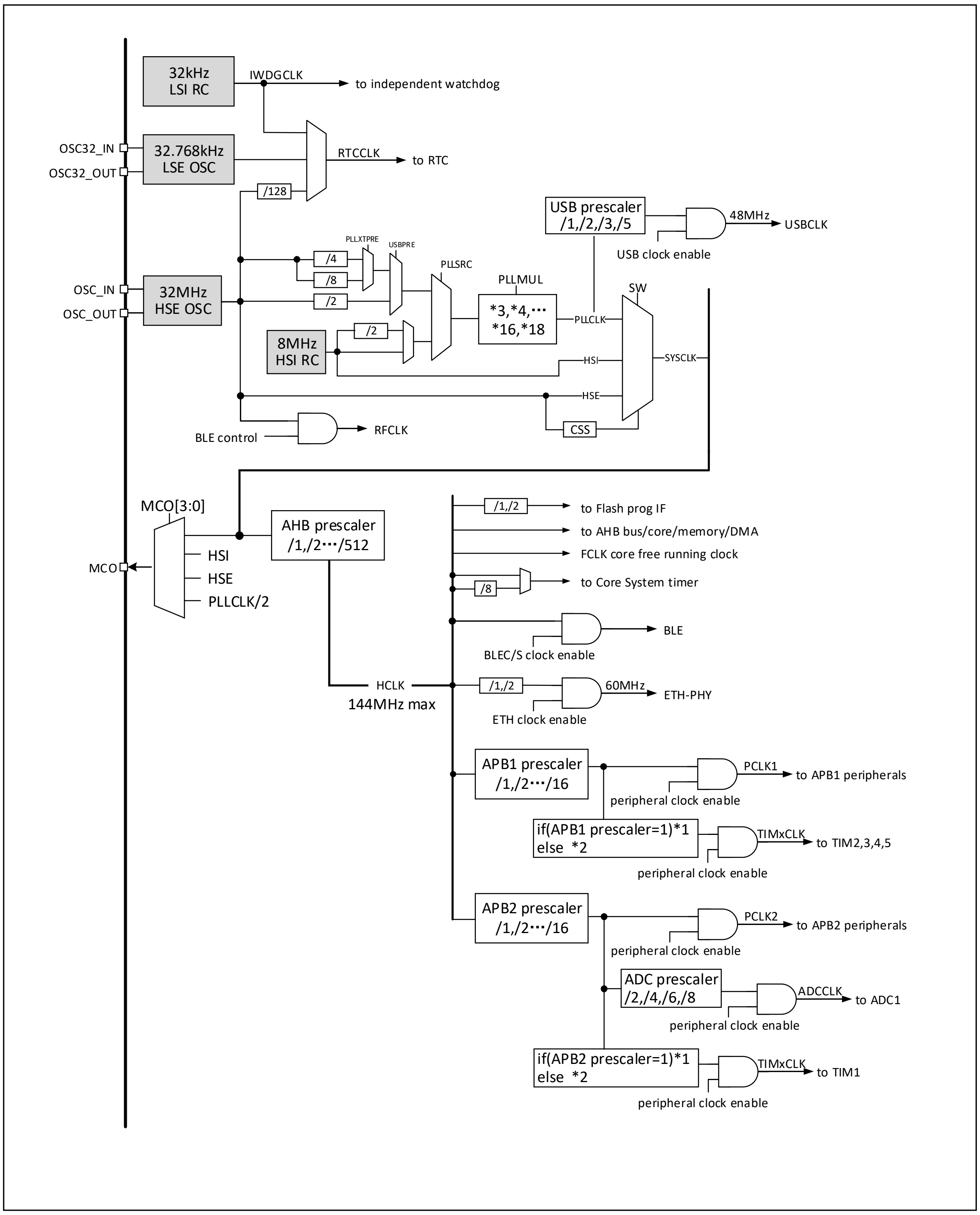
Note: This clock tree structure is applied for CH32F20x_D6, CH32F20x_D8, CH32V20x_D6 and CH32V30x_D8.

Figure 3-4 CH32V203RB clock tree structure



Note: For CH32V203RB, the external crystal or clock (HSE) is 32M. When the external crystal is enabled, no load capacitor is required as it is built in.

Figure 3-5 CH32FV208 clock tree structure



Note: This clock tree structure is applied for CH32F20x_D8W, CH32V20x_D8 and CH32V20x_D8W. When USB and ETH are used simultaneously, set the USBPRE[1:0] bits to 11b. The external crystal or clock (HSE) is 32M. When the external crystal is enabled, no load capacitor is required as it is built in.

3.3.2 High-speed clock (HSI/HSE)

HSI is a high-speed clock signal generated by an 8MHz RC oscillator in the system. The HSI RC oscillator can provide the system clock without depending on any external device. Its startup time is very short but the clock frequency accuracy is poor. HSI is enabled/disabled by setting the HSION bit in the RCC_CTLR register.

The HSIRDY bit indicates whether the HSI RC oscillator is stable. By default, HSION and HSIRDY are set to 1 (recommended not to disable). If the HSIRDYIE bit in the RCC_INTR register is set, a corresponding interrupt is generated.

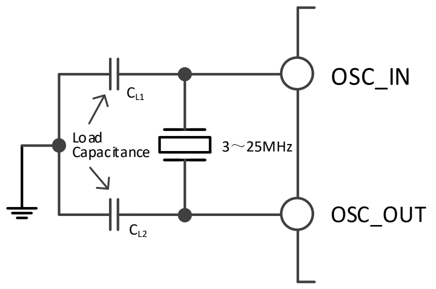
- Factory calibration: The difference in manufacturing process may cause the RC oscillation frequency of each chip to be different, so HSI calibration is performed for each chip before the chip is delivered out of the factory. After the system is reset, the factory calibration value is loaded into the HSICAL[7:0] bits in the RCC_CTLR register.
- User adjustment: Based on different voltages or ambient temperatures, the application program can adjust the HSI frequency through the HSITRIM[4:0] bits in the RCC_CTLR register.

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

HSE is a High-Speed External clock signal, including external crystal/ceramic resonator generation or external high-speed clock input.

- External crystal/ceramic resonator (HSE crystal): An external 3 to 25MHz oscillator provides a more accurate clock source for the system. For details, please refer to the electrical characteristics of this manual. The HSE crystal can be enabled/disabled by setting the HSEON bit in the RCC_CTLR register. The HSERDY bit indicates whether the HSE crystal oscillation is stable. The clock is not released until the HSERDY bit is set to 1 by hardware. If the HSERDYIE bit in the RCC_INTR register is set, a corresponding interrupt can be generated.

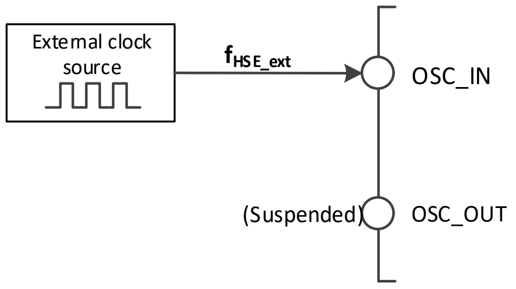
Figure 3-6 HSE crystal circuit



Note: The load capacitor should be placed as close as possible to the oscillator pins, and the loading capacitance values must be adjusted according to the selected oscillator.

- External high-speed clock source (HSE bypass): In this mode, an external clock source is directly provided to the OSC_IN pin, and the OSC_OUT pin is suspended. It can have a frequency of up to 25MHz. The application program needs to set the HSEBYP bit when the HSEON bit is 0, to enable HSE bypass, and then set the HSEON bit.

Figure 3-7 High-speed clock source circuit



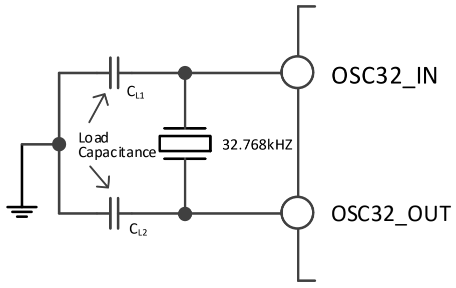
3.3.3 Low-speed clock (LSI/LSE)

LSI is a low-speed clock signal generated by a RC oscillator of approximately 40 KHz in the system. It can keep running in stop and standby modes, and provide a clock reference for the RTC clock, independent watchdog, and wake-up unit. For further information, please refer to the electrical characteristics of this manual. LSI can be enabled/disabled by setting the LSION bit in the RCC_RSTSCKR register, and then it checks whether the LSI RC oscillation is stable by querying the LSIRDY bit. The clock is not released until the LSIRDY bit is set to 1 by hardware. If the LSIRDYIE bit in the RCC_INTR register is set, a corresponding interrupt can be generated.

LSE is an external low-speed clock signal, including external crystal/ceramic resonator generation or external low-speed clock input. It provides a low-power but highly accurate clock source for RTC clock or other timing functions.

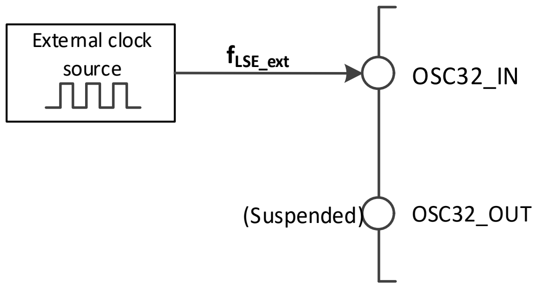
- External crystal/ceramic resonator (LSE crystal): External 32.768 KHz low-speed oscillator. LSE can be enabled/disabled by setting the LSEON bit in the RCC_BDCTLR register. The LSERDY bit indicates whether the LSE crystal oscillation is stable. The clock is not released until the LSERDY bit is set to 1 by hardware. If the LSERDYIE bit in the RCC_INTR register is set, a corresponding interrupt can be generated.

Figure 3-8 LSE crystal circuit



- External low-speed clock source (LSE bypass): In this mode, an external clock source is directly provided to the OSC32_IN pin, and the OSC32_OUT pin is suspended. The application program needs to set the LSEBYP bit when the LSEON bit is 0, to enable the LSE bypass, and then set the LSEON bit.

Figure 3-9 Low-speed clock source circuit



3.3.4 PLL clock

By configuring the RCC_CFGR0 register and the EXTEND_CTR register, 3 clock sources and multiplication factors can be selected for the internal PLL clock. These settings must be completed before each PLL is enabled. Once the PLL is enabled, these parameters cannot be changed. PLL can be enabled/disabled by setting the PLLON bit in the RCC_CTLR register. The PLLRDY bit indicates whether the PLL clock is stable. The clock is not released until the PLLRDY bit is set to 1 by hardware. PLL2 can be enabled/disabled by setting the PLLON2 bit in the RCC_CTLR register. The PLLRDY2 bit indicates whether the PLL2 clock is stable. The clock is not released until the PLLRDY2 bit is set to 1 by hardware. PLL3 can be enabled/disabled by setting the PLLON3 bit in the RCC_CTLR register. The PLLRDY3 bit indicates whether the PLL3 clock is stable. The clock is not released until the PLLRDY3 bit is set to 1 by hardware.If the PLLRDYIE bit, the PLL2RDYIE bit and the PLL3RDYIE bit in the RCC_INTR register are set, a corresponding interrupt can be generated.

PLL clock sources:

- HSI clock
- HSI clock divided by 2
- HSE clock or PLL2 clock through a configurable divider
  PLL2 and PLL3 are clocked by HSE through a configurable divider (PREDIV2)

3.3.5 Bus/Peripheral clock

3.3.5.1 System clock (SYSCLK)

Configure the system clock source by configuring the SW[1:0] bits in the RCC_CFGR0 register. The SWS[1:0] bits indicate the current system clock source status.

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

After the controller reset, HSI clock is selected as the system clock source by default. A switch from one clock source to another occurs only if the target clock source is ready.

3.3.5.2 AHB/APB1/APB2 bus peripheral clock (HCLK/PCLK1/PCLK2)

By configuring the HPRE[3:0], PPRE1[2:0], and PPRE2[2:0] bits in the RCC_CFGR0 register, the clocks of the AHB, APB1, and APB2 buses can be configured respectively. These bus clocks determine the access clock reference of the peripheral interfaces mounted below them. The application program can adjust different values to reduce the power consumption of some peripherals.

Different peripheral modules can be reset by bits in the RCC_AHBRSTR, RCC_APB1PRSTR, and RCC_APB2PRSTR registers to restore them to the initial state.

By setting bits in the RCC_AHBPCENR, RCC_APB1PCENR and RCC_APB2PCENR registers, the communication clock interface of different peripheral modules can be enabled or disabled separately. To use a peripheral, firstly enable the corresponding clock bit to access its register.

3.3.5.3 RTC clock (RTCCLK)

By setting the RTCSEL[1:0] bits in the RCC_BDCTLR register, the RTCCLK clock source can either be HSE/128, LSE or LSI clocks. Before modifying this bit, ensure that the DBP bit in the power control register (PWR_CR) is set to 1, and this bit can be reset only when the backup domain is reset.

- LSE as the RTC clock: As LSE is at the backup domain and is powered by the V_{BAT} supply, as long as V_{BAT} maintains supplying power, RTC will continuously work even though V_{DD} power supply is switched off.
- LSI as the RTC clock: If the V_{DD} power supply is switched off, the Auto-wakeup unit(AWU)clock cannot be guaranteed.
- HSE/128 as the RTC clock: If the V_{DD} supply is powered off or if the internal voltage regulator is powered off (removing power from the 1.8V domain), the RTC state is not guaranteed.

3.3.5.4 Independent watchdog clock

If the independent watchdog(IWDG) is started by either hardware option or software access, the LSI oscillator is forced ON and cannot be switched off. After the LSI oscillator temporization, the clock is provided to IWDG.

3.3.5.5 Microcontroller clock output (MCO)

The microcontroller clock output (MCO) capability allows the clock to output onto the MCO pin. Configure the alternate push-pull output mode in the corresponding GPIO port register. By setting the MCO[3:0] bits in the RCC_CFGR0 register, the following 8 clock signals can be selected as the MCO clock output:

- System clock (SYSCLK)
- HSI clock
- HSE clock
- PLL clock divided by 2
- PLL2 clock
- PLL3 clock
- PLL3 clock divided by 2
- XT1 external 3-25MHz oscillator (for Ethernet)

3.3.5.6 USB clock

USBD 48MHz clock source is a PLL clock through a configurable divider, and the PLL supports 3 types of clock configurations, including 48MHz, 96MHz and 144MHz. By configuring the USBPRE[1:0] bits in the RCC_CFGR0 register, the 48MHz clock is output to USBD.

USBOTG 48MHz clock source is a PLL clock or USBHSPLL clock through a configurable divider, and it can be selected by configuring the USB_CLK_SRC bit in the RCC_CFGR2 register. If a PLL clock through a configurable divider is selected as the clock source, refer to USBD for configuration steps. If USBHSPLL clock is selected as the clock source, USBHS PLL reference clock frequency can be selected by configuring the USBHS_CKREFSEL[1:0] bits in the RCC_CFGR2 register (reference clock frequency must be consistent with USBHS PLL input clock).

USBHD clock source is originated from USBHSPLL clock. USBHS PLL reference clock frequency can be selected by configuring the USBHS_CKREFSEL[1:0] bits in the RCC_CFGR2 register (reference clock frequency must be consistent with USBHS PLL input clock). By setting the USBHS_PLLALIVE bit in the RCC_CFGR2 register, USB PHY internal PLL can be enabled.

3.3.5.7 ETH clock

For details, please refer to Section 27.1.4.5.

3.3.5.8 I2S and RNG clock

I2S and RNG clock source are originated from PLL3VCO or system clock (SYSCLK). By configuring the I2S2SRC/I2S3SRC/RNG_SRC bit in the RCC_CFGR2 register, I2S2/I2S3/RNG clock source can be selected respectively.

3.3.6 Clock security system

The clock security system is an operation protection mechanism of the controller. It can switch to HSI clock when a failure is detected on the HSE clock, generate interrupt notification and allow the application software to perform rescue operations.

By setting the CSSON bit in the RCC_CTLR register to 1, the clock security system can be activated. In this case, the clock detector is enabled after the HSE oscillator startup delay (HSERDY=1), and disabled after the HSE oscillator is stopped. Once a failure is detected on the HSE clock during system operation, the HSE oscillator will be disabled, and the clock failure event will be sent to the break input of the advanced-control timers (TIM1 and TIM8). Then a clock security interrupt is generated, the CSSF bit is set to 1, and the application program enters the non-maskable interrupt (NMI). The CSSF bit can be cleared by setting the CSSC bit, and the NMI interrupt pending bit can be cancelled.

If the current HSE is selected as the system clock, or the current HSE is selected as the PLL input clock and the PLL is selected as the system clock, the clock security system will automatically switch the system clock to the HSI oscillator when the failure occurs, and disable the HSE oscillator and PLL.

3.4 Register description

Table 3-1 RCC registers

Name	Access address	Description	Reset value
R32_RCC_CTLR	0x40021000	Clock control register	0x0000xx83
R32_RCC_CFGR0	0x40021004	Clock configuration register 0	0x00000000
R32_RCC_INTR	0x40021008	Clock interrupt register	0x00000000
R32_RCC_APB2PRSTR	0x4002100C	APB2 peripheral reset register	0x00000000
R32_RCC_APB1PRSTR	0x40021010	APB1 peripheral reset register	0x00000000
R32_RCC_AHBPCENR	0x40021014	AHB peripheral clock enable register	0x00000014
R32_RCC_APB2PCENR	0x40021018	APB2 peripheral clock enable register	0x00000000
R32_RCC_APB1PCENR	0x4002101C	APB1 peripheral clock enable register	0x00000000
R32_RCC_BDCTLR	0x40021020	Backup domain control register	0x00000000
R32_RCC_RSTSCKR	0x40021024	Control/status register	0x0C000000
R32_RCC_AHBRSTR	0x40021028	AHB peripheral reset register	0x00000000
R32_RCC_CFGR2	0x4002102C	Clock configuration register 2	0x00000000

Table 3-2 OSC registers

Name	Access address	Description	Reset value
R32_HSE_CAL_CTRL	0x4002202C	HSE crystal oscillator calibration control register	0x09000000
R16_LSI32K_TUNE	0x40022036	LSI crystal oscillator calibration tune register	0x1011
R8_LSI32K_CAL_CFG	0x40022049	LSI crystal oscillator calibration configuration register	0x01
R16_LSI32K_CAL_STATR	0x4002204C	LSI crystal oscillator calibration status register	0x0000
R8_LSI32K_CAL_OV_CNT	0x4002204E	LSI crystal oscillator calibration counter	0x00
R8_LSI32K_CAL_CTRL	0x4002204F	LSI crystal oscillator calibration control register	0x80

Note: Applied for CH32V20x_D8W and CH32F20x_D8W.

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
Reserved		PLL3 RDY	PLL3 ON	PLL2 RDY	PLL2 ON	PLL RDY	PLL ON	Reserved				CSSO N	HSE BYP	HSE RDY	HSE ON

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSICAL[7:0]								HSITRIM[4:0]					Reser ved	HSI RDY	HSIO N

Bit	Name	Access	Description	Reset value
[31:30]	Reserved	RO	Reserved	0
29	PLL3RDY	RO	PLL3 clock ready lock flag bit (set by hardware): 1: PLL3 locked; 0: PLL3 unlocked. *Note: Applied for CH32F20x_D8C and CH32V30x_D8C.*	0
28	PLL3ON	RW	PLL3 clock enable control bit: 1: PLL3 enabled; 0: PLL3 disabled. *Note: This bit is cleared by hardware when entering stop or standby low-power mode. Applied for CH32F20x_D8C and CH32V30x_D8C.*	0
27	PLL2RDY	RO	PLL2 clock ready lock flag bit (set by hardware): 1: PLL locked; 0: PLL unlocked. *Note: Applied for CH32F20x_D8C and CH32V30x_D8C.*	0
26	PLL2ON	RW	PLL2 clock enable control bit: 1: PLL enabled; 0: PLL disabled. *Note: This bit is cleared by hardware when entering stop or standby low-power mode. Applied for CH32F20x_D8C and CH32V30x_D8C.*	0
25	PLLRDY	RO	PLL clock ready lock flag bit (set by hardware): 1: PLL locked; 0: PLL unlocked.	0
24	PLLON	RW	PLL clock enable control bit: 1: PLL enabled; 0: PLL disabled. *Note: This bit is cleared by hardware when entering stop or standby low-power mode.*	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 HSE clock detector is enabled by hardware. When a HSE clock failure is detected, the	0

			CSSF flag and NMI interrupt is enabled. When the HSE is not ready, the HSE clock detector is disabled by hardware. 0: Disable clock security system.	
18	HSEBYP	RW	HSE clock bypass control bit: 1: Bypass HSE oscillator (using external clock source); 0: Not bypass HSE oscillator. *Note: This bit can be written only when HSEON is 0.*	0
17	HSERDY	RO	HSE clock ready flag bit (set by hardware): 1: HSE oscillator ready; 0: HSE oscillator not ready. *Note: After the HSEON bit is cleared, it takes 6 cycles of the HSE oscillator clock to clear this bit.*	0
16	HSEON	RW	HSE clock enable control bit: 1: Enable HSE oscillator; 0: Disable HSE oscillator. *Note: After entering the stop or standby low-power mode, this bit is cleared by hardware.*	0
[15:8]	HSICAL[7:0]	RO	HSI clock calibration, automatically initialized at startup.	xxh
[7:3]	HSITRIM[4:0]	RW	HSI clock trimming: The user can input a trimming value that is added to the HSICAL[7:0] bits, and adjust the frequency of the internal HSI RC oscillator according to variations in voltage and temperature. The default value is 16, HSI can be trimmed to 8MHz±1%. The trimming step is around 40KHz between 2 consecutive HSICAL steps.	10000
2	Reserved	RO	Reserved	0

1	HSIRDY	RO	HSI clock (8MHz) ready flag bit (set by hardware):
			1: HIS oscillator (8MHz) ready;
			0: HIS oscillator (8MHz) not ready.
			Note: After the HSION bit is cleared, it takes 6 cycles of the HSI oscillator clock to clear this bit.
				1

0	HSION	RW	HSI clock (8MHz) enable control bit: 1: Enable HSI oscillator; 0: Disable HSI oscillator. *Note: When the system returns from standby and stop modes or when the external oscillator HSE selected as the system clock fails, this bit will be set by hardware to start the internal 8MHz RC oscillator.*	1

3.4.2 Clock Configuration Register0 (RCC_CFGR0)

Offset address: 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADCDUTY	Reserved		ETHPRE	MCO[3:0]				USBPRE[1:0]		PLLMUL[3:0]				PLLXTPRE	PLLSRC

15 14	13 12 11	10 9 8	7 6 5 4	3 2	1 0
ADCPRE[1:0]	PPRE2[2:0]	PPRE1[2:0]	HPRE[3:0]	SWS[1:0]	SW[1:0]

Bit	Name	Access	Description	Reset value
31	ADCDUTY	RW	ADC clock duty cycle adjustment: 1: ADC clock at low level for longer; 0: ADC clock duty cycle is 50%.	0
[30:29]	Reserved	RO	Reserved	0
28	ETHPRE	RW	Ethernet clock source prescaler control: 0: Not divided; 1: Divided by 2. *Note: Applied for CH32V20x_D8W, CH32V20x_D8 and CH32F20x_D8W.*	

[27:24]	MCO[3:0]	RW	Microcontroller clock output (MCO) control:
			00xx: No clock;
			0100: System clock (SYSCLK);
			0101: Internal 8MHz RC oscillator clock (HSI);
			0110: External oscillator clock (HSE);
			0111: PLL clock divided by 2;
			1000: PLL2 clock;
			1001: PLL3 clock divided by 2;
			1010: XT1 external oscillator clock;
			1011: PLL3 clock.
			Note: This clock output may have some truncated cycles at startup or during MCO clock source switching. The values from 1000 to 1011 are applied for CH32F20x_D8C and CH32V30x_D8C.
				0

[23:22]	USBPRE[1:0]	RW	USBFS/USBOTG clock prescaler configuration:
			00: Not divided (when PLLCLK=48MHz);
			01: Divided by 2 (when PLLCLK=96MHz);
			10: Divided by 3 (when PLLCLK=144MHz);
			11: Divided by 5, and PLL source is HSE divided by 2 (PLLCLK=240MHz, only applied for CH32V20x_D8W/ CH32F20x_D8W).
			Note: The value 11b is only available to CH32V20x_D8W/CH32F20x_D8W, while reserved for other models.
				0

[21:18]	PLLMUL[3:0]	RW	PLL multiplication factor (these bits can be written only when PLL is disabled):
			For CH32V20x_D8/CH32V20x_D8W/CH32F20x_D8W/ CH32V30x_D8/CH32F20x_D8/CH32V20x_D8/ CH32V20x_D8W/CH32F20x_D8W-
			0000: PLL input clock x 2; 0001: PLL input clock x 3;
			0010: PLL input clock x 4; 0011: PLL input clock x 5;
				0

			0100: PLL input clock x 6; 0101: PLL input clock x 7;
			0110: PLL input clock x 8; 0111: PLL input clock x 9;
			1000: PLL input clock x 10; 1001: PLL input clock x 11;
			1010: PLL input clock x 12; 1011: PLL input clock x 13;
			1100: PLL input clock x 14; 1101: PLL input clock x 15;
			1110: PLL input clock x 16; 1111: PLL input clock x 18;
			For *CH32F20x_D8C/CH32V30x_D8C*-
			0000: PLL input clock x 18; 0001: PLL input clock x 3;
			0010: PLL input clock x 4; 0011: PLL input clock x 5;
			0100: PLL input clock x 6; 0101: PLL input clock x 7;
			0110: PLL input clock x 8; 0111: PLL input clock x 9;
			1000: PLL input clock x 10; 1001: PLL input clock x 11;
			1010: PLL input clock x 12; 1011: PLL input clock x 13;
			1100: PLL input clock x 14; 1101: PLL input clock x 6.5;
			1110: PLL input clock x 15; 1111: PLL input clock x 16.	

17	PLLXTPRE	RW	HSE divider for PLL entry (it can be written only when PLL is disabled) For CH32F20x_D6, CH32F20x_D8, CH32F20x_D8C, CH32V20x_D6, CH32V30x_D8, CH32V30x_D8C. 1: HSE clock divided by 2; 0: HSE clock not divided. For CH32F20x_D8W, CH32V20x_D8, CH32V20x_D8W. 1: HSE clock divided by 8; 0: HSE clock divided by 4;	0
16	PLLSRC	RW	PLL entry clock source (it can be written only when the PLL is disabled): 1: HSE clock not divided or divided by 2; 0: HSI clock not divided or divided by 2.	0

[15:14]	ADCPRE[1:0]	RW	ADC clock source prescaler control:
			00: PCLK2 divided by 2;
			01: PCLK2 divided by 4;
			10: PCLK2 divided by 6;
			11: PCLK2 divided by 8;
			Note: ADC clock shall not exceed 14MHz at most.
				0

[13:11]	PPRE2[2:0]	RW	APB2 clock source prescaler control:
			0xx: HCLK not divided;
			100: HCLK divided by 2;
			101: HCLK divided by 4;
			110: HCLK divided by 8;
			111: HCLK divided by 16.
				0

[10:8]	PPRE1[2:0]	RW	APB1 clock source prescaler control:
			0xx: HCLK not divided;
			100: HCLK divided by 2;
			101: HCLK divided by 4;
			110: HCLK divided by 8;
			111: HCLK divided by 16.
				0

[7:4]	HPRE[3:0]	RW	AHB clock source prescaler control:
			0xxx: SYSCLK not divided;
			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.
			Note: When the prescale factor of the AHB clock source is greater than 1, the prefetch buffer must be switched on.
				0

[3:2]	SWS[1:0]	RO	System clock (SYSCLK) switch status (set by hardware): 00: HSI oscillator used as system clock; 01: HSE oscillator used as system clock; 10: PLL oscillator used as system clock; 11: Not applicable.	0
[1:0]	SW[1:0]	RW	System clock source switch: 00: HSI oscillator used as system clock; 01: HSE oscillator used as system clock; 10: PLL oscillator used as system clock; 11: Not applicable. *Note: When the system returns from standby and stop modes or when the external oscillator HSE used as the system clock fails after the clock security system is enabled (CSSON=1), 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	PLL3 RDYC	PLL2 RDYC	PLL RDYC	HSE RDYC	HSI RDYC	LSE RDYC	LSI RDYC

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	PLL3 RDYIE	PLL2 RDYIE	PLL RDYIE	HSE RDYIE	HSI RDYIE	LSE RDYIE	LSI RDYIE	CSSF	PLL3 RDYF	PLL2 RDYF	PLL RDYF	HSE RDYF	HSI RDYF	LSE RDYF	LSI RDYF

Bit	Name	Access	Description	Reset value
[31:24]	Reserved	RO	Reserved	0
23	CSSC	WO	Clock security system interrupt clear flag bit (CSSF): 1: Clear CSSF interrupt flag; 0: No effect.	0
22	PLL3RDYC	WO	PLL3 ready interrupt clear flag bit:	0

			1: Clear PLL3RDYF interrupt flag; 0: No effect. *Note: Applied for CH32F20x_D8C and CH32V30x_D8C.*	
21	PLL2RDYC	WO	PLL2 ready interrupt clear flag bit: 1: Clear PLL2RDYF interrupt flag; 0: No effect. *Note: Applied for CH32F20x_D8C and CH32V30x_D8C.*	0
20	PLLRDYC	WO	PLL ready interrupt clear flag bit: 1: Clear the PLLRDYF interrupt flag; 0: No effect.	0
19	HSERDYC	WO	HSE oscillator ready interrupt clear flag bit: 1: Clear HSERDYF interrupt flag; 0: No effect.	0
18	HSIRDYC	WO	HSI oscillator ready interrupt clear flag bit: 1: Clear HSIRDYF interrupt flag; 0: No effect.	0
17	LSERDYC	WO	LSE oscillator ready interrupt clear flag bit: 1: Clear LSERDYF interrupt flag; 0: No effect.	0
16	LSIRDYC	WO	LSI oscillator ready interrupt clear flag bit: 1: Clear LSIRDYF interrupt flag; 0: No effect.	0
15	Reserved	RO	Reserved	0
14	PLL3RDYIE	RW	PLL3 ready interrupt enable bit: 1: Enable PLL3 ready interrupt; 0: Disable PLL3 ready interrupt.	0
13	PLL2RDYIE	RW	PLL2 ready interrupt enable bit: 1: Enable PLL2 ready interrupt; 0: Disable PLL2 ready interrupt.	0
12	PLLRDYIE	RW	PLL ready interrupt enable bit: 1: Enable PLL ready interrupt; 0: Disable PLL ready interrupt.	0
11	HSERDYIE	RW	HSE ready interrupt enable bit: 1: Enable HSE ready interrupt; 0: Disable HSE ready interrupt.	0
10	HSIRDYIE	RW	HSI ready interrupt enable bit: 1: Enable HSI ready interrupt; 0: Disable HSI ready interrupt.	0
9	LSERDYIE	RW	LSE ready interrupt enable bit: 1: Enable LSE ready interrupt; 0: Disable LSE ready interrupt.	0
8	LSIRDYIE	RW	LSI ready interrupt enable bit: 1: Enable LSI ready interrupt; 0: Disable LSI ready interrupt.	0
7	CSSF	RO	Clock security system interrupt flag bit: 1:Clock security interrupt caused by HSE clock failure;	0

			0:No clock security system interrupt caused by HSE clock failure. Set by hardware. Write 1 to CSSC bit by software to clear.	
6	PLL3RDYF	RO	PLL3 clock ready interrupt flag: 1: Clock ready interrupt caused by PLL3 clock; 0: No clock ready interrupt caused by PLL3 clock. Set by hardware. Write 1 to PLL3RDYC by software to clear. *Note: Applied for CH32F20x_D8C and CH32V30x_D8C.*	0
5	PLL2RDYF	RO	PLL2 clock ready interrupt flag: 1: Clock ready interrupt caused by PLL2 clock; 0: No clock ready interrupt caused by PLL2 clock. Set by hardware. Write 1 to PLL2RDYC by software to clear. *Note: Applied for CH32F20x_D8C and CH32V30x_D8C.*	0
4	PLLRDYF	RO	PLL clock ready interrupt flag: 1: Clock ready interrupt caused by PLL clock; 0: No clock ready interrupt caused by PLL clock. Set by hardware. Write 1 to PLLRDYC by software to clear.	0
3	HSERDYF	RO	HSE clock ready interrupt flag: 1: Clock ready interrupt caused by HSE clock; 0: No clock ready interrupt caused by HSE clock. Set by hardware. Write 1 to HSERDYC bit by software to clear.	0
2	HSIRDYF	RO	HSI clock ready interrupt flag: 1: Clock ready interrupt caused by HSI clock; 0: No clock ready interrupt caused by HSI clock. Set by hardware. Write 1 to HSIRDYC bit by software to clear.	0
1	LSERDYF	RO	LSE clock ready interrupt flag: 1: Clock ready interrupt caused by LSE clock; 0: No clock ready interrupt caused by LSE clock. Set by hardware. Write 1 to LSERDYC bit by software to clear.	0
0	LSIRDYF	RO	LSI clock ready interrupt flag: 1: Clock ready interrupt caused by LSI clock; 0: No clock ready interrupt caused by LSI clock. Set by hardware. Write 1 to LSIRDYC bit by software to clear.	0

3.4.4 APB2 Peripheral Reset Register (RCC_APB2PRSTR)

Offset address: 0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											TIM10	TIM9RST	Reserved		

											RST				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	USART1 RST	TIM8 RST	SPI1 RST	TIM1 RST	ADC2 RST	ADC1 RST	Reserved		IOPE RST	IOPD RST	IOPC RST	IOPB RST	IOPA RST	Reserved	AFIO RST

Bit	Name	Access	Description	Reset value
[31:21]	Reserved	RO	Reserved.	0
20	TIM10RST	RW	TIM10 reset: 1: Reset TIM10; 0: No effect.	0
19	TIM9RST	RW	TIM9 reset: 1: Reset TIM9; 0: No effect.	0
[18:15]	Reserved	RO	Reserved.	0
14	USART1RST	RW	USART1 reset: 1: Reset USART1; 0: No effect.	0
13	TIM8RST	RW	TIM8 reset: 1: Reset TIM8; 0: No effect.	0
12	SPI1RST	RW	SPI1 reset: 1: Reset SPI1; 0: No effect.	0
11	TIM1RST	RW	TIM1 reset: 1: Reset TIM1; 0: No effect.	0
10	ADC2RST	RW	ADC2 reset: 1: Reset ADC2; 0: No effect.	0
9	ADC1RST	RW	ADC1 reset: 1: Reset ADC1; 0: No effect.	0
[8:7]	Reserved	RO	Reserved	0
6	IOPERST	RW	IO port PE reset: 1: Reset PE; 0: No effect.	0
5	IOPDRST	RW	IO port PD reset: 1: Reset PD; 0: No effect.	0
4	IOPCRST	RW	IO port PC reset: 1: Reset PC; 0: No effect.	0
3	IOPBRST	RW	IO port PB reset: 1: Reset PB; 0: No effect.	0
2	IOPARST	RW	IO port PA reset: 1: Reset PA; 0: No effect.	0
1	Reserved	RO	Reserved	0
0	AFIORST	RW	Alternate function IO reset: 1: Reset alternate function; 0: No effect.	0

3.4.5 APB1 Peripheral Reset Register (RCC_APB1PRSTR)

Offset address: 0x10

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

Reserved		DAC RST	PWR RST	BKP RST	CAN2 RST	CAN1 RST	Reserved	USBD RST	I2C2 RST	I2C1 RST	UART5 RST	UART4 RST	USART3 RST	USART2 RST	Reserved
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3 RST	SPI2 RST	Reserved		WWDG RST	Reserved		UART8RST	UART7RST	UART6RST	TIM7 RST	TIM6 RST	TIM5 RST	TIM4 RST	TIM3 RST	TIM2 RST

Bit	Name	Access	Description	Reset value
[31:30]	Reserved	RO	Reserved.	0
29	DACRST	RW	DAC module reset: 1: Reset DAC; 0: No effect.	0
28	PWRRST	RW	Power interface module reset: 1: Reset Power interface; 0: No effect.	0
27	BKPRST	RW	Backup interface reset: 1: Reset back-up interface; 0: No effect.	0
26	CAN2RST	RW	CAN2 module reset: 1: Reset CAN2; 0: No effect.	0
25	CAN1RST	RW	CAN1 module reset: 1: Reset CAN1; 0: No effect.	0
24	Reserved	RO	Reserved	0
23	USBDRST	RW	USBD module reset: 1: Reset USBD; 0: No effect.	0
22	I^2C2RST	RW	I^2C2 interface reset: 1: Reset I^2C2; 0: No effect.	0
21	I^2C1RST	RW	I^2C1 interface reset: 1: Reset I^2C1; 0: No effect.	0
20	UART5RST	RW	UART5 interface reset: 1: Reset UART5; 0: No effect.	0
19	UART4RST	RW	UART4 interface reset: 1: Reset UART4; 0: No effect.	0
18	USART3RST	RW	USART3 interface reset: 1: Reset USART3; 0: No effect.	0
17	USART2RST	RW	USART2 interface reset: 1: Reset USART2; 0: No effect.	0
16	Reserved	RO	Reserved	0
15	SPI3RST	RW	SPI3 interface reset: 1: Reset SPI3; 0: No effect.	0
14	SPI2RST	RW	SPI2 interface reset: 1: Reset SPI2; 0: No effect.	0
[13:12]	Reserved	RO	Reserved	0
11	WWDGRST	RW	Window watchdog reset: 1: Reset window watchdog; 0: No effect.	0
[10:9]	Reserved	RO	Reserved	0
8	UART8RST	RW	UART8 interface reset:	0

			1: Reset UART8; 0: No effect.	
7	UART7RST	RW	UART7 interface reset: 1: Reset UART7; 0: No effect.	0
6	UART6RST	RW	UART6 interface reset: 1: Reset UART6; 0: No effect.	0
5	TIM7RST	RW	Timer7 module reset: 1: Reset Timer7; 0: No effect.	0
4	TIM6RST	RW	Timer6 module reset: 1: Reset Timer6; 0: No effect.	0
3	TIM5RST	RW	Timer5 module reset: 1: Reset Timer5; 0: No effect.	0
2	TIM4RST	RW	Timer4 module reset: 1: Reset Timer4; 0: No effect.	0
1	TIM3RST	RW	Timer3 module reset: 1: Reset Timer3; 0: No effect.	0
0	TIM2RST	RW	Timer2 module reset: 1: Reset Timer2; 0: No effect.	0

3.4.6 AHB Peripheral Clock Enable Register (RCC_AHBPCENR)

Offset address: 0x14

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETHMACTXEN	ETHMACEN	DVPEN	OTG_FSEN	USBHSEN	SDIOEN	RNGEN	FSMCEN	Reserved	CRCEN	Reserved	Reserved	Reserved	SRAMEN	DMA2EN	DMA1EN

Bit	Name	Access	Description	Reset value
[31:18]	Reserved	RO	Reserved	0
17	BLES	RW	BLES clock enable: 1: BLES clock enabled; 0: BLES clock disabled. *Note: Applied for CH32V20x_D8W and CH32F20x_D8W.*	1
16	BLEC	RW	BLEC clock enable: 1: BLEC clock enabled; 0: BLEC clock disabled. *Note: Applied for CH32V20x_D8W and CH32F20x_D8W.*	1
16	ETHMACRXEN	RW	Ethernet MAC receive clock enable: 1: Ethernet MAC receive clock enabled; 0: Ethernet MAC receive clock disabled. *Note: Applied for CH32V30x_D8C and CH32F20x_D8C with 100M/Gigabit external PHY.*	0
15	ETHMACTXEN	RW	Ethernet MAC transmit clock enable:	0

			1: Ethernet MAC transmit clock enabled; 0: Ethernet MAC transmit clock disabled. *Note: Applied for CH32V30x_D8C and CH32F20x_D8C with 100M/Gigabit external PHY.*	
14	ETHMACEN	RW	Ethernet MAC clock enable: 1: Ethernet MAC clock enabled; 0: Ethernet MAC clock disabled. *Note: Applied for CH32V30x_D8C and CH32F20x_D8C with 100M/Gigabit external PHY.*	0
13	DVPEN	RW	DVP module clock enable bit: 1: DVP clock enabled; 0: DVP clock disabled.	0
12	OTG_FSEN	RW	USBOTG_FS module clock enable bit: 1: USBOTG_FS clock enabled; 0: USBOTG_FS clock disabled.	0
11	USBHSEN	RW	USBHS module clock enable bit: 1: USBHS clock enabled; 0: USBHS clock disabled.	0
10	SDIOEN	RW	SDIO module clock enable bit: 1: SDIO clock enabled; 0: SDIO clock disabled.	0
9	RNGEN	RW	RNG module clock enable bit: 1: RNG clock enabled; 0: RNG clock disabled.	0
8	FSMCEN	RW	FSMCEN module clock enable bit: 1: FSMCEN clock enabled; 0: FSMCEN clock disabled.	0
7	Reserved	RO	Reserved	0
6	CRCEN	RW	CRC module clock enable bit: 1: CRC clock enabled; 0: CRC clock disabled.	0
5	Reserved	RO	Reserved	0
4	Reserved	RO	Reserved	0
3	Reserved	RO	Reserved	0
2	SRAMEN	RW	SRAM interface clock enable bit: 1: In sleep mode, SRAM interface clock enabled; 0: In sleep mode, SRAM interface clock disabled.	1
1	DMA2EN	RW	DMA2 module clock enable bit: 1: DMA2 clock enabled; 0: DMA2 clock disabled.	0
0	DMA1EN	RW	DMA1 module clock enable bit: 1: DMA1 clock enabled; 0: DMA1 clock disabled.	0

Note: When the peripheral clock is not enabled, the value of the peripheral register cannot be read by software, and the returned value is always 0.

3.4.7 APB2 Peripheral Clock Enable Register (RCC_APB2PCENR)

Offset address: 0x18

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											TIM10EN	TIM9EN	Reserved		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	USART1EN	TIM8EN	SPI1EN	TIM1EN	ADC2EN	ADC1EN	Reserved		IOPEEN	IOPDEN	IOPCEN	IOPBEN	IOPAEN	Reserved	AFIOEN

Bit	Name	Access	Description	Reset value
[31:21]	Reserved	RO	Reserved.	0
20	TIM10EN	RW	TIM10 interface clock enable bit: 1: TIM10 clock enabled; 0: TIM10 clock disabled.	0
19	TIM9EN	RW	TIM9 interface clock enable bit: 1: TIM9 clock enabled; 0: TIM9 clock disabled.	0
[18:15]	Reserved	RO	Reserved.	0
14	USART1EN	RW	USART1 interface clock enable bit: 1: USART1 clock enabled; 0: USART1 clock disabled.	0
13	TIM8EN	RW	TIM8 module clock enable bit: 1: TIM8 clock enabled; 0: TIM8 clock disabled.	0
12	SPI1EN	RW	SPI1 interface clock enable bit: 1: SPI1 clock enabled; 0: SPI1 clock disabled.	0
11	TIM1EN	RW	TIM1 module clock enable bit: 1: TIM1 clock enabled; 0: TIM1 clock disabled.	0
10	ADC2EN	RW	ADC2 module clock enable bit: 1: ADC2 clock enabled; 0: ADC2 clock disabled.	0
9	ADC1EN	RW	ADC1 module clock enable bit: 1: ADC1 clock enabled; 0: ADC1 clock disabled.	0
[8:7]	Reserved	RO	Reserved.	0
6	IOPEEN	RW	IO port PE clock enable bit: 1: PE clock enabled; 0: PE clock disabled.	0
5	IOPDEN	RW	IO port PD clock enable bit: 1: PD clock enabled; 0: PD clock disabled.	0
4	IOPCEN	RW	IO port PC clock enable bit: 1: PC clock enabled; 0: PC clock disabled.	0
3	IOPBEN	RW	IO port PB clock enable bit: 1: PB clock enabled; 0: PB clock disabled.	0
2	IOPAEN	RW	IO port PA clock enable bit: 1: PA clock enabled; 0: PA clock disabled.	0
1	Reserved	RO	Reserved.	0
0	AFIOEN	RW	Alternate function IO clock enable bit: 1: Alternate function IO clock enabled; 0: Alternate function IO clock disabled.	0

Note: When the peripheral clock is not enabled, the value of the peripheral register cannot be read by software, and the returned value is always 0.

3.4.8 APB1 Peripheral Clock Enable Register (RCC_APB1PCENR)

Offset address: 0x1C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		DAC EN	PWR EN	BKP EN	CAN 2EN	CAN 1EN	Reser ved	USB DEN	I2C2 EN	I2C1 EN	UART5 EN	UART4 EN	USAR T3EN	USAR T2EN	Reser ved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3 EN	SPI2 EN	Reserved		WWDG EN	Reserved		UAR T8EN	UAR T7EN	UAR T6EN	TIM7 EN	TIM6 EN	TIM5 EN	TIM4 EN	TIM3 EN	TIM2 EN

Bit	Name	Access	Description	Reset value
[31:30]	Reserved	RO	Reserved.	0
29	DACEN	RW	DAC module clock enable bit: 1: DAC clock enabled; 0: DAC clock disabled.	0
28	PWREN	RW	Power interface clock enable bit: 1: Power interface clock enabled; 0: Power interface clock disabled.	0
27	BKPEN	RW	Backup interface clock enable bit: 1: Backup interface clock enabled; 0: Backup interface clock disabled.	0
26	CAN2EN	RW	CAN2 module clock enable bit: 1: CAN2 clock enabled; 0: CAN2 clock disabled.	0
25	CAN1EN	RW	CAN1 module clock enable bit: 1: CAN1 clock enabled; 0: CAN1 clock disabled.	0
24	Reserved	RO	Reserved	0
23	USBDEN	RW	USBD module clock enable bit: 1: USBD clock enabled; 0: USBD clock disabled.	0
22	I2C2EN	RW	I2C2 interface clock enable bit: 1: I2C2 clock enabled; 0: I2C2 clock disabled.	0
21	I2C1EN	RW	I2C1 interface clock enable bit: 1: I2C1 clock enabled; 0: I2C1 clock disabled.	0
20	UART5EN	RW	UART5 interface clock enable bit: 1: UART5 clock enabled; 0: UART5 clock disabled.	0
19	UART4EN	RW	UART4 interface clock enable bit: 1: UART4 clock enabled; 0: UART4 clock disabled.	0
18	USART3EN	RW	USART3 interface clock enable bit: 1: USART3 clock enabled; 0: USART3 clock disabled.	0
17	USART2EN	RW	USART2 interface clock enable bit: 1: USART2 clock enabled; 0: USART2 clock disabled.	0
16	Reserved	RO	Reserved	0
15	SPI3EN	RW	SPI3interface clock enable bit: 1: SPI3 clock enabled; 0: SPI3 clock disabled.	0
14	SPI2EN	RW	SPI2interface clock enable bit: 1: SPI2 clock enabled; 0: SPI2 clock disabled.	0

[13:12]	Reserved	RO	Reserved	0
11	WWDGEN	RW	Window watchdog clock enable bit: 1: Window watchdog clock enabled; 0: Window watchdog clock disabled.	0
[10:9]	Reserved	RO	Reserved	0
8	UART8EN	RW	UART8 enable bit: 1: UART8 clock enabled; 0: UART8 clock disabled.	0
7	UART7EN	RW	UART7 enable bit: 1: UART7 clock enabled; 0: UART7 clock disabled.	0
6	UART6EN	RW	UART6 enable bit: 1: UART6 clock enabled; 0: UART6 clock disabled.	0
5	TIM7EN	RW	Timer7 module clock enable bit: 1: Timer7 clock enabled; 0: Timer7 clock disabled.	0
4	TIM6EN	RW	Timer6 module clock enable bit: 1: Timer6 clock enabled; 0: Timer6 clock disabled.	0
3	TIM5EN	RW	Timer5 module clock enable bit: 1: Timer5 clock enabled; 0: Timer5 clock disabled.	0
2	TIM4EN	RW	Timer4 module clock enable bit: 1: Timer4 clock enabled; 0: Timer4 clock disabled.	0
1	TIM3EN	RW	Timer3 module clock enable bit: 1: Timer3 clock enabled; 0: Timer3 clock disabled.	0
0	TIM2EN	RW	Timer2 module clock enable bit: 1: Timer2 clock enabled; 0: Timer2 clock disabled.	0

Note: When the peripheral clock is not enabled, the value of the peripheral register cannot be read by software, and the returned value is always 0.

3.4.9 Backup Domain Control Register (RCC_BDCTLR)

Offset address: 0x20

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTCEN	Reserved					RTCSEL[1:0]		Reserved					LSEBYP	LSERDY	LSEON

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved.	0
16	BDRST	RW	Backup domain software reset control: 1: Reset the entire backup domain. 0: Reset not activated.	0
15	RTCEN	RW	RTC clock enable control: 1: RTC clock enabled; 0: RTC clock disabled. *Note: The RTC clock can be enabled only when RTCSEL!=0. Otherwise, it is forced to 0 by hardware.*	0

[14:10]	Reserved	RO	Reserved.	0

[9:8]	RTCSEL[1:0]	RW	RTC clock source selection:
			00: No clock;
			01: LSE oscillator clock;
			10: LSI oscillator clock;
			11: HSE oscillator clock divided by 128.
			Note: Once the RTC clock source has been selected (RTCEN=1), it cannot be changed until the next backup domain is reset. The default can be restored by setting the BDRST bit.
				0

[7:3]	Reserved	RO	Reserved.	0
2	LSEBYP	RW	External low-speed (LSE) oscillator bypass control bit: 1: LSE oscillator bypassed (external clock source is used); 0: LSE oscillator not bypassed. *Note: This bit can be written only when LSEON is 0.*	0
1	LSERDY	RO	External low-speed oscillator ready flag bit (set by hardware): 1: LSE oscillator ready; 0: LSE oscillator not ready. *Note: After the LSEON bit is cleared, it takes 6 cycles of LSE clock to clear this bit.*	0
0	LSEON	RW	External low-speed crystal oscillator enable control bit: 1: LSE oscillator enabled; 0: LSE oscillator disabled.	0

Note: The LSEON, LSEBYP, RTCSEL and RTCEN bits in the backup domain control register (RCC_BDCTLR) are in the backup domain. Therefore, these bits are in a write-protected status after reset, and these bits can only be changed after the DBP bit in the power control register (PWR_CR) is set to 1. These bits can only be cleared by the backup domain reset. Any internal or external reset will not affect these bits.

3.4.10 Control/Status Register (RCC_RSTSCKR)

Offset address: 0x24

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPWRRSTF	WWDGRSTF	IWDGRSTF	SFTRSTF	PORRSTF	PINRSTF	Reserved	RMVF	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	LPWRRSTF	RW	Low-power reset flag: 1: Low-power reset occurred; 0: No low-power reset occurred. When low-power management reset occurs, set by hardware. Cleared by writing to the RMVF bit by software.	0

30	WWDGRSTF	RW	Window watchdog reset flag: 1: Window watchdog reset occurred; 0: No window watchdog reset occurred. When the window watchdog reset occurs, set by hardware. Cleared by writing to the RMVF bit by software.	0
29	IWDGRSTF	RW	Independent watchdog reset flag: 1: Independent watchdog reset occurred; 0: No independent watchdog reset occurred. When independent watchdog reset occurs, set by hardware. Cleared by writing to the RMVF bit by software.	0
28	SFTRSTF	RW	Software reset flag: 1: Software reset occurred; 0: No software reset occurred. When software reset occurs, set by hardware. Cleared by writing to the RMVF bit by software.	0
27	PORRSTF	RW	Power-on/power-down reset flag: 1: Power-on/power-down reset occurred; 0: No power-on/power-down reset occurred. When the power-on/power-down reset, set by hardware. Cleared by writing to the RMVF bit by software.	1
26	PINRSTF	RW	External manual reset (NRST pin) flag: 1: NRST pin reset occurred; 0: No NRST pin reset occurred. When the NRST pin reset, set by hardware. Cleared by writing to the RMVF bit by software.	0
25	Reserved	RO	Reserved.	0
24	RMVF	RW	Remove reset flag control: 1: Clear reset flag; 0: No effect.	0
[23:2]	Reserved	RO	Reserved.	0

1	LSIRDY	RO	Internal low-speed clock (LSI) ready flag bit (set by hardware):
			1: Internal low-speed clock (40 KHz) ready;
			0: Internal low-speed clock (40 KHz) not ready;
			Note: After the LSION bit is cleared, it takes 3 cycles of LSI clock to clear this bit.
				0

0	LSION	RW	Internal low-speed clock (LSI) enable control bit: 1:LSI (40KHz) oscillator enabled; 0: LSI (40KHz) oscillator disabled.	0

Note: Except that the reset flag can only be cleared by power-on reset, others can be cleared by system reset.

3.4.11 AHB Peripheral Reset Register (RCC_AHBRSTR)

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	ETH MAC RST	DVP RST	OTGF SRST	Reserved											

Bit	Name	Access	Description	Reset value
[31:15]	Reserved	RO	Reserved	0
14	ETHMACRST	RW	Ethernet MAC reset control: 1: Reset Ethernet MAC; 0: No effect.	0
13	DVPRST	RO	DVP reset control: 1: Reset DVP; 0: No effect.	0
12	OTGFSRST	RW	USBOTG_FS module reset control: 1: Reset USBHD; 0: No effect.	0
[11:0]	Reserved	RO	Reserved	0

3.4.12 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
USBHSSRC	USBHSPLL	USBHSCLK[1:0]		USBHSPLL_SRC	USBHSDIV[2:0]			Reserved	ETH1GEN	ETH1GSRC		RNGSRC	I2S3SRC	I2S2SRC	PREDIV1SRC
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PLL3MUL[3:0]				PLL2MUL[3:0]				PREDIV2[3:0]				PREDIV1[3:0]			

Bit	Name	Access	Description	Reset value
31	USBHSSRC	RW	USBHS 48MHz clock source selection: 1: USB PHY; 0: PLL CLK.	0
30	USBHSPLL	RW	USBHS PHY internal PLL control bit: 1: USB PHY internal PLL enabled; 0: USB PHY internal PLL disabled.	0

[29:28]	USBHSCLK[1:0]	RW	USBHS PLL reference clock selection (USBHS_PLL_SRC/ USBHS_PREDIV):
			00: 3MHz;
			01: 4MHz;
			10: 8MHz;
			11: 5MHz.
				00

27	USBHSPLLSRC	RW	USBHS PLL reference source selection: 1: HSI 0: HSE	0

[26:24]	USBHSDIV[2:0]	RW	USBHS PLL reference source prescaler:
			000: Not divided; 001: Divided by 2;
			010: Divided by 3; 011: Divided by 4;
			100: Divided by 5; 101: Divided by 6;
			110: Divided by 7; 111: Divided by 8.
				000

23	Reserved	RO	Reserved	0
22	ETH1G_125M_EN	RW	Gigabit Ethernet 125M clock enable bit: 1: Enabled; 0: Disabled.	

[21:20]	ETH1GSRC[2:0]	RW	Gigabit Ethernet 125M clock selection:
			00: PLL2 VCO;
			01: PLL3 VCO;
			1x: External PB1 pin input.	

19	RNG_SRC	RW	RNG clock source selection: 1: PLL3 VCO; 0: System clock.	
18	I2S3SRC	RW	I2S3 clock source: 1: PLL3 VCO; 0: System clock (SYSCLK).	0
17	I2S2SRC	RW	I2S2 clock source: 1: PLL3 VCO; 0: System clock (SYSCLK).	0
16	PREDIV1SRC	RW	PREDIV1 clock source: 1: PLL2; 0: HSE.	0

[15:12]	PLL3MUL[3:0]	RW	PLL3 multiplication factor (these bits can be written only when PLL3 is disabled).
			0000: PLL3 input clock x 2.5;
			0001: PLL3 input clock x 12.5;
			0010: PLL3 input clock x 4;
			0011: PLL3 input clock x 5;
			0100: PLL3 input clock x 6;
			0101: PLL3 input clock x 7;
			0110: PLL3 input clock x 8;
			0111: PLL3 input clock x 9;
			1000: PLL3 input clock x 10;
			1001: PLL3 input clock x 11;
			1010: PLL3 input clock x 12;
			1011: PLL3 input clock x 13;
			1100: PLL3 input clock x 14;
			1101: PLL3 input clock x 15;
			1110: PLL3 input clock x 16;
			1111: PLL3 input clock x 20.
				0000

[11:8]	PLL2MUL[3:0]	RW	PLL2 multiplication factor (these bits can be written only when PLL2 is disabled).
			0000: PLL2 input clock x 2.5;
			0001: PLL2 input clock x 12.5;
			0010: PLL2 input clock x 4;
			0011: PLL2 input clock x 5;
			0100: PLL2 input clock x 6;
			0101: PLL2 input clock x 7;
			0110: PLL2 input clock x 8;
			0111: PLL2 input clock x 9;
				0000

			1000: PLL2 input clock x 10;
			1001: PLL2 input clock x 11;
			1010: PLL2 input clock x 12;
			1011: PLL2 input clock x 13;
			1100: PLL2 input clock x 14;
			1101: PLL2 input clock x 15;
			1110: PLL2 input clock x 16;
			1111: PLL2 input clock x 20.	

[7:4]	PREDIV2[3:0]	RW	PREDIV2 prescaler factor (these bits can be written only when PLL2 and PLL3 both are disabled). 0000: PREDIV2 not divided; 0001: PREDIV2 input clock divided by 2; 0010: PREDIV2 input clock divided by 3; 0011: PREDIV2 input clock divided by 4; 0100: PREDIV2 input clock divided by 5; 0101: PREDIV2 input clock divided by 6; 0110: PREDIV2 input clock divided by 7; 0111: PREDIV2 input clock divided by 8 ; 1000: PREDIV2 input clock divided by 9; 1001: PREDIV2 input clock divided by 10; 1010: PREDIV2 input clock divided by 11; 1011: PREDIV2 input clock divided by 12; 1100: PREDIV2 input clock divided by 13; 1101: PREDIV2 input clock divided by 14; 1110: PREDIV2 input clock divided by 15; 1111: PREDIV2 input clock divided by 16.	0000
[3:0]	PREDIV1[3:0]	RW	PREDIV1 prescaler factor (these bits can be written only when PLL is disabled). 0000: PREDIV1 not divided; 0001: PREDIV1 input clock divided by 2; 0010: PREDIV1 input clock divided by 3; 0011: PREDIV1 input clock divided by 4; 0100: PREDIV1 input clock divided by 5; 0101: PREDIV1 input clock divided by 6; 0110: PREDIV1 input clock divided by 7; 0111: PREDIV1 input clock divided by 8 ; 1000: PREDIV1 input clock divided by 9; 1001: PREDIV1 input clock divided by 10; 1010: PREDIV1 input clock divided by 11; 1011: PREDIV1 input clock divided by 12; 1100: PREDIV1 input clock divided by 13; 1101: PREDIV1 input clock divided by 14; 1110: PREDIV1 input clock divided by 15; 1111: PREDIV1 input clock divided by 16. *Note: Bit0 is the same as bit17 in the RCC_CFGR0 register. Bit0 in this register changes when bit17 in the*	0000

			RCC_CFGR0 register is modified.	

Note: Applied for CH32F20x_D8C and CH32V30x_D8C.

3.4.13 HSE Crystal Oscillator Calibration Control Register (HSE_CAL_CTRL)

Offset address: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HSEC				HSEFAULT	Reserved	HSEITRIM [1:0]		Reserved							

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

Bit	Name	Access	Description	Reset value

[31:28]	HSEC[3:0]	RW	HSE internal match capacitor adjustment:
			111: 22pF;
			110: 20pF;
			101: 18pF;
			100: 16pF;
			011: 14pF;
			010: 12pF;
			001: 10pF;
			000: 8pF.
				0

[27]	HSEFAULT	RW	HSE fault detection disable: 1: Analog input HSE fault detection signal disabled; 0: Analog input HSE fault detection signal enabled.	0
[26]	Reserved	RW	Reserved	0
[25:24]	HSEITRIM[1:0]	RW	HSE starting current adjustment bit.	01b
[23:0]	Reserved	RO	Reserved	0

3.4.14 LSI Crystal Oscillator Calibration Tune Register (LSI32K_TUNE)

Offset address: 0x0A

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

Bit	Name	Access	Description	Reset value
[15:13]	Reserved	RO	Reserved	0
[12:5]	HTUNE[7:0]	RW	LSI32K fine tune	80h
[4:0]	LTUNE[4:0]	RW	LSI32Kcoarse tune	11h

3.4.15 LSI Crystal Oscillator Calibration Configuration Register (LSI32K_CAL_CFG)

Offset address: 0x1D

7	6	5	4	3	2	1	0
Reserved	LPEN	WKUPEN	HALTMD	CNTVLU[3:0]			

Bit	Name	Access	Description	Reset value
[7]	Reserved	RO	Reserved	0
[6]	LPEN	RW	Calibration enable in low power mode: 0: Calibration disabled in low power mode; 1: Calibration enabled in low power mode. *Note: It works together with RB_HSE_KEEP_LP in EXTEN.*	0
[5]	WKUPEN	RW	LSI32K wake-up interrupt enable: 0: LSI32K wake-up interrupt disabled; 1: LSI32K wake-up interrupt enabled.	0
[4]	HALTMD	RW	LSI32K calibration count halt duration configuration: 0: Count halts for 1 CK32K cycle; 1: Count halts for 3 CK32K cycles.	0

[3:0]	CNTVLU[3:0]	RW	LSI32K calibration sampling duration configuration:
			0000: 2 CK32K cycles;
			0001: 4 CK32K cycles;
			0010: 32 CK32K cycles;
			0011: 64 CK32K cycles;
			0100: 128 CK32K cycles;
			0101: 256 CK32K cycles;
			0110: 512 CK32K cycles;
			0111: 1024 CK32K cycles;
			1000: 1088 CK32K cycles;
			1001: 1152 CK32K cycles;
			1010: 1216 CK32K cycles;
			1011: 1280 CK32K cycles;
			1100: 2000 CK32K cycles.
			Note: Other configuration values correspond to 2 CK32K cycles.
				0001b

3.4.16 LSI Crystal Oscillator Calibration Status Register (LSI32K_CAL_STATR)

Offset address: 0x20

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

Bit	Name	Access	Description	Reset value
[15]	IFEND	RW1	LSI32K calibration count end interrupt flag: 0: During sampling count, no flag; 1: Sampling count ends, the flag is set.	0
[14]	CNTOV	RW1	LSI32K sampling counter overflow flag: 0: Not overflow; 1: Overflow.	0
[13:0]	CNT[13:0]	RO	Count for several CK32K cycles based on system clock frequency. *Note: The specific number of CK32K cycles can be configured.*	0x0000

3.4.17 LSI Crystal Oscillator Calibration Counter (LSI32K_CAL_OV_CNT)

Offset address: 0x22

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

Bit	Name	Access	Description	Reset value
[7:0]	OVCNT[7:0]	RO	LSI32K sampling counter overflow count. *Note: Clear the overflow flag and this counter will be cleared.*	0x00

3.4.18 LSI Crystal Oscillator Calibration Control Register (LSI32K_CAL_CTRL)

Offset address: 0x23

7	6	5	4	3	2	1	0
HALT	Reserved					CALEN	CALINTEN

Bit	Name	Access	Description	Reset value
[7]	HALT	RO	LSI32K calibration count halt: 0: During count, and the count value is not available; 1: Count halts, and the count value is available.	1
[6:2]	Reserved	RO	Reserved	0
[1]	CALEN	RW	LSI32K calibration enable:	0
[0]	CALINTEN	RW	LSI32K calibration interrupt enable	0

Note: Applied for CH32F20x_D8C and CH32V30x_D8C.

Chapter 4 Backup Register (BKP)

This chapter applies to the whole family of CH32F2x, CH32V2x and CH32V3x.

Backup Register (BKP) provides up to 42 16-bit backup data registers, which can be used to store up to 84 bytes of user data. After the main power supply (V_{DD}) switched off, these data can still be maintained by the power supply of V_{BAT} without being affected by the standby status, system reset or power reset. In addition, the BKP unit also provides tamper detection management, RTC clock calibration and pulse output functions.

4.1 Main features

- Tamper detection (TAMPER) function
- RTC clock calibration function
- Output RTC clock divided by 64, alarm pulse or second pulse on PC13 pin

4.2 Backup data register

After the microcontroller reset, the access to the backup register and RTC will be disabled, and the access to the backup register needs to be enabled by the following operations:

1) Set the PWREN bit and BKPEN bit in the RCC_APB1PCENR register to enable the power supply and the operating clock of the backup interface;
2) Set the DBP bit in the power control register (PWR_CTLR) to enable access to the backup register and RTC register.

4.2.1 Backup data register

The backup data register can be used as a general data buffer. Because of its feature of saving data by V_{BAT} power when V_{DD} is powered off, it can be used to store some important or sensitive data. But these data will be completely cleared after the tamper event occurs.

4.2.2 Tamper Detection

When an external signal (rising or falling edge) is provided, a tamper event is generated, and the important information retained in the current system is cleared automatically by hardware. Tamper detection increases the security of system information.

A tamper event is generated when a transition edge is detected on the tamper detection pin (depending on the TPAL bit). If the tamper detection interrupt is enabled, a tamper detection interrupt is also generated at the same time. As long as a tamper event occurs, all backup data registers are cleared. In addition, the hardware detection adopts the memory mode. Even if the tamper detection function is not enabled (TPE=0), the system still samples and checks whether there is a transition edge, and if the TPAL bit selection is met, the tamper event is locked in advance, and the TPE bit is set to 1 to trigger a tamper event.

For example: When TPAL=0, if it is not enabled (TPE=0) but the TAMPER pin is already high, an additional tamper event is generated (the system locks the rising edge in advance) once TPE=1. When TPAL=1, if it is not enabled (TPE=0) but the TAMPER pin is already low, an additional tamper event is generated (the system locks the falling edge in advance) once TPE=1.

Therefore, in order to avoid unnecessary tamper events from causing the backup registers to be cleared, it is recommended to set the CTE bit in the BKP_TPCSR register to 1 at the beginning of the hardware detection of the tamper pin, to firstly clear the tamper events that may have been remembered by hardware, and ensure that the current tamper detection pin is invalid.

Note: When the V_{DD} power supply is disconnected, the tamper detection function is still valid. In order to avoid unnecessary backup registers of resetting data, the TAMPER pin shall be connected to the correct level correctly.

4.2.3 RTC calibration

For this function, the tamper detection pin must be selected as a common IO port. Clear the TPE bit in the BKP_TPCTLR register.

- Pulse output

Configure the ASOE bit in the BKP_OCTLR register, enable RTC pulse output, set the ASOS bit, and select either the second pulse output or the alarm pulse output.

- RTC calibration

After the CCO bit in the BKP_OCTLR register is configured, the internal RTC clock divided by 64 is output to the tamper detection pin (TAMPER). Through the actual test, adjust the clock and calibrate the RTC by configuring the CAL[6:0] bits by software.

4.2.4 BKP interface reset

The BKP domain can be independently powered-on by V_{BAT} when the V_{DD} power is switched off. The application code controls the reset of BKP domain register, the BKP_DATAR1-10, ASOS bit and ASOE bit in the backup data register are reset by setting the BDRST bit in the RCC_BDCTLR register by software, which is not affected by the RCC peripheral interface control BKPRST bit.

4.3 Register description

Table 4-1 BKP registers

Name	Access address	Description	Reset value
R16_BKP_DATAR1	0x40006C04	Backup data register 1	0x0000
R16_BKP_DATAR2	0x40006C08	Backup data register 2	0x0000
R16_BKP_DATAR3	0x40006C0C	Backup data register 3	0x0000
R16_BKP_DATAR4	0x40006C10	Backup data register 4	0x0000
R16_BKP_DATAR5	0x40006C14	Backup data register 5	0x0000
R16_BKP_DATAR6	0x40006C18	Backup data register 6	0x0000
R16_BKP_DATAR7	0x40006C1C	Backup data register 7	0x0000
R16_BKP_DATAR8	0x40006C20	Backup data register 8	0x0000
R16_BKP_DATAR9	0x40006C24	Backup data register 9	0x0000
R16_BKP_DATAR10	0x40006C28	Backup data register 10	0x0000
R16_BKP_OCTLR	0x40006C2C	RTC calibration register	0x0000
R16_BKP_TPCTLR	0x40006C30	Tamper detection control register	0x0000
R16_BKP_TPCSR	0x40006C34	Tamper detection status register	0x0000
R16_BKP_DATAR11	0x40006C40	Backup data register 11	0x0000
R16_BKP_DATAR12	0x40006C44	Backup data register 12	0x0000

R16_BKP_DATAR13	0x40006C48	Backup data register 13	0x0000
R16_BKP_DATAR14	0x40006C4C	Backup data register 14	0x0000
R16_BKP_DATAR15	0x40006C50	Backup data register 15	0x0000
R16_BKP_DATAR16	0x40006C54	Backup data register 16	0x0000
R16_BKP_DATAR17	0x40006C58	Backup data register 17	0x0000
R16_BKP_DATAR18	0x40006C5C	Backup data register 18	0x0000
R16_BKP_DATAR19	0x40006C60	Backup data register 19	0x0000
R16_BKP_DATAR20	0x40006C64	Backup data register 20	0x0000
R16_BKP_DATAR21	0x40006C68	Backup data register 21	0x0000
R16_BKP_DATAR22	0x40006C6C	Backup data register 22	0x0000
R16_BKP_DATAR23	0x40006C70	Backup data register 23	0x0000
R16_BKP_DATAR24	0x40006C74	Backup data register 24	0x0000
R16_BKP_DATAR25	0x40006C78	Backup data register 25	0x0000
R16_BKP_DATAR26	0x40006C7C	Backup data register 26	0x0000
R16_BKP_DATAR27	0x40006C80	Backup data register 27	0x0000
R16_BKP_DATAR28	0x40006C84	Backup data register 28	0x0000
R16_BKP_DATAR29	0x40006C88	Backup data register 29	0x0000
R16_BKP_DATAR30	0x40006C8C	Backup data register 30	0x0000
R16_BKP_DATAR31	0x40006C90	Backup data register 31	0x0000
R16_BKP_DATAR32	0x40006C94	Backup data register 32	0x0000
R16_BKP_DATAR33	0x40006C98	Backup data register 33	0x0000
R16_BKP_DATAR34	0x40006C9C	Backup data register 34	0x0000
R16_BKP_DATAR35	0x40006CA0	Backup data register 35	0x0000
R16_BKP_DATAR36	0x40006CA4	Backup data register 36	0x0000
R16_BKP_DATAR37	0x40006CA8	Backup data register 37	0x0000
R16_BKP_DATAR38	0x40006CAC	Backup data register 38	0x0000
R16_BKP_DATAR39	0x40006CB0	Backup data register 39	0x0000
R16_BKP_DATAR40	0x40006CB4	Backup data register 40	0x0000
R16_BKP_DATAR41	0x40006CB8	Backup data register 41	0x0000
R16_BKP_DATAR42	0x40006CBC	Backup data register 42	0x0000

Note: The BKP_DATARx (x=11-42) registers are applied for CH32F20x_D8, CH32F20x_D8C, CH32F20x_D8W, CH32V20x_D8, CH32V20x_D8W, CH32V30x_D8 and CH32V30x_D8C.

4.3.1 Backup Data Register (BKP_DATARx) (x=1-42)

Offset address: 0x04 to 0x28, 0x40 to 0xBC

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

Bit	Name	Access	Description	Reset value
[15:0]	D[15:0]	RW	Backup data, which can be written with user data. *Note: They are only reset by backup domain reset (BDRST) or by a TAMPER pin event (if the TAMPER pin function is activated).*	0

4.3.2 RTC Calibration Register (BKP_OCTLR)

Offset address: 0x2C

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

Bit	Name	Access	Description	Reset value
[15:10]	Reserved	RO	Reserved.	0
9	ASOS	RW	TAMPER pin alarm/second output selection: 1: Second pulse output selected; 0: Alarm pulse output selected. *Note: This bit can only be reset by backup domain reset (BDRST).*	0
8	ASOE	RW	TAMPER pin alarm/second output enable: 0: Alarm/second pulse output disabled; 1: Alarm/second pulse output enabled. *Note: This bit can only be reset by backup domain reset (BDRST).*	0
7	CCO	RW	Calibration clock output selection: 1: TEMPER pin output RTC clock divided by 64; 0: No calibration clock output. *Note 1: The tamper detection function must be switched off to enable this function.* *Note2: This bit is cleared when V_{DD} power supply is switched off.*	0
[6:0]	CAL	RW	Calibration value register. The value of this register indicates the number of clock pulses that will be skipped every 2^{20} clock pulses. This allows the calibration of the RTC clock. The clock can be slowed down from 0 to 121ppm.	0

4.3.3 Tamper Detection Control Register (BKP_TPCTLR)

Offset address: 0x30

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

Bit	Name	Access	Description	Reset value
[15:2]	Reserved	RO	Reserved.	0
1	TPAL	RW	TEMPER pin active level: 0: High level on the TEMPER pin cause all backup data registers cleared (hardware lock rising edge); 1: Low level on the TEMPER pin causer all backup data registers cleared (hardware lock falling edge);	0
0	TPE	RW	TEMPER pin enable:	0

			0: TEMPER pin used as common IO port; 1: TEMPER pin used for the tamper detection.	

Note: When the TPAL and TPE bits are cleared at the same time, a false tamper event occurs. It is recommended to change the status of the TPAL bit only when TPE is 0.

4.3.4 Tamper Detection Status Register (BKP_TPCSR)

Offset address: 0x34

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						TIF	TEF	Reserved					TPIE	CTI	CTE

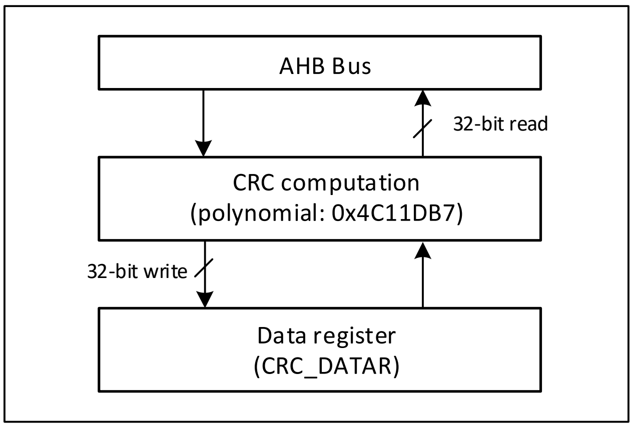
Bit	Name	Access	Description	Reset value
[15:10]	Reserved	RO	Reserved.	0
9	TIF	RO	Tamper interrupt flag. When a tamper event is detected and the TPIE bit is set to 1, this bit is set. Cleared by writing 1 to the CTI bit. If the TPIE bit is reset, this bit is reset at the same time. *Note: This bit is reset only when the system is reset or woken up from standby mode.*	0
8	TEF	RO	Tamper event flag. When a tamper event is detected, this bit is set. Cleared by writing 1 to the CTE bit. *Note: When this bit is 1, all BKP_DATARx registers are cleared, and all write operations to the BKP_DATARx register are invalid before this bit is not reset.*	0
[7:3]	Reserved	RO	Reserved.	0
2	TPIE	RW	Tamper interrupt enable: 0: Tamper detection interrupt disabled; 1: Tamper detection interrupt enabled (TPE needs to be set to 1). *Note 1: The tamper detection interrupt cannot wake up the core from low-power mode.* *Note 2: This bit is reset only when the system is reset or woken up from standby mode.*	0
1	CTI	WO	Clear Tamper detection interrupt. Write 1 to clear it, and the value read out is invalid.	0
0	CTE	WO	Clear Tamper detection event. Write 1 to clear it, and the value read out is invalid.	0

Chapter 5 Cyclic Redundancy Check (CRC)

This chapter applies to the whole family of CH32F2x, CH32V2x and CH32V3x.

The cyclic redundancy check (CRC) calculation unit is used to get a CRC code from a 32-bit data word and a fixed generator polynomial. It is generally used to verify data transmission or storage integrity. The hardware CRC calculation unit provided by the system can greatly save CPU and RAM resources and improve efficiency.

Figure 5-1 CRC structure block diagram



5.1 Main features

- CRC32 polynomial (0x4C11DB7)
  - $X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^{8}+X^{7}+X^{5}+X^{4}+X^{2}+X+1$
- Single input/output 32-bit data register
- CRC computation done in 4 AHB clock cycles (HCLK)

5.2 Functional description

- CRC unit reset

To start a new CRC calculation, it is needed to reset the CRC calculation unit. Write 1 to the RST bit in the CRC_CTLR register, to reset the data register by hardware, and it restores the initial value 0xFFFFFFFF.

- CRC calculation

The CRC calculation unit is a combination of the previous CRC calculation value and the new CRC calculation value. When writing into the CRC_DATAR register, new data is input to the hardware calculation unit. When reading the register, the latest CRC calculation value can be obtained. The hardware calculation interrupts the write operation of the system, so new values can be written continuously.
Note: The CRC unit calculates the whole 32-bit data word, rather than byte per byte.

- Independent data buffer

The CRC unit provides an 8-bit independent data register (CRC_IDATAR), which is used to temporarily store 1 byte of data for the application code and is not affected by the reset of the CRC unit.

5.3 Register description

Table 5-1 CRC registers

Name	Access address	Description	Reset value
R32_CRC_DATAR	0x40023000	Data register	0xFFFFFFFF
R8_CRC_IDATAR	0x40023004	Independent data register	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	RW	Write the original data; read the calculation result.	0xFFFFFFFF

5.3.2 Independent Data Register (CRC_IDATAR)

Offset address: 0x04

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

Bit	Name	Access	Description	Reset value
[7:0]	IDR[7:0]	RW	8-bit general register, can be used as data buffer. This register is not affected by RST in CRC_CTLR.	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 to set. Automatically cleared by hardware. It sets the data register to 0xFFFFFFFF.	0

Chapter 6 Real-Time Clock (RTC)

This chapter applies to the whole family of CH32F2x, CH32V2x and CH32V3x.

The real-time clock (RTC) is an independent timer module. It supports up to 32-bit programmable counter. With software, the real-time clock function can be implemented, and the counter value can be modified to re-configure the current time and date of the system. The RTC module is in the backup power supply area, and is not affected by the system reset and standby mode wake-up.

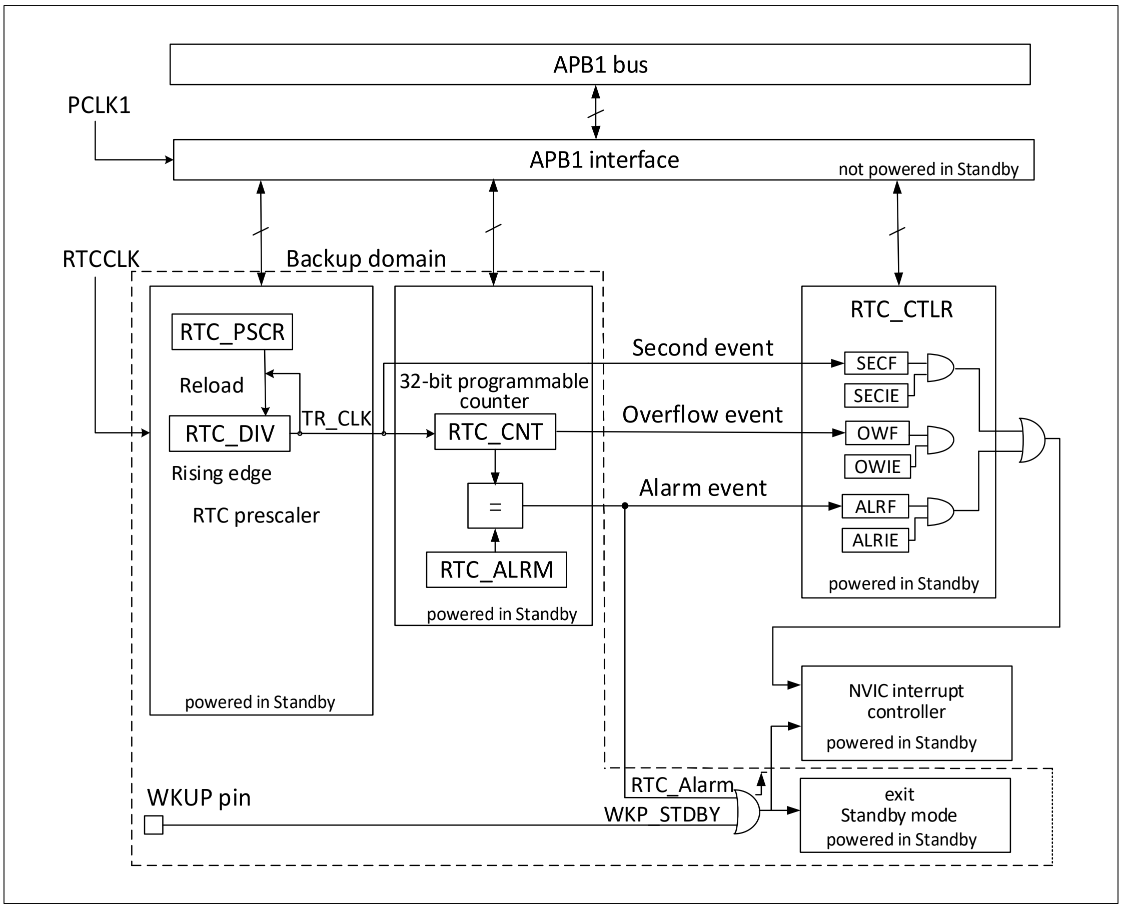
6.1 Main features

- Prescaler factor: up to 2^{20}
- 32-bit programmable counter
- Multiple types of clock sources, interrupt
- Independent reset

6.2 Functional description

6.2.1 Overview

Figure 6-1 RTC structure block diagram



As shown in Figure 6-1, the RTC module is mainly composed of 3 parts: APB1 bus interface, prescaler and counter, control and status registers. The prescaler and counter are in the backup area and are powered by V_{BAT}. After RTCCLK is input to the prescaler (RTC_DIV), it is divided into TR_CLK. It is worth noting that a self-decrement counter is located in the prescaler (RTC_DIV). When the self-decrement reaches the overflow, a TR_CLK is output. Then, take the preset value from the reload value register (RTC_PSCR) and reload it into the prescaler. Reading the prescaler actually means reading its real-time value (read only). The prescaler factor should be written to the reload value register (RTC_PSCR). Generally, the cycle of TR_CLK is set to 1s, TR_CLK triggers the second event, and the main counter (RTC_CNT) will be self-incremented by one at the same time. When the main counter is increased to the same value as the value of the alarm register, the alarm event is triggered. When the main counter is incremented till overflow, an overflow event is triggered. These 3 events can all trigger the interrupt, which is controlled by the corresponding interrupt enable bit.

6.2.2 Reset

Due to the special purposes of the real-time clock, the 4 sets of registers in the backup domain: prescaler, prescaler of reloading value, main counter and alarm clock. They can only be reset by the reset signal in the backup domain. Refer to the chapter of RCC backup domain reset. The control register of the real-time clock is controlled by system reset or power reset.

6.2.3 Special read/write register operations

Due to the special purpose of the real-time clock, the RTC and APB1 buses are independent, and the reading of RTC by APB1 is not necessarily real-time. The RTC register reading through APB1 must go through a RTC rising edge after APB1 is started up. This situation may occur after system reset and power reset, wake-up from standby or stop mode. It is convenient to wait when the RSF bit of the control register (CTLR) is set high. For the write operator of RTC, the operation must be made specifically according to the following steps in the configuration mode when the previous write operation ends:

1) Query RTOFF bit unit it changes to 1;
2) Set CNF bit to enter the configuration mode;
3) Perform write operation to one or more RTC registers;
4) Clear the CNF bit to exit the configuration mode, and the RTC register will start being written on the APB interface1;
5) Query the RTOFF bit until it changes to 1. So far, the write is completed;

6.3 Register description

Table 6-1 RTC registers

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

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.	0
1	ALRIE	RW	Alarm interrupt enable.	0
0	SECIE	RW	Second interrupt enable.	0

6.3.2 RTC Control Register Low (RTC_CTLRL)

Offset address: 0x04

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										RTOFF	CNF	RSF	OWF	ALRF	SECF

Bit	Name	Access	Description	Reset value
[15:6]	Reserved	RO	Reserved.	0
5	RTOFF	RO	RTC operation OFF. It indicates the status of the last write operation performed on the RTC. To perform operation on RTC, this bit must be 1. 1: Last write operation on RTC has been completed; 0: Last write operation on RTC is still ongoing.	1
4	CNF	RW	Configuration flag. Write 1 to this bit to enter the configuration mode, so as to allow to write to R16_RTC_CNTx, R16_RTC_ALRMx and R16_RTC_PSCRx. The write operation can be performed only when this bit is written 1 and cleared by software: 1: Enter the configuration mode; 0: Exit the configuration mode, and start updating the RTC register.	0
3	RSF	RW0	Registers synchronized flag. Ensure that it is set by hardware (registers synchronized) before read/write operation is performed on PSCRx, ALRMx and CNTx. When reading/ writing these registers, or after the APBI is reset or APB1 clock is stopped, the bit should be reset firstly. 1: Register synchronized; 0: Register not synchronized.	0
2	OWF	RW0	Counter overflow flag. When the 32-bit counter overflows, this bit is set by hardware. If the OWIE bit is set, an overflow interrupt is also generated.	0

			This bit can only be cleared by software and cannot be set by software.	
1	ALRF	RW0	Alarm flag. When the counter value reaches the value of the alarm register (ALRMx), this bit will be set by the hardware. If the alarm interrupt enable bit (ALRIE) is set, an alarm interrupt is also generated. This bit can only be cleared by software and cannot be set by software.	0
0	SECF	RW0	Second flag. When the clock generates a falling edge after divided by the prescaler, the counter will self-increase by 1 and generate a second event, and the bit will be set. If the second interrupt is enabled (SECIE bit is set), a second interrupt is also generated at the same time. This bit can only be cleared by software and cannot be set by software.	0

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	Prescaler reload register high.	0

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	Prescaler reload register low. The actual division factor is (PSCR[19:0]+1). For example, if the RTC input frequency is 32768Hz, then this value can be set to 0x7fff to divide the signal with cycle of 1s.	8000h

6.3.5 Prescaler 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	Prescaler divider register high.	0

6.3.6 Prescaler 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	Prescaler divider register low. DIV is actually a self-decrement counter. The DIV counter will decrease by 1 per incoming clock of RTC_CLK. After overflow, it will output a TR_CLK and reload the value from PSCR at the same time. DIV can only be read, and the remaining value of the counter of the current prescaler divider is read.	8000h

6.3.7 RTC Counter Register 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	RTC counter register high	0

6.3.8 RTC Counter Register 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	RTC counter register low. Core component of the RTC timer; the clock is provided with TRCLK (the cycle is generally set to 1 second). Calculate current time by reading CNT[31:0]. Enter the configuration mode to write this value.	0

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	FFFFh

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 register low. When the value of the alarm register ALRM[31:0] is consistent with the value of the counter CNT[31:0], an alarm event is generated. Enter the configuration mode to modify this value.	FFFFh

Chapter 7 Independent Watchdog (IWDG)

This chapter applies to the whole family of CH32F2x, CH32V2x and CH32V3x.

The system is equipped with an Independent Watchdog (IWDG) to detect logic errors and software faults caused by external environmental interference. The IWDG clock source comes from LSI, can run independently of the main program, and is suited for applications with low precision requirements.

7.1 Main features

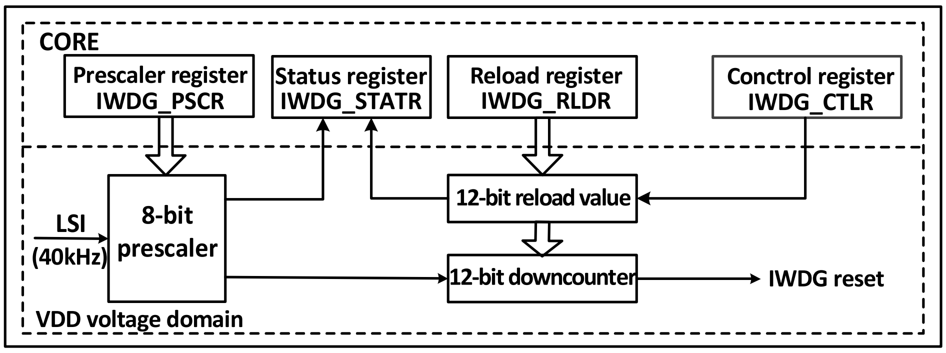
- 12-bit free-running downcounter
- Clocked from LSI clock divided, can run in low-power mode
- Reset condition: The downcounter value reaches 0

7.2 Functional specification

7.2.1 Principle and application

The clock source of the independent watchdog is LSI clock divided, and its function can still work normally in stop and standby mode. When the watchdog downcounter value reaches 0, a system reset is generated, so the timeout period is (reload value + 1) clocks.

Figure 7-1 Structure Block Diagram of Independent Watchdog


CORE
Prescaler register IWDG_PSCR
Status register IWDG_STATR
Reload register IWDG_RLDR
Conctrol register IWDG_CTLR
LSI (40kHz)
8-bit prescaler
12-bit reload value
12-bit downcounter
IWDG reset
VDD voltage domain

- Enable independent watchdog

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

If the hardware independent watchdog enable bit (IWDG_SW) is enabled in User Option Bytes, the IWDG is permanently enabled after the microcontroller reset.

- Watchdog configuration

A 12-bit downcounter is in the watchdog. When the value of downcounter is reduced to 0, the system reset occurs. To enable IWDG, the following operations are needed:

1) Counter time base: IWDG clock source is LSI. Set the LSI divider factor clock as IWDG counter time base through IWDG_PSCR register. Firstly write 0x5555 to the IWDG_CTLR register, and then modify the divider factor in the IWDG_PSCR register. The PVU bit in the IWDG_STATR register indicates the update status of the divider factor. The divider factor can only be modified and read after the update is finished.
2) Reload value: It is used to update the current value of the counter in the independent watchdog, and the counter counts down from this value. Firstly write 0x5555 to the IWDG_CTLR register, and then modify

the IWDG_RLDR register to set the target reload value. The RVU bit in the IWDG_STATR register indicates the update status of the reload value. The IWDG_RLDR register can only be modified and read after the update is finished.

3) Watchdog enable: Write 0xCCCC to the IWDG_CTLR register to enable the watchdog function.

4) Feed dog: The current counter value is refreshed before the watchdog downcounter reaches 0 to prevent the system reset. Write 0xAAAA to the IWDG_CTLR register to update the IWDG_RLDR register value to the watchdog counter by the hardware. This action needs to be executed regularly after the watchdog function is enabled. Otherwise, the watchdog reset action occurs.

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.

8.1 Main features

- Programmable 7-bit free-running downcounter
- Conditional reset
  - Reset when the current counter value is less than 0x40
  - Reset when the counter value is reloaded beyond the window time
- Early wake-up interrupt (EWI), used to feed the dog in time to prevent system reset

7.3 Register description

Table 7-1 IWDG 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	Operation key value lock. 0xAAAA: Feed dog. Load the IWDG_RLDR register value to the independent watchdog counter; 0x5555: Allow to modify R16_IWDG_PSCR and R16_IWDG_RLDR registers; 0xCCCC: Enable the watchdog. If the hardware watchdog is enabled (User Option Bytes	0

			configured), there is no such restriction.	

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 divider factor. Write 0x5555 to KEY before modifying this domain. 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 domain, make sure that the PVU bit in the IWDG_STATR register is 0. Otherwise, the read value is invalid.*	000b

7.3.3 Reload 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 domain. After writing 0xAAAA to KEY, the value of this field will be loaded into the counter by hardware, and then the counter will count down from this value. *Note: Before reading/writing the value of this domain, make sure that the RVU bit in the IWDG_STATR register is 0. Otherwise, read /write operation on this domain is invalid.*	FFFh

Note: This register is reset in standby mode.

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	Reload value update. Set and cleared by hardware. 1: Update of the reload value is ongoing; 0: Update of the reload update completed (5 LSI cycles at most). *Note: The reload value register IWDG_RLDR can only be read /written after the RVU bit is cleared to 0.*	0
0	PVU	RO	Clock prescaler value update. Set and cleared by hardware. 1: Update of the clock prescaler value is ongoing; 0: Update of the clock prescaler value completed (at most 5 LSI cycles). Note: The prescaler factor register (IWDG_PSCR) can only be read and written after the PVU bit is cleared to 0.	0

Note: After the prescale or reload value is updated, there is no need to wait for the RVU or PVU to reset, the following code can be continued. (Even in low-power mode, this write operation can still be performed.)

Chapter 8 Window Watchdog (WWDG)

This chapter applies to the whole family of CH32F2x, CH32V2x and CH32V3x.

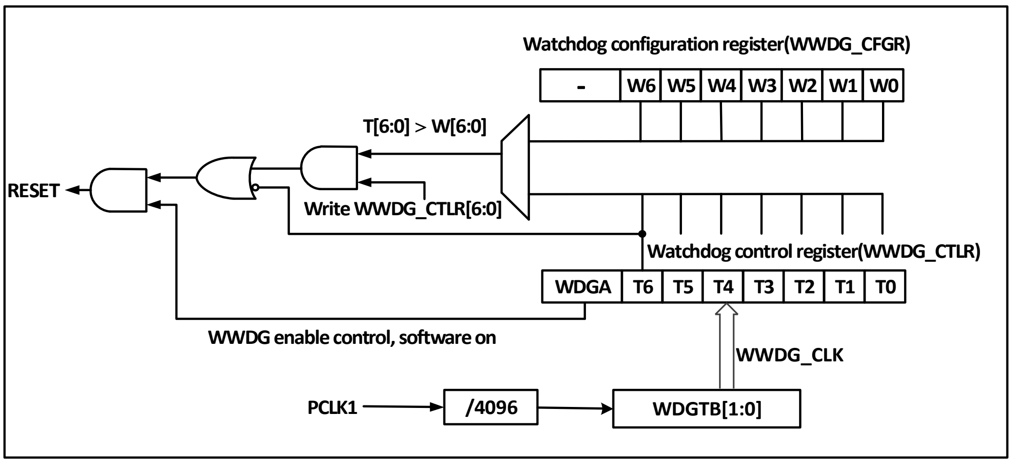
The Window Watchdog is generally used to monitor the software fault of the system operation, such as external interference and unforeseen logic errors. It needs to refresh the counter (feed the dog) within a specific window time (with upper and lower limits). Otherwise, the watchdog circuit will generate a system reset before or after this window time.

8.2 Function specification

8.2.1 Principle and application

The window watchdog runs based on a 7-bit downcounter, which is mounted under the APB1 bus. The counter time base WWDG_CLK originates from the clock prescaler (PCLK1/4096). The clock prescaler factor is set by the WDGTB[1:0] bits in the WWDG_CFGR register. The downcounter is at a free running status. No matter whether the watchdog function is enabled or not, the counter keeps counting down in a cycle. The internal structure block diagram of window watchdog is as shown in Figure 8-1.

Figure 8-1 Window Watchdog block diagram


Watchdog configuration register(WWDG_CFGR)
-
W6
W5
W4
W3
W2
W1
W0
T[6:0] > W[6:0]
RESET
Write WWDG_CTLR[6:0]
Watchdog control register(WWDG_CTLR)
WDGA
T6
T5
T4
T3
T2
T1
T0
WWDG enable control, software on
WWDG_CLK
PCLK1
/4096
WDGTB[1:0]

- 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_APB1PCENR register to suspend WWDG_CLK counting and indirectly stop the watchdog function. Or reset the WWDG module by setting the RCC_APB1PRSTR register, which is equivalent to the function of reset.

- Watchdog configuration

A 7-bit counter which continues downcounting in a cycle is in the watchdog, and it supports read/write access. To enable watchdog reset function, the following operations are needed:

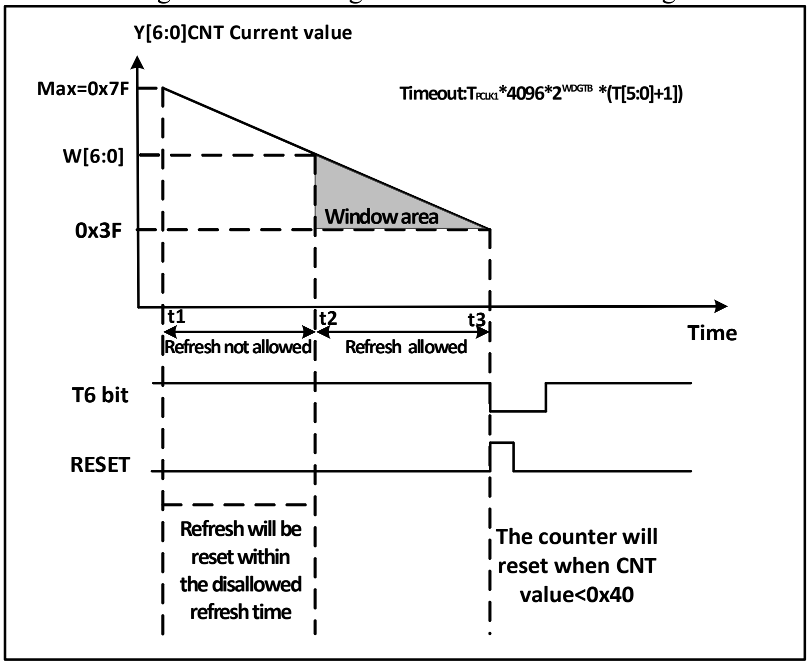
1) Counter time base: The WDGTB[1:0] bits in the WWDG_CFGR register. Note to switch on the WWDG module clock of the RCC unit.
2) Window counter: Set the W[6:0] bits in the WWDG_CFGR register. This counter is used to be compared with the current counter by hardware, the value is configured by the user 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


Y[6:0]CNT Current value
Max=0x7F
W[6:0]
0x3F
Timeout:T_{PCLK1}*4096*2^{WDGTB} *(T[5:0]+1])
Window area
t1
t2
t3
Time
Refresh not allowed
Refresh allowed
T6 bit
RESET
Refresh will be reset within the disallowed refresh time
The counter will reset when CNT value<0x40

- 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 $t_1 \le t \le t_2$, a "window watchdog reset" occurs, and a system reset occurs.

- Early wakeup

In order to avoid system reset caused by not refreshing the counter in time, the watchdog module provides early wake-up interrupt (EWI) notification. When the downcounter reaches 0x40, an early wake-up signal is generated and the EWIF flag is set to 1. If the EWI bit is set, the window watchdog interrupt is triggered at the same time. In this case, it takes single counter clock cycle to the hardware reset (downcounting to 0x3F), and the application can immediately perform feed dog operation within the time limit.

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 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	Window watchdog activation bit. 1:Watchdog enabled (reset signal can be generated); 0:Watchdog disabled. Set by software, and only cleared by hardware after reset.	0
[6:0]	T[6:0]	RW	7-bit downcounter. Decremented every (4096*2^{WDGTB}) PCLK1 cycles. When the counter rolls over from 0x40 to 0x3F, i.e., a watchdog reset is generated when T6 becomes 0.	7Fh

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: not divided, counter time base =PCLK1/4096;	00b

			01: divided by 2, counter time base =PCLK1/4096/2;
			10: divided by 4, counter time base = PCLK1/4096/4;
			11: divided by 8, counter time base = PCLK1/4096/8.	

[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.	7Fh

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 (NVIC/PFIC)

This chapter applies to the whole family of CH32F2x, CH32V2x and CH32V3x.

TheCH32F2x is based on the Cortex-M3 core. It has a built-in Nested Vectored Interrupt Controller (NVIC) to manage 88 maskable external interrupt channels and 10 core interrupt channels, and other interrupt sources are reserved. The interrupt controller is closely connected with the core interface, which provides a flexible interrupt management function with the minimum interrupt delay. For details about the NVIC controller, please refer to the related documents of Cortex-M3.

The CH32V2x and CH32V3x both have built-in Programmable Fast Interrupt Controller (PFIC), which supports up to 255 interrupt vectors. The current system manages 88 peripheral interrupt channels and 8 core interrupt channels, and others are reserved.

9.1 Main features

9.1.1 NVIC controller

- 88 maskable interrupt channels
- Quick response of non-maskable interrupt
- Vectorized interrupt, which implements that the vector entry address directly enters the core
- Automatic stack and recovery when the interrupt enters and exits, with no additional command overhead needed
- 16 nested levels, priority-level dynamically modified

9.1.2 PFIC controller

- 88 peripheral interrupts. each interrupt request has an independent trigger bit and a mask control bit, and has a dedicated status bit
- Programmable multi-level interrupt nested, up to 8 nested levels, 3 hardware stack levels
- Dedicated fast interrupt in/out mechanism, hardware automatic stack and recovery, no command overhead needed
- Vector Table Free (VTF) interrupt response mechanism, 4-channel programmable access interrupt vector address

9.2 System timer

- CH32F2x

The Cortex-M3 core is provided with a 24-bit downcounter (SysTick timer). It supports HCLK or HCLK/8 served as the time base and has a very high priority level (6). It is generally used to control time base of the system. Refer to related documents of Cortex-M3 for details.

- CH32V3x

The core is provided with a 64-bit downcounter (SysTick) that supports HCLK or HCLK/8 as the time base, has a higher priority and can be used for time base after calibration.

9.3 Vector table of interrupt and exception

Table 9-1 CH32F2x vector table

Position	Priority	Priority type	Name	Description	Absolute address
	-	-	-	Reserved	0x00000000
	-3	fixed	Reset	Reset	0x00000004
	-2	fixed	NMI	Non-maskable interrupt	0x00000008
	-1	fixed	HardFault	All class of fault	0x0000000C
	0	settable	MemManage	Memory management	0x00000010
	1	settable	BusFault	Prefetch fault, memory access fault	0x00000014
	2	settable	UsageFault	Undefined instruction or illegal state	0x00000018
	-	-	-	Reserved	0x0000001C-0x0000002B
	3	settable	SVCall	System service call via SWI instruction	0x0000002C
	4	settable	Debug Monitor	Debug monitor	0x00000030
	-	-	-	Reserved	0x00000034
	5	settable	PendSV	Pendable system service	0x00000038
	6	settable	SysTick	System tick timer	0x0000003C
0	7	programmable	WWDG	Window watchdog timer interrupt	0x00000040
1	8	programmable	PVD	PVD through EXTI Line detection	0x00000044
2	9	programmable	TAMPER	Tamper interrupt	0x00000048
3	10	programmable	RTC	Real-time clock (RTC) global interrupt	0x0000004C
4	11	programmable	FLASH	FLASH global interrupt	0x00000050
5	12	programmable	RCC	Reset and clock control (RCC) interrupt	0x00000054
6	13	programmable	EXTI0	EXTI line0 interrupt	0x00000058
7	14	programmable	EXTI1	EXTI line1 interrupt	0x0000005C
8	15	programmable	EXTI2	EXTI line2 interrupt	0x00000060
9	16	programmable	EXTI3	EXTI line3 interrupt	0x00000064
10	17	programmable	EXTI4	EXTI line4 interrupt	0x00000068
11	18	programmable	DMA1_CH1	DMA1 channel1 global interrupt	0x0000006C
12	19	programmable	DMA1_CH2	DMA1 channel2 global interrupt	0x00000070
13	20	programmable	DMA1_CH3	DMA1 channel3 global interrupt	0x00000074
14	21	programmable	DMA1_CH4	DMA1 channel4 global interrupt	0x00000078
15	22	programmable	DMA1_CH5	DMA1 channel5 global interrupt	0x0000007C
16	23	programmable	DMA1_CH6	DMA1 channel6 global interrupt	0x00000080
17	24	programmable	DMA1_CH7	DMA1 channel7 global interrupt	0x00000084
18	25	programmable	ADC1_2	ADC1 and ADC2 global interrupt	0x00000088
19	26	programmable	USB_HP or CAN1_TX	USB_HP or CAN1_TX global interrupt	0x0000008C
20	27	programmable	USB_LP or CAN1_RX0	USB_LP or CAN1_RX0 global interrupt	0x00000090
21	28	programmable	CAN1_RX1	CAN1_RX1 global interrupt	0x00000094
22	29	programmable	CAN1_SCE	CAN1_SCE global interrupt	0x00000098

23	30	programmable	EXTI9_5	EXTI line[9:5] interrupts	0x0000009C
24	31	programmable	TIM1_BRK	TIM1 break interrupt	0x000000A0
25	32	programmable	TIM1_UP	TIM1 update interrupt	0x000000A4
26	33	programmable	TIM1_TRG_COM	TIM1 trigger and communication interrupts	0x000000A8
27	34	programmable	TIM1_CC	TIM1 capture compare interrupt	0x000000AC
28	35	programmable	TIM2	TIM2 global interrupt	0x000000B0
29	36	programmable	TIM3	TIM3 global interrupt	0x000000B4
30	37	programmable	TIM4	TIM4 global interrupt	0x000000B8
31	38	programmable	I2C1_EV	I^2C1 event interrupt	0x000000BC
32	39	programmable	I2C1_ER	I^2C1 error interrupt	0x000000C0
33	40	programmable	I2C2_EV	I^2C2 event interrupt	0x000000C4
34	41	programmable	I2C2_ER	I^2C2 error interrupt	0x000000C8
35	42	programmable	SPI1	SPI1 global interrupt	0x000000CC
36	43	programmable	SPI2	SPI2 global interrupt	0x000000D0
37	44	programmable	USART1	USART1 global interrupt	0x000000D4
38	45	programmable	USART2	USART2 global interrupt	0x000000D8
39	46	programmable	USART3	USART3 global interrupt	0x000000DC
40	47	programmable	EXTI15_10	EXTI line[15:10] interrupts	0x000000E0
41	48	programmable	RTCAlarm	RTC alarm through EXTI line interrupt	0x000000E4
42	49	programmable	USBWakeUp	USB wake-up through EXTI line	0x000000E8
43	50	programmable	TIM8_BRK	TIM8 break interrupt	0x000000EC
44	51	programmable	TIM8_UP	TIM8 update interrupt	0x000000F0
45	52	programmable	TIM8_TRG_COM	TIM8 trigger and communication interrupts	0x000000F4
46	53	programmable	TIM8_CC	TIM8 capture compare interrupt	0x000000F8
47	54	programmable	RNG	RNG global interrupt	0x000000FC
48	55	programmable	FSMC	FSMC global interrupt	0x00000100
49	56	programmable	SDIO	SDIO global interrupt	0x00000104
50	57	programmable	TIM5	TIM5 global interrupt	0x00000108
51	58	programmable	SPI3	SPI3 global interrupt	0x0000010C
52	59	programmable	UART4	UART4 global interrupt	0x00000110
53	60	programmable	UART5	UART5 global interrupt	0x00000114
54	61	programmable	TIM6	TIM6 global interrupt	0x00000118
55	62	programmable	TIM7	TIM7 global interrupt	0x0000011C
56	63	programmable	DMA2_CH1	DMA2 channel1 global interrupt	0x00000120
57	64	programmable	DMA2_CH2	DMA2 channel2 global interrupt	0x00000124
58	65	programmable	DMA2_CH3	DMA2 channel3 global interrupt	0x00000128
59	66	programmable	DMA2_CH4	DMA2 channel4 global interrupt	0x0000012C
60	67	programmable	DMA2_CH5	DMA2 channel5 global interrupt	0x00000130
61	68	programmable	ETH	ETH global interrupt	0x00000134
62	69	programmable	ETH_WKUP	ETH wakeup interrupt	0x00000138
63	70	programmable	CAN2_TX	CAN2_TX global interrupt	0x0000013C
64	71	programmable	CAN2_RX0	CAN2_RX0 global interrupt	0x00000140

65	72	programmable	CAN2_RX1	CAN2_RX1 global interrupt	0x00000144
66	73	programmable	CAN2_SCE	CAN2_SCE global interrupt	0x00000148
67	74	programmable	OTG_FS	Full-speed OTG interrupt	0x0000014C
68	75	programmable	USBHSWakeUp	High-speed USB wakeup interrupt	0x00000150
69	76	programmable	USBHS	High-speed USB global interrupt	0x00000154
70	77	programmable	DVP	DVP global interrupt	0x00000158
71	78	programmable	UART6	UART6 global interrupt	0x0000015C
72	79	programmable	UART7	UART7 global interrupt	0x00000160
73	80	programmable	UART8	UART8 global interrupt	0x00000164
74	81	programmable	TIM9_BRK	TIM9 break interrupt	0x00000168
75	82	programmable	TIM9_UP	TIM9 update interrupt	0x0000016C
76	83	programmable	TIM9_TRG_COM	TIM9 trigger and communication interrupts	0x00000170
77	84	programmable	TIM9_CC	TIM9 capture compare interrupt	0x00000174
78	85	programmable	TIM10_BRK	TIM10 break interrupt	0x00000178
79	86	programmable	TIM10_UP	TIM10 update interrupt	0x0000017C
80	87	programmable	TIM10_TRG_COM	TIM10 trigger and communication interrupts	0x00000180
81	88	programmable	TIM10_CC	TIM10 capture compare interrupt	0x00000184
82	89	programmable	DMA2_CH6	DMA2 channel6 global interrupt	0x00000188
83	90	programmable	DMA2_CH7	DMA2 channel7 global interrupt	0x0000018C
84	91	programmable	DMA2_CH8	DMA2 channel8 global interrupt	0x00000190
85	92	programmable	DMA2_CH9	DMA2 channel9 global interrupt	0x00000194
86	93	programmable	DMA2_CH10	DMA2 channel10 global interrupt	0x00000198
87	94	programmable	DMA2_CH11	DMA2 channel11 global interrupt	0x0000019C

Table 9-2 CH32V2x and CH32V3x vector table

No.	Priority	Type	Name	Description	Entry address
0	-	-	-	-	0x00000000
1	-	-	-	-	0x00000004
2	-5	fixed	NMI	Non-maskable interrupt	0x00000008
3	-4	fixed	HardFault	Exception interrupt	0x0000000C
4	-	-	-	Reserved	0x00000010
5	-3	fixed	Ecall-M	Callback interrupt in machine mode	0x00000014
6-7	-	-	-	Reserved	0x00000018 to 0x0000001C
8	-2	fixed	Ecall-U	Callback interrupt in user mode	0x00000020
9	-1	fixed	BreakPoint	Breakpoint callback interrupt	0x00000024
10-11	-	-	-	Reserved	0x00000028 to 0x0000002C
12	0	programmable	SysTick	System timer interrupt	0x00000030
13	-	-	-	Reserved	0x00000034
14	1	programmable	SW	Software interrupt	0x00000038
15	-	-	-	Reserved	0x0000003C

16	2	programmable	WWDG	Window timer interrupt	0x00000040
17	3	programmable	PVD	Power voltage detector interrupt (EXTI)	0x00000044
18	4	programmable	TAMPER	Tamper interrupt	0x00000048
19	5	programmable	RTC	Real-time clock interrupt	0x0000004C
20	6	programmable	FLASH	Flash memory global interrupt	0x00000050
21	7	programmable	RCC	Reset and clock control interrupt	0x00000054
22	8	programmable	EXTI0	EXTI line0 interrupt	0x00000058
23	9	programmable	EXTI1	EXTI line1 interrupt	0x0000005C
24	10	programmable	EXTI2	EXTI line2 interrupt	0x00000060
25	11	programmable	EXTI3	EXTI line3 interrupt	0x00000064
26	12	programmable	EXTI4	EXTI line4 interrupt	0x00000068
27	13	programmable	DMA1_CH1	DMA1 channel1 global interrupt	0x0000006C
28	14	programmable	DMA1_CH2	DMA1 channel2 global interrupt	0x00000070
29	15	programmable	DMA1_CH3	DMA1 channel3 global interrupt	0x00000074
30	16	programmable	DMA1_CH4	DMA1 channel4 global interrupt	0x00000078
31	17	programmable	DMA1_CH5	DMA1 channel5 global interrupt	0x0000007C
32	18	programmable	DMA1_CH6	DMA1 channel6 global interrupt	0x00000080
33	19	programmable	DMA1_CH7	DMA1 channel7 global interrupt	0x00000084
34	20	programmable	ADC1_2	ADC1 and ADC2global interrupt	0x00000088
35	21	programmable	USB_HP or CAN1_TX	USB_HP or CAN1_TXglobal interrupt	0x0000008C
36	22	programmable	USB_LP or CAN1_RX0	USB_LP or CAN1_RX0global interrupt	0x00000090
37	23	programmable	CAN1_RX1	CAN1_RX1global interrupt	0x00000094
38	24	programmable	CAN1_SCE	CAN1_SCEglobal interrupt	0x00000098
39	25	programmable	EXTI9_5	EXTI line[9:5] interrupts	0x0000009C
40	26	programmable	TIM1_BRK	TIM1 break interrupt	0x000000A0
41	27	programmable	TIM1_UP	TIM1 update interrupt	0x000000A4
42	28	programmable	TIM1_TRG_COM	TIM1 trigger and communication interrupts	0x000000A8
43	29	programmable	TIM1_CC	TIM1 capture compare interrupt	0x000000AC
44	30	programmable	TIM2	TIM2 global interrupt	0x000000B0
45	31	programmable	TIM3	TIM3 global interrupt	0x000000B4
46	32	programmable	TIM4	TIM4 global interrupt	0x000000B8
47	33	programmable	I2C1_EV	I^2C1 event interrupt	0x000000BC
48	34	programmable	I2C1_ER	I^2C1 error interrupt	0x000000C0
49	35	programmable	I2C2_EV	I^2C2 event interrupt	0x000000C4
50	36	programmable	I2C2_ER	I^2C2 error interrupt	0x000000C8
51	37	programmable	SPI1	SPI1 global interrupt	0x000000CC
52	38	programmable	SPI2	SPI2 global interrupt	0x000000D0
53	39	programmable	USART1	USART1 global interrupt	0x000000D4
54	40	programmable	USART2	USART2 global interrupt	0x000000D8
55	41	programmable	USART3	USART3 global interrupt	0x000000DC
56	42	programmable	EXTI15_10	EXTI line[15:10] interrupts	0x000000E0

57	43	programmable	RTCAlarm	RTC alarm interrupt (EXTI)	0x000000E4
58	44	programmable	USBWakeUp	USB wakeup interrupt (EXTI)	0x000000E8
59	45	programmable	TIM8_BRK	TIM8 break interrupt	0x000000EC
60	46	programmable	TIM8_UP	TIM8 update interrupt	0x000000F0
61	47	programmable	TIM8_TRG_COM	TIM8 trigger and communication interrupts	0x000000F4
62	48	programmable	TIM8_CC	TIM8 capture compare interrupt	0x000000F8
63	49	programmable	RNG	RNG global interrupt	0x000000FC
64	50	programmable	FSMC	FSMC global interrupt	0x00000100
65	51	programmable	SDIO	SDIO global interrupt	0x00000104
66	52	programmable	TIM5	TIM5 global interrupt	0x00000108
67	53	programmable	SPI3	SPI3 global interrupt	0x0000010C
68	54	programmable	UART4	UART4 global interrupt	0x00000110
69	55	programmable	UART5	UART5 global interrupt	0x00000114
70	56	programmable	TIM6	TIM6 global interrupt	0x00000118
71	57	programmable	TIM7	TIM7 global interrupt	0x0000011C
72	58	programmable	DMA2_CH1	DMA2 channel1 global interrupt	0x00000120
73	59	programmable	DMA2_CH2	DMA2 channel2 global interrupt	0x00000124
74	60	programmable	DMA2_CH3	DMA2 channel3 global interrupt	0x00000128
75	61	programmable	DMA2_CH4	DMA2 channel4 global interrupt	0x0000012C
76	62	programmable	DMA2_CH5	DMA2 channel5 global interrupt	0x00000130
77	63	programmable	ETH	ETH global interrupt	0x00000134
78	64	programmable	ETH_WKUP	ETH wakeup interrupt	0x00000138
79	65	programmable	CAN2_TX	CAN2_TX global interrupt	0x0000013C
80	66	programmable	CAN2_RX0	CAN2_RX0 global interrupt	0x00000140
81	67	programmable	CAN2_RX1	CAN2_RX1 global interrupt	0x00000144
82	68	programmable	CAN2_SCE	CAN2_SCE global interrupt	0x00000148
83	69	programmable	OTG_FS	Full-speed OTG interrupt	0x0000014C
84	70	programmable	USBHSWakeUp	High-speed USB wakeup interrupt	0x00000150
85	71	programmable	USBHS	High-speed USB global interrupt	0x00000154
86	72	programmable	DVP	DVP global interrupt	0x00000158
87	73	programmable	UART6	UART6 global interrupt	0x0000015C
88	74	programmable	UART7	UART7 global interrupt	0x00000160
89	75	programmable	UART8	UART8 global interrupt	0x00000164
90	76	programmable	TIM9_BRK	TIM9 break interrupt	0x00000168
91	77	programmable	TIM9_UP	TIM9 update interrupt	0x0000016C
92	78	programmable	TIM9_TRG_COM	TIM9 trigger and communication interrupts	0x00000170
93	79	programmable	TIM9_CC	TIM9 capture compare interrupt	0x00000174
94	80	programmable	TIM10_BRK	TIM10 break interrupt	0x00000178
95	81	programmable	TIM10_UP	TIM10 update interrupt	0x0000017C
96	82	programmable	TIM10_TRG_COM	TIM10 trigger and communication interrupts	0x00000180
97	83	programmable	TIM10_CC	TIM10 capture compare interrupt	0x00000184
98	84	programmable	DMA2_CH6	DMA2 channel6 global interrupt	0x00000188

99	85	programmable	DMA2_CH7	DMA2 channel7 global interrupt	0x0000018C
100	86	programmable	DMA2_CH8	DMA2 channel8 global interrupt	0x00000190
101	87	programmable	DMA2_CH9	DMA2 channel9 global interrupt	0x00000194
102	88	programmable	DMA2_CH10	DMA2 channel10 global interrupt	0x00000198
103	89	programmable	DMA2_CH11	DMA2 channel11 global interrupt	0x0000019C

9.4 External interrupt and event controller (EXTI)

9.4.1 Overview

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

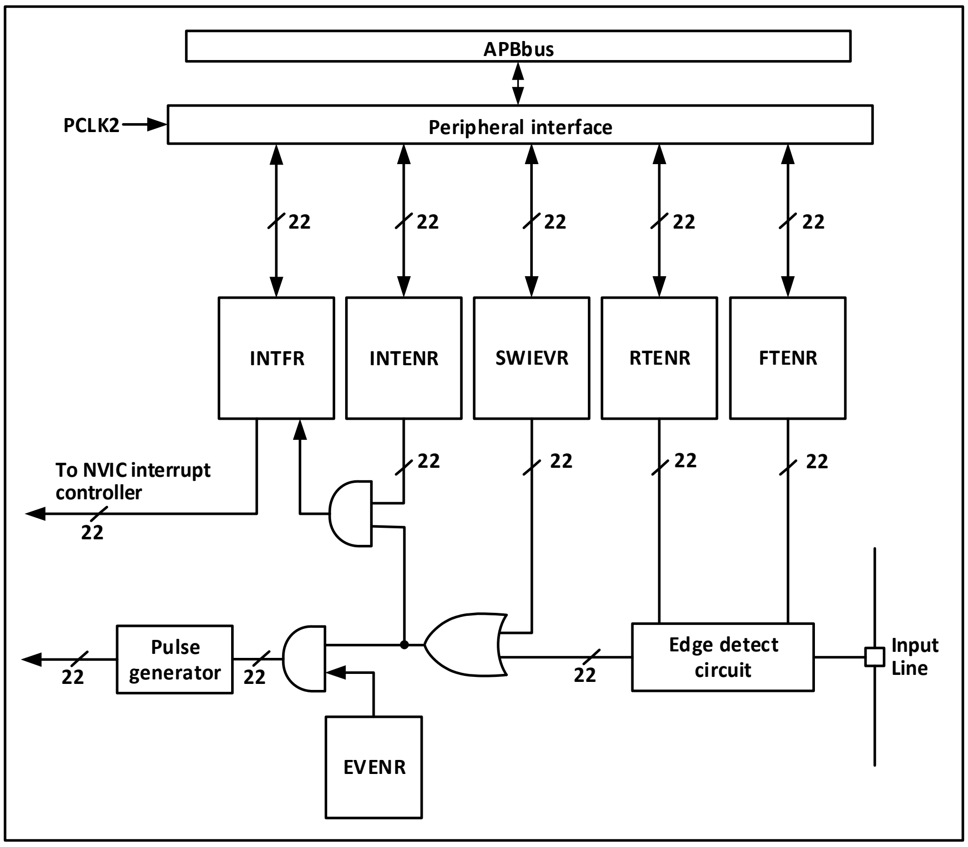


Figure shown in Figure 9-1, the trigger source of an external interrupt can be either a software interrupt (SWIEVR) or an actual external interrupt channel. The signal of the external interrupt channel is firstly filtered by the edge detect circuit. As long as either soft interrupt or external interrupt signal is generated, it is output to dual AND gate circuits of event enable and interrupt enable through the OR circuit in the figure. As long as the interrupt or the event is enabled, an interrupt or event is generated. 6 registers of EXTI are accessed by the processer through APB2 interface.

9.4.2 Wakeup event

The system can wake up from sleep mode caused by WFE command via wake-up event. The wake-up event is generated through the following 2 types of configuration:

- Enable an interrupt in the peripheral register, but do not enable the interrupt in the NVIC or PEIC of the core, and enable the SEVONPEND bit in the core at the same time. Reflected in EXTI, EXTI interrupt is enabled, but EXTI interrupt in NVIC or PEIC is not enabled, at the same time, enable the SEVONPEND bit. When the CPU is woken up from WFE, the EXTI interrupt flag bit and NVIC/PEIC pending bit need to be cleared.

- Enable an EXTI channel as an event channel. It is not necessary to clear the interrupt flag bit and NVIC/PEIC pending bit after the CPU is woken up from WFE.

9.4.3 Description

To use the external interrupt, it is needed to configure the corresponding external interrupt channel, i.e., select the corresponding trigger edge and enable the interrupt. When the set trigger edge appears on the external interrupt channel, an interrupt request is generated and the corresponding interrupt flag bit is also set. Write 1 to the flag bit to clear such flag bit.

Steps to use external hardware interrupt:
1) Configure GPIO;
2) Configure the EXTI_INTENR bit in the corresponding external interrupt channel;
3) Configure the trigger edge (EXTI_RTENR or EXTI_FTENR), select rising edge trigger, falling edge trigger or double edges trigger;
4) Configure the EXTI interrupt in the NVIC/PFIC of the core to ensure that it can respond correctly.

Steps to use external hardware event:
1) Configure GPIO;
2) Configure the EXTI_EVENR bit in the corresponding external interrupt channel;
3) Configure the trigger edge (EXTI_RTENR or EXTI_FTENR), select rising edge trigger, falling edge trigger or double edges trigger.

Steps to use software interrupt/event:
1) Enable external interrupt (EXTI_INTENR) or external event (EXTI_EVENR);
2) To use the interrupt service function, set the EXTI interrupt in the NVIC/PEIC of the core;
3) Set the software interrupt trigger (EXTI_SWIEVR) to generate an interrupt.

9.4.4 External event map

Table 9-3 EXTI interrupt map

External interrupt/event line	Mapping event description
EXTI0～EXTI15	Px0 ～ Px15 (x=A/B/C/D/E). Any IO port can enable the external interrupt/event function, configured by AFIO_EXTICRx register.
EXTI16	PVD event: Exceed the voltage detector threshold
EXTI17	RTC alarm event
EXTI18	USBD/USBFSOTG wakeup event (applied for CH32F20x_D8, CH32F20x_D8C, CH32V30x_D8 and CH32V30x_D8C) USBD wakeup event (applied for other product numbers)
EXTI19	ETH wakeup event
EXTI20	USBHS wakeup event (applied for *CH32F20x_D8C and CH32V30x_D8C)* USBFS wakeup event (applied for other product numbers)
EXTI21	Internal 32K calibration wakeup event (applied for *CH32V20x_D8, CH32V20x_D8W and CH32F20x_D8W)*

9.5 Register description

9.5.1 EXTI registers

Table 9-4 EXTI 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											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
[31:20]	Reserved	RO	Reserved.	0
[20:0]	MRx	RW	Interrupt request signal of external interrupt channelx enable: 1: Interrupt enabled; 0: Interrupt disabled.	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											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
[31:21]	Reserved	RO	Reserved.	0
[20:0]	MRx	RW	Event request signal of external interrupt channelx enable: 1: Event enabled; 0: Event disabled.	0

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											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
[31:21]	Reserved	RO	Reserved.	0
[20:0]	TRx	RW	Rising edge trigger of external interrupt channelx enable: 1: Rising edge trigger enabled; 0: Rising edge trigger disabled.	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											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
[31:21]	Reserved	RO	Reserved	0
[20:0]	TRx	RW	Falling edge trigger of external interrupt channelx enable: 0: Falling edge trigger disabled; 1: Falling edge trigger enabled.	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 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	SWIER14	SWIER13	SWIER12	SWIER11	SWIER10	SWIER9	SWIER8	SWIER7	SWIER6	SWIER5	SWIER4	SWIER3	SWIER2	SWIER1	SWIER0

Bit	Name	Access	Description	Reset value
[31:21]	Reserved	RO	Reserved.	0
[20:0]	SWIERx	RW	Set a software interrupt on the corresponding external trigger interrupt channel. With these bits set, the corresponding bit of the interrupt flag bit	0

			(EXTI_INTFR) is set. If the interrupt (EXTI_INTENR) is enabled or the event (EXTI_EVENR) is enabled, an interrupt or event is generated.	

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											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
[31:21]	Reserved	RO	Reserved.	0
[20:0]	IFx	W1	Interrupt flag. When these bits are set, the corresponding external interrupt has occurred. Write 1 to clear it.	X

9.5.2 PFIC registers

Table 9-5 PFIC registers

Name	Access address	Description	Reset value
R32_PFIC_ISR1	0xE000E000	PFIC interrupt enable status register1	0x0000000C
R32_PFIC_ISR2	0xE000E004	PFIC interrupt enable status register2	0x00000000
R32_PFIC_ISR3	0xE000E008	PFIC interrupt enable status register3	0x00000000
R32_PFIC_ISR4	0xE000E00C	PFIC interrupt enable status register4	0x00000000
R32_PFIC_IPR1	0xE000E020	PFIC interrupt pending status register1	0x00000000
R32_PFIC_IPR2	0xE000E024	PFIC interrupt pending status register2	0x00000000
R32_PFIC_IPR3	0xE000E028	PFIC interrupt pending status register3	0x00000000
R32_PFIC_IPR4	0xE000E02C	PFIC interrupt pending status register4	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	0x00000000
R32_PFIC_VTFADDRR0	0xE000E060	PFIC VTF interrupt0 offset address register	0x00000000
R32_PFIC_VTFADDRR1	0xE000E064	PFIC VTF interrupt1 offset address register	0x00000000

R32_PFIC_VTFADDRR2	0xE000E068	PFIC VTF interrupt2 offset address register	0x00000000
R32_PFIC_VTFADDRR3	0xE000E06C	PFIC VTF interrupt3 offset address register	0x00000000
R32_PFIC_IENR1	0xE000E100	PFIC interrupt enable set register1	0x00000000
R32_PFIC_IENR2	0xE000E104	PFIC interrupt enable set register2	0x00000000
R32_PFIC_IENR3	0xE000E108	PFIC interrupt enable set register3	0x00000000
R32_PFIC_IENR4	0xE000E10C	PFIC interrupt enable set register4	0x00000000
R32_PFIC_IRER1	0xE000E180	PFIC interrupt enable reset register1	0x00000000
R32_PFIC_IRER2	0xE000E184	PFIC interrupt enable reset register2	0x00000000
R32_PFIC_IRER3	0xE000E188	PFIC interrupt enable reset register3	0x00000000
R32_PFIC_IRER4	0xE000E18C	PFIC interrupt enable reset register4	0x00000000
R32_PFIC_IPSR1	0xE000E200	PFIC interrupt pending set register1	0x00000000
R32_PFIC_IPSR2	0xE000E204	PFIC interrupt pending set register2	0x00000000
R32_PFIC_IPSR3	0xE000E208	PFIC interrupt pending set register3	0x00000000
R32_PFIC_IPSR4	0xE000E20C	PFIC interrupt pending set register4	0x00000000
R32_PFIC_IPRR1	0xE000E280	PFIC interrupt pending reset register1	0x00000000
R32_PFIC_IPRR2	0xE000E284	PFIC interrupt pending reset register2	0x00000000
R32_PFIC_IPRR3	0xE000E288	PFIC interrupt pending reset register3	0x00000000
R32_PFIC_IPRR4	0xE000E28C	PFIC interrupt pending reset register4	0x00000000
R32_PFIC_IACTR1	0xE000E300	PFIC interrupt activation register1	0x00000000
R32_PFIC_IACTR2	0xE000E304	PFIC interrupt activation register2	0x00000000
R32_PFIC_IACTR3	0xE000E308	PFIC interrupt activation register3	0x00000000
R32_PFIC_IACTR4	0xE000E30C	PFIC interrupt activation register4	0x00000000
R32_PFIC_IPRIORx	0xE000E400	PFIC interrupt priority configuration register	0x00000000
R32_PFIC_SCTLR	0xE000ED10	PFIC system control register	0x00000000

Note: 1. Interrupts of NMI, EXC, ECALL-M, ECALL-U and BREAKPOINT are always enabled by default.

2. ECALL-M, ECALL-U and BREAKPOINT each is 1 case of EXC. The status is indicated by bit3 in the EXC.

3. NMI and EXC both support interrupt pending clear and set operation, but they do not support interrupt enable clear and set operation.

4. ECALL-M, ECALL-U and BREAKPOINT do not support interrupt pending clear and set operation or interrupt enable clear and set operation.

9.5.2.1 PFIC Interrupt Enable Status Register1 (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	INTENSTA3	INTENSTA2	Reserved

Bit	Name	Access	Description	Reset value
[31:12]	INTENSTA	RO	Enable status of 12# to 31# interrupts. 1: Interrupt enabled; 0: Interrupt disabled.	0
[11:4]	Reserved	RO	Reserved	0
[3:2]	INTENSTA	RO	Enable status of 2# to 3# interrupts. 1: Interrupt enabled; 0: Interrupt disabled.	0
[1:0]	Reserved	RO	Reserved	0

9.5.2.2 PFIC Interrupt Enable Status Register2 (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	Enable status of 32# to 63# interrupts: 1: Interrupt enabled; 0: Interrupt disabled.	0

9.5.2.3 PFIC Interrupt Enable Status Register3 (PFIC_ISR3)

Offset address: 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INTENSTA[95:80]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INTENSTA[79:64]															

Bit	Name	Access	Description	Reset value
[31:0]	INTENSTA	RO	Enable status of 64# to 95# interrupts: 1: Interrupt enabled; 0: Interrupt disabled.	0

9.5.2.4 PFIC Interrupt Enable Status Register4 (PFIC_ISR4)

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								INTENSTA[103:96]							

Bit	Name	Access	Description	Reset value
[31:8]	Reserved	R0	Reserved	0
[7:0]	INTENSTA	RO	Enable status of 96# to 103# interrupts: 1: Interrupt enabled; 0: Interrupt disabled.	0

9.5.2.5 PFIC Interrupt Pending Status Register1 (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								PENDSTA3	PENDSTA2	Reserved	

Bit	Name	Access	Description	Reset value
[31:12]	PENDSTA	RO	Pending status of 12# to 31# interrupts: 1: Interrupt pended; 0: Interrupt not pended.	0
[11:4]	Reserved	RO	Reserved	0
[3:2]	PENDSTA	RO	Pending status of 2# to 3# interrupts. 1: Interrupt pended; 0: Interrupt not pended.	0
[1:0]	Reserved	RO	Reserved	0

9.5.2.6 PFIC Interrupt Pending Status Register2 (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	Pending status of 32# to 63# interrupts: 1: Interrupt pended; 0: Interrupt not pended.	0

9.5.2.7 PFIC Interrupt Pending Status Register3 (PFIC_IPR3)

Offset address: 0x28

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PENDSTA[95:80]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PENDSTA[79:64]															

Bit	Name	Access	Description	Reset value
[31:0]	PENDSTA	RO	Pending status of 64# to 95# interrupts: 1: Interrupt pended; 0: Interrupt not pended.	0

9.5.2.8 PFIC Interrupt Pending Status Register4 (PFIC_IPR4)

Offset address: 0x2C

31	30	29	28	27	26	25	24	23	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[103:96]							

Bit	Name	Access	Description	Reset value
[31:8]	Reserved	RO	Reserved	0
[7:0]	PENDSTA	RO	Pending status of 96# to 103# interrupts: 1: Interrupt pended; 0: Interrupt not pended.	0

9.5.2.9 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. If the interrupt priority value is lower than the current set value, interrupt service is not performed when pended. When this register is 0, it means that the threshold register function is invalid. [7:4]: Priority threshold.	0

			[3:0]: Reserved. Fixed to 0. Invalid if writing.	

9.5.2.10 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								RSTSYS	Reserved						

Bit	Name	Access	Description	Reset value

[31:16]	KEYCODE	WO	Correspond to different target control bits. These bits can only be modified when the corresponding security access identification data is written synchronously, and the read data is fixed to 0.
			KEY1 = 0xFA05;
			KEY2 = 0xBCAF;
			KEY3 = 0xBEEF.
				0

[15:8]	Reserved	RO	Reserved.	0
7	RSTSYS	WO	System reset (write to KEY3 synchronously). Cleared automatically. Valid when writing 1, while invalid when writing 0. *Note: It has the same function as that of the SYSRESET bit in the PFIC_SCTLR register.*	0
[6:0]	Reserved	RO	Reserved.	0

9.5.2.11 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**
Reserved						GPENDSTA	GACTSTA	NESTSTA[7:0]							

Bit	Name	Access	Description	Reset value
[31:10]	Reserved	RO	Reserved.	0
9	GPENDSTA	RO	Whether there is pending interrupt currently: 1: Yes; 0: No.	0
8	GACTSTA	RO	Whether the interrupt is executed currently: 1: Yes; 0: No.	0

[7:0]	NESTSTA	RO	Current interrupt nested status. 8 nested levels currently supported, up to 3 hardware stack levels. If the set nested depth is greater than 3, the lower 3 interrupts should be configured as hardware stack, and other interrupts are configured as software stack.
			0xFF: Level 8 interrupt;
			0x7F: Level 7 interrupt;
			0x3F: Level 6 interrupt;
			0x1F: Level 5 interrupt;
			0x0F: Level 4 interrupt;
			0x07: Level 3 interrupt;
			0x03: Level 2 interrupt;
			0x01: Level 1 interrupt;
			0x00: No interrupt;
			Others: Impossible case.
			Note: Applicable to QingKeV4F core: CH32V30x_D8, CH32V30x_D8C.
			
			Current interrupt nesting status, currently supports a maximum of 2 levels of nesting and a maximum hardware stack depth of 2 levels.
			0x03: Level 2 interrupt.
			0x01: Level 1 interrupt.
			0x00: No interrupt.
			Others: Impossible case.
			Note: Applicable to QingKeV4B and V4C cores: CH32V20x_D6, CH32V20x_D8, CH32V20x_D8W.
				0

9.5.2.12 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 interrupt3 ID	0
[23:16]	VTFID2	RW	Configure VTF interrupt2 ID	0
[15:8]	VTFID1	RW	Configure VTF interrupt1 ID	0
[7:0]	VTFID0	RW	Configure VTF interrupt0 ID	0

9.5.2.13 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 interrupt0 service program address. Bit0 is 0.	0
0	VTF0EN	RW	VTF interrupt0 enable bit: 1: VTF interrupt0 channel enabled; 0: Disabled.	0

9.5.2.14 PFIC VTF Interrupt1 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 interrupt1 service program address. Bit0 is 0.	0
0	VTF1EN	RW	VTF interrupt1 enable bit: 1: VTF interrupt1 channel enabled; 0: Disabled.	0

9.5.2.15 PFIC VTF Interrupt2 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 interrupt2 service program address	0

			bit[31:1]. Bit0 is 0.	
0	VTF2EN	RW	VTF interrupt2 enable bit: 1: VTF interrupt2 channel enabled; 0: Disabled.	0

9.5.2.16 PFIC VTF Interrupt3 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 interrupt3 service program address bit[31:1]. Bit0 is 0.	0
0	VTF3EN	RW	VTF interrupt3 enable bit: 1: VTF interrupt3 channel enabled; 0: Disabled.	0

9.5.2.17 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# to 31# interrupts enable: 1: Interrupt enabled; 0: No effect.	0
[11:0]	Reserved	RO	Reserved	0

9.5.2.18 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# to 63# interrupts enable: 1: Interrupt enabled; 0: No effect.	0

9.5.2.19 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
INTEN[95:80]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INTEN[79:64]															

Bit	Name	Access	Description	Reset value
[31:0]	INTEN	WO	64# to 95# interrupts enable: 1: Interrupt enabled; 0: No effect.	0

9.5.2.20 PFIC Interrupt Enable Set Register 4 (PFIC_IENR4)

Offset address: 0x10C

31	30	29	28	27	26	25	24	23	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[103:96]							

Bit	Name	Access	Description	Reset value
[31:8]	Reserved	RO	Reserved	0
[7:0]	INTEN	WO	96# to 103# interrupts enable: 1: Interrupt enabled; 0: No effect.	0

9.5.2.21 PFIC Interrupt Enable Reset Register 1 (PFIC_IRER1)

Offset address: 0x180

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INTRESET[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# to 31# interrupts reset: 1: Interrupt reset; 0: No effect.	0
[11:0]	Reserved	RO	Reserved	0

9.5.2.22 PFIC Interrupt Enable Reset 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# to 63# interrupts reset: 1: Interrupt reset; 0: No effect.	0

9.5.2.23 PFIC Interrupt Enable Reset Register 3 (PFIC_IRER3)

Offset address: 0x188

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INTRSET[95:80]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INTRSET [79:64]															

Bit	Name	Access	Description	Reset value
[31:0]	INTRSET	WO	64# to 95# interrupts reset: 1: Interrupt reset; 0: No effect.	0

9.5.2.24 PFIC Interrupt Enable Reset Register 4 (PFIC_IRER4)

Offset address: 0x18C

31	30	29	28	27	26	25	24	23	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								INTRST [103:96]							

Bit	Name	Access	Description	Reset value
[31:8]	Reserved	RO	Reserved	0
[7:0]	INTRST	WO	96# to 103# interrupts reset: 1: Interrupt reset; 0: No effect.	0

9.5.2.25 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**
PENDSET15	PENDSET14	PENDSET13	PENDSET12	Reserved								PENDSET3	PENDSET2	Reserved	

Bit	Name	Access	Description	Reset value
[31:12]	PENDSET	WO	12# to 31# interrupts pending set. 13# and 15# interrupts are reserved. 1: Interrupt pended; 0: No effect.	0
[11:4]	Reserved	R0	Reserved	0
[3:2]	PENDSET	WO	2# to 3# interrupts pending set. 1: Interrupt pended; 0: No effect.	0
[1:0]	Reserved	RO	Reserved	0

9.5.2.26 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# to 63# interrupts pending set. 1: Interrupt pended; 0: No effect.	0

9.5.2.27 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
PENDSET[95:80]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PENDSET[79:64]															

Bit	Name	Access	Description	Reset value
[31:0]	PENDSET	WO	64# to 95# interrupts pending set. 1: Interrupt pended; 0: No effect.	0

9.5.2.28 PFIC Interrupt Pending Set Register 4 (PFIC_IPSR4)

Offset address: 0x20C

31	30	29	28	27	26	25	24	23	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 [103:96]							

Bit	Name	Access	Description	Reset value
[31:8]	Reserved	RO	Reserved	0
[7:0]	PENDSET	WO	96# to 103# interrupts pending set. 1: Interrupt pended; 0: No effect.	0

9.5.2.29 PFIC Interrupt Pending Reset 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**
PEND RESET 15	PEND RESET 14	PEND RESE T13	PEND RESET12	Reserved								PEND RESE T3	PEND RESE T2	Reserved	

Bit	Name	Access	Description	Reset value
[31:12]	PENDRST	WO	12# to 31# interrupts pending reset. 13# and 15# are reserved. 1: Interrupt pending status reset; 0: No effect.	0
[11:4]	Reserved	R0	Reserved	0
[3:2]	PENDRST	WO	2# to 3# interrupts pending reset: 1: Interrupt pending status reset;	0

			0: No effect.	
[1:0]	Reserved	RO	Reserved	0

9.5.2.30 PFIC Interrupt Pending Reset 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# to 63# interrupts pending reset: 1: Interrupt pending status reset; 0: No effect.	0

9.5.2.31 PFIC Interrupt Pending Reset Register 3 (PFIC_IPRR3)

Offset address: 0x288

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PENDRST[95:80]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PENDRST[79:64]															

Bit	Name	Access	Description	Reset value
[31:0]	PENDRESET	WO	64# to 95# interrupts pending reset: 1: Interrupt pending status reset; 0: No effect.	0

9.5.2.32 PFIC Interrupt Pending Reset Register 4 (PFIC_IPRR4)

Offset address: 0x28C

31	30	29	28	27	26	25	24	23	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 [103:96]							

Bit	Name	Access	Description	Reset value
[31:8]	Reserved	RO	Reserved	0
[7:0]	PENDSET	WO	96# to 103# interrupts pending reset: 1: Interrupt pending status reset; 0: No effect.	0

9.5.2.33 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								IACTS3	IACTS2	Reserved	

Bit	Name	Access	Description	Reset value
[31:12]	IACTS	RO	12# to 31# interrupts activation status. 13# and 15# are reserved. 1: Interrupt activating; 0: No interrupt activated.	0
[11:4]	Reserved	RO	Reserved	0
[3:2]	IACTS	RO	2# to 3# interrupts activation status: 1: Interrupt activating; 0: No interrupt activated.	0
[1:0]	Reserved	RO	Reserved	0

9.5.2.34 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# to 63# interrupts activation status: 1: Interrupt activating; 0: No interrupt activated.	0

9.5.2.35 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
IACTS[95:80]															

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

Bit	Name	Access	Description	Reset value
[31:0]	IACTS	RO	64# to 95# interrupts activation status: 1: Interrupt activating; 0: No interrupt activated.	0

9.5.2.36 PFIC Interrupt Activation Register 4 (PFIC_IACTR4)

Offset address: 0x30C

31	30	29	28	27	26	25	24	23	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 [103:96]							

Bit	Name	Access	Description	Reset value
[31:8]	Reserved	RO	Reserved	0
[7:0]	IACTS	WO	96# to 103# interrupts activation status: 1: Interrupt activating; 0: No interrupt activated.	0

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

Offset address: 0x400 to 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	See description of IP_0.	0
…	…	…	…	…
[31:24]	IP_3	RW	See description of IP_0.	0
[23:16]	IP_2	RW	See description of IP_0.	0
[15:8]	IP_1	RW	See description of IP_0.	0
[7:0]	IP_0	RW	Interrupt0 priority configuration: [7:4]: Priority control bits. No nested, no preemption bit; For 2 nested levels, bit7 is the preemption bit;	0

			For 4 nested levels, bit7-bit6 are preemption bits; For 8 nested levels, bit7-bit5 are preemption bits. The smaller priority value, the higher the priority. If interrupts with the same preemption priority are pended at the same time, execute the interrupt with the highest priority firstly. [3:0]: Reserved, fixed to 0, write invalid. *Note: Applicable to QingKe V4F core: CH32V30x_D8, CH32V30x_D8C.* Number 0 interrupt priority configuration. [7]: Priority control bits. If no nesting is configured, no preemption bit. If configured with 2 levels of nesting, bit7 is the preemption bit. The smaller the priority value, the higher the priority. If the same preemption priority interrupt hangs at the same time, the interrupt with the higher priority is executed first. [6:0]: Reserved, fixed to 0, write invalid. *Note: Applicable to QingKe V4B and V4C cores: CH32V20x_D6, CH32V20x_D8, CH32V20x_D8W.*	

9.5.2.38 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														

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	WO	System reset. Cleared automatically. Valid when writing 1, while invalid when writing 0. It has the same effect as that of the PFIC_CFGR register.	0
[30:6]	Reserved	RO	Reserved	0
5	SETEVENT	WO	Set event. To wake up WFE.	0
4	SEVONPEND	RW	When an event or interrupt is pended, wake up the system via WFE instruction. If WFE instruction is not executed, system wakes up immediately the next time WFE instruction is executed.	0

			1: The enabled events and all interrupts (including disabled interrupts) can wake up the system; 0: Only enabled events and enabled interrupts can wake up the system.	
3	WFITOWFE	RW	WFI instruction serves as WFE 1: Subsequent WFI instruction serves as WFE instruction; 0: No effect.	0
2	SLEEPDEEP	RW	System low power mode control: 1: Deep sleep; 0: Sleep.	0
1	SLEEPONEXIT	RW	System status control after exiting interrupt service program: 1: System goes into low power mode; 0: System goes into the main program.	0
0	Reserved	RO	Reserved.	0

9.5.3 Dedicated CSR registers

RISC-V defines some Control and Status Control Registers (CSR), which are used to configure, flag or record run status. CSR register is a register in the cores, with dedicated 12-bit address space. The CH32V20x and CH32V30x not only have standard registers defined in RISC-V privileged architecture documentation, but also have some additional vendor-defined registers which need csr instruction to access.

Note: The CSR registers with attribute of MRW/MRO/MRW1 need to be accessed in machine mode.

9.5.3.1 Interrupt System Control Register (INTSYSCR)

CSR address: 0x804

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMTSTA								Reserved		GIHWSTKNEN	HWSTKOVEN	PMTCFG		INESTEN	HWSTKEN

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	MRO	Reserved	0
[15:8]	PMTSTA	MRO	Preemption bit status: 0x00: No preemption bit in priority configuration bits, and no interrupt nested; 0x80: The preemption bit is the highest bit of the priority configuration bits, 2 interrupt nested levels; 0xC0: The preemption bits are the higher 2 bits of the priority configuration bits, 4 interrupt nested levels; 0xE0: The preemption bits are the higher 3 bits of	0

			the priority configuration bits, 8 interrupt nested levels. *Note: Applicable to QingKe V4F core: CH32V30x_D8, CH32V30x_D8C.* Preemption bit status indication: 0x00: No preempted bits in the priority configuration bits, no interrupt nesting occurs. 0x80: The highest bit in the priority configuration bit is a preempt bit, 2-level interrupt nesting. *Note: Applicable to QingKe V4B and V4C cores: CH32V20x_D6, CH32V20x_D8, CH32V20x_D8W.*	
[7:6]	Reserved	MRO	Reserved	0
5	GIHWSTKNEN	MRW1	Global interrupt and hardware stack disable: *Note: This bit is usually used for real-time system operation. When interrupt switches context, set this bit and the global interrupt and hardware stack out can be disabled. When the switch is finished, this bit is cleared automatically by hardware after the interrupt is returned.*	0
4	HWSTKOVEN	MRW	Interrupt enable after hardware stack overflows: 0: Global interrupt disabled after hardware stack overflows; 1: Interrupt still can be executed after hardware stack overflow. *Note: CH32V30x_D8 and CH32V30x_D8C hardware stack depth is 3. When the configuration nesting level is greater than 3, if this bit is set to 1, the low priority three interrupts need to be configured as hardware stack and the high priority as software stack.* *CH32V20x_D6, CH32V20x_D8, CH32V20x_D8W hardware stack depth is 2 levels.*	0
[3:2]	PMTCFG	MRW	Preemption bit configuration: 0b00: No nested, no preemption bit; 0b01: 2 nested levels, 1 preemption bit; 0b10: 4 nested levels, 2 preemption bits; 0b11: 8 nested levels, 3 preemption bits. *Note: Applicable to QingKe V4F core: CH32V30x_D8, CH32V30x_D8C.* Interrupt nesting depth configuration. 00: no nesting, the number of preemption bits is 0. 01: 2 levels of nesting, the number of preemption bits is 1. *Note: Applicable to QingKe V4B and V4C cores: CH32V20x_D6, CH32V20x_D8, CH32V20x_D8W.*	0

1	INESTEN	MRW	Interrupt nested enable: 0: Interrupt nested disabled; 1: Interrupt nested enabled.	0
0	HWSTKEN	MRW	Hardware stack enable: 0: Hardware stack disabled; 1: Hardware stack enabled.	0

9.5.3.2 Machine Trap-vector Base Address Register (MTVEC)

CSR address: 0x305

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASEADDR[15:2]														MODE1	MODE0

Bit	Name	Access	Description	Reset value
[31:2]	BASEADDR[31:2]	MRW	Base address of interrupt vector table	0
1	MODE1	MRW	Interrupt vector table identification mode: 0: Identify by jump instruction. Limited range, non-jump instruction supported; 1: Identify by absolute address. Full range supported, but it must jump.	0
0	MODE0	MRW	Interrupt or trap-vector address mode selection: 0: Unified entry address; 1: Address offset based on Interrupt No. *4.	0

9.5.4 Physical memory protection (PMP)

To improve system security, a set of physical address access restrictions is defined in the RISC-V architecture, and it allows physical memory access privileges (read, write, execute) to be specified for each physical memory region, and the minimum region length is 4 bytes for protection. The PMP unit is always effective in user mode, and optionally effective in machine mode. If the current memory limit is violated, a system exception interrupt (EXC) is generated.

The PMP unit contains 4 sets of 8-bit configuration registers (32bits) and 4 sets of address registers, which need to be accessed using the csr instruction and performed in machine mode.

9.5.4.1 PMP Configuration Register (PMPCFG0)

CSR address: 0x3A0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
pmp3cfg								pmp2cfg							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
pmp1cfg								pmp0cfg							

Bit	Name	Access	Description	Reset value
[31:24]	pmp3cfg	MRW	See description of pmp0cfg.	0
[23:16]	pmp2cfg	MRW	See description of pmp0cfg.	0
[15:8]	pmp1cfg	MRW	See description of pmp0cfg.	0
[7:0]	pmp0cfg	MRW	Bit | Name | Description 7 | L | Lock enable. It can be unlocked in machine mode 0: Not locked; 1: Related registers locked. [6:5] | - | Reserved [4:3] | A | Address aligned and protection region range selection 2 | X | Excute Attibute 1 | W | Write Attibute 0 | R | Read Attibute	0

For address aligned and protection region range selection, it performs memory protection for the region between A_ADDR and B_ADDR (A_ADDR and B_ADDR are both required 4-byte aligned):

1. If B_ADDR – A_ADDR == 2^2, it is based on NA4;
2. If B_ADDR – A_ADDR == $2^{(G+2)}$ and G≥1, and if A_ADDR= $2^{(G+2)}$, it is aligned based on NAPOT;
3. Otherwise it is based on TOR.

A	Name	Description
00b	OFF	No region to protect.

01b	TOR	Top aligned region protection:
		For pmp0cfg, 0≤ region ＜pmpaddr0;
		For pmp1cfg, pmpaddr0≤ region ＜pmpaddr1;
		For pmp2cfg, pmpaddr1≤ region ＜pmpaddr2;
		For pmp3cfg, pmpaddr2≤ region ＜pmpaddr3;
		pmpaddr_{i-1} = A_ADDR >> 2;
		pmpaddr_i = B_ADDR >> 2.

10b	NA4	Fixed 4-byte region protection. For pmp0cfg～pmp3cfg, pmpaddr0 to pmpaddr3 serve as start address. pmpaddr_i = A_ADDR >> 2.
11b	NAPOT	$2^{(G+2)}$ region protection, G≥1, in this case, A_ADDR is aligned on $2^{(G+2)}$. pmpaddri = （(A_ADDR | （2（G+2）-1））& ~(1<<G+1)）>> 2.

9.5.4.2 PMP Address 0 Register (PMPADDR0)

CSR address: 0x3B0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR0[33:18]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR0[17:2]															

Bit	Name	Access	Description	Reset value
[31:0]	ADDR0	MRW	The [33:2] bits of the PMP set address0, actually the higher 2 bits are not used.	0

9.5.4.3 PMP Address 1 Register (PMPADDR1)

CSR address: 0x3B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR1[33:18]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ADDR1[17:2]															

Bit	Name	Access	Description	Reset value
[31:0]	ADDR1	MRW	The [33:2] bits of the PMP set address1, actually the higher 2 bits are not used.	0

9.5.4.4 PMP Address 2 Register (PMPADDR2)

CSR address: 0x3B2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR2[33:18]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ADDR2[17:2]															

Bit	Name	Access	Description	Reset value
[31:0]	ADDR2	MRW	The [33:2] bits of the PMP set address2, actually the higher 2 bits are not used.	0

9.5.4.5 PMP Address 3 Register (PMPADDR3)

CSR address: 0x3B3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR3[33:18]															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ADDR3[17:2]															

Bit	Name	Access	Description	Reset value
[31:0]	ADDR3	MRW	The [33:2] bits of the PMP set address3, actually the higher 2 bits are not used.	0

9.5.5 RISC-V-SysTick registers

Table 9-6 STK registers

Name	Access address	Description	Reset value
R32_STK_CTLR	0xE000F000	System count control register	0x00000000
R32_STK_SR	0xE000F004	System count status register	0x00000000
R32_STK_CNTL	0xE000F008	System counter low register	0x00000000
R32_STK_CNTH	0xE000F00C	System counter high register	0x00000000
R32_STK_CMPLR	0xE000F010	Count/compare low register	0x00000000
R32_STK_CMPHR	0xE000F014	Count/compare high register	0x00000000

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

9.5.5.1 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
SWIE	Reserved														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										INIT	MODE	STRE	STCLK	STIE	STE

Bit	Name	Access	Description	Reset value
31	SWIE	RW	Software interrupt trigger enable (SWI): 1: Software interrupt triggered; 0: Trigger disabled. After going into software interrupt, it needs to be cleared by software, otherwise it is constantly triggered.	0
[30:6]	Reserved	RO	Reserved	
5	INIT	W1	Counter initial value update: 1: Updated to 0 when upcounting, while updated to the compare value when downcounting; 0: No effect.	
4	MODE	RW	Count mode: 1: Downcount; 0: Upcount.	
3	STRE	RW	Auto reload count enable bit: 1: After upcounting to the compare value, start counting from 0 again, and after downcounting to 0, start counting again from the compare value; 0: Continue to upcount after upcounting to the compare value, and downcount from the maximum value after downcounting to 0.	
2	STCLK	RW	Counter system clock sourse selection bit: 1: HCLK serves as time base;	

			0: HCLK/8 serves as time base.	
1	STIE	RW	Counter interrupt enable control bit: 1: Counter interrupt enabled; 0: Counter interrupt disabled.	
0	STE	RW	System counter enable control bit: 1: STK enabled; 0: STK disabled, the counter stops counting.	0

9.5.5.2 System Count Status Register (STK_SR)

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															CNTIF

Bit	Name	Access	Description	Reset value
[31:1]	Reserved	RO	Reserved	0
0	CNTIF	RW0	Count value compare flag. Cleared by writing 0, invalid when writing 1: 1: Upcount to the compare value, downcount to 0; 0: The compare value not reached.	0

9.5.5.3 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	Low 32 bits of the current counter count value.	0

Note: STK_CNTL and STK_CNTH constitute a 64-bit system counter.

9.5.5.4 System Counter High Register (STK_CNTH)

Offset address: 0x0C

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

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

Bit	Name	Access	Description	Reset value
[31:0]	CNT[63:32]	RW	High 32 bits of the current counter count value.	0

Note: STK_CNTL and STK_CNTH constitute a 64-bit system counter.

9.5.5.5 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.	0

Note: STK_CMPLR and STK_CMPHR constitute a 64-bit system counter compare value.

9.5.5.6 Count/Compare High Register (STK_CMPHR)

Offset address: 0x14

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

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

Bit	Name	Access	Description	Reset value
[31:0]	CMP[63:32]	RW	Set the high 32 bits of the compare counter value.	0

Note: STK_CMPLR and STK_CMPHR constitute a 64-bit system counter compare value.

9.5.6 ARM-SysTick registers

Table 9-7 SysTick registers

Name	Access address	Description	Reset value
R32_STK_CTRL	0xE000E010	SysTick control register	0x00000000
R32_STK_LOAD	0xE000E014	SysTick reload register	0x00000000
R32_STK_VAL	0xE000E018	SysTick value register	0x00000000
R32_STK_CALIB	0xE000E01C	SysTick calibration register	0x00000000

Note: Applied for general-purpose MCUs designed based on ARM$^{®}$CortexTM-M3 core.

9.5.6.1 SysTick Control Register (STK_CTRL)

Offset address: 0x00

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

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

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
16	COUNTFLAG	RO	This bit is 1 if SysTick has counted to zero since this register was read last time. Reading this bit will automatically clear it.	0
[15:3]	Reserved	RO	Reserved	0
2	CLKSOURCE	RW	0: External clock source (STCLK); 1: Internal clock (FCLK).	0
1	TICKINT	RW	1: SysTick exception request generated when SysTick counts down to zero. 0: No action when it counts to zero.	0
0	ENABLE	RW	SysTick enable	0

9.5.6.2 SysTick Reload Register (STK_LOAD)

Offset address: 0x04

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

Bit	Name	Access	Description	Reset value
[31:24]	Reserved	RO	Reserved	0
[23:0]	RELOAD	RW	Reloads value when it counts down to zero.	0

9.5.6.3 SysTick Value Register (STK_VAL)

Offset address: 0x08

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

Bit	Name	Access	Description	Reset value
[31:24]	Reserved	RO	Reserved	0
[23:0]	CURRENT	RW	Read it, and it will return current count value. Write to it, and it will be cleared, the COUNTFLAG flag in the SysTick control register will also be cleared.	0

9.5.6.4 SysTick Calibration Register (STK_CALIB)

Offset address: 0x0C

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

31	NOREF	RO	1: No external reference clock (STCLK not available); 0: Available external reference clock.	0
30	SKEW	RO	1: The calibration value is not exactly 10ms; 0: The calibration value is exactly 10ms.	0
[29:24]	Reserved	RO	Reserved	0
[23:0]	TENMS	RW	The number of cells counted down within 10ms. The chip designer should provide this value through the input signal of the Cortex-M3. If the read value is zero, the calibration function cannot be used.	0

Chapter 10 GPIO and Alternate function (GPIO/AFIO)

This chapter applies to the whole family of CH32F2x, CH32V2x and CH32V3x.

The GPIO ports can be configured as a variety of input or output modes, with built-in pull-up/pull-down resistors which can be switched off, and can be configured as push-pull or open-drain functions. GPIO ports also have some other alternate functions.

10.1 Main features

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

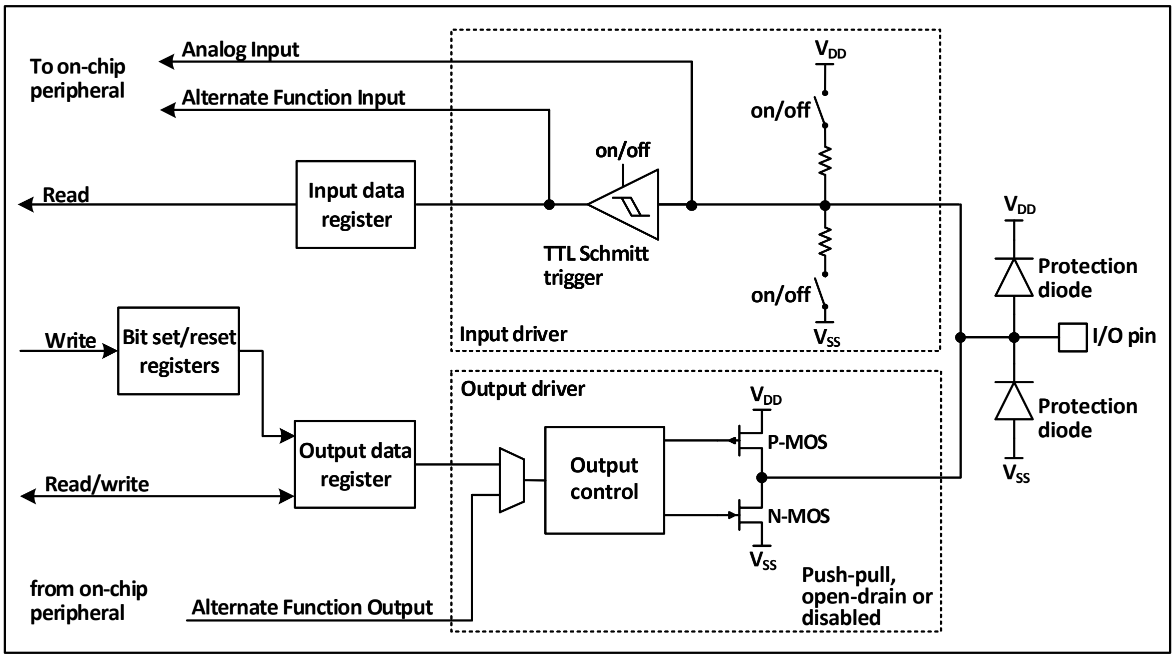
- Floating input
- Pull-up input
- Pull-down input
- Analog input
- Open-drain output
- Push-pull output
- Input and output of alternate function

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 specification

10.2.1 Overview

Figure 10-1 Basic structure of GPIO module



The IO port structure is as shown in Figure 10-1. Each pin has 2 protection diodes inside the chip, and the IO port can be divided into input and output drive modules internally. The weak pull-up and pull-down resistors are optional for input drive, and can be connected to analog input peripherals such as AD; if inputted to digital peripherals, they need to pass through a TTL Schmitt trigger, and then shall be connected to GPIO input register or other alternate peripherals. The output drive has a pair of MOS transistors. The IO port can be configured as open-drain or push-pull output by configuring whether the upper and lower MOS transistors are enabled;

the output drive can also be internally configured to the output controlled by GPIO or other alternate peripherals.

10.2.2 GPIO Initialization

Immediately after reset, the GPIO ports are running in the initial status. At this time, most IO ports are running at the floating input status, but there are also peripheral-related pins such as HSE that is running on peripheral alternate functions. Please refer to related chapters of pins for the specific initialization function.

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 a GPIO pin at most, and the serial number of the external interrupt channel must be consistent with the bit number of the GPIO port, such as PA1 (or PB1, PC1, PD1 and PE1) can only be mapped to EXTI1, and EXTI1 can only accept mappings from one of PA1, PB1, PC1, PD1 and PE1. Both ports have one-to-one relationship.

10.2.4 Alternate function

Attention shall be paid to the following when the alternate function is used:

- To use the alternate function of the input direction, the port must be configured to alternate input mode, the pull-up and pull-down settings can be set according to actual needs
- To use the alternate function of the output direction, the port must be configured to alternate output mode, push-pull or open drain can be set according to the actual situation
- For the bidirectional alternate function, the port must be configured as multiplex output mode, and then the driver will be configured as float control input mode

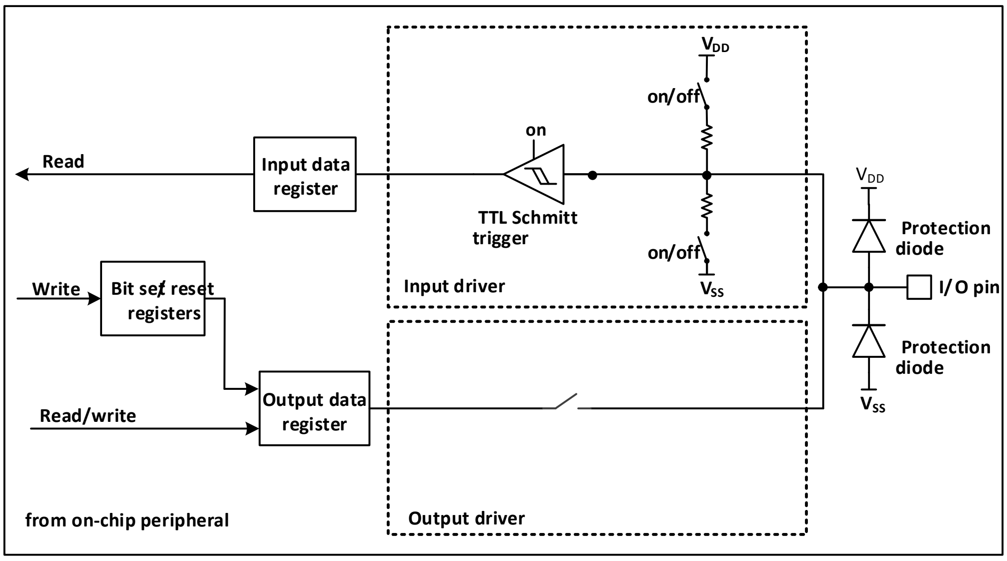
The same IO port may have multiple peripherals alternate to this pin, so in order to give play to the use of each peripheral as far as possible, the alternate pins of the peripherals can be remapped to other pins in addition to the default alternate pins, avoiding occupied pins.

10.2.5 Locking mechanism

The locking mechanism can lock the configuration of 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

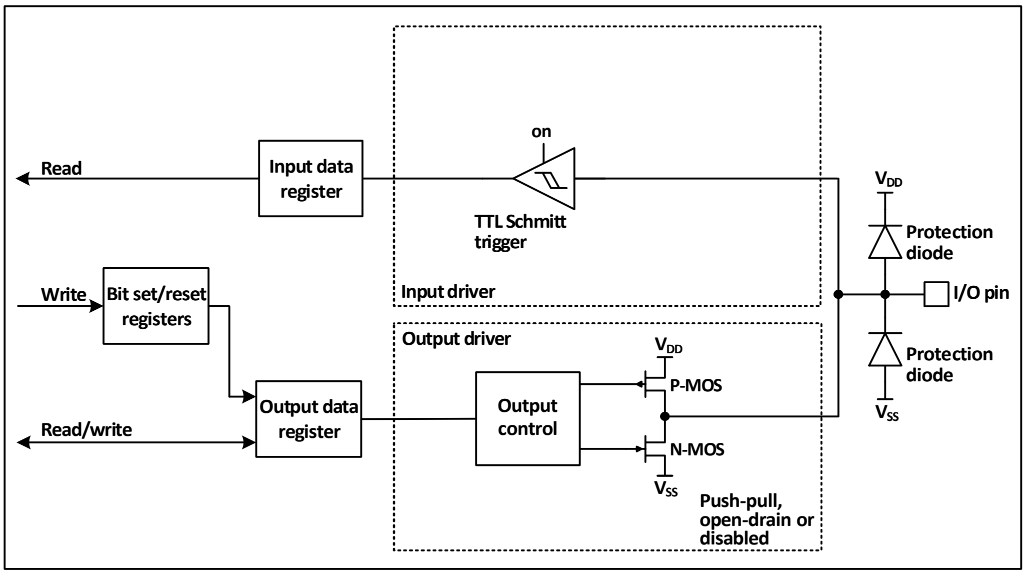
Figure 10-2 Input configuration structure block diagram of GPIO module



When the IO port is configured as input mode, the output driver is disconnected, the input pull-up and pull-down are 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 APB2 clock, and the corresponding bit of the input data register is read to obtain the level state of the corresponding pin.

10.2.7 Output configuration

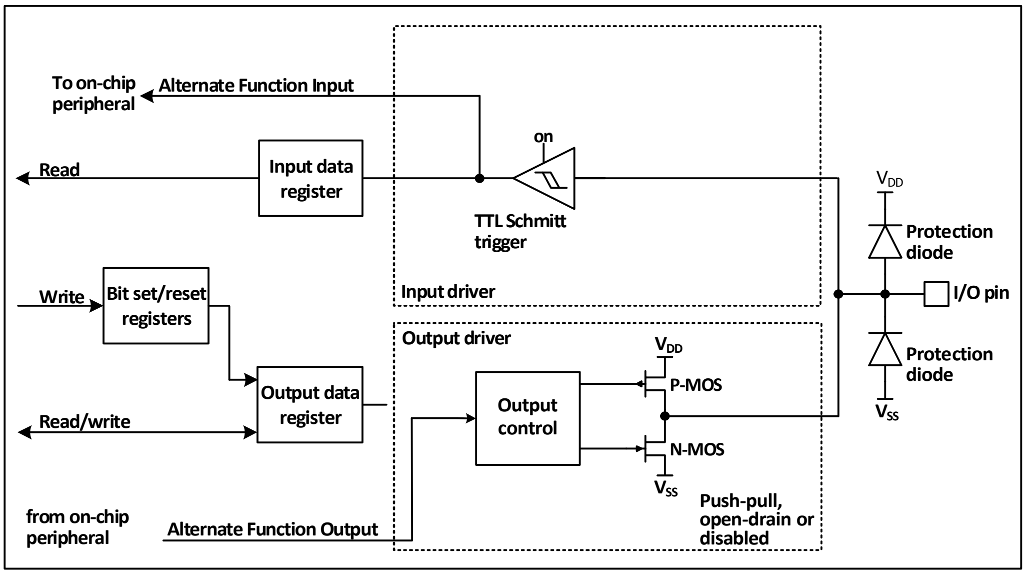
Figure 10-3 Output configuration structure block diagram of GPIO module



When the IO port is configured as output mode, a pair of MOS in the output driver can be configured as push-pull or open-drain mode as required, without using alternate function. The pull-up and pull-down resistors of the input drive are disabled, the TTL Schmitt trigger is activated, and the level appearing on the IO pin will be sampled to the input data register every APB2 clock, so IO status will be obtained by reading the input data register. In the push-pull output mode, the value written last time will be obtained through the access to the output data register.

10.2.8 Alternate function configuration

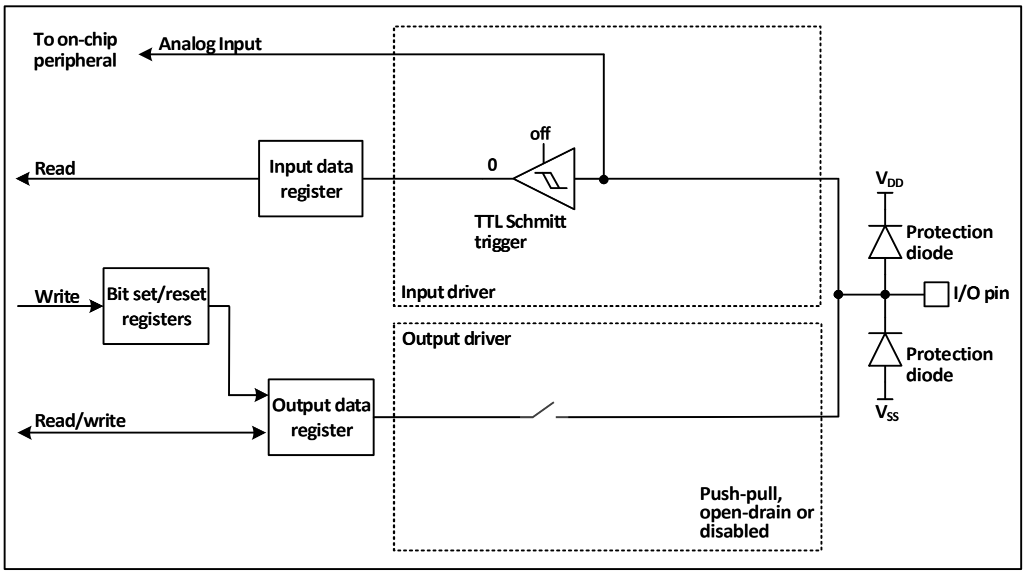
Figure 10-4 Structure block diagram when GPIO module is multiplexed by other peripherals



When the alternate function is enabled, the output driver is enabled and can be configured as open-drain or push-pull mode as required. Schmitt trigger will be also turned on, the input and output lines of the alternate function is connected, but the output data register is disconnected, and the level appearing on the IO pin is sampled to the input data register every APB2 clock. In the open-drain mode, the current status of the IO port is obtained by reading the input data register. In the push-pull mode, the last written value is obtained by reading the output data register.

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 and the Schmitt trigger input in the input driver is disabled to prevent consumption on the IO port. The pull-up and pull-down resistors is disabled, and the read input data register is always 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 (TIM1/8/9/10)

TIM1/8/9/10	Configuration	GPIO configuration
TIM1/8/9/10_CHx	Input capture channel x	Floating input
	Output compare channel x	Push-pull alternate output
TIM1/8/9/10_CHxN	Complementary output channel x	Push-pull alternate output
TIM1/8/9/10_BKIN	Break input	Floating input
TIM1/8/9/10_ETR	External trigger clock input	Floating input

Table 10-2 General-purpose timer (TIM2/3/4/5)

TIM2/3/4/5 pinout	Configuration	GPIO configuration
TIM2/3/4/5_CHx	Input capture channel x	Floating input
	Output compare channel x	Push-pull alternate output
TIM2/3/4/5_ETR	External trigger clock input	Floating input

Table 10-3 Universal Synchronous Asynchronous Receiver Transmitter (USART)

USART pinout	Configuration	GPIO configuration
USARTx_TX	Full duplex mode	Push-pull alternate output
	Half-duplex synchronous mode	Push-pull 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 (SPI) module

SPI pinout	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	Push-pull alternate output
	Software mode	Not used

Table 10-5 Inter IC Sound (I2S) module

I2S pinout	Configuration	GPIO configuration
I2Sx_WS	Master mode	Push-pull alternate output
	Slave mode	Floating input
I2Sx_CK	Master mode	Push-pull alternate output
	Slave mode	Floating input
I2Sx_SD	Transmitter	Push-pull alternate output
	Receiver	Floating, pull-up or pull-down input
I2Sx_MCK	Master mode	Push-pull alternate output
	Slave mode	Not used

Table 10-6 Inter Integrated Circuit (I2C) module

I^{2}C pinout	Configuration	GPIO configuration
I^{2}C_SCL	I^{2}C clock	Open-drain alternate output
I^{2}C_SDA	I^{2}C data	Open-drain alternate output

Table 10-7 Controller LAN (CAN) module

CAN pinout	GPIO configuration
CANx_TX	Push-pull alternate output
CANx_RX	Floating input or pull-up input

Table 10-8 USB Full-speed Device (USBD) controller

USBD pinout	GPIO configuration
USBD_DM/USBD_DP	After the USB is enabled, alternate IO port connects to the internal USBD transceiver automatically

Table 10-9 USB Host Device (USBHD) controller

USBHD pinout	GPIO configuration
USBHD_DM/USBHD_DP	After the USB is enabled, alternate IO port connects to the internal USBHD transceiver automatically

Table 10-10 USB OTG controller

USB OTG pinout	GPIO configuration
OTG_FS_VBUS	Analog input
OTG_FS_ID	Pull-up input
OTG_FS_DM	Automatically controlled by USB disconnected
OTG_FS_DP	Automatically controlled by USB disconnected

Table 10-11 Security Digital Input and Output (SDIO) module

SDIO pinout	Configuration	GPIO configuration
SDIO_CK	Clock	Push-pull alternate output
SDIO_CMD	Command	Push-pull alternate output
SDIO[D7:D0]	Data	Push-pull alternate output

Table 10-12 Flexible Static Memory Controller (FSMC)

FSMC pinout	GPIO configuration
FSMC_A[23:16] FSMC_D[15:0]	Push-pull alternate output
FSMC_CK	Push-pull alternate output
FSMC_NOE FSMC_NWE	Push-pull alternate output
FSMC_NE1 FSMC_NCE2	Push-pull alternate output
FSMC_NWAIT	Floating input or pull-up input
FSMC_NBL[1:0]	Push-pull alternate output

Table 10-13 Analog-to-Digital Converter (ADC) and Digital-to-Analog Converter (DAC)

ADC/DAC pinout	GPIO configuration
ADC/DAC	Analog input

Table 10-14 Other IO function settings

Pinout	Configuration function	GPIO configuration
TAMPER_RTC	RTC output	Hardware automatic setting
	Tamper event input	
MCO	Clock output	Push-pull alternate output
EXTI	External interrupt input	Floating input or pull-up or pull-down input

10.2.11 Remapping GPIO settings with multiplexing functions

10.2.11.1 OSC32_IN/OSC32_OUT as GPIO port PC14/PC15

When LSEON=0, the LSE oscillator pin OSC32_IN/OSC32_OUT can be used as PC14/PC15 of GPIO respectively.

When LSEON=1, it is used as LSE pin.

10.2.11.2 OSC_IN/OSC_OUT as GPIO port PD0/PD1

OSC_IN/OSC_OUT can be used as PD0/PD1 of GPIO, which is achieved by setting remapping register 1 (AFIO_PCFR1).

This remapping is only available for 20, 32, 48 and 64 pin packages, but CH32V203RBT6 only has OSC_IN and OSC_OUT function pins and does not support mapping (for LQFP100 package, there is no need for software remapping since PD0 and PD1 are inherent function pins.)

10.2.11.3 Timer multiplexing function remapping

Table 10-15 TIM1 Multiplexed Function Remapping

Multiplexed Function	TIM1_RM=00 Default Mapping	TIM1_RM=01 Partial Mapping	TIM1_RM=11 Complete Mapping [(1)]
TIM1_ETR	PA12	PA12	PE7
TIM1_CH1	PA8	PA8	PE9
TIM1_CH2	PA9	PA9	PE11
TIM1_CH3	PA10	PA10	PE13
TIM1_CH4	PA11	PA11	PE14
TIM1_BKIN	PB12	PA6	PE15
TIM1_CH1N	PB13	PA7	PE8
TIM1_CH2N	PB14	PB0	PE10
TIM1_CH3N	PB15	PB1	PE12

Note: (1) Only LQFP100 package, support this bit remapping function.

Table 10-16 TIM2 Multiplexed Function Remapping

Multiplexed Function	TIM2_RM=00 Default Mapping	TIM2_RM=01 Partial Mapping	TIM2_RM=10 Partial Mapping	TIM2_RM=11 Complete Mapping
TIM2_ETR	PA0	PA15	PA0	PA15
TIM2_CH1	PA0	PA15	PA0	PA15
TIM2_CH2	PA1	PB3	PA1	PB3
TIM2_CH3	PA2	PA2	PB10	PB10
TIM2_CH4	PA3	PA3	PB11	PB11

Table 10-17 TIM3 Multiplexed Function Remapping

Multiplexed Function	TIM3_RM=00 Default Mapping	TIM3_RM=10 Partial Mapping	TIM3_RM=11 Complete Mapping [(1)]
TIM3_CH1	PA6	PB4	PC6
TIM3_CH2	PA7	PB5	PC7
TIM3_CH3	PB0	PB0	PC8
TIM3_CH4	PB1	PB1	PC9

Note: (1) The bit mapping is not supported in packages below 64 pins.

Table 10-18 TIM4 Multiplexed Function Remapping

Multiplexed Function	TIM4_RM=0 Default Mapping	TIM4_RM=1 Remapping [(1)]
TIM4_CH1	PB6	PD12
TIM4_CH2	PB7	PD13
TIM4_CH3	PB8	PD14
TIM4_CH4	PB9	PD15

Note: (1) Only LQFP100 package, support this bit remapping function.

Table 10-19 TIM5 Multiplexed Function Remapping

Multiplexed Function	TIM5CH4_RM=0 Default Mapping	TIM5CH4_RM=1 Remapping
TIM5_CH4	PA3	LSI internal clock

Table 10-20 TIM8 Multiplexed Function Remapping (1)

Multiplexed Function	TIM8_RM=0 Default Mapping	TIM8_RM=1 Remapping
TIM8_ETR	PA0	PA0
TIM8_CH1	PC6	PB6
TIM8_CH2	PC7	PB7
TIM8_CH3	PC8	PB8
TIM8_CH4	PC9	PC13
TIM8_BKIN	PA6	PB9
TIM8_CH1N	PA7	PA13
TIM8_CH2N	PB0	PA14
TIM8_CH3N	PB1	PA15

Table 10-21 TIM8 Multiplexed Function Remapping

Multiplexed Function	TIM9_RM=00 Default Mapping	TIM9_RM=01 Partial Mapping	TIM9_RM=1x Complete Mapping (1)
TIM9_ETR	PA2	PA2	PD9
TIM9_CH1	PA2	PA2	PD9
TIM9_CH2	PA3	PA3	PD11
TIM9_CH3	PA4	PA4	PD13
TIM9_CH4	PC4	PC14	PD15
TIM9_BKIN	PC5	PA1	PD14
TIM9_CH1N	PC0	PB0	PD8
TIM9_CH2N	PC1	PB1	PD10
TIM9_CH3N	PC2	PB2	PD12

Note: (1) Only LQFP100 package, support this bit remapping function.

Table 10-22 TIM10 Multiplexed Function Remapping

Multiplexed Function	TIM10_RM=00 Default Mapping	TIM10_RM=01 Partial Mapping	TIM10_RM=1x Complete Mapping (1)
TIM10_ETR	PC10	PB11	PD0
TIM10_CH1	PB8	PB3	PD1
TIM10_CH2	PB9	PB4	PD3
TIM10_CH3	PC3	PB5	PD5
TIM10_CH4	PC11	PC15	PD7
TIM10_BKIN	PC12	PB10	PE2
TIM10_CH1N	PA12	PA5	PE3

TIM10_CH2N	PA13	PA6	PE4
TIM10_CH3N	PA14	PA7	PE5

Note: (1) Only LQFP100 package, support this bit remapping function.

10.2.11.4 USART, UART multiplexing function remapping

Table 10-23 USART1 Multiplexing Function Remapping

Multiplexed Function	USART1_RM1=0 (1) USART1_RM2=0 (2) Default Mapping	USART1_RM1=1 (1) USART1_RM2=0 (2) Remapping	USART1_RM1=0 (1) USART1_RM2=1 (2) Remapping	USART1_RM1=1 (1) USART1_RM2=1 (2) Remapping
USART1_CK	PA8	PA8	PA10	PA5
USART1_TX	PA9	PB6	PB15	PA6
USART1_RX	PA10	PB7	PA8	PA7
USART1_CTS	PA11	PA11	PA5	PC4
USART1_RTS	PA12	PA12	PA9	PC5

Note: (1) USART1_RM1 is AFIO_PCFR1 register bit2, which is the low bit of mapping configuration.

(2) USART1_RM2 is AFIO_PCFR2 register bit26, which is the high bit of mapping configuration. CH32V20x_D6 and CH32F20x_D6 do not support this bit mapping.

Table 10-24 USART2 Multiplexing Function Remapping

Multiplexed Function	USART2_RM=0 Default Mapping	USART2_RM=1 Remapping (1)
USART2_CTS	PA0	PD3
USART2_RTS	PA1	PD4
USART2_TX	PA2	PD5
USART2_RX	PA3	PD6
USART2_CK	PA4	PD7

Table 10-25 USART3 Multiplexing Function Remapping

Multiplexed Function	USART3_RM=00 Default Mapping	USART3_RM=01 Partial Mapping (1)	USART3_RM=10 Partial Mapping (2)	USART3_RM=11 Complete Mapping (3)
USART3_TX	PB10	PC10	PA13	PD8
USART3_RX	PB11	PC11	PA14	PD9
USART3_CK	PB12	PC12	PD10	PD10
USART3_CTS	PB13	PB13	PD11	PD11
USART3_RTS	*PB14*	*PB14*	*PD12*	*PD12*

Note: (1) The bit mapping is not supported for packages below 48 pins.

(2) The function is not supported for the fifth bit of the lot number less than 2.

(3) Only the LQFP100 package supports this bit remapping function.

Table 10-26 USART4 Multiplexing Function Remapping

Multiplexed Function [(1)]	UART4_RM=00 Default Mapping	UART4_RM=01 Remapping	UART4_RM=1x Remapping [(2)]
UART4_TX	PC10	PB0	PE0
UART4_RX	PC11	PB1	PE1

Note: (1) Applicable to CH32V30x_D8C, CH32V30x_D8, CH32V30x_D8W, CH32V20x_D8, CH32F20x_D8C, CH32F20x_D8, CH32F20x_D8W.

(2) Only LQFP100 package supports this bit remapping function.

Table 10-27 USART4 Multiplexing Function Remapping [(2)]

Multiplexed Function	USART4_RM=x0 Default Mapping	USART4_RM=x1 Remapping
USART4_CK	PB2	PA6
USART4_TX	PB0	PA5
USART4_RX	PB1	PB5
USART4_CTS	PB3	PA7
USART4_RTS	PB4	PA15

Note: (1) Only LQFP100 package, support this bit remapping function.

(2) This function is not supported for products below 64 pins.

Table 10-28 USART5 Multiplexing Function Remapping

Multiplexed Function	UART5_RM=00 Default Mapping	UART5_RM=01 Remapping	UART5_RM=1x Remapping [(1)]
UART5_TX	PC12	PB4	PE8
UART5_RX	PD2	PB5	PE9

Note: (1) Only LQFP100 package, support this bit remapping function.

(2) This function is not supported for products below 64 pins.

Table 10-29 USART6 Multiplexing Function Remapping [(2)]

Multiplexed Function	UART6_RM=00 Default Mapping [(2)]	UART6_RM=01 Remapping	UART6_RM=1x Remapping [(1)]
UART6_TX	PC0	PB8	PE10
UART6_RX	PC1	PB9	PE11

Note: (1) Only LQFP100 package, support this bit remapping function.

(2) This function is not supported for products below 64 pins.

Table 10-30 USART7 Multiplexing Function Remapping [(2)]

Multiplexed Function	UART7_RM=00 Default Mapping	UART7_RM=01 Remapping	UART7_RM=1x Remapping [(1)]
UART7_TX	PC2	PA6	PE12
UART7_RX	PC3	PA7	PE13

Note: (1) Only LQFP100 package, support this bit remapping function.

(2) This function is not supported for products below 64 pins.

Table 10-31 USART8 Multiplexing Function Remapping (2)

Multiplexed Function	UART8_RM=00 Default Mapping	UART8_RM=01 Remapping	UART8_RM=1x Remapping [(1)]
UART8_TX	PC4	PA14	PE14
UART8_RX	PC5	PA15	PE15

Note: (1) Only LQFP100 package, support this bit remapping function.

(2) This function is not supported for products below 64 pins.

10.2.11.5 SPI multiplexing function remapping

Table 10-32 SPI1 Multiplexing Function Remapping

Multiplexed Function	SPI1_RM=0 Default Mapping	SPI1_RM=1 Remapping
SPI1_NSS	PA4	PA15
SPI1_SCK	PA5	PB3
SPI1_MISO	PA6	PB4
SPI1_MOSI	PA7	PB5

Table 10-33 SPI3 Multiplexing Function Remapping

Multiplexed Function [(1)]	SPI1_RM=0 Default Mapping	SPI1_RM=1 Remapping
SPI3_NSS	PA15	PA4
SPI3_SCK	PB3	PC10
SPI3_MISO	PB4	PC11
SPI3_MOSI	PB5	PC12

10.2.11.6 I2C multiplexing function remapping

Table 10-34 I2C Multiplexing Function Remapping

Multiplexed Function	I2C1_RM=0 Default Mapping	I2C1_RM=1 Remapping
I2C1_SCL	PB6	PB8
I2C1_SDA	PB7	PB9

10.2.11.7 CAN multiplexing function remapping

Table 10-35 CAN1 Multiplexing Function Remapping

Multiplexed Function	CAN1_RM=00 Default Mapping	CAN1_RM=10 Remapping (1)	CAN1_RM=11 Remapping (2)
CAN1_RX	PA11	PB8	PD0
CAN1_TX	PA12	PB9	PD1

Note: (1) Remapping does not apply to chips in packages below 48 pins.

(2) When PD0 and PD1 are not remapped to OSC_IN and OSC_OUT, the remapping function is only available on 100-pin packages.

Table 10-36 CAN2 Multiplexing Function Remapping

Multiplexed Function	CAN2_RM=0 Default Mapping	CAN2_RM=1 Remapping
CAN2_RX	PB12	PB5
CAN2_TX	PB13	PB6

10.2.11.8 ADC multiplexing function remapping

Table 10-37 ADC1 external trigger injection conversion multiplexing function remapping

Multiplexed Function	ADC1_ETRGINJ_RM=0 Default Mapping	ADC1_ETRGINJ_RM=1 Remapping
ADC1 external trigger injection conversion	ADC1 external trigger injection conversion connected to EXTI15	ADC1 external trigger injection conversion connected to TIM8_CH4

Table 10-38 ADC1 external trigger rule conversion multiplexing function remapping

Multiplexed Function	ADC1_ETRGREG_RM=0 Default Mapping	ADC1_ETRGREG_RM=1 Remapping
ADC1 external trigger rule conversion	ADC1 external trigger rule conversion connected to EXTI11	ADC1 external trigger rule conversion connected to TIM8_TRG0

Table 10-39 ADC2 external trigger injection conversion multiplexing function remapping

Multiplexed Function	ADC2_ETRGINJ_RM=0 Default Mapping	ADC2_ETRGINJ_RM=1 Remapping
ADC2 external trigger injection conversion	ADC2 external trigger injection conversion connected to EXTI15	ADC2 external trigger injection conversion connected to TIM8_CH4

Table 10-40 ADC2 external trigger rule conversion multiplexing function remapping

Multiplexed Function	ADC2_ETRGREG_RM=0 Default Mapping	ADC2_ETRGREG_RM=1 Remapping
ADC2 external trigger rule conversion	ADC2 external trigger rule conversion connected to EXTI11	ADC2 external trigger rule conversion connected to TIM8_TRG0

Note: ADC1 remapping is only supported for products where TIM8 is present, see the relevant datasheet for details.

10.2.11.9 ETH multiplexing function remapping

Table 10-41 ETH multiplexing function remapping

Multiplexed Function (1)	ETH_RM=0 Default Mapping	ETH_RM=1 Remapping (1)
ETH_RX_DV	PA7	PD8
ETH_CRS_DV	PA7	PD8
ETH_RXD0	PC4	PD9
ETH_RXD1	PC5	PD10
ETH_RXD2	PB0	PD11
ETH_RXD3	PB1	PD12

Note: (1) Only LQFP100 package interconnect type supports this bit remapping function.

10.2.11.10 FSMC_NADV multiplexing function remapping

Table 10-42 FSMC_NADV multiplexing function remapping

Multiplexed Function (1)	FSMCEN=1 Default Mapping	FSMCEN=1&USBHSEN=1 &RB_UC_RST_SIE=0
FSMC_NADV	PB7	*PD2*

Note: (1) FSMC_NADV output is disabled when FSMC_NADV=1.

(2) The function is not supported if the fifth bit of the lot number is less than 2.

10.2.11.11 DVP multiplexing function remapping

Table 10-43 DVP multiplexing function remapping

Multiplexed Function	DVPEN=1 Default Mapping	DVPEN=1&USBHSEN=1 &RB_UC_RST_SIE=0
DVP_D5	PB6	*PB3*

Note: (1) The function is not supported if the fifth bit of the lot number is less than 2.

10.2.11.12 SDIO multiplexing function remapping

Table 10-44 SDIO multiplexing function remapping

Multiplexed Function	SDIOEN=1	SDIOEN=1&ETHMACEN=1

SDCK	PC12	PC12
CMD	PD2	PD2
SD0	PC8	*PB14*
SD1	PC9	*PB15*
SD2	PC10	PC10
SD3	PC11	PC11
SD4	PB8	PB8
SD5	PB9	PB9
SD6	PC6	PC6
SD7	PC7	PC7

Note: (1) The function is not supported if the fifth bit of the lot number is less than 2.

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 10-15 GPIO 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_GPIOE_CFGLR	0x40011800	PE 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_GPIOE_CFGHR	0x40011804	PE 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_GPIOE_INDR	0x40011808	PE 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_GPIOE_OUTDR	0x4001180C	PE 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_GPIOE_BSHR	0x40011810	PE 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_GPIOE_BCR	0x40011814	PE port reset register	0x00000000
R32_GPIOA_LCKR	0x40010818	PA port configuration lock register	0x00000000
R32_GPIOB_LCKR	0x40010C18	PB port configuration lock register	0x00000000
R32_GPIOC_LCKR	0x40011018	PC port configuration lock register	0x00000000
R32_GPIOD_LCKR	0x40011418	PD port configuration lock register	0x00000000
R32_GPIOE_LCKR	0x40011818	PE port configuration lock register	0x00000000

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

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 bit of port x; configure the corresponding port through these bits.
			Input mode (MODE=00b):
			00: Analog input mode;
			01: Floating input mode;
			10: Mode with pull-up and pull-down
			11: Reserved.
			Output mode (MODE>00b):
			00: General push-pull output mode;
			01: General 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=0-7), port x mode selection, configure the corresponding port through these bits. 00: Input mode; 01: Output mode, maximum speed: 10MHz; 10: Output mode, maximum speed: 2MHz; 11: Output mode, maximum speed: 50MHz;	0

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

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 bit of port x; configure the corresponding port through these bits.
			Input mode (MODE=00b):
			00: Analog input mode;
			01: Floating input mode;
			10: Mode with pull-up and pull-down
			11: Reserved.
			Output mode (MODE>00b):
			00: General push-pull output mode;
			01: General 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 configuration mode of port x; configure the corresponding port through these bits. 00: Input mode; 01: Output mode, maximum speed: 10MHz; 10: Output mode, maximum speed: 2MHz; 11: Output mode, maximum speed: 50MHz;	0

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

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 in 16-bit form. The read value represents the high/low status of the corresponding bit.	X

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

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	(y=0-15), port output data. These bits can only be operated in form of 16 bits. The I/O port outputs the value of these registers externally. For input modes with pull-up and 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/E)

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), setting these bits will clear the corresponding OUTDR bits, and writing 0 has no effect. These bits can only be accessed in form of 16 bits. If the BR and BS bits are set at the same time, the BS bit takes effect.	0
[15:0]	BSy	WO	(y=0-15), setting these bits will set the corresponding OUTDR bits, and writing 0 has no effect. These bits can only be accessed in form of 16 bits. If the BR and BS bits are set at the same time, the BS bit takes effect.	0

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

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), setting these bits will clear the corresponding OUTDR bits, and writing 0 has no effect. These bits can only be accessed in form of 16 bits.	0

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

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. It can be locked by writing in a specific sequence, but it can be read out at any time. When it is read as 0, it means that it is unlocked. When it is read as 1, it means that it is locked. 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 check whether the lock key has been activated. When the sequence is written, any error will not lock the activation. When the sequence is written, the value of LCK[15:0] cannot be changed. After the lock takes effect, the port configuration can be only changed after the next reset.	0
[15:0]	LCKy	RW	(y=0-15), when these bits are 1, it means that the configuration of the corresponding port is locked. These bits can only be changed before the LCKK is unlocked. The locked configuration refers to 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.

10.3.2 AFIO registers

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

Table 10-16 AFIO registers

Name	Access address	Description	Reset value
R32_AFIO_ECR	0x40010000	Event control register	0x00000000
R32_AFIO_PCFR1	0x40010004	Remap register1	0x00000000

R32_AFIO_EXTICR1	0x40010008	External interrupt configuration register1	0x00000000
R32_AFIO_EXTICR2	0x4001000C	External interrupt configuration register2	0x00000000
R32_AFIO_EXTICR3	0x40010010	External interrupt configuration register3	0x00000000
R32_AFIO_EXTICR4	0x40010014	External interrupt configuration register4	0x00000000
R32_AFIO_PCFR2	0x4001001C	Remap register2	0x00000000

10.3.2.1 Event Control Register (AFIO_ECR)

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								EVOE	PORT[2:0]			PIN[3:0]			

Bit	Name	Access	Description	Reset value
[31:8]	Reserved	RO	Reserved	0
7	EVOE	RW	Event output enable bit. By setting this bit, the EVENTOUT of core is connected to the selected IO ports of PORT and PIN.	0

[6:4]	PORT[2:0]	RW	Used to select the port of core output EVENTOUT:
			000: PA port; 001: PB port;
			010: PC port; 011: PD port;
			Others: Reserved.
				0

[3:0]	PIN[3:0]	RW	The value of these bits is used to determine the number of the pin that selects the core EVENTOUT to the port. Values 0-15 correspond to pins 0-15 of the PX selected in PORT.	0

10.3.2.2 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	PTP_PPS_RM	TIM2ITR1_RM	SPI3_RM	Reserved	SW_CFG[2:0]			MII_RMII_SEL	CAN2_RM	ETH_RM	ADC2_ETRGREG_RM	ADC2_ETRGINJ_RM	ADC1_ETRGREG_RM	ADC1_ETRGINJ	TIM5CH4_RM

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PD01 RM	CAN_RM [1:0]		TIM4_RM	TIM3_RM [1:0]		TIM2_RM [1:0]		TIM1_RM [1:0]		USART3_RM [1:0]		USART2_RM	USART1_RM	I2C1_RM	SPI1_RM

Bit	Name	Access	Description	Reset value
31	Reserved	RO	Reserved	0
30	PTP_PPS_RM	RW	Ethernet PTP PPS remap 0: PTP PPS not output to the PB5 pin; 1: PTP PPS output to the PB5 pin.	0
29	TIM2ITR1_RM	RW	TIM2 internal trigger1 remap 0: Internally connect TIM2_ITR1 to Ethernet PTP output; 1: Internally connect TIM2_ITR1 to SOF output of full-speed USB OTG.	0
28	SPI3_RM	RW	SPI3 remap 0: Default map (NSS/PA15, SCK/PB3, MSIO/PB4, MOSI/PB5); 1: Remap (NSS/PA4, SCK/PC10, MSIO/PC11, MOSI/PC12).	0
27	Reserved	RO	Reserved	0

[26:24]	SW_CFG[2:0]	WO	These bits are used to configure the IO port with SW function and tracking function. SWD (SDI) is a debug interface to access the core. After system reset, it is always used as the SWD port.
			0xx: SWD enabled (SDI);
			100: SWD disabled (SDI), served as GPIO;
			Others: Invalid.
				0

23	MII_RMII_SEL	RW	MII or RMII selection. Configure the internal Ethernet MAC to use the external transceiver with MII interface or RMII interface (PHY). 0: Configure the Ethernet MAC to use the external transceiver with MII interface (PHY); 1: Configure the Ethernet MAC to use the external transceiver with RMII interface (PHY).	0
22	CAN2_RM	RW	CAN2 remap bit. 0: Default map (CAN2_RX/PB12, CAN2_TX/PB13); 1: Remap (CAN2_RX/PB5, CAN2_TX/PB6).	0
21	ETH_RM	RW	Ethernet remap bit. 0: Default map (RX_DV-CRS_DV/PA7, RXD0/PC4, RXD1/PC5, RXD2/PB0, RXD3/PB1); 1: Remap (RX_DV-CRS_DV/PD8, RXD0/PD9, RXD1/PD10, RXD2/PD11, RXD3/PD12)	0
20	ADC2_ETRGREG_RM	RW	Remap bit of ADC1 external trigger regular conversion. 0: ADC1 external trigger regular conversion EXTI11; 1: ADC1 external trigger regular conversion TIM8_TRGO.	0
19	ADC2_ETRGINJ_RM	RW	Remap bit of ADC2 external trigger injection conversion	0

			0: ADC2 external trigger injection conversion to EXTI15; 1: ADC2 external trigger injection conversion to TIM8_CH4.	
18	ADC1_ETRGREG_RM	RW	Remap bit of ADC1 external trigger regular conversion 0: ADC1 external trigger regular conversion to EXTI11; 1: ADC1 external trigger regular conversion to TIM8_TRGO.	0
17	ADC1_ETRGINJ_RM	RW	Remap bit of ADC1 external trigger injection conversion 0: ADC1 external trigger injection conversion to EXTI15; 1: ADC1 external trigger injection conversion to TIM8_CH4.	0
16	TIM5CH4_RM	RW	Remap of timer5 channel 0: Default map. Remapping of timer5 channel4; 1: Remap. Map timer5 channel4 to LSI internal clock.	0
15	PD01_RM	RW	Remap bit of the PD0&PD1 pins. This bit can be read and written by the user. It controls whether the GPIO functions of PD0 and PD1 are remapped, i.e., PD0&PD1 is mapped to OSC_IN&OSC_OUT. When the main oscillator HSE is enabled (the system runs in the internal 8MHz RC oscillator), PD0 and PD1 can be mapped to OSC_IN and OSC_OUT pins. 0: The re-mapping of PD0 and PD1 is not conducted; 1: PD0 is mapped to OSC_IN, and PD1 is mapped to OSC_OUT.	0

[14:13]	CAN1_RM[1:0]	RW	CAN1alternate function remap bits, which can be read and written by the user. Control remapping of CAN_RX and CAN_TX:
			00: CAN1_RX is mapped to PA11, and CAN1_TX is mapped to PA12;
			10: CAN1_RX is mapped to PB8, and CAN1_TX is mapped to PB9;
			01: Reserved;
			11: CAN1_RX is mapped to PD0, and CAN1_TX is mapped to PD1.
				0

12	TIM4_RM	RW	Remap bit of timer4. This bit can be read and written by the user. It controls remapping of channel1 to channel4 of timer4: 0: Default map (CH1/PB6, CH2/PB7, CH3/PB8,	0

			CH4/PB9); 1: Remap (CH1/PD12, CH2/PD13, CH3/PD14, CH4/PD15).	

[11:10]	TIM3_RM[1:0]	RW	Remap bits of timer3. These bits can be read and written by the user. It controls remapping of channel1 to channel4 of timer3:
			00: Default map (CH1/PA6, CH2/PA7, CH3/PB0, CH4/PB1);
			01: Reserved;
			10: Partly map (CH1/PB4, CH2/PB5, CH3/PB0, CH4/PB1);
			11: Fully map (CH1/PC6, CH2/PC7, CH3/PC8, CH4/PC9).
			Note: Remapping does not affect TIM3_ETR on PD2.
				0

[9:8]	TIM2_RM[1:0]	RW	Remap bits of timer 2. These bits can be written and read by the user. It controls the mapping of channel1 to channel4 of timer2 and external trigger (ETR) on the GPIO port:
			00: Default map (ETR/PA0, CH1/PA0, CH2/PA1, CH3/PA2, CH4/PA3);
			01: Partly map (ETR/PA15, H1/PA15, CH2/PB3, CH3/PA2, CH4/PA3);
			10: Partly map (ETR/PA0, CH1/PA0, CH2/PA1, CH3/PB10, CH4/PB11);
			11: Fully map (ETR/PA15, CH1/PA15, CH2/PB3, CH3/PB10, CH4/PB11).
				0

[7:6]	TIM1_RM[1:0]	RW	Remap bit of timer 1. These bits can be written and read by the user. It controls the mapping of channel 1to channel4 of timer1, 1N to 3N, external trigger (ETR) and break input (BKIN) on the GPIO port:
			00: Default map (ETR/PA12, CH1/PA8, CH2/PA9, CH3/PA10, CH4/PA11, BKIN/PB12, CH1N/PB13, CH2N/PB14, CH3N/PB15);
			01: Partly map (ETR/PA12, CH1/PA8, CH2/PA9, CH3/PA10, CH4/PA11, BKIN/PA6, CH1N/PA7, CH2N/PB0, CH3N/PB1);
			10: Reserved;
			11: Fully map (ETR/PE7, CH1/PE9, CH2/PE11, CH3/PE13, CH4/PE14, BKIN/PE15, CH1N/PE8, CH2N/PE10, CH3N/PE12).
				0

[5:4]	USART3_RM[1:0]	RW	USART3 remap bits. These bits can be read and written by the user. It controls the mapping of CTS, RTS, CK, TX and RX alternate functions of USART3 on the GPIO port:	0

			00: Default map (TX/PB10, RX/PB11, CK/PB12, CTS/PB13, RTS/PB14);
			01: Partly remap (TX/PC10, RX/PC11, CK/PC12, CTS/PB13, RTS/PB14);
			10: Partial remapping (TX/PA13, RX/PA14, CK/PD10, CTS/PD11, RTS/PD12);
			11: Fully remap (TX/PD8, RX/PD9, CK/PD10, CTS/PD11, RTS/PD12).
			Note: CH32V20x_D6 and CH32F20x_D6 support only default map (00b).	

3	USART2_RM	RW	Remap bit of USART2. This bit can be read and written by the user. It controls the mapping of CTS, RTS, CK, TX and RX alternate functions of USART2 on the GPIO port: 0: Default map (CTS/PA0, RTS/PA1, TX/PA2, RX/PA3, CK/PA4); 1: Remap (CTS/PD3, RTS/PD4, TX/PD5, RX/PD6, CK/PD7). *Note: CH32V20x_D6 and CH32F20x_D6 support only default map (0b).*	0

2	USART1_RM	RW	USART1 map configuration low bit (used with the bit26 (USART1RM1) in the AFIO_PCFR2 register).
			00: Default map (CK/PA8, TX/PA9, RX/PA10, CTS/PA11, RTS/PA12);
			01: Remap (CK/PA8, TX/PB6, RX/PB7, CTS/PA11, RTS/PA12);
			10: Remap (CK/PA10, TX/PB15, RX/PA8, CTS/PA5, RTS/PA9);
			11: Remap (CK/PA5, TX/PA6, RX/PA7, CTS/PC4, RTS/PC5).
			Note: CH32V20x_D6 and CH32F20x_D6 support only default map (00b) and remap (01b).
				0

1	I2C1_RM	RW	Remap of I2C1. This bit can be read and written by the user. It controls the mapping of the SCL and SDA alternate functions of I2C1 at the GPIO port: 0: Default map (SCL/PB6, SDA/PB7); 1: Remap (SCL/PB8, SDA/PB9).	0
0	SPI1_RM	RW	Remap of SPI1. This bit can be read and written by the user. It controls the mapping of NSS, SCK, MISO and MOSI alternate functions of SPI1 at the GPIO port: 0: Default map (NSS/PA4, SCK/PA5, MISO/PA6, MOSI/PA7); 1: Remap (NSS/PA15, SCK/PB3, MISO/PB4, MOSI/PB5).	0

10.3.2.3 External Interrupt Configuration Register1 (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 bits. These bits are used to determine which port pin the external interrupt pin is mapped to: 0000: Pin x of the PA pin; 0001: Pin x of the PB pin; 0010: Pin x of the PC pin; 0011: Pin x of the PD pin; 0100: Pin x of the PE pin; Others: Reserved.	0

10.3.2.4 External Interrupt Configuration Register2 (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 bits, used to determine which port pin the external interrupt pin is mapped to: 0000: Pin x of the PA pin; 0001: Pin x of the PB pin; 0010: Pin x of the PC pin; 0011: Pin x of the PD pin; 0100: Pin x of the PE pin; Others: Reserved.	0

10.3.2.5 External Interrupt Configuration Register3 (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 bits. These bits are used to determine which port pin the external interrupt pin is mapped to: 0000: Pin x of the PA pin; 0001: Pin x of the PB pin; 0010: Pin x of the PC pin; 0011: Pin x of the PD pin; 0100: Pin x of the PE pin; Others: Reserved.	0

10.3.2.6 External Interrupt Configuration Register4 (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 bits. These bits are used to determine which port pin the external interrupt pin is mapped to: 0000: Pin x of the PA pin; 0001: Pin x of the PB pin; 0010: Pin x of the PC pin; 0011: Pin x of the PD pin; 0100: Pin x of the PE pin; Others: Reserved.	0

10.3.2.7 Remap Register 2 (AFIO_PCFR2)

Offset address: 0x1C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					USART1_RM1	UART8_RM[1:0]		UART7_RM[1:0]		UART6_RM[1:0]		UART5_RM[1:0]		UART4_RM[1:0]	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Reserved	FSMC_NADV	Reserved	TIM10_RM [1:0]	TIM9_RM [1:0]	TIM8_RM	Reserved

Bit	Name	Access	Description	Reset value
[31:27]	Reserved	RO	Reserved	0

26	USART1_RM1	RW	USART1 map configuration high bit (used with the bit2 (USART1RM) in the AFIO_PCFR1 register).
			00: Default map (CK/PA8, TX/PA9, RX/PA10, CTS/PA11, RTS/PA12);
			01: Remap (CK/PA8, TX/PB6, RX/PB7, CTS/PA11, RTS/PA12);
			10: Remap (CK/PA10, TX/PB15, RX/PA8, CTS/PA5, RTS/PA9);
			11: Remap (CK/PA5, TX/PA6, RX/PA7, CTS/PC4, RTS/PC5).
			Note: CH32V20x_D6 and CH32F20x_D6 support only default map (00b) and remap (01b).
				0

[25:24]	UART8_RM[1:0]	RW	UART8 remap
			00: Default map (TX/PC4, RX/PC5);
			01: Remap (TX/PA14, RX/PA15);
			1x: Remap (TX/PE14, RX/PE15).
				00b

[23:22]	UART7_RM[1:0]	RW	UART7 remap
			00: Default map (TX/PC2, RX/PC3);
			01: Remap (TX/PA6, RX/PA7);
			1x: Remap (TX/PE12, RX/PE13).
				00b

[21:20]	UART6_RM[1:0]	RW	UART6 remap
			00: Default map (TX/PC0, RX/PC1);
			01: Remap (TX/PB8, RX/PB9);
			1x: Remap (TX/PE10, RX/PE11).
				00b

[19:18]	UART5_RM[1:0]	RW	UART5 remap
			00: Default map (TX/PC12, RX/PD2);
			01: Remap (TX/PB4, RX/PB5);
			1x: Remap (TX/PE8, RX/PE9).
				00b

[17:16]	UART4_RM[1:0]	RW	UART4 remap 00: Default map (TX/PC10, RX/PC11); 01: Remap (TX/PB0, RX/PB1); 1x: Remap (TX/PE0, RX/PE1). *Note: Applied for CH32V30x_D8C, CH32V30x_D8, CH32V30x_D8W, CH32V20x_D8, CH32F20x_D8C, CH32F20x_D8 and CH32F20x_D8W.* x0:Default map (CK/PB2, TX/PB0, RX/PB1, CTS/PB3, RTS/PB4). x1:Remap (CK/PA6, TX/PA5, RX/PB5,	00b

			CTS/PA7, RTS/PA15). *Note: Applied for CH32V20x_D6 and CH32F20x_D6.*	
[15:11]	Reserved	RW	Reserved	0
10	FSMC_NADV	RW	FSMC_NADV remap 0: FSMC NADV is mapped to PB7; 1: FSMC NADV output disabled.	0
[9:7]	Reserved	RO	Reserved	0

[6:5]	TIM10_RM[1:0]	RW	TIM10 remap bits
			00:Default map (ETR/PC10, CH1/PB8, CH2/PB9, CH3/PC3, CH4/PC11, BKIN/PC12, CH1N/PA12, CH2N/PA13, CH3N/PA14);
			01: Partial map (ETR/PB11, CH1/PB3, CH2/PB4, CH3/PB5, CH4/PC15, BKIN/PB10, CH1N/PA5, CH2N/PA6, CH3N/PA7);
			1x: Full map (ETR/PD0, CH1/PD1, CH2/PD3, CH3/PD5, CH4/PD7, BKIN/PE2, CH1N/PE3, CH2N/PE4, CH3N/PE5).
				00b

[4:3]	TIM9_RM[1:0]	RW	TIM9 remap bits
			00: Default map (ETR/PA2, CH1/PA2, CH2/PA3, CH3/PA4, CH4/PC4, BKIN/PC5, CH1N/PC0, CH2N/PC1, CH3N/PC2);
			01: Partial map (ETR/PA2, CH1/PA2, CH2/PA3, CH3/PA4, CH4/PC14, BKIN/PA1, CH1N/PB0, CH2N/PB1, CH3N/PB2);
			1x: Full map (ETR/PD9, CH1/PD9, CH2/PD11, CH3/PD13, CH4/PD15, BKIN/PD14, CH1N/PD8, CH2N/PD10, CH3N/PD12).
				00b

2	TIM8_RM	RW	TIM8 remap bit
			0: Default map (ETR/PA0, CH1/PC6, CH2/PC7, CH3/PC8, CH4/PC9, BKIN/PA6, CH1N/PA7, CH2N/PB0, CH3N/PB1);
			1:Remap (ETR/PA0, CH1/PB6, CH2/PB7, CH3/PB8, CH4/PC13, BKIN/PB9, CH1N/PA13, CH2N/PA14, CH3N/PA15);
				0

[1:0]	Reserved	RW	Reserved	0

Chapter 11 Direct Memory Access Control (DMA)

This chapter applies to the whole family of CH32F2x, CH32V2x and CH32V3x.

Direct memory access controller (DMA) provides a high-speed data transmission method between peripheral and memory or between memories. Without CPU intervention, data can be moved quickly through DMA to save CPU resources for other operations.

Each channel of DMA controllers is dedicated to managing memory access requests from one or more peripherals. There is also an arbiter to coordinate the priority between the 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, APB1, APB2 and AHB 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 inputted to the DMA controller through logic or structure, and currently only a channel request is responded. The internal arbiter of the module selects the peripheral/memory access to be started according to the priority of the channel request.

In software management, the application program can independently configure the priority level for each channel by setting the PL[1:0] bits in the DMA_CFGRx register, including 4 levels: very high, high, medium, and low level. When the software setting levels between the channels are the same, the priority will be selected for the module according to the fixed hardware. The lower number of channel shall have the higher priority than the higher number.

2) DMA configuration

When the DMA controller receives a request signal, it will access the requested peripheral or memory to establish data transmission between the peripheral or memory and the memory. Mainly including the 3 following operation steps:

(1) Fetch data from the peripheral data register or the memory address indicated by the current peripheral/memory address register. The start address of the first transmission is the peripheral base address or memory address specified by the DMA_PADDRx or DMA_MADDRx register.

(2) Save data to the peripheral data register or the memory address indicated by the current peripheral/memory address register. The initial address during the first transmission is the peripheral base address or memory

address specified by the DMA_PADDRx or DMA_MADDRx register.

(3) Perform a decrement operation of the value in the DMA_CNTRx register, which indicates the number of unfinished transfer 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: The memory-to-memory mode does not require peripheral request signals. After configuring this mode (MEM2MEM=1) , the channel will be switched on (EN=1) to start data transmission. This mode does not support cycle mode.

The configuration process is as follows:

(1) Set the initial address of the peripheral register or the memory data address in the memory-to-memory mode (MEM2MEM=1) in the DMA_PADDRx register. When a DMA request occurs, this address will be the source or destination address of the data transmission.

(2) Set the memory data address in the DMA_MADDRx register. When a DMA request occurs, the transmitted data will be read from or written to this address.

(3) Set the number of data to be transmitted in the DMA_CNTRx register. After each data transmission, this value will decrease progressively.

(4) Set the channel priority through the PL[1:0] bits in the DMA_CFGRx register.

(5) In the DMA_CFGRx register, set the direction of data transmission, cycle mode, incremental mode of peripheral and memory, data width of peripheral and memory, DMA Half Transfer, DMA Transfer complete, and tDMA Transfer Error interrupt enable bit,

(6) Set the ENABLE bit in the DMA_CCRx register to enable channel x.

Note: The control bits in DMA_PADDRx/DMA_MADDRx/DMA_CNTRx register and DMA_CFGRx register such as data transmission direction (DIR), cycle mode (location), peripheral and memory incremental mode (MINC/PINC) can only be configured and written in when the DMA channel is switched off.

3) Cycle mode

Set the CIRC bit in the DMA_CFGRx register to 1, to enable the cyclic mode function of the channel data transmission. In the cycle mode, when the number of data transmission becomes 0, the content of the DMA_CNTRx register is automatically reloaded to its initial value, and the internal peripheral and memory address register is also reloaded to the initial address value set by the DMA_PADDRx and DMA_MADDRx registers, DMA operation continues until the channel or DMA mode is switched off.

4) DMA processing status

- DMA Half Transfer: Set the HTIFx bit in the corresponding DMA_INTFR register by the hardware. When the number of DMA transfer bytes is reduced to less than half of the initial set value, the DMA transfer half completion flag is generated. If HTIE is set in the DMA_CCRx register, an interrupt is generated. The hardware reminds the application program through this flag, and can prepare for a new round of data transmission.
- DMA Transfer complete: Set the TCIFx bit in the corresponding DMA_INTFR register by the hardware. When the number of DMA transfer bytes is reduced to 0, a DMA transmission completion flag is generated. If TCIE is set in the DMA_CCRx register, an interrupt is generated.

- DMA Transfer Error: Set the TEIFx bit in the corresponding DMA_INTFR register by the hardware. Reading and writing a reserved address area results in a DMA transmission error. Meanwhile, the module hardware automatically clears the EN bit in the DMA_CCRx register corresponding to the channel where the error is generated, and the channel is switched off. If TEIE is set in the DMA_CCRx register, an interrupt is generated.

When the application program queries the status of the DMA channel, it firstly accesses the GIFx bit in the DMA_INTFR register to determine which channel currently has a DMA event, and then process the specific DAM event content of the channel.

11.2.2 Programmable data transmission total size/data bit width/alignment

The total size of data transmitted in every round of each channel of DMA is programmable, up to 65535 times. The DMA_CNTRx register indicates the number of transfer bytes . When EN=0, write the setting value. After the DMA transmission channel is switched on during EN=1, this register becomes read-only, and the value decreases progressively after each transmission.

The transmission data value of peripherals and memory supports the automatic increment function of the address pointer, and the pointer increment is programmable. The first transmitted data address accessed by them is stored in the DMA_PADDRx and DMA_MADDRx registers. By setting the PINC bit or MINC bit in the DMA_CFGRx register to 1, the peripheral address auto-increment mode or memory address auto-increment mode can be enabled respectively. PSIZE[1: 0] bits are used to set the peripheral address fetch data size and address self-increment size. MSIZE[1:0] bits are used to set the memory address fetch data size and address self-increase size, including 3 options: 8-bit, 16-bit, 32-bit. The specific data transfer method is as shown in the table below:

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

Source bit width	Target bit width	Transferred 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 increment of the source address is aligned with the data bit width set at the source end, and the value is equal to the source data bit width ●The increment of the target address is aligned with the bit width of the target set data, and the value is equal to the bit width of the target data ●Principle for transferring the data to the target end by DMA: In case of insufficient data size, supplement 0 at high bit. In case of data size overflow, the high bit is removed. ●Data storage: Little-endian mode; low bytes are stored at the low address and high bytes are stored at high address
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	0x00/B1B0 0x02/B3B2	

			0x04/B5B4 0x06/B7B6	0x04/B5B4 0x06/B7B6	
16	32	4	0x00/B1B0 0x02/B3B2 0x04/B5B4 0x06/B7B6	0x00/0000B1B0 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 DMA request mapping

QingKe V4F MCUs (CH32V30x_D8C and CH32V30x_D8) and ARM ® CortexTM-M3 MCUs (CH32F20x_D8C and CH32F20x_D8). The DMA controllers provide 18 channels, as DMA1 provides 7 channels and DMA2 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 switched on or off independently.

Figure 11-1 DMA1 request mapping

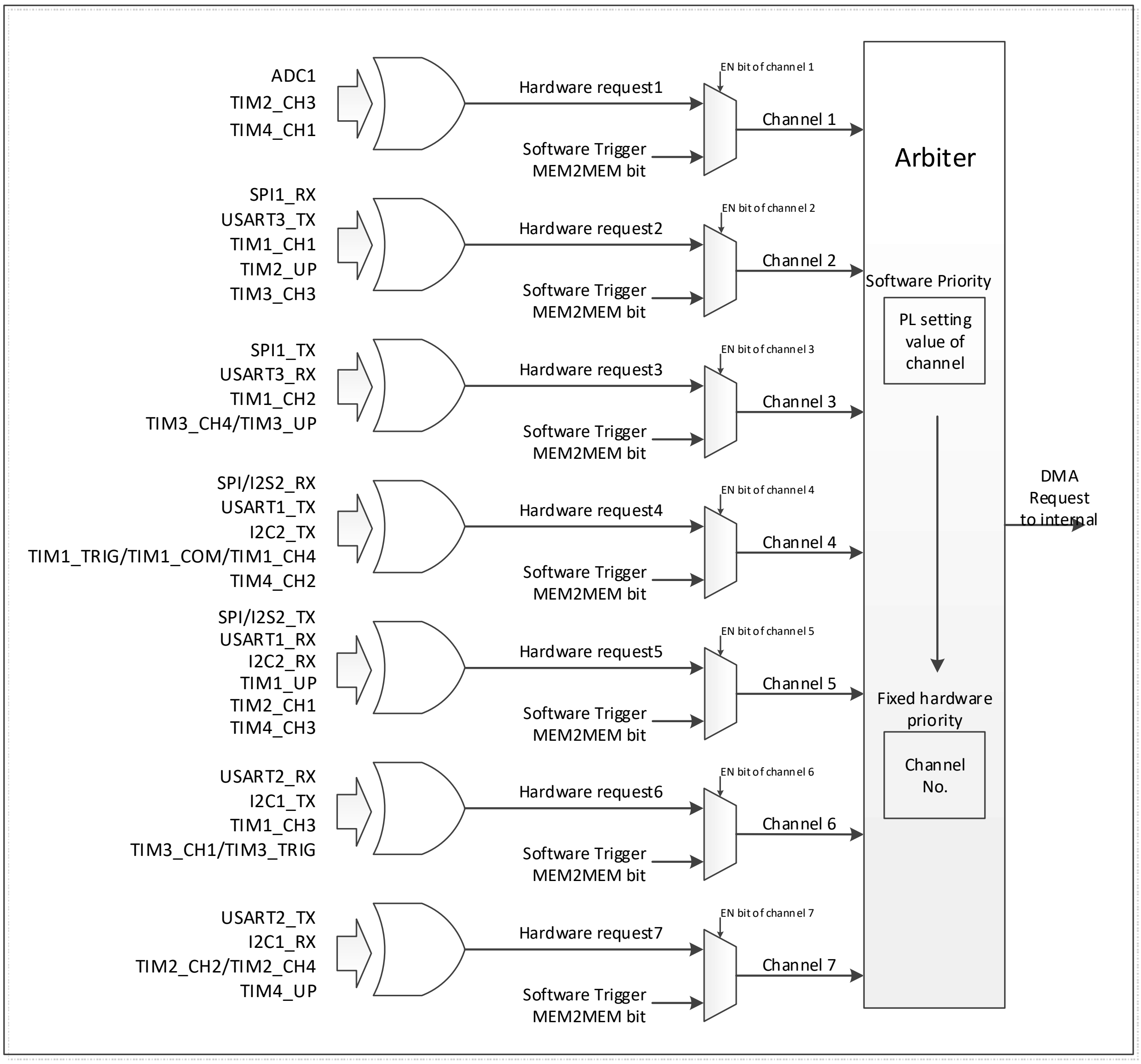


Table 11-2 DMA2 request mapping

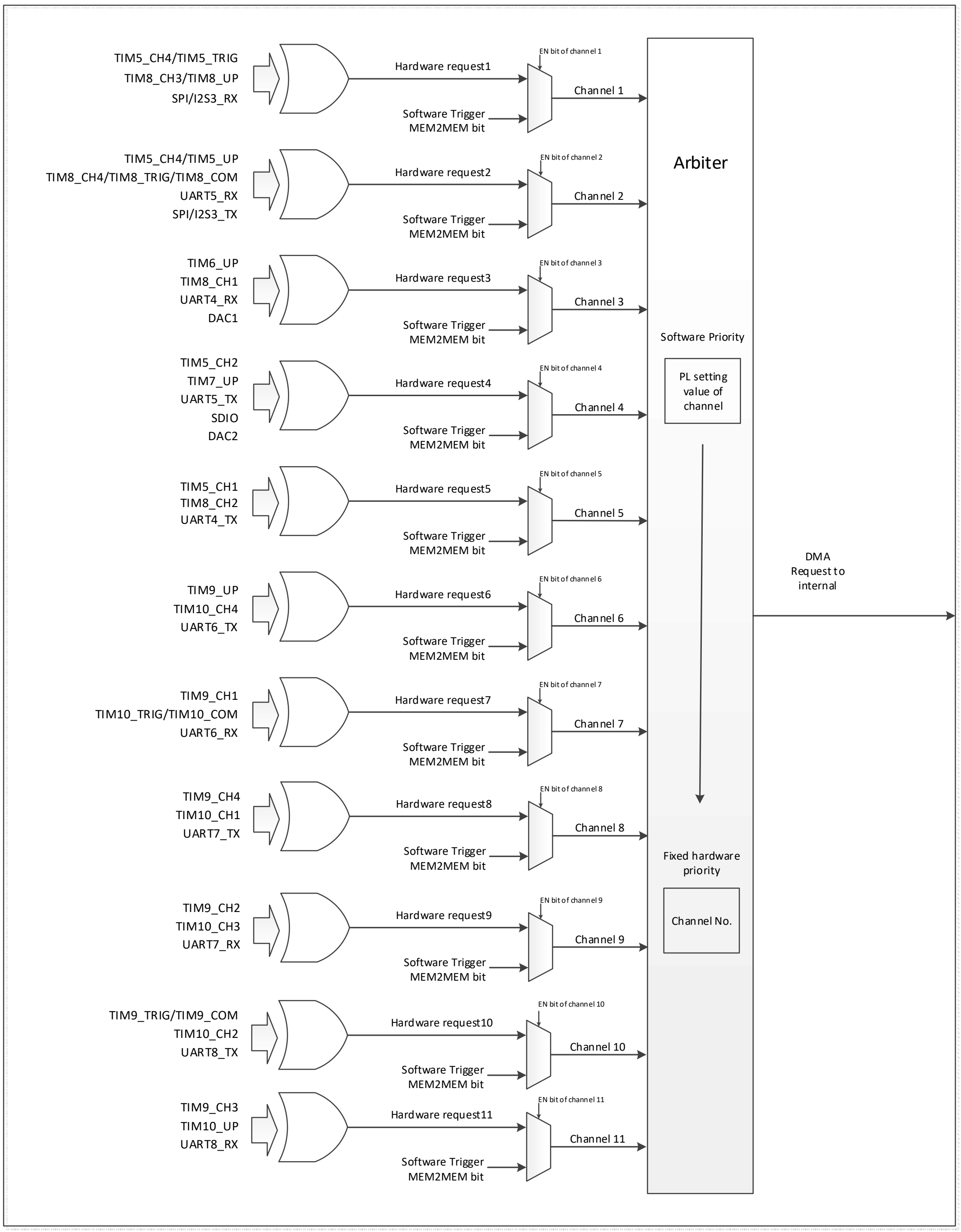


Table 11-2 Peripheral mapping table of each DMA1 channel

Peripheral	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7
ADC1	ADC1						
SPI1		SPI1_RX	SPI1_TX				
SPI/I2S2				SPI/I2S2_RX	SPI/I2S2_TX		
USART1				USART1_TX	USART1_RX		
USART2						USART2_RX	USART2_TX
USART3		USART3_TX	USART3_RX				
I2C1						I2C1_TX	I2C1_RX

I2C2				I2C2_TX	I2C2_RX		
TIM1		TIM1_CH1	TIM1_CH2	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_UP	TIM1_CH3	
TIM2	TIM2_CH3	TIM2_UP			TIM2_CH1		TIM2_CH2 TIM2_CH4
TIM3		TIM3_CH3	TIM3_CH4 TIM3_UP			TIM3_CH1 TIM3_TRIG	
TIM4	TIM4_CH1			TIM4_CH2	TIM4_CH3		TIM4_UP

Table 11-3 Peripheral mapping table of each DMA2 channel (I)

Peripheral	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7
TIM5	TIM5_CH4 TIM5_TRIG	TIM5_CH3 TIM5_UP		TIM5_CH2	TIM5_CH1		
TIM6			TIM6_UP				
TIM7				TIM7_UP			
TIM8	TIM8_CH3 TIM8_UP	TIM8_CH4 TIM8_TRIG TIM8_COM	TIM8_CH1		TIM8_CH2		
TIM9						TIM9_UP	TIM9_CH1
TIM10						TIM10_CH4	TIM10_TRIG TIM10_COM
UART4			USART4_RX		UART4_TX		
UART5		USART5_RX		UART5_TX			
UART6						UART6_TX	UART6_RX
UART7							
UART8							
SPI/I2S3	SPI/I2S3_RX	SPI/I2S3_TX					
SDIO				SDIO			
DAC1			DAC1				
DAC2				DAC2			

Table 11-4 Peripheral mapping table of each DMA2 channel (II)

Peripheral	Channel 8	Channel 9	Channel 10	Channel 11
TIM5				
TIM6				
TIM7				
TIM8				
TIM9	TIM9_CH4	TIM9_CH2	TIM9_TRIG TIM9_COM	TIM9_CH3
TIM10	TIM10_CH1	TIM10_CH3	TIM10_CH2	TIM10_UP
UART4				
UART5				
UART6				
UART7	UART7_TX	UART7_RX		

UART8			UART8_TX	UART8_RX
SPI/I2S3				
SDIO				
DAC1				
DAC2				

QingKe V4B MCUs (CH32V20x_D6) and ARM® Cortex™-M3 MCUs (CH32F20x_D6). The DMA controller provides 8 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 switched on or off independently.

Peripheral	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7	Channel 8
ADC1	ADC1							
SPI1		SPI1_RX	SPI1_TX					
SPI2				SPI2_RX	SPI2_TX			
USART1				USART1_TX	USART1_RX			
USART2						USART2_RX	USART2_TX	
USART3		USART3_TX	USART3_RX					
USART4	USART4_TX							USART4_RX
I2C1						I2C1_TX	I2C1_RX	
I2C2				I2C2_TX	I2C2_RX			
TIM1		TIM1_CH1	TIM1_CH2	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_UP	TIM1_CH3		
TIM2	TIM2_CH3	TIM2_UP			TIM2_CH1		TIM2_CH2 TIM2_CH4	
TIM3		TIM3_CH3	TIM3_CH4 TIM3_UP			TIM3_CH1 TIM3_TRIG		
TIM4	TIM4_CH1			TIM4_CH2	TIM4_CH3		TIM4_UP	

QingKe V4C MCUs (CH32V20x_D8W and CH32V20x_D8) and ARM® Cortex™-M3 MCUs (CH32F20x_D8W). The DMA controller provides 8 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 switched on or off independently.

Peripheral	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7	Channel 8
ADC1	ADC1							
SPI1		SPI1_RX	SPI1_TX					
SPI2				SPI2_RX	SPI2_TX			
USART1				USART1_TX	USART1_RX			
USART2						USART2_RX	USART2_TX	
USART3		USART3_TX	USART3_RX					
USART4	USART4_TX							USART4_RX
I2C1						I2C1_TX	I2C1_RX	
I2C2				I2C2_TX	I2C2_RX			

TIM1		TIM1_CH1	TIM1_CH2	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_UP	TIM1_CH3		
TIM2	TIM2_CH3	TIM2_UP			TIM2_CH1		TIM2_CH2 TIM2_CH4	
TIM3		TIM3_CH3	TIM3_CH4 TIM3_UP			TIM3_CH1 TIM3_TRIG		
TIM4	TIM4_CH1			TIM4_CH2	TIM4_CH3		TIM4_UP	
TIM5	TIM5_CH2		TIM5_CH3			TIM5_CH4	TIM5_CH1 TIM5_TRIG	TIM5_UP

11.3 Register description

Table 11-5 DMA1 registers

Name	Access address	Description	Reset value
R32_DMA1_INTFR	0x40020000	DMA1 interrupt flag register	0x00000000
R32_DMA1_INTFCR	0x40020004	DMA1 interrupt flag clear register	0x00000000
R32_DMA1_CFGR1	0x40020008	DMA1 channel1 configuration register	0x00000000
R32_DMA1_CNTR1	0x4002000C	DMA1 channel1 transferred data register	0x00000000
R32_DMA1_PADDR1	0x40020010	DMA1 channel1 peripheral address register	0x00000000
R32_DMA1_MADDR1	0x40020014	DMA1 channel1 memory address register	0x00000000
R32_DMA1_CFGR2	0x4002001C	DMA1 channel2 configuration register	0x00000000
R32_DMA1_CNTR2	0x40020020	DMA1 channel2 transferred data register	0x00000000
R32_DMA1_PADDR2	0x40020024	DMA1 channel2 peripheral address register	0x00000000
R32_DMA1_MADDR2	0x40020028	DMA1 channel2 memory address register	0x00000000
R32_DMA1_CFGR3	0x40020030	DMA1 channel3 configuration register	0x00000000
R32_DMA1_CNTR3	0x40020034	DMA1 channel3 transferred data register	0x00000000
R32_DMA1_PADDR3	0x40020038	DMA1 channel3 peripheral address register	0x00000000
R32_DMA1_MADDR3	0x4002003C	DMA1 channel3 memory address register	0x00000000
R32_DMA1_CFGR4	0x40020044	DMA1 channel4 configuration register	0x00000000
R32_DMA1_CNTR4	0x40020048	DMA1 channel4 transferred data register	0x00000000
R32_DMA1_PADDR4	0x4002004C	DMA1 channel4 peripheral address register	0x00000000
R32_DMA1_MADDR4	0x40020050	DMA1 channel4 memory address register	0x00000000
R32_DMA1_CFGR5	0x40020058	DMA1 channel5 configuration register	0x00000000
R32_DMA1_CNTR5	0x4002005C	DMA1 channel5 transferred data register	0x00000000
R32_DMA1_PADDR5	0x40020060	DMA1 channel5 peripheral address register	0x00000000
R32_DMA1_MADDR5	0x40020064	DMA1 channel5 memory address register	0x00000000
R32_DMA1_CFGR6	0x4002006C	DMA1 channel6 configuration register	0x00000000
R32_DMA1_CNTR6	0x40020070	DMA1 channel6 transferred data register	0x00000000
R32_DMA1_PADDR6	0x40020074	DMA1 channel6 peripheral address register	0x00000000
R32_DMA1_MADDR6	0x40020078	DMA1 channel6 memory address register	0x00000000
R32_DMA1_CFGR7	0x40020080	DMA1 channel7 configuration register	0x00000000
R32_DMA1_CNTR7	0x40020084	DMA1 channel7 transferred data register	0x00000000
R32_DMA1_PADDR7	0x40020088	DMA1 channel7 peripheral address register	0x00000000

R32_DMA1_MADDR7	0x4002008C	DMA1 channel7 memory address register	0x00000000
R32_DMA1_CFGR8	0x40020094	DMA1 channel8 configuration register	0x00000000
R32_DMA1_CNTR8	0x40020098	DMA1 channel8 transferred data register	0x00000000
R32_DMA1_PADDR8	0x4002009C	DMA1 channel8 peripheral address register	0x00000000
R32_DMA1_MADDR8	0x400200A0	DMA1 channel8 memory address register	0x00000000

Table 11-6 DMA2 registers

Name	Access address	Description	Reset value
R32_DMA2_INTFR	0x40020400	DMA2 interrupt flag register	0x00000000
R32_DMA2_INTFCR	0x40020404	DMA2 interrupt flag clear register	0x00000000
R32_DMA2_CFGR1	0x40020408	DMA2 channel1 configuration register	0x00000000
R32_DMA2_CNTR1	0x4002040C	DMA2 channel1 transferred data register	0x00000000
R32_DMA2_PADDR1	0x40020410	DMA2 channel1 peripheral address register	0x00000000
R32_DMA2_MADDR1	0x40020414	DMA2 channel1 memory address register	0x00000000
R32_DMA2_CFGR2	0x4002041C	DMA2 channel2 configuration register	0x00000000
R32_DMA2_CNTR2	0x40020420	DMA2 channel2 transferred data register	0x00000000
R32_DMA2_PADDR2	0x40020424	DMA2 channel2 peripheral address register	0x00000000
R32_DMA2_MADDR2	0x40020428	DMA2 channel2 memory address register	0x00000000
R32_DMA2_CFGR3	0x40020430	DMA2 channel3 configuration register	0x00000000
R32_DMA2_CNTR3	0x40020434	DMA2 channel3 transferred data register	0x00000000
R32_DMA2_PADDR3	0x40020438	DMA2 channel3 peripheral address register	0x00000000
R32_DMA2_MADDR3	0x4002043C	DMA2 channel3 memory address register	0x00000000
R32_DMA2_CFGR4	0x40020444	DMA2 channel4 configuration register	0x00000000
R32_DMA2_CNTR4	0x40020448	DMA2 channel4 transferred data register	0x00000000
R32_DMA2_PADDR4	0x4002044C	DMA2 channel4 peripheral address register	0x00000000
R32_DMA2_MADDR4	0x40020450	DMA2 channel4 memory address register	0x00000000
R32_DMA2_CFGR5	0x40020458	DMA2 channel5 configuration register	0x00000000
R32_DMA2_CNTR5	0x4002045C	DMA2 channel5 transferred data register	0x00000000
R32_DMA2_PADDR5	0x40020460	DMA2 channel5 peripheral address register	0x00000000
R32_DMA2_MADDR5	0x40020464	DMA2 channel5 memory address register	0x00000000
R32_DMA2_CFGR6	0x4002046C	DMA2 channel6 configuration register	0x00000000
R32_DMA2_CNTR6	0x40020470	DMA2 channel6 transferred data register	0x00000000
R32_DMA2_PADDR6	0x40020474	DMA2 channel6 peripheral address register	0x00000000
R32_DMA2_MADDR6	0x40020478	DMA2 channel6 memory address register	0x00000000
R32_DMA2_CFGR7	0x40020480	DMA2 channel7 configuration register	0x00000000
R32_DMA2_CNTR7	0x40020484	DMA2 channel7 transferred data register	0x00000000
R32_DMA2_PADDR7	0x40020488	DMA2 channel7 peripheral address register	0x00000000
R32_DMA2_MADDR7	0x4002048C	DMA2 channel7 memory address register	0x00000000
R32_DMA2_CFGR8	0x40020490	DMA2 channel8 configuration register	0x00000000
R32_DMA2_CNTR8	0x40020494	DMA2 channel8 transferred data register	0x00000000
R32_DMA2_PADDR8	0x40020498	DMA2 channel8 peripheral address register	0x00000000
R32_DMA2_MADDR8	0x4002049C	DMA2 channel8 memory address register	0x00000000
R32_DMA2_CFGR9	0x400204A0	DMA2 channel9 configuration register	0x00000000
R32_DMA2_CNTR9	0x400204A4	DMA2 channel9 transferred data register	0x00000000
R32_DMA2_PADDR9	0x400204A8	DMA2 channel9 peripheral address register	0x00000000

R32_DMA2_MADDR9	0x400204AC	DMA2 channel9 memory address register	0x00000000
R32_DMA2_CFGR10	0x400204B0	DMA2 channel10 configuration register	0x00000000
R32_DMA2_CNTR10	0x400204B4	DMA2 channel10 transferred data register	0x00000000
R32_DMA2_PADDR10	0x400204B8	DMA2 channel10 peripheral address register	0x00000000
R32_DMA2_MADDR10	0x400204BC	DMA2 channel10 memory address register	0x00000000
R32_DMA2_CFGR11	0x400204C0	DMA2 channel11 configuration register	0x00000000
R32_DMA2_CNTR11	0x400204C4	DMA2 channel11 transferred data register	0x00000000
R32_DMA2_PADDR11	0x400204C8	DMA2 channel11 peripheral address register	0x00000000
R32_DMA2_MADDR11	0x400204CC	DMA2 channel11 memory address register	0x00000000
R32_DMA2_EXTEM_INTFR	0x400204D0	DMA2 extend interrupt flag register	0x00000000
R32_DMA2_EXTEM_INTFCR	0x400204D4	DMA2 extend interrupt flag clear register	0x00000000

11.3.1 DMAx Interrupt Flag Register (DMAx_INTFR) (x=1/2)

Offset address: 0x00 + (x-1)*0x400

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	DMA Transfer Error flag of channel x (x=1/2/3/4/5/6/7/8): 1: A transmission error occurred on channel x; 0: No transmission error occurred on channel x. Set by hardware, and write CTEIFx bit by software to clear this flag.	0
30/26/22/18/14/10/6/2	HTIFx	RO	DMA Half Transfer flag of channel x (x=1/2/3/4/5/6/7/8): 1: A transmission half completion event has occurred on channel x; 0: No transmission half completion event has occurred on channel x. Set by hardware, and write CHTIFx bit by software to clear this flag.	0
29/25/21/17/13/9/5/1	TCIFx	RO	DMA Transfer complete flag of channel x (x=1/2/3/4/5/6/7/8): 1: A transmission completion event has occurred on channel x; 0: No transmission completion event has occurred on channel x. Set by hardware, and write CTCIFx bit by software to clear this flag.	0
28/24/20/16/12/8/4/0	GIFx	RO	Global interrupt flag of channel x (x=1/2/3/4/5/6/7/8): 1: TEIFx or HTIFx or TCIFx is generated on channel x; 0: No TEIFx or HTIFx or TCIFx is generated on channel	0

			x. It is set by hardware, and write CGIFx bit by software to clear this flag.	

Note: Channel8 is applied for CH32V20x_D8, CH32V20x_D8W, CH32F20x_D8W, CH32F20x_D6 and CH32V20x_D6.

11.3.2 DMAx Interrupt Flag Clear Register (DMAx_INTFCR) (x=1/2)

Offset address: 0x04 + (x-1)*0x400

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 DMA Transfer Error flag of channel x (x=1/2/3/4/5/6/7/8): 1: Clear the TEIFx flag in the DMA_INTFR register; 0: No effect.	0
30/26/22/18 /14/10/6/2	CHTIFx	WO	Clear DMA Half Transfer of channel x (x=1/2/3/4/5/6/7/8): 1: Clear the HTIFx flag in the DMA_INTFR register; 0: No effect.	0
29/25/21/17 /13/9/5/1	CTCIFx	WO	Clear DMA Transfer complete flag of channel x (x=1/2/3/4/5/6/7/8): 1: Clear the TCIFx flag in the DMA_INTFR register; 0: No effect.	0
28/24/20/16 /12/8/4/0	CGIFx	WO	Clear the global interrupt flag of channel x (x=1/2/3/4/5/6/7/8): 1: Clear the TEIFx/HTIFx/TCIFx/ GIFx flag in the DMA_INTFR register; 0: No effect.	0

Note: Channel8 is applied for CH32V20x_D8, CH32V20x_D8W, CH32F20x_D8W, CH32F20x_D6 and CH32V20x_D6.

11.3.3 DMAy Channelx Configuration Register (DMAy_CFGRx) (x=1/2/3/4/5/6/7/8, y=0/1)

Offset address: 0x08 + (x-1)*20 + (y-1)*0x400

31	30	29	28	27	26	25	24	23	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	MEM2MEM	PL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN

Bit	Name	Access	Description	Reset value
[31:15]	Reserved	RO	Reserved.	0
14	MEM2MEM	RW	Memory to memory mode enable: 1: Enable memory to memory mode; 0: Disable memory to memory mode.	0
[13:12]	PL[1:0]	RW	Channel priority level setting: 00: Low; 01: Medium; 10: High; 11: Very high.	0
[11:10]	MSIZE[1:0]	RW	Memory size setting: 00: 8 bits; 01: 16 bits; 10: 32 bits; 11: Reserved.	0
[9:8]	PSIZE[1:0]	RW	Peripheral size setting: 00: 8 bits; 01: 16 bits; 10: 32 bits; 11: Reserved.	0
7	MINC	RW	Memory increment mode enable: 1: Enable memory increment mode; 0: Disable memory increment mode.	0
6	PINC	RW	Peripheral increment mode enable: 1: Enable peripheral increment mode; 0: Disable peripheral increment mode.	0
5	CIRC	RW	DMA channel circular mode enable: 1: Enable circular mode; 0: Disable circular mode.	0
4	DIR	RW	Data transfer direction: 1: Read from memory; 0: Read from peripheral.	0
3	TEIE	RW	Transfer error interrupt enable control: 1: Enable transfer error interrupt; 0: Disable transfer error interrupt.	0
2	HTIE	RW	Half transfer interrupt enable control: 1: Enable transmission half interrupt; 0: Disable transmission half interrupt.	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

Note: Channel8 is applied for CH32V20x_D8, CH32V20x_D8W, CH32F20x_D8W, CH32F20x_D6 and CH32V20x_D6.

11.3.4 DMAy Channelx Transferred Data Register (DMAy_CNTRx) (x=1/2/3/4/5/6/7/8, y=0/1)

Offset address: 0x0C + (x-1)*20 + (y-1)*0x400

31	30	29	28	27	26	25	24	23	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 transfer, range: 0-65535. This register can only be written when the channel is not working (EN=0 of DMA_CFGRx). After the channel is enabled, the register will become read-only, indicating the number of remaining data to transfer (the register content decreases progressively after each DMA transmission). When the channel is in the cyclic mode, the contents of the register will be automatically reloaded to the previously configured value.	0

Note: This register can only be changed when EN=0. When EN=1, it is a read-only register, indicating the current number of data to be transmitted. When the register content is 0, no data transmission occurs regardless of whether the channel is switched on or not. Channel8 is applied for CH32V20x_D8, CH32V20x_D8W, CH32F20x_D8W, CH32F20x_D6 and CH32V20x_D6.

11.3.5 DMAy Channelx Peripheral Address Register (DMAy_PADDRx) (x=1/2/3/4/5/6/7/8)

Offset address: 0x10 + (x-1)*20 + (y-1)*0x400

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	Peripheral base address, as the source or destination address of peripheral data transmission. When PSIZE[1:0]='01' (16 bits), the module automatically ignores bit0, and the operation address is automatically 2 bytes aligned. When PSIZE[1:0]='10' (32 bits), the module automatically ignores bit[1:0], and the operation address is automatically 4 bytes aligned.	0

Note: This register can only be changed when EN=0, and cannot be written when EN=1. Channel8 is applied for CH32V20x_D8, CH32V20x_D8W, CH32F20x_D8W, CH32F20x_D6 and CH32V20x_D6.

11.3.6 DMAy Channelx Memory Address Register (DMAy_MADDRx) (x=1/2/3/4/5/6/7/8)

Offset address: 0x14 + (x-1)*20 + (y-1)*0x400

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 target address of data transmission. When MSIZE[1:0]='01' (16 bits), the module automatically ignores bit0, and the operation address is automatically 2 bytes aligned. When MSIZE[1:0]='10' (32 bits), the module automatically ignores the [1:0] bits, and the operation address is automatically4 bytes aligned.	0

Note: This register can only be changed when EN=0, and cannot be written when EN=1. Channel*8 is applied for CH32V20x_D8, CH32V20x_D8W, CH32F20x_D8W, CH32F20x_D6 and CH32V20x_D6.*

11.3.7 DMA2 Channelx Configuration Register (DMA2_CFGRx) (x=8/9/10/11)

Offset address: 0x490 + (x-8)*16

31	30	29	28	27	26	25	24	23	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	MEM2MEM	PL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN

Bit	Name	Access	Description	Reset value
[31:15]	Reserved	RO	Reserved	0
14	MEM2MEM	RW	Memory to memory mode enable: 1: Enable memory to memory mode; 0: Disable memory to memory mode.	0
[13:12]	PL[1:0]	RW	Channel priority level setting: 00: Low; 01: Medium; 10: High; 11: Very high.	0
[11:10]	MSIZE[1:0]	RW	Memory size setting: 00: 8 bits; 01: 16 bits; 10: 32 bits; 11: Reserved.	0
[9:8]	PSIZE[1:0]	RW	Peripheral size setting: 00: 8 bits; 01: 16 bits; 10: 32 bits; 11: Reserved.	0
7	MINC	RW	Memory increment mode enable: 1: Enable memory increment mode; 0: Disable memory increment mode.	0
6	PINC	RW	Peripheral increment mode enable: 1: Enable peripheral increment mode; 0: Disable peripheral increment mode.	0
5	CIRC	RW	DMA channel circular mode enable: 1: Enable circular mode; 0: Disable circular mode.	0
4	DIR	RW	Data transfer direction:	0

			1: Read from memory; 0: Read from peripheral.	
3	TEIE	RW	Transfer error interrupt enable control: 1: Enable transfer error interrupt; 0: Disable transfer error interrupt.	0
2	HTIE	RW	Half transfer interrupt enable control: 1: Enable transmission half interrupt; 0: Disable transmission half interrupt.	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 is cleared to 0 automatically by hardware, and channel is disabled.	0

Note: Applied for CH32F20x_D8, CH32F20x_D8C, CH32V30x_D8 and CH32V30x_D8C.

11.3.8 DMA2 Channelx Transferred Data Register (DMA2_CNTRx) (x=8/9/10/11)

Offset address: 0x494 + (x-8)*16

31	30	29	28	27	26	25	24	23	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 to transfer, range: 0-65535. This register can only be written when the channel is not working (EN of DMA_CFGRx =0). After the channel is enabled, the register will become read-only, indicating the number of remaining transfer bytes (the register content decreases progressively after each DMA transmission). When the channel is in the cyclic mode, the contents of the register will be automatically reloaded to the previously configured value.	0

Note: This register can only be changed when EN=0. When EN=1, it is a read-only register, indicating the current number of transfer bytes . When the register content is 0, no data transmission occurs regardless of whether the channel is switched on or not. Applied for CH32F20x_D8, CH32F20x_D8C, CH32V30x_D8 and CH32V30x_D8C.

11.3.9 DMA2 Channelx Peripheral Address Register (DMA2_PADDRx) (x=8/9/10/11)

Offset address: 0x498 + (x-8)*16

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	Peripheral base address, as the source or target address of peripheral data transmission. When PSIZE[1:0]='01' (16 bits), the module automatically ignores bit0, and the operation address is automatically 2 bytes aligned. When PSIZE[1:0]='10' (32 bits), the module automatically ignores bit[1:0], and the operation address is automatically 4 bytes aligned.	0

Note: This register can only be changed when EN=0, and cannot be written when EN=1. Applied for CH32F20x_D8, CH32F20x_D8C, CH32V30x_D8 and CH32V30x_D8C.

11.3.10 DMA2 Channelx Memory Address Register (DMA2_MADDRx) (x=8/9/10/11)

Offset address: 0x49C + (x-8)*16

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 target address of data transmission. When MSIZE[1:0]='01' (16 bits), the module automatically ignores bit0, and the operation address is automatically 2 bytes aligned. When MSIZE[1:0]='10' (32 bits), the module automatically ignores bit[1:0], and the operation address is automatically 4 bytes aligned.	0

Note: This register can only be changed when EN=0, and cannot be written when EN=1. Applied for CH32F20x_D8, CH32F20x_D8C, CH32V30x_D8 and CH32V30x_D8C.

11.3.11 DMA2 Extend Interrupt Status Register (DMA2_EXTEM_INTFR)

Offset address: 0x4D0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TEIF11	HTIF11	TCIF11	GIF11	TEIF10	HTIF10	TCIF10	GIF10	TEIF9	HTIF9	TCIF9	GIF9	TEIF8	HTIF8	TCIF8	GIF8

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
15/11/7/3	TEIFx	RO	DMA Transfer Error flag of channelx (x=8/9/10/11): 1: A transfer error occurred on channel x; 0: No transfer error occurred on channel x. Set by hardware, and write CTEIFx bit by software to clear	0

			this flag.	
14/10/6/2	HTIFx	RO	DMA Half Transfer flag of channelx (x=8/9/10/11): 1: A transfer half completion event has occurred on channel x; 0: No transfer half completion event has occurred on channel x. Set by hardware, and write CHTIFx bit by software to clear this flag.	0
13/9/5/1	TCIFx	RO	DMA Transfer complete of channelx (x=8/9/10/11): 1: A transfer completion event has occurred on channel x; 0: No transfer completion event has occurred on channel x. Set by hardware, and write CTCIFx bit by software to clear this flag.	0
12/8/4/0	GIFx	RO	Global interrupt flag of channelx (x=8/9/10/11): 1: TEIFx or HTIFx or TCIFx is generated on channel x; 0: No TEIFx or HTIFx or TCIFx is generated on channel x. Set by hardware, and write CGIFx bit by software to clear this flag.	0

Note: Applied for CH32F20x_D8, CH32F20x_D8C, CH32V30x_D8 and CH32V30x_D8C.

11.3.12 DMA2 Extend Interrupt Flag Clear Register (DMA2_EXTEM_INTFCR)

Offset address: 0x4D4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CTEIF11	CHTIF11	CTCIF11	CGIF11	CTEIF10	CHTIF10	CTCIF10	CGIF10	CTEIF9	CHTIF9	CTCIF9	CGIF9	CTEIF8	CHTIF8	CTCIF8	CGIF8

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
15/11/7/3	CTEIFx	WO	Clear DMA Transfer Error flag of channel x (x=8/9/10/11): 1: Clear the TEIFx flag in the DMA_INTFR register; 0: No effect.	0
14/10/6/2	CHTIFx	WO	Clear DMA Half Transfer flag of channel x (x=8/9/10/11): 1: Clear the HTIFx flag in the DMA_INTFR register; 0: No effect.	0
13/9/5/1	CTCIFx	WO	Clear DMA Transfer complete flag of channel x (x=8/9/10/11): 1: Clear the TCIFx flag in the DMA_INTFR register; 0: No effect.	0
12/8/4/0	CGIFx	WO	Clear the global interrupt flag of channel x (x=8/9/10/11): 1: Clear the TEIFx/HTIFx/TCIFx/ GIFx flag in the	0

			DMA_INTFR register; 0: No effect.	

Note: Applied for CH32F20x_D8, CH32F20x_D8C, CH32V30x_D8 and CH32V30x_D8C.

Chapter 12 Analog-to-digital Converter (ADC)

This chapter applies to the whole family of CH32F2x, CH32V2x and CH32V3x.

The ADC module contains 2 12-bit successive approximation analog-to-digital converters with a maximum input clock of 14MHz. It supports sampling sources of 16 external channels and 2 internal sources. It can be performed in single, continuous, automatic scan, discontinuous, external trigger and dual sample modes. The analog watchdog function can be used to monitor whether the channel voltage is within the threshold range.

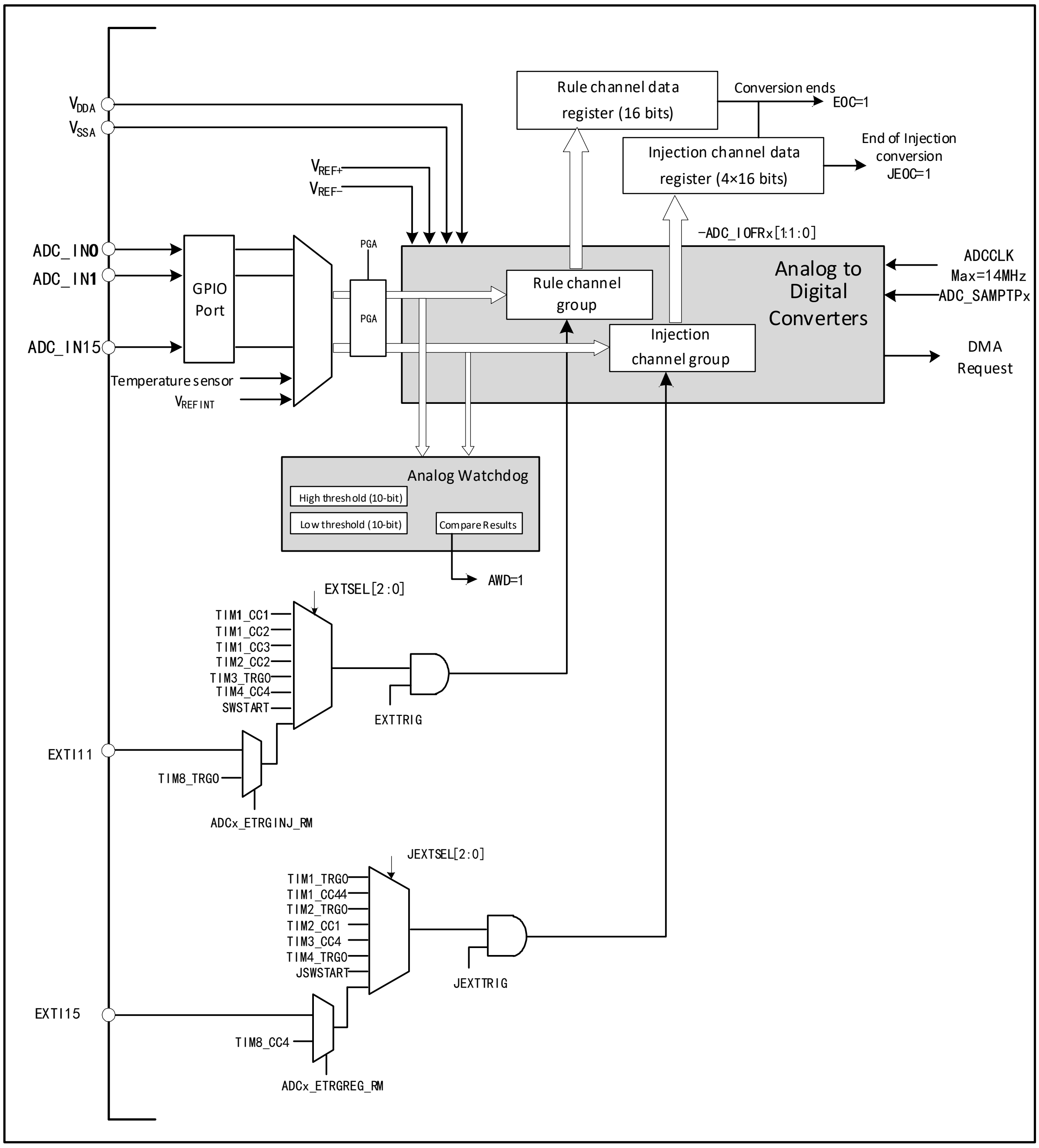
12.1 Main features

- 12-bit resolution
- 16 external channels and 2 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
- Sampling time can be programmed separately per channel
- Both regular conversion and injected conversion support external trigger
- Analog watchdog monitors the channel voltage, and has self-calibration function
- Dual mode
- ADC channel input range: $0 \leq V_{IN} \leq V_{DDA}$
- Adjustable input gain, to implement small signal amplification and sampling

12.2 Functional description

12.2.1 Module structure

Figure 12-1 ADC module block diagram



12.2.2 ADC configuration

1) Module power on

The ADON bit in the ADC_CTLR2 register is 1, indicating that the ADC module is powered on. When the ADC module enters the power-on status (ADON=1) from the power-down mode (ADON=0), it needs to delay a period of time t_{STAB} as the module stabilization time. Afterwards, write the ADON bit as 1 again, to serve as the start signal for software to start ADC conversion. By clearing the ADON bit to 0, you can terminate the current conversion and place the ADC module in power-down mode. In this status, the ADC consumes almost no power.

2) Sample clock

The register operation of the module is based on the PCLK2 (APB2 bus) clock. The clock reference ADCCLK of the conversion unit is synchronized with PCLK2. The frequency division is configured by ADCPRE[1:0] in the RCC_CFGR0 register, and the maximum cannot exceed 14MHz.

3) Channel configuration

The ADC module provides 18 channels of sampling sources, including 16 external channels and 2 internal channels. They can be configured into 2 conversion groups: regular group and injected group, in order to realize the group conversion formed by a series of conversions on any number of channels in any order.

Conversion group:

- Regular group: Composed of up to 16 conversions. The regular channels and their conversion sequence are set in the ADC_RSQRx register. The total number of conversions in the regular group shall be written into RLEN[3:0] in the ADC_RSQR1 register.
- Injected group: Composed of up to 4 conversions. The injected channels and their conversion sequence are set in the ADC_ISQR register. The total number of conversions in the injected group shall be written into ILEN[1:0] in the ADC_ISQR register.

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

Two internal channels:

- Temperature sensor: Connect ADC_IN16 channel to measure the temperature (TA) around the device.
- Internal reference voltage (V_{REFINT}): Connect the ADC_IN17 channel.

4) Calibration

The ADC is provided with a built-in self-calibration mode. After the calibration link, the accuracy error caused by the change of the internal capacitor bank can be greatly reduced. During calibration, an error correction code is calculated on each capacitor to eliminate the error generated on each capacitor in the subsequent conversion.

Initialize the calibration register by writing the RSTCAL bit of the ADC_CTLR2 register to 1. The initiation is completed when the RSTCAL hardware is cleared. Set the CAL bit to start the calibration function. Once the calibration is completed, the hardware will automatically clear the CAL bit and save the calibration code in ADC_RDATAR. Then, the normal conversion function can be used. It is recommended to perform an ADC calibration when the ADC module is powered on.

Note: Before starting calibration, you must ensure that the ADC module is in the power-on status (ADON=1) for more than 2 ADC clock cycles at least.

5) Programmable sample time

Several ADCCLK cycles are used to sample the input voltage. The number of sampling cycles of the channel can be changed by the SMPx[2:0] bits in the ADC_SAMPTR1 and ADC_SAMPTR2 registers. Each channel can be sampled at a different time.

The total conversion time is calculated as follows:

$$T_{CONV} = \text{Sampling time} + 11T_{ADCCLK}$$

The regular channel conversion of ADC supports DMA function. The converted value of the regular channel is stored in the only data register ADC_RDATAR. To prevent continuous conversion of multiple regular channels, the DMA function of ADC can be enabled for the data not timely removed in the ADC_RDATAR register. The hardware will generate a DMA request when the conversion of the regular channel is completed (EOC bit is set), and transmit the converted data from the ADC_RDATAR register to the destination address specified by the user.

After the channel configuration of the DMA controller module is completed, write the DMA bit of the ADC_CTLR2 register to 1 to enable the DMA function of the ADC.

Note: The injected group conversion does not support DMA function.

6) Data alignment

The data storage alignment method after ADC conversion is selected for the ALIGN bit in the ADC_CTLR2 register. 12-bit data supports left alignment and right alignment modes.

The data register ADC_RDATAR of the regular group channel saves the actual converted 12-bit digital value; while the data register ADC_IDATARx of the injected group channel is the value written after the actual converted data is subtracted from the defined offset of the ADC_IOFRx register, the value may be positive or negative, so there will be a sign bit (SIGNB).

Figure 12-2 Data left alignment

Regular 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

Regular 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 ADC conversion start event can be triggered by an external event. If the REXTTRI or IEXTTRIG bit in the ADC_CTLR2 register is set, the conversion of the regular group or the injected group channel can be triggered by an external event, respectively. At this time, the configuration of the REXTSEL[2:0] and IEXTSEL[2:0] bits determines the external event source of the rule group and the injected group.

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

Table 12-1 External trigger sources of regular group channel

REXTSEL[2:0]	Trigger source	Type
000	CC1 event of timer1	Internal signal from on-chip timers
001	CC2 event of timer1	
010	CC3 event of timer1	
011	CC2 event of timer2	
100	TRGO event of timer3	

101	CC4 event of timer4	
110	EXTI line11/TIM8_TRGO	From external pin/internal timer signal
111	SWSTART bit set by software	Software control bit

Table 12-2 External trigger sources of injected group channel

IEXTSEL[2:0]	Trigger source	Type
000	TRGO event of timer1	Internal signal from on-chip timers
001	CC4 event of timer1	
010	TRGO event of timer2	
011	CC1 event of timer2	
100	CC4 event of timer3	
101	TRGO event of timer4	
110	EXTI line15/TIM8_CC4	From external pin/internal timer signal
111	JSWSTART bit set to 1 by software	Software control bit

12.2.4 Conversion mode

Table 12-3 Conversion mode combination

ADC_CTLR1 and ADC_CTLR2 register control bits					ADC conversion mode
CONT	SCAN	RDISCEN/IDISCEN	IAUTO	Start event	
0	0	0	0	ADON bit set to 1	Single single-channel mode: The single conversion is performed through a regular channel.
				External trigger mode	Single single-channel mode: A single conversion is performed through one of the regular channels or injected channels.
	1	0	0	ADON bit set to 1 or external trigger mode	Single scan mode: Perform a single conversion on all selected regular group channels (ADC_RSQRx) or all injected group channels (ADC_ISQR) in sequence. Trigger injected mode: When the regular group channel is converted, all conversions of the injected group channel can be inserted, and then the regular group channel conversion is continued; but the regular group channel conversion is not inserted when injected group channel is converted.
			1	ADON bit set to 1 or external trigger mode	Single scan mode: Perform a single conversion on all selected regular group channels (ADC_RSQRx) or all injected group channels (ADC_ISQR) in sequence.

					Automatic injected mode: After regular group channel is converted, the injected group channel will be automatically converted. *Note: The external trigger signal injected into the channel is not allowed to appear during the conversion process.*
	0	1 (RDISCEN and IDISCEN cannot be 1 at the same time)	0	External trigger mode	Single discontinuous mode: Whenever an event is started, a short sequence (the number defined by DISCNUM[2:0]) of channel number conversion will be performed, and the event will be restarted until the conversion of all selected channels is completed. *Note: The mode control bits selected for the regular group and the injected group are IDISCEN and RDISCEN respectively. The discontinuous mode cannot be configured for the regular group and the injected group at the same time, and the discontinuous mode can only be used for single conversion.*
			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: After each round, a new round of conversion will be repeated, and it can be terminated until CONT is cleared to 0.
	1	0	0		
			1		

Note: The external trigger event of regular group and injected group is different, and the 'ACON' bit can only start the channel conversion of the regular group, so the start event of the channel conversion of the regular group and the injected group is independent.

1) Single single-channel conversion mode

In this mode, only one conversion is performed for the current channel. The first channel in the regular group or injected group is converted in this mode. It can be started up by setting the ADON bit in ADC_CTLR2 register to 1 (only applicable to regular channel), or by an external trigger (applicable to regular channel or injected channel). Once the conversion of the selected channels is completed:

If a regular group channel is converted, the conversion data is saved in the 16-bit ADC_RDATAR register, and the EOC flag is set. If the EOCIE bit is set, the ADC interrupt is triggered.

If the injected group channel is converted, the conversion data is stored in the 16-bit ADC_IDATAR1 register, and the EOC and IEOC flags is set. If the IEOCIE or EOCIE bit is set, the ADC interrupt is triggered.

2) Single scan mode conversion

Enter the ADC scan mode by setting the SCAN bit in the ADC_CTLR1 register to 1. This mode is used to scan a group of analog channels, and perform a single conversion for all channels selected by ADC_RSQRx register (for regular channels) or ADC_ISQR (for injected channels) one by one. When the current channel conversion ends, the next channel in the same group is automatically converted.

In the scan mode, according to the status of the IAUTO bit, it is divided into trigger injected mode and automatic injected mode.

- Trigger injection

The IAUTO bit is 0. When a trigger event for the channel conversion of the injected group occurs during the scanning of the regular group channel, the current conversion is reset, and the sequence of the injected channel is carried out in a single scan mode. After the scanning and conversion of all selected injected group channels are completed, the previous interrupted regular group channel conversion is restored.

If a regular channel start event occurs currently while scanning the injected group channel sequence, the injected group conversion is not interrupted, but the regular sequence conversion is executed after the injected sequence conversion is completed.

Note: When using triggered injected conversion, you must ensure that the interval of the trigger event is longer than the injected sequence. For example, the total conversion time to complete the injected sequence is 28 ADCCLKs, so the minimum event interval time for triggering the injected channel is 29 ADCCLKs.

- Automatic injection

The IAUTO bit is 1. After scanning all the channels selected by the regular group, the conversion of the channels selected by the injected group is automatically performed. This method can be used to convert up to 20 conversion sequences in the ADC_RSQRx and ADC_ISQR registers.

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

Note: For the ADC clock prescale factor(ADCPRE[1:0])of 4 to 8, when switching from regular conversion to injected sequence or from injected conversion to regular sequence, 1 ADCCLK interval will be inserted automatically; when the ADC clock prescale factor is 2, there is a delay of 2 ADCCLK intervals.

3) Single interval mode conversion

Enter the discontinuous mode of the regular group or injected group by setting either the RDISCEN or IDISCEN bit in the ADC_CTLR1 register to 1. This mode differs from scanning a complete set of channels in scan mode, but divides a set of channels into multiple short sequences, and each external trigger event executes a short sequence of scan conversion.

The length of the short sequence n (n<=8) is defined by the DISCNUM[2:0] bits in the ADC_CTLR1 register. When RDISCEN is 1, it is the discontinuous mode of the regular group. The total length to be converted is defined by the RLEN[3:0] bits in the ADC_RSQR1 register. When IDISCEN is 1, it is the discontinuous mode of the injected group, and the total length to be converted is defined by the ILEN[1:0] bits in the ADC_ISQR register. The regular group and injected group cannot be set to discontinuous mode at the same time.

Example of regular group discontinuous mode:

RDISCEN=1, RLEN[3:0]=8, channels to be converted=1, 3, 2, 5, 8, 4, 10, 6
For the first external trigger, conversion sequence: 1, 3, 2
For the second external trigger, conversion sequence: 5, 8, 4
For the third external trigger, conversion sequence: 10, 6, and the EOC event is generated in the meantime
For the fourth external trigger, conversion sequence: 1, 3, 2

Example of injected group discontinuous mode:

IDISCEN=1, DISCNUM[2:0]=1, ILEN[1:0]=3, channels to be converted=1, 3, 2

For the first external trigger, conversion sequence: 1

For the second external trigger, conversion sequence: 3

For the third external trigger, conversion sequence: 2, and the EOC and IEOC events are generated in the meantime

For the fourth external trigger, conversion sequence: 1

Note: 1. When switching a regular group or injected group in the discontinuous mode, it will not automatically start from the beginning after the conversion sequence ends. When all subgroups are converted, the next trigger event will start the conversion of the first subgroup.

2. Automatic injection (IAUTO=1) and intermittent mode cannot be used at the same time.

3. The discontinuous mode cannot be set for the regular group and the injected group at the same time, and the discontinuous mode can only be used for single conversion.

4. In the intermittent mode of the injection group, the number of injection channels to be converted after the external trigger is 1.

4) Continuous conversion

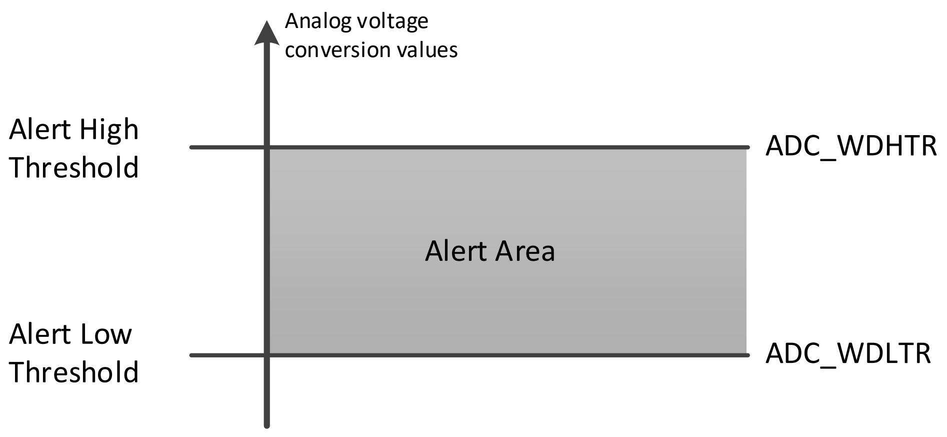
Enter the ADC continuous conversion mode by setting the CONT bit in the ADC_CTLR2 register to 1. In this mode, another conversion is started immediately after the previous ADC conversion is completed. The conversion will not stop on the last channel of the selected group, but will continue from the first channel of the selected group again.

Startup events include external trigger events and ADON bit set to 1. Based on several conversion methods in the previous single mode, it also includes continuous single-channel conversion and continuous scan mode (trigger injection or automatic injection) conversion.

12.2.5 Analog watchdog

If the analog voltage converted by the ADC is lower than the low threshold or higher than the high threshold, the AWD analog watchdog status bit will be set. The threshold setting is located in the low 12 active bits in the ADC_WDHTR and ADC_WDLTR registers. By setting the AWDIE bit in the ADC_CTLR1 register, the corresponding interrupt is allowed to be generated.

Figure 12-4 Analog watchdog threshold area



Configure the AWDSGL, RAWDEN, IAWDEN and AWDCH[4:0] bits in the ADC_CTLR1 register to select the channel for analog watchdog vigilance. The specific relationship is shown in the following table:

Table 12-4 Analog Watchdog Channel Selection

Analog watchdog vigilance channel	ADC_CTLR1 register control bit			
	AWDSGL	RAWDEN	IAWDEN	AWDCH[4:0]
Non-vigilance	Ignore	0	0	Ignore
All injected channels	0	0	1	Ignore
All regular channels	0	1	0	Ignore
All injected and regular 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 Temperature sensor

The module has a built-in temperature sensor, which is connected to the ADC_INT16 channel. The voltage output by the sensor is converted into a digital value through the ADC to feed back the surrounding temperature of the device. The recommended sampling time is 17.1us. The output voltage of the temperature sensor changes linearly with temperature. Due to production differences, the slope and offset of the linear change curve are different. Therefore, the internal temperature sensor is more suitable for detecting temperature changes, rather than measuring absolute temperature. If you need to measure accurate temperature, you shall use an external temperature sensor.

By setting the TSVREFE bit in the ADC_CTLR2 register to 1, wake up the ADC internal sampling channel. The ADC temperature sensor channel conversion is started up by the software or external trigger to read the data results (mV). The conversion formula of digital value and temperature (°C) is as follows:

Temperature (°C) = $((V_{SENSE}-V_{25})/\text{Avg\_Slope})+25$

V25: The voltage value of the temperature sensor at 25°C

Avg_Slope: Average slope of temperature and V_{SENSE} curve (mV/°C)

Refer to the actual values of V_{25} and Avg_Slope in the electrical characteristics chapter of this manual.

Note: A setup time is required for the internal temperature sensor power-on (TSVREFE bit is changed to 1 from 0) and a setup time is also required for ADC module power-on (ADON bit is changed to 1 from 0), so in order to shorten the waiting time, you can set ADON and TSVREFE bits at the same time.

12.2.7 Dual ADC mode

In devices with 2 ADC modules, the 2 ADC modules can be used together to implement dual ADC mode. In dual ADC mode, ADC1 is the master ADC, and ADC2 is the slave ADC. The conversion is triggered alternately or simultaneously, depending on the mode selected by the DAULMODE[3:0] bits in the ADC1_CTLR1 register

Note: In dual ADC mode, when configuring conversion to be triggered by an external event, the external triggers must be enabled on both master and slave ADCs and the user must set the corresponding trigger for the master ADC and set a software trigger for the slave ADC, to prevent spurious triggers to start unwanted slave conversion.

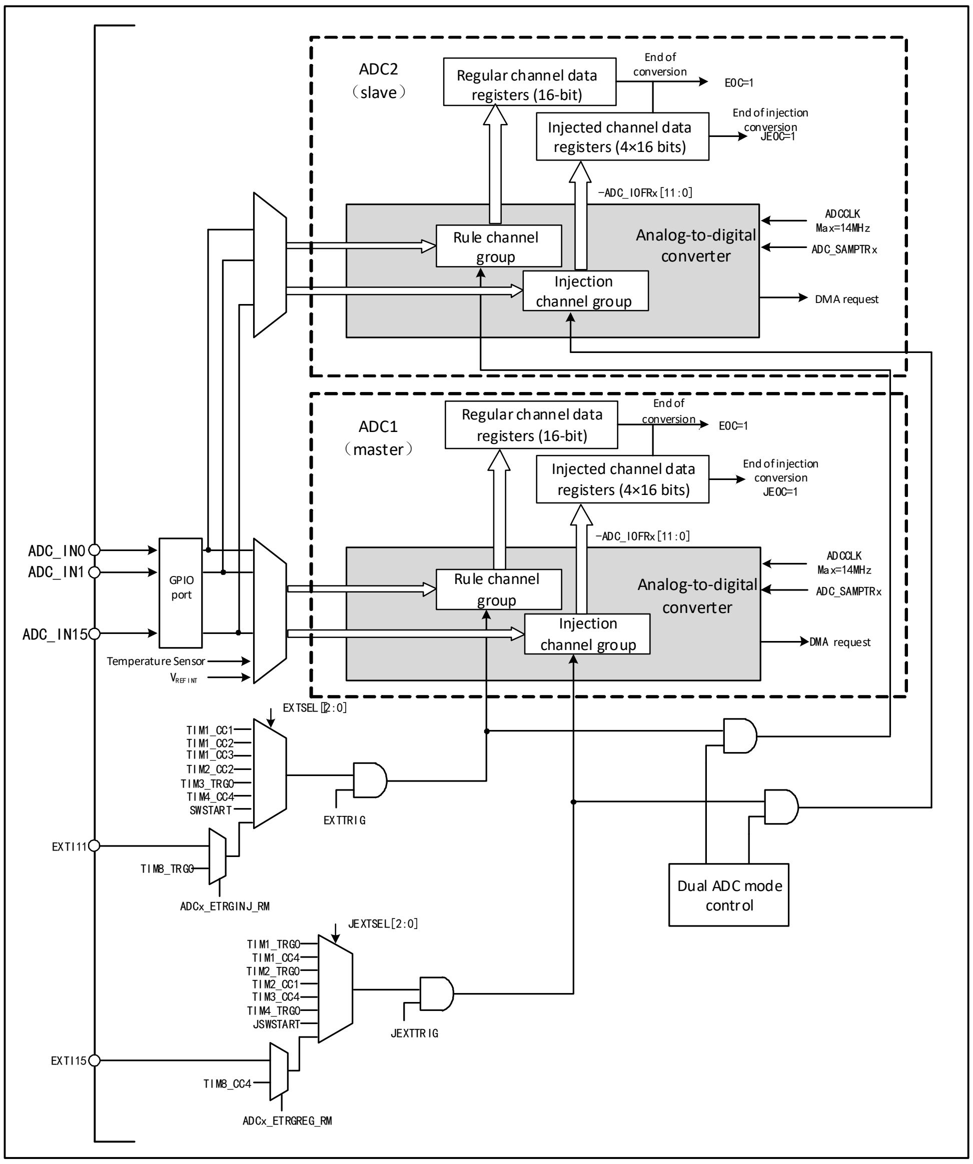
The following modes can be implemented:

- Independent mode
- Injected simultaneous mode
- Regular simultaneous mode
- Fast interleaved mode
- Slow interleaved mode
- Alternate trigger mode
- Regular simultaneous mode + Injected simultaneous mode
- Regular simultaneous mode + Alternate trigger mode
- Injected simultaneous mode + Fast interleaved mode
- Injected simultaneous mode + Slow interleaved mode

Note: 1. In dual ADC mode, the DMA bit must be enabled, to read the slave converted data on the master data register.

2. Only ADC1 has DMA function. the ADC2 converted data only can be transferred by DMA of ADC1 in dual ADC mode.

Figure 12-5 Dual ADC block diagram



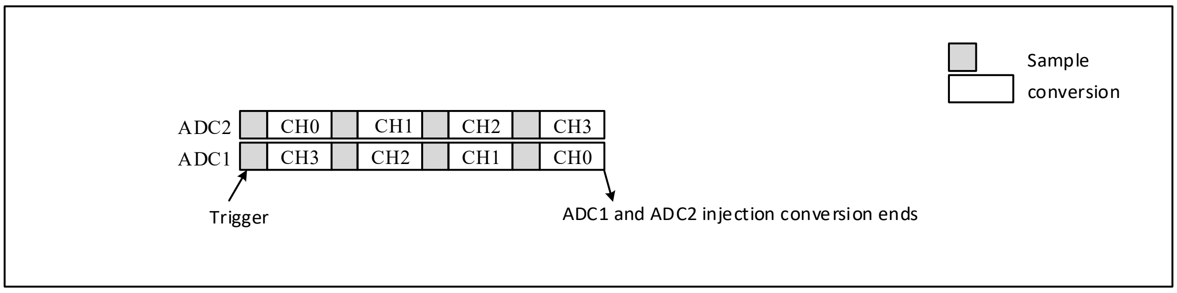
1) Independent mode

In independent mode, the dual ADC modules do not work synchronously, and each ADC works independently.

2) Injected simultaneous mode

This mode converts an injected channel group. By setting the IEXTSEL[2:0] bits in the ADC1_CTLR1 register, the trigger source can be selected. And a simultaneous trigger is provided to ADC2. After conversion, the converted data is stored in the ADC_IDATARx registers of each ADC. If an ADC interrupt is enabled, an IEOC interrupt is generated after conversion.

Figure 12-6 Injected simultaneous mode on 4 channels



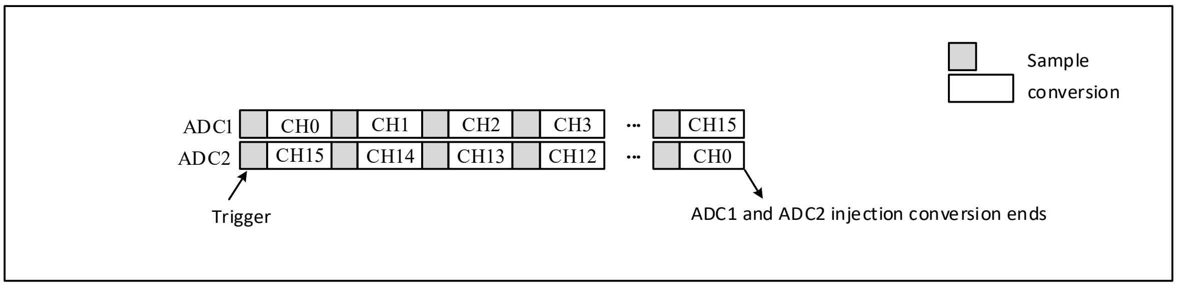
Note: 1. Do not convert the same channel on the 2 ADCs.

2. In simultaneous mode, to ensure that the conversion of the 2 groups can be completed each time, the conversion groups of ADC1 and ADC2 should have the same time, or the time of the longer conversion group is less than the trigger time interval.

3) Regular simultaneous mode

This mode is performed on a regular channel group. By setting the REXTSEL[2:0] bits in the ADC1_CTLR1 register, the trigger source can be selected. And a simultaneous trigger is provided to ADC2. After conversion, a 32-bit DMA transfer request is generated, which transfers the data in the ADC1_RDATAR register to SRAM, and the high 16 bits contain the ADC2 converted data, the low 16 bits contain the ADC1 converted data. If an ADC interrupt is enabled, an EOC interrupt is generated.

Figure 12-7 Regular simultaneous mode on 16 channels



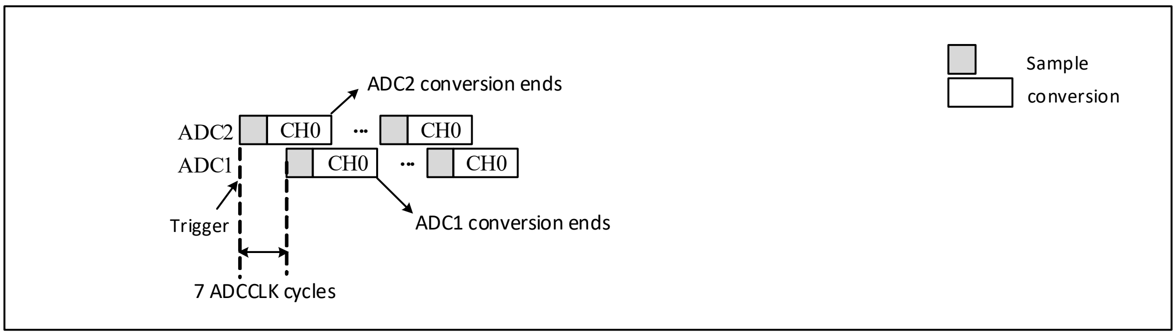
Note: 1. Do not convert the same channel on ADC1 and ADC2;

2. In simultaneous mode, to ensure that the conversion of the 2 groups can be completed each time, the conversion groups of ADC1 and ADC2 should have the same time, or the time of the longer conversion group is less than the trigger time interval.

4) Fast interleaved mode

This mode only applies to a regular channel (usually only one channel). By setting the REXTSEL[2:0] bits in the ADC1_CTLR1 register, the trigger source can be selected. After a trigger occurs, ADC2 starts conversion immediately, while ADC1 starts conversion after a delay of 7 ADC clock cycles. If both ADC1 and ADC2 enable continuous mode (the CONT bit is set), continuous interleaved conversion is performed on regular channel. If an interrupt is enabled, ADC1 generates an EOC interrupt. If both ADC1 and ADC2 enable DMA, a 32-bit DMA transfer request is generated, which transfers the data in the ADC1_RDATAR register to SRAM, and the high 16 bits contain the ADC2 converted data, the low 16 bits contain the ADC1 converted data.

Figure 12-8 Fast interleaved mode on 1 channel in continuous conversion mode

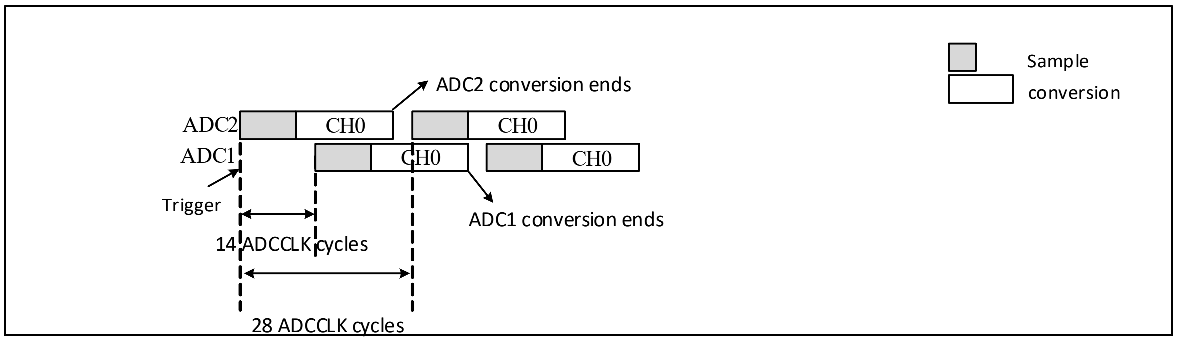


Note: The sampling time should be less than 7 ADC clock cycles, to avoid the overlap between ADC1 and ADC2 sampling phases in the event that they convert the same channel.

5) Slow interleaved mode

This mode only applies to a regular channel (only one channel). By setting the REXTSEL[2:0] bits in the ADC1_CTLR1 register, the trigger source can be selected. After a trigger occurs, ADC2 starts conversion immediately, while ADC1 starts conversion after a delay of 14 ADC clock cycles, and ADC2 starts again after a second delay of 14 ADC clock cycles, and so on. If an interrupt is enabled, ADC1 generates an EOC interrupt. If both ADC1 and ADC2 enable DMA, a 32-bit DMA transfer request is generated, which transfers the data in the ADC1_RDATAR register to SRAM, and the high 16 bits contain the ADC2 converted data, the low 16 bits contain the ADC1 converted data.

Figure 12-9 Slow interleaved mode on 1 channel



Note: 1. The sampling time should be less than 14 ADC clock cycles, to avoid an overlap with the next conversion;

2. After 28 ADC clock cycles, a new ADC2 start is generated automatically;

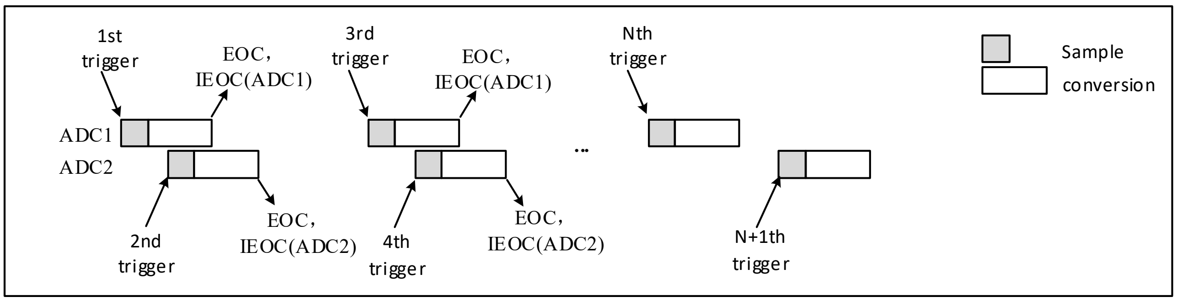
3.The CONT bit cannot be set.

6) Alternate trigger mode

This mode only applies to an injected channel group. By setting the REXTSEL[2:0] bits in the ADC1_CTLR1 register, the trigger source can be selected. When the first trigger occurs, all injected group channels in ADC1 are converted. When the second trigger occurs, all injected group channels in ADC2 are converted. And so on.

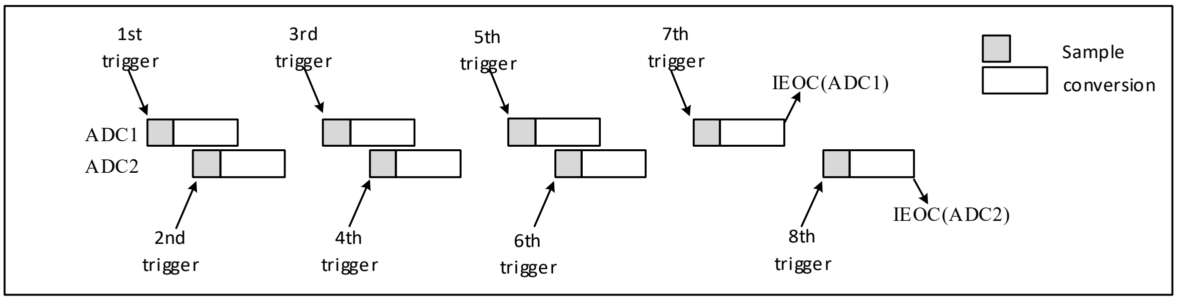
If an interrupt is enabled, an IEOC interrupt is generated after all injected group channels in ADC1 are converted. If an interrupt is enabled, an IEOC interrupt is generated after all injected group channels in ADC2 are converted.

Figure 12-10 Alternate trigger: injected channel group of each ADC



If the injected discontinuous mode is enabled, when the first trigger occurs, the first injected channel in ADC1 is converted. When the second trigger occurs, the first injected channel in ADC2 is converted. And so on. If an interrupt is enabled, an IEOC interrupt is generated after all injected group channels in ADC1 are converted. If an interrupt is enabled, an IEOC interrupt is generated after all injected group channels in ADC2 are converted.

Figure 12-11 Alternate trigger: injected channel group of each ADC in discontinuous mode



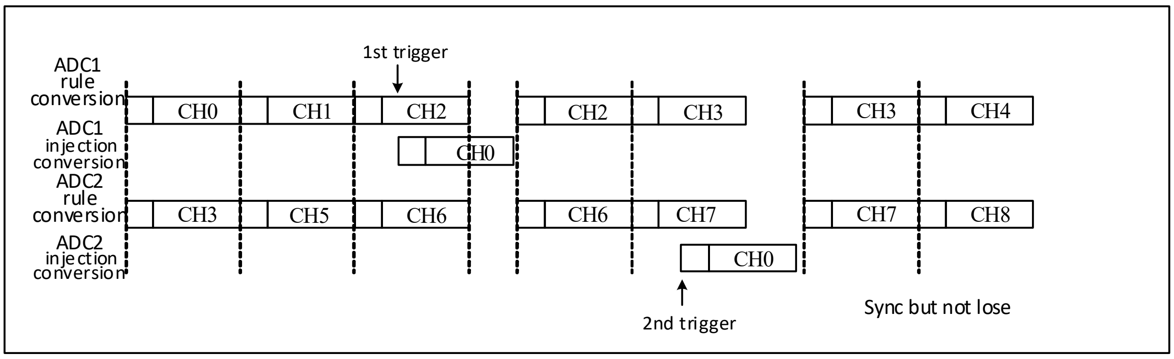
7) Regular simultaneous mode + Injected simultaneous mode

This mode can interrupt simultaneous conversion of a regular group to start simultaneous conversion of an injected group. In this mode, the conversion groups of ADC1 and ADC2 should have the same time, or the time of the longer conversion group is less than the trigger time interval.

8) Regular simultaneous mode + Alternate trigger mode

This mode can interrupt simultaneous conversion of a regular group to start alternate trigger conversion of an injected group. When an injected event occurs, alternate trigger conversion starts immediately. If the regular conversion is already running, the regular conversion of both ADCs is stopped, and it is resumed synchronously at the end of the injected conversion.

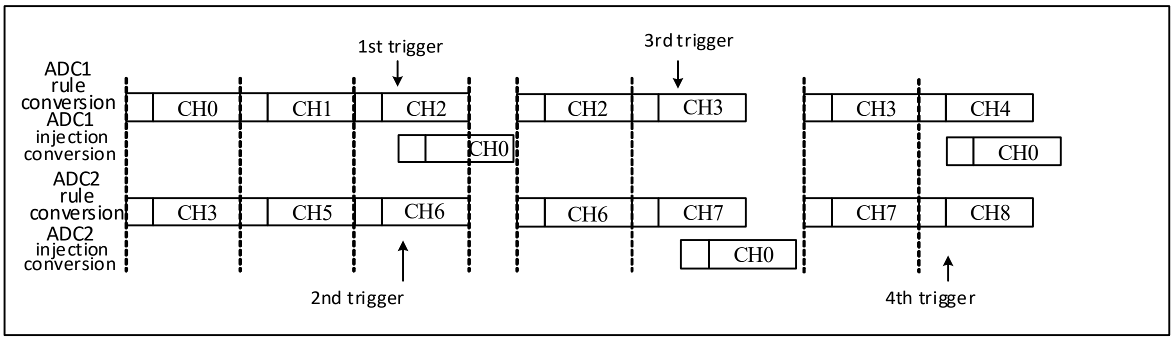
Figure 12-12 Alternate trigger conversion in regular simultaneous mode



Note: In this mode, the conversion groups of ADC1 and ADC2 should have the same time, or the time of the longer conversion group is less than the trigger time interval.

If a trigger occurs during an injected conversion that has interrupted a regular conversion, it will be ignored. The Figure 12-13 shows the behavior in this case (the second trigger is ignored).

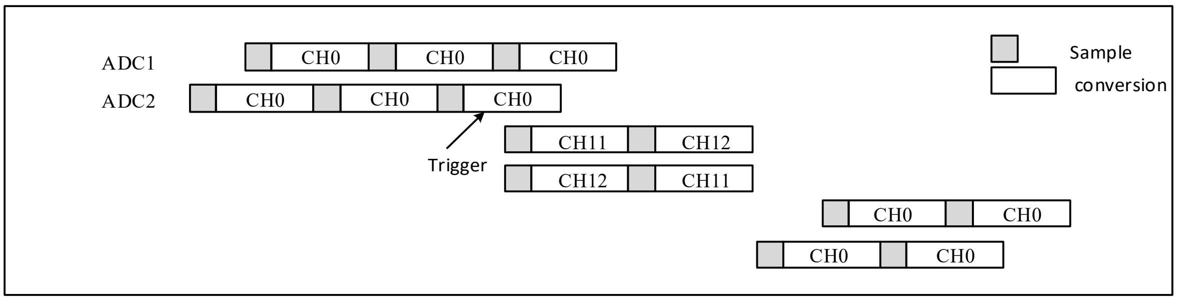
Figure 12-13 Trigger event occurring during injected conversion



9) Injected simultaneous mode + Interleaved mode

In this mode, an injected conversion can stop the interleaved conversion. When an injected event occurs, the interleaved conversion is interrupted, and the injected conversion starts. After injected conversion, the interleaved conversion is resumed.

Figure 12-14 Injected group conversion during interleaved conversion



Note: When the ADC clock prescaler is set to 4, the sampling interval is 8 ADC clock cycles followed by 6 ADC clock cycles, instead of 7 ADC clock cycles followed by 7 ADC clock cycles.

12.3 Register description

Table 12-5 ADC1 registers

Name	Access address	Description	Reset value
R32_ADC1_STATR	0x40012400	ADC1 status register	0x00000000
R32_ADC1_CTLR1	0x40012404	ADC1 control register1	0x00000000
R32_ADC1_CTLR2	0x40012408	ADC1 control register 2	0x00000000
R32_ADC1_SAMPTR1	0x4001240C	ADC1 sample time configuration register1	0x00000000
R32_ADC1_SAMPTR2	0x40012410	ADC1 sample time configuration register2	0x00000000
R32_ADC1_IOFR1	0x40012414	ADC1 injected channel data offset register1	0x00000000
R32_ADC1_IOFR2	0x40012418	ADC1 injected channel data offset register2	0x00000000
R32_ADC1_IOFR3	0x4001241C	ADC1 injected channel data offset register3	0x00000000
R32_ADC1_IOFR4	0x40012420	ADC1 injected channel data offset register4	0x00000000
R32_ADC1_WDHTR	0x40012424	ADC1 watchdog high threshold register	0x00000000
R32_ADC1_WDLTR	0x40012428	ADC1 watchdog low threshold register	0x00000000
R32_ADC1_RSQR1	0x4001242C	ADC1 regular channel sequence register1	0x00000000
R32_ADC1_RSQR2	0x40012430	ADC1 regular channel sequence register2	0x00000000
R32_ADC1_RSQR3	0x40012434	ADC1 regular channel sequence register3	0x00000000
R32_ADC1_ISQR	0x40012438	ADC1 injected channel sequence register	0x00000000
R32_ADC1_IDATAR1	0x4001243C	ADC1 injected data register1	0x00000000
R32_ADC1_IDATAR2	0x40012440	ADC1 injected data register2	0x00000000
R32_ADC1_IDATAR3	0x40012444	ADC1 injected data register3	0x00000000
R32_ADC1_IDATAR4	0x40012448	ADC1 injected data register4	0x00000000
R32_ADC1_RDATAR	0x4001244C	ADC1 regular data register	0x00000000

Table 12-6 ADC2 registers

Name	Access address	Description	Reset value
R32_ADC2_STATR	0x40012800	ADC2 status register	0x00000000
R32_ADC2_CTLR1	0x40012804	ADC2 control register1	0x00000000
R32_ADC2_CTLR2	0x40012808	ADC2 control register 2	0x00000000
R32_ADC2_SAMPTR1	0x4001280C	ADC2 sample time configuration register1	0x00000000
R32_ADC2_SAMPTR2	0x40012810	ADC2 sample time configuration register2	0x00000000
R32_ADC2_IOFR1	0x40012814	ADC2 injected channel data offset register1	0x00000000
R32_ADC2_IOFR2	0x40012818	ADC2 injected channel data offset register2	0x00000000
R32_ADC2_IOFR3	0x4001281C	ADC2 injected channel data offset register3	0x00000000
R32_ADC2_IOFR4	0x40012820	ADC2 injected channel data offset register4	0x00000000
R32_ADC2_WDHTR	0x40012824	ADC2 watchdog high threshold register	0x00000000
R32_ADC2_WDLTR	0x40012828	ADC2 watchdog low threshold register	0x00000000
R32_ADC2_RSQR1	0x4001282C	ADC2 regular channel sequence register1	0x00000000
R32_ADC2_RSQR2	0x40012830	ADC2 regular channel sequence register2	0x00000000
R32_ADC2_RSQR3	0x40012834	ADC2 regular channel sequence register3	0x00000000
R32_ADC2_ISQR	0x40012838	ADC2 injected channel sequence register	0x00000000
R32_ADC2_IDATAR1	0x4001283C	ADC2 injected data register1	0x00000000
R32_ADC2_IDATAR2	0x40012840	ADC2 injected data register2	0x00000000

R32_ADC2_IDATAR3	0x40012844	ADC2 injected data register3	0x00000000
R32_ADC2_IDATAR4	0x40012848	ADC2 injected data register4	0x00000000
R32_ADC2_RDATAR	0x4001284C	ADC2 regular data register	0x00000000

12.3.1 ADCx Status Register (ADCx_STATR) (x=1/2)

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											STRT	JSTRT	JEOC	EOC	AWD

Bit	Name	Access	Description	Reset value
[31:5]	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: 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 ADC_RDATAR is read.	
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 ADC_WDHTR and ADC_WDLTR register range), and is cleared by software (invalid if writing 1).	

12.3.2 ADCx Control Register1 (ADCx_CTLR1) (x=1/2)

Offset address: 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			PGA[1:0]		BUFEN	TKITUNE	TKENABLE	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
[31:29]	Reserved	RO	Reserved	0

[28:27]	PGA[1:0]	RW	ADC channel gain configuration:
			00: x1;
			01: x4;
			10: x16;
			11: x64.
			Note: The input gain can be configured, for amplifying small signals and sampling. ADC_Buffer needs to be enabled to use this function.
				0

26	BUFEN	RW	ADC BUFFER enable: 0: Buffer input disabled; 1: Buffer input enabled. *Note: Cases where buffer needs to be turned on:* *1. When TKENABLE bit or TSVREFE position 1, buffer is on by default and cannot be turned off (so in these two cases, ADC calibration needs to be before TKENABLE bit or TSVREFE position 1 and buffer needs to be turned off).* *2. When the external input impedance is greater than the maximum input impedance requirement, the buffer can be turned on to improve the ADC acquisition data (at this time, the ADC sampling time is not recommended to be less than 7.5T), the external input impedance is detailed in CH32F203DS0/CH32F208DS0/CH32V203DS0/CH32V208DS0 data sheet Table 4-28 CH32V307DS0/CH32F207DS0 datasheet Table 4-41. Refer to EVT related routines for specific operation.*	0
25	TKITUNE	RW	TKEY module charging current configuration: 0: Charging current is 35uA; 1: Charging current is reduced half.	0
24	TKENABLE	RW	TKEY module enable control, including TKEY_F and TKEY_V units: 1: Enable TKEY module; 0: Disable TKEY module.	0
23	AWDEN	RW	Analog watchdog enable bit on regular channels:	0

			1: Enable analog watchdog on regular channels; 0: Disable analog watchdog on regular channels;	
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: Regular simultaneous mode + Injected simultaneous mode;
			0010: Regular simultaneous mode + Alternate trigger mode;
			0011: Injected simultaneous mode + Fast interleaved mode;
			0100: Injected simultaneous mode + Slow interleaved mode;
			0101: Injected simultaneous mode
			0110: Regular simultaneous mode
			0111: Fast interleaved mode;
			1000: Slow interleaved mode;
			1001: Alternate trigger mode.
			Note: These bits in ADC2 are reserved. Modification of a configuration bit must be performed when dual mode is disabled.
				0

[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.	0
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 bit on the regular channel: 1: Enable discontinuous mode on the regular channel; 0: Disable discontinuous mode on the regular channel.	0
10	JAUTO	RW	After regular channel is enabled, automatically injected channel group conversion enable bit: 1: Enable automatic injected channel group conversion; 0: Disable automatic injected channel group conversion; *Note: The external trigger function of the injected channel needs to be disabled in this mode.*	0
9	AWDSGL	RW	In scan mode, analog watchdog enable bit on a single channel: 1: Enable analog watchdog on single channel (AWDCH[4:0] selection); 0: Disable analog watchdog on all channels.	0
8	SCAN	RW	Scan mode enable bit: 1: Enable scan mode (continuous conversion of all channels selected by ADC_IOFRx and ADC_RSQRx);	0

			0: Disable scan mode.	
7	JEOCIE	RW	Injected channel group conversion completion interrupt enable bit: 1: Enable injected channel group transfer completion interrupt (IEOC 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. *Note: In scan mode, if this interrupt occurs, the scan will be aborted.*	0
5	EOCIE	RW	Conversion completion (regular 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 channel0; 00001: Analog input channel1; … 10001: Analog input channel17.	0

12.3.3 ADCx Control Register2 (ADCx_CTLR2) (x=1/2)

Offset address: 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								TSVREFE	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				RSTCAL	CAL	CONT	ADON

Bit	Name	Access	Description	Reset value
[31:24]	Reserved	RO	Reserved.	0
23	TSVREFE	RW	Temperature sensor and internal voltage (V_{REFINT}) channel enable: 1: Enable the temperature sensor and V_{REFINT} channel; 0: Disable the temperature sensor and V_{REFINT} channel. *Note: This bit is only applied for ADC1.*	0
22	SWSTART	RW	Start conversion of a regular channel, set by software to start: 1: Start conversion of a regular channel; 0: Reset status. This bit is set by software, and cleared by hardware after the conversion starts.	0

21	JSWSTART	RW	Start conversion of an injected channel, set by software to start: 1: Start conversion of an injected channels; 0: Reset status. This bit is set by software, and cleared by hardware or software 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 select 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;
			110: EXTI line11/ TRGO event of timer8;
			111: SWSTART software trigger.
			Note: Only high-density devices have TRGO event in timer8.
				0

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/CC4 event of timer8;
			111: JSWSTART software trigger.
			Note: Only high-density devices have CC4 event in timer8.
				0

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
[7:4]	Reserved	RO	Reserved.	0
3	RSTCAL	RW	Reset calibration, this bit is set by software, and cleared by hardware after reset:	0

			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.*	
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: Single conversion mode. If this bit is set, the conversion will continue until the bit is cleared.	0
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 1, writing 1 will start the conversion. 1: Enable ADC and to start conversion; 0: Disable ADC conversion/calibration, and go to power down mode. *Note: When only ADON changes in the register, a conversion will be started. If any other bits are sent to change, a new conversion will not be started.*	0

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

Offset address: 0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								SMP17[2:0]			SMP16[2:0]			SMP15[2:1]	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMP15[0]	SMP14[2:0]			SMP13[2:0]			SMP12[2:0]			SMP11[2:0]			SMP10[2:0]		

Bit	Name	Access	Description	Reset value
[31:24]	Reserved	RO	Reserved.	0

[23:0]	SMPx[2:0]	RW	SMPx[2:0]: Sample time configuration of channel x:
			000: 1.5 cycles; 001: 7.5 cycles;
			010: 13.5 cycles; 011: 28.5 cycles;
			100: 41.5 cycles; 101: 55.5 cycles;
			110: 71.5 cycles; 111: 239.5 cycles;
			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.	

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

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: 1.5 cycles; 001: 7.5 cycles;
			010: 13.5 cycles; 011: 28.5 cycles;
			100: 41.5 cycles; 101: 55.5 cycles;
			110: 71.5 cycles; 111: 239.5 cycles;
			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.
				0

12.3.6 ADCy Injected Channel Data Offset Register (ADCy_IOFRx) (y=1/2; 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[11:0]											

Bit	Name	Access	Description	Reset value
[31:12]	Reserved	RO	Reserved.	0
[11:0]	JOFFSETx[11:0]	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 ADC_IDATARx register	0

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

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[11:0]											

Bit	Name	Access	Description	Reset value
[31:12]	Reserved	RO	Reserved.	0
[11:0]	HT[11:0]	RW	Analog watchdog high 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.

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

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[11:0]											

Bit	Name	Access	Description	Reset value
[31:12]	Reserved	RO	Reserved.	0
[11:0]	LT[11:0]	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 Regular Channel Sequence Register1 (ADCx_RSQR1) (x=1/2)

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 quantity of channels to be converted in the regular channel conversion sequence: 0000-1111: 1 to 16 converted channels.	0
[19:15]	SQ16[4:0]	RW	No. (0 to 17) of the 16th converted channel in regular sequence.	0
[14:10]	SQ15[4:0]	RW	No. (0 to 17) of the 15th converted channel in regular sequence.	0
[9:5]	SQ14[4:0]	RW	No. (0 to 17) of the 14th converted channel in regular sequence.	0
[4:0]	SQ13[4:0]	RW	No. (0 to 17) of the 13th converted channel in regular sequence.	0

12.3.10 ADCx Regular Channel Sequence Register2 (ADCx_RSQR2) (x=1/2)

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	No. (0 to 17) of the 12th converted channel in regular sequence.	0
[24:20]	SQ11[4:0]	RW	No. (0 to 17) of the 11th converted channel in regular sequence.	0
[19:15]	SQ10[4:0]	RW	No. (0 to 17) of the 10th converted channel in regular sequence.	0
[14:10]	SQ9[4:0]	RW	No. (0 to 17) of the 9th converted channel in regular sequence.	0
[9:5]	SQ8[4:0]	RW	No. (0 to 17) of the 8th converted channel in regular sequence.	0
[4:0]	SQ7[4:0]	RW	No. (0 to 17) of the 7th converted channel in regular sequence.	0

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

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	No. (0 to 17) of the 6th converted channel in regular sequence.	0
[24:20]	SQ5[4:0]	RW	No. (0 to 17) of the 5th converted channel in regular sequence.	0
[19:15]	SQ4[4:0]	RW	No. (0 to 17) of the 4th converted channel in regular sequence.	0
[14:10]	SQ3[4:0]	RW	No. (0 to 17) of the 3th converted channel in regular sequence.	0
[9:5]	SQ2[4:0]	RW	No. (0 to 17) of the 2nd converted channel in regular sequence.	0
[4:0]	SQ1[4:0]	RW	No. (0 to 17) of the 1st converted channel in regular sequence.	0

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

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	The quantity of channels to be converted in the injected channel conversion sequence:	0

			00-11: 1 to 4 converted channels.	
[19:15]	JSQ4[4:0]	RW	No. (0 to 17) of the 4th converted channel in injected sequence.	0
[14:10]	JSQ3[4:0]	RW	No. (0 to 17) of the 3rd converted channel in injected sequence.	0
[9:5]	JSQ2[4:0]	RW	No. (0 to 17) of the 2nd converted channel in injected sequence.	0
[4:0]	JSQ1[4:0]	RW	No. (0 to 17) of the 1st converted channel in injected sequence.	0

Note: Different from regular conversion sequence, if the length of ILEN[1:0] is less than 4, the sequence of conversion will start from (4-ILEN).

12.3.13 ADCy Injected Data Register (ADCy_IDATARx) (y=1/2; 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 right alignment).	0

12.3.14 ADCx Regular Data Register (ADCx_RDATAR) (x=1/2)

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[4:0]															

Bit	Name	Access	Description	Reset value
[31:16]	ADC2DATA[15:0]	RO	ADC2 converted data: ADC1: These bits store the regular channel data converted by ADC2 in dual mode. ADC2: These bits are not used.	0
[15:0]	DATA[4:0]	RO	Regular channel converted data (data left or right alignment).	0

Chapter 13 Touch Key Detection (TKEY)

This chapter applies to the whole family of CH32F2x, CH32V2x and CH32V3x.

The Touch Key Detection control (TKEY) unit implements the touch-key detection function by converting the capacitance to the voltage for sampling, with the help of the voltage conversion function of the ADC module. The detection channel multiplexes the 16 external channels of the ADC. It implements the touch-key detection through the single conversion mode of ADC.

13.1 Functional description of TKEY

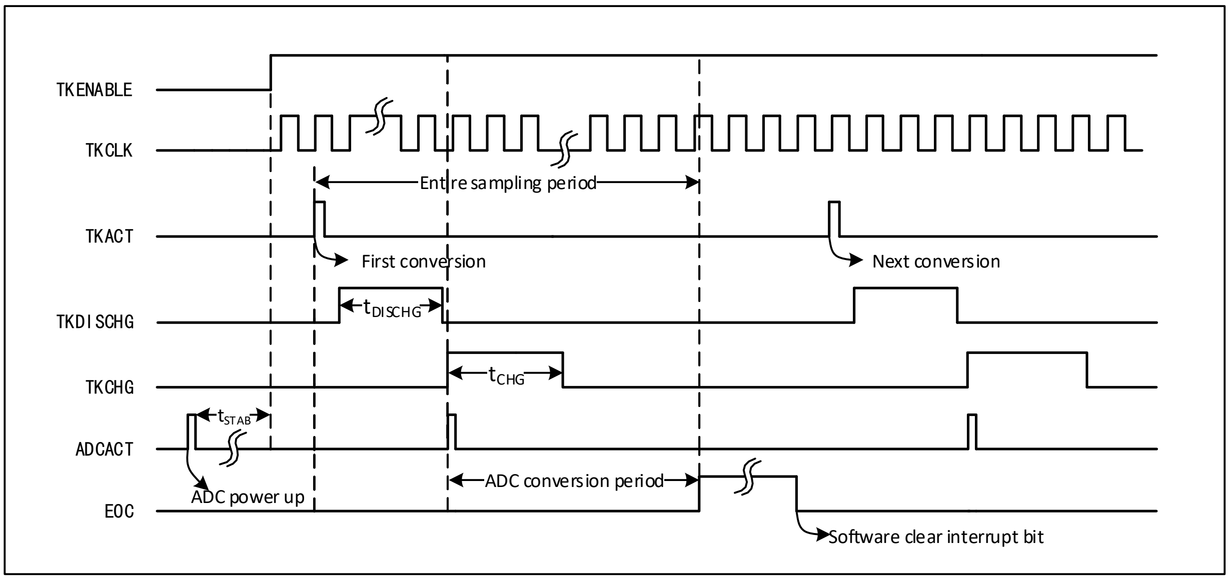
- Enable TKEY

For TKEY detection, the ADC module is needed. To enable TKEY, ensure that ADC is powered on (ADON=1), then set the TKENABLE bit in the ADC_CTLR1 register to 1. The charge current of the TKEY module can be adjusted by the TKITUNE bit.

TKEY only supports single 1-channel conversion mode, which configures the channel to be converted as the first channel in the regular sequence of ADC. And software starts conversion (write to TKEY_ACT_DCG).

Note: When the TKEY conversion is disabled, ADC channel configuration function can still be retained.

Figure 13-1 TKEY working timing diagram



- Programmable sampling time

For TKEY unit conversion, multiple ADCCLK clock cycles (t_{DISCHG}) need to be used for discharge, then charge and sample voltage on the channel through multiple ADCCLK cycles (t_{CHG}). The charge cycle is the total of the configuration values of the TKCGx[2:0] bits in the TKEY_CHARGE1register and the TKEY_CHARGE2 register and the offset value of TKEY_CHGOFFSET. For each channel, different charge cycles can be used to adjust sampling voltage.

13.2 TKEY operations

TKEY detection is an extended function of ADC module. Its working principle is to change the capacitance sensed by the hardware channel through "touch" and "non-touch" methods, and then to convert the capacitance change into the voltage change and finally convert into a digital value by the ADC module.

During sample, ADC needs to be configured as a single 1-channel working mode, and a conversion is started by the "write operation" of the TKEY _ACT register. The specific process is as follows:

1) Initialize the ADC function, configure the ADC module as a single conversion module, set the ACON bit to 1, and wake up the ADC module. Set the TKENABLE bit in the ADC_CTLR1 register to 1, and switch on the TKEY unit.
2) Set the channel to be converted, write the channel serial number into the first conversion position in the ADC regular group sequence (ADC_RSQR3[4:0]), and set RLEN[3:0] to 1.
3) Set the charge sample time of the channel, write to the TKEY_CHARGEx register, to configure different charge times for each channel.
4) Write to TKEY_CHGOFFSET, set the offset charge time of the channel (low 8 bits active), to adjust the charge time.
5) Write to TKEY_ACT_DCG, set the discharge time (low 8 bits active), to start a TKEY sample and conversion.
6) Wait for the EOC conversion end flag bit in the ADC status register to be set to 1, read the ADC_DR register to obtain the conversion value.
7) To perform next conversion, repeat steps 2-6. If you do not need to modify the channel discharge time or charge sampling time, you can skip step 3 or 4.

13.3 TKEY register description

Table 13-1 TKEY1 registers

Name	Access address	Description	Reset value
R32_TKEY1_CHARGE1	0x4001240C	TKEY charge sample time register1	0x00000000
R32_TKEY1_CHARGE2	0x40012410	TKEY charge sample time register2	0x00000000
R32_TKEY1_CHGOFFSET	0x4001243C	TKEY charge time offset register	0x00000000
R32_TKEY1_ACT_DCG	0x4001244C	TKEY activate and discharge time register	X
R32_TKEY1_DR	0x4001244C	TKEY data register	X

Table 13-2 TKEY2 registers

Name	Access address	Description	Reset value
R32_TKEY2_CHARGE1	0x4001280C	TKEY charge sample time register1	0x00000000
R32_TKEY2_CHARGE2	0x40012810	TKEY charge sample time register2	0x00000000
R32_TKEY2_CHGOFFSET	0x4001283C	TKEY charge time offset register	0x00000000
R32_TKEY2_ACT_DCG	0x4001284C	TKEY activate and discharge time register	X
R32_TKEY2_DR	0x4001284C	TKEY data register	X

13.3.1 TKEYx Charge Sample Time Register 1 (TKEYx_CHARGE1) (x=1/2)

Offset address: 0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								TKCG17[2:0]			TKCG16[2:0]			TKCG15[2:1]	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TKCG15	TKCG14[2:0]			TKCG13[2:0]			TKCG12[2:0]			TKCG11[2:0]			TKCG10[2:0]		

Bit	Name	Access	Description	Reset value
[31:24]	Reserved	RO	Reserved	0
[23:0]	TKCGx[2:0]	RW	TKCGx[2:0] (x=10-17): Select the charge sample time of channelx These bits are used to independently select the charge time for each channel. 000: 1.5 cycles 100: 41.5 cycles 001: 7.5 cycles 101: 55.5 cycles 010: 13.5 cycles 110: 71.5 cycles 011: 28.5 cycles 111: 239.5 cycles Time base: ADC clock.	0

Note: This register maps the sample time register1 (ADC_SAMPTR1) of the ADC module. When the ADC function is configured, it is the channel sample time. When the TKEY_F function is configured, it is the channel charge time.

13.3.2 TKEYx Charge Sample Time Register 2 (TKEYx_CHARGE2) (x=1/2)

Offset address: 0x10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		TKCG9[2:0]			TKCG8[2:0]			TKCG7[2:0]			TKCG6[2:0]			TKCG5[2:1]	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TKCG5	TKCG4[2:0]			TKCG3[2:0]			TKCG2[2:0]			TKCG1[2:0]			TKCG0[2:0]		

Bit	Name	Access	Description	Reset value
[31:30]	Reserved	RO	Reserved	0
[29:0]	TKCGx[2:0]	RW	TKCGx[2:0] (x=0-9): Select the charge sample time of channel x These bits are used to independently select the charge time for each channel. 000: 1.5 cycles 100: 41.5 cycles 001: 7.5 cycles 101: 55.5 cycles 010: 13.5 cycles 110: 71.5 cycles 011: 28.5 cycles 111: 239.5 cycles Time base: ADC clock.	0

Note: This register maps the sample time register1 (ADC_SAMPTR2) of the ADC module. When the ADC function is configured, it is the channel sample time. When the TKEY_F function is configured, it is the channel charge time.

13.3.3 TKEYx Charge Time Offset Register (TKEYx_CHGOFFSET) (x=1/2)

Offset address: 0x3C

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

Bit	Name	Access	Description	Reset value
[31:8]	Reserved	RO	Reserved	0
[7:0]	TKCGOFFSET[7:0]	WO	TKEY charge time offset configuration value Total charge time: TCHG=TKCGOFFSET+ TKCGx	0

Note: This register maps the injected data register1 (ADC_IDATAR1) of the ADC module. So when write operation is performed on this address register, it serves as TKEY charge time offset (TKEY_ CHGOFFSET). When read operation is performed, it serves as the injected data register1 (ADC_IDATAR1) of the ADC module.

13.3.4 TKEYx Activate and Discharge Time Register (TKEYx_ACT_DCG) (x=1/2)

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**
Reserved								TKACT_DCG[7:0]							

Bit	Name	Access	Description	Reset value
[31:8]	Reserved	RO	Reserved	0
[7:0]	TKACT_DCG[7:0]	WO	Write the discharge time and activate a TKEY channel detection.	0

Note: This register maps the regular data register (ADC_RDATAR) of the ADC module.

13.3.5 TKEYx Data Register (TKEYx_DR) (x=1/2)

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**
DATA[15:0]															

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:0]	DATA[15:0]	RO	Converted data.	0

Note: This register maps the regular data register (ADC_RDATAR) of the ADC module.

Chapter 14 Advanced-control Timer (ADTM)

This chapter applies to the whole family of CH32F2x, CH32V2x and CH32V3x.

The advanced-control timer module contains a powerful 16-bit auto-reload timer (TIM1, TIM8, TIM9 and TIM10), which can be used to measure pulse width or generate pulse and PWM wave, etc. It is used for fields of motor control and power, etc.

14.1 Main features

Advanced-control timer (TIM1/8/9/10) 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;
- Four 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 zone 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

14.2 Principle and structure

This section describes the internal structure of the advanced-control timer.

14.2.1 Overview

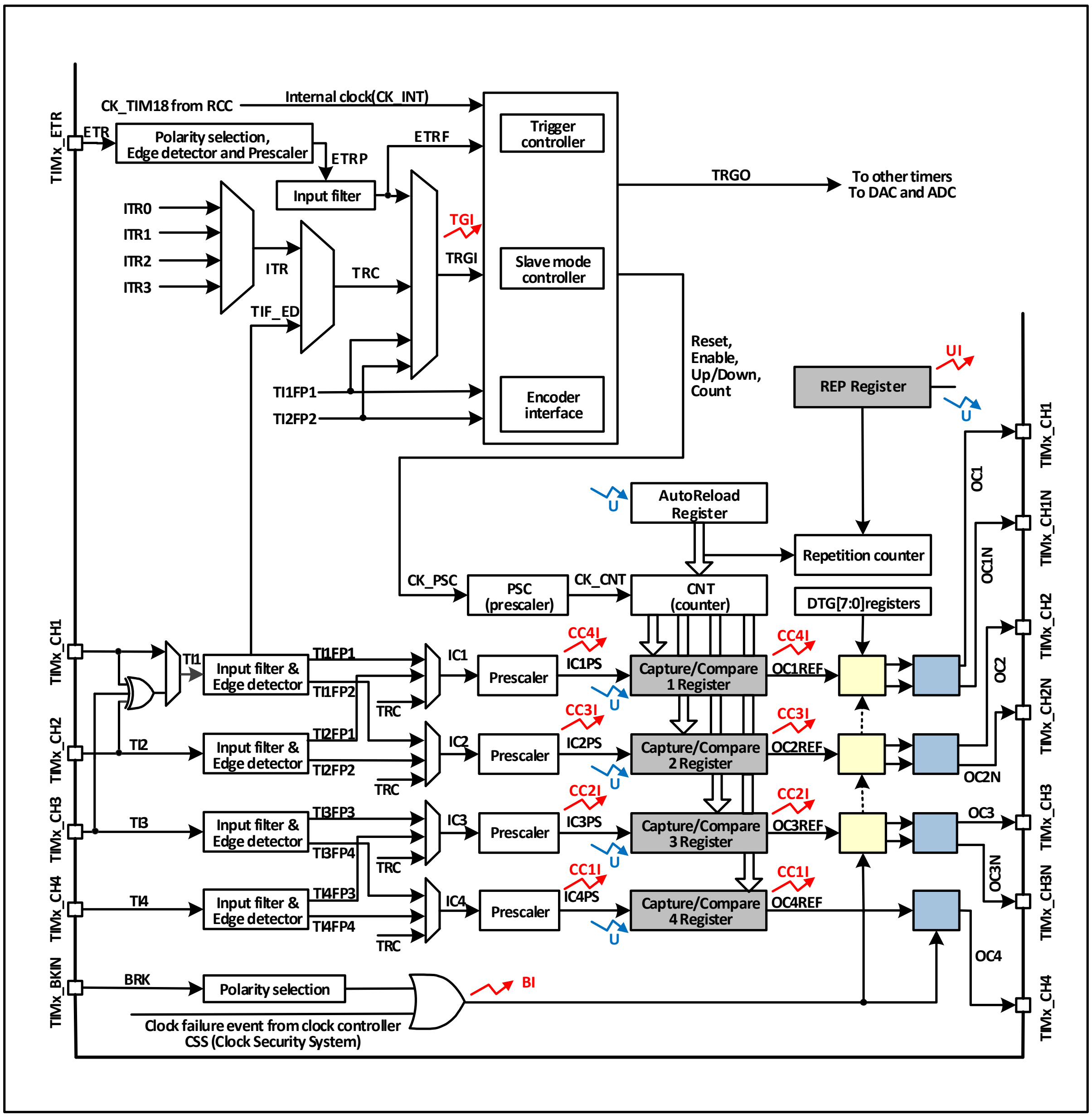
As shown in Figure 14-1, the structure of the advanced-control timer can be roughly divided into 3 parts: Input clock part, core counterpart and compare/capture channel part.

The advanced-control timer clock can come from APB bus clock (CK_INT), external clock input pin (TIMx_ETR), other timers with clock output function (ITRx), or the input end of compare capture channel (TIMx_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 peripherals such as timer, ADC and DAC.

The core of the advanced-control timer is a 16-bit counter (CNT). After CK_PSC is divided by the prescaler (PSC), it becomes CK_CNT and output to CNT. CNT supports up-counting mode, down-counting mode and up/down counting mode, and there is an automatic reload value register (ATRLR). After each counting cycle is completed, CNT will be reloaded with the initial value. In addition, there is an auxiliary counter that counts 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.

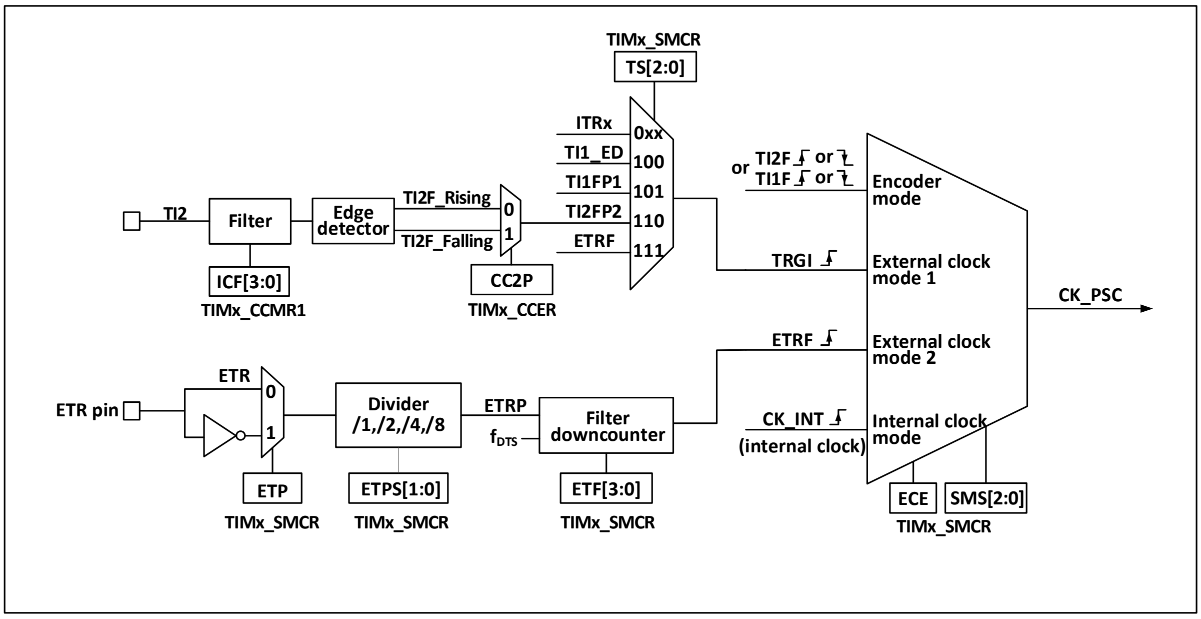
The advanced-control timer has 4 groups of compare/capture channels. On each group of compare/capture channel, pulses can be inputted from its dedicated pins or output waveforms to the pins, i.e., the compare/capture channels support input and output modes. The input of each channel of the compare/capture 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 14-1 Structure block diagram of advanced-control timer



14.2.2 Clock input

Figure 14-2 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 APB clock input route: CK_INT;
3) The route from the compare/capture channel pin (TIMx_CHx): TIMx_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.

14.2.2.1 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.

14.2.2.2 External clock source mode1

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.

14.2.2.3 External clock source mode2

Use external trigger mode2 to count on every rising or falling edge of the external clock pin input. When the ECE bit is set, the external clock source mode2 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.

14.2.2.4 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 14.3.9 for specific functions.

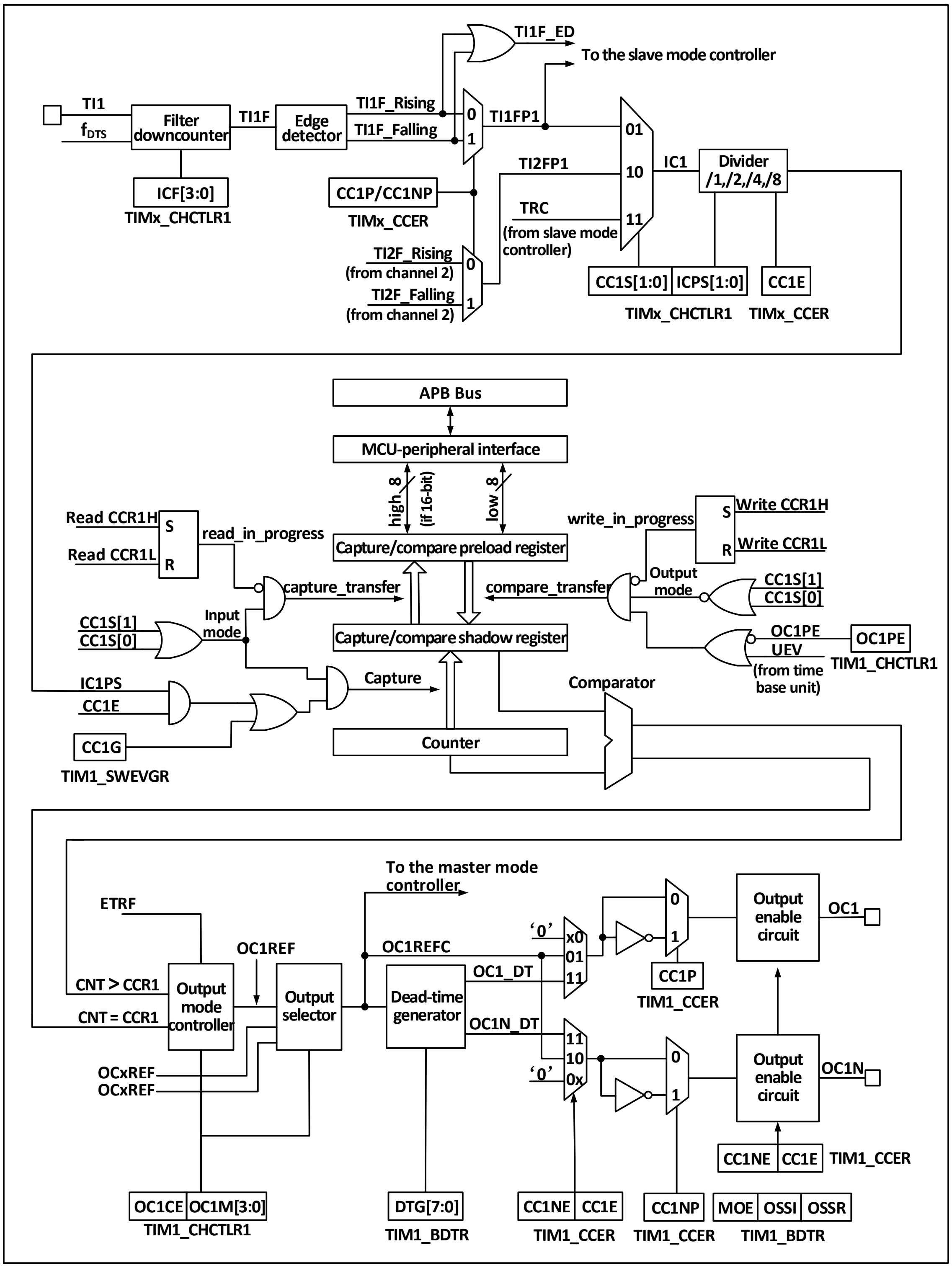
14.2.3 Counter and periphery

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_TIMx_PSC+1. CK_PSC will become CK_INT 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.

14.2.4 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 14-3 Structure block diagram of compare/capture channel



The block diagram of the compare/capture channel is as shown in Figure 14-3. 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 14-1). 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).

14.3 Function and implementation

The advanced-control timer complex functions are implemented by the operation of comparison &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.

14.3.1 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_TIMx_CHCTLRx). When a capture event occurs, CCxIF (in R16_TIMx_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 CCxS and select the source of ICx signal. For example, it is set to 10b, and TI1FP1 is selected as the source of IC1, and the default setting cannot be used. CCxS defaults to use the compare capture module as the output channel;
2) Configure ICxF and set the digital filter of the TI signal. The digital filter will output a jump based on the determined frequency and determined sampling times. The sampling frequency and times are determined by ICxF;
3) Configure CCxP bit and set the polarity of TIxFPx. For example, maintain CC1P bit to be low and select the jump of rising edge;
4) Configure ICxPS and set ICx signal as the frequency division factor between ICxPS. For example, maintain the ICxPS as 00b without frequency division;
5) Configure the CCxE bit to allow to capture the core counter (CNT) value to the compare/capture register. Set the CC1E bit;
6) Configure the CCxIE and CCxDE 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 (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 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_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 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.

14.3.3 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.

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/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.

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 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

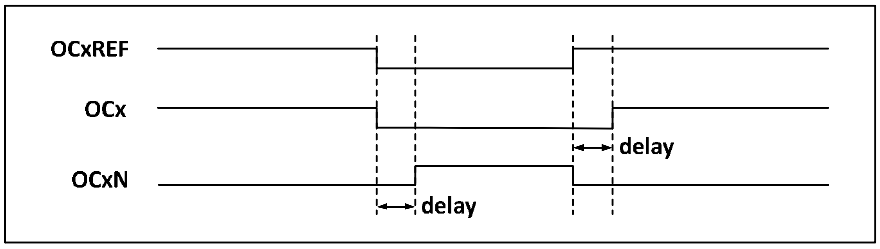
When the center-aligned mode is used, the core counter will run in a mode where up counting and down counting are performed alternately, and OCxREF performs rising and falling jumps when the values of the core counter and the compare/capture register are consistent. However, in 3 types of central alignment mode 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.

14.3.6 Complementary output and dead zone

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 zone 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 zone, each channel has a 10-bit dead zone 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 14-4 shows the relationship between OCx, OCxN and OCxREF, and shows the dead zone.

Figure 14-4 Complementary output and dead zone



14.3.7 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 APB cycle to read the written value correctly.

When the selected level appears on the break pin, the system will generate the following actions:

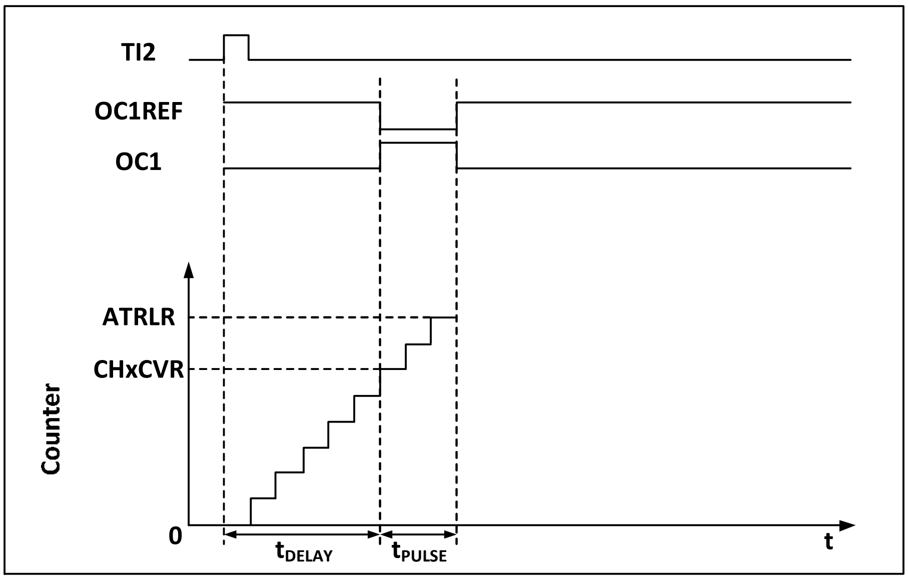
1) The MOE bit is asynchronously cleared, and the output is set to the invalid status, idle status or reset status according to the setting of the SOOI bit;
2) After MOE is cleared, each output channel will output the level determined by OISx;
3) During the supplementary output: the output will be in 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.

14.3.8 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 14-5, 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 14-5 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.

14.3.9 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 14-1 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

14.3.10 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 the timers.

Table 14-2 TIMx internal trigger connection

Slave timer	ITR0(TS=000)	ITR1(TS=001)	ITR2(TS=010)	ITR3(TS=011)
TIM1	TIM5	TIM2	TIM3	TIM4
TIM8	TIM1	TIM2	TIM4	TIM5
TIM9	TIM10	TIM5	TIM6	TIM7
TIM10	TIM9	TIM2	TIM4	TIM5

14.3.11 Debug mode

When the system enters the debug mode, the timer continues to run or stop according to the setting of the DBG module.

14.4 Register description

Table 14-3 TIM1 registers

Name	Access address	Description	Reset value
R16_TIM1_CTLR1	0x40012C00	Control register1	0x0000
R16_TIM1_CTLR2	0x40012C04	Control register2	0x0000
R16_TIM1_SMCFGR	0x40012C08	Slave mode configuration register	0x0000
R16_TIM1_DMAINTENR	0x40012C0C	DMA/interrupt enable register	0x0000
R16_TIM1_INTFR	0x40012C10	Interrupt flag register	0x0000
R16_TIM1_SWEVGR	0x40012C14	Event generation register	0x0000
R16_TIM1_CHCTLR1	0x40012C18	Compare/capture control register1	0x0000
R16_TIM1_CHCTLR2	0x40012C1C	Compare/capture control register2	0x0000
R16_TIM1_CCER	0x40012C20	Compare/capture enable register	0x0000
R16_TIM1_CNT	0x40012C24	Counter	0x0000
R16_TIM1_PSC	0x40012C28	Prescaler	0x0000
R16_TIM1_ATRLR	0x40012C2C	Auto-reload register	0x0000
R16_TIM1_RPTCR	0x40012C30	Repeat count register	0x0000

R16_TIM1_CH1CVR	0x40012C34	Compare/capture register1	0x0000
R16_TIM1_CH2CVR	0x40012C38	Compare/capture register2	0x0000
R16_TIM1_CH3CVR	0x40012C3C	Compare/capture register3	0x0000
R16_TIM1_CH4CVR	0x40012C40	Compare/capture register4	0x0000
R16_TIM1_BDTR	0x40012C44	Break and dead zone register	0x0000
R16_TIM1_DMACFGR	0x40012C48	DMA configuration register	0x0000
R16_TIM1_DMAADR	0x40012C4C	DMA address register in continuous mode	0x0000

Table 14-4 TIM8 registers

Name	Access address	Description	Reset value
R16_TIM8_CTLR1	0x40013400	Control register1	0x0000
R16_TIM8_CTLR2	0x40013404	Control register2	0x0000
R16_TIM8_SMCFGR	0x40013408	Slave mode configuration register	0x0000
R16_TIM8_DMAINTENR	0x4001340C	DMA/interrupt enable register	0x0000
R16_TIM8_INTFR	0x40013410	Interrupt flag register	0x0000
R16_TIM8_SWEVGR	0x40013414	Event generation register	0x0000
R16_TIM8_CHCTLR1	0x40013418	Compare/capture control register1	0x0000
R16_TIM8_CHCTLR2	0x4001341C	Compare/capture control register2	0x0000
R16_TIM8_CCER	0x40013420	Compare/capture enable register	0x0000
R16_TIM8_CNT	0x40013424	Counter	0x0000
R16_TIM8_PSC	0x40013428	Prescaler	0x0000
R16_TIM8_ATRLR	0x4001342C	Auto-reload register	0x0000
R16_TIM8_RPTCR	0x40013430	Repeat count register	0x0000
R16_TIM8_CH1CVR	0x40013434	Compare/capture register1	0x0000
R16_TIM8_CH2CVR	0x40013438	Compare/capture register2	0x0000
R16_TIM8_CH3CVR	0x4001343C	Compare/capture register3	0x0000
R16_TIM8_CH4CVR	0x40013440	Compare/capture register4	0x0000
R16_TIM8_BDTR	0x40013444	Break and dead zone register	0x0000
R16_TIM8_DMACFGR	0x40013448	DMA configuration register	0x0000
R16_TIM8_DMAADR	0x4001344C	DMA address register in continuous mode	0x0000

Table 14-5 TIM9 registers

Name	Access address	Description	Reset value
R16_TIM9_CTLR1	0x40014C00	Control register1	0x0000
R16_TIM9_CTLR2	0x40014C04	Control register2	0x0000
R16_TIM9_SMCFGR	0x40014C08	Slave mode configuration register	0x0000
R16_TIM9_DMAINTENR	0x40014C0C	DMA/interrupt enable register	0x0000
R16_TIM9_INTFR	0x40014C10	Interrupt flag register	0x0000
R16_TIM9_SWEVGR	0x40014C14	Event generation register	0x0000
R16_TIM9_CHCTLR1	0x40014C18	Compare/capture control register1	0x0000
R16_TIM9_CHCTLR2	0x40014C1C	Compare/capture control register2	0x0000
R16_TIM9_CCER	0x40014C20	Compare/capture enable register	0x0000
R16_TIM9_CNT	0x40014C24	Counter	0x0000

R16_TIM9_PSC	0x40014C28	Prescaler	0x0000
R16_TIM9_ATRLR	0x40014C2C	Auto-reload register	0x0000
R16_TIM9_RPTCR	0x40014C30	Repeat count register	0x0000
R16_TIM9_CH1CVR	0x40014C34	Compare/capture register1	0x0000
R16_TIM9_CH2CVR	0x40014C38	Compare/capture register2	0x0000
R16_TIM9_CH3CVR	0x40014C3C	Compare/capture register3	0x0000
R16_TIM9_CH4CVR	0x40014C40	Compare/capture register4	0x0000
R16_TIM9_BDTR	0x40014C44	Break and dead zone register	0x0000
R16_TIM9_DMACFGR	0x40014C48	DMA configuration register	0x0000
R16_TIM9_DMAADR	0x40014C4C	DMA address register in continuous mode	0x0000

Table 14-6 TIM10 registers

Name	Access address	Description	Reset value
R16_TIM10_CTLR1	0x40015000	Control register1	0x0000
R16_TIM10_CTLR2	0x40015004	Control register2	0x0000
R16_TIM10_SMCFGR	0x40015008	Slave mode configuration register	0x0000
R16_TIM10_DMAINTENR	0x4001500C	DMA/interrupt enable register	0x0000
R16_TIM10_INTFR	0x40015010	Interrupt flag register	0x0000
R16_TIM10_SWEVGR	0x40015014	Event generation register	0x0000
R16_TIM10_CHCTLR1	0x40015018	Compare/capture control register1	0x0000
R16_TIM10_CHCTLR2	0x4001501C	Compare/capture control register2	0x0000
R16_TIM10_CCER	0x40015020	Compare/capture enable register	0x0000
R16_TIM10_CNT	0x40015024	Counter	0x0000
R16_TIM10_PSC	0x40015028	Prescaler	0x0000
R16_TIM10_ATRLR	0x4001502C	Auto-reload register	0x0000
R16_TIM10_RPTCR	0x40015030	Repeat count register	0x0000
R16_TIM10_CH1CVR	0x40015034	Compare/capture register1	0x0000
R16_TIM10_CH2CVR	0x40015038	Compare/capture register2	0x0000
R16_TIM10_CH3CVR	0x4001503C	Compare/capture register3	0x0000
R16_TIM10_CH4CVR	0x40015040	Compare/capture register4	0x0000
R16_TIM10_BDTR	0x40015044	Break and dead zone register	0x0000
R16_TIM10_DMACFGR	0x40015048	DMA configuration register	0x0000
R16_TIM10_DMAADR	0x4001504C	DMA address register in continuous mode	0x0000

14.4.1 Control Register 1 (TIMx_CTLR1) (x=1/8/9/10)

Offset address: 0x00

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

Bit	Name	Access	Description	Reset value
[15:10]	Reserved	RO	Reserved.	0

[9:8]	CKD[1:0]	RW	These 2 bits define the frequency, dead-zone time of timer clock (CK_INT) and frequency division ratio of sampling clock used for the dead-zone generator and digital filter (ETR,TIx): 00: Tdts=Tck_int 01: Tdts = 2 * Tck_int 10: Tdts = 4 * Tck_int 11: Reserved.	0
7	ARPE	RW	Auto reload and preload enable bit: 1: Auto reload register (ATRLR) enabled; 0: Auto reload register (ATRLR) disabled.	0
[6:5]	CMS[1:0]	RW	Central alignment mode selection: 00: Edge alignment mode. The counter counts up or down according to the direction bit (DIR). 01: Center alignment mode1. 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 mode2. 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 mode3. 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.*	0
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 is invalid.*	0
3	OPM	RW	Single pulse mode: 1: The counter stops when the next update event (clearing the CEN bit) occurs. 0: The counter does not stop when the next update event occurs.	0
2	URS	RW	Update request source; the software selects the source of UEV event through this bit. 1: If the updating interrupt or DMA request is enabled, only the counter overflow/underflow will generate the	0

			update interrupt or DMA request; 0: If the update interrupt or DMA request is enabled, any of the following events will generate an update interrupt or DMA request. -Counter overflow/underflow -Set the UG bit - Generate update from the mode controller	
1	UDIS	RW	Update disabled. Software allows/disables the generation of UEV events through this bit. 1: Disable UEV. No update event is generated, and the registers (ARR, PSC, CCRx) maintain their values. If the UG bit is set or a hardware reset is issued from the mode controller, the counter and prescaler will be reinitialized; 0: Allow UEV. Update (UEV) events are generated by any of the following events: - Counter overflow/underflow -Set the UG bit - Generate update from the mode controller Registers with buffers are loaded with their preloaded values.	0
0	CEN	RW	Counter enable: 1: Counter enabled; 0: Counter disabled. *Note: After CEN bit is set by software, the external clock, gating mode and encoder mode can only work. The trigger mode can automatically set the CEN bit by hardware.*	0

14.4.2 Control Register2 (TIMx_CTLR2) (x=1/8/9/10)

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 status 4: 1: If the OC4N is implemented for MOE=0, OC1=1 after dead zone; 0: For MOE=0, if the OC4N is implemented, OC1=0 after dead zone. *Note: After levels 1, 2 and 3 have been set for LOCK (TIMx_BDTR register), such bit cannot be modified.*	0

13	OIS3N	RW	During the output idle status 3: 1: When MOE=0, OC1N=1 after the dead zone; 0: When MOE=0, OC1N=0 after the dead zone. *Note: After LOCK (TIMx_BDTR register) levels 1, 2 or 3 have been set, this bit cannot be modified.*	0
12	OIS3	RW	Output idle status 3, refer to OIS4.	0
11	OIS2N	RW	Output idle status 2, refer to OIS3N.	0
10	OIS2	RW	Output idle status 2, refer to OIS4.	0
9	OIS1N	RW	Output idle status 1, refer to OIS3N.	0
8	OIS1	RW	Output idle status 1, refer to OIS4.	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 in the TIMx_EGR register 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 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_SMCR register); 010: Update- An update event is selected as the trigger input (TRGO). For example, the clock of a master timer can be used as a prescaler for a slave timer; 011:Compare 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: Compare-OC1REF signal is used as trigger output (TRGO); 101: Compare-OC2REF signal is used as trigger output (TRGO); 110: Compare-OC3REF signal is used as trigger output	0

			(TRGO); 111: Compare-OC4REF signal is used as trigger output (TRGO).	
3	CCDS	RW	DMA selection of capture/compare 1: When an update event occurs, a DMA request of CHxCVR is transferred; 0: When CHxCVR occurs, a DMA request of CHxCVR is generated.	0
2	CCUS	RW	Compare/capture control update selection bit. 1: If CCPC is set, they can be updated by setting the COM bit or a rising edge on TRGI; 0: If CCPC is set, they can only be updated by setting the COM bit. *Note: This bit only works on channels with complementary outputs.*	0
1	Reserved	RO	Reserved.	0
0	CCPC	RW	Compare/capture preload control bit. 1: CCxE, CCxNE and OCxM bits are pre-loaded. After the bits are set, they will only be updated after setting of the COM bit; 0: CCxE, CCxNE and OCxM bits are not preloaded. Note: This bit only works on channels with complementary outputs.	0

14.4.3 Slave Mode Configuration Mode (TIMx_SMCFGR) (x=1/8/9/10)

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, valid at high level or rising edge.	0
14	ECE	RW	External clock mode 2 enable selection: 1: Enable the external clock mode 2; 2: Disable the external clock mode 2. *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 '111').* *Note 2: When both external clock mode 1 and external clock mode 2 are enabled at the same time, the input of the external clock will be ETRF.*	0

[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.
				0

[11:8]	ETF	RW	External trigger filtering. 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 effect.	0

[6:4]	TS[2:0]	RW	Trigger selection bits. Select the trigger input source used to synchronize the counter through these 3 bits:
			000: Internal trigger0 (ITR0);
			001: Internal trigger1 (ITR1);
			010: Internal trigger2 (ITR2);
			011: Internal trigger3 (ITR3);
			100: Edge detector of TI1 (TI1F_ED);
			101: Timer input1 after filtering (TI1FP1);
			110: Timer input2 after filtering (TI2FP2);
			111: External trigger input (ETRF).
			The values can be changed only when SMS is 0.
			Note: See Table 14-2 for details.
				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 mode 1; according to the level of TI1FP1, the core counter counts up or down on the edge of TI2FP2; 010: Encoder mode 2; according to the level of TI2FP2, the core counter counts up or down on the edge of TI1FP1; 011: Encoder mode 3; according to 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 mode 1; the rising edge of the selected trigger input (TRGI) drives the counter.	0

14.4.4 DMA/Interrupt Enable Register (TIMx_DMAINTENR) (x=1/8/9/10)

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 DMA request enable bit. 1: Trigger DMA request enabled; 0: Trigger DMA request disabled.	0
13	COMDE	RW	DMA request enable bit of COM. 1: DMA request of COM enabled; 0: DMA request of COM disabled.	0
12	CC4DE	RW	DMA request enable bit of compare/capture4. 1: DMA request of compare/capture4 enabled; 0: DMA request of compare/capture4 disabled.	0b
11	CC3DE	RW	DMA request enable bit of compare/capture3. 1: DMA request of compare/capture3 enabled; 0: DMA request of compare/capture3 disabled.	0

10	CC2DE	RW	DMA request enable bit of compare/capture2. 1: DMA request of compare/capture2 enabled; 0: DMA request of compare/capture2 disabled.	0
9	CC1DE	RW	DMA request enable bit 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 bit. 1: Update DMA request enable bit enabled. 0: Update DMA request enable bit disabled.	0b
7	BIE	RW	Break interrupt enable bit. 1: Break interrupt enabled; 0: Break interrupt disabled.	0
6	TIE	RW	Trigger interrupt enable bit. 1: Trigger interrupt enabled; 0: Trigger interrupt disabled.	0
5	COMIE	RW	COM interrupt enable bit. 1: COM interrupt enabled; 0: COM interrupt disabled.	0
4	CC4IE	RW	Interrupt enable bit of compare/capture4. 1: Interrupt of compare/capture4 enabled; 0: Interrupt of compare/capture4 disabled.	0
3	CC3IE	RW	Interrupt enable bit of compare/capture3. 1: Interrupt of compare/capture3 enabled; 0: Interrupt of compare/capture3 disabled.	0
2	CC2IE	RW	Interrupt enable bit of compare/capture2. 1: Interrupt of compare/capture2 enabled; 0: Interrupt of compare/capture2 disabled.	0
1	CC1IE	RW	Interrupt enable bit of compare/capture1. 1: Interrupt of compare/capture1 enabled; 0: Interrupt of compare/capture1 disabled.	0
0	UIE	RW	Update interrupt enable bit. 1: Update interrupt enabled; 0: Update interrupt disabled.	0

14.4.5 Interrupt Flag Register (TIMx_INTFR) (x=1/8/9/10)

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	Overcapture flag bit of compare/capture 4.	0
11	CC3OF	RW0	Overcapture flag bit of compare/capture 3.	0
10	CC2OF	RW0	Overcapture flag bit of compare/capture 2.	0

9	CC1OF	RW0	Overcapture flag bit of compare/capture 1 is only used when the compare/capture is configured in the input capture mode. This flag bit is set by hardware, and cleared by writing 0 by software. 1:When the value of counter is captured into compare/capture register, status of CC1IF has been set; 0: No overcapture is generated.	0b
8	Reserved	RO	Reserved.	0
7	BIF	RW0	Break interrupt flag bit, once the break input is valid, the bit is set by hardware and cleared by software. 1: The set effective level is detected on break pin input; 0: No break event is generated.	0
6	TIF	RW0	Trigger interrupt flag bit; 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	COMIF	RW0	COM interrupt flag bit; once a COM event occurs, this bit is set by hardware and cleared by software. COM interrupt flag bit; once a COM event occurs, this bit is set by hardware and cleared by software. 1: A COM event occurs; 2. No COM event occurs.	0
4	CC4IF	RW0	Interrupt flag bit of compare/capture 4.	0
3	CC3IF	RW0	Interrupt flag bit of compare/capture 3.	0
2	CC2IF	RW0	Interrupt flag bit of compare/capture 2.	0
1	CC1IF	RW0	Interrupt flag bit of compare/capture 1. If the compare/capture is configured as output mode: This bit is set by hardware when the counter value matches the comparison value, except in the center symmetric mode. This bit is cleared by software. 1: The value of the core counter matches the value of the compare/capture register 1; 0: no match occurs. If the compare/capture is configured as input 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 bit. When an update event occurs, this bit is set by hardware and cleared by software. 1: Update interrupt is generated; 0: No update interrupt is generated.	0

			The update event occurs 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;	

14.4.6 Event Generation Register (TIMx_SWEVGR) (x=1/8/9/10)

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	Break event generation bit; this bit is set and cleared by software to generate a break event. 1: A break event is generated. At this time, MOE=0, BIF=1; if the corresponding interrupt and DMA are enabled, the corresponding interrupt and DMA will be generated; 0: No effect.	0
6	TG	WO	Trigger event generation bit; this bit is set by software and cleared by hardware to generate a trigger event. 1: Generate a trigger event; 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	COMG	WO	Compare/capture control update generation bit. Generating compare/capture control update event. This bit is set by software and cleared automatically by hardware. 1: When CCPC=1, it is allowed to update the CCxE, CCxNE and OCxM bits; 0: No effect. *Note: This bit is only valid for channels with complementary outputs (channels 1, 2 and 3).*	0
4	CC4G	WO	Compare/capture event generates bit4. Generating compare/capture event 4.	0
3	CC3G	WO	Compare/capture event generates bit3. Generating the compare/capture event 3.	0
2	CC2G	WO	Compare/capture event generates bit2. Generating compare/capture event 2.	0

1	CC1G	WO	Compare/capture event generates bit1. Generating compare/capture event 1. This bit is set by software and cleared by hardware. It is used to generate a compare/capture event. 1:Generating a compare/capture event on the compare/capture 1: If the compare/capture 1 is configured as output: CC1IF bit is set. If the corresponding interrupt and DMA are enabled, the corresponding interrupt and DMA is generated; If the compare/capture 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 is generated. If the CC1IF bit has been set, set the CC1OF bit. 0: No effect.	0
0	UG	WO	The update event generation bit generates the update event. 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 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=1/8/9/10)

Offset address: 0x18

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
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	Clear enable bit of compare/capture 2. 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	Mode setting of compare/capture 2. The 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 OC1REF; 001: Forced to be an effective level. When the core counter and compare/capture register 1 have the same value, force OC1REF 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 OC1REF to be low; 011: Overturn. When the core counter and compare/capture register 1 have the same value, overturn the level of OC1REF; 100: Force to be inactive level. Force OC1REF to be low. 101: Force to be inactive level. Force OC1REF to be high. 110: PWM mode 1: When counting up, once the core counter is greater than the value of the compare/capture register, channel 1 will be at inactive level. Otherwise, it will be at active level. When counting down, once the core counter is greater than the value of the compare/capture register, channel 1 will be at active level. Otherwise, it will be at inactive level. 111: PWM mode 2: When counting up, once the core counter is greater than the value of the compare/capture register, channel 1 will be at active level. Otherwise, it will be at inactive level. When counting down, once the core counter is greater than the value of the compare/capture register, channel 1 will be at inactive level. Otherwise, it will be at active level (OC1REF=1). *Note: Once the LOCK level is set to 3 and CC1S=00b, this bit cannot be modified. In PWM mode 1 or PWM mode 2, the OC1REF level changes only when compare result changes or when switching from freezing mode to PWM mode in the output comparison mode.*	0
11	OC2PE	RW	Preload enable bit of compare/capture register 2. 1: Enable the preload function of the comparison capture register 2. 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	0

			shadow register when the update event arrives; 0: Disable the preload function of capture register 2; it can be written to compare/capture register 2 at any time, and the newly written value will take effect immediately. *Note: Once the LOCK level is set to 3 and CC1S=00, 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.*	
10	OC2FE	RW	Compare/capture 2 fast enable bit; this bit is used to speed up the response of the compare/capture output to the trigger input event. 1: The effect of the valid 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 the 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 will be 5 clock cycles. OC2FE only works when the channel is configured in PWM1 or PWM2 mode.	0
[9:8]	CC2S[1:0]	RW	Input selection of compare/capture 2. 00: Compare/capture 2 is configured as output; 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 TS bit). *Note: Compare/capture 2 is only writable when the channel is switched off (CC2E is zero).*	0
7	OC1CE	RW	Compare/capture 1 clear enable bit.	0
[6:4]	OC1M[2:0]	RW	Mode setting bits of compare/capture 1.	0
3	OC1PE	RW	Preload enable bit of compare/capture register 1.	0
2	OC1FE	RW	Fast enable bit of compare/capture 1.	0
[1:0]	CC1S[1:0]	RW	Input selection bits of compare/capture 1.	0

Capture mode (pin direction is input):

Bit	Name	Access	Description	Reset value

[15:12]	IC2F[3:0]	RW	Input capture2 filter configuration bits; 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 bits; these 2 bits define prescaler factor of compare/capture 2. Once CC1E=0, the prescaler will be reset. 00: No prescaler, each edge detected on the capture input port triggers a capture; 01: Trigger a capture every 2 events; 10: Trigger a capture every 4 events; 11: Trigger a capture every 8 events;	0
[9:8]	CC2S[1:0]	RW	Compare/capture 2 input selection bits. These 2 bits define the direction of the channel (input/output) and the selection of input pins. 00: Compare/capture 1 is configured as output; 01: Compare/capture 1 is configured as input, and IC1 is mapped on TI1; 10: Compare/capture 1 is configured as input, and IC1 is mapped on TI2; 11: Compare/capture 1 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 is writable only when the channel is closed (CC1E is 0).*	0
[7:4]	IC1F[3:0]	RW	Input capture1 filter configuration bits.	0
[3:2]	IC1PSC[1:0]	RW	Prescale configuration bits of compare/capture 1.	0
[1:0]	CC1S[1:0]	RW	Input selection bits of compare/capture 1.	0

14.4.8 Compare/Capture Control Register 2 (TIMx_CHCTLR2) (x=1/8/9/10)

Offset address: 0x1C

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
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	Clear enable bit of compare/capture 4.	0
[14:12]	OC4M[2:0]	RW	Mode setting of compare/capture 4.	0
11	OC4PE	RW	Preload enable bit of compare/capture register 4.	0
10	OC4FE	RW	Fast enable bit of compare/capture 4.	0
[9:8]	CC4S[1:0]	RW	Input selection of compare/capture 4.	0
7	OC3CE	RW	Clear enable bit of compare/capture 3.	0
[6:4]	OC3M[2:0]	RW	Mode setting of compare/capture 3.	0
3	OC3PE	RW	Preload enable bit of compare/capture register 3.	0
2	OC3FE	RW	Fast enable bit of compare/capture 3.	0
[1:0]	CC3S[1:0]	RW	Input selection of compare/capture 3.	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	Prescale configuration of compare/capture 4.	0
[9:8]	CC4S[1:0]	RW	Input selection of compare/capture 4.	0
[7:4]	IC3F[3:0]	RW	Input capture 3 filter configuration.	0
[3:2]	IC3PSC[1:0]	RW	Prescale configuration of compare/capture 3.	0
[1:0]	CC3S[1:0]	RW	Input selection of compare/capture 3.	0

14.4.9 Compare/Capture Enable Register (TIMx_CCER) (x=1/8/9/10)

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	Output polarity setting bit of compare/capture 4.	0
12	CC4E	RW	Output enable bit of compare/capture 4.	0

11	CC3NP	RW	Complementary output polarity setting bit of compare/capture 3.	0
10	CC3NE	RW	Complementary output enable bit of compare/capture 3.	0
9	CC3P	RW	Output polarity setting bit of compare/capture 3.	0
8	CC3E	RW	Output enable bit of compare/capture 3.	0
7	CC2NP	RW	Complementary output polarity setting bit of compare/capture 2.	0
6	CC2NE	RW	Complementary output enable bit of compare/capture 3.	0
5	CC2P	RW	Output polarity setting bit of compare/capture 2.	0
4	CC2E	RW	Output enable bit of compare/capture 2.	0
3	CC1NP	RW	Complementary output polarity setting bit of compare/capture 1.	0
2	CC1NE	RW	Complementary output enable bit of compare/capture 1.	0
1	CC1P	RW	Output polarity setting bit of compare/capture 1. CC1 channel configured as output: 1: OC1 active low. 0: OC1 active high. 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: Inverted: capture occurs on the falling edge of IC1; when used as an external trigger, IC1 is inverted. 0: Non-inverted: capture occurs on the rising edge of IC1; when used as an external trigger, IC1 is not inverted. *Note: Once the LOCK level (LOCK bit in TIMx_BDTR register) is set to 3 or 2, this bit cannot be modified.*	0
0	CC1E	RW	Output enable bit of compare/capture 1. The CC1 channel is configured as output: 1: On. The OC1 signal is output to the corresponding output pin, and its output level depends on the values of the MOE, OSSI, OSSR, OIS1, OIS1N, and CC1NE bits. 0: off. OC1 disables 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 as an input: This bit determines whether the counter value can be captured into the TIMx_CCR1 register. 1: capture enable. 0: capture disable.	0

14.4.10 Counter of Advanced-control Timer (TIMx_CNT) (x=1/8/9/10)

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's counter.	0

14.4.11 Count Clock Prescaler (TIMx_PSC) (x=1/8/9/10)

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 divider/(PSC+1).	0

14.4.12 Auto-reload Register (TIMx_ATRLR) (x=1/8/9/10)

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 these bits is loaded into the counter. Please refer to Section 14.2.3 for ATRLR acting and update time; when ATRLR is empty, the counter will stop.	0

14.4.13 Repeat Count Register (TIMx_RPTCR) (x=1/8/9/10)

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	Repeated counter value.	0

14.4.14 Compare/Capture Register 1 (TIMx_CH1CVR) (x=1/8/9/10)

Offset address: 0x34

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
[15:0]	CCR1[15:0]	RW	Value of compare/capture register channel 1.	0

14.4.15 Compare/Capture Register 2 (TIMx_CH2CVR) (x=1/8/9/10)

Offset address: 0x38

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
[15:0]	CCR2[15:0]	RW	Value of compare/capture register channel 2.	0

14.4.16 Compare/Capture Register 3 (TIMx_CH3CVR) (x=1/8/9/10)

Offset address: 0x3C

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
[15:0]	CCR3[15:0]	RW	Value of compare/capture register channel 3.	0

14.4.17 Compare/Capture Register4 (TIMx_CH4CVR) (x=1/8/9/10)

Offset address: 0x40

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
[15:0]	CCR4[15:0]	RW	Value of compare/capture register channel 4.	0

14.4.18 Break and Dead Zone Register (TIMx_BDTR) (x=1/8/9/10)

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	Master output enable bit. Once a break signal is effective, it will be cleared asynchronously. 1: Enable to set OCx and OCxN as output; 0: Disable the output of OCx and OCxN or setting it to idle status by force.	0
14	AOE	RW	Automatic 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	Break input polarity setting bit. 1: Break input active at high level; 0: Break input active at low level. *Note: When LOCK level 1 is set, this bit cannot be modified. Writing to this bit requires an APB clock to take effect.*	0
12	BKE	RW	Break function enable bit. 1: Enable break input; 0: Disable break input. *Note: When LOCK level 1 is set, this bit cannot be modified. Writing to this bit requires an APB clock to take effect.*	0
11	OSSR	RW	1: When the timer is not working, once CCxE=1 or CCxNE=1, firstly turn on OC/OCN and output an inactive level, and then set the OCx and OCxN to enable output signal =1; 0: Disable OC/OCN output when the timer is not working. *Note: When LOCK level 1 is set, this bit cannot be modified.*	0
10	OSSI	RW	1: When the timer is not working, once CCxE=1 or CCxNE=1, OC/OCN will firstly output its idle level, and then OCx, OCxN will enable output signal=1; 0: Disable OC/OCN output when the timer is not working. *Note: When LOCK level 1 is set, this bit cannot be modified.*	0
[9:8]	LOCK[1:0]	RW	Lock function setting. 00: Switching off the lock function; 01: Lock level 1; DTG, BKE, BKP, AOE, OISx and OISxN bits cannot be written; 10: Lock level 2; you cannot write the bits in lock level 1, nor can you write the CC polarity bit, OSSR and OSSI bits; 11: Lock level 3; you cannot write each bit in lock level	0

			2, nor can you write the CC control bit. *Note: After the system reset, the LOCK bit can only be written once, and cannot be modified again until reset.*	

[7:0]	DTG[7:0]	RW	Dead zone setting bits; these bits define the duration of the dead zone between complementary outputs.
			Assume that DT represents 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.
				0

14.4.19 DMA Configuration Register (TIMx_DMACFGR) (x=1/8/9/10)

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 DMA continuous transfer; the actual value is the value of these bits + 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

14.4.20 DMA Address Register in Continuous Mode (TIMx_DMAADR) (x=1/8/9/10)

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
[15:0]	DMAB[15:0]	RW	DMA address in continuous mode.	0

Chapter 15 General-purpose Timer (GPTM)

This chapter applies to the whole family of CH32F2x, CH32V2x and CH32V3x.

The general-purpose timer module contains a 16-bit timer (TIM2, TIM3, TIM4 and TIM5) that can be automatically reloaded to measure pulse width or generate specific frequency pulse and PWM wave, etc. It can be used for automatic control and power.

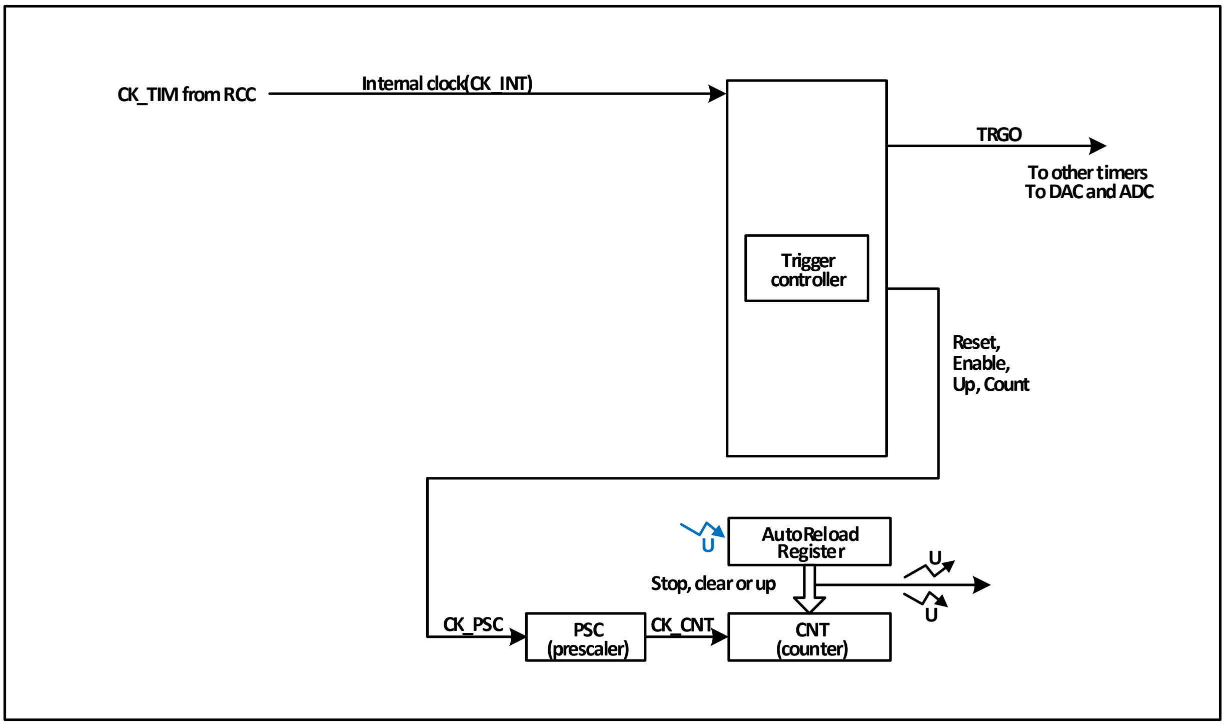
15.1 Main features

General purpose timer (TIM1/8/9/10) features include:

- 16-bit automatic reload counter, supports upcount, downcount and up/down-count
- 16-bit prescaler; the frequency division factor is dynamically adjustable from 1 to 65536
- Four independent compare/captures channels
- Each compare/capture channel supports multiple working modes, such as: input capture, output comparison, PWM generation and single pulse output
- External signal to control timer
- DMA generation in multiple modes

15.2 Principle and structure

Figure 15-1 Structure block diagram of general-purpose timer



15.2.1 Overview

As shown in Figure 15-1, the structure of the general-purpose timer can be roughly divided into 3 parts: Input clock, core counter and compare/capture channel.

The general-purpose timer clock can come from AHB bus clock (CK_INT), external clock input pin

(TIMx_ETR), other timers with clock output function (ITRx), or the input end of compare/capture channel (TIMx_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 peripherals such as timer, ADC and DAC.

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 outputs to CNT. CNT supports up-counting mode, down-counting mode and up/down counting mode, and there is an automatic reload value register (ATRLR). After each counting cycle is completed, CNT will be reloaded with the initial value.

The general-purpose timer has 4 groups of compare/captures. On each group of compare/capture, pulses can be inputted from its dedicated pins or output waveforms to the pins, i.e., the compare/captures 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 has a set of compare/capture register (CHxCVR), which supports comparison with the main counter (CNT) so as to output pulse.

15.2.2 Difference between general-purpose timer and advanced-control timer

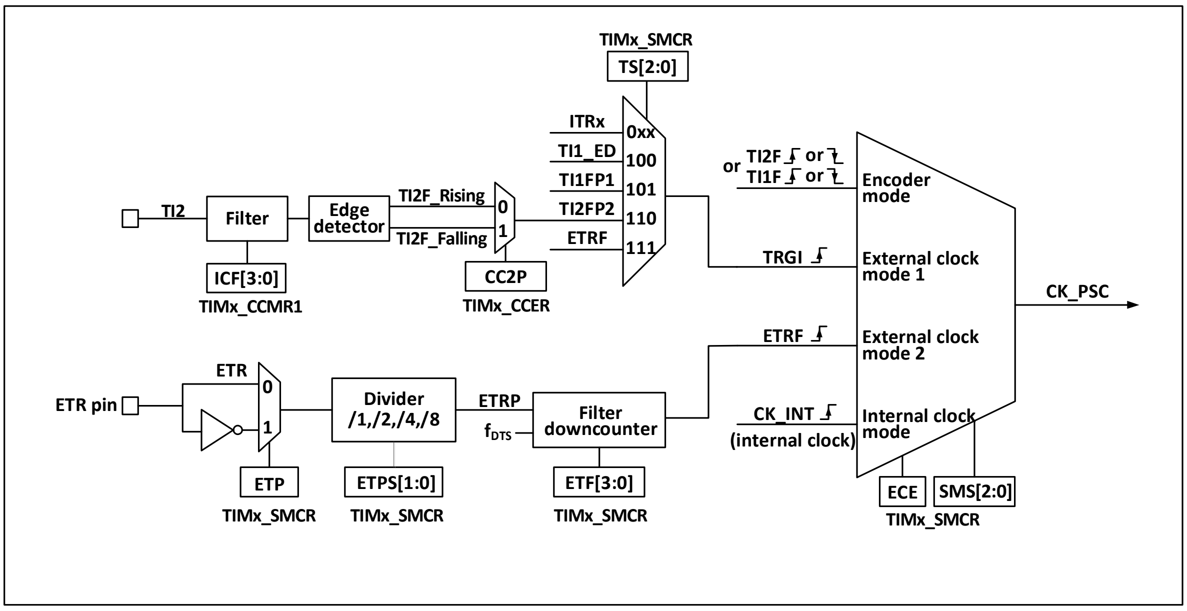
Compared with the advanced-control timer, the general-purpose timer is lack of the following functions:

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 zone generation and has no complementary output.
3) The general-purpose timer has no break signal mechanism.
4) The default clock CK_INT of the general-purpose timer comes from APB1, while the CK_INT of the advanced-control timer comes from APB2.

15.2.3 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 15-2 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 APB 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.

15.2.3.1 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.

15.2.3.2 External clock source mode1

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.

15.2.3.3 External clock source mode2

Use external trigger mode2 to count on every rising or falling edge of the external clock pin input. When the ECE bit is set, the external clock source mode2 is enabled. When the external clock source mode2 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.

15.2.3.4 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 Section14.3.9 for the specific functions.

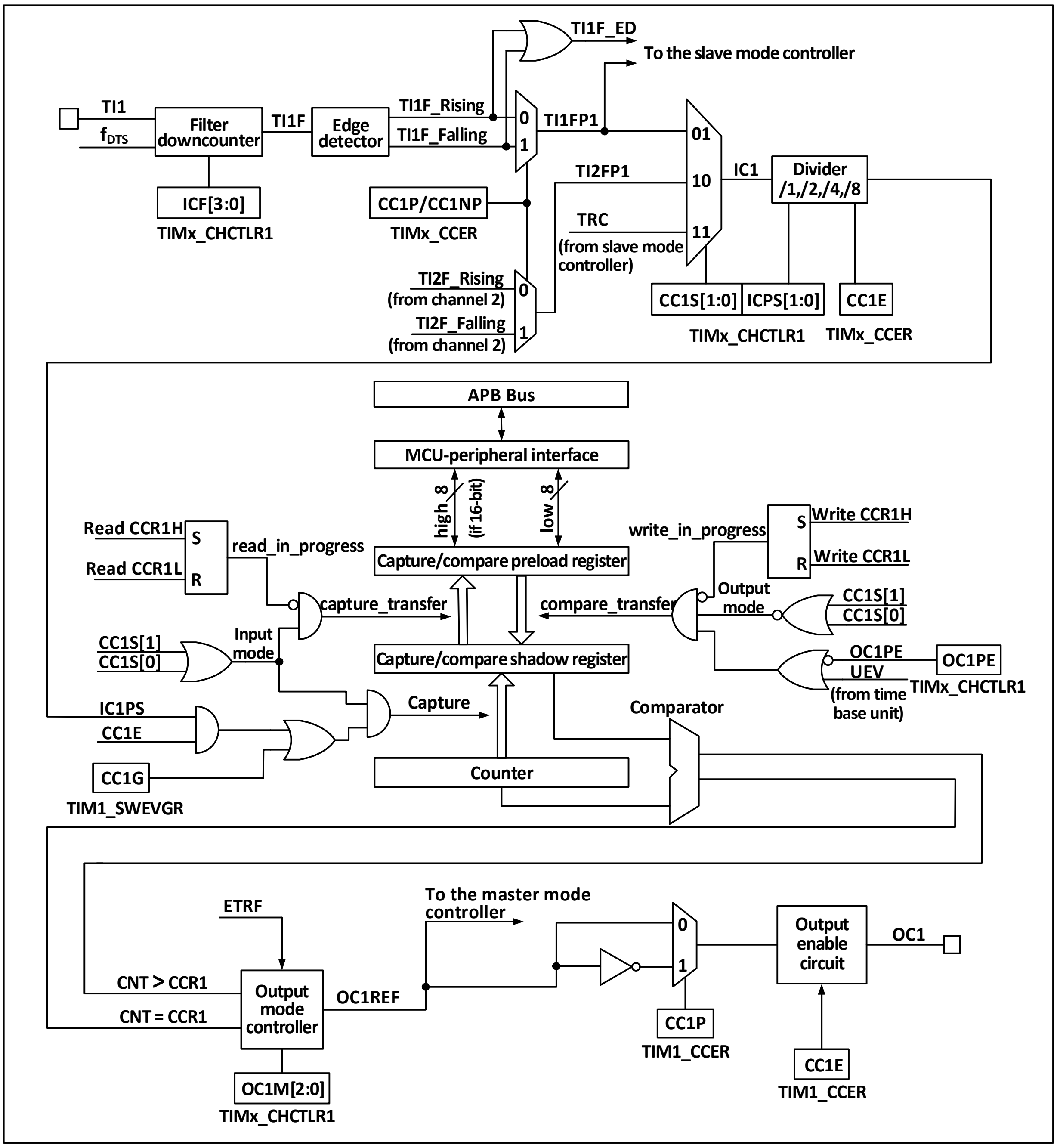
15.2.4 Counter and periphery

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_INT 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.

15.2.5 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 15-3.

Figure 15-3 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 14-1 timer block diagram). 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).

15.3 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.

15.3.1 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_TIMx_CHCTLRx). When a capture event occurs, CCxIF (in R16_TIMx_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 CCxS and select the source of ICx signal. For example, it is set to 10b, and TI1FP1 is selected as the source of IC1, and the default setting cannot be used. CCxS defaults to use the compare capture module as the output channel;
2) Configure ICxF and set the digital filter of the TI signal. The digital filter will output a jump based on the determined frequency and determined sampling times. The sampling frequency and times are determined by ICxF;
3) Configure the CCxP bit and set the polarity of TIxFPx. For example, maintain CC1P bit to be low and select the jump of rising edge;
4) Configure ICxPS and set ICx signal as the frequency division factor between ICxPS. For example, maintain the ICxPS as 00b without frequency division;
5) Configure the CCxE bit to allow to capture the core counter (CNT) value to the compare/capture register. Set the CC1E bit;
6) Configure the CCxIE and CCxDE 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) is recorded in the compare/capture register, and CC1IF is set. When CC1IF has been set before, the CCIOF bit will also be set. If the CC1IE bit 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 (R16_TIMx_SWEVGR).

15.3.2 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;

15.3.3 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.

15.3.4 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.

15.3.5 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.

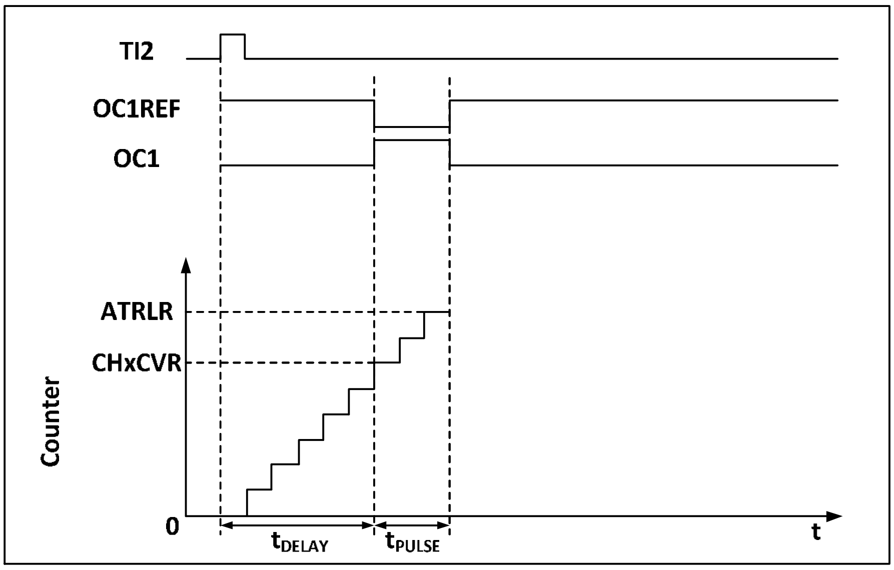
- Central alignment

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.

15.3.6 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 15-4 Event generation and pulse response



As shown in Figure 15-4, 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.

15.3.7 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 15-1 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

15.3.8 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 12-2 shows the internal trigger connection of timers.

Figure 15-2 GTPM internal trigger connection

Slave timer	ITR0(TS=000)	ITR1(TS=001)	ITR2(TS=010)	ITR3(TS=011)
TIM2	TIM1	TIM8/USB/ETH	TIM3	TIM4
TIM3	TIM1	TIM2	TIM5	TIM4
TIM4	TIM1	TIM2	TIM3	TIM8
TIM5	TIM2	TIM3	TIM4	TIM8

15.3.9 Debug mode

When the system enters debug mode, the timer continues to run or stops according to the setting of the DBG module.

15.4 Register description

Table 15-3 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

Name	Offset address	Description	Reset value
R16_TIM2_SWEVGR	0x40000014	TIM2 event generation register	0x0000
R16_TIM2_CHCTLR1	0x40000018	TIM2 compare/capture control register1	0x0000
R16_TIM2_CHCTLR2	0x4000001C	TIM2 compare/capture control register2	0x0000
R16_TIM2_CCER	0x40000020	TIM2 compare/capture enable register	0x0000
R16_TIM2_CNT	0x40000024	TIM2 counter	0x0000
R16_TIM2_PSC	0x40000028	TIM2 prescaler	0x0000
R16_TIM2_ATRLR	0x4000002C	TIM2 auto-reload register	0x0000
R16_TIM2_CH1CVR	0x40000034	TIM2 compare/capture register1	0x0000
R16_TIM2_CH2CVR	0x40000038	TIM2 compare/capture register2	0x0000
R16_TIM2_CH3CVR	0x4000003C	TIM2 compare/capture register3	0x0000
R16_TIM2_CH4CVR	0x40000040	TIM2 compare/capture register4	0x0000
R16_TIM2_DMACFGR	0x40000048	TIM2 DMA configuration register	0x0000
R16_TIM2_DMAADR	0x4000004C	TIM2 DMA address register in continuous mode	0x0000

Table 15-4 TIM3 registers

Name	Offset address	Description	Reset value
R16_TIM3_CTLR1	0x40000400	TIM3 control register1	0x0000
R16_TIM3_CTLR2	0x40000404	TIM3 control register2	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 register1	0x0000
R16_TIM3_CHCTLR2	0x4000041C	TIM3 compare/capture control register2	0x0000
R16_TIM3_CCER	0x40000420	TIM3 compare/capture enable register	0x0000
R16_TIM3_CNT	0x40000424	TIM3 counter	0x0000
R16_TIM3_PSC	0x40000428	TIM3 prescaler	0x0000
R16_TIM3_ATRLR	0x4000042C	TIM3 auto reload register	0x0000
R16_TIM3_CH1CVR	0x40000434	TIM3 compare/capture register1	0x0000
R16_TIM3_CH2CVR	0x40000438	TIM3 compare/capture register2	0x0000
R16_TIM3_CH3CVR	0x4000043C	TIM3 compare/capture register3	0x0000
R16_TIM3_CH4CVR	0x40000440	TIM3 compare/capture register4	0x0000
R16_TIM3_DMACFGR	0x40000448	TIM3 DMA configuration register	0x0000
R16_TIM3_DMAADR	0x4000044C	TIM3 DMA address register in continuous mode	0x0000

Table 15-5 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	0x0000
R16_TIM4_PSC	0x40000828	TIM4 prescaler	0x0000
R16_TIM4_ATRLR	0x4000082C	TIM4 auto-reload register	0x0000
R16_TIM4_CH1CVR	0x40000834	TIM4 compare/capture register1	0x0000
R16_TIM4_CH2CVR	0x40000838	TIM4 compare/capture register2	0x0000
R16_TIM4_CH3CVR	0x4000083C	TIM4 compare/capture register3	0x0000
R16_TIM4_CH4CVR	0x40000840	TIM4 compare/capture register4	0x0000
R16_TIM4_DMACFGR	0x40000848	TIM4 DMA configuration register	0x0000
R16_TIM4_DMAADR	0x4000084C	TIM4 DMA address register in continuous mode	0x0000

Table 15-6 TIM5 registers

Name	Offset address	Description	Reset value
R16_TIM5_CTLR1	0x40000C00	TIM5 control register 1	0x0000
R16_TIM5_CTLR2	0x40000C04	TIM5 control register 2	0x0000
R16_TIM5_SMCFGR	0x40000C08	TIM5 slave mode configuration register	0x0000
R16_TIM5_DMAINTENR	0x40000C0C	TIM5 DMA/interrupt enable register	0x0000
R16_TIM5_INTFR	0x40000C10	TIM5 interrupt flag register	0x0000
R16_TIM5_SWEVGR	0x40000C14	TIM5 event generation register	0x0000
R16_TIM5_CHCTLR1	0x40000C18	TIM5 compare/capture register 1	0x0000
R16_TIM5_CHCTLR2	0x40000C1C	TIM5 compare/capture control register2	0x0000
R16_TIM5_CCER	0x40000C20	TIM5 compare/capture enable register	0x0000
R16_TIM5_CNT	0x40000C24	TIM5 counter	0x0000
R16_TIM5_PSC	0x40000C28	TIM5 prescaler	0x0000
R16_TIM5_ATRLR	0x40000C2C	TIM5 auto-reload register	0x0000
R16_TIM5_CH1CVR	0x40000C34	TIM5 compare/capture register1	0x0000
R16_TIM5_CH2CVR	0x40000C38	TIM5 compare/capture register2	0x0000

R16_TIM5_CH3CVR	0x40000C3C	TIM5 compare/capture register3	0x0000
R16_TIM5_CH4CVR	0x40000C40	TIM5 compare/capture register4	0x0000
R16_TIM5_DMACFGR	0x40000C48	TIM5 DMA configuration register	0x0000
R16_TIM5_DMAADR	0x40000C4C	TIM5 DMA address register in continuous mode	0x0000

15.4.1 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
Reserved						CKD[1:0]		ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN

Bit	Name	Access	Description	Reset value
[15:10]	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.
				0

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: 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.*	0

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 (clearing the CEN bit) occurs; 0: The counter does not stop when the next update event occurs.	0
2	URS	RW	Update request source; the software selects the source of UEV event through this bit. 1: If the update interrupt or DMA request is enabled, only the counter overflow/underflow will generate the update interrupt or DMA request; 0: If the update interrupt or DMA request is enabled, any of the following events will generate an update interrupt or DMA request: -Counter overflow/underflow -Set the UG bit - Update generated by the slave mode controller	0
1	UDIS	RW	Update disable. Software allows/disables the generation of UEV events through this bit. 1: Disable UEV. No update event is generated, and the registers (ATRLR, PSC and CHCTLRx) maintain their values. If the UG bit is set or a hardware reset is issued from the mode controller, the counter and prescaler will be reinitialized. 0: Allow UEV. Update (UEV) events are generated by any of the following events: − Counter overflow/underflow -Set the UG bit - Update generated by the slave mode controller Registers with buffers are loaded with their preloaded values.	0
0	CEN	RW	Counter enable. 1: Counter enabled; 0: Counter disabled. *Note: After the CEN bit is set by software, the external clock, gating mode and encoder mode can only work. The trigger mode can automatically set the CEN bit by hardware.*	0

15.4.2 Control Register2 (TIMx_CTLR2) (x=2/3/4/5)

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	CCUS	Reserved	CCPC

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 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 input (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).	0
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	CCUS	RW	Compare/capture control update selection. 1: If CCPC is set, they can be updated by setting the COM bit or a rising edge on TRGI; 0: If CCPC is set, they can only be updated by setting the COM bit. *Note: This bit only works on channels with complementary outputs.*	0
1	Reserved	RO	Reserved.	0
0	CCPC	RW	Compare/capture preload control. 1: CCxE, CCxNE and OCxM bits are pre-loaded. After the bits are set, they will only be updated after setting of the COM bit; 0: CCxE, CCxNE and OCxM bits are not preloaded. *Note: This bit only works on channels with complementary outputs.*	0

15.4.3 Slave Mode Configuration Register (TIMx_SMCFGR) (x=2/3/4/5)

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 mode2 enable. 1: External clock mode2 enabled; 0: External clock mode2 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 clock mode 2 are enabled at the same time, the input of the external clock will be ETRF.*	0

[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.
				0

[11:8]	ETF[3:0]	RW	External trigger filter. In fact, the digital filter is an	0

			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;	

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	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);
			The values can be changed only when SMS is 0.
				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	0

			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.	

15.4.4 DMA/Interrupt Enable Register (TIMx_DMAINTENR) (x=2/3/4/5)

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	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	COMDE	RW	DMA request enable of COM. 1: DMA request of COM enabled; 0: DMA request of COM disabled.	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. 1: DMA request of compare/capture2 enabled; 0: DMA request of compare/capture2 disabled.	0
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

15.4.5 Interrupt Flag Register (R16_TIMx_INTFR) (x=2/3/4/5)

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	W0	Compare/capture4 overcapture flag.	0
11	CC3OF	W0	Compare/capture3 overcapture flag.	0
10	CC2OF	W0	Compare/capture2 overcapture flag.	0
9	CC1OF	W0	Compare/capture1 overcapture 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. 1: When the value of the counter is captured into the capture comparison register, the status of CC1IF has been set; 0: No Overcapture is generated.	0
[8:7]	Reserved	RO	Reserved.	0
6	TIF	W0	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.	0

			1: Trigger event occurs; 0: No trigger event occurs.	
5	Reserved	RO	Reserved.	0
4	CC4IF	W0	Compare/capture4 interrupt flag.	0
3	CC3IF	W0	Compare/capture3 interrupt flag.	0
2	CC2IF	W0	Compare/capture2 interrupt flag.	0
1	CC1IF	W0	Compare/capture1 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	W0	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

15.4.6 Event Generation Register (TIMx_SWEVGR) (x=2/3/4/5)

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	Break event generation. Set and cleared by software to generate a break event. 1: A break event generated. In this case, MOE=0,	0

			BIF=1; if the corresponding interrupt and DMA are enabled, the corresponding interrupt and DMA will be generated; 0: No effect.	
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	COMG	WO	Compare/capture control update generation bit. Generating compare/capture control update event. This bit is set by software and cleared automatically by hardware. 1: When CCPC=1, it is allowed to update the CCxE, CCxNE and OCxM bits; 0: No effect. *Note: This bit is only valid for channels with complementary outputs (channels 1, 2 and 3).*	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

15.4.7 Compare/Capture Control Register1 (TIMx_CHCTLR1) (x=2/3/4/5)

Offset address: 0x18

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
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/capture2 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 OC1REF; 001: Active by force. When the core counter and compare/capture register1 have the same value, force OC1REF 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 OC1REF to be low; 011: Overturn. When the core counter and compare/capture register1 have the same value, overturn the level of OC1REF; 100: Inactive by force. Force OC1REF to be low. 101: Force to be active level. Force OC1REF to be high. 110: PWM mode 1: When up counting, once the core counter is greater than the value of the compare/capture register, channel 1 is at inactive level. Otherwise, it is at active level. When down counting, once the core counter is greater than the value of the compare/capture register, channel 1 is at active level. Otherwise, it is at inactive level. 111: PWM mode 2: When up counting, once the core	0

			counter is greater than the value of the compare/capture register, channel 1 is at active level. Otherwise, it is at inactive level. When down counting, once the core counter is greater than the value of the compare/capture register, channel 1 is at inactive level. Otherwise, it is at active level (OC1REF=1). *Note: Once the LOCK level is set to 3 and CC1S=00b, this bit cannot be modified. In PWM mode1 or PWM mode2, the OC1REF 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 register2 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 register1 is loaded into the current shadow register when the update event arrives; 0: Disable the pre-loading function of compare/capture register2. Compare/capture register2 can be written at any time, and the newly written value takes effect immediately. *Note: Once the LOCK level is set to 3 and CC1S=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/capture2 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/capture2 is shortened to 3 clock cycles; 0: According to the value of counter and compare/capture register2, 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/capture2 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/capture2 input selection. 00: The compare/capture2 is configured as output; 01: Compare/capture2 is configured as input, and IC2 is mapped on TI2;	0

			10: Compare/capture2 is configured as input, and IC2 is mapped on TI1; 11: Compare/capture2 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/capture2 is only writable when the channel is switched off (CC2E is zero).*	
7	OC1CE	RW	Compare/capture1 clear enable.	0
[6:4]	OC1M[2:0]	RW	Compare/capture1 mode setting.	0
3	OC1PE	RW	Compare/capture register1 preload enable.	0
2	OC1FE	RW	Compare/capture1 fast enable.	0
[1:0]	CC1S[1:0]	RW	Compare/capture1 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/capture2 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; 01: Trigger a capture every 2 events; 10: Trigger a capture every 4 events; 11: Trigger a capture every 8 events;	0

[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 capture1 filter configuration.	0
[3:2]	IC1PSC[1:0]	RW	Compare/capture1 prescaler configuration.	0
[1:0]	CC1S[1:0]	RW	Xompare/capture1 input selection.	0

15.4.8 Compare/Capture Control Register 2 (TIMx_CHCTLR2) (x=2/3/4/5)

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/capture4 clear enable	0
[14:12]	OC4M[2:0]	RW	Compare/capture4 mode setting	0
11	OC4PE	RW	Compare/capture4 preload enable	0
10	OC4FE	RW	Compare/capture4 fast enable	0
[9:8]	CC4S[1:0]	RW	Compare/capture4 input selection	0
7	OC3CE	RW	Compare/capture3 clear enable	0
[6:4]	OC3M[2:0]	RW	Compare/capture3 mode setting	0
3	OC3PE	RW	Compare/capture3 preload enable	0
2	OC3FE	RW	Compare/capture3 fast enable	0
[1:0]	CC3S[1:0]	RW	Compare/capture3 input selection	0

Capture mode (pin direction is input):

Bit	Name	Access	Description	Reset value

[15:12]	IC4F[3:0]	RW	Input capture4 filter configuration	0
[11:10]	IC4PSC[1:0]	RW	Compare/capture4 prescale configuration	0
[9:8]	CC4S[1:0]	RW	Compare/capture4 input selection	0
[7:4]	IC3F[3:0]	RW	Input capture3 filter configuration	0
[3:2]	IC3PSC[1:0]	RW	Compare/capture3 prescale configuration	0
[1:0]	CC3S[1:0]	RW	Compare/capture3 input selection	0

15.4.9 Compare/Capture Enable Register (TIMx_CCER) (x=2/3/4/5)

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/capture4 output polarity	0
12	CC4E	RW	Compare/capture4 output enable	0
[11:10]	Reserved	RO	Reserved.	0
9	CC3P	RW	Compare/capture3 output polarity	0
8	CC3E	RW	Compare/capture3 output enable	0
[7:6]	CC2NP	RO	Reserved.	0
5	CC2P	RW	Compare/capture2 output polarity	0
4	CC2E	RW	Compare/capture2 output enable	0
[3:2]	Reserved	RO	Reserved.	0
1	CC1P	RW	Compare/capture1 output polarity CC1 channel configured as output: 1: OC1 active low. 0: OC1 active high. 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: Inverted: capture occurs on the falling edge of IC1; when used as an external trigger, IC1 is inverted. 0: not inverted: capture occurs on the rising edge of IC1; when used as an external trigger, IC1 is not inverted.	0
0	CC1E	RW	Compare/capture1 output enable The CC1 channel is configured to output: 1: On: OC1 signal is output to the corresponding output 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.	0

			1: capture enable. 0: capture disable.	

15.4.10 Counter of General-purpose Timer (TIMx_CNT) (x=2/3/4/5)

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

15.4.11 Prescaler (TIMx_PSC) (x=2/3/4/5)

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

15.4.12 Auto-reload Register (TIMx_ATRLR) (x=2/3/4/5)

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 14.2.4 for ATRLR acting and update time. When ATRLR is empty, the counter stops.	0

15.4.13 Compare/Capture Register1 (TIMx_CH1CVR) (x=2/3/4/5)

Offset address: 0x34

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
[15:0]	CCR1[15:0]	RW	Value of compare/capture channel 1.	0

15.4.14 Compare/Capture Register2 (TIMx_CH2CVR) (x=2/3/4/5)

Offset address: 0x38

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
[15:0]	CCR2[15:0]	RW	Value of compare/capture channel 2.	0

15.4.15 Compare/Capture Register3 (TIMx_CH3CVR) (x=2/3/4/5)

Offset address: 0x3C

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
[15:0]	CCR3[15:0]	RW	Value of compare/capture channel 3.	0

15.4.16 Compare/Capture Register4 (TIMx_CH4CVR) (x=2/3/4/5)

Offset address: 0x40

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
[15:0]	CCR4[15:0]	RW	Value of compare/capture channel 4.	0

15.4.17 DMA Control Register (TIMx_DMACFGR) (x=2/3/4/5)

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

15.4.18 DMA Address Register in Continuous Mode (TIMx_DMAADR) (x=2/3/4/5)

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
[15:0]	DMAB[15:0]	RW	DMA address in continuous mode.	0

Chapter 16 Basic Timer (BCTM)

This chapter applies to the whole family of CH32F2x, CH32V2x and CH32V3x.

The basic timer module contains a 16-bit auto-reload timer (TIM6 and TIM7), that can be used to count and generate interrupt/DMA request on the update event.

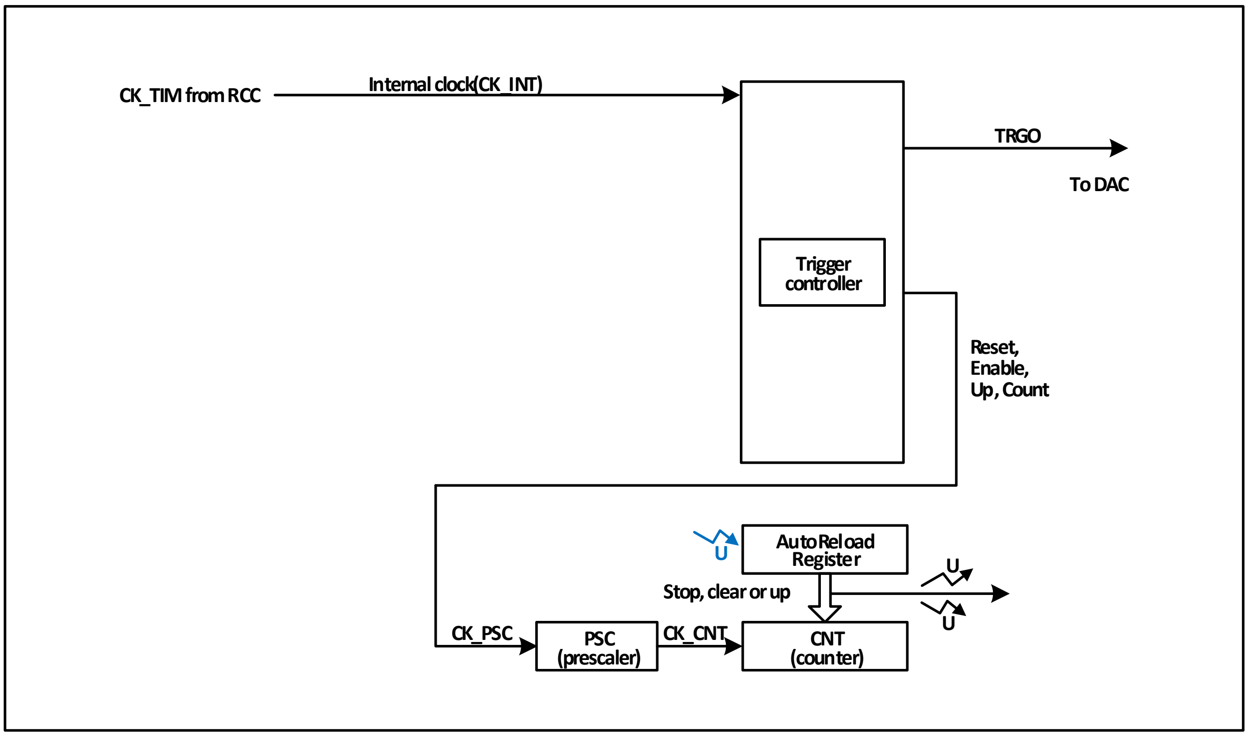
16.1 Main features

Basic timer features include:

- 16-bit auto-reload counter, supports upcount
- 16-bit prescaler, the frequency division factor is dynamically adjustable from 1 to 65536
- Synchronization circuit to trigger the DAC
- Interrupt/DMA request generation on the update event

16.2 Principle and structure

Figure 16-1 Structure block diagram of basic timer



16.2.1 Overview

As shown in Figure 16-1, the structure of the basic timer can be divided into 2 parts: input clock and core counter.

The clock source of basic timer is AHB bus clock (CK_INT). These input clock signals become CK_PSC clock after various set filtering and frequency division operations, and then they are output to the core counter. In addition, these complex clock sources can also be output to DAC peripheral as TRGO.

The core of the basic timer is a 16-bit counter (CNT). CK_PSC is divided by prescaler (PSC) and becomes CK_CNT, finally it is output to CNT. CNT supports upcount, and it has an auto-reload register (ATRLR), that reloads the initial value for the CNT every time a count cycle ends.

16.2.2 Difference between basic timer and general-purpose timer

Compared with general purpose timer, basic timer is lack of the following functions:

1) The basic timer does not support downcount or up/down count.

2) The basic timer does not have the 4 independent compare/capture channels.

3) The basic timer does not support external signals to control timer.

4) The basic timer does not support incremental code, or cascade connection/synchronization between timers.

16.2.3 Clock input

The clock of basic timer is provided by the internal clock (CK_INT).

16.2.4 Counter and periphery

CK_PSC is input 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_INT through PSC. The changed value of R16_TIM1_PSC does not take effect in real time, but can be updated to PSC after the update event. Update events include clearing the UG bit and resetting the UG bit.

16.3 Debug mode

When the system enters debug mode, the timer continues to run or stops according to the setting of the DBG module.

16.4 Register description

Table 16-1 TIM6 registers

Name	Offset address	Description	Reset value
R16_TIM6_CTLR1	0x40001000	TIM6 control register1	0x0000
R16_TIM6_CTLR2	0x40001004	TIM6 control register2	0x0000
R16_TIM6_DMAINTENR	0x4000100C	TIM6 DMA/interrupt enable register	0x0000
R16_TIM6_INTFR	0x40001010	TIM6 interrupt flag register	0x0000
R16_TIM6_SWEVGR	0x40001014	TIM6 event generation register	0x0000
R16_TIM6_CNT	0x40001024	TIM6 counter	0x0000
R16_TIM6_PSC	0x40001028	TIM6 prescaler	0x0000
R16_TIM6_ATRLR	0x4000102C	TIM6 auto-reload register	0x0000

Table 16-2 TIM7 registers

Name	Offset address	Description	Reset value
R16_TIM7_CTLR1	0x40001400	TIM7 control register1	0x0000
R16_TIM7_CTLR2	0x40001404	TIM7 control register2	0x0000
R16_TIM7_DMAINTENR	0x4000140C	TIM7 DMA/interrupt enable register	0x0000
R16_TIM7_INTFR	0x40001410	TIM7 interrupt flag register	0x0000

R16_TIM7_CNT	0x40001424	TIM7 counter	0x0000
R16_TIM7_PSC	0x40001428	TIM7 prescaler	0x0000
R16_TIM7_ATRLR	0x4000142C	TIM7 auto-reload register	0x0000

16.4.1 Control Register 1 (TIMx_CTLR1) (x=6/7)

Offset address: 0x00

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

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved	0
7	ARPE	RW	Auto reload and preload enable: 1: Auto-reload value register (ATRLR) enabled; 0: Auto-reload value register (ATRLR) disabled.	0
[6:4]	Reserved	R0	Reserved	0
3	OPM	RW	Single pulse mode. 1: The counter stops when the next update event (the CEN bit is cleared) occurs; 0: The counter does not stop when the next update event occurs.	0
2	URS	RW	Update request source; the software selects the source of UEV event through this bit. 1: If the update interrupt or if the DMA request is enabled, only the counter overflow/underflow will generate the update interrupt or DMA request; 0: If the update interrupt or if DMA request is enabled, any of the following events will generate an update interrupt or DMA request: -Counter overflow/underflow -Set the UG bit - Update generated by the slave mode controller	0
1	UDIS	RW	Update disable. Software enables/disables the generation of UEV events through this bit. 1: UEV disabled. No update event is generated, and the registers (ATRLR, PSC and CHCTLRx) maintain their values. If the UG bit is set or a hardware reset is sent by the slave mode controller, the counter and prescaler will be reinitialized. 0: UEV enabled. Update (UEV) events are generated by any of the following events: − Counter overflow/underflow -Set the UG bit - Update generated by the slave mode controller Registers with buffers are loaded with their preloaded	0

			values.	
0	CEN	RW	Counter enable. 1: Counter enabled; 0: Counter disabled.	0

16.4.2 Control Register 2 (TIMx_CTLR2) (x=6/7)

Offset address: 0x04

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

Bit	Name	Access	Description	Reset value
[15:7]	Reserved	RO	Reserved	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 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 input (TRGO). For example, the clock of a master timer can be used as a prescaler for a slave timer.	0
[3:0]	Reserved	RO	Reserved	0

16.4.3 DMA/Interrupt Enable Register (TIMx_DMAINTENR) (x=6/7)

Offset address: 0x0C

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

Bit	Name	Access	Description	Reset value
[15:9]	Reserved	RO	Reserved	0
8	UDE	RW	Update DMA request enable. 1: Update DMA request enabled; 0: Update DMA request disabled.	0
[7:1]	Reserved	RO	Reserved	0
0	UIE	RW	Update interrupt enable. 1: Update interrupt enabled; 0: Update interrupt disabled.	0

16.4.4 Interrupt Flag Register (R16_TIMx_INTFR) (x=6/7)

Offset address: 0x10

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

Bit	Name	Access	Description	Reset value
[15:1]	Reserved	RO	Reserved	0
0	UIF	RW0	Update interrupt flag. When an update event occurs, this bit is set by hardware and cleared by software. 1: Update interrupt generated; 0: No update interrupt generated. An update event generates in case of the following circumstances: UDIS=0, the repeat counter value overflows or underflows; URS=0, UDIS=0, the UG bit is set, or the counter core is reinitialized by software.	0

16.4.5 Event Generation Register (TIMx_SWEVGR) (x=6/7)

Offset address: 0x14

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

Bit	Name	Access	Description	Reset value
[15:1]	Reserved	RO	Reserved	0
0	UG	WO	Update event generation, to generate an update event. 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*	0

			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.	

16.4.6 Counter of General-purpose Timer (TIMx_CNT) (x=6/7)

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 counter.	0

16.4.7 Prescaler (TIMx_PSC) (x=6/7)

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	Frequency division factor of the timer's prescaler. The clock frequency of the counter is equal to the input frequency of the divider/(PSC+1).	0

16.4.8 Auto-reload Register (TIMx_ATRLR) (x=6/7)

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 14.2.4 for ATRLR acting and update time. When ATRLR is empty, the counter stops.	0

Chapter 17 Digital-to-Analog Converter (DAC)

This chapter applies to the whole family of CH32F2x, CH32V2x and CH32V3x.

The digital-to-analog converter (DAC) contains 2 converters that converts 8/12-bit digital input to 2-channel analog voltage output. The DAC module has built-in triangular-wave generator and noise-wave generator, and it supports multiple event trigger conversions and supports DMA.

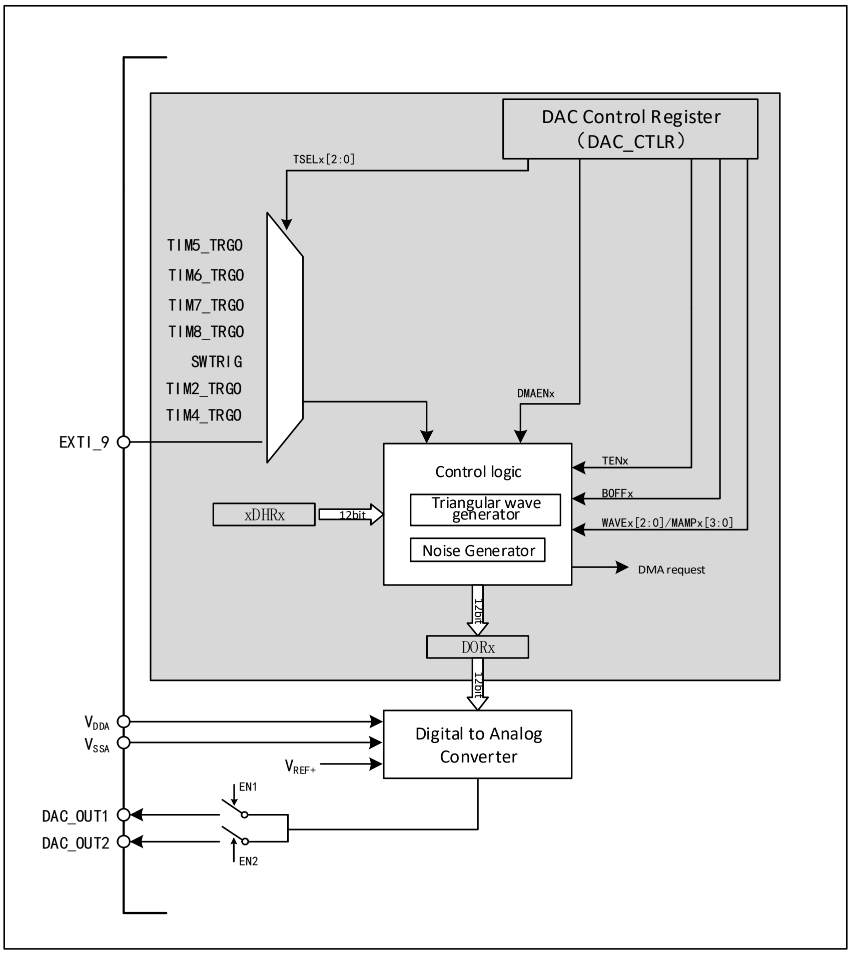
17.1 Main features

- Two DAC converters, and each has single output channel
- Triangular-wave generator and noise-wave generator
- Configurable 8-bit or 12-bit output
- Left or right data alignment in 12-bit mode
- Dual DAC channel for independent or simultaneous conversions
- DMA capability
- Multiple trigger events

17.2 Functional description

17.2.1 DAC structure

Figure 17-1 DAC block diagram



17.2.2 DAC channel configuration

17.2.2.1 Enable DAC:

Set the ENx bit in the DAC_CTLR register to 1, to enable the analog power to DAC channel x. After a period of start-up time, DAC channel x can be enabled. The DAC contains 2 analog output channels which can output independently or simultaneously.

Note: In order to avoid parasitic interference and additional power consumption, the corresponding pins of the DAC channel need to be set to analog input (AIN) mode in advance.

17.2.2.2 Enable output buffer:

DAC integrates output buffers, which can be used to reduce the output impedance and increase the drive capacity to directly drive the external loads. Each DAC channel output buffer can be enabled or disabled by setting the BOFFx bit in the DAC_CTLR register.

17.2.2.3 Data format:

In single DAC channel mode, data formats include 8-bit right alignment, 12-bit left alignment and 12-bit right alignment. Write data to DAC_R8BDHRx[7:0], and the module loads (after single APB1 clock cycle) the left shifted data to DAC_DORx[11:0]. Write data to DAC_R12BDHRx[11:0] , and the module loads (after single APB1 clock cycle) the right aligned data to DAC_DORx[11:0]. Write data to DAC_L12BDHRx[15:4] , and the module loads (after single APB1 clock cycle) the left aligned data to DAC_DORx[11:0]. In dual DAC channel mode, data formats also include 8-bit right alignment, 12-bit left alignment and 12-bit right alignment. Write data to DAC_RD8BDHR[7:0] , and the module loads (after single APB1 clock cycle) the shifted [7:0] bits to DAC_DOR1[11:0], and the shifted [15:8] bits to DAC_DOR2[11:0]. Write data to DAC_LD12BDHR[31:0], and the module loads (after single APB clock cycle) the shifted [15:4] bits to DAC_DOR1[11:4], and the shifted [31:20] bits to DAC_DOR2[11:4]. Write data to DAC_RD12BDHR[31:0], and the module loads (after single APB clock cycle) the shifted [11:0] bits to DAC_DOR1[11:0], and the shifted [27:16] bits to DAC_DOR2[11:0].

Figure 17-2 Data formats in single channel mode

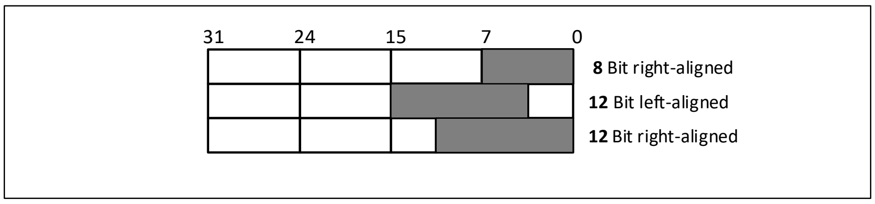
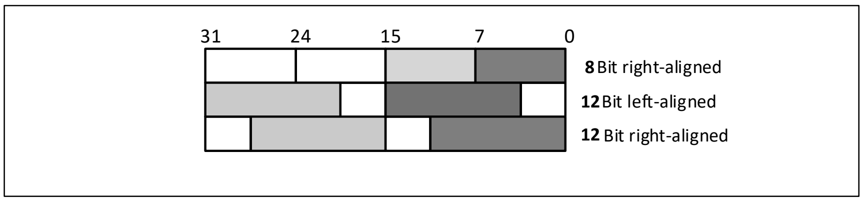


Figure 17-3 Data formats in dual channel mode



17.2.2.4 DMA capability:

DAC channel has the DMA capability. Set the DMAENx bit in the DAC_CTLR register to 1, to enable DMA of the corresponding channel. When a trigger event occurs (software trigger not included), a DMA request is generated. And then the data in the DAC_DORx register can be updated.

17.2.2.5 Trigger event selection:

DAC conversion can be triggered by the following events. When the TENx bit in the DAC_CTLR register is set to 1, a trigger event to trigger DAC conversion can be selected by configuring the TSELx[2:0] bits.

Table 17-1 Trigger events

Trigger source	Type	TSELx[2:0]
Timer 6 TRGO event	Internal signal from on-chip timer	000
Timer 8 TRGO event		001
Timer 7 TRGO event		010
Timer 5 TRGO event		011
Timer 2 TRGO event		100
Timer 4 TRGO event		101
EXTI line9	External pin	110
SWTRIG (software trigger)	Software control bit	111

Every time a DAC interface detects a rising edge on the selected timer TRGO output or on the selected external interrupt line 9, the DAC_DORx register is updated 3 APB1 clock cycles after the trigger occurs.

If the software trigger mode is configured, once the SWTRIG bit is set to 1, a conversion is started. The DAC_DORx register is updated single APB1 clock cycle after the trigger occurs, and the SWTRIG bit can be automatically cleared by hardware.

Note: The TSELx[2:0] bits cannot be changed when ENx is 1.

17.2.3 DAC conversion

The data of the DAC channel comes from the DAC_DORx register, but data cannot be directly written to the register DAC_DORx. Any data output to the DAC channel x must be written into the DAC_R12BDHR1, DAC_L12BDHR1, DAC_R12BDHR2, DAC_L12BDHR2, DAC_RD12BDHR, DAC_LD12BDHR and DAC_RD8BDHR registers. The internal DAC_DHRx register obtains the value of above registers and transfers it to the DAC_DORx register after the corresponding time.

In the non-trigger mode, the data written into the DAC_xDHRx register is shifted into the DAC_DORx register in an APB1 clock cycle.

In software trigger mode, the DAC_DORx register is automatically updated in an APB1 clock cycle after the rising edge of the event trigger.

In hardware trigger mode (timer TRGO event or external interrupt line 9 rising edge), the DAC_DORx register is automatically updated in 3 APB1 clock cycles after the trigger event.

Load the DAC_DORx register data. After the time of $t_{SETTLING}$, the output can be valid, and the length of this period of time will vary depending on the supply voltage and the analog output load.

The digital input is linearly converted to analog voltage output by the DAC, and it ranges from 0 to V_{DDA} . The output voltage on any DAC channel pin shall meet the following relationship:

DAC output voltage = V_{DDA}* (DAC_DORx/ 4096).

17.2.4 DAC triangular-wave generator

The module has a built-in triangular-wave generator, which can add a small amplitude triangle-wave to the reference signal. Set WAVEx[1:0] bit as '10' and select the triangular-wave generation function of DAC. Set

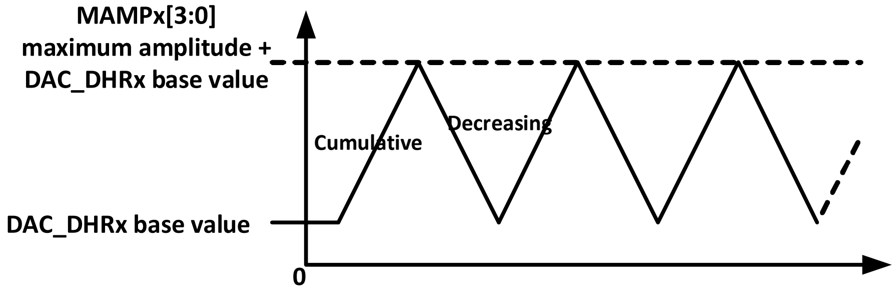
MAMPx[3:0] bit in the DAC_CTLR register to select the amplitude of triangular-wave.

The system contains a triangular-wave counter starting from 0, which accumulates by 1 in 3 APB1 clock cycles after each trigger event. The value of the counter is added to the value of the DAC_DHRx register and the overflow bit is discarded and then written to the DAC_DORx register. When the value transmitted into the DAC_DORx register is smaller than the maximum amplitude defined by the MAMPx[3:0] bits, the triangular-wave counter will gradually accumulate. Once it reaches the set maximum amplitude, the counter will begin to decrease progressively, and then start to accumulate after reaching 0. Repeat this cycle. Set WAVEx[1:0] bits to '00' to reset the generation of triangle-waves.

Note: 1. To generate a triangular-wave, DAC trigger must be enabled, i.e., setting the TENx bit in the DAC_CTLR register to 1.

2. MAMPx[3:0] bits must be set before enable the DAC. Otherwise, its value cannot be modified.

Figure 17-4 Triangular wave generation



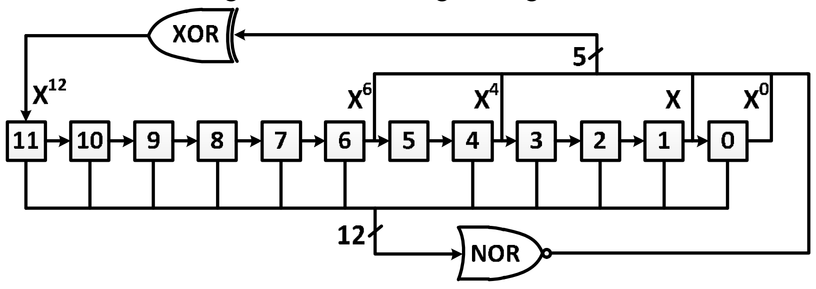
17.2.5 DAC noise-wave generator

The module has a built-in noise-generator that uses the Linear Feedback Shift Register (LFSR) to generate pseudo noise with varying amplitude. Set WAVE[1:0] bits to '01' to select the DAC noise generation function. Set the MAMPx[3:0] bits in the DAC_CTLR register to select the data of the masked part of the LFSR.

The preload value of the register LFSR is 0xAAA. According to a specific algorithm, the value of this register is updated in 3 APB1 clock cycles after each trigger event. Setting the MAMPx[3:0] bits in the DAC_CR register can mask part or all of the LFSR data, so that the LSFR value obtained is added to the value of DAC_DHRx, and the overflow bit is removed and then written into the DAC_DORx register. If the register LFSR value is 0x000, it will inject '1' (anti-lock mechanism). Set WAVEx[1:0] bit to '00' to reset the generation algorithm of LFSR waveform.

Note: To generate a noise-wave, DAC trigger must be enabled, i.e., setting the TENx bit in the DAC_CTLR register to 1.

Figure 17-5 LFSR register algorithm



17.3 Dual DAC conversion

When the 2 DAC channels are required at the same time, the module integrates 3 data registers in dual DAC mode (DAC_RD8BDHR, DAC_LD12BDHR, DAC_RD12BDHR), in order to efficiently operate the DAC module. The conversion value of the 2 DACs can be updated when only one of the 3 registers is operated.

For dual DAC conversion, other registers of the module can be used to implement 11 types of conversion modes with different combinations. The value of data to be converted in the 2 channels needs to be written to one of the 3 data registers.

17.3.1 Independent trigger with the same LFSR generation

Set the TENx bits. TSELx can be set as different values. WAVEx is set to 0b01. MAMPx is set as the same LFSR mask value. When a channel1 trigger event occurs, add the counter (LFSR1) with the same mask value to the channel1 data register (DAC_DHR1), and the sum is transferred to DAC_DOR1 after delay of 3 APB1 clocks, for conversion. Then the LFSR1 counter is updated. When a channel2 trigger event occurs, add the counter (LFSR2) with the same mask value to the channel2 data register (DAC_DHR2), and the sum is transferred to DAC_DOR2 for conversion after delay of 3 APB1 clocks. Then the LFSR2 counter is updated.

17.3.2 Independent trigger with different LFSR generation

Set the TENx bits. TSELx can be set as different values. WAVEx is set to 0b01. MAMPx can be set as different mask values of LFSR. When a channel1 trigger event occurs, add the LFSR1 counter, with the mask value configured by the MAMP1[3:0] bits, to the channel1 data register (DAC_DHR1), and the sum is transferred to DAC_DOR1 for conversion after delay of 3 APB1 clocks. Then the LFSR1 counter is updated. When a channel2 trigger event occurs, add the LFSR2 counter, with the mask value configured by the MAMP2[3:0] bits, to the channel2 data register (DAC_DHR2), and the sum is transferred to DAC_DOR2 for conversion after delay of 3 APB1 clocks. Then the LFSR2 counter is updated.

17.3.3 Independent trigger with the same triangle generation

Set the TENx bits. TSELx can be set as different values. WAVEx is set to 0b1x. MAMPx is set as the same triangle amplitude value. When a channel1 trigger event occurs, add the channel1 triangle counter, with the same triangle amplitude, to the channel1 data register (DAC_DHR1), and the sum is transferred to DAC_DOR1 for conversion after delay of 3 APB1 clocks. Then the channel1 triangle counter is updated. When a channel2 trigger event occurs, add the channel2 triangle counter, with the same triangle amplitude, to the channel2 data register (DAC_DHR2), and the sum is transferred to DAC_DOR2 for conversion after delay of 3 APB1 clocks. Then the channel2 triangle counter is updated.

17.3.4 Independent trigger with different triangle generation

Set the TENx bits. TSELx can be set as different values. WAVEx is set to 0b1x. MAMPx can be set as different triangle amplitude values. When a channel1 trigger event occurs, add the channel1 triangle counter, with a triangle amplitude configured by MAMP1, to the channel1 data register (DAC_DHR1), and the sum is transferred to DAC_DOR1 for conversion after delay of 3 APB1 clocks. Then the channel1 triangle counter is updated. When a channel2 trigger event occurs, add the channel2 triangle counter, with a triangle amplitude configured by MAMP2, to the channel2 data register (DAC_DHR2), and the sum is transferred to DAC_DOR2 for conversion after delay of 3 APB1 clocks. Then the channel2 triangle counter is updated.

17.3.5 Independent trigger with no wave generation

Set the TENx bits. TSELx can be set as different values to select different trigger sources. When a channel1 trigger event occurs, the value of the channel1 data register (DAC_DHR1) delayed 3 APB1 clocks is transferred to DAC_DOR1 for conversion. When a channel2 trigger event occurs, the value of the channel2 data register (DAC_DHR2) delayed 3 APB1 clocks is transferred to DAC_DOR2 for conversion.

17.3.6 Simultaneous software trigger

In this conversion mode, write the dual DAC channel data to the desired register. An APB1 clock cycle later, the data in DAC_DHR1 and DAC_DHR2 is respectively transferred to DAC_DOR1 and DAC_DOR2 for conversion.

17.3.7 Simultaneous trigger with the same LFSR generation

Set the TENx bits. TSELx is set as the same value. WAVEx is set to 0b01. MAMPx is set as the same LFSR mask value. When a trigger event occurs, add the LFDR1 counter, with the same mask value, to the DAC_DHR1 register, and the sum is transferred to DAC_DOR1 after delay of 3 APB1 clocks for conversion. Then the LFSR1 counter is updated. At the same time, add the LFDR2 counter, with the same mask value, to the DAC_DHR2 register, and the sum is transferred to DAC_DOR2 after delay of 3 APB1 clocks for conversion. Then the LFSR2 counter is updated.

17.3.8 Simultaneous trigger with different LFSR generation

Set the TENx bits. TSELx is set as the same value. WAVEx is set to 0b01. MAMPx can be set as different LFSR mask values. When a trigger event occurs, add the LFSR1 counter, with different mask values, to the DAC_DHR1 register, and the sum is transferred to DAC_DOR1 after delay of 3 APB1 clocks for conversion. Then the LFSR1 counter is updated. At the same time, add the LFSR2 counter, with different mask values, to the DAC_DHR2 register, and the sum is transferred to DAC_DOR2 after delay of 3 APB1 clocks for conversion. Then the LFSR2 counter is updated.

17.3.9 Simultaneous trigger with the same triangle generation

Set the TENx bits. TSELx is set as the same value. WAVEx is set to 0b1x. MAMPx is set as the same triangle amplitude value. When a trigger event occurs, add the counter, with the same triangle amplitude value, to the DAC_DHR1 register, and the sum is transferred to DAC_DOR1 after delay of 3 APB1 clocks for conversion. Then the channel1 triangle counter is updated. At the same time, add the counter, with the same triangle amplitude value, to the DAC_DHR2 register, and the sum is transferred to DAC_DOR2 after delay of 3 APB1 clocks for conversion. Then the channel2 triangle counter is updated.

17.3.10 Simultaneous trigger with different triangle generation

Set the TENx bits. TSELx is set as the same value. WAVEx is set to 0b1x. MAMPx can be set as different triangle amplitude values. When a trigger event occurs, add the counter, with a triangle amplitude configured by MAMP1[3:0], to the DAC_DHR1 register, and the sum is transferred to DAC_DOR1 after delay of 3 APB1 clocks for conversion. Then the channel1 triangle counter is updated. At the same time, add the counter, with a triangle amplitude configured by MAMP2[3:0], to the DAC_DHR2 register, and the sum is transferred to DAC_DOR2 after delay of 3 APB1 clocks for conversion. Then the channel2 triangle counter is updated.

17.3.11 Simultaneous trigger with no wave generation

Set the TENx bits. TSELx is set as the same value. In this conversion mode, when a trigger event occurs, the values of the DAC_DHR1 register and the DAC_DHR2 register delayed 3 APB1 clocks are respectively transferred to DAC_DOR1 and DAC_DOR2 for DAC conversion.

17.4 Register description

Table 17-2 DAC registers

Name	Access address	Description	Reset value
R32_DAC_CTLR	0x40007400	DAC control register	0x00000000
R32_DAC_SWTR	0x40007404	DAC software trigger register	0x00000000
R32_DAC_R12BDHR1	0x40007408	DAC channel1 right-aligned 12-bit data hold register	0x00000000
R32_DAC_L12BDHR1	0x4000740C	DAC channel1 left-aligned 12-bit data hold register	0x00000000
R32_DAC_R8BDHR1	0x40007410	DAC channel1 right-aligned 8-bit data hold register	0x00000000
R32_DAC_R12BDHR2	0x40007414	DAC channel2 right-aligned 12-bit data hold register	0x00000000
R32_DAC_L12BDHR2	0x40007418	DAC channel2 left-aligned 12-bit data hold register	0x00000000
R32_DAC_R8BDHR2	0x4000741C	DAC channel2 right-aligned 8-bit data hold register	0x00000000
R32_DAC_RD12BDHR	0x40007420	Dual channel right-aligned 12-bit data hold register	0x00000000
R32_DAC_LD12BDHR	0x40007424	Dual channel left-aligned 12-bit data hold register	0x00000000
R32_DAC_RD8BDHR	0x40007428	Dual channel right-aligned 8-bit data hold register	0x00000000
R32_DAC_DOR1	0x4000742C	DAC channel1 data output register	0x00000000
R32_DAC_DOR2	0x40007430	DAC channel2 data output register	0x00000000

17.4.1 DAC Control Register (DAC_CTLR)

Offset address: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			DMAEN2	MAMP2[3:0]				WAVE2[2:0]		TSEL2[2:0]			TEN2	BOFF2	EN2
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			DMAEN1	MAMP1[3:0]				WAVE1[2:0]		TSEL1[2:0]			TEN1	BOFF1	EN1

Bit	Name	Access	Description	Reset value

[31:29]	Reserved	RO	Reserved.	0
28	DMAEN2	RW	DMA enable of DAC channel2: 1: DMA of DAC channel2 enabled; 0: DMA of DAC channel2 disabled.	0
[27:24]	MAMP2 [3:0]	RW	DAC channel2 mask/amplitude setting. The software sets this area to select the LFSR data mask bit in the noise generation mode, and select the waveform amplitude in the triangle waveform generation mode: 0000: Unmask bit0 of LFSR/Triangle amplitude equal to 1; 0001: Unmask bits[1:0] of LFSR / Triangle amplitude equal to 3; 0010: Unmask bits[2:0] of LFSR / Triangle amplitude equal to 7; 0011: Unmask bits[3:0] of LFSR / Triangle amplitude equal to 15; 0100: Unmask bits[4:0] of LFSR / Triangle amplitude equal to 31; 0101: Unmask bits[5:0] of LFSR / Triangle amplitude equal to 63; 0110: Unmask bits[6:0] of LFSR / Triangle amplitude equal to 127; 0111: Unmask bits[7:0] of LFSR / Triangle amplitude equal to 255; 1000: Unmask bits[8:0] of LFSR / Triangle amplitude equal to 511; 1001: Unmask bits[9:0] of LFSR / Triangle amplitude equal to 1023; 1010: Unmask bits[10:0] of LFSR / Triangle amplitude equal to 2047; ≥1011: Unmask bits[11:0] of LFSR / Triangle amplitude equal to 4095.	0

[23:22]	WAVE2 [1:0]	RW	Noise/triangular wave generation enable of DAC channel2
			00: Wave generator disabled;
			01: Noise-wave generator enabled;
			1x: Triangular-wave generator enabled.
				0

[21:19]	TSEL2[2:0]	RW	Trigger event selection of DAC channel2:
			000: TIM6 TRGO event;
			001: TIM8 TRGO event;
			010: TIM7 TRGO event;
			011: TIM5 TRGO event;
			100: TIM2 TRGO event;
			101: TIM4 TRGO event;
			110: External interrupt line9;
			111: Software trigger;
			Others: Reserved.
				0

18	TEN2	RW	External trigger mode enable of DAC channel2: 1: Trigger of DAC channel2 enabled. The data written into the DAC_xDHR register is sent to the DAC_DOR2 register in 3 APB1 clock cycles. 0: Trigger of DAC channel2 disabled. The data written into the DAC_xDHR register is sent to the DAC_DOR2 register in 1	0

			APB1 clock cycle. *Note: If software trigger is selected, the data in DAC_xDHR only needs to be sent to the DAC_DOR2 register in an APB1 clock cycle.*	
17	BOFF2	RW	DAC channel2 output buffer disable control (recommended to be enabled): 1: DAC channel2 output buffer disabled; 0: DAC channel2 output buffer enabled.	0
16	EN2	RW	DAC channel2 enable: 1: DAC channel2 enabled; 0: DAC channel2 disabled.	0
[15:13]	Reserved	RO	Reserved.	0
12	DMAEN1	RW	DMA enable of DAC channel1: 1: DMA function of DAC channel1 enabled; 0: DMA function of DAC channel1 disabled.	0
[11:8]	MAMP1 [3:0]	RW	DAC channel 1 mask/amplitude setting. The software sets these bits to select the LFSR data mask bit in the noise generation mode, and select the wave amplitude in the triangle waveform generation mode: 0000: Unmask bit0 of LFSR / Triangle amplitude equal to 1; 0001: Unmask bits[1:0] of LFSR / Triangle amplitude equal to 3; 0010: Unmask bits[2:0] of LFSR / Triangle amplitude equal to 7; 0011: Unmask bits[3:0] of LFSR / Triangle amplitude equal to 15; 0100: Unmask bits[4:0] of LFSR / Triangle amplitude equal to 31; 0101: Unmask bits[5:0] of LFSR / Triangle amplitude equal to 63; 0110: Unmask bits[6:0] of LFSR / Triangle amplitude equal to 127; 0111: Unmask bits[7:0] of LFSR / Triangle amplitude equal to 255; 1000: Unmask bits[8:0] of LFSR / Triangle amplitude equal to 511; 1001: Unmask bits[9:0] of LFSR / Triangle amplitude equal to 1023; 1010: Unmask bits[10:0] of LFSR / Triangle amplitude equal to 2047; ≥1011: Unmask bits[11:0] of LFSR / Triangle amplitude equal to 4095.	0
[7:6]	WAVE1 [1:0]	RW	Noise/triangular-wave generation enable of DAC channel1. 00: Wave generator disabled; 01:Noise wave generator enabled; 1x: Triangular wave generator enabled.	0
[5:3]	TSEL1 [2:0]	RW	Trigger event selection of DAC channel1: 000: TIM6 TRGO event;	0

			001: TIM8 TRGO event;
			010: TIM7 TRGO event;
			011: TIM5 TRGO event;
			100: TIM2 TRGO event;
			101: TIM4 TRGO event;
			110: External interrupt line9;
			111: Software trigger;
			Others: Reserved.	

2	TEN1	RW	External trigger mode enable of DAC channel1: 1: Trigger of DAC channel1 enabled. The data written into the DAC_xDHR register is sent to the DAC_DOR1 register in 3 APB1 clock cycles. 0: Trigger of DAC channel1 disabled. The data written into the DAC_xDHR register is sent to the DAC_DOR1 register in 1 APB1 clock cycle. *Note: If software trigger is selected, the data in DAC_xDHR only needs to be sent to the DAC_DOR1 register in an APB1 clock cycle.*	0
1	BOFF1	RW	DAC channel1 output buffer disable control (recommended to be enabled): 1: DAC channel1 output buffer disabled; 0: DAC channel1 output buffer enabled.	0
0	EN1	RW	DAC channel1 enable: 1: DAC channel1 enabled; 0: DAC channel1 disabled.	0

Note: The configuration register includes the configuration of channel1 and channel2. When the outputs of the 2 channels are enabled at the same time (ENx bit is '1'), the same wave is output to 2 hardware channels according to the configuration of channel1. When only the output of channel1 is enabled, the wave is output to the hardware channel1 according to the configuration of channel1, and channel2 does not output. When only the output of channel2 is enabled, the wave is output to the hardware2 according to the configuration of channel2, and channel 1 does not output.

17.4.2 DAC Software Trigger Register (DAC_SWTR)

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														SWTRIG2	SWTRIG1

Bit	Name	Access	Description	Reset value
[31:2]	Reserved	RO	Reserved.	0
1	SWTRIG2	WO	Software trigger control bit of DAC channel2: 1: Software trigger of DAC channel2 enabled; 0: Software trigger of DAC channel2 disabled.	0

			Note: Once the data in DAC_xDHR (after 1 APB1 clock cycle) is sent to the DAC_DOR2 register, this bit can be cleared by hardware.	
0	SWTRIG1	WO	Software trigger control bit of DAC channel1: 1: Software trigger of DAC channel1 enabled; 0: Software trigger of DAC channel1 disabled. *Note: Once the data in DAC_xDHR (after 1 APB1 clock cycle) is sent to the DAC_DOR1 register, this bit can be cleared by hardware.*	0

17.4.3 DAC Channel 1 Right-aligned 12-bit Data Hold Register (DAC_R12BDHR1)

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				DACC1DHR[11:0]											

Bit	Name	Access	Description	Reset value
[31:12]	Reserved	RO	Reserved.	0
[11:0]	DACC1DHR[11:0]	RW	12-bit right-aligned data of DAC channel1.	0

17.4.4 DAC Channel 1 Left-aligned 12-bit Data Hold Register (DAC_L12BDHR1)

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
DACC1DHR[11:0]												Reserved			

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved.	0
[15:4]	DACC1DHR[11:0]	RW	12-bit left-aligned data of DAC channel1.	0
[3:0]	Reserved	RO	Reserved.	0

17.4.5 DAC Channel 1 Right-aligned 8-bit Data Hold Register (DAC_R8BDHR1)

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								DACC1DHR[7:0]							

Bit	Name	Access	Description	Reset value
[31:8]	Reserved	RO	Reserved	0
[7:0]	DACC1DHR[7:0]	RW	8-bit right-aligned data of DAC channel 1.	0

17.4.6 DAC Channel 2 Right-aligned 12-bit Data Hold Register (DAC_R12BDHR2)

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**
Reserved				DACC2DHR[11:0]											

Bit	Name	Access	Description	Reset value
[31:12]	Reserved	RO	Reserved	0
[11:0]	DACC2DHR[11:0]	RW	12-bit right-aligned data of DAC channel 2.	0

17.4.7 DAC Channel 2 Left-aligned 12-bit Data Hold Register (DAC_L12BDHR2)

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**
DACC2DHR[11:0]												Reserved			

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:4]	DACC2DHR[11:0]	RW	12-bit left-aligned data of DAC channel2.	0
[3:0]	Reserved	RO	Reserved	0

17.4.8 DAC Channel 2 Right-aligned 8-bit Data Hold Register (DAC_R8BDHR2)

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								DACC2DHR[7:0]							

Bit	Name	Access	Description	Reset value
[31:8]	Reserved	RO	Reserved	0
[7:0]	DACC2DHR[7:0]	RW	8-bit right-aligned data of DAC channel2.	0

17.4.9 DAC Dual Channel Right-aligned 12-bit Data Hold Register (DAC_RD12BDHR)

Offset address: 0x20

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				DACC2DHR[11:0]											
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				DACC1DHR[11:0]											

Bit	Name	Access	Description	Reset value
[31:28]	Reserved	RO	Reserved	0
[27:16]	DACC2DHR[11:0]	RW	12-bit right-aligned data of DAC channel 2.	0
[15:12]	Reserved	RO	Reserved	0
[11:0]	DACC1DHR[11:0]	RW	12-bit right-aligned data of DAC channel 1.	0

17.4.10 DAC Dual Channel Left-aligned 12-bit Data Hold Register (DAC_LD12BDHR)

Offset address: 0x24

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DACC2DHR[11:0]												Reserved			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DACC1DHR[11:0]												Reserved			

Bit	Name	Access	Description	Reset value
[31:20]	DACC2DHR[11:0]	RW	12-bit left-aligned data of DAC channel 2.	0
[19:16]	Reserved	RO	Reserved	0
[15:4]	DACC1DHR[11:0]	RW	12-bit left-aligned data of DAC channel 1.	0
[3:0]	Reserved	RO	Reserved	0

17.4.11 DAC Dual Channel Right-aligned 8-bit Data Hold Register (DAC_RD8BDHR)

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**
DACC2DHR[7:0]								DACC1DHR[7:0]							

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:8]	DACC2DHR[7:0]	RW	8-bit right-aligned data of DAC channel 2.	0
[7:0]	DACC1DHR[7:0]	RW	8-bit right-aligned data of DAC channel 1.	0

17.4.12 DAC Channel 1 Data Output Register (DAC_DOR1)

Offset address: 0x2C

31	30	29	28	27	26	25	24	23	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				DACC1DOR[11:0]											

Bit	Name	Access	Description	Reset value
[31:12]	Reserved	RO	Reserved.	0
[11:0]	DACC1DOR[11:0]	RO	Output data of DAC channel 1	0

17.4.13 DAC Channel 2 Data Output Register (DAC_DOR2)

Offset address: 0x30

31	30	29	28	27	26	25	24	23	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				DACC2DOR[11:0]											

Bit	Name	Access	Description	Reset value
[31:12]	Reserved	RO	Reserved.	0
[11:0]	DACC2DOR[11:0]	RO	Output data of DAC channel 2	0

Chapter 18 Universal Synchronous Asynchronous Receiver Transmitter (USART)

This chapter applies to the whole family of CH32F2x, CH32V2x and CH32V3x.

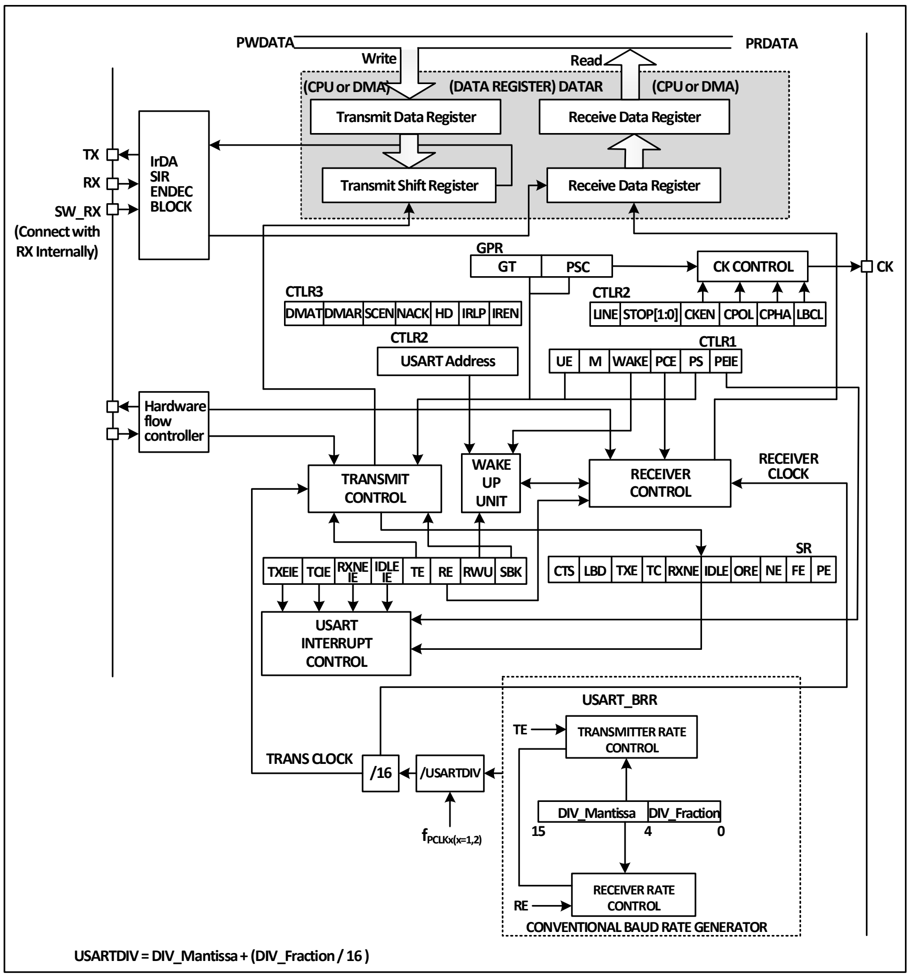
This module contains 3 USARTs (USART1/2/3) and 5 UARTs (UART4/5/6/7/8).
Note: For CH32V20x_D6 and CH32F20x_D6, the serial port 4 is a universal synchronous asynchronous receiver transmitter (USART4).

18.1 Main features

- Full-duplex or half-duplex synchronous or asynchronous communication
- NRZ data format
- Fractional baud rate generator, up to 9 Mbps
- Programmable data word length
- Configurable stop bits
- Supports LIN, IrDA encoder, smart card
- Supports DMA
- Multiple interrupt sources

18.2 Overview

Figure 18-1 USART block diagram



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.

18.3 Baud rate generator

The baud rate of the transceiver = F_{CLK}/(16*USARTDIV); F_{CLK} is the clock of APBx, i.e., PCLK1 or PCLK2, PCLK2 is used for the USART1 module, and PCLK1 shall be used for the rest. The value of USARTDIV is determined according to the 2 domains: DIV_M and DIV_F in USART_BRR. The specific calculation formula is:

$$\text{USARTDIV} = \text{DIV\_M}+(\text{DIV\_F}/16)$$

It shall be noted that the bit rate generated by the baud rate generator may not be exactly 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 APBx clock. For example, when the baud rate is set to 115200bps, the value of USARTDIV will be set to 39.0625, and the baud rate of 115200bps can be obtained at the highest frequency, but if you need a baud rate of 921600bps, the calculated USARTDIV will be 4.88, but the actual closest value filled in USART_BRR can only be 4.875. The actual baud rate is 923076bps, with 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%.

18.4 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 register2 (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:

1) 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;

2) The clock signal is outputted only when TX pin outputs data;

3) 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 register2 (R16_USARTx_CTLR2). These 3 bits need to be set when TE and RE are not enabled. The specific difference is shown in Figure 18-2.

4) 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 18-3.

Figure 18-2 Example of USART clock timing (M=0)

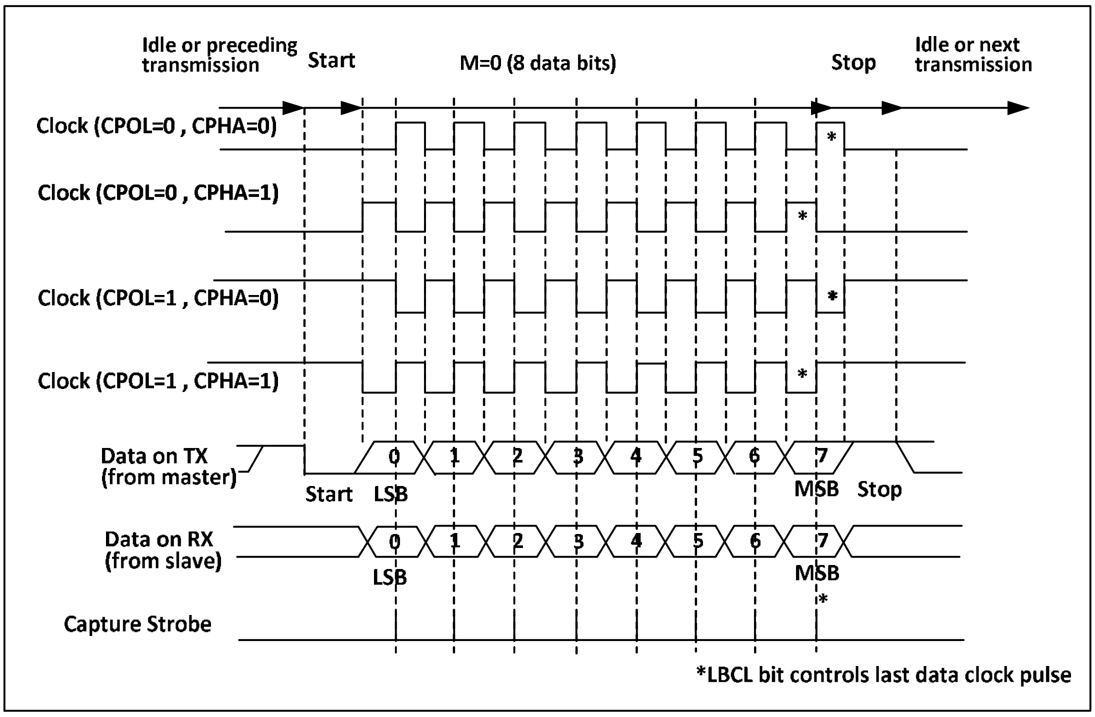
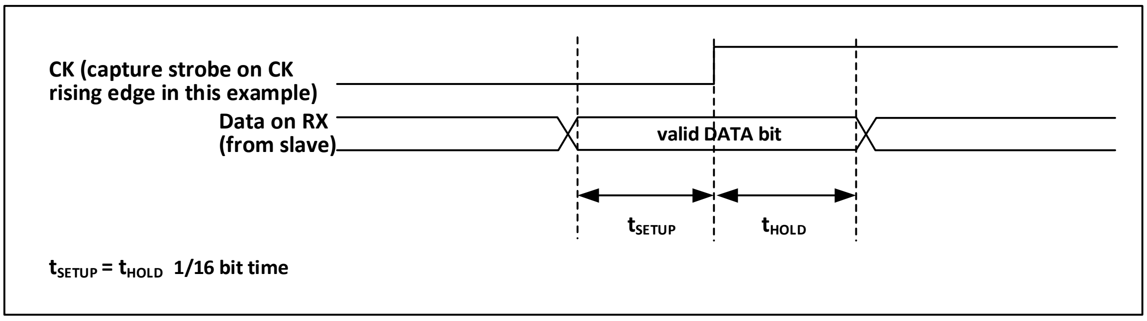


Figure 18-3 Data sample hold time


CK (capture strobe on CK rising edge in this example)
Data on RX (from slave)
valid DATA bit
tSETUP
tHOLD
tSETUP = tHOLD 1/16 bit time

18.5 Single-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 register3 (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 register2 and the control register3 (R16_USARTx_CTLR2 and R16_USARTx_CTLR3).

After setting to half-duplex mode, it is needed to set the TX IO port to floating input or 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.

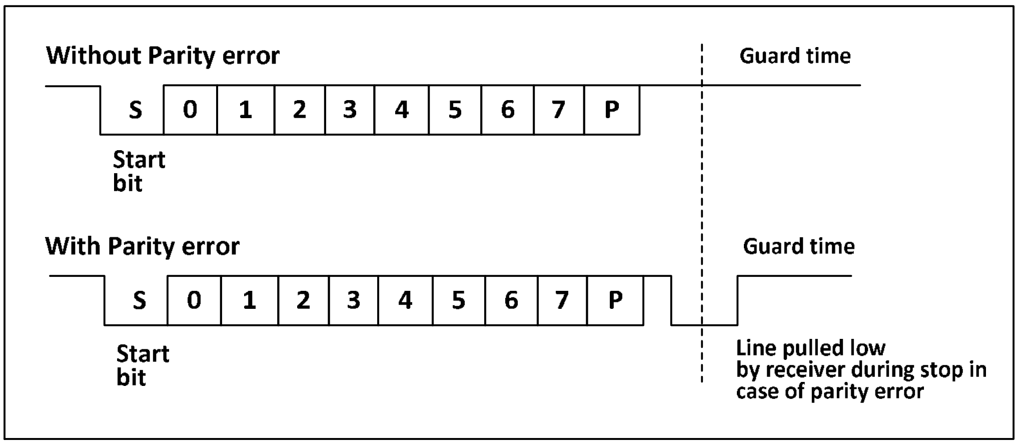
18.6 Smartcard

The smartcard mode supports ISO7816-3 protocol to access the smart card controller.

To enable smartcard mode, set the SCEN bit in the control register3 (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 register2 and the control register3 (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 single-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 17-4 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 18-4 (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 APB 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).

18.7 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.

18.8 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.

18.9 Interrupt

The USART module supports multiple interrupt sources, including transmit data register empty (TXE), CTS, transmission complete (TC), received data ready (TXNE), 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 18-1 Interrupts and the corresponding enable bits

Interrupt source	Enable bit
Transmit data register empty (TXE)	TXEIE
Transmission allowed (CTS)	CTSIE
Transmission complete (TC)	TCIE
Received data ready (TXNE)	TXNEIE
Overrun error detected (ORE)	
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	

18.10 Register description

Table 18-2 USART1 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 register1	0x00000000
R32_USART1_CTLR2	0x40013810	UASRT1 control register2	0x00000000
R32_USART1_CTLR3	0x40013814	UASRT1 control register3	0x00000000
R32_USART1_GPR	0x40013818	UASRT1 guard time and prescaler register	0x00000000

Table 18-3 USART2 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 register1	0x00000000
R32_USART2_CTLR2	0x40004410	UASRT2 control register2	0x00000000
R32_USART2_CTLR3	0x40004414	UASRT2 control register3	0x00000000
R32_USART2_GPR	0x40004418	UASRT2 guard time and prescaler register	0x00000000

Table 18-4 USART3 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 register1	0x00000000
R32_USART3_CTLR2	0x40004810	UASRT3 control register2	0x00000000
R32_USART3_CTLR3	0x40004814	UASRT3 control register3	0x00000000
R32_USART3_GPR	0x40004818	UASRT3 guard time and prescaler register	0x00000000

Table 18-5 UART4 registers

Name	Access address	Description	Reset value
R32_UART4_STATR	0x40004C00	UART4 status register	0x000000C0
R32_UART4_DATAR	0x40004C04	UART4 data register	0x000000XX
R32_UART4_BRR	0x40004C08	UART4 baud rate register	0x00000000
R32_UART4_CTLR1	0x40004C0C	UART4 control register1	0x00000000
R32_UART4_CTLR2	0x40004C10	UART4 control register2	0x00000000
R32_UART4_CTLR3	0x40004C14	UART4 control register3	0x00000000
R32_UART4_GPR	0x40004C18	UART4 guard time and prescaler register	0x00000000

Table 18-6 UART5 registers

Name	Access address	Description	Reset value
R32_UART5_STATR	0x40005000	UART5 status register	0x000000C0
R32_UART5_DATAR	0x40005004	UART5 data register	0x000000XX
R32_UART5_BRR	0x40005008	UART5 baud rate register	0x00000000
R32_UART5_CTLR1	0x4000500C	UART5 control register1	0x00000000
R32_UART5_CTLR2	0x40005010	UART5 control register2	0x00000000
R32_UART5_CTLR3	0x40005014	UART5 control register3	0x00000000
R32_UART5_GPR	0x40005018	UART5 guard time and prescaler register	0x00000000

Table 18-7 UART6 registers

Name	Access address	Description	Reset value
R32_UART6_STATR	0x40001800	UART6 status register	0x000000C0
R32_UART6_DATAR	0x40001804	UART6 data register	0x000000XX

R32_UART6_BRR	0x40001808	UART6 baud rate register	0x00000000
R32_UART6_CTLR1	0x4000180C	UART6 control register1	0x00000000
R32_UART6_CTLR2	0x40001810	UART6 control register2	0x00000000
R32_UART6_CTLR3	0x40001814	UART6 control register3	0x00000000
R32_UART6_GPR	0x40001818	UART6 guard time and prescaler register	0x00000000

Table 18-8 UART7 registers

Name	Access address	Description	Reset value
R32_UART7_STATR	0x40001C00	UART7 status register	0x000000C0
R32_UART7_DATAR	0x40001C04	UART7 data register	0x000000XX
R32_UART7_BRR	0x40001C08	UART7 baud rate register	0x00000000
R32_UART7_CTLR1	0x40001C0C	UART7 control register1	0x00000000
R32_UART7_CTLR2	0x40001C10	UART7 control register2	0x00000000
R32_UART7_CTLR3	0x40001C14	UART7 control register3	0x00000000
R32_UART7_GPR	0x40001C18	UART7 guard time and prescaler register	0x00000000

Table 18-9 UART8 registers

Name	Access address	Description	Reset value
R32_UART8_STATR	0x40002000	UART8 status register	0x000000C0
R32_UART8_DATAR	0x40002004	UART8 data register	0x000000XX
R32_UART8_BRR	0x40002008	UART8 baud rate register	0x00000000
R32_UART8_CTLR1	0x4000200C	UART8 control register1	0x00000000
R32_UART8_CTLR2	0x40002010	UART8 control register2	0x00000000
R32_UART8_CTLR3	0x40002014	UART8 control register3	0x00000000
R32_UART8_GPR	0x40002018	UART8 guard time and prescaler register	0x00000000

18.10.1 USART Status Register (USARTx_STATR) (x=1/2/3/4/5/6/7/8)

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						CTS	LBD	TXE	TC	RXNE	IDLE	ORE	NE	FE	PE

Bit	Name	Access	Description	Reset value
[31:10]	Reserved	RO	Reserved.	0
9	CTS	RW0	CTS status change flag. If the CTSE bit is set, this bit will be set high by hardware when the nCTS output status changes. It is cleared by the software. If the CTSIE bit has been set, an interrupt is generated. 1: A change on the nCTS status line; 0: No change on the nCTS status line.	0
8	LBD	RW0	LIN break detection flag. When LIN disconnection is detected, this bit will be set by hardware. It is cleared by the software. If the	0

			LBDIE bit has been set, an interrupt will be generated. 1: LIN disconnection detected; 0: No LIN disconnection detected.	
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 to clear this bit. 1: Transmission completed; 0: Transmission not completed.	1
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

			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.*	
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 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
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

18.10.2 USART Data Register (USARTx_DATAR) (x=1/2/3/4/5/6/7/8)

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

18.10.3 USART Baud Rate Register (USARTx_BRR) (x=1/2/3/4/5/6/7/8)

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
DIV_Mantissa[11:0]												DIV_Fraction[3:0]			

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved.	0
[15:4]	DIV_ Mantissa	RW	These 12 bits define the mantissa of the USART divider.	0
[3:0]	DIV_ Fraction	RW	These 4 bits define the fraction of the USART divider.	0

18.10.4 USART Control Register1 (USARTx_CTLR1) (x=1/2/3/4/5/6/7/8)

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		UE	M	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	RWU	SBK

Bit	Name	Access	Description	Reset value
[31:14]	Reserved	R0	Reserved.	0
13	UE	RW	USART enable. When this bit is set, 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

			0: Idle line.	
10	PCE	RW	Parity control enable. 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. 0 means even parity, and 1 means odd 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. When this bit is set, the parity check error interrupt is allowed to be generated.	0
7	TXEIE	RW	Transmit buffer empty interrupt enable. When this bit is set, the transmit buffer empty interrupt is allowed to be generated.	0
6	TCIE	RW	Transmission completion interrupt enable. When this bit is set, the transmission complete interrupt is allowed to be generated.	0
5	RXNEIE	RW	Receive buffer non-empty interrupt enable. When this bit is set, the receive buffer not empty interrupt is allowed to be generated.	0
4	IDLEIE	RW	Idle line interrupt enable. When this bit is set, the idle line interrupt is allowed to be generated.	0
3	TE	RW	Transmitter enable. When this bit is set, the transmitter is enabled.	0
2	RE	RW	Receiver enable. When this bit is set, the receiver is enabled, and the receiver starts detecting the start bit on the RX pin.	0
1	RWU	RW	Receiver 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 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

18.10.5 USART Control Register2 (USARTx_CTLR2) (x=1/2/3/4/5/6/7/8)

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	R0	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.
				0

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 SLCK 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 SLCK 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 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.*	0
7	Reserved	RW	Reserved.	0
6	LBDIE	RW	LIN break character detection interrupt enable. This bit can enable the interrupt caused by LBD;	0
5	LBDL	RW	LIN break character 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

18.10.6 USART Control Register 3 (USARTx_CTLR3) (x=1/2/3/4/5/6/7/8)

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
Reserved					CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HDSEL	IRLP	IREN	EIE

Bit	Name	Access	Description	Reset value
[31:11]	Reserved	R0	Reserved.	0
10	CTSIE	RW	CTSIE interrupt enable. When this bit is set, an interrupt is generated when CTS is set.	0
9	CTSE	RW	CTS enable. When this bit is set, the CTS flow control is enabled.	0
8	RTSE	RW	RTS enable. When this bit is set, the RTS flow control is enabled.	0
7	DMAT	RW	DMA transmission enable. When this bit is set, DMA mode is enabled for transmission.	0
6	DMAR	RW	DMA reception enable. When this bit is set, DMA mode is enabled for reception.	0

5	SCEN	RW	Smartcard mode enable. When this bit is set, smartcard mode is enabled.	0
4	NACK	RW	Smart card NACK enable. When this bit is set, NACK is transmitted when the check error occurs.	0
3	HDSEL	RW	Half-duplex mode selection. When this bit is set, half-duplex mode is selected.	0
2	IRLP	RW	Infrared low power selection. When this bit is set, low power mode is selected.	0
1	IREN	RW	Infrared enable. When this bit is set, infrared mode is enabled.	0
0	EIE	RW	Error enable interrupt. When this bit is set, and when DMAR is set, an interrupt is generated if the FE or ORE or NE bit is set.	0

18.10.7 USART Guard Time and Prescaler Register (USARTx_GPR) (x=1/2/3/4/5/6/7/8)

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	R0	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

Chapter 19 Inter-integrated Circuit (I2C) interface

This chapter applies to the whole family of CH32F2x, CH32V2x and CH32V3x.

The inter-integrated circuit (I2C) is widely used in the communication between microcontroller and sensor and other off-chip modules. It supports multi-master mode and multi-slave mode. The communication can be carried out at 2 speeds: 100 KHz (standard) and 400 KHz (fast) only through 2 wires (SDA and SCL). The I2C bus is also compatible with the SMBus protocol. It not only supports I2C timing, but also supports arbitration, timing and DMA, and has CRC check capability.

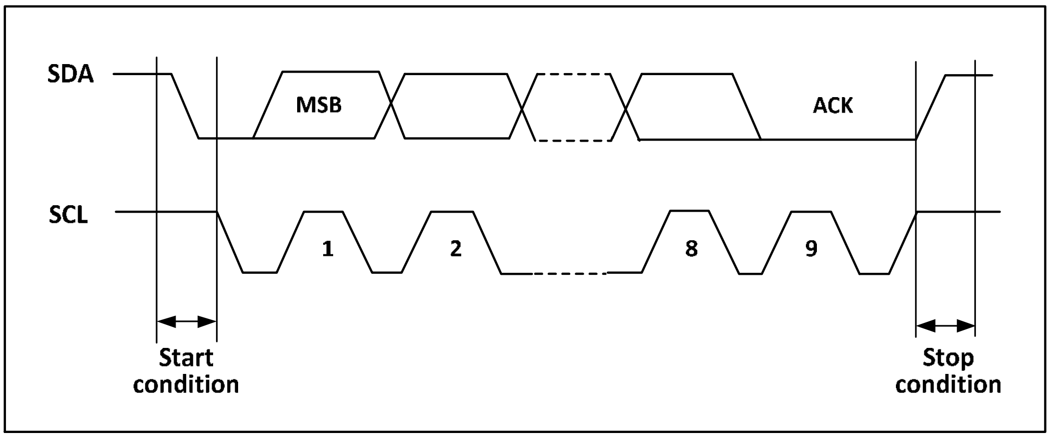
19.1 Main features

- 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
- Compatible with SMBus

19.2 Overview

I2C is a half-duplex bus, and it can only run in one of the following 4 modes at the same time: master transmitter, master receiver, slave transmitter and slave receiver. The I2C module works in the slave mode by default. After the start condition is generated, it automatically switches to master mode. When the arbitration is lost or the stop signal is generated, it switches to the slave mode. I2C module supports multi-host function. When working in master mode, the I2C module actively sends out data and addresses. Both data and address are transferred as 8-bit bytes, and MSB first. One-byte (in 7-bit address mode) or two-byte (in 10-bit address mode) address is located after the initial event. Every time the master device sends 8-bit data or address, the slave device needs to reply an ACK, i.e., pulling the SDA bus to be low, as shown in Figure 19-1.

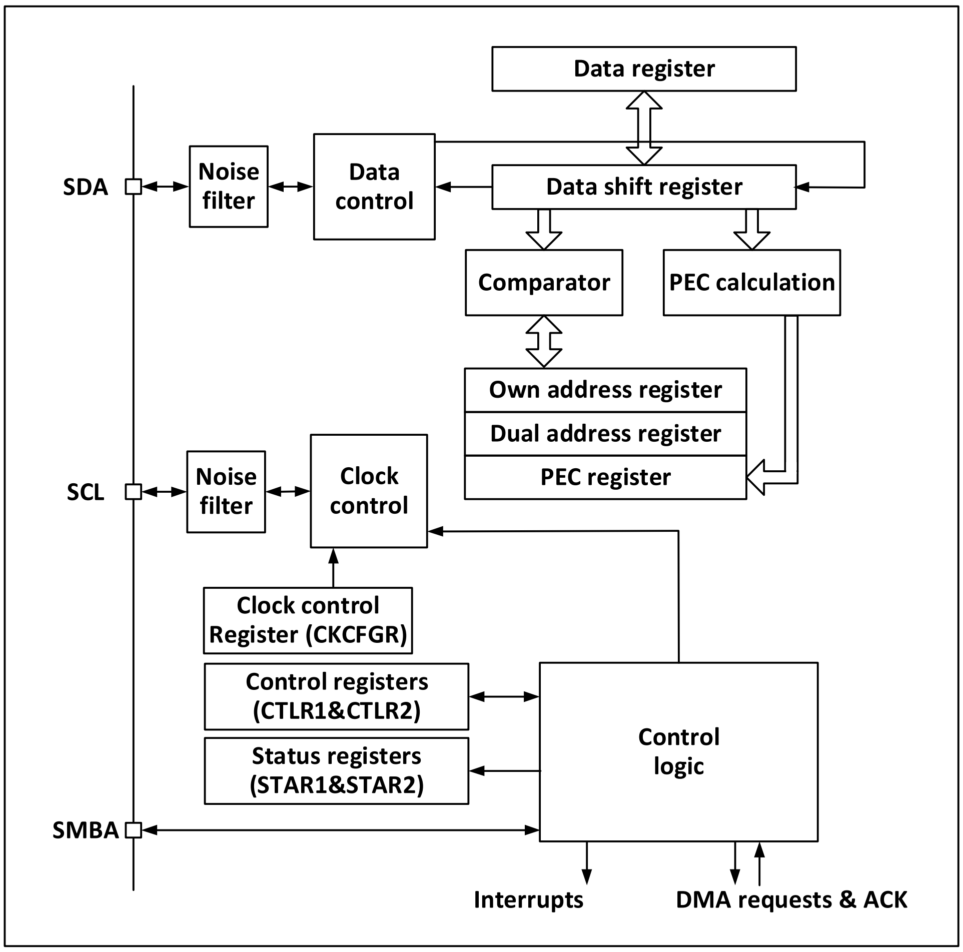
Figure 19-1 I2C timing diagram



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 19-2 shows the block diagram of I2C.

Figure 19-2 I2C block diagram

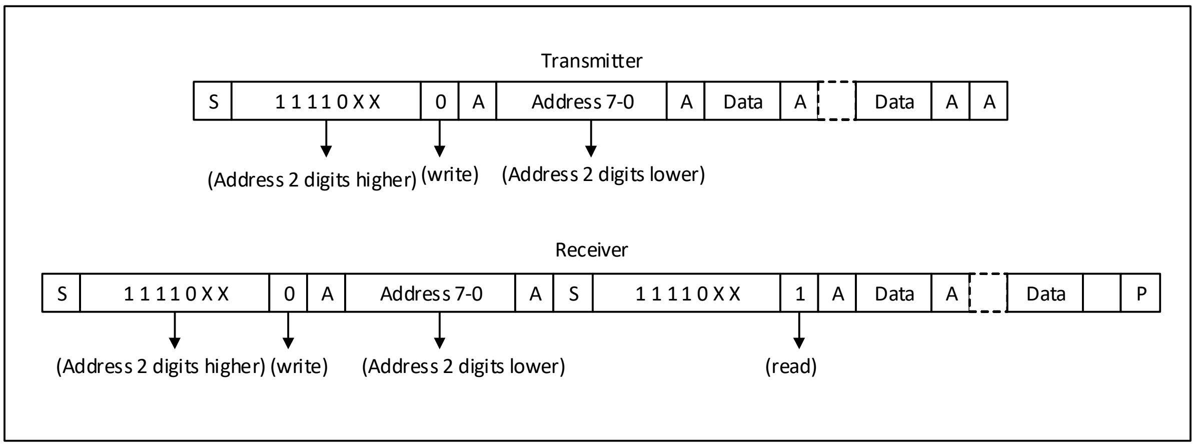


19.3 Master mode

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 register2 (R16_I2Cx_CTLR2) and the clock control register (R16_I2Cx_CKCFGR).
Set a proper rising edge in the rising edge register (R16_I2Cx_RTR).
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

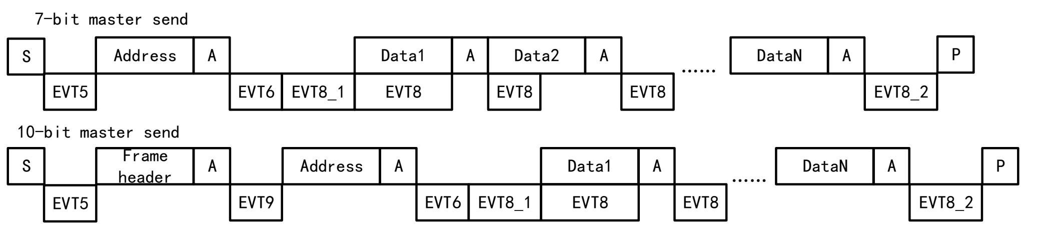
second address byte to the data register. Then, clear the ADD10 bit.
Then, write the data register to send the second address byte. After sending the second address byte, the ADDR bit in the status register is set. If the ITEVTEN bit has been set, an interrupt will be generated. Read the R16_I2Cx_STAR2 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 19-3, 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 19-3 Master receive/transmits data in 10-bit address mode



Master transmit mode:
The master device's internal shift register sends data from the data register to the SDA line. When the master device receives an ACK, TxE in status register 1 (R16_I2Cx_STAR1) is set, and an interrupt is also generated if ITEVTEN and ITBUFEN are set. Writing data to the data register will clear the TxE bit.
If the TxE bit is set and no new data was written to the data register before the last data was sent, then the BTF bit will be set and SCL will remain low until it is cleared, and writing data to the data register after reading R16_I2Cx_STAR1 will clear the BTF bit.

Figure 19-4 Master transmitter transmission sequence diagram



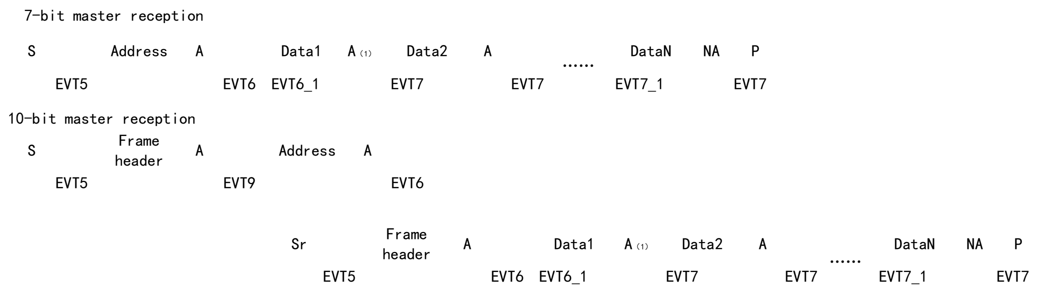
Description: S=Start (start condition), Sr=repeated start condition, P=Stop (stop condition), A=response, NA=non-response, EVTx=event (interrupt generated when ITEVFEN=1)

EVT5: SB=1, reading SR1 and then writing the address to the DR register will clear the event.
EVT6;ADDR=1,reading SR1 then reading SR2 will clear the event.
EVT8_1: TxE=1, shift register empty, data register empty, write DR register.
EVT8: TxE=1,shift register is not empty,data register is empty,writing DR register will clear the event.
EVT8_2: TxE=1, BTF=1, request to set the stop bit. the TxE and BTF bits are cleared by hardware when the stop condition is generated.
EVT9: ADDR10=1, reading SR1 and then writing to DR register will clear the event.

Note: 1: EVT5, EVT6, EVT9, EVT8_1 and EVT8_2 events elongate the SCL low until the end of the corresponding software sequence.
2: The software sequence of EVT8 must be completed before the end of the current byte transfer.

Master receive mode:
The I2C module will receive data from the SDA line and write it into the data register via a shift register. After each byte, if the ACK bit is set, then the I2C module will send an answer low, and the RxNE bit will be set, and an interrupt will be generated if ITEVTEN and ITBUFEN are set. If RxNE is set and the original data is not read before the new data is received, then the BTF bit will be set and SCL will remain low until the BTF is cleared. Reading R16_I2Cx_STAR1 and then reading the data register will clear the BTF bit.

Figure 19-5 Receiver transmission sequence diagram



Description: S=Start (start condition), Sr=repeated start condition, P=Stop (stop condition), A=response, NA=non-response, EVTx=event (interrupt generated when ITEVFEN=1)
EVT5: SB=1, reading SR1 and then writing the address to DR register will clear the event.
EVT6: ADDR=1,reading SR1 and then reading SR2 will erase this event. In 10-bit master receive mode, START=1 of CR2 should be set after this event.
EVT6_1: There is no corresponding event flag and it is only suitable for receiving 1 byte. Exactly after EVT6 (i.e. after ADDR is cleared), the response and stop condition generation bits should be cleared.
EVT7: RxNE=1, read DR register to clear the event.
EVT7_1: RxNE=1, read the DR register to clear this event. Set ACK=0 and STOP request.
EVT9: ADDR10=1, reading SR1 and then writing to DR register will clear this event.

When the master device ends sending data, it will actively send an end event, i.e. set the STOP bit, and the I2C will switch to slave mode. In receive mode, the master device needs to NAK at the answer position of the last data bit, and after receiving NACK, the slave device releases control of the SCL and SDA lines; the master device can then send a stop/restart condition. Note that the I2C module will automatically switch to slave mode after the stop condition is generated.

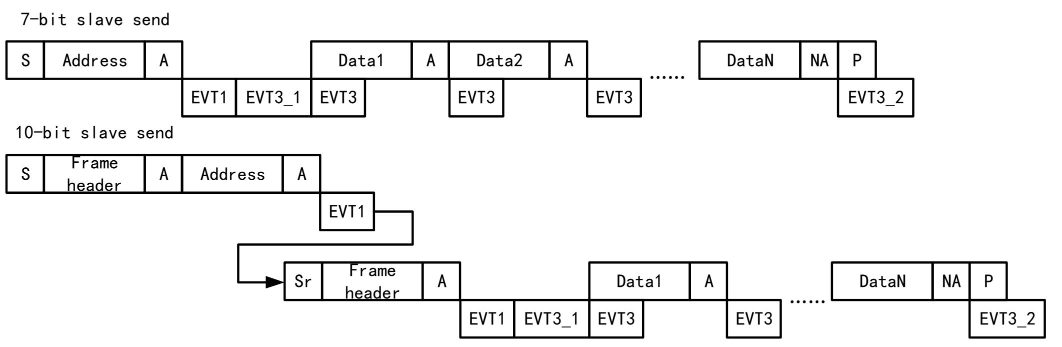
19.4 Slave mode

In the slave mode, the I2C module can identify its own address and the general call address. The software can control to enable or disable the identification of the broadcast calling address. Once the start event is detected,

the I2C module will compare the SDA data with its own address (the number of bits depends on ENDUAL and ADDMODE) or the broadcast address (when ENGC is set) through the shift register. If there is no match, it will be ignored until a new start event is generated. If it matches the header sequence, it will generate an ACK signal and wait for the address of the second byte; if the address of the second byte also matches or the whole-section address matches in the case of a 7-bit address, then:
Firstly generate an ACK response;
The ADDR bit will be set; if the ITEVTEN bit is already set, then the corresponding interrupt will be generated;
If the dual-address mode (the ENDUAL bit is set) is used, you also need to read the DUALF bit to determine which address is woken up by the master device.
The slave mode is the receiving mode by default. 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 the transmitter mode, and the TRA bit will indicate whether it is currently in receiver or transmitter mode.

Slave transmit mode:
After the ADDR bit is cleared, the I2C module sends bytes from the data register to the SDA line via a shift register. The slave device holds SCL low until the ADDR bit is cleared and the data to be sent has been written to the data register. (See EVT1 and EVT3 in the figure below). Upon receipt of an answer ACK, the TxE bit will be set and an interrupt will be generated if ITEVTEN and ITBUFEN are set. The BTF bit will be set if TxE is set but no new data is written to the data register before the end of the next data transmission. SCL will remain low until the BTF is cleared. Reading status register 1 (R16_I2Cx_STAR1) and then writing data to the data register will clear the BTF bit.

Figure 19-6 Slave transmitter transmission sequence diagram



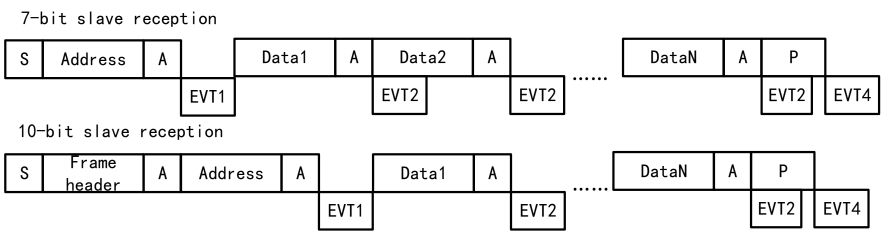
Description: S=Start (start condition), Sr=repeated start condition, P=Stop (stop condition), A=response, NA=non-response, EVTx=event (interrupt is generated when ITEVFEN=1)

EVT1;ADDR=1,read SR1 then read SR2 will eliminate the event.
EVT3_1: TxE=1, shift register empty, data register empty, write DR.
EVT3: TxE=1,shift register is not empty,data register is empty,writing DR will clear the event.
EVT3_2: AF=1, write '0' in AF bit of SR1 register to clear AF bit.

Note: 1: EVT1 and EVT3_1 events elongate SCL low until the end of the corresponding software sequence.
2: The software sequence of EVT3 must be completed before the end of the current byte transfer.

Slave receive mode:
After ADDR is cleared, the I2C module stores the data on SDA into the data register via a shift register. After every byte received, the I2C module sets an ACK bit and sets the RxNE bit. If ITEVTEN and ITBUFEN are set, an interrupt is also generated. If RxNE is set and the old data is not read before the new data is received, then BTF is set. SCL will remain low until the BTF bit is cleared. Reading status register 1 (R16_I2Cx_STAR1) and reading the data in the data register will clear the BTF bit.

Figure 19-7 Receiver transmission sequence diagram



Description: S=Start (start condition), Sr=repeated start condition, P=Stop (stop condition), A=response, NA=non-response, EVTx=event (interrupt generated when ITEVFEN=1)

EVT1;ADDR=1,reading SR1 then SR2 will erase the event.

EVT2: RxNE=1, reading DR will clear the event.

EVT4: STOPF=1,reading SR1 and then writing CR1 register will clear the event.

Note: 1: EVT1 event elongates SCL low until the end of the corresponding software sequence.

2: The software sequence of EVT2 must be completed before the end of the current byte transfer.

The master device will generate a stop condition after the last data byte is transferred. When the I2C module detects a stop event, it will set the STOPF bit, and if the ITEVFEN bit is set, it will also generate an interrupt. The user needs to read the status register (R16_I2Cx_STAR1) and then write the control register (e.g. reset control word SWRST) to clear it. (See EVT4 in the figure above).

19.5 Error

19.5.1 Bus error (BERR)

During address or data transmission, when the I2C module detects an external start or stop event, a bus error will be generated. When a bus error occurs, the BERR bit will be set, and an interrupt will be generated if ITERREN is set. The data is discarded and the hardware releases the bus in the slave mode. For a start signal, the hardware will consider it as a restart signal and begin to wait for an address or stop signal; for a stop signal, it will be operated according to normal stop conditions in advance. In the master mode, the hardware will not release the bus and will not affect the current transmission. The user code decides whether to abort the transmission.

19.5.2 Acknowledge failure (AF)

When the I2C module does not respond after detecting a byte, it will generate an acknowledge failure. When an acknowledge failure occurs, AF is set, and an interrupt is generated if ITERREN is set. If an AF error is encountered and the I2C module is working in the slave mode, the hardware must release the bus. If it is in master mode, the software must generate a stop event.

19.5.3 Arbitration lost (ARLO)

When the I2C module detects that the arbitration is lost, an arbitration loss error will be generated. When an arbitration loss error occurs, the ARLO bit will be set. If ITERREN is set, an interrupt will be generated; the I2C module will switch to the slave mode and no longer respond to the transmission initiated by its slave address, unless the host initiates a new start event; the hardware will release the bus.

19.5.4 Overrun/ Underrun error (OVR)

- Overrun error:

In the slave mode, if clock extension is disabled, the I2C module is receiving data. If 1-byte data has been received, but the data received at the previous time has not been read, an overrun error will occur. When an overrun error occurs, the last received byte will be discarded, and the sender shall retransmit the last byte transmitted.

- Underrun error:

In the slave mode, if the clock extension is disabled, the I2C module is sending data. If new data has not been written to the data register before the next byte of the clock arrives, an underrun error will occur. When an underrun error occurs, the data in the previous data register will be sent twice. If an underrun error occurs, the receiver shall discard the data received repeatedly. In order not to generate an underrun error, the I2C module shall write data into the data register before the first rising edge of the next byte.

19.6 Clock extension

If clock extension is disabled, there is a possibility of overrun/underrun errors. But if clock extension is enabled:

- In transmitter mode, if TxE is set and BTF is set, SCL will always be low, until the user reads the status register and writes the data to be sent to the data register;
- In receiver mode, if RxNE is set and BTF is set, SCL will remain low after receiving data until the user reads the status register and reads the data register;

It can be seen that enabling clock extension can avoid overrun/underrun errors.

19.7 SMBus

SMBus is also a two-wire interface, which is generally used between the system and power management. SMBus and I2C have many similarities. For example, SMBus uses the same 7-bit address mode as I2C.

Similarities between SMBus and I2C:

1) Master-slave communication mode; the host provides the clock and supports multiple masters and multiple slaves;
2) Two-wire communication structure, of which a warning line can be selected for SMBus;
3) Support 7-bit address format.

Differences between SMBus and I2C:

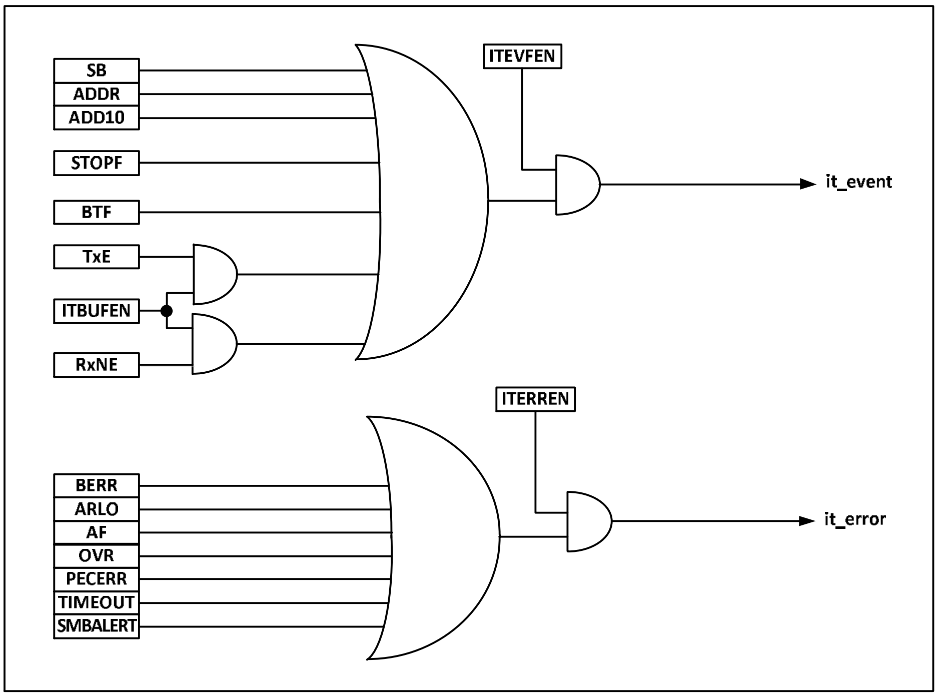
1) I2C supports the maximum speed of 400 KHz, while SMBus supports the maximum speed of 100 KHz, and SMBus has the minimum speed limit of 10 KHz;
2) When the SMBus clock is lower than 35mS, it will report a timeout, but I2C has no such limitation;
3) SMBus has a fixed logic level, but I2C does not, depending on VDD;
4) SMBus has a bus protocol, but I2C does not.

SMBus also includes device identification, address resolution protocol, unique device identifier, SMBus reminder, and various bus protocols. For details, please refer to SMBus specification version 2.0. When SMBus is used, only the SMBus bit of the control register needs to be set, and the SMBTYPE and ENAARP bits need to be configured as needed.

19.8 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 19-4.

Figure 19-4 I2C interrupt request



19.9 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 I2Cx_DATAR register address to the DMA_PADDRx register, and set the memory address in the DMA_MADDRx register, so that the data will be sent from the memory to the I2Cx_DATAR register after each TxE event.
2) Set the required number of transferred bytes in the DMA_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 DMA_CFGRx register.
4) Set the DIR bit in the DMA_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 DMA_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 DMA_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 DMA_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 DMA_CFGRx register.
4) Clear the DIR bit in the DMA_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 DMA_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.

19.10 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.

19.11 Debug mode

After the system enters the debug mode, the DBG_I2Cx_SMBUS_TIMEOUT bit in the DEBUG module can be used to determine whether to continue operating or stop the time-out control of I2CSMBus.

19.12 Register description

Table 19-1 I2C1 registers

Name	Access address	Description	Reset value
R16_I2C1_CTLR1	0x40005400	I2C1 control register1	0x0000
R16_I2C1_CTLR2	0x40005404	I2C1 control register2	0x0000
R16_I2C1_OADDR1	0x40005408	I2C1 address register1	0x0000
R16_I2C1_OADDR2	0x4000540C	I2C1 address register2	0x0000
R16_I2C1_DATAR	0x40005410	I2C1 data register	0x0000
R16_I2C1_STAR1	0x40005414	I2C1 status register1	0x0000
R16_I2C1_STAR2	0x40005418	I2C1 status register2	0x0000
R16_I2C1_CKCFGR	0x4000541C	I2C1 clock register	0x0000
R16_I2C1_RTR	0x40005420	I2C1 rise time register	0x0002

Table 19-2 I2C2 registers

Name	Access address	Description	Reset value
R16_I2C2_CTLR1	0x40005800	I2C2 control register1	0x0000

R16_I2C2_CTLR2	0x40005804	I2C2 control register2	0x0000
R16_I2C2_OADDR1	0x40005808	I2C2 address register1	0x0000
R16_I2C2_OADDR2	0x4000580C	I2C2 address register2	0x0000
R16_I2C2_DATAR	0x40005810	I2C2 data register	0x0000
R16_I2C2_STAR1	0x40005814	I2C2 status register1	0x0000
R16_I2C2_STAR2	0x40005818	I2C2 control register2	0x0000
R16_I2C2_CKCFGR	0x4000581C	I2C2 clock register	0x0000
R16_I2C2_RTR	0x40005820	I2C2 rise time register	0x0002

19.12.1 I2C Control Register (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	ALERT	PEC	POS	ACK	STOP	START	NOSTRETCH	ENGC	ENPEC	ENARP	SMBTYPE	Reserved	SMBUS	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	Reserved	RO	Reserved.	0
13	ALERT	RW	SMBus alert. This bit can be set or cleared by the user code. When PE is set, this bit can be cleared by hardware. 1: Drive the SMBusALERT pin to make it low, and the response address header shall closely follow the ACK signal; 0: Release the SMBusALERT pin to make it high, and the response address header shall closely follow the NACK signal.	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 fail when the 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	0

			received in the PEC shift register is PEC; 0: The ACK bit controls the ACK or NAK of the byte currently being received in the shift register. The PEC bit indicates that the byte of the shift register before the current bit 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 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 ADDB 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	ENARP	RW	ARP enable. Set this bit to enable the ARP. If SMBTYPE=0, the default address of the SMBus device is used. If SMBTYPE=1, the main address of the SMBus is used.	0
3	SMBTYPE	RW	SMBus device type: 1: SMBus master device; 0: SMBus slave device.	0
2	Reserved	RO	Reserved.	0
1	SMBUS	RW	SMBus mode selection. 1: SMBus mode; 0: I2C mode.	0
0	PE	RW	I2C peripheral enable. 1: I2C module enabled; 0: I2C module disabled.	0

19.12.2 I2C Control Register 2 (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	Time 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;
				0

			If ITBUFEN=1, TxE event is 1; If ITBUFEN=1, RxNE event is 1.	
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. The correct clock frequency must be input to generate the correct timing, and the allowable range is between 2~36MHz. It must be set between 000010b and 100100b, and the unit is MHz	0

19.12.3 I2C Address Register 1 (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	MUST1	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	MUST1	RW1	It must always be written 1 by software.	0
[13:10]	Reserved	RO	Reserved.	0
[9:8]	ADD[9:8]	RW	Interface address. The 9th bit and the 8th bit when a 10-bit address is used, while ignored when a 7-bit address is 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

19.12.4 I2C Address Register2 (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

19.12.5 I2C Data Register (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

19.12.6 I2C Status Register 1 (I2Cx_STAR1) (x=1/2)

Offset address: 0x14

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMB ALE RT	TIME OUT	Reser ved	PECE RR	OVR	AF	ARL O	BER R	TxE	RxNE	Reser ved	STOP F	ADD 10	BTF	ADD R	SB

Bit	Name	Access	Description	Reset value
15	SMBALERT	RW0	SMBus alert. It can be reset by user writing 0, or reset by hardware when PE becomes low. In master mode of SMBus: 1: SMBus alert is generated on the pin; 0: No SMBus alert. In slave mode of SMBus: 1: SMBAlert response address header to SMBAlert LOW received; 0: No SMBAlert response address header.	0
14	TIMEOUT	RW0	Timeout or Tlow error flag. It can be reset by user writing 0, or reset by hardware when PE becomes low. 1: SCL is low and has reached 25mS, or the accumulated clock expansion time of the master device low level exceeds 10mS, or the accumulated time of the slave device low level exceeds 25mS; 0: No timeout error. *Note: When this bit is set in slave mode, the slave device resets the communication and the hardware releases the bus. When this bit is set in master mode, hardware issues a stop condition.*	0

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	Data register not empty flag. Reading and writing to the data register will clear this bit, or when PE is 0, the hardware will clear this bit. 1: When data is received, data register not empty; 0: Data register empty.	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	0

			this bit. 1: After the response, the slave device will detect a stop event on the bus; 0: No stop event is detected.	
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, the bit will be changed to be set after the ACK of the second byte of the address is received; in 7-bit address mode, the bit will be set after the ACK of the address is received; 0: The address transmission is not finished. Slave mode: 1: The received address matches; 0: The address does not match or no address is received.	0
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

19.12.7 I2C Status Register 2 (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	SMBHOST	SMBDEFAULT	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	SMBHOST	RO	SMBus host header flag. When the stop bit or start bit is generated, or when PE=0, the hardware will clear this bit. 1: When SMBTYPE=1 and ENARP=1, the SMBus host address will be received; 0: SMBus host address is not received.	0
5	SMBDEFAULT	RO	SMBus device default address flag. When the stop bit or start bit is generated, or when PE=0, the hardware will clear this bit. 1: When ENARP=1, the default address of the SMBus device is received; 0: No address is received.	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

19.12.8 I2C Clock Register (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 of the high-level time in the fast mode and the high-level time. 1:36%; 0:33.3%.	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

19.12.9 I2C Rise Time Register (I2Cx_RTR) (x=1/2)

Offset address: 0x20

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

Bit	Name	Access	Description	Reset value
[15:6]	Reserved	RO	Reserved.	0
[5:0]	TRISE[5:0]	RW	Maximum rise time domain. The rise time of SCL in master mode is set at this bit. The maximum rising edge time is equal to TRISE-1 clock cycle. This bit can only be set when PE is cleared. For example, if the input clock cycle of the IIC module is 125nS and the value of TRISE is 9h, then the maximum rising edge time is (9-1)*125nS, i.e., 1000nS.	000010b

Chapter 20 Serial Peripheral Interface (SPI/I2S)

The module descriptions in this chapter are applicable to the whole series of CH32F2x, CH32V2x and CH32V3x microcontrollers.

SPI supports data exchange in 3-wire synchronous serial mode, and the chip select line supports hardware switching between master and slave modes, and supports communication with a single data line.

I2S is also a 3-wire synchronous serial interface communication protocol, which supports 4 audio standards, including Philips I2S standard, MSB alignment standard, LSB alignment standard and PCM standard.

20.1 Main features

20.1.1 SPI features

- Supports full-duplex synchronous serial mode
- Supports single-wire half-duplex mode
- Support master mode and slave mode, multi-slave mode
- Supports 8-bit or 16-bit data structures
- The highest clock frequency is supported to half of Fpclk
- Data ordering supports MSB or LSB first
- Supports hardware or software control of the NSS pin
- Transceiver supports hardware CRC
- Transceiver buffer supports DMA transfer
- Supports modifying clock phase and polarity

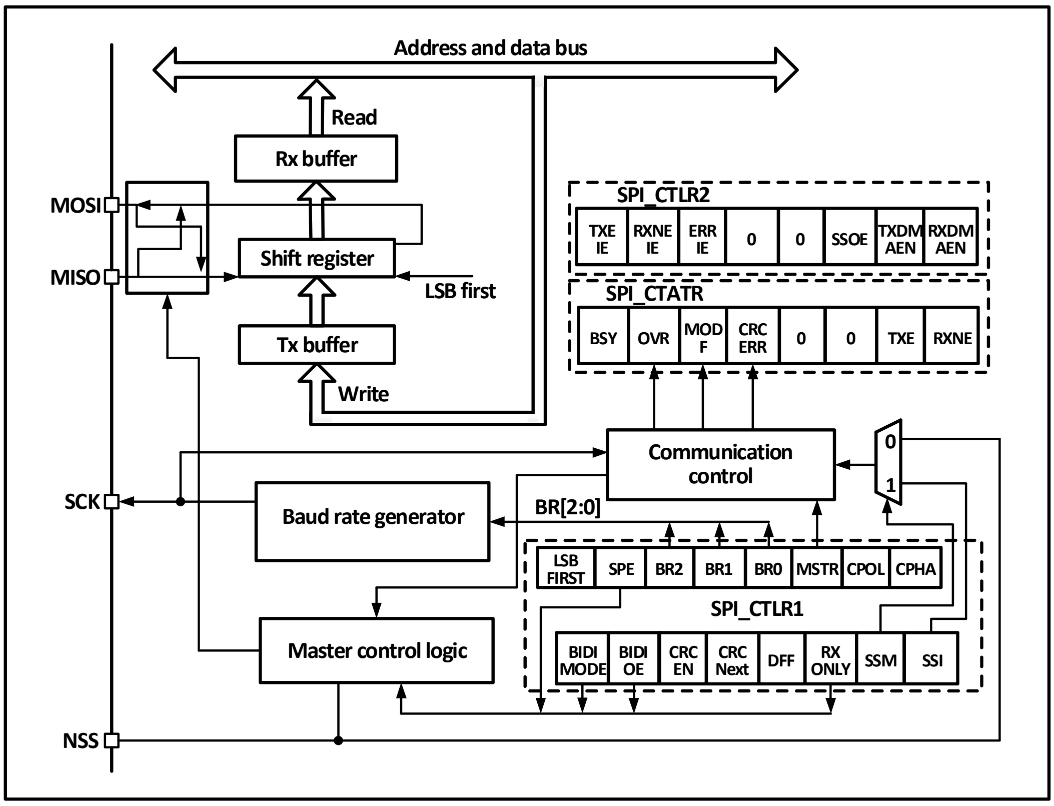
20.1.2 I2S features

- Support simplex communication
- Supports master mode and slave mode
- Supports 16-bit, 24-bit and 32-bit data formats
- Audio sampling frequency support range 8KHz-562.2KHz
- Supports clock polarity programmable
- Support common I2S protocols: Philips standard, MSB alignment standard, MSL alignment standard and PCM standard
- Transceiver buffer supports DMA transfer
- Support master clock output to external audio equipment

20.2 SPI functional description

20.2.1 Overview

Figure 20-1 SPI structure diagram



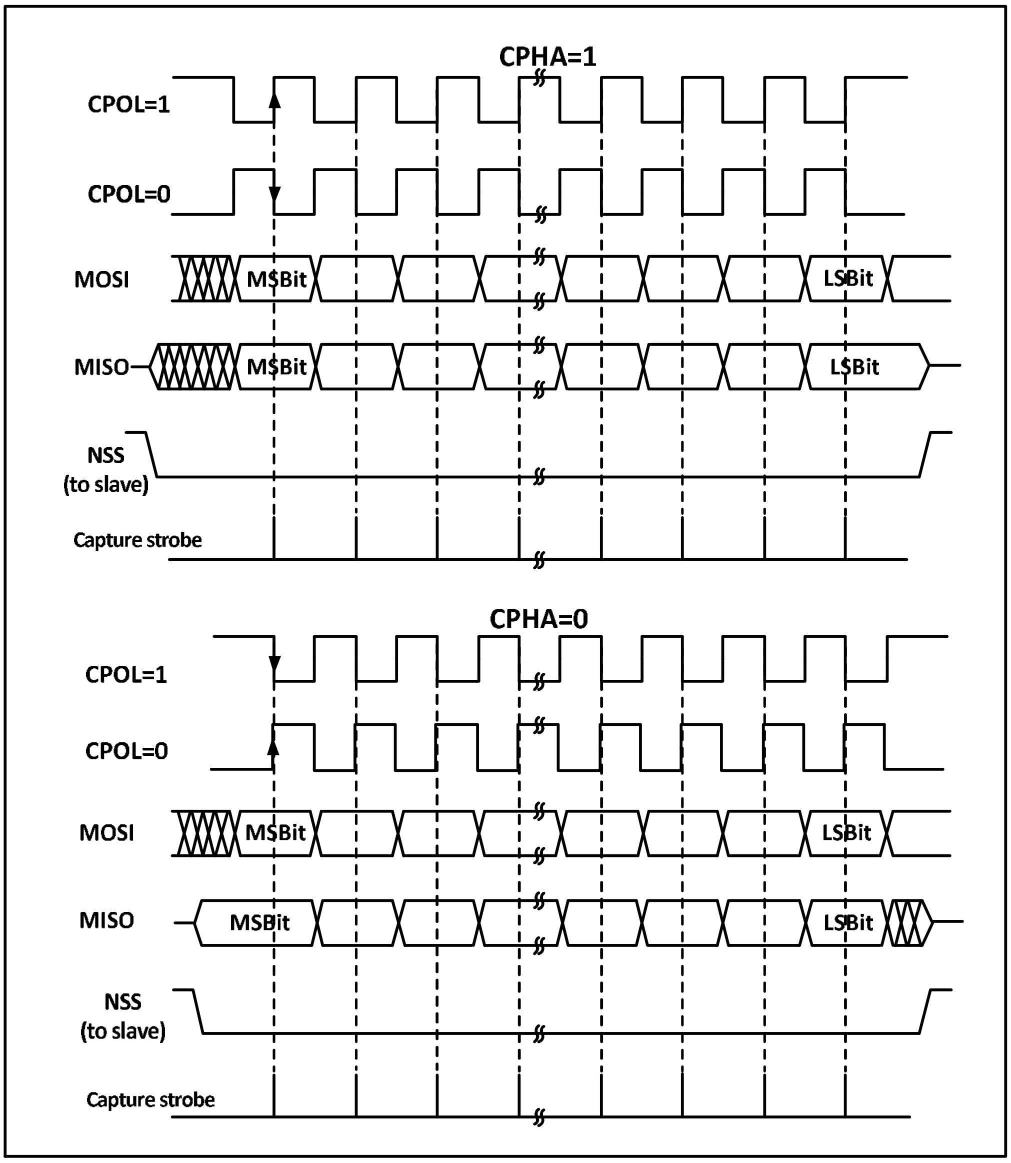
As can be seen from Figure 20-1, the 4 pins related to SPI are MISO, M0SI, SCK and NSS. The MISO pin is the data input pin when the SPI module works in the master mode; it is the data output pin when it works in the slave mode. When the MOSI pin works in master mode, it is a data output pin; when it 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, that is, when the SSOE is set, the NSS pin will be actively pulled down when the SPI host sends the output. If the NSS pin is pulled down, a hardware error will be generated; 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 20-2 below for details.

Figure 20-2 SPI model



The host and device need to be set to the same SPI mode, and the SPE bit needs to be cleared before configuring the SPI mode. The DEF bit can decide whether the single data length of SP is 8 bits or 16 bits. LSBFIRST can control whether a single data word is big-endian or little-endian.

20.2.2 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.

20.2.3 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 the CPOL and CPHA bits to match the host 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.

20.2.4 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.

20.2.5 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.

20.2.6 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.

20.2.7 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.

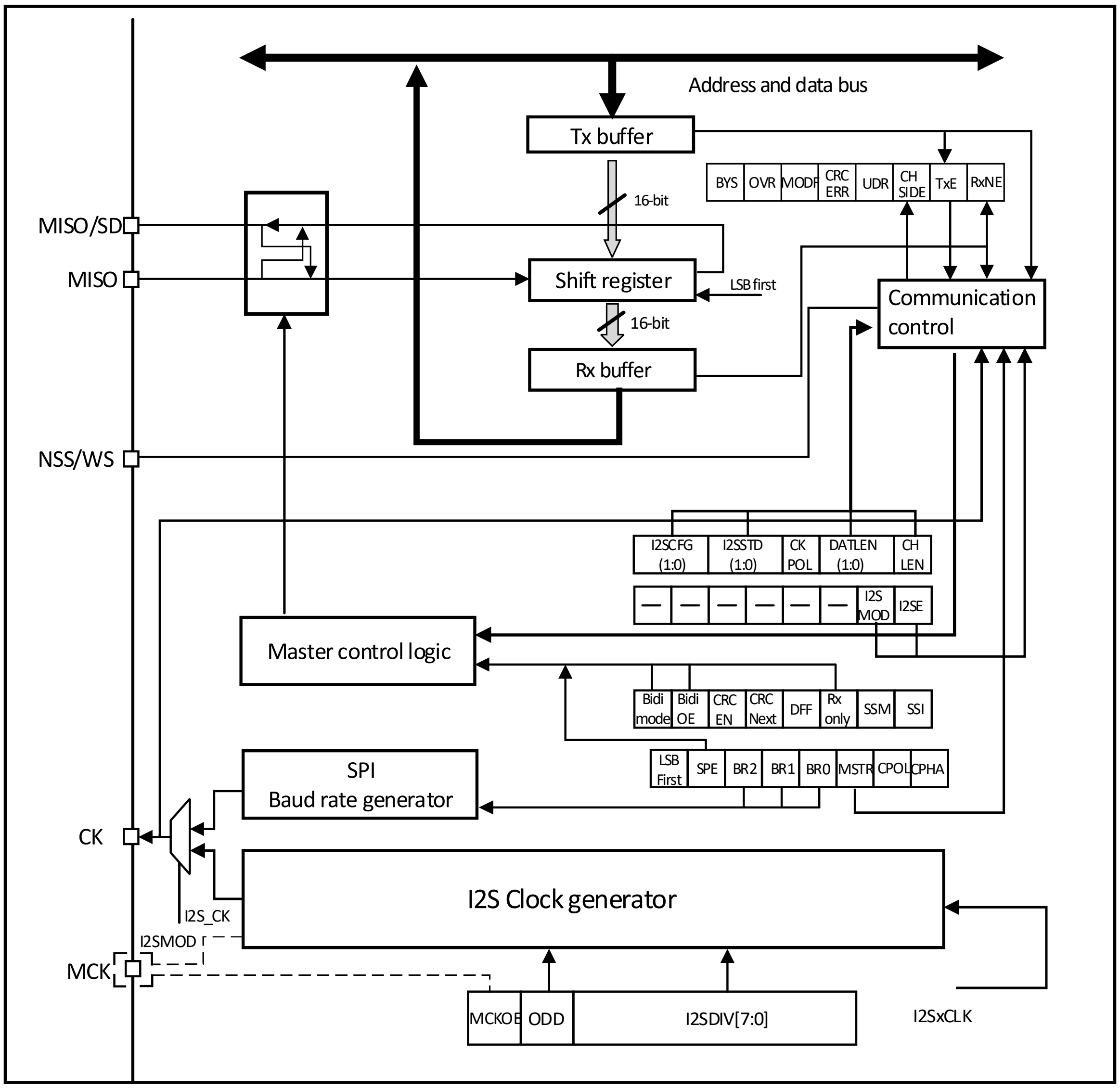
20.2.8 Interrupt

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.

20.3 I2S functional description

20.3.1 I2S overview

Figure 20-3 I2S structure diagram



The I2S function is enabled by setting the I2SMOD bit in register I2SCFGR. At this point, the SPI module can be used as an I2S audio interface. I2S and SPI share 3 pins:

● SD: Serial data (mapped to MOSI pin), used to send and receive 2 time-multiplexed data channels;

● WS: word selection (mapped to NSS pin), output as data control signal in master mode, and input in slave mode;

● CK: Serial clock (mapped to the SCK pin), output as a clock signal in master mode, and input in slave mode.

When the main clock is required for some external audio devices, there is an additional pin to output the clock:

● MCK: main clock (independent mapping), when the I2S is configured as the main mode and the MCKOE bit of the register I2SPR is 1, it is used as an output additional clock signal pin. The frequency of the output clock signal is preset to 256×Fs, where Fs is the sampling frequency of the audio signal.

When set to master mode, I2S uses its own clock generator to generate the clock signal for communication. This clock generator is also the clock source for the master clock output. There are 2 additional registers in

I2S mode, one is the clock generator configuration related register I2SPR, the other is the I2S general configuration register I2SCFGR (can set the audio standard, slave/master mode, data format, packet frame, clock pole parameters, etc.). Register CTLR1 and all CRCR registers are not used in I2S mode. Similarly, the SSOE bit in the CTLR2 register, and the MODF bit and the CRCERR bit in the register STATR are not used in the I2S mode. I2S uses the same register DATAR as SPI for 16-bit wide mode data transfer.

20.3.2 Supported audio protocols

The 3-wire bus supports time division multiplexing of audio data on 2 channels: left and right, but only one 16-bit register is used for transmit or receive. Therefore, when writing data to the data register, the software must write the corresponding data according to the channel currently being transmitted; similarly, when reading the register data, check the CHSIDE bit of the register STATR to determine which channel the received data belongs to. The left channel always sends data before the right channel (the CHSIDE bit has no meaning under the PCM protocol). There are 4 available data and packet frame combinations. Data can be sent in the following 4 data formats:

- 16-bit data is packed into 16-bit frames
- 16-bit data is packed into 32-bit frames
- 24-bit data is packed into 32-bit frames
- 32-bit data is packed into 32-bit frames

When using 16-bit data to expand to a 32-bit frame, the first 16 bits (MSB) are meaningful data, and the last 16 bits (LSB) are forced to 0. This operation requires no software intervention and no DMA request (only requires a read or write operation). 24-bit and 32-bit data frames require the CPU to perform 2 read or write operations on the register DATAR, and 2 DMA transfers are required when using DMA. For 24-bit data, after expansion to 32 bits, the lowest 8 bits are set to 0 by hardware. For all data formats and communication standards, the most significant bit (MSB) is always sent first. The I2S interface supports 4 audio standards, which can be selected by setting the I2SSTD[1:0] bits and PCMSYNC bits in the I2SCFGR register.

20.3.2.1 I2S Philips Standard

Under this standard, pin WS is used to indicate which channel the data being sent belongs to. The pin is valid 1 clock cycle before the first data bit (MSB) is sent. The sender changes the data on the falling edge of the clock signal (CK) and the receiver reads the data on the rising edge. The WS signal also changes on the falling edge of the clock signal.

Figure 20-4 Philips protocol waveform (16/32 full precision, CPOL=0)

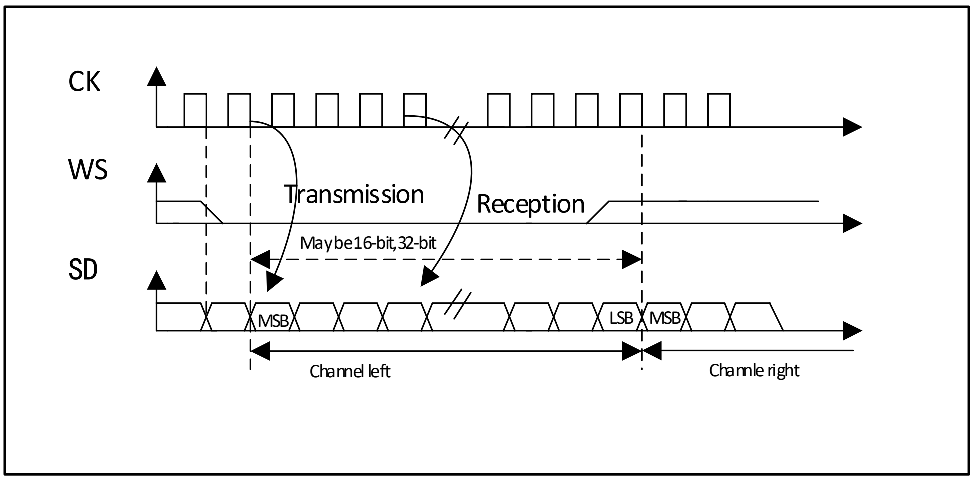
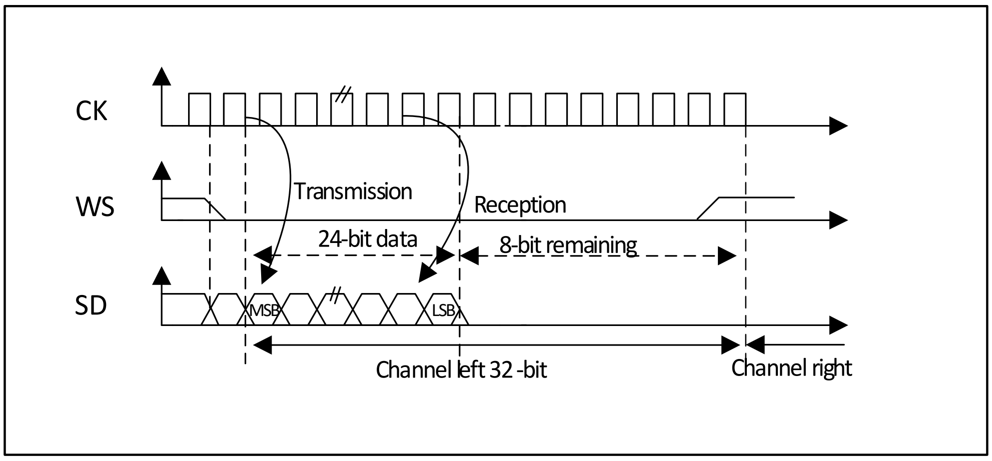
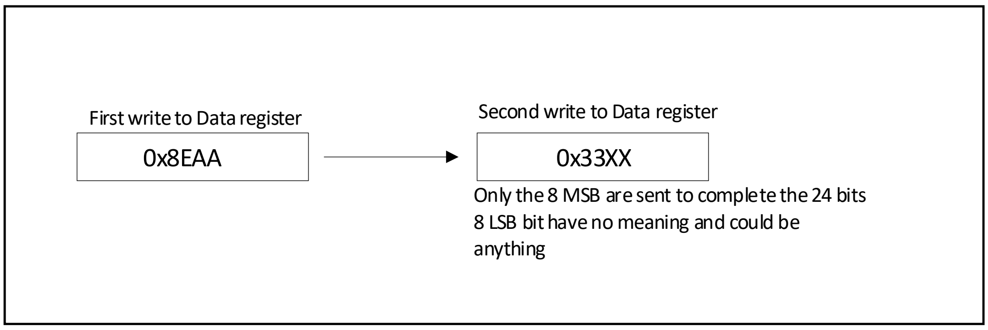


Figure 20-5 Philips protocol waveform (24-bit frame, CPOL=0)



This mode requires 2 read or write operations to the SPI_DATAR register. In transmit mode: if you need to send 0x8EAA33 (24 bits):


First write to Data register
0x8EAA
Second write to Data register
0x33XX
Only the 8 MSB are sent to complete the 24 bits
8 LSB bit have no meaning and could be anything

In receive mode: if 0x8EAA33 is received:

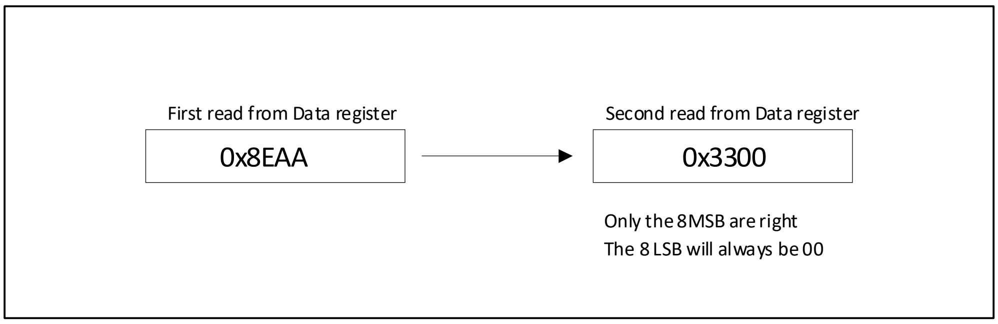
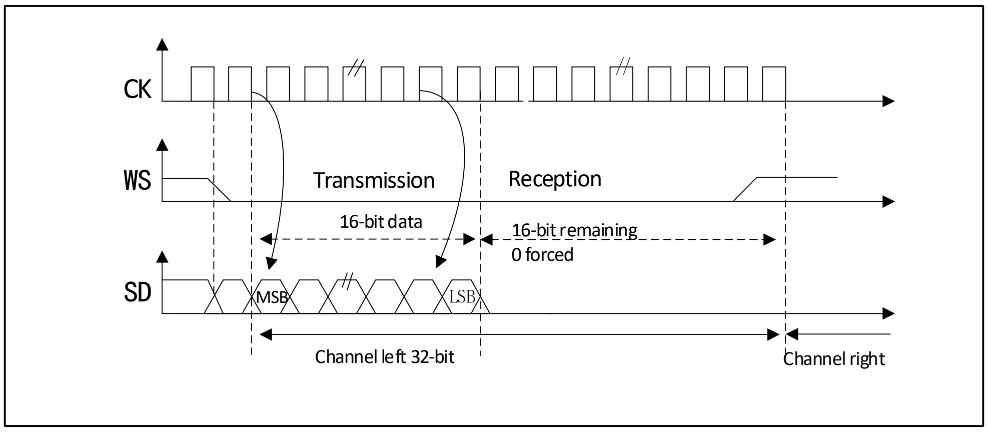

First read from Data register
0x8EAA
Second read from Data register
0x3300
Only the 8MSB are right
The 8 LSB will always be 00

Figure 20-6 Philips protocol standard waveforms (16-bit extended to 32-bit packet frame, CPOL=0)

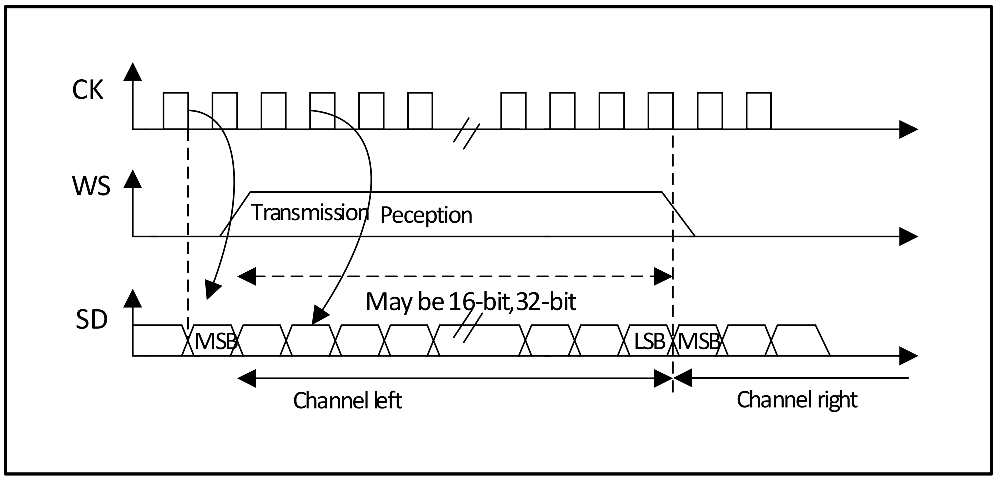


In the I2S configuration stage, if you choose to extend the 16-bit data to a 32-bit channel frame, you only need to access the register DATAR once to extend to 32-bit. The lower 16 bits are set to 0x0000 by hardware. If the data to be transmitted or received is 0x76A3 (extended to 32 bits is 0x76A30000), only one operation of DATAR is required. When sending, the MSB needs to be written into the register DATAR; if the flag bit TXE is 1, it means that new data can be written, and if the corresponding interrupt is allowed, an interrupt can be generated. The transmission is done by hardware, even if the last 16 bits of 0x0000 have not been sent, the TXE will be set and the corresponding interrupt will be generated. When receiving, each time the upper 16-bit half-word (MSB) is received, the flag bit RXNE is set to 1, and if the corresponding interrupt is enabled, an interrupt can be generated. This way, there is more time between the 2 reads and writes, preventing underflow or overflow conditions.

20.3.2.2 MSB alignment standard

Under this standard, the WS signal and the first data bit, the most significant bit (MSB), are generated simultaneously.

Figure 20-7 MSB aligned 16-bit or 32-bit full precision (CPOL = 0)



The sender changes data on the falling edge of the clock signal; the receiver reads data on the rising edge.

Figure 20-8 MSB aligned 24-bit data, CPOL = 0

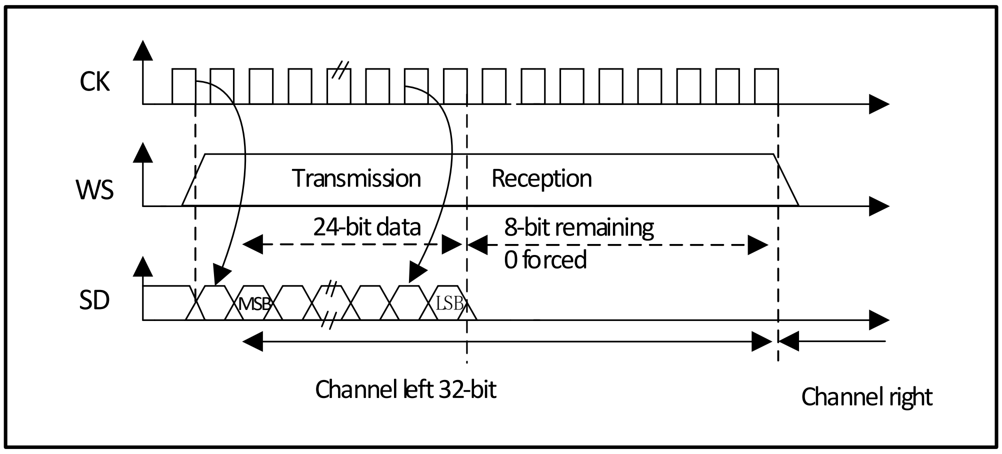
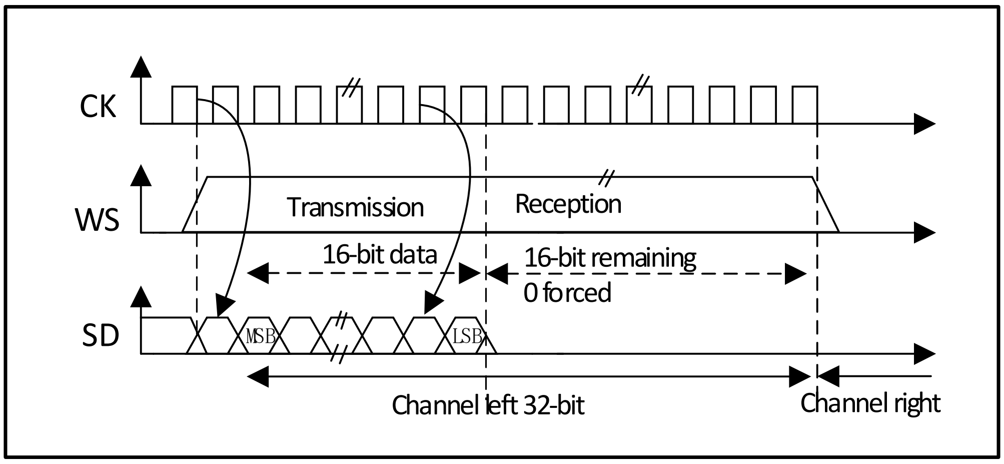


Figure 20-9 MSB-aligned 16-bit data extended to 32-bit packet frame, CPOL = 0



20.3.2.3 LSB alignment standard

This standard is similar to the MSB alignment standard (no difference in 16-bit or 32-bit full-precision frame format).

Figure 20-10 LSB-aligned 16- or 32-bit full precision, CPOL = 0

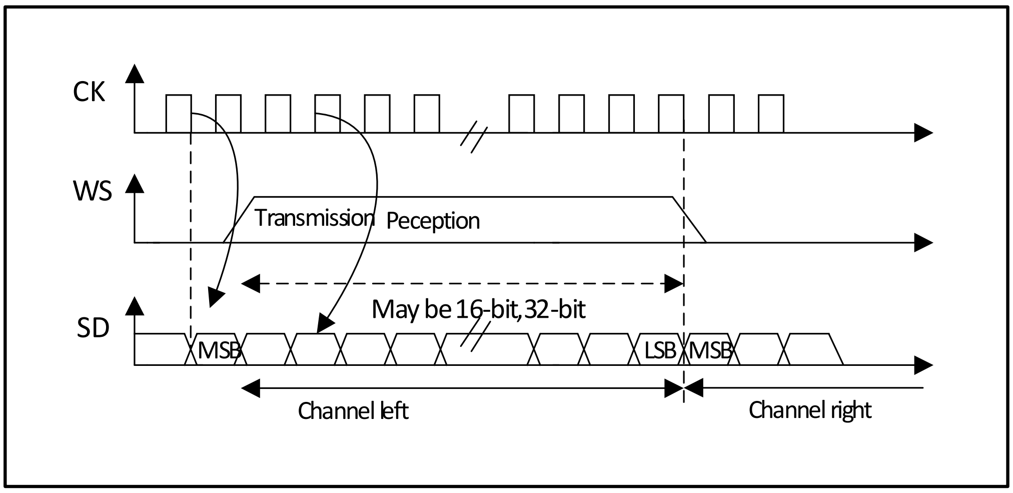
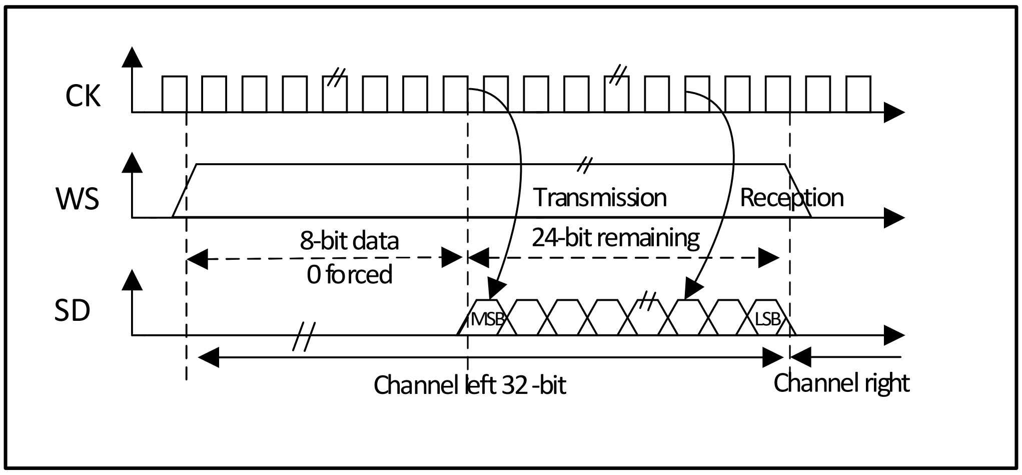
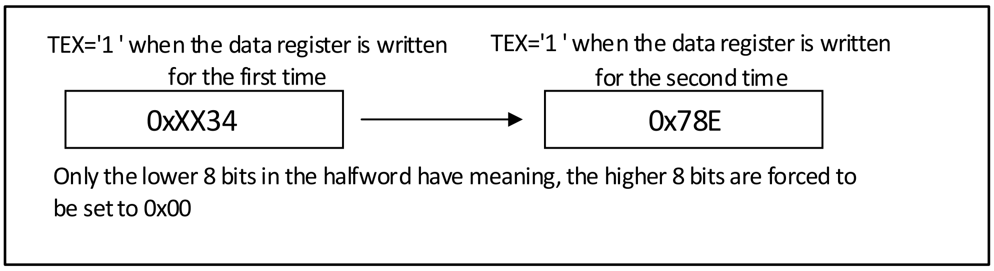


Figure 20-11 LSB-aligned 24-bit data, CPOL = 0



In transmit mode, if you want to send data 0x3478AE, you need to write the register DATAR twice by software or DMA.


TEX='1 ' when the data register is written for the first time
0xXX34
TEX='1 ' when the data register is written for the second time
0x78E
Only the lower 8 bits in the halfword have meaning, the higher 8 bits are forced to be set to 0x00

In receive mode, if you want to receive data 0x3478AE, you need to read the register DATAR once when 2 consecutive RXNE events occur.

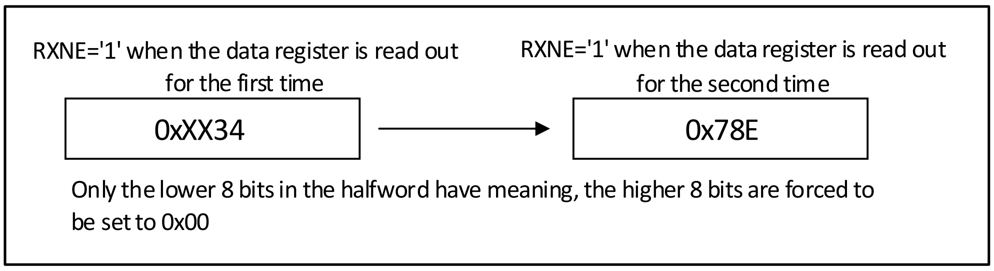
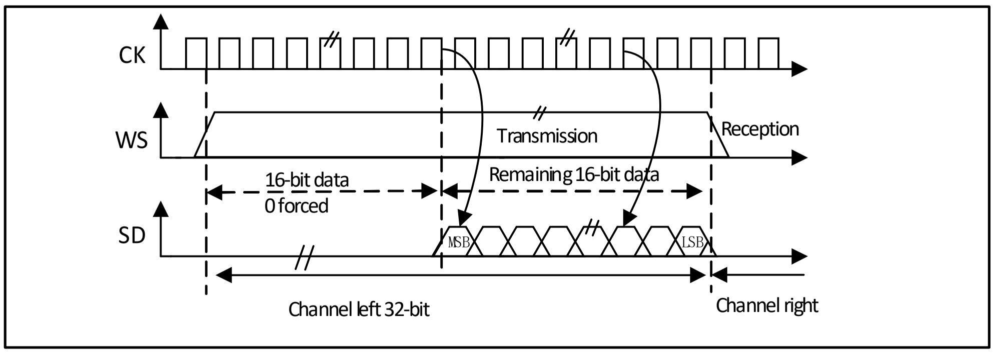

RXNE='1' when the data register is read out for the first time
0xXX34
RXNE='1' when the data register is read out for the second time
0x78E
Only the lower 8 bits in the halfword have meaning, the higher 8 bits are forced to be set to 0x00

Figure 20-12 LSB aligned 16-bit data extended to 32-bit packet frame, CPOL = 0



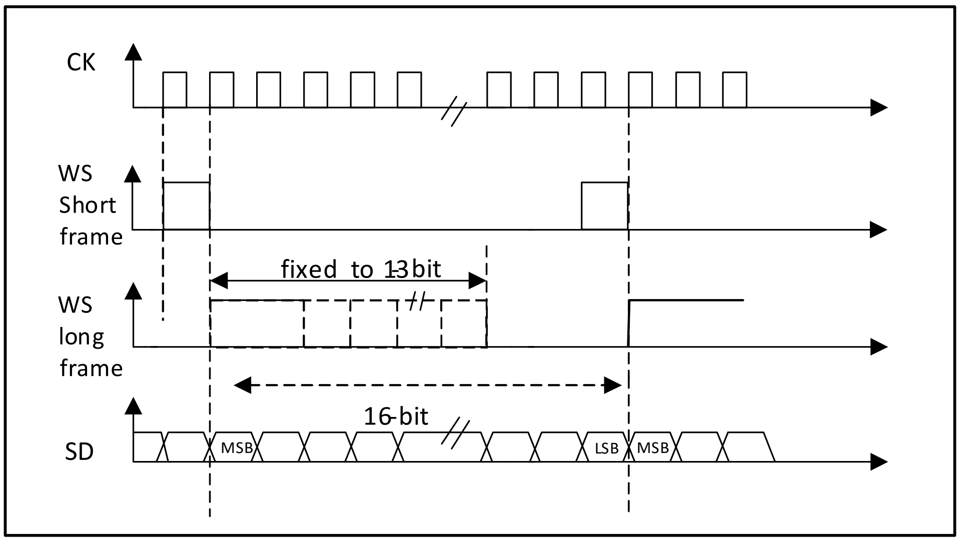
During the I2S configuration stage, if you choose to extend the 16-bit data to a 32-channel frame, you only need to access the DATAR register once. At this point, the high half word (16-bit MSB) after extending to 32 bits is set to 0x0000 by hardware.

If the data to be transmitted or received is 0x76A3 (extended to 32 bits is 0x000076A3), only one operation of DATAR is required. When sending, if TXE is 1, the user needs to write the data to be sent (i.e. 0x76A3). The 0x0000 part, which is used to extend to 32 bits, is first sent by the hardware, and once the valid data starts to be sent from the SD pin, the next TXE event occurs. On reception, the RXNE event occurs as soon as valid data is received (instead of the 0x0000 part). This way, there is more time between the 2 reads and writes, preventing underflow or overflow conditions.

20.3.2.4 PCM standard

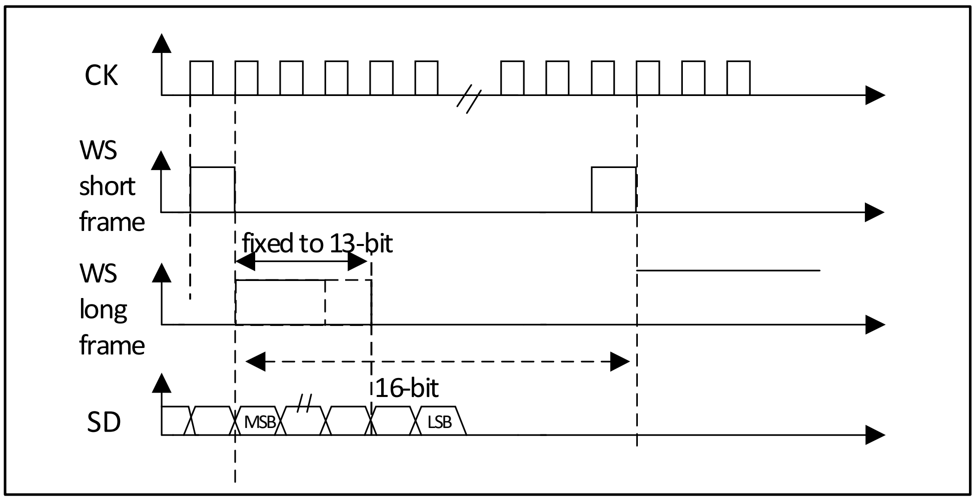
Under the PCM standard, there is no information for channel selection. The PCM standard has 2 available frame structures, short or long, which can be selected by setting the PCMSYNC bit in register I2SCFGR.

Figure 20-13 PCM standard waveform (16-bit)



For the long frame, in the main mode, the valid time of the WS signal used for synchronization is fixed to 13 bits. For short frames, the length of the WS signal used for synchronization is only 1 bit.

Figure 20-14 PCM standard waveform (16-bit)



No matter which mode (master or slave), which synchronization method (short frame or long frame), the time difference between 2 consecutive frames of data and between 2 synchronization signals, (even in slave mode) needs to be set by setting the I2SCFGR register determined by the DATLEN bit and the CHLEN bit.

20.3.3 Clock generator

The bit rate of I2S determines the data flow on the I2S data line and the frequency of the I2S clock signal. I2S bit rate = number of bits per channel × number of channels × audio sampling frequency.

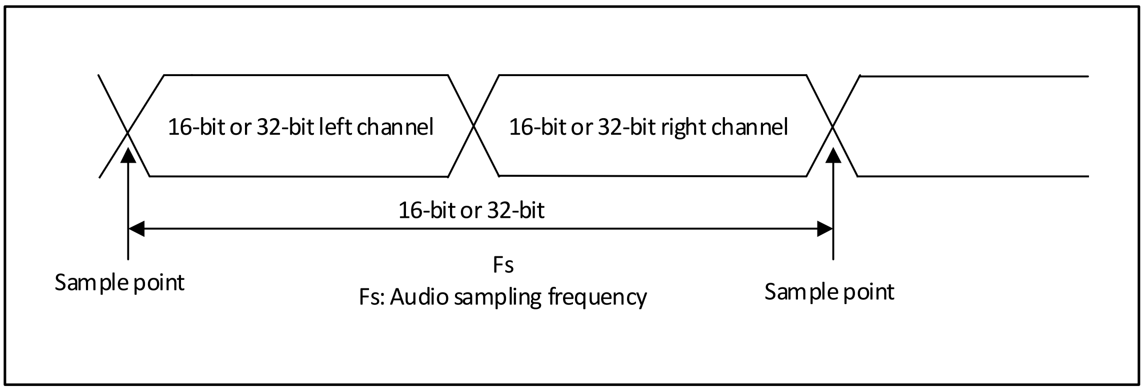
For a signal with left and right channels and 16-bit audio, the I2S bit rate is calculated as follows:

I2S bit rate = 16 × 2 × Fs

If the packet length is 32 bits, the I2S bit rate is calculated as follows:

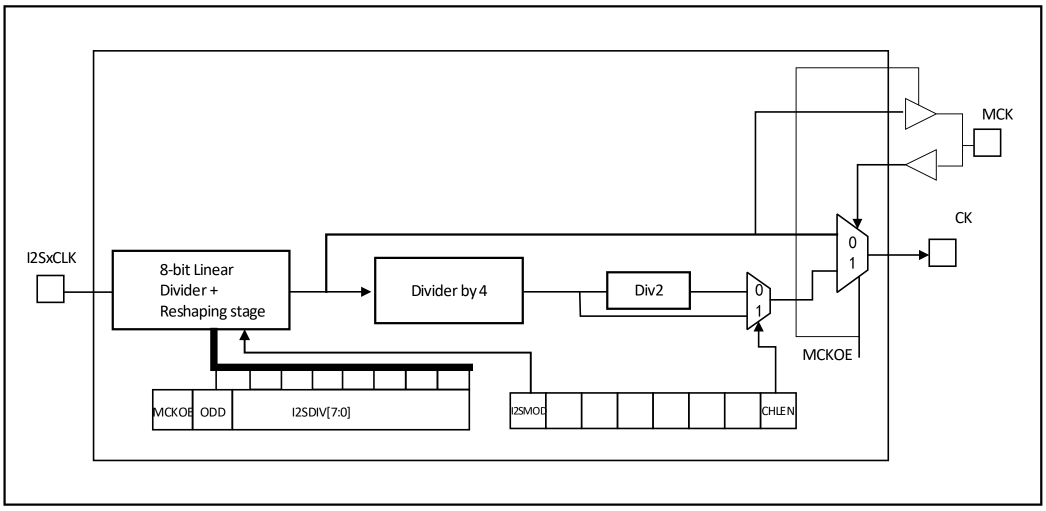
I2S bit rate = 32 × 2 × Fs

Figure 20-15 Audio sampling frequency definition


16-bit or 32-bit left channel
16-bit or 32-bit right channel
16-bit or 32-bit
Sample point
Fs
Fs: Audio sampling frequency
Sample point

In master mode, the linear divider needs to be set correctly in order to obtain the desired audio frequency.

Figure 20-16 I2S clock generator structure



The clock source of I2SxCLK in the figure is the system clock (i.e., the HSI, HSE, or PLL that drives the AHB clock). I2SxCLK can come from SYSCLK, or the PLL3 VCO (2xPLL3CLK) clock, which can be selected by the I2S2SRC and I2S3SRC bits in the RCC_CFGR2 register. Audio can be sampled at 96KHz, 48KHz, 44.1KHz, 32KHz, 22.05KHz, 16KHz, 11.025KHz, or 8KHz (or any value within this range). To obtain the desired frequency, set the linear divider according to the following formula:

When the master clock needs to be generated (the MCKOE bit of the register SPI_I2SPR is 1):
When the frame length of the channel is 16 bits, Fs = I2SxCLK / [(16*2) * ((2*I2SDIV) + ODD)*8]
When the frame length of the channel is 32 bits, Fs = I2SxCLK / [(32*2) * ((2*I2SDIV) + ODD)*4]
When the main clock is turned off (MCKOE bit is '0'):
When the frame length of the channel is 16 bits, Fs = I2SxCLK / [(16*2) * ((2*I2SDIV) + ODD)]
When the frame length of the channel is 32 bits, Fs = I2SxCLK / [(32*2) * ((2*I2SDIV) + ODD)]

20.3.4 I2S master mode

Set I2S to work in master mode, the serial clock is output by pin CK, and the word selection signal is generated by pin WS. The master clock (MCK) can be selected to be output or not output by setting the MCKOE bit in register I2SPR.

20.3.4.1 Configuration process

- Set I2SDIV[7:0] in the I2SPR register to define the serial clock baud rate corresponding to the audio sampling frequency. Also define the ODD bit of the register I2SPR.
- Set the CKPOL bit to define the level state of the communication clock when it is idle. If the main clock MCK needs to be provided to the external DAC/ADC audio device, the MCKOE position of the register I2SPR is required.
- Set the I2SMOD bit in the I2SCFGR register to 1 to activate the I2S function, set the I2SSTD[1:0] and PCMSYNC bits to select the I2S standard used, and set CHLEN to select the number of data bits per channel. Also set the I2SCFG[1:0] in the SPI_I2SCFGR register to select the I2S master mode and direction (transmitter or receiver).
- If required, the required interrupt function and DMA function can be turned on by setting register CR2.
- The I2SE bit in the I2SCFGR register must be set to 1.
- Pins WS and CK need to be configured for output mode. If the MCKOE bit in the SPI_I2SPR register is 1, the pin MCK is also configured to output mode.

20.3.4.2 Transmission process

The transmission process starts when 1 half word (16 bits) of data is written to the transmit buffer. It is assumed that the first data written to the transmit buffer corresponds to the left channel data. When the data is moved from the transmit buffer to the shift register, the flag bit TXE is set to 1. At this time, the data corresponding to the right channel should be written into the transmit buffer. The flag bit CHSIDE indicates which channel the data to be transmitted currently corresponds to. The value of the flag bit CHSIDE is updated when TXE is 1, so it has meaning when TXE is 1. After the data of the left channel and then the right channel have been transmitted, it can be regarded as a complete data frame. It is not possible to transmit only part of the data frame, such as data only for the left channel.

When the first bit of data is sent out, the half-word data is transferred to the 16-bit shift register in parallel, and then the following bits are sent out from pins MOSI/SD in high-order first. Each time data is moved from the transmit buffer to the shift register, the flag bit TXE is set to 1, and an interrupt is generated if the TXEIE bit in the CR2 register is 1.

In order to ensure continuous audio data transmission, it is recommended to write the next data to be transmitted to the register DATAR before the current transmission is completed. It is recommended to wait for the flag bit TXE=1 and BSY=0 when the I2S function is to be turned off, and then clear the I2SE bit to '0'.

20.3.4.3 Reception process

Except for point 3, the configuration steps of the reception process are consistent with the transmission process (see the aforementioned "transmission process"). It is necessary to select the main reception mode by configuring I2SCFG[1:0]. Regardless of the data and channel length, audio data is always received in 16-bit packets. That is, after the receive buffer is filled up each time, the flag bit RXNE is set to 1, and if the RXNEIE bit of the register CR2 is 1, an interrupt is generated. Depending on the configured data and channel length, receiving data from the left or right channel will require 1/2 times to transmit the data to the receive buffer. The RXNE flag is cleared by reading the DATAR register. CHSIDE is updated after each reception. Its value depends on the WS signal generated by the I2S unit. If the previous received data has not been read, and new data is received, that is, an overflow occurs, and the flag bit OVR is set to 1. If the ERRIE bit in the CR2 register is 1, an interrupt is generated, indicating that an error has occurred. To disable I2S, special operations need to be performed to ensure that the I2S module can normally complete the transfer cycle without starting a new data transfer. The operation process is related to the data configuration and channel length, and the mode of the audio protocol:

● 16-bit data extended to 32-bit channel length (DATLEN=00 and CHLEN=1), using LSB (low bit) alignment mode (I2SSTD=10)

a) Wait for the penultimate (n-1) RXNE=1;
b) Wait for 17 I2S clock cycles (using software delay);
c) Disable I2S (I2SE=0).

● 16-bit data extended to 32-bit channel length (DATLEN=00 and CHLEN=1), using MSB (high bit) alignment, I2S or PCM mode (I2SSTD=00, I2SSTD=01 or I2SSTD=11 respectively)

a) Wait for the last RXNE=1;
b) Wait for 1 I2S clock cycle (using software delay);
c) Disable I2S (I2SE=0).

● For all other combinations of DATLEN and CHLEN, any audio mode selected by I2SSTD, use the following method to turn off I2S:

a) Wait for the penultimate (n-1) RXNE=1;
b) Wait for one I2S clock cycle (using software delay);
c) Disable I2S (I2SE=0).

Note: The BSY flag is always low during transmission.

20.3.5 I2S slave mode

In slave mode, I2S can be set to transmission and reception mode. The configuration method of the slave mode basically follows the same process as the configuration of the master mode. In slave mode, no clock is required from the I2S interface. Both the clock signal and the WS signal are provided by an external master I2S device connected to the corresponding pins. So the user does not need to configure the clock.

The configuration steps are as follows:

- Set the I2SMOD bit in the I2SCFGR register to activate the I2S function; set I2SSTD[1:0] to select the I2S standard used; set DATLEN[1:0] to select the number of data bits; set CHLEN to select the number of data bits per channel. Set the I2SCFG[1:0] in the I2SCFGR register to select the data direction (transmitter or receiver) of the I2S slave mode.
- As needed, set register CR2 to turn on the desired interrupt function and DMA function.
- The I2SE bit in the I2SCFGR register must be set to 1.

20.3.5.1 Transmission process

The transmission process starts when the external master sends a clock signal, and when the NSS_WS signal requests data transmission. The slave device must be enabled and the I2S data register must be written before the external master device can start communication. For the MSB-aligned and LSB-aligned modes of I2S, the first data item written to the data register corresponds to the data of the left channel. When the communication starts, the data is transferred from the transmit buffer to the shift register, and then the flag bit TXE is set to 1; at this time, the data item corresponding to the right channel should be written into the I2S data register. The flag bit CHSIDE indicates which channel the data to be transmitted currently corresponds to. Compared to the transmit flow in master mode, in slave mode, CHSIDE depends on the WS signal from the external master I2S. This means that the slave I2S has to prepare the first data to be sent before receiving the clock signal generated by the master. A WS signal of 1 means that the left channel is sent first.

20.3.5.2 Reception process

Except for point 1, the configuration steps are the same as the transmission process. The master reception mode needs to be selected by configuring I2SCFG[1:0]. Regardless of the data and channel length, audio data is always received in the form of 16-bit packets, that is, every time the receive buffer is filled, the flag bit RXNE is set to 1, and if the RXNEIE bit in the I2S_CTLR2 register is 1, an interrupt is generated. According to different data and channel length settings, receiving left channel or right channel data will require 1 or 2 times to transmit data to the receive buffer. CHSIDE is updated every time data is received (to be read out from DATAR), which corresponds to the WS signal generated by the I2S unit. Reading the SPI_DATAR register will clear the RXNE bit. When the previous received data has not been read out and new data is received, an overflow occurs, and the flag bit OVR is set to 1; if the ERRIE bit in the register I2S_CTLR2 is 1, an interrupt is generated, indicating that an error has occurred. To turn off the I2S function, it is necessary to clear the I2SE bit to 0 when the last RXNE=1 is received.

20.3.6 Status flag

There are 3 status flags for the user to monitor the status of the I2S bus.

20.3.6.1 Busy flag (BSY)

The BSY flag is set and cleared by hardware (writing this bit has no effect). This flag bit indicates the status of the I2S communication layer. When this bit is 1, it indicates that I2S communication is in progress, with one exception: in master receive mode (I2SCFG=11), the BSY flag is always low during reception. Before the software closes the SPI module, the BSY flag can be used to detect whether the transfer is over, so as to avoid destroying the last transfer, so it is necessary to strictly follow the procedure below. When a transfer starts, the BSY flag is set to 1, unless the I2S module is in master receive mode. This flag is cleared when the transfer ends or when the I2S module is turned off. When communication is continuous, in master transmit mode, the BSY flag is always high during the entire transfer; in slave mode, between each data item transmission, the BSY flag goes low within 1 I2S clock cycle.

20.3.6.2 TX buffer empty flag (TXE)

When the flag bit is 1, it indicates that the sending buffer is empty, and new data to be sent can be written into the sending buffer. When there is data in the transmit buffer, the flag bit is cleared to 0. When I2S is turned off (I2SE bit is 0), this flag is also 0.

20.3.6.3 RX buffer not empty flag (RXNE)

Setting this flag bit indicates that there is valid data received in the receive buffer. This bit is cleared to 0 when the DATAR register is read.

20.3.6.4 Channel Side flag (CHSIDE)

In transmit mode, this flag is refreshed when TXE is high, indicating which channel the data is sent from the SD pin. If an underflow error occurs in the slave sending mode, the value of this flag is invalid, and the I2S needs to be turned off and on before restarting the communication. In receive mode, this flag bit is refreshed when the register DATAR receives data, indicating the channel where the received data is located. Note that if an error occurs, this flag is meaningless, and I2S needs to be turned off and on again. Under the PCM standard, no matter the short frame format or the long frame format, this flag has no meaning. If the flag bit OVR or UDR in the STATR register is 1, and the ERRIE bit in the CR2 register is 1, an interrupt will be generated. (After the interrupt source has been cleared) The interrupt flag can be cleared by reading the STATR register.

20.3.7 Error flag

20.3.7.1 Underrun flag (UDR)

In the slave transmit mode, if the new data has not been written to the DATAR register when the first clock edge of the data transfer arrives, this flag will be set to 1. This flag is valid only after the I2SMOD bit of the register I2SCFGR is set. An interrupt will be generated if the ERRIE bit in register CR2 is 1. This flag is cleared by reading the STATR register.

20.3.7.2 Overrun flag (OVR)

If the previous received data has not been read, and new data is received, that is, an overflow occurs, and the flag is set to 1. If the ERRIE bit of the register CTLR2 is 1, an interrupt is generated to indicate that an error has occurred. At this time, the content of the receiving buffer will not be refreshed with the new data sent from the sending device. A read of register DATAR returns the last correctly received data. All other 16-bit data sent by the sending device after the overflow occurs are lost. This flag is cleared by first reading the DATAR register and then reading the STATR register.

20.3.8 I2S interrupt

I2S has 4 interrupt sources, of which the send buffer is empty and the receive buffer is not empty. These 2 events will set TXE and RXNE respectively, and an interrupt will be generated when the TXEIE and RXNEIE bits are set respectively. If the previous received data has not been read out, new data is received, that is, an overflow occurs. If ERRIE is set, an overflow interrupt will be generated; in slave transmit mode, if the first clock of data transmission When the edge arrives, the new data is still not written to the DATAR register. If ERRIE is set, an underflow interrupt will be generated.

20.3.9 DMA features

The way DMA works in I2S mode is exactly the same as in SPI mode except that the CRC function is not available. Because there is no data transmission protection system in I2S mode.

20.4 Register description

Table 20-1 SPI1 registers

Name	Access address	Description	Reset value
R16_SPI1_CTLR1	0x40013000	SPI1 control register1	0x0000
R16_SPI1_CTLR2	0x40013004	SPI1 control register2	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_I2S_CFGR	0x4001301C	SPI1 I2S configuration register	0x00
R16_SPI1_HSCR	0x40013024	SPI1 high-speed control register	0x00

Table 20-2 SPI2 registers

Name	Access address	Description	Reset value
R16_SPI2_CTLR1	0x40003800	SPI2 control register1	0x0000
R16_SPI2_CTLR2	0x40003804	SPI2 control register2	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_I2S_CFGR	0x4000381C	SPI2 I2S configuration register	0x00
R16_SPI2_I2SPR	0x40003820	SPI2 I2S prescaler register	0x00
R16_SPI2_HSCR	0x40003824	SPI2 high-speed control register	0x00

Table 20-3 SPI3 registers

Name	Access address	Description	Reset value
R16_SPI3_CTLR1	0x40003C00	SPI3 control register1	0x0000
R16_SPI3_CTLR2	0x40003C04	SPI3 control register2	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_I2S_CFGR	0x40003C1C	SPI3 I2S configuration register	0x00
R16_SPI3_I2SPR	0x40003C20	SPI3 I2S prescaler register	0x00
R16_SPI3_HSCR	0x40003C24	SPI3 high-speed control register	0x00

20.4.1 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
BIDI MODE	BIDI OE	CRC EN	CRC NEXT	DFF	RX ONLY	SSM	SSI	LSB FIRST	SPE	BR[2:0]			MST R	CPO L	CPH A

Bit	Name	Access	Description	Reset value
15	BIDIMODE	RW	Unidirectional data mode enable. 1: Single-line bidirectional mode; 0: 2-wire bidirectional mode.	0
14	BIDIOE	RW	Single-line output enable, used in conjunction with BIDIMODE. 1: Enable output, only send; 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. 1: Enable CRC calculation; 0: Disable CRC calculation.	0
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: Send the CRC result; 0: Continue to send 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 send 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.	0
7	LSBFIRST	RW	Frame format control. This bit cannot be modified during communication. 1: Send LSB first; 0: Send 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.
			000: FPCLK/2; 001: FPCLK/4;
			010: FPCLK/8; 011: FPCLK/16;
			100: FPCLK/32; 101: FPCLK/64;
			110: FPCLK/128; 111: FPCLK/256.
				0

2	MSTR	RW	Master/slave setting. It cannot be modified during communication. 1: Configure as the master device; 0: Configured as a slave device.	0b

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

20.4.2 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

Control Register 2

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved	0
7	TXEIE	RW	Transmit buffer empty interrupt enable. Setting this bit allows an interrupt to be generated when TXE is set.	0
6	RXNEIE	RW	Receive buffer not empty interrupt enable. Setting this bit allows an interrupt to be generated when RXNE is set.	0
5	ERRIE	RW	Error interrupt enable. Setting this bit enables an interrupt to be generated when an error occurs (CRCERR, OVR, MODF).	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: Disable SS output in master mode.	0
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

20.4.3 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	MOD	CRC	UDR	CHSI	TXE	RXN

			F	ERR		D		E

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	RO	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	UDR	RO	Underflow flag. Set by hardware, and reset by software. 1: Underflow occurs; 0: Underflow has not occurred.	0
2	CHSID	RO	Channel. Set by hardware and reset by software. 1: The left channel needs to be transmitted or received; 0: The right channel needs to be transmitted or received.	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: Read DATAR and auto-zero.*	0

20.4.4 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]	DATAR	RW	Data register. It 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 a different area for operations, among which the read pair uses the receive buffer, and the write corresponds to the send buffer. Data reception and transmission can be 8-bit or 16-bit, and it is necessary to determine how many bits of data to use before transmission. When using 8 bits for data transmission, only the lower 8 bits of the data register are used, and the upper 8 bits are forced to 0 when receiving. Using the 16-bit data structure will cause all 16-bit data registers to be used.	0

20.4.5 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.	7

20.4.6 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 to be read when BSY is 0.	0

20.4.7 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	Tx CRC value. Stores the calculated result of the CRC check of the bytes that have been sent. 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 to be read when BSY is 0.	0

20.4.8 SPI_I2S Configuration Register (SPI_I2S_CFGR) (x=1/2/3)

Offset address: 0x1C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				I2S MOD	I2SE	I2SCFG		PCM SYNC	Reserved	I2SSTD		CKPOL	DATLEN		CHL EN

Bit	Name	Access	Description
[15:12]	Reserved	RO	Reserved
11	I2SMOD	RW	I2S mode selection, this bit can only be set when SPI or I2S is disabled. 1: Select I2S mode; 0: Select SPI mode.
10	I2SE	RW	I2S enable, not used in SPI mode. 1: I2S enabled; 0: I2S disabled.

[9:8]	I2SCFG[1:0]	RW	I2S mode selection, this bit can only be set when I2S is disabled:
			00: Slave device transmits;
			01: Slave device receives;
			10: Master device transmit;
			11: Master device receives.

7	PCMSYNC	RW	PCM frame synchronization. This bit is only meaningful when I2SSTD = 11 (using the PCM standard). 1: Long frame synchronization; 0: Short frame sync.
6	Reserved	RO	Reserved
[5:4]	I2SSTD[1:0]	RW	I2S standard selection, this bit can only be set when I2S is disabled. 00: I2S Philips standard; 01: High byte alignment standard (left alignment); 10: Low byte alignment standard (right justified); 11: PCM standard.
3	CKPOL	RW	Quiescent clock polarity, for correct operation, this bit should only be set when I2S is turned off. 1: I2S clock quiescent state is high level;

			0: I2S clock quiescent state is low.

[2:1]	DATLEN	RW	Length of the data to be transmitted. For correct operation, this bit can only be set when I2S is turned off.
			00: 16-bit data length;
			01: 24-bit data length;
			10: 32-bit data length;
			11: Not allowed.

0	CHLEN	RW	Channel length. Only when DATLEN = 00, the write operation of this bit is meaningful, otherwise the channel length is fixed to 32 bits by hardware. 1: 32 bits wide; 0: 16 bits wide.

20.4.9 SPI_I2S Prescaler Register (SPIx_I2SPR) (x=2/3)

Offset Address: 0x20

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						MCKOE	ODD	I2SDIV[7:0]							

Bit	Name	Access	Description
[15:10]	Reserved	RO	Reserved
9	MCKOE	RW	Master clock output enable, for proper operation, this bit can only be set when I2S is turned off. This bit is only used in I2S master mode. 1: Master device clock output enabled; 0: Master clock output disabled.
8	I2SCFG	RW	Odd coefficient prescaler, for correct operation, this bit should only be set when I2S is turned off. This bit is only used in I2S master mode. 1: Actual division factor = (I2SDIV * 2)+1; 0: Actual division factor = I2SDIV *2.
[7:0]	I2SDIV	RW	I2S linear prescaler. For proper operation, this bit should only be set when I2S is turned off. This bit is only used in I2S master mode. Disable setting I2SDIV[7:0] = 0 or I2SDIV[7:0] = 1 See Section 19.3.3.

20.4.10 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															HSRXEN

Bit	Name	Access	Description
[15:1]	Reserved	RO	Reserved

0	HSRXEN	W0	Read enable in SPI high-speed mode (CLK is more than or equal to 36MHz). This mode is valid only when clock is divided by 2 (BR in CTLR1 = 000). This bit cannot be read. 1: High-speed read mode enabled; 0: High-speed read mode disabled.

Chapter 21 USB Full-speed Device Controller (USBD)

The module descriptions in this chapter are applicable to some products of the CH32F2x, CH32V2x and CH32V3x microcontroller series.

The USBD module is a USB full-speed and low-speed protocol communication controller based on the USB2.0 full-speed device technical specification. Built-in hardware automatically handles reverse non-return-to-zero (NRZI) encoding/decoding, bit stuffing of physical signals. The control can drive various states of the USB bus, send and receive protocol packets, and provide functions such as automatic response for flow control to ensure application processing time.

21.1 Main features

- Compliant with USB2.0 full-speed device specification
- Support USB full-speed 12Mbps, low-speed 1.5Mbps mode
- Support configuration of 16 transmission channels
- Supports endpoint address range 0-15
- Support control, interrupt, batch, isochronous transfer
- Double-buffered mechanism supporting bulk/isochronous endpoints
- USB suspend, wake, resume operation
- The hardware automatically performs data PID inversion and transmission flow control
- Frame locked clock pulse generation

Note: USBD and CAN controller share a dedicated 512-byte SRAM area for data transmission and reception in the design, so when using USBD and CAN functions at the same time, this shared area needs to be allocated reasonably to prevent data conflicts.

21.2 Function description

21.2.1 Features

The USBD module provides a communication connection that conforms to the USB specification for the data communication between the USB host (usually a PC) and the microcontroller, and is completed by the application program and the module hardware when used. The module contains a shared 512-byte dedicated SRAM area as a USB transceiver data buffer. The actual use range is determined by the number of configured endpoints and the maximum data packet length of each endpoint, for up to 16 unidirectional or 8 bidirectional endpoints.

USBD module features include:

- Physical signal encoding/decoding: PID detection of token packets, data packets, and handshake packets is implemented according to USB specifications, including bit stuffing, CRC generation and verification, frame header synchronization identification, etc.
- Transaction processing: determine correct transmission and error status, and provide respective flag status and interrupt notification.
- Bus Suspend/Reset/Wakeup status recognition notification.
- Automatic data packet PID: According to the protocol, perform hardware flipping or locking on the sending and receiving data packet PIDs of asynchronous endpoints and synchronous endpoints to reduce

application work.

- Automatic response packet PID: According to the protocol, after a USB transaction is completed, the status of the response packet will be automatically modified for the asynchronous endpoint to provide enough processing and preparation time for the application, but it will not affect the physical transmission and reception on the USB bus.
- Management data sending and receiving: locate the endpoint configuration and buffer description area, and detect the buffer boundary to prevent overflow. Single-buffer/double-buffer area management, interrupt reporting priority management by endpoint type, etc.
- Provide general class, endpoint class, buffer description class register configuration.

Applications can:

- Get the frame interval time point based on USB protocol, bus status: suspended, reset.
- Customize the number of endpoints, endpoint type, endpoint size. Customize the transfer data buffer area.
- Get the current or suspended endpoint's service for processing.
- Get error status like bit stuffing, format, CRC, protocol, missing ACK, buffer overflow/buffer not full, etc.
- The drive module enters a low power consumption mode.

The USBD module maps USB events to 3 different NVIC or PFIC request lines (3 interrupt numbers are used):

1） USB high-priority interrupts: Can only be triggered by correct transfer events for isochronous and double-buffered bulk transfers, in order to guarantee maximum transfer rates.
2） USB low priority interrupt: Can be triggered by all USB events (proper transfer, USB reset, etc.). Firmware should first determine the source of the interrupt before handling the interrupt.
3） USB Wakeup Interrupt: Triggered by a wakeup event from USB suspend mode.

21.2.2 Functional configuration

1） GPIO port

Once the USBD module is enabled, the GPIO ports for UDP and UPM are automatically connected to the internal USB transceiver, and the port settings for its GPIO peripherals are disconnected. Therefore, it is recommended that the GPIO port be configured to output a low level in a push-pull mode to prevent the USB device from being notified in advance when the port is in an uncertain state or when the PC host is connected before the USBD function is turned on.

The USBD module has a built-in 1.5K pull-up resistor in USB device mode, and no external pull-up resistor is required. For specific configuration, please refer to the description of the configuration extension control register (EXTEND_CTR).

2） Module initialization

First, the analog part related to the USB transceiver requires a standard 48MHz clock as the reference clock, which is derived from the AHB bus. The application program needs to ensure that the current USB clock is 48MHz by configuring the corresponding control bit of the clock management logic (RCC_CFGR0 register), and then enable the USB interface clock, so that the program can access the registers of the USBD module.

Second, when the module is forced to reset (the FRES bit in the USBD_CNTR register is 1 by default), the application should initialize the required registers and packet buffer description table. Including: packet buffer description table address register (USBD_BTABLE), endpoint configuration register (USBD_EPRx) and packet buffer description table register. Configure the ADD[6:0] bits in the USBD_DADDR register to be 0 (the default address of the USB protocol), and set the EF bit to enable the endpoint transfer function.

Finally, enable the internal 1.5K pull-up resistor and set the speed mode (EXTEND_CTR register), then, clear the FRES bit on the USBD_CNTR register, cancel the forced reset state of the USBD module to enable the USBD module, and clear the various status flags in the USBD_ISTR register so that Clear unhandled false interrupt flags before enabling operation of any other unit. Turn on the required interrupt control bits in the USBD_CNTR register.

3） USB reset

USB reset includes: USBD module forced reset and USB bus reset (protocol reset). Both will generate the RESET flag in the USBD_ISTR register. When a USB reset occurs, all endpoint communications are disabled (the USBD module will not respond to any packet transfers). After a USB reset, the USBD module is enabled, and the USB endpoint also needs to be enabled to respond to the USB host (the EF bit in the USBD_DADDR register is 1). During the enumeration phase of the USB device, the host will assign a unique address to the device, which must be written into the ADD[6:0] bits in the USBD_DADDR register.

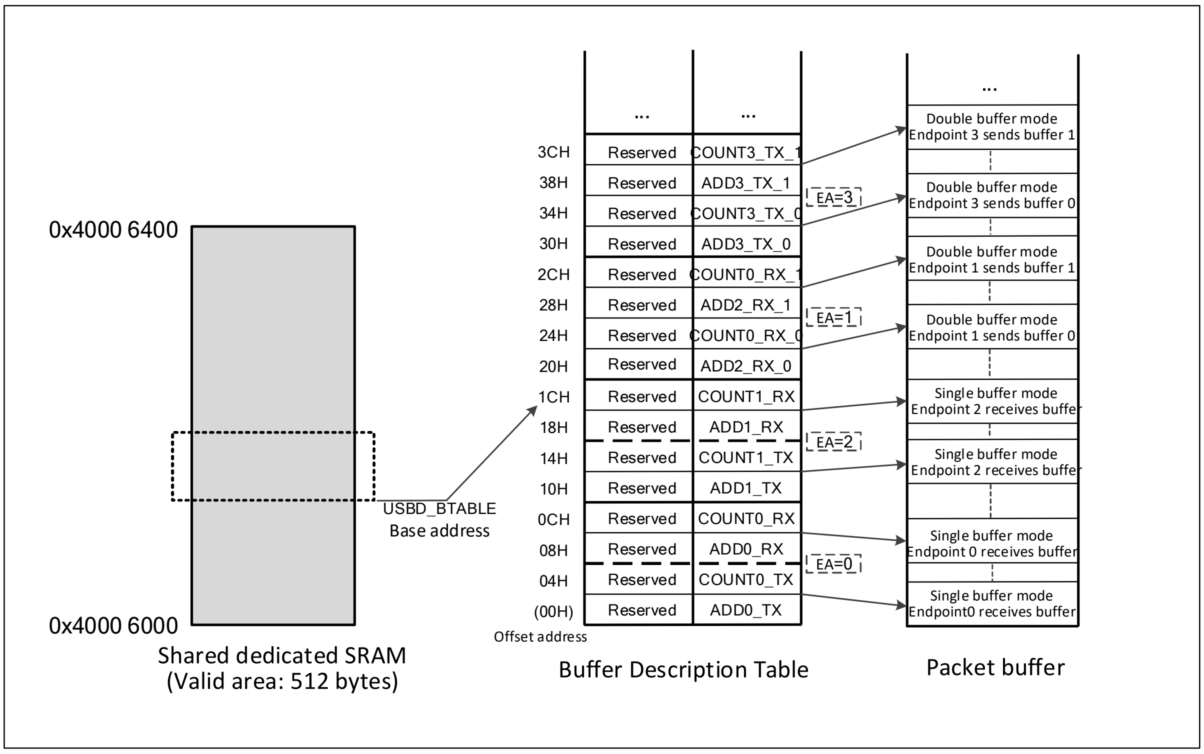
Note: The RESET flag comes from the state of the forced reset control bit (FRES) of the USBD module and the start of the USB bus reset signal.

4） Endpoint configuration and buffer description table

Each endpoint configuration register can configure a bidirectional endpoint single-buffered attribute, or a unidirectional endpoint double-buffered attribute.
For example: configure the bidirectional endpoint single buffer attribute, endpoint configuration register 3 (USBD_EPRx), EA[3:0] is 2, then there can exist endpoint 2 upload channel and endpoint 2 download channel on USB transmission (specifically determined by the descriptor information.); configure the unidirectional endpoint double buffer properties (only for bulk endpoints and synchronous endpoints), endpoint configuration register 3 (USBD_EPRx), EA[3:0] is 2, endpoint type (EPTYPE) is synchronous or bulk endpoint, EP_KIND bit is 1 , then there can exist endpoint 2 upload channel or endpoint 2 downlink channel on USB transmission, choose 1 of 2, the sending and receiving speed is faster than single buffer, the microcontroller processing and USBD module physical sending and receiving can be performed synchronously, reducing waiting time.
Note: The USBD module has a built-in conflict arbitration mechanism, which makes the access of the microcontroller and the USBD module to the packet buffer just like the access to a dual-port SRAM. Even if the microcontroller accesses the buffer continuously, there will be no access conflict.

Each endpoint configuration register corresponds to a set of buffer description class registers (description table) and the corresponding data sending and receiving buffer area, they are all located in the shared 512-byte dedicated SRAM area (base address 0x40006000). Among them, the USBD_BTABLE register defines the starting address of the buffer description table in the SRAM area, and the data transceiver buffer area can be located anywhere in the entire dedicated SRAM area, because their addresses and lengths are defined in the corresponding buffer description table, pay attention to the allocation conflict problem.
Note: When using CAN, the CAN filter table uses the high 256-bytes of the shared 512-byte dedicated SRAM area and the USB uses the low 256-bytes.

Figure 21-1 Buffer description table structure



Whether receiving or sending, the packet buffer is used from the bottom. The USBD module does not change the contents of other buffers beyond the currently allocated buffer area. If the buffer receives a data packet larger than itself, it will only receive data up to its own size, and the others will be discarded, that is, a so-called buffer overflow exception occurs.

1）Endpoint initialization

The first step in initializing an endpoint is to write the appropriate values to the USBD_ADDRx_TX or USBD_ADDRx_RX registers so that the USBD module can find the data to transmit or a buffer that is ready to receive data. The EPTYPE[1:0] bits in the USBD_EPRx register determine the basic type of the endpoint, and the EP_KIND bits determine the special characteristics of the endpoint. As a sender, you need to set the STAT_TX bit of the USBD_EPRx register to enable the endpoint, and configure the COUNTx_TX bit to determine the transmission length. As a receiver, you need to set the STAT_RX bit to enable the endpoint, and set the BL_SIZE and NUM_BLOCK bits to determine the size of the receive buffer to detect buffer overflow exceptions. For unidirectional endpoints with asynchronous non-double-buffered bulk transfers, only one register in the transfer direction needs to be set. Once the endpoint is enabled, the application can no longer modify the value of the USBD_EPRx register and the location of the USBD_ADDRx_TX/USBD_ADDRx_RX, USBD_COUNTx_TX/USBD_COUNTx_RX registers, because these values are modified in real time by the hardware. When the data transfer is completed, the CTR interrupt will be generated, at this time the above registers can be accessed and new transfers can be re-enabled.

2）IN transaction (transmit data)

When receiving an IN token packet, if the received address is consistent with a configured endpoint address, and the STAT_TX bit in the USBD_EPRx register at this time indicates that it can be sent, the USBD module will be based on the contents of the buffer description table and the DTOG_TX bit performs packet encoding to send out packets. If the endpoint corresponding to the received token packet is invalid, a NAK or STALL handshake packet will be sent according to the STAT_TX bit in the USBD_EPRx register instead of a data packet.

After receiving the ACK handshake packet responded by the host, the value of the USBD_EPRx register is updated as follows: the DTOG_TX bit is toggled, the STAT_TX bit is '10' (NAK state), invalidating the endpoint, and the CTR_TX bit is set. The application needs to identify the USB endpoint that generates the interrupt through the EP_ID and DIR bits of the USBD_ISTR register. The interrupt service routine of the CTR_TX event needs to clear the interrupt flag first. If you want to continue sending data (which can be executed whenever data needs to be sent), you need to prepare the data buffer to be sent, and update COUNTx_TX to the number of bytes to be transmitted next time, and finally set the STAT_TX bit to '11' (ACK, the endpoint is valid) to enable data transmission again. When the STAT_TX bit is '10' (NAK state), any IN request sent to the endpoint will be NAKed, and the USB host will resend the IN request until the endpoint acknowledges that the request is valid.

3）OUT transaction and SETUP transaction (receive data)

The USBD module handles these 2 transactions in basically the same way; when an OUT or SETUP packet is received, if the received address matches a configured endpoint address, and if the STAT_RX bit in the USBD_EPRx register at this time indicates that itis ready to receive, the USBD module judges whether the PID of the received data matches according to the DTOG_RX bit. If it matches, it will access the buffer description table, find the ADDRx_RX and COUNTx_RX registers related to the endpoint, and save the received data packet (the one received first is the low byte). Go to the address space defined by ADDRx_RX and detect whether the receive overflows the buffer according to the values of BL_SIZE and NUM_BLOCK. If there are no errors in transmission, send an ACK handshake packet to the host. Even if a CRC error or other type of error occurs (bit stuffing, framing errors, etc.), the data is saved to the packet buffer, at least until the data point where the error occurred, it's just that the ACK handshake is not sent, and the USBD_ISTR register The ERR bit will be set. In this case, the application program usually does not need to intervene in the processing, and the USBD module will automatically recover from the transfer error and be ready for the next transfer. If the endpoint corresponding to the received packet is not ready, the USBD module will send a NAK or STALL handshake packet according to the STAT_RX bit in the USBD_EPRx register, and the data will not be written into the receive buffer.

The value of ADDRx_RX determines the starting address of the receive buffer, and COUNTx_RX determines the size of the receive buffer (expected effective data length + 2-byte CRC). If the length of the received data packet exceeds the range of the buffer, the data beyond the range will not be written into the buffer, the USBD module will report the buffer overflow, send a STALL handshake packet to the host, and set the packet buffer overflow Logo PMAOVR.

If the transfer is completed correctly, the USBD module will send an ACK handshake packet and write the number of valid data bytes in the actual received packet into the COUNTx_RX register. The values of the USBD_EPRx registers are updated as follows: the DTOG_RX bit is toggled, the STAT_RX bit is '10' (NAK state) to invalidate the endpoint, and the CTR_RX bit is set. The application needs to identify the USB endpoint that generates the interrupt through the EP_ID and DIR bits of the USBD_ISTR register. The interrupt service routine of the CTR_RX event must first determine the type of transmission according to the SETUP bit, clear the interrupt flag bit at the same time, and then read the COUNTx_RX register pointed to by the relevant buffer description table entry to obtain the total number of bytes transmitted this time, and process the received data. After processing, the application needs to set the STAT_RX bit in USBD_EPRx to '11' (ACK state) to enable the next transmission. When the STAT_RX bit is '10' (NAK state), any OUT request sent to the endpoint will be NAKed, except for the SETUP request (the protocol specifies that the SETUP request must be received with an ACK handshake). The PC host will continue to resend the NAKed OUT transaction packet until it receives the endpoint's ACK handshake packet.

4）Control Transfer
The control (SETUP) transfer must occur on endpoint 0, so it is also called endpoint 0-bit control endpoint. The control transfer consists of 3 phases, first the SETUP phase in which the host sends a SETUP transaction, then the data phase in which the host sends zero or more data (IN/OUT transactions), and finally the status phase, consisting of the data phase in the opposite direction of the data phase.

The SETUP transaction is very similar to the transmission process of the OUT transaction, so the control endpoint must check the SETUP bit in the USBD_EPRx register every time the CTR_RX interrupt occurs to identify whether it is a normal OUT transaction or a SETUP transaction. When the host sends a SETUP transaction, the USBD module will always reply to the ACK handshake packet and receive it, ignoring the content of STAT_RX and DTOG_RX. Then force DTOG_RX and DTOG_TX to be set to DATA1 state, and set STAT_RX and STAT_TX to '10' (NAK), to ensure that the application can decide whether the subsequent transmission is IN or OUT according to the corresponding data in the SETUP transaction. If subsequent data transmission is rejected or an error occurs, the application can set STAT_RX or STAT_TX to '01' and respond to the STALL handshake packet. If the application receives a SETUP transaction and processes it, CTR_RX remains set at this time, and another SETUP packet is received, the USBD module will discard the SETUP packet and not give any handshake packet response, in order to simulate a reception error, forcing the host to send the SETUP packet again, in order to avoid losing another SETUP transaction transmission immediately following a CTR_RX interrupt.

During the status phase of a control transfer, if an OUT transaction sent by the host to the device is being performed, then the STATUS_OUT bit (EP_KIND in the USBD_EPRx register) should be set, only then if a non-zero length of the transfer is received during the status phase data packets, will generate transmission errors. After completing the status phase transfer, the application should clear the STATUS_OUT bit, and set STAT_RX to ACK to indicate that it is ready to receive a new command request, and set STAT_TX to NAK to not accept any data upload requests.

21.2.3 Dual buffer mechanism
In the USB protocol standard, the application description of different data transmission methods is carried out. Among them, bulk transfer is suitable for bulk data transfer between the USB host and the device, and the host uses as much bandwidth as possible to perform bulk transfer within the frame time. However, this transmission needs to ensure the correctness and integrity of the data, so the transmission includes the sequence of token packets, data packets, and handshake packets. Synchronous transmission is suitable for transmitting data at a constant rate, but has a certain tolerance for errors. It is believed that the transmission can generally be successful. The host has a fixed bandwidth to perform synchronous transmission in each frame time to ensure the transmission rate. Therefore, during transmission Contains token packets, data packets in sequence, without handshake packets to confirm the transmission status and terminate the transmission.

21.2.3.1 One-way dual-buffered bulk endpoint
Bulk transmission, in the single buffer mode, when the application processes the previous data transmission of the batch endpoint, and receives a new data packet, the USBD module will respond to the NAK handshake packet, so that the PC host will continue to resend the same data packet until the application resets the ACK handshake. Such retransmissions take up a lot of bandwidth and affect the rate of bulk transfers. Therefore, the double buffering mechanism is introduced to the bulk endpoint to improve the data transfer rate. In dual buffering mode, a one-way bulk endpoint has 2 data buffers, that is, the receive buffer and the transmit buffer. The data flip bit (DTOG_RX or DTOG_TX) is used to select which one of the 2 buffers is currently used, so

that the application can operate on the other buffer while the USBD module accesses one of the buffers. For example, when transferring an OUT transaction to a double-buffered bulk endpoint, the USBD module saves data from the PC host in a buffer while the application can process data in another buffer (for IN transactions, the case it's the same). In this way, the data processing of the application program is completed by using the time of receiving or sending data of the USBD module, and the efficiency of USB transmission and reception is improved. Because 2 buffers are required for a transmission direction, the bulk endpoint that configures the bidirectional buffer must be configured as a unidirectional endpoint, and its USBD_EPRx register only needs to set the STAT_RX bit (as a double-buffered bulk receive endpoint) or the STAT_TX bit (as a double-buffered endpoint bulk send endpoint). In order to take advantage of double buffering as much as possible and achieve a higher transfer rate, the flow control of the USBD module to handle double- buffered bulk endpoints is slightly different from that of other endpoints. It only sets the endpoint to NAK state when there is an access violation in the buffer, rather than setting the endpoint to NAK state after every successful transfer.

The DTOG_xx bits in the USBD_EPRx register are used to identify the memory buffers currently used by the USBD module and the application respectively to avoid access conflicts. When configured as a unidirectional send double buffer endpoint, DTOG_TX identifies the buffer currently used by the USBD module, and DTOG_RX identifies the buffer currently used by the application; when configured as a unidirectional receive double buffer endpoint, DTOG_RX identifies the buffer currently used by the USBD module area, and DTOG_TX identifies the buffer currently used by the application. We named the USBD module using the buffer ID as DTOG and the application using the buffer ID as SW_BUF. So the double buffered unidirectional bulk endpoint identifier is defined as follows:

Table 21-1 Buffer ID

Buffer Flag	Transmission endpoint	Reception endpoint
DTOG	DTOG_TX (USBD_EPRx register bit6)	DTOG_RX (USBD_EPRx register bit14)
SW_BUF	DTOG_RX (USBD_EPRx register bit14)	DTOG_TX (USBD_EPRx register bit6)

Table21-2 Double-buffered bulk endpoint buffer

Endpoint Type	DTOG	SW_BUF	Buffer used by USBD module	Buffer used by application software
IN Endpoint	0	1	ADDRx_TX_0/COUNTx_TX_0	ADDRx_TX_1/COUNTx_TX_1
	1	0	ADDRx_TX_1/COUNTx_TX_1	ADDRx_TX_0/COUNTx_TX_0
	0	0	Set endpoint in NAK state	ADDRx_TX_0/COUNTx_TX_0
	1	1	Set endpoint in NAK state	ADDRx_TX_1/COUNTx_TX_1
OUT Endpoint	0	1	ADDRx_RX_0/COUNTx_RX_0	ADDRx_RX_1/COUNTx_RX_1
	1	0	ADDRx_RX_1/COUNTx_RX_1	ADDRx_RX_0/COUNTx_RX_0
	0	0	Set endpoint in NAK state	ADDRx_RX_0/COUNTx_RX_0
	1	1	Set endpoint in NAK state	ADDRx_RX_1/COUNTx_RX_1

To configure a double-buffered bulk endpoint, the application needs to set the EPTYPE[1:0] bits in the USBD_EPRx register to '00' and the EP_KIND bit to '1'. The DTOG and SW_BUF bits are initialized according to the buffer used at the beginning of the transfer. After each successful transfer, the USBD module will operate according to the flow control of the double-buffered bulk endpoint and will continue until EP_KIND becomes invalid. At the end of each transfer, either the CTR_RX bit or the CTR_TX bit will be set, depending on the endpoint's transfer direction. At the same time, the hardware will set the corresponding DTOG_xx bit (flip) and implement buffer exchange. If there is no buffer access conflict between the USBD module and the application (i.e. DTOG and SW_BUF are the same value, see Table 154), the status value of the STAT_xx bit is maintained, otherwise it will be set to '10' (NAK status). So after the application accesses the buffer, it needs to flip the SW_BUF bit to notify the USB module that the block buffer has become available.

21.2.3.2 Sync Endpoint

Isochronous transmission is generally used to transmit audio streams, compressed video streams, and other data that have strict requirements on data transmission rates. The endpoint that performs the isochronous transfer is the isochronous endpoint. The USB host will allocate a fixed bandwidth to the synchronous endpoint for IN transaction or OUT transaction transmission in each frame time, and there is no retransmission mechanism, no handshake protocol, and the PID of the transmitted data packet is fixed as DATA0, and DATA0 and DATA1 data rollover mechanism (appears in control/bulk/interrupt transfers)will not appear.

Because there is no handshake mechanism in synchronous transmission, the STAT_RX bit and STAT_TX bit in the USBD_EPRx register can only be set to '00' (transmission disabled) and '11' (running transmission) respectively. Isochronous transfers use a double buffering mechanism to simplify the software process. It also uses 2 buffers to ensure that while the USB module is using one of the buffers, the application can access the other. Different from the double-buffering mechanism of one-way batch endpoints, synchronous endpoints have fixed time intervals and fault tolerance for transmission in the USB standard, so the USBD module does not judge the conflict with the application buffer, and only uses the DTOG bit to identify its current status. The buffer used (DTOG_RX bits in the USBD_EPRx register are used to identify the receive sync endpoint, and DTOG_TX bits are used to identify the transmit sync endpoint).

Table 21-3 Sync endpoint buffer ID

Endpoint Type	DTOG	Buffer used by USBD module	Buffer used by application software
IN endpoint	0	ADDRx_TX_0/COUNTx_TX_0	ADDRx_TX_1/COUNTx_TX_1
	1	ADDRx_TX_1/COUNTx_TX_1	ADDRx_TX_0/COUNTx_TX_0
OUT endpoint	0	ADDRx_RX_0/COUNTx_RX_0	ADDRx_RX_1/COUNTx_RX_1
	1	ADDRx_RX_1/COUNTx_RX_1	ADDRx_RX_0/COUNTx_RX_0

To configure a synchronous endpoint, the application needs to set the EPTYPE[1:0] bits in the USBD_EPRx register to '10'. The DTOG bits are initialized according to the buffer used at the beginning of the transfer. After each successful transfer, either the CTR_RX bit or the CTR_TX bit will be set, depending on the endpoint's transfer direction. At the same time, the hardware will set the corresponding DTOG_xx bits (flipped) to implement the buffer swap, but will not change the expected or sent packet PID (fixed to DATA0). The STAT_RX or STAT_TX bits do not change. In synchronous transmission, even if a CRC error or buffer overflow occurs in the OUT transaction, the transmission is still considered correct, and the CTR_RX interrupt event can be triggered. However, when a CRC error occurs, the hardware will set the ERR bit of the USB_ISTR register to remind the application Program data may be corrupted.

21.2.4 Suspend/Wake process

A bus state is defined in the USB standard - SUSPEND. If the USB bus does not have any activity within 3ms, it enters the suspended state. In this state, the current provided on the USB bus will be reduced (low-speed devices generally do not exceed 500uA, and high-speed devices or devices that support remote wake-up function generally do not exceed 2.5mA). This current limit is critical for bus-powered USB devices, while self-powered devices do not need to strictly adhere to such current consumption limits.

Under normal working conditions, the USB host will send SOF packets at 1ms intervals, so if the USBD module detects 3 consecutive SOF packet loss events, it can determine that the host has issued a suspend request. At this time, it will set the USBD_ISTR register. The SUSP bit, if enabled, also triggers pending interrupts. The USBD module will continuously detect the suspend state of the bus and update the SUSP bit (clearing the SUSP bit flag will still be set again by hardware when the bus is in the suspend state). So when the application receives the USB bus suspend event, it needs to perform the following process:

Set the FSUSP bit in the USBD_CNTR register to 1 to shield the hardware suspend state detection and prevent the suspend event from being triggered continuously.

Eliminate or reduce quiescent current consumption of modules other than the USBD module.

Set the LPMODE bit in the USBD_CNTR register to 1 to put the USBD module in a low-power operating state, but still detect the bus wake-up signal.

The external oscillator and PLL can optionally be turned off to stop any activity on the device.

A USB device or host in a suspend state will be woken up by a wake-up sequence. The so-called wake-up sequence can be initiated by the USB host to wake up the suspended USB device, or it can be triggered by the USB device to wake up the suspended USB host, but the wake-up sequence is finally ended by the USB host. In addition, as a suspended USB device, it needs to be able to detect the function of the RESET signal (bus reset) and perform it as a normal reset operation.

After the suspended USBD module receives the wake-up signal, it will trigger a WKUP interrupt event (channel 42), and set the WKUP bit in the USBD_ISTR register to 1 to automatically clear the LPMODE bit. When the application receives the USB wakeup event, it needs to perform the following process:

Clear the FSUSP bit in the USBD_CNTR register and restart the suspend state detection function of the USB bus;

The external oscillator and PLL can be optionally activated.

Query the RXDP and RXDM bits of the USBD_FNR register to determine what triggers the wake-up event and execute the corresponding software operation.

The USBD module can issue a wake-up sequence to wake up a suspended USB host. In this case, first set the RESUME bit in the USBD_CNTR register to 1, and then clear it to 0 between 1ms-15ms to start the wake-up sequence. After the RESUME bit is cleared, the wake-up process will be done by the host PC (this sequence will continue after the USB host wakes up to wake up other mounted USB devices). The application can query the RXDP and RXDM bits of the USBD_FNR register to determine whether the wake-up is complete.

Note: The RESUME bit can only be set when the USBD module is set to suspend state (set the FSUSP bit in the USB_CNTR register to '1').

Table 21-4 USB Bus Status

RXDP	RXDM	Condition	USB bus status
0	0	>10ms	Bus reset
0	1	>1ms (full-speed device)	Wake-up sequence begins
		>3ms (low-speed device)	Suspended state
1	0	>3ms (full-speed device)	Suspended state
		>1ms (low-speed device)	Wake-up sequence begins
1	1	-	Bus error (or disturbance)

21.3 Register description

The USBD module has the following 3 types of registers:

- Common registers: USBD module control, interrupt related, base address 0x40005C00.
- Endpoint-specific registers: related to endpoint configuration, transmit/receive status, base address 0x40005C00.
- Buffer description register: related to the data transmit/receive buffer, the base address is 0x40006000.

Table 21-5 USBD common registers

Name	Access address	Description	Reset value
R16_USBD_CNTR	0x40005C40	USB control register	0x0003
R16_USBD_ISTR	0x40005C44	USB interrupt status register	0x0000
R16_USBD_FNR	0x40005C48	USB frame number register	0x0XXX
R16_USBD_DADDR	0x40005C4C	USB device address register	0x0000
R16_USBD_BTABLE	0x40005C50	USB packet buffer description table address register	0x0000

Table 21-6 USBD endpoint-specific registers

Name	Access address	Description	Reset value
R16_USBD_EPR0	0x40005C00	USB endpoint configuration register0	0x0000
R16_USBD_EPR1	0x40005C04	USB endpoint configuration register1	0x0000
R16_USBD_EPR2	0x40005C08	USB endpoint configuration register2	0x0000
R16_USBD_EPR3	0x40005C0C	USB endpoint configuration register3	0x0000
R16_USBD_EPR4	0x40005C10	USB endpoint configuration register4	0x0000
R16_USBD_EPR5	0x40005C14	USB endpoint configuration register5	0x0000
R16_USBD_EPR6	0x40005C18	USB endpoint configuration register6	0x0000
R16_USBD_EPR7	0x40005C1C	USB endpoint configuration register7	0x0000

Table 21-7 USBD buffer description registers

Name	Access address	Description	Reset value
R16_USBD_ADDR0_TX	0x40006000+[USBD_BTABLE]	Endpoint transmit buffer address register 0	0x0000
R16_USBD_COUNT0_TX	0x40006004+[USBD_BTABLE]	Endpoint transmit byte count register0	0x0000
R16_USBD_ADDR0_RX	0x40006008+[USBD_BTABLE]	Endpoint receive buffer address register0	0x0000
R16_USBD_COUNT0_RX	0x4000600C+[USBD_BTABLE]	Endpoint Received Data Bytes Register 0	0x0000
R16_USBD_ADDR1_TX	0x40006010+[USBD_BTABLE]	Endpoint transmit buffer address register1	0x0000
R16_USBD_COUNT1_TX	0x40006014+[USBD_BTABLE]	Endpoint transmit byte count register1	0x0000
R16_USBD_ADDR1_RX	0x40006018+[USBD_BTABLE]	Endpoint receive buffer address register1	0x0000

R16_USBD_COUNT1_RX	0x4000601C+[USBD_BTABLE]	Endpoint receive byte count register1	0x0000
R16_USBD_ADDR2_TX	0x40006020+[USBD_BTABLE]	Endpoint transmit buffer address register2	0x0000
R16_USBD_COUNT2_TX	0x40006024+[USBD_BTABLE]	Endpoint transmit byte count register2	0x0000
R16_USBD_ADDR2_RX	0x40006028+[USBD_BTABLE]	Endpoint receive buffer address register2	0x0000
R16_USBD_COUNT2_RX	0x4000602C+[USBD_BTABLE]	Endpoint receive byte count register2	0x0000
R16_USBD_ADDR3_TX	0x40006030+[USBD_BTABLE]	Endpoint transmit buffer address register3	0x0000
R16_USBD_COUNT3_TX	0x40006034+[USBD_BTABLE]	Endpoint transmit byte count register3	0x0000
R16_USBD_ADDR3_RX	0x40006038+[USBD_BTABLE]	Endpoint receive buffer address register3	0x0000
R16_USBD_COUNT3_RX	0x4000603C+[USBD_BTABLE]	Endpoint receive byte count register3	0x0000
R16_USBD_ADDR4_TX	0x40006040+[USBD_BTABLE]	Endpoint transmit buffer address register4	0x0000
R16_USBD_COUNT4_TX	0x40006044+[USBD_BTABLE]	Endpoint transmit byte count register4	0x0000
R16_USBD_ADDR4_RX	0x40006048+[USBD_BTABLE]	Endpoint receive buffer address register4	0x0000
R16_USBD_COUNT4_RX	0x4000604C+[USBD_BTABLE]	Endpoint receive byte count register4	0x0000
R16_USBD_ADDR5_TX	0x40006050+[USBD_BTABLE]	Endpoint transmit buffer address register5	0x0000
R16_USBD_COUNT5_TX	0x40006054+[USBD_BTABLE]	Endpoint transmit byte count register 5	0x0000
R16_USBD_ADDR5_RX	0x40006058+[USBD_BTABLE]	Endpoint receive buffer address register5	0x0000
R16_USBD_COUNT5_RX	0x4000605C+[USBD_BTABLE]	Endpoint receive byte count register5	0x0000
R16_USBD_ADDR6_TX	0x40006060+[USBD_BTABLE]	Endpoint transmit buffer address register6	0x0000
R16_USBD_COUNT6_TX	0x40006064+[USBD_BTABLE]	Endpoint transmit byte count register6	0x0000
R16_USBD_ADDR6_RX	0x40006068+[USBD_BTABLE]	Endpoint receive buffer address register6	0x0000
R16_USBD_COUNT6_RX	0x4000606C+[USBD_BTABLE]	Endpoint receive byte count register6	0x0000
R16_USBD_ADDR7_TX	0x40006070+[USBD_BTABLE]	Endpoint transmit buffer address register7	0x0000
R16_USBD_COUNT7_TX	0x40006074+[USBD_BTABLE]	Endpoint transmit byte count	0x0000

		register7	
R16_USBD_ADDR7_RX	0x40006078+[USBD_BTABLE]	Endpoint receive buffer address register7	0x0000
R16_USBD_COUNT7_RX	0x4000607C+[USBD_BTABLE]	Endpoint receive byte count register7	0x0000

Note: The above buffer description class registers and endpoint configuration registers are used correspondingly. For example: USB endpoint configuration register 0 corresponds to endpoint sending buffer address register 0, endpoint sending data byte count register 0, endpoint receiving buffer address register 0, endpoint receiving data byte count register 0.

21.3.1 USB Control Register (USBD_CNTR)

Offset address: 0x40

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTRM	PMAOVRM	ERRM	WKUPM	SUSPM	RESETM	SOFM	ESOFM	Reserved			RESUME	FSUSP	LPMODE	PDWN	FRES

Bit	Name	Access	Description	Reset value
15	CTRM	RW	Correct transfer interrupt enable: 1: Enable the correct transfer (CTR) interrupt, and generate an interrupt when the corresponding bit of the interrupt register is set to 1; 0: Correct transfer (CTR) interrupt is disabled.	0
14	PMAOVRM	RW	Packet buffer overflow interrupt enable: 1: Enable PMAOVR interrupt and generate an interrupt when the corresponding bit of the interrupt register is set to 1; 0: Disable PMAOVR interrupt.	0
13	ERRM	RW	Error interrupt enable: 1: Enable error interrupt, and generate an interrupt when the corresponding bit of the interrupt register is set to 1; 0: Disable error interrupt.	0
12	WKUPM	RW	Wake-up interrupt enable: 1: Enable wake-up interrupt and generate an interrupt when the corresponding bit of the interrupt register is set to 1; 0: Disable wake-up interrupt.	0
11	SUSPM	RW	Pending interrupt enable: 1: Enable suspend (SUSP) interrupt, and generate an interrupt when the corresponding bit of the interrupt register is set to 1; 0: Suspend (SUSP) interrupt disabled.	0
10	RESETM	RW	USB reset (bus reset or forced reset) interrupt enable: 1: Enable the USB reset interrupt, and generate an interrupt when the corresponding bit of the interrupt register is set to 1; 0: Disable USB reset interrupt.	0

9	SOFM	RW	Start of Frame (SOF) interrupt enable: 1: Enable SOF interrupt and generate an interrupt when the corresponding bit of the interrupt register is set to 1; 0: Disable SOF interrupt.	0
8	ESOFM	RW	Time frame first missing interrupt enable: 1: Enable the ESOF interrupt, and generate an interrupt when the corresponding bit of the interrupt register is set to 1; 0: Disable ESOF interrupt.	0
[7:5]	Reserved	RO	Reserved	0
4	RESUME	RW	Wake-up request control: 1: output wake-up signal; 0: idle state. According to the USB protocol, if this bit remains valid within 1ms to 15ms, the host will wake up the USBD module. *Note: This bit can only be set when the FSUSP bit is 1.*	0
3	FSUSP	RW	Mask suspend detection control: 1: Mask bus suspend state detection. At this time, the clock and static power consumption of the USB analog transceiver are still maintained. If you need to enter a low-power state (bus-powered device), you need to set FSUSP first and then set LPMODE. 0: Enable bus pending state detection. *Note: When there is no data communication (including SOF) on the USB bus for 3ms, the SUSP interrupt will be triggered. At this time, the software must set this bit, otherwise the SUSP interrupt will always be triggered.*	0
2	LPMODE	RW	Low-power mode control: This mode is used to reduce power consumption in USB suspend state. In this mode, except for the power supply of the external pull-up resistor, other static power consumption is turned off, and the system clock will be stopped or reduced to a certain frequency to reduce power consumption. Activity on the USB bus (wakeup event) will clear this bit (can also be cleared by software). 1: low-power mode; 0: Not in low-power mode.	0
1	PDWN	RO	Power down mode: It is used to completely shut down the USB module. The USB module cannot be used when this bit is set. 0: Get out of power down mode; 1: Get into power down mode.	1
0	FRES	RW	Force USB reset control: 1: Forcibly reset the USBD module. The USBD module will remain in reset until software clears this bit. If the USB	1

			reset interrupt is enabled, a reset interrupt will be generated; 0: Clear USB reset.	

21.3.2 USB Interrupt Status Register (USBD_ISTR)

Offset address: 0x44

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTR	PMAOVR	ERR	WKUP	SUSP	RESET	SOF	ESOF	Reserved			DIR	EP_ID[3:0]			

Bit	Name	Access	Description	Reset value
15	CTR	R0	Correct transfer status. This bit is set by hardware after the endpoint has correctly completed a data transfer. The application can identify which endpoint completed the correct data transfer through the DIR and EP_ID bits.	0
14	PMAOVR	RW0	Packet buffer overflow flag. This bit is set by hardware when the microcontroller has not responded to a request to access the USB packet buffer for an extended period of time. The USBD module usually sets this bit when an ACK handshake packet is not sent during reception, or a bit stuffing error occurs during transmission, in both cases the host will require data retransmission. PMAOVR interrupts are not generated during normal data transfers. Since the failed transmission will be retransmitted by the host, the application can accelerate other operations of the device in this interrupted service routine and prepare for retransmission. But this interrupt will not be generated in isochronous transmission (isochronous transmission does not support retransmission) so data may be lost. This bit is readable, write 0 to clear, write 1 to void.	0
13	ERR	RW0	Error flag, this bit is set by hardware when the following errors occur: NANS: No response. The host's reply timed out. CRC: Checksum error. There is an error in the CRC check in the USB packet. BST: Bit stuffing error. A bit stuffing error was detected in the USB data bits. FVIO: Frame format error. Received non-standard frame (like EOP at wrong moment, wrong token, etc.). USB applications can usually ignore these errors because both the USBD module and the host initiate a retransmission mechanism when an error occurs. The interrupt generated by this bit can be used in the	0

			development phase of the application program, and can be used to monitor the transmission quality of the USB bus and identify possible errors that may occur to the user (loose connection cable, severe environmental interference, damaged USB cable, etc.). This bit is readable, write 0 to clear, write 1 to void.	
12	WKUP	RW0	Wake-up signal flag: When the USBD module is in suspend state, this bit will be set by hardware if a wake-up signal is detected. At this time, the LP_MODE bit of the CTLR register will be cleared to 0, and the FSUSP bit needs to be cleared by software to enable the suspension detection. At the same time USB_WAKEUP is activated, notifying other parts of the device (such as the wake-up unit) that the wake-up process will begin. This bit is readable, write 0 to clear, write 1 to void.	0
11	SUSP	RW0	Bus suspend flag: This bit is set by hardware when there is no signal transmission on the USB line for more than 3ms. After the USB reset (bus reset or forced reset) is cancelled, the hardware immediately enables the detection of the suspend signal, but in the suspend mode (FSUSP=1) the hardware will not detect the suspend signal again until the wake-up process ends. This bit is readable, write 0 to clear, write 1 to void.	0
10	RESET	RW0	USB reset (bus reset or forced reset) flag: This bit is set by hardware when the USBD module detects an edge of the USB bus reset signal or a forced reset state. At this time, the USBD module will reset the internal protocol state machine and respond by triggering a reset interrupt when the interrupt is enabled. The transmit and receive parts of the USBD module will be disabled until this bit is cleared. All configuration registers are not reset unless they are cleared by the application. This is used to ensure that the USB transfer can be performed correctly immediately after the reset is reversed. However, the device's address and endpoint registers are reset by a USB reset. This bit is readable, write 0 to clear, write 1 to void.	0
9	SOF	RW0	Start of Frame (SOF) flag: This bit is set by hardware when the USBD module detects a SOF packet on the bus. The interrupt service routine can complete the 1ms synchronization with the host by detecting the SOF event, and correctly read the updated content of the register when it receives the SOF (this function is very meaningful during synchronous	0

			transmission). This bit is readable, write 0 to clear, write 1 to void.	
8	ESOF	RW0	Expected start of frame (ESOF) flag: This bit is set by hardware when the USBD module does not receive the SOF packet on time. The host should send a SOF packet every millisecond, but if the USBD module does not receive it, the suspend timer will trigger this interrupt. If the ESOF interrupt occurs 3 times in a row, that is, the SOF packet is not received 3 times in a row, a SUSP interrupt will be generated. This bit is readable, write 0 to clear, write 1 to void.	0
[7:5]	Reserved	RO	Reserved	0
4	DIR	RO	Transaction data transfer direction. This bit is written by hardware according to the transfer direction after completing the data transfer and generating an interrupt. If DIR=0, the CTR_TX bit of the corresponding endpoint is set, marking the completion of an IN transaction (data transfer from the USBD module to the PC host). If DIR=1, the CTR_RX bit of the corresponding endpoint is set, marking the completion of an OUT transaction (data transfer from the PC host to the USBD module). If the CTR_TX bit is also set at the same time, it indicates that there are both pending OUT and IN transactions. The application can use this information to access the operation corresponding to the USBD_EPnR bit, which indicates the information of the pending interrupt transfer direction.	0
[3:0]	EP_ID[3:0]	RO	Endpoint ID. These bits are written by hardware according to the endpoint ID requested to be interrupted after the USBD module completes the data transfer and generates an interrupt. If there are multiple endpoint request interrupts at the same time, the hardware writes the endpoint ID with the highest priority. The priorities of endpoints are defined as follows: isochronous and double-buffered bulk endpoints have high priority, other endpoints have low priority. If multiple endpoints of the same priority request an interrupt, the priority is determined according to the endpoint ID, that is, endpoint 0 has the highest priority, and the smaller the endpoint ID, the higher the priority. The application can process the interrupt request of the endpoint sequentially through the above-mentioned priority policy.	0

21.3.3 USB Fame Number Register (USBD_FNR)

Offset address: 0x48

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXDP	RXDM	LCK	LSOF[1:0]		FN[10:0]										

Bit	Name	Access	Description	Reset value
15	RXDP	RO	D+ data line level status.	0
14	RXDM	RO	D- data line level status.	0
13	LCK	RO	SOF packet count stop lock. The USBD module will detect SOF packets after the reset or wake-up sequence, if at least 2 SOF packets are detected consecutively, the hardware will set this bit. Once this bit is locked, the frame counter will stop counting and will not resume counting until the USBD module is reset or the bus is suspended.	0
[12:11]	LSOF[1:0]	RO	Frame loss flag. When an ESOF event occurs, the hardware will write the number of lost SOF packets to this field. If the SOF packet is received again, this field is cleared.	X
[10:0]	FN[10:0]	RO	Frame number. This field is the 11-bit frame number in the most recently received SOF packet. Every time the host sends a frame, the frame number will be incremented by itself, which is very meaningful for synchronous transmission. This field is updated when a SOF outage occurs.	X

21.3.4 USB Device Address Register (USBD_DADDR)

Offset Address: 0x4C

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

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved	0
7	EF	RW	USB function enable. This bit is set by the application when the USB device function needs to be enabled. If this bit is 0, the USBD module will stop working, ignore the settings of all registers, and will not respond to any USB communication. 1: Enable USB device function; 0: Stop the USB device function.	0
[6:0]	ADD[6:0]	RW	USB device address. This field is the address value that the USB host assigns to the USB device during enumeration. The address value and the EA bit must match the address information in the USB	0

			token packet in order to perform correct USB transmission at the specified endpoint.	

21.3.5 USB Packet Buffer Description Table Address Register (USBD_BTABLE)

Offset address: 0x50

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

Bit	Name	Access	Description	Reset value
[15:3]	BLABLE[15:3]	RW	Buffer table. This domain is the base address of the packet buffer description table. The packet buffer description table is used to indicate the address and size of the packet buffer of each endpoint, aligned by 8 bytes (i.e. the lowest 3 bits are 000). At the beginning of each transmission, the USBD module reads the packet buffer description table corresponding to the corresponding endpoint to obtain the buffer address and size information.	
[2:0]	Reserved	RO	Reserved	0

21.3.6 USB Endpoint Configuration Register x(USBD_EPRx) (x=0/1/2/3/4/5/6/7)

Offset address: 0x00 to 0x1C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTR_RX	DTOG_RX	STAT_RX[1:0]		SETUP	EPTYPE[1:0]		EP_KIND	CTR_TX	DTOG_TX	STAT_TX[1:0]		EA[3:0]			

Bit	Name	Access	Description	Reset value
15	CTR_RX	RW0	Correct reception flag (OUT/SETUP). This bit is set by hardware when an OUT or SETUP transaction is correctly received (an ACK is sent). If the CTRM bit is set, the corresponding interrupt will be generated and the application needs to clear this bit after handling the event. Whether an OUT transaction or a SETUP transaction was received can be determined by the SETUP bits below. This bit is readable, write 0 to clear, write 1 to have no effect. *Note: This bit is not set for transactions acknowledged with NAK or STALL or for erroneous transfers.*	0
14	DTOG_RX	RW1T	Expect the next received packet PID (OUT/SETUP), hardware settings: 1: expect DATA1; 0: expect DATA0.	0

			For asynchronous endpoints, after receiving the correct PID packet, the USBD module sends an ACK handshake packet, and the hardware automatically flips this bit. For control endpoints, the hardware sets (DATA1) after receiving the correct SETUP packet. For endpoints with double-buffering attributes, in addition to automatically flipping this bit to indicate the expected packet PID, the hardware also supports the exchange of double-buffering according to this bit identification (please refer to the description of the double-buffering mechanism). For synchronous endpoints, the hardware does not judge the PID of the data packet, and only supports the exchange of double buffers through this bit identification. This bit is readable, write 0 is invalid, and write 1 flips. *Note: The application can initialize this bit, or toggle this bit for special purposes.*	
[13:12]	STAT_RX	RW1T	Status bits indicating data reception (in OUT/SETUP transactions): 00: DISABLED, the endpoint ignores all receive requests and does not respond; 01: STALL, the endpoint responds to the request with a STALL packet; 10: NAK, the endpoint responds to the receive request with a NAK packet; 11: ACK, the endpoint responds to the receive request with an ACK packet. When a correct OUT or SETUP data transfer is completed (CTR_RX=1), the hardware will automatically set this bit to the NAK state, so that the application has enough time to process and respond to the next transaction. For double-buffered bulk endpoints, due to the use of a special transfer flow control policy, the transfer state is controlled according to the buffer state used (see Double-buffered endpoints). For isochronous endpoints, hardware will not set this bit after a proper transfer since the endpoint state can only be active or disabled. If this field is set to STALL or NAK, the action that the USBD module responds to is undefined. This field is readable, writing 0 to the bit is invalid, and writing 1 to flip it. *Note: The application program can initialize the field bits.*	0
11	SETUP	RO	SETUP transaction transfer completion flag: 1: It is a SETUP transaction and received correctly (send ACK response);	0

			0: Non-SETUP transaction. *Note: Hardware will only modify this bit when CTR_RX=0.*	

[10:9]	EPTYPE	RW	Transport Endpoint Type:
			00: BULK, bulk endpoint;
			01: CONTROL, control endpoint;
			10: ISO, sync endpoint;
			11: INTERRUPT, interrupt endpoint.
			Only control endpoints will have SETUP transfers, other types of endpoints ignore such transfers. SETUP transmission cannot respond with NAK or STALL packet. If the control endpoint is in NAK state when receiving the SETUP packet, the USBD module will not respond to the request, and a reception error will occur. If the control endpoint is in the STALL state, the SETUP packet will be received correctly, the data will be transmitted correctly, and an interrupt will be generated for the completion of the correct transmission. The OUT packet of the control endpoint is handled in the same way as a normal endpoint. Bulk endpoints and interrupt endpoints are handled in a very similar way, with the only difference being the handling of the EP_KIND bit.
				0

8	EP_KIND	RW	Endpoint special type control bits (used with EP_TYPE):	0

EPTYPE[1:0]	EP_KIND
BULK	DBL_BUF: Enable double buffering.
CONTROL	STATUS_OUT: Control the data packet length judgment in the transmission status stage.
ISO	Unused.
INTERRUTP	Unused.

			DBL_BUF: Setting this bit enables double buffering of bulk endpoints. STATUS_OUT: Setting this bit indicates that the USB device expects the host to send the status phase transaction in the control transfer. At this time, the device responds to the STALL handshake packet for any data packet whose length is not 0. (This feature is only used for control endpoints and is useful for providing detection of protocol layer errors.) If the STATUS_OUT bit is cleared, an OUT transaction in the status phase can contain data of any length.	
7	CTR_TX	RW0	Correct transmit flag (IN): This bit is set by hardware when a correct IN transaction (ACK received) is complete. If the CTRM bit has been set, the corresponding interrupt will be generated, and the	0

			application needs to clear this bit after processing the event. At the end of the IN packet, this bit will not be set if the host responds with a NAK or STALL because the data transfer was unsuccessful. This bit is readable, write 0 to clear, write 1 to have no effect. *Note: This bit is not set if the host responds with a NAK or STALL.*	
6	DTOG_TX	RW1T	Packet PID(IN) to send, hardware setting: 1: send DATA1; 0: Send DATA0. For asynchronous endpoints, after sending the correct PID packet, if the USBD module receives the ACK handshake packet from the host, the hardware automatically flips this bit. For control endpoints, the hardware sets (DATA1) after receiving the correct SETUP packet. For endpoints with double-buffering attributes, in addition to automatically flipping this bit to indicate the PID of the sent data packet, the hardware also supports the exchange of double-buffering according to this bit identification (please refer to the description of the double-buffering mechanism). For isochronous endpoints, the hardware forces the transmission of the data packet DATA0, and this bit indicates that the exchange of double buffers is supported. This bit is readable, writing 0 is invalid, and writing 1 flips. *Note: The application can initialize this bit, or toggle this bit for special purposes.*	0
[5:4]	STAT_TX	RW1T	Status of the transmitted data (in the IN transaction): 00: DISABLED, the endpoint ignores all sending requests and does not respond; 01: STALL, the endpoint responds to the host IN request with a STALL packet; 10: NAK, the endpoint responds to the host IN request with a NAK packet; 11: ACK, the endpoint can send data. When an IN transaction data transmission is completed correctly (CTR_TX=1), the hardware will automatically set this bit to NAK state to ensure that the application has enough time to process and respond to the next transaction transmission. For double-buffered bulk endpoints, due to the use of a special transfer flow control policy, the transfer state is controlled according to the buffer state used (see Double-buffered endpoints).	0

			For isochronous endpoints, hardware will not set this bit after a proper transfer since the endpoint state can only be active or disabled. If this field is set to STALL or NAK, the operation that the USBD module responds to is undefined. This field is readable, writing 0 to the bit is invalid, and writing 1 to flip it. *Note: The application program can initialize the field bits.*	
[3:0]	EA	RW	Endpoint address domain (set the endpoint ID): The application program sets an endpoint address for this endpoint configuration register.	0

21.3.7 Endpoint Transmit Buffer Address Register x(USBD_ADDRx_TX) (x=0/1/2/3/4/5/6/7)

Offset address: [USBD_BTABLE] + x*16

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

Bit	Name	Access	Description	Reset value
[15:1]	ADDRx_TX	RW	Start address of the data buffer to be sent (in an IN transaction).	0
0	-	RZ	The address of the buffer must be 2-byte aligned, so this bit must be 0.	0

21.3.8 Endpoint Transmit Byte Count Register x(USBD_COUNTx_TX) (x=0/1/2/3/4/5/6/7)

Offset address: [USBD_BTABLE] + x*16 + 4

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

Bit	Name	Access	Description	Reset value
[15:10]	Reserved	RO	Reserved	0
[9:0]	COUNTx_TX[9:0]	RW	The number of bytes of data length to be sent (in an IN transaction).	0

Note: There are 2 USBD_ADDRx_TX registers and 2 USB_COUNTx_TX registers for double-buffered and synchronous IN endpoints: USBD_ADDRx_TX_1 and USBD_ADDRx_TX_0, USB_COUNTx_TX_1 and USB_COUNTx_TX_0 respectively, the contents are as follows:

USBD_ADDRx_TX is mapped to USBD_ADDRx_TX_0

USBD_ADDRx_RX is mapped to USBD_ADDRx_TX_1

USBD_COUNTx_TX is mapped to USB_COUNTx_TX_0

USBD_COUNTx_RX is mapped to USB_COUNTx_TX_1

21.3.9 Endpoint Receive Buffer Address Register x(USBD_ADDRx_RX) (x=0/1/2/3/4/5/6/7)

Offset address: [USBD_BTABLE] + x*16 + 8

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

Bit	Name	Access	Description	Reset value
[15:1]	ADDRx_RX[15:1]	RW	Start address of the data buffer to be received (in an OUT or SETUP transaction).	0
0	-	RZ	The address of the buffer must be 2-byte aligned, so this bit must be 0.	0

21.3.10 Endpoint Receive Byte Count Register x(USBD_COUNTx_RX) (x=0/1/2/3/4/5/6/7)

Offset address: [USBD_BTABLE] + x*16 + 12

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

Bit	Name	Access	Description	Reset value
15	BLSIZE	RW	Memory block size: 0: The block size is 2 bytes, used with NUM_BLOCK, the receive buffer range can be allocated 2-62 bytes; 1: The block size is 32 bytes. When used with NUM_BLOCK, the receive buffer range can be allocated from 32 to 512 bytes.	0
[14:10]	NUM_BLOCK [4:0]	RW	The number of memory blocks.	0
[9:0]	COUNTx_RX [9:0]	RO	The number of bytes of data length actually received by the endpoint (in an OUT or SETUP transaction).	X

Note: There are 2 USBD_ADDRx_RX registers and 2 USB_COUNTx_RX registers for double buffer and synchronous IN endpoints: USBD_ADDRx_RX_1 and USBD_ADDRx_RX_0, USB_COUNTx_RX_1 and USB_COUNTx_RX_0 respectively. The contents are as follows:

USBD_ADDRx_TX is mapped to USBD_ADDRx_RX_0

USBD_ADDRx_RX is mapped to USBD_ADDRx_RX_1

USBD_COUNTx_TX is mapped to USB_COUNTx_RX_0

USBD_COUNTx_RX is mapped to USB_COUNTx_RX_1

The upper 6 bits of the USBD_COUNTx_RX register define the size of the receive packet buffer so that the USBD module can detect the overflow boundary of the buffer. The size of the buffer can be expressed according to the parameter maxPacketSize in the endpoint descriptor during the device enumeration process.

Table 20-8 Buffer size definition

NUM_BLOCK[4:0]	Receive Buffer Limit Size	
	BLSIZE = 0	BLSIZE = 1
00000	not allowed	32 bytes
00001	2 bytes	64 bytes
00010	4 bytes	96 bytes
00011	6 bytes	128 bytes
…	…	…
01111	30 bytes	512 bytes
10000	32 bytes	Reserved
…	…	…
11110	60 bytes	Reserved
11111	62 bytes	Reserved

Chapter 22 USB Host/Device Controller (USBHD)

The module descriptions in this chapter are applicable to some products of the CH32F2x, CH32V2x and CH32V3x microcontroller series.

22.1 USB controller introduction

Embedded USB2.0 controller and USB-PHY, with dual roles of host controller and USB device controller. When used as a host controller, it supports low-speed, full-speed and high-speed USB devices/HUBs. When used as a device controller, it can be flexibly set to low-speed, full-speed or high-speed mode to suit various applications.

USB Controller features include:

- Supports USB Host function and USB Device function.
- In the host mode, the downlink port is supported to connect to the high-speed/full-speed HUB.
- In device mode, it supports USB2.0 high-speed 480Mbps, full-speed 12Mbps or low-speed 1.5Mbps.
- Supports USB control transfer, bulk transfer, interrupt transfer and isochronous/real-time transfer.
- Support DMA to directly access the data of each endpoint buffer.
- Support suspend, wake-up/remote wake-up.
- Endpoint 0 supports data packets with a maximum size of 64 bytes. Except for device endpoint 0, other endpoints support data packets with a maximum size of 1024 bytes, and all support double buffering.

22.2 Register description

The USB registers are divided into 3 parts, and some registers are multiplexed in the host and device modes.

- USB global registers
- USB device control registers
- USB host control registers

22.2.1 Global register description

Table 22-1 USBHD registers

Name	Access address	Description	Reset value
R8_USB_CTRL	0x40023400	USB control register	0x06
R8_USB_INT_EN	0x40023402	USB interrupt enable register	0
R8_USB_DEV_AD	0x40023403	USB device address register	0
R16_USB_FRAME_NO	0x40023404	USB frame number register	0
R8_USB_SUSPEND	0x40023406	USB suspend register	0
R8_USB_SPPED_TYPE	0x40023408	USB current speed type register	0
R8_USB_MIS_ST	0x40023409	USB miscellaneous status register	xx10_1000b
R8_USB_INT_FG	0x4002340A	USB interrupt flag register	0
R8_USB_INT_ST	0x4002340B	USB interrupt status register	00xx_xxxxb
R16_USB_RX_LEN	0x4002340C	USB receive length register	xx

22.2.1.1 USB Control Register (R8_USB_CTRL)

Bit	Name	Access	Description	Reset value
7	RB_UC_HOST_MODE	RW	USB operating mode selection: 1: Host mode; 0: Device mode.	0

[6:5]	RB_UC_SPEED_TYPE	RW	USB bus signal transfer rate selection:
			00: Full-speed;
			01: High-speed;
			10: Low-speed.
				0

4	RB_UC_DEV_PU_EN	RW	USB device enable and internal pull-up resistor control in device mode: 1: Enable USB device transfer and enable internal pull-up resistor; 0: Not enabled.	0
3	RB_UC_INT_BUSY	RW	Auto suspend enable before USB transfer completed interrupt flag is not cleared: 1: Automatically suspend before UIF_TRANSFER is cleared, automatically respond busy NAK in device mode, and automatically suspend subsequent transfers in host mode; 0: Not pause.	0
2	RB_UC_RESET_SIE	RW	USB protocol processor software reset control: 1: Forcibly reset the USB protocol processor (SIE), which needs to be cleared by software; 0: No reset. After this bit is cleared, PB6/PB7 automatically switches to USBIO mode.	1
1	RB_UC_CLR_ALL	RW	USB FIFO and interrupt flag clear: 1: Clear the USB interrupt flag and FIFO, which needs to be cleared by software; 0: Not clear.	1
0	RB_UC_DMA_EN	RW	Enable USB DMA, this bit must be set to 1 in normal transfer mode: 1: Enable DMA function and DMA interrupt; 0: Disable DMA.	0

22.2.1.2 USB Interrupt Enable Register (R8_USB_INT_EN)

Bit	Name	Access	Description	Reset value
7	RB_UIE_DEV_NAK	RW	USB device mode, receive NAK interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0

6	RB_UIE_ISO_ACT	RW	Isochronous transfer starts sending/receiving data interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
5	RB_UIE_SETUP_ACT	RW	SETUP transaction completion interrupt: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
4	RB_UIE_FIFO_OV	RW	FIFO overflow interrupt: 1: Enable interrupt; 0: Disable interrupt.	0
3	RB_UIE_SOF_ACT	RW	USB host mode, SOF timing interrupt: 1: Enable interrupt; 0: Disable interrupt. USB device mode, when enabled, receive SOF packets to generate a transfer completion interrupt	0
2	RB_UIE_SUSPEND	RW	USB bus suspend or wakeup event interrupt: 1: Enable interrupt; 0: Disable interrupt.	0
1	RB_UIE_TRANSFER	RW	USB transfer (excluding SETUP transaction) complete interrupt: 1: Enable interrupt; 0: Disable interrupt.	0
0	RB_UIE_DETECT	RW	USB host mode, USB device connect or disconnect event interrupt: 1: Enable interrupt; 0: Disable interrupt.	0
	RB_UIE_BUS_RST	RW	USB device mode, USB bus reset event interrupt: 1: Enable interrupt; 0: Disable interrupt.	0

22.2.1.3 USB Device Address Register (R16_USB_DEV_AD)

Bit	Name	Access	Description	Reset value
7	Reserved	RO	Reserved	0
[6:0]	RB_MASK_USB_ADDR	RW	In host mode, it is the address or HUB address of the currently operating USB device; In device mode, it is USB own address.	0

22.2.1.4 USB Frame Number Register (R16_USB_FRAME_NO)

Bit	Name	Access	Description	Reset value
[15:0]	USB_FRAME_NO	RO	Frame number. In host mode, it indicates the frame number of the SOF packet to be sent. In device mode, it indicates the frame number of the currently received SOF packet. Among them, the lower 11 bits are the effective frame number, and the upper 3 bits are the micro frame number of the high-speed mode.	0

Note: USB_FRAME_NO is a 16-bit register, in which the lower 11 bits represent the frame number of the SOF packet, and the upper 3 bits represent the current micro-frame, which can be used for interrupt, synchronization/real-time transmission under the operation of high-speed HUB.

22.2.1.5 USB Suspend Register (R8_USB_SUSPEND)

Bit	Name	Access	Description	Reset value
[7:6]	Reserved	RO	Reserved	0
[5:4]	RB_USB_LINESTATE	RO	PHY Linestate signal	X
3	Reserved	RO	Reserved	0
2	RB_USB_WAKEUP_ST	RO	In the suspend state, if a wake-up signal from the host is received, this bit is set until the suspend state is exited.	X
[1:0]	RB_USB_SYS_MOD	RW	Test mode in Host mode	00b

Note: When remote wake-up is required, pull the RB_UH_REMOTE_WKUP bit high and then low.

22.2.1.6 USB Speed Type Register (R8_USB_SPEED_TYPE)

Bit	Name	Access	Description	Reset value
[7:2]	Reserved	RO	Reserved	0

[1:0]	RB_USB_SPEED_TYPE	RO	In host mode, it indicates the speed type of the currently connected device; in device mode, it indicates the speed type of the current device;
			00: Full-speed;
			01: High-speed;
			10: Low-speed.
				00b

Note: Different from RB_UC_SPEED_TYPE in the R8_USB_CTRL register, RB_UC_SPEED_TYPE represents the expected maximum speed. Assuming that in device mode, set RB_UC_SPEED_TYPE to high-speed, when the device is connected to a full-speed host, the actual speed type is full-speed. By querying R8_USB_SPEED_TYPE registers can be known. In host mode, set RB_UC_SPEED_TYPE to high-speed. When a full-speed device is connected, the actual communication speed is full-speed, which can be known by querying the R8_USB_SPEED_TYPE register.

22.2.1.7 USB Miscellaneous Status Register (R8_USB_MIS_ST)

Bit	Name	Access	Description	Reset value
7	RB_UMS_SOF_PRES	RO	SOF packet presage status in USB host mode: 1: SOF packet is about to be sent, if there are other USB packets at this time, it will be automatically delayed; 0: No SOF packet is sent.	X

6	RB_UMS_SOF_ACT	RO	SOF packet transfer status in USB host mode: 1: SOF packet is being sent; 0: Send complete or idle.	X
5	RB_UMS_SIE_FREE	RO	USB protocol handler free: 1: The protocol device is free; 0: Busy, USB transfer in progress.	1
4	RB_UMS_R_FIFO_RDY	RO	USB receive FIFO data ready: 1: The receive FIFO is not empty; 0: The receive FIFO is empty.	0
3	RB_UMS_BUS_RESET	RO	USB bus reset: 1: The current USB bus is reset; 0: The current USB bus is not reset.	X
2	RB_UMS_SUSPEND	RO	USB suspend: 1: The USB bus is in a suspended state, and there is no USB activity for a period of time; 0: The USB bus is in a non-suspend state.	0
1	RB_UMS_DEV_ATTACH	RO	USB device attach status for the port in USB host mode: 1: The port has been connected to a USB device; 0: The port has no USB device connected.	0
0	RB_UMS_SPLIT_CAN	RO	In USB host mode, SPLIT packet transmit enable: 1: Allow sending SPLIT packets; 0: Disable sending	0

22.2.1.8 USB Interrupt Flag Register (R8_USB_INT_FG)

Bit	Name	Access	Description	Reset value
7	Reserved	RO	Reserved	0
6	RB_UIF_ISO_ACT	RO	Synchronous transmission starts to send/receive data interrupt flag, write 1 to clear: 1: Start sending/receiving data trigger; 0: No event. *Note: For receiving, if the CRC16 error is received, UIF_TRANSFER will not be generated, otherwise UIF_TRANSFER will still be generated after the transaction is completed.*	0
5	RB_ UIF_SETUP_ACT	RO	SETUP transaction completion interrupt flag, write 1 to clear: 1: The SETUP transaction is completed;	1

			0: No event.	
4	RB_UIF_FIFO_OV	RW	USB FIFO overflow interrupt flag, write 1 to clear: 1: FIFO overflow trigger; 0: no event.	0
3	RB_UIF_HST_SOF	RW	SOF timer interrupt flag in USB host mode, write 1 to clear: 1: SOF packet transmission completion trigger; 0: no event.	0
2	RB_UIF_SUSPEND	RW	USB bus suspend or wake-up event interrupt flag, write 1 to clear: 1: Triggered by USB suspend event or wake-up event; 0: No event.	0
1	RB_UIF_TRANSFER	RW	USB transfer completion interrupt flag, write 1 to clear: 1: A USB transfer completion trigger; 0: No event.	0
0	RB_UIF_DETECT	RW	In USB host mode, USB device connect or disconnect event interrupt flag, write 1 to clear: 1: Detection USB device connection or disconnection trigger; 0: No event.	0
	RB_UIF_BUS_RST	RW	In USB device mode, USB bus reset event interrupt flag, write 1 to clear: 1: USB bus reset event trigger; 0: No event.	0

22.2.1.9 USB Interrupt Status Register (R8_USB_INT_ST)

Bit	Name	Access	Description	Reset value
7	RB_UIS_IS_NAK	RO	NAK response status in USB device mode, same as RB_U_IS_NAK: 1: Respond to NAK during the current USB transfer process; 0: No NAK response.	0
6	RB_UIS_TOG_OK	RO	After the USB transaction is received, the Toggle of the received data packet and the set expected value Match Status Bits: 1: toggle match; 0: toggle does not match.	0
[5:4]	MASK_UIS_TOKEN	RO	In device mode, the token PID identifier of the current USB transfer transaction.	XXb

[3:0]	MASK_UIS_ENDP	RO	In device mode, the endpoint number of the current USB transfer transaction.	XXXXb
	MASK_UIS_H_RES	RO	In host mode, the response PID identifier of the current USB transfer transaction, 0000 means the device has no response or timed out; other values indicate the response PID.	XXXXb

Note: MASK_UIS_TOKEN is used to identify the token PID of the current USB transfer transaction in USB device mode: 00 means OUT packet; 01 means SOF packet; 10 means IN packet; 11 means SETUP packet. MASK_UIS_H_RES is only valid in host mode. In the host mode, if the host sends the OUT/SETUP token packet, the PID is the handshake packet ACK/NAK/STALL, or the device has no response/timeout. If the host sends an IN token packet, the PID is the PID of the data packet (DATA0/DATA1) or the PID of the handshake packet.

22.2.1.10 USB Receive Length Register (R16_USB_RX_LEN)

Bit	Name	Access	Description	Reset Value
[15:0]	R16_USB_RX_LEN	RO	The current number of data bytes received by the USB endpoint	X

22.2.2 Device register description

In USB device mode, the USBHD module provides a total of 16 groups of bidirectional endpoints from endpoints 0 to 15. The maximum packet length of all endpoints except endpoint 0 is 1024 bytes, and the maximum packet length of endpoint 0 is 64 bytes.

- Endpoint 0 is the default endpoint, which supports control transmission. Sending and receiving share a 64-byte data buffer.
- Endpoints 1-15 each include a sending endpoint IN and a receiving endpoint OUT, each with an independent data buffer for sending and receiving, supporting bulk transfer, interrupt transfer and real-time/synchronous transfer.
- Endpoint 0 has an independent DMA address, which is shared for sending and receiving. Endpoints 1 to 15 each have a DMA address for sending and receiving. By entering the R32_UEPn_BUF_MOD register, the mode of the data buffer can be set to double buffer or single buffer. If using double buffer mode, the endpoint can only use one-way transfers.
- Each group of endpoints has transceiver control registers R8_UEPn_TX_CTRL, R8_UEPn_RX_CTRL and transmit length registers R16_UEPn_T_LEN and R32_UEPn_*¬_DMA (n=0~15), which are used to configure the synchronization trigger bits of the endpoint, responses to OUT transactions and IN transactions, and transmit data length, etc.

The USB bus pull-up resistor necessary as a USB device can be set by software at any time to enable or not. When RB_UC_DEV_PU_EN in the USB control register R8_USB_CTRL is set to 1, the controller is set according to the speed of RB_UC_SPEED_TYPE, which is the DP/DM pin of the USB bus internally. Connect a pull-up resistor and enable the USB device function.

When a USB bus reset, USB bus suspend or wake-up event is detected, or when the USB successfully processes data transmission or data reception, the USB protocol processor will set the corresponding interrupt flag. If the interrupt enable is turned on, it will also generate corresponding interrupt request. The application can query directly or query and analyze the interrupt flag register R8_USB_INT_FG in the USB interrupt service routine, and perform corresponding processing according to RB_UIF_BUS_RST and RB_UIF_SUSPEND; and, if RB_UIF_TRANSFER is valid, then it is necessary to continue to analyze the USB interrupt status register R8_USB_INT_ST, according to the current endpoint number MASK_UIS_ENDP and the current transaction token PID identify MASK_UIS_TOKEN for corresponding processing. If the synchronization trigger bit RB_UEP_R_TOG of the OUT transaction of each endpoint is set in advance, then RB_U_TOG_OK or RB_UIS_TOG_OK can be used to judge whether the synchronization trigger bit of the currently received data packet matches the synchronization trigger bit of the endpoint. If the data is synchronized, the data Valid; if the data is out of sync, the data should be discarded. After each USB transmission or reception interrupt is processed, the synchronization trigger bit of the corresponding endpoint should be modified correctly to detect whether the next data packet sent or received next time is synchronously detected; in addition, setting RB_UEP_T_TOG_AUTO or RB_UEP_R_TOG_AUTO enables automatic modification of the corresponding synchronization trigger bit (flip or self-decrease) after successful transmission or successful reception.

The data to be sent by each endpoint is in its own buffer, and the length of the data to be sent is independently set in R16_UEPn_T_LEN; the data received by each endpoint is in its own buffer, but the length of the received data is in the USB receive length. In the register R16_USB_RX_LEN, it can be distinguished according to the current endpoint number when the USB receives an interrupt.

Table 22-2 Device registers

Name	Access address	Description	Reset value
R32_UEP_CONFIG	0x40023410	Endpoint enable configuration register	00000000h
R32_UEP_TYPE	0x40023414	Endpoint type configuration register	00000000h
R32_UEP_BUF_MOD	0x40023418	Endpoint buffer mode register	00000000h
R32_UEP0_DMA	0x4002341C	Endpoint0 buffer start address	xxxxh
R32_UEP1_RX_DMA	0x40023420	Endpoint1 receive buffer start address	xxxxh
R32_UEP2_RX_DMA	0x40023424	Endpoint2 receive buffer start address	xxxxh
R32_UEP3_RX_DMA	0x40023428	Endpoint3 receive buffer start address	xxxxh
R32_UEP4_RX_DMA	0x4002342C	Endpoint4 receive buffer start address	xxxxh
R32_UEP5_RX_DMA	0x40023430	Endpoint5 receive buffer start address	xxxxh
R32_UEP6_RX_DMA	0x40023434	Endpoint6 receive buffer start address	xxxxh
R32_UEP7_RX_DMA	0x40023438	Endpoint7 receive buffer start address	xxxxh
R32_UEP8_RX_DMA	0x4002343C	Endpoint8 receive buffer start address	xxxxh
R32_UEP9_RX_DMA	0x40023440	Endpoint9 receive buffer start address	xxxxh
R32_UEP10_RX_DMA	0x40023444	Endpoint10 receive buffer start address	xxxxh
R32_UEP11_RX_DMA	0x40023448	Endpoint11 receive buffer start address	xxxxh
R32_UEP12_RX_DMA	0x4002344C	Endpoint12 receive buffer start address	xxxxh
R32_UEP13_RX_DMA	0x40023450	Endpoint13 receive buffer start address	xxxxh
R32_UEP14_RX_DMA	0x40023454	Endpoint14 receive buffer start address	xxxxh
R32_UEP15_RX_DMA	0x40023458	Endpoint15 receive buffer start address	xxxxh
R32_UEP1_TX_DMA	0x4002345C	Endpoint1 transmit buffer start address	xxxxh
R32_UEP2_TX_DMA	0x40023460	Endpoint2 transmit buffer start address	xxxxh
R32_UEP3_TX_DMA	0x40023464	Endpoint3 transmit buffer start address	xxxxh
R32_UEP4_TX_DMA	0x40023468	Endpoint4 transmit buffer start address	xxxxh

R32_UEP5_TX_DMA	0x4002346C	Endpoint5 transmit buffer start address	xxxxh
R32_UEP6_TX_DMA	0x40023470	Endpoint6 transmit buffer start address	xxxxh
R32_UEP7_TX_DMA	0x40023474	Endpoint7 transmit buffer start address	xxxxh
R32_UEP8_TX_DMA	0x40023478	Endpoint8 transmit buffer start address	xxxxh
R32_UEP9_TX_DMA	0x4002347C	Endpoint9 transmit buffer start address	xxxxh
R32_UEP10_TX_DMA	0x40023480	Endpoint10 transmit buffer start address	xxxxh
R32_UEP11_TX_DMA	0x40023484	Endpoint11 transmit buffer start address	xxxxh
R32_UEP12_TX_DMA	0x40023488	Endpoint12 transmit buffer start address	xxxxh
R32_UEP13_TX_DMA	0x4002348C	Endpoint13 transmit buffer start address	xxxxh
R32_UEP14_TX_DMA	0x40023490	Endpoint14 transmit buffer start address	xxxxh
R32_UEP15_TX_DMA	0x40023494	Endpoint15 transmit buffer start address	xxxxh
R16_UEP0_MAX_LEN	0x40023498	Endpoint0 maximum length packet register	xxxxh
R16_UEP1_MAX_LEN	0x4002349C	Endpoint1 maximum length packet register	xxxxh
R16_UEP2_MAX_LEN	0x400234A0	Endpoint2 maximum length packet register	xxxxh
R16_UEP3_MAX_LEN	0x400234A4	Endpoint3 maximum length packet register	xxxxh
R16_UEP4_MAX_LEN	0x400234A8	Endpoint4 maximum length packet register	xxxxh
R16_UEP5_MAX_LEN	0x400234AC	Endpoint5 maximum length packet register	xxxxh
R16_UEP6_MAX_LEN	0x400234B0	Endpoint6 maximum length packet register	xxxxh
R16_UEP7_MAX_LEN	0x400234B4	Endpoint7 maximum length packet register	xxxxh
R16_UEP8_MAX_LEN	0x400234B8	Endpoint8 maximum length packet register	xxxxh
R16_UEP9_MAX_LEN	0x400234BC	Endpoint9 maximum length packet register	xxxxh
R16_UEP10_MAX_LEN	0x400234C0	Endpoint10 maximum length packet register	xxxxh
R16_UEP11_MAX_LEN	0x400234C4	Endpoint11 maximum length packet register	xxxxh
R16_UEP12_MAX_LEN	0x400234C8	Endpoint12 maximum length packet register	xxxxh
R16_UEP13_MAX_LEN	0x400234CC	Endpoint13 maximum length packet register	xxxxh
R16_UEP14_MAX_LEN	0x400234D0	Endpoint14 maximum length packet register	xxxxh
R16_UEP15_MAX_LEN	0x400234D4	Endpoint15 maximum length packet register	xxxxh
R16_UEP0_T_LEN	0x400234D8	Endpoint0 transmit length register	xxxxh
R8_UEP0_TX_CTRL	0x400234DA	Endpoint0 transmit control register	00h

R8_UEP0_RX_CTRL	0x400234DB	Endpoint0 receive control register	00h
R16_UEP1_T_LEN	0x400234DC	Endpoint1 transmit length register	xxxxh
R8_UEP1_TX_CTRL	0x400234DE	Endpoint1 transmit control register	00h
R8_UEP1_RX_CTRL	0x400234DF	Endpoint1 receive control register	00h
R16_UEP2_T_LEN	0x400234E0	Endpoint2 transmit length register	xxxxh
R8_UEP2_TX_CTRL	0x400234E2	Endpoint2 transmit control register	00h
R8_UEP2_RX_CTRL	0x400234E3	Endpoint2 receive control register	00h
R16_UEP3_T_LEN	0x400234E4	Endpoint3 transmit length register	xxxxh
R8_UEP3_TX_CTRL	0x400234E6	Endpoint3 transmit control register	00h
R8_UEP3_RX_CTRL	0x400234E7	Endpoint3 receive control register	00h
R16_UEP4_T_LEN	0x400234E8	Endpoint4 transmit length register	xxxxh
R8_UEP4_TX_CTRL	0x400234EA	Endpoint4 transmit control register	00h
R8_UEP4_RX_CTRL	0x400234EB	Endpoint4 receive control register	00h
R16_UEP5_T_LEN	0x400234EC	Endpoint5 transmit length register	xxxxh
R8_UEP5_TX_CTRL	0x400234EE	Endpoint5 transmit control register	00h
R8_UEP5_RX_CTRL	0x400234EF	Endpoint5 receive control register	00h
R16_UEP6_T_LEN	0x400234F0	Endpoint6 transmit length register	xxxxh
R8_UEP6_TX_CTRL	0x400234F2	Endpoint6 transmit control register	00h
R8_UEP6_RX_CTRL	0x400234F3	Endpoint6 receive control register	00h
R16_UEP7_T_LEN	0x400234F4	Endpoint7 transmit length register	xxxxh
R8_UEP7_TX_CTRL	0x400234F6	Endpoint7 transmit control register	00h
R8_UEP7_RX_CTRL	0x400234F7	Endpoint7 receive control register	00h
R16_UEP8_T_LEN	0x400234F8	Endpoint8 transmit length register	xxxxh
R8_UEP8_TX_CTRL	0x400234FA	Endpoint8 transmit control register	00h
R8_UEP8_RX_CTRL	0x400234FB	Endpoint8 receive control register	00h
R16_UEP9_T_LEN	0x400234FC	Endpoint9 transmit length register	xxxxh
R8_UEP9_TX_CTRL	0x400234FE	Endpoint9 transmit control register	00h
R8_UEP9_RX_CTRL	0x400234FF	Endpoint9 receive control register	00h
R16_UEP10_T_LEN	0x40023500	Endpoint10 transmit length register	xxxxh
R8_UEP10_TX_CTRL	0x40023502	Endpoint10 transmit control register	00h
R8_UEP10_RX_CTRL	0x40023503	Endpoint10 receive control register	00h
R16_UEP11_T_LEN	0x40023504	Endpoint11 transmit length register	xxxxh
R8_UEP11_TX_CTRL	0x40023506	Endpoint11 transmit control register	00h
R8_UEP11_RX_CTRL	0x40023507	Endpoint11 receive control register	00h
R16_UEP12_T_LEN	0x40023508	Endpoint12 transmit length register	xxxxh
R8_UEP12_TX_CTRL	0x4002350A	Endpoint12 transmit control register	00h
R8_UEP12_RX_CTRL	0x4002350B	Endpoint12 receive control register	00h
R16_UEP13_T_LEN	0x4002350C	Endpoint13 transmit length register	xxxxh
R8_UEP13_TX_CTRL	0x4002350E	Endpoint13 transmit control register	00h
R8_UEP13_RX_CTRL	0x4002350F	Endpoint13 receive control register	00h
R16_UEP14_T_LEN	0x40023510	Endpoint14 transmit length register	xxxxh
R8_UEP14_TX_CTRL	0x40023512	Endpoint14 transmit control register	00h
R8_UEP14_RX_CTRL	0x40023513	Endpoint14 receive control register	00h

R16_UEP15_T_LEN	0x40023514	Endpoint15 transmit length register	xxxxh
R8_UEP15_TX_CTRL	0x40023516	Endpoint15 transmit control register	00h
R8_UEP15_RX_CTRL	0x40023517	Endpoint15 receive control register	00h

22.2.2.1 USB Endpoint Configuration Register (R32_UEP_CONFIG)

Bit	Name	Access	Description	Reset value
[31:17]	RB_UEP_R_EN	RW	Endpoint 1 to 15 receive enable	0
16	Reserved	RO	Reserved	0
[15:1]	RB_UEP_T_EN	RW	Endpoint 1 to 15 transmit enable	0
0	Reserved	RO	Reserved	0

Note: The transceiver enable signal of endpoint 0 is always valid.

22.2.2.2 USB Endpoint Type Control Register (R32_UEP_TYPE)

Bit	Name	Access	Description	Reset value
[31:17]	RB_UEP_R_TYPE	RW	Endpoint 1 to 15, OUT direction transmission type, 1 means synchronous transmission	0
16	Reserved	RO	Reserved	0
[15:1]	RB_UEP_T_TYPE	RW	Endpoint 1 to 15, IN direction transmission type, 1 means synchronous transmission	0
0	Reserved	RO	Reserved	0

22.2.2.3 USB Endpoint Buffer Mode Control Register (R32_UEP_BUF_MOD)

Bit	Name	Access	Description	Reset value
[31:16]	RB_UEP_ISO_BUF_MOD	RW	Synchronous endpoint buffer mode control, 1 is valid.	0
[15:0]	RB_UEP_BUF_MOD	RW	Endpoint buffer mode control	0

Note: When RB_UEP_ISO_BUF_MOD is 1, for the synchronous IN endpoint, after receiving the SOF packet, the hardware will perform the following operations: load the content of EPx_R_TOG into EPx_T_TOG; load the value of EPx_MAX_LEN into EPx_T_LEN; load the value of UEPn_RX_DMA into UEPn_TX_DMA.
When RB_UEP_ISO_BUF_MOD is 1, for the synchronous OUT endpoint, after receiving the SOF packet, the hardware will perform the following operations: load the content of EPx_T_TOG into EPx_R_TOG; load the value of UEPn_TX_DMA into UEPn_RX_DMA.

Table 21-3 Endpoint n buffer mode (n=1-15)

UEPn_RX_EN	UEPn_TX_EN	UEPn_BUF_MOD	Description: Arrange from low to high with UEPn_DMA as the starting address
0	0	x	The endpoint is disabled and the UEPn_*_DMA buffers are not used.

1	0	0	The first address of the receive (OUT) buffer is UEPn_RX_DMA

1	0	1	RB_UEPn_RX_TOG[0]=0, use buffer UEPn_RX_DMA
			RB_UEPn_RX_TOG[0]=1, use buffer UEPn_TX_DMA

0	1	0	The first address of the transmit (IN) buffer is UEPn_TX_DMA.

0	1	1	RB_UEPn_TX_TOG[0]=0, use buffer UEPn_TX_DMA
			RB_UEPn_TX_TOG[0]=1, use buffer UEPn_RX_DMA

22.2.2.4 Endpoint n buffer start address (R32_UEP0_DMA)

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
[16:0]	R32_UEPn_DMA	RW	Endpoint 0 buffer start address. The lower 16 bits are valid, and the address must be 4-byte aligned.	X

22.2.2.5 USB Endpoint n transmit buffer start address (R32_UEPn_TX_DMA) (n=1-15)

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
[16:0]	R32_UEPn_TX_DMA	RW	Endpoint n send buffer start address. The lower 16 bits are valid, and the address must be 4-byte aligned.	X

22.2.2.6 USB Endpoint n receive buffer start address (R32_UEPn_RX_DMA) (n=1-15)

Bit	Reserved	Access	Description	Reset value
[16:0]	R32_UEPn_RX_DMA	RW	Endpoint n receives the buffer start address. The lower 16 bits are valid, and the address must be 4-byte aligned.	X

22.2.2.7 Endpoint n maximum length packet register (R16_UEPn_MAX_LEN) (n=0-15)

Bit	Reserved	Access	Description	Reset value
[15:11]	Reserved	RO	Reserved	0
[10:0]	UEPn_MAX_LEN	RW	The maximum packet length of data received by endpoint n.	xxxxh

Note: This maximum packet length determines the maximum length of data that the endpoint can receive. Data beyond this length will be discarded and will not be written to the buffer.

22.2.2.8 Endpoint n transmit length register (R16_UEPn_T_LEN) (n=0-15)

Bit	Reserved	Access	Description	Reset value
[10:0]	UEPn_T_LEN	RW	Set the number of data bytes to be sent by USB endpoint n. For the control endpoint (0), the lower 7 bits are valid.	X

22.2.2.9 Endpoint n transmit control register (R8_UEPn_TX_CTRL) (n=0-15)

Bit	Name	Access	Description	Reset value
[7:6]	Reserved	RO	Reserved	0
5	RB_UEP_T_TOG_AUTO	RW	Synchronization trigger bit auto toggle enable, which can be modified by software: 1: For non-synchronous endpoints, MASK_UEP_T_TOG [0] is automatically toggled after data transmission is successful. For synchronous endpoints, MASK_UEP_T_TOG is automatically decremented by 1 after data transmission is successful. 0: No automatic toggle, can be switched manually. *Note: This bit of endpoint 0 is reserved.*	0

[4:3]	MASK_UEP_T_TOG	RW	Synchronization trigger bit prepared by the transmitter of USB endpoint n (handling IN transaction):
			00: Send DATA0;
			01: Send DATA1;
			10: Send DATA2;
			11: Send MDATA.
				0

2	Reserved	RO	Reserved	00b
[1:0]	MASK_UEP_T_RES	RW	Response control by the sender of endpoint n to IN transaction: 00: data ready and expecting ACK; 10: Answer NAK or busy; 11: Response STALL or error.	00b

22.2.2.10 Endpoint n receive control register (R8_UEPn_RX_CTRL) (n=0-15)

Bit	Name	Access	Description	Reset value
[7:6]	Reserved	RO	Reserved	0
5	RB_UEP_R_TOG_A	RO	Synchronization trigger bit auto toggle	0

	UTO		enable: 1: Automatically toggle MASK_UEP_R_TOG [0] after data reception is successful; 0: No automatic toggle, can be switched manually. *Note: This bit of endpoint 0 is reserved.*	

[4:3]	MASK_UEP_R_TOG	RW	Sync trigger bit prepared by receiver of USB endpoint n (processing OUT transaction):
			00: Expect DATA0;
			01: Expect DATA1;
			10: Expect DATA2;
			11: Expect MDATA.
			Invalid for isochronous transfer.
				0

2	Reserved	RO	Reserved	00b
[1:0]	MASK_UEP_R_RES	RW	Receiver of endpoint n responds to OUT transaction control: 00: Data ready and expect ACK; 10: Answer NAK or busy; 11: Response STALL or error; 01: Respond to NYET. Not valid for real time/isochronous transfers.	00b

22.2.3 USB host register

In USB host mode, the chip provides a set of bidirectional host endpoints, including a sending endpoint OUT and a receiving endpoint IN, the maximum length of a data packet is 1024 bytes (synchronous transfer), and supports control transfer, interrupt transfer, bulk transfer and real-time/simultaneous transmission.

Each USB transaction initiated by the host endpoint always automatically sets the RB_UIF_TRANSFER interrupt flag after processing. The application can directly query or query and analyze the interrupt flag register R8_USB_INT_FG in the USB interrupt service routine, and perform corresponding processing according to each interrupt flag; and, if RB_UIF_TRANSFER is valid, then continue to analyze the USB interrupt status register R8_USB_INT_ST, process accordingly according to the response PID identification MASK_UIS_H_RES of the current USB transfer transaction.

If the synchronization trigger bit (RB_UH_R_TOG) of the IN transaction of the host receiving endpoint is set in advance, then RB_U_TOG_OK or RB_UIS_TOG_OK can be used to judge whether the synchronization trigger bit of the currently received data packet matches the synchronization trigger bit of the host receiving endpoint. If synchronized, the data is valid; if the data is not synchronized, the data should be discarded. After each USB send or receive interrupt is processed, the synchronization trigger bit of the corresponding host endpoint should be modified correctly to synchronize the data packet sent next time and detect whether the data packet received next time is synchronized; in addition, by setting RB_UH_T_AUTO_TOG and RB_UH_R_AUTO_TOG can automatically flip the corresponding synchronization trigger bit after successful transmission or successful reception.

The USB host token setting register R8_UH_EP_PID is used to set the endpoint number of the target device to be operated and the token PID packet identification of this USB transfer transaction. The data corresponding to the SETUP token and the OUT token is provided by the host sending endpoint, the data to be sent is in the R16_UH_TX_DMA buffer, and the length of the data to be sent is set in R16_UH_TX_LEN; the data corresponding to the IN token is returned by the target device to the host for reception Endpoint, the received data is stored in the R16_UH_RX_DMA buffer, and the received data length is stored in R16_USB_RX_LEN.

Table 22-3 Host registers

Name	Access address	Description	Reset value
R8_UHOST_CTRL	0x40023401	USB host control register	00h
R32_UH_CONFIG	0x40023410	USB host endpoint configuration register	00000000h
R32_UH_EP_TYPE	0x40023414	USB host endpoint type register	00000000h
R32_UH_RX_DMA	0x40023424	USB host receive buffer start address	16hxxxx
R32_UH_TX_DMA	0x40023464	USB host transmit buffer start address	16hxxxx
R16_UH_RX_MAX_LEN	0x400234A0	USB host receive maximum length packet register	16hxxxx
R8_UH_EP_PID	0x400234E0	USB host token setup register	8h00
R8_UH_RX_CTRL	0x400234E3	USB host receive endpoint control register	8h00
R16_UH_TX_LEN	0x400234E4	USB host transmit length register	16hxxxx
R8_UH_TX_CTRL	0x400234E6	USB host transmit endpoint control Register	8h00
R16_UH_SPLIT_DATA	0x400234E8	USB host transmit data of the SPLIT packet	16hxxxx

22.2.3.1 USB Host Control Register (R8_UHOST_CTRL)

Bit	Name	Access	Description	Reset value
7	RB_UH_SOF_EN	RW	Automatically generate SOF packet enable control bit: 1: The host automatically generates a SOF packet; 0: Do not generate SOF packets. This bit is automatically cleared by hardware when changing from connected to disconnected state.	0
6	RB_UH_SOF_FREE	RO	Bus Idle.	0
5	Reserved	RO	Reserved.	0
4	RB_UH_PHY_SUSPENDM	RW	The USB-PHY is in a suspended state, the internal USB-PLL will be turned off, active low.	0
3	RB_UH_REMOTE_WKUP	RW	Wake up remotely.	0
2	RB_UH_TX_BUS_	RW	In host mode, it means that the host wakes	0

	RESUME		up the device. After the software pulls it high for 50ns, the hardware automatically sends a 30ms wake-up signal.	
1	RB_UH_TX_BUS_SUSPEND	RW	The USB host sends a suspend signal, which needs to be pulled up by software for 10ms.	0
0	RB_UH_TX_BUS_RESET	RW	The USB host sends a bus reset signal, which needs to be pulled up by software for 10ms.	0

Note: The reset time is determined by the high-level duration of RB_UH_TX_BUS_RESET (recommended at least 10ms, and the speed type is directly inquired after 10ms). If the host wakes up the device, after bUH_TX_BUS_RESUME is pulled high, the hardware automatically sends a wake-up signal (K) for 30ms, and bUH_TX_BUS_RESUME needs to be cleared manually so as not to affect the next host suspend (bUH_TX_BUS_RESUME remains high for at least 50ns).

22.2.3.2 USB Host Endpoint Configuration Control Register (R32_UH_CONFIG)

Bit	Name	Access	Description	Reset value
[31: 19]	Reserved	RO	Reserved	0
18	RB_UH_EP_RX_EN	RW	Host receive enable	0
[17:4]	Reserved	RO	Reserved	0
3	RB_UH_EP_TX_EN	RW	Host send enable	0
[2:0]	Reserved	RO	Reserved	0

22.2.3.3 USB Host Endpoint Type Register (R32_UH_EP_TYPE)

Bit	Name	Access	Description	Reset value
[31:19]	Reserved	RO	Reserved	0
[18]	RB_UH_EP_RX_TYPE	RW	Host Rx endpoint type 1: Isochronous transfer.	0
[17:4]	Reserved	RO	Reserved	0
[3]	RB_UH_EP_TX_TYPE	RW	Host Tx endpoint type 1: Isochronous transfer.	0
[2:0]	Reserved	RO	Reserved	0

22.2.3.4 USB Host Receive Buffer Start Address (R32_UH_RX_DMA)

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
[16:0]	R16_UH_RX_DMA	RW	Start address of the host endpoint Rx buffer, the lowest 2 bits are fixed to 0 (4-byte alignment).	X

22.2.3.5 USB Host Transmit Buffer Start Address (R32_UH_TX_DMA)

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
[16:0]	R16_UH_TX_DMA	RW	Host endpoint Tx buffer start address (4-byte alignment is not required).	X

22.2.3.6 USB Host Receive Maximum Length Packet Register (R16_UH_RX_MAX_LEN)

Bit	Name	Access	Description	Reset value
[15:11]	Reserved	RO	Reserved	0
[10:0]	UH_RX_MAX_LEN	RW	The maximum packet length of data received by the host endpoint.	xxxxh

Note: This maximum packet size determines the maximum length of data that the endpoint can receive, and data beyond this length will be discarded.

22.2.3.7 USB Host Token Setup Register (R8_UH_EP_PID)

Bit	Name	Access	Description	Reset value
[7:4]	MASK_UH_TOKEN	RW	Set the token PID identifier of this USB transfer transaction.	0
[3:0]	MASK_UH_ENDP	RW	Set the endpoint number of the target device to be operated this time.	0

22.2.3.8 USB Host Receive Endpoint Control Register (R8_UH_RX_CTRL)

Bit	Name	Access	Description	Reset value
7	Reserved	RO	Reserved	0
6	RB_UH_R_DATA_NO	RW	1: Not expect data packets, used for operating high-speed HUB in host mode; 0: Expect data packet (IN).	0
5	RB_UH_R_AUTO_TOG	RW	Synchronization trigger bit auto toggle enable: 1: For asynchronous transfer, the corresponding MASK_UH_R_TOG[0] will be automatically toggled after data reception is successful; for synchronous transfer, MASK_UH_R_TOG will be automatically decremented by 1 after data reception is successful. 0: No automatic toggle, can be switched manually.	0
[4:3]	MASK_UH_R_TOG	RW	Synchronization trigger bit expected by the master receiver (handling the IN	0

			transaction),
			00: Expect DATA0;
			01: Expect DATA1;
			10: Expect DATA2;
			11: Expect MDATA.	

2	RB_UH_R_RES_NO	RW	1: No acknowledgment, for real-time/isochronous transfers other than endpoint 0. MASK_UEP_R_RES is ignored at this time; 0: Send a response after receiving data successfully.	0
[1:0]	MASK_UH_R_RES	RW	Master Receiver Response Control Bits to IN Transactions: 00: reply ACK; Not valid for real-time/isochronous transfers.	0

22.2.3.9 USB Host Transmit Length Register (R16_UH_TX_LEN)

Bit	Name	Access	Description	Reset value
[15:11]	Reserved	RO	Reserved	0
[10:0]	R16_UH_TX_LEN	RW	Set the data bytes that the USB host transmit endpoint is ready to transmit.	X

22.2.3.10 USB Host Transmit Endpoint Control Register (R8_UH_TX_CTRL)

Bit	Name	Access	Description	Reset value
7	Reserved	RO	Reserved	0
6	RB_UH_T_DATA_NO	RW	1: Do not send data packets (PING/SPLIT); 0: Send data packets (OUT/SETUP).	0
5	RB_UH_T_AUTO_TO G	RW	Synchronization trigger bit auto toggle enable, which can be modified by software: 1: For asynchronous transfer, MASK_UH_T_TOG[0] is automatically toggled after data is sent successfully. 0: No automatic toggle, can be switched manually.	0

[4:3]	MASK_UH_T_TOG	RW	Sync trigger bit prepared by USB host transmitter (handling SETUP/OUT transactions)
			00: Send DATA0;
			01: Send DATA1;
			10: Send DATA2;
				0

			11: Send MDATA.	
2	RB_UH_T_RES_NO	RW	1: No acknowledgment, for real-time/isochronous transfers other than endpoint0. MASK_UEP_T_RES is ignored at this time; 0: Expect a response after sending data successfully.	0

[1:0]	MASK_UH_T_RES	RW	USB Host transmitter response control bits to SETUP/OUT transactions
			00: Expect to reply ACK;
			10: expect to answer NAK or busy;
			11: Expect to answer STALL or error;
			01: Expect to answer NYET.
			Invalid for isochronous transfer.
				0

22.2.3.11 USB Host Transmit SPLIT Packet Data (R16_UH_SPLIT_DATA)

Bit	Name	Access	Description	Reset value
[15:12]	Reserved	RO	Reserved	0
[11:0]	UH_SPLIT_DATA	RW	Data content of the SPLIT packet sent by the host endpoint, the lower 12 bits are valid, and the upper 4 bits are fixed to 0.	0xxxh

Chapter 23 USB OTG Full-speed Controller (OTG_FS)

The module descriptions in this chapter are applicable to some products of the CH32F2x, CH32V2x and CH32V3x microcontroller series.

23.1 USB controller introduction

The chip is embedded with a USB controller and transceiver, and the features include:

- Dual-role device controller, supporting USB Host function and USB Device function.
- Compliant with the On-The-Go Supplement to the USB2.0 specification, both host and device modes support USB2.0 full-speed 12Mbps or low-speed 1.5Mbps.
- Support software HNP and SRP protocols.
- Support USB control transfer, batch transfer, interrupt transfer, synchronous/real-time transfer.
- Supports data packets up to 64 bytes, built-in FIFO, supports interrupts and DMA.

23.2 Register description

The USB registers are divided into 3 parts, and some registers are multiplexed in the host and device modes.

- USB global registers
- USB device control registers
- USB host control registers

23.2.1 Global register description

Table 23-1 USB OTG registers

Name	Access address	Description	Reset value
R8_USB_CTRL	0x50000000	USB control register	0x06
R8_USB_INT_EN	0x50000002	USB interrupt enable register	0x00
R8_USB_DEV_AD	0x50000003	USB Device address register	0x00
R32_USB_STATUS	0x50000004	USB status register	0xXX20XXXX
R8_USB_MIS_ST	0x50000005	USB miscellaneous status register	0xXX
R8_USB_INT_FG	0x50000006	USB interrupt flag register	0x20
R8_USB_INT_ST	0x50000007	USB interrupt status register	0xXX
R16_USB_RX_LEN	0x50000008	USB receive length register	0xXX
R32_USB_OTG_CR	0x50000054	USB OTG control register	0x00
R32_USB_OTG_SR	0x50000058	USB OTG status register	0x00

23.2.1.1 USB Control Register (R8_USB_CTRL)

Bit	Name	Access	Description	Reset value
7	RB_UC_HOST_MODE	RW	USB operating mode selection: 1: Host mode; 0: Device mode.	0
6	RB_UC_LOW_SPEED	RW	USB bus signal transfer rate selection: 1: 1.5Mbps; 0: 12Mbps.	0

5	RB_UC_DEV_PU_EN	RW	USB device enable and internal pull-up resistor control in USB device mode: 1: Enable USB device transfer and enable the internal pull-up resistor.	0
[5:4]	MASK_UC_SYS_CTRL	RW	See the table below to configure the USB system.	0
3	RB_UC_INT_BUSY	RW	Auto pause enable bit before USB transfer completed interrupt flag is not cleared: 1: Auto pause before UIF_TRANSFER is cleared, auto respond busy NAK in device mode, and auto pause subsequent transfers in host mode; 0: Not pause.	0
2	RB_UC_RESET_SIE	RW	USB protocol processor software reset control: 1: Forcibly reset the USB protocol processor (SIE), which needs to be cleared by software; 0: No reset.	1
1	RB_UC_CLR_ALL	RW	USB FIFO and interrupt flags clear: 1: Forced clearing and clearing; 0: Not clear.	1
0	RB_UC_DMA_EN	RW	DMA and DMA interrupt control for USB: 1: Enable DMA function and DMA interrupt; 0: Disable DMA.	0

The USB system control combination is composed of RB_UC_HOST_MODE and MASK_UC_SYS_CTRL:

Table 23-2 USB System control combination

RB_UC_HOST_MODE	MASK_UC_SYS_CTRL	USB system control description
0	00	Disable the USB device function, turn off the internal pull-up resistor.
0	01	Enable the USB device function, close the internal pull-up resistor, and need to add an external pull-up.
0	1x	Enable USB device function, enable internal 1.5K pull-up resistor. This pull-up resistor takes precedence over the pull-down resistor and can also be used in GPIO mode.
1	00	USB host mode, normal working state.
1	01	USB host mode, forcing DP/DM to output SE0 state.
1	10	USB host mode, forces DP/DM to output J state.
1	11	USB host mode, force DP/DM to output K state/wake up.

23.2.1.2 USB Interrupt Enable Register (R8_USB_INT_EN)

Bit	Name	Access	Description	Reset value
7	RB_UIE_DEV_SOF	RW	USB device mode, receive SOF packet interrupt: 1: Enable interrupt; 0: Disable interrupt.	0
6	RB_UIE_DEV_NAK	RW	USB device mode, receive NAK interrupt: 1: Enable interrupt; 0: Disable interrupt.	0
5	Reserved	RO	Reserved	0
4	RB_UIE_FIFO_OV	RW	FIFO overflow interrupt: 1: Enable interrupt; 0: Disable interrupt.	0
3	RB_UIE_HST_SOF	RW	USB host mode, SOF timing interrupt: 1: Enable interrupt; 0: Disable interrupt.	0
2	RB_UIE_SUSPEND	RW	USB bus suspend or wakeup event interrupt: 1: Enable interrupt; 0: Disable interrupt.	0
1	RB_UIE_TRANSFER	RW	USB transfer complete interrupt: 1: Enable interrupt; 0: Disable interrupt.	0
0	RB_UIE_DETECT	RW	USB host mode, USB device connect or disconnect event interrupt: 1: Enable interrupt; 0: Disable interrupt.	0
	RB_UIE_BUS_RST	RW	USB device mode, USB bus reset event interrupt: 1: Enable interrupt; 0: Disable interrupt.	0

23.2.1.3 USB Device Address Register (R8_USB_DEV_AD)

Bit	Name	Access	Description	Reset value
7	RB_UDA_GP_BIT	RW	USB general-purpose flag, user-defined.	0
[6:0]	MASK_USB_ADDR	RW	Host mode: current operated USB device address; Device Mode: USB own address.	0

23.2.1.4 USB Miscellaneous Status Register (R8_USB_MIS_ST)

Bit	Name	Access	Description	Reset value
7	RB_UMS_SOF_PRES	RO	SOF packet presage status in USB host mode: 1: SOF packet is about to be sent, if there are other USB packets at this time, it will be automatically delayed; 0: No SOF packet is sent.	X

6	RB_UMS_SOF_ACT	RO	SOF packet transfer status in USB host mode: 1: SOF packet is being sent; 0: Send complete or idle.	X
5	RB_UMS_SIE_FREE	RO	USB protocol handler free: 1: The protocol device is free; 0: Busy, USB transfer in progress.	1
4	RB_UMS_R_FIFO_RDY	RO	USB receive FIFO data ready: 1: The receive FIFO is not empty; 0: The receive FIFO is empty.	0
3	RB_UMS_BUS_RESET	RO	USB bus reset: 1: The current USB bus is in reset state; 0: The current USB bus is in a non-reset state.	X
2	RB_UMS_SUSPEND	RO	USB suspend: 1: The USB bus is in a suspended state, and there is no USB activity for a period of time; 0: The USB bus is in a non-suspend state.	0
1	RB_UMS_DM_LEVEL	RO	In USB host mode, level state of the DM pin when the device is just connected to the USB port is used to judge the speed: 1: High level/low-speed; 0: Low level/full-speed.	0
0	RB_UMS_DEV_ATTACH	RO	USB device attach status for the port in USB host mode: 1: The port has been connected to a USB device; 0: The port has no USB device connected.	0

23.2.1.5 USB Interrupt Flag Register (R8_USB_INT_FG)

Bit	Name	Access	Description	Reset value
7	RB_U_IS_NAK	RO	In USB device mode, NAK response status: 1: Respond to NAK during the current USB transfer process; 0: No NAK response.	0
6	RB_U_TOG_OK	RO	Current USB transfer DATA0/1 sync flag match status: 1: Synchronized; 0: Not synchronized.	0
5	RB_U_SIE_FREE	RO	USB Protocol Handler free: 1: USB idle; 0: Busy, USB transfer in progress.	1

4	RB_UIF_FIFO_OV	RW	USB FIFO overflow interrupt flag, write 1 to clear: 1: FIFO overflow trigger; 0: no event.	0
3	RB_UIF_HST_SOF	RW	SOF timer interrupt flag in USB host mode, write 1 to clear: 1: SOF packet transmission completion trigger; 0: no event.	0
2	RB_UIF_SUSPEND	RW	USB bus suspend or wake-up event interrupt flag, write 1 to clear: 1: Triggered by USB suspend event or wake-up event; 0: No event.	0
1	RB_UIF_TRANSFER	RW	USB transfer completion interrupt flag, write 1 to clear: 1: Trigger when a USB transfer is completed; 0: No event.	0
0	RB_UIF_DETECT	RW	In USB host mode, USB device connect or disconnect event interrupt flag, write 1 to clear: 1: Detected USB device connection or disconnection trigger; 0: No event.	0
	RB_UIF_BUS_RST	RW	In USB device mode, USB bus reset event interrupt flag bit, write 1 to clear: 1: USB bus reset event trigger; 0: No event.	0

23.2.1.6 USB Interrupt Status Register (R8_USB_INT_ST)

Bit	Name	Access	Description	Reset value
7	RB_UIS_IS_NAK	RO	In USB device mode, NAK response status, same as RB_U_IS_NAK: 1: Respond to NAK during the current USB transfer process; 0: No NAK response.	0
6	RB_UIS_TOG_OK	RO	Current USB transmission DATA0/1 synchronization flag match status, same as RB_U_TOG_OK: 1: Synchronized; 0: Not synchronized.	0
[5:4]	MASK_UIS_TOKEN	RO	In device mode, the token PID identifier of the current USB transfer transaction.	XXb
[3:0]	MASK_UIS_ENDP	RO	In device mode, the endpoint number of	XXXXb

			the current USB transfer transaction.	
	MASK_UIS_H_RES	RO	In host mode, the response PID identifier of the current USB transmission transaction, 0000 means the device has no response or timed out; other values indicate the response PID.	XXXXb

MASK_UIS_TOKEN is used to identify the token PID of the current USB transfer transaction in the USB device mode: 00 means OUT packet; 01 means SOF packet; 10 means IN packet; 11 means SETUP packet.

MASK_UIS_H_RES is only valid in host mode. In the host mode, if the host sends the OUT/SETUP token packet, the PID is the handshake packet ACK/NAK/STALL, or the device has no response/timeout. If the host sends an IN token packet, the PID is the PID of the data packet (DATA0/DATA1) or the PID of the handshake packet.

23.2.1.7 USB Receive Length Register (R16_USB_RX_LEN)

Bit	Name	Access	Description	Reset value
[9:0]	R16_USB_RX_LEN	RO	The current number of data bytes received by the USB endpoint	X

23.2.1.8 USB OTG Control Register (R32_USB_OTG_CR)

Bit	Name	Access	Description	Reset value
[31:6]	Reserved	R0	Reserved	0
5	RB_CR_SESS_VTH	RW	OTG session valid threshold voltage setting: 1: SESS_VLD level is 1.4V; 0: SESS_VLD level is 0.8V.	0
4	RB_CR_VBUS_VTH	RW	OTG VBUS threshold voltage setting: 1: VBUS_VLD level is 4.4V; 0: VBUS_VLD level is 4.8V.	0
3	RB_CR_OTG_EN	RW	OTG function enable: 1: Enable; 0: Disable	0
2	RB_CR_IDPU	RW	USB_OTG_ID pin pull-up enable: 1: Enable; 0: Disable	0
1	RB_CR_CHARGE_VBUS	RW	OTG VBUS charge enable: 1: Enable; 0: Disable	0
0	RB_CR_DISCHAR_VBUS	RW	OTG VBUS discharge enable: 1: Enable; 0: Disable	0

Note: This register is only applicable to CH32V305, CH32V307, CH32F205 and CH32F207

23.2.1.9 USB OTG Control Register (R32_USB_OTG_SR)

Bit	Name	Access	Description	Reset value
[31:4]	Reserved	R0	Reserved	0

3	RB_SR_ID_DIG	RO	OTG ID flag: 1: B device; 0: A device.	0
2	RB_SR_SESS_END	RO	OTG session end valid flag 1: Valid; 0: Invalid	0
1	RB_SR_SESS_VLD	RO	OTG session valid flag 1: Valid, the session valid level is greater than the threshold voltage 0: Invalid, the session valid level is less than the threshold voltage	0
0	RB_SR_VBUS_VLD	RO	OTG VBUS input level 1: VBUS voltage is greater than the threshold voltage; 0: VBUS voltage is less than the threshold voltage.	X

Note: This register is only applicable to CH32V305, CH32V307, CH32F205 and CH32F207

23.2.2 Device register description

In USB device mode, the USB OTG module provides 8 sets of bidirectional endpoint configuration registers with endpoint numbers 0-7, which can map the configuration of endpoint numbers 8-15. The maximum packet length of all endpoints except endpoint 3 is 64 bytes. The maximum packet length of the endpoints is 1023 bytes (synchronous transmission).

- Endpoint 0 is the default endpoint and supports control transmission. Sending and receiving share a 64-byte data buffer
- Endpoints 1-15 can be configured with independent 64-byte send and receive buffers or dual 64-byte data buffers, supporting bulk transfers, interrupt transfers, and real-time/synchronous transfers.

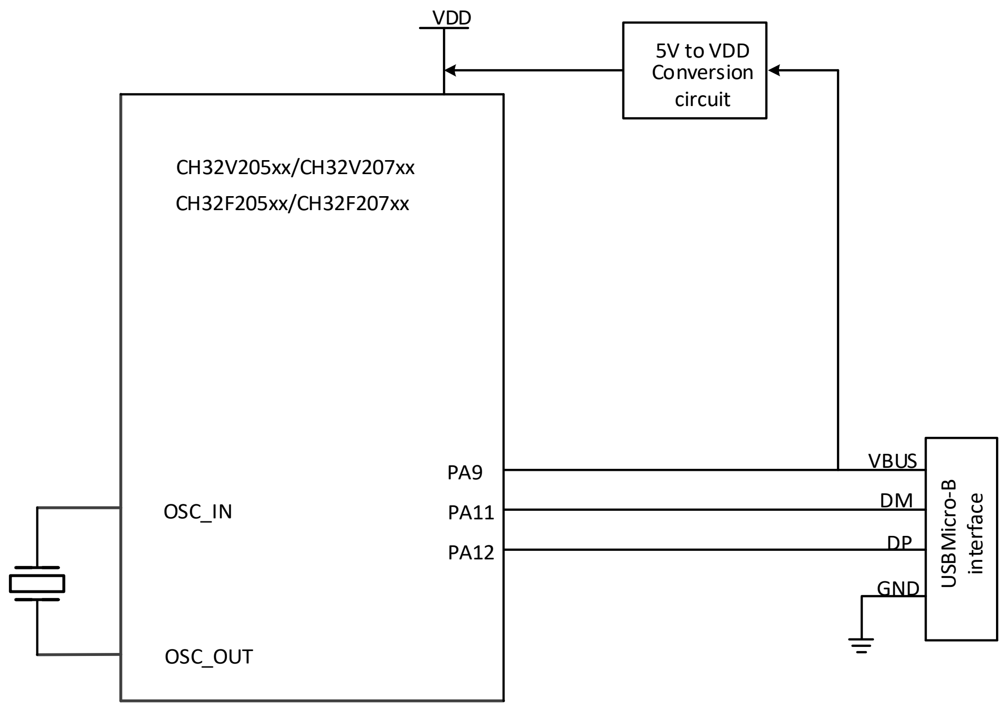
Each group of endpoints has a control register R8_UEPn_CTRL and a send length register R16_UEPn_T_LEN, which are used to set the synchronization trigger bit of the endpoint, the response to the OUT transaction and the IN transaction, and the length of the transmitted data.

The roles of the USB OTG module host and device are determined by the state of the OTG_FS_ID pin. When the OTG_FS_ID pin is floating, its built-in pull-up resistor will make the RB_SR_ID_DIG bit of the USB OTG status register R32_USB_OTG_SR set to 1. At this time, the controller should be initialized as a B device. When the OTG_FS_ID pin is grounded, the RB_SR_ID_DIG bit of the USB OTG status register R32_USB_OTG_SR is 0 at this time, and the controller should be initialized as an A device.

When the B device starts a session, it ensures that the level of VBUS is lower than $V_{SESS\_VLD,min}$ according to RB_SR_SESS_VLD of the OTG status register R32_USB_OTG_SR. If this bit is 0, a new session can be started. If this bit is 1, the RB_CR_DISCHAR_VBUS bit of the OTG control register R32_USB_OTG_CR can be set to 1 to discharge, so that VBUS is less than the session threshold level.

As a Class B device, it needs to take power from VBUS, and an external conversion circuit is required, as shown in Figure 22-1.

Figure 23-1 OTG Class B device connection



The USB bus pull-up resistor necessary as a USB device can be set by software at any time to enable or not. When RB_UC_DEV_PU_EN in the USB control register R8_USB_CTRL is set to 1, the controller is set according to the speed of RB_UD_LOW_SPEED, which is the DP/DM pin of the USB bus internally. Connect a pull-up resistor and enable the USB device function.

When a USB bus reset, USB bus suspend or wake-up event is detected, or when the USB successfully processes data transmission or data reception, the USB protocol processor will set the corresponding interrupt flag, and if the interrupt enable is turned on, it will also generate a corresponding interrupt request. The application can directly query or query and analyze the interrupt flag register R8_USB_INT_FG in the USB interrupt service routine, and perform corresponding processing according to RB_UIF_BUS_RST and RB_UIF_SUSPEND; and, if RB_UIF_TRANSFER is valid, then it is necessary to continue to analyze the USB interrupt status register R8_USB_INT_ST, according to the current endpoint number MASK_UIS_ENDP and the current transaction token PID identify MASK_UIS_TOKEN for corresponding processing. If the synchronization trigger bit RB_UEP_R_TOG of the OUT transaction of each endpoint is set in advance, then RB_U_TOG_OK or RB_UIS_TOG_OK can be used to judge whether the synchronization trigger bit of the currently received data packet matches the synchronization trigger bit of the endpoint. If the data is synchronized, the data valid; if the data is out of sync, the data should be discarded. After each USB send or receive interrupt is processed, the synchronization trigger bit of the corresponding endpoint should be modified correctly to detect whether the next data packet sent or received next time is synchronously detected; in addition, setting RB_UEP_AUTO_TOG enables to automatically flip the corresponding synchronization trigger bit after successful transmission or reception.

The data to be sent by each endpoint is in its own buffer, and the length of the data to be sent is independently set in R8_UEPn_T_LEN; the data received by each endpoint is in its own buffer, but the length of the received data is in the USB receive length. In the register R8_USB_RX_LEN, it can be distinguished according to the current endpoint number when the USB receives an interrupt.

Table 23-3 Device registers

Name	Address	Description	Reset value
R8_UDEV_CTRL	0x50000001	USB device physical port control register	0xX0
R8_UEP4_1_MOD	0x5000000C	Endpoint 1(9)/4(8/12) mode control register	0x00
R8_UEP2_3_MOD	0x5000000D	Endpoint 2(10)/3(11) mode control register	0x00
R8_UEP5_6_MOD	0x5000000E	Endpoint 5(13)/6(14) mode control register	0x00
R8_UEP7_MOD	0x5000000F	Endpoint 7(15) mode control register	0x00
R16_UEP0_DMA	0x50000010	Endpoint 0 buffer start address	0xXXXX
R16_UEP1_DMA	0x50000014	Endpoint 1(9) buffer start address	0xXXXX
R16_UEP2_DMA	0x50000018	Endpoint 2 (10) buffer start address	0xXXXX
R16_UEP3_DMA	0x5000001C	Endpoint 3 (11) buffer start address	0xXXXX
R16_UEP4_DMA	0x50000020	Endpoint 4 (8/12) buffer start address	0xXXXX
R16_UEP5_DMA	0x50000024	Endpoint 5(13) buffer start address	0xXXXX
R16_UEP6_DMA	0x50000028	Endpoint 6(14) buffer start address	0xXXXX
R16_UEP7_DMA	0x5000002C	Endpoint 7(15) buffer start address	0xXXXX
R8_UEP0_T_LEN	0x50000030	Endpoint 0 transmit length register	0xXX
R8_UEP0_TX_CTRL	0x50000032	Endpoint 0 transmit control register	0x00
R8_UEP0_RX_CTRL	0x50000033	Endpoint 0 receive control register	0x00
R8_UEP1_T_LEN	0x50000034	Endpoint 1(9) transmit length register	0xXX
R8_UEP1_TX_CTRL	0x50000036	Endpoint 1(9) transmit control register	0x00
R8_UEP1_RX_CTRL	0x50000037	Endpoint 1(9) receive control register	0x00
R8_UEP2_T_LEN	0x50000038	Endpoint 2(10) transmit length register	0xXX
R8_UEP2_TX_CTRL	0x5000003A	Endpoint 2(10) transmit control register	0x00
R8_UEP2_RX_CTRL	0x5000003B	Endpoint 2(10) receive control register	0x00
R8_UEP3_T_LEN	0x5000003C	Endpoint 3(11) transmit length register	0xXX
R8_UEP3_TX_CTRL	0x500003E	Endpoint 3(11) transmit control register	0x00
R8_UEP3_RX_CTRL	0x500003F	Endpoint 3(11) receive control register	0x00
R8_UEP4_T_LEN	0x50000040	Endpoint 4 (8/12) transmit length register	0xXX
R8_UEP4_TX_CTRL	0x50000042	Endpoint 4 (8/12) transmit control register	0x00
R8_UEP4_RX_CTRL	0x50000043	Endpoint 4 (8/12) receive control Register	0x00
R8_UEP5_T_LEN	0x50000044	Endpoint 5(13) transmit length register	0xXX

R8_UEP5_TX_CTRL	0x50000046	Endpoint 5(13) transmit control register	0x00
R8_UEP5_RX_CTRL	0x50000047	Endpoint 5(13) receive control register	0x00
R8_UEP6_T_LEN	0x50000048	Endpoint 6(14) transmit length register	0xXX
R8_UEP6_TX_CTRL	0x5000004A	Endpoint 6(14) transmit control register	0x00
R8_UEP6_RX_CTRL	0x5000004B	Endpoint 6(14) receive control register	0x00
R8_UEP7_T_LEN	0x5000004C	Endpoint 7(15) transmit length register	0xXX
R8_UEP7_TX_CTRL	0x5000004E	Endpoint 7(15) transmit control register	0x00
R8_UEP7_RX_CTRL	0x5000004F	Endpoint 7(15) receive control register	0x00

23.2.2.1 USB Device Physical Port Control Register (R8_UDEV_CTRL)

Bit	Name	Access	Description	Reset value
7	RB_UD_PD_DIS	RW	USB device port UD+/UD- pin internal pull-down resistor control: 1: Disable internal pull-down; 0: Enable internal pull-down. Can be used in GPIO mode to provide pull-down resistors.	1
6	Reserved	RO	Reserved	0
5	RB_UD_DP_PIN	RO	Current UD+ pin status: 1: High level; 0: Low level.	X
4	RB_UD_DM_PIN	RO	Current UD-pin status: 1: High level; 0: Low level.	X
3	Reserved	RO	Reserved	0
2	RB_UD_LOW_SPEED	RW	USB Device physical port low speed mode enable: 1: 1.5Mbps low-speed mode; 0: 12Mbps full-speed mode.	0
1	RB_UD_GP_BIT	RW	USB device mode general-purpose flag, user-defined.	0
0	RB_UD_PORT_EN	RW	USB Device physical port enable: 1: Enable physical port; 0: Disable physical port.	0

23.2.2.2 Endpoint 1(9)/4(8/12) Mode Control Register (R8_UEP4_1_MOD)

Bit	Name	Access	Description	Reset value
7	RB_UEP1_RX_EN	RW	1: Enable endpoint 1(9) to receive (OUT); 0: Endpoint 1(9) reception disabled.	0
6	RB_UEP1_TX_EN	RW	1: Enable endpoint 1(9) transmission (IN); 0: Disable endpoint 1(9) transmission.	0
5	Reserved	RO	Reserved	0
4	RB_UEP1_BUF_MOD	RW	Endpoint 1(9) data buffer mode control. *Note: When this bit is 1, UEP1_RX_EN and UEP1_TX_EN cannot be 1 at the same time.*	0
3	RB_UEP4_RX_EN	RW	1: Enable endpoint 4 (8/12) reception (OUT); 0: Disable endpoint 4 (8/12) reception.	0
2	RB_UEP4_TX_EN	RW	1: Enable endpoint 4 (8/12) transmission (IN); 0: Endpoint 4 (8/12) transmission disabled.	0
1	Reserved	RO	Reserved	0
0	RB_UEP4_BUF_MOD	RW	Endpoint 4 (8/12) data buffer mode control. *Note: When this bit is 1, UEP4_RX_EN and UEP4_TX_EN cannot be 1 at the same time.* *Note: This bit control only supports CH32V103x series.*	0

Note: The endpoint 1 configuration option maps endpoint 9, and the endpoint 4 configuration option maps endpoints 8 and 12.

23.2.2.3 Endpoint 2(10)/3(11) Mode Control Register (R8_UEP2_3_MOD)

Bit	Name	Access	Description	Reset value
7	RB_UEP3_RX_EN	RW	1: Enable endpoint 3 (11) reception (OUT); 0: Disable endpoint 3 (11) reception.	0
6	RB_UEP3_TX_EN	RW	1: Enable endpoint 3 (11) transmission (IN); 0: Disable endpoint 3 (11) transmission.	0
5	Reserved	RO	Reserved	0
4	RB_UEP3_BUF_MOD	RW	Endpoint 3(11) data buffer mode control. *Note: When this bit is 1, UEP3_RX_EN and UEP3_TX_EN cannot be 1 at the same time.*	0
3	RB_UEP2_RX_EN	RW	1: Enable endpoint 2 (10) reception (OUT); 0: Disable endpoint 2 (10) reception.	0
2	RB_UEP2_TX_EN	RW	1: Enable endpoint 2 (10) transmission (IN); 0: Disable endpoint 2 (10) transmission.	0
1	Reserved	RO	Reserved	0
0	RB_UEP2_BUF_MOD	RW	Endpoint 2(10) data buffer mode control. *Note: When this bit is 1, UEP2_RX_EN and*	0

			UEP2_TX_EN cannot be 1 at the same time.	

Note: The endpoint 2 configuration options map to endpoint 10, and the endpoint 3 configuration options map to endpoint 11.

23.2.2.4 Endpoint 5(13)/6(14) Mode Control Register (R8_UEP5_6_MOD)

Bit	Name	Access	Description	Reset value
7	RB_UEP6_RX_EN	RW	1: Enable endpoint 6 (14) reception (OUT); 0: Disable endpoint 6 (14) reception.	0
6	RB_UEP6_TX_EN	RW	1: Enable endpoint 6 (14) transmission (IN); 0: Disable endpoint 6(14) transmission.	0
5	Reserved	RO	Reserved	0
4	RB_UEP6_BUF_MOD	RW	Endpoint 6(14) data buffer mode control. *Note: When this bit is 1, UEP6_RX_EN and UEP6_TX_EN cannot be 1 at the same time.*	0
3	RB_UEP5_RX_EN	RW	1: Enable endpoint 5 (13) reception (OUT); 0: Disable endpoint 5(13) reception.	0
2	RB_UEP5_TX_EN	RW	1: Enable endpoint 5(13) transmission (IN); 0: Disable endpoint 5(13) transmission.	0
1	Reserved	RO	Reserved	0
0	RB_UEP5_BUF_MOD	RW	Endpoint 5(13) data buffer mode control. *Note: When this bit is 1, UEP5_RX_EN and UEP5_TX_EN cannot be 1 at the same time.*	0

Note: The endpoint 5 configuration options map to endpoint 13, and the endpoint 6 configuration options map to endpoint 14.

23.2.2.5 Endpoint 7(15) Mode Control Register (R8_UEP7_MOD)

Bit	Name	Access	Description	Reset value
[7:4]	Reserved	RO	Reserved	0
3	RB_UEP7_RX_EN	RW	1: Enable endpoint 7(15) reception (OUT); 0: Disable endpoint 7 (15) reception.	0
2	RB_UEP7_TX_EN	RW	1: Enable endpoint 7(15) transmission (IN); 0: Disable endpoint 7(15) transmission.	0
1	Reserved	RO	Reserved	0
0	RB_UEP7_BUF_MOD	RW	Endpoint 7(15) data buffer mode control.	0

Note: The endpoint 7 configuration option maps to endpoint 15.

The data buffer mode of USB endpoints 1-15 is configured by the combination of RB_UEPn_RX_EN and RB_UEPn_TX_EN and RB_UEPn_BUF_MOD, please refer to Table 23-4 for details. Among them, in the dual 64-byte buffer mode, the first 64-byte buffer will be selected according to RB_UEP_*_TOG=0 during USB data transmission, and the last 64-byte buffer will be selected according to RB_UEP_*_TOG=1, and RB_UEP_AUTO_TOG=1 can be set to realize automatic switching.

Table 23-4 Endpoint n buffer mode (n=1 to 7)

RB_UEPn_RX_EN	RB_UEPn_TX_EN	RB_UEPn_BUF_MOD	Description: R16_UEPn_DMA is the starting address arranged from low to high
0	0	X	The endpoint is disabled and the R16_UEPn_DMA buffer is not used.
1	0	0	Single 64-byte receive buffer (OUT).
1	0	1	Dual 64-byte receive buffer (OUT), selected by RB_UEP_R_TOG.
0	1	0	Single 64-byte transmit buffer (IN).
0	1	1	Dual 64-byte transmit buffer (IN), selected by RB_UEP_T_TOG.
1	1	0	A single 64-byte receive buffer (OUT), a single 64-byte transmit buffer (IN).
1	1	1	Dual 64-byte receive buffer (OUT), selected by RB_UEP_R_TOG, Dual 64-byte transmit buffer (IN), selected by RB_UEP_T_TOG. All 256 bytes are arranged as follows: UEPn_DMA+0 address: endpoint receiving address when RB_UEP_R_TOG=0; UEPn_DMA+64 address: endpoint receiving address when RB_UEP_R_TOG=1; UEPn_DMA+128 address: endpoint sending address when RB_UEP_T_TOG=0; UEPn_DMA+192 address: endpoint sending address when RB_UEP_T_TOG=1.

Note: The configuration options in Table 21-4 support n=1-7, and endpoint 8-15 configuration maps endpoint 1-7 configuration.

Endpoint n buffer start address (R8_UEPn_DMA) (n=0 to 7)

Bit	Name	Access	Description	Reset Value
[17:0]	R8_UEPn_DMA	RW	Endpoint n buffer start address. The lower 15 bits are valid, and the address must be 4-byte aligned.	X

Note: 1. The length of the buffer for receiving data >= min (maximum packet length that may be received + 2 bytes, 64 bytes).

2. Endpoint DMA configuration supports 0-7 endpoints, and can map and configure endpoints 8-15 endpoints.

23.2.2.6 Endpoint n Transmit Length Register (R16_UEPn_T_LEN) (n=0 to 7)

Bit	Name	Access	Description	Reset value
[9:0]	R8_UEP3_T_LEN	RW	Sets the number of bytes of data that USB endpoint 3 is ready to send.	X
[7:0]	R8_UEPn_T_LEN	RW	Set the number of data bytes that the USB endpoint n is ready to send (n=0, 1, 2, 4, 5, 6, 7).	X

Note 1. Endpoint sending length configuration supports endpoint0 to endpoint7, which can be mapped to configure the sending of endpoint8 to endpoint15.

2. The host sends a maximum of 1023 bytes (for isochronous endpoints)

23.2.2.7 Endpoint n Control Register (R8_UEPn_TX_CTRL) (n=0 to 7)

Bit	Name	Access	Description	Reset value
[7:4]	Reserved	RO	Reserved	0
3	RB_UEP_T_AUTO_TOG	RW	Synchronization trigger bit auto toggle enable: 1: After the data is sent successfully, the corresponding synchronization trigger bit is automatically toggled; 0: No automatic toggle, can be switched manually. *Note: This bit is reserved for endpoint 0.*	0
2	RB_UEP_T_TOG	RW	Synchronization trigger bit prepared by the transmitter of USB endpoint n (handling IN transaction): 1: Send DATA1; 0: Send DATA0.	0
[1:0]	MASK_UEP_T_RES	RW	Response control by the sender of endpoint n to IN transaction: 00: DATA0/DATA1 data is ready and ACK is expected; 01: Reply to DATA0/DATA1 and expect no response, for real-time/synchronous transmission of non-endpoint 0; 10: Answer NAK or busy; 11: Response STALL or error.	00b

Note: Endpoint configuration supports 0-7 endpoints, which can be mapped to configure endpoints 8-15 endpoints.

23.2.2.8 Endpoint n Control Register (R8_UEPn_ RX_CTRL) (n=0-7)

Bit	Name	Access	Description	Reset value
[7:4]	Reserved	RO	Reserved	0
3	RB_UEP_R_AUTO_	RW	Synchronization trigger bit auto toggle	0

	TOG		enable: 1: Automatically toggle the corresponding synchronization trigger bit after the data is successfully received; 0: No automatic toggle, can be switched manually. *Note: This bit is reserved for endpoint 0.*	
2	MASK_UEP_R_TOG	RW	Sync trigger bit expected by receiver of USB endpoint n (handling OUT transaction): 1: DATA1 is expected; 0: DATA0 is expected.	0
[1:0]	MASK_UEP_R_RES	RW	Receiver of endpoint n responds to OUT transaction control: 00: reply ACK; 01: timeout/no response, for real-time/synchronous transmission of non-endpoint 0; 10: Answer NAK or busy; 11: Response STALL or error.	00b

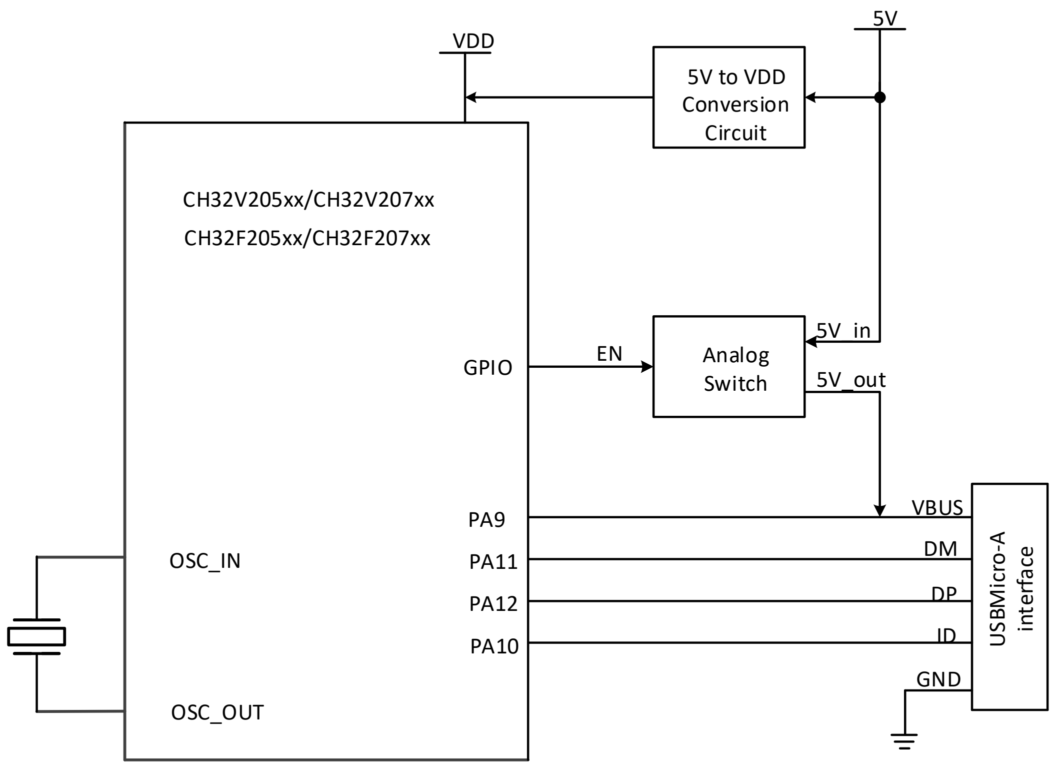
Note: Endpoint configuration supports 0-7 endpoints, which can be mapped to configure endpoints 8-15.

22.2.3 USB host register

In USB OTG host mode, the chip provides a set of bidirectional host endpoints, including a sending endpoint OUT and a receiving endpoint IN, the maximum length of a data packet is 1023 bytes, and supports control transfer, interrupt transfer, bulk transfer and real-time/ synchronous transmission.

In USB OTG host mode, if the controller cannot provide 5V power for VUBS, an external charge pump is required. If the application board can provide 5V power, the analog switch can be used to control the on and off of VBUS, as shown in Figure 23-2.

Figure 23-2 OTG Class A device connection



Each USB transaction initiated by the host endpoint always automatically sets the RB_UIF_TRANSFER interrupt flag after processing. The application can directly query or query and analyze the interrupt flag register R8_USB_INT_FG in the USB interrupt service routine, and perform corresponding processing according to each interrupt flag; and, if RB_UIF_TRANSFER is valid, then continue to analyze the USB interrupt status register R8_USB_INT_ST, process accordingly according to the response PID identification MASK_UIS_H_RES of the current USB transfer transaction .

If the synchronization trigger bit (RB_UH_R_TOG) of the IN transaction of the host receiving endpoint is set in advance, then RB_U_TOG_OK or RB_UIS_TOG_OK can be used to judge whether the synchronization trigger bit of the currently received data packet matches the synchronization trigger bit of the host receiving endpoint. If synchronized, the data is valid; if the data is not synchronized, the data should be discarded. After each USB send or receive interrupt is processed, the synchronization trigger bit of the corresponding host endpoint should be modified correctly to synchronize the data packet sent next time and detect whether the data packet received next time is synchronized; in addition, setting RB_UEP_AUTO_TOG and RB_UH_R_AUTO_TOG enables to automatically flip the corresponding synchronization trigger bit after successful transmission or reception..

The USB host token setting register R8_UH_EP_PID is used to set the endpoint number of the target device to be operated and the token PID packet identification of this USB transfer transaction. The data corresponding to the SETUP token and the OUT token is provided by the host sending endpoint, the data to be sent is in the R16_UH_TX_DMA buffer, and the length of the data to be sent is set in R16_UH_TX_LEN; the data corresponding to the IN token is returned by the target device to the host for reception Endpoint, the received data is stored in the R16_UH_RX_DMA buffer, the received data length is stored in R16_USB_RX_LEN, and the maximum packet length that the host endpoint can receive needs to be written to the R16_UH_RX_MAX_LEN register in advance.

Table 23-5 Host registers

Name	Access address	Description	Reset value
R8_UHOST_CTRL	0x50000001	USB host physical port control register	0xX0
R32_UH_EP_MOD	0x5000000D	USB host endpoint mode control register	0x00
R16_UH_RX_DMA	0x50000018	USB host receive buffer start address	X
R16_UH_TX_DMA	0x5000001C	USB host transmit buffer start address	X
R16_UH_SETUP	0x50000036	USB host auxiliary setup register	0x00
R8_UH_EP_PID	0x50000038	USB host token setup register	0x00
R8_UH_RX_CTRL	0x5000003A	USB host receive endpoint control register	0x00
R16_UH_TX_LEN	0x5000003C	USB host transmit length register	X
R8_UH_TX_CTRL	0x5000003E	USB host transmit endpoint control register	0x00

22.2.3.1 USB Host Physical Port Control Register (R8_UHOST_CTRL)

Bit	Name	Access	Name	Reset value
7	RB_UH_PD_DIS	RW	Internal pull-down resistor control for USB host port UD+/UD- pins: 1: Disable internal pull-down; 0: Enable internal pull-down. Can be used in GPIO mode to provide pull-down resistors.	1
6	Reserved	RO	Reserved	0
5	RB_UH_DP_PIN	RO	Current UD+ pin status: 1: High level; 0: Low level.	X
4	RB_UH_DM_PIN	RO	Current UD-pin status: 1: High level; 0: Low level.	X
3	Reserved	RO	Reserved	0
2	RB_UH_LOW_SPEED	RW	USB host port low-speed mode enable: 1: Select 1.5Mbps low-speed mode; 0: Select 12Mbps full-speed mode.	0
1	RB_UH_BUS_RESET	RW	USB host mode bus reset control: 1: Force output USB bus reset; 0: End output.	0
0	RB_UH_PORT_EN	RW	USB host port enable: 1: Enable the host port; 0: Disable the host port. This bit is automatically cleared to 0 when the USB device is disconnected.	0

22.2.3.2 USB Host Endpoint Mode Control Register (R32_UH_EP_MOD)

Bit	Name	Access	Description	Reset value
7	Reserved	RO	Reserved	0
6	RB_UH_EP_TX_EN	RW	Host transmission endpoint transmit (SETUP/OUT) enable: 1: Enable endpoint transmit; 0: Disable endpoint transmit.	0
5	Reserved	RO	Reserved	0
4	RB_UH_EP_TBUF_MOD	RW	Host transmission endpoint transmit buffer mode control.	0
3	RB_UH_EP_RX_EN	RW	Host reception endpoint receive (IN) enable: 1: Enable endpoint receive; 0: Disable endpoint receive.	0
[2:1]	Reserved	RO	Reserved	0
0	RB_UH_EP_RBUF_MOD	RW	USB host reception endpoint receive buffer mode control	0

The host transmit endpoint buffer mode is controlled by the combination of RB_UH_EP_TX_EN and RB_UH_EP_TBUF_MOD, refer to the table below.

Table 23-6 Host transmit buffer mode

RB_UH_EP_TX_EN	RB_UH_EP_TBUF_MOD	Description: Take R16_UH_TX_DMA as the starting address
0	X	The endpoint is disabled and the R16_UH_TX_DMA buffer is not used.
1	0	Single 64-byte transmit buffer (SETUP/OUT).
1	1	Dual 64-byte transmit buffer, selected by RB_UH_T_TOG: When RB_UH_T_TOG=0, select the first 64-byte buffer; The last 64-byte buffer is selected when RB_UH_T_TOG=1.

The host receive endpoint data buffer mode is controlled by the combination of RB_UH_EP_RX_EN and RB_UH_EP_RBUF_MOD, refer to the table below.

Table23-7 Host receive buffer mode

RB_UH_EP_RX_EN	RB_UH_EP_RBUF_MOD	Structure description: R16_UH_TX_DMA is the starting address
0	X	The endpoint is disabled and the R16_UH_RX_DMA buffer is not used.
1	0	Single 64-byte receive buffer (IN).

1	1	Dual 64-byte receive buffers, selected by RB_UH_R_TOG: When RB_UH_R_TOG=0, select the first 64-byte buffer; The last 64-byte buffer is selected when RB_UH_R_TOG=1.

USB Host Receive Buffer Start Address (R16_UH_RX_DMA)

Bit	Name	Access	Description	Reset value
[15:0]	R16_UH_RX_DMA	RW	Host endpoint data receive buffer start address. The lower 15 bits are valid, and the address must be 4-byte aligned.	X

USB Host Transmit Buffer Start Address (R16_UH_TX_DMA)

Bit	Name	Access	Description	Reset value
[15:0]	R16_UH_TX_DMA	RW	Host endpoint data send buffer start address. The lower 15 bits are valid, and the address must be 4-byte aligned.	X

22.2.3.3 USB Host Auxiliary Setup Register (R16_UH_SETUP)

Bit	Name	Access	Description	Reset value
[15:11]	Reserved	RO	Reserved	0
10	RB_UH_PRE_PID_EN	RW	Low-speed preamble packet PRE PID enable: 1: Enable, used to communicate with low-speed USB devices through an external HUB. 0: Disable low-speed preamble packets.	0
[9:3]	Reserved	RO	Reserved	0
2	RB_UH_SOF_EN	RW	Auto generate SOF packet enable: 1: The host automatically generates a SOF package; 0: Disable the automatic SOF function.	0
[1:0]	Reserved	RO	Reserved	0

22.2.3.4 USB Host Token Setup Register (R8_UH_EP_PID)

Bit	Name	Access	Description	Reset value
[7:4]	MASK_UH_TOKEN	RW	Set the token PID identifier of this USB transfer transaction.	0
[3:0]	MASK_UH_ENDP	RW	Set the endpoint number of the target device to be operated this time.	0

22.2.3.5 USB Host Receive Endpoint Control Register (R8_UH_RX_CTRL)

Bit	Name	Access	Description	Reset value
[7:4]	Reserved	RO	Reserved	0
3	RB_UH_R_AUTO_TOG	RW	Synchronization trigger bit auto toggle enable: 1: Automatically toggle the corresponding expected synchronization trigger bit (RB_UH_R_TOG) after successful data reception; 0: Manual control synchronization trigger bit (RB_UH_R_TOG).	0
2	RB_UH_R_TOG	RW	Synchronization trigger bit prepared by the Master Receiver (handling IN Transactions): 1: no response, for real-time/synchronous transmission of non-zero endpoints; 0: Acknowledge ACK.	0
1	Reserved	RO	Reserved	0
0	RB_UH_R_RES	RO	Master Receiver Response Control Bits to IN Transactions: 1: no response, for real-time/synchronous transmission of non-zero endpoints; 0: Acknowledge ACK.	0

22.2.3.6 USB Host Transmit Length Register (R16_UH_TX_LEN)

Bit	Name	Access	Description	Reset value
[16:0]	R16_UH_TX_LEN	RW	Set the number of bytes of data that the USB host sending endpoint is ready to send.	X

22.2.3.7 USB Host Transmit Endpoint Control Register (R8_UH_TX_CTRL)

Bit	Name	Access	Description	Reset value
[7:4]	Reserved	RO	Reserved	0

3	RB_UH_T_AUTO_TOG	RO	Synchronization trigger bit auto toggle enable: 1: Automatically toggle the corresponding synchronization trigger bit (RB_UH_T_TOG) after the data is sent successfully; 0: Manual control synchronization trigger bit (RB_UH_T_TOG).	0
2	RB_UH_T_TOG	RW	Sync trigger bit prepared by USB host transmitter (handling SETUP/OUT transactions): 1: Send DATA1; 0: Send DATA0.	0
1	Reserved	RO	Reserved	0
0	RB_UH_T_RES	RW	USB Host transmitter response control to SETUP/OUT transactions: 1: expect no response, for real-time/synchronous transmission of non-zero endpoints; 0: ACK is expected.	0

Chapter 24 Controller Area Network (CAN)

The module descriptions in this chapter are applicable to some products of the CH32F2x, CH32V2x and CH32V3x microcontroller series.

Controller Area Network is a high-performance communication protocol for serial data communication. The CAN controller provides a complete implementation of the CAN protocol, supporting CAN protocols 2.0A and 2.0B. The CAN controller can be used to construct a powerful local area network to realize safe distributed real-time control, process a large number of data packets with a small CPU load, and has a wide range of applications in the industrial and automotive fields.

24.1 Main features

- Compatible with CAN specification 2.0A and 2.0B
- Programmable transfer rate up to 1Mbit/s
- Support time-triggered communication function to avoid low-priority message blocking
- Support 3 sending mailboxes, the priority of sending messages can be determined by the message identifier or the order of sending requests, and can record the time stamp of the SOF moment of sending the message
- Support 2 receive FIFOs with 3-level mailbox depth, 28 message filter groups can be configured, high-capacity products CAN1 and CAN2 share 28 filters, each filter group can be configured as 32 or 16-bit mode, mask bit or identifier list mode, which can minimize the software's intervention in packet filtering, flexible FIFO overflow processing, and can record the time stamp of the SOF moment of the received packet
- Occupies 4 interrupt vectors, each interrupt source can be configured independently

24.2 CAN controller working mode

The CAN controller can operate the SLEEP or INRQ bits in the register CAN_CTLR to switch between the initialization mode, sleep mode and normal mode.

24.2.1 Initialization mode

After reset, CAN works in sleep mode by default to reduce power consumption. At this time, the sending and receiving of messages is prohibited, the internal pull-up resistor of the TX pin is enabled, and the TX pin outputs a recessive bit. The INRQ bit in the register CAN_CTLR is set to 1, and the CAN controller is requested to enter the initialization mode. When the INAK bit of the register CAN_STATR is automatically set to 1, the initialization state is successfully entered. Similarly, clear the INRQ bit in the register CAN_CTLR to request the CAN controller to exit the initialization mode. When the INAK bit of the register CAN_STATR is automatically cleared to 0, the initialization state is successfully exited.

The filter group can be initialized in the non-initialization mode, but the FINIT bit of the register CAN_FCTLR must be set to 1, and the reception of the message is prohibited at this time.

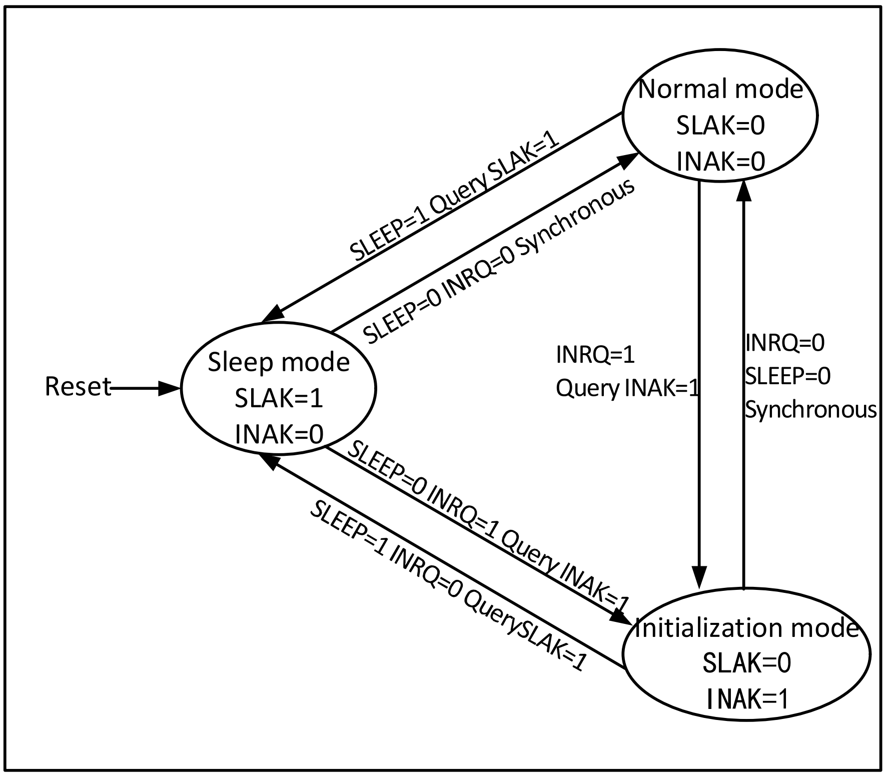
24.2.2 Sleep mode

Set the SLEEP bit in the CAN_CTLR register to 1 to request the CAN controller to enter the sleep mode. When the SNAK bit of the CAN_STATR register is automatically set to 1, the CAN successfully enters the sleep mode. At this time, the clock of the CAN controller stops, but the mailbox register is still accessible.

To enter the initialization mode from sleep mode, the SLEEP bit of CAN_CTLR must be cleared to 0 and the INRQ bit to 1. When the INAK bit of the register CAN_STATR is automatically set to 1, the switch to the initialization state is completed.

To enter normal mode from sleep mode, the SLEEP bit of CAN_CTLR must be cleared to 0. When the SNAK bit of the register CAN_STATR is automatically cleared to 0, the normal mode is entered.

Figure 24-1 CAN working mode switch



24.3 CAN controller test mode

In the initialization mode, operate the SILM and LBKM bits of the register CAN_BTIMR to select a test mode, and then exit the initialization mode and enter the test mode by clearing the INRQ bit of the register CAN_CTLR. There are 3 test modes: silent mode, loopback mode and silent loopback mode.

24.3.1 Silent mode

Setting the SILM bit in register CAN_BTIMR to 1 can optionally enter silent mode. In this mode, the CAN controller can receive, but cannot send messages to the outside world. It is always in a recessive bit to the outside world, which can avoid affecting the bus, but the message can be received by the controller of the node where it is located. Usually, silent mode is used for the status analysis of the CAN bus.

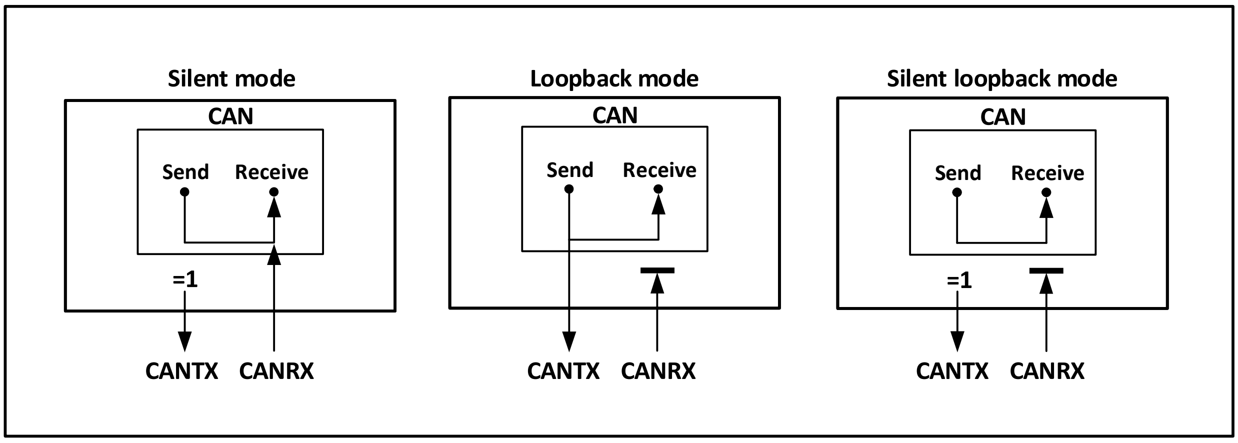
24.3.2 Loopback mode

By setting the LBKM bit of the register CAN_BTIMR to 1, the loopback mode can be selected. In this mode, the CAN controller can send external messages, but cannot receive external messages, but the sent messages can be received by the controller of the node where it is located, and the reception filtering mechanism is effective. Usually, loopback mode is used for transceiver testing of CAN controllers.

24.3.3 Silent loopback mode

Setting the SILM and LBKM bits in register CAN_BTIMR to 1 can optionally enter silent loopback mode. This mode is usually used for the closed self-test of the CAN controller. In this mode, it has no effect on the CAN bus, the RX pin is disconnected from the bus, and the TX pin is set to a recessive bit.

Figure 24-2 CAN 3 test modes of the bus



24.4 MCU working state of CAN controller in debug mode

When the MCU enters the debug mode, the kernel is in a suspended state, but it can be determined whether the CAN controller is in a normal operation or a stop state through the configuration bits in the debug module.

24.5 CAN controller functional description

24.5.1 Transmit processing flow

The transmit processing flow is as follows: If there are vacant mailboxes among the 3 sending mailboxes, the application layer software only has write access to the registers of the vacant mailboxes, and operates the registers CAN_TXMIRx, CAN_TXMDTRx, CAN_TXMDLRx and CAN_TXMDHRx, and can set the message identifier, message length, time stamp, and message data. After the data is ready, the TXRQ bit of the register CAN_TXMIRx is set to 1 to request transmission, the mailbox enters the registered state, and the priority is queued; once it becomes the highest priority mailbox, it becomes the scheduled transmission state and waits for the CAN bus to be idle; when the CAN bus is idle When the message is scheduled to be sent, the message of the mailbox will enter the sending state immediately; after the message is sent, the mailbox will become a vacant mailbox again, and the RQCP and TXOK bits of the register CAN_TSTATR are set to 1 to indicate that the sending is successful; if the arbitration fails during sending, the ALST of the register CAN_TSTATR Set to 1, TERR set to 1 if an error is sent.

24.5.2 Transmit priority

The transmit priority can be determined by the identifier or the order of sending requests. The TXFP position of the register CAN_CTLR is 1 and sent according to the order of sending requests. According to the order of sending requests, it is mainly used for segmented sending; the TXFP bit is cleared to 0 and sent according to the priority of the identifier. In order, the smaller the identifier is, the higher the priority is. In the case of the same identifier, the mailbox with the lower number has a higher priority.

24.5.3 Transmit abort processing

If the ABRQ bit in the register CAN_TSTATR is set, the transmission request can be aborted. When the mailbox status is registered or scheduled to send, the sending request is directly aborted; when the mailbox is in the sending state, the abort request may succeed (stop sending) or fail (sending complete), and the result can be queried by the TXOK bit in the CAN_TSTATR register.

24.5.4 Time-based trigger mode

When the traditional CAN communication bus is busy, it is easy to cause low-priority messages to be blocked for a long time, and even cannot meet the requirements of its time limit. In order to solve the bottleneck, related protocols based on time-triggered mode have been introduced. Such protocols have a certain scale of application in the industry, and the functions based on time-triggered mode are the application of such protocols.

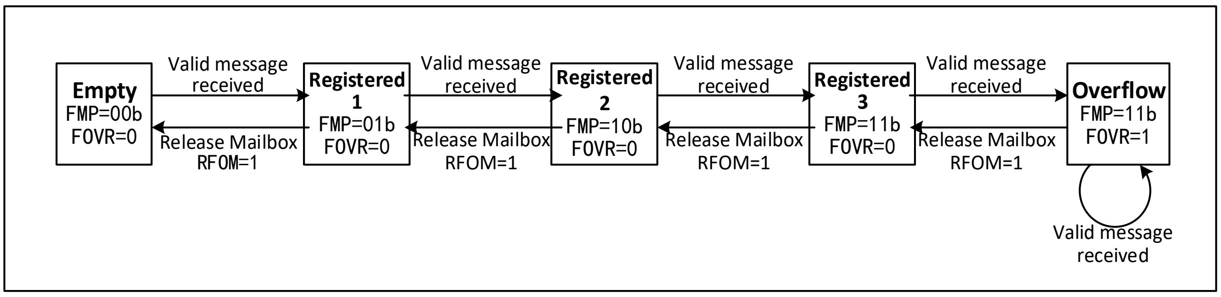
There are 2 modes to choose from in the time-triggered mode. To use this mode, the automatic retransmission function needs to be turned off. The default mode and enhanced mode are selected by configuring the MODE bit of the CAN_TTCTLR register. Set the TTCM and NART bits of the register CAN_CTLR to 1, enable the time-triggered mode and disable automatic retransmission. The MODE bit of the CAN_TTCTLR register defaults to 0. At this time, it works in the default mode, and the internal timer is activated to generate timestamps of the transmit and receive mailboxes. The timer accumulates at the CAN bit time, and the internal timer is sampled and generates a timestamp at the sample point position of the received and transmitted frame start bits. If the enhanced mode is used, the MODE bit of the CAN_TTCTLR register needs to be set to 1 to enable the enhanced mode. Using this mode, there must be 3 or more nodes in the entire CAN network, one of which sends the time reference, and the other nodes receive the time stamp of the reference node, they reset the internal counter by writing 1 to the TIMRST bit of the CAN_TTCTLR register to synchronize the internal counter. So that in addition to the node that sends the time reference, the rest of the CAN nodes achieve time synchronization. Afterwards, write the data to be sent to the sending mailbox, configure the time trigger count value (TIMCNT of the CAN_TTCNT register) and the internal counter count end value (TIMCMV of the CAN_TTCTLR register) of each node in turn . The time-triggered count value and the final count value of the internal counter are determined by the CAN nodes, the CAN communication rate and the number of data bits in a frame. After the configuration is completed, each node waits for the internal counter to count to the time-triggered count value, and then triggers the sending action.

24.5.5 Receive processing flow

The reception of CAN bus messages is completed by the controller hardware without the intervention of the MCU, which reduces the processing load of the MCU. The received messages are stored in 2 FIFOs with 3-level mailbox depth according to the setting of the register CAN_FAFIFOR. If the application layer needs to obtain the message, it can only read the valid received message through the receiving FIFO mailbox.

Initially, the receiving FIFO is empty, and the value of FMR[1:0] in the receiving FIFO register CAN_RFIFOx is binary 00b. After receiving a valid receiving message, it becomes the registered 1 state, and the controller automatically sets the FMR[1:0] in the receiving FIFO register CAN_RFIFOx to binary 01b; if the mailbox data registers CAN_RXMDLRx and CAN_RXMDHRx are read at this time, the mailbox is released by setting the RFOM bit of the receiving FIFO register CAN_RFIFOx to 1, and the receiving FIFO state becomes empty again; if the mailbox is not released in the registered 1 state, after the next valid receiving message is received, the receiving FIFO state switches to the registered 2 state; at this time, the FMR[1:0] of the receiving FIFO register CAN_RFIFOx is automatically set to binary 10b. If the mailbox data register is read and the mailbox is released, then the state returns to registered 1; if the mailbox is not released in the registered 2 state, the receiving FIFO enters the registered 3 state; also in the registered 3 state, the message is read and the mailbox is released, then the registered 2 state is returned; if the registered 3 state is not If the mailbox is released, when the next valid packet is received, packet loss will inevitably occur.

Figure 24-3 Receive FIFO state switching diagram



In the case of message loss above, that is, the receiving FIFO is full, and the message overflow causes the message to be lost. The FOVR bit of the receiving FIFO register CAN_RFIFOx will be automatically set to 1 by hardware for overflow query. When the RFLM bit of the register CAN_CTLR is set to 1, the receiving FIFO locking function is enabled, and the discarded message is a new received message; when the RFLM bit of the register CAN_CTLR is cleared to 0, the receiving FIFO locking function is disabled. Among the 3 original messages of the receiving FIFO, the last received message will be overwritten by the new message.

When the relevant bit of the register CAN_INTENR is set, an interrupt can be generated when the state of the receiving FIFO is switched, so as to process the received message more efficiently, see section 24.6 CAN interrupt for details.

24.5.6 Received message identifier filtering

There are up to 28 filter groups in the module. By setting the filter group, each CAN node can receive the packets that meet the filtering rules, and the packets that do not meet the filtering rules are discarded by hardware without software intervention.

Each filter bank consists of 2 32-bit registers CAN_FxR0 and CAN_FxR1. The bit width of the filter group can be independently configured as a 32-bit filter or 2 16-bit filters by setting each bit of the register CAN_FSCFGR. Each filter group can be configured as mask bit or identifier list mode by setting each bit of register CAN_FMCFGR, and each filter group can be enabled or disabled by setting each bit of register CAN_FWR. Setting each bit of the register CAN_FAFIFOR can select which receive FIFO the message that passes the filter is stored in.

As shown in Table 24-1 below, in the mask bit mode, the 2 registers are the identifier register and the mask register, which need to be used together. Each bit of the identifier register indicates that the expected value of the corresponding bit is dominant or recessive. Each bit of the register indicates whether the corresponding bit needs to be consistent with the expected value of the corresponding identifier register bit.

Table 24-1 32-bit mask bit mode

Identifier register	CAN_FxR1[31:24]	CAN_FxR1[23:16]		CAN_FxR1[15:8]	CAN_FxR1[7:0]			
Mask bit register	CAN_FxR2[31:24]	CAN_FxR2[23:16]		CAN_FxR2[15:8]	CAN_FxR2[7:0]			
Map	STID[10:3]	STID[2:0]	EXID[17:13]	EXID[12:5]	EXID[4:0]	IDE	RTR	0

In the identifier list mode, both registers are used as identifier registers, and each bit of the received message identifier must be consistent with one of the registers to pass the filter.

Table 24-2 32-bit identifier list mode

Identifier register	CAN_FxR1[31:24]	CAN_FxR1[23:16]		CAN_FxR1[15:8]	CAN_FxR1[7:0]			
Mask bit register	CAN_FxR2[31:24]	CAN_FxR2[23:16]		CAN_FxR2[15:8]	CAN_FxR2[7:0]			
Map	STID[10:3]	STID[2:0]	EXID[17:13]	EXID[12:5]	EXID[4:0]	IDE	RTR	0

In 16-bit mode, the register group is divided into 4 registers, and the mask bit mode of each group of filters can have 2 filters, each of which contains a 16-bit identifier register and a 16-bit mask register; all 4 registers are used as identifier registers in identifier list mode.

Table 24-3 16-bit mask bit mode

Identifier register n	CAN_FxR1[15:8]	CAN_FxR1[7:0]			
Mask bit register n	CAN_FxR1[31:24]	CAN_FxR1[23:16]			
Identifier register n+1	CAN_FxR2[15:8]	CAN_FxR2[7:0]			
Mask bit register n+1	CAN_FxR2[31:24]	CAN_FxR2[23:16]			
Map	STID[10:3]	STID[2:0]	RTR	IDE	EXID[17:15]

Table 24-4 16-bit identifier list mode

Identifier register n	CAN_FxR1[15:8]	CAN_FxR1[7:0]			
Mask bit register n	CAN_FxR1[31:24]	CAN_FxR1[23:16]			
Identifier register n+1	CAN_FxR2[15:8]	CAN_FxR2[7:0]			
Mask bit register n+1	CAN_FxR2[31:24]	CAN_FxR2[23:16]			
Map	STID[10:3]	STID[2:0]	RTR	IDE	EXID[17:15]

When the message enters the FIFO mailbox, it will be read and stored by the application program. Usually, the application program distinguishes the message data according to the message identifier. The CAN controller provides a filter number for the messages filtered by different filters in the receiving FIFO, and the number is stored in FMI[7:0] of the CAN_RXMDTRx register, regardless of whether the filter group is enabled or not. The numbering scheme is detailed in the example in Figure 22-4.

When a packet can be filtered by multiple filters, the filter number stored in the receiving mailbox determines which filter number is stored according to the filter priority rules. The filter priority rules are as follows:

- All 32-bit filters have higher priority than 16-bit filters
- For filters of the same width, the filter of the identifier list has higher priority than the filter of the mask bit pattern
- Filters with the same width and pattern, filters with smaller numbers have higher priority

As shown in Figure 22-5: When receiving a message, the identifier is first matched and filtered with the 32-bit identifier list pattern filter. If there is no match, the 32-bit masked bit pattern filter will be matched and filtered . If there is no match, the 16-bit identifier list pattern filter will be matched and filtered . If there is no match, the 16-bit masked bit pattern filter will be matched and filtered . Finally, if there is no match, the message will be discarded. If there is a match, the message will be stored in the mailbox of the receiving FIFO. The identifier number is stored in the FMI in the CAN_RXMDTRx register.

Figure 24-4 Example of filter number

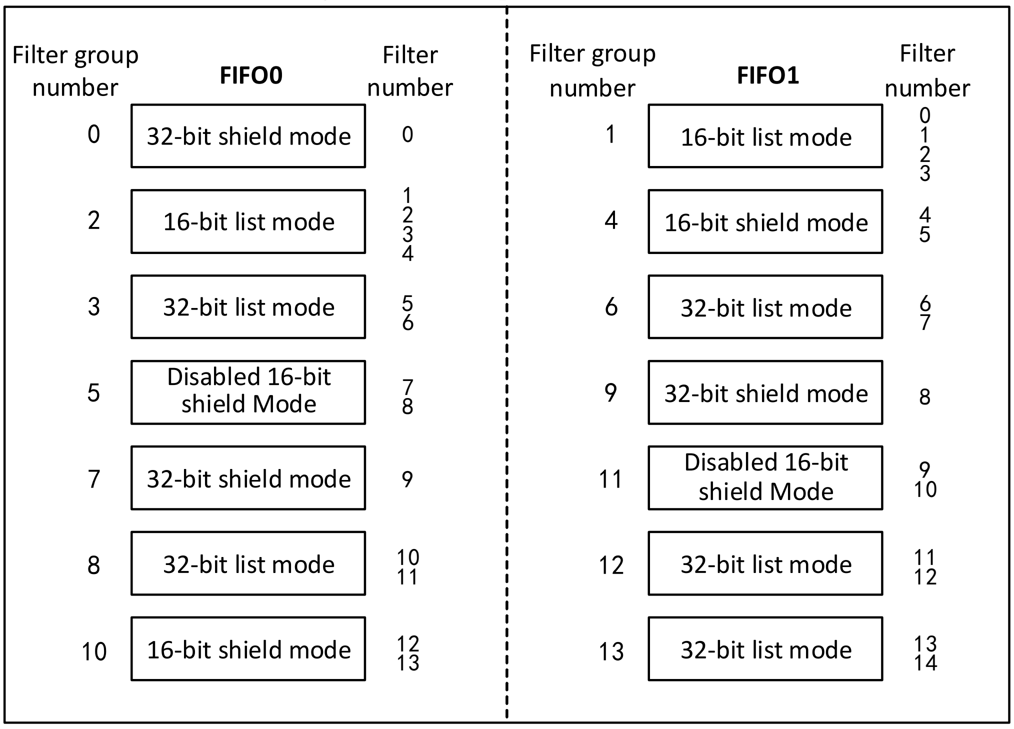
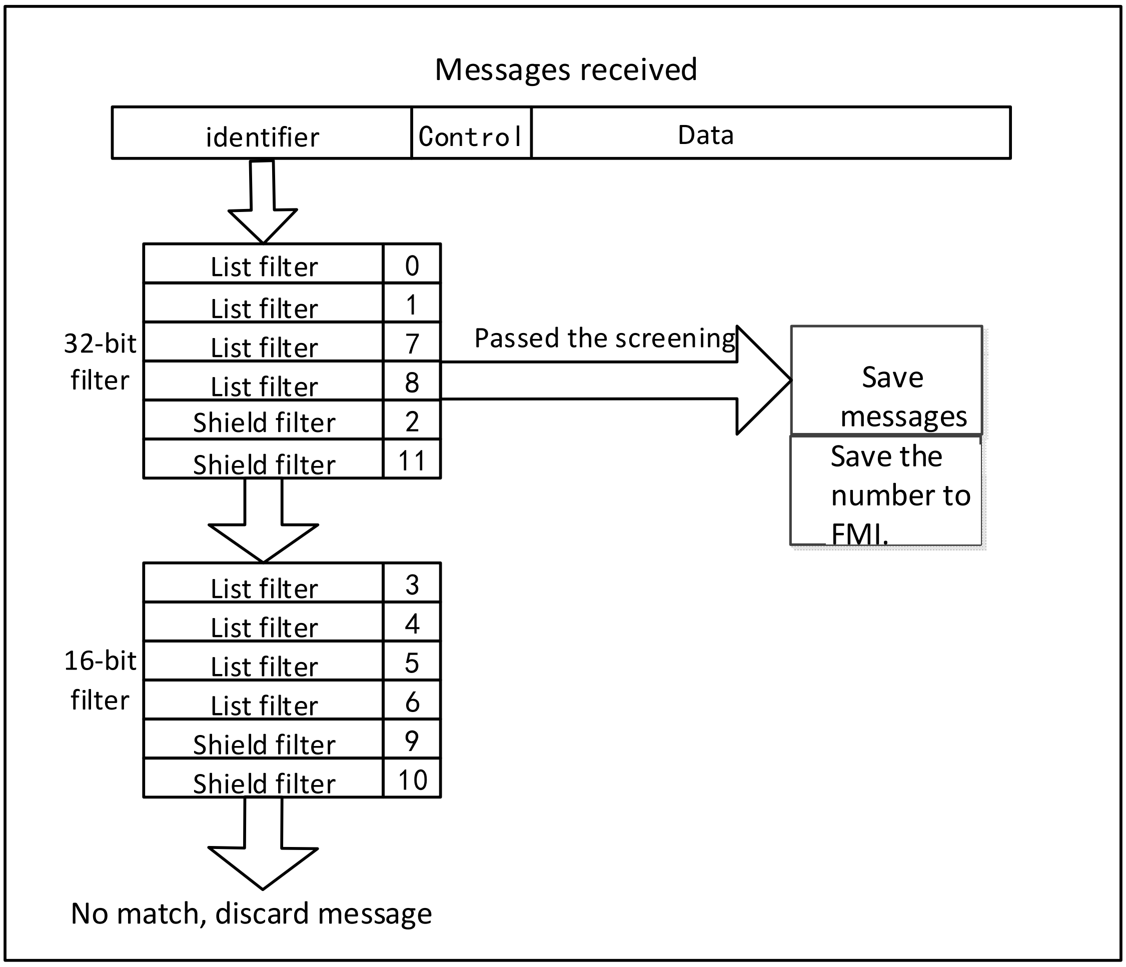


Figure 24-5 Filter example



24.5.7 Error handling

The CAN controller relies on the status error register CAN_ERRSR for error management on the bus. The TEC and REC in the status error register CAN_ERRSR represent the sending and receiving error count values respectively, which increase with the increase of the sending and receiving errors and decrease when the sending and receiving is successful. The stability of the CAN bus can be judged according to their values.

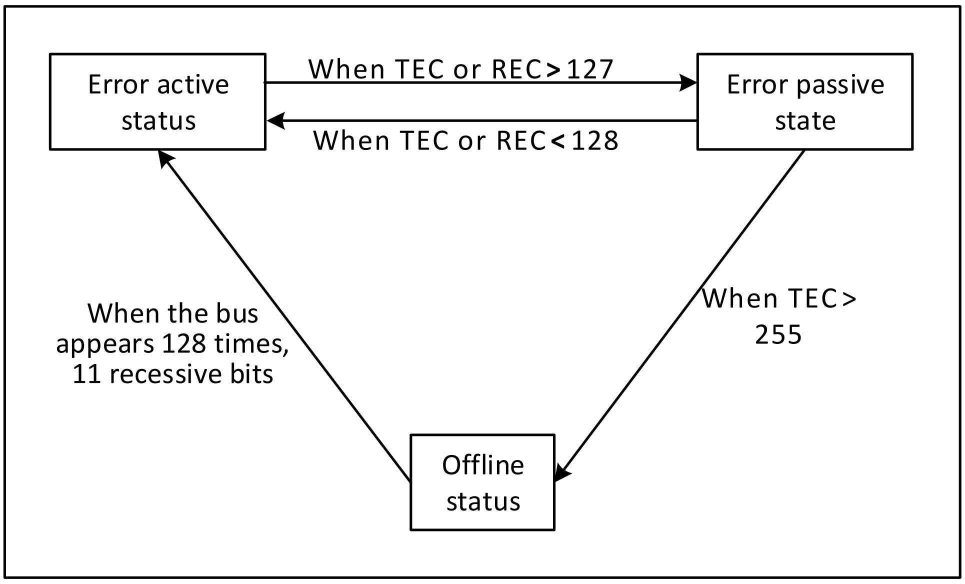
When the TEC and REC in the status error register CAN_ERRSR are less than 128, the current CAN node is in an error active state, can normally participate in bus communication, and sends an active error flag when an error is detected.

When the TEC and REC in the status error register CAN_ERRSR are greater than 127, the current CAN node is in an error passive state, and when an error is detected, it is not allowed to issue an active error flag, but only a passive error flag.

When the TEC in the status error register CAN_ERRSR is greater than 255, the current CAN node enters the offline state.

When the bus monitors the occurrence of 11 consecutive recessive bits for 128 times, it restores to the active state of the error. The restoration method is affected by the ABOM bit in the main control register CAN_CTLR. If ABOM is set to 1, the hardware automatically exits the offline state. If ABOM is 0, the software needs to operate the INRQ bit to enter the initialization mode, and then exit the initialization to exit the offline state.

Figure 24-6 CAN error state switch diagram



24.5.8 Bit timing

According to the CAN bus standard, each bit time is divided into 4 segments: synchronization segment, propagation time segment, phase buffer segment 1 and phase buffer segment 2. These segments consist of minimum time units Tq. The CAN controller monitors CAN bus changes by sampling and synchronizes by the edge of the frame start bit

The CAN controller redivides the above 4 segments into 3 segments, which are:

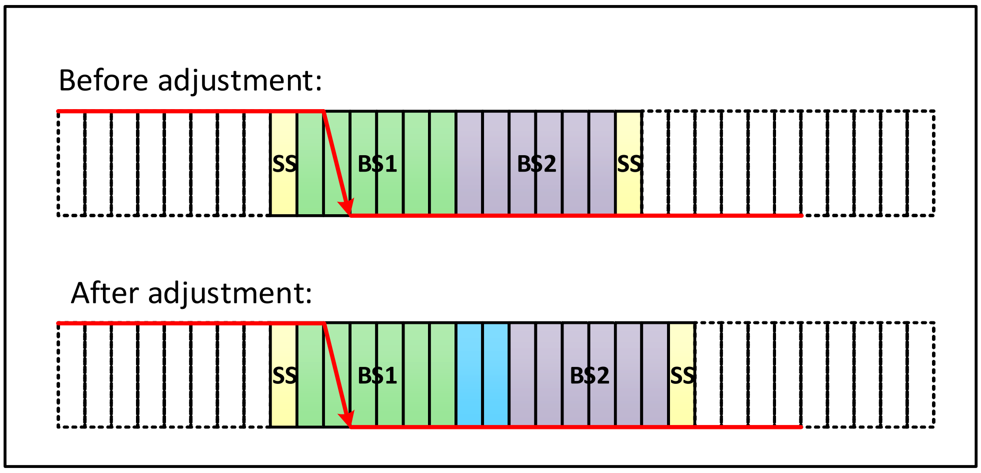
- Synchronization segment (SS): That is, the synchronization segment in the CAN standard, which is fixed as a minimum time unit. Under normal circumstances, the expected bit transition occurs within this time period.

- Bit segment 1 (BS1): contains the propagation time period and phase buffer period 1 in the CAN standard, can be set to contain 1 to 16 minimum time units, and can be automatically extended to compensate for positive phase drift due to frequency accuracy errors at different nodes on the CAN bus. The time period ends at the sampling point location.
- Bit segment 2 (BS2): that is, the phase buffer section 2 in the CAN standard, which can be set to 1 to 8 minimum time units, and can be automatically shortened to compensate for negative phase drift due to frequency accuracy errors at different nodes on the CAN bus.

The resynchronization jump width (SJW) is the upper limit of the minimum number of time units that can be extended and reduced in each bit, and the range can be set to 1 to 4 minimum time units.

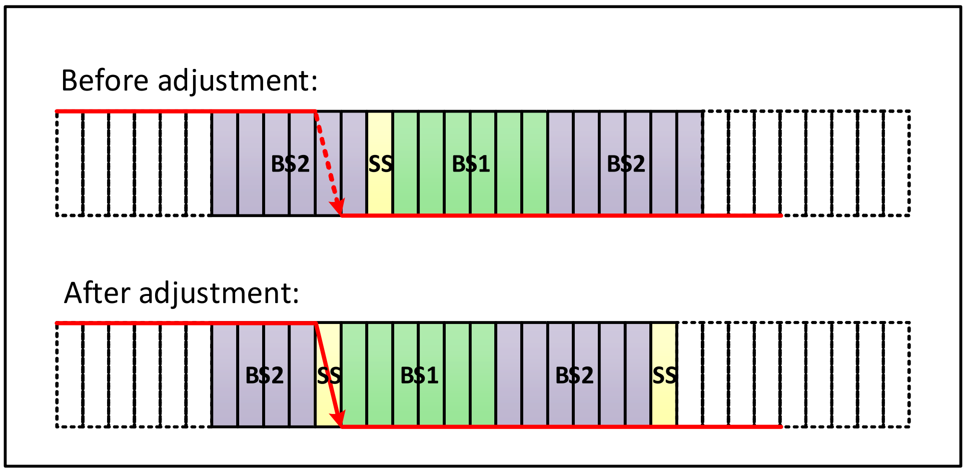
The above parameters can be configured in the CAN bus timing register CAN_BTIMR.

Figure 24-7 The jump appears in BS1



As shown in Figure 24-7, the SJW is 2, and the bus level transition is detected in the time period 1, then the length of the time period 1 needs to be extended, and the maximum SJW is extended, thereby delaying the position of the sampling point.

Figure 24-8 The jump appears in BS2



As shown in Figure 24-8, the SJW is 2, and the bus level transition is detected in the time period 2, so it is necessary to reduce the length of the time period 2 and reduce the SJW at the maximum, so as to advance the position of the sampling point.

The CAN baud rate is calculated by the following formula:

$$\mathrm{CANbps} = \frac{\mathrm{tpclk1}}{(\mathrm{TS1}[3:0] + 1 + \mathrm{TS2}[2:0] + 1 + 1) \times \mathrm{BRP}[9:0]}$$

Here tpclk1 is the APB1 clock cycle and BRP[9:0], TS1[3:0], TS2[2:0] are the corresponding bits of the CANx_BTIMR register.

24.6 CAN interrupt

The CAN controller has 4 interrupt vectors, which are transmit interrupt, FIFO_0 interrupt, FIFO_1 interrupt, error and state change interrupt.

Setting the CAN interrupt enable register CAN_INTENR can enable or disable each interrupt source.

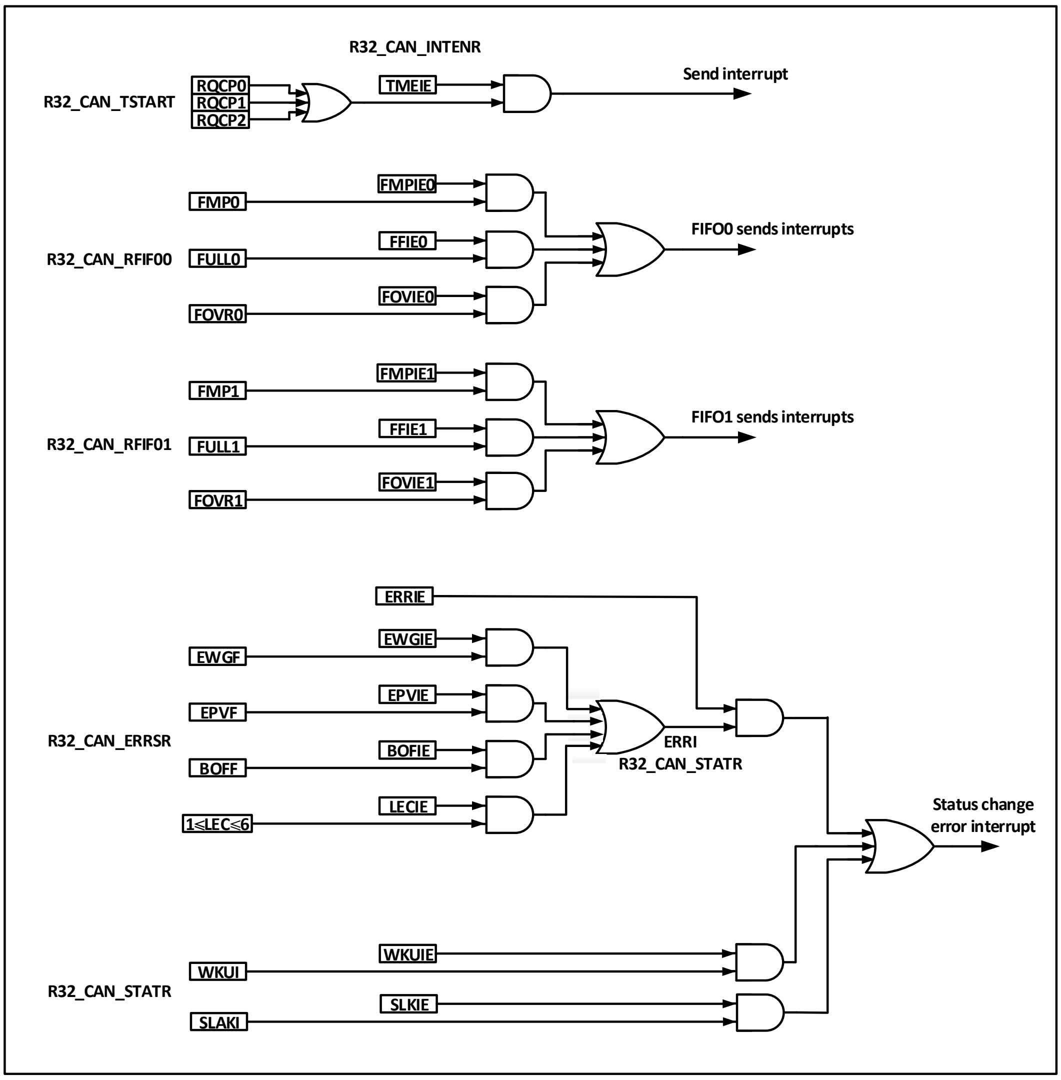
The sending interrupt is generated by the event that the sending mailbox becomes empty. After the interrupt occurs, query the RQCP0, RQCP1 and RQCP2 bits of the register CAN_TSTATR to determine which mailbox becomes empty.

FIFO0 interrupt is generated by receiving new message, receiving mailbox full and overflow event. After the interrupt occurs, query the FMP0, FULL0 and FOVER0 bits of register CAN_RFIFO0 to determine which mailbox becomes empty event.

FIFO1 interrupt is generated by receiving new message, receiving mailbox full and overflow event. After the interrupt occurs, query the FMP1, FULL1 and FOVER1 bits of register CAN_RFIFO1 to determine which mailbox becomes empty event.

Error and state change interrupts are generated by error, wakeup, and sleep events.

Figure 24-9 CAN interrupt logic diagram



24.7 Register description

The registers related to the CAN controller must be manipulated in 32-bit words. In order to avoid the influence of the current node on the entire CAN bus, the application software can only modify the bit timing register CAN_BTIMR in the initialization mode.

Table 24-5 CAN1 registers

Name	Access address	Description	Reset value
R32_CAN1_CTLR	0x40006400	CAN1 master control register	0x00010002
R32_CAN1_STATR	0x40006404	CAN1 master status register	0x00000C02
R32_CAN1_TSTATR	0x40006408	CAN1 transmit status register	0x1C000000
R32_CAN1_RFIFO0	0x4000640C	CAN1 receive FIFO0 control and status register	0x00000000
R32_CAN1_RFIFO1	0x40006410	CAN1 receive FIFO1 control and status register	0x00000000
R32_CAN1_INTENR	0x40006414	CAN1 interrupt enable register	0x00000000
R32_CAN1_ERRSR	0x40006418	CAN1 error status register	0x00000000
R32_CAN1_BTIMR	0x4000641C	CAN1 bit timing register	0x01230000

R32_CAN1_TTCTLR	0x40006420	CAN1 time trigger control register	0x0000FFFF
R32_CAN1_TTCNT	0x40006424	CAN1 time trigger count value register	0x00000000

Table 24-6 CAN2 registers

Name	Access address	Description	Reset value
R32_CAN2_CTLR	0x40006800	CAN2 master control register	0x00010002
R32_CAN2_STATR	0x40006804	CAN2 master status register	0x00000C02
R32_CAN2_TSTATR	0x40006808	CAN2 transmit status register	0x1C000000
R32_CAN2_RFIFO0	0x4000680C	CAN2 receive FIFO0 control and status register	0x00000000
R32_CAN2_RFIFO1	0x40006810	CAN2 receive FIFO1 control and status register	0x00000000
R32_CAN2_INTENR	0x40006814	CAN2 interrupt enable register	0x00000000
R32_CAN2_ERRSR	0x40006818	CAN2 error status register	0x00000000
R32_CAN2_BTIMR	0x4000681C	CAN2 bit timing register	0x01230000
R32_CAN2_TTCTLR	0x40006820	CAN2 time trigger control register	0x0000FFFF
R32_CAN2_TTCNT	0x40006824	CAN2 time trigger count value register	0x00000000

Table 24-7 CAN1 mailbox registers

Name	Access address	Description	Reset value
R32_CAN1_TXMIR0	0x40006580	CAN1 Tx mailbox0 identifier register	X
R32_CAN1_TXMDTR0	0x40006584	CAN1 Tx mailbox0 data length and timestamp register	X
R32_CAN1_TXMDLR0	0x40006588	CAN1 Tx mailbox0 data low register	X
R32_CAN1_TXMDHR0	0x4000658C	CAN1 Tx mailbox0 data high register	X
R32_CAN1_TXMIR1	0x40006590	CAN1 Tx mailbox1 identifier register	X
R32_CAN1_TXMDTR1	0x40006594	CAN1 Tx mailbox1 data length and timestamp register	X
R32_CAN1_TXMDLR1	0x40006598	CAN1 Tx mailbox1 data low register	X
R32_CAN1_TXMDHR1	0x4000659C	CAN1 Tx mailbox1 data high register	X
R32_CAN1_TXMIR2	0x400065A0	CAN1 Tx mailbox2 identifier register	X
R32_CAN1_TXMDTR2	0x400065A4	CAN1 Tx mailbox2 data length and timestamp register	X
R32_CAN1_TXMDLR2	0x400065A8	CAN1 Tx mailbox2 data low register	X
R32_CAN1_TXMDHR2	0x400065AC	CAN1 Tx mailbox2 data high register	X
R32_CAN1_RXMIR0	0x400065B0	CAN1 Rx FIFO0 mailbox identifier register	X
R32_CAN1_RXMDTR0	0x400065B4	CAN1 Rx FIFO0 mailbox data length and timestamp register	X
R32_CAN1_RXMDLR0	0x400065B8	CAN1 Rx FIFO0 mailbox data low register	X
R32_CAN1_RXMDHR0	0x400065BC	CAN1 Rx FIFO0 mailbox data high register	X
R32_CAN1_RXMIR1	0x400065C0	CAN1 Rx FIFO1 mailbox identifier register	X
R32_CAN1_RXMDTR1	0x400065C4	CAN1 Rx FIFO1 mailbox data length and timestamp register	X
R32_CAN1_RXMDLR1	0x400065C8	CAN1 Rx FIFO1 mailbox data low register	X
R32_CAN1_RXMDHR1	0x400065CC	CAN1 Rx FIFO1 mailbox data high register	X

Table 24-8 CAN2 Mailbox registers

Name	Access address	Description	Reset Value
R32_CAN2_TXMIR0	0x40006980	CAN2 Tx mailbox0 identifier register	X
R32_CAN2_TXMDTR0	0x40006984	CAN2 Tx mailbox0 data length and timestamp register	X
R32_CAN2_TXMDLR0	0x40006988	CAN2 Tx mailbox0 data low register	X
R32_CAN2_TXMDHR0	0x4000698C	CAN2 Tx mailbox0 data high register	X
R32_CAN2_TXMIR1	0x40006990	CAN2 Tx mailbox1 identifier register	X
R32_CAN2_TXMDTR1	0x40006994	CAN2 Tx mailbox1 data length and timestamp register	X
R32_CAN2_TXMDLR1	0x40006998	CAN2 Tx mailbox1 data low register	X
R32_CAN2_TXMDHR1	0x4000699C	CAN2 Tx mailbox1 data high register	X
R32_CAN2_TXMIR2	0x400069A0	CAN2 Tx mailbox2 identifier register	X
R32_CAN2_TXMDTR2	0x400069A4	CAN2 Tx mailbox2 data length and timestamp register	X
R32_CAN2_TXMDLR2	0x400069A8	CAN2 Tx mailbox2 data low register	X
R32_CAN2_TXMDHR2	0x400069AC	CAN2 Tx mailbox2 data high register	X
R32_CAN2_RXMIR0	0x400069B0	CAN2 Rx FIFO0 mailbox identifier register	X
R32_CAN2_RXMDTR0	0x400069B4	CAN2 Rx FIFO0 mailbox data length and timestamp register	X
R32_CAN2_RXMDLR0	0x400069B8	CAN2 Rx FIFO0 mailbox data low register	X
R32_CAN2_RXMDHR0	0x400069BC	CAN2 Rx FIFO0 mailbox data high register	X
R32_CAN2_RXMIR1	0x400069C0	CAN2 Rx FIFO1 mailbox identifier register	X
R32_CAN2_RXMDTR1	0x400069C4	CAN2 Rx FIFO1 mailbox data length and timestamp register	X
R32_CAN2_RXMDLR1	0x400069C8	CAN2 Rx FIFO1 mailbox data low register	X
R32_CAN2_RXMDHR1	0x400069CC	CAN2 Rx FIFO1 mailbox data high register	X

Table 24-9 CAN1 Filter registers

Name	Access address	Description	Reset value
R32_CAN1_FCTLR	0x40006600	CAN1 filter master control register	0x2A1C0E01
R32_CAN1_FMCFGR	0x40006604	CAN1 filter mode register	0x00000000
R32_CAN1_FSCFGR	0x4000660C	CAN1 filter bit width register	0x00000000
R32_CAN1_FAFIFOR	0x40006614	CAN1 filter FIFO association register	0x00000000
R32_CAN1_FWR	0x4000661C	CAN1 filter activation register	0x00000000
R32_CAN1_F0R1	0x40006640	CAN1 filter Bank0 register 1	X
R32_CAN1_F0R2	0x40006644	CAN1 filter Bank0 register 2	X
R32_CAN1_F1R1	0x40006648	CAN1 filter Bank1 register 1	X
R32_CAN1_F1R2	0x4000664C	CAN1 filter Bank1 register 2	X
R32_CAN1_F2R1	0x40006650	CAN1 filter Bank2 register 1	X
R32_CAN1_F2R2	0x40006654	CAN1 filter Bank2 register 2	X
R32_CAN1_F3R1	0x40006658	CAN1 filter Bank3 register 1	X
R32_CAN1_F3R2	0x4000665C	CAN1 filter Bank3 register 2	X
R32_CAN1_F4R1	0x40006660	CAN1 filter Bank4 register 1	X

R32_CAN1_F4R2	0x40006664	CAN1 filter Bank4 register 2	X
R32_CAN1_F5R1	0x40006668	CAN1 filter Bank5 register 1	X
R32_CAN1_F5R2	0x4000666C	CAN1 filter Bank5 register 2	X
R32_CAN1_F6R1	0x40006670	CAN1 filter Bank6 register 1	X
R32_CAN1_F6R2	0x40006674	CAN1 filter Bank6 register 2	X
R32_CAN1_F7R1	0x40006678	CAN1 filter Bank7 register 1	X
R32_CAN1_F7R2	0x4000667C	CAN1 filter Bank7 register 2	X
R32_CAN1_F8R1	0x40006680	CAN1 filter Bank8 register 1	X
R32_CAN1_F8R2	0x40006684	CAN1 filter Bank8 register 2	X
R32_CAN1_F9R1	0x40006688	CAN1 filter Bank9 register 1	X
R32_CAN1_F9R2	0x4000668C	CAN1 filter Bank9 register 2	X
R32_CAN1_F10R1	0x40006690	CAN1 filter Bank10 register 1	X
R32_CAN1_F10R2	0x40006694	CAN1 filter Bank10 register 2	X
R32_CAN1_F11R1	0x40006698	CAN1 filter Bank11 register 1	X
R32_CAN1_F11R2	0x4000669C	CAN1 filter Bank11 register 2	X
R32_CAN1_F12R1	0x400066A0	CAN1 filter Bank12 register 1	X
R32_CAN1_F12R2	0x400066A4	CAN1 filter Bank12 register 2	X
R32_CAN1_F13R1	0x400066A8	CAN1 filter Bank13 register 1	X
R32_CAN1_F13R2	0x400066AC	CAN1 filter Bank13 register 2	X
R32_CAN1_F14R1	0x400066B0	CAN1 filter Bank14 register 1	X
R32_CAN1_F14R2	0x400066B4	CAN1 filter Bank14 register 2	X
R32_CAN1_F15R1	0x400066B8	CAN1 filter Bank15 register 1	X
R32_CAN1_F15R2	0x400066BC	CAN1 filter Bank15 register 2	X
R32_CAN1_F16R1	0x400066C0	CAN1 filter Bank16 register 1	X
R32_CAN1_F16R2	0x400066C4	CAN1 filter Bank16 register 2	X
R32_CAN1_F17R1	0x400066C8	CAN1 filter Bank17 register 1	X
R32_CAN1_F17R2	0x400066CC	CAN1 filter Bank17 register 2	X
R32_CAN1_F18R1	0x400066D0	CAN1 filter Bank18 register 1	X
R32_CAN1_F18R2	0x400066D4	CAN1 filter Bank18 register 2	X
R32_CAN1_F19R1	0x400066D8	CAN1 filter Bank19 register 1	X
R32_CAN1_F19R2	0x400066DC	CAN1 filter Bank19 register 2	X
R32_CAN1_F20R1	0x400066E0	CAN1 filter Bank20 register 1	X
R32_CAN1_F20R2	0x400066E4	CAN1 filter Bank20 register 2	X
R32_CAN1_F21R1	0x400066E8	CAN1 filter Bank21 register 1	X
R32_CAN1_F21R2	0x400066EC	CAN1 filter Bank21 register 2	X
R32_CAN1_F22R1	0x400066F0	CAN1 filter Bank22 register 1	X
R32_CAN1_F22R2	0x400066F4	CAN1 filter Bank22 register 2	X
R32_CAN1_F23R1	0x400066F8	CAN1 filter Bank23 register 1	X
R32_CAN1_F23R2	0x400066FC	CAN1 filter Bank23 register 2	X
R32_CAN1_F24R1	0x40006700	CAN1 filter Bank24 register 1	X
R32_CAN1_F24R2	0x40006704	CAN1 filter Bank24 register 2	X
R32_CAN1_F25R1	0x40006708	CAN1 filter Bank25 register 1	X
R32_CAN1_F25R2	0x4000670C	CAN1 filter Bank25 register 2	X
R32_CAN1_F26R1	0x40006710	CAN1 filter Bank26 register 1	X
R32_CAN1_F26R2	0x40006714	CAN1 filter Bank26 register 2	X

R32_CAN1_F27R1	0x40006718	CAN1 filter Bank27 register 1	X
R32_CAN1_F27R2	0x4000671C	CAN1 filter Bank27 register 2	X

Table 24-10 CAN2 Filter registers

Name	Access address	Description	Reset value
R32_CAN2_FCTLR	0x40006A00	CAN2 filter master control register	0x2A1C0E01
R32_CAN2_FMCFGR	0x40006A04	CAN2 filter mode register	0x00000000
R32_CAN2_FSCFGR	0x40006A0C	CAN2 filter bit width register	0x00000000
R32_CAN2_FAFIFOR	0x40006A14	CAN2 filter FIFO associated register	0x00000000
R32_CAN2_FWR	0x40006A1C	CAN2 filter activation register	0x00000000
R32_CAN2_F0R1	0x40006A40	CAN2 filter Bank0 register 1	X
R32_CAN2_F0R2	0x40006A44	CAN2 filter Bank0 register 2	X
R32_CAN2_F1R1	0x40006A48	CAN2 filter Bank1 register 1	X
R32_CAN2_F1R2	0x40006A4C	CAN2 filter Bank1 register 2	X
R32_CAN2_F2R1	0x40006A50	CAN2 filter Bank2 register 1	X
R32_CAN2_F2R2	0x40006A54	CAN2 filter Bank2 register 2	X
R32_CAN2_F3R1	0x40006A58	CAN2 filter Bank3 register 1	X
R32_CAN2_F3R2	0x40006A5C	CAN2 filter Bank3 register 2	X
R32_CAN2_F4R1	0x40006A60	CAN2 filter Bank4 register 1	X
R32_CAN2_F4R2	0x40006A64	CAN2 filter Bank4 register 2	X
R32_CAN2_F5R1	0x40006A68	CAN2 filter Bank5 register 1	X
R32_CAN2_F5R2	0x40006A6C	CAN2 filter Bank5 register 2	X
R32_CAN2_F6R1	0x40006A70	CAN2 filter Bank6 register 1	X
R32_CAN2_F6R2	0x40006A74	CAN2 filter Bank6 register 2	X
R32_CAN2_F7R1	0x40006A78	CAN2 filter Bank7 register 1	X
R32_CAN2_F7R2	0x40006A7C	CAN2 filter Bank7 register 2	X
R32_CAN2_F8R1	0x40006A80	CAN2 filter Bank8 register 1	X
R32_CAN2_F8R2	0x40006A84	CAN2 filter Bank8 register 2	X
R32_CAN2_F9R1	0x40006A88	CAN2 filter Bank9 register 1	X
R32_CAN2_F9R2	0x40006A8C	CAN2 filter Bank9 register 2	X
R32_CAN2_F10R1	0x40006A90	CAN2 filter Bank10 register 1	X
R32_CAN2_F10R2	0x40006A94	CAN2 filter Bank10 register 2	X
R32_CAN2_F11R1	0x40006A98	CAN2 filter Bank11 register 1	X
R32_CAN2_F11R2	0x40006A9C	CAN2 filter Bank11 register 2	X
R32_CAN2_F12R1	0x40006AA0	CAN2 filter Bank12 register 1	X
R32_CAN2_F12R2	0x40006AA4	CAN2 filter Bank12 register 2	X
R32_CAN2_F13R1	0x40006AA8	CAN2 filter Bank13 register 1	X
R32_CAN2_F13R2	0x40006AAC	CAN2 filter Bank13 register 2	X
R32_CAN2_F14R1	0x40006AB0	CAN2 filter Bank14 register 1	X
R32_CAN2_F14R2	0x40006AB4	CAN2 filter Bank14 register 2	X
R32_CAN2_F15R1	0x40006AB8	CAN2 filter Bank15 register 1	X
R32_CAN2_F15R2	0x40006ABC	CAN2 filter Bank15 register 2	X
R32_CAN2_F16R1	0x40006AC0	CAN2 filter Bank16 register 1	X
R32_CAN2_F16R2	0x40006AC4	CAN2 filter Bank16 register 2	X
R32_CAN2_F17R1	0x40006AC8	CAN2 filter Bank17 register 1	X

R32_CAN2_F17R2	0x40006ACC	CAN2 filter Bank17 register 2	X
R32_CAN2_F18R1	0x40006AD0	CAN2 filter Bank18 register 1	X
R32_CAN2_F18R2	0x40006AD4	CAN2 filter Bank18 register 2	X
R32_CAN2_F19R1	0x40006AD8	CAN2 filter Bank19 register 1	X
R32_CAN2_F19R2	0x40006ADC	CAN2 filter Bank19 register 2	X
R32_CAN2_F20R1	0x40006AE0	CAN2 filter Bank20 register 1	X
R32_CAN2_F20R2	0x40006AE4	CAN2 filter Bank20 register 2	X
R32_CAN2_F21R1	0x40006AE8	CAN2 filter Bank21 register 1	X
R32_CAN2_F21R2	0x40006AEC	CAN2 filter Bank21 register 2	X
R32_CAN2_F22R1	0x40006AF0	CAN2 filter Bank22 register 1	X
R32_CAN2_F22R2	0x40006AF4	CAN2 filter Bank22 register 2	X
R32_CAN2_F23R1	0x40006AF8	CAN2 filter Bank23 register 1	X
R32_CAN2_F23R2	0x40006AFC	CAN2 filter Bank23 register 2	X
R32_CAN2_F24R1	0x40006B00	CAN2 filter Bank24 register 1	X
R32_CAN2_F24R2	0x40006B04	CAN2 filter Bank24 register 2	X
R32_CAN2_F25R1	0x40006B08	CAN2 filter Bank25 register 1	X
R32_CAN2_F25R2	0x40006B0C	CAN2 filter Bank25 register 2	X
R32_CAN2_F26R1	0x40006B10	CAN2 filter Bank26 register 1	X
R32_CAN2_F26R2	0x40006B14	CAN2 filter Bank26 register 2	X
R32_CAN2_F27R1	0x40006B18	CAN2 filter Bank27 register 1	X
R32_CAN2_F27R2	0x40006B1C	CAN2 filter Bank27 register 2	X

24.7.1 CANx Master Control Register (CANx_CTLR) (x=1/2)

Offset address: 0x00

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESET	Reserved							TTCM	ABOM	AWUM	NART	RFLM	TXFP	SLEEP	INRQ

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
16	DBF	RW	Debug whether to disable CAN bus from working 1: During debugging, the CAN transceiver is prohibited, but the control and read and write operations of the receiving FIFO are normal; 0: When debugging, the CAN controller works normally.	1
15	RESET	RW1	CAN controller software reset request. Writing 0 to this bit is invalid 1: Reset the CAN controller. After reset, the controller enters sleep mode, and then the	0

			hardware automatically clears it to 0; 0: The CAN controller is in normal state.	
[14:8]	Reserved	RO	Reserved	0
7	TTCM	RW	Whether to allow time trigger mode 1: Enable time trigger mode; 0: Disable time trigger mode. The time trigger mode is mainly used with the TTCAN protocol.	0
6	ABOM	RW	Offline automatic exit control 1: The hardware detects 11 consecutive implicit bits 128 times and automatically exits the offline state; 0: The INRQ bit of the software operation register CAN_CTLR is required to be set to 1 and then cleared to 0. After 11 consecutive implicit bits are detected 128 times, the offline state is exited.	0
5	AWUM	RW	CAN controller automatic wake-up enable 1: When a message is detected, the hardware automatically wakes up, and the SLEEP and SLAK bits of the register CAN_STATR are automatically cleared to 0; 0: Software operation is required to clear the SLEEP bit of the CAN_CTLR register to wake up the CAN controller.	0
4	NART	RW	The automatic packet retransmission function is disabled 1: No matter whether the transmission is successful or not, the message can only be sent once; 0: The CAN controller keeps retransmitting until the transmission is successful.	0
3	RFLM	RW	Receive FIFO message lock mode enable 1: When the receiving FIFO overflows, the received mailbox message is not read, and the newly received message is discarded when the mailbox is not released; 0: When the receiving FIFO overflows, the received mailbox message is not read out, and when the mailbox is not released, the new received message will overwrite the original message.	0
2	TXFP	RW	Send mailbox priority method selection 1: The priority is determined by the order in which the requests are sent; 0: The priority is determined by the message	0

			identifier.	
1	SLEEP	RW	Sleep mode request 1: Set to 1 to request the CAN controller to enter sleep mode. After the current activity is completed, the controller enters sleep mode. If the AWUM bit is set to 1, the controller clears the SLEEP bit to 0 when a message is received; 0: After the software clears to 0, the controller exits the sleep mode.	1
0	INRQ	RW	Initialize mode request 1: Set to 1 to request the CAN controller to enter the initialization mode. After the current activity is completed, the controller enters the initialization mode, and the hardware sets the INAK bit of the register CAN_STATR to 1; 0: Set to 0 to request the CAN controller to exit the initialization mode and enter the normal mode, and the hardware clears the INAK bit of the CAN_STATR register to 0.	0

24.7.2 CANx Master Status Register (CANx_STATR) (x=1/2)

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				RX	SAMP	RXM	TXM	Reserved			SLAKI	WKUI	ERRI	SLAK	INAK

Bit	Name	Access	Description	Reset value
[31:12]	Reserved	RO	Reserved	0
11	RX	RO	Current actual level of CAN controller Rx pin.	1
10	SAMP	RO	Level of a receive bit on the CAN controller RX pin	1
9	RXM	RO	Rx mode query 1: CAN controller is in Rx mode currently; 0: CAN controller is not in Rx mode currently.	0
8	TXM	RO	Tx mode query 1: CAN controller is in Tx mode currently; 0: CAN controller is not in Tx mode currently.	0
[7:5]	Reserved	RO	Reserved	0
4	SLAKI	RW1	Sleep interrupt enable, that is, interrupt generation flag when the SLKIE bit in the CAN_INTENR register is set to 1, write 1 to clear it, and writing 0 is invalid. 1: When entering sleep mode, an interrupt is generated and the hardware is set to 1;	0

			0: When exiting sleep mode, it can be cleared by hardware or by software.	
3	WKUI	RW1	Wake-up interrupt flag. When the WKUI bit in the CAN_INTENR register is set to 1, if the SOF bit is detected when the CAN controller is in sleep mode, the hardware will set it to 1. Set to 1 by software to clear to 0, and set to 0 is invalid.	0
2	ERRI	RW1	Error interrupt. When the ERRIE bit in the CAN_INTENR register is set to 1, an error and status change interrupt is generated. This bit is set to 1 and cleared to 0 by software, and set to 0 is invalid.	0
1	SLAK	RO	Sleep mode indication. 1: CAN controller is in sleep mode; 0: CAN controller is not in sleep mode.	1
0	INAK	RO	Initialization mode indication. 1: CAN controller is in initialization mode; 0: CAN controller is not in initialization mode.	0

24.7.3 CANx Transmit Status Register (CANx_TSTATR) (x=1/2)

Offset Address: 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOW2	LOW1	LOW0	TME2	TME1	TME0	CODE[1:0]		ABRQ2	Reserved			TERR2	ALST2	TXOK2	RQCP2
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ABRQ1	Reserved			TERR1	ALST1	TXOK1	RQCP1	ABRQ0	Reserved			TERR0	ALST0	TXOK0	RQCP0

Bit	Name	Access	Description	Reset value
31	LOW2	RO	Lowest priority flag for Tx mailbox2 1: Tx mailbox2 has the lowest priority; 0: The priority of Tx mailbox2 is not the lowest.	0
30	LOW1	RO	Lowest priority flag for Tx mailbox1 1: The priority of Tx mailbox1 is the lowest; 0: The priority of Tx mailbox1 is not the lowest.	0
29	LOW0	RO	Lowest priority flag for Tx mailbox0 1: The priority of Tx mailbox0 is the lowest; 0: The priority of Tx mailbox0 is not the lowest.	0
28	TME2	RO	Empty flag bit for Tx mailbox2 1: No message waiting to be sent by the Tx mailbox2; 0: Tx mailbox2 has a message waiting to be sent.	1
27	TME1	RO	Empty flag bit for Tx mailbox1 1: No message waiting to be sent by the Tx mailbox1; 0: Tx mailbox1 has a message waiting to be sent.	1

26	TME0	RO	Empty flag bit for Tx mailbox0 1: No message waiting to be sent by the Tx mailbox0; 0: Tx mailbox0 has a message waiting to be sent.	1
[25:24]	CODE	RO	Mailbox code When more than one mailbox is empty, it indicates the next empty mailbox number; when all mailboxes are empty, it indicates the mailbox number with the lowest priority.	0
23	ABRQ2	RW1	Send a send abort request for mailbox2. Set to 1 by software to abort the sending request of mailbox 2. When the sent message is cleared, the hardware clears it to 0. If mailbox 2 is cleared, the software is invalidated by setting 1.	0
[22:20]	Reserved	RO	Reserved	0
19	TERR2	RW1	Tx mailbox2 transmit error. When Tx mailbox2 fails, this bit is automatically set to 1. Set to 1 by software to clear, software write 0 is invalid.	0
18	ALST2	RW1	Tx mailbox 2 arbitration failure flag. When the Tx mailbox 2 has a low arbitration priority and fails to send, this bit is automatically set to 1. Set to 1 by software to clear, software write 0 is invalid.	0
17	TXOK2	RW1	Tx mailbox 2 transmit OK. 1: The last transmission was successful; 0: The last transmission failed. Set to 1 by software to clear, software write 0 is invalid.	0
16	RQCP2	RW	Tx mailbox 2 request completion flag, this bit is automatically set to 1 when the sending or aborting request of sending mailbox 2 is completed. Set to 1 by software to clear, software write 0 is invalid.	0
15	ABRQ1	RW0	Send a send abort request for mailbox 1. Set to 1 by software to abort the sending request of mailbox 1, and reset to 0 by hardware when the sent message is cleared. Software write 0 is invalid.	0
[14:12]	Reserved	RO	Reserved	0
11	TERR1	RW1	Tx mailbox1 transmit error. When Tx mailbox1 fails, this bit is automatically set to 1. Set to 1 by software to clear, software write 0 is invalid.	0
10	ALST1	RW1	Tx mailbox1 arbitration failure flag. When the Tx mailbox1 has a low arbitration priority and fails to send, this bit is automatically set to 1.	0
9	TXOK1	RW1	Tx mailbox1 transmit OK. 1: The last transmission was successful;	0

			0: The last transmission failed. Set to 1 by software to clear, software write 0 is invalid.	
8	RQCP1	RW	Tx mailbox1 request completion flag, this bit is automatically set to 1 when the sending or aborting request of sending mailbox 1 is completed. Set to 1 by software to clear, software write 0 is invalid.	0
7	ABRQ0	RW0	Send a send abort request for mailbox 0. Set to 1 by software to abort the sending request of mailbox 0, and reset to 0 by hardware when the sent message is cleared. Software write 0 is invalid.	0
[6:4]	Reserved	RO	Reserved	0
3	TERR0	RW1	Tx mailbox0 transmit error. When Tx mailbox0 fails, this bit is automatically set to 1. Set to 1 by software to clear, software write 0 is invalid.	0
2	ALST0	RW1	Tx mailbox0 arbitration failure flag. When Tx mailbox0 has a low arbitration priority and fails to send, this bit is automatically set to 1. Set to 1 by software to clear, software write 0 is invalid.	0
1	TXOK0	RW1	Tx mailbox0 transmit OK. 1: The last transmission was successful; 0: The last transmission failed. Set to 1 by software to clear, software write 0 is invalid.	0
0	RQCP0	RW	Tx mailbox0 request completion flag. When the send or abort request of Tx mailbox0 is completed, this bit is automatically set to 1. Set to 1 by software to clear, software write 0 is invalid.	0

24.7.4 CANx Receive FIFO 0 Status Register (CANx_RFIFO0) (x=1/2)

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										RFOM0	FOVR0	FULL0	Reserved	FMP0[1:0]	

Bit	Name	Access	Description	Reset value
[31:6]	Reserved	RO	Reserved	0
5	RFOM0	RW1	When the software sets this bit to 1, it releases the current mailbox message of the receiving	0

			FIFO_0, and automatically clears it to 0 after the release, and software write 0 is invalid.	
4	FOVR0	RW1	Receive FIFO_0 overflow flag. When there are 3 messages in FIFO_0, a new message is received, and the hardware is set to 1. This bit needs software to be set to 1 and cleared to 0, and software write 0 is invalid.	0
3	FULL0	RW1	Receive FIFO_0 full flag. Set by hardware when there are 3 messages in FIFO_0. This bit needs software to be set to 1 and cleared to 0, and software write 0 is invalid.	0
2	Reserved	RO	Reserved	0
[1:0]	FMP0[1:0]	RO	Number of received FIFO_0 messages.	0

24.7.5 CANx Receive FIFO 1 Status Register (CANx_RFIFO1) (x=1/2)

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										RFOM1	FOVR1	FULL1	Reserved	FMP1[1:0]	

Bit	Name	Access	Description	Reset value
[31:6]	Reserved	RO	Reserve.	0
5	RFOM1	RW1	When the software sets this bit to 1, the current mailbox message of the receiving FIFO_1 is released, and it is automatically cleared to 0 after the release, and software write 0 is invalid.	0
4	FOVR1	RW1	Receive FIFO_1 overflow flag. When there are 3 messages in FIFO_1, a new message is received, and the hardware is set to 1. This bit needs software to be set to 1 and cleared to 0, and software write 0 is invalid.	0
3	FULL1	RW1	Receive FIFO_1 full flag. Set to 1 by hardware when there are 3 messages in FIFO_1. This bit needs software to be set to 1 and cleared to 0, and software write 0 is invalid.	0
2	Reserved	RF	Reserved	0
[1:0]	FMP1[1:0]	RO	Number of received FIFO_1 messages.	0

24.7.6 CANx Interrupt Enable Register (CANx_INTENR) (x=1/2)

Offset address: 0x14

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRIE	Reserved			LECIE	BOFIE	EPVIE	EWGIE	Reserved	FOVIE1	FFIE1	FMPIE1	FOVIE0	FFIE0	FMPIE0	TMEIE

Bit	Name	Access	Description	Reset value
[31:18]	Reserved	RO	Reserved	0
17	SLKIE	RW	Sleep interrupt enable. 1: When entering the sleep state, an interrupt is generated; 0: No interrupt is generated when entering sleep state.	0
16	WKUIE	RW	Wake-up interrupt enable. 1: When the CAN controller is woken up, an interrupt is generated; 0: No interrupt is generated when the CAN controller wakes up.	0
15	ERRIE	RW	Error interrupt enable, CAN error interrupt always enable. 1: When the CAN controller generates an error, an interrupt is generated; 0: No interrupt is generated when the CAN controller generates an error.	0
[14:12]	Reserved	RF	Reserved.	0
11	LECIE	RW	Last error number interrupt enable. 1: When an error is detected, the hardware updates LEC[2:0], updates the ERRI bit to 1, and triggers an error interrupt; 0: When an error is detected, the hardware updates LEC[2:0], does not update the ERRI bit, and does not trigger an error interrupt.	0
10	BOFIE	RW	Offline interrupt enable. 1: When entering the offline state, update the ERRI bit to 1, triggering an error interrupt; 0: When entering the offline state, the ERRI bit will not be updated, and the error interrupt will not be triggered.	0
9	EPVIE	RW	Error passive interrupt enable. 1: When entering the error passive state, update the ERRI bit to 1, triggering an error interrupt; 0: When entering the error passive state, the ERRI bit is not updated and the error interrupt is not triggered.	0

8	EWGIE	RW	Error warning interrupt enable. 1: When the number of errors reaches the warning threshold, update the ERRI bit to 1, triggering an error interrupt; 0: When the number of errors reaches the warning threshold, the ERRI bit will not be updated, and the error interrupt will not be triggered.	0
7	Reserved	RF	Reserved	0
6	FOVIE1	RW	Receive FIFO_1 overflow interrupt enable. 1: When FIFO_1 overflows, trigger FIFO_1 interrupt; 0: When FIFO_1 overflows, do not trigger FIFO_1 interrupt.	0
5	FFIE1	RW	Receive FIFO_1 full interrupt enable. 1: When FIFO_1 is full, trigger FIFO_1 interrupt; 0: When FIFO_1 is full, do not trigger FIFO_1 interrupt.	0
4	FMPIE1	RW	Receive FIFO_1 message registration interrupt enable. 1: When FIFO_1 updates the FMP bit and is not 0, trigger FIFO_1 interrupt; 0: When FIFO_1 updates the FMP bit, and it is not 0, the FIFO_1 interrupt is not triggered.	0
3	FOVIE0	RW	Receive FIFO_0 overflow interrupt enable. 1: When FIFO_0 overflows, trigger FIFO_0 interrupt; 0: When FIFO_0 overflows, not trigger FIFO_0 interrupt.	0
2	FFIE0	RW	Receive FIFO_0 full interrupt enable. 1: When FIFO_0 is full, trigger FIFO_0 interrupt; 0: When FIFO_0 is full, not trigger FIFO_0 interrupt.	0
1	FMPIE0	RW	Receive FIFO_0 message registration interrupt enable. 1: When FIFO_0 updates the FMP bit and is not 0, trigger FIFO_0 interrupt; 0: When FIFO_0 updates the FMP bit, and it is not 0, the FIFO_0 interrupt is not triggered.	0
0	TMEIE	RW	Tx mailbox empty interrupt. 1: When the Tx mailbox is empty, an interrupt is generated; 0: No interrupt is generated when the Tx mailbox is empty.	0

24.7.7 CANx Error Status Register (CANx_ERRSR) (x=1/2)

Offset Address: 0x18

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REC[7:0]								TEC[7:0]							

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

Bit	Name	Access	Description	Reset value
[31:24]	REC[7:0]	RO	Receive error counter. When the CAN receives an error, according to the error condition, the counter is incremented by 1 or 8; after successful reception, the counter is decreased by 1 or set to 120 (the error count value is greater than 127). When the counter value exceeds 127, the CAN enters the error passive state.	0
[23:16]	TEC[7:0]	RO	Transmit error counter. When the CAN sends an error, according to the error condition, the counter is incremented by 1 or 8; after the transmission is successful, the counter is decremented by 1 or set to 120 (the error count value is greater than 127). When the counter value exceeds 127, the CAN enters the error passive state.	0
[15:7]	Reserved	RO	Reserved	0

[6:4]	LEC[2:0]	RW	Last error code.
			When detecting the sending error on the CAN bus, the controller will set according to the error condition, and set 000b when sending and receiving the message correctly.
			000: no error;
			001: Bit stuffing error;
			010: FORM format error;
			011: ACK confirmation error;
			100: recessive bit error;
			101: Dominant bit error;
			110: CRC error;
			111: Software settings.
			Usually when the application software reads the error, the code name is set to 111b, and the code name update can be detected.
				0

3	Reserved	RO	Reserved	0

2	BOFF	RO	Offline status flag. When the CAN controller enters the offline state, the hardware automatically sets it to 1; when it exits the offline state, the hardware automatically clears it to 0.	0
1	EPVF	RO	Error passive flag. When the transceiver error counter reaches the error passive threshold, that is, greater than 127, the hardware is set to 1.	0
0	EWGF	RO	Error warning flag bit. When the sending and receiving error counter reaches the warning threshold, that is, greater than or equal to 96, the hardware is set to 1.	0

24.7.8 CANx Bit Timing Register (CANx_BTIMR) (x=1/2)

Offset address: 0x1C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SILM	LBKM	Reserved				SJW[1:0]		Reserved	TS2[2:0]			TS1[3:0]			

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

Bit	Name	Access	Description	Reset value
31	SILM	RW	Silent mode setting. 1: Enter silent mode; 0: Exit silent mode.	0
30	LBKM	RW	Loopback mode setting. 1: Enter loopback mode; 0: Exit loopback mode.	0
[29:26]	Reserved	RO	Reserved	0
[25:24]	SJW[1:0]	RW	Defines the resync jump width setting value. When implementing resynchronization, the upper limit of the minimum number of time units that can be extended and reduced in the bit, the actual value is (SJW[1:0]+1), and the range can be set to 1 to 4 minimum time units.	01b
23	Reserved	RO	Reserved	0
[22:20]	TS2[2:0]	RW	Time period 2 set value. It defines how many minimum time units are occupied by time period 2, and the actual value is (TS2[2:0]+1).	010b
[19:16]	TS1[3:0]	RW	Time period 1 set value. It defines how many minimum time units are occupied by time period 1, and the actual value	0011b

			is (TS1[3:0]+1).	
[15:10]	Reserved	RO	Reserved.	0
[9:0]	BRP[9:0]	RW	Minimum time unit length setting valueTq = (BRP[9:0]+1) × t_{pclk} *Note: The CAN baud rate is calculated by the following formula:* *CANbps=PCLK1/((TSI[3:0]+1+TS2[2:0]+1+1)*(BPR[9:0]+1))*	0

24.7.9 CANx Time Trigger Control Register (CANx_TTCTLR) (x=1/2)

Offset address: 0x20

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

Bit	Name	Access	Description	Reset value
[31:18]	Reserved	RO	Reserved	0
17	MODE	RW	Time-triggered mode selection. 1: Enhanced mode; 0: Default mode.	0
16	TIMRST	WZ	Internal counter reset control. Write 1 to reset the internal counter, the hardware will automatically clear 0	0
[15:0]	TIMCMV[15:0]	RW	Internal counter count end value	ffffh

24.7.10 CANx Time Trigger Count Value Register (CANx_TTCNT) (x=1/2)

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**
TIMCNT[15:0]															

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserve.	0
[15:0]	TIMCNT[15:0]	RW	time-triggered count value	0

24.7.11 CANx Tx Mailbox Identifier Register (CANx_TXMIRy) (x=0/1, y=0/1/2)

Offset address: 0x180, 0x190, 0x1A0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STID[10:0]/EXID[28:18]											EXID[17:13]				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXID[12:0]													IDE	RTR	TXRQ

Bit	Name	Access	Description	Reset value
[31:21]	STID[10:0]/EXID[28:18]	RW	The upper 11 bits of a standard or extended identifier.	X
[20:3]	EXID[17:0]	RW	The lower 18 bits of the extended identifier.	X
2	IDE	RW	Identifier selection flag. 1: Extended identifier; 0: Standard identifiers.	X
1	RTR	RW	Remote frame selection flag. 1: Remote frame; 0: Data frame.	X
0	TXRQ	RW	Data transmission request flag. When the software is set to 1, the data in the mailbox is requested to be sent. When the mailbox is empty after sending, the hardware is cleared to 0.	0

24.7.12 CANx Tx Mailbox Data Length and Timestamp Register (CANx_TXMDTRy) (x=0/1, y=0/1/2)

Offset address: 0x184, 0x194, 0x1A4

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

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

Bit	Name	Access	Description	Reset value
[31:16]	TIME[15:0]	RW	The 16-bit timer value used to send the message SOF time.	X
[15:9]	Reserved	RO	Reserved	X
8	TGT	RW	Message timestamp transmission selection flag. This bit is valid when TTCM is set to 1 and the message length is 8. 1: Send timestamp, the value is the immediate value of TIME[15:0], replacing the last 2 bytes of the 8-byte message; 0: No timestamp is sent.	X
[7:4]	Reserved	RO	Reserved	X
[3:0]	DLC[3:0]	RW	Data length of data frame or remote frame request data length	X

			The data length can be set from 0 to 8.	

24.7.13 CANx Tx Mailbox Data Low Register (CANx_TXMDLRy) (x=0/1, y=0/1/2)

Offset address: 0x188, 0x198, 0x1A8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA3[7:0]								DATA2[7:0]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DATA1[7:0]								DATA0[7:0]							

Bit	Name	Access	Description	Reset value
[31:24]	DATA3[7:0]	RW	Send the content of data byte 3.	X
[23:16]	DATA2[7:0]	RW	Send the content of data byte 2.	X
[15:8]	DATA1[7:0]	RW	Send the content of data byte 1.	X
[7:0]	DATA0[7:0]	RW	Send the contents of data byte 0.	X

24.7.14 CANx Tx Mailbox Data High Register (CANx_TXMDHRy) (x=0/1, y=0/1/2)

Offset address: 0x18C, 0x19C, 0x1AC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA7[7:0]								DATA6[7:0]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DATA5[7:0]								DATA4[7:0]							

Bit	Name	Access	Description	Reset value
[31:24]	DATA7[7:0]	RW	Send the content of data byte 7.	X
[23:16]	DATA6[7:0]	RW	Send the content of data byte 6.	X
[15:8]	DATA5[7:0]	RW	Send the content of data byte 5.	X
[7:0]	DATA4[7:0]	RW	Send the content of data byte 4.	X

24.7.15 CANx Rx Mailbox Identifier Register (CANx_RXMIRy) (x=0/1, y=0/1)

Offset address: 0x1B0, 0x1C0

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

Bit	Name	Access	Description	Reset value
[31:21]	STID[10:0] /EXID[28:18]	RO	The upper 11 bits of a standard or extended identifier.	X

[20:3]	EXIDL[17:0]	RO	The lower 18 bits of the extended identifier.	X
2	IDE	RO	Identifier selection flag. 1: Select extended identifier; 0: Use standard identifiers.	X
1	RTR	RO	Remote frame selection flag. 1: Remote frame; 0: Data frame.	X
0	Reserved	RO	Reserved	X

24.7.16 CANx Rx Mailbox Data Length and Timestamp Register (CANx_RXMDTRy) (x=0/1, y=0/1)

Offset address: 0x1B4, 0x1C4

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

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

Bit	Name	Access	Description	Reset value
[31:16]	TIME[15:0]	RO	16-bit timer value used to receive the SOF time of the message.	X
[15:8]	FMI[7:0]	RO	The filter number matched by the packet.	X
[7:4]	Reserved	RO	Reserved	X
[3:0]	DLC[3:0]	RO	Received message data length. The data frame length is 0 to 8, and the remote frame is 0.	X

24.7.17 CANx Rx Mailbox Data Low Register (CANx_RXMDLRy) (x=0/1, y=0/1)

Offset address: 0x1B8, 0x1C8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA3[7:0]								DATA2[7:0]							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA1[7:0]								DATA0[7:0]							

Bit	Name	Access	Description	Reset value
[31:24]	DATA3[7:0]	RO	Data byte 3 of the received message.	X
[23:16]	DATA2[7:0]	RO	Data byte 2 of the received message.	X
[15:8]	DATA1[7:0]	RO	Data byte 1 of the received message.	X
[7:0]	DATA0[7:0]	RO	Data byte 0 of the received message.	X

24.7.18 CANx Rx Mailbox Data High Register (CANx_RXMDHRy) (x=0/1, y=0/1)

Offset address: 0x1BC, 0x1CC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA7[7:0]								DATA6[7:0]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DATA5[7:0]								DATA4[7:0]							

Bit	Name	Access	Description	Reset value
[31:24]	DATA7[7:0]	RO	Data byte 7 of the received message.	X
[23:16]	DATA6[7:0]	RO	Data byte 6 of the received message.	X
[15:8]	DATA5[7:0]	RO	Data byte 5 of the received message.	X
[7:0]	DATA4[7:0]	RO	Data byte 4 of the received message.	X

24.7.19 CANx Filter Control Register (CANx_FCTLR) (x=0/1)

Offset address: 0x200

31	30	29	28	27	26	25	24	23	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		CAN2SB[5:0]						Reserved							FINIT

Bit	Name	Access	Description	Reset value
[31:14]	Reserved	R0	Reserved	Reset value
[13:8]	CAN2SB[5:0]	RW	CAN2 filter start bank (value ranges from 1 to 27).	01110b
[7:1]	Reserved	R0	Reserved	Reset value
0	FINIT	RW	Filter initialization mode enable flag. 1: Filter bank is in initialization mode; 0: Filter bank is in normal mode.	1

24.7.20 CANx Filter Mode Register (CANx_FMCFGR) (x=0/1)

Offset address: 0x204

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				FBM27	FBM26	FBM25	FBM24	FBM23	FBM22	FBM21	FBM20	FBM19	FBM18	FBM17	FBM16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FBM15	FBM14	FBM13	FBM12	FBM11	FBM10	FBM9	FBM8	FBM7	FBM6	FBM5	FBM4	FBM3	FBM2	FBM1	FBM0

Bit	Name	Access	Description	Reset value
[31:28]	Reserved	RO	Reserved	0

[27:0]	FBMx	RW	Working mode control of filter bank x. Only can be written when FINT is 1. 0: The register of filter bank x is in mask bit mode; 1: The register of filter bank x is in identifier list mode.	0000000h

24.7.21 CANx Filter Bit Width Register (CANx_FSCFGR) (x=0/1)

Offset address: 0x20C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				FSC27	FSC26	FSC25	FSC24	FSC23	FSC22	FSC21	FSC20	FSC19	FSC18	FSC17	FSC16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FSC15	FSC14	FSC13	FSC12	FSC11	FSC10	FSC9	FSC8	FSC7	FSC6	FSC5	FSC4	FSC3	FSC2	FSC1	FSC0

Bit	Name	Access	Description	Reset value
[31:28]	Reserved	RO	Reserved	0
[27:0]	FSCx	RW	Bit width control bit of filter bank x. Only can be written when FINT is 1. 1: The register of filter bank x is a single 32-bit; 0: The register of filter bank x is 2 16-bit.	0

24.7.22 CANx Filter FIFO Association Register (CANx_FAFIFOR) (x=0/1)

Offset address: 0x214

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				FFA27	FFA26	FFA25	FFA24	FFA23	FFA22	FFA21	FFA20	FFA19	FFA18	FFA17	FFA16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FFA15	FFA14	FFA13	FFA12	FFA11	FFA10	FFA9	FFA8	FFA7	FFA6	FFA5	FFA4	FFA3	FFA2	FFA1	FFA0

Bit	Name	Access	Description	Reset value
[31:28]	Reserved	RO	Reserved	0
[27:0]	FFAx	RW	Associated FIFO control of filter bank x. Only can be written when FINT is 1. 1: Filter bank x is associated to FIFO_1; 0: Filter bank x is associated to FIFO_0.	0

24.7.23 CANx Filter Activation Register (CANx_FWR) (x=0/1)

Offset address: 0x21C

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

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

Bit	Name	Access	Description	Reset value
[31:28]	Reserved	RO	Reserved	0
[27:0]	FACTx	RW	Filter bank x activation control bit. 1: Filter bank x active; 0: Filter bank x disabled.	0

24.7.24 CANx Filter Register for Filter Bank (CANx_FiRy) (x=1/2, i=0-27, y=1/2)

Offset address: 0x240 to 0x31C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FB31	FB30	FB29	FB28	FB27	FB26	FB25	FB24	FB23	FB22	FB21	FB20	FB19	FB18	FB17	FB16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FB15	FB14	FB13	FB12	FB11	FB10	FB9	FB8	FB7	FB6	FB5	FB4	FB3	FB2	FB1	FB0

Bit	Name	Access	Description	Reset value
[31:0]	FB	RW	Register flag bit in the filter bank. Only can be written when FINT is 1. In identifier mode: 1: Expected level of the corresponding bit is a recessive bit; 0: Expected level of the corresponding bit is a dominant bit. In mask bit mode: 1: Must be consistent with the corresponding identifier register bit; 0: No need to be consistent with the corresponding identifier register bit.	0

Chapter 25 Digital Video Port (DVP)

The module descriptions in this chapter are applicable to some products of the CH32F2x, CH32V2x and CH32V3x microcontroller series.

Digital Video Port (DVP), which supports the use of DVP interface timing to obtain image data streams, supports image data organized in original line and frame formats, such as YUV, RGB, etc., and also supports compressed image data such as JPEG format, which can receive high-speed parallel data stream output by external 8-bit, 10-bit and 12-bit camera modules.

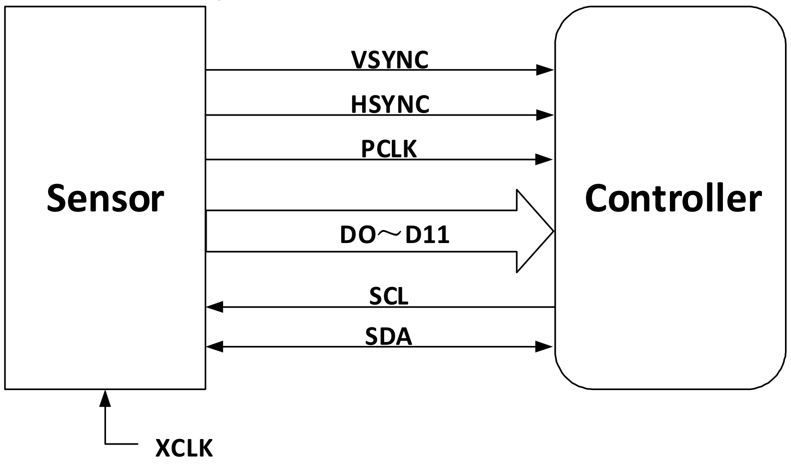
25.1 Main features

- Configurable 8/10/12-bit data width mode
- Support YUV, RGB format data
- Support JPEG compression format data
- Built-in FIFO, support DMA transfer
- Support dual-buffered reception
- Support crop function
- Support continuous mode and snapshot mode

25.2 Functional description

25.2.1 Connected to the sensor

Figure 25-1 DVP interface connection



- PLCK (Pixel clk): pixel clock, each clock corresponds to single pixel data (uncompressed data). The PCLK clock output by the external DVP interface sensor supports a maximum of 96MHz.
- HSYNC (horizontal synchronization): horizontal synchronization signal.
- VSYNC (vertical synchronization): Frame synchronization signal.
- DATA: pixel data or compressed data, the bit width supports 8/10/12 bits.
- XCLK: The reference clock of the Sensor, which can be provided by the microcontroller or externally, generally using a crystal oscillator.
- I2C interface: The register used to configure the sensor, which can simulate the I2C timing communication through the controlled common GPIO port, or use the hardware I2C interface to operate.

Table 25-1 DVP pins

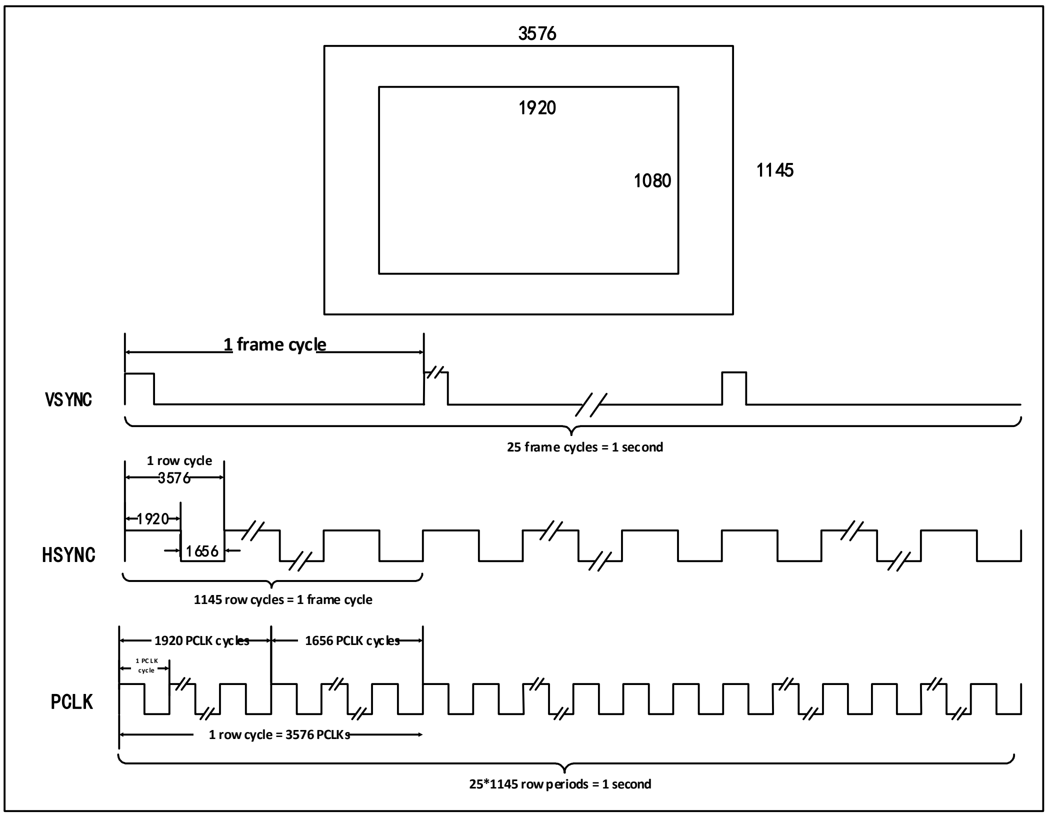
Name	Signal Type Description
PLCK	Pixel clock input
DATA[11:0]	Pixel data input
VSYNC	Frame sync signal input
HSYNC	Line sync signal input

25.2.2 Working timing

There is a certain relationship between the digital signal data stream output by the commonly used DVP interface sensor module and the image size. The following is an example of image data.

As shown in Figure 25-2, the inside of the sensor is a complete image size of 3576*1145. After internal scaling process, the final image data size output from the interface is 1920*1080, and the image refresh rate is 25.

Figure 25-2 Timing description



PCLK is the transmission time of a pixel, so HSYNC is 3576 times more than PCLK. Among the 3576 pixels, only 1920 pixels are valid, and the sensor does not transmit data during the remaining 1656 pixels. VSYNC is a frame synchronization signal, so the VSYNC time is 3576*1145 times of PCLK. Similarly, the sensor is transmitting data only during the 1920*1080 effective pixel time. If the sensor is transmitting JPEG compressed data, the HSYNC signal may not be needed.

The relationship between DVP interface signal and image data is mainly based on the description of the selected sensor data sheet.

25.2.3 RGB/YUV/JPEG compressed data format description

- RGB
  3 primary colors: red, green, blue
- YUV
  The luminance signal Y, the chrominance signal U and V, Pb and Pr, Cb and Cr.
- JPEG (Joint Photographic Experts Group)

There is lossy compression, but the lost part is the part that is not easily perceptible by human vision, and the human eye is insensitive to the high-frequency information part of the computer color. Remove the visual redundant information (spatial redundancy) and remove the redundant information of the data itself (structural redundancy).

25.3 Application of digital video port

25.3.1 Digital video port configuration description

1） In the data reception of DVP, each frame of data is stored alternately by BUF0 and BUF1, starting from BUF0. For RGB and YUV data streams, the hardware resets and selects BUF0 every time the frame signal changes from an inactive level to an active level. When a line of data is full, BUF1 will be switched to realize alternate storage; for JPEG compressed data, the hardware will set the switching threshold of BUF0 and BUF1 according to the set DMA receive length.

2） When the data bus width is 10-bit or 12-bit, each time a data is received, the system will automatically unsigned-extend the data to 16-bit and then store it.

3） The R16_DVP_ROW_NUM and R16_DVP_COL_NUM registers must match the image size actually output by the sensor.

4） In video stream RGB mode, R16_DVP_COL_NUM represents the number of valid PCLK cycles of 1-line data, and R16_DVP_ROW_NUM represents the number of lines contained in a frame of image data; in image JPEG mode, R16_DVP_COL_NUM is used to configure the DMA length, and in this mode, the R16_DVP_ROW_NUM register cannot be used effectively.

25.3.2 Digital video port application note

When using the digital image interface to receive image data, the DVP control registers must be correctly configured to match the mode of the image sensor. The specific operation steps are as follows:

1） Clear RB_DVP_ALL_CLR and RB_DVP_RCV_CLR using the R8_DVP_CR1 register.

2） Configure the image mode, data bit width, PCLK polarity, HSYNC polarity and VSYNC polarity through the R8_DVP_CR0 register to match the output of the sensor.

3） According to the effective image pixels output by the configured image sensor, configure the R16_DVP_ROW_NUM and R16_DVP_COL_NUM registers to match the output of the sensor. In the image JPEG mode, only configure the R16_DVP_COL_NUM register.

4） Configure the DMA receive address through the R32_DVP_DMA_BUF0/1 register.

5） If the snapshot mode is used, the RB_DVP_CM field needs to be configured through the R8_DVP_CR1 register to enable the snapshot mode.

6） If the cropping mode is used, the RB_DVP_CROP and RB_DVP_FCRC fields need to be configured through the R8_DVP_CR1 register to enable the cropping function and control the frame capture rate. At the same time, configure the R16_DVP_HOFFCNT, R16_DVP_VST, R16_DVP_CAPCNT and

R16_DVP_VLINE to set the size of the cropped image.

7） According to the requirements, enable the corresponding interrupt through the R8_DVP_IER register, configure the interrupt priority through the interrupt controller NVIC or PFIC, and enable the DVP interrupt.

8） The DMA is enabled through the R8_DVP_CR1 register, and the DVP interface is enabled through the R8_DVP_CR0 register.

9） Wait for the generation of the relevant receiving interrupt, and process the received data in time.

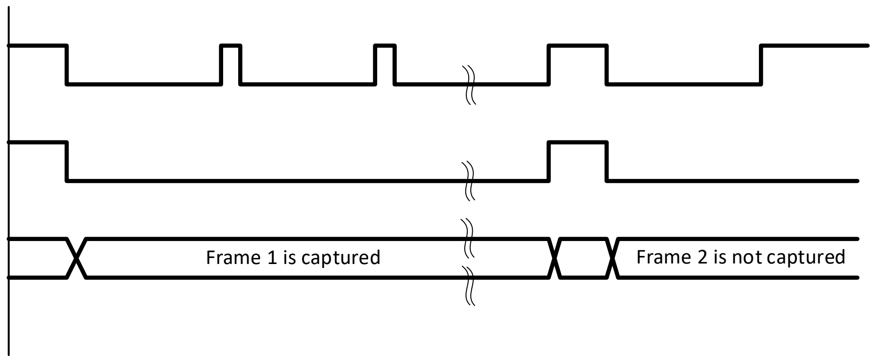
25.3.3 Capture mode description

The DVP interface supports 2 capture modes: snapshot (single frame) mode and continuous mode.

25.3.3.1 Snapshot mode

In this mode, only a single frame is captured (RB_DVP_CM is set to 1 in the R8_DVP_CR1 register). When the DVP interface is enabled, wait for the system to detect the start of the frame, and then start to sample the image data. After receiving a complete frame of data, the DVP interface will be turned off. (the RB_DVP_ENABLE field of the R8_DVP_CR0 register is cleared to 0). Figure 25-3 shows the single frame capture waveform in snapshot mode.

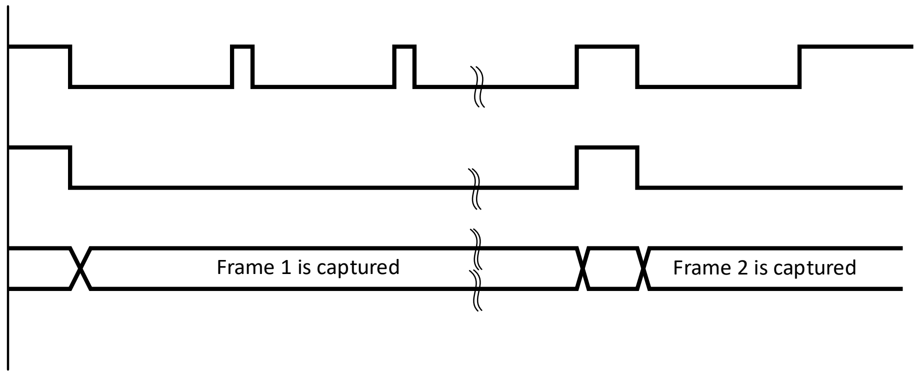
Figure 25-3 Single frame capture waveform in Snapshot mode



25.3.3.2 Continuous mode

In this mode (RB_DVP_CM is cleared to 0 in the R8_DVP_CR1 register), after the DVP interface is enabled, each frame of data will be continuously sampled before the RB_DVP_ENABLE field of the R8_DVP_CR0 register is cleared to 0. The frame capture waveform in continuous mode is shown in Figure 25-4.

Figure 25-4 Frame capture waveform in continuous mode



25.3.4 Cropping function description

DVP can use the crop function to cut a rectangular window from the received image. The starting coordinates of the rectangular window (X coordinate of the upper left corner of the rectangle R16_DVP_HOFFCNT, Y coordinate R16_DVP_VST) and window size (R16_DVP_CAPCNT represents the horizontal size, R16_DVP_VLINE represents the vertical size) are configurable. The coordinates and size of the cropping window are shown in Figure 25-5. The cropped window data capture waveform is shown in Figure 25-6.

Figure 25-5 Crop window coordinates and size

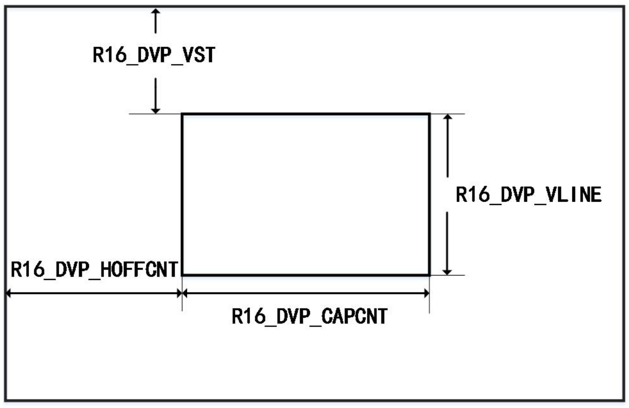
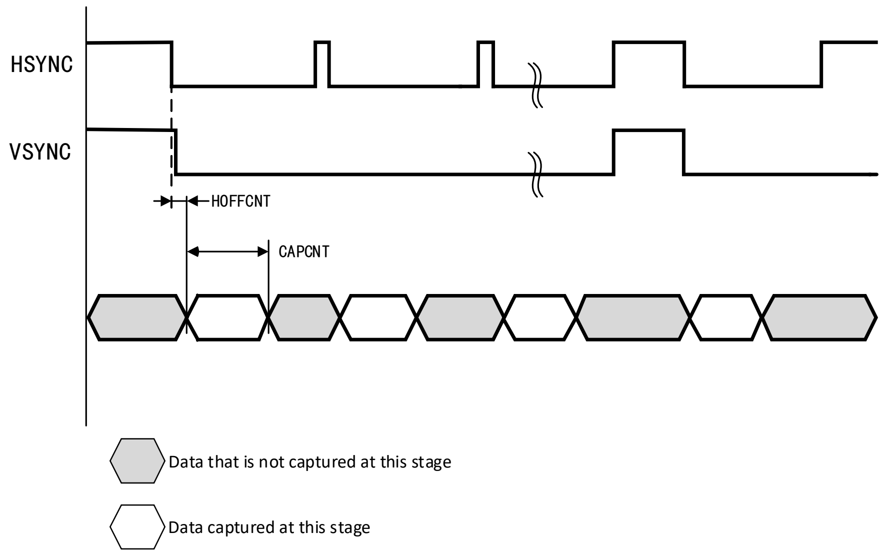


Figure 25-6 Crop window data capture waveform


HSYNC
VSYNC
HOFFCNT
CAPCNT
Data that is not captured at this stage
Data captured at this stage

25.4 Register description

Table 25-2 DVP registers

Name	Access address	Description	Reset value
R8_DVP_CR0	0x50050000	DVP control register0	0x00
R8_DVP_CR1	0x50050001	DVP control register1	0x06
R8_DVP_IER	0x50050002	DVP interrupt enable register	0x00
R16_DVP_ROW_NUM	0x50050004	DVP row number configuration register	0x0000
R16_DVP_COL_NUM	0x50050006	DVP column number configuration register	0x0000

R32_DVP_DMA_BUF0	0x50050008	DVP DMA address0 register	0x00000000
R32_DVP_DMA_BUF1	0x5005000C	DVP DMA address1 register	0x00000000
R8_DVP_IFR	0x50050010	DVP interrupt flag register	0x00
R8_DVP_STATUS	0x50050011	DVP receive FIFO status register	0x00
R16_DVP_ROW_CNT	0x50050014	DVP row count register	0x0000
R16_DVP_HOFFCNT	0x50050018	Horizontal displacement register for the start of the window	0x0000
R16_DVP_VST	0x5005001A	Window start line number register	0x0000
R16_DVP_CAPCNT	0x5005001C	DVP capture count register	0x0000
R16_DVP_VLINE	0x5005001E	DVP vertical line count register	0x0000
R32_DVP_DR	0x50050020	DVP data register	0x00000000

25.4.1 DVP Control Register (R8_DVP_CR0)

Offset address: 0x00

Bit	Name	Access	Description	Reset value
6	RB_DVP_JPEG	RW	JPEG mode enable. 1: JPEG compression format; 0: Raw data format.	0

[5:4]	RB_DVP_MSK_DAT_MOD	RW	DVP data bit width configuration.
			00: 8-bit mode;
			01: 10-bit mode;
			1x: 12-bit mode.
				0

3	RB_DVP_P_POLAR	RW	PCLK polarity configuration. 1: Sample data on the falling edge of PCLK; 0: Sample data on the rising edge of PCLK.	0
2	RB_DVP_H_POLAR	RW	HSYNC polarity configuration. 1: HSYNC low level data is valid; 0: HSYNC high level data is valid.	0
1	RB_DVP_V_POLAR	RW	VSYNC polarity configuration. 1: VSYNC high level data is valid; 0: VSYNC low level data is valid.	0
0	RB_DVP_ENABLE	RW	DVP enable. 1: Enable DVP; 0: Disable DVP.	0

25.4.2 DVP Control Register (R8_DVP_CR1)

Offset address: 0x01

Bit	Name	Access	Description	Reset value
[7:6]	RB_DVP_FCRC	RW	DVP frame capture rate control. 00: Capture all frames; 01: Capture every other frame; 10: Capture every third frame. 11: Reserved	0

5	RB_DVP_CROP	RW	Crop function control. 0: Capture full image; 1: Capture only the data in the window specified by the clipping register.	0
4	RB_DVP_CM	RW	Capture mode. 0: Continuous mode; 1: Snapshot mode.	0
3	RB_DVP_BUF_TOG	RWT	Buffer address flag. Hardware control toggle, the software sets 1 to flip this bit, and writing 0 is invalid. 1: Data is stored at receive address 1; 0: Data is stored at receive address 0.	0
2	RB_DVP_RCV_CLR	RW	Receive logic reset control. 1: Reset the receiving logic circuit; 0: Cancel the reset operation.	1
1	RB_DVP_ALL_CLR	RW	Flag and FIFO clear control, write 1 or write 0 by software: 1: Reset flag and FIFO; 0: Cancel the reset operation.	1
0	RB_DVP_DMA_ENABLE	RW	DMA enable: 1: Enable DMA; 0: Disable DMA.	0

25.4.3 DVP Interrupt Enable Register (R8_DVP_IER)

Offset address: 0x02

Bit	Name	Access	Description	Reset value
[7:5]	Reserved	RO	Reserved	-
4	RB_DVP_IE_STP_FRM	RW	Frame stop interrupt enable. (Interrupt is generated when VSYNC changes from an active level to an inactive level.) 1: Enable frame stop interrupt; 0: Disable frame stop interrupt.	0
3	RB_DVP_IE_FIFO_OV	RW	Receive FIFO overflow interrupt enable. 1: Enable FIFO overflow interrupt; 0: Disable FIFO overflow interrupt.	0
2	RB_DVP_IE_FRM_DONE	RW	Frame reception done interrupt enable. (Interrupt is generated when the counter reaches the RAW/COL_NUM configuration value, indicating that the last data has been written to RAM) 1: Enable frame reception done interrupt; 0: Disable frame reception done interrupt.	0
1	RB_DVP_IE_ROW_DONE	RW	Row done interrupt enable. (Interrupt is generated when the counter reaches the COL_NUM configuration value)	0

			1: Enable row done interrupt; 0: Disable row done interrupt.	
0	RB_DVP_IE_STR_FRM	RW	New frame start interrupt enable. (Interrupt is generated when VSYNC changes from an inactive level to an active level, indicating that a new frame starts and data is coming.) 1: Enable new frame start interrupt; 0: Disable new frame start interrupt.	0

25.4.4 DVP Row Number Configuration Register (R16_DVP_ROW_NUM)

Offset address: 0x04

Bit	Name	Access	Description	Reset value
[15:0]	RB_DVP_ROW_NUM	RW	In RGB mode, it indicates the number of rows contained in a frame of image data. In JPEG mode, this register has no practical meaning.	0

25.4.5 DVP Column Number Configuration Register (16_DVP_COL_NUM)

Offset address: 0x06

Bit	Name	Access	Description	Reset value
[15:0]	RB_DVP_COL_NUM	RW	In RGB mode, indicates the number of PCLK cycles contained within a line of data. In JPEG mode, it is used to configure the DMA receive length.	0

25.4.6 DVP DMA Address 0 Register (R32_DVP_DMA_BUF0)

Offset address: 0x08

Bit	Name	Access	Description	Reset value
[16:0]	RB_DVP_DMA_BUF0	RW	DMA receive address 0.	0

25.4.7 DVP DMA Address 1 Register (R32_DVP_DMA_BUF1)

Offset address: 0x0C

Bit	Name	Access	Description	Reset value
[16:0]	RB_DVP_DMA_BUF1	RW	DMA receive address 1.	0

25.4.8 DVP Interrupt Flag Register (R8_DVP_IFR)

Offset Address: 0x10

Bit	Name	Access	Description	Reset value
[7:5]	Reserved	RO	Reserved.	-

4	RB_DVP_IF_STP_FRM	RW	Frame stop interrupt flag, active high, clear by writing 0.	0
3	RB_DVP_IF_FIFO_OV	RW	Receive FIFO overflow interrupt flag, active high, clear by writing 0.	0
2	RB_DVP_IF_FRM_DONE	RW	Frame receive completion interrupt flag, valid high, clear by writing 0.	0
1	RB_DVP_IF_ROW_DONE	RW	Row done interrupt flag, active high, clear by writing 0.	0
0	RB_DVP_IF_STR_FRM	RW	Frame start interrupt flag, active high, clear by writing 0.	0

25.4.9 DVP Receive FIFO Status Register (R8_DVP_STATUS)

Offset address: 0x11

Bit	Name	Access	Description	Reset value
7	Reserved	RO	Reserved.	0
[6:4]	RB_DVP_FIFO_CNT	RO	FIFO counter.	0
3	Reserved	RO	Reserved.	0
2	RB_DVP_FIFO_OV	RO	FIFO overflow status. 1: FIFO overflow; 0: FIFO not overflow.	0
1	RB_DVP_FIFO_FULL	RO	FIFO full status. 1: The buffer is full; 0: The FIFO is not full.	0
0	RB_DVP_FIFO_RDY	RO	FIFO ready status. 1: There is data in the FIFO; 0: No data in the FIFO.	0

25.4.10 DVP Row Count Register (R16_DVP_ROW_CNT)

Offset address: 0x14

Bit	Name	Access	Description	Reset value
[15:0]	RB_DVP_ROW_CNT	RO	In actual reception, the number of rows contained in a frame of image data, this register is updated at the end of the frame. In JPEG format, the value of this register has no meaning.	0

25.4.11 DVP Horizontal Shift Register for the Start of the Window (R16_DVP_HOFFCNT)

Offset address: 0x18

Bit	Name	Access	Description	Reset value
[15:0]	RB_DVP_HOFFCNT	RW	Within the window line, each row needs to be blanked for the number of PCLK cycles before	0

			capturing data. The lower 14 bits are valid.	

25.4.12 DVP Window Start Line Number Register (R16_DVP_VST)

Offset address: 0x1A

Bit	Name	Access	Description	Reset value
[15:0]	RB_DVP_VST	RW	The number of lines the image to start capturing, data before this line is not captured. The lower 13 bits are valid.	0

25.4.13 DVP Capture Count Register (R16_DVP_CAPCNT)

Offset address: 0x1C

Bit	Name	Access	Description	Reset value
[15:0]	RB_DVP_CAPCNT	RW	The number of PCLK cycles to capture within the clipping window. The lower 14 bits are valid.	0

25.4.14 DVP Vertical Line Count Register (R16_DVP_VLINE)

Offset address: 0x1E

Bit	Name	Access	Description	Reset value
[15:0]	RB_DVP_VLINE	RW	The number of lines to capture within the crop window. The lower 14 bits are valid.	0

25.4.15 DVP Data Register (R32_DVP_DR)

Offset address: 0x20

Bit	Name	Access	Description	Reset value
[31:0]	RB_DVP_DR	RO	Each time the DVP interface receives 4 bytes of data, a DMA request is triggered, and the 4-byte deep FIFO can leave sufficient time for DMA transfers. It can effectively prevent the occurrence of DMA overflow.	0

Chapter 26 Flexible Static Memory Controller (FSMC)

The module descriptions in this chapter are applicable to some products of the CH32F2x, CH32V2x and CH32V3x microcontroller series.

The flexible static memory controller (FSMC) supports a variety of static memory types and rich storage operation methods, and can expand different types of large-capacity static memory according to the needs of system applications.

26.1 Main features

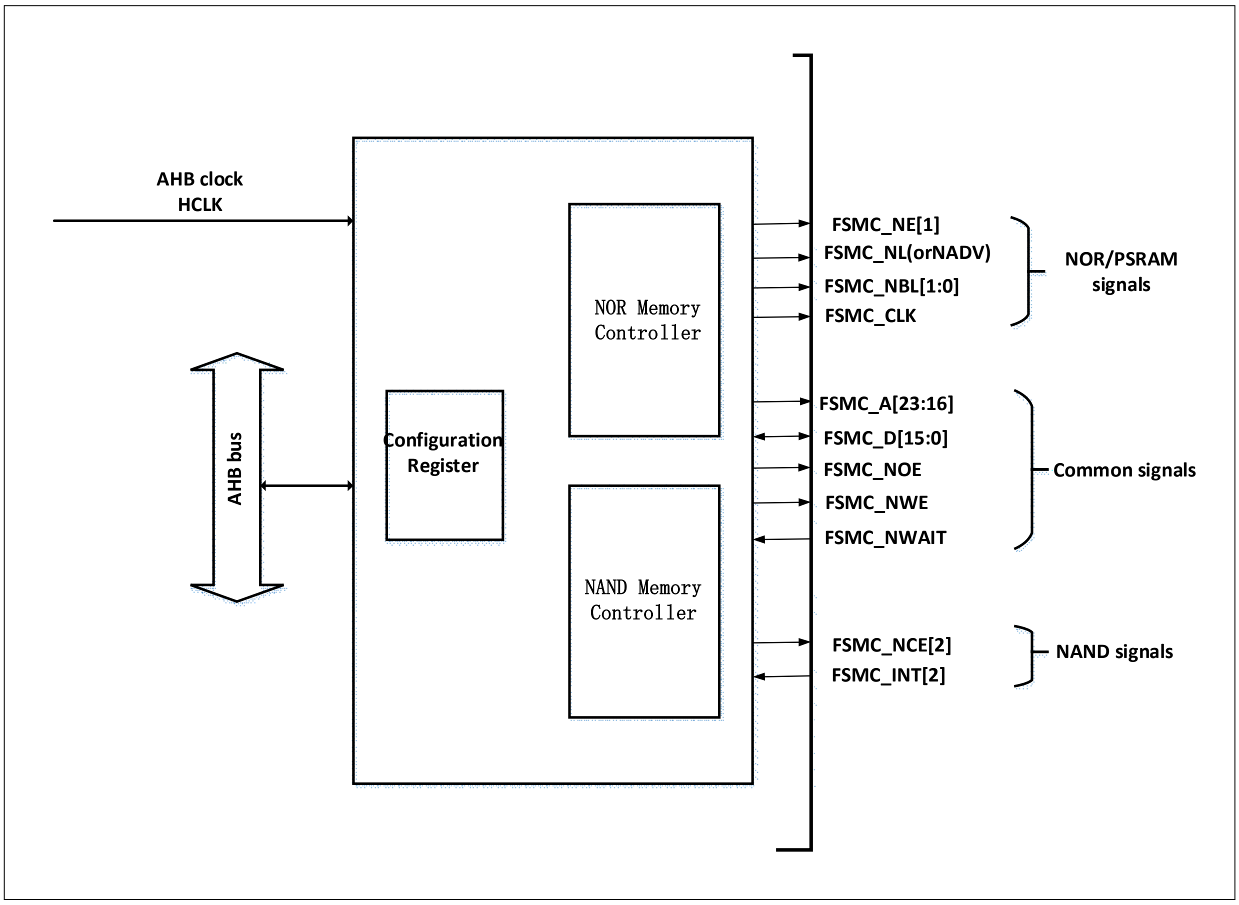
- Support to operate SRAM, ROM, NOR FLASH and PSRAM
- Support NAND FLASH operation, and built-in hardware ECC, can detect up to 8KByte data
- Support for synchronous device operation, such as PSRAM
- Support 8bit or 16bit data bus width
- Timing signals can be programmed by software

26.2 Functional description

26.2.1 Module structure

FSMC mainly includes AHB bus interface, NOR memory controller, NAND memory controller and external device interface.

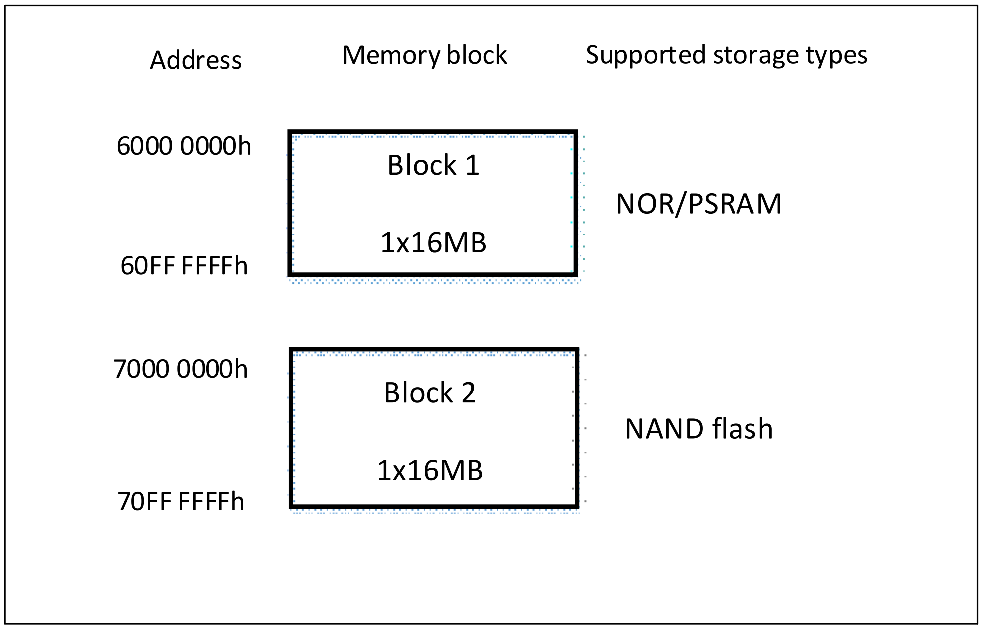
Figure 26-1 FSMC block diagram



26.2.2 External device address mapping

The FSMC divides the memory block into 2 memory blocks with a fixed size of 16MByte according to the address line, as shown in Figure 26-2.

Figure 26-2 FSMC memory block



26.2.2.1 NOR/PSRAM address mapping

Table 26-1 External memory address

Data width	Address lines connected to memory	Maximum access memory space (bit)
8bit	HADDR[23:0] is correspondingly connected to FSMC_A[23:0]	16MBx8=128Mbit
16bit	HADDR[23:1] is correspondingly connected to FSMC_A[22:0], HADDR[0] is not connected	16MB/2x16=128Mbit

Note: Those with 100 pins support FSMC, and only support address and data line multiplexing mode.

NOR FLASH and PSRAM support unaligned access. For asynchronous mode, each operation requires an accurate address; for synchronous mode, only single address signal is issued, and then batches of data are sequentially processed through CLK. For NOR FLASH that supports unaligned batch access, the unaligned access mode of the memory can be set to the same mode as AHB. If it cannot be set, the unaligned access mode is disabled, and the unaligned access request is divided into 2 consecutive access operations.

26.2.2.2 NAND address mapping

NAND FLASH image and timing registers, see Table 26-2.

Table 26-2 Memory map and timing registers

Start address	End address	FSMC memory block	Storage	Timing register

0x78000000	0x78FFFFFF	Piece 2-NAND FLASH	Attribute	FSMC_PATT2(0x6C)
0x70000000	0x70FFFFFF		General	FSMC_PMEM2(0x68)

The general and attribute space can be divided into address area (second 128KB area), command area (second 64KB area) and data area (first 64KB area) in the lower 256KB, see Table 26-3.

The software accesses the specific process of NAND FLASH by operating these 3 areas:

- Send read and write operation commands to NAND FLASH: the software can operate any address in the command area to send commands;
- Send the address to be operated to the NAND FLASH: the software can operate any address in the address area to send commands;
- Read and write data from NAND FLASH: software can operate any address in the data area to write or read data;

Table 26-3 NAND memory block selection

Area	HADDR[17:16]	Range
Address area	1X	0x020000 to 0x03FFFF
Command area	01	0x010000 to 0x01FFFF
Data area	00	0x000000 to 0x00FFFF

26.2.3 NOR/PSRAM controller

FSMC supports 8bit, 16bit and 32bit asynchronous operation SRAM and ROM, supports asynchronous mode and burst mode operation PSRAM, supports asynchronous mode and burst mode operation NOR FLASH. The output signals of all controllers change on the rising edge of the internal clock HCLK. For synchronous write mode (PSRAM), the output data changes on the falling edge of the memory clock (CLK). For details, refer to the synchronous transfer and asynchronous transfer timing diagrams. The read and write parameters of the memory can be configured by software, see Table 26-4.

Table 26-4 Software controllable NOR/PSRAM read and write parameters

Parameters	Read and write	Parameter value range
Address setup time	Asynchronous	1<< T <<16 (AHB HCLK)
Address hold time	Asynchronous	1<< T <<16 (AHB HCLK)
Data setup time	Asynchronous	2<< T <<256 (AHB HCLK)
Bus recovery time	Asynchronous or synchronous read	1<< T <<16 (AHB HCLK)
Clock divider	Synchronous	2<< T <<16 (AHB HCLK)
Data generation time	Synchronous	2<< T <<17 (Memory CLK)

26.2.3.1 External memory multiplexing interface signal

For NOR FLASH and PSRAM interfaces, see Table 26-5 and Table 26-6.

Table 26-5 Multiplexed NOR FLASH interface

FSMC Pins	Direction	Description
CLK	Output	Clock line (for synchronous burst mode only)

A[23:16]	Output	Address line
AD[15:0]	Input / Output	16bit address/data line (multiplexed)
NE[1]	Output	Chip select line
NOE	Output	Output enable
NWE	Output	Write enable
NL(NADV)	Output	Latch Enable
NWAIT	Input	Wait for signal line (only for NOR FLASH)
NBL[1]	Output	High byte enable (NUB)
NBL[0]	Output	Low byte enable (NLB)

Note: The signal with the prefix "N" means that it is active low.

Table 26-6 Multiplexed PSRAM interface

FSMC Pins	Direction	Description
CLK	Output	Clock line (for synchronous burst mode only)
A[23:16]	Output	Address line
AD[15:0]	Input / Output	16bit address/data line (multiplexed)
NE[1]	Output	Chip select line
NOE	Output	Output enable
NWE	Output	write enable
NL(NADV)	Output	Latch enable
NWAIT	Input	Wait for signal line (only for NOR FLASH)

26.2.3.2 Supported memory and operation methods

See Table 26-7 for supported memories and operation methods.

Table 26-7 Supported memories and operation methods.

Memory	Mode	AHB data width	Memory width	Description
NOR FLASH	Asynchronous read	8	16	
	Asynchronous read	16	16	
	Asynchronous write	16	16	
	Asynchronous read	32	16	Divided into 2 FSMC access
	Asynchronous write	32	16	Divided into 2 FSMC access
	Synchronous read	16	16	
	Synchronous read	32	16	Divided into 2 FSMC access
PSRAM	Asynchronous read	8	16	
	Asynchronous write	8	16	Use byte letter NBL[1:0]
	Asynchronous read	16	16	
	Asynchronous write	16	16	
	Asynchronous read	32	16	Divided into 2 FSMC access
	Asynchronous write	32	16	Divided into 2 FSMC access
	Synchronous read	16	16	
	Synchronous read	32	16	Divided into 2 FSMC access
	Synchronous write	8	16	Use byte letter NBL[1:0]
	Synchronous write	16/32	16	

SRAM and ROM	Asynchronous read	8/16/32	8/16	Use byte letter NBL[1:0]
	Asynchronous write	8/16/32	8/16	Use byte letter NBL[1:0]

26.2.3.3 NOR/PSRAM timing diagram of asynchronous transfer address/data multiplexing

Note the following for asynchronous static memory (NOR FLASH and PSRAM) operations:

- When extended mode is enabled, read and write operations to memory using modes A, B, C, and D can be mixed.

Figure 26-3 Mode 1 read operation (multiplexed)

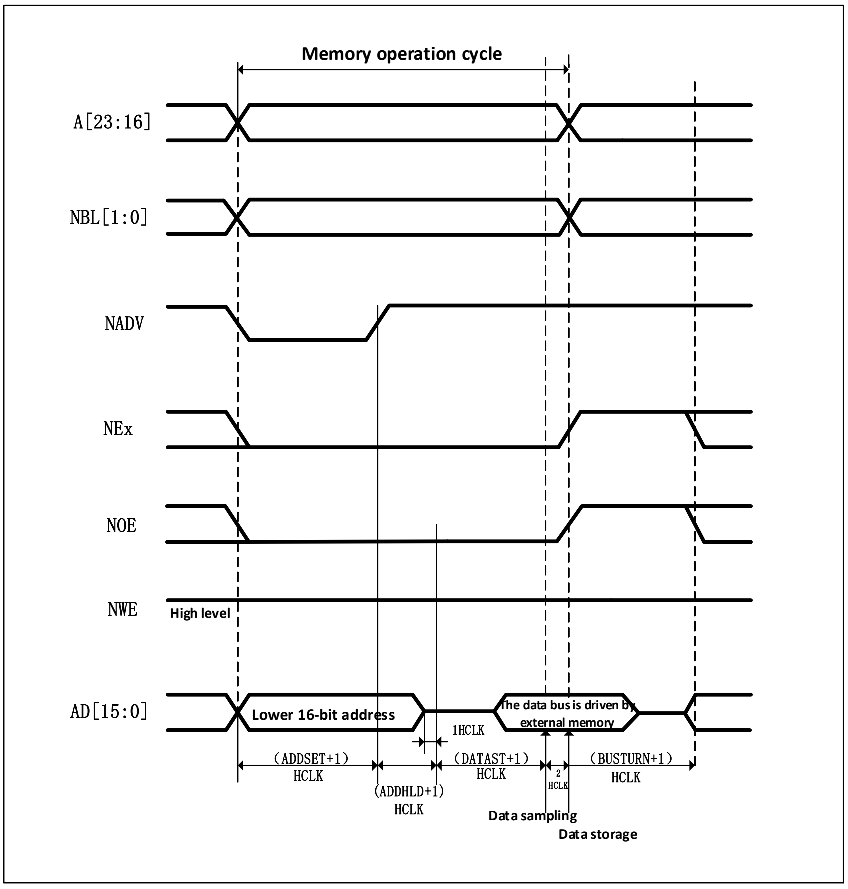


Figure 26-4 Mode 1 write operation (multiplexed)

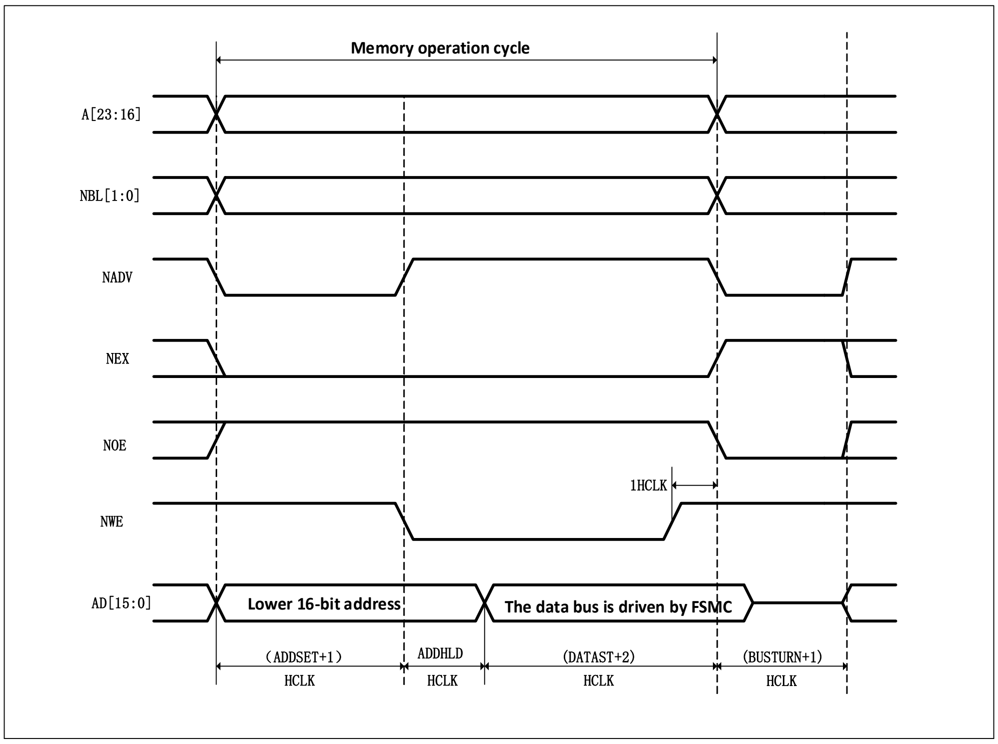


Table 26-8 Mode 1 FSMC_BCR1 bit field

Bitx	Name	Configuration Value
[31:20]	Reserved	0x000
19	CBURSTRW	0x0
[18:16]	Reserved	0x0
15	ASYNCWAIT	It is 1 if the memory supports this function, otherwise it is 0.
14	EXTMOD	0x0
13	WAITEN	0x0
12	WREN	Set as needed
11	WAITCFG	No effect
10	WRAPMOD	0x0
9	WAITPOL	When bit15 is 1, it is meaningful
8	BURSTEN	0x0
7	Reserved	0x1
6	FACCEN	No effect
[5:4]	MWID	Set as needed
[3:2]	MTYP	Set as needed, excluding 0x2 (NOR FLASH)
1	MUXEN	0x1
0	MBKEN	0x1

Table 26-9 Mode 1 FSMC_BTR1 bit field

Bitx	Name	Configuration Value
[31:30]	Reserved	0x0
[29:28]	ACCMOD	No effect

[27:24]	DATLAT	No effect
[23:20]	CLKDIV	No effect
[19:16]	BUSTURN	0x0
[15:8]	DATAST	Data setup time. The write operation is (DATAST+1 HCLK), and the read operation is (DATAST+3 HCLK). The minimum value of this bit is 1.
[7:4]	ADDHLD	Address hold time. (ADDHLD+1 HCLK)
[3:0]	ADDSET	Address setup time. (ADDSET+1 HCLK).

Figure 26-5 Mode A read operation (multiplexed)

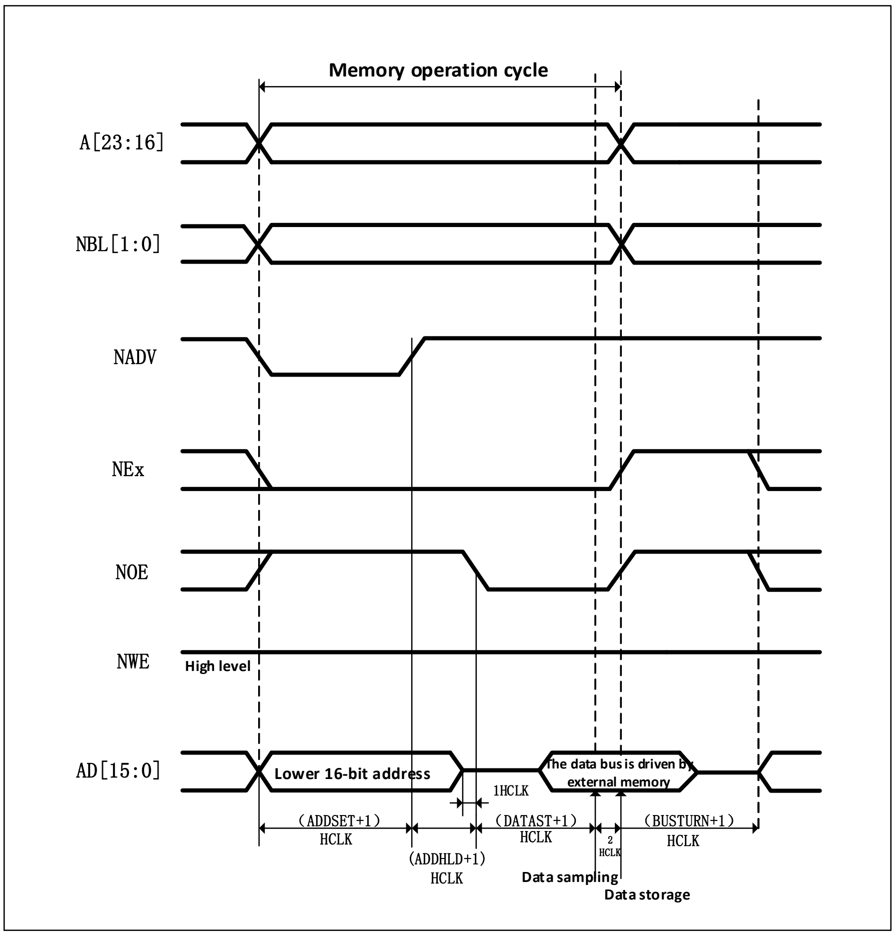


Figure 26-6 Mode A write operation (multiplexed)

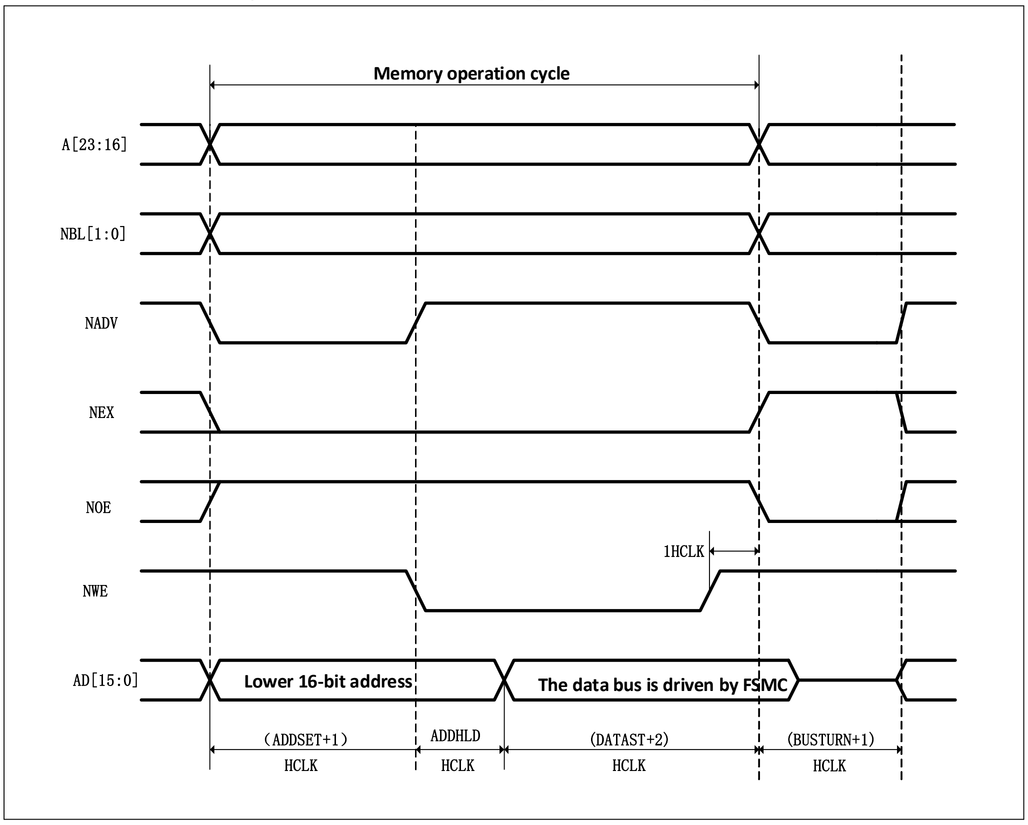


Table 26-10 Mode A FSMC_BCR1 bit field

Bitx	Name	Configuration Value
[31:20]	Reserved	0x000
19	CBURSTRW	0x0
[18:16]	Reserved	0x0
15	ASYNCWAIT	It is 1 if the memory supports this function, otherwise it is 0.
14	EXTMOD	0x1
13	WAITEN	0x0
12	WREN	Set as needed
11	WAITCFG	No effect
10	WRAPMOD	0x0
9	WAITPOL	When bit15 is 1, it is meaningful
8	BURSTEN	0x0
7	Reserved	0x1
6	FACCEN	No effect
[5:4]	MWID	Set as needed
[3:2]	MTYP	Set as needed, excluding 0x2 (NOR FLASH)
1	MUXEN	0x1
0	MBKEN	0x1

Table 26-11 Mode A FSMC_BTR1 bit field

Bitx	Name	Configuration Value
[31:30]	Reserved	0x0
[29:28]	ACCMOD	0x0
[27:24]	DATLAT	No effect
[23:20]	CLKDIV	No effect
[19:16]	BUSTYRN	0x0
[15:8]	DATAST	Data setup time. The write operation is (DATAST+1 HCLK), and the read operation is (DATAST+3 HCLK). The minimum value of this bit is 1.
[7:4]	ADDHLD	No effect
[3:0]	ADDSET	Address setup time. The read operation is (ADDSET+1 HCLK).

Table 26-12 Mode A FSMC_BWTR1 bit field

Bitx	Name	Configuration Value
[31:30]	Reserved	0x0
[29:28]	ACCMOD	0x0
[27:24]	DATLAT	No effect
[23:20]	CLKDIV	No effect
[19:16]	BUSTURN	0x0
[15:8]	DATAST	Data setup time. The write operation is (DATAST+1 HCLK), and the read operation is (DATAST+3 HCLK). The minimum value of this bit is 1.
[7:4]	ADDHLD	Address hold time. (ADDHLD+1 HCLK)
[3:0]	ADDSET	Address setup time. The write operation is (ADDSET+1 HCLK).

Figure 26-7 Mode B read operation (multiplexed)

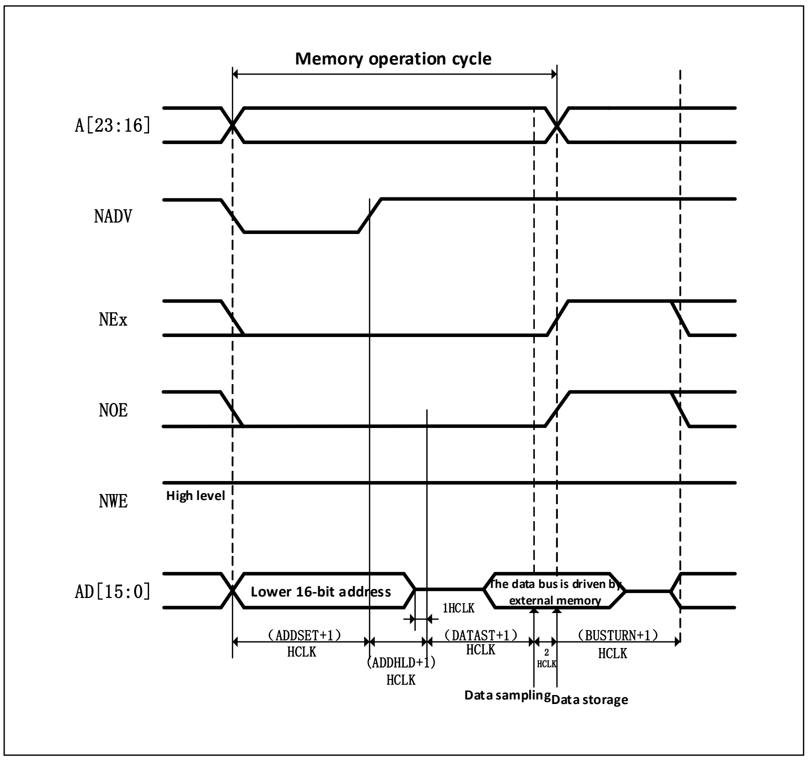


Figure 26-8 Mode B write operation (multiplexed)

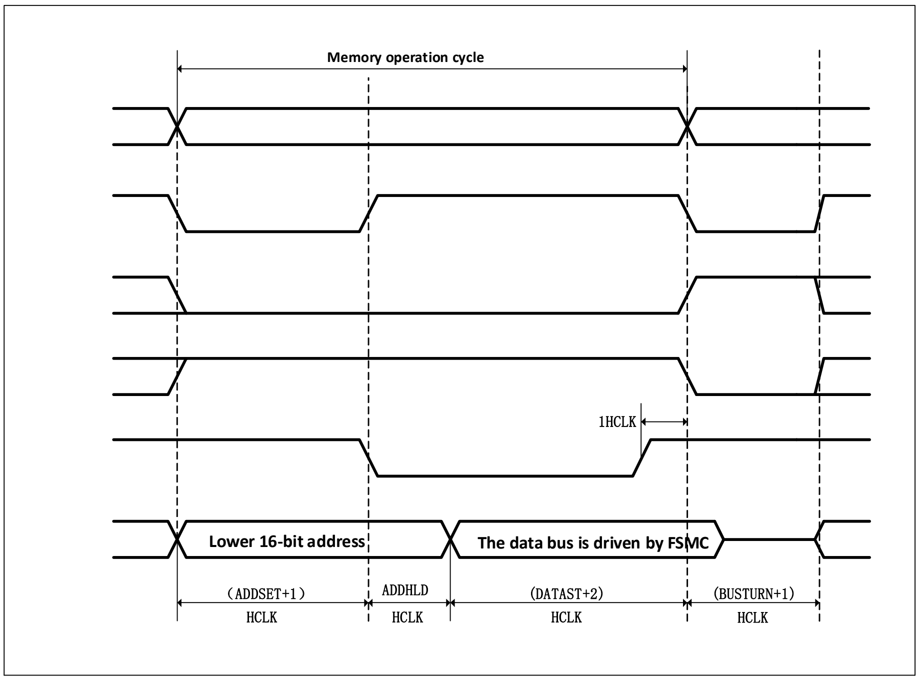


Table 26-13 Mode B FSMC_BCR1 bit field

Bitx	Name	Configuration Value
[31:20]	Reserved	0x000
19	CBURSTRW	0x0
[18:16]	Reserved	0x0
15	ASYNCWAIT	It is 1 if the memory supports this function, otherwise it is 0.
14	EXTMOD	0x1

13	WAITEN	0x0
12	WREN	Set as needed
11	WAITCFG	No effect
10	WRAPMOD	0x0
9	WAITPOL	When bit15 is 1, it is meaningful.
8	BURSTEN	0x0
7	Reserved	0x1
6	FACCEN	0x1
[5:4]	MWID	Set as needed
[3:2]	MTYP	0x2(NOR FLASH)
1	MUXEN	0x1
0	MBKEN	0x1

Table 26-14 Mode B FSMC_BTR1 bit field

Bitx	Name	Configuration Value
[31:30]	Reserved	0x0
[29:28]	ACCMOD	0x1
[27:24]	DATLAT	No effect
[23:20]	CLKDIV	No effect
[19:16]	BUSTURN	0x0
[15:8]	DATAST	Data setup time. The write operation is (DATAST+1 HCLK), and the read operation is (DATAST+3 HCLK). The minimum value of this bit is 1.
[7:4]	ADDHLD	Address hold time. (ADDHLD+1 HCLK)
[3:0]	ADDSET	Address setup time. The read operation is (ADDSET+1 HCLK).

Table 26-15 Mode B FSMC_BWTR1 bit field

Bitx	Name	Configuration Value
[31:30]	Reserved	0x0
[29:28]	ACCMOD	0x1
[27:24]	DATLAT	No effect
[23:20]	CLKDIV	No effect
[19:16]	BUSTURN	0x0
[15:8]	DATAST	Data setup time. The write operation is (DATAST+1 HCLK), and the read operation is (DATAST+3 HCLK). The minimum value of this bit is 1.
[7:4]	ADDHLD	Address hold time. (ADDHLD+1 HCLK)
[3:0]	ADDSET	Address setup time. The write operation is (ADDSET+1 HCLK).

Figure 26-9 Mode C read operation (multiplexed)

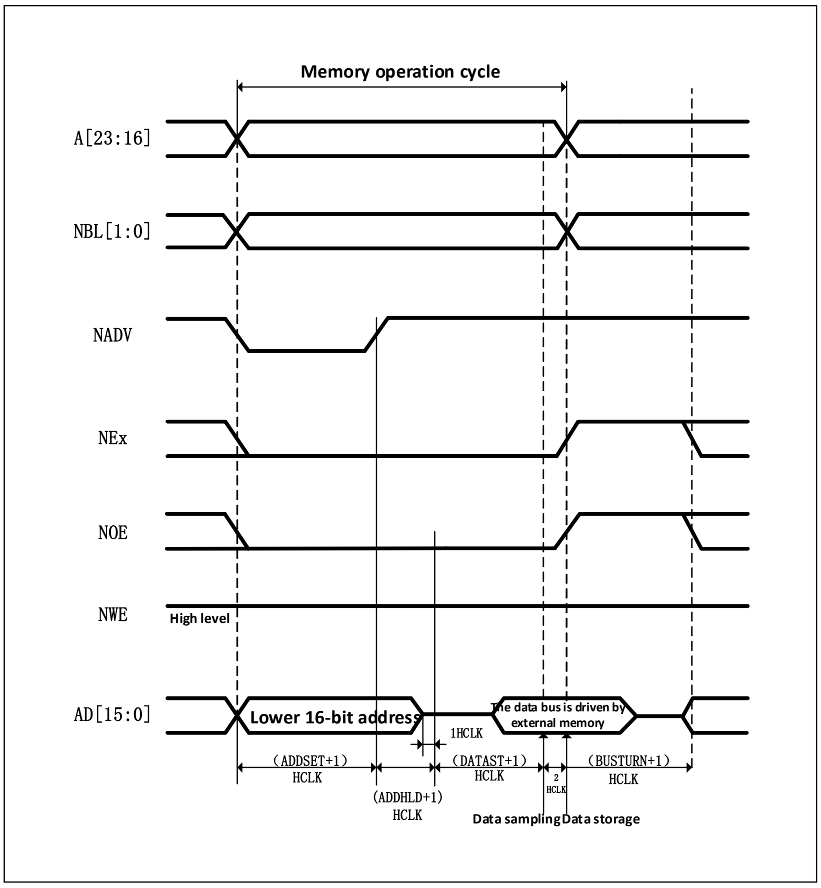


Figure 26-10 Mode C write operation (multiplexed)

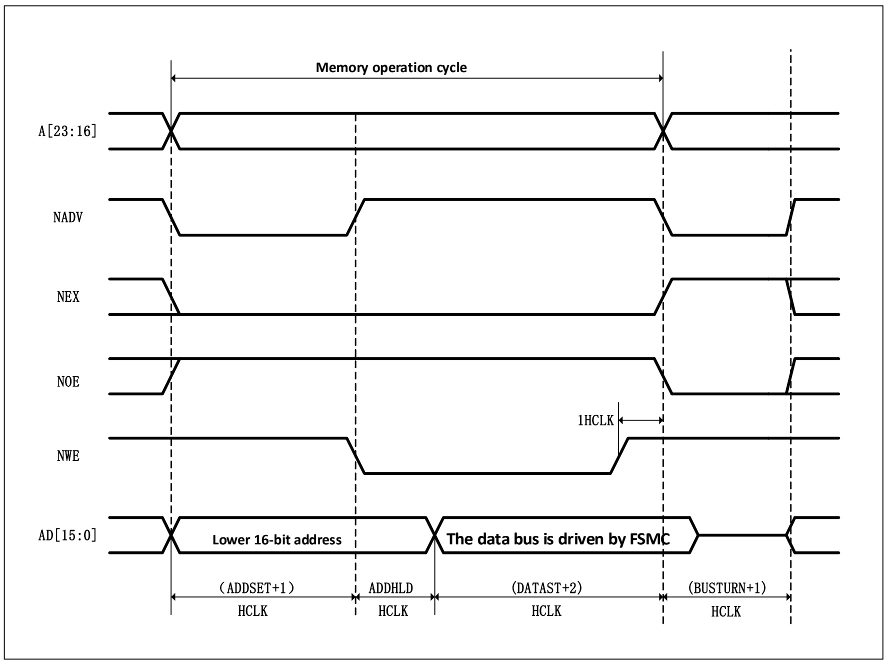


Table 26-16 Mode C FSMC_BCR1 bit field

Bitx	Name	Configuration Value
[31:20]	Reserved	0x000
19	CBURSTRW	0x0
[18:16]	Reserved	0x0
15	ASYNCWAIT	It is 1 if the memory supports this

		function, otherwise it is 0.
14	EXTMOD	0x1
13	WAITEN	0x0
12	WREN	Set as needed
11	WAITCFG	No effect
10	WRAPMOD	0x0
9	WAITPOL	When bit15 is 1, it is meaningful.
8	BURSTEN	0x0
7	Reserved	0x1
6	FACCEN	0x1
[5:4]	MWID	Set as needed
[3:2]	MTYP	0x2(NOR FLASH)
1	MUXEN	0x1
0	MBKEN	0x1

Table 26-17 Mode C FSMC_BTR1 bit field

Bitx	Name	Configuration Value
[31:30]	Reserved	0x0
[29:28]	ACCMOD	0x2
[27:24]	DATLAT	No effect
[23:20]	CLKDIV	No effect
[19:16]	BUSTURN	0x0
[15:8]	DATAST	Data setup time. The write operation is (DATAST+1 HCLK), and the read operation is (DATAST+3 HCLK). The minimum value of this bit is 1.
[7:4]	ADDHLD	Address hold time. (ADDHLD+1 HCLK)
[3:0]	ADDSET	Address setup time. The read operation is (ADDSET+1 HCLK).

Table 26-18 Mode C FSMC_BWTR1 bit field

Bitx	Name	Configuration Value
[31:30]	Reserved	0x0
[29:28]	ACCMOD	0x2
[27:24]	DATLAT	No effect
[23:20]	CLKDIV	No effect
[19:16]	BUSTURN	0x0
[15:8]	DATAST	Data setup time. The write operation is (DATAST+1 HCLK), and the read operation is (DATAST+3 HCLK). The minimum value of this bit is 1.
[7:4]	ADDHLD	Address hold time. (ADDHLD+1 HCLK)
[3:0]	ADDSET	Address setup time. The write operation is (ADDSET+1 HCLK).

Figure 26-11 Mode D read operation (multiplexed)

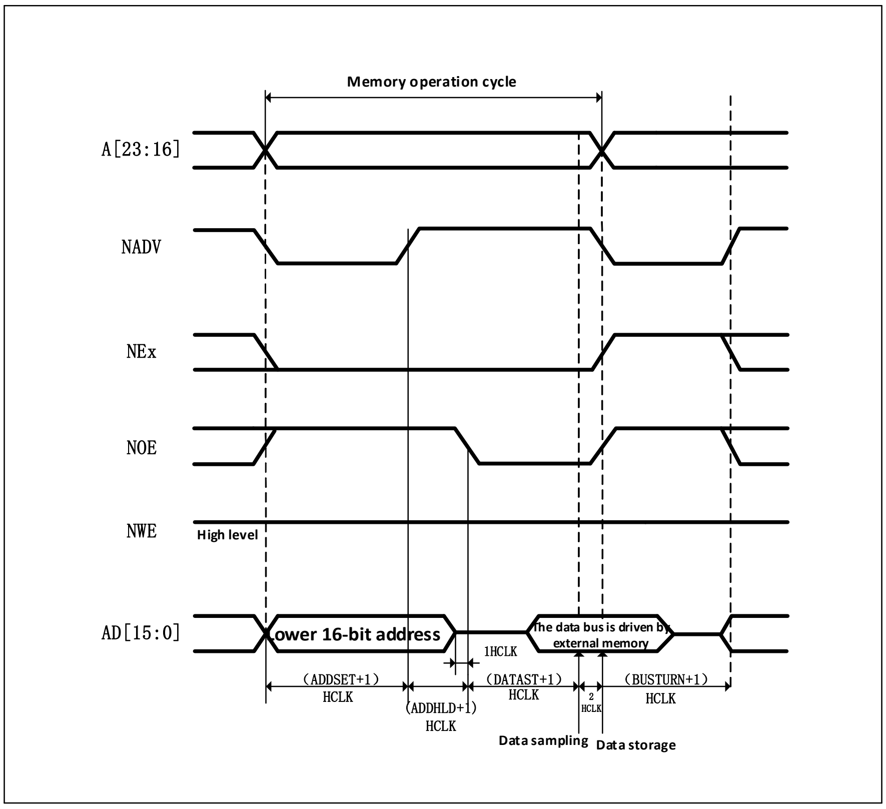


Figure 26-12 Mode D write operation (multiplexed)

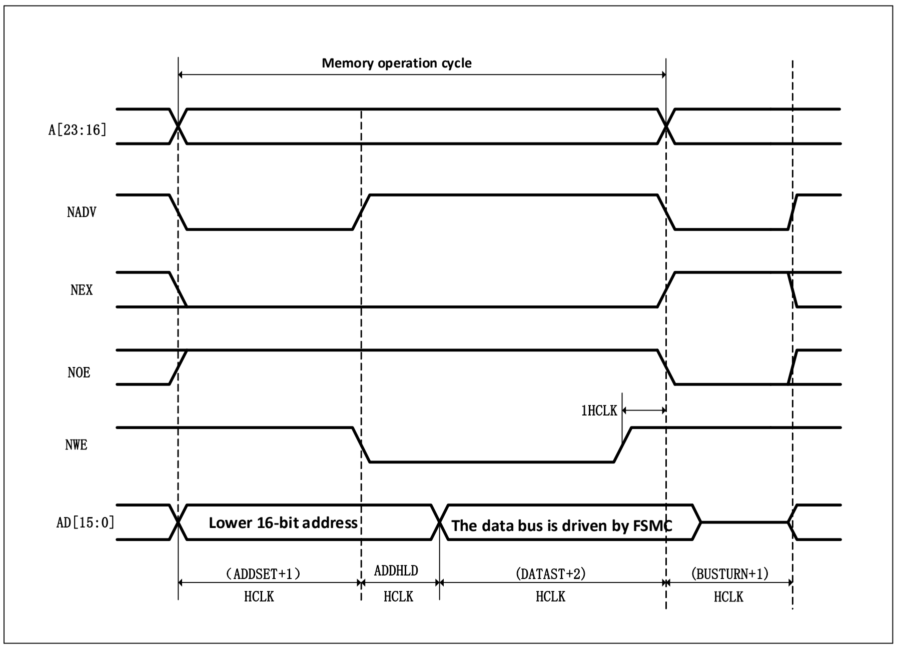


Figure 26-19 Mode D FSMC_BCR1 bit field

Bitx	Name	Configuration Value
[31:20]	Reserved	0x000
19	CBURSTRW	0x0
[18:16]	Reserved	0x0
15	ASYNCWAIT	It is 1 if the memory supports this function, otherwise it is 0.
14	EXTMOD	0x1

13	WAITEN	0x0
12	WREN	Set as needed
11	WAITCFG	No effect
10	WRAPMOD	0x0
9	WAITPOL	When bit15 is 1, it is meaningful.
8	BURSTEN	0x0
7	Reserved	0x1
6	FACCEN	Set as needed
[5:4]	MWID	Set as needed
[3:2]	MTYP	Set as needed
1	MUXEN	0x1
0	MBKEN	0x1

Table 26-20 Mode D FSMC_BTR1 bit field

Bitx	Name	Configuration Value
[31:30]	Reserved	0x0
[29:28]	ACCMOD	0x3
[27:24]	DATLAT	No effect
[23:20]	CLKDIV	No effect
[19:16]	BUSTURN	0x0
[15:8]	DATAST	Data setup time. The write operation is (DATAST+1 HCLK), and the read operation is (DATAST+3 HCLK). The minimum value of this bit is 1.
[7:4]	ADDHLD	Address hold time. (ADDHLD+1 HCLK).
[3:0]	ADDSET	Address setup time. The read operation is (ADDSET+1 HCLK).

Table 26-21 Mode D FSMC_BWTR1 bit field

Bitx	Name	Configuration Value
[31:30]	Reserved	0x0
[29:28]	ACCMOD	0x3
[27:24]	DATLAT	0x0
[23:20]	CLKDIV	0x0
[19:16]	BUSTYRN	0x0
[15:8]	DATAST	Data setup time. The write operation is (DATAST+1 HCLK), and the read operation is (DATAST+3 HCLK). The minimum value of this bit is 1.
[7:4]	ADDHLD	Address hold time. (ADDHLD+1 HCLK).
[3:0]	ADDSET	Address set time. The write operation is (ADDSET+1 HCLK).

26.2.3.4 Timing diagram of NOR/PSRAM synchronous transfer address/data multiplexing

Data delay and NOR FLASH delay should be noted that the DATALAT value must be consistent with the definition in the NOR FLASH configuration register. The clock period when the NADV signal is low is not counted in the delay parameter. The DATLAT parameter of the FSMC can be DATALAT+2 or DATALAT+3. For special memories, the NWAIT signal will be generated during the data retention period, and DATLAT needs to be set to the minimum value. For other memories, the NWAIT signal will not be generated during the data hold time phase, the data hold time of FSMC and memory must be set consistently.

For a single batch transmission, it should be noted that the memory configuration bit is synchronized in batch mode. If 16bit data is transmitted, FSMC will perform a batch transmission with a length of 1. If 32bit data is transmitted, the FSMC is divided into 2 times of 16bit transmission, and the length of an execution is 1. 2 batch transfers.

In NOR FLASH synchronous batch mode access, after the hold time (DATLAT+1 CLK), if the NWAIT signal is detected to be low, the FSMC needs to insert a wait period before NWAIT becomes high. When NWAIT becomes high, FSMC considers the data valid. During the wait state insertion period controlled by the NWAIT signal, the controller will continue to send clock pulses to the memory, hold the chip select signal and output the valid signal, while ignoring the invalid data signal. In bulk transfer mode, the NWAIT signal of NOR FLASH selects 2 timing configurations by configuring the WAITCFG bit.

Figure 26-13 Synchronous bus multiplexing read operations (multiplexed)

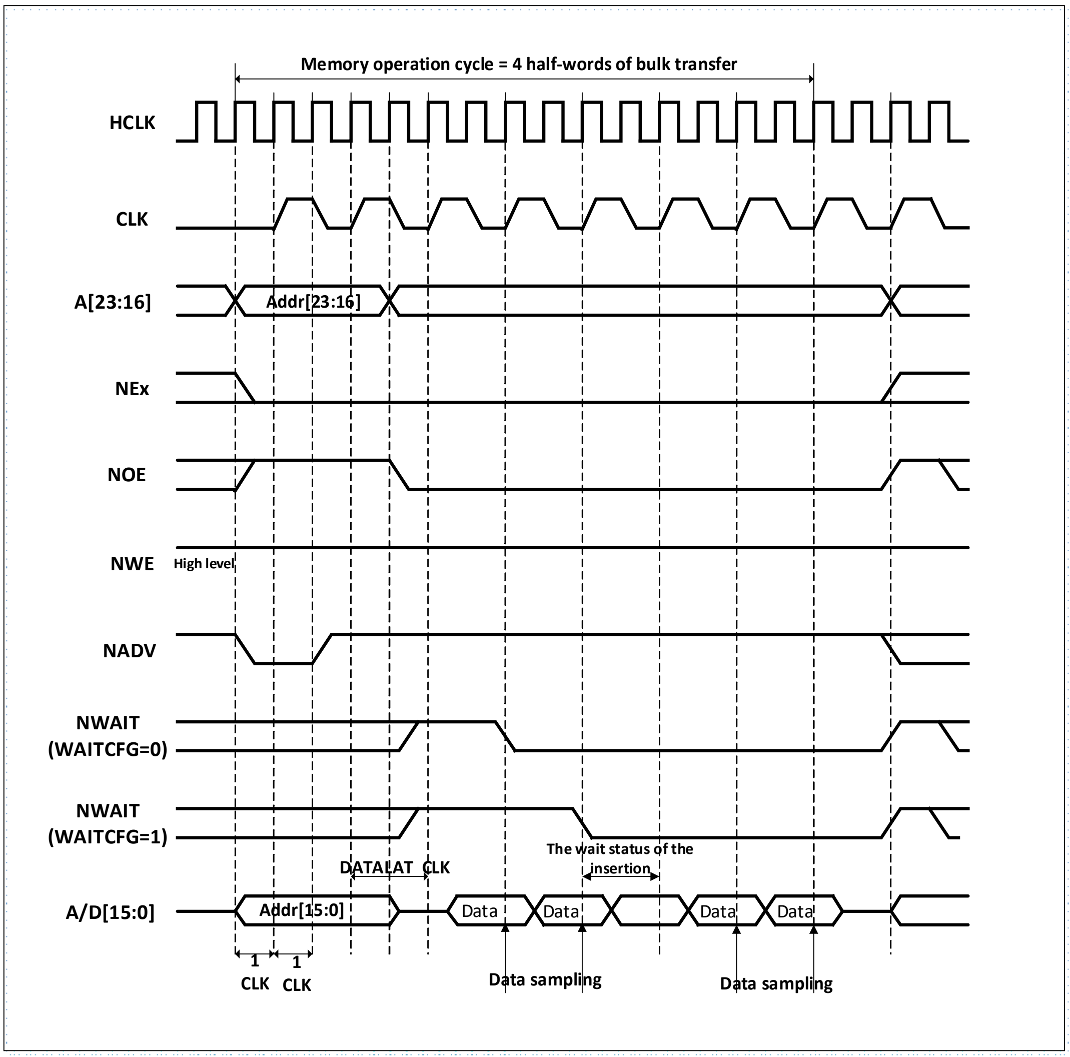


Table 26-22 Mode D FSMC_BCR1 bit field

Bitx	Name	Configuration Value
[31:20]	Reserved	0x000
19	CBURSTRW	No effect
[18:16]	Reserved	0x0
15	ASYNCWAIT	0x0
14	EXTMOD	0x0
13	WAITEN	0x0
12	WREN	No effect
11	WAITCFG	Set as needed
10	WRAPMOD	0x0
9	WAITPOL	Set as needed
8	BURSTEN	0x1
7	Reserved	0x1
6	FACCEN	Set as needed
[5:4]	MWID	Set as needed
[3:2]	MTYP	0x1or 0x2
1	MUXEN	0x1
0	MBKEN	0x1

Table 26-23 Mode D FSMC_BTR1 bit field

Bitx	Name	Configuration Value
[31:30]	Reserved	0x0
[29:28]	ACCMOD	0x0
[27:24]	DATLAT	Data latch time
[23:20]	CLKDIV	0x0: CLK=1 HCLK 0x1: CLK=2 HCLK
[19:16]	BUSTURN	No effect
[15:8]	DATAST	No effect
[7:4]	ADDHLD	No effect
[3:0]	ADDSET	No effect

Figure 26-14 Synchronous bus multiplexing write operations (multiplexed)

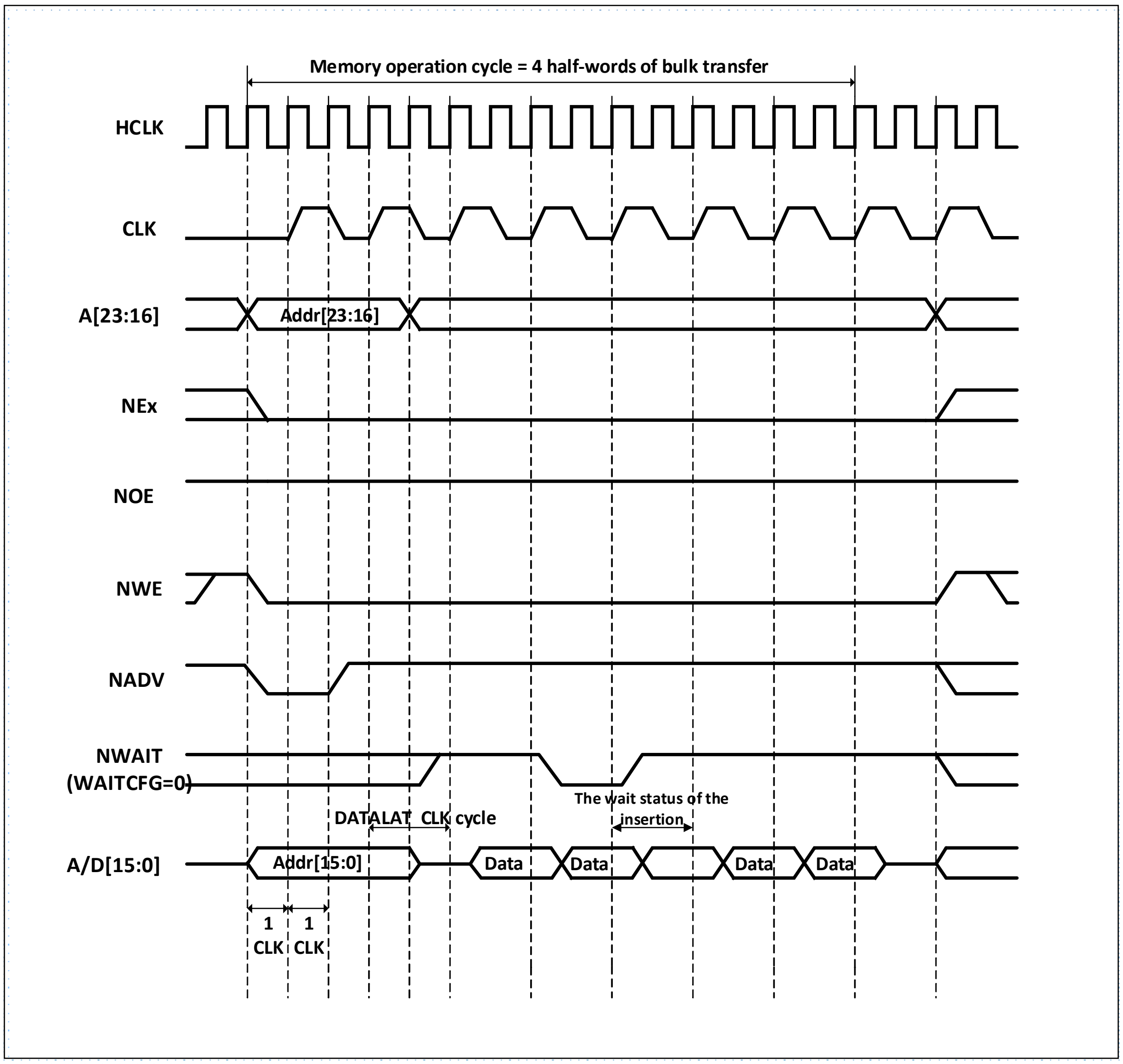


Table 26-24 Mode D FSMC_BCR1 bit field

Bitx	Name	Configuration Value
[31:20]	Reserved	0x000
19	CBURSTRW	0x1
[18:16]	Reserved	0x0
15	ASYNCWAIT	0x0
14	EXTMOD	0x0
13	WAITEN	It is 1 if the memory supports this function, 0 otherwise it is 0.
12	WREN	0x1
11	WAITCFG	0x0
10	WRAPMOD	0x0
9	WAITPOL	Set as needed
8	BURSTEN	No effect
7	Reserved	0x1
6	FACCEN	Set as needed
[5:4]	MWID	Set as needed

[3:2]	MTYP	0x1
1	MUXEN	0x1
0	MBKEN	0x1

Table 26-25 Mode D FSMC_BTR1 bit field

Bitx	Name	Configuration Value
[31:30]	Reserved	0x0
[29:28]	ACCMOD	0x0
[27:24]	DATLAT	Data latch time
[23:20]	CLKDIV	0x0: CLK=1 HCLK 0x1: CLK=2 HCLK
[19:16]	BUSTURN	No effect
[15:8]	DATAST	No effect
[7:4]	ADDHLD	No effect
[3:0]	ADDSET	No effect

26.2.4 NAND FLASH controller

FSMC supports 8bit and 16bit operation of NAND FLASH. FSMC can generate multiple address cycles as needed. In theory, FSMC has no limit to the capacity of NAND FLASH. The read and write parameters of NAND FLASH can be configured by software, see Table 26-26.

Table26-26 Software controllable NAND read and write parameters

Parameter	Function	Read and write	Parameter value range
Memory setup time	Time to setup the address before sending command	read/write	1<< T <<256 (AHB HCLK)
Memory wait time	Minimum duration to send a command	read/write	1<< T <<256 (AHB HCLK)
Memory hold time	Time to hold the address at the end of command transmission, also the data hold time during write operation	read/write	1<< T <<255 (AHB HCLK)
Memory data bus high resistance state time	The data bus remains high resistance state time after initiating a write operation	write	0<< T <<255 (AHB HCLK)

26.2.4.1 External memory interface signals

8bit and 16bit NAND FLASH interface, see Table 26-27, Table 26-28.

Table26-27 8-bit NAND FLASH

FSMC Pin	Direction	Description
A[17]	Output	Address latch enable signal (ALE)
A[16]	Output	Command latch enable signal (CLE)

D[7:0]	Input/output	8-bit bidirectional address/data multiplexing bus
NCE[2]	Output	Chip select line
NOE	Output	Output enable
NWE	Output	Write enable
NWAIT	Input	Wait signal line

Table 26-28 16-bit NAND FLASH

FSMC Pins	Direction	Description
A[17]	Output	Address latch enable signal (ALE)
A[16]	Output	Command latch enable signal (CLE)
D[15:0]	Input/output	16-bit bidirectional address/data multiplexing bus
NCE[2]	Output	Chip select line
NOE	Output	Output enable
NWE	Output	Write enable
NWAIT	Input	Wait signal line

26.2.4.2 Supported memory and operation methods

See Table 26-29 for supported memories and operation methods.

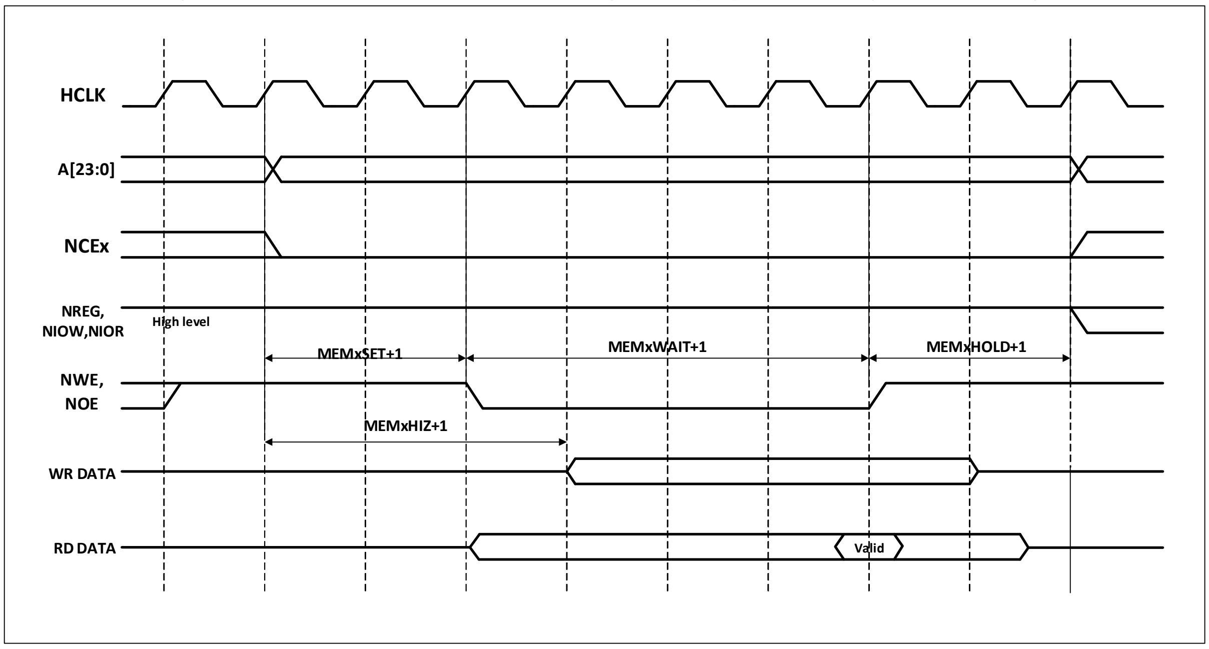
Table 26-29 Supported memory and operation modes

Memory	Mode	AHB data width	Memory width	Description
8bit NAND	Asynchronous read	8	8	
	Asynchronous write	8	8	
	Asynchronous read	16	8	Divided into 2 FSMC access
	Asynchronous write	16	8	Divided into 2 FSMC access
	Asynchronous read	32	8	Divided into 4 FSMC access
	Asynchronous write	32	8	Divided into 4 FSMC access
16bit NAND	Asynchronous read	8	16	
	Asynchronous read	16	16	
	Asynchronous write	16	16	
	Asynchronous read	32	16	Divided into 2 FSMC access
	Asynchronous write	32	16	Divided into 2 FSMC access

26.2.4.3 NAND FLASH timing diagram (including pre-wait function)

NAND FLASH operation timing control, involving FSMC_PMEM2 and FSMC_PATT2 timing registers, each register contains 4 parameters. The 3 parameters MEMHOLD, MEMWAIT and ATTSET correspond to the number of HCLK cycles in the 3 stages of operating the NAND FLASH, and the MEMHIZ parameter corresponds to the timing when the FSMC starts to drive the data bus. NAND FLASH controller general memory space access timing, see Figure 26-15.

Figure 26-15 NAND FLASH controller general purpose memory access timing



26.2.4.4 NAND FLASH operating procedure

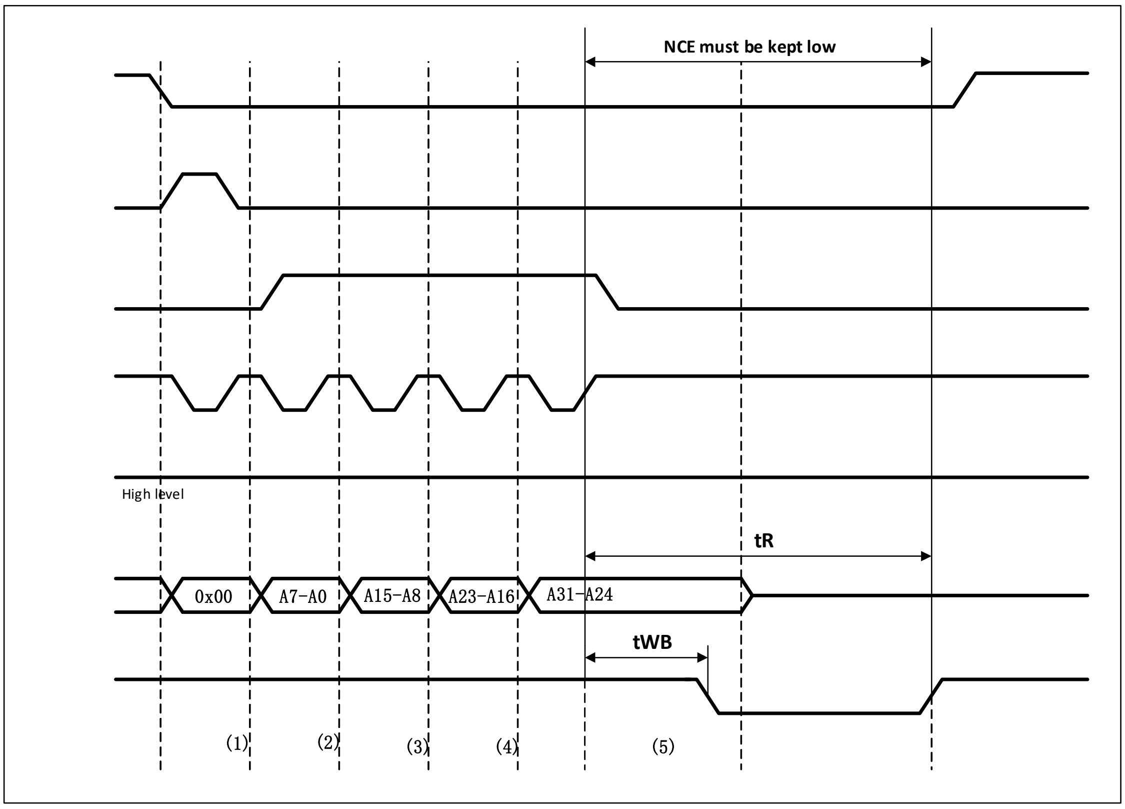
The command latch enables signal (CLE) and address latch enable signal (ALE) of NAND FLASH are driven by FSMC address lines A16 and A17 respectively. Therefore, when sending commands or addresses to NAND FLASH operations, it is necessary to perform a write operation on the specified address of the storage space. The read operation process of NAND FLASH is as follows:

1） Configure the PWID, PTYP, PWAITEN and PBKEN of the FSMC_PCR2, FSMC_PMEM2 and FSMC_PATT2 registers according to the NAND FLASH characteristics to be operated.
2） Write a command byte in the general storage space. During the period when NWE is low, CLE outputs a high level. At this time, the command byte is recognized as a command by NAND FLASH, and the command is latched. The subsequent page read operation does not need to send the command repeatedly.
3） Then, by writing 4 bytes to the memory space, it is used as the starting address of the read operation. During the period when NWE is low, ALE outputs high level, and 4 bytes are recognized by NAND FLASH as the starting address of the read operation at this time. When operating the attribute storage space, the FSMC can be made to generate different timings and realize some NAND FLASH pre-wait functions.
4） Before the FSMC controller starts a new operation, it needs to wait for the R/NB signal of the NAND FLASH to become a high level. During the wait period, the FSMC controller needs to keep the NCE signal at a low level.
5） The FSMC controller can read the memory page byte by byte from the NAND FLASH by operating the general memory space.

26.2.4.5 NAND FLASH pre-wait function

For some special NAND FLASHs, after the last address byte is input, the R/NB signal changes to a low level, see Figure 26-16. In the figure (1) is the CPU write byte command 0x00 to 0x70010000; in the figure (2) the CPU writes the NAND FLASH address A7-A0 to 0x70020000; in the figure (3) the CPU writes the NAND FLASH address A15-A8 to 0x70020000; (4) in the figure is the address A23-A16 to 0x70020000 of the CPU writing NAND FLASH; (5) in the figure is the address A31-A24 to 0x70020000 of the CPU writing NAND FLASH;

Figure 26-16 Special operation on NAND FLASH



When using this function, the timing of tWB is guaranteed by configuring the MEMHOLD bit in the FSMC_PMEM2 register. After reading and writing operations to NAND FLASH, the FSMC controller will insert (MEMHOLD+1) HCLK between the rising edge of the NEW signal and the next operation. Keep the delay. In order to solve this problem, the attribute space is used to configure the ATTHOLD value so that it conforms to the timing of tWB, and at the same time, it is necessary to keep MEMHOLD at a minimum value. At this time, only when writing the last byte of the NAND FLASH address, the FSMC controller needs to write the attribute storage space, and the rest of the NAND FLASH read and write operations use the general storage space.

26.2.4.6 NAND FLASH ECC features

FSMC's NAND FLASH controller includes an error correction code calculation hardware module, which can effectively reduce the software workload of the CPU when processing error correction codes. When the ECC module reads and writes NAND FLASH, it supports correction of 1 bit error and detection of 2-bit errors in every 256, 512, 1024, 2048, 4096 or 8192 bytes. The page size corresponds to the valid bits of the ECC result, see Table 26-30. After the ECC calculation circuit is enabled, the ECC module monitors the data bus and read/write signals (NCE and NWE) of the NAND FLASH. Pay attention to the following when using ECC:

- When accessing the NAND FLASH, the data appearing on the D[15:0] bus is latched and used for ECC calculations.
- After the specified number of bytes have been written into or read from the NAND FLASH, the software reads the ECC value from the FSMC_ECCR2 register. To calculate ECC again, first clear

ECCEN to 0, and then write 1 to re-enable ECC calculation.

Table 26-30 The page size corresponds to the valid bits of the ECC result

ECCPS[2:0]	Page size (bytes)	ECC significant bit
000	256	ECC[21:0]
001	512	ECC[23:0]
010	1024	ECC[25:0]
011	2048	ECC[27:0]
100	4096	ECC[29:0]
101	8192	ECC[31:0]

26.3 Register description

Table 26-31 FSMC registers

Name	Access address	Description	Reset value
R32_FSMC_BCR1	0xA0000000	SRAM/NOR-Flash chip select control register 1	0x000030DX
R32_FSMC_BTR1	0xA0000004	SRAM/NOR-Flash chip select timing register 1	0x0FFFFFFF
R32_FSMC_PCR2	0xA0000060	NAND-Flash control register 2	0x00000018
R32_FSMC_SR2	0xA0000064	FIFO status and interrupt register 2	0x00000040
R32_FSMC_PMEM2	0xA0000068	General purpose memory timing register 2	0xFCFCFCFC
R32_FSMC_PATT2	0xA000006C	Attribute memory timing register 2	0xFCFCFCFC
R32_FSMC_ECCR2	0xA0000074	ECC result register 2	0x00000000
R32_FSMC_BWTR1	0xA0000104	SRAM/NOR-Flash write timing register 1	0x0FFFFFFF

26.3.1 SRAM/NOR-Flash Chip Select Control Register 1 (FSMC_BCR1)

Offset address: 0x00

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ASYNCWAIT	EXTMOD	WAITEN	WREN	WAITCFG	WRAPMOD	WAITPOL	BURSTEN	Reserved	FACCEN	MWID[1:0]		MTYP[1:0]		MUXEN	MBKEN

Bit	Name	Access	Description	Reset value
[31:20]	Reserved	RO	Reserved	0
19	CBURSTRW	RW	Bulk write enable. 1: Write operation is in synchronous mode; 0: Write operation is in asynchronous mode.	0
[18:16]	Reserved	RO	Reserved	0
15	ASYNCWAIT	RW	Wait signal during asynchronous transfer. 1: Enable wait signal during asynchronous	0

			transfer; 0: Disable wait signal during asynchronous transfer.	
14	EXTMOD	RW	Extended mode enable. 1: Enable writing to use FSMC_BWTR; 0: Disable writing to use FSMC_BWTR.	0
13	WAITEN	RW	Wait enable. When the flash memory is in bulk transfer mode, this bit controls whether the wait signal is inserted according to the NWAIT signal. 1: Enable NWAIT signal; 0: Disable NWAIT signal.	1
12	WREN	RW	Write enable. 1: Enable FSMC to write to the memory; 0: Disable FSMC to write to the memory.	1
11	WAITCFG	RW	Wait timing configuration. 1: NWAIT signal is valid during the wait state; 0: NWAIT signal is valid one data cycle before the wait state.	0
10	WRAPMOD	RW	Batch mode supports unaligned enable. 1: Supports direct unaligned batch operations; 0: Direct unaligned bulk operations are not supported.	0
9	WAITPOL	RW	Wait signal polarity. 1: NWAIT is active high; 0: NWAIT is active low.	0
8	BURSTEN	RW	Bulk mode enable. 1: Enable bulk mode; 0: Disable bulk mode.	0
7	Reserved	RO	Reserved	1
6	FACCEN	RW	NOR FLASH access enable. 1: Enable access to NOR FLASH; 0: Disable access to NOR FLASH.	1

[5:4]	MWID[1:0]	RW	Data bus width.
			11: Reserved;
			10: Reserved;
			01: 16 bits;
			00: 8 bits.
				01b

[3:2]	MTYP[1:0]	RW	Memory type.
			11: Reserved;
			10: NOR FLASH;
			01: PSRAM;
			00: SRAM, ROM.
				reset value

1	MUXEN	RW	Address/data multiplexing enable. 1: Address/data multiplexing; 0: Address/data are not multiplexed.	reset value

0	MBKEN	RW	Memory block enable. 1: Enable the corresponding memory block; 0: Disable the corresponding memory block.	reset value

26.3.2 SRAM/NOR-Flash Chip Select Timing Register 1 (FSMC_BTR1)

Offset address: 0x04

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

Bit	Name	Access	Description	Reset value
[31:30]	Reserved	RO	Reserved	0

[29:28]	ACCMOD[1:0]	RW	Asynchronous access mode.
			11: Mode D;
			10: Mode C;
			01: Mode B;
			00: Mode A.
				00b

[27:24]	DATLAT[3:0]	RW	Data latch time. For synchronous bulk mode (only for NOR FLASH) 1111: The first data latch time is 17 CLK clocks; 1110: The first data latch time is 16 CLK clocks; ... 0000: The first data latch time is 2 CLK clocks.	1111b

[23:20]	CLKDIV[3:0]	RW	Clock divider ratio (CLK signal). This parameter has no effect when accessing asynchronous NOR FLASH, SRAM and ROM.
			1111: 1 CLK cycle = 16 HCLK cycles;
			1110: 1 CLK cycle = 15 HCLK cycles;
			...
			0001: 1 CLK cycle = 2 HCLK cycles;
			0000: Reserved.
				1111b

[19:16]	BUSTURN[3:0]	RW	Bus recovery time. (only for NOR FLASH) 1111: bus recovery time = 16 HCLK cycles; 1110: bus recovery time = 15 HCLK cycles; ... 0000: bus recovery time = 1 HCLK cycle.	1111b

[15:8]	DATAST[7:0]	RW	Data hold time.
			11111111: data hold time = 256 HCLK cycles;
			11111110: data hold time = 255 HCLK cycles;
			...
				0xFF

			00000001: data hold time = 2 HCLK cycles; 00000000: Reserved.	
[7:4]	ADDHLD[3:0]	RW	Address hold time. (Only for asynchronous operations) 1111: Address hold time = 16 HCLK cycles; 1110: Address hold time = 15 HCLK cycles; ... 0000: Address hold time = 1 HCLK cycle.	1111b
[3:0]	ADDSET[3:0]	RW	Address setup time. (Only for asynchronous operations) 1111: Address setup time = 16 HCLK cycles; 1110: Address setup time = 15 HCLK cycles; ... 0000: Address setup time = 1 HCLK cycle.	1111b

26.3.3 NAND-Flash Control Register 2 (FSMC_PCR2)

Offset address: 0x60

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												ECCPS[2:0]			TAR[3]
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TAR[2:0]			TCLR[3:0]				Reserved		ECCEN	PWID[1:0]		PTYP	PBKEN	PWAITEN	Reserved

Bit	Name	Access	Description	Reset value
[31:20]	Reserved	RO	Reserved	0

[19:17]	ECCPS[2:0]	RW	ECC page size.
			111: Reserved;
			101: 8192Byte;
			100: 4096Byte;
			011: 2048Byte;
			100: 1024Byte;
			100: 512Byte;
			100: 256Byte.
				0

[16:13]	TAR[2:0]	RW	ALE to RE delay. 1111: 16 HCLK cycles; 1110: 15 HCLK cycles; ... 0000: 1 HCLK cycle.	0
[12:9]	TCLR[3:0]	RW	CLE to RE delay. 1111: 16 HCLK cycles; 1110: 15 HCLK cycles; ... 0000: 1 HCLK cycle.	0
[8:7]	Reserved	RO	Reserved	0

6	ECCEN	RW	ECC enable. 1: Enable ECC; 0: Disable ECC.	0

[5:4]	PWID[1:0]	RW	Data bus width.
			11: Reserved;
			10: Reserved;
			01: 16 bits;
			00: 8 bits.
				01b

3	PTYP	RW	Memory Type. 1: NAND flash memory; 0: Reserved.	1
2	PBKEN	RW	NAND memory enable. 1: Enable the corresponding memory block; 0: Disable the corresponding memory block.	0
1	PWAITEN	RW	Wait function enable. 1: Enable; 0: Disable.	0
0	Reserved	RO	Reserved	0

26.3.4 FIFO Status and Interrupt Registers 2 (FSMC_SR2)

Offset address: 0x64

31	30	29	28	27	26	25	24	23	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									FEMPT	IFEN	ILEN	IREN	IFS	ILS	IRS

Bit	Name	Access	Description	Reset value
[31:7]	Reserved	RO	Reserved	0
6	FEMPT	RO	FIFO empty flag. 1: FIFO empty; 0: FIFO not empty.	1
5	IFEN	RW	Interrupt falling edge detection enable. 1: Enable; 0: Disable.	0
4	ILEN	RW	Interrupt high level detection enable. 1: Enable; 0: Disabled.	0
3	IREN	RW	Rising edge interrupt detection enable. 1: Enable; 0: Disable.	0
2	IFS	RW	Interrupt falling edge status. 1: Interrupt falling edge generated; 0: No falling edge generated.	0

1	ILS	RW	Interrupt high status. 1: Interrupt high level generated; 0: No interrupt high level generated.	0
0	IRS	RW	Interrupt rising edge status. 1: No interrupt rising edge generated; 0: Interrupt rising edge generated.	0

26.3.5 General-purpose Memory Timing Registers 2 (FSMC_PMEM2)

Offset address: 0x68

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

Bit	Name	Access	Description	Reset value
[31:24]	MEMHIZx	RW	High-impedance time of the general memory data bus. 11111111: NAND FLASH is 255 HCLK cycles; 11111110: NAND FLASH is 254 HCLK cycles; ... 00000000: NAND FLASH is 0 HCLK cycles.	0xFC

[23:16]	MEMHOLDx	RW	General memory hold time.
			11111111: 255 HCLK cycles;
			11111110: 254 HCLK cycles;
			...
			00000001: 1 HCLK cycle;
			00000000: Reserved.
				0xFC

[15:8]	MEMWAITx	RW	General memory wait time. (Need to add the wait period introduced by the NWAIT signal going low)
			11111111: 256 HCLK cycles;
			11111110: 255 HCLK cycles;
			...
			00000001: 2 HCLK cycles;
			00000000: Reserved.
				0xFC

[7:0]	MEMSETx	RW	General memory setup time. 11111111: NAND FLASH 257 HCLK cycles; 11111110: NAND FLASH 256 HCLK cycles; ... 00000000: NAND FLASH 2 HCLK cycles.	0xFC

26.3.6 Attribute Memory Timing Register 2 (FSMC_PATT2)

Offset address: 0x6C

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

Bit	Name	Access	Description	Reset value
[31:24]	ATTHIZx	RW	High-impedance time of the attribute memory data bus. 11111111: 255 HCLK cycles; 11111110: 254 HCLK cycles; ... 00000000: 0 HCLK cycles.	0xFC

[23:16]	ATTHOLDx	RW	Attribute memory hold time.
			11111111: 255 HCLK cycles;
			11111110: 254 HCLK cycles;
			...
			00000001: 1 HCLK cycle;
			00000000: Reserved.
				0xFC

[15:8]	ATTWAITx	RW	Attribute memory wait time. (Need to add the wait period introduced by the NWAIT signal going low)
			11111111: 256 HCLK cycles;
			11111110: 255 HCLK cycles;
			...
			00000001: 2 HCLK cycles;
			00000000: 1 HCLK cycle.
				0xFC

[7:0]	ATTSETx	RW	Attribute memory setup time. 11111111: 256 HCLK cycles; 11111110: 255 HCLK cycles; ... 00000000: 1 HCLK cycle.	0xFC

26.3.7 ECC Result Register2 (FSMC_ ECCR2)

Offset address: 0x74

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

Bit	Name	Access	Description	Reset value
[31:0]	ECC	RO	ECC calculation result	0

26.3.8 SRAM/NOR-Flash Write Timing Register1 (FSMC_BWTR1)

Offset address: 0x104

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		ACCMOD [1:0]		DATLAT[3:0]				CLKDIV[3:0]				BUSTYRN[3:0]			

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

Bit	Name	Access	Description	Reset value
[31:30]	Reserved	RO	Reserved	0

[29:28]	ACCMOD[1:0]	RW	Access mode. (Valid when EXTMOD is 1)
			00: Access mode A;
			01: Access mode B;
			10: Access mode C;
			11: Access mode D.
				00b

[27:24]	DATLAT[3:0]	RW	Data latch time. (NOR FLASH synchronous batch mode) CLK is the flash clock. 1111: 17 CLK cycles; 1110: 16 CLK cycles; ... 0000: 1 HCLK cycle.	1111b

[23:20]	CLKDIV[3:0]	RW	Clock division ratio (CLK).
			1111: 1 CLK cycle = 16 HCLK cycles;
			1110: 1 CLK cycle = 15 HCLK cycles;
			...
			0001: 1 CLK cycle = 2 HCLK cycles;
			0000: Reserved.
				1111b

[19:16]	Reserved	RO		1111b

[15:8]	DATAST[7:0]	RW	Data retention time.
			11111111: 256 HCLK cycles;
			11111110: 255 HCLK cycles;
			...
			00000001: 2 HCLK cycles;
			00000000: Reserved.
				0xFF

[7:4]	ADDHLD[3:0]	RW	Address hold time.
			1111: 16 HCLK cycles;
			1110: 15 HCLK cycles;
			...
				1111b

			0001: 2 HCLK cycles; 0000: Reserved.	

[3:0]	ADDSET[3:0]	RW	Address setup time.
			1111: 16 HCLK cycles;
			1110: 15 HCLK cycles;
			...
			0001: 2 HCLK cycles;
			0000: 1 HCLK cycle.
				1111b

Chapter 27 Ethernet Transceiver (ETH)

The module descriptions in this chapter are applicable to some products of the CH32V3x and CH32F20x series.

The "Ethernet transceiver" mentioned in this chapter is a proper term, which means the Ethernet (Ethernet) data link layer transceiver inside the microcontroller, the communication rate is up to gigabit per second (1Gbps), it is a communication peripheral. The "MAC" mentioned in this chapter refers to the role name of the Ethernet transceiver at the data link layer and is a component of the Ethernet transceiver.

Ethernet transceiver (Ethernet Transceiver MAC) is an important high-speed communication component of microcontroller, which allows microcontroller to access Ethernet at Gigabit connection speed to achieve extremely fast data communication.

27.1 CH32F20x_D8C and CH32V30x_D8C Series

27.1.1 Main features

The Ethernet transceiver of the microcontroller is an important high-speed communication peripheral of the microcontroller. It integrates Gigabit MAC (media access controller), 32-bit wide DMA controller, management counter (MMC), precision time protocol control device (PTP) and a 10M-speed Ethernet physical layer (10BASE-T PHY). With the external Gigabit Ethernet physical layer (PHY), it can access Ethernet at a Gigabit (1Gbps) speed for data transmission and reception. The Ethernet transceiver works at the data link layer and needs to implement the TCP/IP protocol stack and interface through software. The Ethernet transceiver supports 3 MII interfaces of standard MII, RMII and RGMII to connect to the PHY. If the user wants to achieve Gigabit access speed, the RGMII interface must be used; the Ethernet transceiver controls the PHY through the SMI interface, and the timing of the interface is automatically implemented by the MAC and does not need to be generated by the user through software. The RGMII interface supports transmit clock phase inversion and relative data delay, with a maximum delay of 4 nanoseconds. The MAC of the Ethernet transceiver supports the Ethernet of the standard IEEE802.3 protocol, and supports magic frames and specific wake-up frames. The DMA controller matched with the Ethernet transceiver performs data transceiver management and memory transfer in the form of descriptors. The number of descriptors is determined by the user according to the communication intensity. The DMA controller can write received data or fetch data to be sent in the memory space specified by the descriptor at a speed of 32 bits wide. In addition, the MAC of the Ethernet transceiver also supports the IEEE1588 Precision Time Protocol.

27.1.1.1 MAC features

- The MAC is mounted on the AHB bus
- Support 10M/100M/1000M Ethernet
- Support MII/RMII/RGMII interface
- RGMII supports transmit clock delay and inversion
- Support full-duplex and half-duplex
- Support automatic insertion of frame header sequence and SFD
- Support automatic insertion and check CRC
- Support automatic calculation and verification of the checksum of IP/ICMP/TCP/UDP protocol packets
- Support sending frame length control
- Support sending gap adjustment

- Support VLAN frame
- Support perfect address filtering and HSAH filtering of frame receiving addresses
- Support frame sending address filtering
- Support multicast broadcast frame reception control
- Support promiscuous mode
- Support SMI management interface (one set each for RGMII and MII)
- Support magic frame and custom wake-up frame to wake up microcontroller
- Support dedicated Ethernet wake-up interrupt entry
- Supports data loopback at the link layer

27.1.1.2 DMA features

- 32-bit wide MAC dedicated DMA
- Minimize CPU operations
- DMA is mounted on the AHB bus, and is also the master device of the AHB bus to directly access the RAM
- Support access to RAM in a byte-aligned manner
- Manage send and receive buffers in the form of descriptors
- Part of the status feedback sent by the receiver is in the descriptor
- Can dynamically modify descriptors and buffers that are not in use
- Support manual stop or start
- Support chained or ringed formal connection descriptors
- Support multiple interrupt sources such as transceiver completion interrupt

27.1.1.3 MMC module features

- Support manual reset, stop or freeze
- Multiple counters supporting multiple counting modes for sending and receiving
- Support multiple interrupts

27.1.1.4 PTP module features

- Support IEEE 1588 protocol
- Support to automatically save the current moment when sending and receiving
- Come with a 32-bit second timer and a 32-bit signed sub-second timer
- Support to adjust the time in 2 ways: fine adjustment and coarse adjustment
- Single interrupt source
- Support PPS output

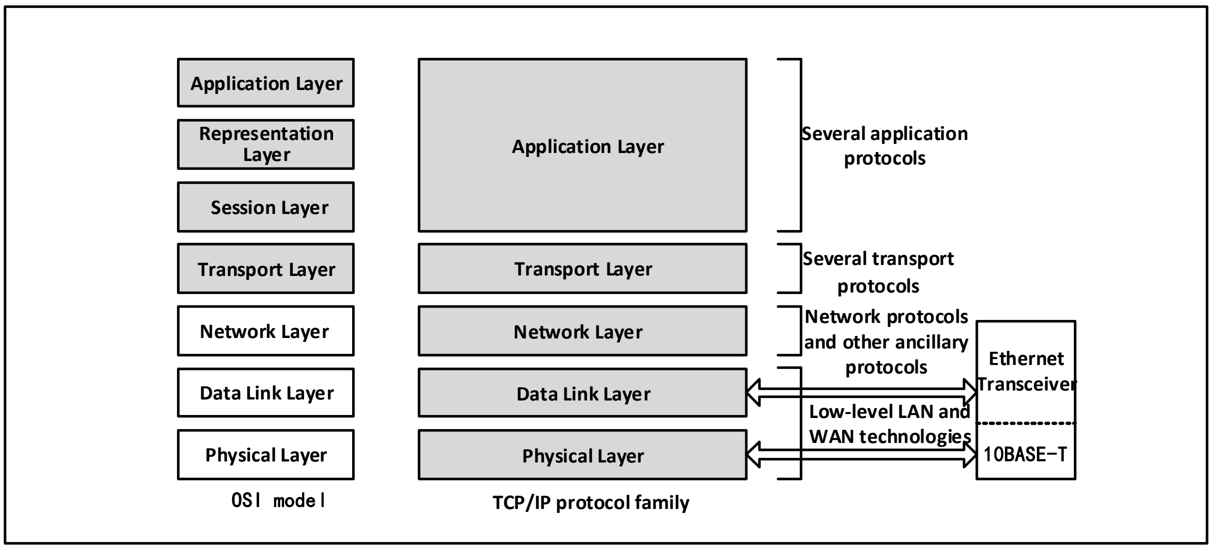
27.1.1.5 Internal 10BASE-T physical layer features

- Support 10BASE-T
- Management via SMI interface
- Support MDIX automatic rollover
- Support differential pair p/n polarity flip of MDI interface
- Support manual reset
- Support data loopback at the physical layer
- Support half-duplex

27.1.2 Overview

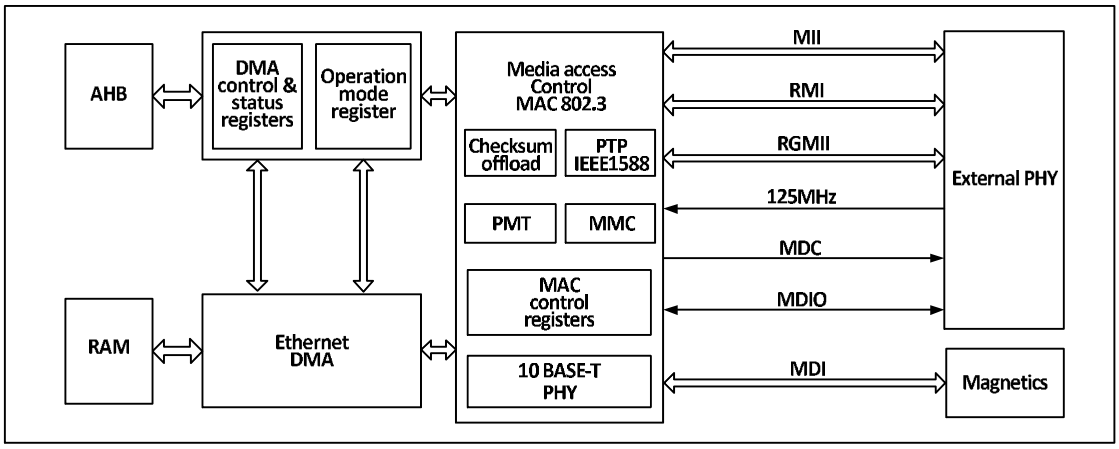
Ethernet transceivers work at the data link layer and physical layer in the OSI seven-layer model. If a data transmission rate greater than 10Mbps needs to be achieved, an external 100M or 1000M physical layer chip (PHY) is required to achieve physical connection. In order to establish the communication of protocols such as IP, TCP and UDP in the widely used Ethernet, the user also needs to implement the TCP/IP protocol stack in software. The Ethernet transceiver is composed of a media access control layer (MAC), a matching DMA and their control registers, and a 10M physical layer.

Figure 27-1 Position of Ethernet Transceivers in OSI Model and TCP/IP Model



The MAC of the Ethernet transceiver is designed according to the specifications of the IEEE802.3 protocol, and is equipped with a 32-bit wide DMA to ensure that the data can be quickly forwarded from the network cable to the memory of the microcontroller. The Ethernet transceiver has powerful and complete DMA control registers, MAC control registers and mode control registers. The microcontroller's CPU operates the registers of the Ethernet transceiver through the AHB bus. The Ethernet transceiver is connected to the gigabit Ethernet physical layer through the RGMII interface. If only the speed of 100M Ethernet is required, it can be connected to the 100M Ethernet physical layer through the MII or RMII interface. The pins of the MII, RMII and RGMII interfaces of the Ethernet transceiver are multiplexed, see section 27.3 for details. The Ethernet transceiver manages the Ethernet physical layer through the SMI interface. When using an Ethernet transceiver, the clock of the AHB bus cannot be lower than 50MHz.

Figure 27-2 Block diagram of Ethernet Transceiver



In addition, the Ethernet transceiver also supports the IEEE1588 Precision Time Protocol (PTP) to provide precise time data for microcontroller systems or off-chip.

27.1.3 Ethernet transceiver pinouts and configuration

The microcontroller supports 3 MII interfaces, namely Standard MII, RMII and RGMII. The MII/RMII interface is dedicated to 10M and 100M Ethernet physical layers, and the RGMII interface can be used for 10M, 100M and Gigabit Ethernet. The following table shows the distribution of standard MII, RMII and RGMII interfaces and internal physical layers on package pins.

Table 27-1 Media independent interface of microcontroller, media dependent interface with built-in physical layer and other related pinouts and required configuration

Pinout	RGMII	RGMII pin configuration	MII	RMII	MII/RMII pin configuration
PA2	TXCLK		MDIO	MDIO	Push-pull multiplexed output
PC1	RXCTL	Floating input	MDC	MDC	Push-pull multiplexed output
PA1	RXD3	Floating input	RX_CLK	REF_CLK	Floating input
PA7	TXD0	Push-pull multiplexed output	RX_DV	CRS_DV	Floating input
PC4	TXD1	Push-pull multiplexed output	RXD0	RXD0	Floating input
PC5	TXD2	Push-pull multiplexed output	RXD1	RXD1	Floating input
PB0	TXD3	Push-pull multiplexed output	RXD2		Floating input
PB1	125MHz_IN	Floating input	RXD3		Floating input
PB10			RXER		Floating input
PC3	RXD1	Floating input	TX_CLK		Floating input
PB11			TX_EN	TX_EN	Push-pull multiplexed output
PB12	MDC	Push-pull multiplexed output	TXD0	TXD0	Push-pull multiplexed output
PB13	MDIO	Push-pull multiplexed output	TXD1	TXD1	Push-pull multiplexed output

PC2	RXD0	Floating input	TXD2		Push-pull multiplexed output
PB8			TXD3		Push-pull multiplexed output
PA0	RXD2	Floating input	CRS		Floating input
PA3	TXCTL	Push-pull multiplexed output	COL		Floating input
PB5	PPS_OUT (Push-pull multiplexed output)				
PC6	10BASE-T_TX_p (no IO configuration required)				
PC7	10BASE-T_TX_n (no IO configuration required)				
PC8	10BASE-T_RX_p (no IO configuration required)				
PC9	10BASE-T_RX_n (no IO configuration required)				

27.1.4 Physical Layer (PHY) management and data interaction

The MAC of the Ethernet transceiver manages the PHY through the site management interface (SMI interface), and uses the media independent interface (MII interface) to exchange data with the PHY. The MII interfaces supported by the microcontroller include standard MII (usually written as MII), reduced MII (RMII) and reduced Gigabit MII (RGMII). The microcontroller uses different pins to bring out the SMI interface in MII/RMII mode and in RGMII mode.

27.1.4.1 SMI interface

The SMI interface is a serial communication interface. It uses MDC (clock line) and MDIO (data line) to access the registers of the PHY to manage the PHY, and can manage up to 32 PHY chips. Where MDC is the clock line, which is kept low when idle, and MDIO is the data line. The read and write operations of SMI and the composition of the frame are dominated by the MAC, and the user only needs to write the address and data. The relevant registers are the MII address register (R32_ETH_MACMIIAR) and the MII data register (R32_ETH_MACMIIDR).

27.1.4.1.1 Frame format

The format of the SMI interface management frame is shown in Table 27-2 below.

Table 27-2 SMI Frame Format

	Preamble	STR	OP	PHY ADR	REG ADR	T	DATA	P
Read	32 x “1”	01	10	PPPPP	RRRRR	Z0	DDDDDDDDDDDDDDDD	Z
Write	32 x “1”	01	01	PPPPP	RRRRR	10	DDDDDDDDDDDDDDDD	Z

The fields of the management frame are defined as follows:

1） Preamble: Preamble, consisting of 32 "1"s, used for MAC and PHY synchronization;

2） STR: start character, fixed to "01";

3） OP: operator, read as "10", written as "01";

4） PHY ADR: physical layer address, 5 bits;

5） REG ADR: register address, 5 bits;

6） T: Converter, 2 bits, used to switch the control of the MDIO line between the MAC and the PHY. During the read operation, the MAC maintains a high impedance to the MDIO line, and the PHY maintains a high impedance to the first bit, pulls down on the second bit, and obtains control of the MDIO; during a write operation, the MAC first sets the MDIO line high and then pulls it down. PHY maintains a high-impedance state for MDIO, and MAC maintains control over MDIO;

7） DATA: data field, 16 bits, data read and written by MAC to PHY, MSB first;

8） P: Both MAC and PHY maintain a high-impedance state for MDIO, but the pull-up resistor of PHY will pull MDIO high.

27.1.4.1.2 Read and write timing

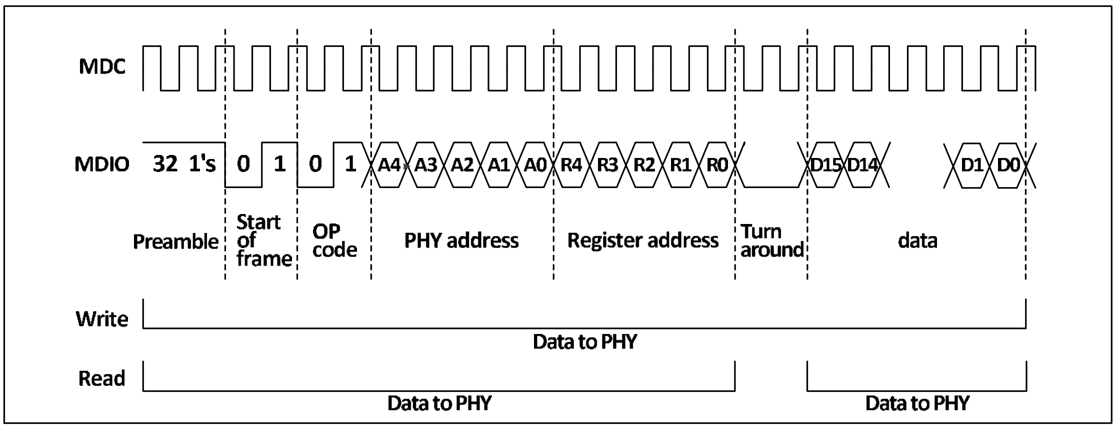
Writing to the PHY register operates as follows:

When the user sets the MII write bit (ETH_MACMIIAR:MW) and the busy bit (ETH_MACMIIAR:MB), the SMI interface will send the PHY address and register address to the PHY, and then send the data (ETH_MACMIIDR). In the process of sending data through the SMI interface, the values of the MII address register and MII data register cannot be modified; during this process, the busy bit remains high, and the write operation to the MII address register or MII data register will be ignored, and the entire transmission is affected. When the write operation is completed, the SMI interface will clear the busy bit, and the user can judge the end of the write operation according to the busy bit. The write part of Figure 27-3.

The operation of reading the PHY register is as follows:

When the user sets the MII busy bit of the MII address register (ETH_MACMIIAR) of the Ethernet MAC and keeps the MII write bit reset, the SMI interface sends the PHY address and register address, and performs the operation of reading the PHY register. During the entire transfer process, the user cannot modify the contents of the MII address register and the MII data register. During the transfer, the busy bit remains high, and writes to the MII address register or the MII data register will be ignored and will not affect the correct completion of the entire transfer. After the read operation is completed, the SMI interface will clear the busy bit and write the data read back from the PHY into the MII data register. The read section of Figure 27-3.

Figure 27-3 SMI interface read and write time diagram



27.1.4.1.3 SMI clock

Generally speaking, the SMI clock needs to be kept in a fixed range to ensure that the SMI clock actually needs to be received by the PHY. For details, refer to the manual of the PHY chip selected by the user.

The SMI clock frequency is divided from the AHB clock by the CR field of the MII address register (ETH_MACMIIAR). The following table shows the frequency division relationship between SMI clock and AHB clock. By default, divide by 42 is selected.

Table 27-3 Frequency division relationship between SMI clock and AHB clock

	Range of AHB clocks suitable for matching	SMI clock
0000b	Above 60MHz	AHB clock/42
0010b	20-35MHz	AHB clock/16
0011b	35-60MHz	AHB clock/26
other values	meaningless	

27.1.4.2 MII/RMII interface

27.1.4.2.1 Overview

Media Independent Interface (MII) is the interface for data communication between MAC and PHY. According to the speed and the number of leads, there are standard MII, RMII, GMII, RGMII and SGMII. The media independent interface generally has a group of receiving and transmitting, and each group is composed of a clock line, a number of data lines and auxiliary lines. The supported MII interfaces are: standard MII/RMII supports 10M and 100M physical layers, and RGMII supports 10M, 100M and 100M physical layers. Since the physical layers commonly used in the market are generally 10M/100M auto-negotiation physical layer, 10M/100M/1000M auto-negotiation physical layer, users should use standard MII/RMII interface when using 10M/100M auto-negotiation physical layer, and when using 10M/100M auto-negotiation physical layer /100M/1000M physical layer, users should use RGMII interface.

In general, the clock and data in the transmit direction (at the beginning of TX) of the media independent interface are sent by the MAC, and the clock and data in the receive direction (at the beginning of RX) are sent by the PHY, but the RMII clock is common. For the pin wiring in the same direction, pay attention to the equal-length wiring. For specific wiring points, please refer to the Layout manual of the physical layer chip selected by the user or Section 27.1.4.2.2 in this manual.

27.1.4.2.2 Pins

MII Pins and functions of the interface are as follows:

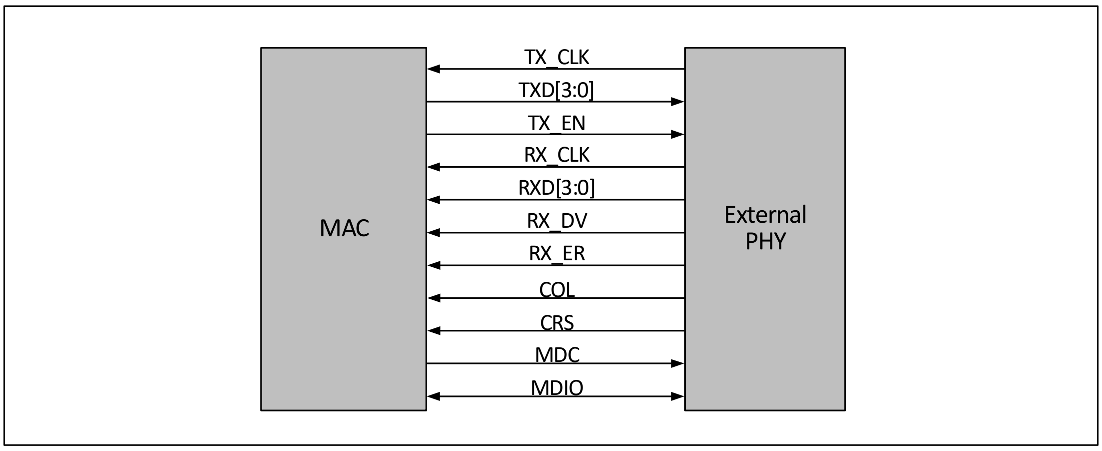
Table 27-4 MII Interface pins and functions

Pins	Features
TXC	Transmit clock (2.5MHz or 25MHz)
TXEN	Transmit data enable
TXD0	Transmit data lines [0:3]
TXD1	
TXD2	
TXD3	
RXC	Receive clock (2.5MHz or 25MHz)
RXDV	Received data valid
RXER	Received data error
RXD0	Receive data lines [0:3]
RXD1	
RXD2	
RXD3	
COL	Collision detection

CRS	Carrier detect

Figure 27-4 shows how the MAC and PHY are connected using the definition of the MII interface.

Figure 27-4 How the MAC and PHY are wired when the Ethernet transceiver uses the MII specification



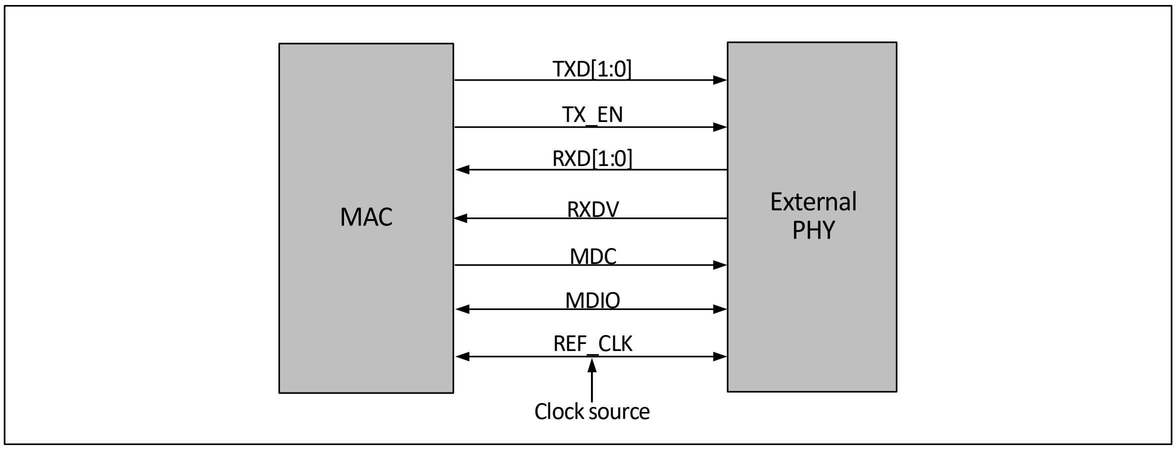
The pins and functions of the RMII interface are as follows:

Table 27-5 Pins and functions of the RMII interface

Pins	Features
CLK_REF	Receive/transmit clock (50MHz)
TXEN	Transmit data enable
TXD0	Transmit data lines [0:1]
TXD1	
CRSDV	Received data valid
RXD0	Receive data lines [0:1]
RXD1	

When using the RMII interface, the transceiver clocks share the same line for transmission, and the clock frequency is fixed at 50MHz. When RMII is used at a rate of 10Mbps, RX and TX sample data every 10 cycles, and the data line needs to retain data for 10 cycles; MAC and PHY need to use the same clock source. The MCO output of the MAC can be connected to the external clock source input pin of the PHY. Figure 27-5 shows how the MAC and PHY are wired when using the RMII interface.

Figure 27-5 Wiring diagram of MAC and PHY when using RMII interface



Note: The Ethernet transceiver needs to input the externally provided REF_CLK 50MHz clock frequency from the RXC pin.

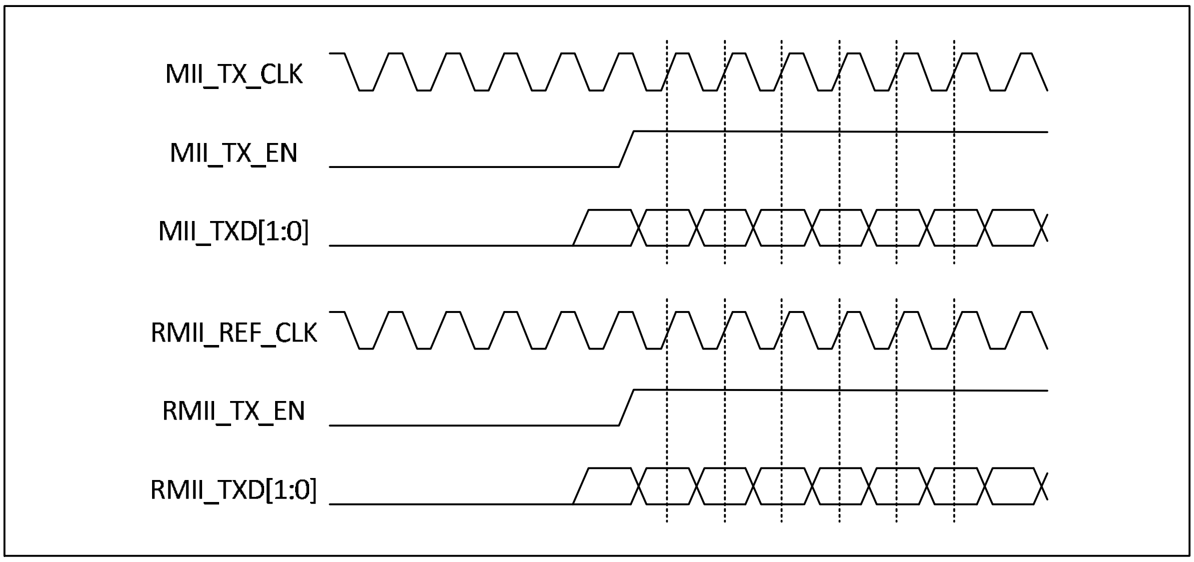
27.1.4.2.3 Timing

Difference in timing between RMII and MII:

- The clock of RMII is fixed at 50MHz, while the transceiver clock of MII works at 2.5MHz or 25MHz depending on the actual connection speed;
- RMII transceivers share the same clock line;
- RMII is a 2-bit data line, while MII is a 4-bit data line.

Figure 27-6 shows the timing of RMII and MII.

Figure 27-6 Timing diagram of MII and RMII



27.1.4.3 RGMII interface

27.1.4.3.1 Pins

The pins and functions of the RGMII are as follows

Table 27-6 RGMII pins and functions

Pins	Features
TXC	Transmit clock
TXCTL	Transmit data control
TXD0	Transmit data lines [0:3]
TXD1	
TXD2	
TXD3	
RXC	Receive clock
RXCTL	Receive data control
RXD0	Receive data lines[0:3]
RXD1	
RXD2	

RXD3	

Note: 1. TXCTL/RXCTL is used in the Ethernet transceiver of this microcontroller as a valid signal for sending and receiving data;

2. RGMII has an independent SMI interface.

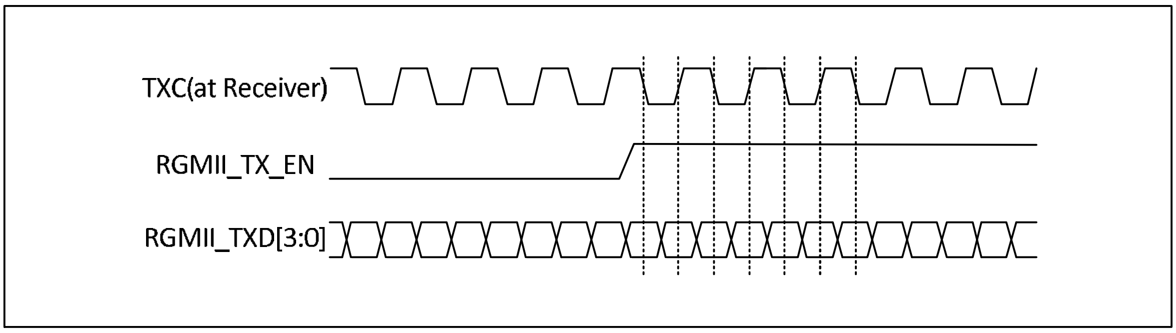
27.1.4.3.2 Timing

RGMII works in 10M and 100M rate modes, and the timing is similar to MII; when working in gigabit, the clock of RGMII is 125MHz, and the double-edge sampling method is adopted. RGMII does not support half-duplex.

Since RGMII uses double-edge sampling and operates at a clock frequency of 125MHz, circuit designers should be aware of signal integrity issues.

The clock received by the receiver of the RGMII should lag the data by 90° to ensure correct sampling. Ethernet transceivers are designed to transmit clock delay output and phase inversion.

Figure 27-7 RGMII timing diagram



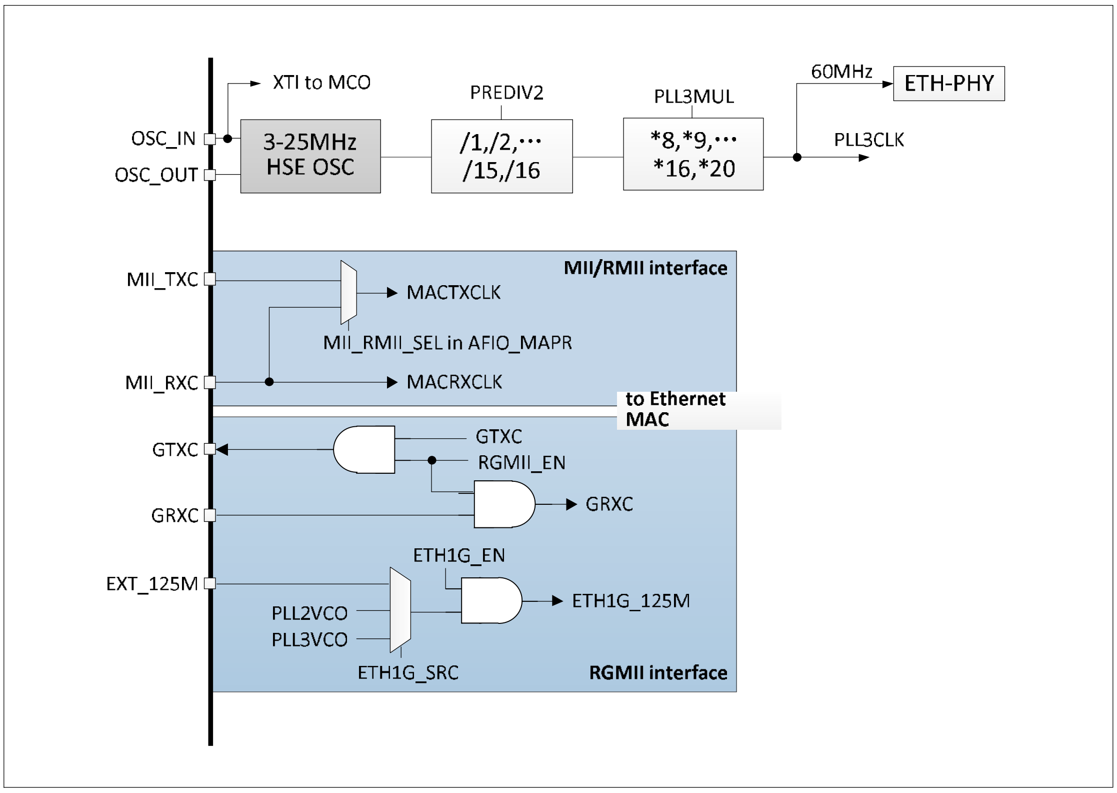
27.1.4.4 Precautions when using the internal 10M physical layer

When the internal physical layer enable bit of the extension register is set, all MII-related configurations will be ignored, the read and write of the SMI interface will be directly mapped to the internal physical layer, and the physical layer address written by the SMI interface will be ignored. Users can access the internal physical layer registers to get the physical layer connection status, see Section 27.1.8.5 for details.

27.1.4.5 Clock generation and output

27.1.4.5.1 Peripheral clock source configuration

Figure 27-8 Ethernet peripheral clock tree



As can be seen from the above figure, the clock of the internal 10M Ethernet physical layer is provided by PLL3 and must be 60MHz. When using the internal physical layer, the second bit of the extension register needs to be set. After setting, the settings related to MII/RMII/RGMII are invalid.

When using MII, the transceiver clock is provided by the external physical layer, which is 2.5MHz or 25MHz. When using RMII, REF_CLK is the only clock, which is input to the microcontroller from the RXC pin, and is fixed at 50MHz. When using RMII, the 23rd bit in the AFIO remapping register needs to be set.

When using RGMII, the MAC needs a clock with a frequency of 125MHz, which is generated by the VCO of the internal PLL2/PLL3 (the VCO output is twice the output frequency of the PLL) or an external input, and the 125MHz clock is input through the EXT_125M pin. The transmit clock of RGMII is generated by the MAC, and the receive clock is generated by the PHY. To turn on RGMII, you need to set bit 3 in the extension registers, set bit 22 in the second configuration register of RCC, and select the appropriate 125MHz clock source in [21:20]. For details, see EVT routines on the official website.

27.1.4.5.2 MCO output

It is known that MCO supports output of 8 clocks, namely HSE, HSI, system clock frequency, PLLCLK/2, PLL2CLK, PLL3CLK, PLL3CLK/2 and XTI. Users can use MCO to output the frequency required in the application, such as 25MHz clock required by common physical layer chips to save a crystal. The output frequency of the MCO should not be greater than 100MHz.

27.1.5 IEEE802.3 and IEEE1588

The IEEE802.3 protocol and its supplementary protocols constitute the official standard of the current Ethernet, which defines all aspects of the currently used Ethernet in detail. We can think of Ethernet as part of the physical layer and data link layer in the OSI model. This section discusses the parts that users need to use in the IEEE802.3 protocol from the application point of view, that is, the frame format and the content related to the MAC address. At the same time, the Ethernet transceiver of the microcontroller also supports the IEEE1588

Precision Time Protocol, which will also be discussed in part in this article.

In the Ethernet model constructed by the IEEE802.3 protocol, the data transmission unit is an Ethernet frame. Ethernet frames have different encoding forms when transmitted on different media at different rates. After receiving, they are decoded by the physical layer and sent to the MAC through the MII interface. After the MAC checksum is filtered, it is extracted by the TCP/IP protocol stack. Application information is sent to different application processes.

27.1.5.1 Frame format

The frame format of an Ethernet frame is shown in Table 27-7.

Table 27-7 Normal Ethernet frame format

Preamble	SFD	Target address	Source address	Length/Type	Data field	CRC
7 Bytes	1Byte	6 Bytes	6 Bytes	2 Bytes	46-1500 Bytes	4 Bytes
Total length: 64 to 1518 bytes plus 8 bytes of physical layer header (preamble and SFD)						

Preamble: 56-bit (7 bytes) alternating low-level and high-level jumps, the value is fixed, which is 0xAA-0xAA-0xAA-0xAA-0xAA-0xAA-0xAA in hexadecimal. This field is used for clock synchronization. This field is added/removed automatically by the hardware and the user does not need to care about it.

SFD: Frame start delimiter, 8 bits (1 byte), the value is 10101011b, SFD is used to remind the receiver that this is the last opportunity to synchronize the clock, followed by the target address. This byte is automatically added/removed by the hardware, and the user does not need to care about it.

Destination address: The device that sends this frame wants to receive the device address of this frame, and the address here refers to the hardware address, also known as the MAC address, which is assigned by the IEEE for the manufacturer, globally unique, 6-byte (48-bit). The MAC addresses of WCH microcontrollers have been programmed into the devices when they left the factory. The transmission of the address follows the principle of the least significant bit first.

Source Address: The hardware address of the device that sent this frame. The source address must be a unicast address.

Length/Type: 2 bytes, Ethernet is generally used as a type, and is also used as a length in the IEEE802.3 standard. It is demarcated by 1536 (i.e. 0x0600). The protocol above 1536 indicates which upper-layer protocol the data part is organized according to, for example, 0x0806 means ARP, 0x0800 means IPv4, 0x86dd means IPv6; below 1536 means data length.

Data field: Minimum 46 bytes, maximum 1500 bytes. When the data field is less than 46 bytes, you need to add padding to 46 bytes, and if it exceeds 1500 bytes, please create another frame. The data field is loaded with the data that needs to be actually sent.

CRC: Cyclic Redundancy Check, the CRC32 check is used here.

27.1.5.2 Frame sending

When the Ethernet transceiver sends a data frame, its dedicated DMA controller takes out the data to be sent from the RAM specified by the transmit descriptor (trans descript in Section 14.1.1) and pushes it into the MAC through the dedicated data bus. The filling level of the FIFO will be returned to the DMA controller. When all the data to be sent is sent, the DMA controller will send a data start signal to the MAC, and the MAC will start sending; read the data from the FIFO and send pilot, SFD, and sends actual data to MII interface (RMII/RGMII) ; after the DMA controller sends the data end signal to the MAC, the MAC adds the automatically calculated CRC.

The MAC will calculate the checksum for the checksum field according to the type of the frame sent by the configured automatic device, and replace the value of the original checksum field. This feature can be turned off.

The MAC will automatically add the CRC32 check to the padding, or it can be set to not add the CRC check bit. According to the provisions of the IEEE802.3 protocol, the length of the data field is not less than 46 bytes. When the MAC detects that the data field to be sent is shorter than 46 bytes, it will automatically add padding after the number field. If automatic padding is selected, then The CRC32 checksum must be added, ignoring the register configuration. The CRC32 verification formula used by the MAC is as follows:

$$G(x)=x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^{8}+x^{7}+x^{5}+x^{4}+x^{2}+x+1$$

When the IEEE 1588 (PTP) function is enabled, the MAC will save the current timestamp in the descriptor when sending an Ethernet frame, but will overwrite some information at the same time, and the user can read the timestamp and complete the descriptor in time to receive the interrupt..

27.1.5.3 Frame receiving

When the Ethernet transceiver receives the Ethernet frame, the Ethernet frame enters the MAC from MII, RMII or RGMII, first enters the FIFO in the MAC, and then is forwarded to the buffer in the RAM by the DMA. The MAC will filter and check the Ethernet frame. The filtering includes perfect address filtering and HASH filtering. The inspection then typically performs frame length, IP/ICMP/TCP/UDP checks and CRC checks on frames that do not pass the filtering or checks are marked out by the descriptor or dropped, and packets that exceed the length may be choked off.

With the IEEE 1588 (PTP) function enabled, the MAC will save the current timestamp in the descriptor when the frame is received, but at the same time it will overwrite part of the information, so the user can read the timestamp and complete the descriptor in time to receive the interrupt.

27.1.5.4 Frame filtering

Use the MAC frame filter to perform perfect filtering or HASH filtering on the destination MAC address and source MAC address of the received frame, and can set the frame that does not pass the filtering to be discarded, received, or marked in the receive descriptor. Please read the description of the MAC frame filter register (R32_ETH_MACFFR) for this function. The MAC has 4 built-in MAC address registers, of which MAC address register 0 is used by default to store its own MAC address, and the remaining 3 MAC address registers can be used for perfect filtering, compared with the source address or destination address of the received frame. After setting R32_ETH_MACFFR:RA, all received frames will be received, but the corresponding status will be marked in the status field of the first word of the descriptor after that, if R32_ETH_MACFFR:PM is set, there will have a similar effect, but the state will not be marked. After setting R32_ETH_MACFFR: DAIF/SAIF, the result will be flipped. Those that pass the filtering will be discarded or marked, and those that do not pass the filtering will be transferred to the buffer in RAM.

27.1.5.4.1 Unicast filtering

The MAC uses the HPF bit and the HU bit to determine whether the unicast frame is HASH filtering or perfect address filtering.

27.1.5.4.2 Multicast filtering

The MAC uses the HPF bit and the HM bit to determine whether the unicast frame is HASH filtered or perfect address filtering. When the PAM bit is set, all multicast packets can pass through the filter.

27.1.5.4.3 Broadcast filtering

By setting the BFD bit, the MAC can block all broadcast packets.

27.1.5.4.4 Source address filtering selection

The MAC address register can be enabled by setting the AE bit in the MAC address register, while setting the SA bit determines whether the MAC address register is to be compared as a source or target address sample.

27.1.5.4.5 Summary

The filter settings for target address and source address are shown in Table 27-8 and Table 27-9 respectively.

Table 27-8 Acceptance of the setting of each bit of R32_ETH_MACFFR to the target MAC address of the received frame

Frame type	M	PF	U	DAIF	M	PAM	Effect
Broadcast frame	1	-	-	-	-	-	Pass
	0	-	-	-	-	-	Fail (BFD set)
Unicast frame	1	-	-	-	-	-	All frames pass
	0	-	0	0	-	-	Pass when perfect filter matches
	0	-	0	1	-	-	Fail when perfect filter matches
	0	0	1	0	-	-	Pass when HSAH filter matches
	0	0	1	1	-	-	Fail when HSAH filter matches
	0	1	1	0	-	-	Pass when perfect filter or HSAH filter matches
	0	1	1	1	-	-	Fail when perfect filter or HSAH filter matches
Multicast frame	1	-	-	-	-	-	Pass
	-	-	-	-	-	1	Pass
	0	-	-	0	0	0	Pass when perfect filter matches
	0	0	-	0	1	0	Pass when HSAH filter matches
	0	1	-	0	1	0	Pass when perfect filter or HSAH filter matches
	0	-	-	1	0	0	Fail when perfect filter matches
	0	0	-	1	1	0	Fail when HSAH filter matches
	0	1	-	1	1	0	Fail when perfect filter or HSAH filter matches

Table 27-9 Acceptance of the setting of each bit of R32_ETH_MACFFR to the source MAC address of the received frame

Frame Type	RA	SAIF	SAF	Effect

Unicast frame	1	-	-	All frames pass
	0	0	0	Passes on perfect filter match, marks but does not discard failed frames
	0	1	0	Do not pass on perfect filter match, mark but do not discard failed frames
	0	0	1	Pass when perfect filter matches, discard failed frames
	0	1	1	Fail when perfect filter matches, discard failed frames

Note: "-" indicates that the setting of this bit is of no concern.

27.1.5.5 MMC

The function of the management counter (MMC) is mainly to count the receiving and sending status of various instruction frames and the running status of the MAC. It can generate settings and generate interrupts. In general, we can get the number of currently received good frames by receiving the "good" frame counter (MMCRGUFCR), get the number of successfully sent frames by sending the "good" frame counter (MMCTGFCR), and get the number of frames successfully sent by receiving the CRC error frame counter (MMCRFCECR) to check whether there is a frame with CRC error received, in general, if the data is correct but the CRC is wrong, it can be considered that there is a problem with the RGMII line layout.

Users need to be clear about a concept: what kind of frame is a "good" frame. Under normal circumstances, as long as a frame starts to send, its frame length meets the requirements (automatic padding is turned on), and automatic CRC calculation is turned on, then it will be a "good" frame. And a frame is received, as long as its CRC is correct, the frame length is within the Ethernet frame length range, the frame length and the value of the length field are the same (if the length type field indicates the length), or there is no misalignment, then it will be considered a "good" received frame.

27.1.5.6 PMT

27.1.5.6.1 Overview

The function of the PMT (power management) part is to wake up the microcontroller from a low-power mode via Ethernet. The Gigabit Ethernet controller supports 2 types of frames to wake up the system, magic frames and (remote) wake-up frames. When Ethernet receives these 2 kinds of frames, because the microcontroller is in low-power mode, it does not necessarily generate a receive interrupt, even if it is legal, but if it is recognized by the MAC's magic frame and wake-up frame, it will generate PMT interrupt, PMT interrupt is a separate interrupt from the Ethernet interrupt. By querying the PMT control and status registers, you can find out which Ethernet frame generated the interrupt.

27.1.5.6.2 Magic frame

The magic frame (magic package, also called magic data packet) defined by AMD is a commonly used Ethernet frame to wake up the microcontroller. It has a fixed and special frame format, that is, the target network card can receive it through frame filtering. It can be encapsulated into a broadcast frame, and there are 6 consecutive bytes of full height (OXFF) in the frame, followed by the MAC address of the target network card repeated 16 times. This combination can exist anywhere in the frame, so magic frames can be encapsulated into Mac frames or IP packets or even UDP packets. Below is the format of the magic frame.
xx xx xx xx xx xx xx xx(Unlimited previous data, even no previous data)—ff ff ff ff ff ff 84 c2 e4 01 02 02 84 c2 e4 01 02 02 84 c2 e4 01 02 02 84 c2 e4 01 02 02 84 c2 e4 01 02 02 84 c2 e4 01 02 02 84 c2 e4 01 02 02 84 c2 e4 01 02 02 84 c2 e4 01 02 02 84 c2 e4 01 02 02 84 c2 e4 01 02 02 84 c2 e4 01 02 02 84 c2 e4 01 02 02 84 c2 e4 01 02 02 84 c2 e4 01 02 02 84 c2 e4 01 02 02 xx xx xx xx xx xx (Some network cards require the magic frame to be attached with a password at the end).

27.1.5.6.3 Wakeup frame

Due to the limitation of the format of the magic frame, the format of the (remote) wakeup frame can be defined by the user. As long as the Ethernet frame passes through the remote wakeup frame filter register (ETH_MACRWUFFR) group, it is regarded as a wakeup frame. Users can define the wakeup frame that meets their needs by setting the remote wakeup frame filter register group. It should be noted that the remote wakeup frame filter register group has only single address entry, the user can write data to it through 8 consecutive write operations, and then read the data originally written into the remote wakeup frame filter register by reading data 8 consecutive times.

The structure of the remote wakeup frame filter register group is as follows:

Table 27-10 Structure of the remote wakeup frame filtering register bank

Filter register 0	Byte mask register 0							
Filter register 1	Byte mask register 1							
Filter register 2	Byte mask register 2							
Filter register 3	Byte mask register 3							
Filter register 4	Reserved	Command 3	Reserved	Command 2	Reserved	Command 1	Reserved	Command 0
Filter register 5	Offset 3		Offset 2		Offset 1		Offset 0	
Filter register 6	Filter 1				Filter 0			
Filter register 7	Filter 3				Filter 2			

As can be seen from the above table, the 4 fields of byte mask register, command, and offset filter work together to determine whether a frame is a remote wakeup frame. In fact, 4 different frames that meet the requirements can be set. The highest bit of the 4-bit command field indicates what kind of frame it works on, 1 is valid only for multicast addresses, 0 is valid only for unicast addresses; the second and first bits of the command field are reserved, and the 0th bit is Enable bit, set high to enable this group of filters; the offset field indicates how many bytes are offset from the frame header to start calculating the CRC16 value, the minimum fill is 12, and the actual effective value adds one to the value of this field, if the offset field is 1, the value of CRC16 is calculated from the 13th byte from the beginning, that is, it starts from the first byte of the network layer. The 32-bit byte mask indicates the 31 bytes from the beginning of the offset field definition, which need to participate in the calculation of CRC16. The highest bit of the byte mask register must be 0, and a maximum of 31 bytes are involved in the calculation. The filter field stores the value of the CRC result that the user expects to calculate. The MAC will compare the CRC16 value calculated by itself with the value in this field. If it is consistent, the current frame will be filtered by the remote wakeup frame. The filtering of the device is recognized as a (remote) wakeup frame by the MAC. If the wake-up frame interrupt and PMT interrupt are enabled, a PMT interrupt will also be generated.

In addition, according to the bit definition of the PMT control status register, if the GU bit is set, then the unicast frame passing the frame filter is also regarded as a wakeup frame.

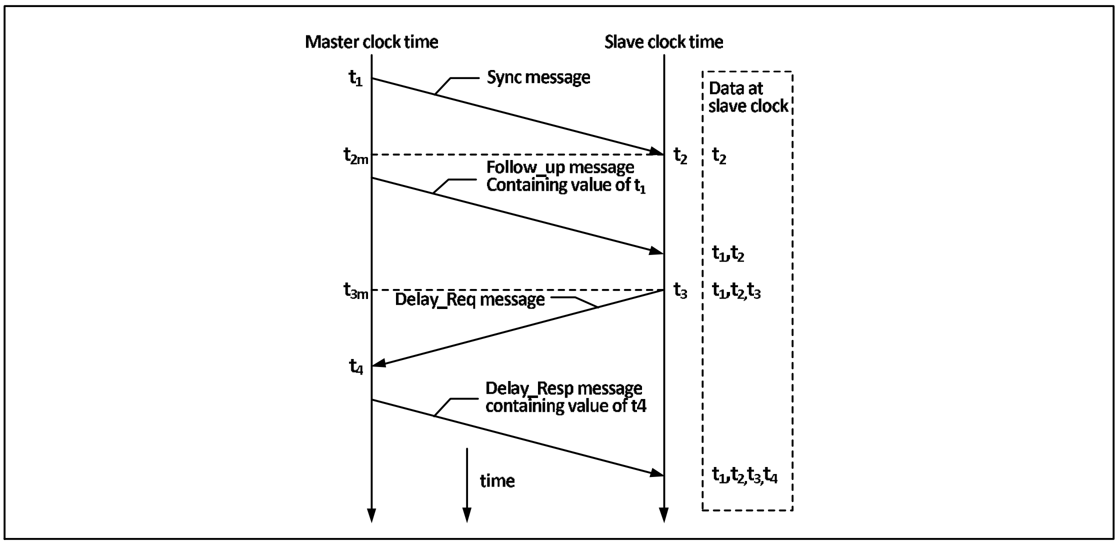
27.1.5.7 IEEE1588 PTP

27.1.5.7.1 PTP principle and implementation

The IEEE1588 standard defines a set of precise protocols for acquiring time, and its goal is to achieve time synchronization within 10 microseconds of error. The original NTP protocol can already achieve time synchronization at the level of 200 microseconds, but in fact this level cannot meet the time synchronization requirements in the field of industrial automation, so the Network Precision Clock Synchronization Committee drafted the PTP protocol, which was approved by the IEEE Standards Committee at the end of 2002 as the IEEE1588 standard.

The implementation of the PTP protocol requires both the host and the slave to accurately record the time when the MAC receives and sends Ethernet frames, which requires both the host and the slave to have their own set of high-precision time counters. Then, the master and slave supporting PTP perform time synchronization through a set of procedures, and the slave can obtain the time difference between itself and the master and correct it. The following figure shows the process of master-slave time synchronization:

Figure 27-9 IEEE1588 PTP protocol synchronization message timing diagram



Step-by-step description:

- The master sends a syn message to the slave, and the slave receives this message and records the local time t_2 when the syn message is received;
- The master sends a follow_up message to the slave, including the master time t_1 when the master sends the syn message;
- The slave sends a delay_req message to the master, and the slave records the sending time t_3;
- The master sends a delay_resq message to the slave, including the reception time t_4 of the delay_req message;

In actual use, the host sends out a syn message every 2 seconds, and every other syn message can be regarded as the follow_up message of the previous syn message, and they will all be accompanied by the sending time of the last syn message. Through this process, the slave can know the delay time Tdelay from the master network to the slave network, and then calculate the time offset between the master time and the slave time.

$$\text{Tdelay} = \frac{(t_2-t_1)+(t_4-t_3)}{2}$$

The time sent by either host minus the offset is the host's time.

The synchronization process of PTP is generally implemented through the UDP protocol, of course, the user can also implement a custom protocol. PTP is highly dependent on the latency stability of the intranet.

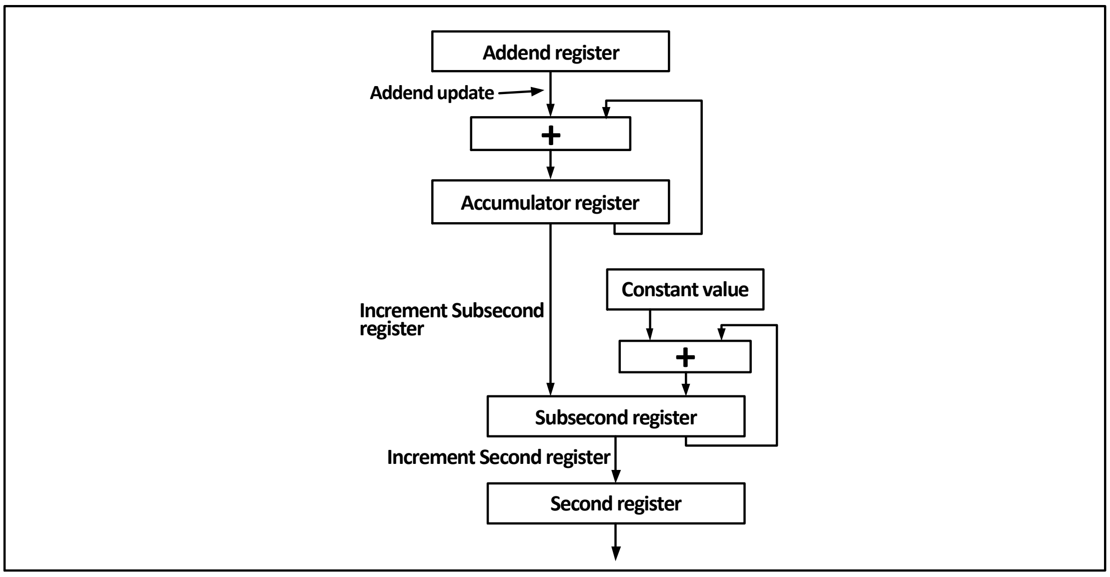
27.1.5.7.2 Local time update and correction

In order to implement the PTP host, the device needs to have a high-precision time counter locally, at least to the nanosecond level. The PTP module of the microcontroller's Gigabit Ethernet counter has a 32-bit counter in seconds, and a 31-bit sub-second counter. When the sub-second counter overflows, it will cause the second counter to increment automatically, so the resolution of the local time. The rate can be done around 0.46 nanoseconds.

There are 2 ways to update the local time, namely coarse correction and fine correction. The update timing of the coarse correction mode is determined externally. When the time update is required, the time update bit (PTPTSCR:TSSTU) of the time stamp control register is set, and the PTP module of the microcontroller subtracts or adds the value of the timestamp update registers (TSHUR, TSLUR) to the second counter and sub-second counter. Coarse correction is a simpler and more convenient time update mechanism, but less precise.

The fine correction time update method is more commonly used. The update process is as follows:

Figure 27-10 Process for updating time using fine correction



The way to use the fine correction mode requires a clear understanding of the system frequency and the fine correction process. Different from the coarse correction mode, the timing of the update event in the fine correction mode is the overflow of the accumulator (32 bits, not listed in the register list. The Accumulator register in the figure) overflows, and the accumulator will automatically increment the number register in each system clock cycle. The value of (PTPTSAR, Addend register in the figure) will generate a time update event once it overflows, that is, the value of the sub-second auto-increment register (PTPSSIR, Constant Value in the figure) will be added to the sub-second counter (Sub-second register), complete time update. The second counter is incremented when the sub-second counter overflows. In fact, the time for the sub-second counter to increment by one bit is 1/(2^31)=0.46566128730...nanoseconds. The user needs to ensure that the time spent on the accumulator overflow is exactly equal to the value of the sub-second auto-increment register multiplied by the sub-second the time for the counter to increment by one bit.

The correction of local time is relatively simple. Set the time correction bit (PTPTSCR:TSSTI) of the time stamp control register, and the value of the second counter and sub-second counter will be replaced by the value of the time stamp update counter (TSHUR, TSLUR).

27.1.6 DMA operation

27.1.6.1 Overview

In an Ethernet transceiver, data enters the FIFO from the MII interface and is then transferred into RAM by DMA. Even for the largest ordinary Ethernet frame, the data part reaches the maximum 1500 bytes, and it only takes about tens of microseconds to receive and transmit. Even with the receiving detection of the MAC, the DMA transfer time and the frame interval time are counted, a frame needs to be processed by the CPU in 100us. The advantages of Ethernet transceivers are fast speed and large throughput. In order to maintain this advantage, it is necessary to minimize the intervention of the CPU in the process of receiving and sending Ethernet frames. Here, a dedicated DMA for Ethernet transceivers is used.

The DMA used by Ethernet is 32-bit, and it is managed by the CPU through 2 data structures: traditional control and status registers, and receive and transmit descriptors. Since the DMA is 32 bits wide, the base address of the transceiver descriptor queue in RAM is required to be aligned with 4 bytes. There is no alignment requirement for the receive/transmit buffer.

27.1.6.2 DMA descriptor

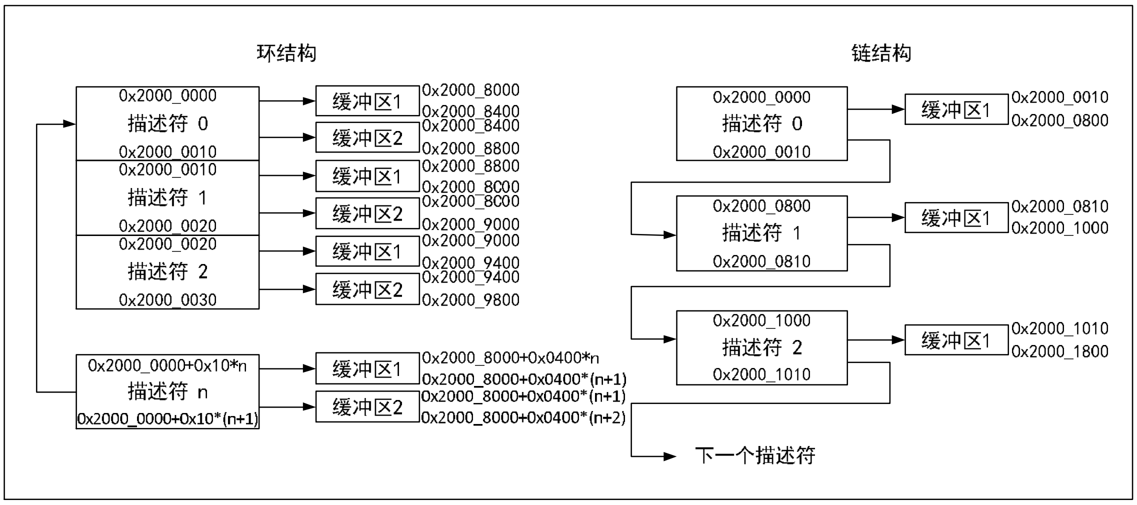
Users use traditional peripherals mainly by writing the control bits of the register, and by reading the status register to obtain the status and return information of the peripheral. These registers are independent of the core, space outside of SRAM and non-volatile memory, and actually exist, and can be called "hardware registers". Traditional communication peripherals, such as UART or SPI, have a data register to temporarily store the data sent and received, and a set of DMA to store all received data in a specific address space.

The Ethernet transceiver is different from the traditional communication transceiver because of its extremely high data transmission speed, great data throughput and unique data frame organization. The Ethernet transceiver receives a large amount of data as densely as possible. It must store the received data stream in a separate memory space in units of frames, and complete the receiving and sending operations by itself, so that the CPU can read it with a minimum of operations. And dispose of these data, release the corresponding memory space, and ensure that the DMA controller and the CPU will not conflict with the memory usage rights. The single size of the buffer that is opened up to temporarily store the Ethernet frame in the memory is generally set to the maximum packet length of the Ethernet frame. IEEE802.3 stipulates that it is 1518 bytes (including the source-end hardware address, length or type field and CRC32 calibration). The number of buffers for sending and receiving Ethernet frames is determined by the user according to the frequency of actual interaction and the memory resources of the microcontroller.

The buffer of the Ethernet frame is organized in the form of a queue. In the receiving direction, the Ethernet frame is written into the memory by DMA through the MAC, and is read and dequeued by the CPU; in the sending direction, the Ethernet frame is written to the memory and queued by the CPU, read by DMA and pushed into the transmit FIFO and out of the queue. The depth of the queue is the number of buffers. Different from ordinary communication peripherals, the starting address and number of the buffer of the Ethernet frame are not fixed in a certain register, but are managed by a special type of data structure, which is stored in the memory. A single such data structure unit manages a buffer, which holds the start address, length of the buffer, the settings that the DMA controller needs to make when calling this buffer, the send status of the write-back when the DMA send is completed, and the next such data structure the address of. This data structure plays the role of the traditional communication peripheral control register and status register, but the actual location is in the memory, so it can be called "software register", and the official name is "DMA descriptor".

DMA descriptors are divided into 2 types: sending and receiving, and each format is fixed. The storage structure of DMA descriptors is divided into 2 types, one is linked list form, that is, the fourth word of each descriptor is the address of the next descriptor, and the DMA controller will directly read the next descriptor from TDes3/RDes3. The other is a ring structure, where all descriptors must be closely arranged, DMA control takes a descriptor from the end of the current descriptor, when the DMA controller detects the TER/RER bit of TDes4/RDes4, i.e., from the beginning of the descriptor list. The address of the descriptor array must be 4-byte aligned, and the size of a single descriptor is 16 bytes.

Figure 27-11 A buffer allocation scheme which describes the ring structure and the chain structure for reference when users allocate space.



27.1.6.2.1 Transmit DMA descriptor

Figure 27-11 shows the structure of the transmit descriptor.

Table 27-11 Structure of the transmit descriptor

31 0

TDes0	OWM (31)	CTRL (30:26)	TTSE (25)	Reserved	Control （23:20）	Reserved (19:18)	TTSS (17)	State （16:0）
TDes1	Reserved （31:29）	Buffer 2-byte count （28:16）			Reserved （15:13）	Buffer 1-byte count （12:0）		
TDes2	Buffer 1 address, timestamp low							
TDes3	Buffer 2 address, address of next descriptor, timestamp high							

It can be seen from the above table that the send descriptor is composed of 4 32-bit words, namely TDes0, TDes1, TDes2 and TDes3, of which TDes9 is used for control and return transmission status, TDes1 is used to indicate the transmission length, and TDes2 is used to indicate the transmission buffer. The position of the zone or the low bit of the return transmit timestamp, TDes3 is used to indicate the address for the second transmit buffer (when TCH is not set) or the address of the next descriptor (when TCH is set), timestamp is enabled When it is sent, the high bit of the IEEE1588 timestamp will be returned. Each 32-bit word is described below.

Table 27-12 Definitions of TDes0 bits

31	30	29	28	27	26	25	24	23:22	21	19	19:18	17	16
OWN	IC	LS	FS	DC	DP	TTE	Res	CIC	TER	TCH	Res	TSS	IHE

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ES	JT	FF	IPE	LCA	NC	LCO	EC	VF	CC				ED	UF	DB

Bit	Name	Description
31	OWN	Descriptor own This bit indicates who occupies this descriptor. 0: This descriptor is owned by the CPU, and the CPU can modify the value of this descriptor; 1: This descriptor is owned by the DMA, and the CPU cannot modify this descriptor. When this bit is 0, it is set to 1 by the CPU after the CPU completes the operation on the descriptor and buffer. When this bit is 1, 0 is automatically written by the DMA after the DMA completes the operation on the descriptor and the buffer. In this way, the user software and hardware complete the operation handover of the descriptor and the receive/transmit buffer.
30	IC	Transmission completed interrupt enable When this bit is set, the transmission interrupt bit flag (ETH_DMASR: TS) will be set after the current frame is sent.
29	LS	Last segment This bit is set to indicate that the buffer indicated by this descriptor contains the end of the frame.
28	FS	First segment This bit is set to indicate that the buffer indicated by this descriptor contains the beginning of the frame.
27	DC	Disable automatic CRC calculation When this bit is set, the DMA controller will not calculate the CRC32 value of the Ethernet frame, and no value will be appended to the end of the frame. This bit is only valid when the FS bit is set. Also, the setting of the DP bit takes precedence over this bit.
26	DP	Disable auto-padding When this bit is set, the DMA controller will not add auto-padding for Ethernet frames less than 64 bytes. When this bit is 0, DMA will automatically add padding and CRC value to Ethernet frames less than 64 bytes, regardless of whether DC is set.
25	TTE	Timestamp transmission enable On the premise that ETH_PTPTSCR:TSE is set, the DMA controller will enable the IEEE1588 timestamp function in the

		Ethernet frame indicated by the face descriptor after this bit is set. This bit is only valid when FS is set.
24	Reserved	Reserved
23:22	CIC	Checksum and insert control 00: Prohibit insertion of checksum; 01: Only enable the calculation and insertion of IP header checksum; 10: Reserved; 11: Enable calculation and insertion of IP header checksum and payload checksum, enable calculation of pseudo-header checksum.
21	TER	Transmit descriptor end flag setting Users set this bit to indicate to the DMA controller that the current transmit descriptor is the last descriptor of the transmit descriptor array. The DMA controller will next read the first descriptor in the transmit descriptor array.
20	TCH	Next descriptor addresses valid This bit is set to indicate that the second address is the address of the next descriptor and not the address of the next buffer. When this bit is set, the value of TBS2 has no effect. This bit is only valid when the FS bit is set. The TER bit has higher priority than this bit.
19:18	Reserved	Reserved
17	TSS	Transmit timestamp status DMA controller sets this bit to indicate that the transmit timestamp is captured and stored in TDes2 and TDes3.
16	IHE	IP header error DMA controller checks the IP header according to the received data: for IPv4, the DMA controller checks whether the length field of the header is correct; for IPv6, the DMA controller checks whether the header is 40 bytes. In addition, the IP protocol type must be the same as the type/length field in the Ethernet frame.
15	ES	Error summary Set by the DMA controller if an error occurs while sending a frame. The ES bit is set when one of the following bits is set: UF[TDES0:1] data underflow error bits; IPE[TDES0:12] IP data error bits; FF[TDES0:13] frame empty bits; JT[TDES0:14] Jabber timeout bits; IHE[TDES0:16] IP header error bits.
14	JT	Jabber timeout When this bit is set, a jabber timeout error occurs at the MAC transmitter. This bit is only set when the JD bit (ETH_MACCR:22) is not set.

13	FF	Frame flushed This bit is set to indicate that the DMA controller clears the frame from FIFO due to the command issued by the CPU.
12	IPE	IP payload error The MAC compares the total packet length in the IPv4 or IPv6 header of the received TCP/UDP/ICMP packet with the actual packet length, and set this bit if they are inconsistent.
11	LCA	Lose carrier This bit is set to indicate that a carrier loss occurs when the frame is sent, that is, the CSR signal is invalid. This bit only works in half-duplex mode.
10	NC	No carrier This bit indicates that the carrier sense signal of the physical layer is invalid when the frame is sent. This bit only works in half-duplex mode.
9	LCO	Late collision This bit indicates that the frame collided after sending the preamble. This bit only works in half-duplex mode.
8	EC	Excessive collisions This bit indicates that a collision of more than 16 bits occurs when the frame is sent. If the RD bit in MACCR is set, this bit indicates that only one collision is sent. This bit only works in half-duplex mode.
7	VF	VLAN frame. Set when sending VLAN frames.
6:3	CC	Collision counter These bits indicate how many collisions the frame had when it was sent. No effect when EC is set. These bits only works in half-duplex mode.
2	EC	Excessive deferral This bit is set to indicate that when the DC in MACCR is set, the transmission of the frame fails due to the delay of more than 24288 bits. This bit only works in half-duplex mode.
1	UF	UF data underflow error When the DMA controller fetches data from the specified RAM and finds that the buffer is empty, the transmission enters the suspended state, and sets this bit and the relevant bit of the DMASR register.
0	DB	Deferred bit This bit is set to indicate that the transmission fails due to carrier occupancy. This bit only works in half-duplex mode.

Table 27-13 Definitions of TDes1 bits

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			TBS2												

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			TBS1												

Bit	Name	Description
31:29	Reserved	Reserved
28:16	TBS2	Size of the transmit buffer 2.
15:13	Reserved	Reserved
12:0	TBS1	Size of the transmit buffer 1.

Table 27-14 Definitions of TDes2 bits

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TBAD1/TTSL															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBAD1/TTSL															

Bit	Name	Description
31:0	TBAD1/TTSL	All 32 bits of TDes2 are used as a whole unit to store the address of the transmit buffer 1. When IEEE1588 mode is enabled, TDes2 is used to store the timestamp returned by the MAC at the end of transmission, and the MAC clears the OWN bit (TDes0: 31). This field stores the lower 32 bits of the timestamp.

Table 27-15 Definitions of TDes3 bits

31	30	29	28	27	26	25	24	23	22	21	20	19	20	17	16
TDAD2/TTSH															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDAD2/TTSH															

Bit	Name	Description
31:0	TDAD2/TTSH	This field is used to store the address of the next descriptor. When TCH is not set, this field stores the address of the transmit buffer 2. When IEEE1588 mode is enabled, TDes3 is used to store the timestamp returned by the MAC at the end of transmission, and the MAC clears the OWN bit (TDes0: 31). This field stores the upper 32 bits of the timestamp.

27.1.6.2.2 Receive DMA descriptor

Table 27-16 Receive descriptor structure

	31					0
RDes0	OWM(31)	State (30:0)				
RDes1	Control(31)	Reserved (30:29)	Buffer 2-byte count (28:16)	RER (15:14)	Reserved 13	Buffer 1 byte count (12:0)
RDes2	Buffer 1 address, timestamp low					
RDes3	Buffer 2 address, address of next descriptor, timestamp high					

It can be seen from the above figure that the receiving descriptor is also composed of 4 32-bit words, of which the first 32-bit word mainly returns the state when receiving, the second 32-bit word contains the received data length, and the third 32-bit word defines the address of the receive buffer or the low order of the return timestamp, the fourth 32-bit word is the address of the second buffer (RCH is not set), the address of the next buffer (RCH is set) or Returns the high order bit of the timestamp.
The meanings of bits of the receive descriptor are as follows:

Table 27-17 Definitions of RDes0

31	30	29:16
OWN	AFM	FL

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ES	DE	SAF	LE	OE	VLAN	FS	LS	IPHCE	LCO	PT	RWT	RE	DE	CE	PCE

Bit	Name	Description
31	OWN	Descriptor own This bit indicates who occupies this descriptor. 0: This descriptor is owned by the CPU, and the CPU can modify the value of this descriptor; 1: This descriptor is owned by the DMA, and the CPU cannot modify this descriptor. When this bit is 0, it is set to 1 by the CPU after the CPU completes the operations on the descriptor and buffer. When this bit is 1, 0 is automatically written by the DMA after the DMA completes the operations on the descriptor and buffer. In this way, the operation handover of the descriptor and the receive/transmit buffer between the user software and the hardware is completed.
30	AFM	Destination address filter fail This flag is set if the received frame does not pass the MAC's target address filter.
29:16	FL	Frame length This field is valid when ES (RDes0:15) is 0. When LS (RDes0:8) is 1, this field indicates the length of the frame received by the DMA controller, including the CRC. When LS is 0, this field indicates the cumulative length sent to the memory by the DMA controller so far, the unit is in bytes.

15	ES	Error summary
		Set when the MAC detects any of the following errors:
		CE[RDes0:1]: CRC error;
		RE[RDes0:3]: Receive error;
		RWT[RDes0:4]: Watchdog timeout;
		IPHCE[RDes0:7]: Jumbo frame (note that it is necessary to distinguish whether it is a jumbo frame or an IP header error when reporting IPHCE);
		OE[RDes0:11]: Overflow error;
		DE[RDes0:14]: Descriptor error.

14	DE	Descriptor error This bit is set to indicate that the buffer indicated by the descriptor is cut off as the buffer indicated by the descriptor cannot fit the current frame, and the DMA does not occupy the next descriptor. This bit is only valid when the LS bit (RDes0 [8]) is set.
13	SAF	Source address filter fail This bit is set to indicate that the frame does not pass the source address filter of the MAC.
12	LE	Length error Set to indicate that the actual received frame length does not match the length indicated in the Ethernet Type/Length field. This bit is only valid when FT (RDes0 [5]) is set.
11	OE	Overflow error Set to indicate that the received frame is corrupted due to an overflow of the receive FIFO.
10	VLAN	VLAN tag Set to indicate that a VLAN frame is received.
9	FS	First descriptor This bit indicates that this descriptor contains the beginning of the frame.
8	LS	Last descriptor It indicates that this descriptor contains the end of the frame.
7	IPHCE	IP header checksum error This bit is set to indicate that there is an error in the IPv4 or IPv6 header. The specific reasons may be: 1. The protocol indicated by the Ethernet frame type/length field is inconsistent with the actual IP version; 2. The IP header checksum is incorrect; 3. The length indicated by the IP header is incorrect.
6	LCO	Late collision Set to indicate that a late collision occurs. This bit only works in half-duplex mode.
5	FT	Frame type indication bit. 1 indicates that the received frame is an Ethernet type encapsulated frame (RFC 894). 0 indicates that the received frame is an IEEE802.3 type encapsulated frame (RFC1042). This bit is invalid when the frame length is less than 14 bytes. *Note: Refer to Table 27-18 for the special meaning of FT.*

4	RWT	Receive watchdog timeout Set to indicate that when the current frame is received, the watchdog times out and the current frame is truncated.
3	RE	Receive error Set to indicate that the RX_ERR signal is valid when RX_DV is valid in the process of receiving a frame.
2	DE	Dribble bit error Set to indicate that the length of the frame received by the MAC is not multiples of the 8-bit frame, and the period may be missed.
1	CE	CRC error This bit is set to indicate that the received frame has a CRC check error. This bit is valid only when the LS bit (RDes0 [8]) is set.
0	PCE	Payload checksum error This bit is set to indicate that the TCP/UDP/ICMP packet received by the MAC does not match the value indicated in the checksum field.

It can be seen that the verification mechanism is implemented in the receiving process of the MAC. In fact, the data length of this layer is described at the Ethernet frame layer (data link layer), network layer (IPv4/IPv6) and transport layer (TCP/UDP/SCTP), and some check measures are taken to ensure the correctness of the data content. The 0th, 5th and 7th bits of RDES0 are related to the checksum of the data, which is summarized in the following table.

Table 27-18 Relationship between the value of RDES:7/5/0 and the status of the received frame

RDES0:7	RDES0:5	RDES0:0	Frame status
IPHCE	FT	PCE	
1	1	1	For IP frames, there are parity errors at the IP layer and the transport layer.
1	1	0	For IP frames, there is a check error at the IP layer.
1	0	1	For IP frames, there is a check error at the transport layer.
1	0	0	For IP frames, no checksum errors were detected.
0	1	1	Non-IP frame, no inspection detection performed.
0	1	0	Reserved
0	0	1	For IP frames, unsupported transport layer protocols, no IP layer checksum errors were detected.
0	0	0	Frames whose length/type fields are less than 0x0600.

Table 27-19 Definitions of RDes1 bits

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIC	Reserved		RBS2												

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RER	RCH	Res	RBS1												

Bit	Name	Description
31	DIC	Disable reception completion interrupt setting bit.

30:29	Reserved	Unused.
28:16	RBS2	Receive buffer 2 size.
15	RER	End receive descriptor flag. It indicates that the current descriptor is the last descriptor. The DMA controller will go back to the Descriptor Pair Column Base Address Register (ETH_DMARDLAR) to fetch the next descriptor.
14	RCH	Next receive descriptor address valid bit. This bit is set to indicate that in the last 32-bit word is the address of the next receive descriptor, otherwise the address of the second buffer.
13	Reserved	Unused.
12:0	RBS1	Receive buffer 1 size.

Table 27-20 Definitions of RDes2 bits

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RBAD1/RTSL															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RBAD1/RTSL															

Bit	Name	Description
31:0	RBAD/RTSL	All 32 bits of RDes2 are used as a whole unit to store the address of the receive buffer. When IEEE1588 mode is enabled, RDes2 is also used to store the timestamp returned by the MAC at the end of reception, and the MAC will clear the OWN bit (RDes0:31).

Table 27-21 Definitions of RDes3 bits

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDAD2/RTSH															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDAD2/RTSH															

Bit	Name	Description
31:0	RDAD/RTSH	Used to store the address of the next buffer when RCR is set, otherwise used to store the address of the second buffer. When IEEE1588 mode is enabled, RDes3 is used to store the timestamp returned by the MAC at the end pf reception, and the MAC will clear the OWN bit (RDes0:31). This field stores the upper 32 bits of the timestamp.

27.1.6.3 Data cache alignment

Since both the transceiver buffer and the transceiver descriptor are called by the DMA controller and exist in the RAM space, and the DMA is 32-bit, it is necessary to ensure that their start addresses are aligned in 4 bytes when setting the transceiver descriptor queue, but setting the buffer does not require this forcibly.

In order to improve efficiency, it is most appropriate that an Ethernet frame is received by a buffer. For example, setting the buffer to 1518 bytes can accommodate the largest Ethernet ordinary frame, including source-end address field, length type field, Data padding field and CRC field. It should be noted that the maximum frame length of the tagged VLAN frame is 4 bytes longer than the general Ethernet frame, that is, 1522 bytes. If the user wants to achieve 4-byte alignment, a buffer can be 1524 bytes.

27.1.6.4 DMA transceiver configuration

The data received and sent is automatically dumped and pushed by the DMA, but if a fatal error is encountered, the DMA will stop and update the DMA status register. The user needs to manually start the DMA after initializing the DMA to continue running.

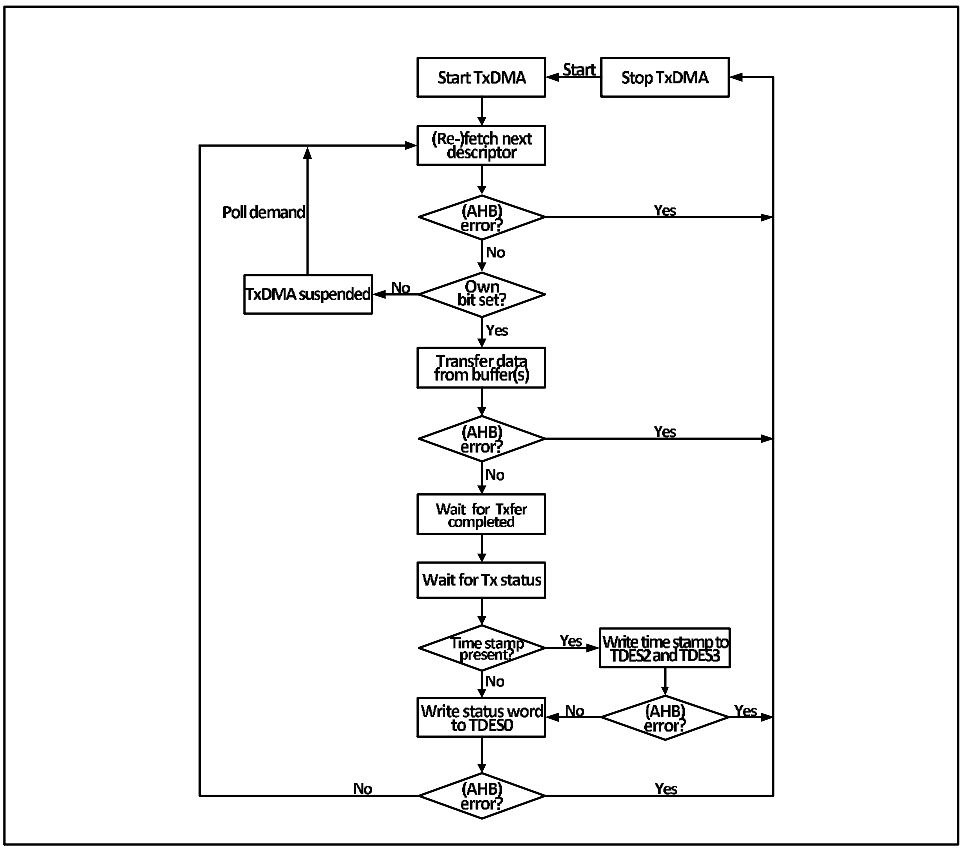
27.1.6.4.1 Transmit DMA configuration

The steps for the DMA controller to establish the operating mechanism are as follows:

1） Set up the transmit buffer queue and fill the transmit buffer with the content to be sent. Set up the transmit descriptor queue, fill in the fields of the transmit descriptor, set OWN, and deliver the management right of the transmit descriptor to the DMA controller; register the initial address of the descriptor in the DMATDLAR register;
2） Set the ST bit (DMAOMR: 13) to enable DMA;
3） In Run mode, the DMA descriptor will automatically read the content of the transmit descriptor, and push the data to the transmit FIFO according to the address and length indicated by it. After finishing the DMA, it will read the next transmit descriptor immediately following the chain structure for the next transmission . When the DMA controller detects that the OWM bit is not set, resulting in no access to the transmit descriptor or other normal errors, it will terminate the transfer and set the TBUS bit (DMASR: 2) or other bits (other than errors caused by OWN being 0) and the NIS bit (DMASR:16) is set.
4） A single frame is not allowed to span multiple descriptors; a frame must be clearly associated with a descriptor and a buffer.
5） If the MAC turns on the IEEE1588 PTP mode, after the DMA controller pushes the data to the FIFO, the MAC will write the transmit timestamp to TDes2 and TDes3, and reset the OWN bit.
6） After sending a frame, if the transmit descriptor enables the transmit completion interrupt (set TDes1:31), the DMA controller will set the transmit completion interrupt flag (DMASR:0), and then continue to fetch the next transmit descriptor.

The following figure shows the default process of transmission.

Figure 27-11 Process of transmission



27.1.6.4.2 Receive DMA configuration

The steps to establish a DMA receive flow mechanism are as follows:

1） Set the transmit buffer queue and descriptor queue, and set up the bits of each field of the receive descriptor; register the initial address of the descriptor in the DMARDLAR register; set the OWN bit to give the descriptor use permission to the DMA controller;

2） Set the SR bit to start the receiving process;

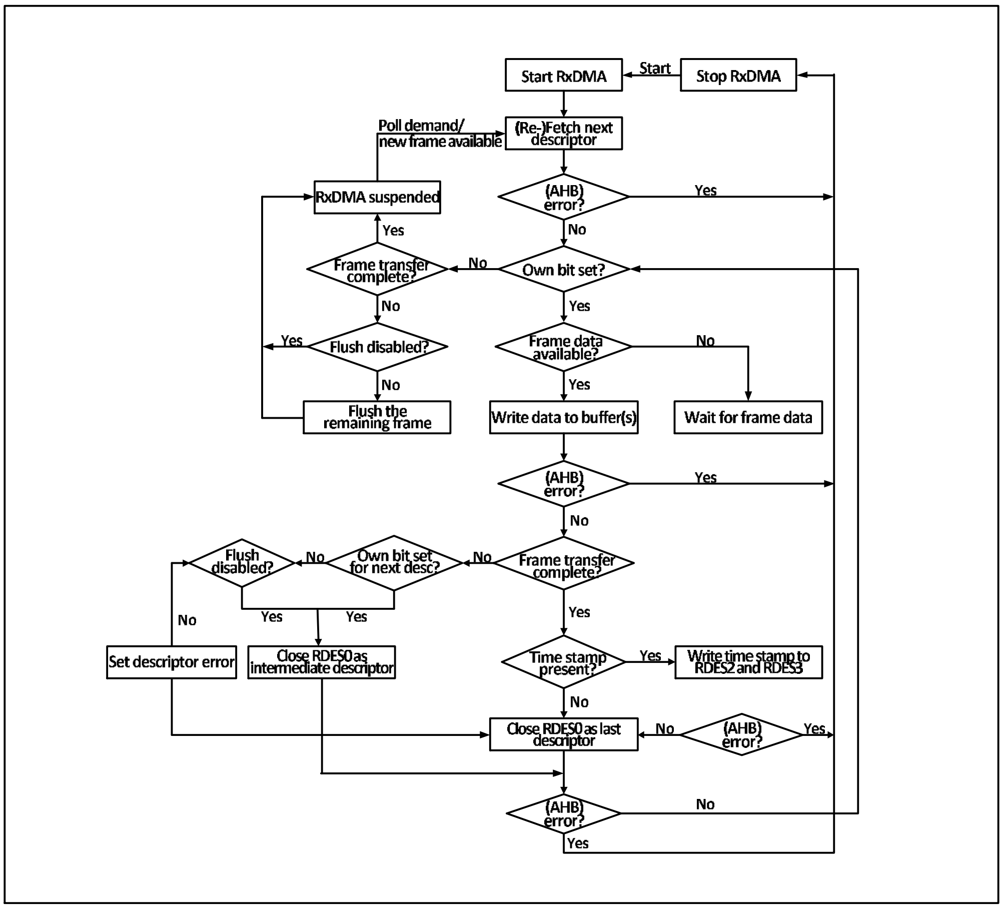
3） When the receive flow mechanism is running, the DMA controller obtains the next descriptor, checks the receiving descriptor configuration, and when the FIFO receives the next frame, performs multiple checks such as filtering and identifying the frame content, and forwards the frame data to the buffer specified by the symbol, writes the descriptor status field and obtains the next receiving descriptor, and reports the receiving completion interrupt. If the frame does not pass the filtering, it will be marked or discarded. If the frame has a checksum error, CRC error or the frame is too short, it will be marked. If the frame is too long, it may be cut off or receive a watchdog timeout error. The DMA controller will stop the receiving process when it encounters a fatal error such as the unavailability of the receive descriptor, and the user needs to pay special attention;

4） The user needs to enable at least one receive completion interrupt, and restore the used receive descriptor to the standby state in the Ethernet interrupt function to ensure that the receive process can run uninterrupted. The user can pass the address of the data buffer to be processed in the interrupt function, or handle some abnormal events that interrupt the receiving process.

5） If the PTP timestamp is enabled, the DMA controller dumps the data and writes the descriptor status field, and also writes the current timestamp into the last 2 words of the descriptor. The user should read the timestamp in time and complete the buffer address and the next descriptor address originally written in the last 2 words.

The following figure shows the default flow mechanism of reception:

Figure 27-12 Process of reception



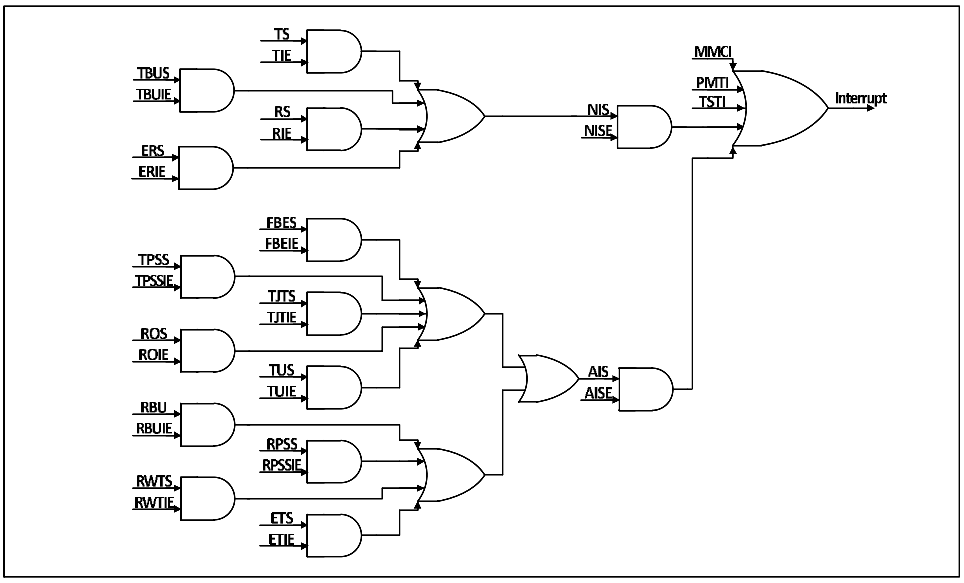
27.1.7 Interrupt

The Ethernet transceiver has 2 interrupt vectors, one is the Ethernet wake-up event and the other is the normal transceiver event. When a wake-up frame or magic frame is detected, an Ethernet wake-up event is triggered. Common Ethernet transceiver interrupt events include DMA interrupt and ETH interrupt.

27.1.7.1 DMA interrupt

DMA interrupts can be roughly divided into 2 groups, namely normal interrupts (NIS) and abnormal interrupts (AIS). Abnormal interrupts generally mean abnormal data transmission and reception, which requires special attention. When processing interrupts, users need to retrieve all interrupt flag bits and process all generated interrupt sources. The following figure is a schematic diagram of the interrupt of the Ethernet transceiver.

Figure 27-13 Interrupt schematic



The DMA interrupt is the most important interrupt in the Ethernet transceiver. Generally, user logic needs to rely on interrupts to receive frames, confirm that the frames are sent, or deal with the interrupted transceiver logic in time.

27.1.7.2 ETH interrupt

The ETH interrupt mainly includes the time alarm trigger of PTP, the sending and receiving count to a certain set value of the MMC register, and the PMT. The usefulness of ETH is mainly functional, not very complex and important relative to DMA interrupts. Users can use PTP interrupt for alarm clock, MMC interrupt for speed measurement, or PMT interrupt for remote wake-up. In addition, ETH also supports interrupts when the state of the built-in physical layer connection changes.

27.1.7.3 PMT interrupt

The reason why the PMT interrupt is raised separately and reiterated is that this interrupt has an independent interrupt vector and needs to be paid attention to when using it.

27.1.8 Register description

Table 27-22 Ethernet transceiver registers

MAC control register address mapping

Name	Offset Address	Description	Reset Value
R32_ETH_MACCR	0x40028000	MAC control register	0x00008000
R32_ETH_MACFFR	0x40028004	Frame filter register	0x00000000
R32_ETH_MACHTHR	0x40028008	Hash value list register high	0x00000000
R32_ETH_MACHTLR	0x4002800C	Hash value list register low	0x00000000
R32_ETH_MACMIIAR	0x40028010	MII address register	0x00000000
R32_ETH_MACMIIDR	0x40028014	MII data register	0x00000000
R32_ETH_MACFCR	0x40028018	MAC flow control register	0x00000000
R32_ETH_MACVLAN	0x4002801C	VLAN tag register	0x00000000
R32_ETH_MACRWUFFR	0x40028028	Wake-up frame filter register	0x00000000
R32_ETH_MACPMTCSR	0x4002802C	PMT control and status Register	0x00000000

R32_ETH_MACSR	0x40028038	MAC interrupt status register	0x00000000
R32_ETH_MACIMR	0x4002803C	MAC interrupt mask register	0x00000000
R32_ETH_MACA0HR	0x40028040	MAC address 0 register high	0x0010FFFF
R32_ETH_MACA0LR	0x40028044	MAC address 0 register low	0xFFFFFFFF
R32_ETH_MACA1HR	0x40028048	MAC address 1 register high	0x0000FFFF
R32_ETH_MACA1LR	0x4002804C	MAC address 1 register low	0xFFFFFFFF
R32_ETH_MACA2HR	0x40028050	MAC address 2 register high	0x0000FFFF
R32_ETH_MACA2LR	0x40028054	MAC address 2 register low	0xFFFFFFFF
R32_ETH_MACA3HR	0x40028058	MAC address 3 register high	0x0000FFFF
R32_ETH_MACA3LR	0x4002805C	MAC address 3 register low	0xFFFFFFFF

MMC control register address mapping, note: the addresses are not consecutive

Name	Address	Description	Reset Value
R32_ETH_MMCCR	0x40028100	MMC control register	0x00000000
R32_ETH_MMCRIR	0x40028104	MMC receive register	0x00000000
R32_ETH_MMCTIR	0x40028108	MMC transmit interrupt register	0x00000000
R32_ETH_MMCRIMR	0x4002810C	MMC receive interrupt mask register	0x00000000
R32_ETH_MMCTIMR	0x40028110	MMC transmit interrupt mask register	0x00000000
R32_ETH_MMCTGFSCCR	0x4002814C	MMC transmit good frame counter after single collision	0x00000000
R32_ETH_MMCTGFMSCCR	0x40028150	MMC transmit good frame counter after multiple collisions	0x00000000
R32_ETH_MMCTGFCR	0x40028168	MMC transmit good frame count register	0x00000000
R32_ETH_MMCRFCECR	0x40028194	MMC receive frame with CRC error count register	0x00000000
R32_ETH_MMCRFAECR	0x40028198	MMC receive frame with alignment error count register	0x00000000
R32_ETH_MMCRGUFCR	0x400281C4	MMC receive good unicast frame count register	0x00000000

IEEE1588 (PTP) register address mapping

Name	Address	Description	Reset Value
R32_ETH_PTPTSCR	0x40028700	PTP timestamp control register	0x00000000
R32_ETH_PTPSSIR	0x40028704	PTP subsecond increment register	0x00000000
R32_ETH_PTPTSHR	0x40028708	PTP timestamp register high	0x00000000
R32_ETH_PTPTSLR	0x4002870C	PTP timestamp register low	0x00000000
R32_ETH_PTPTSHUR	0x40028710	PTP timestamp update register high	0x00000000
R32_ETH_PTPTSLUR	0x40028714	PTP timestamp update register low	0x00000000
R32_ETH_PTPTSAR	0x40028718	PTP timestamp adder register	0x00000000
R32_ETH_PTPTTHR	0x4002871C	PTP target register high	0x00000000
R32_ETH_PTPTTLR	0x40028720	PTP target register low	0x00000000

DMA register address mapping. Note: The addresses are not consecutive

Name	Address	Description	Reset Value
R32_ETH_DMABMR	0x40029000	DMA bus mode register	0x00002101
R32_ETH_DMATPDR	0x40029004	DMA transmit query register	0x00000000
R32_ETH_DMARPTR	0x40029008	DMA receive query register	0x00000000
R32_ETH_DMARDLAR	0x4002900C	DMA receive descriptor address register	0x00000000
R32_ETH_DMATDLAR	0x40029010	DMA transmit descriptor address register	0x00000000
R32_ETH_DMASR	0x40029014	DMA status register	0x00000000
R32_ETH_DMAOMR	0x40029018	DMA operation mode register	0x00000000
R32_ETH_DMAIER	0x4002901C	DMA interrupt enable register	0x00000000
R32_ETH_DMAMFBOCR	0x40029020	DMA lost frame register	0x00000000
R32_ETH_DMACHTDR	0x40029048	DMA current transmit descriptor register	0x00000000
R32_ETH_DMACHRDR	0x4002904C	DMA current receive descriptor register	0x00000000
R32_ETH_DMACHTBAR	0x40029050	DMA current transmit buffer register	0x00000000
R32_ETH_DMACHRBAR	0x40029054	DMA current receive buffer register	0x00000000

Internal 10M physical layer register address

Name	Offset Address	Description	Reset Value
BMCR	0x00	Basic control register	0x2100
BMSR	0x01	Basic status register	0x1809
PHY_SR	0x10	Physical layer status register	0x0000
PHY_MDIX	0x1E	Auto-flip register	0x0000

Note: the offset address of the internal physical layer registers is used in the SMI interface

27.1.8.1 MAC control registers

27.1.8.1.1 MAC Control Register (R32_ETH_MACCR)

Offset Address: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TCD			Reserved					WD	JD	PI	PR	IFG			CSD

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FES		ROD	LM	DM	IPCO	RD	Res	APCS	BL		DC	TE	RE	TCF	Res

Name	Name	Access	Description	Reset Value
[31:29]	TCD	RW	Transmit clock delay field, this field is used to delay the transmit clock. The MAC controller delays the	000b

			output of the transmit clock selected by the CES bit (ETH_MACCR[1]). Delay time calculation formula: Tdalay=TCD(ETH_MACCR[31:29])*0.5ns.	
[28:24]	Reserved	RO	Reserved	
23	WD	RW	Watchdog setting. 0: The MAC opens the watchdog, and can only accept Ethernet frames with a maximum size of 2048 bytes, and the super-long part will be cut off; 1: MAC closes the door dog, and can receive Ethernet frames with a maximum size of 16384 bytes.	0b
22	JD	RW	Jabber setting. 0: If the user tries to send an Ethernet frame with a length of more than 2048 bytes, the MAC will close the transmitter; 1: The MAC closes the Jabber timer, and can send up to 16384 bytes of Ethernet frames.	0b
21	PI	RW	Built-in 10MPHY transmit drive bias current setting bits. 0: Rated drive; 1: Energy-saving sending.	0b
20	PR	RW	Built-in 10MPHY on-chip 50-ohm resistor pull-up to open the setting bit. 0: On-chip 50-ohm resistor disconnected; 1: On-chip 50-ohm resistor connected.c	0b

[19:17]	IFG	RW	Frame gap setting, used to set the shortest time gap between sending 2 frames.
			000: 96-bit time;
			001: 88-bit time;
			010: 80-bit time;
			011: 72-bit time;
			100: 64-bit time;
			101: 56-bit time;
			110: 48-bit time;
			111: 40-bit time.
				000b

16	Reserved	RO	Reserved	0b

[15:14]	FES	RW	Ethernet speed setting.
			00: 10Mbit/s;
			01: 100Mbit/s;
			10: 1Gbit/s;
			11: Reserved, unused.
				00b

13	Reserved	RO	Reserved	0b
12	LM	RW	Self-loop mode enable. Setting this bit enables self-loop mode.	0b
11	DM	RW	Duplex mode enable. Setting this bit enables full duplex mode.	0b
10	IPCO	RW	IPv4 checksum check enable.	0b

		RW	0: Disable the checksum verification function of IPv4 at the receiving end, and the corresponding PCE and PHCE flags are always 0. See the definition of each bit of the receive descriptor; 1: Enable IPv4 checksum verification, the MAC controller will check the TCP, UDP and ICMP headers.	
[9:8]	Reserved	RO	Unused.	
7	APCS	RW	Padding & CRC auto-stripping enable. 0: MAC does not change the content of the frame; 1: When receiving an Ethernet frame with a length less than or equal to 1500 bytes, the MAC automatically removes the padding bytes and CRC fields of the frame; in an Ethernet frame greater than 1500 bytes, the MAC does not change.	0b
[6:4]	Reserved	RO	Unused	
3	TE	RW	Transmit enable. 0: Disable the transmitter, and no frame will be transmitted after the MAC finishes sending the current frame; 1: Enable the MAC transmitter.	0b
2	RE	RW	Receive enable. 0: Disable the receiver, and no frame will be received after the MAC finishes receiving the current frame; 1: Enable the MAC receiver.	0b
1	TCF	RW	Transmit clock toggle setting. 0: TXC selected by TCES is directly used as the GTX_CLK output by the chip; 1: Use the inversion of TXC selected by TCES as the GTX_CLK output by the chip.	0b
0	Res	R0	Unused.	0b

27.1.8.1.2 MAC Frame Filter Register (R32_ETH_MACFFR)

Offset Address: 0x04

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					HPF	SAF	SAIF	PCF		BFD	PAM	DAIF	HM	HU	PM

Bit	Name	Access	Description	Reset Value
31	RA	RW	Receive all 0: MAC only forwards frames that have passed the filter to the receive queue;	0b

			1: The MAC forwards all received frames to the receive queue, regardless of whether they pass the filter or not.	
[30:11]	Reserved	RO	Reserved	
10	HPF	RW	HASH filter or perfect filter 0: On the premise that HM or HU is set, as long as it meets the HASH filter, it can pass the address filter; 1: Determine which filtering method is used in unicast/multicast mode according to the value of HM or HU.	0b
9	SAF	RW	Source address filter 0: MAC marks frames that do not pass the source MAC address filtering; 1: MAC will directly discard frames that do not pass the source MAC address filtering.	0b
8	SAIF	RW	Source address filter result inverted. 0: If the source address of the received frame is inconsistent with the source address enabled in the MAC address register, it is considered to have failed the source address filter; 1: If the source address of the received frame is consistent with the source address enabled in the MAC address register, it is considered to have failed the source address filter.	0b
[7:6]	PCF	RW	Flow control frames pass control. 00/01: The MAC does not forward any flow control frames to the application; 10: MAC forwards all flow control frames to the application, including flow control frames that fail the address filter; 11: MAC only forwards flow control frames that pass the address filter.	0
5	BFD	RW	Broadcast frame reception control 0: Receive all broadcast frames; 1: Discard all broadcast frames.	0b
4	PAM	RW	Pass all multicast frame control. 0: Whether the multicast frame can pass filtering depends on the value of HM; 1: All multicast frames can be filtered by address.	0b
3	DAIF	RW	Destination MAC address filter results inverted control. 0: The filter result takes effect normally; 1: For multicast frames and unicast frames, whether the result of the filter passing is reversed before it takes effect.	0b
2	HM	RW	Multicast frame filter mode 0: Perfect address filtering; 1: HSAH address filtering.	0b

1	HU	RW	Unicast frame filter mode 0: Perfect address filtering; 1: HSAH address filtering.	0b
0	PM	RW	Promiscuous mode enable All frames pass the address filter without marking the result of the filter.	0b

27.1.8.1.3 Hash List Register (R32_ETH_MACHTHR, R32_ETH_MACHTLR)

Offset Address: 0x08

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

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

Offset Address: 0x0C

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

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

Bit	Name	Access	Description	Reset Value
[31:0]	HTH	RW	Hash list high 32 bits	00000000h
[31:0]	HTL	RW	Hash list low 32 bits	00000000h

27.1.8.1.4 MII Address Register (R32_ETH_MACMIIAR)

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
PA					MR					Res	CR			MW	MB

Bit	Name	Access	Description	Reset Value
[31:16]	Reserved	RO	Unused.	0
[15:11]	PA	RW	Physical layer address. The user writes the physical layer address to be operated into this field.	0
[10:6]	MR	RW	Physical layer register address field. The user writes the register address to be operated into this field.	0
5	Reserved	RO	Unused.	0
[4:2]	CR	RW	Clock range setting. Users, please keep it as 000b, so that the MII frequency is divided by 42 of the main frequency.	0

1	MW	RW	Read and write setting. 0: Read the physical layer; 1: Write to the physical layer.	0
0	MB	W1	MII busy flag. This bit is set by the user, indicating that the hardware is instructed to start reading or writing. The physical address, register address and data field should remain unchanged during reading and writing. After the hardware clears this bit, the operation is completed.	0

27.1.8.1.5 MII Data Register (R32_ETH_MACMIIDR)

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
MD															

Bit	Name	Access	Description	Reset Value
[31:16]	Reserved	RO	Reserved.	0
[15:0]	MD	RW	MII operation data. This field is used to store the data to be read from the physical layer through the MII interface, or to store the data written to the physical layer.	0

27.1.8.1.6 MAC Flow Control Register (R32_ETH_MACFCR)

Offset address: 0x18

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								ZQPD	Res	PLT		UPFD	RFCE	TFCE	FCB

Bit	Name	Access	Description	Reset Value
[31:16]	PT	RW	Pause interval. This field is used to control the value of the Pause time field, and the unit is the time it takes for the current MII interface to send 64 bytes.	0
[15:8]	Reserved	RO	Unused.	0
7	ZQPD	RW	Zero value Pause function disable. When this bit is set, the generation of automatic zero-value Pause control frames is disabled.	0
6	Reserved	RO	Reserved	0

[5:4]	PLT	RW	Automatically resend Pause frame threshold. The value of this field should be less than the value of PT. The retransmission time is equal to the time represented by the PT minus the PLT.
			00: 4 time slots;
			01: 28 time slots;
			10: 144 time slots;
			11: 256 time slots.
				00b

3	UPFD	RW	Unicast Pause frame detection. 0: The MAC only receives Pause frames with the unique address defined by the protocol specification; 1: The MAC also detects whether the Pause frame is a unicast address defined in the MAC address register 0.	0
2	RFCE	RW	Receive flow control enable. 0: The MAC does not parse the Pause frame; 1: The MAC parses the Pause frame and turns off the transmitter for a while.	0
1	TFCE	RW	Transmit flow control enable. 0: MAC closes sending flow control and does not send Pause frames; 1: MAC enables sending flow control and can send Pause frames.	0
0	FCB	W1	Flow control busy flag. This bit can be set to send a Pause frame, which is cleared by hardware after the transmission is completed. When operating on the entire MACFCR register, it is necessary to ensure that the FCB bit is 0.	0

27.1.8.1.7 VLAN Tag Register (MACVLAN)

Offset address: 0x1C

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

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

Bit	Name	Access	Description	Reset Value
[31:17]	Reserved	RO	Reserved	0
16	VLANT	WR	Tag compare control. 0: Compare all 16-bit data of the 15th and 16th bytes of the VLAN frame with the VLANTI field; 1: Only use the [11:0] bits of the 15th and 16th bytes of the VLAN frame and compare them with the corresponding bits of the VLANTI field.	0

[15:0]	VLANTI	WR	Label vs. sample domain. According to the IEEE 802.1 protocol, bits [15:13] of the VLAN frame are the user priority, [12] is the canonical format indicator, and bits [11:0] of the VLAN identifier field. If VLANTI is all 0, then the MAC will no longer care about the 15th and 16th bytes of the VLAN frame, and when the 13th and 14th bytes are 0x8100 (pay attention to the size end), it will be judged as a VLAN frame.	0

27.1.8.1.8 Wake-up Frame Filter Register (R32_MACRWUFFR)

Offset Address: 0x28

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

Bit	Name	Access	Description	Reset Value
[31:0]	RWUFFR	RW	The wake-up frame filter register is actually 8 different registers, and 8 consecutive read operations can read out all registers, and 8 consecutive write operations can be written into all 8 registers. The description of each bit of these 8 registers is described in Section 14.5.6.3.	0

27.1.8.1.9 PMT Control and Status Register (R32_ETH_MACPMTCSR)

Offset address: 0x2C

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

Bit	Name	Access	Description	Reset Value
31	WFFRP	W	Register reset control. Setting this bit clears all PMTCSR registers to zero. This bit is automatically cleared in single system cycle. Register reset control bits. Setting this bit clears all PMTCSR registers to zero. This bit is automatically cleared in single system cycle.	0b
[30:10]	Reserved	RO	Reserved.	
9	GU	RW	Global unicast. Setting this bit causes the MAC to consider all unicast frames that pass the filter to be wake-up frames.	0b

[8:7]	Reserved	RO	Reserved.	
6	WFR	RC	Wake-up frame received flag. This bit is set when a wake-up frame is received. Read it and it will be cleared automatically.	0b
5	MPR	RC	Magic frame received flag. This bit is set when a magic frame is received. Read it and it will be cleared automatically.	0b
[4:3]	Reserved	RO	Reserved.	
2	WFE	RW	Wake-up frame enable. Setting this bit allows a PMT event to be generated when a wake-up frame is received.	0b
1	MPE	RW	Magic frame enable. Setting this bit allows a PMT event to be generated when a magic frame is received.	0b
0	PD	W	Power-down control. Setting this bit will put the MAC into power-down mode: all other frames are discarded until it receives a wake-up frame or magic frame, and the MAC is automatically cleared after wake-up. Before setting this bit, WFE or MPE needs to be set.	0b

27.1.8.1.10 MAC Interrupt Status Register (R32_ETH_MACSR)

Offset address: 0x38

31	30	29	28	27	26	25	24	23	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						TSTS	Reserved		MMCTS	MMCRS	MMCS	PMTS	Reserved		

Bit	Name	Access	Description	Reset Value
[31:10]	Reserved	RO	Reserved	0
9	TSTS	RC	Timestamp trigger interrupt flag. When the time set by the PTP system clock is reached, this flag will be set.	0
[8:7]	Reserved	RO	Reserved	0
6	MMCTS	R	MMC transmit interrupt flag. This bit is set when any interrupt is generated in the MMCTIR register in the MMC register bank. This bit is also cleared when all the MMCTIR registers in the MMC register bank are cleared.	0
5	MMCRS	R	MMC receive interrupt flag. This bit is set when any interrupt is generated in the MMCRIR register in the MMC register bank. This bit is also cleared when the MMCRIR registers in the MMC register bank are all cleared.	0
4	MMCS	R	MMC status flag. When MMCTS or MMCRS is set, this bit is triggered. When MMCTS and MMCRS are all cleared, this bit is cleared.	0

3	PMTS	R	PMT status flag. In power-down mode, if the MAC receives a magic frame or wake-up frame wakes up the MAC, this bit will be set. After clearing the WFR and MPR bits, this bit is cleared.	0
[2:0]	Reserved	RO	Reserved	0

27.1.8.1.11 MAC Interrupt Mask Register (R32_ETH_MACIMR)

Offset Address: 0x3C

31	30	29	28	27	26	25	24	23	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						TSTIM	Reserved					PMTIM	Reserved		

Bit	Name	Access	Description	Reset Value
[31:10]	Reserved	RO	Reserved	0
9	TSTIM	RW	Timestamp interrupt mask. Setting this bit disables generation of timestamp interrupts.	0
[8:4]	Reserved	RO	Reserved	0
3	PMTIM	RW	PMT interrupt mask. Setting this bit will mask the PMT interrupt.	0
[2:0]	Reserved	RO	Reserved	0

27.1.8.1.12 MAC Address Register 0 High 32 Bits (R32_ETH_MACA0HR)

Offset Address: 0x40

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACA0H															

Bit	Name	Access	Description	Reset Value
31	MO	RO	Always 1.	1
[30:16]	Reserved	RO	Reserved	0
[15:0]	MACA0H	RW	High 16 bits of the MAC address, that is, 47:32 bits.	FFFFh

27.1.8.1.13 MAC Address Register 0 Low 32 Bits (R32_ETH_MACA0LR)

Offset Address: 0x44

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MACA0L															

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

MACA0L

Bit	Name	Access	Description	Reset Value
[31:0]	MACA0L	RW	Low 32 bits of the MAC address, namely 31:0 bits. In general, the address of the MAC address register 0 is the address of the MAC itself.	FFFFFFFFh

27.1.8.1.14 MAC Address Register 1 High 32 Bits (R32_ETH_MACA1HR)

Offset address: 0x48

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AE	SA	MBC						Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MACA1H															

Bit	Name	Access	Description	Reset Value
31	AE	RW	Address filter enable. 0: The address filter ignores MAC address 1. 1: The address filter uses MAC address 1 for perfect filtering;	0
30	SA	RW	Address role selection. 0: MAC address 1 is used to compare with the destination address of the received frame; 1: MAC address 1 is used to compare with the source address of the received frame.	0
[29:24]	MBC	RW	Mask word control. Each bit of the MBC is used to correspond to a certain byte of the masked MAC address 1, and is used to prohibit a certain byte of the MAC address 1 from participating in the comparison of the source address or the destination address of the received frame. R32_ETH_MACA1HR[29] is set to mask MACA1H[15:8]; R32_ETH_MACA1HR[28] is set to mask MACA1H[7:0]; R32_ETH_MACA1HR[27] is set to mask MACA1L[31:24]; R32_ETH_MACA1HR[26] is set to mask MACA1L[23:16]; R32_ETH_MACA1HR[25] is set to mask MACA1L[15:8]; R32_ETH_MACA1HR[24] is set to mask MACA1L[7:0].	0
[23:16]	Reserved	RO	Reserved	0

[15:0]	MACA1H	RW	High 16 bits of the MAC address, that is, 47:32 bits.	FFFFh

27.1.8.1.15 MAC Address Register 1 Low 32 Bits (R32_ETH_MACA1LR)

Offset Address: 0x4C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MACA1L															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MACA1L															

Bit	Name	Access	Description	Reset Value
[31:0]	MACA1L	RW	Low 32 bits of the MAC address, namely 31:0 bits.	FFFFFFFFh

27.1.8.1.16 MAC Address Register 2 High 32 Bits (R32_ETH_MACA2HR)

Offset Address: 0x50

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AE	SA	MBC						Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MACA2H															

Bit	Name	Access	Description	Reset value
31	AE	RW	Address filter enable. 0: The address filter ignores the MAC address 2. 1: The address filter uses MAC address 2 for perfect filtering.	0
30	SA	RW	Address role selection. 0: MAC address 2 is used to compare with the destination address of the received frame; 1: MAC address 2 is used to compare with the source address of the received frame.	0
[29:24]	MBC	RW	Mask word control. Each bit of the MBC is used to correspond to a certain byte of the masked MAC address 2, and is used to prohibit a certain byte of the MAC address 2 from participating in the comparison of the source address or the destination address of the received frame. R32_ETH_MACA2HR[29] is set to mask MACA2H[15:8]; R32_ETH_MACA2HR[28] is set to mask MACA2H[7:0]; R32_ETH_MACA2HR[27] is set to mask	0

			MACA2L[31:24]; R32_ETH_MACA2HR[26] is set to mask MACA2L[23:16]; R32_ETH_MACA2HR[25] is set to mask MACA2L[15:8]; R32_ETH_MACA2HR[24] is set to mask MACA2L[7:0].	
[23:16]	Reserved	RO	Reserved	0
[15:0]	MACA2H	RW	High 16 bits of the MAC address, that is, 47:32 bits.	FFFFh

27.1.8.1.17 MAC Address Register 2 Low 32 Bits (R32_ETH_MACA2LR)

Offset address: 0x54

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MACA2L															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACA2L															

Bit	Name	Access	Description	Value
31:0	MACA2L	RW	Low 32 bits of the MAC address, that is, 31:0 bits.	FFFFFFFFh

27.1.8.1.18 MAC Address Register 3 High 32 Bits (R32_ETH_MACA3HR)

Offset address: 0x58

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACA3H															

Bit	Name	Access	Description	Reset Value
31	AE	RW	Address filter enable. 0: The address filter ignores the MAC address 3; 1: The address filter uses MAC address 3 for perfect filtering.	0b
30	SA	RW	Address role selection. 0: MAC address 3 is used to compare with the destination address of the received frame; 1: MAC address 3 is used to compare with the source address of the received frame.	0b
[29:24]	MBC	RW	Mask word control. Each bit of the MBC is used to correspond to a certain byte of the masked MAC address 3, and is used to prohibit a certain byte of the	000000b

			MAC address 3 from participating in the comparison of the source address or the destination address of the received frame. R32_ETH_MACA3HR[29] is set to mask MACA3H[15:8]; R32_ETH_MACA3HR[28] is set to mask MACA3H[7:0]; R32_ETH_MACA3HR[27] is set to mask MACA2L[31:24]; R32_ETH_MACA3HR[26] is set to mask MACA2L[23:16]; R32_ETH_MACA3HR[25] is set to mask MACA3L[15:8]; R32_ETH_MACA3HR[24] is set to mask MACA3L[7:0].	
[23:16]	Reserved	RO	Reserved	
15:0	MACA3H	RW	High 16 bits of the MAC address, that is, 47:32 bits.	FFFFFFFFh

27.1.8.1.19 MAC Address Register 3 Low 32 Bits (R32_ETH_MACA3LR)

Offset address: 0x5C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MACA3L															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MACA3L															

Bit	Name	Access	Description	Reset Value
[31:0]	MACA3L	RW	Low 32 bits of the MAC address, that is, 31:0 bits.	FFFFFFFFh

27.1.8.2 MMC registers

27.1.8.2.1 MMC Control Register (R32_ETH_MMCCR)

Offset address: 0x0100

31	30	29	28	27	26	25	24	23	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												MCF	ROR	CSR	CR

Bit	Name	Access	Description	Reset Value
[31:4]	Reserved	RO	Reserved	0

3	MCF	RW	Counter freeze control. Setting this bit will freeze all MMC counter values. Resetting this bit will restore the count of each counter. Setting ROR while frozen and then reading any counter will cause that counter to be cleared.	0
2	ROR	RW	Reset control on read. Setting this bit will clear the value of the counter after reading either counter.	0
1	CSR	RW	Counter roll stop. Setting this bit causes the counter to increment to its maximum value and stop without auto-zeroing.	0
0	CR	W	Counter reset control. Setting this bit will reset all counters. This bit is automatically cleared after single system cycle.	0

27.1.8.2.2 MMC Receive Interrupt Register (R32_ETH_MMRIR)

Offset address: 0x0104

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

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

Bit	Name	Access	Name	Reset Value
[31:18]	Reserved	RO	Reserved	0
17	RGUFS	RC	This bit is set when more than half of the good frames are received.	0
[16:6]	Reserved	RO	Reserved	0
5	RFCES	RC	This bit is set when more than half of the frames with CRC errors are received.	0
4:0	Reserved	RO	Reserved	0

27.1.8.2.3 MMC Transmit Interrupt Register (R32_ETH_MMCTIR)

Offset address: 0x0108

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

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

Bit	Name	Access	Description	Reset Value
[31:22]	Reserved	RO	Reserved	0
21	TGFS	RC	This bit is set when more than half of the frames are sent.	0

[20:0]	Reserved	RO	Reserved	0

27.1.8.2.4 MMC Receive Interrupt Mask Register (R32_ETH_MMRIMR)

Offset address: 0x010C

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

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

Bit	Name	Access	Description	Reset Value
[31:18]	Reserved	RO	Reserved	0
17	FGUFM	RW	Receive a good frame half interrupt mask. Setting this bit will mask the interrupt generated when the received frame counter reaches half the value.	0
[16:6]	Reserved	RO	Reserved	0
5	FRCRM	RW	Receive CRC error frame half interrupt mask. Setting this bit will mask the interrupt that occurs when the frame counter value reaches half the value of the frame counter that receives a CRC error.	0
[4:0]	Reserved	RO	Reserved	0

27.1.8.2.5 MMC Transmit Interrupt Mask Register (R32_ETH_MMCTIMR)

Offset address: 0x0110

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

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

Bit	Name	Access	Description	Reset Value
[31:22]	Reserved	RO	Reserved	0
21	TGFM	RW	Transmit good frame half interrupt mask. Setting this bit will mask the interrupt generated when the sent frame counter reaches half the value.	0
[20:0]	Reserved	RO	Reserved	0

27.1.8.2.6 MMC Good Frame Counter After Single Collision (R32_ETH_MMCTGFSCCR)

Offset address: 0x014C

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

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

Bit	Name	Access	Description	Reset Value
[31:0]	TGFSCCR	RO	This field is used to count the counters of only one collision when a frame occurs in half-duplex mode.	0

27.1.8.2.7 MMC Good Frame Counter after Multiple Collisions (R32_ETH_MMCTGFMSCCR)

Offset address: 0x0150

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

Bit	Name	Access	Description	Reset Value
[31:0]	TGFMSCCR	RO	This field is used to count the counters that only encounter more than one collision when a frame occurs in half-duplex mode.	0

27.1.8.2.8 MMC Transmit Good Frame Count Register (R32_ETH_MMCTGFCR)

Offset address: 0x0168

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

Bit	Name	Access	Description	Reset Value
[31:0]	TGFC	RO	Count of correct frames sent by the MAC.	0

27.1.8.2.9 MMC Receive Frame with CRC Error Count Register (R32_ETH_MMCRFCECR)

Offset address: 0x0194

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

Bit	Name	Access	Description	Reset Value

[31:0]	RFCECR	RO	Counter of frames received by the MAC with CRC errors.	0

27.1.8.2.10 MMC Receive Frame with Alignment Error Count Register (R32_ETH_MMCRFAECR)

Offset address: 0x0198

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

Bit	Name	Access	Description	Reset Value
[31:0]	RFAECR	RO	Counter of frames received by the MAC with alignment errors.	0

27.1.8.2.11 MMC Receive Good Frame Count Register (R32_ETH_MMCRGUFCR)

Offset address: 0x01C4

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

Bit	Name	Access	Description	Reset Value
[31:0]	RGUFCR	RO	The number of normal frames received by the MAC.	0

27.1.8.3 IEEE 1588 (PTP) registers

27.1.8.3.1 PTP Timestamp Control Register (R32_ETH_PTPTSCR)

Offset address: 0x0700

31	30	29	28	27	26	25	24	23	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										TSARU	TSITE	TSSTU	TSSTI	TSFCU	TSE

Bit	Name	Access	Description	Reset Value
[31:6]	Reserved	RO	Reserved	0
5	TSARU	RW	Addend register update control. When this bit is set, the value of the addend register will be added to the accumulator. This bit is used in fine tuning mode. This bit is automatically cleared after the accumulator is incremented. This bit can only be set when it is 0.	0

4	TSITE	RW	Timestamp interrupt trigger enable. When this bit is set, an interrupt will be generated when the PTP system time reaches the value set in the target time register.	0
3	TSSTU	RW	System time update control. When this bit is set, the PTP system time will be added to the value in the update register. This bit is automatically cleared after the update is complete.	0
2	TSSTI	RW	Timestamp initialization control. When this bit is set, the system time of the PTP will be replaced with the value in the update register. This bit is automatically cleared after the update is complete.	0
1	TSFCU	RW	Update mode selection. 0: Coarse mode; 1: Fine adjustment mode.	0
0	TSE	RW	Additional timestamp enable. 0: No timestamp is added to the descriptor; 1: Add a timestamp to the descriptor at the end of reception or transmission.	0

27.1.8.3.2 PTP Subsecond Increment Register (R32_ETH_PTPSSIR)

Offset address: 0x0704

31	30	29	28	27	26	25	24	23	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								STSSI							

Bit	Name	Access	Description	Reset Value
[31:8]	Reserved	RO	Reserved	0
[7:0]	STSSI	RW	Sub-second step value. In coarse mode, the PTP system time will automatically increase by this value every main frequency cycle. In the fine adjustment mode, when the accumulator overflows, the PTP system time will automatically increase by this value.	00h

27.1.8.3.3 PTP Timestamp Register High (R32_ETH_PTPTSHR)

Offset address: 0x0708

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

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

Bit	Name	Access	Description	Reset Value
[31:0]	TSHR	RO	PTP system time value, real-time value, in seconds.	0

27.1.8.3.4 PTP Timestamp Register Low (R32_ETH_PTPTSLR)

Offset address: 0x070C

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

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

Bit	Name	Access	Description	Reset Value
31	STPNS	RO	System time positive/negative flag. 1: Negative; 0: Positive.	0
[30:0]	STSS	RO	PTP system time value, real-time value, in sub-seconds, i.e., about 0.46ns. When STSS overflows, STS increments by 1s.	0

27.1.8.3.5 PTP Timestamp Update Register High 32 Bits (R32_ETH_PTPTSHUR)

Offset address: 0x0710

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

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

Bit	Name	Access	Description	Reset Value
[31:0]	TSUS	RW	Timestamp update second value, which is used to replace the high bit of the PTP system time or the second value representing its addition or subtraction to the system time.	0

27.1.8.3.6 PTP Timestamp Update Register Low 32 Bits (R32_ETH_PTPTSLUR)

Offset address: 0x0714

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

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

Bit	Name	Access	Description	Reset Value
31	TSUPNS	RW	Timestamp update positive/negative flag. When the update time method is to use TSUS and TSUSS to directly replace the PTP system time, TSUPNS should be 0; when the update time method is to use TSUS and TSUSS to add or subtract the original PTP system time value, TSUPNS is 1, indicating that the PTP system Subtract TSUS and TSUSS from the time base, and 0 means add TSUS and TSUSS to the PTP system time.	0
[30:0]	TSUSS	RW	Timestamp update sub-second value, which is used to replace the high bit of the PTP system time or a sub-second value that represents its addition or subtraction from the system time.	0

27.1.8.3.7 PTP Timestamp Adder Register (R32_ETH_PTPTSAR)

Offset address: 0x0718

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

Bit	Name	Access	Description	Reset Value
[31:0]	TSA	RW	Timestamp adder value. This register is only used in fine tuning mode. The value of TSA is added to the accumulator every system cycle, and if the accumulator overflows, it will trigger a system time update in fine tuning mode.	0

27.1.8.3.8 PTP Target Time Register High 32 Bits (R32_ETH_PTPTTHR)

Offset address: 0x071C

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

Bit	Name	Access	Description	Reset Value
[31:0]	TTSH	RW	Target time in seconds. If the PTP system time reaches or exceeds this value and the associated interrupt is enabled, an interrupt will be generated.	0

27.1.8.3.9 PTP Target Time Register High 32 Bits (R32_ETH_PTPTTHR)

Offset address: 0x0720

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

Bit	Name	Access	Description	Reset Value
[31:0]	TTSL	RW	Target time sub-second value. An interrupt is generated if the PTP system time reaches or exceeds this value and the associated interrupt is enabled.	0

27.1.8.4 DMA control registers

27.1.8.4.1 DMA Bus Mode Register (R32_ETH_DMATPDR)

Offset address: 0x1000

31	30	29	28	27	26	25	24	23	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									DSL					Reserved	SR

Bit	Name	Access	Description	Reset Value
[31:7]	Reserved	RO	Reserved	0
[6:2]	DSL	RW	Descriptor hop length These bits define the jump distance in words (32 bits) between 2 descriptors that are not connected in a chained structure. The address hop is the address difference from the end of the current descriptor to the beginning of the next descriptor. When the DSL field is 0, the DMA considers the descriptors to be contiguous under a ring structure.	0
1	Reserved	RO	Reserved	0
0	SR	RW	Software Reset When set to 1, the MAC's DMA controller resets the internal registers and logic circuits of all subsystems of the MAC. This bit is automatically cleared after the reset operation is completed for the different clock domain modules inside the MAC. Before rewriting the MAC's registers, you should ensure that this bit is 0.	1

27.1.8.4.2 DMA Transmit Query Register (R32_ETH_DMATPDR)

Offset address: 0x1004

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

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

Bit	Name	Access	Description	Reset Value
[31:0]	TPDR	RW	Send query commands. The user initiates a suspended transmit process by writing any value to this register. After restarting the transmission process, this register will be automatically cleared.	0

27.1.8.4.3 DMA Receive Query Register (R32_ETH_DMARPDR)

Offset address: 0x1008

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

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

Bit	Name	Access	Description	Reset Value
[31:0]	RPDR	RW	Receive query commands. The receiving process may be interrupted by various accidents, and the user needs to write any value to this register to restart the receiving process. After restarting the receive process, this register will be automatically cleared.	0

27.1.8.4.4 DMA Receive Descriptor Address Register (R32_ETH_DMARDLAR)

Offset address: 0x100C

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

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

Bit	Name	Access	Description	Reset Value
[31:0]	RDLAR	RW	This register is used to store the address of the first received DMA descriptor. Note that descriptors require 16-byte alignment, so the last 4 bits of this register should be 0.	0

27.1.8.4.5 DMA Transmit Descriptor Address Register (R32_ETH_DMATDLAR)

Offset address: 0x1010

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

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

Bit	Name	Access	Description	Reset Value
[31:0]	TDLAR	RW	This register is used to store the address of the first transmit DMA descriptor. Note that descriptors require 16-byte alignment, so the last 4 bits of this register should be 0.	0

27.1.8.4.6 DMA Status Register (R32_ETH_DMASR)

Offset address: 0x1014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PLS	Res	TSTS	PMTS	MMCS	Res		EBS			TPS			RPS		NIS

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AIS	ERS	FBES	Reserved		ETS	RWTS	RPSS	RBUS	RS	TUS	ROS	TJTS	TBUS	TPSS	TS

Bit	Name	Access	Description	Reset Value
31	PLS	RW1Z	Internal 10M physical layer connection status. This bit is set to indicate that the physical layer connection is up or down. Or write 1 to clear.	0
30	Reserved	RO	Reserved	0
29	TSTS	RO	Timestamp trigger status. This bit will be set when an interrupt event occurs in the PTP section, or an interrupt will be generated if the PTP interrupt is enabled. This bit is automatically cleared after clearing all flag bits in the PTP section.	0
28	PMTS	RO	PMT trigger status. This bit will be set when an interrupt event is generated in the PMT part, and an interrupt will be generated if the PMT interrupt is enabled. This bit is automatically cleared after clearing all flag bits in the PMT section.	0
27	MMCS	RO	MMC trigger status. This bit will be set when an interrupt event is generated in the MMC part, and an interrupt will be generated if the MMC interrupt is enabled. This bit is automatically cleared after clearing all flag bits in the MMC section.	0

26	Reserved	RO	Reserved	0
[25:23]	EBS	RO	Error status. This field indicates the type of bus error that caused it. This field is only valid when the DMASR[13] bit is set. DMASR[23]:0: Error sending DMA forwarding data; 1: Error receiving DMA forwarding data; DMASR[24]:0: Error reading data forwarding; RO 1: Error writing data forwarding; DMASR[25]:0: Error accessing descriptor; 1: Error accessing data cache.	0
[22:20]	TPS	RO	Transmit process status. This field is used to indicate the status of the current transmit DMA. 000: Stop. Receive reset or stop sending command; 001: Run. Fetching the send descriptor; 010: Run. Waiting for status information; 011: Run. Reading the transmit buffer data and pushing it into the FIFO; 100,101: Reserved; 110: Pause. The transmit descriptor is not available or the transmit buffer data underflows; 111: Run. Closing the transmit descriptor.	000b
[19:17]	RPS	RO	Receive process status. This field is used to indicate the current state of receiving DMA. 000: Stop. Receive reset or stop sending command; 001: Run. Fetching the receive descriptor; 010: Reserved; 011: Run. Waiting to receive data packets; 100: Pause. The receive descriptor is not available; 101: Run. Closing the receive descriptor; 110: Reserved; 111: Run. Pushing the received data from the FIFO into the memory.	000b
16	NIS	RW1Z	Normal interrupt summary. With interrupts enabled in the DMAIER register, if any of the following bits are set, the NIS bit will also be set. -DMASR[0]: Send interrupt; -DMASR[2]: The transmit buffer is not available; -DMASR[6]: Receive interrupt; -DMASR[14]: Early receive interrupt; -DMASR[31]:Internal 10M physical layer connection state change The NIS bit is sticky, when the corresponding interrupt status bit that causes NIS location 1 is cleared, the NIS bit must also be cleared by writing 1.	0
15	AIS	RW1Z	Abort interrupt summary. With interrupts enabled in the DMAIER register, the AIS bit will also be set if any	0

			of the following bits are set. -DMASR[1]: The transmission is stopped; -DMASR[3]: Transmit Jabber timeout; -DMASR[4]: Receive FIFO overflow; -DMASR[5]: Transmit data underflow; -DMASR[7]: The transmit buffer is not available; -DMASR[8]: The reception is stopped; -DMASR[9]: Receive watchdog timeout; -DMASR[10]: Send early; -DMASR[13]: Bus error; If all the above bits are cleared, the AIS bit will be automatically cleared.	
14	ERS	RW1Z	Early receive status. When this bit is set, it means that the DMA has filled the first buffer when the data frame is received, but the complete frame has not been received yet. After RS is set, the ERS bit is automatically cleared.	0
13	FBES	RW1Z	Bus error. When this bit is set, it indicates that a bus error was sent. See [25:23] bits for specific reasons. When this bit is set, the corresponding DMA controller will close the bus access.	0
[12:11]	Reserved	RO	Reserved	0
10	ETS	RW1Z	Early transmit status. When set, it indicates that the transmit frame has been fully pushed into the FIFO.	0
9	RWTS	RW1Z	Watchdog time-out status. When set, it indicates that the frame length has exceeded 2048 bytes.	0
8	RPSS	RW1Z	Receive process stop status. This bit is set to indicate that the receive process has stopped.	0
7	RBUS	RW1Z	Receive buffer unavailable status. This bit is set to indicate that the permission to receive the descriptor belongs to the CPU, the DMA cannot be obtained, and the receiving process has been suspended. The user needs to release the descriptor and fill the RPDR register with a value to resume the receive flow. The DMA will retry to acquire the descriptor when the next frame is received.	0
6	RS	RW1Z	Receive completed status. This bit is set to indicate that the reception of a frame has been completed, and the frame information is also updated into the descriptor.	0
5	TUS	RW1Z	Transmit data underflow status. This bit is set to indicate that the transmit buffer under flowed the transmit data while transmitting the frame. At this point the sending process is halted and the data underflow error bit is set.	0
4	ROS	RW1Z	Receive status overflow. This bit is set to indicate that a data overflow has occurred in the receive buffer.	0

3	TJTS	RW1Z	Transmit Jabber timer timeout status. This bit is set to indicate that the transmitter is too busy, transmission has stopped, and the descriptor's Jabber timeout bit has been set.	0
2	TBUS	RW1Z	Transmit buffer unavailable status bit. This bit is set to indicate that the transmit descriptor is occupied by the CPU, the DMA cannot be acquired, and the transmit process has been suspended. Bits [22:20] show the current transmit status. The application needs to release the send descriptor.	0
1	TPSS	RW1Z	Transmit process stop status. This bit is set to indicate that the transmission process has stopped.	0
0	TS	RW1Z	Transmit completed status. This bit is set to indicate that the transmission of a frame has been completed, and the ownership of the descriptor has been returned to the CPU.	0

27.1.8.4.7 DMA Operation Mode Register (R32_ETH_DMAOMR)

Offset address: 0x1018

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		ST	Reserved					FEF	FUGF	Reserved				SR	Res

Bit	Name	Access	Description	Reset Value
[31:27]	Reserved	RO	Reserved	0
26	DTCEFD	RW	Discard TCP/IP checksum error frame disable 0: If the FEF bit is 0, the MAC will discard all frames with errors; 1: When an error is found in the checksum of protocols such as TCP/IP/ICMP/UDP, the frame will not be discarded.	0
[25:22]	Reserved	RO	Reserved	0
21	TSF	RW	Transmit store and forward. 0: After the data written into the FIFO in the sending process reaches a certain value, the sending will be started; 1: The sending process will start sending after the complete frame is written into the FIFO.	0
20	FTF	RW	Transmit FIFO empty control. Setting this bit will reset the transmit FIFO.	0
[19:14]	Reserved	RO	Reserved	0
13	ST	RW	Start or stop transmission.	0

			0: After sending the current frame, the sending process enters the stop state; 1: Put the sending process in the running state.	
[12:8]	Reserved	RO	Reserved	0
7	FEF	RW	Forward error frame. 0: All frames are forwarded to the DMA except for frames that are too short; 1: Receive FIFO discards frames with errors.	0
6	FUGF	RW	Forward too short frame. 0: The receive FIFO discards all frames less than 64 bytes in length; 1: The receive FIFO forwards a frame whose length is too short.	0
[5:2]	Reserved	RO	Reserved	0
1	SR	RW	Start or stop reception. 0: After forwarding the currently received frame, the receiving DMA enters the stop mode, and the next transmission starts from the current receiving descriptor position; 1: Open the receiving process, DMA fetches the receiving descriptor from the current position or fetches the receiving descriptor from the identifier header position.	0
0	Reserved	RO	Reserved	0

27.1.8.4.8 DMA Interrupt Enable Register (R32_ETH_DMAIER)

Offset address: 0x101C

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AISE	ERS	FBES	Reserved		ETIE	RWTIE	RPSIE	RBUIE	RIE	TUIE	ROIE	TJTIE	TBUIE	TPSIE	TIE

Bit	Name	Access	Description	Reset Value
31	PLE	RW	Internal 10M physical layer connection state change interrupt enable.	0
[30:17]	Reserved	RO	Reserved	0
16	NISE	RW	Normal interrupt enable. Enabling this bit will enable the following interrupts. -DMASR[0]: Transmit interrupt; -DMASR[2]: Transmit buffer unavailable; -DMASR[6]: Receive interrupt; -DMASR[14]: Early receive interrupt.	0
15	AISE	RW	Abnormal interrupt. Enabling this bit will enable the following interrupts.	0

			-DMASR[1]: The sending process is stopped; -DMASR[3]: Send Jabber timeout; -DMASR[4]: Receive FIFO overflow; -DMASR[5]: transmit data underflow; -DMASR[7]: Send buffer unavailable; -DMASR[8]: The receiving process is stopped; -DMASR[9]: Receive watchdog timeout; -DMASR[10]: Early transmit; -DMASR[13]: Bus error.	
14	ERS	RW	Early receive interrupt enable. Enable this bit to generate an early receive interrupt. NISE must be set. AISE must be set.	0
13	FBES	RW	Bus fatal error interrupt enable. Enable this bit to generate a bus error interrupt. AISE must be set.	0
[12:11]	Reserved	RO	Reserved	0
10	ETIE	RW	Early transmit interrupt enable. Enable this bit to generate an early transmit interrupt. AISE must be set.	0
9	RWTIE	RW	Receive watchdog interrupt enable. Enable this bit to generate a receive watchdog timeout interrupt. AISE must be set.	0
8	RPSIE	RW	Receive process stop interrupt enable. Enable this bit to generate a receive process stop interrupt.	0
7	RBUIE	RW	Receive buffer unavailable interrupt enable. Enable this bit to generate a receive buffer unavailable interrupt. AISE must be set.	0
6	RIE	RW	Receive completed interrupt enable bit. Enable this bit to generate receive completed interrupt. NISE must be set.	0
5	TUIE	RW	Transmit underflow interrupt enable. Enable this bit to generate a transmit underflow interrupt. AISE must be set.	0
4	ROIE	RW	Receive overflow interrupt. Enable this bit to generate a receive overflow interrupt. AISE must be set.	0
3	TJTIE	RW	Transmit Jabber timeout interrupt enable. Enable this bit to generate transmit Jabber timeout interrupt. AISE must be set.	0
2	TBUIE	RW	Transmit buffer unavailable interrupt enable. Enable this bit to generate transmit buffer unavailable interrupt. NISE must be set	0
1	TPSIE	RW	Transmit process stop interrupt enable. Enable this bit to generate transmit process stop interrupt. AISE must be set.	0
0	TIE	RW	Transmit completed interrupt enable. Enable this bit to generate transmit completed interrupt. NISE must be set.	0

27.1.8.4.9 DMA Lost Frame Register (R32_ETH_DMAMFBOCR)

Offset address: 0x1020

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

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

Bit	Name	Access	Description	Reset Value
[31:17]	Reserved	RO	Reserved.	0
16	OMFC	RW	Missing frame counter overflow.	0
[15:0]	MFC	RW	Lost frame counter. This field indicates the number of frames lost due to unavailability of receive buffers.	0

27.1.8.4.10 DMA Current Transmit Descriptor Register (R32_ETH_DMACHTDR)

Offset address: 0x1048

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

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

Bit	Name	Access	Description	Reset Value
[31:0]	CHTDR	RO	This register value points to the transmit descriptor currently in use. This register is updated in real time by DMA.	0

27.1.8.4.11 DMA Current Receive Descriptor Register (R32_ETH_DMACHRDR)

Offset address: 0x104C

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

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

Bit	Name	Access	Description	Reset Value
[31:0]	CHRDR	RO	This register value points to the receive descriptor currently in use. This register is updated in real time by DMA.	0

27.1.8.4.12 DMA Current Transmit Buffer Register (R32_ETH_DMACHTBAR)

Offset address: 0x1050

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

Bit	Name	Access	Description	Reset Value
[31:0]	CHTBAR	RO	This register value points to the transmit buffer currently in use. This register is updated in real time by DMA.	0

27.1.8.4.13 DMA Current Receive Buffer Register (R32_ETH_DMACHRBAR)

Offset address: 0x1054

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

Bit	Name	Access	Description	Reset Value
[31:0]	CHRBAR	RO	This register value points to the receive buffer currently in use. This register is updated in real time by DMA.	0

27.1.8.5 Internal 10M physical layer register address

27.1.8.5.1 Basic Control Register (BMCR)

Offset address: 0x00

Bit	Name	Access	Description	Reset Value
15	Reset	RW/SC	1: PHY reset; 0: Normal operation.	0
14	Loopback	RW	1: Enable loopback; 0: Normal operation.	0
13	Speed Selection	RO	0: 10Mb/s;	0
12	Auto-Negotiation	RW	1: Enable auto-negotiation; 0: Disable auto-negotiation.	1
[11:10]	Reserved	RO	Reserved	0
9	Restart Auto-Negotiation	RW/SC	1: Restart auto-negotiation; 0: Normal operation.	0
8	Duplex Mode	RW	1: Full-duplex; 0: Half-duplex.	1
7	Collision Test	RW	1: Conflict test enabled; 0: Normal operation.	0
[6:0]	Reserved	RO	Reserved	0

27.1.8.5.2 Basic Status Register (BMSR)

Offset address: 0x01

Bit	Name	Access	Description	Reset Value
[15:6]	Reserved	RO	Reserved	0
5	Auto-Negotiation Complete	RO	1: Auto-negotiation completed; 0: Auto-negotiation not completed.	0
[4:3]	Reserved	RO	Reserved	0
2	Link	RO	1: Physical layer establishes a link; 0: Physical layer has not established a link.	0
[1:0]	Reserved	RO	Reserved	0

27.1.8.5.3 Auto-Negotiation Linker Capability Register(R16_ETH_ANLPAR)

Offset address: 0x05

Bit	Name	Access	Description	Reset Value
15	NP	RO	Next page bit. 1: Transmit protocol details data page. 0: Transmit basic capability data page.	0
14	ACK	RO	1: The linker acknowledges receipt of the capability data word from the local node. 0: No answer.	0
13	RF	RO	1: The linker is indicating a remote fault. 0: The linker is not indicating a remote fault.	0
12	Reserved	RO	Reserved	0
11	ASYPAUSE	RO	1: The linker supports asymmetric flow control. 0: The linker does not support asymmetric flow control. When Auto-Negotiation is enabled, this bit reflects the linker's capabilities.	0
10	PAUSE	RO	1: The linker supports flow control. 0: The linker does not support flow control. When Auto-Negotiation is enabled, this bit reflects the linker's capabilities. (read-only).	0
9	100Base-T4	RO	1: 100Base-T4 is supported by the linker. 0: 100Base-T4 is not supported by the linker.	0
8	100Base-TX-FD	RO	1: 100Base-TX full duplex is supported by the linker. 0: 100Base-TX full duplex is not supported by the linker.	0
7	100Base-TX	RO	1: 100Base-TX is supported by the linker. 0: 100Base-TX is not supported by the linker. This bit is also set after a 100Base-TX link is established by parallel detection.	0
6	10Base-T-FD	RO	1: 10Base-TX full duplex is supported by the linker. 0: 10Base-TX full duplex is not supported by the linker.	0

5	10Base-T	RO	1: 10Base-T is supported by the linker. 0: 10Base-T is not supported by the linker. This bit is also set after a 10 Base-T link is established by parallel detection.	0
[4:0]	SELECT	RO	Binary encoding node selector for the linker, currently only CSMA/CD <00001> is specified.	00001b

27.1.8.5.4 Physical Layer Status Register (PHYSR)

Offset address: 0x10

Bit	Name	Access	Description	Reset Value
[15:4]	Reserved	RO	Reserved	0
3	Loopback_10M	RO	1: PHY works in 10M self-loop; 0: Normal mode.	0
2	Full_10M	RO	1: PHY works at 10M full-duplex; 0: Half-duplex mode.	0
[1:0]	Reserved	RO	Reserved	0

27.1.8.5.5 Auto-flip Register (MDIX)

Offset Address: 0x1E

Bit	Name	Access	Description	Reset Value
[15:4]	Reserved	RO	Reserved	0

[3:2]	P/N polarity	RW	1x: Reserved;
			00: P/N polarity is normal;
			01: P/N polarity reversed.
				00b

[1:0]	T/R selection	RW	00: Automatic;
			01: MDIX;
			1x: MDI.
				00b

27.2 For CH32F20x_D8W, CH32V20x_D8, CH32V20x_D8W

27.2.1 Introduction to Ethernet controller

The device integrates MAC, PHY, and DMA Ethernet controllers, compatible with the IEEE802.3 protocol. The internal DMA receives and transmit data to the system RAM. The PHY physical layer is a 10Mbit/s Ethernet transceiver. Several network PHY registers are available, and the receive/transmit performance can be set.

Regarding the bottom operation of the network, the subroutine library is provided mainly for application support, and the registers are not introduced in detail.

Main features:

- Support full-duplex and half-duplex.
- Support short-packet padding settings.
- Support CRC setting and padding.
- Support jumbo frame reception.
- Support different filter mode combinations.

- Support pause frame transmitting and setting.
- Support auto-negotiation mechanism.
- Support DMA for receiving and transmitting data.
- PHY transceiver is compatible with 10BASE-T, and the transmit module supports energy-saving mode.
- Built-in 50ohm transfer impedance matching resistor, also can be connected externally.
- Provide a globally unique MAC address assigned by IEEE.

27.2.2 Register description

Table 27-23 Ethernet controller registers

Name	Offset Address	Description	Reset Value
R8_ETH_EIE	0x40028003	Interrupt enable register	0x00
R8_ETH_EIR	0x40028004	Interrupt flag register	0x00
R8_ETH_ESTAT	0x40028005	Status register	0x00
R8_ETH_ECON2	0x40028006	PHY analog parameter setting register	0x06
R8_ETH_ECON1	0x40028007	Receive/transmit control register	0x00
R32_ETH_TX	0x40028008	Transmit DMA control register	0xXXXXXXXX
R16_ETH_ETXST	0x40028008	Transmit DMA buffer start address register	0xXXXX
R16_ETH_ETXLN	0x4002800A	Transmission length register	0xXXXX
R32_ETH_RX	0x4002800C	Receive DMA control register	0x00000000
R16_ETH_ERXST	0x4002800C	Receive DMA buffer start address register	0x0000
R16_ETH_ERXLN	0x4002800E	Reception length register	0x0000
R32_ETH_HTL	0x40028010	Hash table low register	0x484EA033
R32_ETH_HTH	0x40028014	Hash table high register	0x5000EF97
R32_ETH_MACON	0x40028018	Receive filter control register	0x10000000
R8_ETH_ERXFCON	0x40028018	Receive packet filter control register	0x00
R8_ETH_MACON1	0x40028019	Mac layer flow control register	0x00
R8_ETH_MACON2	0x4002801A	Mac layer packet control register	0x00
R8_ETH_MABBIPG	0x4002801B	Minimum packet interval register	0x10
R32_ETH_TIM	0x4002801C	Flow control pause frame time register	0xXXXXXXXX
R16_ETH_EPAUS	0x4002801C	Flow control pause frame time register	0xXXXX
R16_ETH_MAMXFL	0x4002801E	Maximum receive packet length register	0x0000
R16_ETH_MIRD	0x40028020	MII read register	0x1100
R32_ETH_MIWR	0x40028024	MII write register	0x00000000
R8_ETH_MIREGADR	0x40028024	MII address register	0x00
R8_ETH_MISTAT	0x40028025	MII status register	0x00
R16_ETH_MIWR	0x40028026	MII write register	0x0000
R32_ETH_MAADRL	0x40028028	MAC address low register	0xXXXXXXXX

R16_ETH_MAADRH	0x4002802C	MAC address high register	0xXXXX

27.2.2.1 Interrupt Enable Register (R8_ETH_EIE)

Bit	Name	Access	Description	Reset value
7	RB_ETH_EIE_INTIE	RW	Ethernet interrupt enable 0: Interrupt disabled; 1: Interrupt enabled.	0
6	RB_ETH_EIE_RXIE	RW	Receive completed interrupt enable 0: Interrupt disabled; 1: Interrupt enabled.	0
5	Reserved	RO	Reserved	0
4	RB_ETH_EIE_LINKIE	RW	Link change interrupt enable 0: Interrupt disabled; 1: Interrupt enabled.	0
3	RB_ETH_EIE_TXIE	RW	Transmit completed interrupt enable 0: Interrupt disabled; 1: Interrupt enabled.	0
2	RB_ETH_EIE_R_EN50	RW	Built-in 50ohm impedance matching resistor enable 0: On-chip resistor disconnected; 1: On-chip resistor connected.	0
1	RB_ETH_EIE_TXERIE	RW	Transmit error interrupt enable 0: Interrupt disabled; 1: Interrupt enabled.	0
0	RB_ETH_EIE_RXERIE	RW	Receive error interrupt enable 0: Interrupt disabled; 1: Interrupt enabled.	0

27.2.2.2 Interrupt Flag Register (R8_ETH_EIR)

Bit	Name	Access	Description	Reset value
7	Reserved	RO	Reserved	0
6	RB_ETH_EIR_RXIF	RW1	Receive completed flag	0
5	Reserved	RO	Reserved	0
4	RB_ETH_EIR_LINKIF	RW1	Link change flag	0
3	RB_ETH_EIR_TXIF	RW1	Transmit completed flag	0
2	Reserved	RO	Reserved	0
1	RB_ETH_EIR_TXERIF	RW1	Transmit error flag	0
0	RB_ETH_EIR_RXERIF	RW1	Receive error flag	0

27.2.2.3 Status Register (R8_ETH_ESTAT)

Bit	Name	Access	Description	Reset value

7	RB_ETH_ESTAT_INT	RW1	Interrupt	0
6	RB_ETH_ESTAT_BUFER	RW1	Buffer error	0
5	RB_ETH_ESTAT_RXCRCER	RO	Receive CRC error	0
4	RB_ETH_ESTAT_RXNIBBLE	RO	Receive nibble error	0
3	RB_ETH_ESTAT_RXMORE	RO	Receive more than the set maximum packets	0
2	RB_ETH_ESTAT_RXBUSY	RO	Receive in progress	0
1	RB_ETH_ESTAT_TXABRT	RO	Transmission interrupted by MCU	0
0	Reserved	RO	Reserved	0

27.2.2.4 PHY Analog Parameter Setting Register (R8_ETH_ECON2)

Bit	Name	Access	Description	Reset value
[7:4]	Reserved	RO	Reserved	0000b
[3:1]	RB_ETH_ECON2_RX RB_ETH_ECON2_MUST	RW	Reserved. Must write 011.	011b
0	RB_ETH_ECON2_TX	RW	Transmitter energy-saving driver control 0: Rated driver. 1: Energy-saving driver.	0

27.2.2.5 Receive/Transmit Control Register (R8_ETH_ECON1)

Bit	Name	Access	Description	Reset value
7	RB_ETH_ECON1_TXRST	RW	Transmit module reset 0: Not reset. 1: Reset the transmit module.	0
6	RB_ETH_ECON1_RXRST	RW	Receive module reset 0: Not reset. 1: Reset the receive module.	0
[5:4]	Reserved	RO	Reserved	00b
3	RB_ETH_ECON1_TXRTS	RW	Transmit start. Cleared automatically after the transmission is completed. 1: Start transmission. 0: No effect.	0
2	RB_ETH_ECON1_RXEN	RW	Receive enable 0: Receive disabled. 1: Receive enabled.	0
[1:0]	Reserved	RO	Reserved	00b

27.2.2.6 Transmit DMA Buffer Address Register (R16_ETH_ETXST)

Bit	Name	Access	Description	Reset value
[15:0]	R16_ETH_ETXST	RW	Transmit DMA buffer start address Low 15 bits are valid. The address must	XXXXh

			be 4-byte aligned.	

27.2.2.7 Transmission Length (R16_ETH_ETXLN)

Bit	Name	Access	Description	Reset value
[15:0]	R16_ETH_ETXLN	RW	Transmission length	XXXXh

27.2.2.8 Receive DMA Buffer Address Register (R16_ETH_ERXST)

Bit	Name	Access	Description	Reset value
[15:0]	R16_ETH_ERXST	RW	Receive DMA buffer start address Low 15 bits are valid. The address must be 4-byte aligned.	XXXXh

27.2.2.9 Reception Length Register (R16_ETH_ERXLN)

Bit	Name	Access	Description	Reset value
[15:0]	R16_ETH_ERXLN	RO	Reception length	0000h

27.2.2.10 Hash Table Low Register (R32_ETH_HTL)

Bit	Name	Access	Description	Reset value
[31:24]	R8_ETH_EHT3	RW	Hash Table byte 3	xxh
[23:16]	R8_ETH_EHT2	RW	Hash Table byte 2	xxh
[15:8]	R8_ETH_EHT1	RW	Hash Table byte 1	xxh
[7:0]	R8_ETH_EHT0	RW	Hash Table byte 0	xxh

27.2.2.11 Hash Table High Register (R32_ETH_HTH)

Bit	Name	Access	Description	Reset value
[31:24]	R8_ETH_EHT7	RW	Hash Table byte 7	xxh
[23:16]	R8_ETH_EHT6	RW	Hash Table byte 6	xxh
[15:8]	R8_ETH_EHT5	RW	Hash Table byte 5	xxh
[7:0]	R8_ETH_EHT4	RW	Hash Table byte 4	xxh

27.2.2.12 Receive Filter Control Register (R8_ETH_ERXFCON)

Bit	Name	Access	Description	Reset value
7	RB_ETH_ERXFCON_UCEN	RW	Unicast match filter settings 0: discard all unicast packets. 1: Receive packets with matching destination address.	0
6	Reserved	RO	Reserved	0
5	RB_ETH_ERXFCON_C	RW	CRC checksum filter settings	0

	RCEN		0: Discard packets with CRC errors. 1: Receive packets with CRC errors.	
4	RB_ETH_ERXFCON_EN	RW	Receive filtering enable 0: Disable the receive filtering function. 1: Enable the receive filtering function.	0
3	RB_ETH_ERXFCON_MPEN	RW	Magic packet filter settings 0: Discard magic packets. 1: Receive magic packets.	0
2	RB_ETH_ERXFCON_HTEN	RW	Hash table matching filter settings 0: discard hash table match packets. 1: Receive packets with matching hash table.	0
1	RB_ETH_ERXFCON_MCEN	RW	Multicast packet matching filter settings 0: Discard all multicast packets. 1: Receive all multicast packets.	0
0	RB_ETH_ERXFCON_BCEN	RW	Broadcast packet matching filter settings 0: Discard all broadcast packets. 1: Receive all broadcast packets.	0

27.2.2.13 Mac Layer Flow Control Register (R8_ETH_MACON1)

Bit	Name	Access	Description	Reset value
[7:6]	Reserved	RO	Reserved	00b
[5:4]	RB_ETH_MACON1_FCEN	RW	Pause frame setting. Active at full-duplex 00: Stop transmitting pause frame; 01: Transmit a pause frame, and then stop transmitting; 10: Periodically transmit pause frames; 11: Transmit 0 timer pause frame, and then stop transmitting.	00b
3	RB_ETH_MACON1_TXPAUS	RW	Transmit pause frame enable control 0: Not transmit pause frame; 1: Transmit enabled.	0
2	RB_ETH_MACON1_RXPAUS	RW	Receive pause frame enable 0: Not receive pause frame; 1: Receive enabled.	0
1	RB_ETH_MACON1_PASSALL	RW	Control frame setting 0: Control frame will be filtered; 1: Control frame that is not filtered will be written to the buffer.	0
0	RB_ETH_MACON1_MARXEN	RW	MAC layer receive enable 0: MAC does not receive data; 1: MAC receive enabled.	0

27.2.2.14 Mac Layer Packet Control Register (R8_ETH_MACON2)

Bit	Name	Access	Description	Reset value
[7:5]	RB_ETH_MACON2_PADCFG	RW	Short-packet pad configuration 7: All short packets are padded with 0 to 64 bytes, then 4-byte CRC; 6: Not pad the short packet; 5: The detected VLAN network packet whose field is 8100h is automatically padded with 0 to 64 bytes. Otherwise the short packet is padded with 60 bytes of 0, and then padded with 4-byte CRC; 4: Not pad the short packet; 3: All short packets are padded with 0 to 64 bytes, then 4-byte CRC; 2: Not pad the short packet; 1: All short packets are padded with 0 to 64 bytes, then 4-byte CRC; 0: Not pad the short packet.	000b
4	RB_ETH_MACON2_TXCRCEN	RW	Transmit add CRC control 0: Hardware does not pad CRC; 1: Hardware pads CRC.	0
3	RB_ETH_MACON2_PHDREN	RW	Special 4 bytes are not involved in CRC.	0
2	RB_ETH_MACON2_HFRMEN	RW	Jumbo frame received enable 0: Disabled to receive jumbo frame; 1: Enable to receive jumbo frame.	0
1	Reserved	RO	Reserved	0
0	RB_ETH_MACON2_FULDPX	RW	Ethernet communication mode 0: Half-duplex; 1: Full-duplex.	0

27.2.2.15 Minimum Packet Interval Register (R8_ETH_MABBIPG)

Bit	Name	Access	Description	Reset value
7	Reserved	RO	Reserved	0
[6:0]	R8_ETH_MABBIPG	RW	Minimum number of packet interval bytes	0010000b

27.2.2.16 Flow Control Pause Frame Time Register (R16_ETH_EPAUS)

Bit	Name	Access	Description	Reset value
[15:0]	R16_ETH_EPAUS	RW	Flow control pause frame time	XXXXh

27.2.2.17 Maximum Receive Packet Length Register (R16_ETH_MAMXFL)

Bit	Name	Access	Description	Reset

				value
[15:0]	R16_ETH_MAMXFL	RW	Maximum receive packet length	0000h

27.2.2.18 MII Read Register (R16_ETH_MIRD)

Bit	Name	Access	Description	Reset value
[15:0]	R16_ETH_MIRD	RW	MII read register	1100h

27.2.2.19 MII Address Register (R8_ETH_MIREGADR)

Bit	Name	Access	Description	Reset value
[7:5]	Reserved	RO	Reserved	000b
[4:0]	RB_ETH_MIREGADR_MIRDL	RW	PHY register address	00000b

27.2.2.20 MII Status Register (R8_ETH_MISTAT)

Bit	Name	Access	Description	Reset value
[7:1]	Reserved	RO	Reserved	0000000b
0	R8_ETH_MII_STA	RO	MII register status 1: Write to MII register; 0: Read MII register.	0

27.2.2.21 MII Write Register (R16_ETH_MIWR)

Bit	Name	Access	Description	Reset value
[15:0]	R16_ETH_MIWR	WO	MII write register	0000h

27.2.2.22 MAC Address Register (R32_ETH_MAADRL, R16_ETH_MAADRH)

Bit	Name	Access	Description	Reset value
[31:0]	R32_ETH_MAADRL	RW	MAC Address bytes 1 to 4	XXXXXXXXh
[15:0]	R16_ETH_MAADRH	RW	MAC Address bytes 5 to 6	XXXXh

Note: The contents of the internal 10M physical layer related registers are detailed in 27.1.8.5 Internal 10M physical layer related register address section.

27.2.3 Operation guide

1. Initialization

(1) Configure the safe register to enter safe mode. Enable Ethernet clock and power.

(2) Enable the corresponding interrupt, which is optional. Enable impedance matching resistor.

(3) Configure the receive filter mode, CRC function, MAC address.

(4) Set the buffer.

(5) Start receiving and enable interrupt.

2. Transmit data
   (1) Write to R16_ETH_ETXLN data length.
   (2) Write to R16_ETH_ETXST data address.
   (3) Enable RB_ETH_ECON1_TXRTS to start transmitting.

3. Receive data
   (1) Preset the receive address and enable the reception.
   (2) Use the interrupt or query the receive completed status.
   (3) Read R16_ETH_ERXLN reception length.
   (4) Update R16_ETH_ERXST receiving address.

For specific applications, please use based on the Ethernet protocol stack library, and refer to the provided network application routines.

Chapter 28 SDIO Interface (SDIO)

The module description in this chapter applies to the full range of CH32F2x, CH32V2x and CH32V3x microcontrollers

The "SDIO" mentioned separately in this chapter refers to a communication interface on the microcontroller designed to operate external memory cards such as SD cards or other devices, and is a peripheral of the microcontroller. The SDIO of the microcontroller is directly mounted on the AHB bus, and the clock is directly provided by HCLK, which can achieve higher communication speed. The SDIO of the microcontroller is used as the SDIO host, and the controlled devices are also collectively referred to as SDIO devices. In applications, SDIO is generally used to read and write SD cards, TF cards or eMMC particles, or to control other devices that use SDIO as a communication interface, such as Wi-Fi/4G modules.

28.1 Main features

28.1.1 Features

- Support SD card, SDIO card and MMC card
- 1-bit, 4-bit and 8-bit bus support
- The clock of SDIO can reach half of HCLK at maximum
- Compatible with MMC Specification 4.5
- Compatible with SD card specification 2.0, SDIO card specification 2.0
- Not compatible with SPI or QSPI

28.1.2 Overview

The SDIO of the microcontroller supports communication with memory such as SD cards or MMC cards. It needs to be clear that SDIO only provides a set of clocks, data and command control sequences required to realize the single command transmission of SD cards and MMC cards. The sequence and combination of commands need to be determined by the user through the program. In addition, for various memory cards, SDIO can only realize the function of reading and writing, and the function of the file system provided by the file system needs to be realized by the user to build the file system through the program.

SDIO is different from the QSPI interface. It has no chip select pin and has an extra CMD pin. CMD can be regarded as a special data line, which is specially used to transmit commands and responses; SDIO is available in 1-bit, 4-bit and 8-bit data line widths; the SDIO clock generally works at a frequency below 400KHz during configuration. When the data transmission is officially performed, it can be configured as the maximum clock supported by the SDIO device. The maximum SDIO clock output supported by the microcontroller is Half of HCLK, according to the protocol, when the clock received by the SDIO device is greater than a certain threshold, the waveform peaks of the clock line, data line and command line need to be reduced to save the time-consuming of waveform rising and falling.

Unlike SD cards, SDIO cards often refer to non-storage devices such as WIFI/Bluetooth modules and 4G modules that use SDIO interfaces. If there is no special instruction, the content described in this chapter must be applicable to SD card, and the content only applicable to SDIO card or MMC card will be specially pointed out. In the description of this chapter, the SD card is regarded as the potential operation object first, followed by the SDIO card, and finally the MMC card.

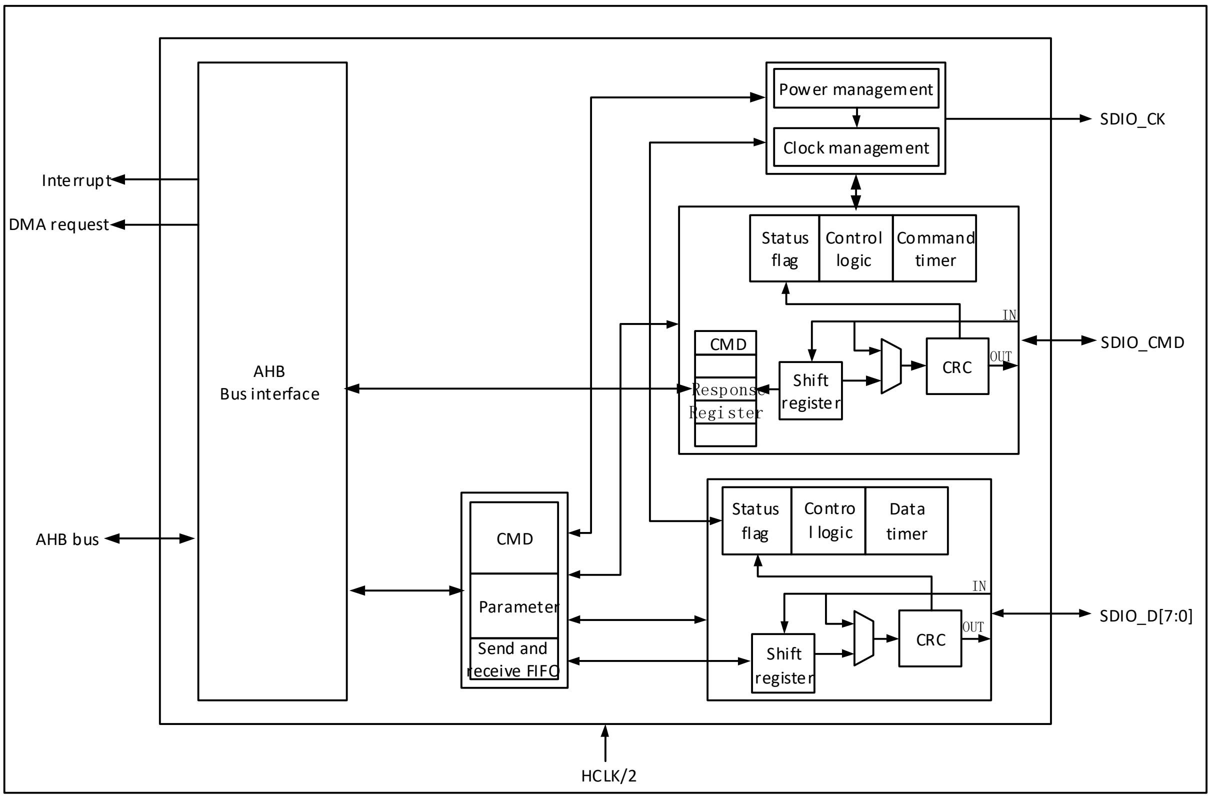
28.2 Interface and clock

SDIO receives the control of the CPU through the AHB bus. There is a FIFO interface in the SDIO register. The CPU or DMA obtains or generates data by reading and writing the FIFO. SDIO is directly clocked by HCLK, and has an interrupt entry that supports multiple interrupt sources. The pins directly controlled by SDIO are SDIO_CK, SDIO_CMD, SDIO_D [7:0], which are connected to SDIO devices through several pins. SDIO is a host-dominated communication interface, and all transfers must be initiated by the microcontroller.

28.2.1 Peripheral structure

The structure of SDIO is shown in Figure 28-1.

Figure 28-1 SDIO structure diagram



SDIO is a peripheral that the microcontroller directly operates the SDIO device as a host. It is roughly composed of 5 modules: AHB interface, clock control part, CMD line control part, data line control part and control register part. SDIO is a half-duplex for peripherals, the CPU writes the commands and data to be sent to the control register, and the command line and data line control module is responsible for pushing the command or data to the IO and adding CRC. The data flow of SDIO is responsible for the transfer from RAM to SDIO FIFO by general-purpose DMA. SDIO FIFO has 32 32-bit sizes.

28.2.2 Pins and their configuration

The pins that need to be configured for SDIO and their modes are shown in Table 28-1.

Table 28-1 SDIO pin configuration

GPIO	SDIO alternate function	Required pin mode	Required speed
PC[8:11]	SDIO_D[0:3]	Push-pull multiplexed output	50MHz
PC12	SDIO_CK	Push-pull multiplexed output	50MHz

PD2	SDIO_CMD	Push-pull multiplexed output	50MHz
PB8	SDIO_D4	Push-pull multiplexed output	50MHz
PB9	SDIO_D5	Push-pull multiplexed output	50MHz
PC6	SDIO_D6	Push-pull multiplexed output	50MHz
PC7	SDIO_D7	Push-pull multiplexed output	50MHz

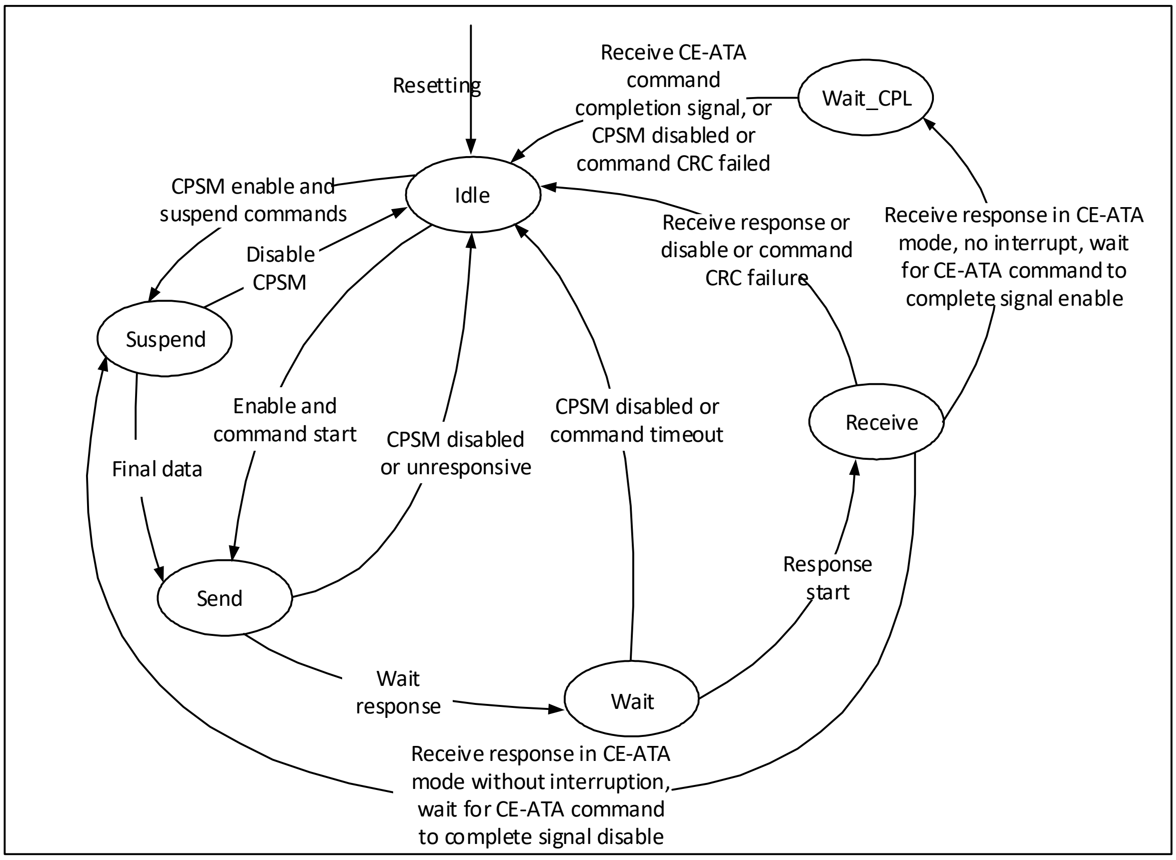
28.2.3 Clock

The clock pin of SDIO is SDIO_CK, and its output clock is obtained by dividing the frequency of HCLK, and the frequency dividing coefficient can be configured as any integer value between 2 and 261. SDIO devices generally only support a single bus mode with a clock of up to 400KHz during initialization. After initialization, the host generally initiates a switch to a low voltage operation, and at the same time increases the clock to the maximum clock that both the microcontroller and the SDIO device can accept. Different versions and speed grades of SD cards support different clock speeds and switching processes, and users need to understand for themselves.

28.2.4 Command path state machine

The command workflow of SDIO follows the state machine shown in the figure below.

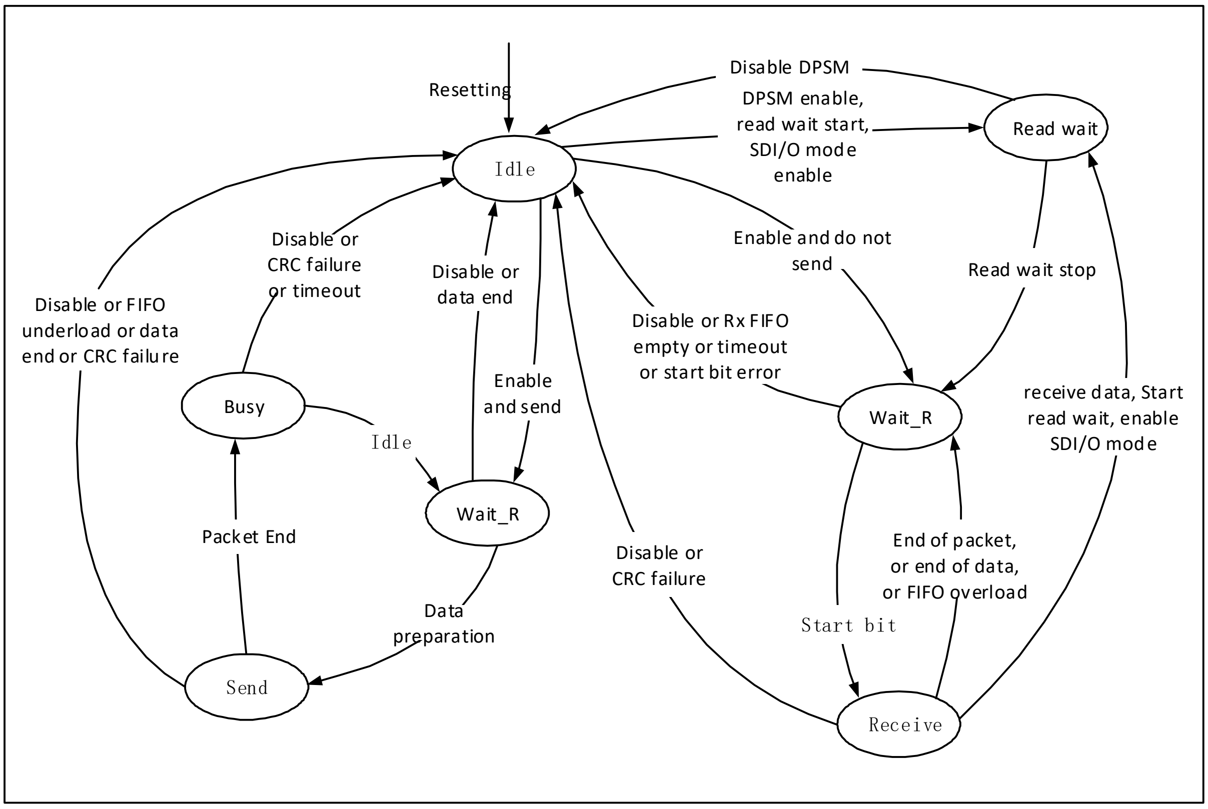
Figure 28-2 Command path state machine


Resetting
Receive CE-ATA command completion signal, or CPSM disabled or command CRC failed
Wait_CPL
Idle
CPSM enable and suspend commands
Disable CPSM
Receive response or disable or command CRC failure
Receive response in CE-ATA mode, no interrupt, wait for CE-ATA command to complete signal enable
Suspend
Enable and command start
CPSM disabled or unresponsive
CPSM disabled or command timeout
Receive
Final data
Send
Response start
Wait response
Wait
Receive response in CE-ATA mode without interruption, wait for CE-ATA command to complete signal disable

28.2.5 Data path state machine

The data workflow of SDIO follows the state machine shown in the figure below.

Figure 28-3 Data path state machine


Resetting
Idle
Disable DPSM
DPSM enable, read wait start, SDI/O mode enable
Read wait
Disable or CRC failure or timeout
Disable or data end
Enable and do not send
Read wait stop
Disable or FIFO underload or data end or CRC failure
Disable or Rx FIFO empty or timeout or start bit error
Enable and send
Busy
Idle
Wait_R
receive data, Start read wait, enable SDI/O mode
Wait_R
Packet End
End of packet, or end of data, or FIFO overload
Disable or CRC failure
Data preparation
Start bit
Send
Receive

28.3 SDIO protocol description

The communication on SDIO is based on transmission as the smallest unit. Each transmission always starts with the host sending a command on the CMD line. After some commands are sent, the SDIO device will also send a piece of data on the CMD line to reply to the host, which is called " Response", sometimes accompanied by the transmission of data, the transmission of data is on the D line. The format of commands and responses is fixed, and the definition of each field is determined according to different commands or responses. Responses and data transfers need to be issued or stopped within a specified time after the end of the command or response, otherwise a timeout error will be generated.

The purpose of this section is to let users have a preliminary understanding of the details of some specifications necessary to use SDIO in a relatively small space, and it does not guarantee detailed and timely updates. The SDIO of the microcontroller only guarantees that it implements the hardware operation basis of the SD 2.0, SDIO 2.0 and MMC 4.5 specifications, and does not necessarily support the functions defined in the higher version specifications, such as double-edge sampling. Users should carry out SDIO interactive programming based on SD specification, SDIO specification and MMC specification when doing specific development.

28.3.1 Bus timing

The transmission of the SD card is initiated by the host initiating the CMD. The SD card may not reply with a response, but may reply with a short response and a long response, and some responses will be accompanied by data transmission. In the communication of the SDIO card or MMC card, there may also be a situation where the SDIO device actively reports an interruption. The timing is shown in the group of figures below.

Figure 28-4 SDIO no response timing and no data timing (taking SD card as an example)

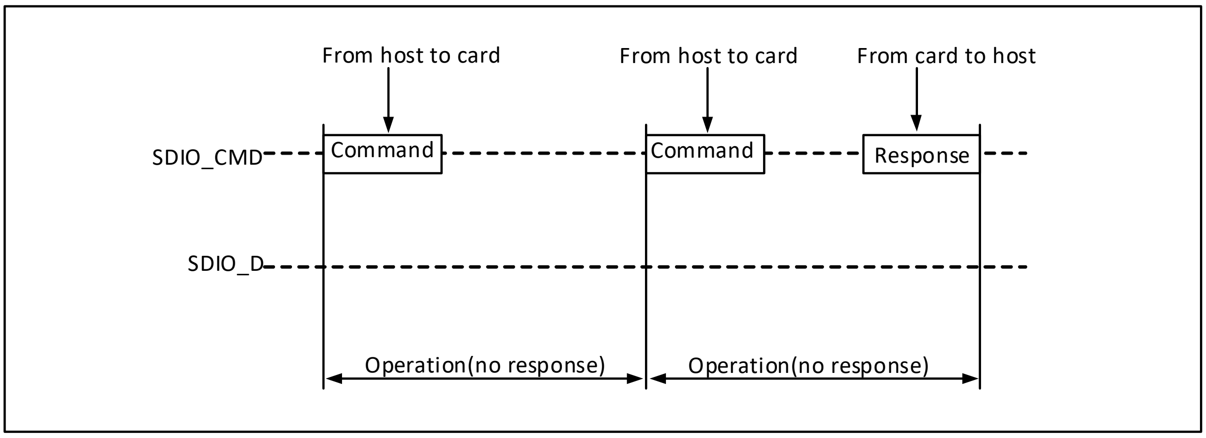


Figure 28-5 SDIO multi-block read timing (taking SD card as an example)

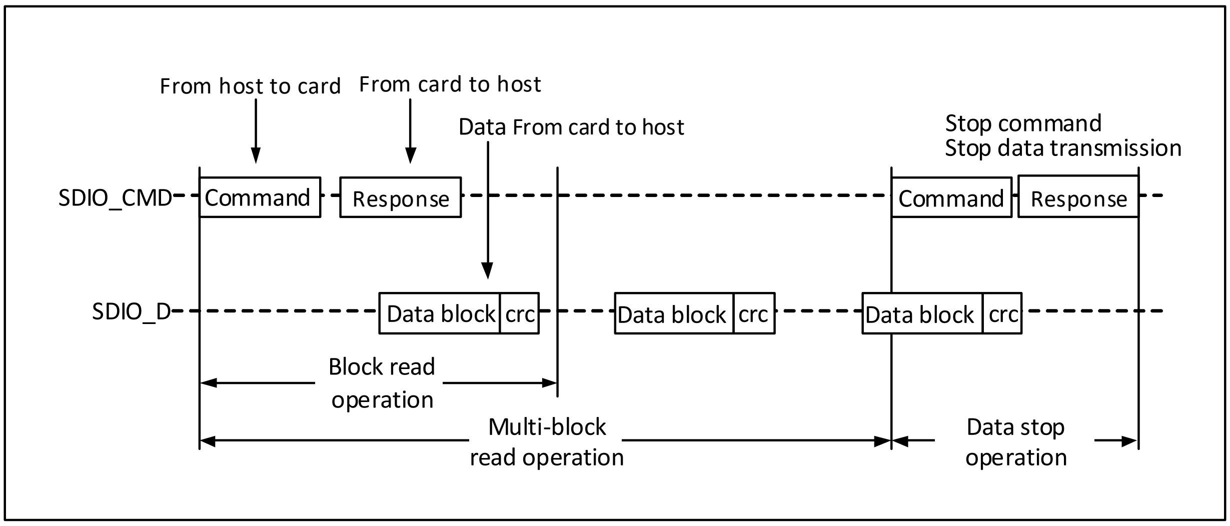


Figure 28-6 SDIO multi-block write timing (taking SD card as an example)

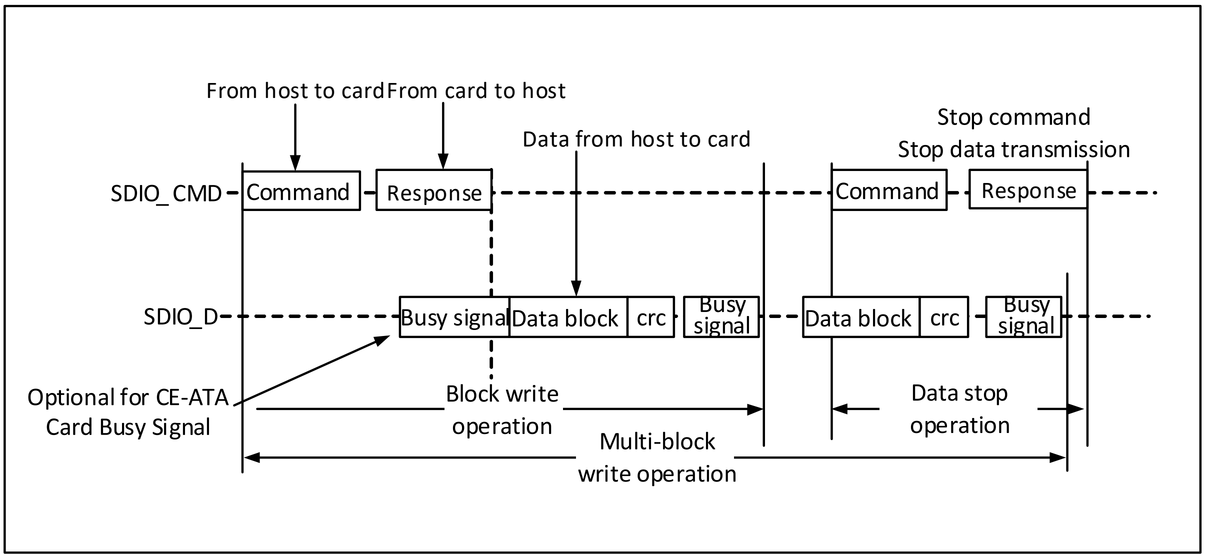


Figure 28-7 SDIO data stream read timing (taking SDIO card as an example)

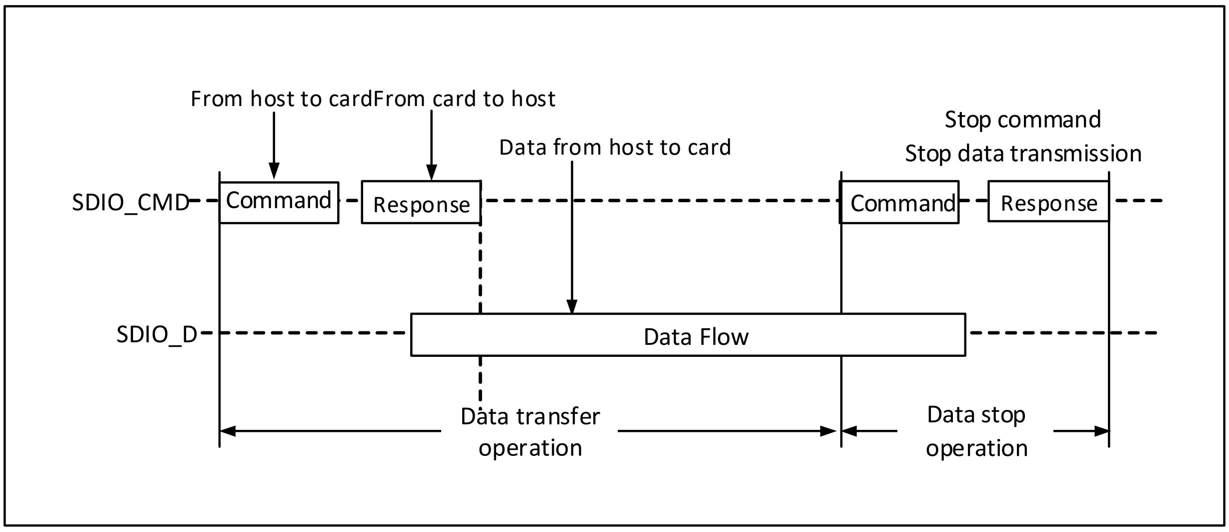
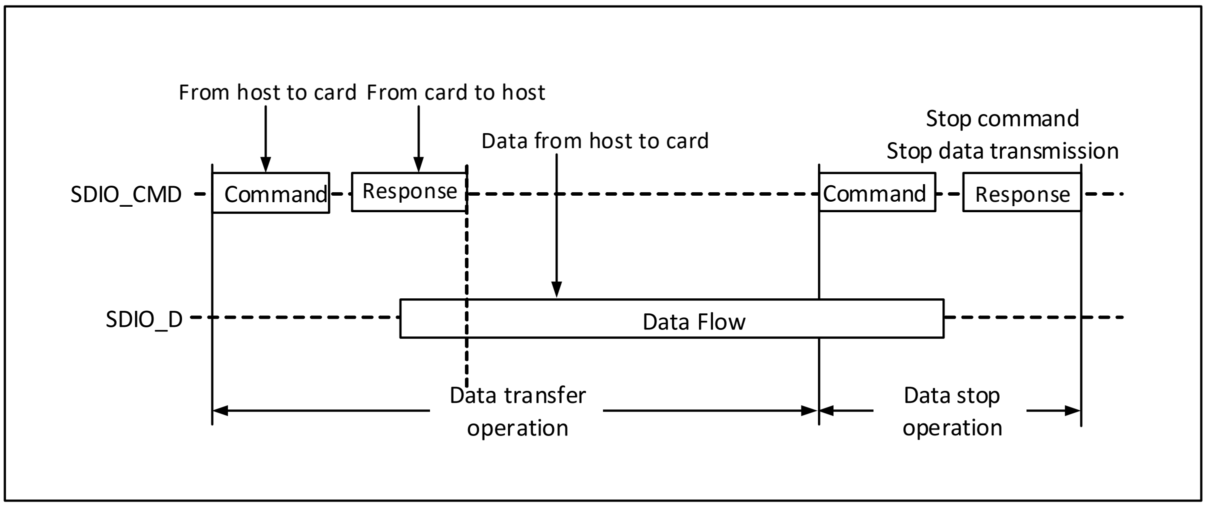


Figure 28-8 SDIO data stream write timing (taking SDIO card as an example)



28.3.2 Command

Most SDIO transfers start with a command (CMD), through which the SDIO host informs the device of its intentions. The format of the command is as follows.

Table 28-2 Command format

Bit/Field name	Start bit	Transmission bit	Command index	Command parameters	CRC7	End bit
Rank/Width	47	46	[45:40] / 6 bits	[39:8] / 32 bits	[7:1] / 7 bits	0
Value	0b	1b	X	X	X	1b

4 types of commands:

1） Broadcast command (bc): sent to all cards on the bus, no response returned;
2） Broadcast command with response (bcr): sent to all cards on the bus, and a response is returned;
3） Point-to-point command (ac): issued to a specific card, but no data transmission;
4） Point-to-point command with data transmission (adtc): issued to a specific card with data transmission attached.

Below are some commonly used commands.

28.3.2.1 Basic command

Basic commands are some of the more basic functions supported by the SD card.

Table 28-3 SD Card Basic Commands

Command index	Type	Parameter	Response format	Abbreviation	Description
CMD0	bc	[31:0] stuff bits are meaningless.	None	GO_IDLE_STATE	Reset all cards to IDLE state
CMD2	bcr	[31:0] stuff bits are meaningless.	R2	ALL_SEND_CID	All cards reply to CID
CMD3	bcr	[31:0] stuff bits are meaningless.	R6	SEND_RELATIVE_ADDR	Reply to new RCA
CMD7	ac	[31:16] RCA; [15:0] stuff bits are meaningless.	R1b Selected Card	SELECT/DESELECT_CARD	Check or uncheck a card
CMD8	bcr	[31:12] reserved bits; [11:8] powered; [7:0] check mode.	R7	SEND_IF_COND	Send SD card interface conditions.
CMD9	ac	[31:16] RCA; [15:0] stuff bits are meaningless.	R2	SEND_CSD	Require CSD。
CMD10	ac	[31:16] RCA; [15:0] stuff bits are meaningless.	R2	SEND_CID	Require CID。
CMD11	ac	[31:0] stuff bits are meaningless.	R1	VOLATGE_SWITCH	Switch to 1.8V.
CMD12	ac	[31:0] stuff bits are meaningless.	R1b	STOP_TRANSMISSION	Force the card to stop transmission;
CMD13	ac	[31:16] RCA; [15] stuff bits are meaningless; [14:0] stuff bits.	R1	SEND_STATUS/SEND_TASK_STATUS	Transmit Status or Task Status Register
CMD15	ac	[31:16] RCA, [15:0] stuff bits are meaningless.	None	GO_INACTIVE_STATE	The card is required to enter the INACTIVE mode;

28.3.2.2 Erase command

The SD card is also a FLASH structure, and the FLASH needs to be erased before writing, but the SD card integrates erasing logic. If it is found that there is no erasure before executing the write command, it will automatically supplement the erasing operation. In many cases, especially before large batch writes, it can help to improve efficiency if the erase is actively performed.

Table 28-4 SD Card Erase Command [1]

Command index	Type	Parameter	Response format	Abbreviation	Description
CMD32	ac	[31:0] address to start erasing [2]	R1	ERASE_WR_BLK_START	Set the first address for erasing
CMD33	ac	[31:0] address to start erasing [2]	R1	ERASE_WR_BLK_END	Set the first address for erasing
CMD38	ac	[31:0] erase mode	R1b	ERASE	The parameter is 0, that is, ordinary wipe

Note 1: The erase command here is the erase command of SD card defined by SD protocol specification, and the erase command of SDIO card and MMC card is different from this.
Note 2: Currently commonly used SDHC/SDXC (2GB to 2TB) level erase address written by the card must be a block address, that is, 512-byte alignment.

28.3.2.3 Block read command

Table 28-5 SD Card block read command [1]

Command index	Type	Parameter	Response format	Abbreviation	Description
CMD16	ac	[31:0] block length	R1	SET_BLOCKLEN	write block length, 512
CMD17	adtc	[31:0] address of a single block [1]	R1	READ_SINGLE_BLOCK	Set the start address of a single read
CMD18	adtc	[31:0] address of a single block [1]	R1	READ_MULTIPLE_BLOCK	Set the start address of a single read
CMD19	adtc	[31:0] reserved bits	R1	SEND_TUNING_BLOCK	Send a 64-byte sequence representing the mode change
CMD20	ac	[31:28] reserved bits [27:0] speed control bits	R1b	SPEED_CLASS_CONTROL	speed control class
CMD22	ac	[31:6] reserved bits [5:0] extended address	R1	ADDRESS_EXTENSION	SDUC will be used.
CMD23	ac	[31:0] block counter	R1	SET_BLOCK_COUNT	block counter

Note: The erasing address written by the SDHC/SDXC (2GB to 2TB) level cards commonly used at present must be a block address, that is, 512-byte alignment.

28.3.2.4 Write command for block transfer

Table 28-6 SD Card block read command [1]

Command index	Type	Parameter	Response format	Abbreviation	Description
CMD16	ac	[31:0] block length	R1	SET_BLOCKLEN	write block length, 512
CMD24	adtc	[31:0] block address [1]	R1	WRITE_BLOCK	Set the start address of a single block write
CMD25	adtc	[31:0] addresses of multiple blocks [1]	R1	WRITE_MULTIPLE_BLOCK	Set the start address of multiple blocks write
CMD27	adtc	[31:0] stuff bits	R1	PROGRAM_CSD	Programmable word programming for CSD

Note: The erasing address written by the SDHC/SDXC (2GB to 2TB) level cards commonly used at present must be a block address, that is, 512-byte alignment.

28.3.3 Response

As a necessary reply from the SDIO device to the host, the response is also transmitted on the CMD line and must be replied within a specified time. The response transmission is high-order first and low-order last. The length of the response and the definition of each bit and each field are determined by the type of response, but all responses start with a fixed 0 start bit, followed by a fixed 0 transmission Direction bit [1]. All responses have a stop bit at the end, fixed at 1[1]. There are roughly 7 kinds of responses, SD card supports R1 / R1b / R2 / R3 / R6 / R7, SDIO card also supports R4 / R5, see the format below.

28.3.3.1 R1 response

Normal response, the total length is 48 bits, with CRC7, and the card status field is 32 bits. The format is as follows.

Table 28-7 R1 response

Bit/Field name	Start bit	Transmission bit	Response index	Card Status	CRC7	End bit
Rank/Width	47	46	[45:40] / 6 bits	[39:8] / 32 bits	[7:1] / 7 bits	0
value	0b	0b	Follow CMD Index	X	X	1b

28.3.3.2 R1b response

The format of R1b is the same as that of R1, but a busy signal can be added after the response, that is, the data line D2 is clamped. The host needs to deal with it after receiving the busy signal (detecting that SDIO_D2 is low).

28.3.3.3 R2 response

Responses to specific commands, the total length is 136 bits, the CRC7 is included in the card status field, and the card status field is 128 bits. The card status field stores the value of the CID register or CSD register. The CID register is generally used as the reply to CMD2/CMD10, and the CSD register is generally used as the reply to CMD9. For the specific meaning of the CID/CSD register, see the section 28.4.2 Device Registers. It should be noted that R2 will only reply to the [127:1] segment of the CID/CSD register. The reserved bit [2] of the CID/CSD [0] fixed to 1 is occupied by the end bit, and the end bit is also fixed to 1. The format is as follows.

Table 28-8 R2 response

Bit/Field name	Start bit	Transmission bit	Command index	Card status	End bit
Rank/Width	135	134	[133:128] / 6 bits	[127:1] / 127 bits	0
Value	0b	0b	111111b	CID/CSD	1b

28.3.3.4 R3 response

Reply to the dedicated response of the OCR register, the total length is 48 bits, and there is no CRC7. The OCR register is generally used as a reply to ACMD41. The format is as follows.

Table 28-9 R3 response

Bit/Field name	Start bit	Transmission bit	Command index	Card Status	Reserved	End bit
Rank/Width	47	46	[45:40] / 6 bits	[39:8] / 32 bits	[7:1] / 7 bits	0
Value	0b	0b	111111b	OCR	1111111b	1b

28.3.3.5 R4 response

In response to the response of CMD5, reply to the special response of OCR and related registers, with a total length of 48 bits and a CRC7. R4 is used in SDIO card, the format is as follows.

Table 28-10 R4 response

Bit/Field name	Bit position	Width (bit)	value
Start bit	47	1	0b
Transmission bit	46	1	0b
Reserve	[45:40]	6	111111b
Card ready	39	1	X
IO number of functions	[38:36]	3	X
Current register	[35]	1	X
Stuff bits	[34:33]	2	00b
S18A	32	1	X
IO OCR	[31:8]	24	OCR
CRC field	[7:1]	7	1111111b
End bit	0	1	1b

Note: The format of MMC card R4 is different from SDIO card.

28.3.3.6 R5 response

A dedicated response to CMD5, with a total length of 48 bits and a CRC7. R5 is used in SDIO card, the format is as follows.

Table 28-11 R5 response

Bit/Field name	Start bit	Transmission bit	Command index	Stuff bit	Response format	Read and write data	CRC7	End bit
Rank	47	46	[45:40]	[39:24]	[23:16]	[15:8]	[7:1]	0
width	1	1	6	16	8	8	7	1
Value	0b	0b	110100b	0000h	X	X	X	1b

Note: The format of MMC card R5 is different from SDIO card.

28.3.3.7 R6 response

Reply to the special response of RCA, with a total length of 48 bits and a CRC7. The format is as follows.

Table 28-12 R6 response

Bit/Field name	Start bit	Transmission bit	Command index	Card RCA	Card status bit	CRC7	End bit
rank	47	46	[45:40]	[39:24]	[23:8]	[7:1]	0
Width	1	1	6	16	16	7	1
Value	0b	0b	000011b	X	X	X	1b

28.3.3.8 R7 response

In response to the dedicated response of CMD8, it indicates the information of the supported voltage, with a total length of 48 bits, with CRC7, and the format is as follows.

Table 28-13 R7 response

Bit/Field name	Start bit	Transmission bit	Command index	Reserved bit	PCIe 1V2 support	PCIe response	Accepted Voltage	Check Feedback	CRC7	End bit
Bit position	47	46	[45:40]	[39:22]	21	20	[19:16]	[15:8]	[7:1]	0
Width	1	1	6	18	1	1	4	8	7	1
Value	0b	0b	001000b	00000h	X	X	X	X	X	1b

28.3.4 Data transfer

Data transfer is carried out on the data line SDIO_D, which has 3 widths of 1/4/8 bits. During data transfer, the start bit of a clock is generally at the front, the high bit of each byte is in the front, and the low bit is in the back. For the block transfer mode only supported by the SD card, the data transfer of each block is followed by a CRC. The MMC card also supports the data stream transfer mode, which does not come with CRC at this time. The following figure shows the format of data transfer.

Figure 28-9 SD card single bus data transfer byte format

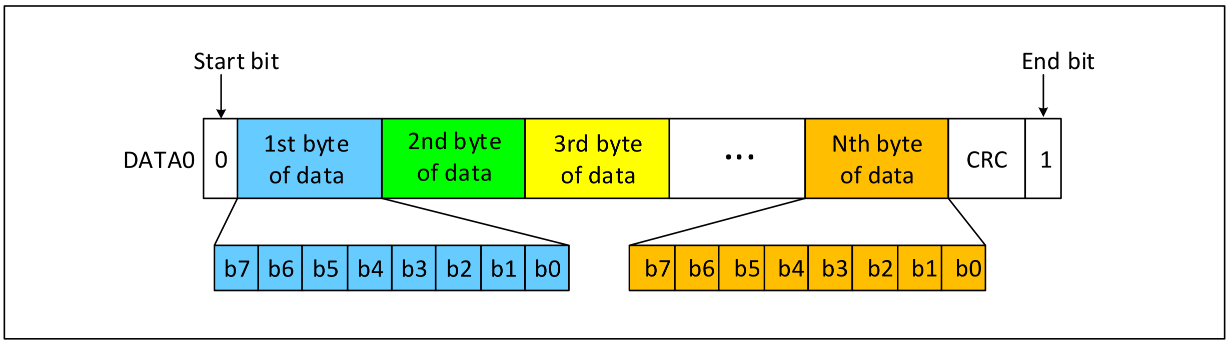


Figure 28-10 SD card 4 bus data transfer byte format

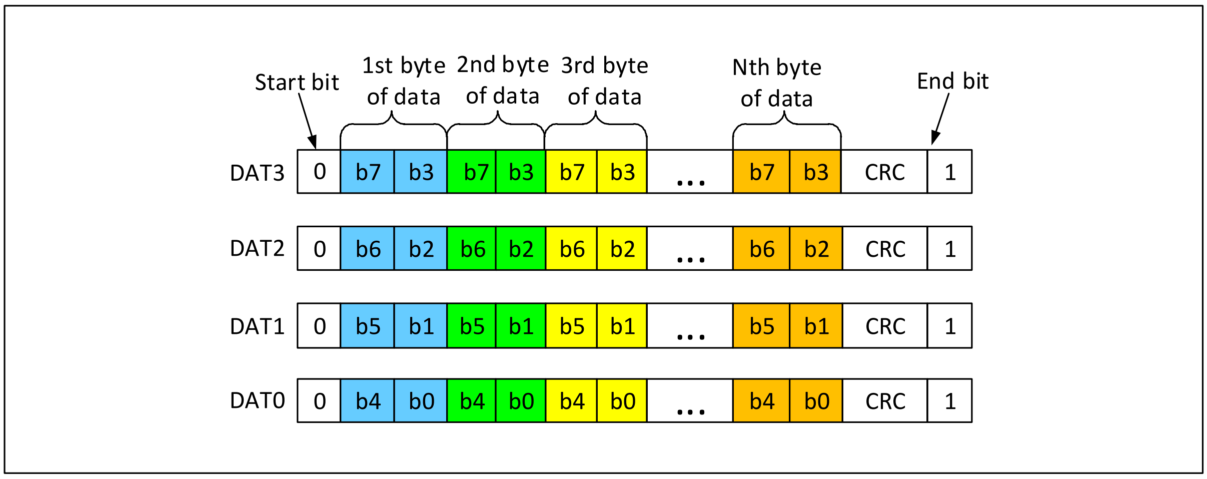

Start bit
1st byte of data
2nd byte of data
3rd byte of data
Nth byte of data
End bit
DAT3 0 b7 b3 b7 b3 b7 b3 … b7 b3 CRC 1
DAT2 0 b6 b2 b6 b2 b6 b2 … b6 b2 CRC 1
DAT1 0 b5 b1 b5 b1 b5 b1 … b5 b1 CRC 1
DAT0 0 b4 b0 b4 b0 b4 b0 … b4 b0 CRC 1

Figure 28-11 SD card single bus data transfer in a 512-bit word format

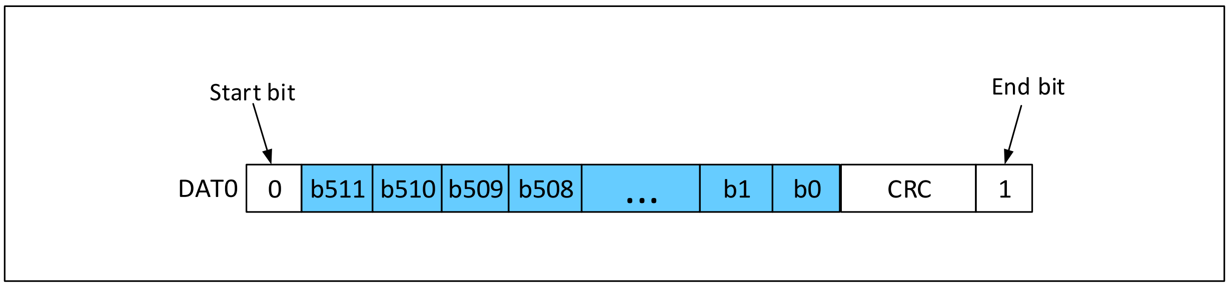
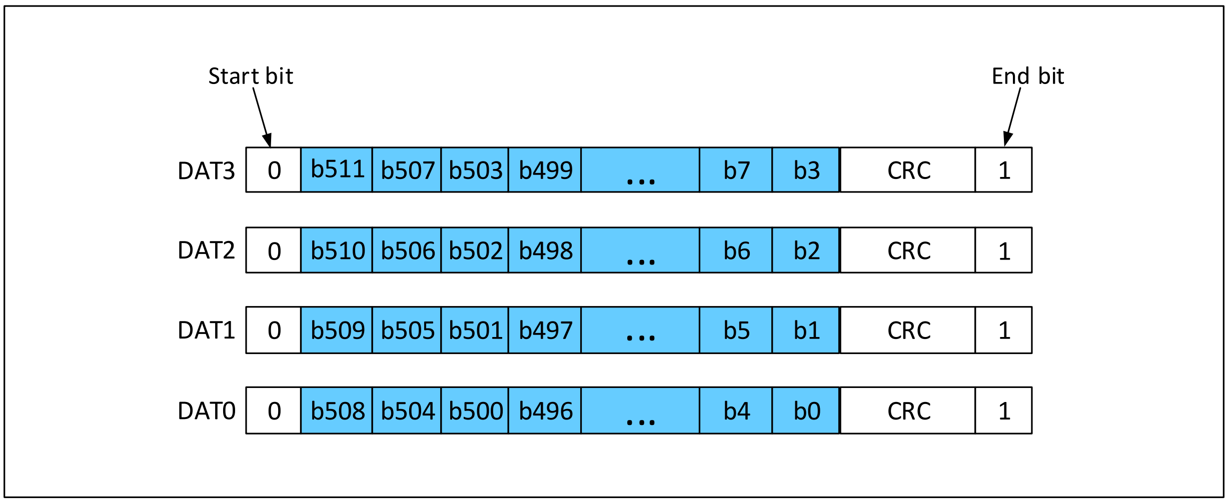


Figure 28-12 SD card 4 bus data transfer in a 512-bit word format



28.4 Application

28.4.1 Device initialization and device registers

28.4.1.1 OCR

The Operation Conditions Register stores some information about the power supply voltage and related status bits of the SD card. The relevant bits are defined in the following table.

Table 28-14 OCR bit definitions

Bit	Bit definition	Description
0-3	Reserved	Supported VDD voltage range in volts.
4	Reserved	
5	Reserved	
6	Reserved	
7	Reserved for low voltage range	
8	Reserved	
9	Reserved	
10	Reserved	
11	Reserved	
12	Reserved	
13	Reserved	
14	Reserved	
15	2.7-2.8	
16	2.8-2.9	
17	2.9-3.0	
18	3.0-3.1	
19	3.1-3.2	
20	3.2-3.3	
21	3.3-3.4	
22	3.4-3.5	
23	3.5-3.6	
24	accept switching to 1.8V	

25-26	Reserved	
27	Over 2TB support status bit	
28	Reserved	
29	UHS-II card status bits	This bit is set to indicate that the card supports the UHS-II interface.
30	Card capacity status (CCS)	This bit is set to indicate that the card capacity is greater than 2GB.
31	Card power-on status bit (busy)	This bit is set when the card is powered up. After power-on, other bits have meaning.

28.4.1.2 CID

CID stores some identifying information.

Table 28-15 CID bit definitions

Bit/Field name	Abbreviation	Width	Bit position
Manufacturer ID	MID	8	[127:120]
App ID	OID	16	[119:104]
Product Name	PNM	40	[103:64]
Product Revision	PRV	8	[63:56]
Product Serial Number	PSN	32	[55:24]
Reserved	None	4	[23:20]
Date of Manufacture	MDT	12	[19:8]
CRC7	CRC	7	[7:1]
Fixed bit, always 1	None	1	[0]

28.4.1.3 CSD

The CSD register stores the characteristic data of the SD card. Taking the most commonly used SDHC and SDXC cards of the second version of the CSD as an example, the definitions of each field are as follows.

Table 28-16 CSD bit definitions

Name	Abbreviation	Width	Value	Read and write	Bit position
CSD version	CSD_STRUCTURE	2	01b	RO	[127:126]
Reserved	None	6	00_0000b	RO	[125120
Reading access time	TAAC	8	0Eh	RO	[119:112]
Reading access time in clock cycles	NSAC	8	00h	RO	[111:104]
Maximum data transmission speed	TRAN_SPEED	8	32h5Ah0Bh2Bh	RO	[103:96]
Card command class	CCC	12	X1X1101101X1b	RO	[95:84]

Maximum length of read data block	READ_BL_LEN	4	9	RO	[83:80]
Allow block partial read	READ_BL_PARTIAL	1	0	RO	[79]
block write misalignment	WRITE_BLK_MISALIGN	1	0	RO	[78]
block read misaligned	READ_BLK_MISALIGN	1	0	RO	[77]
Executed DSR	DSR_IMP	1	X	RO	[76]
Reserved	None	6	00_0000b	RO	[75:70]
Device size	C_SIZE	22	XXXXXXh	RO	[69:48]
Reserved	None	1	0	RO	[47]
Single block erase enable	ERASE_BLK_EN	1	1	RO	[46]
Erase sector size	SECTOR_SIZE	7	7Fh	RO	[45:39]
Write-protect group size	WP_GRP_SIZE	7	0000000b	RO	[38:32]
Write Protect Group Enable	WP_GRP_ENABLE	1	0	RO	[31]
Reserved	None	2	00b	RO	[30:29]
Write speed factor	R2W_FACTOR	3	010b	RO	[28:26]
Maximum write data block length	WRITE_BL_LEN	4	9	RO	[25:22]
Allow block partial writes	WRITE_BL_PARTIAL	1	0	RO	[21]
Reserved	None	5	00000b	RO	[20:16]
file format group	FILE_FORMAT_GRP	1	0	RO	[15]
Copy sign	TMP_WRITE_PROTECT	1	X	RW OTP	[14]
Permanent write protection	PERM_WRITE_PROTECT	1	X	RW OTP	[13]
Temporary write protection	TMP_WRITE_PROTECT	1	X	RW	[12]
File format	FILE_FOMAT	2	00b	RO	[11:10]
Reservef	None	2	00b	RO	[9:8]
CRC	CRC	7	0000000b	RW	[7:1]
Not used, must use 1	None	1	1b	RO	[0]

28.4.1.4 RCA

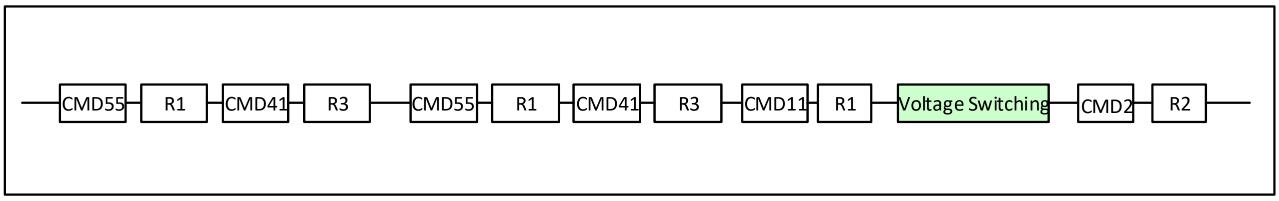
The relative card address register stores the address of the card, which is 16 bits, and the default value is 0.

28.4.2 Voltage switch

In the later stage of SD card initialization, it is necessary to switch the interface level, and switch the IO level of the clock line, data line and command line of the SD card to the 1.8V level. For devices whose slew rate is not good enough, using a lower-level standard can help increase the frequency. However, the power supply voltage of SD does not necessarily change, and only SD cards with low voltage power supply appear in the newer version of the protocol.

The steps for switching the voltage are shown in the figure below.

Figure28-13 Voltage switch sequence



28.4.3 Clock switch

During initialization, the clock of the SD card is only 400KHz. After the voltage switch is completed, the clock can be increased to a higher level. For example, the first speed of the SDHC card UHS-I mode, the bus clock can reach 80MHz. In view of the IO output capability of the microcontroller, the clock should be limited to 50MHz.

28.5 Interrupt

28.5.1 SDIO interrupt

SDIO supports a variety of interrupt sources. As shown in the interrupt enable register (R32_SDIO_IER), there are 24 situations that can trigger interrupts, and users can enable them as appropriate.

28.5.2 SDIO device disconnect

It should be noted that not only SDIO peripherals can report interrupts to the CPU, but also external SDIO cards and MMC cards can report interrupts to SDIO peripherals. In 4-bit bus mode, the interrupt line is D1, and in 8-bit bus mode, the interrupt line is D7, active low. If SDIO detects that D1 or D7 is low level in the idle state, it should read the status register or interrupt flag register of the SDIO device to respond to the interrupt in time. The CPU can get whether the SDIO host receives an interrupt through R32_SDIO_SR: 22 bits.

28.6 Register description

Table 28-17 SDIO registers

Name	Address	Description	Reset Value
R32_SDIO_POWER	0x40018000	Power register	0x00000000
R32_SDIO_CLKCR	0x40018004	Clock register	0x00000000

R32_SDIO_ARG	0x40018008	Command parameter register	0x00000000
R32_SDIO_CMD	0x4001800C	Command register	0x00000000
R32_SDIO_RESPCMD	0x40018010	Response register	0x00000000
R128_SDIO_RESPX	0x40018014	Response parameter register	0x00000000
R32_SDIO_DTIMER	0x40018024	Data timing register	0x00000000
R32_SDIO_DLEN	0x40018028	Data length register	0x00000000
R32_SDIO_DCTLR	0x4001802C	Data control register	0x00000000
R32_SDIO_DCOUNT	0x40018030	Data count register	0x00000000
R32_SDIO_STA	0x40018034	Status register	0x00000000
R32_SDIO_ICR	0x40018038	Interrupt clear register	0x00000000
R32_SDIO_MASK	0x4001803C	Interrupt enable register	0x0000_0000
R32_SDIO_ FIFOCNT	0x40018048	FIFO counter	0x0000_0000
R32_SDIO_FIFO	0x40018080	FIFO register	0x0000_0000

28.6.1 Power Register (R32_SDIO_POWER)

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														PWRCTRL	

Bit	Name	Access	Description	Reset value
[31:2]	Reserved	RO	Reserved	0
[1:0]	PWRCTRL	RW	Power control: 00: Power down, clock stops; 01: Reserved; 10: Reserved power-on status; 11: Power-on, the card clock is turned on;	0

28.6.2 Clock Register (R32_SDIO_CLKCR)

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
Res	HWFC_EN	NEGEDGE	WIDBUS		BYPASS	PWRSAV	CLKEN	CLKDIV							

Bit	Name	Access	Description	Reset Value
[31:15]	Reserved	RO	Reserved	0
[14]	HWFC_EN	RW	Hardware flow control enable. TXFIFOE and RXFIFOF signals have effect after this bit is set. 0: Disable hardware flow control; 1: Enable hardware flow control.	0
[13]	NEGEDGE	RW	SDIO_CK phase selection.	0

			0: SDIO_CK is generated on the rising edge of HCLK; 1: SDIO_CK is generated on the falling edge of HLCK.	
[12:11]	WIDBUS	RW	Bus width configuration. 00: 1-bit bus mode, use SDIO_D0; 01: 4-bit bus mode, use SDIO_D[3:0]; 10: 8-bit bus mode, use SDIO_D[7:0]; 11: Not used.	0
[10]	BYPASS	RW	Clock bypass enable. 0: SDIO_CK is obtained after divider; 1: SDIO_CK is directly connected to HCLK/2.	0
[9]	PWRSAV	RW	Free clock state configuration. After this bit is set, SDIO_CK output is disabled when bus is free, to save power. 0: SDIO_CK is always output; 1: SDIO_CK is only output when needed.	0
[8]	CLKEN	RW	Clock enable. 0: SDIO_CK output is disabled; 1: SDIO_CK output is enabled.	0
[7:0]	CLKDIV	RW	Clock frequency division factor. This field represents the relationship between SDIO_CK and HCLK. SDIO_CK=HCLK/(CLKDIV+2). Note that SDIO_CK should be lower than 400KHz during initialization.	0

Note: The clock configuration register is used to control SDIO_CK related parameters. It should be noted that this register cannot be changed during the period of reading and writing data to 7 HCLK cycles.

28.6.3 Command Parameter Register (R32_SDIO_ARG)

Offset address: 0x08

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

Bit	Name	Access	Description	Reset Value
[31:0]	CMDARG	RW	Command parameter field. This field stores the parameters in the command, which will be sent to the CMD line as part of the command.	0

28.6.4 Command Register (R32_SDIO_CMD)

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
Res	ATACMD	NIEN	ENCMDCompl	SDIOSuspend	CPSMEN	WAITPEND	WAITINT	WAITRESP		CMDINDEX					

Bit	Name	Access	Description	Reset Value
[31:15]	Reserved	RO	Reserved	0
[14]	ATACMD	RW	Carry out CE-ATA command. If this bit is set, CPSM will go to CMD61.	0
[13]	NIEN	RW	Not enable CE-ATA interrupt setting. If this bit is set, CE-ATA will not generate an interrupt.	0
[12]	ENCMDCompl	RW	Enable CMD completed signal enable. If this bit is set, command completion will be signaled.	0
[11]	SDIOSuspend	RW	Suspend command. If this bit is set, a suspend signal will be sent. Applies to SDIO cards only.	0
[10]	CPSMEN	RW	CPSM (Command Path State Machine) enable. If set, CPSM is enabled.	0
[9]	WAITPEND	RW	Command wait control. If this bit is set, the CPSM will wait for the data transfer to complete before sending the command.	0
[8]	WAITINT	RW	Command wait interrupt control. If this bit is set, the CPSM will turn off the timeout control and wait for an interrupt to be generated.	0
[7:6]	WAITRESP	RW	Response type. This field indicates the type of response the CPSM expects to receive. 00: No response, wait for the CMDSENT flag; 01: Short response, wait for CMDREND or CCRCFAIL flag; 10: No response, wait for the CMDSENT flag; 11: Long response, wait for CMDREND or CCRCFAIL flag;	0
[5:0]	CMDINDEX	RW	Command index. This field indicates the specific command value.	0

28.6.5 Response Register (R32_SDIO_RESPCMD)

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										RESPCMD					

Bit	Name	Access	Description	Reset Value
[31:6]	Reserved	RO	Reserved	0
[5:0]	RESPCMD	RO	This field records the index value of the received response.	000000b

28.6.6 Response Parameter Register (R128_SDIO_RESPX)

28.6.6.1 Response Parameter Register High 32 Bits (R128_SDIO_RESP1[127:96])

Offset address: 0x14

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CARDSTATUS1															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARDSTATUS1															

28.6.6.2 Response Parameter Register Second High 32 Bits (R128_SDIO_RESP2[95:64])

Offset address: 0x18

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CARDSTATUS2															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARDSTATUS2															

28.6.6.3 Response Parameter Register Second Low 32 Bits (R128_SDIO_RESP3[63:32])

Offset address: 0x1C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CARDSTATUS3															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARDSTATUS3															

28.6.6.4 Response Parameter Register Low 32 Bits (R128_SDIO_RESP4[31:0])

Offset address: 0x20

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CARDSTATUS4															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARDSTATUS4															

Bit	Name	Access	Description	Reset Value
[128:0]	CARDSTATUSx	RO	When the response is a long response, the 128 bits represent the card status. When the response is a short response, the lower 32 bits represent the card status. The SDIO peripheral first receives the highest bit of the card state and stores it from the lowest bit of R128_SDIO_RESX.	0

28.6.7 Data Timing Register (R32_SDIO_DTIMER)

Offset address: 0x24

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

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

Bit	Name	Access	Description	Reset Value
[31:0]	DATATIME	RW	Data timeout period. SDIO_CK cycle as the unit.	0

28.6.8 Data Length Register (R32_SDIO_DLEN)

Offset address: 0x28

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

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

Bit	Name	Access	Description	Reset Value
[31:25]	Reserved	RO	Reserved	0
[24:0]	DATALENGTH	RW	Data length. The value of this field is loaded into the transfer counter when a transfer is started. For block transfers, the value of this field must be an integer multiple of the block size. The block size is defined by the SDIO device and stored in R32_SDIO_DCTLR[7:4]. Common values are 512B, etc.	0

28.6.9 Data Control Register (R32_SDIO_DCTRL)

Offset address: 0x2C

31	30	29	28	27	26	25	24	23	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				SDIOEN	RWMOD	RWSTOP	RWSTART	DBLOCKSIZE				DMAEN	DTMODE	DTDIR	DTEN

Bit	Name	Access	Description	Reset Value
[31:12]	Reserved	RO	Reserved	0
[11]	SDIOEN	RW	SDIO enable. After this bit is set, the DPSM can operate.	0
[10]	RWMOD	RW	Read wait mode. 0: Stop SDIO_CK control read wait; 1: Use SDIO_D2 to control read wait.	0
[9]	RWSTOP	RW	Read wait stop. If the RWSTART bit is set, the read wait will be stopped.	0
[8]	RWSTART	RW	Read wait start. Setting this bit will perform a read wait operation.	0
[7:4]	DBLOCKSIZE	RW	Data block size. This field stores the size of the data block, and the block transfer length must be	0

			defined before using the block transfer. The value that can be written in this field is between 0 and 1110b, which indicates that the block transfer length is 2^{BLKLEN} (between 0 and 16384 bytes).	
[3]	DMAEN	RW	DMA enable. Setting this bit enables DMA.	0
[2]	DTMODE	RW	Data mode. Set this bit to configure the transfer mode. 0: Block transfer; 1: Stream transfer.	0
[1]	DTDIR	RW	Data direction. Set this bit to configure the transfer direction. 0: From controller to card; 1: From card to controller.	0
[0]	DTEN	RW	Data enable. Set this bit to start data transfer. Specific process: after setting this bit, DPSM enters the process of Wait_S or Wait_R (depending on the data direction).	0

28.6.10 Data Count Register (R32_SDIO_DCOUNT)

Offset address: 0x30

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

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

Bit	Name	Access	Description	Reset Value
[31:25]	Reserved	RO	Reserved	0
[24:0]	DATACOUNT	RO	Data count. The value of the transmission length register will be loaded into this counter when a transfer is initiated and decremented as the transfer progresses.	0

28.6.11 Status Register (R32_SDIO_STA)

Offset address: 0x34

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								CEATAEND	SDIOIT	RXDAVL	TXDAVL	RXFIFOE	TXFIFOE	RXFIFOF	TXFIFOF

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXFIFOHF	TXFIFOHE	RXACT	TXACT	CMDACT	DBCKEND	STBITERR	DATAEND	CMDSENT	CMDREND	RXOVERR	TXUNDERR	DTIMEOUT	CTIMEOUT	DCRCFAIL	CCRCFAIL

Bit	Name	Access	Description	Reset Value
[31:24]	Reserved	RO	Reserved	0
[23]	CEATAEND	RO	When this bit is set, CMD61 receives the CE-ATA	0

			completed signal.	
[22]	SDIOIT	RO	When this bit is set, SDIO receives a device interrupt.	0
[21]	RXDAVL	RO	When this bit is set, data in the Rx FIFO is valid.	0
[20]	TXDAVL	RO	When this bit is set, data in the Tx FIFO is valid.	0
[19]	RXFIFOE	RO	When this bit is set, the Rx FIFO is empty.	0
[18]	TXFIFOE	RO	When this bit is set, the Tx FIFO is empty.	0
[17]	RXFIFOF	RO	When this bit is set, the Rx FIFO is full.	0
[16]	TXFIFOF	RO	When this bit is set, the Tx FIFO is full.	0
[15]	RXFIFOHF	RO	When this bit is set, the Rx FIFO is half full.	0
[14]	TXFIFOHE	RO	When this bit is set, the Tx FIFO is half empty.	0
[13]	RXACT	RO	When this bit is set, data is being received.	0
[12]	TXACT	RO	When this bit is set, data is being transmitted.	0
[11]	CMDACT	RO	When this bit is set, a command is being transmitted.	0
[10]	DBCKEND	RO	When this bit is set, the data block has been sent or received and the CRC has passed.	0
[9]	STBITERR	RO	When this bit is set, no start signal is detected on all data lines in wide bus mode.	0
[8]	DATAEND	RO	When this bit is set, the data transfer is complete (the transfer counter is zero).	0
[7]	CMDSENT	RO	When this bit is set, the command has been sent.	0
[6]	CMDREND	RO	When this bit is set, the response has been received and the CRC is successful.	0
[5]	RXOVERR	RO	When this bit is set, the Rx FIFO overflows.	0
[4]	TXUNDERR	RO	When this bit is set, the transmit FIFO underflows.	0
[3]	DTIMEOUT	RO	When this bit is set, the data times out.	0
[2]	CTIMEOUT	RO	When this bit is set, the command timeout exceeds 64 SDIO_CK cycles.	0
[1]	DCRCFAIL	RO	When this bit is set, a data block has been sent or received but the CRC failed.	0
[0]	CCRCFAIL	RO	When this bit is set, the response has been received but the CRC has failed.	0

28.6.12 Interrupt Clear Register (R32_SDIO_ICR)

Offset address: 0x38

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					DBCKENDC	STBITERRC	DATAENDC	CMDSENTC	CMDRENDC	RXOVERRC	TXUNDERRC	DTIMEOUTC	CTIMEOUTC	DCRCFAILC	CCRCFAILC

Bit	Name	Access	Description	Reset Value
[31:24]	Reserved	RO	Reserved	0
[23]	CEATAENDC	RW	Setting this bit clears the CEATAEND bit in the	0

			status register.	
[22]	SDIOITC	RW	Setting this bit clears the SDIOIT bit in the status register.	0
[21:11]	Reserved	RW	Reserved	0
[10]	DBCKENDC	RW	Setting this bit clears the DBCKEND bit in the status register.	0
[9]	STBITERRC	RW	Setting this bit clears the STBITERR bit in the status register.	0
[8]	DATAENDC	RW	Setting this bit clears the DATAEND bit in the status register.	0
[7]	CMDSENTC	RW	Setting this bit clears the CMDSENT bit in the status register.	0
[6]	CMDRENDC	RW	Setting this bit clears the CMDREND bit in the status register.	0
[5]	RXOVERRC	RW	Setting this bit clears the RXOVERR bit in the status register.	0
[4]	TXUNDERRC	RW	Setting this bit clears the TXUNDERR bit in the status register.	0
[3]	DTIMEOUTC	RW	Setting this bit clears the DTIMEOUT bit in the status register.	0
[2]	CTIMEOUTC	RW	Setting this bit clears the CTIMEOUT bit in the status register.	0
[1]	DCRCFAILC	RW	Setting this bit clears the DCRCFAIL bit in the status register.	0
[0]	CCRCFAILC	RW	Setting this bit clears the CCRCFAIL bit in the status register.	0

28.6.13 Interrupt Enable Register (R32_SDIO_MASK)

Offset address: 0x3C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								CEATAENDIE	SDIOITIE	RXDAVLIE	TXDAVLIE	RXFIFOEIE	TXFIFOEIE	RXFIFOFIE	TXFIFOFIE

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXFIFOHFIE	TXFIFOHEIE	RXACTIE	TXACTIE	CMDACTIE	DBCKENDIE	STBITERRIE	DATAENDIE	CMDSENTIE	CMDRENDIE	RXOVERRIE	TXUNDERRIE	DTIMEOUTIE	CTIMEOUTIE	DCRCFAILIE	CCRCFAILIE

Bit	Name	Access	Description	Reset Value
[31:24]	Reserved	RO	Reserved	0
[23]	CEATAENDIE	RW	After this bit is set, the status register will generate an interrupt when the CEATAEND bit is set.	0
[22]	SDIOITIE	RW	After this bit is set, the status register will generate an interrupt when the SDIOIT bit is set.	0
[21]	RXDAVLIE	RW	After this bit is set, the status register will generate an	0

			interrupt when the RXDAVL bit is set.	
[20]	TXDAVLIE	RW	After this bit is set, the status register will generate an interrupt when the TXDAVL bit is set.	0
[19]	RXFIFOEIE	RW	After this bit is set, the status register will generate an interrupt when the RXFIFOE bit is set.	0
[18]	TXFIFOEIE	RW	After this bit is set, the status register will generate an interrupt when the TXFIFOE bit is set.	0
[17]	RXFIFOFIE	RW	After this bit is set, the status register will generate an interrupt when the RXFIFOF bit is set.	0
[16]	TXFIFOFIE	RW	After this bit is set, the status register will generate an interrupt when the TXFIFOF bit is set.	0
[15]	RXFIFOHFIE	RW	After this bit is set, the status register will generate an interrupt when the RXFIFOHF bit is set.	0
[14]	TXFIFOHEIE	RW	After this bit is set, the status register will generate an interrupt when the TXFIFOHE bit is set.	0
[13]	RXACTIE	RW	After this bit is set, the status register will generate an interrupt when the RXACT bit is set.	0
[12]	TXACTIE	RW	After this bit is set, the status register will generate an interrupt when the TXACT bit is set.	0
[11]	CMDACTIE	RW	After this bit is set, the status register will generate an interrupt when the CMDACT bit is set.	0
[10]	DBCKENDIE	RW	After this bit is set, the status register will generate an interrupt when the DBCKEND bit is set.	0
[9]	STBITERRIE	RW	After this bit is set, the status register will generate an interrupt when the STBITERR bit is set.	0
[8]	DATAENDIE	RW	After this bit is set, the status register will generate an interrupt when the DATAEND bit is set.	0
[7]	CMDSENTIE	RW	After this bit is set, the status register will generate an interrupt when the CMDSENT bit is set.	0
[6]	CMDRENDIE	RW	After this bit is set, the status register will generate an interrupt when the CMDREND bit is set.	0
[5]	RXOVERRIE	RW	After this bit is set, the status register will generate an interrupt when the RXOVERR bit is set.	0
[4]	TXUNDERRIE	RW	After this bit is set, the status register will generate an interrupt when the TXUNDERR bit is set.	0
[3]	DTIMEOUTIE	RW	After this bit is set, the status register will generate an interrupt when the DTIMEOUT bit is set.	0
[2]	CTIMEOUTIE	RW	After this bit is set, the status register will generate an interrupt when the CTIMEOUT bit is set.	0
[1]	DCRCFAILIE	RW	After this bit is set, the status register will generate an interrupt when the DCRCFAIL bit is set.	0
[0]	CCRCFAILIE	RW	After this bit is set, the status register will generate an interrupt when the CCRCFAIL bit is set.	0

28.6.14 FIFO Count Register (R32_SDIO_FIFOCNT)

Offset address: 0x48

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

Bit	Name	Access	Description	Reset Value
[31:0]	FIFOCOUNT	RO	The FIFO contains the number of words (32 bits) that have not been written to or read from the FIFO. When setting R32_SDIO_DCTLR: R32_SDIO_DCTLR, if the DPSM is idle, the FIFO counter will load the transfer length value from R32_SDIO_TLEN, if this value is not divisible by 4, the last 1 to 3 bytes will be treated as a word.	0

28.6.15 FIFO (R32_SDIO_FIFO)

Offset address: 0x80

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

Bit	Name	Access	Description	Reset Value
[31:0]	FIFOData	RW	FIFO data field. Reading and writing this field will read the received data or send the pending data. The SDIO FIFO has a total of 32 words (one word is 32 bits).	0

Chapter 29 Random Number Generator (RNG)

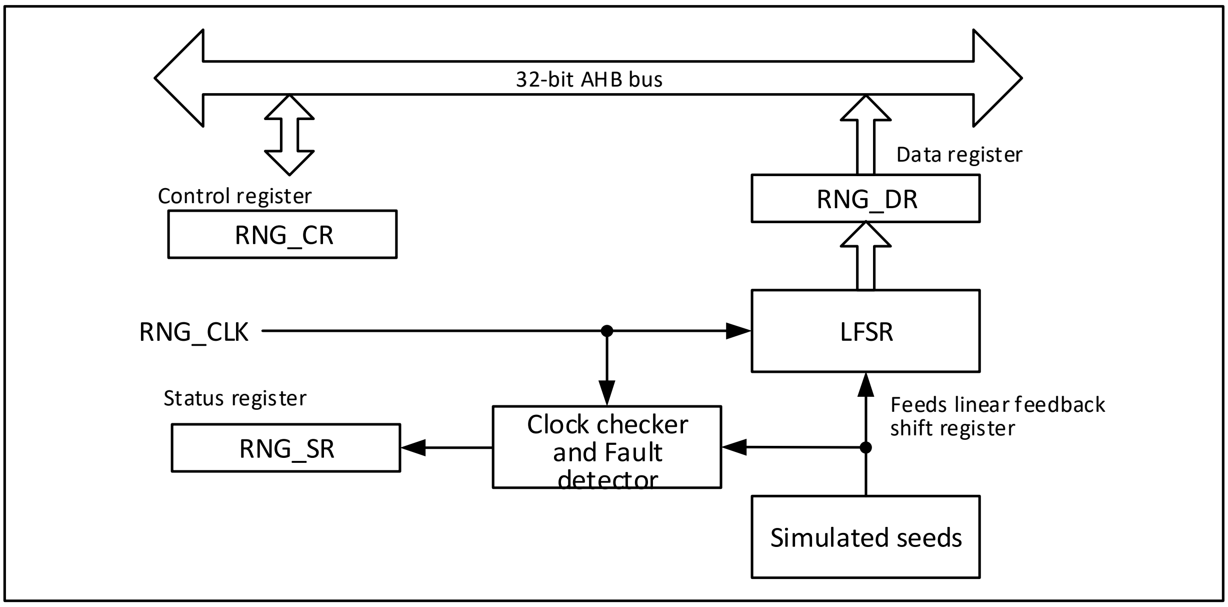
The module description in this chapter applies to the full range of CH32F2x and CH32V3x microcontrollers

Random number generator (RNG), based on continuous analog noise, provides 32-bit random numbers when the host reads data

29.1 Main features

- Can generate 32-bit random numbers
- Error management is possible
- Can be disabled individually to reduce power consumption

Figure 29-1 RNG block diagram



29.2 Functional description

The random transmitter is implemented using an analog circuit that generates a seed for a linear feedback shift register (RNG_LFSR) for generating 32-bit random numbers. RNG_LFSR provides clock information at a constant frequency by a dedicated clock (PLL48CLK), so the quality of random numbers is related to the HCLK clock. When a large number of seeds are introduced into RNG_LFSR, the content of RNG_LFSR will be transferred to the data register (RNG_DR).

29.2.1 RNG operations

RNG operation procedure:

1) If the interrupt is enabled, by setting the IE bit in the RNG_CR register (the interrupt is generated when a random number is ready or an error occurs).
2) Random number generation is enabled by configuring the RNGEN bit in the RNG_CR register, while the analog section, RNG_LFSR and error detector are activated.
3) If the interrupt is enabled, every time an interrupt is generated, the SEIS and CEIS bits in the RNG_SR register are 0 to confirm that no error occurs and that DRDY is 1 to confirm that the random number is

ready. The contents of the RNG_DR register can then be read.

29.2.2 Error management

RNG errors include clock errors and seed errors. When a clock error occurs, the RNG can no longer generate random numbers. At this time, it is necessary to check whether the clock controller is configured correctly and whether the RNG clock can be provided, and then clear the CEIS bit. When the CECS bit is 0, the RNG can work normally. When a clock error occurs, it has no effect on the last random number and can be used normally. When a seed error occurs, the value in the RNG_DR register cannot use the random number at this time. If the RNG is to be reused, the SEIS bit needs to be cleared first, then the RNGEN bit is cleared and set to 1, and the RNG is reinitialized and restarted.

29.3 Register description

Table 29-1 OPA registers

Name	Address	Description	Reset value
R32_RNG_CR	0x40023C00	RNG control register	0x00000000
R32_RNG_SR	0x40023C04	RNG status register	0x00000000
R32_RNG_DR	0x40023C08	RNG data register	0x00000000

29.3.1 RNG Control Register (RNG_CR)

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												IE	RNG EN	Reserved	

Bit	Name	Access	Description	Reset value
[31:4]	Reserved	RO	Reserved	0
3	IE	RW	Interrupt enable: 0: Disable RNG interrupt. 1: Enable RNG interrupt.	0
2	RNGEN	RW	Random number generator enable: 0: Disable random number generator. 1: Enable random number generator.	0
[1:0]	Reserved	RW	Reserved	0

29.3.2 RNG Status Register (RNG_SR)

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									SEIS	CEIS	Reserved		SECS	CECS	DRDY

Bit	Name	Access	Description	Reset Value
[31:7]	Reserved	RO	Reserved	0
6	SEIS	RW	Seed error interrupt status (this bit is set at the same time as SECS) 0: No inverted error sequence detected. 1: One of the following error sequences is detected: - more than 64 identical consecutive bits; - more than 32 consecutive alternating 0's and 1's.	0
5	CEIS	RW	Clock error interrupt status (this bit is set at the same time as CECS) 0: PLL48CLK clock detected correctly 1: PLL48CLK clock not detected correctly	0
[4:3]	MODE4	RW	Reserved	0
2	SECS	RO	Seed error current status 0: No error sequence detected; 1: One of the following error sequences was detected: - more than 64 identical consecutive bits; - more than 32 consecutive alternating 0s and 1s.	0
1	CECS	RO	Clock error current status 0: PLL48CLK clock detected correctly 1: PLL48CLK clock not detected correctly	0
0	DRDY	RO	Data ready (this bit is cleared after reading RNG_DR) 0: RNG_DR register is invalid, this random number is not available 1: RNG_DR register is valid, this random number is available	0

29.3.3 RNG Data Register (RNG_DR)

Offset address: 0x08

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

Bit	Name	Access	Description	Reset Value
[31:0]	RNDATA	RO	32-bit random number	0

Chapter 30 Operational Amplifier (OPA)

The module description in this chapter applies to the full range of CH32F2x, CH32V2x and CH32V3xmicrocontrollers

The Operational Amplifier Module (OPA) contains 4 independently configurable operational amplifiers. The input and output of each op-amp are connected to the I/O port, and the input pin can be selected, and the output pin can be optionally configured to the general-purpose I/O port or multiplexed as the I/O of the ADC sampling channel.

30.1 Main features

- Input pin selectable
- The output pin can select general I/O port or ADC sampling channel

30.2 Functional description

Set OPAx_EN to enable the corresponding OPAx, configure OPAx_MODE to select the output channel of OPAx as ADC sampling channel or ordinary I/O port, configure OPAx_PSEL, select the positive input pin of OPAx, configure OPAx_NSEL, select OPAx the negative input pin.

Note: For the detailed input and output pins of each OPA, please refer to the pin description in the datasheet.

30.3 Register description

Table 30-1 OPA register

Name	Address	Description	Reset value
R32_OPA_CTLR	0x40023804	OPA control register	0x00000000

30.3.1 OPA Control Register (OPA_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
PSEL4	NSEL4	MODE4	EN4	PSEL3	NSEL3	MODE3	EN3	PSEL2	NSEL2	MODE2	EN2	PSEL1	NSEL1	MODE1	EN1

Bit	Name	Access	Description	Reset Value
[31:16]	Reserved	RO	Reserved	0
15	PSEL4	RW	OPA4 positive input selection 0: CHP0. 1: CHP1. *Note: Applied for CH32F20x_D8, CH32F20x_D8C, CH32V30x_D8 and CH32V30x_D8C.*	0

14	NSEL4	RW	OPA4 negative input selection 0: CHN0. 1: CHN1. *Note: Applied for CH32F20x_D8, CH32F20x_D8C, CH32V30x_D8 and CH32V30x_D8C.*	0
13	MODE4	RW	OPA4 output channel selection 0: OPA4_OUT0. 1: OPA4_OUT1. *Note: Applied for CH32F20x_D8, CH32F20x_D8C, CH32V30x_D8 and CH32V30x_D8C.*	0
12	EN4	RW	OPA4 enable 0: Disable OPA4. 1: Enable OPA4. *Note: Applied for CH32F20x_D8, CH32F20x_D8C, CH32V30x_D8 and CH32V30x_D8C.*	0
11	PSEL3	RW	OPA3 positive input selection 0: CHP0. 1: CHP1. *Note: Applied for CH32F20x_D8, CH32F20x_D8C, CH32V30x_D8 and CH32V30x_D8C.*	0
10	NSEL3	RW	OPA3 negative input selection 0: CHN0. 1: CHN1. *Note: Applied for CH32F20x_D8, CH32F20x_D8C, CH32V30x_D8 and CH32V30x_D8C.*	0
9	MODE3	RW	OPA3 output channel selection 0: OPA3_OUT0. 1: OPA3_OUT1. *Note: Applied for CH32F20x_D8, CH32F20x_D8C, CH32V30x_D8 and CH32V30x_D8C.*	0
8	EN3	RW	OPA3 enable 0: Disable OPA3. 1: Enable OPA3. *Note: Applied for CH32F20x_D8, CH32F20x_D8C, CH32V30x_D8 and CH32V30x_D8C.*	0
7	PSEL2	RW	OPA2 positive input selection 0: CHP0. 1: CHP1.	0
6	NSEL2	RW	OPA2 negative input selection 0: CHN0. 1: CHN1.	0
5	MODE2	RW	OPA2 output channel selection 0: OPA2_OUT0. 1: OPA2_OUT1.	0
4	EN2	RW	OPA2 enable 0: Disable OPA2.	0

			1: Enable OPA2.	
3	PSEL1	RW	OPA1 positive input selection 0: CHP0. 1: CHP1.	0
2	NSEL1	RW	OPA1 negative input selection 0: CHN0. 1: CHN1.	0
1	MODE1	RW	OPA1 output channel selection 0: OPA1_OUT0. 1: OPA1_OUT1.	0
0	EN1	RW	OPA1 enable 0: Disable OPA1. 1: Enable OPA1.	0

Chapter 31 Electronic Signature (ESIG)

The module description in this chapter applies to the full range of CH32F2x, CH32V2x and CH32V3x microcontrollers.

The electronic signature contains chip identification information: The capacity of the flash memory area and the unique identification. It is programmed into the system storage area of the memory module by the manufacturer at the factory, and can be read by SWD (SDI) or application code.

31.1 Functional description

Flash memory area capacity: Indicates the available size of the current chip user application program.

Unique ID: 96-bit binary code, unique to any microcontroller; users can only read and access but cannot modify it. This unique identification information can be used as the security password, encryption 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 conduct read access according to 8/16/32 bits。

31.2 Register description

Table 31-1 ESIG registers

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

31.2.1 Flash Capacity Register (ESIG_FLACAP)

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

Bit	Name	Access	Description	Reset Value
[15:0]	FLASHSIZE	RO	Flash capacity, in unit of Kbyte Example: 0x0080 = 128 Kbytes	X

31.2.2 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 to 31 of UID	X

31.2.3 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	Descriptor	Reset Value
[31:0]	U_ID[63:32]	RO	Bits 32 to 63 of UID	X

31.2.4 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	Descriptor	Reset Value
[31:0]	U_ID[95:64]	RO	Bits 64 to 95 of UID	X

Chapter 32 Flash Memory and User Option Bytes

The module description in this chapter applies to the full range of CH32F2X, CH32V2X and CH32V3X microcontrollers.

32.1 Flash memory organization

The internal flash memory organization structure of the chip is as follows (taking xVCT6 as an example):

Table 32-1 Flash memory organization structure

Block	Name	Address Range	Size(byte)
Main memory	Page 0	0x08000000 to 0x080000FF	256
	Page 1	0x08000100 to 0x080001FF	256
	Page 2	0x08000200 to 0x080002FF	256
	Page 3	0x08000300 to 0x080003FF	256
	Page 4	0x08000400 to 0x080004FF	256
	Page 5	0x08000500 to 0x080005FF	256
	Page 6	0x08000600 to 0x080006FF	256
	Page 7	0x08000700 to 0x080007FF	256
	…	…	…
	Page 1919	0x08077F00 to 0x08077FFF	256
Information block	System boot loader storage	0x1FFF8000 to 0x1FFFEFFF	28K
	User option bytes	0x1FFFF800 to 0x1FFFF87F	128

Note:

1) The main memory area is used for the user's application program storage, and the write protection is divided in unit of 4K bytes (32 pages). Except for the locked "Vendor Configuration Word" area before delivery which is inaccessible to users, other areas can be operated by users under certain conditions.

2) When carrying out FLASH-related operations, it is strongly recommended that the system main frequency is not greater than 120M.

If the actual application must require the use of system mains frequency greater than 120M, it should be noted that:

In the non-zero-wait area FLASH and zero-wait area FLASH, user word read/write and vendor configuration word and Boot area read, the following operations need to be done, firstly, HCLK is divided into two (related peripheral clocks are also divided at the same time, the impact needs to be evaluated), and then restored after the FLASH operation is completed to ensure that the FLASH access clock frequency does not exceed 60Mhz (FLASH_ CTLR register bit[25]-SCKMOD can configure FLASH access clock frequency as system clock or half of system clock, the default configuration of this bit is half of system clock).

32.2 Flash memory programming and safety

32.2.1 Two program / erase methods

- Standard programming: This method is the default programming method (compatible method). In this mode, the CPU executes programming in a single 2-byte manner, and executes erasure and entire chip erasure operations in a single 4K byte

- Fast programming: The page operation mode (recommended) is used for this method. After unlocking in a specific sequence, a single 256-byte programming and 256-byte erasing are performed, 32K-byte erasing, 64K-byte erasing and whole chip erasure are performed.

32.2.2 Security-preventing against Illegal access (read, write and erase)

- Page write protection
- Read protection

Under the read protection state:

1） The main memory pages 0-15 (4K bytes) are automatically write-protected and are not controlled by the FLASH_WPR register; when the read protection status is released, all main memory pages will be controlled by the FLASH_WPR register.

2） The main memory cannot be erased or programmed in the system boot code area, SWD or SDI mode, and RAM area, except for the entire chip erasure. User-selected word area can be erased or programmed. If you try to release the read protection (program user word), the chip will automatically erase the entire user area

Note: When programming/erasing operations of flash memory are made, the internal RC oscillator (HSI) must be switched on.

32.3 Flash enhanced read mode

The FLASH enhanced read mode is suitable for the user program running in FLASH (the user code space exceeds the CODE size space configured by the user-selected SRAM_CODE_MODE [1:0] bits). Turning on this mode can improve the FLASH access efficiency. To enable this mode, set the ENHANCE_MOD bit in the FLASH_CTLR register to 1. To disable this mode, you must first clear the ENHANCE_MOD bit to 0, and then set RSEN_ACT to 1. At the same time, the access clock frequency can be selected by configuring the SCK_MOD bit of the FLASH_CTLR register.

Note: When using FLASH Enhanced Read Mode, the following points should be noted:

1） Before performing any mode of erasing or programming on FLASH (including unlocking user word programming such as read protection), you must exit the enhanced read mode, otherwise the erasing and programming operations will fail;

2） Before entering the stop mode, you must exit the enhanced read mode, otherwise it may cause an abnormal stop mode;

3） After power reset and system reset, the chip is controlled by hardware to automatically exit the enhanced read mode.

32.4 Register description

Table 32-2 FLASH registers

Name	Access address	Description	Reset value
R32_FLASH_KEYR	0x40022004	FPEC key register	X
R32_FLASH_OBKEYR	0x40022008	OBKEY register	X

R32_FLASH_STATR	0x4002200C	Status register	0x00000000
R32_FLASH_CTLR	0x40022010	Control register	0x00000080
R32_FLASH_ADDR	0x40022014	Address register	0x00000000
R32_FLASH_OBR	0x4002201C	Selection word register	0x03FFFFFC
R32_FLASH_WPR	0x40022020	Write protection register	0xFFFFFFF
R32_FLASH_MODEKEYR	0x40022024	Extension key register	X

32.4.1 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

32.4.2 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, used to input the selection word key to release OPTWRE.	X

32.4.3 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
Reserved								EHMODS	Reserved	EOP	WRPRTERR	Reserved	Reserved	WRBSY	BSY

Bit	Name	Access	Description	Reset value
[31:8]	Reserved	RO	Reserved	0
7	EHMODS	R0	FLASH enhanced read mode start: 1: FLASH enhanced read mode is enabled. 0: FLASH enhanced read mode is off.	0
6	Reserved	RO	Reserved	0
5	EOP	RW1	End of operation. Write 1 to clear it. The bit is set by hardware every time it is successfully erased or programmed.	0
4	WRPRTERR	RW1	Write protection error. Write 1 to clear it. The bit is set by hardware when the write protection address is programmed.	0
3	Reserved	RO	Reserved.	0
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, this bit is set to '1' when data is written, and it is automatically cleared to '0' by hardware. If this bit is '0', it means that the next data is allowed to be written.	0
0	BSY	RO	Busy: 1: Flash memory operation is in the process; 0: Operation completion.	0

Note: When performing the programming operation, you need to make sure that the STRT bit in the FLASH_CTLR register is set to 0.

32.4.4 Control Register (FLASH_CTLR)

Offset address: 0x10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
						SCKMOD	EHMOD	Reserved	RSENACT	PGSTRT	Reserved	BER64	BER32	FTER	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	MER	SER	PG

Bit	Name	Access	Description	Reset Value
[31:26]	Reserved	RO	Reserved	0
25	SCKMOD	RW	FLASH access clock configuration: 1: FLASH access clock frequency = SYSCLK. 0: FLASH access clock frequency = SYSCLK/2. *Note: FLASH access clock frequency cannot be more than 60 MHz.*	0

24	EHMOD	RW	FLASH enhanced read mode: This mode can improve access efficiency when the program runs in FLASH. 1: Enable FLASH enhanced read mode; 0: Turn off the FLASH enhanced read mode, which needs to be operated together with the RSEN_ACT bit. To exit, first clear the ENHANCE_MOD bit to 0, and then set RSEN_ACT to 1.	0
23	Reserved	RO	Reserved	0
22	RSENACT	WO	To exit the enhanced read mode, the hardware will automatically clear it. It needs to be operated together with the ENHANCE_MOD bit. To exit, first clear the ENHANCE_MOD bit to 0, and then set RSEN_ACT to 1.	0
21	PGSTRT	RW0	Start. Set to 1 to start a page programming, which is automatically cleared by hardware.	0
20	Reserved	R0	Reserved	0
19	BER64	RW	Perform a 64KB erase.	0
18	BER32	RW	Perform a 32KB erase.	0
17	FTER	RW	Perform a fast page (256Byte) erase operation.	0
16	FTPG	RW	Perform a fast page programming operation.	0
15	FLOCK	RW1	Fast programming lock. Only ‘1’ can be written. When this bit is '1', it means that the fast programming/erasure mode is not available. After detecting the correct unlock sequence, the hardware will clear this bit to ‘0’. Set 1 by software, and relock it	1
[14:13]	Reserved	RO	Reserved	0
12	EOPIE	RW	EOP interrupt enable (EOP is set in the FLASH_STATR register) 1: Enable to generate interrupt; 0: Disable to generate interrupt.	0
11	Reserved	RO	Reserved	0
10	ERRIE	RW	Error status interrupt enable (PGERR/WRPRTERR is set in the FLASH_STATR register): 1: Enable to generate interrupt; 0: Disable to generate interrupt.	0
9	OBWRE	RW0	User option bytes lock; cleared by software: 1: User option bytes can be programmed. It needs to be set by hardware after the correct	0

			sequence is written to the FLASH_OBKEYR register. 0: Re-lock the user option bytes after cleared by software.	
8	Reserved	RO	Reserved	0
7	LOCK	RW1	Lock. Only ‘1’ can be written. When this bit is '1', it means that FPEC and FLASH_CTLR are locked and cannot be written. After detecting the correct unlock sequence, the hardware will clear this bit to ‘0’. After an unsuccessful unlock operation, this bit will not change until the next system reset.	1
6	STRT	RW1	Start. Set to 1 to start an erase/program operation, and the hardware will automatically clear it to 0 (BSY becomes ‘0’).	0
5	OBER	RW	Execute the user option bytes erasure	0
4	OBPG	RW	Execute the user option bytes programming	0
3	Reserved	RO	Reserved	0
2	MER	RW	Execute the whole erasure operation (erasing the whole user area).	0
1	PER	RW	Execute the standard page (4KB) erasure operation.	0
0	PG	RW	Execute the standard programming operation.	0

32.4.5 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[31:0]	WO	Flash address, the programming address when programming, and the erasing start address when erasing. When the BSY bit in the FLASH_SR register is '1', this register cannot be written.	0

32.4.6 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															

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

Reserved	USER Reserved	PORCTR	USBDPU	USBDMODE	STANDYRST	STOPRST	IWDGSW	RDPRT	OBERR

Bit	Name		Access	Description	Reset Value
[31:10]	Reserved		RO	Reserved	0
[9:8]	USER	SRAM_CODE_MODE	RO	00: CODE-192KB + RAM-128KB 01: CODE-224KB + RAM-96KB 10: CODE-256KB + RAM-64KB 11: CODE-288KB + RAM-32KB *Note: Applied for CH32V303RC, CH32V303VC, CH32V307RC, CH32V307WC, CH32V307VC, CH32F203RC, CH32F203VC and CH32F207VC.* 00: CODE-128KB + RAM-64KB 01: CODE-144KB + RAM-48KB 1x: CODE-160KB + RAM-32KB *Note: Applied for CH32V20x_D8W, CH32V20x_D8 and CH32F20x_D8W.*	X
[7:5]		Reserved	RO	Reserved	X
4		STANDY_RST	RO	System reset control in Standby mode.	X
3		STOP_RST	RO	System reset control in Stop mode.	X
2		IWDG_SW	RO	Independent watchdog (IWDG) hardware enable bit.	1
1	RDPRT		RO	Read protection. 1: Current read protection of flash is valid.	1
0	OBERR		RO	Option byte error. 1: Option byte does not match its inverted code.	0

Note: USER and RDPRT are loaded from the user-selected word area after system reset.

32.4.7 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 write protection. 1: Write protection invalid; 0: Write protection valid.	X

			Each bit represents the write protection state of 4-Kbytes (16 pages) memory.	

Note: WPR is loaded from the user-selected word area after system reset.

32.4.8 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

32.5 Flash operation procedure

32.5.1 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.

32.5.2 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.

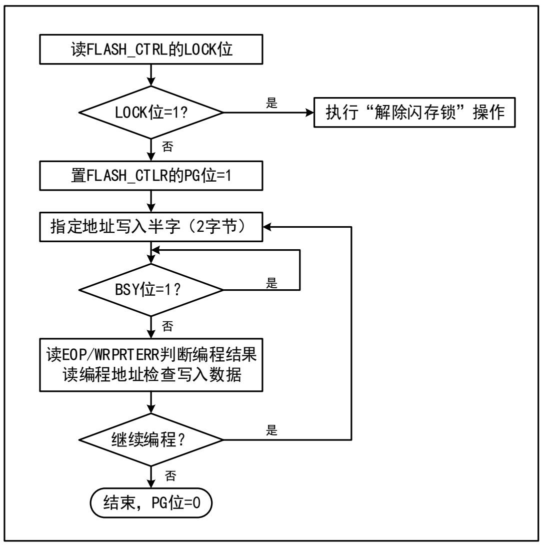
The flash memory controller (FPEC) and the FLASH_CTLR register can be locked again by setting the "LOCK" bit in the FLASH_CTLR register to 1.

32.5.3 Main memory standard programming

You can write 2 bytes each time through the standard programming. When the PG bit in the FLASH_CTLR register is '1', a programming will be started every time a halfword (2 bytes) is written to the flash memory address. When any non-halfword data is written, FPEC will generate a bus error. During the programming process, the BSY bit is ‘1’. After the programming is completed, the BSY bit is ‘0’ and the EOP bit is ‘1’.

Note: When the BSY bit is ‘1’, writing to any register will be disabled.

Figure 32-1 FLASH programming

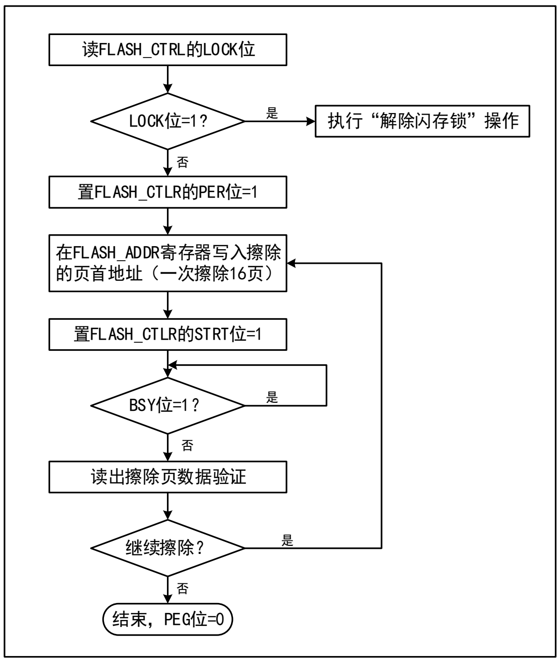


1) Check the LOCK bit in the FLASH_CTLR register. If it is 1, you need to perform the "Release Flash Memory Lock" operation.
2) Set the PG bit in the FLASH_CTLR register to ‘1’ to enable the standard programming mode.
3) Write the half word to be programmed to the designated flash memory address (even address).
4) When the BYS bit changes to '0' or the EOP bit in the FLASH_STATR register to be '1', it indicates the end of programming. Clear the EOP bit to 0.
5) Check the FLASH_STATR register to see if there is an error, or read the programming address data for verification.
6) To continue programming, you can repeat steps 3-5, end programming and clear the PG bit to 0.

32.5.4 Main memory standard erase

The flash memory can be erased in the standard pages (4K bytes) or in whole chips.

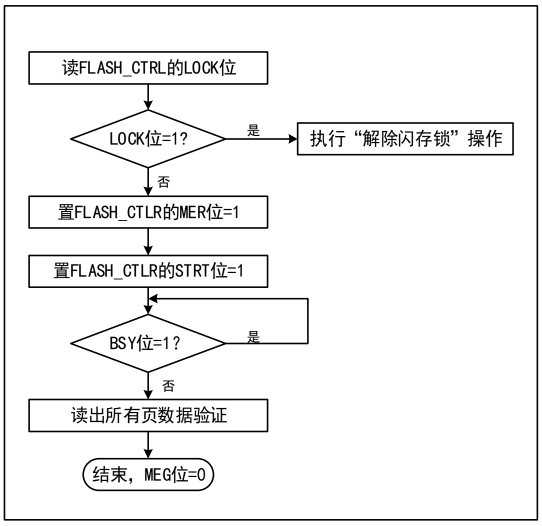
Figure 32-2 FLASH page erase



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 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 standard page continuously, you can repeat steps 3-5 to end erasing and clear the PEG bit to 0.

Note: After erasing is successful, word read - 0xe339e339, half word read - 0xe339, even address byte read - 0x39, odd address read 0xe3.

Figure 32-3 FLASH whole chip erase



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 MEG bit in the FLASH_CTLR register to ‘1’ to enable the whole chip erasure mode.
3) Set the STAT bit in the FLASH_CTLR register to ‘1’ to start an erasure action.
4) 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.
5) Read the data of the erasure page for verification.
6) Clear the MER bit to 0.

32.5.5 Fast programming mode unlock

The quick programming mode operation can be unlocked by writing the sequence to the FLASH_MODEKEYR register. After unlocking, the FLOCK bit of the FLASH_CTLR register will be cleared to 0, indicating that quick erasure and programming operations can be made. Set 1 by the software through the “FLOCK” bit of FLASH_CTLR register.

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 continuously made in sequence. Otherwise, it will be locked in case of wrong operation, and will be re-unlocked until the next system reset.

Note: For the quick programming operation, it needs to release the 2 layers of "LOCK" and "FLOCK".

32.5.6 Main memory fast programming

The fast programming (128 bytes) is made according to the page.

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) Check the FLOCK bit in the FLASH_CTLR register, if it is '1', you need to perform the "fast programming mode unlock" operation.
3) Check the BSY bit in the FLASH_STATR register to confirm that there are no other programming operations in progress.
4) Set the FTPG bit in the FLASH_CTLR register to '1' to enable fast page programming mode.
5) Use 32-bit mode to write data to the FLASH address, for example
   (uint32_t)0x80000000 = 0x12345678;
6) Wait for the WR_BSY of the FLASH_STATR register to be '0' and write the next data.
7) Repeat steps 5-6 a total of 64 times.
8) Set the PG_START bit of the FLASH_CTLR register to '1' to start fast page programming.
9) Wait for the BSY bit to become '0' or the EOP bit of the FLASH_STATR register to be '1' to indicate that a fast page programming is completed, and clear the EOP bit to 0.
10) Read the FLASH_STATR register to know if there is an error, or read the programming address data for verification.
11) To continue fast page programming, steps 5-10 can be repeated, and the FTPG bit will be cleared to 0 when the programming ends.

32.5.7 Main memory fast erasure

Quick erasure is also performed according to the pages (256 bytes).

1) Check the LOCK bit in the FLASH_CTLR register, if it is 1, you need to perform the "Release Flash Memory" operation.
2) Check the FLOCK bit in the FLASH_CTLR register, if it is 1, you need to perform the "fast programming mode unlock" operation.
3) Check the BSY bit in the FLASH_STATR register to confirm that there are no other programming operations in progress.
4) Set the FTER bit in the FLASH_CTLR register to '1' to enable the fast page erase (256 bytes) mode function.
5) Write the first address of the fast erase page to the FLASH_ADDR register.
6) Set the STAT bit in the FLASH_CTLR register to '1' to start a fast page erase (256 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 erasing, and clear the EOP bit to 0.
8) Read the FLASH_STATR register to know if there is an error, or read and erase the page address data verification.
9) To continue the fast page erasing, you can repeat steps 5-8, and clear the FTER bit to 0 when the erasing ends.

Note: After erasing is successful, word read - 0xe339e339, half word read - 0xe339, even address byte read - 0x39, odd address read 0xe3.

Fast Erase erases in blocks (32K bytes).

1) Check the LOCK bit in the FLASH_CTLR register, if it is 1, you need to perform the "unlock the flash memory" operation.
2) Check the FLOCK bit in the FLASH_CTLR register, if it is 1, you need to perform the "fast programming mode unlock" operation.
3) Check the BSY bit in the FLASH_STATR register to confirm that there are no other programming operations in progress.
4) Set the BER32 bit in the FLASH_CTLR register to '1' to enable the fast block erase (32K bytes) mode function.
5) Write the first address of the flash erase block to the FLASH_ADDR register.
6) Set the STAT bit in the FLASH_CTLR register to '1' to start a fast block erase (32K bytes) action.
7) Wait for the BYS bit to become '0' or the EOP bit in the FLASH_STATR register to be '1' to indicate the end of erasing, and clear the EOP bit to 0.
8) Query the FLASH_STATR register to see if there is an error, or read and erase the page address data verification.
9) To continue fast page erasing, you can repeat steps 5-8, and clear the BER32 bit to 0 after erasing.

Note: After erasing is successful, word read - 0xe339e339, half word read - 0xe339, even address byte read - 0x39, odd address read 0xe3.

Fast Erase erases in blocks (64K bytes).

1) Check the LOCK bit in the FLASH_CTLR register, if it is 1, you need to perform the "unlock the flash memory" operation.
2) Check the FLOCK bit in the FLASH_CTLR register, if it is 1, you need to perform the "fast programming mode unlock" operation.
3) Check the BSY bit in the FLASH_STATR register to confirm that there are no other programming operations in progress.
4) Set the BER64 bit in the FLASH_CTLR register to '1' to enable the fast block erase (64K bytes) mode function.
5) Write the first address of the flash erase block to the FLASH_ADDR register.
6) Set the STAT bit in the FLASH_CTLR register to '1' to start a fast block erase (64K bytes) action.
7) Wait for the BYS bit to become '0' or the EOP bit in the FLASH_STATR register to be '1' to indicate the end of erasing, and clear the EOP bit to 0.
8) Read the FLASH_STATR register to know if there is an error, or read and erase the page address data verification.
9) To continue the fast page erase, you can repeat steps 5-8, and clear the BER64 bit to 0 after the erase is completed.

Note: After erasing is successful, word read - 0xe339e339, half word read - 0xe339, even address byte read - 0x39, odd address read 0xe3

32.6 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-selected 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 32-3 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 32-4 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: Code read protection status, unreadable; pages 0 to 31 pages (4K) will be automatically write-protected and not controlled by WRPR0.	0x01
USER	[7:6]	RAM_CODE_MOD	00: CODE-192KB + RAM-128KB 01: CODE-224KB + RAM-96KB 10: CODE-256KB + RAM-64KB 11: CODE-288KB + RAM-32KB *Note: Applied for CH32V30x_D8C, CH32V30x_D8, CH32F20x_D8C and CH32F20x_D8.* 00: CODE-128KB + RAM-64KB 01: CODE-144KB + RAM-48KB 1x: CODE-160KB + RAM-32KB *Note: Applied for CH32V20x_D8W, CH32V20x_D8 and CH32F20x_D8W.*	xxb
	[5:3]	Reserved	Reserved	1
	2	STANDYRST	System reset control in Standby mode: 1: Disable; system is not reset when entering Standby mode; 0: Enable; system is reset when entering Standby mode.	1
	1	STOPRST	System reset control in Stop mode: 1: Disable; system will not be reset when entering Stop mode; 0: Enable; system will be reset when entering Stop mode.	1
	0	IWDGSW	Independent watchdog (IWDG) hardware enable:	1

			1: The IWDG function is enabled by software, and disabled by hardware; 0: The IWDG function is enabled by software (depends on the LSI clock).	
Data0–Data1			Saving the user's data 2 bytes.	FFFFh
WRPR0 - WRPR3			Write protection control. Each bit is used to control the write protection status of 1 sector (4Kbytes/sector) in main memory: 1: Disable write protection; 0: Enable write protection. 4 bytes are used to protect a total of 512K bytes of main memory. WRP0: Sectors 0-7 memory write protection control; WRP1: Sectors 8-15 memory write protection control; WRP2: Sectors 16-23 memory write protection control; WRP3: Bits 0-6 provide write protection for sectors 24-30; bit 7 provides write protection for sectors 31-127.	FFFFFFFFh

32.6.1 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.

32.6.2 User option bytes programming

It only supports the standard programming mode. The half word (2 bytes) is written at a time. In the actual process, when programming the user option bytes, FPEC only uses the low byte in the half-word, and automatically calculates the high byte (the high byte is the inverse code of the low byte), and then starts the programming operation. Ensure that byte in the user option bytes and its inverse code are always correct.

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) Check the BSY bit in the FLASH_STATR register to ensure that there is no other programming operation in progress.
3) Check the OBWRE bit in the FLASH_CTLR register. If it is 0, you need to perform the "User Option Bytes Unlock" operation.
4) Set the OBPG bit in the FLASH_CTLR register to ‘1’, then set the STAT bit in the FLASH_CTLR register to ‘1’, to enable the user option bytes programming.
5) Write the half word (2 bytes) to be programmed to the designated address.
6) When the BYS bit changes to '0' or the EOP bit in the FLASH_STATR register to be '1', it indicates the end

of programming. Clear the EOP bit to 0.
7) Read the programming address data for verification.
8) To continue programming, you can repeat steps 5-7, end programming and clear the OBPG bit to 0.

Note: When the "read protection" in the option bytes is modified and becomes "non-protected", the main memory area will be erased automatically once. If you modify the selections other than "read protection", the entire chip erasure operation will not occur.

32.6.3 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 1, you need to perform the "Release Flash Memory Lock" operation.
2) Check the BSY bit in the FLASH_STATR register to ensure that there is no programming operation in progress.
3) Check the OBWRE bit in the FLASH_CTLR register. If it is 0, you need to perform the "User Option Bytes Unlock" operation.
4) Set the OBER bit in the FLASH_CTLR register to ‘1’, then set the STAT bit in the FLASH_CTLR register to ‘1’, to enable the user option bytes erasure.
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 erasure address data for verification.
7) Clear the OBER bit.

Note: After erasing is successful, word read - 0xe339e339, half word read - 0xe339, byte read - 0x39.

32.6.4 Read protection release

The read protection of flash memory is determined by the user option bytes. Read the FLASH_OBR register. When the RDPRT bit is '1', it means that the current flash memory is in the read protection state, and the flash memory operation is subject to a series of safety protections in the read protection state. The process of releasing the read protection is as follows:

1) Erase the entire User Option Bytes area. At this time, the read protection field RDPR will become 0xFF, and the read protection will be still valid.
2) The user option bytes programming and writes the correct RDPR code 0xA5 to release the read protection of the flash memory. (This step will first cause the system to automatically perform a whole chip erasure operation on the flash memory).
3) Perform a power-on reset to reload the selection byte (including the new RDPR code), and the read protection is released at this time.

32.6.5 Write protection release

The write protection of flash memory is determined by the user option bytes. Read the FLASH_WPR register. Each bit represents 4K bytes of flash memory space. When the bit is ‘1’, it means the non-write-protected state, and ‘0’ means write-protected. The process of releasing write-protected is as follows:

1) Erase the whole user option bytes area.
2) Write the correct RDPR code 0xA5, and the read access is allowed;
3) Perform a system reset and reload the selection byte (including the new WRPR[3:0] byte) to release the write protection.

Chapter 33 Extended configuration

33.1 Extended configuration

The system provides an EXTEN extended configuration unit (EXTEN_CTR register). This unit uses AHB clock, and it is only reset when system reset occurs. It mainly includes the following extended control bit functions:

1) Adjust the internal voltage: LDOTRIM and ULLDOTRIM fields select default values, which can be modified when tuning performance and power consumption.
2) PLL clock selection: The HSIPRE field cooperates with the original clock configuration register to provide the option of dividing or not dividing the HSI clock as PLL input clock.
3) Lock-up monitor: When the LKUPEN field is enabled, the Lock-up monitor of the system will be enabled. Once a Lock-up condition occurs, the system will perform software reset, and the LKUPRESET field will be set to 1. It can be cleared by writing 1 after reading.
4) USBD internal resistor and transfer rate control: For USB full-speed device controller (USBD), the internal pull-up resistor (1.5KΩ) is enabled/disabled by configuring the USBDPU field. An external pull-up resistor is needed to be connected to USB pin (UD-pin when in low-speed mode, UD+ pin when in full-speed mode) when the internal pull-up resistor is disabled. The USBDLS field can be used to configure the current USB device speed mode.
5) ETH module 10M Ethernet and 1000M Ethernet RGMII interface enable control: 10M Ethernet can be enabled by configuring ETH_10M, and 1000M Ethernet RGMII interface can be enabled by configuring ETH_RGMII.
6) HSE oscillator control in low-power mode: It can be used to control whether HSE oscillates in low-power mode.

Note: For different types of MCUs, the extend register bits are defined differently, please refer to the description of EXTEN_CTR for details.

33.2 Register description

Table 33-1 EXTEN register

Name	Access address	Description	Reset value
R32_EXTEN_CTR	0x40023800	Configuration extend control register	0x00000A00

33.2.1 Configure Extended Control Register (EXTEN_CTR)

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			HSEKPLP	LDOTRIM[1:0]		ULLDOTRIM[1:0]		LKUPRST	LKUPEN	Reserved	HSIPRE	ETHRGMII	ETH10M	USBDPU	USBDLS

Bit	Name	Access	Description	Reset value
[31:13]	Reserved	RO	Reserved	0

[12]	HSEPLP	RW	HSE oscillator control in low-power mode: 1: HSE keeps oscillating in low-power mode. 0: HSE does not oscillate in low-power mode. *Note: Applies for CH32V20x_D8, CH32V20x_D8W and CH32F20x_D8W.*	0
[11:10]	LDOTRIM[1:0]	RW	Adjust digital core voltage value, LDO voltage value	10b
[9:8]	ULLDOTRIM[1:0]	RW	Adjust ULLDO voltage value in low-power mode	10b
7	LKUPRST	RW1	LOCKUP reset: 1: LOCKUP occurs and causes system reset. Write 1 to clear it. 0: Normal.	0
6	LKUPEN	RW	LOCKUP monitor function: 1: Enable. System reset occurs and set the LOCKUP_RESET bit when lock-up occurs. 0: Disable.	1
5	Reserved	RO	Reserved	0
4	HSIPRE	RW	HSI clock: (Only can be written when PLL is disabled.) 1: HSI clock selected as PLL input clock. 0: HSI clock divided by 2 selected as PLL input clock.	0
3	ETHRGMII	RW	1000M Ethernet RGMII interface enable and clock enable: 1: 1000M Ethernet RGMII interface enabled, and clock enabled. 0: 1000M Ethernet RGMII interface disabled, and clock disabled. *Note: Applies for CH32F20x_D8C and CH32V30x_D8C.*	0
2	ETH10M	RW	10M Ethernet enable and clock enable: 1: 10M Ethernet enabled, and clock enabled. 0: 10M Ethernet disabled, and clock disabled. *Note: Applies for CH32F20x_D8C, CH32V30x_D8C, CH32V20x_D8, CH32V20x_D8W, CH32F20x_D8W and CH32F20x_D8。*	0
1	USBDPU	RW	USBD internal pull-up resistor enable: 1: Enable (no external pull-up resistor is required). 0: Disable (an external pull-up resistor is required).	0
0	USBDLS	RW	USBD operating mode selection: 1: Low speed mode. 0: Full speed mode.	0

Chapter 34 Debug Support (DBG)

34.1 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
- Support for I2CSMBus timeout control
- Support for bxCAN communication

34.2 Register description

34.2.1 RISC-V 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								TIM10_STOP	TIM9_STOP	CAN2_STOP	CAN1_STOP	TIM8_STOP	TIM7_STOP	TIM6_STOP	TIM5_STOP

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIM4_STOP	TIM3_STOP	TIM2_STOP	TIM1_STOP	I2C2_SMBUS_TIMEOUT	I2C1_SMBUS_TIMEOUT	WWDG_STOP	IWDG_STOP	Reserved					STANDBY	STOP	SLEEP

Bit	Name	Access	Description	Reset value
[31:24]	Reserved	RW	Reserved	
23	TIM10_STOP	RW	Timer 10 debug stop bit. The counter stops when the core enters the debug state. 1: Timer 10's counter stops working. 0: Timer 10's counter is still working normally.	0
22	TIM9_STOP	RW	Timer 9 debug stop bit. The counter stops when the core enters the debug state. 1: Timer 9's counter stops working. 0: Timer 9's counter is still working normally.	0
21	CAN2_STOP	RW	CAN2 debug stop bit. CAN2 stops when the core enters the debug state. 1: CAN2's receive register does not continue to receive data. 0: CAN2 still operates normally.	0
20	CAN1_STOP	RW	CAN1 debug stop bit. CAN1 stops when the core enters the debug state. 1: CAN1's receive register does not continue to receive data. 0: CAN1 still operates normally.	0

19	TIM8_STOP	RW	Timer 8 debug stop bit. The counter stops when the core enters the debug state. 1: Timer 8's counter stops working. 0: Timer 8's counter is still working normally.	0
18	TIM7_STOP	RW	Timer 7 debug stop bit. The counter stops when the core enters the debug state. 1: Timer 7's counter stops working. 0: Timer 7's counter is still working normally.	0
17	TIM6_STOP	RW	Timer 6 debug stop bit. The counter stops when the core enters the debug state. 1: Timer 6's counter stops working. 0: Timer 6's counter is still working normally.	0
16	TIM5_STOP	RW	Timer 5 debug stop bit. The counter stops when the core enters the debug state. 1: Timer 5's counter stops working. 0: Timer 5's counter is still working normally.	0
15	TIM4_STOP	RW	Timer 4 debug stop bit. The counter stops when the core enters the debug state. 1: Timer 4's counter stops working. 0: Timer 4's counter is still working normally.	0
14	TIM3_STOP	RW	Timer 3 debug stop bit. The counter stops when the core enters the debug state. 1: Timer 3's counter stops working. 0: Timer 3's counter is still working normally.	0
13	TIM2_STOP	RW	Timer 2 debug stop bit. The counter stops when the core enters the debug state. 1: Timer 2's counter stops working. 0: Timer 2's counter is still working normally.	0
12	TIM1_STOP	RW	Timer 1 debug stop bit. The counter stops when the core enters the debug state. 1: Timer 1's counter stops working. 0: Timer 1's counter is still working normally.	0
11	I2C2_SMBUS _TIMEOUT	RW	SMBUS timeout mode debug stop bit. Stops SMBUS timeout mode when the core enters debug state. 1: freezes the SMBUS timeout control. 0: Same as normal mode operation.	0
10	I2C1_SMBUS _TIMEOUT	RW	SMBUS timeout mode debug stop bit. Stops SMBUS timeout mode when the core enters debug state. 1: freezes the SMBUS timeout control. 0: Same as normal mode operation.	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 is still working normally.	0
8	IWDG_STOP	RW	IWDG debug stop bit. The debug IWDG stops working when the core enters the debug state.	0

			1: IWDG counter stops working. 0: IWDG counter is still working normally.	
[7:3]	Reserved	RW	Reserved	0
2	STANDBY	RW	Debug the Standby mode bits. 1: (FCLK on, HCLK on) The digital circuitry section is not powered down, and the FCLK and HCLK clocks are clocked by the internal RL oscillator. Alternatively, the microcontroller exits STANDBY mode and reset by generating a system reset is the same. 0: (FCLK off, HCLK off) The entire digital circuitry section is powered down. From the software point of view, exiting STANDBY mode is the same as a reset (except that some status bits indicate that the microcontroller has just exited from STANDBY state).	0
1	STOP	RW	Debug stop mode bits. 1: (FCLK on, HCLK on) When in Stop mode, the FCLK and HCLK clocks are provided by the internal RC oscillator. When exiting Stop mode, software must reconfigure the clock system to start the PLL, crystal, etc. (same operation as when configuring this bit to 0). 0: (FCLK off, HCLK off) When in STOP mode, the clock controller disables all clocks (including HCLK and FCLK). When exiting from STOP mode, the clock configuration is the same as after reset (the microcontroller is clocked by an 8 MHz internal RC oscillator (HIS)). Therefore, the software must reconfigure the clock control system to start the PLL, crystal, etc.	0
0	SLEEP	RW	Debug sleep mode bits. 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, the software does not need to reconfigure the clock system when exiting from Sleep mode.	0

34.2.2 ARM Debug MCU Configuration Register (DBGMCU_CR)

Offset address: 0xE0042004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								TIM10_STOP	TIM9_STOP	CAN2_STOP	TIM8_STOP	TIM7_STOP	TIM6_STOP	TIM5_STOP	I2C2_SMBUS_T

															IME OUT
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2C1_SMBUS_TIMEOUT	CAN1_STOP	TIM4_STOP	TIM3_STOP	TIM2_STOP	TIM1_STOP	WWDG_STOP	IWDG_STOP	TRACE_MODE		TRACE_IOEN	Reserved		STANDBY	STOP	SLEEP

Bit	Name	Access	Description	Reset value
[31:24]	Reserved	RW	Reserved	
23	TIM10_STOP	RW	Timer 10 debug stop bit. The counter stops when the core enters the debug state. 1: Timer 10's counter stops working. 0: Timer 10's counter is still working normally.	0
22	TIM9_STOP	RW	Timer 9 debug stop bit. The counter stops when the core enters the debug state. 1: Timer 9's counter stops working. 0: Timer 9's counter is still working normally.	0
21	CAN2_STOP	RW	CAN2 debug stop bit. CAN2 stops when the core enters the debug state. 1: CAN2's receive register does not continue to receive data. 0: CAN2 still operates normally.	0
20	TIM8_STOP	RW	Timer 8 debug stop bit. The counter stops when the core enters the debug state. 1: Timer 8's counter stops working. 0: Timer 8's counter is still working normally.	0
19	TIM7_STOP	RW	Timer 7 debug stop bit. The counter stops when the core enters the debug state. 1: Timer 7's counter stops working. 0: Timer 7's counter is still working normally.	0
18	TIM6_STOP	RW	Timer 6 debug stop bit. The counter stops when the core enters the debug state. 1: Timer 6's counter stops working. 0: Timer 6's counter is still working normally.	0
17	TIM5_STOP	RW	Timer 5 debug stop bit. The counter stops when the core enters the debug state. 1: Timer 5's counter stops working. 0: Timer 5's counter is still working normally.	0
16	I2C2_SMBUS_TIMEOUT	RW	SMBUS timeout mode debug stop bit. Stops SMBUS timeout mode when the core enters debug state. 1: Freezes the SMBUS timeout control. 0: Same as normal mode operation.	0
15	I2C1_SMBUS_TIMEOUT	RW	SMBUS timeout mode debug stop bit. Stops SMBUS timeout mode when the core enters debug state. 1: freezes the SMBUS timeout control.	0

			0: Same as normal mode operation.	
14	CAN1_STOP	RW	CAN1 debug stop bit. CAN1 stops when the core enters the debug state. 1: CAN1's receive register does not continue to receive data. 0: CAN1 still operates normally.	0
13	TIM4_STOP	RW	Timer 4 debug stop bit. The counter stops when the core enters the debug state. 1: Timer 4's counter stops working. 0: Timer 4's counter is still working normally.	0
12	TIM3_STOP	RW	Timer 3 debug stop bit. The counter stops when the core enters the debug state. 1: Timer 3's counter stops working. 0: Timer 3's counter is still working normally.	0
11	TIM2_STOP	RW	Timer 2 debug stop bit. The counter stops when the core enters the debug state. 1: Timer 2's counter stops working. 0: Timer 2's counter is still working normally.	0
10	TIM1_STOP	RW	Timer 1 debug stop bit. The counter stops when the core enters the debug state. 1: Timer 1's counter stops working. 0: Timer 1's counter is still working normally.	0
9	WWDG_STOP	RW	Window watchdog debug stop bit. The debug window watchdog stops working when the core enters the debug state. 1: The window watchdog counter stops working. 0: The window watchdog counter is still working normally.	0
8	IWDG_STOP	RW	Independent watchdog debug stop bit. The watchdog stops when the core enters the debug state. 1: The watchdog counter stops working. 0: The watchdog counter is still working normally.	0
[7:6]	TRACE_MODE	RW	Trace pin assignment control bit. This bit is used in conjunction with TRACE_IOEN. When TRACE_IOEN = 0: xx: No trace pins are assigned (default state). When TRACE_IOEN = 1: 00: asynchronous mode is used for the trace pins. 01: The trace pin uses synchronous mode and has a data length of 1. 10: The trace pin uses synchronous mode and has a data length of 2. 11: the trace pin is in synchronous mode and the data length is 4.	xx
5	TRACE_IOEN	RW	Trace pin assignment enable bit. This bit is used in conjunction with TRACE_MODE. 0: No trace pin is assigned (default state). 1: Assign the trace pin.	0
[4:3]	Reserved	RW	Reserved	0
2	STANDBY	RW	Debug the standby mode bits.	0

			1: (FCLK on, HCLK on) The digital circuitry section is not powered down, and the FCLK and HCLK clocks are clocked by the internal RL oscillator. Alternatively, the microcontroller exits STANDBY mode and reset by generating a system reset is the same. 0: (FCLK off, HCLK off) The entire digital circuitry section is powered down. From the software point of view, exiting STANDBY mode is the same as a reset (except that some status bits indicate that the microcontroller has just exited from STANDBY state).	
1	STOP	RW	Debug stop mode bits. 1: (FCLK on, HCLK on) When in stop mode, the FCLK and HCLK clocks are provided by the internal RC oscillator. When exiting stop mode, software must reconfigure the clock system to start the PLL, crystal, etc. (same operation as when configuring this bit to 0). 0: (FCLK off, HCLK off) When in STOP mode, the clock controller disables all clocks (including HCLK and FCLK). When exiting from STOP mode, the clock configuration is the same as after reset (the microcontroller is clocked by an 8 MHz internal RC oscillator (HIS)). Therefore, the software must reconfigure the clock control system to start the PLL, crystal, etc.	0
0	SLEEP	RW	Debug sleep mode bits. 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, the software does not need to reconfigure the clock system when exiting from sleep mode.	0

Note: Applicable to CH32F2x series. When the system enters debug mode, the chip has a certain peripheral, the debug module has the function to configure that peripheral, and the debug MCU configuration register has the configuration bits corresponding to that peripheral.