Overview

The CH32F103x is general microcontroller designed 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 CH32V103R8T6/CH32V103C8T6/CH32V103C8U6/CH32V103C6T6 in the CH32V103x series are general microcontrollers designed based on the 32-bit RISC-V instruction set (IMAC) and RISC-V3A processor, mounted with abundant peripheral interfaces and functional modules. Its internal organizational structure meets low-cost and low-power embedded application scenarios.

This datasheet provides detailed application information of CH32F103x series and CH32V103x series products for user application development. It is applicable to products with different memory capacities, functional resources, and packages in the series. If there are differences, special instructions will be given in the corresponding function chapters.

CH32F103x compare with CH32V103x

Function / Differences	Description		
	CH32F103x	Difference	CH32V103x
Core (command)	Cortex-M3 (ARM)	Different commands and frameworks	RISC-V3A (RV32IMAC)
Interrupt controller	NVIC	Different actual application methods	PFIC
Bit segment mapping	Support	-	Not support
TKEY	TKEY_F	Different application methods	TKEY_V
USBHD	5 configurable USB device endpoints	● Different number of endpoints ● Different endpoint register addresses ● Different receive/transmit sizes of USB host endpoint ● Different pins of physical USB port	16 configurable USB device endpoints
CAN/DAC/USBD	Support	-	Not support
DEBUG	SWD	Different protocols	RVSWD
Others	Consistent		

Abbreviated description of bit attribute in the register

Bit Attribute	Description
RF	Read-only, the value read out is fixed.
RO	Read-only, it can be changed by hardware.
RZ	Read-only, the bit is automatically cleared after read operation
WO	Write only (unreadable, uncertain read value)

WA	Write only, it can be written in the safe mode.
WZ	Write only, automatically cleared after write operation
RW	Readable, writable.
RWA	Readable, it can be written in the safe mode.
RW1	Readable, valid when writing 1; invalid when writing 0.
RW0	Readable; valid when writing 0; invalid when writing 1.
RW1T	Readable; invalid when writing 0, overturn when writing 1.

Chapter 1 Memory and Bus Framework

1.1 Bus Framework

The CH32F103 is a microcontroller designed based on the Cortex-M3 core. The core, arbitration unit, DMA module, and SRAM memory, etc. in the framework interact through multiple sets of buses. The system architecture is as shown in Figure 1-1.

The CH32V103 is a general microcontroller designed based on the RISC-V3A core. The core, arbitration unit, DMA module and SRAM memory, etc. of the architecture interact through multiple sets of buses. The 2-level assembly line processing is adopted for the core, and is equipped with static branch prediction and command prefetch mechanisms to achieve the best performance ratio of the system with low power, low cost, and high-speed operation. The system architecture is as shown in Figure 1-2.

Figure 1-1 CH32F103 System architecture

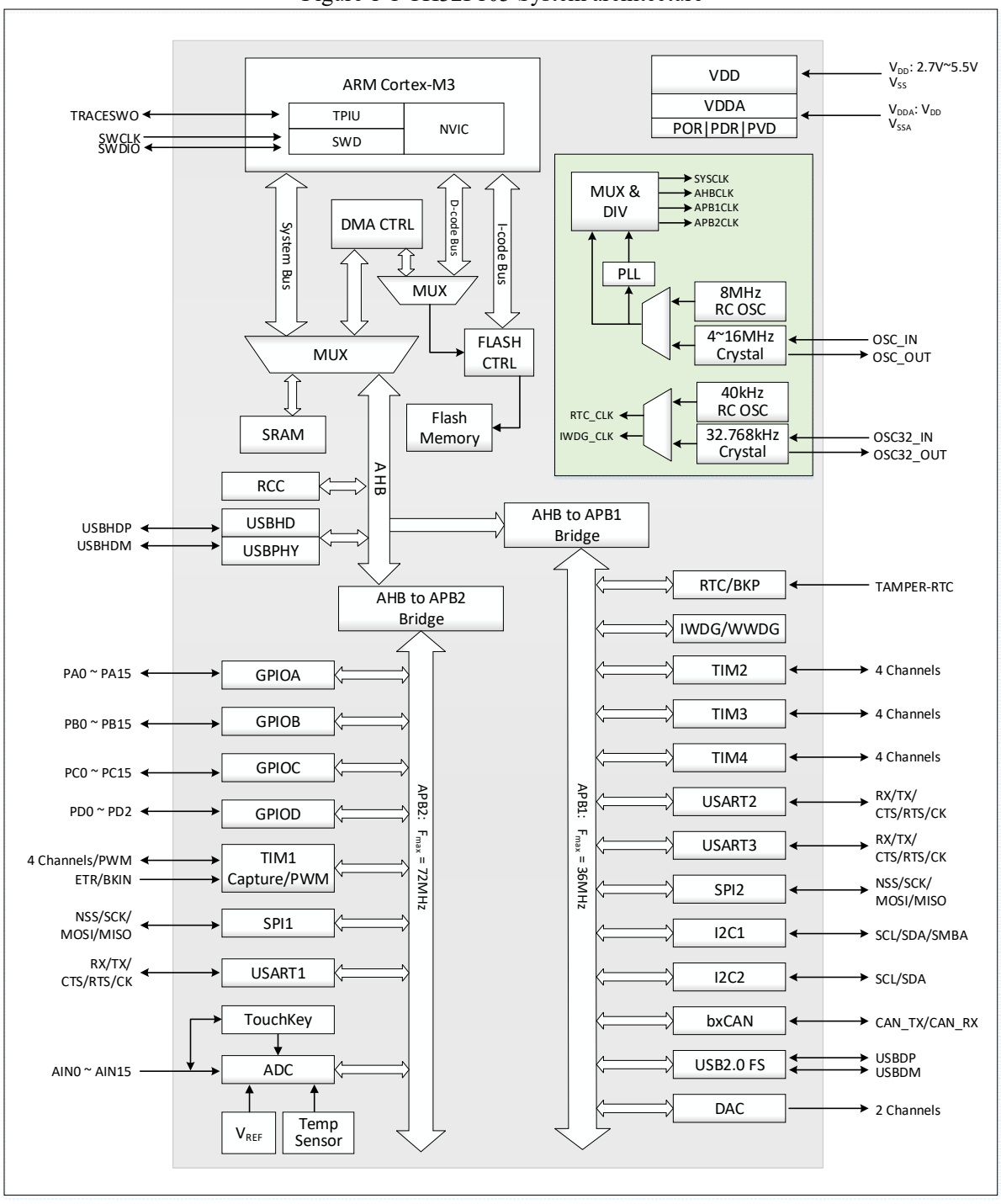
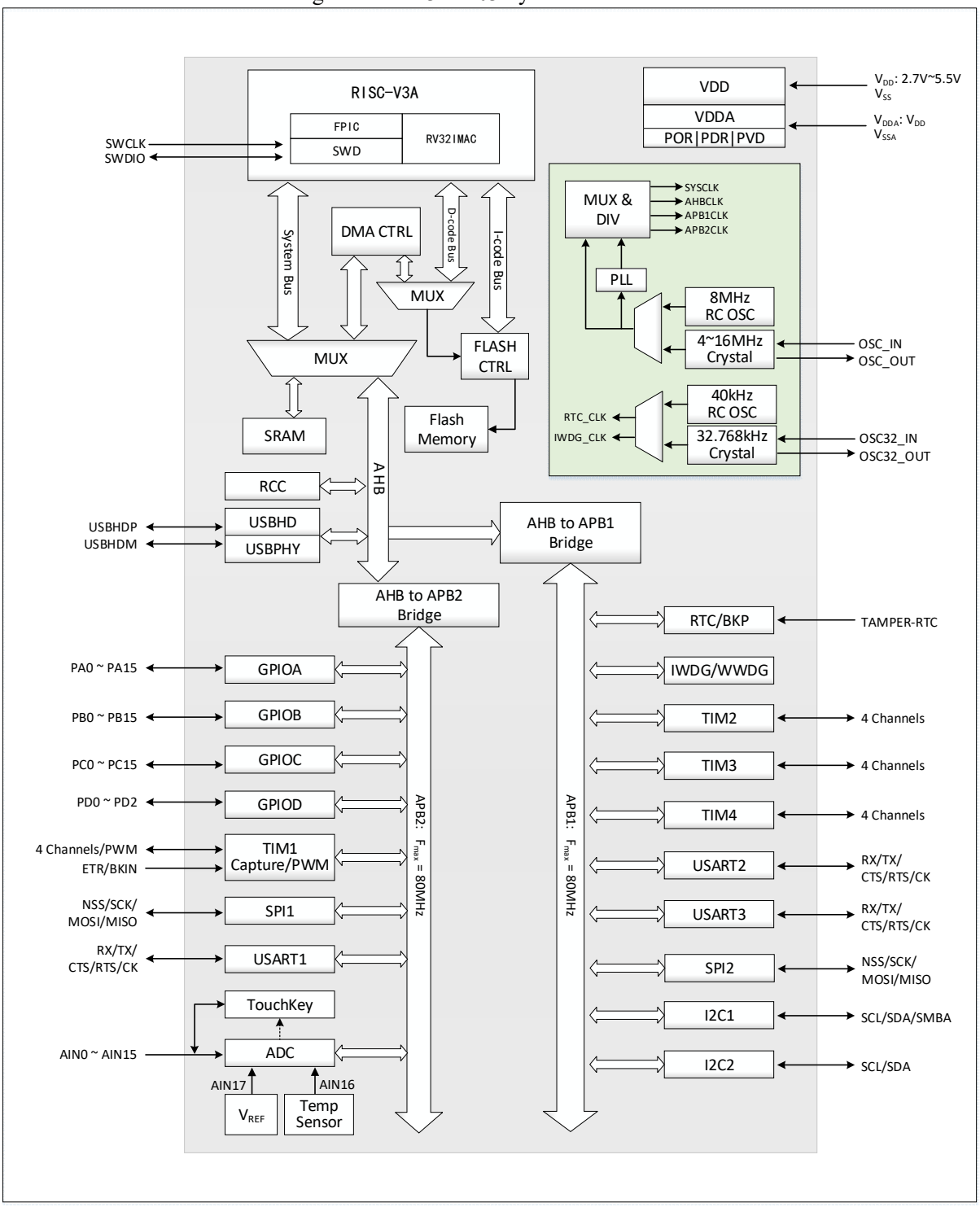


Figure 1-2 CH32V103 System architecture



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 between 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 two APB buses. Different peripherals are connected to different APB buses, and different bus clocks can be configured according to actual needs to optimize performance.

1.2 Memory Mapping

Both CH32F103 and CH32V103 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.

Figure 1-3 CH32F103 Storage Mapping

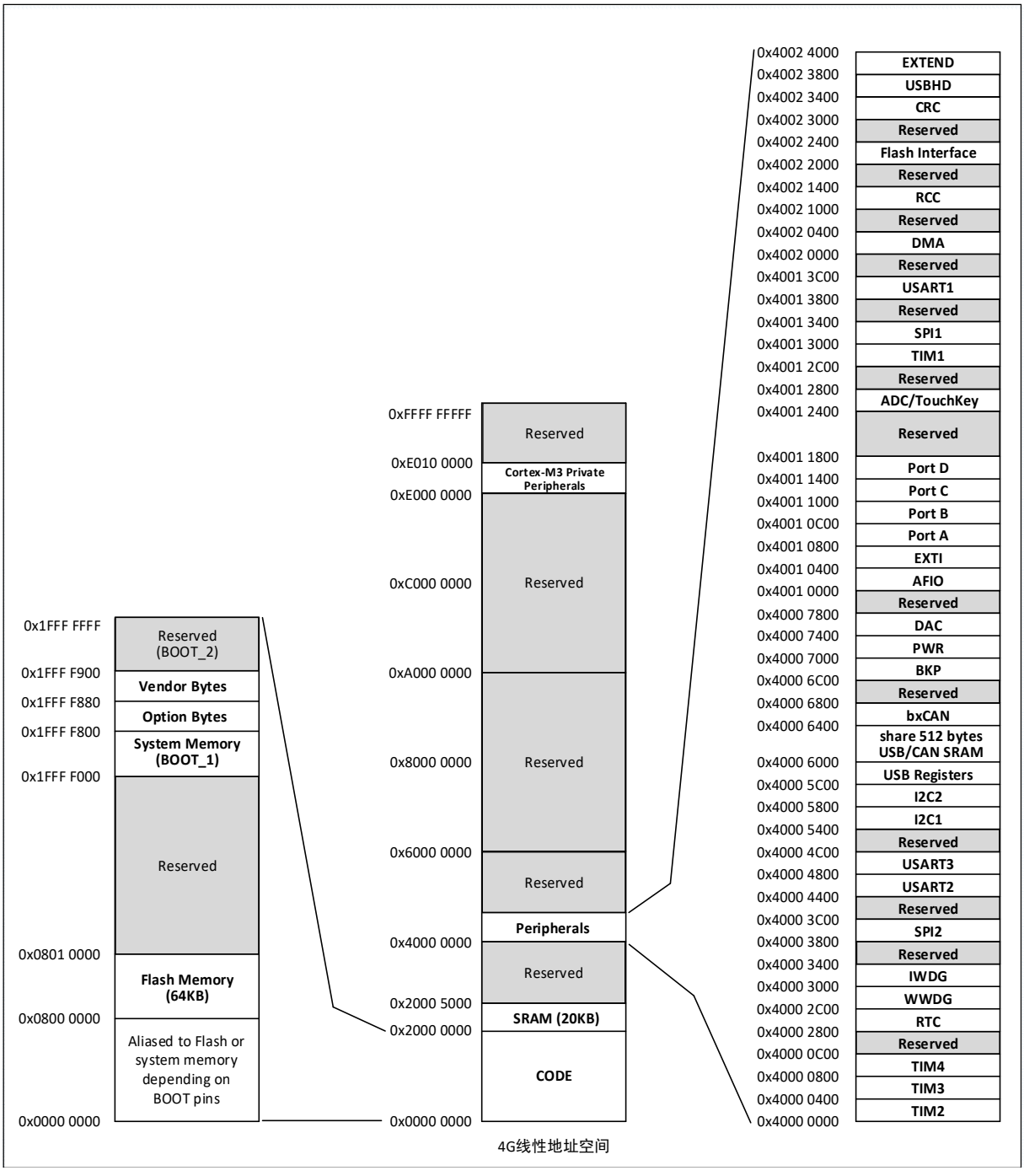
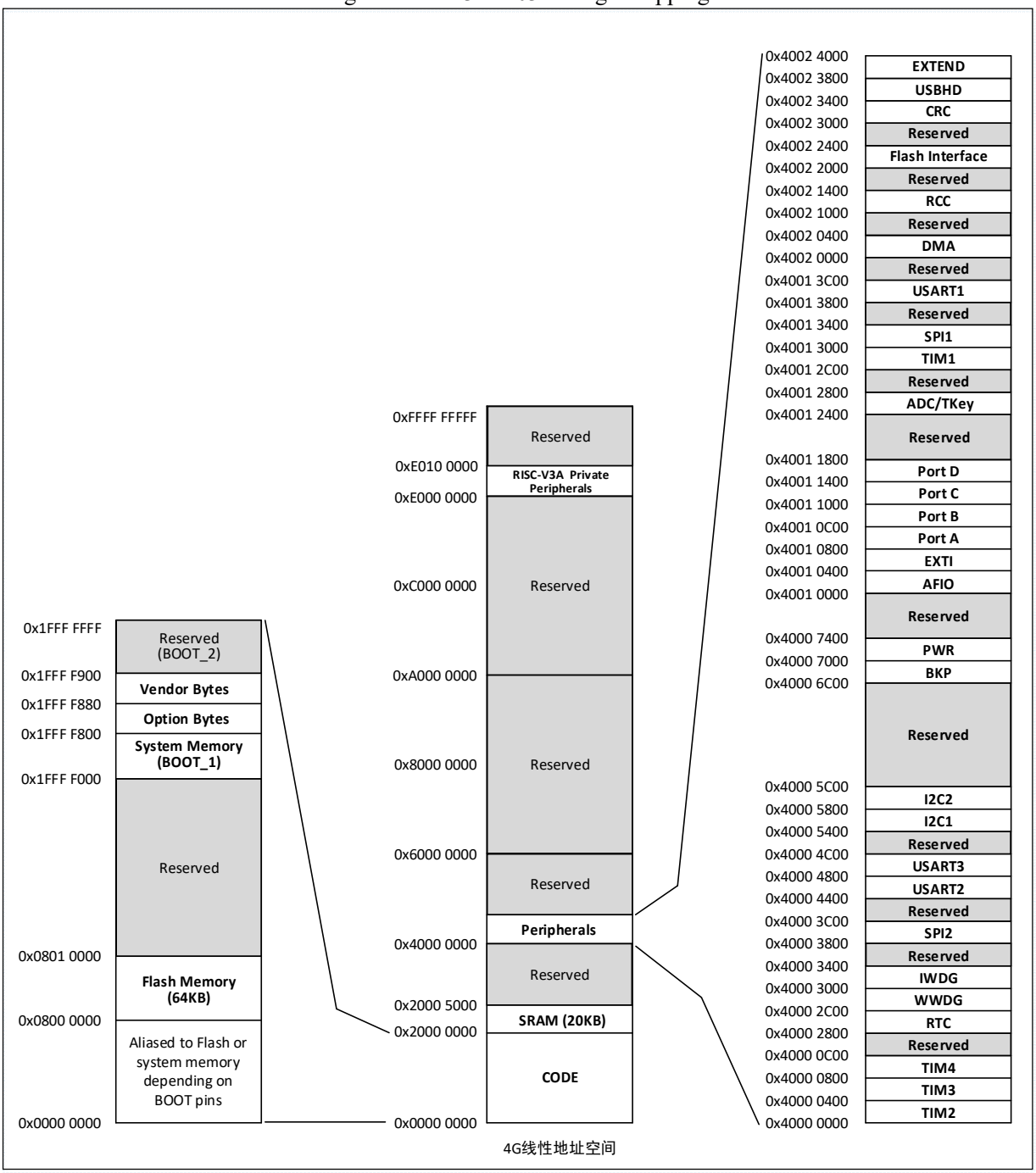


Figure 1-4 CH32V103 Storage Mapping



1.2.1 Bit Segment Access

Bit operation means independently reading and writing a bit operation. The CH32F103 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

$$Mapaddr = BEaddr + 0x2000000 + (Ofaddr \times 32) + (BitN \times 4)$$

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 know 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: CH32V103 does not support bit segment mapping access mode.

1.2.2 Memory Distribution

Built-in maximum 20K bytes of SRAM, initial address 0x20000000, supporting byte, half word (2 bytes), full word (4 bytes) access.

Built-in 64K bytes of program flash storage area (CodeFlash), for storing user applications.

Built-in 3.75K bytes of system memory (bootloader), for storing the system boot program (manufacturer's solidified bootloading program).

The built-in 128-byte space is 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.

1.3 Startup Configuration

The system can select three different startup modes through the BOOT0 and BOOT1 pins.

Table 1-1 Startup Mode

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

The user selects the startup mode after reset by setting the status value of BOOT pin. 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 original 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 being started up from the internal SRAM, it can be only accessed from 0x20000000 address area. When CH32F103 series products are 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 CH32V103, such action is not required.

Chapter 2 Power Control (PWR)

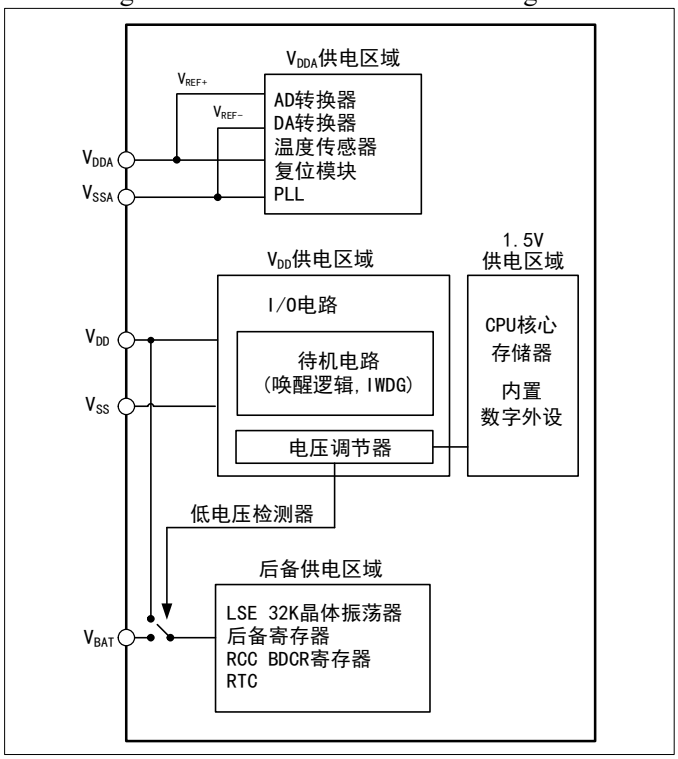
The module description in this chapter applies to the full range of CH32F103 and CH32V103 microcontrollers.

2.1 Overivew

The system operating voltage V_{DD} ranges from 2.7 to 5.5V, 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 such as battery can provide power to the real-time clock (RTC) and backup registers through the V_{BAT} pin. If the backup power 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. V_{REF+} and V_{REF-} are used as reference points for some analog circuits and 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} terminals.

Figure 2-1 Power Structure Block Diagram



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, PC13-15 cannot be used as GPIO, and only the following functions can be used:

- 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 to be powered by V_{DD}, and PC13～15 can be used as GPIO functions.

When PC13～15 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.

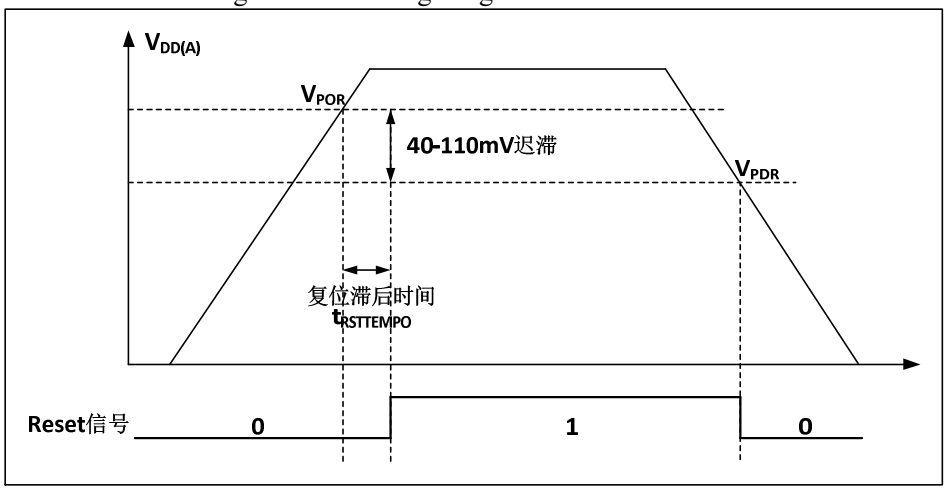
Note: During the restoration of the main power V_{DD}, the internal V_{BAT} power is still connected to the external backup power through the corresponding V_{BAT} pin. If VDD is less than the reset delay time $t_{RSTTEMPO}$, it will be stabilized and be higher than the value of V_{BAT} by more than 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 such as the battery will be injected through the V_{BAT} pin. If the backup power cannot withstand such instantaneously injected current, it is recommended to add a forward conducting low-dropout diode between the backup power and V_{BAT} pin.

2.2 Power Management

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 the chip supply voltage V_{DD} and V_{DDA} are lower than the corresponding threshold voltage, the system will be reset by the relevant circuits, without an external reset circuit. Please refer to the corresponding data sheet for the parameters of the power-on threshold voltage V_{POR} and the power-down threshold voltage V_{PDR}.

Figure 2-2 Working Diagram of POR and PDR



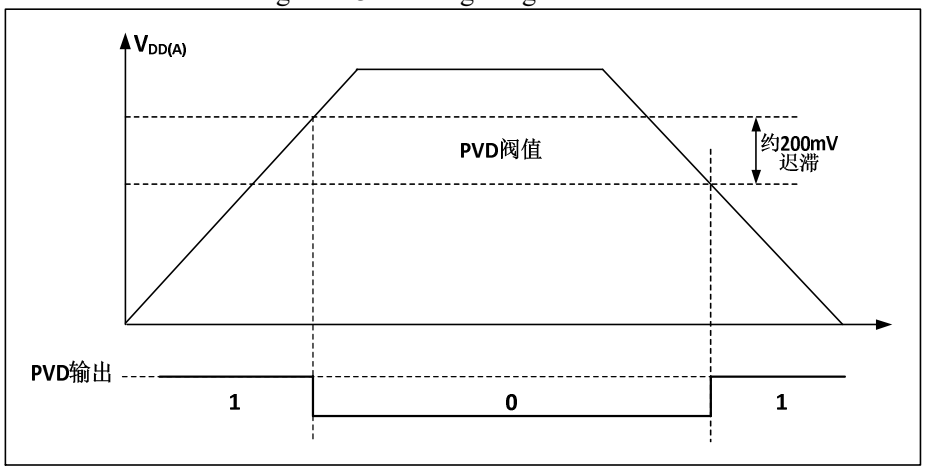
2.2.2 Programmable Voltage Detector

The programmable voltage detector PVD is mainly used to monitor the changes of the main power of the system and compare the power with the threshold voltage set by the PLS[2:0] of 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.

The specific configuration is as follows:

1） Set the PLS[2:0] field of the PWR_CTLR register and select the voltage threshold to be monitored.
2） Selectable interrupt processing. The PVD function is internally connected to the line16 of the EXTI module to trigger the setting of rising/falling edges. Turn on this interrupt (with EXTI configured). When VDD drops to be less than the PVD threshold or rises above the PVD threshold, a PVD interrupt will be generated.
3） Set the PVDE bit of the PWR_CTLR register to enable the PVD function.
4） Read the PVD0 bit of the PWR_CSR status register to obtain the relationship between the main power of the current system and the threshold set by PLS[2:0], and perform the corresponding soft processing.

Figure 2-3 Working Diagram of PVD



2.3 Low Power Mode

After the system is reset, the microcontroller will be at a normal working status (running mode). At this time, the system power consumption can be saved by reducing the main frequency of the system or switching off the peripheral clock or reducing the working peripheral clock. If the system does not need to work, you can set the system to enter low-power mode, and let the system jump out of this status through specific event.

The microcontroller currently provides three low-power-consumption modes, which are divided into the following 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	Access	Wake-up source	Effect on clock	Voltage regulator
Sleep	WFI	Arbitrary interrupt wakes up	Core clock off, no effect on other clocks	Normal
	WFE	Wake-up event wakes up		
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	Switch off the HSE, HIS, PLL and peripheral clock	Normal: 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 external interrupt/event can also wake up the system, but the system will not be reset after wake-up.*	Switch off the HSE, HIS, PLL and peripheral clock	Normal: LPDS=0 or Low power: LPDS=1

Note: The SLEEPDEEP bit belongs to the core private peripheral control bit. For CH32F103 products, refer to the Cortex-M3 core manual, and for CH32V103 products, refer to the PFIC_SCTLR register.

2.3.1 Low Power Configuration

- WFI and WFE Modes

WFI: The microcontroller is woken up by an interrupt source that has an interrupt controller response. After the system is woken up, the interrupt service function will be executed firstly (except for the microcontroller

reset).

WFE: When a wake-up event triggers the microcontroller, it will exit the low-power mode. Wake-up events include:

1) Configure an external or internal EXTI line as time 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 and start up the peripheral interrupt enable, but do not start up the interrupt enable in the interrupt controller, and the interrupt pending bit needs to be cleared after the system is wakened up.

- SLEEPONEXIT

Enable: After the WFI or WFE command is executed, the microcontroller will ensure entering the low-power mode after all pending interrupt services exit.

Disable: After the WFI or WFE command is executed, the microcontroller will immediately enter the low-power mode.

- SEVONPEND

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

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

2.3.2 Sleep Mode

In this mode, all IO pins keep their status in the running mode, and all peripheral clocks are normal, so turn off useless peripheral clocks as far as possible to reduce power consumption before entering sleep mode. The time required by the wake-up in such mode is the shortest.

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

Exit: Any interrupt or wake-up event.

2.3.3 Stop mode

The stop mode is a clock control mechanism based on the deep sleep mode (SLEEPDEEP) of the core and combined with peripherals, and allows the voltage regulator to operate in a lower-power-consumption status. In this mode, the high-frequency clock (HSE/HSI/PLL) domain is closed, SRAM and register contents are retained, and the IO pin status is maintained. The system can continue to run after being woken up from this mode, and HSI is called as the default system clock.

If the flash memory is being programmed, the system will not enter the stop mode until the memory access is completed; if the APB is being accessed, the system will not enter the stop mode until the APB access is 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 is optional. Execute WFI or WFE, and SEVONPEND and SLEEPONEXIT are optional.

Exit: Any external interrupt/event (set in the external interrupt register), rising edge of WKUP pin.

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 external interrupt/event (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.

Note: In debug mode, if the microprocessor enters the stop or standby mode, the debugging connection will be lost.

2.3.5 RTC Automatic Wake-up

RTC can realize automatic wake-up without external interrupt. By programming the time base, you can periodically wake up from the stop or standby mode.

A precise external low-frequency 32.768 kHz crystal oscillator LSE can be selected as the RTC clock source, or an internal LSI oscillator 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 this function, the external interrupt line 17 needs to be configured as a rising edge interrupt, and the RTC can be set to generate an alarm event. To wake up from the standby mode, you only need to set the RTC to generate an alarm event.

2.4 Register Description

Table 2-2 List of PWR Related Registers

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

Power control register (PWR_CTLR)

Offset address: 0x00

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

Bit	Name	Access	Description	Reset value
[31:9]	Reserved	RO	Reserved	0
8	DBP	RW	Write enable of backup area. When the RTC clock is the 128 frequency divisions of the external clock, this bit must be set to 1. 1: Allowed to write RTC and backup registers; 0: Disabled to write RTC and backup registers.	0
[7:5]	PLS[2:0]	RW	PVD voltage detect threshold setting 000: 2.65V at the rising edge/2.5V at the falling edge; 001: 2.87V at the rising edge/2.7V at the falling edge; 010: 3.07V at the rising edge/2.89V at the falling edge; 011: 3.27V at the rising edge/3.08V at the falling edge; 100: 3.46V at the rising edge/3.27V at the falling edge; 101: 3.76V at the rising edge/3.55V at the falling edge; 110: 4.07V at the rising edge/3.84V at the falling edge; 111: 4.43V at the rising edge/4.13V at the falling edge;	0
4	PVDE	RW	Supply voltage monitoring function enable flag bit 1: Enable supply voltage function; 0: Disable supply voltage function.	0
3	CSBF	RW1	Clear the standby status flag bit, and the value read out is always 0. 1: Set 1 to clear the SBF standby status flag bit; 0: Cleared to 0 and invalid.	0
2	CWUF	RW1	Clear the wake-up status flag bit, and the value read out is always 0. 1: After it is set to 1, the WUF flag bit will be cleared in 2 system clock cycles; 0: Cleared to 0 and invalid.	0
1	PDDS	RW	Standby/stop mode selection bit when power-down and in deep sleep. 1: Enter standby mode; 0: Enter stop mode.	0
0	LPDS	RW	Voltage regulator working mode selection bit in stop mode. 1: The voltage regulator works in the low-power mode; 0: The voltage regulator works in the normal mode.	0

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

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
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							EWUP						PVD0	SBF	WUF

Bit	Name	Access	Description	Reset value
[31:9]	Reserved	RO	Reserved.	0
8	EWUP	RW	WKUP pin enable bit 1: WKUP is forcibly configured as input pull-down status, used to wake up the MCU from the standby status; 0: WKUP pin can be used for general purpose IO, without standby wake-up function.	0
[7:3]	Reserved	RO	Reserved.	0
2	PVD0	RO	PVD output status flag bit 1: V_{DD} and V_{DDA} are lower than the PVD threshold set by PLS[2:0]; 0: V_{DD} and V_{DDA} are higher than the PVD threshold set by PLS[2:0].	0
1	SBF	RO	Standby state flag bit, cleared by setting CSBF bit to 1. 1: MCU enters standby mode; 0: MCU is not in standby mode.	0
0	WUF	RO	Wake-up event status flag bit, cleared by setting CWUF bit to 1. 1: A wake-up event or RTC alarm event is detected on the WKUP pin; 0: No wake-up event occurs.	0

Note: This register remains unchanged after being woken up from the standby mode.

Chapter 3 Reset, Expansion and Clock Control

The module description in this chapter applies to the full range of CH32F103 and CH32V103 microcontrollers.

The controller provides different reset forms 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

- Multiple reset forms
- Multiple clock sources, bus clock management
- Built-in external crystal oscillation monitoring and clock security system
- Independent management of each peripheral clock: reset, switching on, switching off
- Supporting internal clock output

3.2 Reset

The controller provides three types of reset forms: Power reset, system reset and backup area reset.

3.2.1 Power Reset

When power reset occurs, all registers except for backup area will be reset (the backup area is powered by V_{BAT}), and PC pointer in the application program is fixed at address 0x00000004 (Reset vector table).

The generation conditions include:

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

3.2.2 System Reset

When a system reset occurs, all registers except for the reset flag and backup area in the control/status register RCC_RSTSCKR will be reset. Identify the source of the reset event by viewing the reset status flag bit in the RCC_RSTSCKR register.

The generation conditions include:

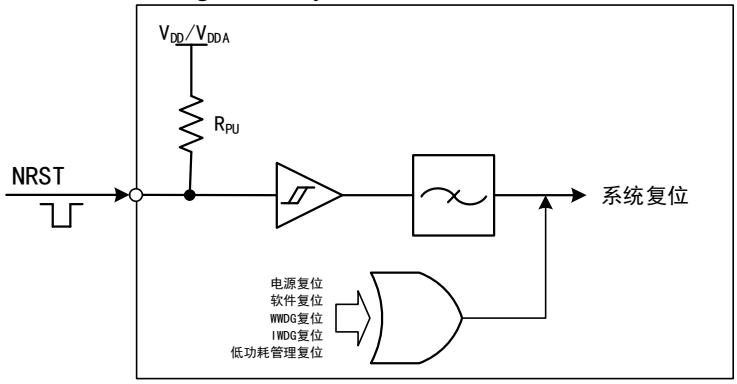
- Low level signal on NRST pin (external reset)
- Window watchdog counting termination (WWDG reset)
- Independent watchdog counting termination (IWDG reset)
- Software reset (SW reset)
- Low power reset

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 chapter.

Software reset: CH32F103 product resets the system by setting bit2 in the core register AIRCR. For specific operations, please refer to the Cortex-M3 Core Manual for more detailed information. The CH32V103 product resets the system by setting the SYSRESET bit of the interrupt configuration register PFIC_CFGR in the programmable interrupt controller PFIC to 1. Refer to the corresponding chapter for details.

Low power management reset: By setting the STANDY_RST bit in the user selection byte to 1, the standby mode reset will be enabled. After the process of entering the standby mode is executed at this time, a system reset will be performed instead of entering the standby mode. By setting the STOP_RST bit in the user selection byte to 1, 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 the stop mode.

Figure 3-1 System Reset Structure



3.2.3 Backup Area Reset

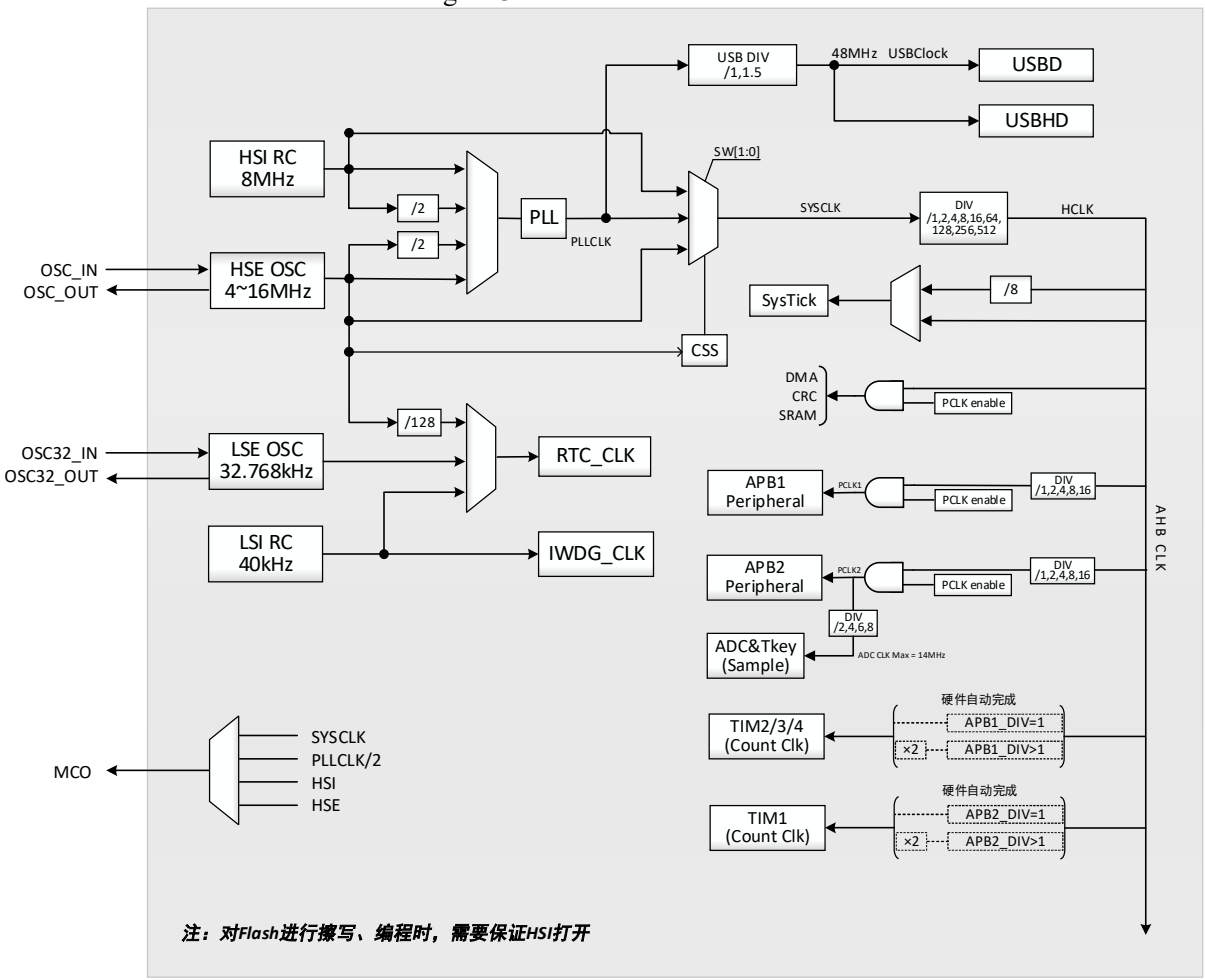
When the backup area reset occurs, only the backup area register will be reset, including the backup register, RCC_BDCTLR register (RTC enable and LSE oscillator). The generation conditions include:

- 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 BDRST bit of the RCC_BDCTLR register to 1
- Set BKPRST bit of the RCC_APB1PRSTR register to 1

3.3 Clock

3.3.1 System Clock Structure

Figure 3-2 Clock Tree Structure



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 any external device. Its startup time is very short but the clock frequency accuracy is poor. HSI is enabled and 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 (it is recommended not to disable it). If the HSIRDYIE bit of the RCC_INTR register is set, a corresponding interrupt will be generated.

- Factory calibration: The difference in manufacturing process will cause the RC oscillation frequency of each chip to be different, so HSI calibration will be performed for each chip before the chip is delivered out of the factory. After the system is reset, the factory calibration value will be loaded into HSICAL[7:0] of 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.

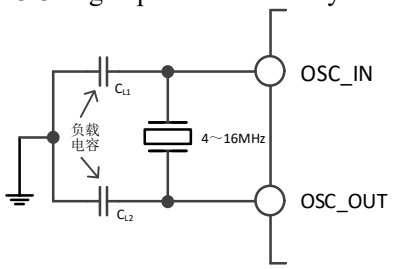
Note: If the HSE crystal oscillator fails, the HSI clock will be used as a backup clock source (clock security system).

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

- External crystal/ceramic resonator (HSE crystal): A 4-16MHz external oscillator provides a more accurate clock source for the system. For further information, please refer to the electrical

characteristics of the datasheet. The HSE crystal can be enabled and disabled by setting the HSEON bit in the RCC_CTLR register. The HSERDY bit indicates whether the HSE crystal oscillation is stable. The hardware will not send the clock to the system until the HSERDY bit is set to 1. If the HSERDYIE bit of the RCC_INTR register is set, a corresponding interrupt will be generated.

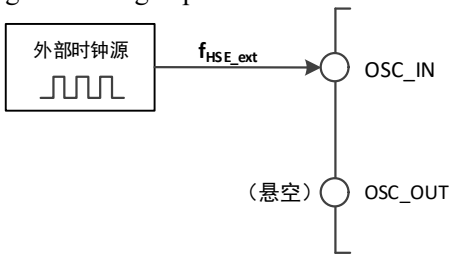
Figure 3-3 High-speed External Crystal Circuit



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

- External high-speed clock source (HSE bypass): In this mode, the clock source is directly sent to the OSC_IN pin from the outside, and the OSC_OUT pin is suspended. It supports 25MHz frequency at most. The application program needs to set the HSEBYP bit when the HSEON bit is 0. Switch on the HSE bypass function, and then set the HSEON bit.

Figure 3-4 High-speed Clock Source Circuit



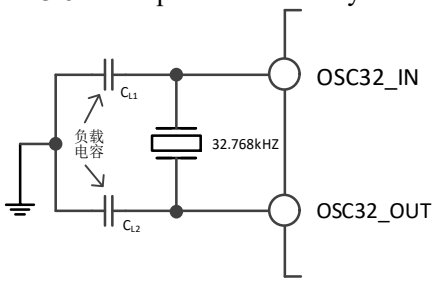
3.3.3 Low-Speed Clock (LSI/LSE)

LSI is a low-speed clock signal generated by an RC oscillator of approximately 40 kHz in the system. It can keep running in the stop and standby modes, providing a clock reference for the RTC clock, independent watchdog, and wake-up unit. For further information, please refer to the electrical characteristics of the datasheet. LSI can be enabled and 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, and the hardware will input the clock after the LSIRDY bit is set to 1. If the LSIRDYIE bit of the RCC_INTR register is set, a corresponding interrupt will 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 and accurate clock source for RTC clock or other timing functions.

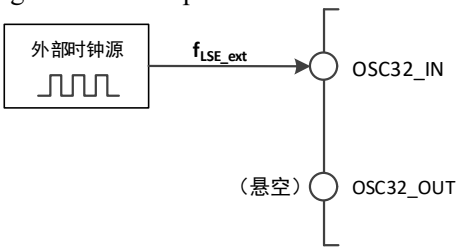
- External crystal/ceramic resonator (LSE crystal): External low-speed oscillator of 32.768 kHz. LSE can be enabled and disabled by setting the LSEON bit in the RCC_BDCTLR register. The LSERDY bit indicates whether the LSE crystal oscillation is stable. The hardware will not send the clock to the system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

Figure 3-5 Low-speed External Crystal Circuit



- External low-speed clock source (LSE bypass): In this mode, the clock source is directly sent to the OSC32_IN pin from the outside, and the OSC32_OUT pin is suspended. The application program needs to set the LSEBYP bit , switch on the LSE bypass function and then set the LSEON bit when the LSEON bit is 0.

Figure 3-6 Low-speed Clock Source Circuit



3.3.4 PLL Clock

By configuring the RCC_CFGR0 register and the extension register EXTEND_CTR, 4 clock sources and multiplication factors can be selected for the internal PLL clock. These settings must be completed before the PLL is turned on. Once the PLL is turned on, these parameters cannot be changed. PLL can be enabled and disabled by setting the PLLON bit in the RCC_CTLR register. The PLLRDY bit indicates whether the PLL clock is stable. The hardware will not send the clock to the system until the PLLRDY bit is set to 1. If the PLLRDYIE bit of the RCC_INTR register is set, a corresponding interrupt will be generated.

If you need to use the USBD or USBHD module function in your application, the PLL must be set to output a 48MHz or 72MHz clock to provide a 48MHz USBCLK clock. Because the analog transceiving clock of the USBD or USBHD module is based on the PLL clock.

PLL clock source:

- HSI clock input
- HSI clock input through two frequency divisions
- HSE clock input
- HSE clock input through two frequency divisions

3.3.5 Bus/Peripheral Clock

1) System Clock (SYSCLK)

Configure the system clock source by configuring the SW[1:0] bits of the RCC_CFGR0 register. SWS[1:0] indicates the current system clock source.

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

After the controller is reset, the default HSI clock is selected as the system clock source. Switching between clock sources must occur after the target clock source is ready.

2) AHB/APB1/APB2 Bus Peripheral Clock (HCLK/PCLK1/PCLK2)

By configuring the HPRE[3:0], PPRE1[2:0], and PPRE2[2:0] bits of 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 each bit in the RCC_AHBRSTR, RCC_APB1PRSTR, and RCC_APB2PRSTR registers to restore them to the initial state.

Through each bit in RCC_AHBPCENR, RCC_APB1PCENR and RCC_APB2PCENR registers, the communication clock interface of different peripheral modules can be enabled or disabled separately. When using a peripheral, you shall firstly switch on its clock enable bit to access its register.

3) RTC Clock (RTCCLK)

By setting the RTCSEL[1:0] bits in the RCC_BDCTLR register, the RTCCLK clock source can be provided by the 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 area is reset.

- LSE is used as the RTC clock: Because LSE is at the backup domain and is powered by V_{BAT}, as long as V_{BAT} maintains supplying power, RTC will continuously work even though V_{DD} power supply is disconnected.
- LSI is used as the RTC clock: If the V_{DD} power supply is cut off, the RTC automatic wake-up cannot be guaranteed.
- HSE/128 is used as the RTC clock: If the VDD power supply is disconnected or the internal voltage regulator is turned off (the power supply in the 1.8V domain is switched off), the RTC status will be uncertain.

4) Independent Watchdog Clock

If the independent watchdog has been activated by the hardware configuration setting or software, the LSI oscillator will be forcibly switched on and cannot be switched off. After the LSI oscillator is stabilized, the clock will be supplied to the IWDG.

5) Clock Output(MCO)

The microcontroller allows to output a clock signal to the MCO pin. Configure the multiplexed push-pull output mode in the corresponding GPIO port register. By setting the MCO[2:0] bits in the RCC_CFGR0 register, the following four clock signals can be selected as the MCO clock output:

- System clock (SYSCLK) output
- HSI clock output
- HSE clock output
- PLL clock output through 2 frequency divisions

Note: Ensure that the output clock frequency does not exceed the maximum frequency of the I/O port 50MHz.

3.3.6 Clock Security System

The clock security system is an operation protection mechanism of the controller. It can switch to the HSI clock when the HSE clock fails to transmit, and generate an interrupt notification to allow the application software to complete the rescue operation.

Activate the clock security system by setting the CSSON bit of the RCC_CTLR register to 1. At this time, the clock monitor will be enabled after the HSE oscillator delay is started up (HSERDY=1) and will be switched off after the HSE clock is switched off. Once the HSE clock fails during system operation, the HSE oscillator will be turned off, and the clock failure event will be sent to the break input end of the advanced-control timers (TIM1 and TIM8). A clock safety interrupt will be generated, the CSSF bit will be set to 1, and the application program will enter the non-maskable interrupt (NMI). The CSSF bit flag can be cleared by setting the CSSC bit, and the NMI interrupt pending bit can be cancelled.

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

3.4 Extension Configuration

The system provides EXTEND extension configuration unit (EXTEND_CTR register). When AHB clock is used for this unit, the reset action will be executed only during the system power-on reset. It mainly includes the following extension control bit functions:

1) Adjusting the built-in voltage: The default values shall be selected for the LDOTRIM and ULLDOTRIM, and their values can be modified during the adjustment of performance and power consumption.
2) PLL clock selection: The HSIPRE cooperates with the original clock configuration register to provide the choice of frequency division or not for the HSI clock as the input clock of the PLL.
3) Lock-up function monitoring: When the LKUPEN field is enabled, the Lock-up monitoring of the system will be switched on. Once a Lock-up situation occurs, the system will perform software reset and set the LKUPRESET field to 1. After reading it, you can write 1 to clear this flag.
4) Built-in resistor and communication speed control of USBD module: USB full-speed device controller (USBD) selects whether to use the built-in pull-up resistor (1.5KΩ) through the USBDPU field. When it is not enabled, a pull-up resistor needs to be connected to the USB pin (connected to UD- pin in low-speed mode, connected to UD+ pin in full-speed mode). The current USB device speed mode is configured for USBDLS. In order to better match the USB signal, USB5VSEL will be set to 1 when the system supplies power at the rating of 5V, and USB5VSEL will be cleared to 0 when the system supplies power at the rating of 3.3 V.
5) USBHD interface pin selection: USBHD full-speed host/device controller needs to switch on the USBHDIO control bit, and use the PB6/PB7 pin as the USB signal communication function. In order to better match the USB signal, USB5VSEL will be set to 1 when the system supplies power at the rating of 5V, and USB5VSEL will be cleared to 0 when the system supplies power at the rating of 3.3 V.

3.5 Register Description

Table 3-1 List of RCC Related 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

Table 3-2 List of EXTEND Related Registers

Name	Access address	Description	Reset value
R32_EXTEND_CTR	0x40023800	Configuration extension control register	0x00000020

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						PLL RDY	PLLON	Reserved				CSSON	HSE BYP	HSE RDY	HSEON

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSICAL[7:0]								HSITRIM[4:0]					Reserved	HSI RDY	HSION

Bit	Name	Access	Description	Reset value
[31:26]	Reserved	RO	Reserved.	0
25	PLLRDY	RO	PLL clock ready lock flag bit (set by hardware): 1: PLL clock locked; 0: PLL clock unclocked.	0
24	PLLON	RW	PLL clock enable control bit: 1: Enable PLL clock; 0: Disable PLL clock. *Note: After entering the stop or standby low-power mode, this bit is cleared by hardware.*	0
[23:20]	Reserved	RO	Reserved.	0
19	CSSON	RW	Clock security system enable control bit: 1: Enable clock security system. When the HSE is ready (HSERDY is set to 1), the hardware will switch on the HSE clock monitoring function. When the HSE abnormally is found, the CSSF flag and NMI interrupt will be interrupted; when the HSE is not ready, the hardware will switch off the HSE clock monitoring function. 0: Disable clock security system.	0
18	HSEBYP	RW	External high-speed crystal bypass control bit: 1: Bypass external high-speed crystal/ceramic resonator (using external clock source); 0: Not bypass external high-speed crystal/ceramic resonators.	0

			Note: This bit shall be written when HSEON is 0.	
17	HSERDY	RO	External high-speed crystal oscillation stability ready flag bit (set by hardware): 1: Stable external high-speed crystal oscillation; 0: Unstable external high-speed crystal oscillation; *Note: After the HSEON bit is cleared, it takes 6 HSE cycles to clear this bit.*	0
16	HSEON	RW	External high-speed crystal oscillation 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	RO	Internal high-speed clock calibration value, automatically initialized during the system startup.	xxh
[7:3]	HSITRIM	RW	Internal high-speed clock adjustment value: The user can input an adjustment value to superimpose on the HSICAL[7:0] value, and adjust the frequency of the internal HSI RC oscillator according to changes in voltage and temperature. The default value is 16, HSI can be adjusted to 8MHz±1%; the change of HSICAL is adjusted to about 40kHz per step.	10000
2	Reserved	RO	Reserved.	0
1	HSIRDY	RO	Stability ready flag bit of internal high-speed clock (8MHz) (set by hardware): 1: Stable internal high-speed clock (8MHz); 0: Unstable internal high-speed clock (8MHz). *Note: After the HSION bit is cleared, it takes 6 HSI cycles to clear this bit.*	1
0	HSION	RW	Internal high-speed 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 used as the system clock fails, this bit will be set by hardware to start the internal 8MHz RC oscillator.*	1

Clock configuration register 0 (RCC_CFGR0)

Offset address: 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					MCO[2:0]			Reserved	USB PRE	PLLMUL[3:0]				PLL XTPRE	PLL SRC

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
[31:27]	Reserved	RO	Reserved.	0
[26:24]	MCO	RW	Microcontroller MCO pin clock output control: 0xx: No clock output; 100: System clock (SYSCLK) output; 101: Internal 8MHz RC oscillator clock (HSI) output; 110: External oscillator clock (HSE) output; 111: Output after PLL clock two frequency divisions *Note: When the MCO clock is started up or switched, several cycles of the clock may be lost. Ensure that the output clock frequency does not exceed 50MHz (I/O port maximum frequency).*	0
23	Reserved	RO	Reserved.	0
22	USBPRE	RW	USBD/USBHD module prescaler clock configuration: 1: PLL clock is directly used as the USBD/USBHD module clock; 0: PLL clock 1.5 frequency divisions as the USBD/USBHD module clock. *Note: 48MHz is required for the USBD/USBHD module clock. This bit must be configured (in the RCC_AHBPCENR and RCC_APB1PCENR registers) before the USBD and USBHD clocks are enabled.*	0
[21:18]	PLLMUL	RW	PLL clock frequency multiplication factor (it can be written only when the PLL is switched off): 0000: PLL output frequency x 2; 0001: PLL output frequency x 3; 0010: PLL output frequency x 4; 0011: PLL output frequency x 5; 0100: PLL output frequency x 6; 0101: PLL output frequency x 7; 0110: PLL output frequency x 8; 0111: PLL output frequency x 9; 1000: PLL output frequency x 10; 1001: PLL output frequency x 11; 1010: PLL output frequency x 12; 1011: PLL output frequency x 13; 1100: PLL output frequency x 14; 1101: PLL output frequency x 15; 1110: PLL output frequency x 16; 1111: PLL output frequency x 16. *Note: The output frequency of PLL cannot exceed 72MHz.*	0
17	PLLXTPRE	RW	HSE divider for PLL entry (it can be written only when PLL is disabled) 1: HSE clock divided by 2 for PLL entry; 0: HSE clock not divided for PLL entry.	0

16	PLLSRC	RW	PLL entry clock source (it can be written only when the PLL is switched off): 1: HSE clock not divided or divided by 2 for PLL entry; 0: HSI clock not divided or divided by 2 for PLL entry.	0
[15:14]	ADCPRE	RW	ADC clock source prescaler control: 00: PCLK2 divided by 2 as the ADC clock; 01: PCLK2 divided by 4 as the ADC clock; 10: PCLK2 divided by 6 as the ADC clock; 11: PCLK2 divided by 8 as the ADC clock; *Note: ADC clock shall not exceed 14MHz at most.*	0
[13:11]	PPRE2	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	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	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	RO	System clock (SYSCLK) status (set by hardware): 00: The system clock source is HSI; 01: The system clock source is HSE; 10: The system clock source is PLL; 11: Not applicable.	0
[1:0]	SW	RW	System clock source selection: 00: HSI used as the system clock; 01: HSE used as the system clock; 10: PLL output used as the system clock; 11: Not applicable. *Note: When the system returns from standby and stop modes or when the external oscillator HSE used as the*	0

			system clock fails after the clock security system is enabled, HSI is forcibly selected as the system clock by hardware.	

Clock interrupt register (RCC_INTR)

Offset address: 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								CSSC	Reserved		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			PLL RDYIE	HSE RDYIE	HSI RDYIE	LSE RDYIE	LSI RDYIE	CSSF	Reserved		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	Clear the clock security system interrupt flag bit (CSSF): 1: Clear CSSF interrupt flag; 0: No effect.	0
[22:21]	Reserved	RO	Reserved.	0
20	PLLRDYC	WO	Clear PLL ready interrupt flag bit: 1: Clear the PLLRDYF interrupt flag; 0: No effect.	0
19	HSERDYC	WO	Clear the HSE oscillator ready interrupt flag bit: 1: Clear the HSERDYF interrupt flag; 0: No effect.	0
18	HSIRDYC	WO	Clear the HSI oscillator ready interrupt flag bit: 1: Clear HSIRDYF interrupt flag; 0: No effect.	0
17	LSERDYC	WO	Clear the LSE oscillator ready interrupt flag bit: 1: Clear LSERDYF interrupt flag; 0: No effect.	0
16	LSIRDYC	WO	Clear the LSI oscillator ready interrupt flag bit: 1: Clear LSIRDYF interrupt flag; 0: No effect.	0
[15:13]	Reserved	RO	Reserved.	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:	0

			1: Enable LSE ready interrupt; 0: Disable LSE ready interrupt;	
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: The HSE clock fails and a clock security interrupt CSSI is generated; 0: No clock security system interrupt. Set by hardware. Write 1 to CSSC bit by software to clear.	0
[6:5]	Reserved	RO	Reserved.	0
4	PLLRDYF	RO	PLL clock ready lock interrupt flag: 1: PLL clock lock generates interrupt; 0: No PLL clock lock interrupt. Set by hardware. Write 1 to PLLRDYC by software to clear.	0
3	HSERDYF	RO	HSE clock ready interrupt flag: 1: Enable the HSE clock ready interrupt; 0: No HSE clock ready interrupt. Set by hardware. Write 1 to HSERDYC bit by software to clear.	0
2	HSIRDYF	RO	HSI clock ready interrupt flag: 1: Enable the HSI clock ready interrupt; 0: No HSI clock ready interrupt. Set by hardware. Write 1 to HSIRDYC bit by software to clear.	0
1	LSERDYF	RO	LSE clock ready interrupt flag: 1: Enable the LSE clock ready interrupt; 0: No LSE clock ready interrupt. Set by hardware. Write 1 to LSERDYC bit by software to clear.	0
0	LSIRDYF	RO	LSI clock ready interrupt flag: 1: Enable the LSI clock ready interrupt; 0: No LSI clock ready interrupt. Set by hardware. Write 1 to LSIRDYC bit by software to clear.	0

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

Bit	Name	Access	Description	Reset value
[31:15]	Reserved	RO	Reserved.	0
14	USART1RST	RW	USART1 reset: 1: Reset USART1; 0: No effect.	0
13	Reserved	RO	Reserved.	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	Reserved	RO	Reserved.	0
9	ADCRST	RW	ADC reset l: 1: Reset ADC; 0: No effect.	0
[8:6]	Reserved	RO	Reserved.	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	Auxiliary function module reset of IO: 1: Reset module; 0: No effect.	0

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	Reserved	CAN RST	Reserved	USBD RST	I2C2 RST	I2C1 RST	Reserved		USART3 RST	USART2 RST	Reserved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	SPI2 RST	Reserved		WWDG RST	Reserved								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:	0

			1: Reset Power interface; 0: No effect.	
27	BKPRST	RW	Backup interface reset: 1: Reset back-up interface; 0: No effect.	0
26	Reserved	RO	Reserved.	0
25	CANRST	RW	CAN module reset: 1: Reset CAN; 0: No effect.	0
24	Reserved	RO	Reserved.	0
23	USBDRST	RW	USBD module reset: 1: Reset USBD; 0: No effect.	0
22	I2C2RST	RW	I2C2 interface reset: 1: Reset I2C2; 0: No effect.	0
21	I2C1RST	RW	I2C1 interface reset: 1: Reset I2C1; 0: No effect.	0
[20:19]	Reserved	RO	Reserved.	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
[16:15]	Reserved	RO	Reserved.	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
[10:3]	Reserved	RO	Reserved.	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

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															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			USBHDEN	Reserved					CRCEN	Reserved	FLITFEN	Reserved	SRAMEN	Reserved	DMAEN

Bit	Name	Access	Description	Reset value
[31:13]	Reserved	RO	Reserved.	0
12	USBHDEN	RW	USBHD module clock enable bit: 1: Enable module clock; 0: Disable module clock.	0

[11:7]	Reserved	RO	Reserved.	0
6	CRCEN	RW	CRC module clock enable bit: 1: Enable module clock; 0: Disable module clock.	0
5	Reserved	RO	Reserved.	0
4	FLITFEN	RW	Flash memory interface module clock enable bit: 1: Enable flash memory module clock in sleep mode; 0: Disable flash memory module clock in sleep mode.	1
3	Reserved	RO	Reserved.	0
2	SRAMEN	RW	SRAM interface module clock enable bit: 1: Enable SRAM interface module clock in sleep mode; 0: Disable SRAM interface module clock in sleep mode.	1
1	Reserved	RO	Reserved.	0
0	DMAEN	RW	DMA module clock enable bit: 1: Enable module clock; 0: Disable module clock.	0

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

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															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	USART1 EN	Reserved	SPI1 EN	TIM1 EN	Reserved	ADC EN	Reserved			IOPD EN	IOPC EN	IOPB EN	IOPA EN	Reserved	AFIO EN

Bit	Name	Access	Description	Reset value
[31:15]	Reserved	RO	Reserved.	0
14	USART1EN	RW	USART1 interface clock enable bit: 1: Enable module clock; 0: Disable module clock.	0
13	Reserved	RO	Reserved.	0
12	SPI1EN	RW	SPI1 interface clock enable bit: 1: Enable module clock; 0: Disable module clock.	0
11	TIM1EN	RW	TIM1 module clock enable bit: 1: Enable module clock; 0: Disable module clock.	0
10	Reserved	RO	Reserved.	0
9	ADCEN	RW	ADC module clock enable bit: 1: Enable module clock; 0: Disable module clock.	0
[8:6]	Reserved	RO	Reserved.	0
5	IOPDEN	RW	PD port module clock enable bit of IO: 1: Enable module clock; 0: Disable module clock.	0
4	IOPCEN	RW	PC port module clock enable bit of IO: 1: Enable module clock; 0: Disable module clock.	0
3	IOPBEN	RW	PB port module clock enable bit of IO: 1: Enable module clock; 0: Disable module clock.	0

2	IOPAEN	RW	PA port module clock enable bit of IO: 1: Enable module clock; 0: Disable module clock.	0
1	Reserved	RO	Reserved.	0
0	AFIOEN	RW	Auxiliary function module clock enable bit of IO: 1: Enable module clock; 0: Disable module clock.	0

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

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	Reserved	CAN EN	Reserved	USBD EN	I2C2 EN	I2C1 EN	Reserved		USART3 EN	USART2 EN	Reserved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	SPI2 EN	Reserved		WWDG EN	Reserved								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: Enable module clock; 0: Disable module clock.	0
28	PWREN	RW	Power interface module clock enable bit: 1: Enable module clock; 0: Disable module clock.	0
27	BKPEN	RW	Backup interface clock enable bit: 1: Enable module clock; 0: Disable module clock.	0
26	Reserved	RO	Reserved.	0
25	CANEN	RW	CAN module clock enable bit: 1: Enable module clock; 0: Disable module clock.	0
24	Reserved	RO	Reserved.	0
23	USBDEN	RW	USBD module clock enable bit: 1: Enable module clock; 0: Disable module clock.	0
22	I2C2EN	RW	I2C2 interface clock enable bit: 1: Enable module clock; 0: Disable module clock.	0
21	I2C1EN	RW	I2C1 interface clock enable bit: 1: Enable module clock; 0: Disable module clock.	0
[20:19]	Reserved	RO	Reserved.	0
18	USART3EN	RW	USART3 interface clock enable bit: 1: Enable module clock; 0: Disable module clock.	0
17	USART2EN	RW	USART2 interface clock enable bit: 1: Enable module clock; 0: Disable module clock.	0
[16:15]	Reserved	RO	Reserved.	0
14	SPI2EN	RW	SPI2 interface clock enable bit: 1: Enable module clock; 0: Disable module clock.	0
[13:12]	Reserved	RO	Reserved.	0
11	WWDGEN	RW	Window watchdog clock enable bit:	0

			1: Enable module clock; 0: Disable module clock.	
[10:3]	Reserved	RO	Reserved.	0
2	TIM4EN	RW	Timer4 module clock enable bit: 1: Enable module clock; 0: Disable module clock.	0
1	TIM3EN	RW	Timer3 module clock enable bit: 1: Enable module clock; 0: Disable module clock.	0
0	TIM2EN	RW	Timer2 module clock enable bit: 1: Enable module clock; 0: Disable module clock.	0

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

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					LSE BYP	LSE RDY	LSEON

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved.	0
16	BDRST	RW	Backup domain software reset control: 1: Reset the whole backup domain. 0: Cancel the reset.	0
15	RTCEN	RW	RTC clock enable control: 1: Enable RTC clock; 0: Disable RTC clock. *Note: The RTC clock can be enabled only when RTCSEL!=0. Otherwise, the hardware will be forced to 0.*	0
[14:10]	Reserved	RO	Reserved.	0
[9:8]	RTCSEL	RW	RTC clock source selection: 00: No clock; 01: LSE oscillator clock used as RTC clock; 10: LSI oscillator clock used as RTC clock; 11: HSE oscillator clock divided by 128 used as RTC clock. *Note: Once the RTC clock source is 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	Bypass control bit of external low-speed crystal (LSE): 1: Bypass external low-speed crystal/ceramic resonator (using external clock source); 0: Not bypass low-speed external crystal/ceramic resonators.	0

			Note: This bit shall be written when LSEON is 0.	
1	LSERDY	RO	External low-speed crystal oscillation stability ready flag bit (set by hardware): 1: Stable external low-speed crystal oscillation; 0: Unstable external low-speed crystal oscillation; *Note: After the LSEON bit is cleared, it takes 6 LSE cycles to clear this bit.*	0
0	LSEON	RW	External low-speed crystal oscillation enable control bit: 1: Enable LSE oscillator; 0: Disable LSE oscillator;	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.

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	RO	Low-power reset flag: 1: Enable the low-power reset; 0: Disable low-power reset. When low-power management reset occurs, set it to 1 by hardware; write RMVF bit by software to clear.	0
30	WWDGRSTF	RO	Window watchdog reset flag: 1: Enable window watchdog reset; 0: Disable window watchdog reset. When the window watchdog reset occurs, set it to 1 by hardware; write RMVF bit by software to clear.	0
29	IWDGRSTF	RO	Independent watchdog reset flag: 1:Enable independent watchdog reset; 0: Disable independent watchdog reset. When independent watchdog reset occurs, set it to 1 by hardware; write RMVF bit by software to clear.	0
28	SFTRSTF	RO	Software reset flag: 1: Enable the software reset; 0: Disable the software reset. When software reset occurs, set it to 1 by hardware; write	0

			RMVF bit by software to clear.	
27	PORRSTF	RO	Power-on/power-down reset flag: 1: Enable power-on/power-down reset; 0: Disable power-on/power-down reset. During the power-on/power-down reset, set it to 1 by the hardware; write RMVF bit by software to clear.	1
26	PINRSTF	RO	External manual reset (NRST pin) flag: 1: Enable NRST pin reset; 0: Disable NRST pin reset. During the NRST pin reset, set it to 1 by the hardware; write RMVF bit by software to clear.	1
25	Reserved	RO	Reserved.	0
24	RMVF	RW	Clear reset flag control: 1: Clear reset flag; 0: No effect.	0
[23:2]	Reserved	RO	Reserved.	0
1	LSIRDY	RO	Stability ready flag bit of internal high-speed clock (LSI) (set by hardware): 1: Stable internal low-speed clock (40 kHz); 0: Unstable internal low-speed clock (40 kHz); *Note: After the LSION bit is cleared, it takes 3 LSI cycles to clear this bit.*	0
0	LSION	RW	Internal low-speed clock (LSI) enable control bit: 1: Enable LSI (40kHz) oscillator; 2: Disable LSI (40kHz) oscillator.	0

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

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			USBHDRST	Reserved											

Bit	Name	Access	Description	Reset value
[31:13]	Reserved	RO	Reserved.	0
12	USBHDRST	RW	USBHD module reset control: 1: Reset module; 0: No effect.	0
[11:0]	Reserved	RO	Reserved.	0

Configure the extension control register (EXTEND_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					LDO TRIM	ULLDO TRIM[1:0]		LKUP RESET	LKUP EN	Reserved	HSI PRE	USB 5VSEL	USBHD IO	USBD PU	USBD LS

Bit	Name	Access	Description	Reset value
[31:11]	Reserved	RO	Reserved.	0
10	LDOTRIM	RW	Adjust the digital core voltage value: 1:1.62V; 0:1.5V (recommended).	0
[9:8]	ULLDOTRIM	RW	Adjust the ULLDO voltage value in the low power mode	10b
7	LKUPRESET	RW1	LOCKUP reset flag: 1: LOCKUP causes system reset; write 1 to clear it; 0: Normal.	0
6	LKUPEN	RW	LOCKUP monitoring function: 1: Enable; the system will execute reset when system lock-up occurs and set the bit of LOCKUP_RESET; 0: Disable.	0
5	Reserved	RO	Reserved.	0
4	HSIPRE	RW	Whether the frequency of HSI clock is divided: (it can only be written when the PLL is turned off) 1: HSI clock is used as PLL input clock; 0: The HSI clock is used as the PLL input clock through the 2 frequency divisions.	0
3	USB5VSEL	RW	The USB functional configuration is used under the power supply of different systems: 1: System rated power supply (V_{DD}) 5V; 0: System rated power supply (V_{DD}) 3.3V.	0
2	USBHDIO	RW	PB6/PB7 pin function configuration: 1: Multiplexing as USBHD function; 0: Other functions	0
1	USBDPU	RW	Whether the internal pull-up resistor of USBD is enabled: 1: Enable (the external pull-up resistor is not required); 0: Disable (external pull-up resistor shall be connected)	0
0	USBDLS	RW	Selection of USBD operation mode: 1: Low-speed mode; 0: Full-speed mode.	0

Chapter 4 Backup Register (BKP)

The module description in this chapter applies to the full range of CH32F103 and CH32V103 microcontrollers.

The backup register (BKP) provides ten 16-bit general data registers that can be used to store 20 bytes of user data. After the main power supply (V_{DD}) is 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 intrusion detection management, RTC clock calibration and pulse output functions.

4.1 Main Features

- Provide 20-byte backup data register
- Intrusion detection (TAMPER) function
- RTC clock calibration function
- Output RTC clock 64-frequency-division alarm pulse or second pulse on PC13 pin

4.2 Functional Specification

After the microcontroller is 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 of the register RCC_APB1PCENR to switch on the power supply and the operating clock of the backup interface;
2) Set the DBP bit of 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 all cleared after the intrusion event occurs.

4.2.2 Instrusion Detection

Intrusion detection means that the hardware will automatically clear the important information retained in the current system when a signal (rising or falling edge) is provided externally, indicating that there is an "intrusion event". Through this mode, the security of the system information can be added.

An intrusion event will be generated when a transition edge (depending on the TPAL bit) appears on the intrusion detection pin. If the intrusion detection interrupt is enabled, an intrusion detection interrupt will also be generated at the same time. As long as there is an intrusion event, the backup data register will be all cleared. In addition, the hardware detection adopts the memory method. Even if the intrusion detection function is not enabled (TPE=0), the system will sample and check whether there is a transition edge, and if the TPAL bit selection is met, the intrusion event will be locked in advance, and the TPE position will be set to 1 to trigger an intrusion event.

For example: When TPAL=0, the function will not be enabled if TPE=0, but the TAMPER pin is already at the high level. Once TPE=1, an additional intrusion event will be generated (the system locks the rising edge in advance). When TPAL=1, the function will not be enabled if TPE=0, but the TAMPER pin is already at

the low level. Once TPE=1, an additional intrusion event will be generated (the system locks the falling edge in advance).

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

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

4.2.3 RTC Calibration

For this function, the intrusion detection pin shall serve as a normal IO port. Clear the TPE bit of configured BKP_TPCTLR register.

- Pulse output

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

- RTC calibration

After the CCO bit of the BKP_OCTLR register is configured, the internal RTC clock will be outputted to the instrusion detection pin (TAMPER) after 64 frequency divisions. Through the actual test, the software coordinates to modify the CAL[6:0] bit to adjust the clock and calibrate the RTC.

4.2.4 BKP Interface Reset

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

4.3 Register Description

Table 4-1 List of BKP Related 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	Intrusion detection control register	0x0000

R16_BKP_TPCSR	0x40006C34	Intrusion detection status register	0x0000

Backup data register (BKP_DATARx) (x=1-10)

Offset address: 0x04-0x28

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
[15:0]	DATA	RW	Backup data, which can be called by the user program. *Note: They are only reset by the backup domain reset (BDRST) or (if the intrusion detection pin TAMPER function is enabled) by the intrusion pin event.*	0

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 pulse output selection. 1: Output second pulse; 0: Output alarm pulse. *Note: This bit can only be reset by the backup domain reset (BDRST).*	0
8	ASOE	RW	The TAMPER pin enable pulse output bit 1: Disable the output alarm pulse or second pulse; 0: Enable the output alarm pulse or second pulse. *Note: This bit can only be reset by the backup domain reset (BDRST).*	0
7	CCO	RW	Calibration clock output selection bit 1: RTC clock of TEMPER pin output through the 64 frequency divisions; 0: Not outputting the calibration clock. *Note 1: The intrusion setection function must be switched off to enable such function.* *Note2: Such bit will be cleared when V_{DD} power supply is switched off.*	0
[6:0]	CAL	RW	Calibration value register. The value of this register indicates number of skipped pulses per 220 clock pulses. This function is used to calibrate the RTC clock. RTC clock can be reduced by 0~121ppm.	0

Intrusion 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	Effective level setting of intrusion detection pin (TEMPER pin) 0: The high level on the intrusion detection pin will clear all backup data registers (hardware lock rising edge); 1: The low level on the intrusion detection pin will clear all backup data registers (hardware lock falling edge);	0
0	TPE	RW	Intrusion detection pin enable bit 0: TEMPER pin is used as ordinary IO port; 1: TEMPER pin is used for the intrusion detection.	0

Note: Clearing the TPAL and TPE bits at the same time will generate a false intrusion event. It is recommended to change the status of the TPAL bit only when TPE is 0.

Intrusion detection status register (BKP_TPCTLR)

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	Intrusion interrupt flag bit; when an intrusion event is detected and the TPIE bit is set to 1, this bit will be set. Clear this flag bit by writing 1 to the CTI bit. If the TPIE bit is reset, then this bit will also be 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	Intrusion event flag bit; when an intrusion event is detected, this bit will be set. This bit is cleared by writing 1 to the CTE bit. *Note: When this bit is 1, all BKP_DATARx register values will be 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	Instrusion interrupt enable bit:	0

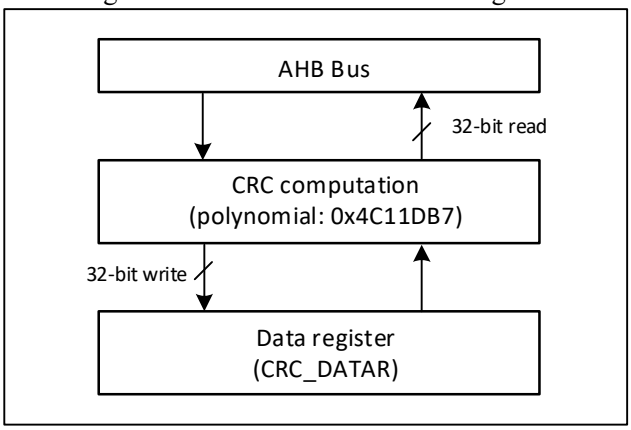
			0: Disable instrusion detection interrupt; 1: Enable instrusion detection interrupt (TPE needs to be set to 1). *Note 1: The intrusion interrupt cannot wake up the core from low-power mode.* *Note 2: This bit is reset only when the system is reset or woken up from standby mode.*	
1	CTI	WO	Intrusion detection interrupt clearing bit; write 1 to clear it, and the reading is invalid.	0
0	CTE	WO	Intrusion detection clearing bit; write 1 to clear it, and the reading is invalid.	0

Chapter 5 Cyclic Redundancy Check (CRC)

The module description in this chapter applies to the full range of CH32F103 and CH32V103 microcontrollers.

The calculation unit of cyclic redundancy check (CRC) obtains the CRC calculation results of any 32-bit data according to a fixed generator polynomial. It is generally used in the fields of data storage and data communication to verify the correctness of data. 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) is used:
  $X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$;
- The same 32-bit register is used as data input and CRC32 calculation output
- Single conversion time: 4 AHB clock cycles (HCLK)

5.2 Functional Description

- CRC unit reset

To start the CRC calculation for a new data group, you need to reset the CRC calculation unit. Write 1 to the RST bit of the control register CRC_CTLR, and the hardware will reset the data register and restore the initial value 0xFFFFFFFF.

- CRC calculation

The calculation of the CRC unit is the previous CRC calculation result and the CRC result of the newly involved data. Write operation to CRC_DATAR data register to send the new data to the hardware calculation unit; execute read operation to acquire the latest round of CRC calculation value. The hardware calculation will interrupt the write operation of the system, so new values can be written continuously.

Note: The CRC unit calculates the entire 32-bit data, rather than byte by 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 List of CRC Related Registers

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

Data register (CRC_DATAR)

Offset address: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATAR															
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
[31:0]	DATA	RW	Write the original data; read the calculation result.	0xFFFFFFFF

Independent data buffer (CRC_IDATAR)

Offset address: 0x04

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

Bit	Name	Access	Description	Reset value
[7:0]	IDATA	RW	An 8-bit general register can be used as a data buffer. This register is not affected by RST of the control register.	0

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 execute, and it will be automatically cleared by	0

			hardware. After execution, the data register is 0xFFFFFFFF.	

Chapter 6 Real-Time Clock (RTC)

The module description in this chapter applies to the full range of CH32F103 and CH32V103 microcontrollers.

The real-time clock (RTC) is an independent timer module. Its programmable counter can reach up to 32 bits. With software, the real-time clock function can be realized, and the counter value can be modified to reconfigure the current time and date of the system. The RTC module is in the backup power supply area, and will not be affected by the system reset and standby mode wake-up.

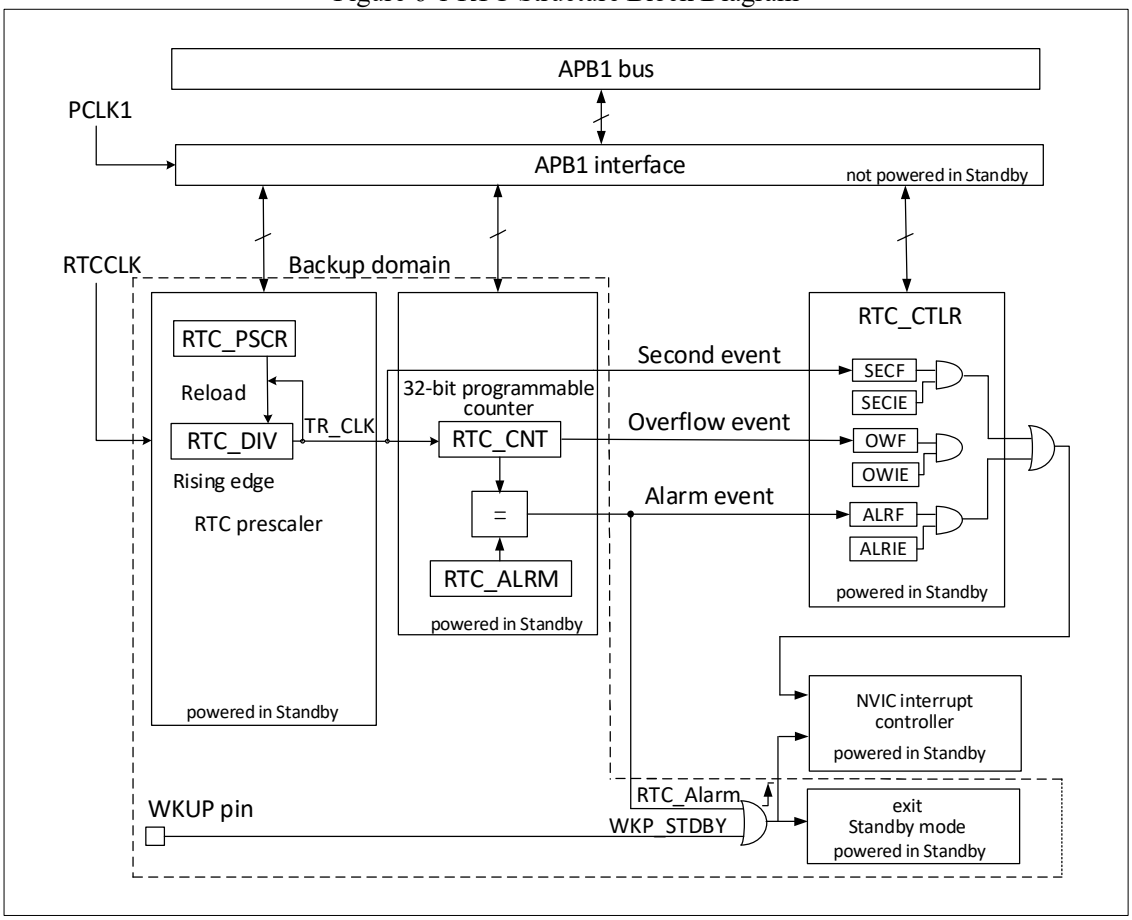
6.1 Main Features

- Maximum prescale factor: 2^{20}
- 32-bit programmable counter
- Multiple 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 three parts: APB1 bus interface, frequency divider and counter, control and status register. The frequency divider and counter are in the backup area and can be powered by V_{BAT}. After RTCCK is inputted to the frequency divider (RTC_DIV), the frequency is

divided into TR_CLK. It is worth noting that a self-decrement counter is located in the frequency divider (RTC_DIV). When the self-decrement reaches the overflow, a TR_CLK will be outputted. Then, take the preset value from the reload value register (RTC_PSCR) and reload it into the freqeuency divider. Reading the frequency divider actually means reading its real-time value (read only). The frequency division factor shall be written to the reload value register (RTC_PSCR). Generally, the period of TR_CLK is set to 1 second, TR_CLK will trigger the second event, and 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 alarm register, the alarm event will be triggered; when the main counter is incremented to overflow, an overflow event will be triggered. The above three events can trigger an interrupt, and there is a corresponding interrupt enable bit control.

6.2.2 Reset

Due to the special purpose of the real-time clock, the four sets of registers in the backup domain: prescaler, prescale reload value, main counter and alarm clock can only be reset by the reset signal of 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 and 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) Reset 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 becomes 1. So far, the write is completed;

6.3 Register Description

Table 6-1 List of RTC Related Registers

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

R16_RTC_ALRML	0x40002824	Alarm clock register low bit	0xFFFF

RTC control register high bit (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 bit.	0
1	ALRIE	RW	Alarm clock interrupt enable bit.	0
0	SECIE	RW	Second interrupt enable bit.	0

RTC control register low bit (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	The RTC operation status indicator bit indicates the execution status of the last operation on the RTC. The operation of the RTC must be made when this bit is 1. 1: The previous operation for RTC has been completed; 0: The previous operation for RTC is in progress.	1
4	CNF	RW	Configure the flag bit, and write this bit to enter the configuration mode, so as to allow to write values to the counter (R16_RTC_CNTx), alarm register (R16_RTC_ALRMx) and prescaler reload value register (R16_RTC_PSCRx). The write operation can be executed only when this bit is written to 1 and cleared by the software: 1: Enter the configuration mode; 0: Exit the configuration mode, and start updating the RTC register.	0
3	RSF	RW0	The register synchronization flag bit shall be set by the hardware before reading and writing the prescaler (PSCRx), alarm (ALRMx) and counter (CNTx) to ensure that these registers have been synchronized; when reading or writing these registers, or after the APBI is reset or APB1 clock is stopped, the bit shall be	0

			reset firstly. 1: The register is synchronized; 0: The register is not synchronized.	
2	OWF	RW0	Counter overflow flag. When the 32-bit counter overflows, this bit will be set by hardware. If the OWIE bit is set, an overflow interrupt will also be generated. This bit can only be cleared by software and cannot be set by software.	0
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 will also be generated. This bit can only be cleared by software and cannot be set by software.	0
0	SECF	RW0	Second event flag. When the clock generates a falling edge after frequency division 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 will be also generated at the same time. This bit can only be cleared by software and cannot be set by software.	0

Prescaler reload value register high bit (RTC_PSCRH)

Offset address: 0x08

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

Bit	Name	Access	Description	Reset value
[15:4]	Reserved	RO	Reserved.	0
[3:0]	PSCRH	WO	Reload value high bit.	0

Prescaler reload value register low bit (RTC_PSCRL)

Offset address: 0x0C

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

Bit	Name	Access	Description	Reset value
[15:0]	PSCRL	WO	Reload value lower bit. The actual frequency division factor is (PSCR[19:0]+1). For example, if the RTC input frequency is 32768Hz, then this value will be set to 0x7fff to divide the signal with a 1-second cycle.	8000h

Frequency divider register high bit (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]	DIVH	RO	Frequency divider register high bit.	0

Frequency divider register low bit (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]	DIVL	RO	Frequency divider register lower bit. 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 frequency divider is read.	8000h

RTC counter high bit (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]	CNTH	RW	Counter high bit	0

RTC counter low bit (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]	CNTL	RW	The low bit of the counter, the core component of the RTC timer; the clock is provided with TRCLK (the period is generally set to 1 second). Calculate current	0

			time by reading CNT[31:0]. Enter the configuration mode to write this value.	

Alarm register high bit (RTC_ALRMH)

Offset address: 0x20

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

Bit	Name	Access	Description	Reset value
[15:0]	ALRMH	WO	Alarm clock register high bit	FFFFh

Alarm register lower bit (RTC_ALRML)

Offset address: 0x24

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

Bit	Name	Access	Description	Reset value
[15:0]	ALRML	WO	Alarm clock register lower bit When the value of the alarm register ALRM[31:0] is consistent with the value of the counter CNT[31:0], an alarm event will be generated. Enter the configuration mode to modify this value.	FFFFh

Chaper 7 Independent Watchdog (IWDG)

The module description in this chapter applies to the full range of CH32F103 and CH32V103 microcontrollers.

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 suitable for occasions with low precision requirements.

7.1 Main Features

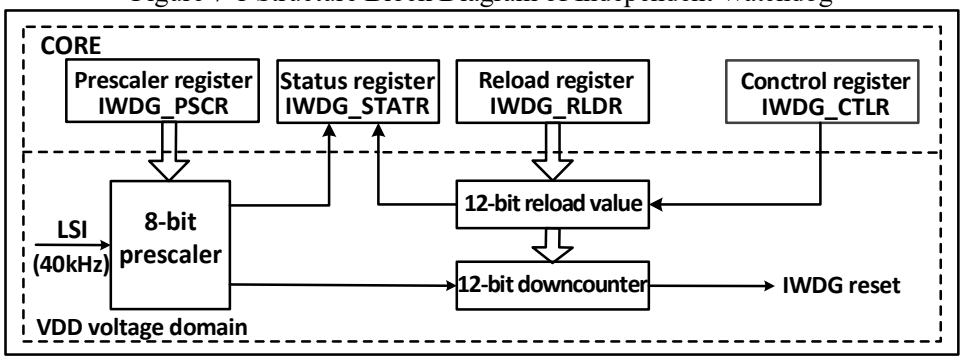
- 12-bit self-decrement type counter
- The clock source LSI frequency division can run in low-power mode
- Reset conditions: Reduce the counter value to 0

7.2 Functional Specification

7.2.1 Principle and Using Method

Different from the window watchdog, the clock of the independent watchdog originates from LSI clock frequency division, and its function can still work normally in the stop and standby mode. When the watchdog counter decreases to 0, a system reset will be generated, so the timeout period is (reload value + 1) clocks, with a maximum of 26.2s and a minimum of 100us.

Figure 7-1 Structure Block Diagram of Independent Watchdog



- Start up the independent watchdog

After the system is reset, the watchdog will be at the off status. Write 0xCCCC to the IWDG_CTLR register to switch on the watchdog, and then it can no longer be turned off unless a reset occurs.

If the hardware independent watchdog enable bit (IWDG_SW) is enabled in the user selection word, the IWDG will be permanently enabled after the microcontroller is reset.

- Watchdog configuration

A 12-bit counter runs through progress decrease in the watchdog. When the value of counter is reduced to 0, the system reset will be generated. To switch on the IWDG functions, the following operations shall be made:

1) Counting time base: IWDG clock source LSI; set the LSI frequency division value clock as IWDG counting time base through IWDG_PSCR register. The operation method is to write 0x5555 to IWDG_CTLR, and then modify the frequency division value in the IWDG_PSCR register. The PVU bit in the IWDG_STATR register indicates the update status of the frequency division value. The frequency

division value can be modified and read only after the update is completed.

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. The operation method is to firstly write 0x5555 to the IWDG_CTLR register, and then modify the IWDG_RLDR register to set the target reload value. The RUV bit in the IWDG_STATR register indicates the update status of the reload value. The IWDG_RLDR register can be modified and read only after the update is completed.
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 counter decreases to 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 will occur.

7.2.2 Debug Mode

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

7.3 Register Description

Table 7-8 List of IWDG Related Registers

Name	Access address	Description	Reset value
R16_IWDG_CTLR	0x40003000	Control register	0x0000
R16_IWDG_PSCR	0x40003004	Frequency division factor register	0x0000
R16_IWDG_RLDR	0x40003008	Reload value register	0x7FFF
R16_IWDG_STATR	0x4000300C	Status register	0x0000

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	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 (the user selects word configuration), there is no such restriction.	0

Frequency division factor register (IWDG_PSCR)

Offset address: 0x04

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

Bit	Name	Access	Description	Reset value
[15:3]	Reserved	RO	Reserved.	0
[2:0]	PSCR	RW	IWDG clock frequency division 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/ frequency 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.*	0

Reload value register (IWDG_RLDR)

Offset address: 0x08

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

Bit	Name	Access	Description	Reset value
[15:12]	Reserved	RO	Reserved.	0
[11:0]	RLDR	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 and writing the value of this domain, make sure that the RUV bit in the IWDG_STATR register is 0. Otherwise, the domain read and write are invalid.*	FFFh

Note: This register will be reset in standby mode.

Status register (IWDG_STATR)

Offset address: 0x0C

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

Bit	Name	Access	Description	Reset value
[15:2]	Reserved	RO	Reserved.	0

1	RUV	RO	Reload value update flag bit, set or cleared by hardware. 1: The reload value update is in progress; 0: Reload update completed (5 LSI cycles at most). *Note: The reload value register IWDG_RLDR can only be read and written after the RVU bit is cleared to 0.*	0
0	PVU	RO	Clock frequency division factor update flag bit, set or cleared by hardware. 1: Clock frequency division value update in progress; 0: Clock frequency division value update completed (at most 5 LSI cycles). *Note: The frequency division factor register IWDG_PSCR can only be read and written after the PVU bit is cleared to 0.*	0

Note: After the prescale and/or reload value is updated, there is no need to wait for the RVU or PVU to reset. You can continue to execute the following code. (Even in the low-power mode, this write operation will be executed continuously.)

Chapter 8 Window Watchdog (WWDG)

The module description in this chapter applies to the full range of CH32F103 and CH32V103 microcontrollers.

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.

8.1 Main Features

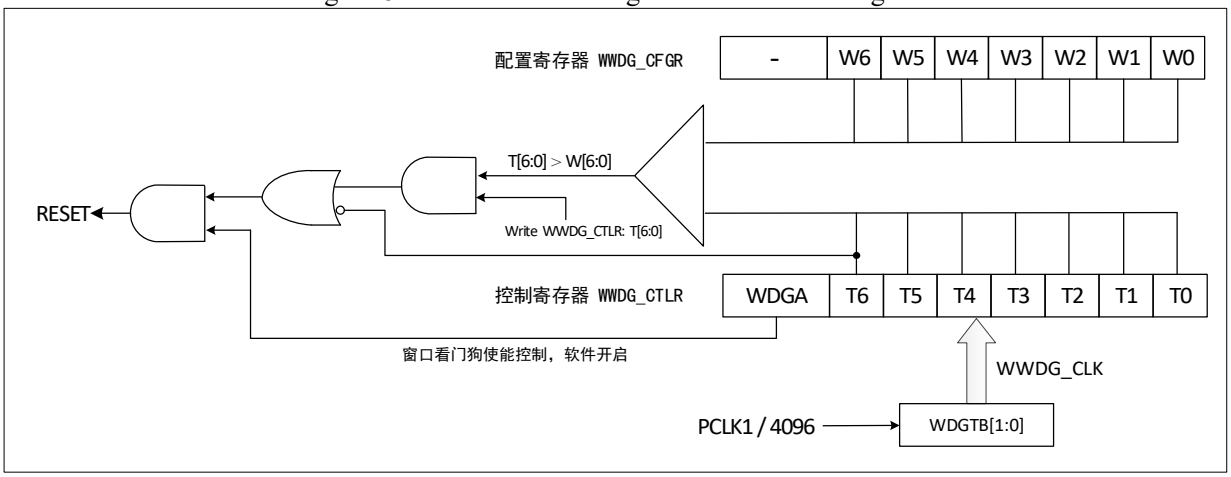
- Programmable 7-bit self-decrement type counter
- Double conditions reset: The current counter value is less than 0x40, or the counter value is reloaded beyond the window time
- Wake-up advance notification function (EWI), used to feed the dog in time to prevent system reset

8.2 Function Specification

8.2.1 Principle and Using Method

The window watchdog runs based on a 7-bit down counter, which is mounted under the APB1 bus. The counting time base WWDG_CLK originates from the frequency division of clock (PCLK1/4096). The frequency division is set in WDGTB[1:0] of configuration register WWDG_CFGR. The down counter 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 Structure Block Diagram



- Enabe window watchdog

After the system is reset, the watchdog will be at the off status. Set WDGA bit of WWDG_CTLR register to switch on the watchdog, and then it can no longer be turned off unless a reset occurs.

Note: You can switch off the clock source of WWDG 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

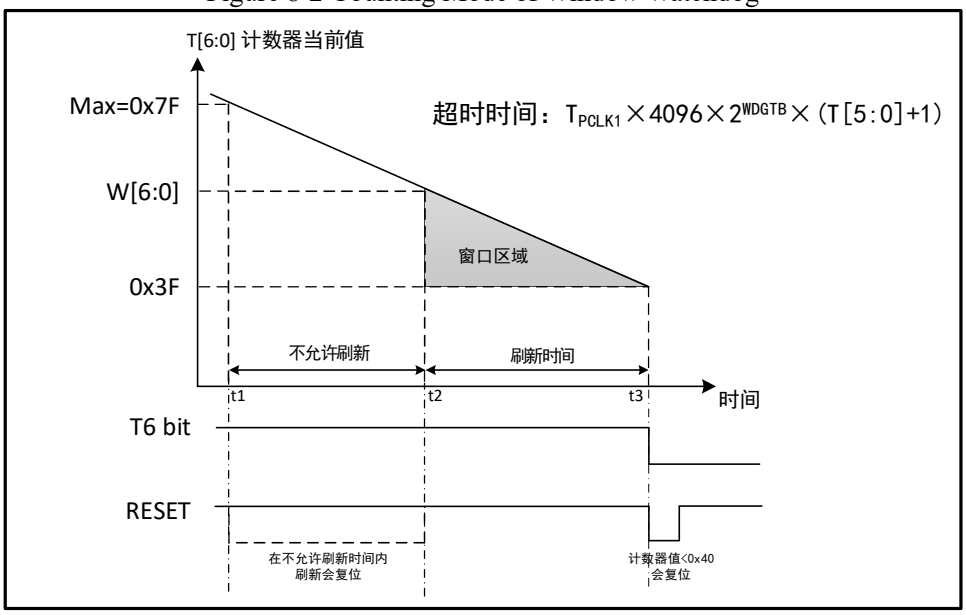
A 7-bit counter which continues to run by decreasing progressively in a cycle is located in the watchdog and supports read and write access. When the watchdog reset function is used, the following operations need to be executed:

1) Counting time base: Through the WDGTB[1:0] bit field of the WWDG_CFGR register, note to switch on the WWDG module clock of the RCC unit.
2) Window counter: Set the W[6:0] bit field of the WWDG_CFGR register. This counter is used by the hardware to compare with the current counter. 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 of 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] bit field of 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 will occur.

- Feed dog window time

As shown in Figure 8-2, the gray area is the monitoring window area of the window watchdog. Its upper limit time t2 corresponds to the time point when the current counter value reaches the window value W[6:0]; its lower limit time t3 corresponds to the current counter value reaching the time point of 0x3F. Within this area time (t2<t<t3), the dog feeding operation can be made (write T[6:0]) to refresh the current counter value.

Figure 8-2 Counting Mode of Window Watchdog



- Watchdog reset:

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

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

2) When the counter refresh action is executed during the period when the dog is not allowed to be fed, i.e., when the T[6:0] bit field is written within the time t1≤t≤t2, a "window watchdog reset" will appear, and a system reset will occur.

- In-advance wake-up

In order to prevent the system from resetting if the counter is not refreshed in time, the watchdog module provides early wake-up interrupt (EWI) notification. When the counter self-decreases to 0x40, an early wake-up signal will be generated and the WEIF flag will be set to 1. If the EWI bit is set, the window watchdog interrupt will be triggered at the same time. At this time, there is 1 counter clock cycle (self-decreasing to 0x3F) from the hardware reset, and the application can immediately feed the dog within the time limit.

8.2.2 Debug Mode

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

8.3 Register Description

Table 8-1 List of WWDG Related Registers

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

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 reset enable bit. 1: Enable the watchdog function (reset signal may be generated); 0: Disable the watchdog function. Write 1 to enable the software, but the hardware is allowed to clear 0 only after reset.	0
[6:0]	T	RW	7-bit self-decrement counter self-decreases by 1 every 4096*2^{WDGTB} cycles of PCLK1. When the counter self-decreases from 0x40 to 0x3F, i.e., a watchdog reset will be generated when T6 jumps to 0.	7Fh

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	Wake up interrupt enable bit in advance. If this bit is set to 1, an interrupt will be generated when the counter value reaches 0x40. This bit can be only cleared to 0 by hardware after reset.	0
[8:7]	WDGTB	RW	Window watchdog clock frequency division selection: 00: divided by 1, counter clock =PCLK1/4096; 01: divided by 2, counter clock =PCLK1/4096/2; 10: divided by 4, counter clock = PCLK1/4096/4; 11: divided by 8, counter clock= PCLK1/4096/8.	0
[6:0]	W	RW	Window watchdog 7-bit window value. It is used to compare with the counter value. The dog feeding operation can be made only when the counter value is less than the window value and is greater than 0x3F.	7Fh

WWDG configuration register (WWDG_STATR)

Offset address: 0x04

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

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

Chapter 9 Interrupt and Events

The module description in this chapter applies to the full range of CH32F103 and CH32V103 microcontrollers.

The Cortex-M3 core is adopted for CH32F103 series productst. The built-in Nested Vectored Interrupt Controller (NVIC) manages 44 shieldable external interrupt channels and 10 core interrupt channels, and other interrupt sources are reserved. The interrupt controller is closely connected with the core interface, providing a flexible interrupt management function with the minimum interrupt delay. For specific instructions for the use of the NVIC controller, please refer to the related documents of Cortex-M3.

The built-in programmable fast interrupt controller (PFIC) of CH32V103 series supports up to 255 interrupt vectors. The current system manages 44 peripheral interrupt channels and 5 core interrupt channels, and others are reserved.

9.1 Main Features

9.1.1 NVIC Controller

- 44 maskable interrupt channels
- Providing the prompt response of non-maskable interrupt
- In the vectorized interrupt design, the vector entry address directly enters the core
- Automatic stacking and recovery when the interrupt enters and exits, without additional command overhead
- 16-level nesting, priority-level dynamic modification

9.1.2 PFIC Controller

- 44+3 individually maskable interrupts; each interrupt request has an independent trigger and mask bits, status bit
- Providing a non-maskable interrupt NMI
- For 2-level nesting interrupt entry and exit and hardware automatic stacking and recovery; no instruction overhead is required
- 4-channel programmable fast interrupt channel; custom interrupt vector address

9.2 System Timer

- CH32F103

The Cortex-M3 core is provided with a 24-bit self-decrement counter (SysTick timer). It supports HCLK or HCLK/8 serving 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.

- CH32V103

The RISC-V3A core is provided with a 64-bit self-increment counter (SysTick) that supports HCLK/8 as the time base, has a higher priority and can be used for time base after calibration.

9.3 Interrupt and Exception Vector

Table 9-1 Vector Table of CH32F103

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	Failure of all types	0x0000000C
	0	Settable	MemManage	Memory management	0x00000010
	1	Settable	BusFault	Prefetch failure, memory access failure	0x00000014
	2	Settable	UsageFault	Undefined command or illegal status	0x00000018
	-	-	-	Reserved	0x0000001C ~0x0000002B
	3	Settable	SVCall	System service call through SWI command	0x0000002C
	4	Settable	Debug Monitor	Debugging monitor	0x00000030
	-	-	-	Reserved	0x00000034
	5	Settable	PendSV	Mountable system service	0x00000038
	6	Settable	SysTick	System tick timer	0x0000003C
0	7	Settable	WWDG	Window watchdog timer interrupt	0x00000040
1	8	Settable	PVD	Connected to power voltage detection (PVD) interrupt of EXTI	0x00000044
2	9	Settable	TAMPER	Intrusion detection interrupt	0x00000048
3	10	Settable	RTC	Global interrupt of real-time clock (RTC)	0x0000004C
4	11	Settable	FLASH	FLASH global interrupt	0x00000050
5	12	Settable	RCC	Reset and clock control (RCC)	0x00000054
6	13	Settable	EXTI0	EXTI line 0 interrupt	0x00000058
7	14	Settable	EXTI1	EXTI line 1 interrupt	0x0000005C
8	15	Settable	EXTI2	EXTI line 2 interrupt	0x00000060
9	16	Settable	EXTI3	EXTI line 3 interrupt	0x00000064
10	17	Settable	EXTI4	EXTI line 4 interrupt	0x00000068
11	18	Settable	DMA1 channel 1	DMA1 channel 1 global interrupt	0x0000006C
12	19	Settable	DMA1 channel 2	DMA1 channel 2 global interrupt	0x00000070
13	20	Settable	DMA1 channel 3	DMA1 channel 3 global interrupt	0x00000074
14	21	Settable	DMA1 channel 4	DMA1 channel 4 global interrupt	0x00000078
15	22	Settable	DMA1 channel 5	DMA1 channel 5 global interrupt	0x0000007C
16	23	Settable	DMA1 channel 6	DMA1 channel 6 global interrupt	0x00000080
17	24	Settable	DMA1 channel 7	DMA1 channel 7 global interrupt	0x00000084
18	25	Settable	ADC	Global interrupt of ADC1 and ADC2	0x00000088

19	26	Settable	USB_HP_CAN_TX	USBD high priority or CAN transmission interrupt	0x0000008C
20	27	Settable	USB_LP_CAN_RX0	USBD low priority or CAN reception 0 interrupt	0x00000090
21	28	Settable	CAN_RX1	CAN reception 1 interrupt	0x00000094
22	29	Settable	CAN_SCE	CAN SCE interrupt	0x00000098
23	30	Settable	EXTI9_5	EXTI[9:5] interrupt	0x0000009C
24	31	Settable	TIM1_BRK	TIM1 break interrupt	0x000000A0
25	32	Settable	TIM1_UP	TIM1 update interrupt	0x000000A4
26	33	Settable	TIM1_TRG_COM	TIM1 trigger and communication interrupt	0x000000A8
27	34	Settable	TIM1_CC	Capture/compare interrupt	0x000000AC
28	35	Settable	TIM2	TIM2 global interrupt	0x000000B0
29	36	Settable	TIM3	TIM3 global interrupt	0x000000B4
30	37	Settable	TIM4	TIM4 global interrupt	0x000000B8
31	38	Settable	I2C1_EV	I^2C1 event interrupt	0x000000BC
32	39	Settable	I2C1_ER	I^2C1 error interrupt	0x000000C0
33	40	Settable	I2C2_EV	I^2C2 event interrupt	0x000000C4
34	41	Settable	I2C2_ER	I^2C2 error interrupt	0x000000C8
35	42	Settable	SPI1	SPI1 global interrupt	0x000000CC
36	43	Settable	SPI2	SPI2 global interrupt	0x000000D0
37	44	Settable	USART1	USART1 global interrupt	0x000000D4
38	45	Settable	USART2	USART2 global interrupt	0x000000D8
39	46	Settable	USART3	USART3 global interrupt	0x000000DC
40	47	Settable	EXTI15_10	EXTI[15:10] interrupt	0x000000E0
41	48	Settable	RTCAlarm	Connected to RTC clock interrupt of EXTI	0x000000E4
42	49	Settable	USBWakeUp	Connected to USB wake-up interrupt of EXTI	0x000000E8
43	50	Settable	USBHD	USBHD transmission interrupt	0x000000EC

Table 9-2 Vector Table of CH32V103

No.	Priority	Type	Name	Description	Entry address
0		-	-		0x00000000
1	-3	Fixed	Reset	Reset	0x00000004
2	-2	Fixed	NMI	Non-maskable interrupt	0x00000008
3	-1	Fixed	EXC	Exception interrupt	0x0000000C
4-11	-	-	-	Reserved	
12	0	Programmable	SysTick	System timer interrupt	0x00000030
13	-	-	-	Reserved	
14	1	Programmable	SWI	Software interrupt	0x00000038
15	-	-	-	Reserved	
16	2	Programmable	WWDG	Window timer interrupt	0x00000040

17	3	Programmable	PVD	Power voltage detection interrupt (EXTI)	0x00000044
18	4	Programmable	TAMPER	Intrusion detection 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 line 0 interrupt	0x00000058
23	9	Programmable	EXTI1	EXTI line 1 interrupt	0x0000005C
24	10	Programmable	EXTI2	EXTI line 2 interrupt	0x00000060
25	11	Programmable	EXTI3	EXTI line 3 interrupt	0x00000064
26	12	Programmable	EXTI4	EXTI line 4 interrupt	0x00000068
27	13	Programmable	DMA1_CH1	DMA1 channel 1 global interrupt	0x0000006C
28	14	Programmable	DMA1_CH2	DMA1 channel 2 global interrupt	0x00000070
29	15	Programmable	DMA1_CH3	DMA1 channel 3 global interrupt	0x00000074
30	16	Programmable	DMA1_CH4	DMA1 channel 4 global interrupt	0x00000078
31	17	Programmable	DMA1_CH5	DMA1 channel 5 global interrupt	0x0000007C
32	18	Programmable	DMA1_CH6	DMA1 channel 6 global interrupt	0x00000080
33	19	Programmable	DMA1_CH7	DMA1 channel 7 global interrupt	0x00000084
34	20	Programmable	ADC	ADC global interrupt	0x00000088
35-38	-	-	-	Reserved	
39	21	Programmable	EXTI9_5	EXTI line [9:5] interrupt	0x0000009C
40	22	Programmable	TIM1_BRK	TIM1 break interrupt	0x000000A0
41	23	Programmable	TIM1_UP	TIM1 update interrupt	0x000000A4
42	24	Programmable	TIM1_TRG_COM	TIM1 trigger and communication interrupt	0x000000A8
43	25	Programmable	TIM1_CC	TIM1 capture comparison interrupt	0x000000AC
44	26	Programmable	TIM2	TIM2 global interrupt	0x000000B0
45	27	Programmable	TIM3	TIM3 global interrupt	0x000000B4
46	28	Programmable	TIM4	TIM4 global interrupt	0x000000B8
47	29	Programmable	I2C1_EV	I^2C1 event interrupt	0x000000BC
48	30	Programmable	I2C1_ER	I^2C1 error interrupt	0x000000C0
49	31	Programmable	I2C2_EV	I^2C2 event interrupt	0x000000C4
50	32	Programmable	I2C2_ER	I^2C2 error interrupt	0x000000C8
51	33	Programmable	SPI1	SPI1 global interrupt	0x000000CC
52	34	Programmable	SPI2	SPI2 global interrupt	0x000000D0
53	35	Programmable	USART1	USART1 global interrupt	0x000000D4
54	36	Programmable	USART2	USART2 global interrupt	0x000000D8
55	37	Programmable	USART3	USART3 global interrupt	0x000000DC
56	38	Programmable	EXTI15_10	EXTI line [15:10] interrupt	0x000000E0
57	39	Programmable	RTCAlarm	RTC alarm interrupt (EXTI)	0x000000E4
58	40	Programmable	USBWakeUp	USB wake-up interrupt (EXTI)	0x000000E8
59	41	Programmable	USBHD	USBHD transmission interrupt	0x000000EC

9.4 External Interrupt and Event Controller (EXTI)

9.4.1 Overview

Figure 9-1 External Interrupt (EXTI) Interface Block Diagram

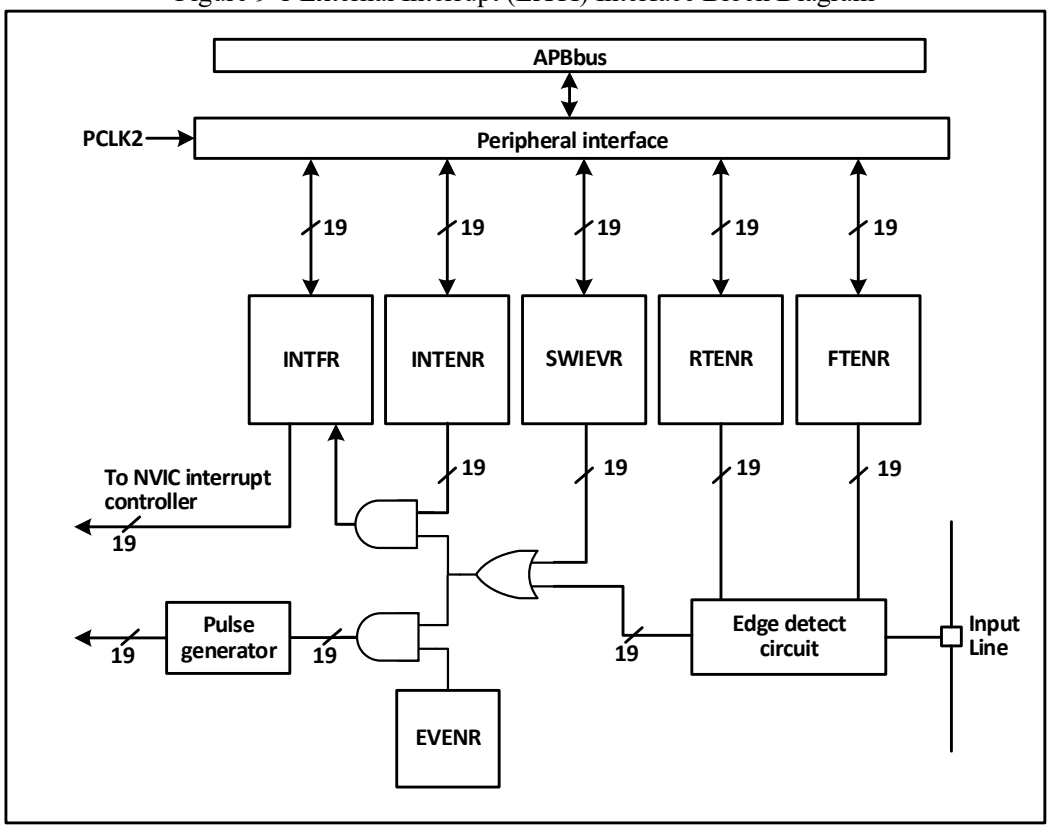


Figure 9-1 shows that the trigger source of an external interrupt can be either a software interrupt (SWIEVR) or an actual external interrupt channel. The signal of the external interrupt channel will firstly be filtered by the edge detect circuit. As long as either soft interrupt or external interrupt signal is generated, it will be outputted to two 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 interruption or event will be generated. Six registers of EXTI are accessed by the processer through APB2 interface.

9.4.2 Wake-up Event Description

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

- Enable an interrupt in the peripheral register, but do not enable the interrupt in the NVIC of the core, and enable the SEVONPEND bit in the core at the same time. Reflected in EXTI, it is to enable EXTI interrupt, but not to enable EXTI interrupt in NVIC, while enabling SEVONPEND bit. When the CPU is woken up from WFE, the interrupt flag bit of EXTI and suspension bit of NVIC need to be cleared.
- Enable an EXTI channel as an event channel. It is not necessary to clear the interrupt flag bit and the NVIC suspension bit operation after the CPU is woken up from WFE.

9.4.3 Description

To use the external interrupt, you need to configure the external interrupt channel, i.e., selecting the trigger edge and enabling the interrupt. When the set trigger edge appears on the external interrupt channel, an interrupt request will be generated and the corresponding interrupt flag bit will also be set. Write 1 in the flag bit to clear such flag bit.

Steps for interrupt with external hardware:

1) Configure GPIO;
2) Configure the interrupt enable bit (EXTI_INTENR) of 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 of the core to ensure that it can respond correctly.

Steps for enabling event with external hardware:

1) Configure GPIO;
2) Configure the event enable bit (EXTI_EVENR) of 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 for enabling interrupt/event with software:

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

9.4.4 External Event Mapping

Table 9-3 EXTI Interrupt Mapping

External interrupt/event route	Mapping event description
EXTI0～EXTI15	Px0～Px15 (x=A/B/C/D); any IO port can enable the external interrupt/event function, configured by AFIO_EXTICRx register.
EXTI16	PVD event: Exceed the voltage monitoring threshold
EXTI17	RTC alarm event
EXTI18	USB wake-up event

9.5 Register Description

9.5.1 EXTI Register Description

Table 9-4 List of EXTI Related Registers

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

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													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
[31:19]	Reserved	RO	Reserved.	0
[18:0]	MRx	RW	Interrupt request signal of enable external interrupt channel x: 1: Enable interrupt of such channel; 0: Disable interrupt of such channel.	0

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													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
[31:19]	Reserved	RO	Reserved.	0
[18:0]	MRx	RW	Event request signal of enable external interrupt channel x: 1: Enable event of such channel; 0: Disable event of such channel.	0

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													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
[31:19]	Reserved	RO	Reserved.	0
[18:0]	TRx	RW	Rising edge trigger of enable external interrupt channel x: 1: Enable rising edge trigger of such channel; 0: Disable rising edge trigger of such channel;	0

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													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
[31:19]	Reserved	RO	Reserved	0
[18:0]	TRx	RW	Falling edge trigger of enable external interrupt channel x: 0: Disable the faling edge trigger of such channel; 1: Enable the faling edge trigger of such channel;	0

Soft 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 18	SWIER 17	SWIER 16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWIER15	SWIER 14	SWIER13	SWIER12	SWIER11	SWIER 10	SWIER 9	SWIER 8	SWIER 7	SWIER 6	SWIER 5	SWIER 4	SWIER 3	SWIER 2	SWIER 1	SWIER 0

Bit	Name	Access	Description	Reset value
[31:19]	Reserved	RO	Reserved.	0
[18:0]	SWIERx	RW	Set a software interrupt on the corresponding external trigger interrupt channel. Set this bit here to set the corresponding bit of the interrupt flag bit (EXTI_INTFR). If the interrupt enable (EXTI_INTENR) or event enable(EXTI_EVENR) is switched on, then an interrupt or event will be generated.	0

Interrupt flag bit register (EXTI_INTFR)

Offset address: 0x14

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													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
[31:19]	Reserved	RO	Reserved.	0
[18:0]	IFx	W1	Interrupt flag bit, indicating that the corresponding external interrupt has occurred. Write 1 to clear it.	X

9.5.2 PFIC Register Description

Table 9-5 List of PFIC Related Registers

Name	Access address	Description	Reset value
R32_PFIC_ISR1	0xE000E000	PFIC interrupt enable status register 1	0x0000000C
R32_PFIC_ISR2	0xE000E004	PFIC interrupt enable status register 2	0x00000000
R32_PFIC_IPR1	0xE000E020	PFIC interrupt suspension status register 1	0x00000000
R32_PFIC_IPR2	0xE000E024	PFIC interrupt suspension status register 2	0x00000000
R32_PFIC_ITHRESDR	0xE000E040	PFIC interrupt priority threshold configuration register	0x00000000
R32_PFIC_FIBADDRR	0xE000E044	PFIC fast interrupt service base address register	0x80000000
R32_PFIC_CFGR	0xE000E048	PFIC interrupt configuration register	0x00000000
R32_PFIC_GISR	0xE000E04C	PFIC interrupt global status register	0x00000000
R32_PFIC_FIOFADDRR0	0xE000E060	PFIC fast interrupt 0 offset address register	0x00000000
R32_PFIC_FIOFADDRR1	0xE000E064	PFIC fast interrupt 1 offset address register	0x00000000
R32_PFIC_FIOFADDRR2	0xE000E068	PFIC fast interrupt 2 offset address register	0x00000000
R32_PFIC_FIOFADDRR3	0xE000E06C	PFIC fast interrupt 3 offset address register	0x00000000
R32_PFIC_IENR1	0xE000E100	PFIC interrupt enable setting register 1	0x00000000
R32_PFIC_IENR2	0xE000E104	PFIC interrupt enable setting register 2	0x00000000
R32_PFIC_IRER1	0xE000E180	PFIC interrupt enable clearing register 1	0x00000000
R32_PFIC_IRER2	0xE000E184	PFIC interrupt enable clearing register 2	0x00000000
R32_PFIC_IPSR1	0xE000E200	PFIC interrupt suspension setting register 1	0x00000000
R32_PFIC_IPSR2	0xE000E204	PFIC interrupt suspension setting register 2	0x00000000
R32_PFIC_IPRR1	0xE000E280	PFIC interrupt suspension clearing register 1	0x00000000
R32_PFIC_IPRR2	0xE000E284	PFIC interrupt suspension clearing register 2	0x00000000
R32_PFIC_IACTR1	0xE000E300	PFIC interrupt activation status register 1	0x00000000
R32_PFIC_IACTR2	0xE000E304	PFIC interrupt activation status register 2	0x00000000
R32_PFIC_IPRIORx	0xE000E400	PFIC interrupt priority configuration register	0x00000000
R32_PFIC_SCTLR	0xE000ED10	PFIC system control register	0x00000000

PFIC interrupt enable status register 1 (PFIC_ISR1)

Offset address: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INTS[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	INTS14	-	INTS12	Reserved								1	1	Reserved	

Bit	Name	Access	Description	Reset value
[31:12]	INTSTA	RO	31# and below interrupt current enable status. 1: Enable the current number interrupt; 0: Disable the current number interrupt.	Ch

PFIC interrupt enable status register 2 (PFIC_ISR2)

Offset address: 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				INTS[59:48]											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTS[47:32]															

Bit	Name	Access	Description	Reset value
[31:28]	Reserved	RO	Reserved.	0
[27:0]	INTENSTA	RO	32# and above interrupt current enable status. 1: Enable the current number interrupt; 0: Disable the current number interrupt.	0

PFIC interrupt suspension status register 1 (PFIC_IPR1)

Offset address: 0x20

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IPS[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	IPS14	-	IPS12	Reserved								IPS3	IPS2	Reserved	

Bit	Name	Access	Description	Reset value
[31:12]	PENDSTA	RO	31# and below interrupt current suspension status. 1: Current number intrrupt has been suspended; 0: Current number intrrupt is not suspended;	0

PFIC interrupt suspension status register 2 (PFIC_IPR2)

Offset address: 0x24

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				IPS[59:48]											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPS[47:32]															

Bit	Name	Access	Description	Reset value

[31:28]	Reserved	RO	Reserved.	0
[27:0]	PENDSTA	RO	32# and above interrupt current suspension status. 1: Current number intrrupt has been suspended; 0: Current number intrrupt is not suspended;	0

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
THRESHOLD[31:0]																															

Bit	Name	Access	Description	Reset value
[31:8]	Reserved	RO	Reserved.	0
[7:0]	THRESHOLD	RW	Interrupt priority threshold setting value. If the priority value of the interrupt is lower than the current setting value, interrupt service will not be performed when suspended; when this register is 0, it means that the threshold register function is invalid. [7:4]: Priority threshold. [3:0]: Reserved; 0 constantly; invalid if writing.	0

PFIC fast interrupt service base address register (PFIC_FIBADDRR)

Offset address: 0x44

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BASEADDR[3:0]				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]	BASEADDR	RW	The higher 4 bits of target jump address of fast interrupt response. Together with PFIC_FIOFADDRR*, it forms the corresponding-number fast interrupt vector (the 32-bit jump address of the interrupt service program).	8h
[27:0]	Reserved	RO	Reserved.	0

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								SYS RESET	PFIC RESET	EXC RESET	EXC SET	NMI RESET	NMI SET	NEST CTRL	HWSTK CTRL

Bit	Name	Access	Description	Reset value
[31:16]	KEYCODE	WO	For corresponding target control bits, the corresponding security access identification data needs to be written synchronously for modification, and the read data is fixed to 0. KEY1 = 0xFA05; KEY2 = 0xBCAF; KEY3 = 0xBEEF.	0
[15:8]	Reserved	RO	Reserved.	0
7	SYSRESET	WO	System reset (Write into KEY3 synchronously). Cleared to 0 automatically. Valid when writing 1; invalid when writing 0.	0
6	PFICRESET	WO	PFIC control module reset; clear 0 automatically. Valid when writing 1; invalid when writing 0.	0
5	EXCRESET	WO	Exception interrupt clearing suspension (write into KEY2 synchronously). Valid when writing 1; invalid when writing 0.	0
4	EXCSET	WO	Exception interrupt suspension (write into KEY2 synchronously). Valid when writing 1; invalid when writing 0.	0
3	NMIRESET	WO	NMI interrupt clearing suspension (write into KEY2 synchronously). Valid when writing 1; invalid when writing 0.	0
2	NMISET	WO	NMI interrupt suspension (write into KEY2 synchronously). Valid when writing 1; invalid when writing 0.	0
1	NESTCTRL	RW	Nesting interrupt enable control: 1: Off; 0: On (write into KEY1 synchronously).	0
0	HWSTKCTRL	RW	Hardware stack enable control: 1: Off; 0: On (write into KEY1 synchronously).	0

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 interrupt suspended 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 nesting status, supporting 2-level nesting currently, [1:0] valid. 3: Level 2 interrupt in process; 1: Level 1 interrupt in process; 0: No interrupt occurs; Others: Impossible circumstances.	0

PFIC fast interrupt 0 offset address register (PFIC_FIOFADDRR0)

Offset address: 0x60

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IRQID0[7:0]								OFFADDR0[23:16]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFADDR0[15:0]															

Bit	Name	Access	Description	Reset value
[31:24]	IRQID0	RW	Number of fast interrupt 0.	0
[23:0]	OFFADDR0	RW	Lower 24-bit address of the fast interrupt 0 service program, of which the low 20-bit configuration is valid, and [23:20] is fixed to 0.	0

PFIC fast interrupt 1 offset address register (PFIC_FIOFADDRR1)

Offset address: 0x64

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IRQID1[7:0]								OFFADDR1[23:16]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFADDR1[15:0]															

Bit	Name	Access	Description	Reset value
[31:24]	IRQID1	RW	Number of fast interrupt 1.	0
[23:0]	OFFADDR1	RW	Lower 24-bit address of the fast interrupt 1 service program, of which the low 20-bit configuration is valid, and [23:20] is fixed to 0.	0

PFIC fast interrupt 2 offset address register (PFIC_FIOFADDRR2)

Offset address: 0x68

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IRQID2[7:0]								OFFADDR2[23:16]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFADDR2[15:0]															

Bit	Name	Access	Description	Reset value
[31:24]	IRQID2	RW	Number of fast interrupt 2.	0
[23:0]	OFFADDR2	RW	Lower 24-bit address of the fast interrupt 2 service program, of which the low 20-bit configuration is valid, and [23:20] is fixed to 0.	0

PFIC fast interrupt 3 offset address register (PFIC_FIOFADDRR0)

Offset address: 0x6C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IRQID3[7:0]								OFFADDR3[23:16]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFADDR3[15:0]															

Bit	Name	Access	Description	Reset value
[31:24]	IRQID3	RW	Number of fast interrupt 3.	0
[23:0]	OFFADDR3	RW	Lower 24-bit address of the fast interrupt 3 service program, of which the low 20-bit configuration is valid, and [23:20] is fixed to 0.	0

PFIC interrupt enable setting 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
-	INTEN14	-	INTEN12	Reserved											

Bit	Name	Access	Description	Reset value
[31:12]	INTEN	WO	31# and below interrupt enable control. 1: Enable the current number interrupt; 0: No effect.	0
[11:0]	Reserved	RO	Reserved.	0

PFIC interrupt enable setting register 2 (PFIC_IENR2)

Offset address: 0x104

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				INTEN[59:48]											
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
[31:28]	Reserved	RO	Reserved.	0
[27:0]	INTEN	WO	32# and above interrupt enable control. 1: Enable the current number interrupt; 0: No effect.	0

PFIC interrupt enable clearing 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
-	INT RESET14	-	INT RESET12	Reserved											

Bit	Name	Access	Description	Reset value
[31:12]	INTRESET	WO	31# and below interrupt disable control. 1: Disable the current number interrupt; 0: No effect.	0
[11:0]	Reserved	RO	Reserved.	0

PFIC interrupt enable clearing register 2 (PFIC_IRER2)

Offset address: 0x184

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				INTRESET[59:48]											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTRESET[47:32]															

Bit	Name	Access	Description	Reset value
[31:28]	Reserved	RO	Reserved.	0
[27:0]	INTRESET	WO	32# and above interrupt disable control. 1: Disable the current number interrupt; 0: No effect.	0

PFIC interrupt suspension setting 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
-	PENDSET14	-	PENDSET12	Reserved								PEND SET3	PEND SET2	Reserved	

Bit	Name	Access	Description	Reset value
[31:12]	PENDSET	WO	31# and below interrupt suspension setting. 1: Current number intrrupt is suspended; 0: No effect.	0
[11:0]	Reserved	RO	Reserved.	0

PFIC interrupt suspension setting register 1 (PFIC_IPSR2)

Offset address: 0x204

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				PENDSET[59:48]											
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
[31:28]	Reserved	RO	Reserved.	0
[27:0]	PENDSET	WO	32# and above interrupt suspension setting. 1: Current number intrrupt is suspended; 0: No effect.	0

PFIC interrupt suspension clearing register 1 (PFIC_IPRR1)

Offset address: 0x280

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PENDRESET[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	PENDRESET14	-	PENDRESET12	Reserved								PENDRESET3	PENDRESET2	Reserved	

Bit	Name	Access	Description	Reset value
[31:12]	PENDRESET	WO	31# and below interrupt suspension clearing. 1: Current number intrrupt clearing suspension status; 0: No effect.	0
[11:0]	Reserved	RO	Reserved.	0

Note: The above registers are invalid for the interrupts with number of Reset, NMI and EXC.

PFIC interrupt suspension clearing register 2 (PFIC_IPRR2)

Offset address: 0x284

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				PENDRESET[59:48]											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PENDRESET[47:32]															

Bit	Name	Access	Description	Reset value
[31:28]	Reserved	RO	Reserved.	0
[27:0]	PENDRESET	WO	32# and above interrupt suspension clearing. 1: Current number intrrupt clearing suspension status; 0: No effect.	0

PFIC interrupt activation status 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
-	IACTS14	-	IACTS12	Reserved											

Bit	Name	Access	Description	Reset value
[31:12]	IACTS	RO	31# and below interrupt execution status. 1: Executing the current number interrupt; 0: Not executing the current number interrupt.	0
[11:0]	Reserved	RO	Reserved.	0

PFIC interrupt activation status register 2 (PFIC_IACTR2)

Offset address: 0x304

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				IACTS[59:48]											
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
[31:28]	Reserved	RO	Reserved.	0
[27:0]	IACTS	RO	32# and above interrupt execution status. 1: Executing the current number interrupt; 0: Not executing the current number interrupt.	0

PFIC interrupt priority configuration register (PFIC_IPRIORx) (x=0-63)

Offset address: 0x400–0x4FF

The controller supports 256 interrupts (0-255), and 8bits are used to set the control priority for each interrupt.

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

Bit	Name	Access	Description	Reset value
[2047:2040]	IP_255	RW	Same as IP_0 description.	0
…	…	…	…	…
[31:24]	IP_3	RW	Same as IP_0 description.	0
[23:16]	IP_2	RW	Same as IP_0 description.	0
[15:8]	IP_1	RW	Same as IP_0 description.	0
[7:0]	IP_0	RW	Number 0 interrupt priority configuration: [7:4]: Priority control bit. [3:0]: Reserved; 0 constantly; invalid if writing. The smaller priority value means higher priority. There is 2-level interrupt nesting, i.e., it can be only seized once.	0

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
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										SET EVENT	SEV ONPEND	WFITO WFE	SLEEP DEEP	SLEEP ONEXIT	Reserved

Bit	Name	Access	Description	Reset value
[31:6]	Reserved	RO	Reserved.	0
5	SETEVENT	WO	Set an event to wake up the WFE.	0
4	SEVONPEND	RW	When an event or interrupt suspension status occurs, the system can be woken up by the WFE command. If the WFE command is not executed, the system will be woken up immediately after the next execution of the command. 1: Enabled events and all interrupts (including non-enabled interrupts) can wake up the system; 0: Only enabled events and enabled interrupts can wake up the system.	0
3	WFITOWFE	RW	The WFI command is executed as WFE. 1: The subsequent WFI command is deemed as WFE command; 0: No effect.	0
2	SLEEPDEEP	RW	Low power mode of control system: 1: deepsleep 0: sleep	0
1	SLEEPONEXIT	RW	The system status after the control leaves the interrupt service program: 1: The system enters low power mode; 0: The system enters the main program.	0
0	Reserved	RO	Reserved.	0

9.5.3 STK Register Description

Table 9-6 List of STK Related Registers

Name	Access address	Description	Reset value
R32_STK_CTLR	0xE000F000	System count control register	0x00000000
R32_STK_CNTL	0xE000F004	System counter low-bit register	0x00000000
R32_STK_CNTH	0xE000F008	System counter high-bit register	0x00000000
R32_STK_CMPLR	0xE000F00C	Count comparison low-bit register	0x00000000
R32_STK_CMPHR	0xE000F010	Count comparison high-bit register	0x00000000

System count control register (STK_CTLR)

Offset address: 0x00

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

Bit	Name	Access	Description	Reset value
[31:1]	Reserved	RO	Reserved.	0
0	STE	RW	System counter enable control bit: 1: Enable system counter STK (HCLK/8 time base); 0: Disable system counter STK; the counter stops counting.	0

System counter low-bit register (STK_CNTL)

Offset address: 0x04

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

Bit	Name	Access	Description	Reset value
[31:0]	CNTL	RW	Lower 32 bits of count value of current counter. Count increment. This register can be read in 8-bit/16-bit/32-bit mode, but can only be modified in 8-bit mode.	0

Note: Register STK_CNTL and register STK_CNTH together form a 64-bit increment system counter.

System counter high-bit register (STK_CNTH)

Offset address: 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT[63:56]								CNT[55:48]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[47:40]								CNT[39:32]							

Bit	Name	Access	Description	Reset value
[31:0]	CNTH	RW	Higher 32 bits of count value of current counter. Count increment. This register can be read in 8-bit/16-bit/32-bit mode, but can only be modified in 8-bit mode.	0

Note: Register STK_CNTL and register STK_CNTH together form a 64-bit increment system counter.

Count comparison low-bit register (STK_CMPLR)

Offset address: 0x0C

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

Bit	Name	Access	Description	Reset value
[31:0]	CMPL	RW	Set the comparison counter value lower 32 bits. When the CNT[63:0] and CMP[63:0] values are equal, the STK interrupt service will be triggered. This register can be read in 8-bit/16-bit/32-bit mode, but can only be modified in 8-bit mode.	0

Note: Register STK_CMPLR and register STK_CMPHR together compose the 64-bit counter comparison value.

Count comparison high-bit register (STK_CMPHR)

Offset address: 0x10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CMP[63:56]								CMP[55:48]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP[47:40]								CMP[39:32]							

Bit	Name	Access	Description	Reset value
[31:0]	CMPH	RW	Set the comparison counter value high 32 bits. When the CNT[63:0] and CMP[63:0] values are equal, the STK interrupt service will be triggered. This register can be read in 8-bit/16-bit/32-bit mode, but can only be modified in 8-bit mode.	0

Note: Register STK_CMPLR and register STK_CMPHR together compose the 64-bit counter comparison value.

Chapter 10 GPIO and Its Multiplexing Function

The module description in this chapter applies to the full range of CH32F103 and CH32V103 microcontrollers.

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

10.1 Main Features

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

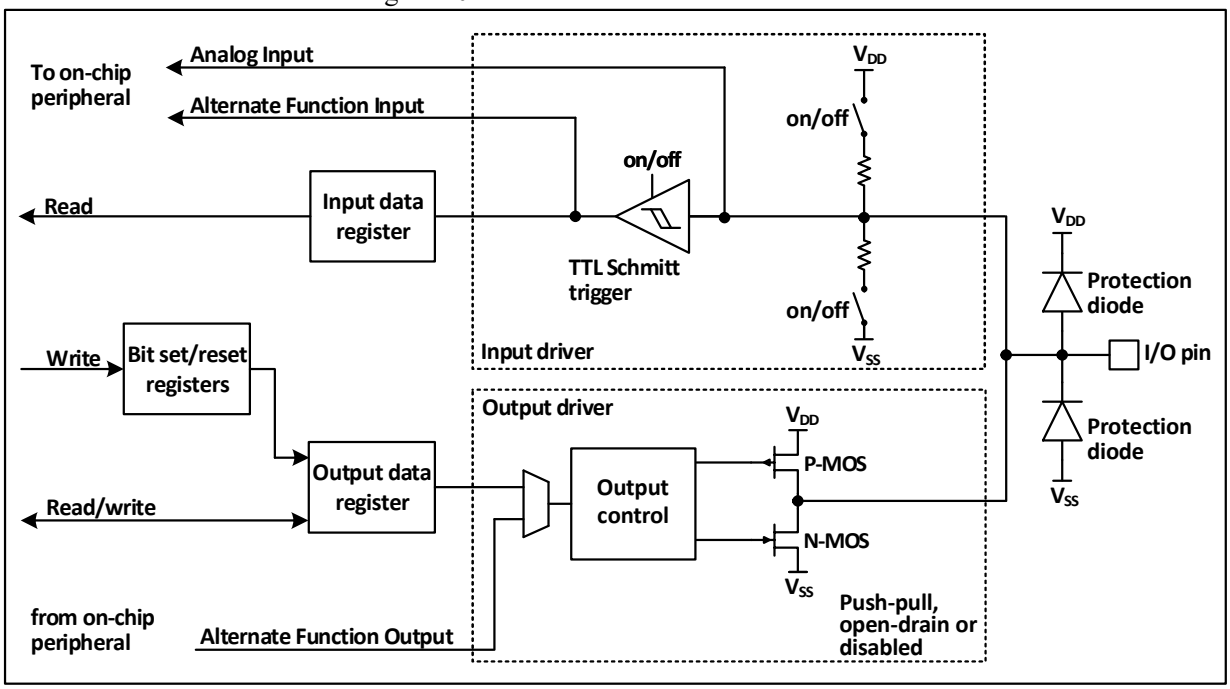
- Floating input
- Pull-up input
- Pull-down input
- Analog input
- Open-drain output
- Push-pull output
- Input and output of multiplexing function

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

10.2.2 Initialization Function of GPIO

Immediately after reset, the GPIO port is 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 JTAG and HSE that are running on peripheral multiplexing functions. Please refer to related chapters of pins for the specific initialization function. In which:

PA15 is in the pull-up mode as JTDI pin by default;
PA14 is in the pull-down mode as JTCK pin by default;
PA13 is in the pull-up mode as JTMS pin by default;
PB4 is in the pull-up mode as the JNTRS pin by default.

10.2.3 External Interrupt

All GPIO ports can be configured with external interrupt input channels, but one external interrupt input channel can only be mapped to one 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, or PE1. Both ports have one-to-one relationship.

10.2.4 Multiplexing Function

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

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

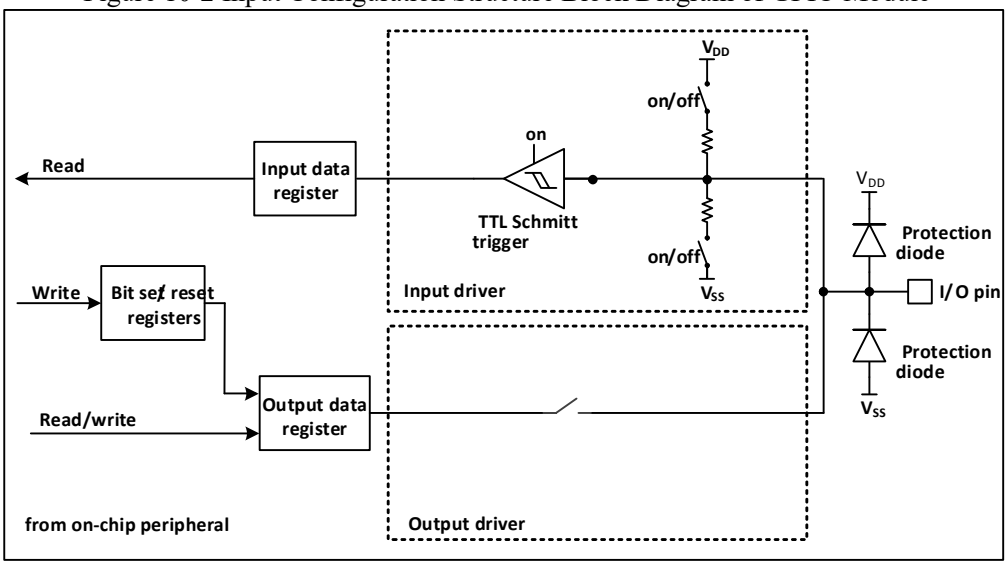
The same IO port may have multiple peripherals multiplexed to this pin, so in order to give play to the use of each peripheral as far as possible, the multiplexed pins of the peripherals can be remapped to other pins in addition to the default multiplexed pins, avoiding occupied pins.

10.2.5 Lock Mechanism

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

10.2.6 Input Configuration

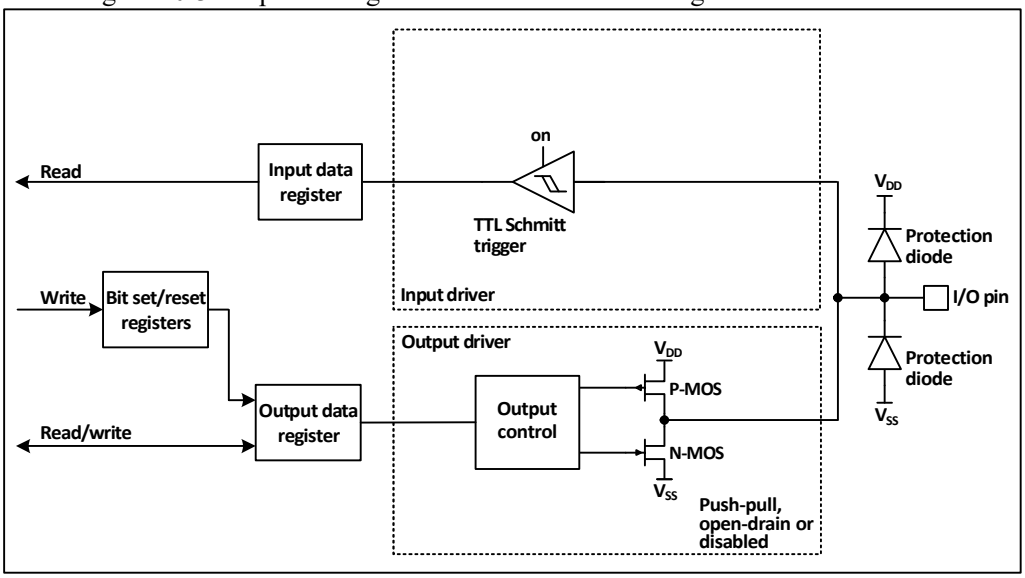
Figure 10-2 Input Configuration Structure Block Diagram of GPIO Module



When the IO port is configured as input mode, the output drive will be disconnected, the input pull-up and pull-down are optional, and the multiplex 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

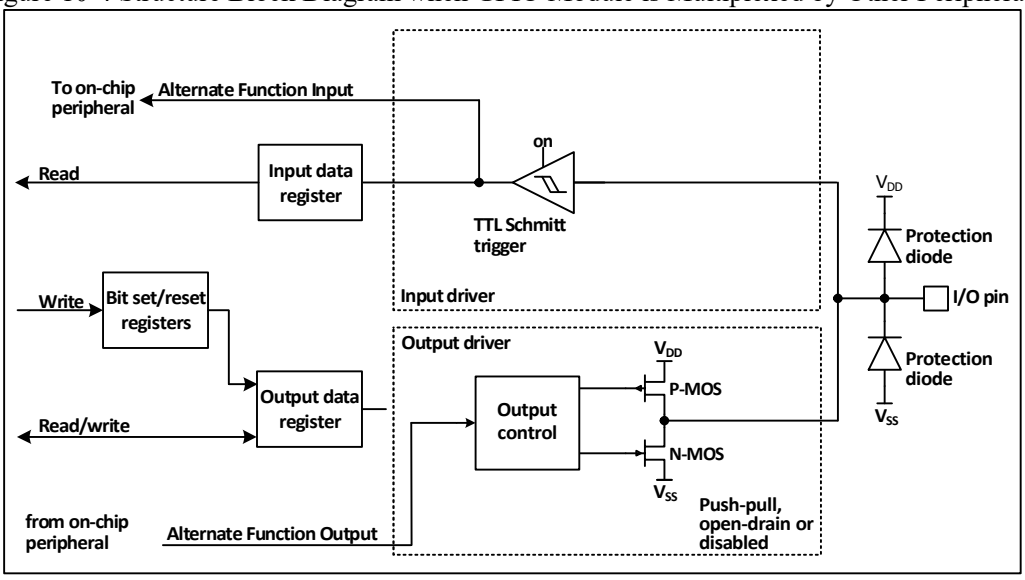
Figure 10-3 Output Configuration Structure Block Diagram of GPIO Module



When the IO port is configured in output mode, a pair of MOS in the output driver can be configured in push-pull or open-drain mode as required, without using multiplexing 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 Multiplexing Function Configuration

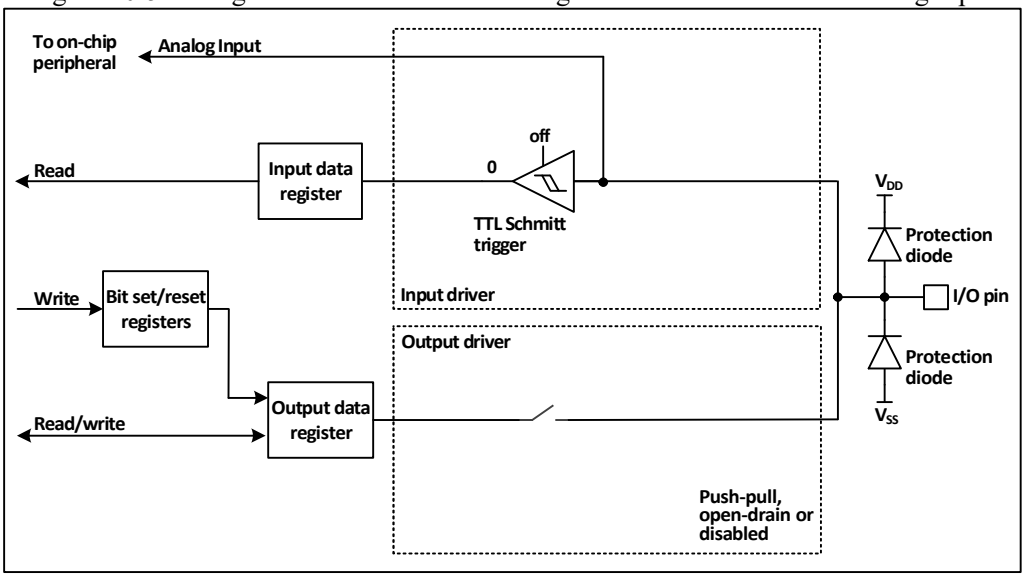
Figure 10-4 Structure Block Diagram when GPIO Module is Multiplexed by Other Peripherals



When the multiplexing function is enabled, the output driver will be 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 multiplexing function will be connected, but the output data register will be disconnected, and the level appearing on the IO pin will be sampled to the input data register every APB2 clock. In the open-drain mode, the current status of the IO port will be obtained by reading the input data register. In the push-pull mode, the last written value will be 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 will be disconnected and the Schmitt trigger input in the input driver will be disabled to prevent consumption on the IO port. The pull-up and pull-down resistors will be disabled, and the read input data register will always be 0.

10.2.10 GPIO Setting of Peripheral

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

Table 10-1 Advanced-control Timer (TIM1)

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

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

TIM2/3/4 pins	Configuration	GPIO configuration
TIM2/3/4_CHx	Input capture channel x	Floating input
	Output comparison channel x	Push-pull multiplexing output
TIM2/3/4_ETR	External trigger clock input	Floating input

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

USART pin	Configuration	GPIO configuration
USARTx_TX	Full duplex mode	Push-pull multiplexing output
	Half-duplex synchronous mode	Push-pull multiplexing output
USARTx_RX	Full duplex mode	Floating input or pull-up input
	Half-duplex synchronous mode	Not used
USARTx_CK	Synchronous mode	Push-pull multiplexing output
USARTx_RTS	Hardware flow control	Push-pull multiplexing output
USARTx_CTS	Hardware flow control	Floating input or pull-up input

Table 10-4 Serial Peripheral Interface (SPI) Module

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

Table 10-5 Inter Integrated Circuit (I2C) Module

I^2C pin	Configuration	GPIO configuration
I^2C_SCL	I^2C clock	Open-drain multiplexing output
I^2C_SDA	I^2C data	Open-drain multiplexing output

Table 10-6 Controller LAN (CAN) Module

CAN pin	GPIO configuration
CAN_TX	Push-pull multiplexing output
CAN_RX	Floating input or with pull-up input

Table 10-7 Universal Serial Bus (USB) Controller

USB pin	GPIO configuration
USB_DM/USB_DP	After the USB module is enabled, the multiplexed IO port will be automatically connected to the internal USB transceiver

Table 10-8 Analog-to-Digital Converter (ADC) and Digital-to-Analog Converter (DAC)

ADC/DAC pin	GPIO configuration
ADC/DAC	Analog input

Table 10-9 Other IO function settings

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

10.3 Register Description

10.3.1 Register Description of GPIO

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

Table 10-10 List of GPIO Related Registers

Name	**Access address**	**Description**	**Reset value**
R32_GPIOA_CFGLR	0x40010800	PA port configuration register low bit	0x44444444
R32_GPIOB_CFGLR	0x40010C00	PB port configuration register low bit	0x44444444
R32_GPIOC_CFGLR	0x40011000	PC port configuration register low bit	0x44444444
R32_GPIOD_CFGLR	0x40011400	PD port configuration register low bit	0x44444444
R32_GPIOA_CFGHR	0x40010804	PA port configuration register high bit	0x44444444
R32_GPIOB_CFGHR	0x40010C04	PB port configuration register high bit	0x44444444
R32_GPIOC_CFGHR	0x40011004	PC port configuration register high bit	0x44444444
R32_GPIOD_CFGHR	0x40011404	PD port configuration register high bit	0x44444444
R32_GPIOA_INDR	0x40010808	PA port input data register	0x0000XXXX
R32_GPIOB_INDR	0x40010C08	PB port input data register	0x0000XXXX
R32_GPIOC_INDR	0x40011008	PC port input data register	0x0000XXXX
R32_GPIOD_INDR	0x40011408	PD port input data register	0x0000XXXX

R32_GPIOA_OUTDR	0x4001080C	PA port output data register	0x00000000
R32_GPIOB_OUTDR	0x40010C0C	PB port output data register	0x00000000
R32_GPIOC_OUTDR	0x4001100C	PC port output data register	0x00000000
R32_GPIOD_OUTDR	0x4001140C	PD port output data register	0x00000000
R32_GPIOA_BSHR	0x40010810	PA port set/ reset register	0x00000000
R32_GPIOB_BSHR	0x40010C10	PB port set/reset register	0x00000000
R32_GPIOC_BSHR	0x40011010	PC port set/ reset register	0x00000000
R32_GPIOD_BSHR	0x40011410	PD port set/ reset register	0x00000000
R32_GPIOA_BCR	0x40010814	PA port reset register	0x00000000
R32_GPIOB_BCR	0x40010C14	PB port reset register	0x00000000
R32_GPIOC_BCR	0x40011014	PC port reset register	0x00000000
R32_GPIOD_BCR	0x40011414	PD port reset register	0x00000000
R32_GPIOA_LCKR	0x40010818	PA port lock configuration register	0x00000000
R32_GPIOB_LCKR	0x40010C18	PB port lock configuration register	0x00000000
R32_GPIOC_LCKR	0x40011018	PC port lock configuration register	0x00000000
R32_GPIOD_LCKR	0x40011418	PD port lock configuration register	0x00000000

GPIO configuration register low bit (GPIOx_CFGLR) (x=A/B/C/D)

Offset address: 0x00

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

Bit	Name	Access	Description	Reset value
[31:30] [27:26] [23:22] [19:18] [15:14] [11:10] [7:6] [3:2]	CNFy	RW	(y=0-7), the configuration bit of port x; configure the corresponding port through these bits. During the input mode (MODE=00b): 00: Analog input mode; 01: Floating input mode; 10: Mode with pull-up and pull-down 11: Reserved. During the output mode (MODE>00b): 00: General push-pull output mode; 01: General open-drain output mode; 10: Multiplexing function push-pull output mode; 11: Multiplexing function open-drain output mode.	01b
[29:28] [25:24] [21:20] [17:16] [13:12] [9:8]	MODEy	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

[5:4] [1:0]				

GPIO configuration register high bit (GPIOx_CFGHR) (x=A/B/C/D)

Offset address: 0x04

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

Bit	Name	Access	Description	Reset value
[31:30] [27:26] [23:22] [19:18] [15:14] [11:10] [7:6] [3:2]	CNFy	RW	(y=8-15), the configuration bit of port x; configure the corresponding port through these bits. During the input mode (MODE=00b): 00: Analog input mode; 01: Floating input mode; 10: Mode with pull-up and pull-down 11: Reserved. During the output mode (MODE>00b): 00: General push-pull output mode; 01: General open-drain output mode; 10: Multiplexing function push-pull output mode; 11: Multiplexing function open-drain output mode.	01b
[29:28] [25:24] [21:20] [17:16] [13:12] [9:8] [5:4] [1:0]	MODEy	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

Port input register (GPIOx_INDR) (x=A/B/C/D)

Offset address: 0x08

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

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved.	0
[15:0]	IDR	RO	Port input data. These bits are read-only and can only	X

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

Port output register (GPIOx_OUTDR) (x=A/B/C/D)

Offset address: 0x0C

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

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved.	0
[15:0]	ODR	RW	Port output data. The data can only be operated in form of 16 bits. The IO port outputs the value of these registers externally.	0

Port reset/setting register (GPIOx_BSHR) (x=A/B/C/D)

Offset address: 0x10

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

Bit	Name	Access	Description	Reset value
[31:16]	BR	WO	Setting these bits will clear the corresponding ODR 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 will take effect.	0
[15:0]	BS	WO	Setting these bits will set the corresponding ODR 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 will take effect.	0

Port reset register (GPIOx_BCR) (x=A/B/C/D)

Offset address: 0x14

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
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]	BR	WO	Setting these bits will clear the corresponding ODR bits, and writing 0 has no effect. These bits can only be accessed in form of 16 bits.	0

Configuration lock register (GPIOx_LCKR) (x=A/B/C/D)

Offset address: 0x18

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LCKK
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCK15	LCK14	LCK13	LCK12	LCK11	LCK10	LCK9	LCK8	LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
16	LCKK	RW	Lock key; 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]	LCK	RW	When the bit is 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 the configuration registers GPIOx_CFGLR and GPIOx_CFGHR.	0

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

10.3.2 AFIO Register

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

Table 10-11 List of AFIO Related Registers

Name	Access address	Description	Reset value
R32_AFIO_ECR	0x40010000	Event control register	0x00000000

R32_AFIO_PCFR	0x40010004	Remap register	0x00000000
R32_AFIO_EXTICR1	0x40010008	External interrupt configuration register 1	0x00000000
R32_AFIO_EXTICR2	0x4001000C	External interrupt configuration register 2	0x00000000
R32_AFIO_EXTICR3	0x40010010	External interrupt configuration register 3	0x00000000
R32_AFIO_EXTICR4	0x40010014	External interrupt configuration register 4	0x00000000

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	Enable event output bit; by setting this bit, the EVENTOUT of core will be connected to the selected IO ports of PORT and PIN.	0
[6:4]	PORT	RW	For selecting the port of core output EVENTOUT: 000: Select PA port; 001: Select PB port; 010: Select PC port; 011: Select PD port; Others: Reserved.	0
[3:0]	PIN	RW	The value of this bit is used to determine the number of specific pin that selects the core EVENTOUT to the port. Values 0-15 correspond to pins 0-15 of the PX selected in PORT.	0

Remap register (AFIO_PCFR)

Offset address: 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					SWCFG[2:0]			Reserved							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PD01RM	CANRM[1:0]		Reserved	TIM3RM[1:0]		TIM2RM[1:0]		TIM1RM[1:0]		USART3RM[1:0]		USART2RM	USART1RM	I2C1RM	SPI1RM

Bit	Name	Access	Description	Reset value
[31:27]	Reserved	RO	Reserved.	0
[26:24]	SWCFG	WO	These bits are used to configure the IO port of the SW function and tracking function. SWD (RVSWD) is a debugging interface to access the core. After the system is reset, it is always used as the SWD port.	0

			0xx: Enable SWD (RVSWD); 100: Disable SWD (RVSWD), served as GPIO; Other: Invalid.	
[23:16]	Reserved	RO	Reserved.	0
15	PD01RM	RW	Pin PD0&PD1 remap bit; 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]	CANRM	RW	The CAN multiplexing function remaps bits, which can be read and written by the user. Control of CAN_RX and CAN_TX remapping: 00: CAN_RX is mapped to PA11, CAN_TX is mapped to PA12; 10: CAN_RX is mapped to PB8, CAN_TX is mapped to PB9; 01/11: Reserved.	0
12	Reserved	RO	Reserved.	0
[11:10]	TIM3RM	RW	Timer 3 remap bits; these bits can be read and written by the user. It controls the remapping of channels 1 to 4 of timer 3 on the GPIO port: 00: Default mapping (CH1/PA6, CH2/PA7, CH3/PB0, CH4/PB1); 01: Reserved; 10: Partial mapping (CH1/PB4, CH2/PB5, CH3/PB0, CH4/PB1); 11: Complete mapping (CH1/PC6, CH2/PC7, CH3/PC8, CH4/PC9); *Note: Remapping does not affect TIM3_ETR on PD2.*	0
[9:8]	TIM2RM	RW	Remap bits of timer 2. These bits can be written and read by the user. It controls the mapping of timer 2 channels 1 to 4 and external trigger (ETR) on the GPIO port: 00: Default mapping (CH1/ETR/PA0, CH2/PA1, CH3/PA2, CH4/PA3); 01: Partial mapping (CH1/ETR/PA15, CH2/PB3, CH3/PA2, CH4/PA3); 10: Partial mapping (CH1/ETR/PA0, CH2/PA1, CH3/PB10, CH4/PB11); 11: Complete mapping (CH1/ETR/PA15, CH2/PB3, CH3/PB10, CH4/PB11).	0

[7:6]	TIM1RM	RW	Remap bit of timer 1. These bits can be written and read by the user. It controls the mapping of timer 1 channels 1 to 4, 1N to 3N, external trigger (ETR) and break input (BKIN) on the GPIO port: 00: Default mapping (ETR/PA12, CH1/PA8, CH2/PA9, CH3/PA10, CH4/PA11, BKIN/PB12, CH1N/PB13, CH2N/PB14, CH3N/PB15); 01: Partial mapping (ETR/PA12, CH1/PA8, CH2/PA9, CH3/PA10, CH4/PA11, BKIN/PA6, CH1N/PA7, CH2N/PB0, CH3N/PB1); 10: Reserved; 11: Complete mapping (ETR/PE7, CH1/PE9, CH2/PE11, CH3/PE13, CH4/PE14, BKIN/PE15, CH1N/PE8, CH2N/PE10, CH3N/PE12).	0
[5:4]	USART3RM	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 multiplexing functions of USART3 on the GPIO port: 00: Default mapping (TX/PB10, RX/PB11, CK/PB12, CTS/PB13, RTS/PB14); 01: Partial remapping (TX/PC10, RX/PC11, CK/PC12, CTS/PB13, RTS/PB14); 10: Reserved; 11: Complete remapping (TX/PD8, RX/PD9, CK/PD10, CTS/PD11, RTS/PD12).	0
3	USART2RM	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 multiplexing functions of USART2 on the GPIO port: 0: Default mapping (CTS/PA0, RTS/PA1, TX/PA2, RX/PA3, CK/PA4); 1: Remapping (CTS/PD3, RTS/PD4, TX/PD5, RX/PD6, CK/PD7).	0
2	USART1RM	RW	Remap bit of USART1. This bit can be read and written by the user. It controls the mapping of the TX and RX multiplexing functions of USART1 at the GPIO port: 0: Default mapping (TX/PA9, RX/PA10); 1: Remapping (TX/PB6, RX/PB7).	0
1	I2C1RM	RW	Remap of I2C1. This bit can be read and written by the user. It controls the mapping of the SCL and SDA multiplexing functions of I2C1 at the GPIO port: 0: Default mapping (SCL/PB6, SDA/PB7); 1: Remapping (SCL/PB8, SDA/PB9).	0
0	SPI1RM	RW	Remap of SPI1. This bit can be read and written by the	0

			user. It controls the mapping of NSS, SCK, MISO and MOSI multiplexing functions of SPI1 at the GPIO port: 0: Default mapping (NSS/PA4, SCK/PA5, MISO/PA6, MOSI/PA7); 1: Remapping (NSS/PA15, SCK/PB3, MISO/PB4, MOSI/PB5).	

External interrupt configuration register 1 (AFIO_EXTICR1)

Offset address: 0x08

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

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved.	0
[15:12] [11:8] [7:4] [3:0]	EXTIx	RW	External interrupt input pin configuration bit. It is 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; Others: Reserved.	0

External interrupt configuration register 2 (AFIO_EXTICR2)

Offset address: 0x0C

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

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved.	0
[15:12] [11:8] [7:4] [3:0]	EXTIx	RW	External interrupt input pin configuration bit, 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;	0

			Others: Reserved.	

External interrupt configuration register 3 (AFIO_EXTICR3)

Offset address: 0x10

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

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved.	0
[15:12] [11:8] [7:4] [3:0]	EXTIx	RW	External interrupt input pin configuration bit. It is 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; Others: Reserved.	0

External interrupt configuration register 4 (AFIO_EXTICR4)

Offset address: 0x14

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

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved.	0
[15:12] [11:8] [7:4] [3:0]	EXTIx	RW	External interrupt input pin configuration bit. It is 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; Others: Reserved.	0

Chapter 11 Direct Memory Access Control (DMA)

The module description in this chapter applies to the full range of CH32F103 and CH32V103 microcontrollers.

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

The DMA controller has 7 channels, and each channel 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

- 7 independent configurable channels
- Each channel is directly connected to dedicated hardware DMA request, and supports software trigger
- Supporting cyclic buffer management
- The priority of requests between multiple channels can be set by software programming (highest, high, medium and low level). When the priority settings are equal, it is determined by the channel number (the channel number is low and the priority is high)
- Supporting 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 transmission: 65535 at most

11.2 Functional Description

11.2.1 DMA Channel Processing

1) Arbitration Priority

DMA requests generated by 7 independent channels are inputted to the DMA controller through logic or structure, and currently only one 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 of the DMA_CFGRx register, including 4 levels: the highest, 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 channle 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 includes 3 DMA data transfer methods:

- 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 this mode (MEM2MEM=1) is configured, 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 in the PL[1:0] bits of 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, transmission half completion, transmission completion, and transmission error interrupt enable bit,
6. Set the ENABLE bit of the DMA_CCRx register to enable channel x (x=1/2/3/4/5/6/7).

Note: The control bits of 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 of 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 will be automatically reloaded to its initial value, and the internal peripheral and memory address register will also be reloaded to the initial address value set by the DMA_PADDRx and DMA_MADDRx registers, DMA operation will continue until the channel or DMA mode is switched off.

4) DMA Processing Status

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

will be generated. If TCIE is set in the DMA_CCRx register, an interrupt will be generated.

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

When the application program queries the status of the DMA channel, it will firstly access the GIFx bit of 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 Total Data Transmission Amount/Data Width/Alignment

The total amount 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 data to be transmitted. When EN=0, write the setting value. After the DMA transmission channel is switched on during EN=1, this register will become a read-only attribute, and the value will decrease 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 of 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] is used to set the peripheral address fetch data size and address self-increment size. MSIZE[1:0] is 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 port width	Destination port width	Number of data items to transfer	Source: Address/ data	Destination: 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 setting 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,
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/B1 0x02/B2 0x03/B3	
16	16	4	0x00/B1B0 0x02/B3B2	0x00/B1B0 0x02/B3B2	

			0x04/B5B4 0x06/B7B6	0x04/B5B4 0x06/B7B6	the high bit will be removed. ● Data storage method: Littile-endian mode; low bytes are stored at the low address and low bytes are stored at high address
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/B1 0x02/B2 0x03/B3	
32	16	4	0x00/B3B2B1B0 0x04/B7B6B5B4 0x08/BBBAB9B8 0x0C/BFBEBDBC	0x00/B1B0 0x02/B3B2 0x04/B5B4 0x06/B7B6	
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

The DMA controller provides 7 channels, and 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 11-2 DMA Request Mapping

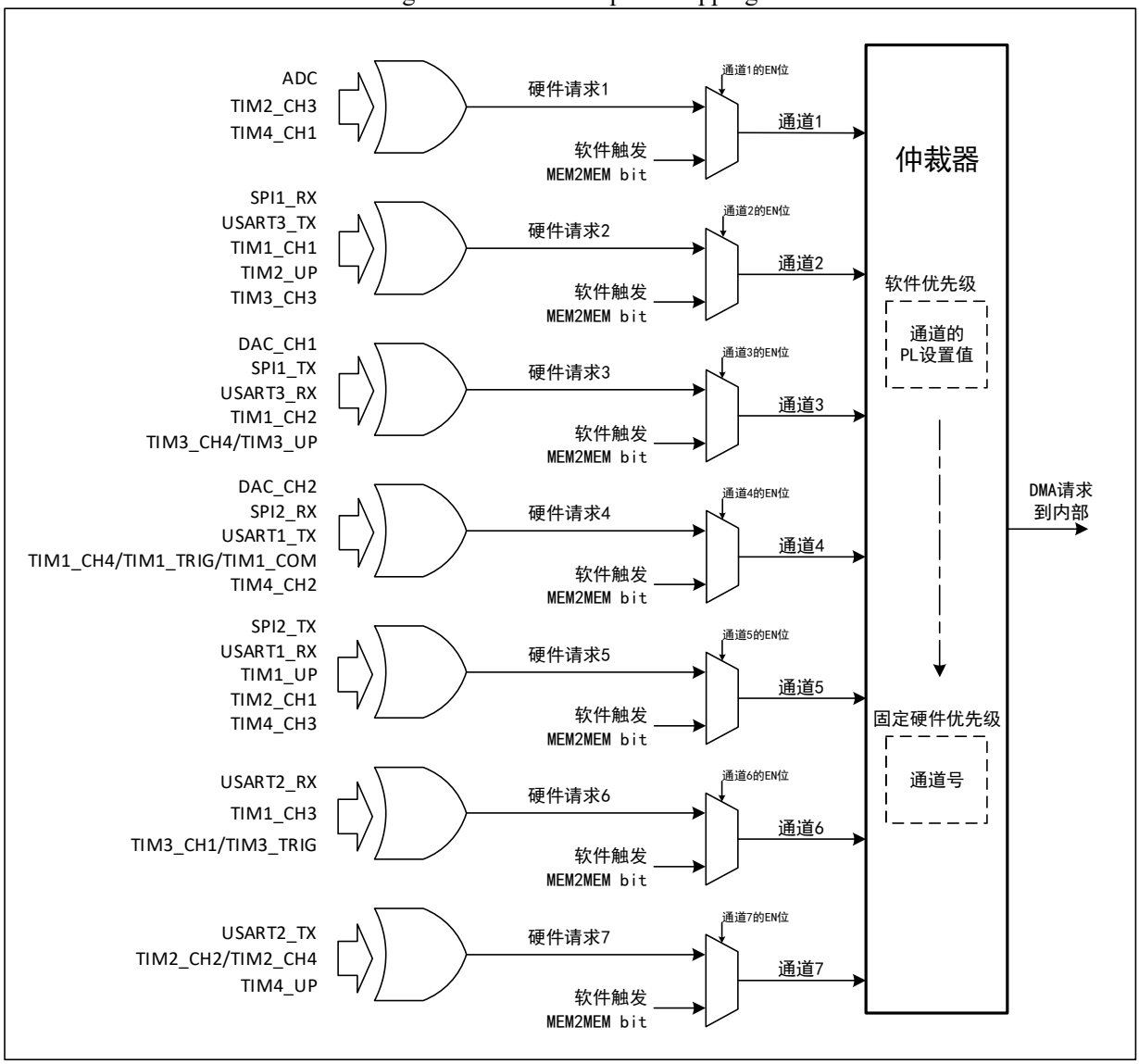


Table 11-2 Peripheral Mapping Table at Each Channel of DMA

Peripheral	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7
ADC	ADC						
DAC			DAC_CH1	DAC_CH2			
SPIx		SPI1_RX	SPI1_TX	SPI2_RX	SPI2_TX		
USARTx		USART3_TX	USART3_RX	USART1_TX	USART1_RX	USART2_RX	USART2_TX
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

11.3 Register Description

Table 11-3 List of DMA Related Registers

Name	Access address	Description	Reset value
R32_DMA_INTFR	0x40020000	DMA interrupt status register	0x00000000
R32_DMA_INTFCR	0x40020004	DMA interrupt flag clear register	0x00000000
R32_DMA_CFGR1	0x40020008	DMA channel 1 configuration register	0x00000000
R32_DMA_CNTR1	0x4002000C	DMA channel 1 number of data register	0x00000000
R32_DMA_PADDR1	0x40020010	DMA channel 1 peripheral address register	0x00000000
R32_DMA_MADDR1	0x40020014	DMA channel 1 memory address register	0x00000000
R32_DMA_CFGR2	0x4002001C	DMA channel 2 configuration register	0x00000000
R32_DMA_CNTR2	0x40020020	DMA channel 2 number of data register	0x00000000
R32_DMA_PADDR2	0x40020024	DMA channel 2 peripheral address register	0x00000000
R32_DMA_MADDR2	0x40020028	DMA channel 2 memory address register	0x00000000
R32_DMA_CFGR3	0x40020030	DMA channel 3 configuration register	0x00000000
R32_DMA_CNTR3	0x40020034	DMA channel 3 number of data register	0x00000000
R32_DMA_PADDR3	0x40020038	DMA channel 3 peripheral address register	0x00000000
R32_DMA_MADDR3	0x4002003C	DMA channel 3 memory address register	0x00000000
R32_DMA_CFGR4	0x40020044	DMA channel 4 configuration register	0x00000000
R32_DMA_CNTR4	0x40020048	DMA channel 4 number of data register	0x00000000
R32_DMA_PADDR4	0x4002004C	DMA channel 4 peripheral address register	0x00000000
R32_DMA_MADDR4	0x40020050	DMA channel 4 memory address register	0x00000000
R32_DMA_CFGR5	0x40020058	DMA channel 5 configuration register	0x00000000
R32_DMA_CNTR5	0x4002005C	DMA channel 5 number of data register	0x00000000
R32_DMA_PADDR5	0x40020060	DMA channel 5 peripheral address register	0x00000000
R32_DMA_MADDR5	0x40020064	DMA channel 5 memory address register	0x00000000
R32_DMA_CFGR6	0x4002006C	DMA channel 6 configuration register	0x00000000
R32_DMA_CNTR6	0x40020070	DMA channel 6 number of data register	0x00000000
R32_DMA_PADDR6	0x40020074	DMA channel 6 peripheral address register	0x00000000
R32_DMA_MADDR6	0x40020078	DMA channel 6 memory address register	0x00000000
R32_DMA_CFGR7	0x40020080	DMA channel 7 configuration register	0x00000000
R32_DMA_CNTR7	0x40020084	DMA channel 7 number of data register	0x00000000
R32_DMA_PADDR7	0x40020088	DMA channel 7 peripheral address register	0x00000000
R32_DMA_MADDR7	0x4002008C	DMA channel 7 memory address register	0x00000000

DMA interrupt status register (DMA_INTFR)

Offset address: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				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
[31:28]	Reserved	RO	Reserved.	0
27/23/19/15/11/7/3	TEIFx	RO	Transmission error flag of channel x (x=1/2/3/4/5/6/7): 1: A transmission error has occurred on channel x; 0: No transmission error has occurred on channel x. It is set by hardware, and write CTEIFx bit by software to clear this flag.	0
26/22/18/14/10/6/2	HTIFx	RO	Transmission half completion of channel x (x=1/2/3/4/5/6/7): 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
25/21/17/13/9/5/1	TCIFx	RO	Transmission completion flag of channel x (x=1/2/3/4/5/6/7): 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
24/20/16/12/8/4/0	GIFx	RO	Global interrupt flag of channel x (x=1/2/3/4/5/6/7): 1: TEIFx or HTIFx or TCIFx is generated on channel x; 0: No TEIFx or HTIFx or TCIFx is generated on channel x. It is set by hardware, and write CGIFx bit by software to clear this flag.	0

DMA interrupt flag clear register (DMA_INTFCR)

Offset address: 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				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
[31:28]	Reserved	RO	Reserved.	0
27/23/19/15/11/7/3	CTEIFx	WO	Clear the transmission error flag of channel x (x=1/2/3/4/5/6/7): 1: Clear the TEIFx flag in the DMA_INTFR register;	0

			0: No effect.	
26/22/18/ 14/10/6/2	CHTIFx	WO	Clear the transmission half completion of channel x (x=1/2/3/4/5/6/7): 1: Clear the HTIFx flag in the DMA_INTFR register; 0: No effect.	0
25/21/17/ 13/9/5/1	CTCIFx	WO	Clear the transmission completion flag of channel x (x=1/2/3/4/5/6/7): 1: Clear the TCIFx flag in the DMA_INTFR register; 0: No effect.	0
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): 1: Clear the TEIFx/HTIFx/TCIFx/ GIFx flag in the DMA_INTFR register; 0: No effect.	0

DMA channel x configuration register (DMA_CFGRx) (x=1/2/3/4/5/6/7)

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	MEM2 MEM	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	RW	Channel priority level setting: 00: Low; 01: Medium; 10: High; 11: Very high.	0
[11:10]	MSIZE	RW	Memory size setting: 00: 8 bits; 01: 16 bits; 10: 32 bits; 11: Reserved.	0
[9:8]	PSIZE	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:	0

			1: Enable circular mode; 0: Disable circular mode.	
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

DMA channel x number of data register (DMA_CNTRx) (x=1/2/3/4/5/6/7)

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

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

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved.	0
[15:0]	NDT	RW	Number of data to 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 will occur regardless of whether the channel is switched on or not.

DMA channel x peripheral address register (DMA_PADDRx) (x=1/2/3/4/5/6/7)

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

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

Bit	Name	Access	Description	Reset value
[31:0]	PA	RW	Peripheral base address, as the source or destination address of peripheral data transmission. When PSIZE[1:0]='01' (16 bits), the module will automatically ignore bit0, and the operation address will be automatically aligned with 2 bytes. When PSIZE[1:0]='10' (32 bits), the module will automatically ignore bit[1:0], and the operation address will be automatically aligned with 4 bytes.	0

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

DMA channel x memory address register (DMA_MADDRx) (x=1/2/3/4/5/6/7)

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

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

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

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

Chapter 12 Analog-to-digital Converter (ADC)

The module description in this chapter applies to the full range of CH32F103 and CH32V103 microcontrollers.

The ADC module contains a 12-bit successive approximation analog-to-digital converter 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 and external trigger mode. The analog watchdog function can be used to monitor whether the channel voltage is within the threshold range.

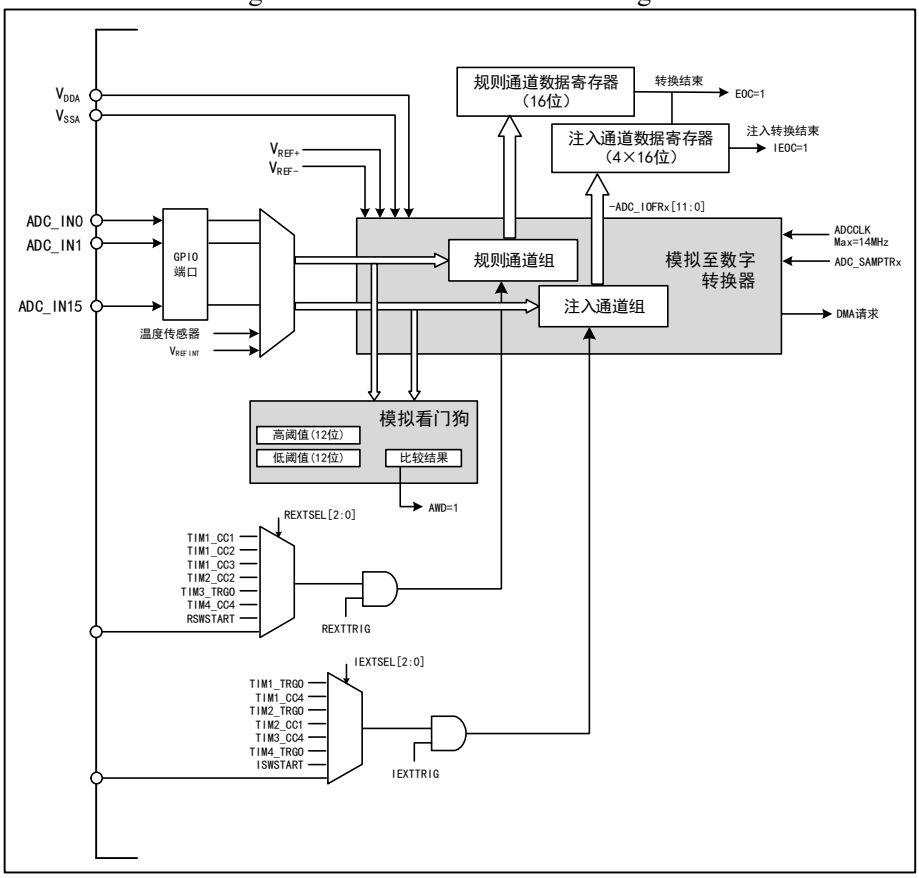
12.1 Main Features

- 12-bit resolution
- It supports sampling of 16 external channels and 2 internal sources
- Multiple sampling conversion methods 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 triggering
- Analog watchdog monitors the channel voltage, and has self-calibration function
- ADC channel input range: $0 \le V_{IN} \le V_{DDA}$

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) Sampling 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] of the RCC_CFGR0 register, and the maximum cannot exceed 14MHz.

3) Channel Cofiguration

The ADC module provides 18 channels of sampling sources, including 16 external channels and 2 internal channels. They can be configured into two 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] of 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] of 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 of VREFINT: 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 two ADC clock cycles at least.

5) Progammable Sampling 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:

TCONV = Sampling time + 12.5T_{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 contniuous 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 substracted 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 of 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.

Table 12-1 External Trigger Source of Regular Group Channel

REXTSEL[2:0]	Trigger source	Type
000	CC1 event of timer 1	Internal signal from on-chip

001	CC2 event of timer 1	timers
010	CC3 event of timer 1	
011	CC2 event of timer 2	
100	TRGO event of timer 3	
101	CC4 event of timer 4	
110	EXTI line 11	From external pin
111	RSWSTART bit set to 1 by software	Software control bit

Table 12-2 External Trigger Source of Injected Group Channel

IEXTSEL[2:0]	Trigger source	Type
000	TRGO event of timer 1	Internal signal from on-chip timers
001	CC4 event of timer 1	
010	TRGO event of timer 2	
011	CC1 event of timer 2	
100	CC4 event of timer 3	
101	TRGO event of timer 4	
110	EXTI line 15	From external pin
111	ISWSTART 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 channel or injected channel.
	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 one group of 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 of 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 will be saved in the 16-bit ADC_RDATAR register, and the EOC flag will be set. If the EOCIE bit is set, the ADC interrupt will be triggered.

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

2) Single Scanning Mode Conversion

Enter the ADC scanning mode by setting the SCAN bit of 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 will be 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 of channel, the current conversion will be reset, and the sequence of the injected channel will be carried out in a single scan mode. After the scannnig and conversion of all selected injected group channels are completed, the previous interrupted regular group channel conversion will be restored.

If a regular channel start event occurs currently while scanning the injected group channel sequence, the injected group conversion will not be interrupted, but the regular sequence conversion will be 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 will be 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]) 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 the RDISCEN or IDISCEN bit of 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 will execute a short sequence of scan conversion.

The length of the short sequence n (n<=8) is defined in the DISCNUM[2:0] of 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 in the RLEN[3: of the ADC_RSQR1 register. 0]; when IDISCEN is 1, it is the discontinuous mode of the injected group, and the total length to be converted is defined in ILEN[1:0] of 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, DISCNUM[2:0]=3, RLEN[3:0]=8, channels to be converted=1, 3, 2, 5, 8, 4, 10, 6

The first external trigger: Conversion sequence is 1, 3 and 2

The second external trigger: Conversion sequence is 5, 8 and 4

The third external trigger: Conversion sequence is 10 and 6, and the EOC event is generated in the meantime

The fourth external trigger: Conversion sequence is 1, 3 and 2

Example of injected group discontinuous mode:

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

The first external trigger: Conversion sequence is 1

The second external trigger: Conversion sequence is 3

The third external trigger: Conversion sequence is 2, and the EOC and IEOC events are generated in the meantime

The fourth external trigger: Conversion sequence is 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 one group of conversion.

4) Continuous Conversion

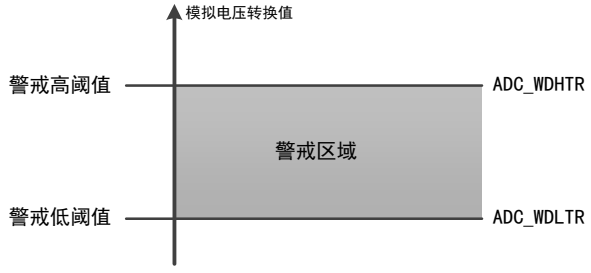
Enter the ADC continuous conversion mode by setting the CONT bit of the ADC_CTLR2 register to 1. In this mode, another conversion is started immediately after the previous ADC conversion is completed. The conversion will not stop on the last channel of the selected group, but will continue from the first 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 lowest 12 valid bits of 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 of 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 of 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 (℃) is as follows:

Temperature (℃) = ((V_{SENSE}-V_{25})/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/℃)

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

Note: A setup time is required for the internal temperature sensor power-on (TSVREFE bit is changed to 1 from 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.

12.3 Register Description

Table 12-5 List of ADC Related Registers

Name	Access address	Description	Reset value
R32_ADC_STATR	0x40012400	ADC status register	0x00000000
R32_ADC_CTLR1	0x40012404	ADC control register 1	0x00000000
R32_ADC_CTLR2	0x40012408	ADC control register 2	0x00000000
R32_ADC_SAMPTR1	0x4001240C	ADC sample time configuration register 1	0x00000000
R32_ADC_SAMPTR2	0x40012410	ADC sample time configuration register 2	0x00000000
R32_ADC_IOFR1	0x40012414	ADC injected channel data offset register 1	0x00000000
R32_ADC_IOFR2	0x40012418	ADC injected channel data offset register2	0x00000000
R32_ADC_IOFR3	0x4001241C	ADC injected channel data offset register 3	0x00000000
R32_ADC_IOFR4	0x40012420	ADC injected channel data offset register 4	0x00000000
R32_ADC_WDHTR	0x40012424	ADC watchdog high threshold register	0x00000000

R32_ADC_WDLTR	0x40012428	ADC watchdog low threshold register	0x00000000
R32_ADC_RSQR1	0x4001242C	ADC regular channel sequence register 1	0x00000000
R32_ADC_RSQR2	0x40012430	ADC regular channel sequence register 2	0x00000000
R32_ADC_RSQR3	0x40012434	ADC regular channel sequence register 3	0x00000000
R32_ADC_ISQR	0x40012438	ADC injected channel sequence register	0x00000000
R32_ADC_IDATAR1	0x4001243C	ADC injected data register 1	0x00000000
R32_ADC_IDATAR2	0x40012440	ADC injected data register 2	0x00000000
R32_ADC_IDATAR3	0x40012444	ADC injected data register 3	0x00000000
R32_ADC_IDATAR4	0x40012448	ADC injected data register 4	0x00000000
R32_ADC_RDATAR	0x4001244C	ADC regular data register	0x00000000

ADC status register (ADC_STATR)

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											RSTRT	ISTRT	IEOC	EOC	AWD

Bit	Name	Access	Description	Reset value
[31:5]	Reserved	RO	Reserved.	0
4	RSTRT	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	ISTRT	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	IEOC	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).	

ADC control register 1 (ADC_CTLR1)

Offset address: 0x04

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISCNUM[2:0]			IDISCEN	RDISCEN	IAUTO	AWDSGL	SCAN	IEOCIE	AWDIE	EOCIE	AWDCH[4:0]				

Bit	Name	Access	Description	Reset value
[31:25]	Reserved	RO	Reserved.	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	RAWDEN	RW	Analog watchdog enable bit on regular channels: 1: Enable analog watchdog on regular channels; 0: Disable analog watchdog on regular channels;	0
22	IAWDEN	RW	Analog watchdog enable bit on injected channels: 1: Enable analog watchdog on injected channels; 0: Disable analog watchdog on injected channels;	0
[21:16]	Reserved	RO	Reserved.	0
[15:13]	DISCNUM	RW	In discontinuous mode, the number of regular channels to be converted after external triggering: 000: 1 channel; … 111: 8 channels.	0
12	IDISCEN	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	RDISCEN	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	IAUTO	RW	After regular channel is enabled, automatically converted to injected channel group 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*	0

			needs to be disabled in this mode.	
9	AWDSGL	RW	In scan mode, analog watchdog enable bit on a single channel: 1: Enable analog watchdog on a 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: Disable scan mode.	0
7	IEOCIE	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	RW	Analog watchdog channel selection bit: 00000: Analog input channel 0; 00001: Analog input channel 1; … 10001: Analog input channel 17.	0

ADC control register 2 (ADC_CTLR2)

Offset address: 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								TS VREFE	RSW START	ISW START	REXT TRIG	REXTSEL[2:0]			Reserved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IEXT TRIG	IEXTSEL[2:0]			ALIGN	Reserved		DMA	Reserved				RST CAL	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 bit:	0

			1: Enable the temperature sensor and V_{REFINT} channel; 0: Disable the temperature sensor and V_{REFINT} channel;	
22	RSWSTART	RW	Start conversion of regular channels, set by software to start: 1: Start conversion of regular channels; 0: Reset status. This bit is set by software, and cleared by hardware after the conversion starts.	0
21	ISWSTART	RW	Start conversion of injected channels, set by software to start: 1: Start conversion of injected channels; 0: Reset status. This bit is set by software, and cleared by hardware or software after the conversion starts.	0
20	REXTTRIG	RW	External trigger conversion mode enable for regular channels: 1: Enable conversion on external event; 0: Disable conversion on external event.	0
[19:17]	REXTSEL	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 line 11; 111: RSWSTART software trigger.	0
16	Reserved	RO	Reserved.	0
15	IEXTTRIG	RW	External trigger conversion mode enable for injected channels: 1: Enable conversion on external event; 0: Disable conversion on external event.	0
[14:12]	IEXTSEL	RW	External trigger event select for injected channels: 000: TRGO event of timer 1; 001: CC4 event of timer 1; 010: TRGO event of timer 2; 011: CC1 event of timer 2; 100: CC4 event of timer 3; 101: TRGO event of timer 4; 110: EXTI line 15; 111: ISWSTART software trigger.	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:	0

			1: Enable DMA mode; 0: Disable DMA mode.	
[7:4]	Reserved	RO	Reserved.	0
3	RSTCAL	RW	Reset calibration, this bit is set by software, and cleared by hardware after reset: 1: Initialize calibration register; 0: The calibration register initialized. *Note: If RSTCAL is set while the conversion is in progress, it takes extra cycles to clear the calibration register.*	0
2	CAL	RW	A/D calibration, set by software and cleared by hardware when the calibration is completed. 1: Enable the calibration: 0: Calibration completed.	0
1	CONT	RW	Continuous conversion enable: 1: Continuous conversion mode; 0: 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

ADC sample time configuration register 1 (ADC_SAMPTR1)

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	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.	

ADC sample time configuration register 2 (ADC_SAMPTR2)

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	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.	

ADC injected channel data offset register x (ADC_IOFRx) (x=1/2/3/4)

Offset address: 0x14-0x20

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

Bit	Name	Access	Description	Reset value
[31:12]	Reserved	RO	Reserved.	0
[11:0]	IOFFSETx	RW	Data offset for injected channel x. When the injected channel is converted, this value defines the value to be subtracted from the original conversion data. The result of the conversion can be read in the ADC_IDATARx register	0

ADC watchdog high threshold register (ADC_WDHTR)

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	RW	Analog watchdog high threshold setting value.	0

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

ADC watchdog low threshold register (ADC_WDLTR)

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				LT[11:0]											

Bit	Name	Access	Description	Reset value
[31:12]	Reserved	RO	Reserved.	0
[11:0]	LT	RW	Analog watchdog low threshold setting value.	0

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

ADC regular channel sequence register 1 (ADC_RSQR1)

Offset address: 0x2C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								RLEN[3:0]				RSQ16[4:1]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSQ16[0]	RSQ15[4:0]					RSQ14[4:0]						RSQ13[4:0]			

Bit	Name	Access	Description	Reset value
[31:24]	Reserved	RO	Reserved.	0
[23:20]	RLEN	RW	The number of channels to be converted in the regular channel conversion sequence: 0000-1111: 1-16 conversions.	0
[19:15]	RSQ16	RW	Number (0-17) of the 16th conversion in regular sequence.	0
[14:10]	RSQ15	RW	Number (0-17) of the 15th conversion in regular sequence.	0
[9:5]	RSQ14	RW	Number (0-17) of the 14th conversion in regular sequence.	0
[4:0]	RSQ13	RW	Number (0-17) of the 13th conversion in regular sequence.	0

ADC regular channel sequence register 2 (ADC_RSQR2)

Offset address: 0x30

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		RSQ12[4:0]					RSQ11[4:0]					RSQ10[4:1]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSQ10[0]		RSQ9[4:0]					RSQ8[4:0]					RSQ7[4:0]			

Bit	Name	Access	Description	Reset value
[31:30]	Reserved	RO	Reserved.	0
[29:25]	RSQ12	RW	Number (0-17) of the 12th conversion in regular sequence.	0
[24:20]	RSQ11	RW	Number (0-17) of the 11th conversion in regular sequence.	0
[19:15]	RSQ10	RW	Number (0-17) of the 10th conversion in regular sequence.	0
[14:10]	RSQ9	RW	Number (0-17) of the 9th conversion in regular sequence.	0
[9:5]	RSQ8	RW	Number (0-17) of the 8th conversion in regular sequence.	0
[4:0]	RSQ7	RW	Number (0-17) of the 7th conversion in regular sequence.	0

ADC regular channel sequence register 3 (ADC_RSQR3)

Offset address: 0x34

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		RSQ6[4:0]					RSQ5[4:0]					RSQ4[4:1]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSQ4[0]		RSQ3[4:0]					RSQ2[4:0]					RSQ1[4:0]			

Bit	Name	Access	Description	Reset value
[31:30]	Reserved	RO	Reserved.	0
[29:25]	RSQ6	RW	Number (0-17) of the 6th conversion in regular sequence.	0
[24:20]	RSQ5	RW	Number (0-17) of the 5th conversion in regular sequence.	0
[19:15]	RSQ4	RW	Number (0-17) of the 4th conversion in regular sequence.	0
[14:10]	RSQ3	RW	Number (0-17) of the 3th conversion in regular sequence.	0
[9:5]	RSQ2	RW	Number (0-17) of the 2nd conversion in regular sequence.	0
[4:0]	RSQ1	RW	Number (0-17) of the 1st conversion in regular sequence.	0

ADC injected channel sequence register (ADC_ISQR)

Offset address: 0x38

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										ILEN[1:0]		ISQ4[4:1]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISQ4[0]		ISQ3[4:0]					ISQ2[4:0]					ISQ1[4:0]			

Bit	Name	Access	Description	Reset value
[31:22]	Reserved	RO	Reserved.	0
[21:20]	ILEN	RW	The number of channels to be converted in the injected channel conversion sequence: 00-11: 1-4 conversions.	0
[19:15]	ISQ4	RW	Number (0-17) of the 4th conversion in injected sequence.	0
[14:10]	ISQ3	RW	Number (0-17) of the 3rd conversion in injected sequence.	0
[9:5]	ISQ2	RW	Number (0-17) of the 2nd conversion in injected sequence.	0
[4:0]	ISQ1	RW	Number (0-17) of the 1st conversion 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).

ADC injected data register x (ADC_IDATARx) (x=1/2/3/4)

Offset address: 0x3C-0x48

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

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved.	0
[15:0]	IDATA	RO	Injected channel conversion data (data left alignment or right alignment).	0

ADC regular data register (ADC_RDATAR)

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
RDATA[15:0]															

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved.	0
[15:0]	RDATA	RO	Regular channel conversion data (data left or right alignment).	0

Chapter 13 Touch Key Detection (TKEY)

The module description in this chapter applies to the full range of CH32F103 and CH32V103 microcontrollers.

The touchkey detection control (TKEY_F) unit of the CH32F103 series products, with the help of the voltage conversion function of the ADC module, realizes the touch button detection function by converting the capacitance to the voltage for sampling. The detection channel multiplexes the 16 external channels of the ADC, and the touchkey detection is realized through the single conversion mode of the ADC module.

The touchkey detection control (TKEY_V) unit of CH32V103 series products realizes the touchkey detection function by converting the capacitance change into the frequency change for sampling. The detection channel multiplexes the 16 external channels of the ADC. The application program judges the status of touch key based on the change of digital value.

13.1 Functional Description of TKEY_F

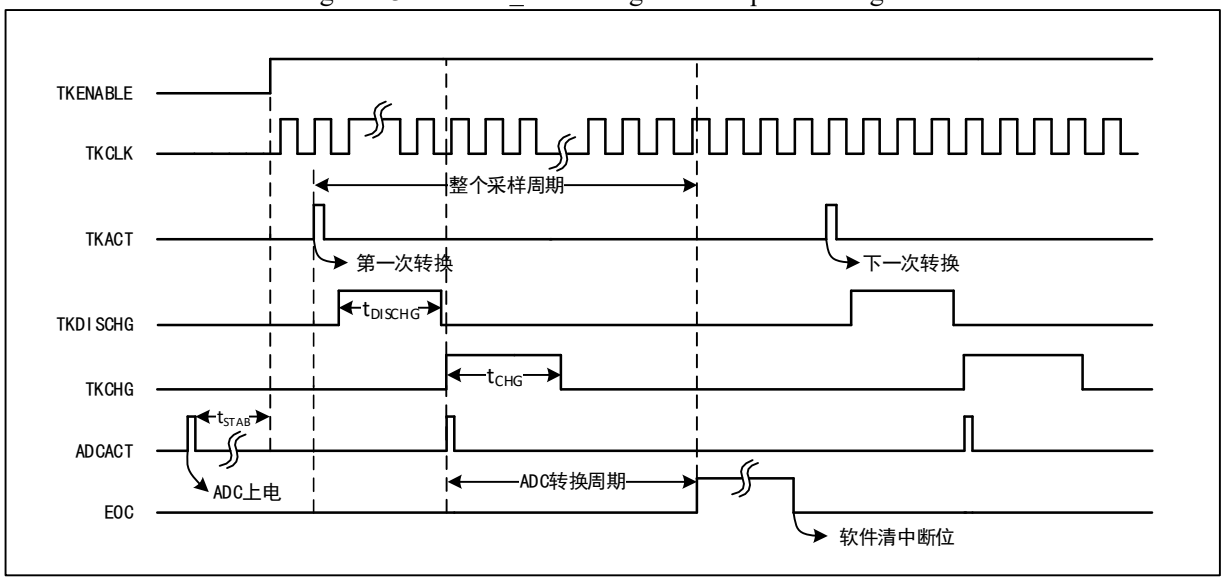
- TKEY_F enable

The coordination of ADC module is required during the TKEY_F detection process, so ADC module shall be at the power-down status (ADON=1) when the TKEY_F function is used. Then, set the TKENABLE bit of the ADC_CTLR1 register to 1, and switch on the TKEY_F unit function.

TKEY_F only supports single-time single-channel conversion mode. Configure the channel to be converted to the first of the regular group sequence of ADC module, and the software will start the conversion (writing the TKEY_ACT register).

Note: When TKEY_F conversion is not performed, the ADC channel configuration conversion function can still be retained.

Figure 13-1 TKEY_F Working Time Sequence Diagram



- Programmable sample time

For TKEY unit conversion, multiple system clock cycles (t_{DISCHG}) need to be used for discharge. Then, the channel is charged through multiple ADCCLK cycles (t_{CHG}) for sampling. The number of charging cycles is changed by the TKCGx[2:0] bit in the TKEY_CHARGE1 and TKEY_CHARGE2 registers. In each channel,

the sampling voltage can be regulated by different charging cycles.

The total process conversion time is calculated as follows:

T_{TKCONV} =Number of discharge cycles (T_{SYSCLK}) + number of charge cycles (T_{ADCCLK}) + 13.5T_{ADCCLK}

13.2 TKEY_F Operation Procedure

TKEY_F 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-time single-channel working mode, and a conversion is started by the "write operation" of the TKEY_F_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 of the ADC_CTLR1 register to 1, and switch on the TKEY_F unit.
2) Set the channel to be converted, write the channel 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 discharge time of the channel and write the TKEY_F_DISCHARGE register. The minimum discharge time is 1 system clock (Tsys) and the discharge time of all channels is the same. If you want to set it differently, you need to rewrite it.
4) Set the charging sampling time of the channel and write the TKEY_F_CHARGEx register to configure different charging time for each channel.
5) Write the TKEY_F_ACT register to start a sampling and conversion of TKEY_F. It is recommended to write 0x00 to reach the internal 0 and wait for the operation.
6) After the EOC conversion end flag bit of the ADC status register is set to 1, read the ADC_DR register to get the conversion value.
7) If you need to perform the 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 Description of TKEY_F Register

Table 13-1 List of TKEY_F Related Registers

Name	Access address	Description	Reset value
R32_TKEY_F_CHARGE1	0x4001240C	TKEY_F charge sample time register 1	0x00000000
R32_TKEY_F_CHARGE2	0x40012410	TKEY_F charge sample time register 2	0x00000000
R32_TKEY_F_DISCHARGE	0x4001243C	TKEY_F discharge time register	X
R32_TKEY_F_ACT	0x4001244C	TKEY_F start register	X
R32_TKEY_F_DR	0x4001244C	TKEY_F data register	X

TKEY_F charge sample time register 1 (TKEY_F_CHARGE1)

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	RW	TKCGx[2:0]: Select the charging sample time of channel x These are used to independently select the charging 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 sampling time register 1 (ADC_SAMPTR1) of the ADC module. When the ADC function is configured, it is the channel adoption time; when the TKEY_F function is configured, it is the channel charging time.

TKEY_F charge sample time register 2 (TKEY_F_CHARGE2)

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
[31:24]	Reserved	RO	Reserved.	0
[23:0]	TKCGx	RW	TKCGx[2:0]: Select the charging sample time of channel x These are used to independently select the charging 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 sampling time register 1 (ADC_SAMPTR2) of the ADC module. When the ADC function is configured, it is the channel adoption time; when the TKEY_F function is configured, it is the channel charging time.

TKEY_F discharge time register (TKEY_F_DISCHARGE)

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								TKDCRGT[7:0]							

Bit	Name	Access	Description	Reset value
[31:8]	Reserved	RO	Reserved.	0
[7:0]	TKDCRGT	WO	TKEY_F discharge time configuration value. The actual discharge time is (TKDCRGT+1) Tsys.	0

Note: This register maps the data input register 1 (ADC_IDATAR1) of the ADC module. Therefore, when the address register performs a "write operation", it will be executed as the discharge time register (TKEY_F_DISCHARGE) of the TKEY_F module; when a "read operation" is performed, it is executed as the injected data register 1 (ADC_IDATAR1) of the ADC module.

TKEY_F start register (TKEY_F_ACT)

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[7:0]							

Bit	Name	Access	Description	Reset value
[31:8]	Reserved	RO	Reserved.	0
[7:0]	TKACT	WO	Start. This register "write operation" starts a TKEY_F channel detection. It is recommended to always write 0x00.	0

TKEY_F data register (TKEY_F_DR)

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	RO	Converted data.	0

Note: This register maps the regular data register (ADC_RDATAR) of the ADC module.

13.4 TKEY_V Functional Description

- TKEY_V enable

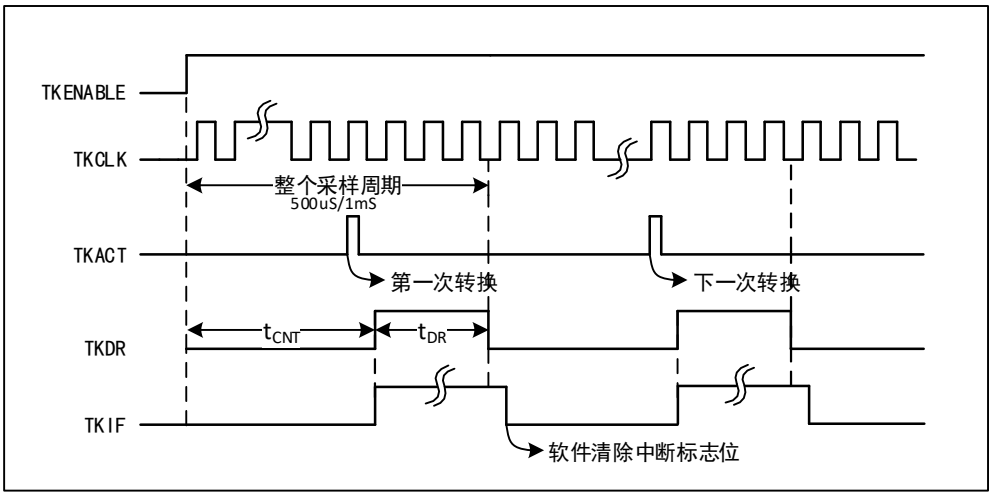
The TKEY_V unit detects that the channel selection and partial register addresses of the ADC module which are multiplexed internally. To use the TKEY_V function, you need to enable the ADC module (ADON=1) and switch on the ADC clock to access the relevant registers. Then, set the TOKENABLE bit of the TKEY_V_CTLR (ADC_CTLR1) register to 1 and switch on the TKEY_V unit function.

Note: Because the sampling channel selection is shared, the ADC and TKEY_V detection functions cannot be used at the same time.

- Operate Principle

Once the TKEY_V function is enabled, the hardware will automatically perform a periodic sampling and counting conversion process, and will notify the application code to take away the data within a fixed time (tDR) and start the next conversion after completing a conversion. This cycle process is is conducted automatically when TKEY_V is enabled. As shown in Figure 13-2, the hardware internally provides the pulse source TKCLK for counting. The application software selects 500us or 1ms as the current hardware counting cycle. When the internal counting statistics within the cycle are completed, the TKIF flag will be generated to notify the application code to read this conversion value. The application code needs to take away the data within the maximum length of 43uS (t_{DR}). Otherwise, the next round of conversion will affect the contents of the data register.

Figure 13-2 TKEY_V Working Time Sequence Diagram



13.5 TKEY_V Operation Step

TKEY_V detects the touch key with the statistical value based on the principle that the internal oscillation frequency change is affected by changing the capacitance. The concrete operation process is as follows:

1) Enable the ADCEN bit of RCC module and switch on the TKEY_V register operation authority.
2) Switch on the TKEY_V function, set the ACON bit to 1, and wake up the ADC module. Set the TKENABLE bit of the ADC_CTLR1 register to 1, and switch on the TKEY_V unit.
3) Configure the sampling period, operate the CCSEL[2:0] and TKCPS bits of the TKEY_V_CTLR register, and select 500us or 1ms cycle. The unit is internally timed by the AHB clock, so CCSEL[2:0] is required to be equal to the current AHB frequency. Otherwise, the sampling period will be too large or too small.
4) Configure the sampling channel and set the TKEY_V_CHANNEL register. The write operation of this register will trigger the start of a new cycle.

5) When the TKIF flag is set to 1, indicating that a conversion is completed, the count value of TKDR[13:0] in the TKEY_V_SDR register can be read. TKSTA indicates whether current TKDR[13:0] count value is valid. You need to write 1 and clear 0 by the software for the TKIF flag. If the TKIEN bit is set, the TKEY_V (ADC) interrupt will be triggered synchronously to enter the ADC interrupt service function.
6) Repeat the steps 3-5 to acquire the next count value. 3-4 are optional configuration.

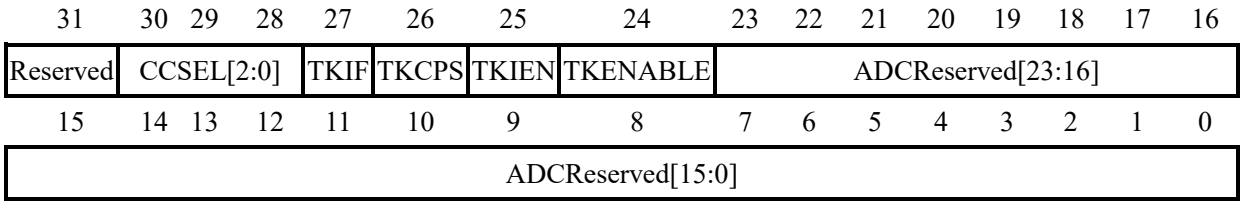
13.6 TKEY_V Register Description

Table 13-2 List of TKEY_V Related Registers

Name	Access address	Description	Reset value
R32_TKEY_V_CTLR	0x40012404	TKEY_V control register	0x00000000
R32_TKEY_V_CHANNEL	0x40012434	TKEY_V channel selection register	0x00000000
R32_TKEY_V_SDR	0x4001244C	TKEY_V status data register	X

TKEY_V control register (TKEY_V_CTLR)

Offset address: 0x04



Bit	Name	Access	Description	Reset value
31	Reserved	RO	Reserved.	0
[30:28]	CCSEL	RW	TKEY_V counting cycle time base: 000: 8MHz; 001: 12MHz; 010: 24MHz; 011: 36MHz; 100: 48MHz; 101: 56MHz; 110/111: Reserved, not matching. *Note: This bit selection shall match the current AHB clock frequency.*	0
27	TKIF	RW1	Counting conversion completion flag, set to 1 by hardware (automatically cleared to 0 after 43us), and write 1 by software to clear. 1: TKEY_V count completed; 0: TKEY_V count is being converted.	0
26	TKCPS	RW	TKEY_V count cycle selection: 1: The count conversion is conducted in a cycle of 1ms; 0: The count conversion is conducted in a cycle of 500us. *Note: This bit needs to match with CCSEL selection to guarantee the accurate time. If there is a deviation in the time base, the counting*	0

			period will change accordingly.	
25	TKIEN	RW	Count conversion completion interrupt enable. 1: Enable TKEY_V interrupt; interrupt service means ADC interrupt; 0: Disable TKEY_V interrupt.	0
24	TKENABLE	RW	TKEY module enable control. 1: Enable TKEY_V unit; 0: Disable TKEY_V unit.	0
[23:0]	ADCReserved	-	Reserved, with the same function as ADC_CTLR1 register.	-

Note: This register maps the control register 1 (ADC_CTLR1) of the ADC module.

TKEY_V channel selection register (TKEY_V_CHANNEL)

Offset address: 0x34

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

Bit	Name	Access	Description	Reset value
[31:5]	Reserved	RO	Reserved.	0
[4:0]	CHSEL	RO	TKEY_V count conversion channel selection. 00000b～01111b: Corresponding to channel 0～channel 15. *Note: If CHSEL is written during the TKEY_V unit count conversion period, the hardware will stop the conversion process and start a new count conversion cycle.*	0

Note: This register maps the regular channel sequence register 3 (ADC_RDATAR) of the ADC module.

TKEY_V status data register (TKEY_V_SDR)

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
TKSTA	Reserved							TKDR[13:0]							

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved.	0
15	TKSTA	RO	Current TKEY_V operating status: 1: Count conversion in process; the value in the TKDR[13:0] is invalid;	1

			0: Count suspended; TKDR can be read.	
14	Reserved	RO	Reserved.	0
[13:0]	TKDR	RO	TKEY_V count conversion value	0

Note: This register maps the regular data register (ADC_RDATAR) of the ADC module.

Chapter 14 Advanced-control Timer (TIM1)

The module description in this chapter applies to the full range of CH32F103 and CH32V103 microcontrollers.

The advanced-control timer module contains a powerful 16-bit auto-reload timer (TIM1), 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

The main features of advanced-control timer (TM1) include:

- 16-bit automatic reload counter, support up counting mode, down counting mode and up/down counting mode;
- 16-bit prescaler; the frequency division coefficient is dynamically adjustable from 1 to 65536;
- It supports four independent compare/capture;
- Each compare/capture channel supports multiple working modes, such as: input capture, output comparison, PWM generation and single pulse output;
- Supports complementary output with programmable dead zone time;
- Supports external signal control timer;
- Supports use a repeat counter to update the timer after the determination of the cycle;
- Supports use the break signal to reset the timer or put it in a determined status;
- Supports use DMA in multiple modes;
- Supports incremental encoder;
- Supports cascade connection and synchronization between timers

14.2 Principle and Structure

This section describes the internal structure of the advanced-control timer to lay the foundation for understanding the functional principles of the next section.

14.2.1 Overview

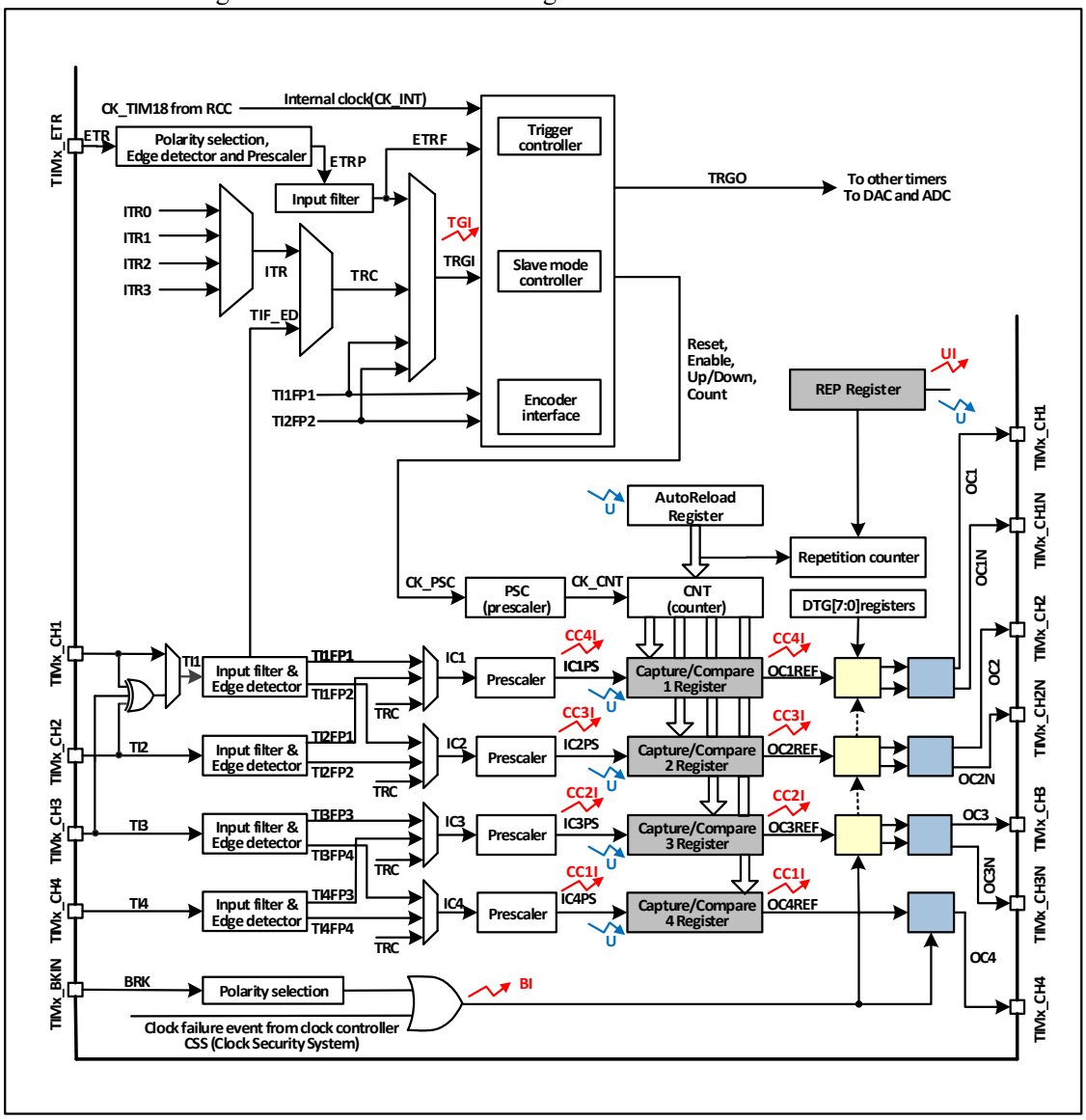
As shown in Figure 14-1, the structure of the advanced-control timer can be roughly divided into three parts: Input clock part, core counter part and compare/capture channel part.

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 counter part. 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.

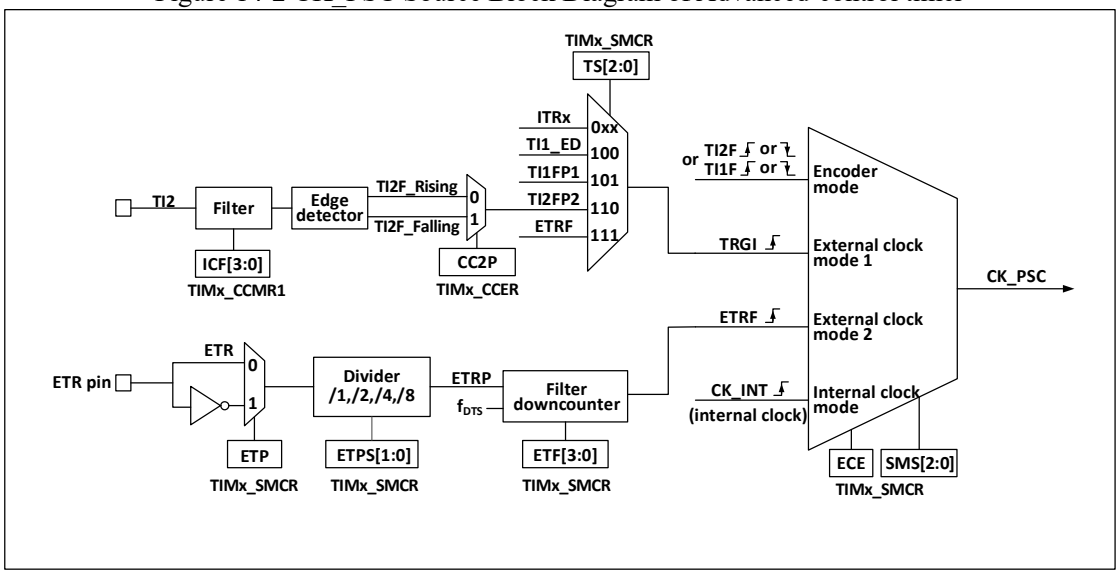
The advanced-control timer has four groups of compare/capture channels. On each group of compare/capture channel, pulses can be inputted from its dedicated pins or output waveforms to the pins, i.e., the compare/capture channels support input and output modes. The input of each channel of the compare/capture 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 Mode 1;

If SMS is set to 111b, the external clock source mode 1 will be enabled. When external clock source 1 is enabled, TRGI will be 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 Mode 2

Use external trigger mode 2 to count on every rising or falling edge of the external clock pin input. When the

ECE bit is set, the external clock source mode 2 will be used. When the external clock source mode 2 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.

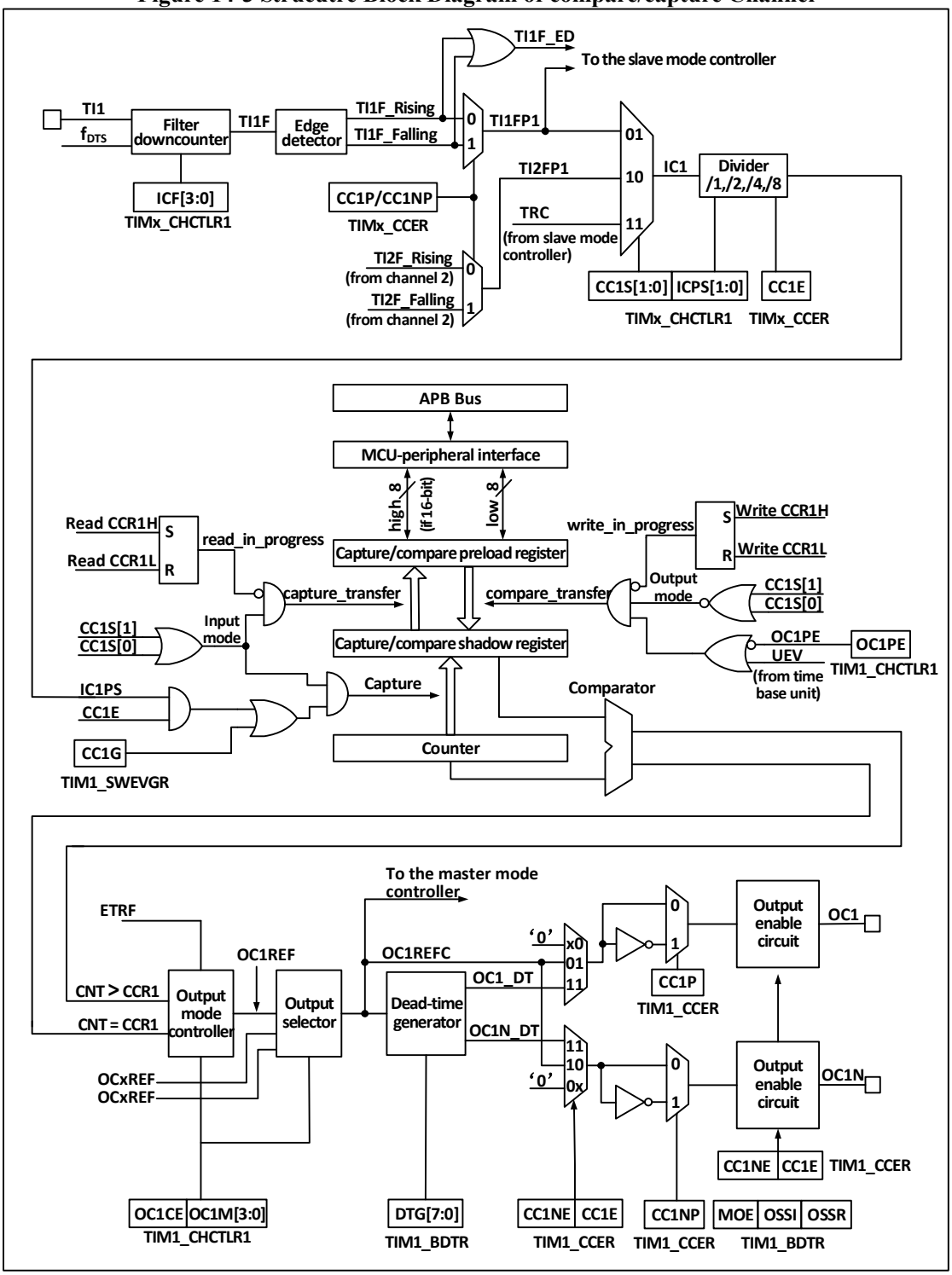
14.2.3 Counter and Periphery

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 partcomparator and output control.

Figure 14-3 Strucutre 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 two 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 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 Realization

The advanced-control timer complex functions are realized 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.
   So far, the comparison & capiture channel configuration has been 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 Comparison Output Mode

The comparison output mode is one of basic functions of timer. The principle of the comparison 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 comparison 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 comparison register;
5) Set the output mode, and set OCxM and CCxP bit;
6) Enable the output and set the CCxE bit;
7) Set the CEN bit and start the timer.

14.3.3 Forced Output Mode

The output mode of the compare/capture 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 method 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 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 two compare/capture channels, and the input polarity of the two 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 as 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 as 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

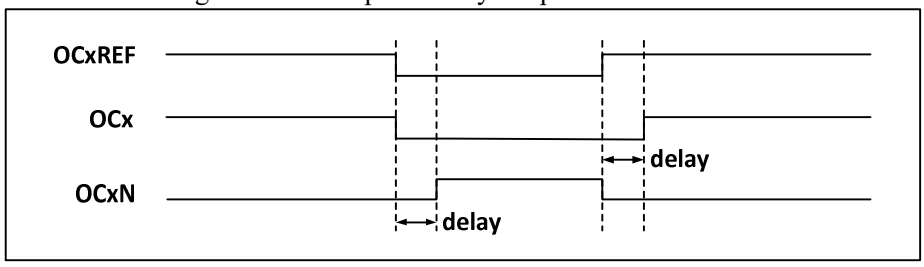
When the center-aligned mode is used, the core counter will run in a mode where up counting and down counting are performed alternately, and OCxREF performs rising and falling jumps when the values of the core counter and the compare/capture register are consistent. However, in three types of central 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 two output pins (compare/capture channel 4 has only one output pin), to output two 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 is 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:

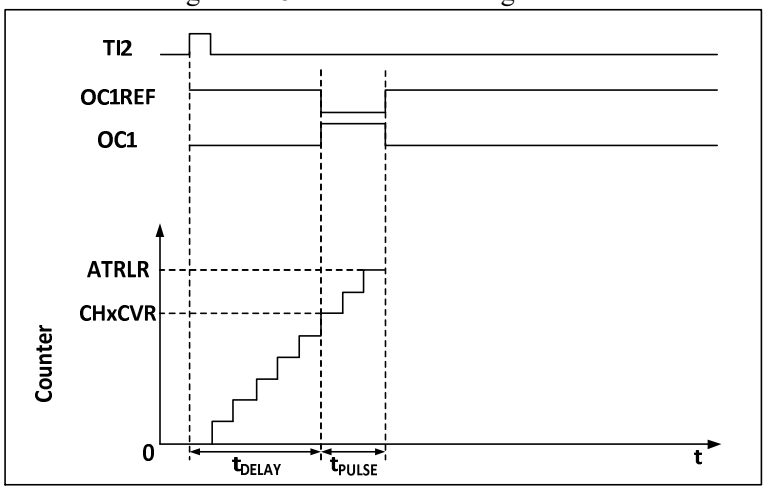
1) The MOE bit is asynchronously cleared, and the output is set to the invalid status, idle status or reset status according to the setting of the SOOI bit;
2) After MOE is cleared, each output channel will output the level determined by OSIx;
3) During the supplementary output: the output will be in 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 14-4, 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:

Figure 14-5 Generation of Single Pulse



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 substracting 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 effective 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 Synchronization 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.

14.3.11 Debug Mode

When the system enters the debug mode, the timer will continue to run or stop according to the setting of the DBG module.

14.4 Register Description

Table 14-2 List of TIM1 Related Registers

Name	Access address	Description	Reset value
R16_TIM1_CTLR1	0x40012C00	Control register 1	0x0000
R16_TIM1_CTLR2	0x40012C04	Control register 2	0x0000
R16_TIM1_SMCFGR	0x40012C08	Slave mode control register	0x0000
R16_TIM1_DMAINTENR	0x40012C0C	DMA/Interrupt enable register	0x0000
R16_TIM1_INTFR	0x40012C10	Interrupt status register	0x0000
R16_TIM1_SWEVGR	0x40012C14	Event generation register	0x0000
R16_TIM1_CHCTLR1	0x40012C18	Compare/ Capture control register 1	0x0000
R16_TIM1_CHCTLR2	0x40012C1C	Compare/ Capture control register 2	0x0000
R16_TIM1_CCER	0x40012C20	Compare/ Capture enable register	0x0000
R16_TIM1_CNT	0x40012C24	Counter	0x0000
R16_TIM1_PSC	0x40012C28	Timing clock prescaler	0x0000
R16_TIM1_ATRLR	0x40012C2C	Reload value register	0x0000
R16_TIM1_RPTCR	0x40012C30	Repeat count value register	0x0000
R16_TIM1_CH1CVR	0x40012C34	Compare/ Capature register 1	0x0000
R16_TIM1_CH2CVR	0x40012C38	Compare/ Capature register 2	0x0000
R16_TIM1_CH3CVR	0x40012C3C	Compare/ Capature register 3	0x0000
R16_TIM1_CH4CVR	0x40012C40	Compare/ Capature register 4	0x0000
R16_TIM1_BDTR	0x40012C44	Break and dead zone register	0x0000
R16_TIM1_DMACFGR	0x40012C48	DMA control register	0x0000
R16_TIM1_DMAADR	0x40012C4C	DMA address register in continuous mode	0x0000

Control register 1 (TIM1_CTLR1)

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	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 digitial filter (ETR,TIx): 00: Tdts=Tck_int 01: Tdts = 2 x Tck_int 10: Tdts = 4 x Tck_int 11: Reserved.	0
7	ARPE	RW	Automatic reload and preload enable bit: 1: Enable the automatic reload value register (ATRLR); 0: Disable the automatic reload value register (ATRLR).	0
[6:5]	CMS	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: 1: The counting mode of the counter is counting up; 0: The counting mode of the counter is counting down. *Note: When the counter is configured in the center lignment mode or encoder mode, this bit will be invalid.*	0

3	OPM	RW	Single pulse mode: 1: The counter will stop when the next update event (clearing the CEN bit) occurs. 0: The counter will 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 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	0
1	UDIS	RW	Update is disabled; the 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: Allowing 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	Enable counter. 1: Enable counter; 0: Disable counter. *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

Control register 2 (TIM1_CTLR2)

Offset address: 0x04

15	14	13	12	11	10	9	8	7	6 5 4	3	2	1	0
Reserved	OIS4	OSI3N	OIS3	OSI2N	OIS2	OSI1N	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=00 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	OSI3N	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	OSI2N	RW	Output idle status 2, refer to OIS3N.	0
10	OIS2	RW	Output idle status 2, refer to OIS4.	0
9	OSI1N	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	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 of 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: Comparison pulse-when a capture occurs or a comparison is successful, and the CC1IF flag is to be set	0

			(even if it is already high), the trigger output will send a positive pulse (TRGO); 100: Comparison-OC1REF signal is used as trigger output (TRGO); 101: Comparison-OC2REF signal is used as trigger output (TRGO); 110: Comparison-OC3REF signal is used as trigger output (TRGO); 111: Comparison-OC4REF signal is used as trigger output (TRGO).	
3	CCDS	RW	1: When an update event occurs, send a DMA request of CHxCVR; 0: When CHxCVR occurs, a DMA request of CHxCVR will be generated.	0
2	CCUS	RW	The 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

Salve mode control register (TIM1_SMCFGR)

Offset address: 0x08

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]			Reserved	SMS[2:0]		

Bit	Name	Access	Description	Reset value
15	ETP	RO	ETR trigger polarity selection; this bit selects whether to 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*	0

			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.*	
[13:12]	ETPS	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	RW	Trigger selection field; 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);	0

			110: Timer input 2 (TI12FP2) after filtering; 011: Internal trigger 3 (ITR3); 111: External trigger input (ETRF); The values will be changed only when SMS is 0.	
3	Reserved	RO	Reserved.	0
[2:0]	SMS	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

DMA/interrupt enable register (TIM1_DMAINTENR)

Offset address: 0x0C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE

Bit	Name	Access	Description	Reset value
15	Reserved	RO	Reserved.	0
14	TDE	RW	Trigger DMA request enable bit. 1: Enable to trigger DMA request; 0: Disable to trigger DMA request.	0
13	COMDE	RW	DMA request enable bit of COM. 1: Enable DMA request of COM; 0: Disable DMA request of COM.	0
12	CC4DE	RW	DMA request enable bit of compare/capture 4. 1: Enable DMA request of compare/capture 4;	0b

			0: Disable DMA request of compare/capture 4.	
11	CC3DE	RW	DMA request enable bit of compare/capture 3. 1: Enable DMA request of compare/capture 3; 0: Disable DMA request of compare/capture 3.	0
10	CC2DE	RW	DMA request enable bit of compare/capture 2. 1: Enable DMA request of compare/capture 2; 0: Disable DMA request of compare/capture 2.	0
9	CC1DE	RW	DMA request enable bit of compare/capture 1. 1: Enable DMA request of compare/capture 1; 0: Disable DMA request of compare/capture 1.	0
8	UDE	RW	Update DMA request enable bit. 1: Enable update DMA request enable bit. 0: Disable update DMA request enable bit.	0b
7	BIE	RW	Break interrupt enable bit. 1: Enable break interrupt; 0: Disable break interrupt.	0
6	TIE	RW	Trigger interrupt enable bit. 1: Enable trigger interrupt; 0: Disable trigger interrupt.	0
5	COMIE	RW	COM interrupt enable bit. 1: Enable COM interrupt; 0: Disable COM interrupt.	0
4	CC4IE	RW	Interrupt enable bit of compare/capture 4. 1: Enable interrupt of compare/capture 4; 0: Disable interrupt of compare/capture 4.	0
3	CC3IE	RW	Interrupt enable bit of compare/capture 3. 1: Enable interrupt of compare/capture 3; 0: Disable interrupt of compare/capture 3.	0
2	CC2IE	RW	Interrupt enable bit of compare/capture 2. 1: Enable interrupt of compare/capture 2; 0: Disable interrupt of compare/capture 2.	0
1	CC1IE	RW	Interrupt enable bit of compare/capture 1. 1: Enable interrupt of compare/capture 1; 0: Disable interrupt of compare/capture 1.	0
0	UIE	RW	Update interrupt enable bit. 1: Enable the update interrpt; 0: Disable the update interrpt;	0

Interrupt status register (TIM1_INTFR)

Offset address: 0x10

15 14 13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	CC4OF	CC3OF	CC2OF	CC1OF	Reserved	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF

Bit	Name	Access	Description	Reset value

[15:13]	Reserved	RO	Reserved.	0
12	CC4OF	RW0	Overcapture flag bit of compare/capture 4.	0
11	CC3OF	RW0	Overcapature 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 the hardware, write 0 by the software to clear the bit. 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 will be set by hardware and can be 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 the bit by the hardware and clear it 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 will be set by hardware and cleared by software. COM interrupt flag bit; once a COM event occurs, this bit will be 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 the 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

			0: No input capture is generated.	
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. The update event will generate 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

Event generation register (TIM1_SWEVGR)

Offset address: 0x14

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved.	0
7	BG	WO	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	The compare/capture event generates bit 4. Generating	0

			compare/capture event 4.	
3	CC3G	WO	The compare/capture event generates bit 3. Generating the compare/capture event 3.	0
2	CC2G	WO	The compare/capture event generates bit 2. Generating compare/capture event 2.	0
1	CC1G	WO	The compare/capture event generates bit 1. 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 will be set. If the corresponding interrupt and DMA are enabled, the corresponding interrupt and DMA will be 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 will be 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 centrosymmetric 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

Compare/Capture control register 1 (TIM1_CHCTLR1)

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]			

Comparison 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	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 the active level. Otherwise, it will be inactive; when counting down, once the core counter is greater than the value of the compare/capture register, channel 1 will be inactive level. Otherwise, it will be active level (OC1REF=1). 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 the inactive level. Otherwise, it will be an active level; when counting down, once the core counter is greater than the value of the compare/capture register, Channel 1 will be at the active level. Otherwise, it will be at the inactive level. *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 compareison capture register. Read and write operations are only made on the	0

			preload register. The preload value of the compare/capture register 1 is loaded into the current 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	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	RW	Mode setting 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	RW	Input selection of compare/capture 1.	0

Capture mode (pin direction is input):

Bit	Name	Access	Description	Reset value

[15:12]	IC2F	RW	Input capture 2 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	RW	Compare/capture 2 prescaler configuration; 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	RW	Compare/capture 2 input selection. 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	RW	Input capture 1 filter configuration.	0
[3:2]	IC1PSC	RW	Prescale configuration of compare/capture 1.	0
[1:0]	CC1S	RW	Input selection of compare/capture 1.	0

Compare/Capture control register 2 (TIM1_CHCTLR 2)

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]			

Comparison 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	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	RW	Input selection of compare/capture 4.	0
7	OC3CE	RW	Clear enable bit of compare/capture 3.	0
[6:4]	OC3M	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	RW	Input selection of compare/capture 3.	0

Capture mode (pin direction is input):

Bit	Name	Access	Description	Reset value
[15:12]	IC4F	RW	Input capture 4 filter configuration.	0
[11:10]	IC4PSC	RW	Prescale configuration of compare/capture 4.	0
[9:8]	CC4S	RW	Input selection of compare/capture 4.	0
[7:4]	IC3F	RW	Input capture 3 filter configuration.	0
[3:2]	IC3PSC	RW	Prescale configuration of compare/capture 3.	0
[1:0]	CC3S	RW	Input selection of compare/capture 3.	0

Compare/ Capture enable register (TIM1_CCER)

Offset address: 0x20

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E

Bit	Name	Access	Description	Reset value
[15:14]	Reserved	RO	Reserved.	0
13	CC4P	RW	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	0

			compare/capture 3.	
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.	0
0	CC1E	RW	Output enable bit of compare/capture 1.	0

Counter of Advanced-control timer (TIM1_CNT)

Offset address: 0x24

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

Bit	Name	Access	Description	Reset value
[15:0]	CNT	RW	Real-time value of timer counter.	0

Count clock prescaler (TIM1_PSC)

Offset address: 0x28

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

Bit	Name	Access	Description	Reset value
[15:0]	PSC	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

Automatic reload value register (TIM1_ATRLR)

Offset address: 0x2C

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

Bit	Name	Access	Description	Reset value
[15:0]	ATRLR	RW	The value of this domain will be loaded into the counter.	0

			Please read 14.2.3 for ATRLR acting and update time; when ATRLR is empty, the counter will stop.	

Recounting value register (TIM1_RPTCR)

Offset address: 0x30

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

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved.	0
[7:0]	RPTCR	RW	Repeated counter value.	0

Compare/Capture register 1 (TIM1_CH1CVR)

Offset address: 0x34

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1CVR[15:0]															

Bit	Name	Access	Description	Reset value
[15:0]	CH1CVR	RW	Value of compare/capture register channel 1.	0

Compare/Capture register 2 (TIM1_CH2CVR)

Offset address: 0x38

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH2CVR[15:0]															

Bit	Name	Access	Description	Reset value
[15:0]	CH2CVR	RW	Value of compare/capture register channel 2.	0

Compare/Capture register 3 (TIM1_CH3CVR)

Offset address: 0x3C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3CVR [15:0]															

Bit	Name	Access	Description	Reset value
[15:0]	CH3CVR	RW	Value of compare/capture register channel 3.	0

Compare/Capture register 4 (TIM1_CH4CVR)

Offset address: 0x40

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH4CVR [15:0]															

Bit	Name	Access	Description	Reset value
[15:0]	CH4CVR	RW	Value of compare/capture register channel 4.	0

Break and dead zone register (TIM1_BDTR)

Offset address: 0x44

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							

Bit	Name	Access	Description	Reset value
15	MOE	RW	Main output enable bit. Once 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,	0

			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.*	
[9:8]	LOCK	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 2, nor can you write the CC control bit. *Note: After the system is reset, the LOCK bit can only be written once, and cannot be modified again until reset.*	0
[7:0]	DTG	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

DMA control register (TIM1_DMACFGR)

Offset address: 0x48

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

Bit	Name	Access	Description	Reset value
[15:13]	Reserved	RO	Reserved.	0
[12:8]	DBL	RW	Length of DMA continuous transfer; the actual value is the value of this domain + 1.	0
[7:5]	Reserved	RO	Reserved.	0
[4:0]	DBA	RW	These bits define the offset of DMA from the address of control register 1 in continuous mode.	0

DMA address register in continuous mode (TIM1_DMAADR)

Offset address: 0x4C

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

Bit	Name	Access	Description	Reset value
[15:0]	DMAADR	RW	DMA address in continuous mode.	0

Chapter 15 General-purpose Timer (TIM2/3/4)

The module description in this chapter applies to the full range of CH32F103 and CH32V103 microcontrollers.

The general-purpose timer module contains a 16-bit timer 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.

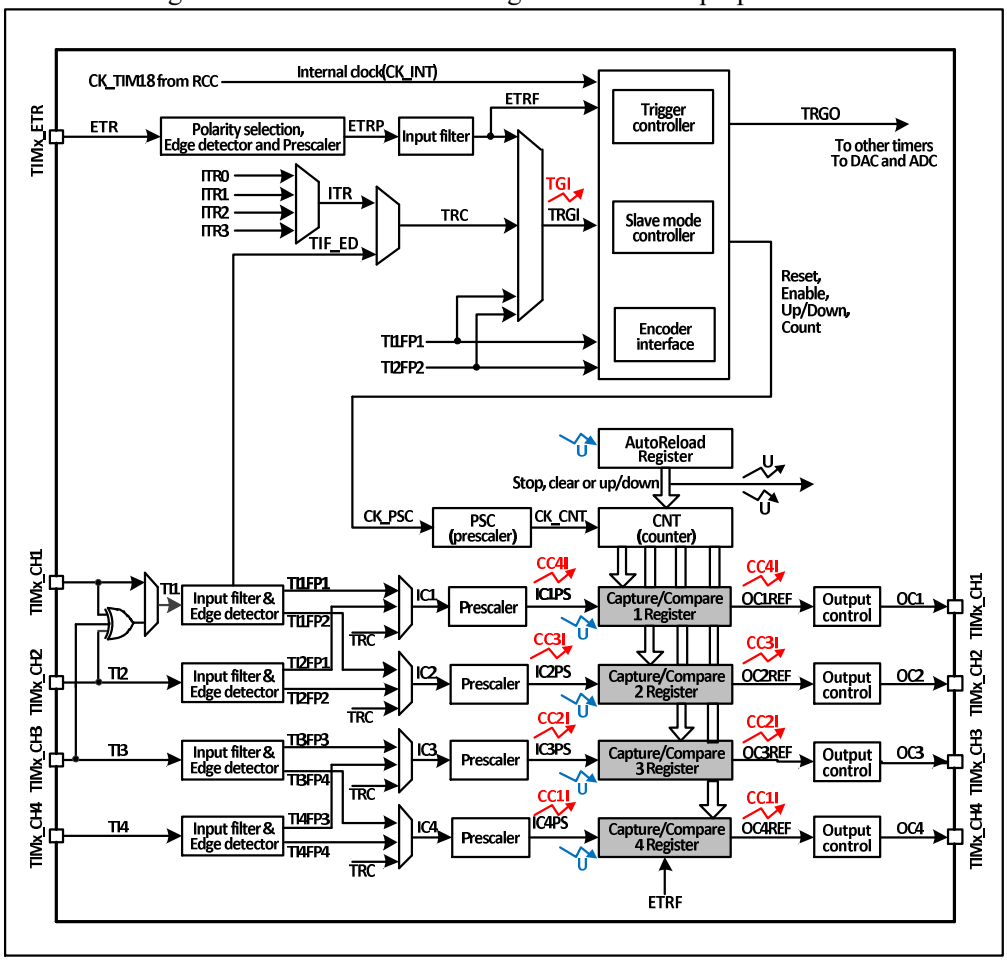
15.1 Main Features

The main features of general-purpose timer include:

- 16-bit automatic reload counter, support upcounting mode, downcounting mode and up/down-counting mode
- 16-bit prescaler; the frequency division coefficient is dynamically adjustable from 1 to 65536
- It supports four independent compare/captures
- Each compare/capture supports multiple working modes, such as: input capture, output comparison, PWM generation and single pulse output
- Supports external signal control timer
- Supports the use of DMA in multiple modes
- Supports the incremental code, cascade connection and synchronization between timers

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 three parts: Input clock part, core counter part and compare/capture part.

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 counter part. 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 four 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

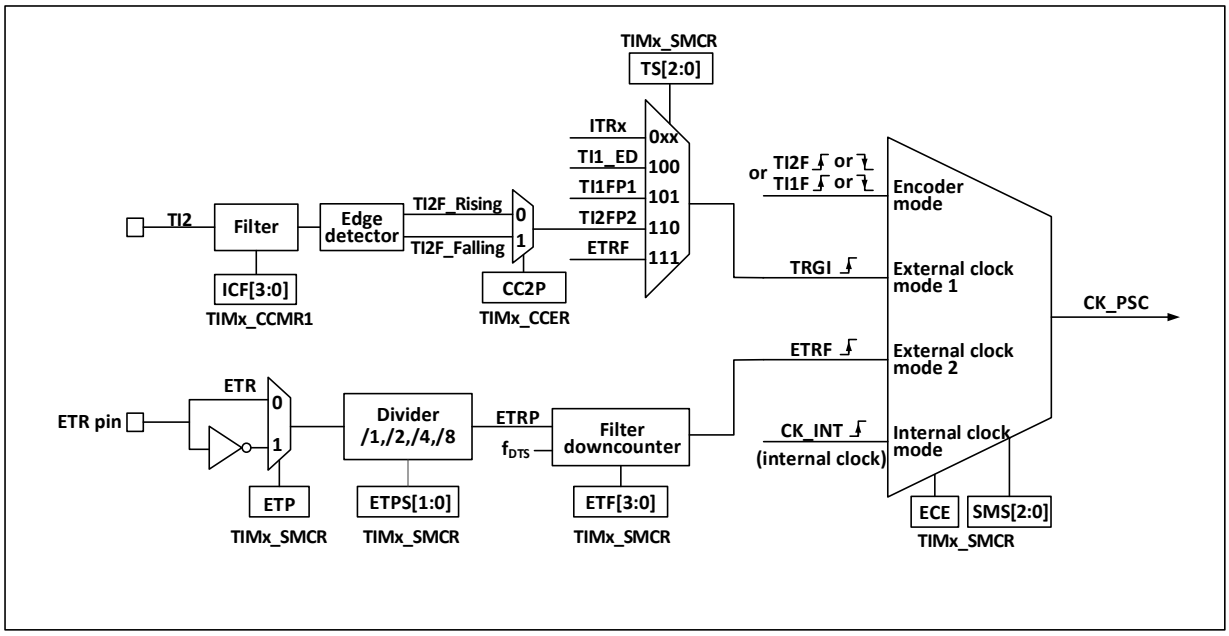
Compared with the advanced-control timer, the general-purpose timer is lack of the following functions:

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 APB2, while the CK_INT of the advanced-control timer (TIM1) comes from APB1.

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 Source Block Diagram of General-purpose timer



The optional input clock can be divided into 4 categories:

1) The route of external clock pin (ETR) input: ETR→ETRP→ETRF;
2) Internal APB clock input route: CK_INT;
3) The route from the compare/capture 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 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 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.

15.2.3.2 External Clock Mode 1

If SMS is set to 111b, the external clock source mode 1 will be enabled. When external clock source 1 is enabled, TRGI will be 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 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;
4) Signal ETRF from external clock pin.

15.2.2.3 External Clock Source Mode 2

Use external trigger mode 2 to count on every rising or falling edge of the external clock pin input. When the ECE bit is set, the external clock source mode 2 will be used. When the external clock source mode 2 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.

15.2.3.4 Encoder Mode

Set SMS as 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.

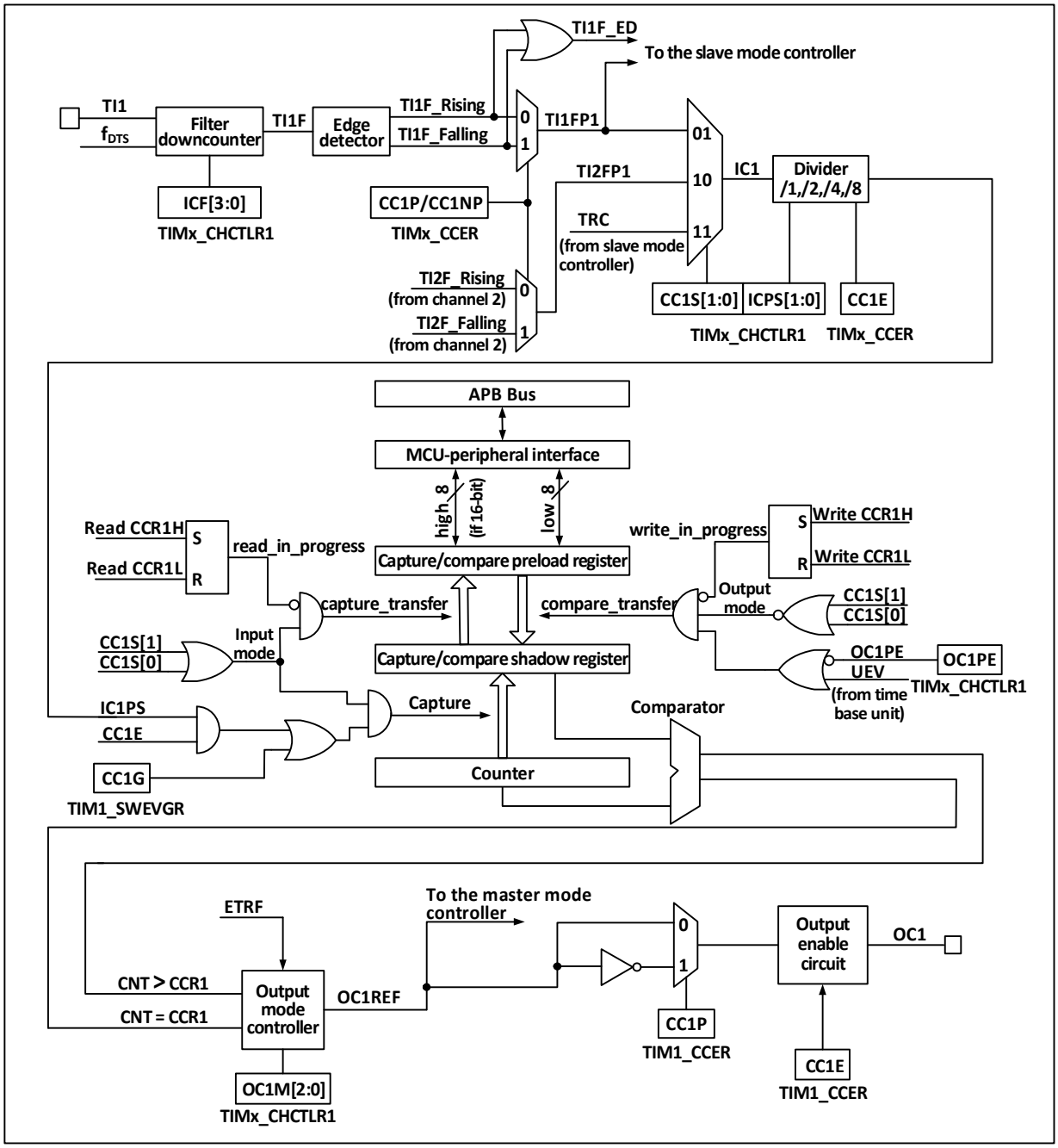
15.2.4 Counter and Periphery

CK_PSC inputs to the prescaler (PSC) for frequency division. PSC 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.

15.2.5 Comparison 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 partcomparator and output control. The block diagram of the compare/capture is as shown in Figure 15-3.

Figure 15-3 Strucutre Block Diagram of Compare/capture



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 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 two 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 Realization

The general-purpose timer complex functions are realized 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. 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 out put 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.
So far, the comparison & capiture channel configuration has been 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 bit, 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 comparison 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 method 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 comparison 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 two compare/captures, and the input polarity of the two 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 as 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 as 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.

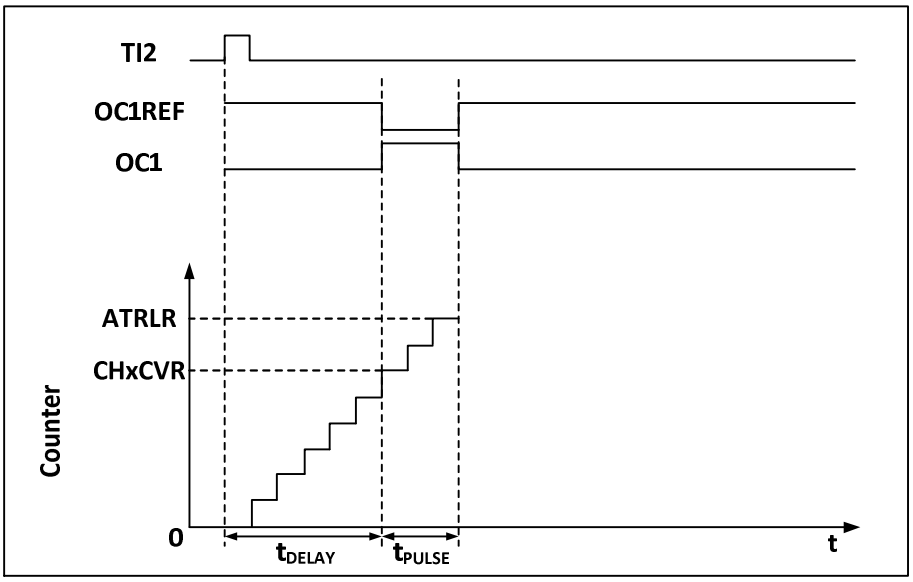
- Central alignment

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 three 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.

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 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 substracting 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 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 15-1 Relationship between Counting Direction of Timer Encoder Mode and Encoder Signal

Counting effective 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

15.3.8 Timer Synchronization 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.

15.3.9 Debug Mode

When the system enters debug mode, the timer will continue to run or stop according to the setting of the DBG module.

15.4 Register Description

Table 15-2 List of TIM2 Related Registers

Name	Offset address	Description	Reset value
R16_TIM2_CTLR1	0x40000000	TIM2 control register 1	0x0000
R16_TIM2_CTLR2	0x40000004	TIM2 control register 2	0x0000
R16_TIM2_SMCFGR	0x40000008	TIM2 slave mode control register	0x0000
R16_TIM2_DMAINTENR	0x4000000C	TIM2 DMA/Interrupt enable register	0x0000
R16_TIM2_INTFR	0x40000010	TIM2 interrupt state register	0x0000
R16_TIM2_SWEVGR	0x40000014	TIM2 event generation register	0x0000
R16_TIM2_CHCTLR1	0x40000018	TIM2 compare/capture control register 1	0x0000
R16_TIM2_CHCTLR2	0x4000001C	TIM2 compare/capture control register 2	0x0000
R16_TIM2_CCER	0x40000020	TIM2 compare/ capture enable register	0x0000
R16_TIM2_CNT	0x40000024	TIM2 counter	0x0000
R16_TIM2_PSC	0x40000028	TIM2 timing clock prescaler	0x0000
R16_TIM2_ATRLR	0x4000002C	TIM2 automatic reload value register	0x0000
R16_TIM2_CH1CVR	0x40000034	TIM2 compare/capture register 1	0x0000
R16_TIM2_CH2CVR	0x40000038	TIM2 comparie/capture register 2	0x0000
R16_TIM2_CH3CVR	0x4000003C	TIM2 compare/capture register 3	0x0000
R16_TIM2_CH4CVR	0x40000040	TIM2 compare/capture register 4	0x0000
R16_TIM2_DMACFGR	0x40000048	TIM2 DMA control register	0x0000
R16_TIM2_DMAADR	0x4000004C	DMA address register of TIM2 continuous mode	0x0000

Table 15-3 List of TIM3 Related Registers

Name	Offset address	Description	Reset value
R16_TIM3_CTLR1	0x40000400	TIM3 control register 1	0x0000
R16_TIM3_CTLR2	0x40000404	TIM3 control register 2	0x0000
R16_TIM3_SMCFGR	0x40000408	TIM3 slave mode control register	0x0000
R16_TIM3_DMAINTENR	0x4000040C	TIM3 DMA/Interrupt enable register	0x0000
R16_TIM3_INTFR	0x40000410	TIM3 interrupt state register	0x0000
R16_TIM3_SWEVGR	0x40000414	TIM3 event generation register	0x0000
R16_TIM3_CHCTLR1	0x40000418	TIM3 compare/capture control register 1	0x0000
R16_TIM3_CHCTLR2	0x4000041C	TIM3 compare/capture control register 2	0x0000
R16_TIM3_CCER	0x40000420	TIM3 compare/ capture enable register	0x0000
R16_TIM3_CNT	0x40000424	TIM3 counter	0x0000
R16_TIM3_PSC	0x40000428	TIM3 timing clock prescaler	0x0000
R16_TIM3_ATRLR	0x4000042C	TIM3 automatic reload value register	0x0000
R16_TIM3_CH1CVR	0x40000434	TIM3 compare/capture register 1	0x0000
R16_TIM3_CH2CVR	0x40000438	TIM3 compare/capture register 2	0x0000
R16_TIM3_CH3CVR	0x4000043C	TIM3 compare/capture register 3	0x0000
R16_TIM3_CH4CVR	0x40000440	TIM3 compare/capture register 4	0x0000
R16_TIM3_DMACFGR	0x40000448	TIM3 DMA control register	0x0000
R16_TIM3_DMAADR	0x4000044C	DMA address register of TIM3 continuous mode	0x0000

Table 15-4 List of TIM4 Related Registers

Name	Offset address	Description	Reset value
R16_TIM4_CTLR1	0x40000800	TIM4 control register 1	0x0000
R16_TIM4_CTLR2	0x40000804	TIM4 control register 2	0x0000
R16_TIM4_SMCFGR	0x40000808	TIM4 slave mode control register	0x0000
R16_TIM4_DMAINTENR	0x4000080C	TIM4 DMA/Interrupt enable register	0x0000
R16_TIM4_INTFR	0x40000810	TIM4 interrupt state register	0x0000
R16_TIM4_SWEVGR	0x40000814	TIM4 event generation register	0x0000
R16_TIM4_CHCTLR1	0x40000818	TIM4 compare/capture control register 1	0x0000
R16_TIM4_CHCTLR2	0x4000081C	TIM4 compare/capture control register 2	0x0000
R16_TIM4_CCER	0x40000820	TIM4 compare/capture enable register	0x0000
R16_TIM4_CNT	0x40000824	TIM4 counter	0x0000
R16_TIM4_PSC	0x40000828	TIM4 timing clock prescaler	0x0000
R16_TIM4_ATRLR	0x4000082C	TIM4 automatic reload value register	0x0000
R16_TIM4_CH1CVR	0x40000834	TIM4 compare/capture register 1	0x0000
R16_TIM4_CH2CVR	0x40000838	TIM4 compare/capture register 2	0x0000
R16_TIM4_CH3CVR	0x4000083C	TIM4 compare/capture register 3	0x0000
R16_TIM4_CH4CVR	0x40000840	TIM4 compare/capture register 4	0x0000
R16_TIM4_DMACFGR	0x40000848	TIM4 DMA control register	0x0000
R16_TIM4_DMAADR	0x4000084C	DMA address register of TIM4 continuous mode	0x0000

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	RW	These 2 bits define the frequency division ratio of timer clock (CK_INT) frequency and sampling clock used for the digitial filter: 00: Tdts=Tck_int; 01: Tdts= 2xTck_int; 10: Tdts= 4xTck_int; 11: Reserved.	0
7	ARPE	RW	Automatic reload and preload enable bit: 1: Enable the automatic reload value register (ATRLR); 0: Disable the automatic reload value register (ATRLR).	0
[6:5]	CMS	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: 1: The counting mode of the counter is counting up; 0: The counting mode of the counter is counting down. *Note: When the counter is configured in the center lignment mode or encoder mode, this bit will be invalid.*	0
3	OPM	RW	Single pulse mode.	0

			1: The counter will stop when the next update event (clearing the CEN bit) occurs; 0: The counter will not stop when the next update event occurs.	
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 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	0
1	UDIS	RW	Update is disabled; the 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: Allowing 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	Enable counter. 1: Enable counter; 0: Disable counter. *Note: After the software sets the CEN bit, the external clock, gating mode and encoder mode can only work. The trigger mode can automatically set the CEN bit by hardware.*	0

Control register 2 (TIMx_CTLR2) (x=2/3/4)

Offset address: 0x04

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TI1S	MMS[2:0]			CCDS	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	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 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

Slave mode control register (TIMx_SMCFGR) (x=2/3/4)

Offset address: 0x08

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]			Reserved	SMS[2:0]		

Bit	Name	Access	Description	Reset value
15	ETP	RO	ETR trigger polarity selection; this bit selects whether to directly input ETR or input inverted ETR. 1: ETR inverted, active at low level or falling edge; 0: ETR, active at high level or rising edge.	0
14	ECE	RW	External clock mode 2 enable 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 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	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 is divided by 8.	0
[11:8]	ETF	RW	External trigger filter. In fact, the digital filter is an event counter. N events are needed to validate a transition on the output. 0000: No filter, sampling is done at Fdts; 0001: Fsampling=Fck_int, N=2; 0010: Fsampling=Fck_int，N=4; 0011: Fsampling=Fck_int, N=8; 0100: Fsampling=Fdts/2, N=6; 0101: Fsampling=Fdts/2, N=8;	0

			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	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; depending on TI1FP1 level, the core counter counts up or down on edge of TI2FP2; 010: Encoder mode 2; depending on TI2FP2 level, the core counter counts up or down on edge of TI1FP1; 011: Encoder mode 3; depending on the input level of another signal, the core counter counts up and down on the edge of TI1FP1 and TI2FP2; 100: Reset mode; the rising edge of the trigger input (TRGI) will initialize the counter and generate a signal for updating the register; 101: Gating mode; when the trigger input (TRGI) is high, the clock of the counter will be turned on; when the trigger input becomes low, the counter will stop, and the start and stop of the counter will be controlled;	0

			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.	

DMA/interrupt enable register (TIMx_DMAINTENR) (x=2/3/4)

Offset address: 0x0C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TDE	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 bit. 1: Enable to trigger DMA request; 0: Disable to trigger DMA request.	0
13	COMDE	RW	DMA request enable bit of COM. 1: Enable DMA request of COM; 0: Disable DMA request of COM.	0
12	CC4DE	RW	DMA request enable bit of compare/capture 4. 1: Enable DMA request of compare/capture 4; 0: Disable DMA request of compare/capture 4.	0
11	CC3DE	RW	DMA request enable bit of compare/capture 3. 1: Enable DMA request of compare/capture 3; 0: Disable DMA request of compare/capture 3.	0
10	CC2DE	RW	DMA request enable bit of compare/capture 2. 1: Enable DMA request of compare/capture 2; 0: Disable DMA request of compare/capture 2.	0
9	CC1DE	RW	DMA request enable bit of compare/capture 1. 1: Enable DMA request of compare/capture 1; 0: Disable DMA request of compare/capture 1.	0
8	UDE	RW	Updated DMA request enable bit. 1: Enable updated DMA request 0: Disable updated DMA request	0
7	Reserved	RO	Reserved.	0
6	TIE	RW	Trigger interrupt enable bit. 1: Enable trigger interrupt; 0: Disable trigger interrupt.	0
5	Reserved	RO	Reserved.	0
4	CC4IE	RW	Interrupt enable bit of compare/capture 4. 1: Enable interrupt of compare/capture 4; 0: Disable interrupt of compare/capture 4.	0
3	CC3IE	RW	Interrupt enable bit of compare/capture 3.	0

			1: Enable interrupt of compare/capture 3; 0: Disable interrupt of compare/capture 3.	
2	CC2IE	RW	Interrupt enable bit of compare/capture 2. 1: Enable interrupt of compare/capture 2; 0: Disable interrupt of compare/capture 2.	0
1	CC1IE	RW	Interrupt enable bit of compare/capture 1. 1: Enable interrupt of compare/capture 1; 0: Disable interrupt of compare/capture 1.	0
0	UIE	RW	Update interrupt enable bit. 1: Enable update interrpt; 0: Disable update interrpt;	0

Interrupt status register (R16_TIMx_INTFR) (x=2/3/4)

Offset address: 0x10

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CC4OF	CC3OF	CC2OF	CC1OF	Reserved		TIF	Reserved	CC4IF	CC3IF	CC2IF	CC1IF	UIF

Bit	Name	Access	Description	Reset value
[15:13]	Reserved	RO	Reserved.	0
12	CC4OF	W0	Overcapture flag bit of compare/capture 4.	0
11	CC3OF	W0	Overcapature flag bit of compare/capture 3.	0
10	CC2OF	W0	Overcapture flag bit of compare/capture 2.	0
9	CC1OF	W0	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 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 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	Reserved	RO	Reserved.	0
4	CC4IF	W0	Interrupt flag bit of compare/capture 4.	0
3	CC3IF	W0	Interrupt flag bit of compare/capture 3.	0
2	CC2IF	W0	Interrupt flag bit of compare/capture 2.	0
1	CC1IF	W0	Interrupt flag bit of compare/capture 1.	0

			If the compare/capture is configured as the output mode, this bit will be 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	UIF	W0	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. The update event will generate 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

Event generation register (TIMx_SWEVGR) (x=2/3/4)

Offset address: 0x14

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

			corresponding interrupt and DMA are enabled, the corresponding interrupt and DMA will be generated; 0: No effect.	
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/capture 1 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 sffect.	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 the prescaler is also cleared, but the prescaler factor remains unchanged. In centrosymmetric 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

Compare/Capture control register 1 (TIMx_CHCTLR1) (x=2/3/4)

Offset address: 0x18

The channel can be used for input (capture mode) or output (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	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 active 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 the active level. Otherwise, it will be inactive; when counting down, once the core counter is greater than the value of the compare/capture register, channel 1 will be inactive level. Otherwise, it will be active level (OC1REF=1). 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 the inactive level. Otherwise, it will be an active level; when counting	0

			down, once the core counter is greater than the value of the compare/capture register, Channel 1 will be at the active level. Otherwise, it will be at the invalid level. *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 the comparison result changes or when switching from freezing mode to PWM mode in the output comparison mode.	
11	OC2PE	RW	Preload enable bit of compare/capture register 2. 1: Enable the preload function of the compare/capture register. Read and write operations are only made on the preload register. The preload value of the compare/capture register 1 is loaded into the current shadow register when the update event arrives; 0: Disable the pre-loading function of compare/capture register 2. You can write 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=00b, this bit cannot be modified; only in single pulse mode (OPM=1) you can use PWM mode without confirming the preload register; otherwise its action is uncertain.*	0
10	OC2FE	RW	Compare/capture 2 fast enable 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	RW	Input selection of compare/capture 2. 00: The 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;	0

			11: Compare/capture 2 is configured as an input, and IC2 is mapped on TRC. This mode only works when the internal trigger input is selected (selected by the TS bit). *Note: Compare/capture 2 is only writable when the channel is switched off (CC2E is zero).*	
7	OC1CE	RW	Compare/capture 1 clear enable bit.	0
[6:4]	OC1M	RW	Mode setting 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	RW	Input selection of compare/capture 1.	0

Capture mode (pin direction is input):

Bit	Name	Access	Description	Reset value
[15:12]	IC2F	RW	Input capture 2 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 calidate 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	RW	Compare/capture 2 prescaler configuration; these 2 bits define the prescaler factor of compare/capture 2. Once CC1E=0, the prescaler will be reset. 00: Prescaler OFF, each edge detected on the capture input port triggers a capture; 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	RW	Compare/capture 2 input selection. These 2 bits define the direction of the channel (input/output) and selection of input pins.	0

			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).*	
[7:4]	IC1F	RW	Input capture 1 filter configuration.	0
[3:2]	IC1PSC	RW	Prescale configuration of compare/capture 1.	0
[1:0]	CC1S	RW	Input selection of compare/capture 1.	0

Compare/Capture control register 2 (TIMx_CHCTLR2) (x=2/3/4)

Offset address: 0x1C

The channel can be used for input (capture mode) or output (comparison mode), and the direction of the channel is defined by the corresponding CCxS bit. The functions of other bits of this register are different in input and output modes. OCxx describes the function of the channel in output mode, and ICxx describes the function of the channel in input mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M[2:0]			OC4PE	OC4FE	CC4S[1:0]		OC3CE	OC3M[2:0]			OC3PE	OC3FE	CC3S[1:0]	
IC4F[3:0]				IC4PSC[1:0]				IC3F[3:0]				IC3PSC[1:0]			

Comparison 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	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	RW	Input selection of compare/capture 4.	0
7	OC3CE	RW	Clear enable bit of compare/capture 3.	0
[6:4]	OC3M	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	RW	Input selection of compare/capture 3.	0

Capture mode (pin direction is input):

Bit	Name	Access	Description	Reset value
[15:12]	IC4F	RW	Input capture 4 filter configuration.	0
[11:10]	IC4PSC	RW	Prescale configuration of compare/capture 4.	0
[9:8]	CC4S	RW	Input selection of compare/capture 4.	0
[7:4]	IC3F	RW	Input capture 3 filter configuration.	0

[3:2]	IC3PSC	RW	Prescale configuration of compare/capture 3.	0
[1:0]	CC3S	RW	Input selection of compare/capture 3.	0

Compare/Capture enable register (TIMx_CCER）(x=2/3/4)

Offset address: 0x20

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		CC4P	CC4E	Reserved		CC3P	CC3E	Reserved		CC2P	CC2E	Reserved		CC1P	CC1E

Bit	Name	Access	Description	Reset value
[15:14]	Reserved	RO	Reserved.	0
13	CC4P	RW	Output polarity setting bit of compare/capture 4.	0
12	CC4E	RW	Output enable bit of compare/capture 4.	0
[11:10]	Reserved	RO	Reserved.	0
9	CC3P	RW	Output polarity setting bit of compare/capture 3.	0
8	CC3E	RW	Output enable bit of compare/capture 3.	0
[7:6]	CC2NP	RO	Reserved.	0
5	CC2P	RW	Output polarity setting bit of compare/capture 2.	0
4	CC2E	RW	Output enable bit of compare/capture 2.	0
[3:2]	Reserved	RO	Reserved.	0
1	CC1P	RW	Output polarity setting bit of compare/capture 1.	0
0	CC1E	RW	Output enable bit of compare/capture 1.	0

Counter of general-purpose timer (TIMx_CNT) (x=2/3/4)

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	RW	Real-time value of timer counter.	0

Counting clock prescaler (TIMx_PSC) (x=2/3/4)

Offset address: 0x28

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

Bit	Name	Access	Description	Reset value
[15:0]	PSC	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

Automatic reload value register (TIMx_ATRLR) (x=2/3/4)

Offset address: 0x2C

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

Bit	Name	Access	Description	Reset value
[15:0]	ATRLR	RW	The value of ATRLR[15:0] will be loaded into the counter. Please refer to Section 14.2.4 for ATRLR acting and update time; when ATRLR is empty, the counter will stop.	0

Compare/Capture control register 1 (TIMx_CH1CVR) (x=2/3/4)

Offset address: 0x34

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1CVR[15:0]															

Bit	Name	Access	Description	Reset value
[15:0]	CH1CVR	RW	Value of compare/capture 1.	0

Compare/Capture control register 2 (TIMx_CH2CVR) (x=2/3/4)

Offset address: 0x38

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH2CVR[15:0]															

Bit	Name	Access	Description	Reset value
[15:0]	CH2CVR	RW	Value of compare/capture 2.	0

Compare/Capture control register 3 (TIMx_CH3CVR) (x=2/3/4)

Offset address: 0x3C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3CVR[15:0]															

Bit	Name	Access	Description	Reset value
[15:0]	CH3CVR	RW	Value of compare/capture 3.	0

Compare/Capture control register 4 (TIMx_CH4CVR) (x=2/3/4)

Offset address: 0x40

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH4CVR[15:0]															

Bit	Name	Access	Description	Reset value
[15:0]	CH4CVR	RW	Value of compare/capture 4.	0

DMA control register (TIMx_DMACFGR) (x=2/3/4)

Offset address: 0x48

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

Bit	Name	Access	Description	Reset value
[15:13]	Reserved	RO	Reserved.	0
[12:8]	DBL	RW	Length of DMA continuous transfer; the actual value is the value of this domain + 1.	0
[7:5]	Reserved	RO	Reserved.	0
[4:0]	DBA	RW	These bits define the offset of DMA from the address of control register 1 in continuous mode.	0

DMA address register in continuous mode (TIMx_DMAADR) (x=2/3/4)

Offset address: 0x4C

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

Bit	Name	Access	Description	Reset value
[15:0]	DMAADR	RW	DMA address in continuous mode.	0

Chapter 16 Digital-to-Analog Converter (DAC)

The module description in this chapter applies to the full range of CH32F103 microcontrollers.

Digital-to-analog converter (DAC), including a 12-bit digital input converter to convert 2-channel analog voltage outputs. Built-in triangle wave and noise waveform generator, supporting a variety of event trigger conversion, DMA function, etc.

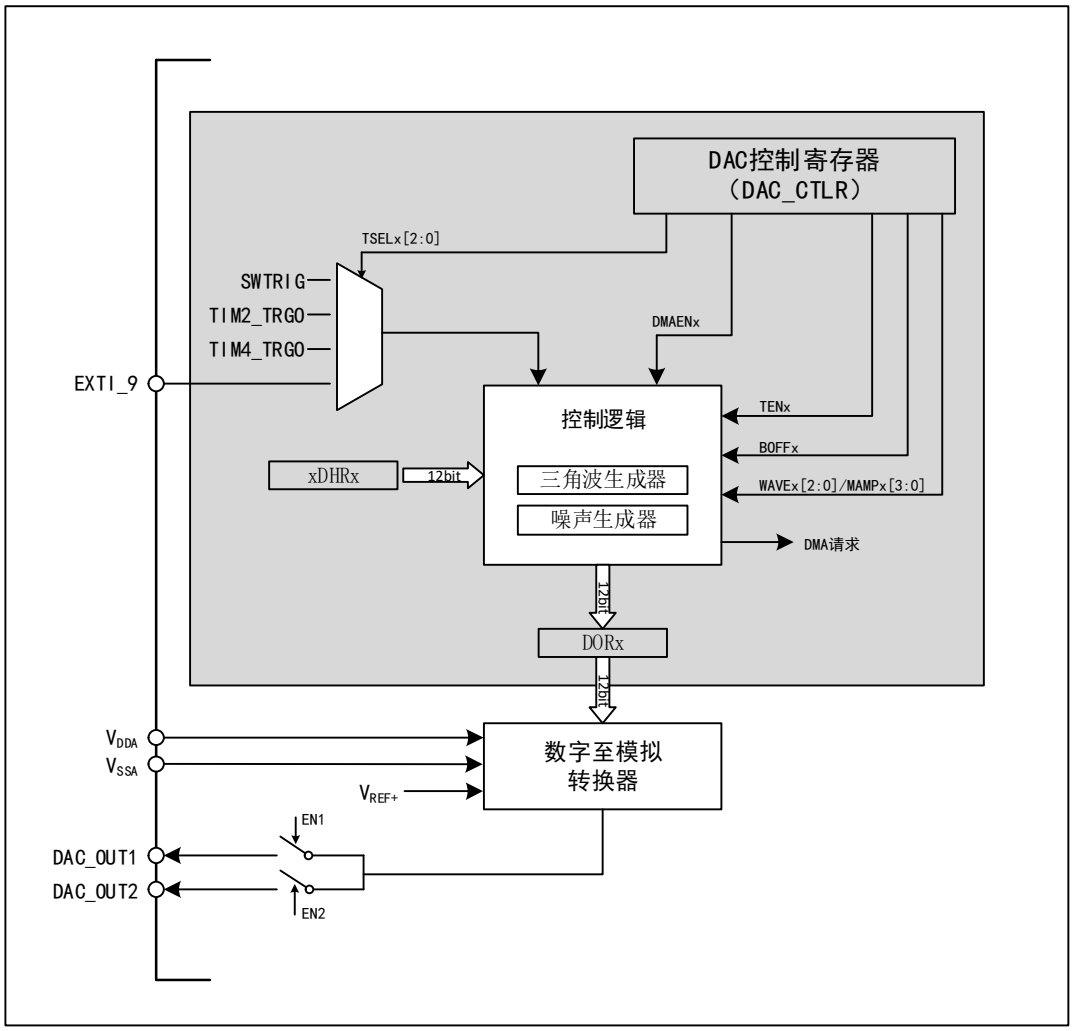
16.1 Main Features

- 1 DAC converter corresponds to 2 channels of monotonic output
- Triangular-wave, noise-wave generation
- Left or right data alignment in 12-bit mode
- Support DMA function
- Multiple trigger events

16.2 Functional Description

16.2.1 DAC Structure

Figure 16-1 DAC Block Diagram



16.2.2 DAC Channel Configuration

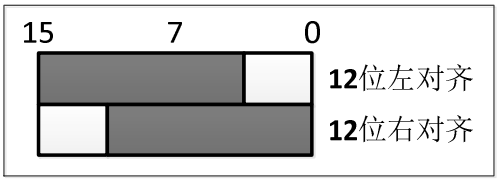
1) Enable DAC function: Set the ENx bit of the DAC_CTLR register to 1, to turn on the analog power to DAC channel x. After a period of start-up time, DAC channel x will be enabled. The DAC contains 2 analog output channels. When the outputs of the 2 channels are enabled at the same time (EN1 bit 1 and EN2 1), the same waveform will be outputted to the 2 analog channels in the configuration of channel 1; when only output of the channel 1 is enabled, the waveform will be outputted to the analog channel 1 according to the channel 1 configuration, and the channel 2 will not output; when only the output of the channel 2 is enabled, the output waveform will be outputted to the analog channel 2 according to the channel 2 configuration, and channel 1 will not output.

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.

2) Enable output buffer: DAC integrates the output buffer, which can be used to reduce the output impedance and increase the driving capacity to directly drive the external load. Each DAC channel output buffer can be enabled or disabled by setting the BOFFx bit of DAC_CTLR register.

3) Data format: Including 12-bit data left alignment and 12-bit data right alignment. Write data to DAC_R12BDHRx[11:0], the module will load (after 1 APB1 clock cycle) right-aligned data to the data output register DAC_DORx[11:0]; write data to DAC_L12BDHRx[15:3], the module will go through the corresponding shift to load the left-aligned data (after 1 APB1 clock cycle) to the data output register DAC_DORx[11:0].

Figure 16-2 Data Format



4) DMA function: DAC channel has the DMA function. Set the DMAENx bit of the DAC_CTLR register to 1, to enable the DMA function of the corresponding channel. When a trigger event (excluding software trigger) occurs, a DMA request will be generated, and then the data in DAC_DORx register will be updated.

5) Trigger event selection: DAC conversion can be triggered by the following 4 events. When the TENx bit of the DAC_CTLR register is configured as 1, the TSELx[2:0] control bits will be configured to select one of the four trigger events to trigger the DAC conversion.

Table 16-1 Trigger Events

Trigger source	Type	TSELx[2:0]
TRGO event of timer 3	Internal signal from on-chip timer	001
TRGO event of timer 2		100
TRGO event of timer 4		101
EXTI line 9	External pin	110
SWTRIG (software trigger)	Software control bit	111

The DAC interface will monitor the rising edge from the selected timer TRGO output or external interrupt line 9, and update the register DAC_DORx to the new value after 3 APB1 clock cycles following the trigger.

If the software trigger mode is configured, once the SWTRIG bit is set to 1, a conversion will be started. After 1 APB1 clock cycle following the trigger, the register DAC_DORx will be updated to the new value, and the SWTRIG bit will be automatically cleared by hardware.

Note: The TSELx[2:0] bits cannot be changed when ENx is 1.

16.2.3 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 registers. The register DAC_DHRx internally held by the system will obtain the above register value and send it to the DAC_DORx register after the corresponding time.

In the non-trigger mode, the data written into the register DAC_xDHRx will be shifted into the DAC_DORx register in1 APB1 clock cycle.

Under software trigger, the DAC_DORx register will be automatically updated in one APB1 clock cycle after the rising edge of the event trigger.

Under hardware trigger (timer TRGO event or external interrupt line 9 rising edge), the DAC_DORx register will be automatically updated in 3 APB1 clock cycles after the trigger event.

Load the DAC_DORx register data. After the time $t_{SETTLING}$, the output will 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 an analog voltage output by the DAC, and its range is 0 to V_{DDA} . The output voltage on any DAC channel pin shall satisfy the following relationship:

DAC output voltage = V_{DDA}* (DAC_DORx/ 4096).

16.2.4 DAC Triangular Wave Generator

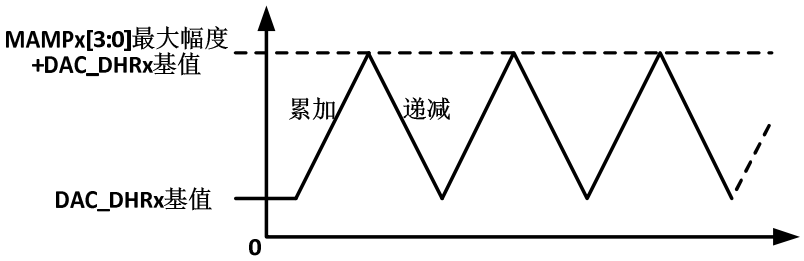
The module has a built-in triangular wave generator, which can add a small amplitude triangle wave to the reference signal. Set WAVEx[1:0] bit as '10' and select the triangular wave generation function of DAC. Set MAMPx[3:0] bit of DAC_CTLR register to select the amplitude of triangular wave.

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] bit, 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] bit 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 of 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 16-3 Triangular Wave Generation



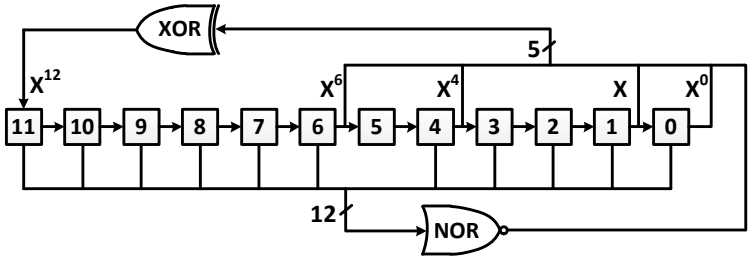
16.2.5 DAC Noise Generator

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 of 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 of 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: 1. To generate a triangular noise, DAC trigger must be enabled, i.e., setting the TENx bit of the DAC_CTLR register to 1.

Figure 16-4 LFSR Register Algorithm



16.3 Register Description

Table 16-2 List of DAC Related Registers

Name	Access address	Description	Reset value
R32_DAC_CTLR	0x40007400	DAC configuration register	0x00000000
R32_DAC_SWTR	0x40007404	DAC software trigger register	0x00000000
R32_DAC_R12BDHR1	0x40007408	DAC channel 1 12-bit right-aligned data holding register	0x00000000
R32_DAC_L12BDHR1	0x4000740C	DAC channel 1 12-bit left-aligned data holding register	0x00000000
R32_DAC_R12BDHR2	0x40007414	DAC channel 2 12-bit right-aligned data holding register	0x00000000
R32_DAC_L12BDHR2	0x40007418	DAC channel 2 12-bit left-aligned data holding register	0x00000000

R32_DAC_DOR1	0x4000742C	Data output register of DAC channel 1	0x00000000
R32_DAC_DOR2	0x40007430	Data output register of DAC channel 2	0x00000000

DAC configuration 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 channel 2: 1: Enable DMA function of DAC channel 2; 0: Disable DMA function of DAC channel 2.	0
[27:24]	MAMP2	RW	DAC channel 2 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 LSFR/ Triangle amplitude equal to 1; 0001: Unmask bit[1:0] of LSFR/ Triangle amplitude equal to 3; 0010: Unmask bit[2:0] of LSFR/ Triangle amplitude equal to 7; 0011: Unmask bit[3:0] of LSFR/ Triangle amplitude equal to 15; 0100: Unmask bit[4:0] of LSFR/ Triangle amplitude equal to 31; 0101: Unmask bit[5:0] of LSFR/ Triangle amplitude equal to 63; 0110: Unmask bit[6:0] of LSFR/ Triangle amplitude equal to 127; 0111: Unmask bit[7:0] of LSFR/ Triangle amplitude equal to 255; 1000: Unmask bit[8:0] of LSFR/ Triangle amplitude equal to 511; 1001: Unmask bit[9:0] of LSFR/ Triangle amplitude equal to 1023; 1010: Unmask bit[10:0] of LSFR/ Triangle amplitude equal to 2047; ≥1011: Unmask bit[11:0] of LSFR/ Triangle amplitude equal to 4095.	0
[23:22]	WAVE2	RW	Noise/Triangular wave generation enable of DAC	0

			channel 2 00: Disable the waveform generator; 01: Enable the noise waveform generator; 1x: Enable the triangular waveform generator.	
[21:19]	TSEL2	RW	Trigger event selection setting of DAC channel 2: 001: TRGO event of TIM3; 100: TRGO event of TIM2; 101: TRGO event of TIM4; 110: External interrupt line 9; 111: Software trigger; Others: Reserved.	0
18	TEN2	RW	External trigger mode enable of DAC channel 2: 1: Enable the trigger function of DAC channel 2; the data written into the DAC_xDHR register will be sent to the DAC_DOR2 register in 3 APB1 clock cycles. 0: Disable the trigger function of DAC channel 2; the data written into the DAC_xDHR register will be sent to the DAC_DOR2 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_DOR2 register in one APB1 clock cycle.*	0
17	BOFF2	RW	DAC channel 2 output buffer disable control (recommended to be enabled): 1: Disable DAC channel 2 output buffer; 0: Enable DAC channel 2 output buffer.	0
16	EN2	RW	DAC channel 2 enable: 1: Enable DAC channel 2; 2: Disable DAC channel 2.	0
[15:13]	Reserved	RO	Reserved.	0
12	DMAEN1	RW	DMA enable of DAC channel 1: 1: Enable DMA function of DAC channel 1; 0: Disablee DMA function of DAC channel 1.	0
[11:8]	MAMP1	RW	DAC channel 1 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 LSFR/ Triangle amplitude equal to 1; 0001: Unmask bit[1:0] of LSFR/ Triangle amplitude equal to 3; 0010: Unmask bit[2:0] of LSFR/ Triangle amplitude equal to 7; 0011: Unmask bit [3:0] of LSFR/ Triangle amplitude equal to 15; 0100: Unmask bit[4:0] of LSFR/ Triangle amplitude equal to 31;	0

			0101: Unmask bit[5:0] of LSFR/ Triangle amplitude equal to 63; 0110: Unmask bit[6:0] of LSFR/ Triangle amplitude equal to 127; 0111: Unmask bit[7:0] of LSFR/ Triangle amplitude equal to 255; 1000: Unmask bit[8:0] of LSFR/ Triangle amplitude equal to 511; 1001: Unmask bit[9:0] of LSFR/ Triangle amplitude equal to 1023; 1010: Unmask bit[10:0] of LSFR/ Triangle amplitude equal to 2047; ≥1011: Unmask bit[11:0] of LSFR/ Triangle amplitude equal to 4095.	
[7:6]	WAVE1	RW	Noise/Triangular wave generation enable of DAC channel 1. 00: Disable the waveform generator; 01: Enable the noise waveform generator; 1x: Enable the triangular waveform generator.	0
[5:3]	TSEL1	RW	Trigger event selection setting of DAC channel 1: 001: TRGO event of TIM3; 100: TRGO event of TIM2; 101: TRGO event of TIM4; 110: External interrupt line 9; 111: Software trigger; Others: Reserved.	0
2	TEN1	RW	External trigger mode enable of DAC channel 1: 1: Enable the trigger function of DAC channel 1; the data written into the DAC_xDHR register will be sent to the DAC_DOR1 register in 3 APB1 clock cycles. 0: Disable the trigger function of DAC channel 1; the data written into the DAC_xDHR register will be 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 one APB1 clock cycle.*	0
1	BOFF1	RW	DAC channel 1 output buffer disable control (recommended to be enabled): 1: Disable the DAC channel 1 output buffer; 0: Enable the DAC channel 1 output buffer.	0
0	EN1	RW	DAC channel 1 enable: 1: Enable DAC channel 1; 2: Disable DAC channel 1.	0

Note: The configuration register includes the configuration of channel 1 and channel 2. When the output of 2 channels is enabled at the same time (ENx bit is '1'), the same waveform will be outputted to 2 hardware channels according to the configuration of channel 1; if the output of channel 1 is enabled, the waveform will be outputted to the hardware channel 1 according to the configuration of channel 1 and channel 2 will

not be outputted; when the output of channel 2 is only enabled, the waveform will be outputted to the hardware 2 according to the configuration of channel 2 on the channel, and channel 1 will not output.

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														SW TRIG2	SW TRIG1

Bit	Name	Access	Description	Reset value
[31:2]	Reserved	RO	Reserved.	0
1	SWTRIG2	WO	Software trigger control bit of DAC channel 2: 1: Enable the software trigger of DAC channel 2. 0: Disable the software trigger of DAC channel 2. *Note: Once the data in DAC_xDHR (after 1 APB1 clock cycle) is sent to the DAC_DOR2 register, this bit will be cleared by hardware.*	0
0	SWTRIG1	WO	Software trigger control bit of DAC channel 1: 1: Enable the software trigger of DAC channel 1. 0: Disable the software trigger of DAC channel 1. *Note: Once the data in DAC_xDHR (after 1 APB1 clock cycle) is sent to the DAC_DOR1 register, this bit will be cleared by hardware.*	0

DAC channel 1 12-bit right-aligned data holding 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	RW	12-bit right-aligned data of DAC channel 1.	0

DAC channel 1 12-bit left-aligned data holding 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
[31:12]	Reserved	RO	Reserved.	0
[11:4]	DACC1DHR	RW	12-bit left-aligned data of DAC channel 1.	0
[3:0]	Reserved	RO	Reserved.	0

DAC channel 2 12-bit right-aligned data holding register (DAC_R12BDHR1)

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	RW	12-bit right-aligned data of DAC channel 2.	0

DAC channel 2 12-bit left-aligned data holding 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
[31:12]	Reserved	RO	Reserved.	0
[11:4]	DACC2DHR	RW	12-bit left-aligned data of DAC channel 2.	0
[3:0]	Reserved	RO	Reserved.	0

Data output register of DAC channel 1 (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	RO	Output data of DAC channel 1	0

Data output register of DAC channel 2 (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	RO	Output data of DAC channel 2	0

Chapter 17 Universal Synchronous Asynchronous Receiver Transmitter (USART)

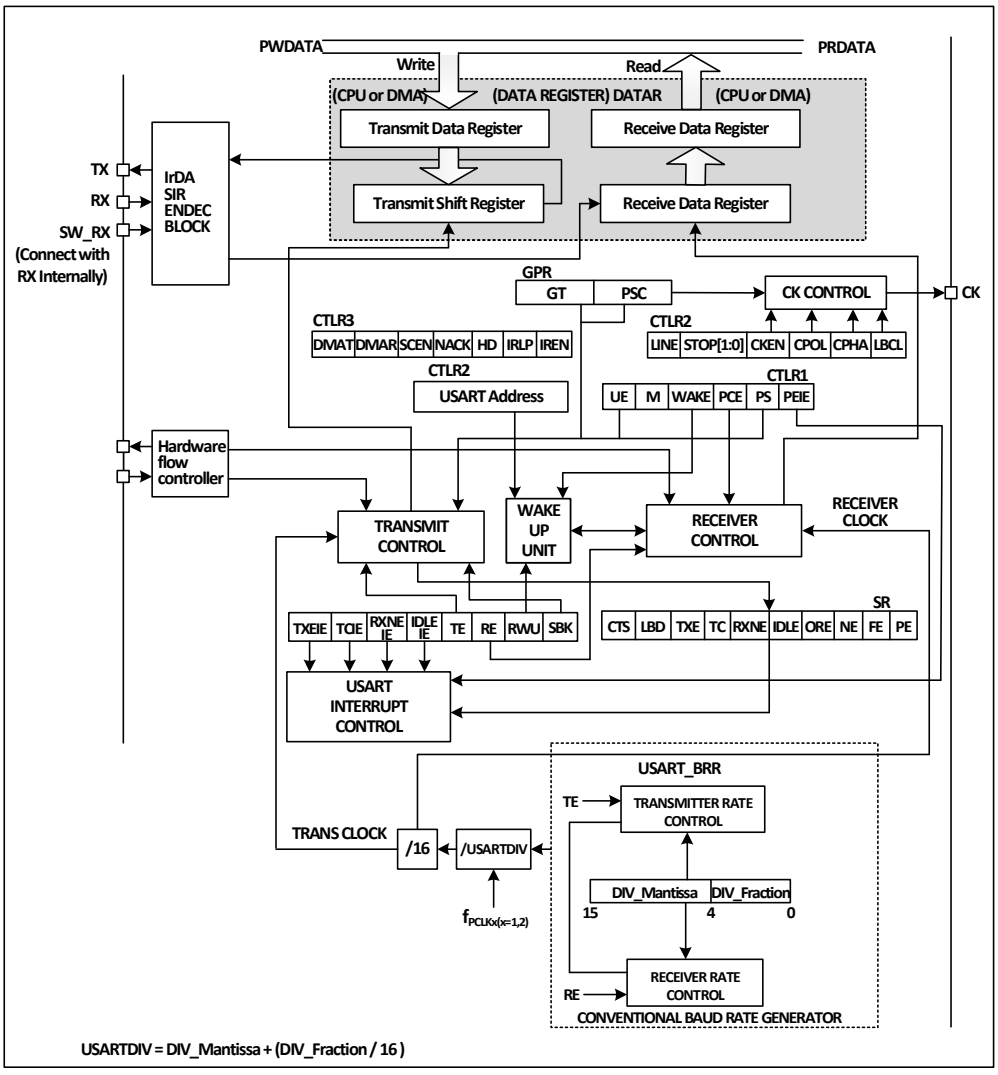
The module description in this chapter applies to the full range of CH32F103 and CH32V103 microcontrollers.

17.1 Main Features

- Full-duplex or half-duplex synchronous or asynchronous communication
- NRZ data format
- Fractional baud rate generator, up to 4.5Mbps
- Programmable data word length
- Configurable stop bit
- Support LIN, IrDA encoder, smart card
- Support DMA
- Multiple interrupt sources

17.2 Overview

Figure 17-1 Block Diagram of USART



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 eight or nine bits of 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 will be the check bit.

After TE is set, an idle frame will be 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.

17.3 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 two domains: DIV_M and DIV_F in USART_BRR. The specific calculation formula is:

$$USARTDIV = DIV\_M+(DIV\_F/16)$$

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 three 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 three 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% .

17.4 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 of the control register 2 (R16_USARTx_CTLR2), but you need to switch off the LIN mode, smart card mode, infrared mode and half-duplex mode at the same time, i.e., to ensure that the SCEN, HDSEL and IREN bits are in the reset status. These three bits are in the control register 3 (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 three bits are in the control register 2 (R16_USARTx_CTLR2). These three bits need to be set when TE and RE are not enabled. The specific difference is shown in Figure 17-2.
4) In the synchronous mode, the receiver will only sample when outputting the clock, and the slave device needs to maintain a certain signal setup time and hold time, specifically as shown in Figure 17-3.

Figure 17-2 USART Clock Timing Example (M=0)

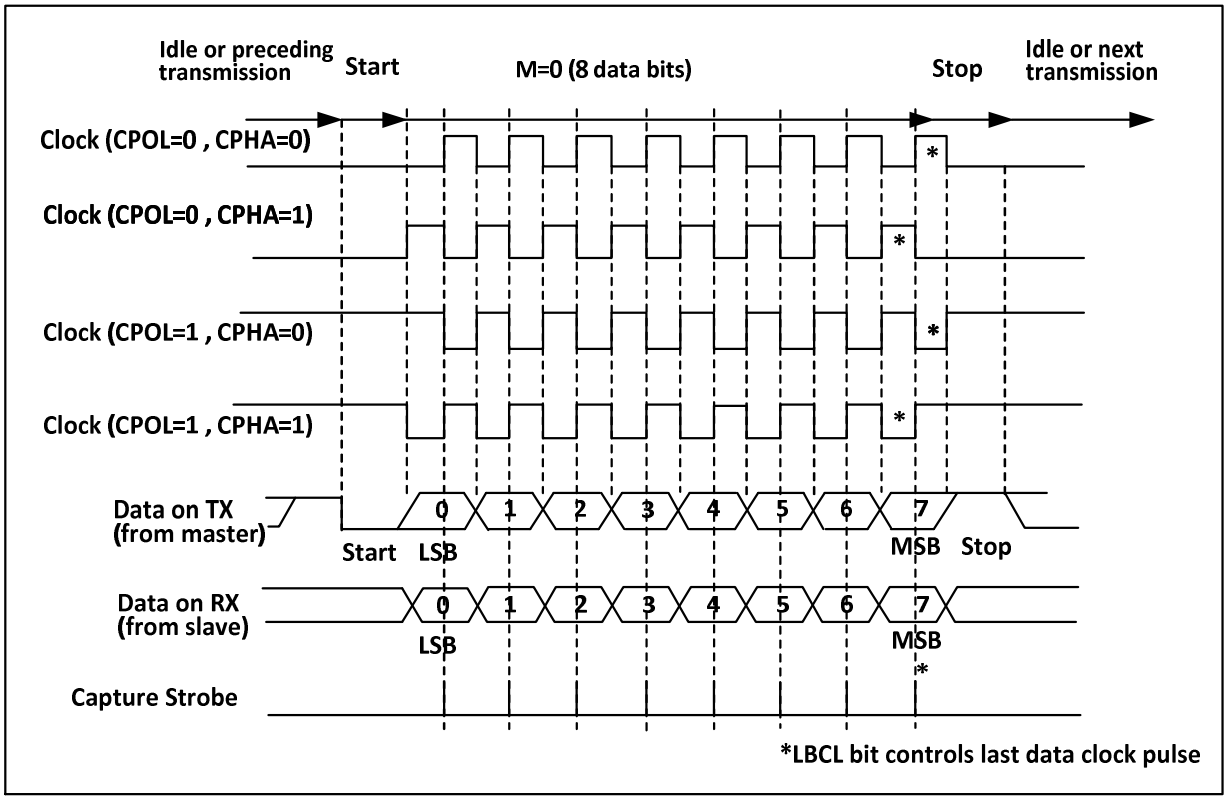
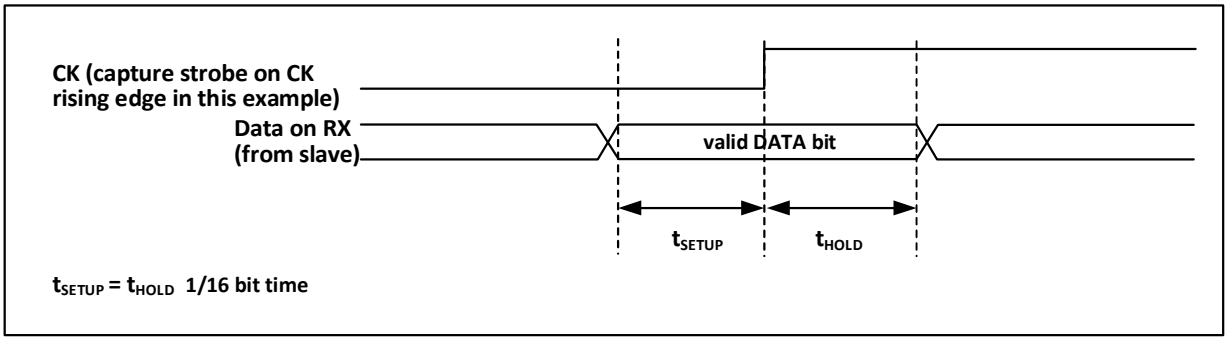


Figure 17-3 Data Sampling Hold Time



17.5 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 of the control register 3 (R16_USARTx_CTLR3), but you need

to disable LIN mode, smart card 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 three bits are in the control registers 2 and 3 (R16_USARTx_CTLR2 and R16_USARTx_CTLR3).

After set to half-duplex mode, it is needed to set the TX IO port to suspended 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.

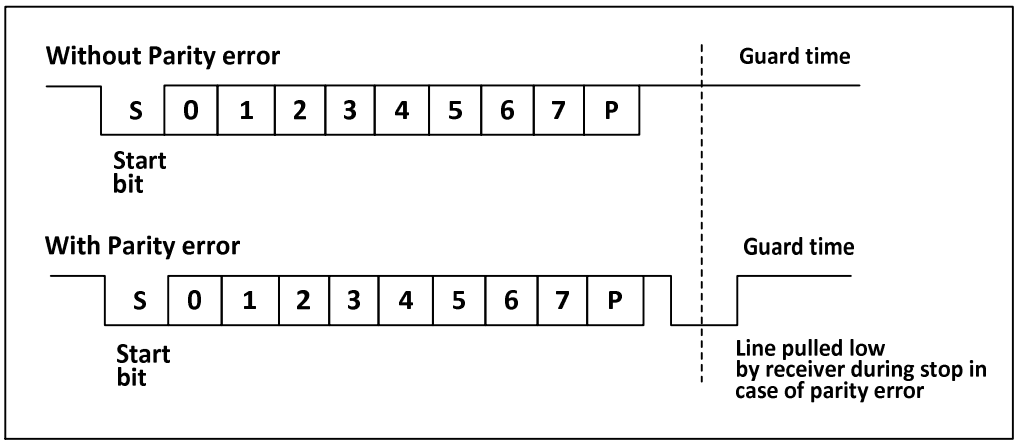
17.6 Smart Card

The smart card mode supports ISO7816-3 protocol to access the smart card controller.

To enable smart card mode, set SCEN bit of the control register 3 (R16_USARTx_CTLR3), but it is needed to disable LIN mode, half-duplex mode and infrared mode at the same time, i.e., to ensure that the LINEN, HDSEL and IREN bits are in the reset status, but CLKEN can be switched on to output the clock. These three bits are in the control registers 2 and 3 (R16_USARTx_CTLR2 and R16_USARTx_CTLR3).

In order to support smart card 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 one 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 17-4 Schematic Diagram of (No) Parity Check Error



In smart card mode, the outputted 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 five-bit settable clock frequency division (the frequency division value is double of PSC, and the highest is frequency division 62).

17.9 IrDA

USART module supports controll 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 115200.

IrDA is a half-duplex protocol. If UASRT sends data to the SIR physical layer, the IrDA decoder will ignore the newly sent infrared signal. If the USART receives data from SIR, then SIR will not accept USART signal. The level logic sent by USART to SIR and SIR to USART is different. In SIR receiving logic, the high level is 1 and the low level is 0. However, in the SIR sending logic, the high level is 0 and the low level is1.

17.10 DMA

The USART module supports the DMA function, and can use DMA to realize fast continuous sending and receiving. When DMA is enabled and TXE is set, DMA will write data to the transmission buffer from the set memory space. When DMA is used for receiving, DMA will transfer the data in the receive buffer to a specific memory space each time RXNE is set,

17.11 Interrupt

The USART module supports a variety of interrupt sources, including transmission data register empty (TXE), CTS, transmission completion (TC), receiving data ready (TXNE), data overflow (ORE), line idle (IDLE), parity check error (PE), disconnection flag (LBD), noise (NE), multi-buffer communication overflow (ORT) and frame error (FE).

Table 17-1 Relationship between Interrupt and Corresponding Enable Bit

Interrupt sources	Enable bit
Data register empty (TXE)	TXEIE
Transmission allowed (CTS)	CTSIE
Transmission completion (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)	
Frame error (FE) in multi-buffer communication	

17.12 Register Description

Table 17-2 USART1 Related Registers List

Name	Access address	Description	Reset value
R32_USART1_STATR	0x40013800	UASRT1 status register	0x000000C0
R32_USART1_DATAR	0x40013804	UASRT1 data register	0x000000XX
R32_USART1_BRR	0x40013808	UASRT1 baud rate register	0x00000000
R32_USART1_CTLR1	0x4001380C	UASRT1 control register 1	0x00000000
R32_USART1_CTLR2	0x40013810	UASRT1 control register 2	0x00000000

R32_USART1_CTLR3	0x40013814	UASRT1 control register 3	0x00000000
R32_USART1_GPR	0x40013818	UASRT1 guard time and prescaler register	0x00000000

Table 17-3 List of USART2 Related Registers

Name	Access address	Description	Reset value
R32_USART2_STATR	0x40004400	UASRT2 status register	0x000000C0
R32_USART2_DATAR	0x40004404	UASRT2 data register	0x000000XX
R32_USART2_BRR	0x40004408	UASRT2 baud rate register	0x00000000
R32_USART2_CTLR1	0x4000440C	UASRT2 control register 1	0x00000000
R32_USART2_CTLR2	0x40004410	UASRT2 control register 2	0x00000000
R32_USART2_CTLR3	0x40004414	UASRT2 control register 3	0x00000000
R32_USART2_GPR	0x40004418	UASRT2 guard time and prescaler register	0x00000000

Table 17-4 List of USART3 Related Registers

Name	Access address	Description	Reset value
R32_USART3_STATR	0x40004800	UASRT3 status register	0x000000C0
R32_USART3_DATAR	0x40004804	UASRT3 data register	0x000000XX
R32_USART3_BRR	0x40004808	UASRT3 baud rate register	0x00000000
R32_USART3_CTLR1	0x4000480C	UASRT3 control register 1	0x00000000
R32_USART3_CTLR2	0x40004810	UASRT3 control register 2	0x00000000
R32_USART3_CTLR3	0x40004814	UASRT3 control register 3	0x00000000
R32_USART3_GPR	0x40004818	UASRT3 guard time and prescaler register	0x00000000

USART status register (R32_USARTx_STATR) (x=1/2/3)

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 will be generated. 1: There is a change on the nCTS status line; 0: There is 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	0

			cleared by the software. If the 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 completion 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 pending 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 will 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

USART data register (USARTx_DATAR) (x=1/2/3)

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	RW	Data register. This register is actually composed of two 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

USART baud rate register (USARTx_BRR) (x=1/2/3)

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_M[11:0]												DIV_F[3:0]			

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved.	0
[15:4]	DIV_M	RW	These 12 bits define the mantissa of the USART divider.	0
[3:0]	DIV_F	RW	These 4 bits define the fraction of the USART divider.	0

USART control register 1 (USARTx_CTLR1) (x=1/2/3)

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	RO	Reserved.	0
13	UE	RW	USART enable bit. When this bit is set, the frequency divider and output of USART will stop working after the current byte transmission is completed.	0
12	M	RW	Word length bit. 1: 9 data bits; 0: 8 data bits.	0
11	WAKE	RW	Wake-up bit. This bit decides the method to wake up USART: 1: Address flag; 0: Idle line.	0
10	PCE	RW	Parity control enable. For the receiver, the parity of	0

			the data is performed; for the transmitter, the check bit is inserted. Once this bit is set, the parity control enable will take effect only after the current byte transmission is completed.	
9	PS	RW	Parity selection. 0 means even parity, and 1 means odd parity. After this bit is set, the parity control enable will take effect only after the current byte transmission is completed.	0
8	PEIE	RW	Parity check interrupt enable bit. When this bit is set, it means that the parity check error interrupt is allowed to be generated.	0
7	TXEIE	RW	Transmit buffer empty interrupt enable. When this bit is set, it means that the transmit buffer empty interrupt is allowed to be generated.	0
6	TCIE	RW	Transmission completion interrupt enable. When this bit is set, it means that the transmission completion interrupt is allowed to be generated.	0
5	RXNEIE	RW	Receive buffer non-empty interrupt enable. When this bit is set, it means that the receive buffer non-empty interrupt is allowed to be generated.	0
4	IDLEIE	RW	Idle line interrupt enable. When this bit is set, it means that the idle line interrupt is allowed to be generated.	0
3	TE	RW	Transmitter enable. When this bit is set, the transmitter will be enabled.	0
2	RE	RW	Receiver enable. When this bit is set, the receiver will be enabled, and the receiver will start 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 bit. 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 will be transmitted; 0: No break character will be transmitted.	0

USART control register 2 (USARTx_CTLR2) (x=1/2/3)

Offset address: 0x10

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	LINEN	STOP		CLKEN	CPOL	CPHA	LBCL	Reserved	LBDIE	LBDL	Reserved	ADD[3:0]			

Bit	Name	Access	Description	Reset value
[31:15]	Reserved	RO	Reserved.	0
14	LINEN	RW	LIN mode enable bit; when this bit is set, the LIN mode will be 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 two 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, enable CK pin. 1: Enable; 0: Disable.	0
10	CPOL	RW	Clock polarity setting bit. In synchronous mode, you can use this bit 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: The high level is maintained on the CK pin when the bus is idle; 0: The 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 bit. 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	The last clock pulse control bit. In synchronous mode, it is used to control whether	0

			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 outputted from CK; 0: The clock pulse of the last bit of data is not outputted from CK. *Note: This bit cannot be modified after enabling transmission.*	
7	Reserved	RW	Reserved.	0
6	LBDIE	RW	LIN break character detection interrupt enable; this bit will 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	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 will be used to wake up a certain USART device.	0

USART control register 3 (USARTx_CTLR3) (x=1/2/3)

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	RO	Reserved.	0
10	CTSIE	RW	CTSIE interrupt enable bit. When this bit is set, an interrupt will be generated when CTS is set.	0
9	CTSE	RW	CTS enable bit; when this bit is set, the CTS flow control will be enabled.	0
8	RTSE	RW	RTS enable bit; when this bit is set, the RTS flow control will be enabled.	0
7	DMAT	RW	DMA transmission enable bit. Set 1 at the bit and use DMA when transmitting.	0
6	DMAR	RW	DMA reception enable bit. Set 1 at the bit and use DMA when receiving.	0
5	SCEN	RW	Smart card mode enable bit; set 1 to enable the smart card mode.	0
4	NACK	RW	Smart card NACK enable bit; when this bit is set, NACK will be transmitted when the check error	0

			occurs.	
3	HDSEL	RW	Half-duplex mode selection bit; set this bit to select half-duplex mode.	0
2	IRLP	RW	Infrared low power selection bit; set this bit to enable low power mode when infrared mode is selected.	0
1	IREN	RW	Infrared enable bit; set this bit to enable the infrared mode.	0
0	EIE	RW	Error enable interrupt bit; after this bit is enabled, under the premise that DMAR is set, an interrupt will be generated if FE or ORE or NE bit is set.	0

USART guard time and prescaler register (USARTx_GPR) (x=1/2/3)

Offset address: 0x18

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

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved.	0
[15:8]	GT	RW	Guard time value. This domain specifies the guard time in unit of baud rate clock. In the smart card mode, the transmission completion flag will be set after the guard time has passed.	0
[7:0]	PSC	RW	Pre-scaler value. 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 the infrared normal mode, this bit can only be set to 1; In the smart card mode, the source clock is frequency-divided by twice the value (the lower 5 bits are valid) to provide the clock to the smart card. A value of 0 means reservation.	0

Chapter 18 Inter-integrated Circuit (I2C) interface

The module description in this chapter applies to the full range of CH32F103 and CH32V103 microcontrollers.

The inter-integrated circuit (I2C) is widely used in the communication between microcontroller and sensor and other off-chip modules. It supports multi-master and multi-slave modes. The communication can be carried out at two speeds: 100 KHz (standard) and 400 KHz (fast) only through two 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 a CRC check function.

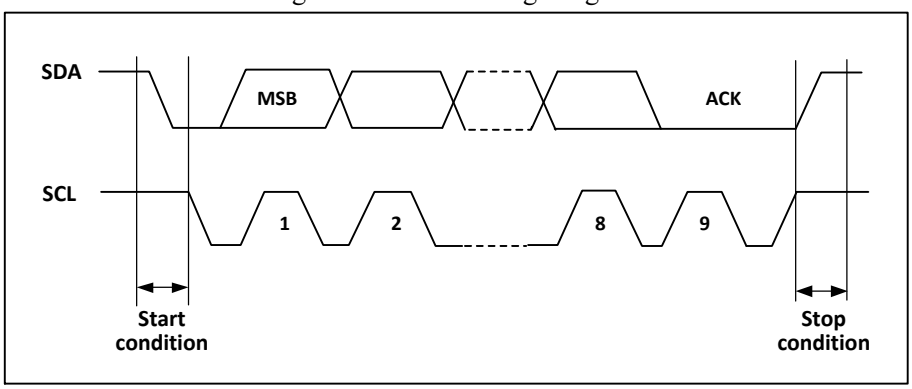
18.1 Main Features

- Support master mode and slave mode
- Support 7-bit or 10-bit address
- Slave device supports dual 7-bit address.
- Support two speed modes: 100KHz and 400KHz
- Multiple status modes, multiple error flags
- Support extended clock function
- 2 interrupt vectors
- Support DMA
- Support PEC
- Compatible with SMBus

18.2 Overview

I2C is a half-duplex bus, and it can only run in one of the following four modes at the same time: master device transmission mode, master device receiving mode, slave device transmission mode and slave device receiving mode. The I2C module works in the slave mode by default. After the start condition is generated, it will automatically switch to the master mode. When the arbitration is lost or the stop signal is generated, it will switch to the slave mode. I2C module supports multi-host function. When working in master mode, the I2C module will actively send out data and addresses. Both data and address are transmitted in the unit of 8 bits, and the high bits are in front of the lower bits. One-byte (under the 7-bit address mode) or two-byte (under the 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 18-1.

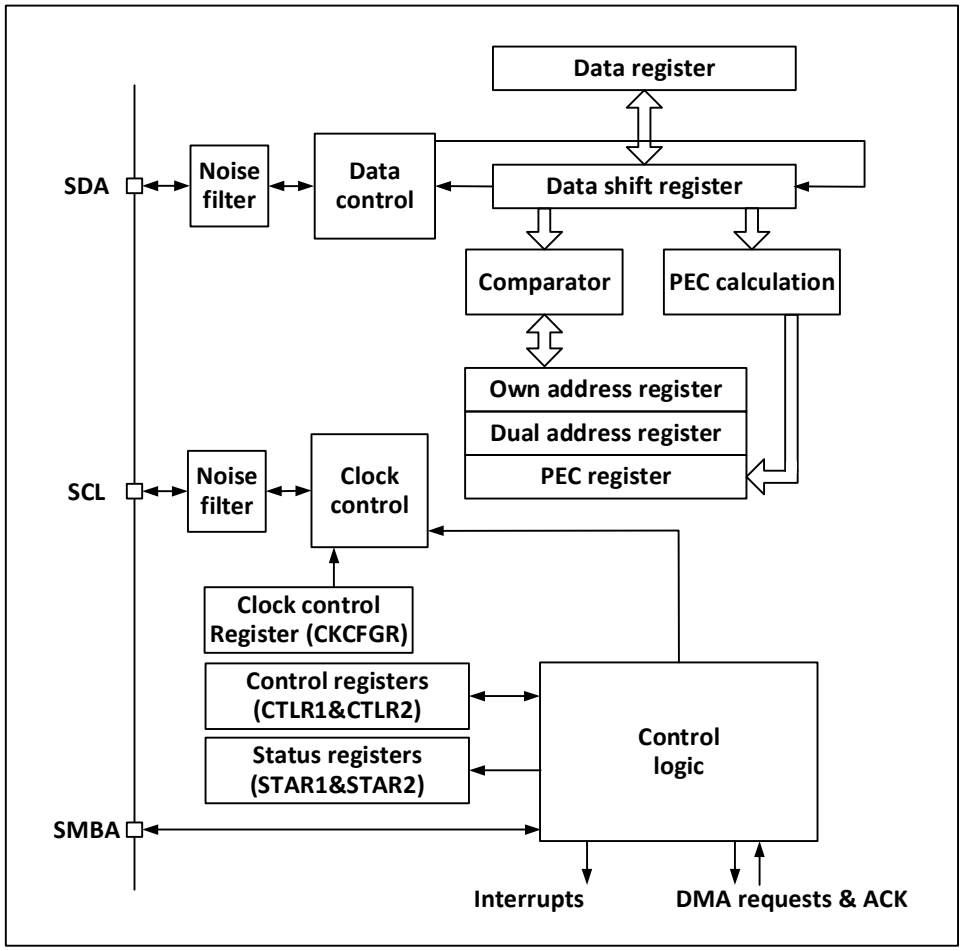
Figure 18-1 I2C Timing Diagram



For normal use, the correct clock must be inputted to I2C. In the standard mode, the minimum input clock is 2MHz, and the minimum input clock is 4MHz in the fast mode.

Figure 18-2 shows the block diagram of I2C.

Figure 18-2 I2C Block Diagram



18.3 Master Mode

In master mode, the I2C module leads the data transmission and outputs the clock signal. The data transmission starts with a Start event and ends with a Stop event. The following is the required sequence in master mode:

1) Set the correct clock in the control register 2 (R16_I2Cx_CTLR2) and the clock control register (R16_I2Cx_CKCFGR);
2) Set a proper rising edge in the rising edge register (R16_I2Cx_RTR);
3) Set the PE bit in the control register (R16_I2Cx_CTLR1) to start the peripheral;
4) 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 will automatically switch to be in master mode, the MSL bit will be set, and the start event will be generated. After the start event is generated, the SB bit will be set. If the ITEVTEN bit (in R16_I2Cx_CTLR2) is set, an interrupt will be generated. At this time, you shall read the status register 1 (R16_I2Cx_STAR1). After writing the slave address to the data register, the SB bit will be automatically cleared;
5) If the 10-bit address mode is used, then write the data register to send the header sequence (the header sequence is 11110xx0b, of which the xx bits are the highest two bits of the 10-bit address).

After the header sequence is transmitted, the ADD10 bit of the status register will be 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.

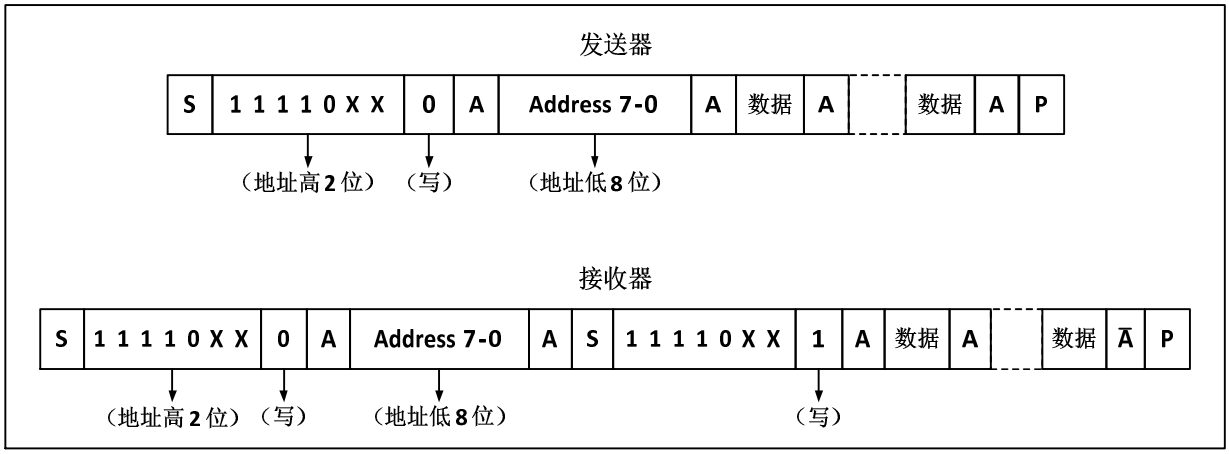
Then, write the data register to send the second address byte. After sending the second address byte, the ADDR bit of the status register will be set. If the ITEVTEN bit has been set, an interrupt will be generated. Read the R16_I2Cx_STAR2 register at this time and then read R16_I2Cx_STAR1 register again to clear the ADDR bit;

If the 7-bit address mode is used, the write data register will tranmit the address byte. After the address byte is sent, the ADDR bit of the status register will be 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 18-3, 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 device transmission mode, send data continuously; if the device enters the master device receiving mode subsequently, a start condition needs to be re-sent, and a byte of 11110xx1 will be sent together. Then, enter the master device receiving mode.

Figure 18-3 Diagram of Master Device Transmission and Receiving Data in case of 10-bit Address



6) In the transmission mode, the shift register inside the master device sends data from the data register to the SDA line. When the master device receives an ACK, the TxE of the status register 1 (R16_I2Cx_STAR1) will be set. If ITEVTEN and ITBUFEN are set, an interrupt will also be generated. Writing data to the data register will clear the TxE bit.

If the TxE bit is set and no new data is written into the data register before the last data is sent, then the BTF bit will be set. Before it is cleared, SCL will remain to be at low level. After reading R16_I2Cx_STAR1, write data into the data register to clear the BTF bit.

In the receiving mode, the I2C module will receive data from the SDA line. Write it into the data register through the shift register. After each byte, if the ACK bit is set, the I2C module will send an acknowledgment low level, and the RxNE bit will be set at the same time. If ITEVTEN and ITBUFEN are set, an interrupt will also be generated. If RxNE is set and the original data is not read before the new

data is received, the BTF bit will be set. Before the BTF is cleared, SCL will remain to be at low level. After R16_I2Cx_STAR1 is read, read the data fetch register to clear the BTF bit.

7) When the master device finishes sending data, it will actively send an end event, i.e., setting the STOP bit. In the receiving mode, the master device needs NAK in the response position of the last data bit. Note that the I2C module will switch to slave mode after NAK is generated.

18.4 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.

In the send mode, after the ADDR bit is cleared, the I2C module will send the byte from the data register to the SDA line through the shift register. After an ACK is received, the TxE bit will be set, and if ITEVTEN and ITBUFEN are set, an interrupt will also be generated. If TxE bit is set but no new data is written into the data register before the next data transmission is completed, the BTF bit will be set. Before BTF is cleared, SCL will remain to be at low level. After the status register 1 (R16_I2Cx_STAR1) is read, writing data to the data register will clear the BTF bit.

In the receiving mode, after the ADDR is cleared, the I2C module will store the data on the SDA into the data register through the shift register. After each byte is received, the I2C module will set an ACK bit and set the RxNE bit. If ITEVTEN and ITBUFEN are set, an interrupt will also be generated. If RxNE bit is set, and the old data is not read before the new data is received, then BTF will be set. SCL will remain to be at low level before 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.

When I2C detects a stop event, it will set the STOPF bit, and if the ITEVFEN bit is set, an interrupt will also be generated. The user needs to read the status register (R16_I2Cx_STAR1) and then write the control register (such as reset control word SWRST) to clear it.

18.5 Error

18.5.1 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.

18.5.2 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 will be set, and an interrupt will be 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.

18.5.3 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.

18.5.4 Overrun/ Underrun Error (OVR)

- Overrun error:

In the slave mode, if clock extension is disabled, the I2C module is receiving data. If one byte of 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.

18.6 Clock Extension

If clock extension is disabled, there is a possibility of overrun/underrun errors. But if clock extension is enabled:

- In sending 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 the receiving 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.

18.7 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.

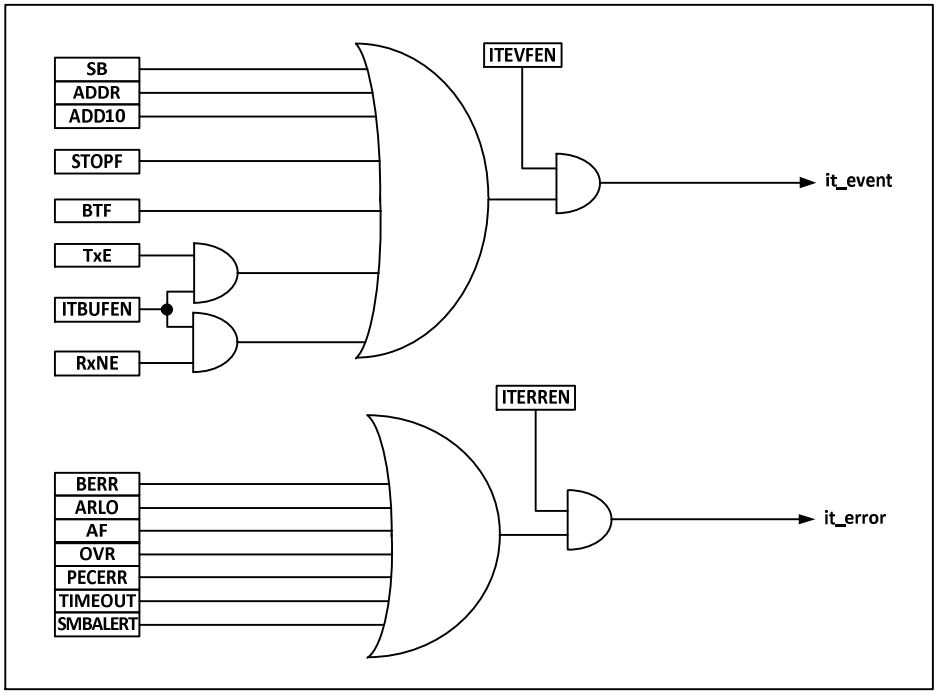
Differences between SMBus and I2C:
1) I2C supports the maximum speed of 400 KHz, while SMBus supports the maximum speed of 100 KHz, and SMBus has the minimum speed limit of 10 KHz;
2) When the SMBus clock is lower than 35mS, it will report a timeout, but I2C has no such limitation;
3) SMBus has a fixed 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.

18.8 Interrupt

Each I2C module has two interrupt vectors: event interrupt and error interrupt. Two types of interrupts support the interrupt sources as shown in Figure 18-4.

Figure 18-4 I2C Interrupt Request



18.9 DMA

DMA can be used to send and receive 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 control register. As long as the TxE bit is

set, the data will 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] bit 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 wholy completed.
5) Activate the channel by setting the EN bit on the DMA_CFGRx register.

When the number of data transfers 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 DMAEN bit is set, DMA receiving mode can be started. When DMA is used for receiving, DMA will transfer 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 using the PL[0:1] 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 wholy completed.
5) Activate the channel by setting the EN bit in the DMA_CFGRx register.

When the number of data transfers 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.

18.10 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 of 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 receiving 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.

18.11 Debug Mode

After the system enters the debug mode, DBG_I2Cx_SMBUS_TIMEOUT bit of the DEBUG module can be used to determine whether to continue operating or stop the time-out control of I2CSMBus.

18.12 Register Description

Table 18-1 List of I2C1 Related Registers

Name	Access address	Description	Reset value
R16_I2C1_CTLR1	0x40005400	I2C1 control register 1	0x0000
R16_I2C1_CTLR2	0x40005404	I2C1 control register 2	0x0000
R16_I2C1_OADDR1	0x40005408	I2C1 address register 1	0x0000
R16_I2C1_OADDR2	0x4000540C	I2C1 address register 2	0x0000
R16_I2C1_DATAR	0x40005410	I2C1 data register	0x0000
R16_I2C1_STAR1	0x40005414	I2C1 status register 1	0x0000
R16_I2C1_STAR2	0x40005418	I2C1 status register 2	0x0000
R16_I2C1_CKCFGR	0x4000541C	I2C1 clock register	0x0000
R16_I2C1_RTR	0x40005400	I2C1 rising time register	0x0002

Table 18-2 List of I2C2 Related Registers

Name	Access address	Description	Reset value
R16_I2C2_CTLR1	0x40005800	I2C2 control register 2	0x0000
R16_I2C2_CTLR2	0x40005804	I2C2 control register 2	0x0000
R16_I2C2_OADDR1	0x40005808	I2C2 address register 1	0x0000
R16_I2C2_OADDR2	0x4000580C	I2C2 address register 2	0x0000
R16_I2C2_DATAR	0x40005810	I2C2 data register	0x0000
R16_I2C2_STAR1	0x40005814	I2C2 status register 1	0x0000
R16_I2C2_STAR2	0x40005818	I2C2 control register 2	0x0000
R16_I2C2_CKCFGR	0x4000581C	I2C2 clock register	0x0000
R16_I2C2_RTR	0x40005800	I2C2 rising time register	0x0002

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 IIC peripheral. Before reset, make sure that the pins of the IIC bus are released and the bus is in idle status. *Note: This bit can reset the IIC module when no stop condition is detected on the bus but the busy*	0

			bit is 1.	
14	Reserved	RO	Reserved.	0
13	ALERT	RW	SMBus alert bit; 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	Data packet error checking enable bit; 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 will be cleared by the 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 bit; this bit can be set or cleared by user code, and it can be cleared by hardware after PE is cleared; 1: The ACK bit controls the ACK or NAK of the next byte received in the shift register. The next byte received in the PEC shift register is PEC; 0: The ACK bit controls the ACK or NAK of the byte currently being 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.*	0
10	ACK	RW	Acknowledge enable bit; 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 bit; 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.	0

			In master mode: 1: A stop event will be 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.	
8	START	RW	Start event generation bit; this bit can be set or cleared by the user code. When the start condition is issued or PE is cleared, it will 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 will be generated; 0: No start event is generated.	0
7	NOSTRETCH	RW	Clock stretching disable bit. 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: Disable clock stretching; 0: Enable clock stretching.	0
6	ENGC	RW	General call enable bit; set this bit to enable the general call, and respond to general address 00h.	0
5	ENPEC	RW	PEC enable bit; set this bit to enable PEC calculation.	0
4	ENARP	RW	ARP enable bit; set this bit to enable the ARP. If SMBTYPE=0, the default address of the SMBus device will be used; if SMBTYPE=1, the main address of the SMBus will be used.	0
3	SMBTYPE	RW	SMBus device type; set 1 as the SMBus master device, and set 0 as the SMBus slave device.	0
2	Reserved	RO	Reserved.	0
1	SMBUS	RW	SMBus mode selection bit; set 1 to use SMBus mode, and set 0 to use IIC mode.	0
0	PE	RW	IIC peripheral enable bit. 1: Enable IIC module; 0: Disable IIC module.	0

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 bit 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 receivier mode and can generate a NAK when the data is received at the last time.*	0
11	DMAEN	RW	DMA request enable bit; set this bit to allow DMA request when TxE or RxEN bit is set.	0
10	ITBUFEN	RW	Buffer interrupt enable bit. 1: When TxE or RxEN bit is set, an event interrupt will be generated; 0: When TxE or RxEN bit is set, no interrupt will be generated.	0
9	ITEVTEN	RW	Time interrupt enable bit; set this bit to enable event interrupt. Under the following conditions, the interrupt will be generated: SB=1 (master mode); ADDR=1(master and slave modes); ADDR10=1 (master mode); STOPF=1 (slave mode); BTF=1, but no TxE or RxEN event occurs; If ITBUFEN=1, TxE event will be 1; If ITBUFEN=1, RxNE event will be 1.	0
8	ITERREN	RW	Error interrupt enable bit; when the bit is set, it indicates that the error interrupt is allowed. Under the following conditions, the interrupt will 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	RW	IIC module clock frequency, the correct clock frequency must be inputted 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

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
ADD MODE	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]	ADD9_8	RW	Interface address, the 9th to 8th bit when a 10-bit address is used, and is ignored when a 7-bit address is used.	0
[7:1]	ADD7_1	RW	Interface address, bit 7-1.	0
0	ADD0	RW	Interface address, the 0th bit when a 10-bit address is used, and is ignored when a 7-bit address is used.	0

I2C address register 2 (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	RW	Interface address; bit 7-1 of address under the dual-address mode.	0
0	ENDUAL	RW	Dual addressing mode enable bit; set this bit to enable the ADD2 to be also identified.	0

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								DATAR[7:0]							

Bit	Name	Access	Description	Reset value
15:8	Reserved	RO	Reserved.	0
7:0	DATAR	RW	Data register. This domain is used to store received data or store data to be transmitted to the bus.	0

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
SMBALERT	TIMEOUT	Reserved	PECERR	OVR	AF	ARLO	BERR	TxE	RxNE	Reserved	STOPF	ADD10	BTF	ADDR	SB

Bit	Name	Access	Description	Reset value
15	SMBALERT	RW0	SMBus alert bit; 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 bit; 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 the slave mode, the slave device will reset the communication and the hardware will release the bus; when this bit is set in the master mode, the hardware will issue a stop condition.*	0
13	Reserved	RO	Reserved.	0
12	PECERR	RW0	PEC error flag bit 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 will be returned; 0: No PEC error.	0
11	OVR	RW0	Overrun and underrun flag bit. 1: Overrun or underrun event occurs: In case of NOSTRETCH=1, when a new byte is received in the receiving mode and the content in the data register has not been read, the newly received byte will be lost; in the transmission 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 bit; this bit can be reset by the user writing 0, or reset by hardware when	0

			PE becomes low. 1: Acknowledge error; 0: Normal acknowledge.	
9	ARLO	RW0	Arbitration lost flag bit; 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 bit; 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 bit, 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 transmitting data register will be empty; 0: The data register is non-empty.	0
6	RxNE	RO	Data register not empty flag bit. 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 bit. After the user reads the status register 1, writing to the control register 1 will clear this bit, or when PE is 0, the hardware will clear this bit. 1: After the response, the slave device will detect a stop event on the bus; 0: No stop event is detected.	0
3	ADD10	RO	10-bit address header sent flag bit. After the user reads the status register 1, writing to the control register 1 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 bit. After the user reads the status register 1, 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	0

			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	
1	ADDR	RW0	Address transmitted/matched flag. After the user reads the status register 1, the read operation of the status register 2 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 bit. 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

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	RO	Packet error checking. When PEC is enabled (ENPEC is set), this domain will store the value of PEC.	0
7	DUALF	RO	Matched detection flag bit, when the stop bit or start bit is generated, or when PE=0, the hardware will clear the bit to zero. 1: The received address matched with OAR2; 0: The received address matched with OAR1.	0
6	SMBHOST	RO	SMBus host header flag bit; when the stop bit or start bit is generated, or when PE=0, the hardware will clear the bit to zero. 1: When SMBTYPE=1 and ENARP=1, the SMBus	0

			host address will be received; 0: SMBus host address is not received.	
5	SMBDEFAULT	RO	SMBus device default address flag bit; when the stop bit or start bit is generated, or when PE=0, the hardware will clear the bit to zero. 1: When ENARP=1, the default address of the SMBus device will be received; 0: No address is received.	0
4	GENCALL	RO	General call address flag bit; when the stop bit or start bit is generated, or when PE=0, the hardware will clear the bit to zero. 1: When ENGC=1, the address of general call will be received; 0: No general call address is received.	0
3	Reserved	RO	Reserved.	0
2	TRA	RO	Transmitter/receiver flag bit, 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 bit; this bit will be cleared when a stop bit is detected. When the interface is disabled (PE=0), the information will be 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 indicator bit. 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

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		Master mode selection bit. 1: Fast mode; 0: Standard mode.	0
14	DUTY		Duty cycle of the high-level time in the fast mode and the high-level time.	0

			1:36%; 0:33.3%.	
[13:12]	Reserved		Reserved.	0
[11:0]	CCR		Clock frequency division factor, determines the frequency waveform of the SCL clock.	0

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	RW	Maximum rise time. The rise time of SCL in the 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 will be (9-1)*125nS, i.e., 1000nS.	000010b

Chapter 19 Serial Peripheral Interface (SPI)

The module description in this chapter applies to the full range of CH32F103 and CH32V103 microcontrollers.

SPI supports data interaction in three-wire synchronous serial mode, supports hardware switching master-slave mode with the chip selection line, and supports communication with a single data line.

19.1 Main Features

- Support full duplex synchronous serial mode
- Support single-wire half-duplex mode
- Support master mode and slave mode, multi-slave mode
- Support 8-bit or 16-bit data structure
- Support half of clock frequency of Fpclk at most
- Support first MSB or LSB in data sequence
- Support hardware or software control NSS pin
- Support the hardware CRC check for transceiving
- Transceiver buffer supports DMA transmission
- Support the modification of the clock phase and polarity

19.2 SPI Functional Description

19.2.1 Overview

Figure 19-1 Structure Block Diagram of Serial Peripheral Bus

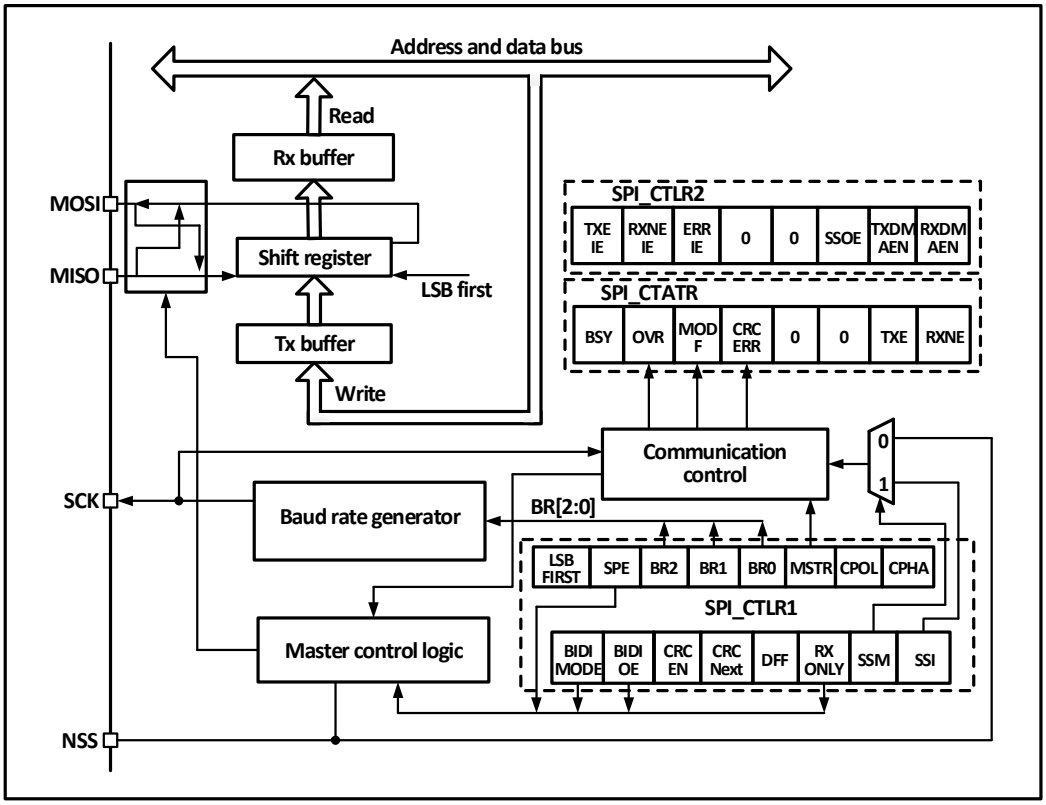


Figure 19-1 shows that the four pins related to SPI are MISO, MOSI, SCK and NSS. The MISO pin is a data

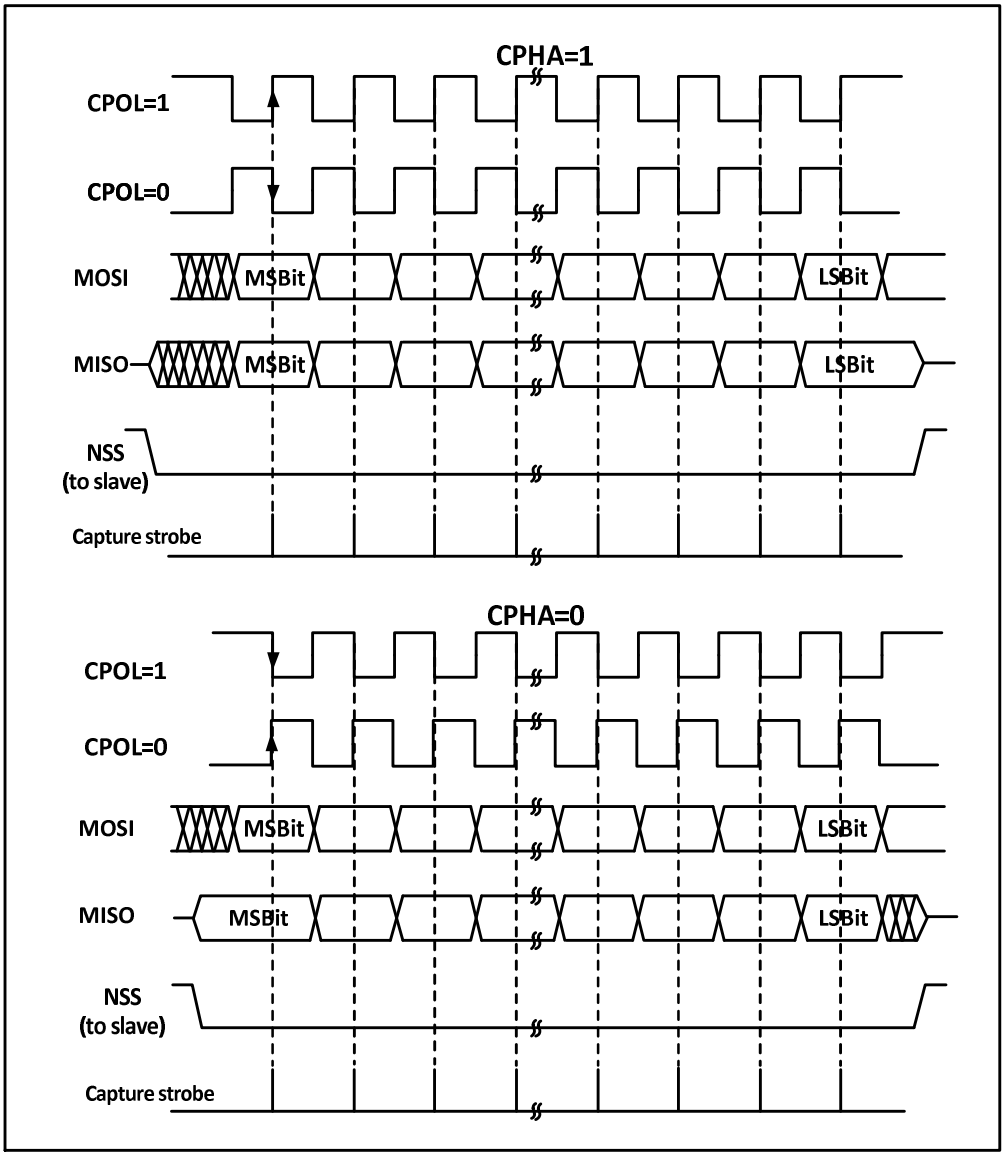
input pin when the SPI module works in the master mode; it is a data output pin when working in the slave mode. When the MOSI pin works in the master mode, it is the data output pin; when it works in the slave mode, it is the data input pin. SCK is a clock pin; the clock signal is always outputted by the master, and the slave receives the clock signal and synchronizes the transmission and receiving of data.

NSS pin is a chip selection pin and can be used as follows:

1) NSS is controlled by software: SSM is set at this time, and internal NSS signal is determined by SSI to output high or low value. This circumstance is generally applied in the SPI master mode;
2) NSS is controlled by hardware: when the NSS output is enabled, i.e., when the SSOE is set, the SPI host will actively pull down the NSS pin when sending out the output. If the NSS pin is pulled down, a hardware error will occur; if SSOE bit is not set, it can be used in the multi-master mode. If it is pulled low, it will force to enter the slave mode, and the MSTR bit will be automatically cleared.

The operating mode of SPI can be configured through CPHA and CPOL. When CPHA bit is set, it means that the module performs data sampling on the second edge of the clock and the data is latched. When CPHA is latched, 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 to be high or low when there is no data. See the Figure 18-2 for details.

Figure 19-2 SPI Mode



The host and device need to be set to the same SPI mode, and the SPE bit needs to be cleared before the SPI

mode is configured. The DEF bit can determine whether the single data length of SP is 8 bits or 16 bits. LSBFIRST can control whether the single data word is high bit in front or low bit in front.

19.2.2 Master Mode

When the SPI module is working in the master mode, the serial clock will be generated by SCK. The master mode is configured in the following steps:

1) Configure BR[2:0] of the control register to determine the clock;
2) Configure CPOL and CPHA bits to determine the SPI mode;
3) Configure DEF to determine the data word length;
4) Configure LSBFIRST to determine the frame format;
5) Configure the NSS pin, such as setting the SSOE bit to enable the hardware to set NSS. SSM bit can be also set and SSI bit can be also set to be high;
6) Set the MSTR bit and SPE bit to ensure that the NSS is already high at this time.

When data needs to be sent, only the data to be sent needs to be written to the data register. SPI will send data from the transmission 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 is set, the data register needs to be filled with data to maintain a complete data flow.

When the receiver receives data and the last sampling clock edge of the data word arrives, the data will be transmitted from the shift register to the receive buffer in parallel, and the RXNE bit will be set. If the RXNEIE bit is previously set, an interrupt will be generated. At this time, the data register shall be read as soon as possible to remove the data

19.2.3 Slave Mode

When the SPI module is working in slave mode, SCK will be used to receive the clock sent by the host, and its own baud rate setting will be invalid. The steps for configuring to the slave mode are as follows:

1) Configure the DEF bit to set the data bit length;
2) Configure the CPOL and CPHA bits to match the host mode;
3) Configure the LSBFIRST to match the host data frame format;
4) In hardware management mode, the NSS pin needs to be kept at low level. If NSS is set to software management (SSM bit is set), then please keep SSI bit not set;
5) Clear the MSTR bit, set SPE bit and enable the SPI mode.

When the first slave receiving sampling edge appears in SCK during transmission, the slave will start sending. The process of transmission is to move the data in the transmission buffer to the transmitter shift register. When the data in the transmission buffer is moved to the shift register, the TXE flag will be set. If the TXEIE bit is set before, then an interrupt will be generated.

During reception, the RXNE bit will be set after the last clock sampling edge, the byte received by the shift register will be transferred to the receive buffer, and the data in the receive buffer can be obtained by reading the data register. If the RXNEIE bit has been set before RXNE bit is set, the corresponding interrupt will be generated.

19.2.4 Simplex Mode

The SPI interface can work in half-duplex mode, i.e., the master device uses the MOSI pin, and the slave device uses the MISO pin for communication. When the half-duplex communication is used, you need to set

BIDIMODE and use BIDIOE to control the transmission direction.

The SPI module can be set to the simplex mode of receiving only by setting the RXONLY bit in the normal full-duplex mode. After RXONLY bit is set, a data pin will be released. The pins released in master mode and slave mode are different. The received data can be also ignored to set SPI to the only transmitting mode.

19.2.5 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 for 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 will enable the CRC check and reset 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. Meanwhile, if the last received value of the receiving shift register is not the same as the locally calculated value of the RXCRCR, the CRCERR bit will be set. To use the CRC check, 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 CRC and receive CRC check. Note that the CRC calculation polynomials of the sender and receiver shall be unified.

19.2.6 DMA

The SPI module supports the use of DMA to speed up data communication. DMA can be used to fill in data in the transmission buffer, or DMA can be used to timely take data from the receive buffer. DMA will use RXNE and TXE as signals to timely fetch or send data. DMA can also work in simplex or CRC check mode.

19.2.7 Error

- Master Mode Failure Error

When the SPI is working in the NSS pin hardware management mode, and the NSS pin is pulled down externally; or the SSI bit is cleared in the NSS pin software management mode; or the SPE bit is cleared, switching off the SPI; or the MSTR bit is cleared, SPI will enter the slave mode. If the ERRIE bit has been set, an interrupt will be generated.

- Overflow error

If the host sends data and there is still unread data in the receive buffer of the slave, an overflow error will occur and the OVR bit will be set. An interrupt will be generated if ERRIE is set. When an overflow error is sent, the current transmission will be restarted. Read the data register and then the status register to clear this bit.

- CRC Error

When the received CRC check word does not match the value of RXCRCR, a CRC check error will occur, and the CRCERR bit will be set.

19.2.8 Interrupt

The interrupt of the SPI module supports five interrupt sources, of which for the empty transmission buffer and the non-empty receive buffer, TXE and RXNE bits wil be set respectively, and an interrupt will be generated when the TXEIE and RXNEIE bits are set respectively. In addition, the three errors mentioned above will also generate interrupts: MODF, OVR and CRCERR. After the ERRIE bit is enabled, these three errors will also generate errors.

19.3 Register Description

Table 19-1 List of SPI1 Related Registers

Name	Access address	Description	Reset value
R16_SPI1_CTLR1	0x40013000	SPI1 control register 1	0x0000
R16_SPI1_CTLR2	0x40013004	SPI1 control register 2	0x0000
R16_SPI1_STATR	0x40013008	SPI1 status register	0x0002
R16_SPI1_DATAR	0x4001300C	SPI1 data register	0x0000
R16_SPI1_CRCR	0x40013010	SPI1 polynomial register	0x0007
R16_SPI1_RCRCR	0x40013014	SPI1 receiving CRC register	0x0000
R16_SPI1_TCRCR	0x40013018	SPI1 transmission CRC register	0x0000

Table 19-2 List of SPI2 Related Registers

Name	Access address	Description	Reset value
R16_SPI2_CTLR1	0x40003800	SPI2 control register 1	0x0000
R16_SPI2_CTLR2	0x40003804	SPI2 control register 2	0x0000
R16_SPI2_STATR	0x40003808	SPI2 status register	0x0002
R16_SPI2_DATAR	0x4000380C	SPI2 data register	0x0000
R16_SPI2_CRCR	0x40003810	SPI2 polynomial register	0x0007
R16_SPI2_RCRCR	0x40003814	SPI2 receiving CRC register	0x0000
R16_SPI2_TCRCR	0x40003818	SPI2 transmission CRC register	0x0000

SPI control register 1 (SPIx_CTLR1) (x=1/2)

Offset address: 0x00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIDI MODE	BIDI OE	CRCEN	CRC NEXT	DFF	RX ONLY	SSM	SSI	LSB FIRST	SPE	BR[2:0]			MSTR	CPOL	CPHA

Bit	Name	Access	Description	Reset value
15	BIDIMODE	RW	One-way data mode enable bit. 1: Select single-line bidirectional mode; 0: Select double-line bidirectional mode.	0
14	BIDIOE	RW	Single-line output enable bit, and used with BIDIMODE. 1: Enable output, only transmit; 0: Disable output, only receivr.	0
13	CRCEN	RW	Hardware CRC check enable bit. This bit can only be written when SPE is 0. This bit can only be used in the full duplex mode. 1: Enable CRC calculation; 0: Disable CRC calculation.	0
12	CRCNEXT	RW	After the next data transmission, send the value of the CRC register. This bit shall be set immediately after the last data is written to the data register.	0

			1: Transmit CRC check result; 0: Continuously transmit the data of data register.	
11	DFF	RW	Data frame length bit; this bit can only be written when SPE is 0. 1: The 16-bit data length is used for transmission and receiving; 0: The 8-bit data length is used for transmission and receiving;	0
10	RXONLY	RW	In two-wire mode, only a bit is received, and this bit is used with BIDIMODE. Set this bit to enable this device to only receive rather than transmitting. 1: Only receiving; simplex mode; 0: Full-duplex mode.	0
9	SSM	RW	Chip selection pin management bit; this bit determines whether the level of the NSS pin is controlled by hardware or software. 1: Software control NSS pin; 0: Hardware control NSS pin.	0
8	SSI	RW	Chip selection pin control bit. When SSM is set, this bit determines the level of the NSS pin. 1: NSS is high level; 0: NSS is low level.	0
7	LSBFIRST	RW	Frame format control bit. This bit cannot be modified during communication. 1: Transmit LSB firstly; 0: Transmit MSB firstly.	0
6	SPE	RW	SPI enable bit. 1: Enable SPI; 0: Disable SPI.	0
[5:3]	BR	RW	Baud rate setting; cannot be modified during communication. 000: $F_{PCLK}/2$; 001: $F_{PCLK}/4$; 010: $F_{PCLK}/8$; 011: $F_{PCLK}/16$; 100: $F_{PCLK}/32$; 101: $F_{PCLK}/64$; 110: $F_{PCLK}/128$; 111: $F_{PCLK}/256$.	0
2	MSTR	RW	Master/slave setting bit; this bit cannot be modified during communication. 1: Configure as the master device; 0: Configure as the slave device.	0b
1	CPOL	RW	Clock polarity selection bit; this bit cannot be modified during communication. 1: SCK is kept at high level when idle; 0: SCK is kept at low level when idle;	0
0	CPHA	RW	Clock phase setting bit; this bit cannot be modified during communication. 1: Sample from the second clock edge;	0

			0: Sample from the first clock edge.	

SPI control register 2 (SPIx_CTLR1) (x=1/2)

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 bit. When this bit is set, TXE is allowed to generate interrupt when set.	0
6	RXNEIE	RW	Receive buffer not empty interrupt enable bit. When the bit is set, RXNE is allowed to generate interrupt during the bit setting.	0
5	ERRIE	RW	Error interrupt enable bit. When the bit is set, interrupt is allowed to be generated during error generation (CRCERR, OVR and MODF).	0
[4:3]	Reserved	RO	Reserved.	0
2	SSOE	RW	SS output enable. Disable the 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 bit. 1: Enable the transmit buffer DMA; 0: Disable the transmit buffer DMA.	0
0	RXDMAEN	RW	Receive buffer DMA enable bit. 1: Enable the receive buffer DMA; 0: Disable the receive buffer DMA;	0

SPI status register (SPIx_STATR) (x=1/2)

Offset address: 0x08

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BSY	OVR	MODF	CRCERR	Reserved		TXE	RXNE

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved.	0
7	BSY	RO	Busy flag bit, set or reset by hardware. 1: SPI is in the process of communication, or the transmitter buffer is not empty; 0: SPI is not in the process of communication.	0
6	OVR	RWO	Overrun flag bit, set by hardware and reset by	0

			software. 1: An overrun error is generated; 0: No overrun error is generated.	
5	MODF	RO	Mode error flag bit, set by hardware and reset by software. 1: A mode error is generated; 0: No mode error is generated.	0
4	CRCERR	RW0	CRC error flag bit, set by hardware and reset by software. 1: The received CRC value does not match the RCRCR value; 0: The received CRC value matched the RCRCR value.	0
[3:2]	Reserved	RO	Reserved.	0
1	TXE	RO	Transmit buffer empty flag bit. 1: The transmit buffer is empty; 0: The transmit buffer is not empty.	1
0	RXNE	RO	Receive buffer non-empty flag bit. 1: The receive buffer is not empty; 0: The receive buffer is empty.	0

SPI data register (SPIx_DATAR) (x=1/2)

Offset address: 0x0C

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

Bit	Name	Access	Description	Reset value
[15:0]	DATAR	RW	Data register. The data register is used to store the received data or pre-store the data to be sent out. Therefore, the reading and writing of the data register actually corresponds to different areas of the operation. The read corresponds to the receive buffer, and the write corresponds to the transmission buffer. 8-bit or 16-bit data can be received and transmitted, so it is necessary to determine bits of data to be used before transmission. When 8 bits are used for data transmission, only the lower 8 bits of the data register are used, and the higher 8 bits will be forced to 0 during reception. All 16-bit data registers to will be used when the 16-bit data structure is used.	0

SPI polynomial register (SPIx_CRCR) (x=1/2)

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	RW	CRC polynomial. This domain defines the polynomial used in the CRC calculation.	7

SPI receive CRC register (SPIx_RCRCR) (x=1/2)

Offset address: 0x14

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

Bit	Name	Access	Description	Reset value
[15:0]	RXCRC	RO	Receive CRC value. The calculated CRC check result of the received byte is stored. The register will be reset by setting the CRCEN bit. The polynomial used by CRCPOLY is used as the calculation method. Only the lower 8 bits are involved in the calculation in the 8-bit mode, and all 16 bits are involved in the calculation in the 16-bit mode. This register needs to be read when the BSY is 0.	0

Transmit CRC register (SPIx_TCRCR) (x=1/2)

Offset address: 0x18

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

Bit	Name	Access	Description	Reset value
[15:0]	TXCRC	RO	Transmit CRC value. The calculated CRC check result of the transmitted byte is stored. The register will be reset by setting CRCEN bit. The polynomial used by CRCPOLY is used as the calculation method. Only the lower 8 bits are involved in the calculation in the 8-bit mode, and all 16 bits are involved in the calculation in the 16-bit mode. This register needs to be read when the BSY is 0.	0

Chapter 20 USB Full Speed Device Controller (USBD)

The module description in this chapter applies to the full range of CH32F103 microcontrollers.

The USBD module is a USB full-speed, low-speed protocol communication controller designed based on the USB2.0 full-speed device technical specification. The built-in hardware automatically processes the reverse non-return-to-zero (NRZI) encoding/decoding and bit stuffing of the physical signal. The control can drive multiple states of the USB bus, protocol packet transceiving, and provide functions such as automatic response for flow control to ensure application program processing time.

20.1 Main Properties

- USB2.0 full-speed device technical specification compliant
- Support 12Mbps USB full-speed mode and 1.5Mbps low-speed mode
- Support configure 16 transmission channels
- Support endpoint address range 0-15
- Support control, interrupt, bulk and synchronous transmission
- Support bulk/synchronous endpoint dual-buffer mechanism
- USB suspension, wake-up and resume operation
- The hardware automatically performs data PID flipping and transmission flow control
- Frame lock clock pulse generation

Note: USBD and CAN controllers share a dedicated 512-byte SRAM area in the design for data transmission and reception. Therefore, when USBD and CAN functions are used at the same time, this shared area needs to be allocated reasonably to prevent data conflicts.

20.2 Functional Description

20.2.1 Introduction to Functions

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, which is completed through the coordination of the application program and the module hardware. The module contains a shared 512-byte dedicated SRAM area as the USB transceiving data buffer. The actual use range is determined by the number of configured endpoints and the maximum packet size of each endpoint. The 512-byte buffer can be used for each endpoint at most, and can be used for up to 16 unidirectional or 8 bidirectional endpoints.

The USBD module has the following functions:

- Physical signal encoding/decoding: According to the USB specification, the PID detection of token packet, data packet and handshake packet is realized, including bit stuffing, CRC generation and verification and frame header synchronization recognition.
- Transaction processing: Judging the correct transmission and error status and providing respective flag status and interrupt notification.
- Bus suspension /reset/wake-up status recognition notification.
- Automatic data packet PID: According to the protocol, the PID of the transceiving data packets of the asynchronous and synchronous endpoints is flipped or locked by hardware to reduce the application program 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 sufficient processing and preparation time for the application, but does not affect the physical transmission and reception on the USB bus.

- Managment data transmission and reception: Locating the endpoint configuration and buffer description area, and detecting the buffer boundary to prevent overflow. Single buffer/double buffer management, interrupt reporting priority management by endpoint type, etc.
- Providing general type, endpoint type and buffer description type register configuration.

The application program can:

- Acquire the frame interval time point based on the USB protocol, and the bus status: suspended and reset.
- Self-define the number of end-points, end-point type and end-point size. Self-define the transmission data buffer.
- Acquire the service at the current or suspended endpoint for processing.
- Acquire the error status such as bit stuffing, format, CRC, protocol, missing ACK and buffer overflow/underfilled buffer.
- The drive module enters low power mode.

The USBD module maps USB events to 3 different NVIC request lines (3 interrupt numbers are used):

1) USB high priority interrupt (channel 19): It can only be triggered by the correct transmission event of synchronous and double-buffer bulk transmission so as to ensure the maximum transmission rate.
2) USB low priority interrupt (channel 20): It can be triggered by all USB events (correct transmission and USB reset, etc.). The firmware shall firstly determine the source of the interrupt source before processing the interrupt.
3) USB wake-up interrupt (channel 42): It is triggered by a wake-up event in the USB suspension mode.

20.2.2 Functional Configuration

GPIO Port:

Once the USBD module is enabled, the GPIO ports used as UDP and UPM will be automatically connected to the internal USB transceiver and disconnect the port settings of its GPIO peripherals. Therefore, it is recommended that the GPIO port shall be configured to output low level in the push-pull mode to prevent the indeterminate state of the port before the USBD function is enabled or notify the USB device access in advance during connection to the PC host.

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).

Module Initialization:

First of all, the analog part related to the USB transceiver requires a standard 48MHz clock as the reference clock, which comes from the AHB bus. The application program needs to firstly configure the corresponding control bit (RCC_CFGR0 register) of the clock management logic to ensure that the current USB clock is 48MHz, and then enable the USB interface clock so that the program can access the register of the USBD module.

Secondly, when the module is forced to be reset (the FRES bit on the USBD_CNTR register is 1 by default), the application program shall initialize the required registers and packet buffer description table. Including: packet buffer description table address register (USBD_BTABLE), endpoint configuration register x

(USBD_EPRx) and packet buffer description table register. Configure the ADD[6:0] domain of the USBD_DADDR register to 0 (the default address of the USB protocol), and set the EF bit to enable the endpoint transmission 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 various status flags in the USBD_ISTR register so as to clear the non-processed false interrupt flag before enabling the operation of any other unit. Switch on the interrupt control bit required in the USBD_CNTR register.

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 will be disabled (the USBD module will not respond to any packet transmission). After the USB is reset, the USBD module will be enabled, and the USB endpoint also needs to be enabled to respond to the USB host (the EF bit of the USB_DADDR register is 1). During the enumeration phase of the USB device, the host will assign a unique address to the device, and the address must be written into the ADD[6:0] bit of the USB_DADDR register.

Note: The RESET flag comes from the status of the forced reset control bit (FRES) of the USBD module and the start of the USB bus reset signal.

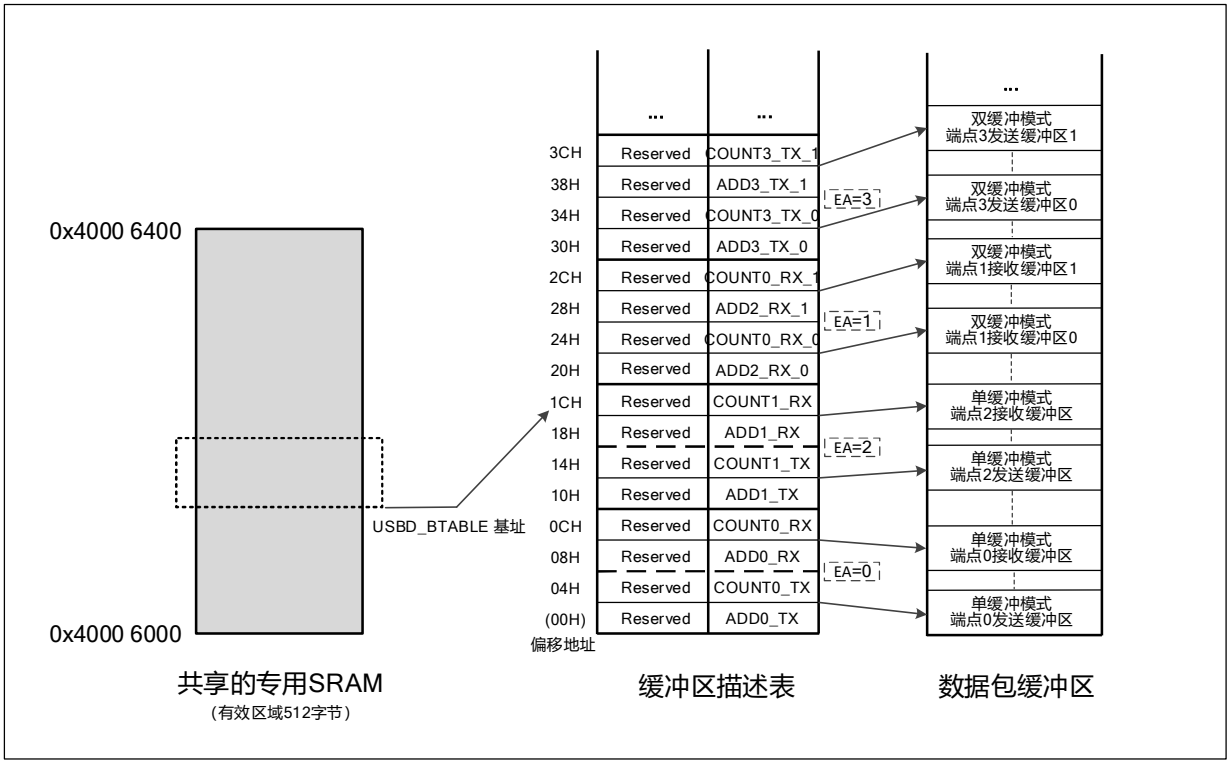
Endpoint configuration and buffer description table
Each endpoint configuration register can be configured with a two-way endpoint single-buffering attribute, or a one-way endpoint double-buffering attribute.

For example: Configure the two-way endpoint single buffer attribute. Configure the register 3 (USBD_EPRx) at the endpoint and set EA[3:0] to 2, so there can be an endpoint 2 upload channel and an endpoint 2 download channel on USB transmission (specifically determined by the descriptor information); configure the one-way endpoint double buffer attributes (only specific to the bulk endpoint and synchronous endpoint). Configure the register 3 (USBD_EPRx) at the endpoint and set EA[3:0] to 2. The endpoint type (EPTYPE) is synchronous or bulk endpoint. Set the EP_KIND bit to 1, so there can be an endpoint 2 upload channel or an endpoint 2 download channel on USB transmission. Select 1 of 2. The transceiving is faster compared with the single buffer. The microcontroller processing and the USBD module physical transceiving can be conducted simultaneously to reduce the waiting time.

Note: The USBD module has a built-in conflict arbitration mechanism, so that the microcontroller and USBD module access the packet buffer as a dual-port SRAM. Even if the microcontroller continuously accesses the buffer, there will be no access conflict.

Each endpoint configuration register corresponds to a set of buffer description registers (description table) and the corresponding data transceiving buffer. They are all located at the shared 512-byte dedicated SRAM area (base address 0x40006000). The USBD_BTABLE register defines the initial address of the buffer description table in the SRAM area, and the data transceiving buffer 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 problem of allocation conflicts.

Figure 20-1 Buffer Description Table Structure



For receiving or transmission, the packet buffer is used from the bottom. The USBD module will not change the contents of other buffers beyond the buffer currently allocated. If the buffer receives a data packet larger than itself, it will only receive data up to its own size, and discard the others, i.e., a so-called buffer overflow exception occurs.

1) Endpoint Initialization

The first step of the endpoint initialization is to write the appropriate value to the USBD_ADDRx_TX or USBD_ADDRx_RX register, so that the USBD module can find the data to be transmitted or the buffer that is ready to receive the data. The EPTYPE[1:0] bit of the USBD_EPRx register determines the basic type of the endpoint, and the EP_KIND bit determines the special characteristics of the endpoint. The sender needs 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. The receiver needs to set the STAT_RX bit to enable the endpoint, set the BL_SIZE and NUM_BLOCK bits and determine the size of the receive buffer to detect buffer overflow exception. For the one-way endpoint of asynchronous non-double-buffer bulk transmission, only one register at the transmission direction needs to be set. Once the endpoint is enabled, the application program can no longer modify the value of the USBD_EPRx register and the location of the USBD_ADDRx_TX/USBD_ADDRx_RX and USBD_COUNTx_TX/ USBD_COUNTx_RX registers, because these values will be modified by the hardware real-timely. When the data transmission is completed, the CTR interrupt will be generated. At this time, the above-mentioned registers can be accessed and a new transmission can be re-enabled.

2) IN Transaction (for Data Transmission)

When an IN token packet is received, if the received address matches a configured endpoint address, and the STAT_TX bit on the register USBD_EPRx indicates that it can be transmitted, the USBD module will packet the coding and send out the data packet according to the contents of the buffer description table and DTOG_TX bit. If the endpoint corresponding to the received token packet is invalid, the NAK or STALL handshake packet rather than the data packet will be sent according to STAT_TX bit in the USBD_EPRx

register.

After the ACK handshake packet responded by the host is received, the value of the USBD_EPRx register has the following updates: DTOG_TX bit is flipped, STAT_TX bit is ‘10’ (NAK status), the endpoint is invalidated, and CTR_TX bit is set. The application program needs to identify the USB endpoint where the interrupt is generated through the EP_ID and DIR bits of the USBD_ISTR register. The interrupt service program of the CTR_TX event needs to firstly clear the interrupt flag bit. If the data needs to be transmitted (which can be executed when the data needs to be transmitted), the buffer of the data to be transmitted shall be prepared. The COUNTx_TX shall be updated as the number of bytes to be transmitted next time. Finally, set the STAT_TX bit as '11' (ACK, valid endpoint) and then enable the data transmission. When the STAT_TX bit is '10' (NAK status), any IN request sent to the endpoint will be negatively acknowledged, and the USB host will retransmit the IN request until the endpoint confirms that the request is valid.

3) OUT Transaction and SETUP Transaction (for Data Reception)

The USBD module processes these two transactions basically in the same way; when an OUT or SETUP packet is received, if the received address matches a configured endpoint address, and the STAT_RX bit on the register USBD_EPRx indicates that it can be received, the USBD module will judge whether the received data matches the PID according to the DTOG_RX bit. If it matches, the module 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 low byte is received first) in the address space defined by ADDRx_RX and detect whether the receiving overflows the buffer according to the values of BL_SIZE and NUM_BLOCK. If no error occurs during the transmission, an ACK handshake packet will be sent to the host. Even if a CRC error or other type of error (bit stuffing, frame error, etc.) occurs, the data will still be saved in the packet buffer, at least to the data point where the error occurred, but the ACK handshake packet will not be sent, and the ERR bit of USBD_ISTR register will be set. In this case, the application program usually does not need to intervene in processing, and the USBD module will be automatically recovered from the transmission error and will get ready for the next transmission 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 of the USBD_EPRx register, and the data will not be written into the receive buffer.

The value of ADDRx_RX determines the initial address of the receive buffer, and COUNTx_RX determines the size of the receive buffer (expected effective data length + 2 bytes 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, and the USBD module will report that the buffer has overflowed, and send a STALL handshake packet to the host, and set the packet buffer overflow flag PMAOVR.

If the transmission is completed correctly, the USBD module will send an ACK handshake packet and write the number of valid data bytes in the actual received data packet into the COUNTx_RX register. The value of the USBD_EPRx register has the following updates: the DTOG_RX bit is flipped, the STAT_RX bit is ‘10’ (NAK status) to invalidate the endpoint, and the CTR_RX bit is set. The application program needs to identify the USB endpoint where the interrupt is generated through the EP_ID and DIR bits of the USBD_ISTR register. For the interrupt service program of the CTR_RX event, firstly 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 program needs to set the STAT_RX bit in USBD_EPRx to ‘11’ (ACK status) to enable the next transmission. When the STAT_RX bit is ‘10’ (NAK status), any OUT request sent to the endpoint will be negatively acknowledged, except for SETUP requests (the protocol specifies that SETUP request must be received as ACK handshake

packet). The PC host will continuously retransmit the OUT transaction packet of NAK until the ACK handshake packet is received from the endpoint.

4) Control Transmission

Control (SETUP) transmission shall occur at endpoint 0, so it is also called endpoint 0 bit control endpoint. The control transmission is composed of 3 phases. The first is the SETUP phase in which the host sends a SETUP transaction, the second is the data phase in which the host sends zero or more data (IN/OUT transaction), and the last is the status phase, which is composed by the data transaction at the direction opposite to the one of the data phase.

SETUP transaction is very similar to the transmission process of OUT transaction, so every time a CTR_RX interrupt occurs, the control endpoint must check the SETUP bit of the USBD_EPRx register 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 for receiving, and ignore the judgment of the content of STAT_RX and DTOG_RX. Then, set DTOG_RX and DTOG_TX to DATA1 status by force, and set STAT_RX and STAT_TX to '10' (NAK) to ensure that the application program can determine whether the subsequent transmission is IN or OUT according to the corresponding data in the SETUP transaction. If the subsequent data transmission is rejected or an error occurs, the application program can set STAT_RX or STAT_TX to '01' and respond to the STALL handshake packet. If the application program receives and processes a SETUP transaction, CTR_RX bit will remain set at this time, another SETUP packet will be received, and the USBD module will discard the SETUP packet and will not provide any handshake packet response to simulate a reception error to force the host to send the SETUP packet again. This is to avoid losing another SETUP transaction transmission following a CTR_RX interrupt.

In the status phase of the control transmission, if the OUT transaction sent by the host to the device is executed, the STATUS_OUT bit (EP_KIND in the USBD_EPRx register) shall be set. Only in this way, the transmission error can be generated when the non-zero-length data packet is received during the status phase transmission. After the status phase transmission is completed, the application program shall clear the STATUS_OUT bit, and set STAT_RX to ACK to indicate that it is ready to receive a new command request. Set STAT_TX to NAK, and no data upload request will be received.

20.2.3 Dual-Buffer Mechanism

In the USB protocol standard, application descriptions are provided for different data transmission methods. The bulk transmission is suitable for mass data transmission between the USB host and the device, and the host uses as much bandwidth as possible to perform the bulk transmission within the frame time. But for this type of transmission, the correctness and integrity of the data shall be guaranteed, so the transmission is carried out in the sequence of token packet, data packet and handshake packet. Synchronous transmission is suitable for data transmission at a constant rate, but has a certain tolerance for errors. It is believed that transmission can generally be successful. The host has a fixed bandwidth to perform synchronous transmission within each frame time to ensure the transmission rate, so the transmission is carried out in sequence of token packet and data packet. There is no handshake packet for verifying the transmission status and terminating the transmission.

One-way Double-buffer Bulk Endpoint

Bulk transmission; when the application program processes the previous data transmission of the bulk endpoint in the single buffer mode and receives a new data packet, the USBD module will respond to the NAK handshake packet to make the PC host continuously retransmit the same data packet until the

application program resets the ACK handshake packet. Such retransmission occupies a lot of bandwidth and affects the rate of bulk transmission. Therefore, a double buffering mechanism is introduced to the bulk endpoint to increase the data transmission rate. In the double buffering mode, the one-way bulk endpoint has two data buffers: data receive and transmit buffers of the endpoint. The data flipping bit (DTOG_RX or DTOG_TX) is used to select which of the two buffers is currently used, so that the application program can operate the other buffer while the USBD module accesses one of the buffers. For example, when the OUT transaction is transmitted to a double-buffer bulk endpoint, the USBD module will save the data from the PC host to a buffer, and the application prorgam can process the data in the other buffer (for IN transactions, the situation is the same). In this way, the data processing of the application program is completed within the time of receiving or sending data of the USBD module, which improves the efficiency of USB transceiving. Because two buffers are required for one transmission direction, the bulk endpoint of the bidirectional buffer must be configured as a unidirectional endpoint. For the USBD_EPRx register, only the STAT_RX bit (as a double-buffer bulk receiving endpoint) or the STAT_TX bit (as a double-buffer bulk transmission endpoint) needs to be set. In order to utilize the advantages of double buffering as much as possible to achieve a high transmission rate, the USBD module processes the flow control of double-buffer bulk endpoint differently from other endpoints to some extent. It only sets the endpoint to NAK status when an access conflict occurs in the buffer, instead of setting the endpoint to NAK status after each successful transmission.

The DTOG_xx bits in the USBD_EPRx register are used to identify the storage buffers currently used by the USBD module and the application program to avoid access conflicts. When configured to send a dual-buffer endpoint at one direction, DTOG_TX will identify the buffer currently used by the USBD module, and DTOG_RX will identify the buffer currently used by the application program; when configured to receive a double-buffer-area endpoint at one direction, DTOG_RX will identify the buffer currently used by the USBD module, and DTOG_TX will identify the buffer currently used by the application program. We name the buffer identifier used for the USBD module as DTOG, and the buffer identifier used for the application program as SW_BUF. Thus, the double-buffer one-way bulk endpoint identification is defined as follows:

Table 20-1 Buffer Identification

Buffer identification bit	Transmission endpoint	Receiving 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)

Table 20-2 Double-Buffer Bulk Endpoint Buffer

Endpoint type	DTOG	SW_BUF	Buffer used for USBD module	Buffer used for application program
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 the endpoint to NAK status	ADDRx_TX_0/COUNTx_TX_0
	1	1	Set the endpoint to NAK status	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 the endpoint to NAK status	ADDRx_RX_0/COUNTx_RX_0
	1	1	Set the endpoint to NAK status	ADDRx_RX_1/COUNTx_RX_1

To configure a double-buffer bulk endpoint for the application program, the EPTYPE[1:0] of the

USBD_EPRx register needs to be set to ‘00’ and the EP_KIND bit needs to be set to ‘1’. Initialize the DTOG and SW_BUF bits according to the buffer used at the beginning of the transmission. After each successful completion of a transmission, the USBD module will control the flow according to the double-buffer bulk endpoint and continuously control it until EP_KIND becomes invalid. At the end of each transmission, the CTR_RX bit or CTR_TX bit will be set according to the transmission direction of the endpoint. Meanwhile, the hardware will set the corresponding DTOG_xx bit (flip) and realize buffer exchange. If there is no buffer access conflict between the USBD module and the application program (i.e., DTOG and SW_BUF are the same value, see Table 154), then keep the status value of the STAT_xx bit. Otherwise, it will be set to '10' (NAK status). Therefore, after the application program accesses the buffer, it needs to flip the SW_BUF bit to notify the USB module that the buffer has become available.

Synchronous endpoint

Synchronous transmission is generally used to transmit audio streams, compressed video streams, and other data that have strict requirements on data transmission rate. The endpoint that synchronous transmission is executed is the synchronous endpoint. The USB host will allocate a fixed bandwidth to the synchronous endpoint for IN transaction or OUT transaction transmission within each frame time, and there is no retransmission mechanism, no handshake protocol, and the PID of the transmitted data packet is always DATA0, and DATA0 and DATA1 data flipping mechanism will not appear (Appearing in the control/bulk/interrupt transmission).

Because there is no handshake mechanism in synchronous transmission, the STAT_RX and STAT_TX bits of the USBD_EPRx register can only be set to two statuses: ‘00’ (transmission disabled) and ‘11’ (running transmission) respectively. Synchronous transmission simplifies the software process with a double buffer mechanism. It also uses two buffers to ensure that the application program can access the other buffer when the USB module uses one buffer. Different from the double-buffering mechanism of one-way bulk endpoint, the synchronous endpoint has fixed time interval for transmission in the USB standard and has fault tolerance, so the USBD module does not judge the conflict with the application area buffer, and only uses the DTOG bit to identify the current buffer used (the DTOG_RX bit in the USBD_EPRx register is used to identify the synchronous endpoint for receiving, and the DTOG_TX bit is used to identify the transmission synchronous endpoint).

Table 20-3 Synchronous Endpoint Buffer Identification

Endpoint type	DTOG	Buffer used for USBD module	Buffer used for application program
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

The application program configures a synchronous endpoint and needs to set the EPTYPE[1:0] of the USBD_EPRx register to ‘10’. Initialize the DTOG bit according to the buffer used at the beginning of the transmission. After each successful completion of a transmission, the CTR_RX bit or CTR_TX bit will be set according to the transmission direction of the endpoint. At the same time, the hardware will set the corresponding DTOG_xx bit (flip) to achieve buffer exchange, but will not change the expected or transmitted data packet PID (fixed to DATA0). The STAT_RX or STAT_TX bit will not change. In synchronous transmission, even if a CRC error or buffer overflow occurs in the OUT transaction, this transmission will be still regarded as correct and can trigger the CTR_RX interrupt event. However, when a

CRC error occurs, the hardware will set the ERR bit in the USB_ISTR register to remind the application program data may be damaged.

20.2.4 Suspension /Wake-up Process

A bus status is defined in the USB standard- bus suspension. If the USB bus has no activity within 3ms, it will enter the suspended state. In this status, the current provided on the USB bus will be reduced (generally not exceeding 500uA for low-speed device, and not exceeding 2.5mA for high-speed device or the device supporting remote wake-up function). This current limit is essential for bus-powered USB devices, while self-powered devices do not need to strictly comply with such current consumption limits.

Under normal working conditions, the USB host will send SOF packets at an interval of 1ms, so if the USBD module detects 3 consecutive SOF packet loss events, it can determine that the host has issued a suspension request. At this time, it will set the SUSP bit of USBD_ISTR register. If the interrupt is enabled, the suspension interrupt will be triggered. The USBD module will continuously detect the suspension status of the bus and update the SUSP bit (The cleared SUSP bit flag in the suspension status of the bus will still be set again by the hardware). So the application needs to perform the following process when receiving the USB bus suspension event:

1) Set the FSUSP bit of the USBD_CNTR register to 1, shield the hardware suspension status detection, and prevent the suspension event from being triggered continuously.
2) Eliminate or reduce the static current consumption of modules other than the USBD module.
3) Set the LPMODE bit of the USBD_CNTR register to 1, so that the USBD module is at a low-power operation status, but the bus wake-up signal can still be detected.
4) You can choose to disable external oscillator and PLL to stop any activity of the device.

The USB device or host in the suspension status will be woken up by the "wake-up" sequence. The so-called "wake up" sequence can be initiated by the USB host to wake up the suspended USB device, or triggered by the USB device to wake up the suspended USB host, but the USB host finally ends the "wake-up" sequence. In addition, the suspended USB device needs to be capable of detecting the function of the RESET signal (bus reset) and performing it as a normal reset operation.

The suspended USBD module will trigger a WKUP interrupt event (channel 42) after receiving the wake-up signal, set the WKUP bit of the USBD_ISTR register to 1 and automatically clear the LPMODE bit. When the application program receives the USB wake-up event, it needs to perform the following process:

1) Clear the FSUSP bit in the USBD_CNTR register, and restart the suspension status detection function of USB bus;
2) You may select to start the external oscillator and PLL.
3) Query the RXDP and RXDM bits of the USBD_FNR register to determine what triggered the wake-up event, and carry out the corresponding software operation.

The USBD module can issue a wake-up sequence to wake up the suspended USB host. In this case, firstly set the RESUME bit of the USBD_CNTR register to 1, and then clear it to 0 within 1ms-15ms to start the wake-up sequence. After the RESUME bit is cleared, the wake-up process will be completed by the host PC (the USB host will continuously execute this sequence to wake up other mounted USB devices). The application program can query the RXDP and RXDM bits of the USBD_FNR register to determine whether the wake-up is complete.

Note: Only when the USBD module is set to the suspension state (set the FSUSP bit of the USB_CNTR register to '1'), the RESUME bit can be set.

Table 20-4 USB Bus Status

RXDP	RXDM	Condition	USB bus status
0	0	>10ms	Bus reset
0	1	>1ms (full-speed device)	Wake-up sequence start
		>3ms (low-speed device)	Suspension status
1	0	>3ms (full-speed device)	Suspension status
		>1ms (low-speed device)	Wake-up sequence start
1	1	-	Bus error (or interference)

20.3 Register Description

The USBD module has the following 3 types of registers:

- General register: Related to USBD module control and interrupt correlation, base address 0x40005C00.
- Endpoint register: endpoint configuration, transceiving status correlation and base address 0x40005C00.
- Buffer description register: Related to the data transceiving buffer, the base address 0x40006000.

Table 20-5 USBD General Registers List

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	X
R16_USBD_DADDR	0x40005C4C	USB device address register	0x0000
R16_USBD_BTABLE	0x40005C50	USB packet buffer description table address register	0x0000

Table 20-6 USBD Endpoint Registers List

Name	Access address	Description	Reset value
R16_USBD_EPR0	0x40005C00	USB endpoint configuration register 0	0x0000
R16_USBD_EPR1	0x40005C04	USB endpoint configuration register 1	0x0000
R16_USBD_EPR2	0x40005C08	USB endpoint configuration register 2	0x0000
R16_USBD_EPR3	0x40005C0C	USB endpoint configuration register 3	0x0000
R16_USBD_EPR4	0x40005C10	USB endpoint configuration register 4	0x0000
R16_USBD_EPR5	0x40005C14	USB endpoint configuration register 5	0x0000
R16_USBD_EPR6	0x40005C18	USB endpoint configuration register 6	0x0000
R16_USBD_EPR7	0x40005C1C	USB endpoint configuration register 7	0x0000

Table 20-7 USBD Buffer Description Registers List

Name	Access address	Description	Reset value
R16_USBD_ADDR0_TX	0x40006000+[USBD_BTABLE]	Endpoint transmission buffer address register 0	0x0000
R16_USBD_COUNT0_TX	0x40006004+[USBD_BTABLE]	Endpoint transmission data byte count register 0	0x0000
R16_USBD_ADDR0_RX	0x40006008+[USBD_BTABLE]	Endpoint reception buffer address register 0	0x0000
R16_USBD_COUNT0_RX	0x4000600C+[USBD_BTABLE]	Endpoint reception data	0x0000

		byte count register 0	
R16_USBD_ADDR1_TX	0x40006010+[USBD_BTABLE]	Endpoint transmission buffer address register 1	0x0000
R16_USBD_COUNT1_TX	0x40006014+[USBD_BTABLE]	Endpoint transmission data byte count register 1	0x0000
R16_USBD_ADDR1_RX	0x40006018+[USBD_BTABLE]	Endpoint reception buffer address register 1	0x0000
R16_USBD_COUNT1_RX	0x4000601C+[USBD_BTABLE]	Endpoint reception data byte count register 1	0x0000
R16_USBD_ADDR2_TX	0x40006020+[USBD_BTABLE]	Endpoint transmission buffer address register 2	0x0000
R16_USBD_COUNT2_TX	0x40006024+[USBD_BTABLE]	Endpoint transmission data byte count register 2	0x0000
R16_USBD_ADDR2_RX	0x40006028+[USBD_BTABLE]	Endpoint reception buffer address register 2	0x0000
R16_USBD_COUNT2_RX	0x4000602C+[USBD_BTABLE]	Endpoint reception data byte count register 2	0x0000
R16_USBD_ADDR3_TX	0x40006030+[USBD_BTABLE]	Endpoint transmission buffer address register 3	0x0000
R16_USBD_COUNT3_TX	0x40006034+[USBD_BTABLE]	Endpoint transmission data byte count register 3	0x0000
R16_USBD_ADDR3_RX	0x40006038+[USBD_BTABLE]	Endpoint reception buffer address register 3	0x0000
R16_USBD_COUNT3_RX	0x4000603C+[USBD_BTABLE]	Endpoint reception data byte count register 3	0x0000
R16_USBD_ADDR4_TX	0x40006040+[USBD_BTABLE]	Endpoint transmission buffer address register 4	0x0000
R16_USBD_COUNT4_TX	0x40006044+[USBD_BTABLE]	Endpoint transmission data byte count register 4	0x0000
R16_USBD_ADDR4_RX	0x40006048+[USBD_BTABLE]	Endpoint reception buffer address register 4	0x0000
R16_USBD_COUNT4_RX	0x4000604C+[USBD_BTABLE]	Endpoint reception data byte count register 4	0x0000
R16_USBD_ADDR5_TX	0x40006050+[USBD_BTABLE]	Endpoint transmission buffer address register 5	0x0000
R16_USBD_COUNT5_TX	0x40006054+[USBD_BTABLE]	Endpoint transmission data byte count register 5	0x0000
R16_USBD_ADDR5_RX	0x40006058+[USBD_BTABLE]	Endpoint reception buffer address register 5	0x0000
R16_USBD_COUNT5_RX	0x4000605C+[USBD_BTABLE]	Endpoint reception data byte count register 5	0x0000
R16_USBD_ADDR6_TX	0x40006060+[USBD_BTABLE]	Endpoint transmission buffer address register 6	0x0000
R16_USBD_COUNT6_TX	0x40006064+[USBD_BTABLE]	Endpoint transmission data byte count register 6	0x0000

R16_USBD_ADDR6_RX	0x40006068+[USBD_BTABLE]	Endpoint reception buffer address register 6	0x0000
R16_USBD_COUNT6_RX	0x4000606C+[USBD_BTABLE]	Endpoint reception data byte count register 6	0x0000
R16_USBD_ADDR7_TX	0x40006070+[USBD_BTABLE]	Endpoint transmission buffer address register 7	0x0000
R16_USBD_COUNT7_TX	0x40006074+[USBD_BTABLE]	Endpoint transmission data byte count register 7	0x0000
R16_USBD_ADDR7_RX	0x40006078+[USBD_BTABLE]	Endpoint reception buffer address register 7	0x0000
R16_USBD_COUNT7_RX	0x4000607C+[USBD_BTABLE]	Endpoint reception data byte count register 7	0x0000

Note: The above buffer description registers correspond to endpoint configuration registers during use. For example: USB endpoint configuration register 0 corresponds to endpoint transmission buffer address register 0, endpoint transmission data bytes count register 0, endpoint receive buffer address register 0 and endpoint receiving data bytes count register 0.

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	LP MODE	Reserved	FRES

Bit	Name	Access	Description	Reset value
15	CTRM	RW	Correct transfer interrupt enable bit: 1: Enable correct transfer (CTR) interrupt, and generating an interrupt when the corresponding bit of the interrupt register is set to 1. 0: Disable correct transfer (CTR) interrupt.	0
14	PMAOVRM	RW	Packet buffer overrun interrupt enable bit: 1: Enable PMAOVR interrupt, and generating an interrupt when the corresponding bit of interrupt register is set to 1; 0: Disable PMAOVR interrupt.	0
13	ERRM	RW	Error interrupt enable bit: 1: Enable error interrupt, and generating 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 bit: 1: Enable wake-up interrupt, and generating an interrupt when the corresponding bit of interrupt register is set to 1; 0: Disable wake-up interrupt.	0
11	SUSPM	RW	Suspension interrupt enable bit: 1: Enable the suspension (SUSP) interrupt, and generating an interrupt when the corresponding bit of the interrupt	0

			register is set to 1. 0: Disable the suspension (SUSP) interrupt.	
10	RESETM	RW	USB reset (bus reset or forced reset) interrupt enable bit: 1: Enable USB reset interrupt, and generating an interrupt when the corresponding bit of interrupt register is set to 1; 0: Disable USB reset interrupt.	0
9	SOFM	RW	Frame start (SOF) interrupt enable bit: 1: Enable SOF interrupt, and generating an interrupt when the corresponding bit of the interrupt register is set to 1; 0: Disable SOF interrupt.	0
8	ESOFM	RW	Interrupt enable bit for the timing frame start loss: 1: Enable the ESOF interrupt, and generating an interrupt when the corresponding bit of interrupt register is set to 1; 0: Disable ESOF interrupt.	0
[7:5]	Reserved	RO	Reserved.	0
4	RESUME	RW	Wake-up request control bit: 1: Output wake-up signal; 0: Idle status. 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 be set only when the FSUSP bit is 1.*	0
3	FSUSP	RW	Mask suspension detection control bit: 1: Mask bus suspension status detection. At this time, the clock and static power consumption of the USB analog transceiver are still maintained. If you need to enter low-power status (bus-powered device), you need to set FSUSP bit and then LPMODE bit. 0: Enable bus supension status 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 bit: This mode is used to reduce power consumption when the USB is suspended. In this mode, except for the power supply of the external pull-up resistor, other static power consumption will be turned off, and the system clock will be stopped or reduced to a certain frequency to reduce the power consumption. Activity on the USB bus (wake-up event) will clear this bit (software can also clear this bit). 1: Low-power mode; 0: No low-power mode.	0
1	Reserved	RO	Reserved.	1
0	FRES	RW	Force USB reset control bit: 1: Reset the USBD module by force. The USBD module	1

			will remain in the reset status until the software clears this bit. If the USB reset interrupt is enabled, a reset interrupt will be generated; 0: Clear USB reset.	

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	RO	Correct transfer status indication. This bit is set by hardware after the data transfer is completed correctly at the endpoint. The application program can identify the endpoint that the correct data transfer has been completed through the DIR and EP_ID bits.	0
14	PMAOVR	RW0	Packet buffer overrun flag. This bit is set by hardware when the microcontroller does not respond to a request to access the USB packet buffer for a long time. The USBD module usually sets this bit in the following situations: an ACK handshake packet is not sent in the receiving process, or a bit stuffing error occurs in the transmission process, and the host will require data retransmission in both cases. No PMAOVR interrupt will be generated during normal data transmission. Since the failed transmission will be retransmitted by the host, the application program can accelerate other operations of the device in this interrupted service program and get ready for retransmission. But this interruption will not be generated during synchronous transmission (synchronous transmission does not support retransmission), so data may be lost. This bit is readable; write 0 to clear it; invalid if writing 1.	0
13	ERR	RW0	Error flag; the hardware will set this bit when the following errors occur: NANS: No answer. The host response timeout. CRC: Check error. CRC check error in the USB packet. BST: Bit stuffing error. Bit stuffing error is detected in the USB data bit. FVIO: Frame format error. Receipt of non-standard frames (e.g. EOP appears at the wrong time; wrong token). The USB application program can usually ignore these errors, because the USBD module and the host will start the retransmission mechanism when an error occurs. The	0

			interrupt generated by this bit can be used in the development phase of the application program, can be used to monitor the transmission quality of the USB bus, and identify the errors that may occur to the user (loose connection line, serious environmental interference and damaged USB line). This bit is readable; write 0 to clear it; invalid if writing 1.	
12	WKUP	RW0	Wake-up signal flag: When the USBD module is in the suspension status, if a wake-up signal is detected, this bit will be set by hardware. At this time, the LP_MODE bit of the CTLR register will be cleared to 0, and the FSUSP bit needs to be cleared to 0 by software to enable suspension detection. At the same time, USB_WAKEUP is activated, notifying other parts of the device (such as the wake-up unit) to start the wake-up process. This bit is readable; write 0 to clear it; invalid if writing 1.	0
11	SUSP	RW0	Bus suspension 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 will immediately enable the detection of the suspension signal, but the hardware will not detect the suspension signal in the suspend mode (FSUSP=1) until the wake-up process ends. This bit is readable; write 0 to clear it; invalid if writing 1.	0
10	RESET	RW0	USB reset (bus reset or forced reset) flag: This bit is set by hardware when the USBD module detects the USB bus reset signal edge or forced reset status. At this time, the USBD module will reset the internal protocol status device and trigger the reset interrupt to respond when the interrupt is enabled. The transmission and receiving parts of the USBD module will be disabled until this bit is cleared. All configuration registers will not be reset unless the application program clears them. This is used to ensure that the USB transmission can be correctly executed immediately after the reset is cancelled. But the address and endpoint register of the device will be reset by USB. This bit is readable; write 0 to clear it; invalid if writing 1.	0
9	SOF	RW0	Frame start (SOF) flag: This bit is set by hardware when the USBD module detects the SOF packet on the bus. The interrupt service program can complete the 1ms synchronization with the host by detecting the SOF event, and correctly read the updated content of the register when the SOF is received (this function is very meaningful during synchronous	0

			transmission). This bit is readable; write 0 to clear it; invalid if writing 1.	
8	ESOF	RW0	Timing start of frame (ESOF) loss flag: This bit is set by hardware when the USBD module does not receive the SOF packet on time. The host shall send SOF packet every millisecond, but if the USBD module does not receive it, the suspended timer will trigger this interrupt. If three consecutive ESOF interrupts occur, i.e., if no SOF packet is received for three consecutive times, a SUSP interrupt will be generated. This bit is readable; write 0 to clear it; invalid if writing 1.	0
[7:5]	Reserved	RO	Reserved.	0
4	DIR	RO	Transaction data transmission direction. This bit is written by the hardware according to the transmission direction after the data transmission is completed and interrupt is generated. If DIR=0, the CTR_TX bit at the corresponding endpoint will be set, symbolizing the completion of an IN transaction transmission (data transmission from the USBD module to the PC host). If DIR=1, the CTR_RX bit of the corresponding endpoint will be set, symbolizing the completion of an OUT transaction transmission (data transmission 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 pending OUT transactions and IN transactions at the same time. The application program can use this information to access the operation corresponding to the USBD_EPnR bit, which indicates the information about the direction of the suspended interrupt transmission.	0
[3:0]	EP_ID	RO	Endpoint No. This bit is written by hardware according to endpoint number of the request interrupt after USBD module completes data transmission and generates an interrupt. If there are request interrupts at multiple endpoints, hardware will write the endpoint number with the highest priority. The priority of the endpoint is defined as follows: the synchronous endpoint and the double-buffer bulk endpoint have high priority, and the other endpoints have low priority. If multiple endpoints at the same priority request an interrupt, the priority will be determined according to the endpoint number, i.e., endpoint 0 has the highest priority. The smaller number of endpoint means the higher priority. The application program can process the interrupt request of endpoint according to this priority scheme.	0

USB frame 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 bit. The USBD module will detect SOF packet after the reset or wake-up sequence ends. If at least 2 SOF packets are continuously detected, the hardware will set this bit. Once this bit is locked, the frame counter will stop counting and resume counting when the USBD module is reset or the bus is suspended.	0
[12:11]	LSOF	RO	Frame start loss flag bit. When the ESOF event occurs, the hardware will write the number of lost SOF packets into this domain. If the SOF packet is received again, this domain will be cleared.	X
[10:0]	FN	RO	Frame number. This domain is the 11-bit frame number in the lately received SOF packet. Each time the host sends a frame, the frame number will be self-increased, which is very meaningful for the synchronous transmission. This domain is updated in the event of SOF interrupt.	X

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 bit. This bit is set by the application program when the USB device function needs to be enabled. If this bit is 0, the USBD module will stop working, ignore all register settings, and will not respond to any USB communication. 1: Enable USB device function; 0: Stop USB device function.	0
[6:0]	ADD	RW	USB device address. This domain is the address value assigned by the USB host	0

			to the USB device in the enumeration process. The address value and the EA bit must match the address information in the USB token packet in order to perform correct USB transmission at the specified endpoint.	

USB packet buffer description table address register (USBD_BTABLE)

Offset address: 0x4C

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	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

USB endpoint configuration register x (USBD_EPRx) (x=0/1/2/3/4/5/6/7)

Offset address: 0x00-0x1C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTR_RX	DTOG_RX	STAT_RX[1:0]		STEUP	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 bit (OUT/SETUP). This bit is set by hardware when the OUT or SETUP transaction is correctly received (ACK is sent). If the CTRM bit is set, the corresponding interrupt will be generated. The application program needs to clear this bit after processing the event. The reception of OUT transaction or SETUP transaction can be determined through the following SETUP bit. This bit is readable; write 0 to clear it; invalid if writing 1. *Note: This bit will not be set for transmission of transactions responded by NAK or STALL or error.*	0
14	DTOG_RX	RW1T	Data packet PID(OUT/SETUP) expected to be received next time; set by hardware: 1: Expected DATA1;	0

			0: Expected DATA0. For asynchronous endpoints, after receiving the correct PID data packet, USBD module will send an ACK handshake packet, and the hardware will automatically flip this bit. For control endpoint, the hardware sets the bit (DATA1) after receiving the correct SETUP packet. For the endpoint with the double-buffer attribute, the hardware not only automatically flips this bit to indicate the expected packet PID, but also supports the exchange of double-buffers based on this bit identification (please refer to the description in the double-buffer mechanism). For the synchronous endpoint, the hardware does not judge the PID of the data packet, but only supports the exchange of double buffers through this bit. This bit is readable; invalid if writing 0; write 1 to flip. *Note: The application program can set the initial value of this bit, or flip this bit for special purpose.*	
[13:12]	STAT_RX	RW1T	Status bit of data received (in OUT/SETUP transmission): 00: DISABLED; the endpoint ignores all receive requests and does not respond; 01: STALL; the endpoint responds to the receive request with STALL packet; 10: NAK; the endpoint responds to the receive request with NAK packet; 11: ACK; the endpoint responds to the receive request with ACK packet. When a correct OUT or SETUP data transmission is completed (CTR_RX=1), the hardware will automatically set this bit to NAK status, so that the application program has enough time to process and respond to the next transaction. For the double-buffer bulk endpoint, due to the use of a special transmission flow control strategy, the transmission status will be controlled according to the buffer status used (please refer to double-buffer endpoint). For the synchronous endpoint, since the endpoint status can only be valid or disabled, the hardware will not set this bit after the correct transmission. If the domain is set to STALL or NAK, the operation responded by the USBD module will be undefined. This domain is readable; invalid if writing 0 at the bit; write 1 to flip. *Note: The application program can set the initial value of the domain.*	0
11	STEUP	RO	SETUP transaction transmission completion flag bit: 1: It is SETUP transaction and received correctly	0

<table>
<tr><td></td><td></td><td></td><td>(transmitting ACK);
0: Non-SETUP transaction.
Note: The hardware may modify this bit only when CTR_RX=0.</td><td></td></tr>
<tr><td>[10:9]</td><td>EPTYPE</td><td>RW</td><td>Transmission endpoint types:
00: BULK, bulk endpoint;
01: CONTROL, control endpoint;
10: ISO, synchronous endpoint;
11: INTERRUPT, interrupt endpoint.
Only control endpoints will have SETUP transmission, and other types of endpoints ignore this type of transmission. SETUP transmission cannot be responded with NAK or STALL packet. If the control endpoint is in NAK status when receiving the SETUP packet, the USBD module will not respond to the request, and a receiving error will occur. If the control endpoint is at the STALL status, the SETUP packet will be received correctly, the data will be correctly transmitted, and a correct transmission completion interrupt will be generated. The OUT packet of the control endpoint is processed in the same way as a normal endpoint.
The processing methods of bulk endpoints and interrupt endpoints are very similar, except for the processing of EP_KIND bits.</td><td>0</td></tr>
<tr><td>8</td><td>EP_KIND</td><td>RW</td><td>Endpoint special type control bit (used with EP_TYPE):
<table>
<tr><th>EPTYPE[1:0]</th><th>EP_KIND</th></tr>
<tr><td>BULK</td><td>DBL_BUF: Enable double buffers.</td></tr>
<tr><td>CONTROL</td><td>STATUS_OUT: Control the data packet length judgment during the transmission status.</td></tr>
<tr><td>ISO</td><td>Not used.</td></tr>
<tr><td>INTERRUTP</td><td>Not used.</td></tr>
</table>
DBL_BUF: Set this bit to enable double-buffer mode of bulk endpoint.
STATUS_OUT: Set this bit to indicate that the USB device expects the host to send the status phase transaction in the control transmission. At this time, the device responds to the STALL handshake packet for any data packet whose length is not 0. (This function is only used to control the endpoint, which helps to provide detection of protocol layer errors.) If the STATUS_OUT bit is cleared, the OUT transaction in the status phase can contain data in any length.</td><td>0</td></tr>
<tr><td>7</td><td>CTR_TX</td><td>RW0</td><td>Correct transmission flag bit (IN):
This bit is set by hardware when the correct IN transaction (ACK is received) is completed. If the CTRM bit is set, the corresponding interrupt will be generated. The application</td><td>0</td></tr>
</table>

			program needs to clear this bit after processing the event. At the end of the IN packet, if the host responds to NAK or STALL, this bit will not be set because the data transmission is not successful. This bit is readable; write 0 to clear it; invalid if writing 1. *Note: If the host responds with NAK or STALL, this bit will not be set.*	
6	DTOG_TX	RW1T	PID (IN) of data to be transmitted, set by hardware: 1: Transmit DATA1 ; 0: Transmit DATA 0. For asynchronous endpoints, after the correct PID data packet is received, the hardware will automatically flip this bit if the USBD module receives an ACK handshake packet of the host. For the control endpoint, the hardware sets the bit (DATA1) after receiving the correct SETUP packet. For the endpoint with the double-buffer attribute, the hardware not only automatically flips this bit to indicate sending the packet PID, but also supports the exchange of double-buffers based on this bit identification (please refer to the description in the double-buffer mechanism). For synchronous endpoint, the hardware sends the data packet DATA0 by force, and supports the exchange of double buffers through this bit identification. This bit is readable; invalid if writing 0; write 1 to flip. *Note: The application program can set the initial value of this bit, or flip this bit for special purpose.*	0
[5:4]	STAT_TX	RW1T	Status bit of data transmission: 00: DISABLED; the endpoint ignores all transmission requests and does not respond; 01: STALL; the endpoint responds to the host IN request with STALL packet; 10: NAK; the endpoint responds to the host IN request with NAK packet; 11: ACK; the data can be sent through this endpoint. When the data transmission of an IN transaction is completed correctly (CTR_TX=1), the hardware will automatically set this bit to NAK status to ensure that the application has enough time to process and respond to the next transaction transmission. For the double-buffer bulk endpoint, due to the use of a special transmission flow control strategy, the transmission status will be controlled according to the buffer status used (please refer to double-buffer endpoint). For the synchronous endpoint, since the endpoint status can only be valid or disabled, the hardware will not set this bit	0

			after the correct transmission. If the domain is set to STALL or NAK, the operation responded by the USBD module will be undefined. This domain is readable; invalid if writing 0 at the bit; write 1 to flip. *Note: The application program can set the initial value of the domain.*	
[3:0]	EA	RW	Endpoint address domain (setting the endpoint number): The application program needs to set an endpoint address for this endpoint configuration register.	0

Endpoint transmission 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	The initial address of the buffer where the data is to be sent (during IN transaction).	0
0	-	RZ	The buffer address must be aligned with 2 bytes, so this bit must be 0.	0

Endpoint transmission data 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	RW	The length and bytes count of the data to be sent (during the IN transaction).	0

Note: There are 2 USBD_ADDRx_TX registers and 2 USB_COUNTx_TX registers for the double-buffer and synchronous IN endpoint: USBD_ADDRx_TX_1 and USBD_ADDRx_TX_0, USB_COUNTx_TX_1 and USB_COUNTx_TX_0, and 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

Endpoint reception 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	RW	The initial ddress of the buffer where the data is to be received (during the OUT or SETUP transaction).	0
0	-	RZ	The buffer address must be aligned with 2 bytes, so this bit must be 0.	0

Endpoint reception data byte count register x (USBD_COUNTx_RX) (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
BLSIZE	NUM_BLOCK[4:0]					COUNTx_RX[9:0]									

Bit	Name	Access	Description	Reset value
15	BLSIZE	RW	Storage block size: 1: The block size is 2 bytes, and the block is used with NUM_BLOCK; the distributable receive buffer range is 2-62 bytes; 0: The block size is 32 bytes, and the block is used with NUM_BLOCK; the distributable receive buffer range is 32-512 bytes.	0
[14:10]	NUM_BLOCK	RW	Number of storage blocks.	0
[9:0]	COUNTx_RX	RO	Length and bytes count of data actually received at the endpoint (during OUT or SETUP transaction).	X

Note: There are 2 USBD_ADDRx_RX registers and 2 USB_COUNTx_RX registers for the double-buffer and synchronous IN endpoints: USBD_ADDRx_RX_1 and USBD_ADDRx_RX_0, USB_COUNTx_RX_1 and USB_COUNTx_RX_0, and 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 higher 6 bits of the USBD_COUNTx_RX register define the size of the receiving 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 in the device enumeration process.

Table 20-8 Buffer Size Definition

NUM_BLOCK[4:0]	Receive buffer size limit	
	BLSIZE = 0	BLSIZE = 1
00000	Not allowed to be used	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 21 USB Full Speed Host/ Device Controller (USBHD)

The module description in this chapter applies to the full range of CH32F103 and CH32V103 microcontrollers.

21.1 USB Controller Introduction

The chip is embedded with a USB master-slave controller and transceiver, the features are as follows:

- Support USB Host function and USB Device function.
- Support USB2.0 full-speed 12Mbps or low-speed 1.5Mbps.
- Support USB control transmission, bulk transmission and interrupt transmission, synchronous/real-time transmission.
- Support data packets up to 64 bytes, built-in FIFO, supporting interrupt and DMA.

21.2 Register Description

USB related registers are divided into 3 parts, some of which are multiplexed in host and device modes.

- USB global register
- USB device control register
- USB host control register

21.2.1 Global Register Description

Table 21-1 List of USBHD Related Register (those marked in grey are controlled by RB_UC_RESET_SIE reset)

Name	Access address	Description	Reset value
R8_USB_CTRL	0x40023400	USB control register	0x06
R8_USB_INT_EN	0x40023402	USB interrupt enable register	0x00
R8_USB_DEV_AD	0x40023403	USB device address register	0x00
R32_USB_STATUS	0x40023404	USB status register	0xXX20XXXX
R8_USB_MIS_ST	0x40023405	USB miscellaneous status register	0xXX
R8_USB_INT_FG	0x40023406	USB interrupt flag register	0x20
R8_USB_INT_ST	0x40023407	USB interrupt status register	0xXX
R8_USB_RX_LEN	0x40023408	USB receive length register	0xXX

USB control register (R8_USB_CTRL)

Bit	Name	Access	Description	Reset value
7	RB_UC_HOST_MODE	RW	USB working mode selection bit: 1: Host mode; 0: Device mode.	0
6	RB_UC_LOW_SPEED	RW	USB bus signal transmission rate selection bit: 1: 1.5Mbps; 0:12Mbps.	0
5	RB_UC_DEV_PU_EN	RW	USB device enable and internal pull-up resistor control bit in USB device mode, if it is 1, USB device transmission will be enabled	0

			and internal pull-up resistor will be enabled.	
[5:4]	MASK_UC_SYS_CTRL	RW	See the table below to configure USB system.	0
3	RB_UC_INT_BUSY	RW	Auto-suspend enable bit before USB transmission completion interrupt flag is not cleared: 1: It will automatically suspend before interrupt flag UIF_TRANSFER is not cleared. In device mode, it will automatically respond to busy NAK and will automatically suspend subsequent transmission in host mode; 0: Not suspend.	0
2	RB_UC_RESET_SIE	RW	Software reset control bit of USB protocol processor: 1: Forced to reset USB protocol processor (SIE); it needs to be cleared by software; 0: Not reset.	1
1	RB_UC_CLR_ALL	RW	USB FIFO and interrupt flag clear: 1: Cleared by force; 0: Not clear.	1
0	RB_UC_DMA_EN	RW	DMA and DMA interrupt control bit of USB: 1: Enable DMA mode and DMA interrupt; 0: Disable DMA.	0

RB_UC_HOST_MODE and MASK_UC_SYS_CTRL constitute the USB system control combination:

Table 21-2 USB System Control Combination

RB_UC_HOST_MODE	MASK_UC_SYS_CTRL	USB system control description
0	00	Disable the USB device function and disable the pull-up resistor.
0	01	Enable the USB device function and disable the internal pull-up resistor. The external pull-up is needed.
0	1x	Enable the USB device function and enable the internal 1.5K pull-up resistor. The pull-up resistor has priority over the pull-down resistor, and can also be used in GPIO mode.
1	00	Normal working state in USB host mode.
1	01	In USB host mode, DP/DM is forced to output SE0 status.
1	10	In USB host mode, DP/DM is forced to output J status.
1	11	In USB host mode, DP/DM is forced to output K status/wake-up.

USB interrupt enable register (R8_USB_INT_EN)

Bit	Name	Access	Description	Reset value
7	RB_UIE_DEV_SOF	RW	In USB device mode, receive SOF packet interrupt: 1: Enable interrupt; 0: Disable interrupt.	0
6	RB_UIE_DEV_NAK	RW	In 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 overrun interrupt: 1: Enable interrupt; 0: Disable interrupt.	0
3	RB_UIE_HST_SOF	RW	In USB host mode, SOF timing interrupt: 1: Enable interrupt; 0: Disable interrupt.	0
2	RB_UIE_SUSPEND	RW	USB bus suspend/wake-up event interrupt: 1: Enable interrupt; 0: Disable interrupt.	0
1	RB_UIE_TRANSFER	RW	USB transfer completion interrupt: 1: Enable interrupt; 0: Disable interrupt.	0
0	RB_UIE_DETECT	RW	In USB host mode, USB device connection or disconnection event interrupt: 1: Enable interrupt; 0: Disable interrupt.	0
	RB_UIE_BUS_RST	RW	In USB device mode, USB bus reset event interrupt: 1: Enable interrupt; 0: Disable interrupt.	0

USB device address register (R8_USB_DEV_AD)

Bit	Name	Access	Description	Reset value
7	RB_UDA_GP_BIT	RW	USB general flag, user self-defined.	0
[6:0]	MASK_USB_ADDR	RW	Host mode: address of USB device currently operated; Device mode: the address of the USB itself.	0

USB miscellaneous status register (R8_USB_MIS_ST)

Bit	Name	Access	Description	Reset value
7	RB_UMS_SOF_PRES	RO	SOF packet indicator status bit in USB host mode: 1: SOF packet will be sent, and it will be automatically delayed if there are other USB data packets; 0: No SOF package is sent.	X
6	RB_UMS_SOF_ACT	RO	SOF packet transmission status bit in USB host mode: 1: SOF packet is being sent out; 0: The transmission is completed or idle.	X
5	RB_UMS_SIE_FREE	RO	Idle status bit of USB protocol processor: 1: Idle protocol processor;	1

			0: Busy; USB transmission is in progress.	
4	RB_UMS_R_FIFO_RDY	RO	USB receiver FIFO data ready status bit: 1: Receiver FIFO is not empty; 0: Receiver FIFO is empty.	0
3	RB_UMS_BUS_RESET	RO	USB bus reset status bit: 1: The current USB bus is at reset status; 0: The current USB bus is not at reset status.	X
2	RB_UMS_SUSPEND	RO	USB suspension status bit: 1: The USB bus is in suspended state, and there is no USB activity for a period of time; 0: USB bus is not at suspended status.	0
1	RB_UMS_DM_LEVEL	RO	In USB host mode, the level status 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 connection status bit of the port in USB host mode: 1: Port has been connected to USB device; 0: No USB device is connected to the port.	0

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 acknowledge status bit: 1: NAK during current USB transmission; 0: No NAK.	0
6	RB_U_TOG_OK	RO	Current USB transmit DATA0/1 synchronous flag match status bit: 1: Synchronous; 0: Asynchronous.	0
5	RB_U_SIE_FREE	RO	USB protocol processor idle status bit: 1: Idle USB; 0: Busy; USB transmission is in progress.	1
4	RB_UIF_FIFO_OV	RW	USB FIFO overrun interrupt flag bit; write 1 to clear it: 1: FIFO overrun trigger; 0: No event.	0
3	RB_UIF_HST_SOF	RW	SOF timing interrupt flag bit in USB host mode; cleared by writing 1: 1: Triggered after the completion of SOF packet transmission; 0: No event.	0
2	RB_UI F_SUSPEND	RW	USB bus suspend/wake-up event interrupt flag bit; write 1 to clear it:	0

			1: USB suspend/wake-up event trigger; 0: No event.	
1	RB_UIF_TRANSFER	RW	USB transfer completion interrupt flag bit, write 1 to clear it: 1: A USB transmission completion trigger; 0: No event.	0
0	RB_UIF_DETECT	RW	In the USB host mode, the USB device connection or disconnection event interrupt flag bit; write 1 to clear it: 1: USB device connection or disconnection trigger is detected; 0: No event.	0
	RB_UIF_BUS_RST	RW	USB bus reset event interrupt flag bit in USB device mode; write 1 to clear it: 1: USB bus reset event trigger; 0: No event.	0

USB interrupt flag register (R8_USB_INT_ST)

Bit	Name	Access	Description	Reset value
7	RB_UIS_IS_NAK	RO	In USB device mode, NAK status bit, the same as RB_U_IS_NAK: 1: NAK during the current USB transmission; 0: No NAK acknowledge.	0
6	RB_UIS_TOG_OK	RO	Current USB transmit DATA0/1 synchronous flag match status bit, the same as RB_U_TOG_OK: 1: Synchronous; 0: Asynchronous.	0
[5:4]	MASK_UIS_TOKEN	RO	In device mode, the token PID 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 the host mode, the response PID of the current USB transmission transaction is identified. 0000: Device has no response or timeout; Others: Respond PID.	XXXXb

MASK_UIS_TOKEN is used to identify the token PID of the current USB transmission 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 host mode, if the host sends an OUT/SETUP token packet, the PID will be the handshake packet ACK/NAK/STALL, or the device has no response/timeout. If the host sends an IN token packet, the PID will the PID of the data packet (DATA0/DATA1) or the handshake packet PID.

USB receive length register (R8_USB_RX_LEN)

Bit	Name	Access	Description	Reset value
[9:0]	R16_USB_RX_LEN	RO	The number of bytes of the data received by the current USB endpoint (applied to V103).	X
[7:0]	R8_USB_RX_LEN	RO	The number of bytes of the data received by the current USB endpoint (applied to F103).	X

21.2.2 Device Register Description

In USB device mode, USBHD module of CH32F103x is equipped with 5 sets of bidirectional endpoints, including endpoint0, endpoint1, endpoint2, endpoint3 and endpoint4. The maximum data packet length of all endpoints is 64 bytes.

- Endpoint 0 is the default endpoint and supports control transmission. Transmission and receiving share a 64-byte data buffer.
- Endpoint 1, endpoint 2, endpoint 3 each includes a transmission endpoint IN and a reception endpoint OUT. The transmission endpoint and reception endpoint each has a separate 64 bytes or double 64 bytes data buffer, support bulk transmission, interrupt transmission, and real-time/synchronous transmission.
- Endpoint 4 includes a transmission endpoint IN and a reception endpoint OUT. The transmission endpoint and reception endpoint each has a separate 64 bytes data buffer, support bulk transmission, interrupt transmission, and real-time/synchronous transmission.

In the USB device mode, the USBHD module of CH32V103x series products is equipped with 8 sets of bidirectional endpoint configuration registers with endpoint numbers 0-7, which can be mapped to the configuration of endpoint numbers 8-15, and the maximum data packet length of all endpoints is 64 bytes .

- Endpoint 0 is the default endpoint, which supports control transmission. The transmission and receiving share a 64-byte data buffer
- Endpoint1 ~ endpoint15 can be configured with independent 64-byte transmission and reception buffers or double-64-byte data buffer, and bulk transmission, interrupt transmission and real-time/synchronous transmission are supported.

Each group of endpoints has a control register R8_UEPn_CTRL and a length transmit register R8_UEPn_T_LEN, which are used to set the synchronization trigger bit of endpoint, the response to OUT transactions and IN transactions and the length of data to be sent.

As the necessary USB bus pull-up resistor of USB device, it can be set whether to be enabled by the software at any time. When RB_UC_DEV_PU_EN in the R8_USB_CTRL is set to 1, the controller will set according to the speed of RB_UD_LOW_SPEED, internally connect the pull-up resistor with the DP/DM pin of the USB bus and enable the USB device function.

When a USB bus reset, USB bus suspend/wake-up event is detected, or when USB successfully processes data sending or receiving, the USB protocol processor will set corresponding interrupt flag. If the interrupt enable is switched on, the corresponding interrupt request will be also generated. The application program can directly query or query and analyze the interrupt flag register R8_USB_INT_FG in the USB interrupt service program, and perform corresponding processing according to RB_UIF_BUS_RST and RB_UIF_SUSPEND. In addition, if RB_UIF_TRANSFER is valid, it is required to continue to analyze the USB interrupt state register R8_USB_INT_ST, and perform the corresponding processing according to the current endpoint number MASK_UIS_ENDP and the current transaction token PID identifier

MASK_UIS_TOKEN. If the synchronization trigger bit RB_UEP_R_TOG of OUT transaction of each endpoint is set in advance, you can judge whether the synchronization trigger bit of the data packet received matches the synchronization trigger bit of the endpoint through RB_U_TOG_OK or RB_UIS_TOG_OK; if the data is synchronized, the data is valid; if the data is not synchronized, the data should be discarded. After the USB sending or receiving interrupt is processed each time, the synchronization trigger bit of corresponding 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, RB_UEP_AUTO_TOG can be set to automatically flip the corresponding synchronization trigger bit after sending or receiving successfully.

The data to be sent by each endpoint is in their 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 their own buffer, but the length of the data received is in the USB length receiving register R8_USB_RX_LEN, and it can be distinguished according to the current endpoint number when the USB is receiving an interrupt.

Table 21-3 List of Device Related Registers (those marked in grey are controlled by RB_UC_RESET_SIE reset) (applied to F103)

Name	Access address	Description	Reset value
R8_UDEV_CTRL	0x40023401	USB device physical port control register	0xX0
R8_UEP4_1_MOD	0x4002340c	Endpoint 1/4 mode control register	0x00
R8_UEP2_3_MOD	0x4002340d	Endpoint 2/3 mode control register	0x00
R16_UEP0_DMA	0x40023410	Start address of endpoint 0 buffer	0xXXXX
R16_UEP1_DMA	0x40023414	Start address of endpoint 1 buffer	0xXXXX
R16_UEP2_DMA	0x40023418	Start address of endpoint 2 buffer	0xXXXX
R16_UEP3_DMA	0x4002341c	Start address of endpoint 3 buffer	0xXXXX
R8_UEP0_T_LEN	0x40023420	Endpoint 0 transmission length register	0xXX
R8_UEP0_CTRL	0x40023422	Endpoint 0 control register	0x00
R8_UEP1_T_LEN	0x40023424	Endpoint 1 transmit length register	0xXX
R8_UEP1_CTRL	0x40023426	Endpoint 1 control register	0x00
R8_UEP2_T_LEN	0x40023428	Endpoint 2 transmit length register	0xXX
R8_UEP2_CTRL	0x4002342a	Endpoint 2 control register	0x00
R8_UEP3_T_LEN	0x4002342c	Endpoint 3 transmit length register	0xXX
R8_UEP3_CTRL	0x4002342e	Endpoint 3 control register	0x00
R8_UEP4_T_LEN	0x40023430	Endpoint 4 transmit length register	0xXX
R8_UEP4_CTRL	0x40023432	Endpoint 4 control register	0x00

Table 21-4 List of Device Related Registers (those marked in grey are controlled by RB_UC_RESET_SIE reset) (applied to V103)

Name	Access address	Description	Reset value
R8_UDEV_CTRL	0x40023401	USB device physical port control register	0xX0
R8_UEP4_1_MOD	0x4002340c	Endpoint 1(9)/4(8/12) mode control register	0x00
R8_UEP2_3_MOD	0x4002340d	Endpoint 2(10)/3(11) mode control register	0x00
R8_UEP5_6_MOD	0x4002340e	Endpoint 5(13)/6(14) mode control register	0x00
R8_UEP7_MOD	0x4002340f	Endpoint 7 (15) mode control register	0x00
R16_UEP0_DMA	0x40023410	Start address of endpoint 0 buffer	0xXXXX
R16_UEP1_DMA	0x40023414	Endpoint 1 (9) buffer start address	0xXXXX

R16_UEP2_DMA	0x40023418	Endpoint 2 (10) buffer start address	0xXXXX
R16_UEP3_DMA	0x4002341c	Endpoint 3 (11) buffer start address	0xXXXX
R16_UEP4_DMA	0x40023420	Endpoint 4 (8/12) buffer start address	0xXXXX
R16_UEP5_DMA	0x40023424	Endpoint 5 (13) buffer start address	0xXXXX
R16_UEP6_DMA	0x40023428	Endpoint 6 (14) buffer start address	0xXXXX
R16_UEP7_DMA	0x4002342c	Endpoint 7 (15) buffer start address	0xXXXX
R16_UEP0_T_LEN	0x40023430	Endpoint 0 transmit length register	0xXX
R8_UEP0_CTRL	0x40023432	Endpoint 0 control register	0x00
R16_UEP1_T_LEN	0x40023434	Endpoint 1 (9) transmit length register	0xXX
R8_UEP1_CTRL	0x40023436	Endpoint 1 (9) control register	0x00
R16_UEP2_T_LEN	0x40023438	Endpoint 2 (10) transmit length register	0xXX
R8_UEP2_CTRL	0x4002343a	Endpoint 2 (10) control register	0x00
R16_UEP3_T_LEN	0x4002343c	Endpoint 3 (11) transmit length register	0xXX
R8_UEP3_CTRL	0x4002343e	Endpoint 3(11) control register	0x00
R16_UEP4_T_LEN	0x40023440	Endpoint 4(8/12) transmit length register	0xXX
R8_UEP4_CTRL	0x40023442	Endpoint 4(8/12) control register	0x00
R16_UEP5_T_LEN	0x40023444	Endpoint 5(13) transmit length register	0xXX
R8_UEP5_CTRL	0x40023446	Endpoint 5(13) control register	0x00
R16_UEP6_T_LEN	0x40023448	Endpoint 6(14) transmit length register	0xXX
R8_UEP6_CTRL	0x4002344a	Endpoint 6(14) control register	0x00
R16_UEP7_T_LEN	0x4002344c	Endpoint 7(15) transmit length register	0xXX
R8_UEP7_CTRL	0x4002344e	Endpoint 7 (15) control register	0x00

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- pull-down resistor control bit: 1: Disable internal pull-down; 0: Enable internal pull-down. It also can be used in GPIO mode to provide pull-down resistor	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 bit: 1: Select 1.5Mbps low-speed mode; 0: Select 12Mbps full-speed mode.	0
1	RB_UD_GP_BIT	RW	USB device mode general flag, user-defined.	0
0	RB_UD_PORT_EN	RW	USB device physical port enable bit: 1: Enable the physical port;	0

			0: Disable the physical port.	

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) reception (OUT); 0: Disable endpoint 1(9) reception.	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 bit. *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: Disable endpoint 4 (8/12) transmission.	0
1	Reserved	RO	Reserved.	0
0	RB_UEP4_BUF_MOD	RW	Endpoint 4 (8/12) data buffer mode control bit. *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 applies to CH32V103x series.*	0

Note: For CH32V103x, endpoint 1 configuration option is mapped to endpoint 9, and endpoint 4 configuration option is mapped to endpoints 8 and 12. CH32F103x series products have no such mapping.

(Applied to F103) P4_RX_EN and bUEP4_TX_EN configure the data buffer mode of USB endpoint0 and endpoint 4. For details, refer to the following table:

Table 21-5 Endpoint 0 and endpoint 4 Buffer Mode (Applied to F103)

bUEP4_RX_EN	bUEP4_TX_EN	Description: arrange from low to high with UEP0 DMA as start address
0	0	Endpoint 0 single 64-byte receive/transmit shared buffers (IN and OUT).
1	0	Endpoint 0 single 64-byte receive/transmit shared buffers; endpoint 4 single 64-byte receive buffers (OUT).
0	1	Endpoint 0 single 64-byte receive/transmit shared buffers; endpoint 4 single 64-byte transmit buffers (IN).
1	1	Endpoint 0 single 64-byte receive/transmit shared buffers; endpoint 4 single 64-byte receive buffers (OUT); Endpoint 4 single 64-byte receive buffers (IN). All 192 bytes are arranged as follows: UEP0_DMA+0 address: The 64-byte start address of endpoint 0 receive/transmit shared buffer; UEP0_DMA+64 address: The 64-byte start address of endpoint 4 receive buffer; UEP0_DMA+128 address: The 64-byte start address of endpoint 4 transmit buffer.

Note: For CH32F103x, endpoint 4 does not support double buffer mode, and its DMA address allocation is related to endpoint 0 buffer. For details, refer to Table 21-5 above. For CH32V103x series products, refer to Table 21-6 for the distribution of endpoint 4 mode.

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 bit. *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 bit. *Note: When this bit is 1, UEP2_RX_EN and UEP2_TX_EN cannot be 1 at the same time.*	0

Note: For CH32V103x, endpoint 2 configuration option is mapped to endpoint 10, and the endpoint 3 configuration option is mapped to endpoint 11. CH32F103x series products have no such mapping.

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 bit. *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 bit. *Note: When this bit is 1, UEP5_RX_EN and UEP5_TX_EN cannot be 1 at the same time*	0

Note: For CH32V103x, endpoint 5 configuration option is mapped to endpoint 13, and endpoint 6 configuration option is mapped to endpoint 14. CH32F103x series products do not have this register.

Endpoint 7 (15) mode control register (R8_UEP7_MOD)

Bit	Name	Access	Description	Reset value
[7:5]	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 bit.	0

Note: For CH32V103x, endpoint 7 configuration option is mapped to endpoint 15. CH32F103x series products do not have this register.

The data buffer mode of USB endpoint 1 ~ endpoint 15 is configured in combination for RB_UEPn_RX_EN, RB_UEPn_TX_EN and RB_UEPn_BUF_MOD, respectively. Refer to Table 21-6 for details. Among them, in the double 64-byte buffer mode, the first 64-byte buffer will be selected based on RB_UEP_*_TOG=0 and the last 64-byte buffer will be selected based on RB_UEP_*_TOG=1 during USB data transmission, and RB_UEP_AUTO_TOG=1 is set to realize automatic switch.

Table 21-6 Endpoint n Buffer Mode (n=1-7)

RB_UEPn_RX_EN	RB_UEPn_TX_EN	RB_UEPn_BUF_MOD	Description: Arrange from low to high with R16_UEPn_DMA as start address
0	0	X	Endpoint is disabled, and the R16_UEPn_DMA buffer is not used.
1	0	0	Single 64-byte reception buffer (OUT).
1	0	1	Double 64-byte reception buffers (OUT), selected by RB_UEP_R_TOG.
0	1	0	Single 64-byte transmission buffers (IN).
0	1	1	Double 64-byte transmission buffers (IN), selected by RB_UEP_T_TOG.
1	1	0	Single 64-byte reception buffer (OUT), and single 64-byte transmission buffer (IN).
1	1	1	Double 64-byte reception buffers (OUT), selected by RB_UEP_R_TOG. Double 64-byte transmission buffers (IN), selected by RB_UEP_T_TOG. All 256 bytes are arranged as follows: UEPn_DMA+0 address: endpoint reception address when RB_UEP_R_TOG=0; UEPn_DMA+64 address: endpoint reception address when RB_UEP_R_TOG=1; UEPn_DMA+128 address: endpoint reception address when RB_UEP_T_TOG=0;

			UEPn_DMA+192 address: endpoint reception address when RB_UEP_T_TOG=1.

Note: For CH32F103x, the configuration options in Table 21-6 support n=1-3; for CH32V103x series products, the configuration options in Table 21-6 support n=1-7, and endpoints 8-15 configuration is mapped to endpoints 1-7 configuration.

Endpoint n buffer start address (R16_UEPn_DMA) (n=0/7)

Bit	Name	Access	Description	Reset value
[15:0]	R16_UEPn_DMA	RW	Endpoint n buffer start address. Lower 15 bits are valid, and 4 bytes must be aligned for the address.	X

Note 1: The length of the buffer that receives data >= min (maximum data packet length possibly received + 2 bytes, 64 bytes)
Note 2: F103 product endpoint DMA configuration supports 0-3 endpoints, V103 product endpoint DMA configuration supports 0-7 endpoints, and can be mapped to the configuration of endpoints 8-15.

Endpoint n sending length register (R8_UEPn_T_LEN) (n=0-7)

Bit	Name	Access	Description	Reset value
[7:0]	R8_UEPn_T_LEN	RW	Set the number of data bytes that USB endpoint n is ready to send (applied to F103) n=0-7.	X

Note 1: F103 product endpoint sending length configuration supports 0-4 endpoints, and V103 product endpoint sending length configuration supports 0-7 endpoints and can be mapped to configure the sending of 8-15 endpoints.
Note 2: Since the host endpoint sending register multiplexes the sending register of the endpoint 3 of the device, in V103, the host sending supports a maximum of 1023 bytes (for synchronous endpoint), so it is valid when the endpoint 3 sending register is expanded to 16 bits.

Endpoint n control register (R8_UEPn_CTRL) (n=0-7)

Bit	Name	Access	Description	Reset value
7	RB_UEP_R_TOG	RW	Expected synchronization trigger bit of the receiver (processing OUT transactions) of USB endpoint n: 1: Expect DATA1; 0: Expect DATA0.	0
6	RB_UEP_T_TOG	RW	Synchronization trigger bit of USB endpoint n transmitter (processing IN transactions) 1: Transmit DATA1; 0: Transmit DATA0.	0
5	Reserved	RO	Reserved.	0
4	RB_UEP_AUTO_TOG	RW	Synchronous trigger bit auto flip enable control bit: 1: After the data is transmitted or successfully received, the corresponding synchronization	0

			trigger bit is automatically flipped; 0: It is not flipped automatically, and can be switched manually. It only supports endpoint 1/2/3.	
[3:2]	MASK_UEP_R_RES	RW	Control on acknowledge to OUT transactions by the receiver of USB endpoint n: 00: ACK; - 01: Timeout/no response, used for real-time /synchronous transmission of non-endpoint 0; 10: NAK or busy; 11: STALL or errror.	00b
[1:0]	MASK_UEP_T_RES	RW	Response control by transmitter of endpoint n to IN services: 00: DATA0/DATA1 data is ready and ACK is expected; 01: No response to DATA0/DATA1 acknowledgment and expection, used for real-time/synchronous transmission of non-endpoint 0; 10: Response NAK or busy; 11: Response STALL or errror.	00b

Note: F103 product endpoint configuration supports endpoint 0 ~ endpoint 4, V103 product endpoint configuration supports endpoint 0 ~ endpoint 7, and can be mapped to the configuration of endpoint 8 ~ endpoint 15.

21.2.3 USB Host Register

In the USB host mode, the chip is equipped with 1 set of bidirectional host endpoints, including a sending endpoint OUT and a receiving endpoint IN. The maximum data length of a packet is 64 bytes (F103) or 1023 bytes (V103), supporting control transmission, interrupt transmission, bulk transmission and real-time/synchronous transmission.

Each USB transaction initiated by host endpoint always automatically sets the RB_UIF_TRANSFER interrupt flag after the processing ends. The application program can directly query or query and analyze the interrupt flag register R8_USB_INT_FG in the USB interrupt service program, and perform corresponding processing according to each interrupt flag; in addition, if RB_UIF_TRANSFER is valid, it is required to continue to analyze the USB interrupt status register R8_USB_INT_ST, and perform the corresponding processing according to the response PID identification MASK_UIS_H_RES of the current USB transmission transaction.

If the synchronization trigger bit RB_UH_R_TOG of IN transaction of host receiving endpoint is set in advance, you can judge whether the synchronization trigger bit of the data packet received matches the synchronization trigger bit of the endpoint through RB_U_TOG_OK or RB_UIS_TOG_OK; if the data is synchronized, the data is valid; if the data is not synchronized, the data should be discarded. After the USB sending or receiving interrupt is processed each time, the synchronization trigger bit of 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, RB_UH_T_AUTO_TOG and RB_UH_R_AUTO_TOG can be set to automatically flip the corresponding synchronization trigger bit after sending or receiving successfully.

USB host token setting register R8_UH_EP_PID is used to set the endpoint number of the target device being operated and the token PID packet identification of the USB transmission transaction. The data corresponding to the SETUP token and 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 receiving endpoint, the received data is stored in the R16_UH_RX_DMA buffer, and the received data length is stored in R8_USB_RX_LEN.

Table 21-7 List of Host Related Registers (those marked in grey are controlled by RB_UC_RESET_SIE reset) (applied to F103)

Name	Access address	Description	Reset value
R8_UHOST_CTRL	0x40023401	Physical port control register of USB host	0xX0
R8_UH_EP_MOD	0x4002340d	USB host endpoint mode control register	0x00
R16_UH_RX_DMA	0x40023418	USB host reception buffer start address	X
R16_UH_TX_DMA	0x4002341c	USB host transmission buffer start address	X
R8_UH_SETUP	0x40023426	USB host auxiliary setting register	0x00
R8_UH_EP_PID	0x40023428	USB host token setting register	0x00
R8_UH_RX_CTRL	0x4002342a	USB host reception endpoint control register	0x00
R8_UH_TX_LEN	0x4002342c	USB host transmit length register	X
R8_UH_TX_CTRL	0x4002342e	USB host transmission endpoint control register	0x00

Table 21-8 List of Host Related Registers (those marked in grey are controlled by RB_UC_RESET_SIE reset) (applied to V103)

Name	Access address	Description	Reset value
R8_UHOST_CTRL	0x40023401	Physical port control register of USB host	0xX0
R8_UH_EP_MOD	0x4002340d	USB host endpoint mode control register	0x00
R16_UH_RX_DMA	0x40023418	USB host reception buffer start address	X
R16_UH_TX_DMA	0x4002341c	USB host transmission buffer start address	X
R8_UH_SETUP	0x40023436	USB host auxiliary setting register	0x00
R8_UH_EP_PID	0x40023438	USB host token setting register	0x00
R8_UH_RX_CTRL	0x4002343a	USB host reception endpoint control register	0x00
R16_UH_TX_LEN	0x4002343c	USB host transmit length register	X
R8_UH_TX_CTRL	0x4002343e	USB host transmission endpoint control register	0x00

USB host physical port control register (R8_UHOST_CTRL)

Bit	Name	Access	Description	Reset value
7	RB_UH_PD_DIS	RW	USB host port UD+/UD-pull-down resistor control bit: 1: Disable internal pull-down; 0: Enable internal pull-down. It also can be used in GPIO mode to provide pull-down resistor	1
6	Reserved	RO	Reserved.	0
5	RB_UH_DP_PIN	RO	Current UD + pin status:	X

			1: High level; 0: Low level.	
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 bit: 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 bit: 1: Output USB bus reset by force; 0: The output is completed.	0
0	RB_UH_PORT_EN	RW	USB host port enable bit: 1: Enable the host port; 0: Disable the host port. The bit will be automatically cleared when the USB device is disconnected	0

USB host endpoint mode control register (R8_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 bit: 1: Enable endpoint transmission; 0: Disable endpoint transmission.	0
5	Reserved	RO	Reserved.	0
4	RB_UH_EP_TBUF_MOD	RW	Host transmission endpoint transmit data buffer mode control bit.	0
3	RB_UH_EP_RX_EN	RW	Host reception endpoint receive (IN) enable bit: 1: Enable endpoint reception; 0: Disable endpoint reception.	0
[2:1]	Reserved	RO	Reserved.	0
0	RB_UH_EP_RBUF_MOD	RW	USB host reception endpoint data buffer mode control bit.	0

The data buffer modes of host transmission endpoint are controlled by a combination of RB_UH_EP_TX_EN and RB_UH_EP_TBUF_MOD. Refer to the following table.

Table 21-9 Host Transmission Buffer Mode

RB_UH_EP_TX_EN	RB_UH_EP_TBUF_MOD	Description: Take R16_UH_TX_DMA as start address
0	X	Endpoint is disabled, and the R16_UH_TX_DMA buffer is not used.
1	0	Single 64-byte transmission buffers (SETUP/OUT).
1	1	Double 64-byte transmission buffers, selected by RB_UH_T_TOG: When RB_UH_T_TOG=0, select the first 64 bytes of

		the buffer; When bUH_R_TOG=1, select the last 64-byte buffer.

The data buffer modes of USB host reception endpoint are controlled by a combination of RB_UH_EP_RX_EN and RB_UH_EP_RBUF_MOD. Refer to the following table.

Table 21-10 Host Reception Buffer Mode

RB_UH_EP_RX_EN	RB_UH_EP_RBUF_MOD	Structure description: Take R16_UH_TX_DMA as start address
0	X	Endpoint is disabled, and the R16_UH_RX_DMA buffer is not used.
1	0	Single 64-byte reception buffers (IN).
1	1	Double 64-byte reception buffers, selected by RB_UH_R_TOG: When RB_UH_R_TOG=0, select the first 64 bytes of the buffer; When RB_UH_R_TOG=1, select the last 64-byte buffer.

USB host reception buffer start address (R16_UH_RX_DMA)

Bit	Name	Access	Description	Reset value
[15:0]	R16_UH_RX_DMA	RW	Host endpoint data reception buffer start address. Lower 15 bits are valid, and 4 bytes must be aligned for the address.	X

USB host transmission buffer start address (R16_UH_TX_DMA)

Bit	Name	Access	Description	Reset value
[15:0]	R16_UH_TX_DMA	RW	Host endpoint data transmission buffer start address. Lower 15 bits are valid, and 4 bytes must be aligned for the address.	X

USB host auxiliary setting register (R8_UH_SETUP)

Bit	Name	Access	Description	Reset value
7	RB_UH_PRE_PID_EN	RW	Low-speed preamble packet PRE PID enable bit: 1: Enable, used to communicate with low-speed USB device through an external HUB. 0: Disable the low-speed preamble packet.	0
6	RB_UH_SOF_EN	RW	Automatically generate SOF packet enable	0

			bit: 1: The host automatically generates SOF packet; 0: Disable the automatic SOF function.	
[5:0]	Reserved	RO	Reserved.	0

USB host token setting register (R8_UH_EP_PID)

Bit	Name	Access	Description	Reset value
[7:4]	MASK_UH_TOKEN	RW	Set the token PID identification of this USB transmission transaction.	0
[3:0]	MASK_UH_ENDP	RW	Set the endpoint number of the target device being operated this time.	0

USB host reception endpoint control register (R8_UH_RX_CTRL)

Bit	Name	Access	Description	Reset value
7	RB_UH_R_TOG	RW	Synchronous trigger bit expected by the USB host receiver (processing IN transactions): 1: Expect DATA1; 0: Expect DATA0.	0
[6:5]	Reserved	RO	Reserved.	0
4	RB_UH_R_AUTO_TOG	RW	Synchronous trigger bit auto flip enable control bit: 1: After the data is successfully received, the corresponding expected synchronous trigger bit (RB_UH_R_TOG) is automatically flipped; 0: Manually control the synchronous trigger bit (RB_UH_R_TOG).	0
3	Reserved	RO	Reserved.	0
2	RB_UH_R_RES	RW	Control on response to IN transactions by the receiver of host: 1: No response, used for real-time /synchronous transmission of non-endpoint0; 0: Response ACK.	0
[1:0]	Reserved	RO	Reserved.	0

USB host transmit length register (R8_UH_TX_LEN/R16_UH_TX_LEN)

Bit	Name	Access	Description	Reset value
[9:0]	R16_UH_TX_LEN	RW	Set the number of data bytes to be sent by USB host transmission endpoint (applied to V103).	X
[7:0]	R8_UH_TX_LEN	RW	Set the number of data bytes that USB host	X

			transmission endpoint is ready to send (applied to F103).	

USB host transmission endpoint control register (R8_UH_TX_CTRL)

Bit	Name	Access	Description	Reset value
7	Reserved	RO	Reserved.	0
6	RB_UH_T_TOG	RW	Synchronous trigger bit prepared by USB host transmitter (processing SETUP/OUT transactions): 1: Transmit DATA1; 0: Transmit DATA0.	0
5	Reserved	RO	Reserved.	0
4	RB_UH_T_AUTO_TOG	RW	Synchronous trigger bit auto flip enable control bit: 1: The corresponding synchronous trigger bit (RB_UH_T_TOG) is flipped after the data is sent successfully; 0: Manually control the synchronous trigger bit (RB_UH_T_TOG).	0
[3:1]	Reserved	RO	Reserved.	0
0	RB_UH_T_RES	RW	Response control bit of USB host transmitter to SETUP/OUT transaction: 1: No response to the expectation, used for real-time/synchronous transmission of non-endpoint0; 0: Expected acknowledgment ACK.	0

Chapter 22 Controller LAN (CAN)

The module description in this chapter applies to the full range of CH32F103 microcontrollers.

Controller LAN is a high-performance communication protocol for serial data communication. The CAN controller provides a complete CAN protocol implementation scheme, supporting CAN protocols 2.0A and 2.0B. The CAN controller can be used to construct a powerful LAN to realize safe distributed real-time control, and process a large number of data messages with a small CPU load. It is widely applied in the industrial and automotive fields.

22.1 Main Features

- Compatible with CAN specification 2.0A and 2.0B
- Programmable transmission rate, up to 1Mbit/s
- Support TTCAN protocol to avoid low-priority message blocking
- Support three transmit mailboxes; the priority of sending messages can be determined by the message identifier, or the sending mailbox can be configured as a sending FIFO, and the time stamp of the SOF moment when the message is sent can be recorded
- Support 2 receiver FIFOs with three-level mailbox depth; 14 message filter groups are available for configuration, each filter group can be configured in 32 or 16 bits mode, mask bit or identifier list mode to minimize intervention of message filtering by the software. The FIFO overflow processing method is flexible, and the time stamp of the moment when the message SOF is received can be recorded
- 4 interrupt vectors are occupied, and each interrupt source enable can be independently configured

22.2 CAN Controller Working Mode

The CAN controller can operate the SLEEP or INRQ bit in the CAN_CTLR register to switch between the three operating modes: initialization, sleep and normal.

22.2.1 Initialization Mode

After reset, CAN works in sleep mode by default to reduce power consumption. At this time, message transceiving is disabled, the internal pull-up resistor of the TX pin is enabled, and the TX pin outputs a recessive bit. Set the INRQ bit in the CAN_CTLR register to 1, and request the CAN controller to enter the initialization mode. When the INAK bit in the CAN_STATR register is automatically set to 1, the initialization status will be successfully entered. Similarly, clear the INRQ bit in the CAN_CTLR register to zero and request the CAN controller to exit the initialization mode. When the INAK bit in the CAN_STATR register is automatically cleared to 0, the initialization state will be successfully exited.

The filter group can be initialized in the non-initialization mode, but the FINIT bit of the CAN_FCTLR register must be set to 1. At this time, the message transceiving is disabled.

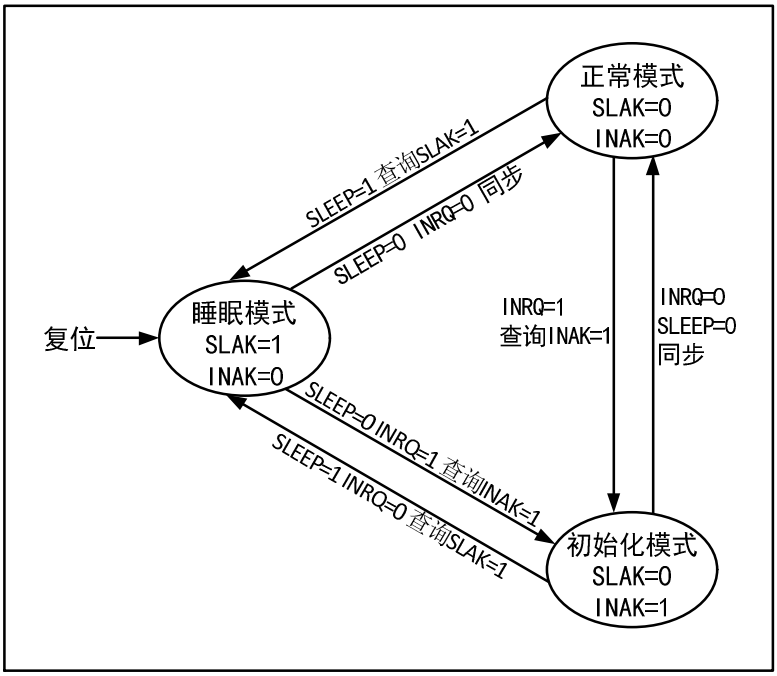
22.2.2 Sleep Mode

Set the SLEEP bit in the CAN_CTLR register to 1, and request the CAN controller to enter the sleep mode. When the SNAK bit in the CAN_STATR register is automatically set to 1, the CAN will successfully enter the sleep mode. At this time, the clock of the CAN controller will stop, but the mailbox register will still be accessible.

To enter the initialization mode from sleep mode, the SLEEP bit of CAN_CTLR must be cleared to 0 and the INRQ bit shall be set to 1. When the INAK bit of the register CAN_STATR is automatically set to 1, the switch to the initialization state will be completed.

To enter the normal mode from the sleep mode, the SLEEP bit of CAN_CTLR must be cleared to 0. When the SNAK bit of the CAN_STATR register is automatically cleared to 0, the normal mode will be entered.

Figure 22-1 CAN Working Mode Switch



22.3 CAN Controller Test Mode

In the initialization mode, operate the SILM and LBKM bits of the CAN_BTIMR register to prepare to enter one of the test modes, and then clear the INRQ bit of the CAN_CTLR register to zero to exit the initialization mode and enter the test mode. There are three test modes: silent mode, loopback mode and silent loopback mode.

22.3.1 Silent Test Mode

Set the SILM bit of the CAN_BTIMR register to 1, and you can choose to prepare to enter the silent mode. In this mode, the CAN controller can receive but cannot send messages externally. It is always in a recessive bit to avoid affecting the bus, but the message can be received by the controller of the node where it is located. Generally, the silent mode is used for CAN bus status analysis.

22.3.2 Loopback Test Mode

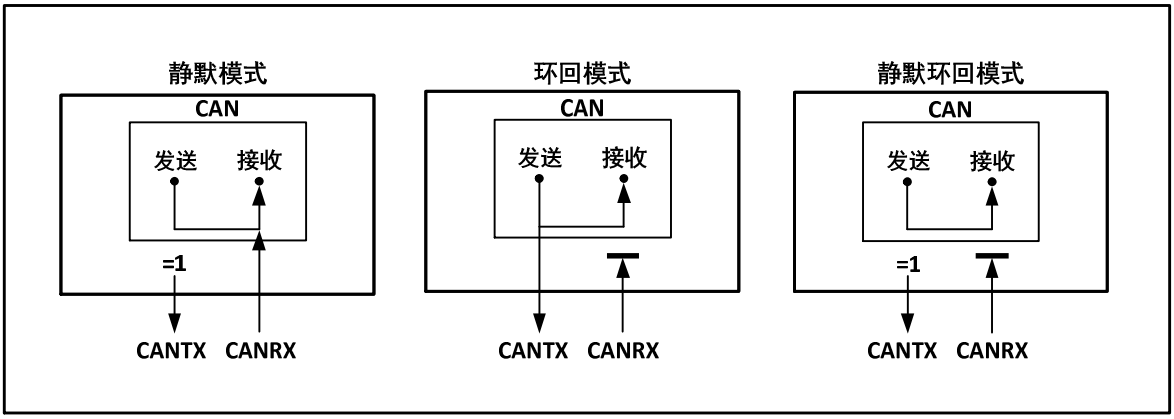
Set the LBKM bit of the CAN_BTIMR register to 1, and you can choose to prepare to enter the loopback mode. In this mode, the CAN controller can send external messages, but cannot receive external messages. However, the sent messages can be received by the controller at the node where it is located, and the reception filtering mechanism is effective. Generally, the loopback mode is used for the transceiving test of CAN controller.

22.3.3 Silent Loopback Test Mode

Set the SILM and LBKM bits of the CAN_BTIMR register to 1, and you can choose to prepare to enter the silent loopback mode. This mode is usually used for the closed self-test of the CAN controller. In this mode, it

has no impact on the CAN bus, the RX pin is disconnected from the bus, and the TX pin is set to a recessive bit.

Figure 21-2 Three Test Modes of CAN Bus



22.4 Working Status of CAN Controller when MCU is in Debug Mode

When the MCU enters debug mode, the core will be in a suspended state, but the configuration bit can be used to determine whether the CAN controller is in normal operation or stop state.

22.5 Functional Description of CAN Controller

22.5.1 Transmission Processing Flow

The transmission processing flow is as follows: If there are vacant mailboxes in the three transmitting mailboxes, the application layer software only has write access to the registers of the vacant mailboxes, and operates on the registers CAN_TXMIRx, CAN_TXMDTRx, CAN_TXMDLRx and CAN_TXMDHRx. The message identifier and message Text length, time stamp and message data, etc. can be set. After the data is ready, set the TXRQ bit of register CAN_TXMIRx to 1 to request sending. The mailbox will enter the registered state, and priority queuing will be conducted; once it becomes the highest priority mailbox, it will become the scheduled transmission state, waiting for the CAN bus to be idle; when the CAN bus is idle, the message of the scheduled transmission mailbox will enter the transmission status immediately; after the message is transmitted, the mailbox will become vacant again, and the RQCP and TXOK bits of the CAN_TSTATR register will be set to 1, indicating that the transmission is successful; if the arbitration fails during transmission, the ALST bit in the CAN_TSTATR register will be set to 1; for the transmission error, set TERR bit to 1.

22.5.2 Transmission Priority

The transmission priority can be determined by the identifier or the request order of the transmission. Set the TXFP bit of the register CAN_CTLR to 1, send in the order of transmission requests, and the transmission request order is mainly used for segmented transmission; clearing to determine the transmission order according to the priority of the identifier. The smaller identifier means the higher priority. In the case of the same identifier, the mailbox with the lower number has a higher priority.

22.5.3 Transmission Abort Processing

If the ABRQ bit of the CAN_TSTATR register is set to 1, the transmission request can be aborted. When the mailbox status is registered or scheduled to transmission, the transmission request will be directly aborted;

when the mailbox is at the transmission status, the aborting request may succeed (transmission stop) or fail (transmission completion). The result can be inquired by the TXOK bit in the CAN_TSTATR register.

22.5.4 Time Based Trigger Mode

Traditional CAN is based on an event trigger mode. When the bus is busy in this mode, low-priority messages are likely to be blocked for a long time, even failing to meet its time limit requirements. In order to solve this bottleneck, related protocols based on the time-trigger mode have been introduced. This type of protocol has a certain scale of application in the industry. The function based on the time-trigger mode is the application of this type of protocol.

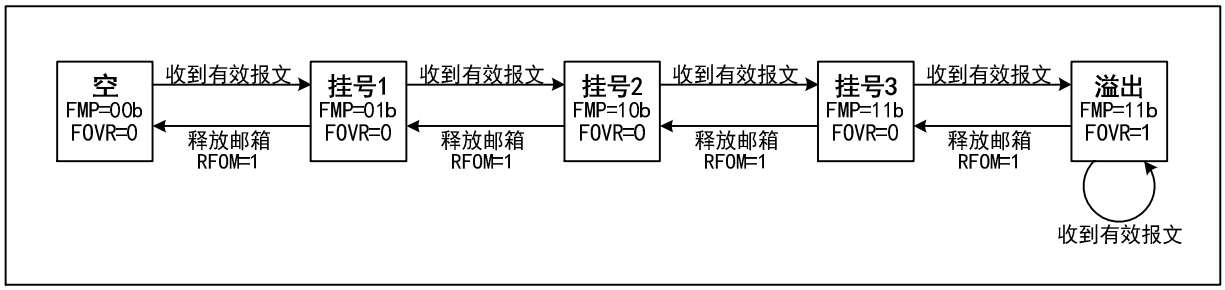
Set the TTCM bit of the CAN_CTLR register to 1 to enable the time trigger mode. At this time, the internal timer will be activated to generate the time stamp of the sending and receiving mailboxes. The timer accumulates time at the CAN bit and is sampled at the bit time sampling point to generate the timestamp.

22.5.5 Reception Handling Process

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 two FIFOs with 3-level mailbox depth according to the setting of the CAN_FAFIFOR register. If the application layer needs to obtain messages, it can only read valid received messages through the receiver FIFO mailbox.

Initially, the receiver FIFO is empty, and the FMR[1:0] value of the receiver FIFO register CAN_RFIFOx is binary 00b. After receiving a valid receiving message, it will become the registered 1 status, and the controller will automatically set the FMR [1:0] of receiver FIFO register CAN_RFIFOx to binary 01b; if the mailbox data registers CAN_RXMDLRx and CAN_RXMDHRx are read at this moment, the mailbox will be released by setting the RFOM bit of receiver FIFO register CAN_RFIFOx to 1, and the receiver FIFO state will become vacant; if the mailbox is not released at the suspended 1 state, the state of receiver FIFO is switched to the suspended 2 state after the next valid receiving message is received. At this time, FMR[1:0] of the receiver FIFO register CAN_RFIFOx will be automatically set to binary 10b; if the mailbox data register is read and the mailbox is released, the state will return to registered 1; if the mailbox is not released at the registered 2 state, the receiver FIFO will enter the registered 3 stats; also read the message and release into the mailbox in the registered 3 state, it will return to the registered 2 state; if the mailbox is not released at the registered 3 state, the message will be inevitably lost when the next valid message is received.

Figure 21-3 Receiver FIFO state Switching Diagram



The above message loss situation means that the receiver FIFO is full, and the message overflow causes the message to be lost. The FOVR bit of the receiver FIFO register CAN_RFIFOx will be automatically set to 1 by hardware for overflow query. If the RFLM bit in the CAN_CTLR register is set to 1, the receiver FIFO lock function will be enabled, and the discarded message will be a new received message; the RFLM bit in the CAN_CTLR register will be cleared, and the receiver FIFO lock function will be disabled. Among the three original messages in the receiver FIFO, the last message received will be overwritten by the new message.

When the related bit of register CAN_INTENR is set, an interrupt can be generated if the receiver FIFO status is switched, so as to process the received message more efficiently. For details, see section 22.6 CAN Interrupt.

22.5.6 Receiving Message Identifier Filter

There are as many as 14 filter groups in the module. By setting the filter group, each CAN node can receive messages that meet the filter rules, and the messages that do not meet the filter rules are discarded by the hardware without software intervention.

Each filter group is composed of two 32-bit registers CAN_FxR0 and CAN_FxR1. The bit width of the filter group can be independently configured into one 32-bit filter or two 16-bit filters by setting each bit of the CAN_FSCFGR register. Each filter group can be configured as a mask bit or identifier list mode by setting each bit of the register CAN_FMCFGR, and each filter group can be enabled or disabled by setting each bit of the register CAN_FWR. You can store the message selected to pass through the filter in the required receiver FIFO by setting each bit of register CAN_FAFIFOR.

As shown in the following table 22-1, in the mask bit mode, two registers are the identifier register and the mask register. The two registers need to be used together. Each bit of the identifier register indicates the dominant or recessive expected value of the corresponding bit. Each bit of the mask register indicates whether the corresponding bit needs to be consistent with the expected value of the corresponding identifier register bit.

Table 22-1 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]			
mapping	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 receiving message identifier must be consistent with one of the registers, so the screening can be passed.

Table 22-2 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]			
mapping	STID[10:3]	STID[2:0]	EXID[17:13]	EXID[12:5]	EXID[4:0]	IDE	RTR	0

In the 16-bit mode, the register group is divided into four registers. The mask bit mode of each group of filter can have 2 filters, and each filter contains a 16-bit identifier register and a 16-bit mask register; all four registers in the identifier list mode are used as identifier registers.

Table 22-3 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]			
mapping	STID[10:3]	STID[2:0]	RTR	IDE	EXID[17:15]

Table 22-4 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]			
mapping	STID[10:3]	STID[2:0]	RTR	IDE	EXID[17:15]

Messages entering the FIFO mailbox 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 the filter number for the messages filtered by different filters in the receiver FIFO. The number is stored in the FMI[7:0] of the register CAN_RXMDTRx. You do not need to consider whether the filter group is enabled or not during numbering. See the example in Figure 22-4 for the numbering rules.

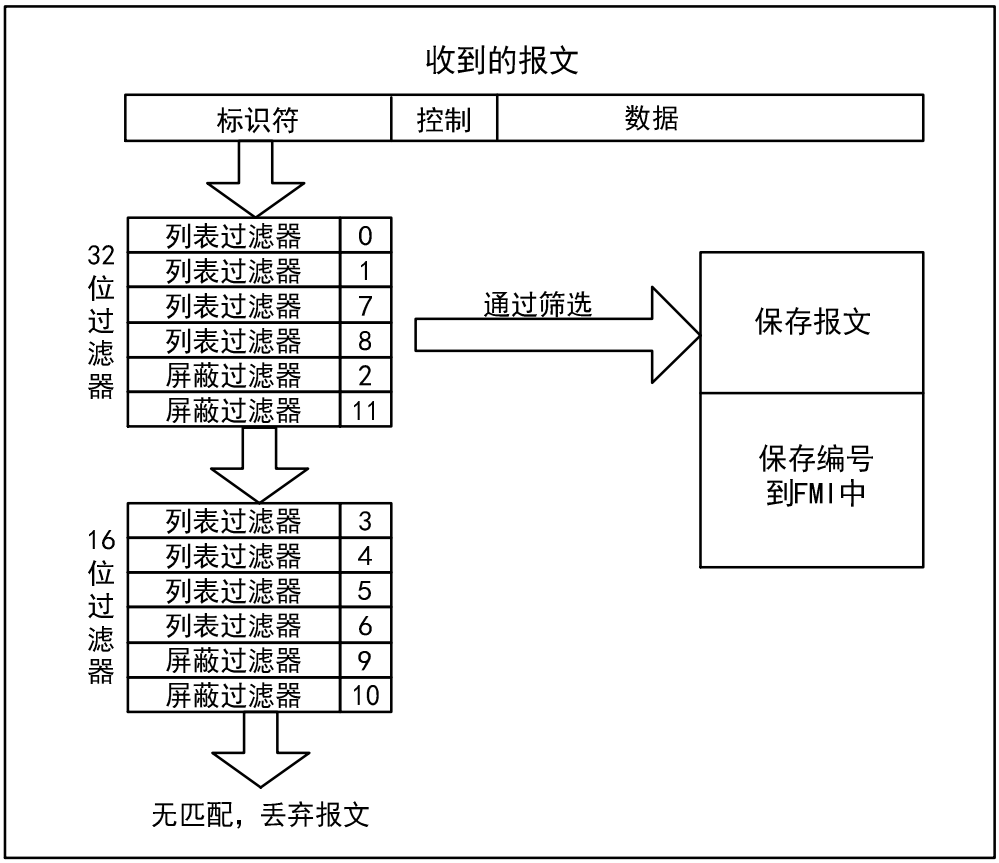
When a message can be filtered by multiple filters, the filter number stored in the receiving mailbox determines which filter number is stored according to the filter priority rules. The filter priority rules are as follows:

- All 32-bit filters have higher priority than 16-bit filters.
- For filters of the same width, the priority of the filter of the identifier list is higher than the filter of the mask bit mode.
- For filters with the same width and pattern, filters with lower number have higher priority

As shown in Figure 22-5: When receiving a message, firstly match the identifier with the 32-bit identifier list mode filter during screening; if there is no match, then match and filter with the 32-bit mask bit mode filter; if there is no match, continuously filter with the 16-bit identifier list mode filter. If there is no match, match with the 16-bit mask bit mode filter 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 receiver FIFO, and the identifier number will be stored in the FMI of the register CAN_RXMDTRx.

Figure 22-4 Example of Filter Number

Figure 22-5 Filtering Example of Filter



22.5.7 Error Processing

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 respectively represent the transmission and receiving error count values, which increase according to the increase in sending and receiving errors and decrease when the sending and receiving succeeds. The stability of the CAN bus can be judged according to their values.

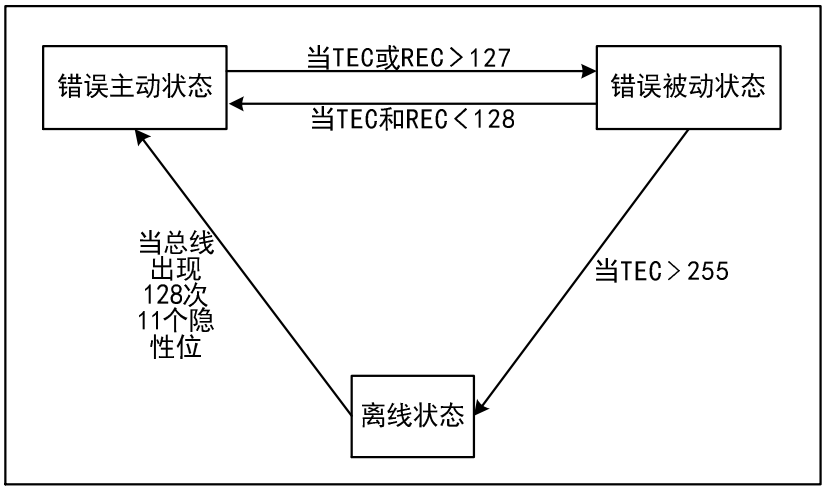
When the TEC and REC in the status error register CAN_ERRSR are less than 128, the current CAN node will be in the error active state, can participate in bus communication normally, and will send an active error flag when an error is detected.

When the TEC and REC in the status error register CAN_ERRSR are greater than 127, the current CAN node will be in an error passive state, and the active error flag will not be allowed to be sent when an error is detected, only the passive error flag can be sent.

When the TEC in the status error register CAN_ERRSR is greater than 255, the current CAN node will enter the offline state.

When 11 consecutive recessive bits appear for 128 times in the bus monitoring, it will recover to the error active state. This recovery method is affected by the ABOM bit in the main control register CAN_CTLR. If ABOM is set to 1, the hardware will automatically exit the offline state. If ABOM is 0, the software needs to operate the INRQ bit to enter the initialization mode, and then exits the initialization to exit the offline state.

Figure 22-6 CAN Error State Switching Diagram



22.5.8 Bit Timing

According to the CAN bus standard, each bit time is divided into four segments: synchronization segment, propagation time segment, phase buffer segment 1 and phase buffer segment 2. These segments are composed of the minimum time unit Tq. The CAN controller monitors the CAN bus changes through sampling and synchronizes through the edge of the frame start bit

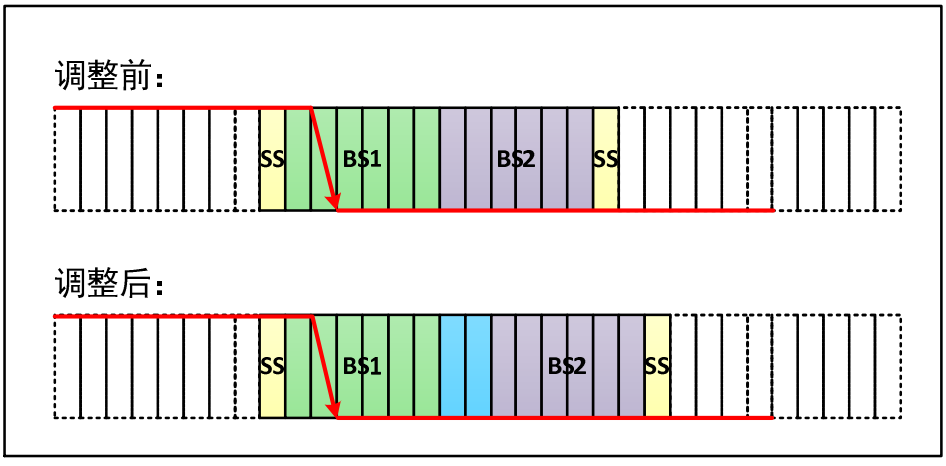
The CAN controller re-divides the above four segments into three segments, namely:

- Synchronization segment (SS): It is the synchronization segment in the CAN standard, which is fixed to 1 minimum time unit. Under normal circumstances, the expected bit jump occurs in this time period.
- Bit segment 1 (BS1): It contains the propagation time period and phase buffer section 1 in the CAN standard. It can be set to include the minimum time unit from 1 to 16 and can be automatically extended to compensate the corresponding forward drift arising from the frequency accuracy error at different nodes on the CAN bus. The end of the time period is the sampling point position.
- Bit segment 2 (BS2): It 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 the phase negative drift arising from the frequency accuracy error 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 per bit, and the range can be set to 1 to 4 minimum time units.

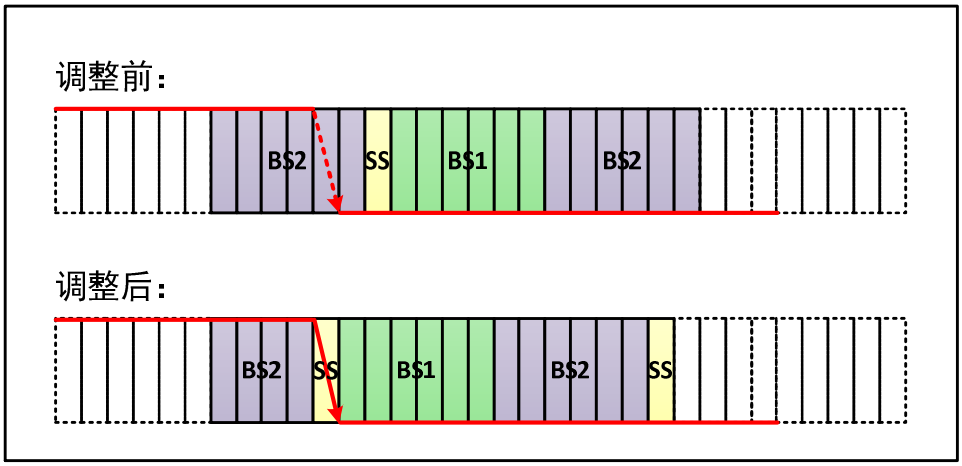
The above parameters can be configured in the CAN bus timing register CAN_BTIMR.

Figure 22-7 Jump Appears in BS1



As shown in Figure 22-7, SJW is 2, and the bus level jump is detected in time period 1. You need to extend the length of time period 1, and extend SJW to the greatest extent to delay the position of the sampling point.

Figure 22-8 Jump Appears in BS2



As shown in Figure 22-8, SJW is 2, and the bus level jump is detected in time period 2. Then, the length of time period 2 needs to be reduced, and SJW is reduced to the greatest extent to advance the position of the sampling point.

22.6 CAN Interrupt

The CAN controller has four interrupt vectors: transmit interrupt, FIFO_0 interrupt, FIFO_1 interrupt, error and status change interrupt.

Set the CAN interrupt enable register CAN_INTENR to enable or disable each interrupt source.

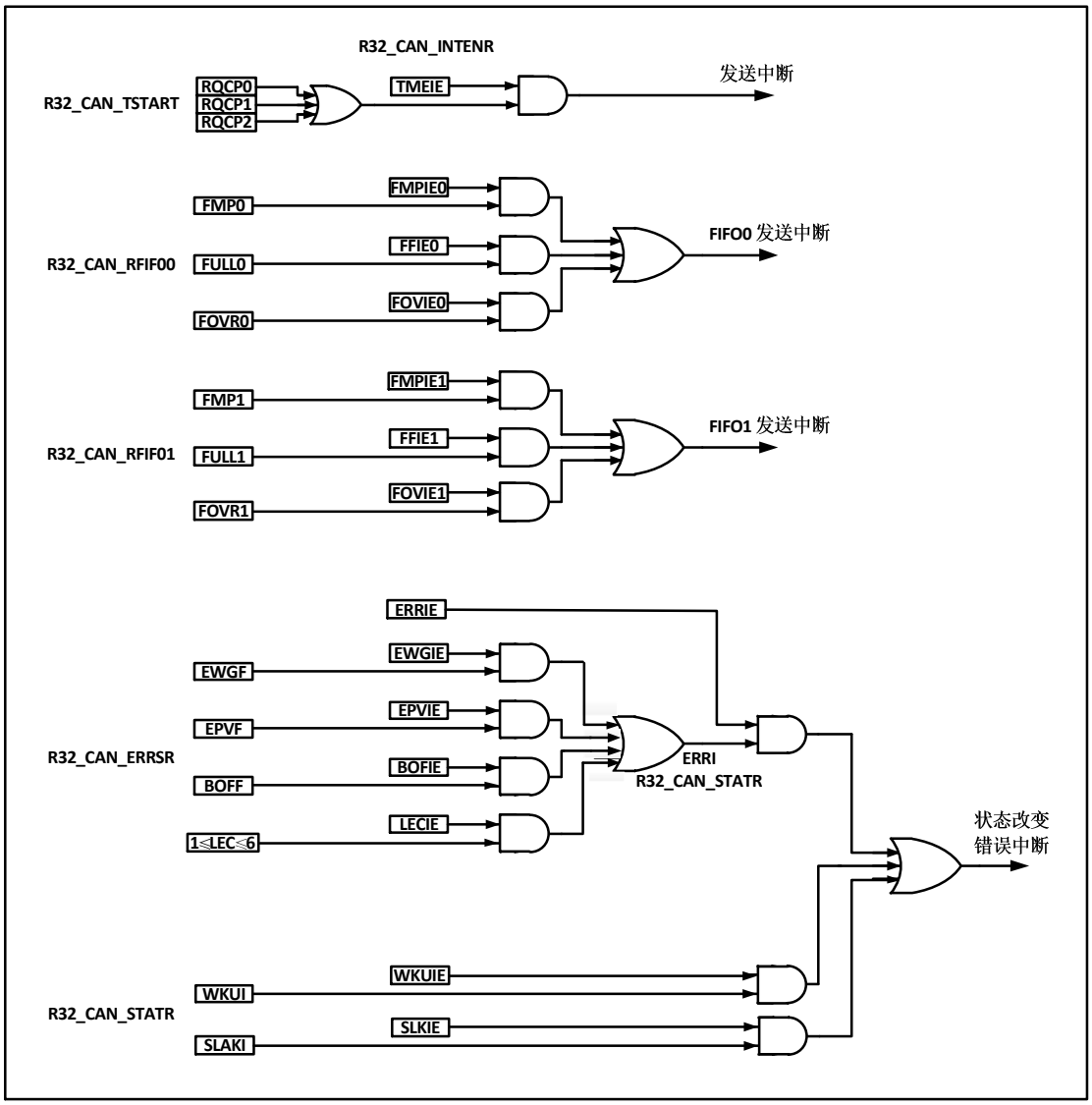
The transmission interrupt is generated by the emptying event of the transmission mailbox. After the interrupt is generated, the RQCP0, RQCP1 and RQCP2 bits of the CAN_TSTATR register will be checked to determine which mailbox emptying event is generated.

The FIFO0 interrupt is generated by receiving new messages, full receiving mailbox and overflow events. After the interrupt is generated, query the FMP0, FULL0 and FOVER0 bits of the CAN_RFIFO0 register to determine which mailbox has become empty event.

The FIFO1 interrupt is generated by receiving new messages, full receiving mailbox and overflow events. After the interrupt is generated, query the FMP1, FULL1 and FOVER1 bits of the CAN_RFIFO1 register to determine which mailbox has become empty event.

Error and status change interrupts are generated by error, wake-up and sleep events.

Figure 22-9 CAN Interrupt Logic Diagram



22.7 Register Description

The registers related to the CAN controller must be operated 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 sequence register CAN_BTIMR in the initialization mode.

Table 22-5 List of CAN Related Registers

Name	Access address	Description	Reset value
R32_CAN_CTLR	0x40006400	CAN master control register	0x00010002
R32_CAN_STATR	0x40006404	CAN master status register	0x00000C02
R32_CAN_TSTATR	0x40006408	CAN transmit status register	0x1C000000
R32_CAN_RFIFO0	0x4000640C	CAN receiver FIFO0 control and status register	0x00000000
R32_CAN_RFIFO1	0x40006410	CAN receiver FIFO1 control and status register	0x00000000
R32_CAN_INTENR	0x40006414	CAN interrupt enable register	0x00000000
R32_CAN_ERRSR	0x40006418	CAN error status register	0x00000000
R32_CAN_BTIMR	0x4000641C	CAN bit timing register	0x01230000

Table 22-6 List of CAN Mailbox Related Registers

Name	Access address	Description	Reset value
R32_CAN_TXMIR0	0x40006580	CAN transmit mailbox 0 identifier register	X
R32_CAN_TXMDTR0	0x40006584	CAN transmit mailbox 0 data length or time stamp register	X
R32_CAN_TXMDLR0	0x40006588	CAN transmit mailbox 0 low byte data register	X
R32_CAN_TXMDHR0	0x4000658C	CAN transmit mailbox 0 high byte data register	X
R32_CAN_TXMIR1	0x40006590	CAN transmit mailbox 1 identifier register	X
R32_CAN_TXMDTR1	0x40006594	CAN transmit mailbox 1 data length or time stamp register	X
R32_CAN_TXMDLR1	0x40006598	CAN transmit mailbox 1 low byte data register	X
R32_CAN_TXMDHR1	0x4000659C	CAN transmit mailbox 1 high byte data register	X
R32_CAN_TXMIR2	0x400065A0	CAN transmit mailbox 2 identifier register	X
R32_CAN_TXMDTR2	0x400065A4	CAN transmit mailbox 2 data length or time stamp register	X
R32_CAN_TXMDLR2	0x400065A8	CAN transmit mailbox 2 low byte data register	X
R32_CAN_TXMDHR2	0x400065AC	CAN transmit mailbox 2 high byte data register	X
R32_CAN_RXMIR0	0x400065B0	CAN receiver FIFO0 mailbox identifier register	X
R32_CAN_RXMDTR0	0x400065B4	CAN receiver FIFO0 mailbox data length and time stamp register	X
R32_CAN_RXMDLR0	0x400065B8	CAN receiver FIFO0 mailbox low byte data register	X
R32_CAN_RXMDHR0	0x400065BC	CAN receiver FIFO0 mailbox high byte data register	X
R32_CAN_RXMIR1	0x400065C0	CAN receiver FIFO1 mailbox identifier register	X
R32_CAN_RXMDTR1	0x400065C4	CAN receiver FIFO1 mailbox data length and time stamp register	X
R32_CAN_RXMDLR1	0x400065C8	CAN receiver FIFO1 mailbox low byte data register	X
R32_CAN_RXMDHR1	0x400065CC	CAN receiver FIFO1 mailbox high byte data register	X

Table 22-7 List of CAN Filter Related Registers

Name	Access address	Description	Reset value
R32_CAN_FCTLR	0x40006600	CAN filter master control register	0x2A1C0E01
R32_CAN_FMCFGR	0x40006604	CAN filter mode register	0x00000000
R32_CAN_FSCFGR	0x4000660C	CAN filter bit width register	0x00000000
R32_CAN_FAFIFOR	0x40006614	CAN filter FIFO associated register	0x00000000
R32_CAN_FWR	0x4000661C	CAN filter activation register	0x00000000
R32_CAN_F0R1	0x40006640	CAN filter group 0 register 1	X
R32_CAN_F0R2	0x40006644	CAN filter group 0 register 2	X
R32_CAN_F1R1	0x40006648	CAN filter group 1 register 1	X
R32_CAN_F1R2	0x4000664C	CAN filter group 1 register 2	X
R32_CAN_F2R1	0x40006650	CAN filter group 2 register 1	X
R32_CAN_F2R2	0x40006654	CAN filter group 2 register 2	X

R32_CAN_F3R1	0x40006658	CAN filter group 3 register 1	X
R32_CAN_F3R2	0x4000665C	CAN filter group 3 register 2	X
R32_CAN_F4R1	0x40006660	CAN filter group 4 register 1	X
R32_CAN_F4R2	0x40006664	CAN filter group 4 register 2	X
R32_CAN_F5R1	0x40006668	CAN filter group 5 register 1	X
R32_CAN_F5R2	0x4000666C	CAN filter group 5 register 2	X
R32_CAN_F6R1	0x40006670	CAN filter group 6 register 1	X
R32_CAN_F6R2	0x40006674	CAN filter group 6 register 2	X
R32_CAN_F7R1	0x40006678	CAN filter group 7 register 1	X
R32_CAN_F7R2	0x4000667C	CAN filter group 7 register 2	X
R32_CAN_F8R1	0x40006680	CAN filter group 8 register 1	X
R32_CAN_F8R2	0x40006684	CAN filter group 8 register 2	X
R32_CAN_F9R1	0x40006688	CAN filter group 9 register 1	X
R32_CAN_F9R2	0x4000668C	CAN filter group 9 register 2	X
R32_CAN_F10R1	0x40006690	CAN filter group 10 register 1	X
R32_CAN_F10R2	0x40006694	CAN filter group 10 register 2	X
R32_CAN_F11R1	0x40006698	CAN filter group 11 register 1	X
R32_CAN_F11R2	0x4000669C	CAN filter group 11 register 2	X
R32_CAN_F12R1	0x400066A0	CAN filter group 12 register 1	X
R32_CAN_F12R2	0x400066A4	CAN filter group 12 register 2	X
R32_CAN_F13R1	0x400066A8	CAN filter group 13 register 1	X
R32_CAN_F13R2	0x400066AC	CAN filter group 13 register 2	X

CAN master control register (CAN_CTLR)

Offset address: 0x04

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	Whether to disable CAN bus during debug 1: During debug, CAN transmission and reception are disabled, but the control and read and write operations of the receiver FIFO are normal; 0: During debug, CAN controller operates normally.	1
15	RESET	RW1	CAN controller software reset request; invalid if writing this bit to 0 1: After the CAN controller is reset, the controller will enter the sleep mode, and then the hardware will be automatically cleared to 0;	0

			0: Normal state of CAN controller.	
[14:8]	Reserved	RO	Reserved	0
7	TTCM	RW	Whether to allow time trigger mode 1: Enable the time trigger mode; 0: Disable the 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 will automatically exit the offline state when detecting consecutive 11 recessive bits for 128 times; 0: The software need to opeate to set the INRQ bit of register CAN_CTLR to 1 and clear to 0. When detecting consecutive 11 recessive bits for 128 times, it will exit the offline state.	0
5	AWUM	RW	CAN controller automatic wake-up enable 1: When a message is detected, the hardware will automatically wake up, and the SLEEP and SLAK bits in the CAN_STATR register will be automatically cleared to 0; 0: The software needs to operate to clear the SLEEP bit of the register CAN_CTLR to 0.	0
4	NART	RW	Message automatic retransmission function disable 1: No matter whether the transmission is successful or not, the message can only be sent once; 0: The CAN controller keeps being retransmitted until the transmission is successful.	0
3	RFLM	RW	Receiver FIFO message lock mode enable 1: When the receiver FIFO overflows, the received mailbox message will not be read; when the mailbox is not released, the newly received message will be discarded; 0: When the receiver FIFO overflows, the received mailbox message will not be read. When the mailbox is not released, the newly received message will overwrite the original message.	0
2	TXFP	RW	Transmission mailbox priority mode selection 1: The priority is decided by the sequence of transmission request; 0: The priority is decided by the message identifier.	0
1	SLEEP	RW	Sleep mode request bit 1: Set 1 to request the CAN controller to enter the sleep mode. After the current activity is completed, the controller will enter the sleep mode. If the AWUM bit is set to 1, the controller will clear the SLEEP bit to 0 when a message is received;	1

			0: After software clears it to 0, the controller will exit the sleep mode.	
0	INRQ	RW	Initialization mode request bit 1: Set 1 to request the CAN controller to enter the initialization mode. After the current activity is completed, the controller will enter the initialization mode, and the hardware will set the INAK bit of the CAN_STATR register to 1; 0: Set to 0 to request the CAN controller to exit the initialization mode and enter the normal mode, and the hardware will clear the INAK bit of the CAN_STATR register to 0.	0

CAN master status register (CAN_STATR)

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	CAN controller receives the current actual level of pin RX.	1
10	SAMP	RO	CAN controller receives the level of a received bit on the pin RX	1
9	RXM	RO	Receiving mode query bit 1: The CAN controller is currently receiver; 0: The CAN controller is not currently receiver.	0
8	TXM	RO	Transmission mode query bit 1: The CAN controller is currently transmitter; 0: The CAN controller is not currently transmitter.	0
[7:5]	Reserved	RO	Reserved	0
4	SLAKI	RW1	When the sleep interrupt is enabled, i.e., when the SLKIE bit of the CAN_INTENR register is set to 1, the interrupt will generate a flag bit. Write 1 to clear 0; invalide if writing 0. 1: When the sleep mode is entered, an interrupt will be generated and the hardware will be set to 1; 0: When the sleep mode is exited, the hardware will clear 0 and the software can also clear 0.	0
3	WKUI	RW1	Wake-up interrupt flag bit. When the WKUI bit of the register CAN_INTENR is set to 1, if the CAN controller is at the sleep mode, the SOF bit will be	0

			detected, and the bit will be set to 1 by the hardware. Set it to 1 by the software to clear 0. Invalid if it is set to 0.	
2	ERRI	RW1	Error interrupt status flag bit. When the ERRIE bit of the CAN_INTENR register is set to 1, an error and status change interrupt will be generated. Set 1 by the software to clear 0. It is invalid if it is set 0.	0
1	SLAK	RO	Sleep mode indication bit. 1: CAN controller is in the sleep mode; 0: CAN controller is not in sleep mode.	1
0	INAK	RO	Initialization mode indication bit. 1: CAN controller is in the initialization mode; 0: CAN controller works in the non-initialization mode.	0

CAN transmission status register (CAN_TSTATR)

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 bit for transmit mailbox 2 1: Priority of transmit mailbox 2 is the lowest; 0: Priority of transmit mailbox 2 is not the lowest.	0
30	LOW1	RO	Lowest priority flag bit of transmit mailbox 1 1: Priority of transmit mailbox 1 is the lowest; 0: Priority of transmit mailbox 1 is not the lowest.	0
29	LOW0	RO	Lowest priority flag bit of transmit mailbox 0 1: Priority of transmit mailbox 0 is the lowest; 0: Priority of transmit mailbox 0 is not the lowest.	0
28	TME2	RO	Empty flag bit of transmit mailbox 2 1: Transmit mailbox 2 does not have the pending sending message; 0: Transmit mailbox 2 has the pending sending message.	1
27	TME1	RO	Empty flag bit of transmission mailbox 1 1: Transmit mailbox 1 does not have the pending sending message; 0: Transmit mailbox 1 has the pending sending message.	1
26	TME0	RO	Empty flag bit of transmit mailbox 0 1: Transmit mailbox 0 does not have the pending	1

			sending message; 0: Transmit mailbox 0 has the pending sending message.	
[25:24]	CODE	RO	Mailbox No. When more than one mailbox is empty, it means the next empty mailbox number; when all the mailboxes are empty, it means the mailbox number with the lowest priority.	0
23	ABRQ2	RW1	Transmission termination request of transmit mailbox 2. Set to 1 by software, the transmission request of mailbox 2 can be aborted. The hardware will be cleared to 0 when the transmitted message is cleared. If the mailbox 2 is cleared, it will be invalid when this bit is set to 1 by software.	0
[22:20]	Reserved	RO	Reserved	0
19	TERR2	RW1	Transmit mailbox 2 transmission failure flag bit; when the transmit mailbox 2 fails to send, this bit will be automatically set to 1 by software to clear. It is invalid if writing 0 by software.	0
18	ALST2	RW1	Transmit mailbox 2 arbitration failure flag bit. When the transmit mailbox 2 has a low arbitration priority which results in the transmission failure, this bit will be automatically set to 1. Set to 1 by software to clear. It is invalid if writing 0 by software.	0
17	TXOK2	RW1	Successful transmission flag bit of transmit mailbox 2 1: Previous transmission success; 0: Previous transmission failure. Set 1 by software to clear; it is invalid if writing 0 by the software.	0
16	RQCP2	RW	Transmit mailbox 2 request completion flag bit; this bit will be automatically set to 1 when the transmission or abort request of transmit mailbox 2 is completed. Set to 1 by the software to clear to 0; it is invalid if writing 0 by the software.	0
15	ABRQ1	RW0	Transmission abort request of transmit mailbox 1. Set to 1 by the software to abort the transmission request of mailbox 1, and cleared by hardware when transmission message is cleared. It is invalid if writing 0 by software.	0
[14:12]	Reserved	RO	Reserved	0
11	TERR1	RW1	Transmit mailbox 1 transmission failure flag bit; when the transmit mailbox 1 fails to send, this bit will be automatically set to 1. Set to 1 by software to clear. It is invalid if writing 0 by the software.	0

10	ALST1	RW1	Transmit mailbox 1 arbitration failure flag bit. When the transmit mailbox 1 has a low arbitration priority which results in the transmission failure, this bit will be automatically set to 1.	0
9	TXOK1	RW1	Successful transmission flag bit of transmit mailbox 1 1: Previous transmission success; 0: Previous transmission failure. Set 1 by software to clear it; it is invalid if writing 0 by software.	0
8	RQCP1	RW	Transmit mailbox 1 request completion flag bi; this bit will be automatically set to 1 when the transmission or abort request of transmit mailbox 1 is completed. Set 1 by software to clear; it is invalid if writing 0 by software.	0
7	ABRQ0	RW0	Transmission abort request of transmit mailbox 0. Set to 1 by software to abort transmission request of mailbox 0, and cleared by hardware when trnasmission message is cleared. It is invalid if writing 0 by software.	0
[6:4]	Reserved	RO	Reserved	0
3	TERR0	RW1	Transmission mailbox 0 transmission failure flag bit; when the transmit mailbox 0 fails to send, this bit will be automatically set to 1. Set to 1 to clear by software. It is invalid if writing 0 by software.	0
2	ALST0	RW1	Transmission mailbox 0 arbitration failure flag bit. When the transmission mailbox 0 has a low arbitration priority which results in the transmission failure, this bit will be automatically set to 1. Set to 1 by software to clear it. It is invalid if writing 0 by software.	0
1	TXOK0	RW1	Successful transmission flag bit of transmit mailbox 0 1: Previous transmission success; 0: Previous transmission failure. Set 1 by software to clear it; it is invalid if writing 0 by software.	0
0	RQCP0	RW	Transmit mailbox 0 request completion flag bit; this bit will be automatically set to 0 when the transmission or abort request of transmit mailbox 0 is completed. Set to 1 by software to clear; it is invalid if writing 0 by software.	0

CAN receiver FIFO0 status register (CAN_RFIFO0)

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	Set the bit to 1 by software, the current mailbox message of receiver FIFO_0 will be released. After release, it will be automatically cleared. It will be invalid if writing to 0 by software.	0
4	FOVR0	RW1	Receiver FIFO_0 overrun flag bit. When there are three messages in FIFO_0, a new message will be received and the hardware will be set to 1. This bit needs to be set to 1 by software to clear it; it will be invalid if writing 0 by software.	0
3	FULL0	RW1	Receiver FIFO_0 full flag bit. When there are three messages in FIFO_0, the hardware will be set to 1. This bit needs to be set to 1 by software to clear it; it will be invalid if writing 0 by software.	0
2	Reserved	RO	Reserved.	0
[1:0]	FMP0	RO	Number of receiver FIFO_0 messages.	0

CAN receiver FIFO1 status register (CAN_RFIFO 1)

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	Reserved.	0
5	RFOM1	RW1	Set the bit to 1 by software, the current mailbox message of receiver FIFO_1 will be released. After release, it will be automatically cleared. It will be invalid if writing to 0 by software.	0
4	FOVR1	RW1	Receiver FIFO_0 overrun flag bit. When there are three messages in FIFO_1, a new message will be received and will be set to 1 by hardware. This bit needs to be set to 1 by software to clear; it will be invalid if writing 0 by software.	0
3	FULL1	RW1	Receiver FIFO_0 full flag bit. When there are three	0

			messages in FIFO_1, the hardware will be set to 1. This bit needs to be set to 1 by software to clear it; it will be invalid if writing 0 by software.	
2	Reserved	RF	Reserved	0
[1:0]	FMP1	RO	Number of receiver FIFO_1 messages.	0

CAN interrupt enable register (CAN_INTENR)

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 bit. 1: When the sleep state is entered, an interrupt will be generated; 0: When the sleep state is entered, an interrupt will not be generated;	0
16	WKUIE	RW	Wake-up interrupt enable bit. 1: When the CAN controller is awakened, an interrupt will be generated; 0: When the CAN controller is awakened, no interrupt will be generated.	0
15	ERRIE	RW	Error interrupt enable bit; CAN error interrupt total enable bit. 1: When the CAN controller generates an error, an interrupt will be generated; 0: When the CAN controller generates an error, no interrupt will be generated.	0
[14:12]	Reserved	RF	Reserved.	0
11	LECIE	RW	Previous error number interrupt enable bit. 1: When an error is detected, the hardware will update LEC[2:0] and update the ERRI bit to 1 to trigger an error interrupt; 0: When an error is detected, the hardware will update LEC[2:0], and will not update the ERRI bit and will not trigger the error interrupt.	0
10	BOFIE	RW	Offline interrupt enable bit. 1: When the offline state is entered, the ERRI bit will be updated to 1 to trigger the error interrupt; 0: When the offline state is entered, the ERRI bit	0

			will not be updated, and the error interrupt will not be triggered.	
9	EPVIE	RW	Error passive interrupt enable bit. 1: When the error passive state is entered, the ERRI bit will be updated to 1 to trigger the error interrupt; 0: When the error passive state is entered, the ERRI bit will not be updated, and the error interrupt will not be triggered.	0
8	EWGIE	RW	Error warning interrupt enable bit. 1: When the number of errors reaches the warning threshold, the ERRI bit will be updated to 1 to trigger 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	Receiver FIFO_1 overflow interrupt enable bit. 1: When FIFO_1 overflows, FIFO_1 interrupt will be triggered; 0: When FIFO_1 overflows, FIFO_1 interrupt will not be triggered.	0
5	FFIE1	RW	Receiver FIFO_1 full interrupt enable bit. 1: When FIFO_1 is full, FIFO_1 interrupt will be triggered; 0: When FIFO_1 is full, FIFO_1 interrupt will not be triggered.	0
4	FMPIE1	RW	Receiver FIFO_1 message register interrupt enable bit. 1: When FIFO_1 updates the FMP bit and it is not 0, FIFO_1 interrupt will be triggered; 0: When FIFO_1 updates the FMP bit and is not 0, the FIFO_1 interrupt will not be triggered.	0
3	FOVIE0	RW	Receiver FIFO_0 overflow interrupt enable bit. 1: When FIFO_0 overflows, FIFO_0 interrupt will be triggered; 0: When FIFO_0 overflows, FIFO_0 interrupt will not be triggered.	0
2	FFIE0	RW	Receiver FIFO_0 full interrupt enable bit. 1: When FIFO_0 is full, FIFO_0 interrupt will be triggered; 0: When FIFO_0 is full, FIFO_0 interrupt will not be triggered.	0
1	FMPIE0	RW	Receiver FIFO_0 message register interrupt enable bit. 1: When FIFO_0 updates the FMP bit and it is not	0

			0, FIFO_0 interrupt will be triggered; 0: When FIFO_0 updates the FMP bit and is not 0, the FIFO_0 interrupt will not be triggered.	
0	TMEIE	RW	Empty transmission mailbox interrupt. 1: When the sending mailbox is empty, an interrupt will be generated; 0: When the sending mailbox is empty, no interrupt will be generated.	0

CAN error status register (CAN_ERRSR)

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	RO	Receive error counter. When a CAN receive error occurs, the counter will be increased by 1 or 8 according to the error condition; after successful reception, the counter will be decreased by 1 or set to 120 (the error count value is greater than 127). When the counter value exceeds 127, CAN will enter an error passive state.	0
[23:16]	TEC	RO	Transmit error counter. When a CAN transmit error occurs, counter will be increased by 1 or 8 according to error condition; after successful transmission, the counter will be decreased by 1 or set to 120 (the error count value is greater than 127). When the counter value exceeds 127, CAN will enter an error passive state.	0
[15:7]	Reserved	RO	Reserved.	0
[6:4]	LEC{2:0}	RW	Last error code. When a transmit error on the CAN bus is detected, the controller will set it according to the error situation, and will set 000b when the message is sent and received correctly. 000: No error; 001: Bit stuffing error; 010: FORM error; 011: ACK acknowledge error; 100: Recessive bit error; 101: Dominant bit error; 110: CRC error;	0

			111: Software setting. When the application software reads the error, the code number will be set to 111b, and the code update can be detected.	
3	Reserved	RO	Reserved.	0
2	BOFF	RO	Offline state flag bit. When the CAN controller enters the offline state, the hardware will automatically set to 1; when the offline state is exited, the hardware will be automatically reset to 0.	0
1	EPVF	RO	Error passive flag bit. When the transceiving error counter reaches the error passive threshold, i.e., greater than 127, the hardware will be set to 1.	0
0	EWGF	RO	Error warning flag bit. When the transceiving error counter reaches the warning threshold, i.e., greater than 96, the hardware will be set to 1.	0

CAN bit timing register (CAN_BTIMR)

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 bit. 1: Enter the silent mode; 0: Exit the silent mode.	0
30	LBKM	RW	Loopback mode setting bit. 1: Enter the loopback mode; 0: Exit the loopback mode.	0
[29:26]	Reserved	RO	Reserved.	0
[25:24]	SJW	RW	Define resynchronization jump width setting value. When the resynchronization is realized, the upper limit of the minimum time unit that can be extended and reduced in the bit, the actual value will be (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	RW	Setting value of time segment 2 It defines the number of minimum time units occupied by time segment 2, and the actual value is	010b

			(TS2[1:0]+1).	
[19:16]	TS1	RW	Setting value of time segment 1. It defines the number of minimum time units occupied by time segment 1, and the actual value is (TS1[1:0]+1).	0011b
[15:10]	Reserved	RO	Reserved.	0
[9:0]	BRP	RW	Minimum time unit length setting value Tq = (BRP[9:0]+1) × t_{pclk}	0

CAN transmit mailbox identifier register (CAN_TXMIRx) (x=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_EXIDH	RW	The higher 11 bits of the standard identifier or extended identifier.	X
[20:3]	EXIDL	RW	The lower 18 bits of the extended identifier.	X
2	IDE	RW	Identifier selection flag bit. 1: Select the extended identifier; 0: Select the standard identifier.	X
1	RTR	RW	Remote frame (also called remote control frame) selection flag. 1: Remote frame; 0: Data frame.	X
0	TXRQ	RW	Data transmission request flag bit. When it is set to 1 by software, data in the mailbox will be requested to be sent. When the mailbox is empty, the hardware will be cleared to 0.	0

CAN transmit mailbox data length and time stamp register (CAN_TXMDTRx) (x=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	RW	16-bit timer value used to send the message SOF	X

			moment	
[15:9]	Reserved	RO	Reserved.	X
8	TGT	RW	The message timestamp transmission selection flag bit. This bit is set to 1 in TTCM and valid when the message length is 8. 1: Transmission time stamp; the value is the instant value of TIME[15:0], which replaces the last two bytes of the 8-byte message; 0: Not sending time stamp.	X
[7:4]	Reserved	RO	Reserved.	X
[3:0]	DLC	RW	Data length of data frame or remote frame request data length The settable range of data length is from 0 to 8.	X

CAN transmit mailbox low byte data register (CAN_TXMDLRx) (x=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	RW	Contents of transmission data byte 3.	X
[23:16]	DATA2	RW	Content of transmission data byte 2.	X
[15:8]	DATA1	RW	Content of transmission data byte 1.	X
[7:0]	DATA0	RW	Content of transmission data byte 0.	X

CAN transmission mailbox high byte data register (CAN_TXMDHRx) (x=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	RW	Content of transmission data byte 7.	X
[23:16]	DATA6	RW	Content of transmission data byte 6.	X
[15:8]	DATA5	RW	Content of transmission data byte 5.	X
[7:0]	DATA4	RW	Content of transmission data byte 4.	X

CAN receive mailbox identifier register (CAN_RXMIRx) (x=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_EXIDH	RO	The higher 11 bits of the standard identifier or extended identifier.	X
[20:3]	EXIDL	RO	The lower 18 bits of the extended identifier.	X
2	IDE	RO	Identifier selection flag bit. 1: Select the extended identifier; 0: Select the standard identifier.	X
1	RTR	RO	Remote frame (also called remote frame) selection flag. 1: It is a remote frame currently; 0: It is a data frame currently.	X
0	Reserved	RO	Reserved.	X

CAN receive mailbox data length and time stamp register (CAN_RXMDTRx) (x=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	RO	The 16-bit timer value used to receive the message SOF moment.	X
[15:8]	FMI	RO	Number of filter matching the message.	X
[7:4]	Reserved	RO	Reserved.	X
[3:0]	DLC	RO	Reception message data length. Data frame length 0 to 8, remote frame: 0.	X

CAN receive mailbox low byte data register (CAN_RXMDLRx) (x=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 reception message	X
[23:16]	DATA2[7:0]	RO	Data byte 2 of reception message.	X
[15:8]	DATA1[7:0]	RO	Data byte 1 of reception message	X
[7:0]	DATA0[7:0]	RO	Data byte 0 of reception message.	X

CAN receive mailbox high byte data register (CAN_RXMDHRx) (x=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 reception message.	X
[23:16]	DATA6[7:0]	RO	Data byte 6 of reception message.	X
[15:8]	DATA5[7:0]	RO	Data byte 5 of reception message.	X
[7:0]	DATA4[7:0]	RO	Data byte 4 of reception message.	X

CAN filter master control register (CAN_FCTLR)

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															FINT

Bit	Name	Access	Description	Reset value
[31:1]	Reserved	RO	Reserved.	Reset value
0	FINT	RW	Filter initialization mode enable flag bit. 1: The filter group is in initilization mode; 0: The filter group is in normal mode.	1

CAN filter mode register (CAN_FMCFGR)

Offset address: 0x204

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	FBM13	FBM12	FBM11	FBM10	FBM9	FBM8	FBM7	FBM6	FBM5	FBM4	FBM3	FBM2	FBM1	FBM0	

Bit	Name	Access	Description	Reset value
[31:14]	Reserved	RO	Reserved.	0
[13:0]	FBMx	RW	Working mode flag bit of filter group x can be written when FINT is 1. 1: Register of filter group x is in mask bit mode; 0: Register of filter group x is in identifier list mode.	0

CAN filter bit width register (CAN_FSCFGR)

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		FSC13	FSC12	FSC11	FSC10	FSC9	FSC8	FSC7	FSC6	FSC5	FSC4	FSC3	FSC2	FSC1	FSC0

Bit	Name	Access	Description	Reset value
[31:14]	Reserved	RO	Reserved.	0
[13:0]	FSCx	RW	Bit width flag of filter group x can be written when FINT is 1. 1: The register of filter group x is a single 32-bit; 0: The register of filter group x is two 16 bits.	0

CAN filter FIFO associated register (CAN_FAFIFOR)

Offset address: 0x214

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		FFA13	FFA12	FFA11	FFA10	FFA9	FFA8	FFA7	FFA6	FFA5	FFA4	FFA3	FFA2	FFA1	FFA0

Bit	Name	Access	Description	Reset value
[31:14]	Reserved	RO	Reserved.	0
[13:0]	FFAx	RW	Associated FIFO flag bit of filter group x; it can be written when FINT is 1. 1: Filter group x is associated with FIFO_1; 0: Filter group x is associated with FIFO_0.	0

CAN filter activation register (CAN_FWR)

Offset address: 0x21C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	FACT13	FACT12	FACT11	FACT10	FACT9	FACT8	FACT7	FACT6	FACT5	FACT4	FACT3	FACT2	FACT1	FACT0	

Bit	Name	Access	Description	Reset value
[31:14]	Reserved	RO	Reserved.	0
[13:0]	FACTx	RW	Activation flag bit of filter group x; it can be written when FINT is 1. 1: Filter group x is activated; 0: Filter group x is disabled.	0

Filter register of CAN filter group (CAN_FiRx) (i=0-13, x=0/1)

Offset address: 0x240-0x31C

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	Flag bit of the register in the filter group, can be written only when FINT is 1. Identifier mode 1: The expected level of the corresponding bit is recessive bit; 0: The expected level of the corresponding bit is dominant bit. Mask bit mode 1: It must be consistent with the corresponding identifier register; 0: It does not need to be consistent with the corresponding identifier register bit.	0

Chapter 23 Electronic Signature (ESIG)

The module description in this chapter applies to the full range of CH32F103 and CH32V103 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 (RVSWD) or application code.

23.1 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.

23.2 Register Description

Table 23-1 List of ESIG Related Registers

Name	Access address	Description	Reset value
R16_ESIG_FLACAP	0x1FFFF7E0		
R32_ESIG_UNIID1	0x1FFFF7E8	UID register 1	0xXXXXXXXX
R32_ESIG_UNIID2	0x1FFFF7EC	UID register 2	0xXXXXXXXX
R32_ESIG_UNIID3	0x1FFFF7F0	UID register 3	0xXXXXXXXX

Flash memory 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 memory capacity in unit of Kbyte. Example: 0x0080 = 128 K bytes	X

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	RO	0-30 bits of UID	X

UID register (ESIG_UNIID2)

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

Bit	Name	Access	Description	Reset value
[31:0]	U_ID	RO	32-63 bits of UID.	X

UID register (ESIG_UNIID3)

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

Bit	Name	Access	Description	Reset value
[31:0]	U_ID	RO	64-95 bits of UID.	X

Chapter 24 Flash Memory and User Selection Word

The module description in this chapter applies to the full range of CH32F103 and CH32V103 microcontrollers.

24.1 Flash Memory Organization

The internal flash memory organization structure of the chip is as follows (taking xR8T6 as an example):

Table 24-1 Flash Memory Organization Structure

Block	Name	Address range	Size (byte)
Main memory	Page 0	0x08000000 – 0x0800007F	128
	Page 1	0x08000080 – 0x080000FF	128
	Page 2	0x08000100 – 0x0800017F	128
	Page 3	0x08000180 – 0x080001FF	128
	Page 4	0x08000200 – 0x0800027F	128
	Page 5	0x08000280 – 0x080002FF	128
	Page 6	0x08000300 – 0x0800037F	128
	Page 7	0x08000380 – 0x080003FF	128
	…	…	…
	Page 511	0x0800FF80 – 0x0800FFFF	128
Information block	System bootloader storage 1	0x1FFFF000 – 0x1FFFF7FF	2K
	User selection word	0x1FFFF800 – 0x1FFFF87F	128
	Vendor configuration word	0x1FFFF880 – 0x1FFFF8FF	128
	System bootloader storage 2	0x1FFFF900 – 0x1FFFFFFF	1792

The main memory area aforesaid 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.

24.2 Flash Memory Programming and Safety

1) Two Programming / Erasure 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 1K byte.
- Fast programming: The page operation mode (recommended) is used for this method. After unlocking in a specific sequence, a single 128-byte programming and 128-byte erasing are performed.

2) Security-Preventing against Illegal Access (read, write and erasure)

- Page write protection
- Read protection

Under the read protection state:

1) The main memory pages 0-31 (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 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.

24.3 Register Description

Table 24-2 List of FLASH Related Registers

Name	Access address	Description	Reset value
R32_FLASH_ACTLR	0x40022000	Access control register	0x00000030
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	Configuration register	0x00000080
R32_FLASH_ADDR	0x40022014	Address register	X
R32_FLASH_OBR	0x4002201C	Selection word register	0x03FFFFFC
R32_FLASH_WPR	0x40022020	Write protection register	0xFFFFFFF
R32_FLASH_MODEKEYR	0x40022024	Extension key register	X

Access control register (FLASH_ACTLR)

Offset address: 0x00

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

Bit	Name	Access	Description	Reset value
[31:6]	Reserved	RO	Reserved.	0
5	PRFTBS	RO	Pre-fetch buffer status: 1: Enable the pre-fetch buffer; 0: Disable the pre-fetch buffer.	1
4	PRFTBE	RW	Pre-fetch buffer enable 1: Enable the pre-fetch buffer; 0: Disable the pre-fetch buffer.	1
3	Reserved	RO	Reserved.	0
[2:0]	LATENCY	RW	Delay. The ratio of system clock (SYSCLK) to flash memory access time: 000: Zero waiting; it recommended that	000b

			0<SYSCLK≤24MHz; 001: 1 piece of waiting; it is recommended that 24MHz<SYSCLK≤48MHz; 010: 2 pieces of waiting; it is recommended that 48MHz <SYSCLK≤72MHz.	

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	WO	FPEC key; the unlock key used to enter FPEC includes: RDPRT key = 0x000000A5; KEY1 = 0x45670123; KEY2 = 0xCDEF89AB.	X

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	WO	Selection word key; used to input the selection word key to release OPTWRE.	X

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										EOP	WRPRTERR	Reserved	PGERR	Reserved	BSY

Bit	Name	Access	Description	Reset value
[31:6]	Reserved	RO	Reserved.	0

5	EOP	RW1	End of the 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, must be kept at clear status '0'.	
2	PGERR	RW1	Program error, write 1 to clear it. It will be set by hardware when programming the address with write protection.	
1	Reserved	RO	Reserved, must be kept at clear status '0'.	0
0	BSY	RO	Busy state: 1: Flash memory operation is in the process; 0: Operation completion or error generation.	0

Note: When performing the programming operation, you need to make sure that the STRT bit of the FLASH_CTLR register is set to 0.

Configuration register (FLASH_CTLR)

Offset address: 0x10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												BUF RST	BUF LOAD	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	PER	PG

Bit	Name	Access	Description	Reset value
[31:20]	Reserved	RO	Reserved.	0
19	BUFRST	RW	Clear the internal buffer data.	0
18	BUFLOAD	RW	Load data into the internal buffer.	0
17	FTER	RW	Fast page (128Byte) erase operation	0
16	FTPG	RW	Fast 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	Operation completion interrupt control (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 control (PGERR/WRPRTERR is set in the FLASH_STATR register): 1: Enable to generate interrupt; 0: Disable to generate interrupt.	0

9	OBWRE	RW0	User selection word lock; clear it by the software: 1: Indicates that the user-selected word can be programmed. It needs to be set by hardware after the correct sequence is written in the FLASH_OBKEYR register. 0: Re-lock the user-selected word after clearing by software.	0
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 erasure or programming action, and the hardware will automatically clear it to 0 (BSY becomes ‘1’).	0
5	OBER	RW	Execute the user selection word erasure	0
4	OBPG	RW	Execute the user selection word 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 (1 KB) erasure operation.	0
0	PG	RW	Execute the standard programming operation.	0

Address register (FLASH_ADDR)

Offset address: 0x14

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

Bit	Name	Access	Description	Reset value
[31:0]	ADDR	WO	Erased flash address. When the BSY bit in the FLASH_SR register is '1', this register cannot be written.	X

Selection word 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		POR CTR	USBD PU	USBD MODE	STANDY RST	STOP RST	IWDG SW	RDPRT	OBERR

Bit	Name		Access	Description	Reset value
[31:10]	Reserved		RO	Reserved.	0
9	USER	Reserved	RO	Not used.	X
8					
7		POR_CTR	RO	Power-on reset time.	X
6		USBD_PU	RO	USBD (compatible) internal pull-up resistor configuration.	X
5		USBD_MODE	RO	USBD (compatible) speed mode configuration.	X
4		STANDY_RST	RO	System reset control under the standby mode.	X
3		STOP_RST	RO	System reset control under the stop mode.	X
2		IWDG_SW	RO	Independent watchdog (IWDG) hardware enable bit.	X
1	RDPRT		RO	Read protection status. 1: Current read protection of flash memory is valid.	X
0	OBERR		RO	Selection word error. 1: Selection word does not match its inverted code.	X

Note: USER and RDPRT are loaded from the user-selected word area after system reset.

Write protection register (FLASH_WPR)

Offset address: 0x20

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

Bit	Name	Access	Description	Reset value
[31:0]	WPR	RO	Flash memory write protection status. 1: Write protection failure; 0: Valid write protection. Each bit represents 4K bytes (32 pages) to store the write protection status.	X

Note: WPR is loaded from the user-selected word area after system reset.

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	WO	Input the following sequence to unlock the fast programming/erasure mode: KEY1 = 0x45670123; KEY2 = 0xCDEF89AB.	X

24.4 Flash Memory Operation Process Flow

24.4.1 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.

24.4.2 Flash Memory Unlock

After the system is 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.

Unlocking sequence:
1) Write KEY1 = 0x45670123 to the FLASH_KEYR register (the first step must be KEY1);
2) Write KEY2 = 0xCDEF89AB to the FLASH_KEYR register (the second step must be KEY2).

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.

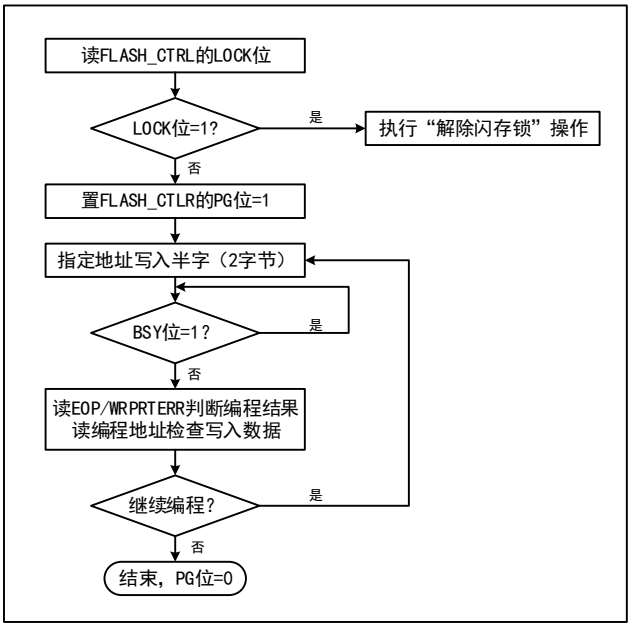
The flash memory controller (FPEC) and the FLASH_CTLR register can be locked again by setting the "LOCK" bit of the FLASH_CTLR register to 1.

24.4.3 Main Memory Standard Programming

You can write 2 bytes each time through the standard programming. When the PG bit of 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 24-1 FLASH Programming

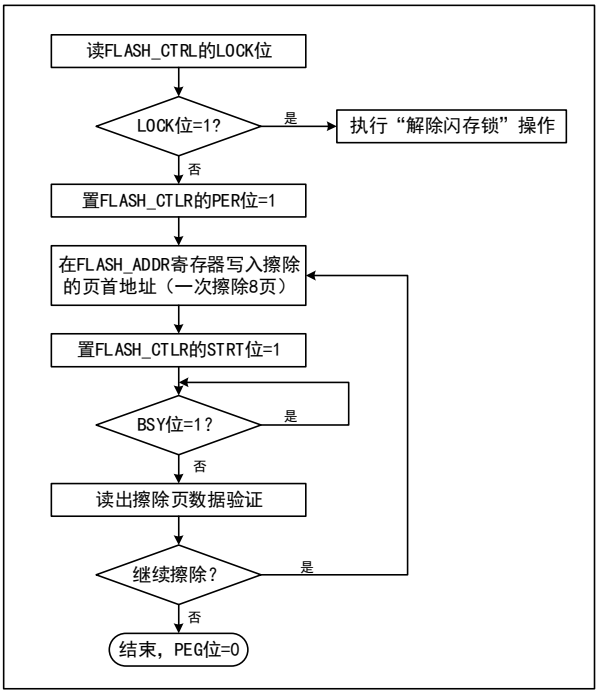


1) Check the FLASH_CTLR register LOCK. If it is 1, you need to perform the "Release Flash Memory Lock" operation.
2) Set the PG bit of the FLASH_CTLR register to ‘1’ to enable the standard programming mode.
3) Write the half word to be programmed to the designated flash memory address (even address).
4) When the BYS bit changes to '0' or the EOP bit of 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.

24.4.4 Main Memory Standard Erasure

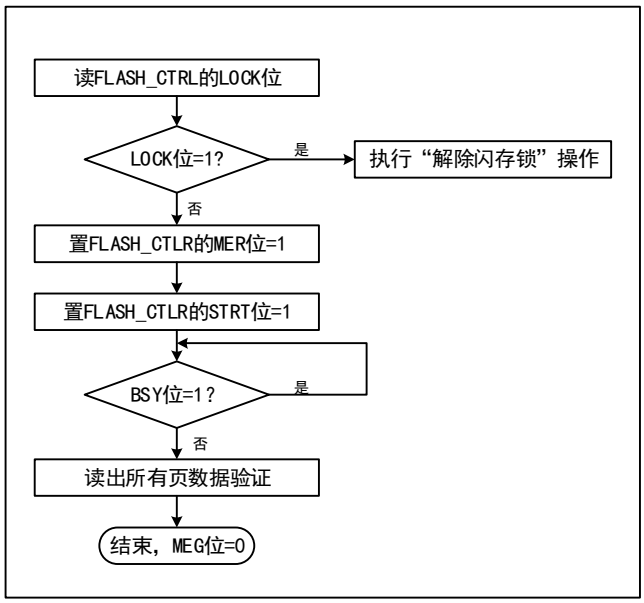
The flash memory can be erased in the standard pages (1K bytes) or in whole chips.

Figure 24-2 FLASH Page Erasure



1) Check the FLASH_CTLR register LOCK bit. If it is 1, you need to perform the "Release Flash Memory Lock" operation.
2) Set the PEG bit of 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 of the FLASH_CTLR register to ‘1’ to start an erase action.
5) When the BYS bit changes to '0' or the EOP bit of 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.

Figure 24-1 FLASH Full Chip Erasure



1) Check the FLASH_CTLR register LOCK bit. If it is 1, you need to perform the "Release Flash Memory Lock" operation.
2) Set the MEG bit of the FLASH_CTLR register to ‘1’ to enable the whole chip erasure mode.
3) Set the STAT bit of the FLASH_CTLR register to ‘1’ to start an erasure action.
4) When the BYS bit changes to '0' or the EOP bit of 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 MEG bit to 0.

24.4.5 Fast Programming Molde 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.

Unlocking 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 two layers of "LOCK" and "FLOCK".

24.4.6 Main Memory Fast Programming

The fast programming (128 bytes) is made according to the page. The system has a built-in 128-byte buffer. The data line to be programmed is saved in the buffer and a programming operation is performed, which is more efficient.

1) Check the FLASH_CTLR register LOCK bit. 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 FLASH_CTLR register FLOCK bit. If it is 1, you need to perform the "Fast Programming Mode Unlock" operation.
4) Set the FTPG bit in the FLASH_CTLR register to enable the fast programming mode function.
5) Set the BUFRST bit of the FLASH_CTLR register and execute the operation to clear the internal 128-byte buffer.
6) When the BYS bit changes to '0' or the EOP bit of the FLASH_STATR register to be '1', it indicates the end of clearing. Clear the EOP bit to 0.
7) Continuously write 16 bytes of data to the specified address (4 bytes per operation; the offset of the written address is 4 each time), and then set the BUFLOAD bit of the FLASH_CTLR register to load it into the buffer.
8) When the BYS bit changes to '0' or the EOP bit of the FLASH_STATR register to be '1', it indicates the end of loading. Clear the EOP bit to 0.
9) Repeat steps 7-8 for a total of 8 times to load all 128 bytes of data into the buffer (the main 8 rounds of operation address shall be continuous).
10) Write the page heading address of the fast programming page to the FLASH_ADDR register.
11) Set the STAT bit of the FLASH_CTLR register to ‘1’ to start the fast page programming action.
12) When the BYS bit changes to '0' or the EOP bit of the FLASH_STATR register to be '1', it indicates the end of programming. Clear the EOP bit to 0.
13) Check the FLASH_STATR register to see if there is an error, or read the programming address data for verification.
14) To continue the fast page programming, you can repeat steps 5-13, end programming and clear the FTPG bit to 0.

24.4.7 Main Memory Fast Erasure

Quick erasure is also performed according to the pages (128 bytes).

1) Check the FLASH_CTLR register LOCK bit. 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 FLASH_CTLR register FLOCK bit. If it is 1, you need to perform the "Fast Programming Mode Unlock" operation.
4) Set the FTER bit in the FLASH_CTLR register to enable the fast erasure mode function.
5) Write the page heading address of the fast erasure page to the FLASH_ADDR register.
6) Set the STAT bit of the FLASH_CTLR register to ‘1’ to start the fast page erasure action.
7) When the BYS bit changes to '0' or the EOP bit of the FLASH_STATR register to be '1', it indicates the end of erasure. Clear the EOP bit to 0.

8) Check the FLASH_STATR register to see if there is an error, or read the erasure page address data for verification.
9) To continue fast page erasure, you can repeat steps 5-8 to end erasing and clear the FTER bit to 0.

24.5 User Selection Word

The user-selected word is solidified in FLASH and will be reloaded into the corresponding register after the system is 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 24-3 32-bit Selection Word Format Division

[31:24]	[23:16]	[15:8]	[7:0]
Inverse code of selection word byte 1	Selection word byte 1	Inverse code of selection word byte 0	Selection word byte 0

Table 24-4 User Selection Word 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 bit; configure 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: Indicate the code read protection status, unreadable; pages 0-31 (4K) will be automatically write-protected and not controlled by WRPR0.	0x01
USER	[7:6]	Reserved	Reserved.	11b
	5	POR_CTR	Power-on reset time configuration: 1: Reset time: 16.384ms; 0: Reset time: 40.96ms.	1
	4	USBD_PU	USBD (compatible) internal pull-up resistor configuration; 1: Disable the USBD internal pull-up resistor (recommended); 0: Enable the USBD internal pull-up resistor. *Note: CH32V103 does not have this configuration value, and this bit is reserved.*	1

	3	USBD_MODE	USBD (compatible) speed mode configuration: 1: Full speed mode USBD (recommended); 0: Low speed mode USBD. *Note: CH32V103 does not have this configuration value, and this bit is reserved.*	1
	2	STANDY_RST	System reset control in standby mode: 1: Disable; system will not be reset when entering standby mode; 0: Enable; system will be reset when entering standby mode.	1
	1	STOP_RST	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	IWDG_SW	Independent watchdog (IWDG) hardware enable bit: 1: The IWDG function is enabled by the software, and disabled by hardware; 0: The IWDG function is enabled by the software (decided along with the LSI clock).	1
Data0–Data1			Saving the user's data 2 bytes.	FFFFh
WRPR0 - WRPR3			Write protection control bit. Each bit is used to control the write protection status of 4K bytes in the main memory: 1: Disable the write protection; 0: Enable the write protection. 4 bytes are used to protect a total of 128K bytes of main memory. WRPO: 0–32K bytes address storage write protection control; WRP1: 32K-64K bytes address storage write protection control; WRP2: 64K-96K bytes address storage write protection control; WRP3: 96K-128K byte address storage write protection control.	X

Note: After the USBD_PU bit is set to 0, the function of the extended register will not effective (hardware action).

24.5.1 User Selection Word Unlock

The user selection word operation can be unlocked by writing the sequence to the FLASH_OBKEYR register. After unlocking, the OBWRE bit of the FLASH_CTLR register will be set to 1, indicating that user-selected words can be erased and programmed. By setting the "OBWRE" bit of the FLASH_CTLR register, the software will be set to 0 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 two layers: "LOCK" and "OBWRE" for word selection.

24.5.2 User Selection Word 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-selected word, 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 the byte in the user-selected word and its inverse code are always correct.

1) Check the FLASH_CTLR register LOCK bit. 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 FLASH_CTLR register OBWRE bit. If it is 0, you need to perform the "User Selection Word Unlock" operation.
4) Set the OBPG bit of the FLASH_CTLR register to ‘1’ to enable the user selection word 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 of 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 modified selection word 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.

24.5.3 User Selection Word Erasure

Erase the entire 128-byte user-selected word area directly.

1) Check the FLASH_CTLR register LOCK bit. 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 FLASH_CTLR register OBWRE bit. If it is 0, you need to perform the "User Selection Word Unlock" operation.
4) Set the OBER bit of the FLASH_CTLR register to ‘1’ to enable the user selection word erasure.
5) When the BYS bit changes to '0' or the EOP bit of 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.

24.5.4 Read Protection Release

The read protection of flash memory is determined by the user's selection of words. 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 selection word area. At this time, the read protection field RDPR will become 0xFF, and the read protection will be still valid.
2) The user selects word 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.

24.5.5 Write Protection Release

The write protection of flash memory is determined by the user's selection of words. Read the FLASH_WPR register. Each bit represents 4K bytes of flash memory space. When the bit is '1', it means the non-write-protection state, and '0' means write protection. The process of releasing write protection is as follows:

1) Erase the whole user selection word 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.