CH32V003 Reference Manual

V1.5
https://wch-ic.com

Overview

CH32V003 series are industrial-grade general-purpose microcontrollers designed based on 32-bit RISC-V instruction set and architecture. It adopts QingKe V2A core, RV32EC instruction set, and supports 2 levels of interrupt nesting. The series are mounted with rich peripheral interfaces and function modules. Its internal organizational structure meets the low-cost and low-power embedded application scenarios.

This manual provides detailed information on the use of the CH32V003 series for the user's application development, and is applicable to products with different memory capacities, functional resources, and packages in the series; any differences will be specially explained in the corresponding functional chapters.

Please refer to the Datasheet *CH32V003DS0* for the device characteristics of this product.
For information about the core, please refer to the *QingKeV2_Processor_Manual*.

RISC-V core version overview

Features / Core versions	Instruction set	Hardware stack levels	Interrupt nesting levels	Fast interrupt channels	Flow line	Vector table model	Extensions instruction	Debug interface
QingKe V2A	RV32EC	2	2	2	2	Address or command	Support	1-wire

Abbreviated description of the bit attribute in the register:

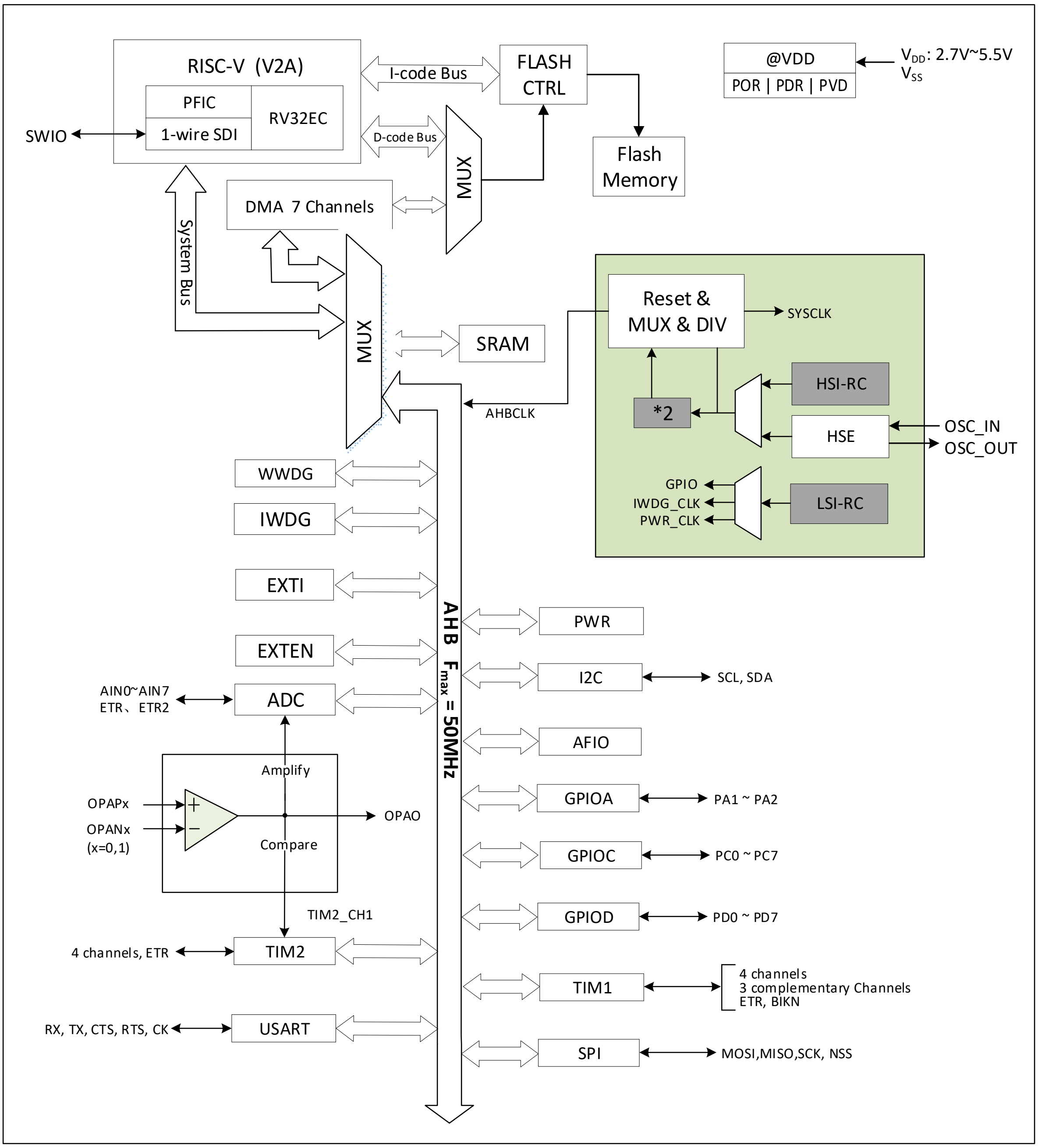
Register bit properties	Property description
RF	Read-only property that reads a fixed value.
RO	Read-only attribute, changed by hardware.
RZ	Read-only property, auto bit clear 0 after read operation.
WO	Write only attribute (not readable, read value uncertain)
WA	Write-only attribute, writable in Safe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

Chapter 1 Memory and Bus Architecture

1.1 Bus Architecture

The CH32V003 series is designed based on the RISC-V instruction set, and its architecture interacts the core, arbitration unit, DMA module, SRAM storage and other parts through multiple buses. The design integrates a general-purpose DMA controller to reduce the CPU load and improve access efficiency, as well as data protection mechanisms, automatic clock switching protection mechanisms and other measures to increase system stability. The system block diagram is shown in Figure 1-1.

Figure 1-1 CH32V003 system block diagram



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

- The instruction bus (I-Code) connects the core to the FLASH instruction interface and prefetching is done on this bus.

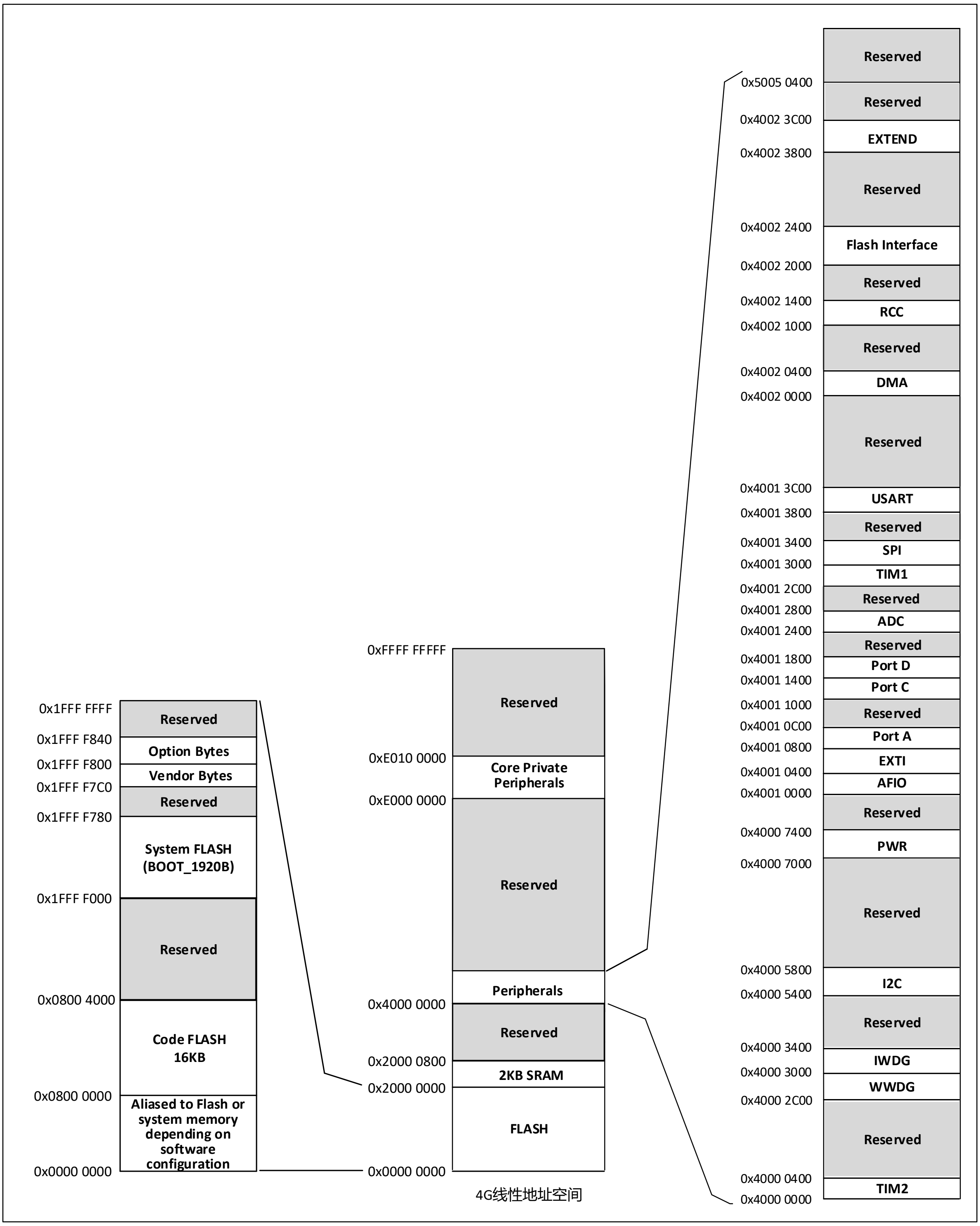
- The data bus (D-Code) connects the core to the FLASH data interface for constant loading and debugging.
- The system bus connects the core to the bus matrix and is used to coordinate accesses to the core, DMA, SRAM and peripherals.
- The DMA bus is responsible for the DMA of the AHB master interface connected to the bus matrix, which is accessed by FLASH data, SRAM and peripherals.
- The bus matrix is responsible for the access coordination between the system bus, data bus, DMA bus, SRAM and AHB bridge.

1.2 Memory Image

The CH32V003 family contains program memory, data memory, core registers, peripheral registers, and more, all addressed in a 4GB linear space.

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

Figure 1-2 Storage image



1.2.1 Memory Allocation

Built-in 2KB SRAM, starting address 0x20000000, supports byte, half-word (2 bytes), and full-word (4 bytes) access.

Built-in 16KB program Flash memory (CodeFlash) for storing user applications.

Built-in 1920B System memory (bootloader) for storing the system bootloader (factory-cured bootloader).

Built-in 64B space for vendor configuration word storage, factory-cured and unmodifiable by users.

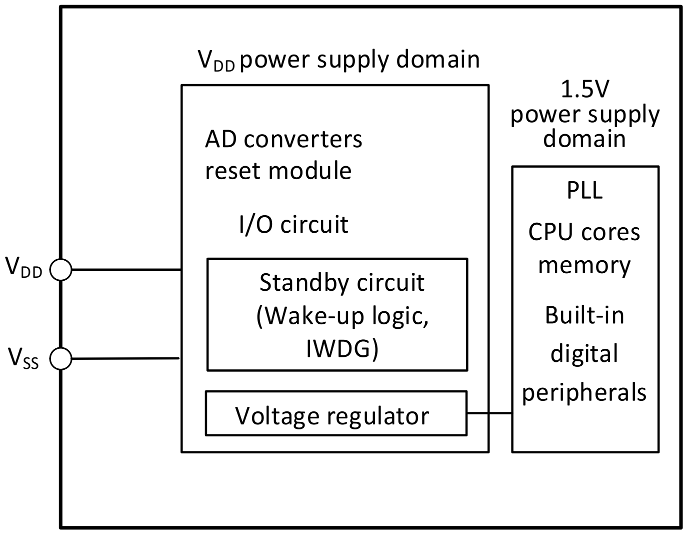
Built-in 64B space for user-option bytes storage.

Chapter 2 Power Control (PWR)

2.1 Overview

The system operating voltage V_{DD} ranges from 2.7 to 5.5V, and the built-in voltage regulator provides the 1.5V power supply required by the core.

Figure 2-1 Block diagram of power supply structure

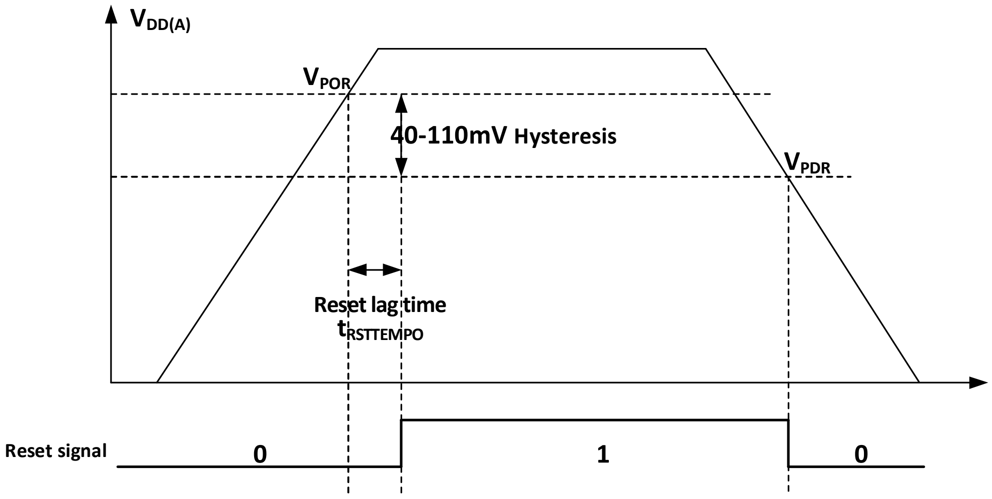


2.2 Power Management

2.2.1 Power-on Reset and Power-down Reset

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

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



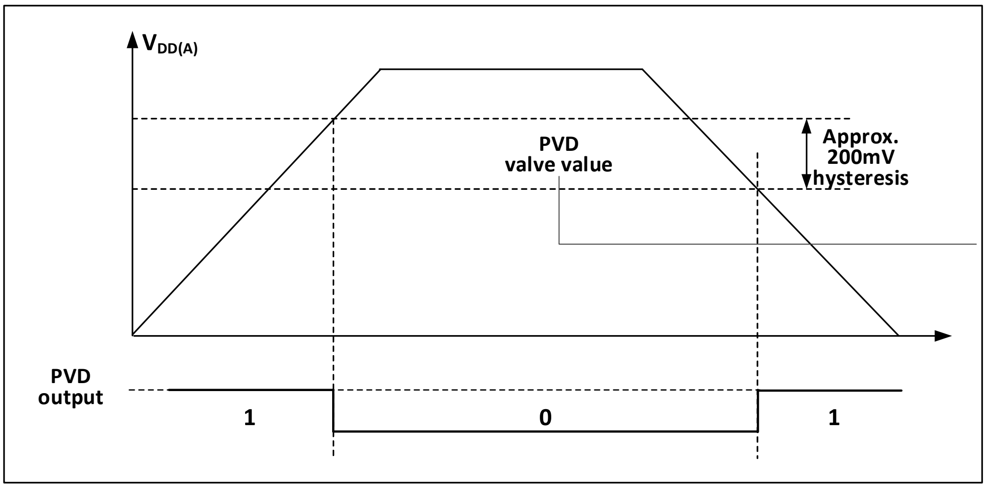
2.2.2 Programmable Voltage Detector

The programmable voltage monitor, PVD, is mainly used to monitor the change of the main power supply of the system and compare it with the threshold voltage set by PLS[2:0] of the power control register PWR_CTLR, and with the external interrupt register (EXTI) setting, it can generate relevant interrupts to notify the system in time for pre-power down operations such as data saving.

The specific configuration is as follows.

1) Set the PLS[2:0] field of the PWR_CTLR register to select the voltage threshold to be monitored.
2) Optional interrupt handling. the PVD function internally connects to the rising/falling edge trigger setting of line 8 of the EXTI module, turns on this interrupt (configures EXTI), and generates a PVD interrupt when VDD drops below the PVD threshold or rises above the PVD threshold.
3) Set the PVDE bit of PWR_CTLR register to enable the PVD function.
4) Read the PVD0 bit of PWR_CSR status register to obtain the current system main power and PLS[2:0] setting threshold relationship, and perform the corresponding soft processing.

Figure 2-3 Schematic diagram of PVD operation



2.3 Low-power Modes

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

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

- Sleep mode: The core stops running and all peripherals (including core private peripherals) are still running.
- Standby mode: Stop all clocks, wake up and switch the clock to HSI.

Table 2-1 Low-power mode list

Mode	Entry	Wake-up source	Effect on clock	Voltage regulator
Sleep	WFI	Any interrupt	Core clock OFF, no effect on other clocks	ON
	WFE	Wake-up event		
Standby	Set SLEEPDEEP to 1 Set PDDS to 1 WFI or WFE	AWU event *Note: Any event can also wake up the system, but the system does not reset after waking up.*	HSE, HSI, PLL and peripheral clock OFF	OFF

Note: The SLEEPDEEP bit belongs to the core private peripheral control bit, CH32V003 product reference PFIC_SCTLR register.

2.3.1 Low-power Configuration Options

- WFI and WFE

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

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

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

- SLEEPONEXIT

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

  Not enabled: The microcontroller enters low-power mode immediately after executing the WFI or WFE command.

- SEVONPEND

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

  Not enabled: Only interrupts or wake-up events enabled in the interrupt controller can wake up the low-power consumption entered by executing WFE.

2.3.2 Sleep Mode

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

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

Exit: Arbitrary interrupt or wakeup event.

2.3.3 Standby Mode

Standby mode is a combination of peripheral clock control mechanisms based on the core's deep Sleep mode (SLEEPDEEP) and allows the voltage regulator to operate at a much lower-power consumption. This mode has the high frequency clock (HSE/HSI/PLL) domain turned off, the SRAM and register contents held, and the I/O pin state held. The system can continue to run after this mode wakes up, and the HSI is called the default system clock.

If flash programming is in progress, the system does not enter Standby mode until access to memory is complete.

Standby mode can work modules: Independent Watchdog (IWDG), Low Frequency Clock (LSI).

Enter: Configure the core register control bit SLEEPDEEP=1, PDDS=1 in the power control register, and execute WFI or WFE, optionally SEVONPEND and SLEEPONEXIT.

Exit:

1) Any interrupt/event (set in the external interrupt register).
2) AWU event, clock switches to HSI after this wakeup, system does not reset.

2.3.4 Auto-wakeup (AWU)

Auto-wakeup without external interrupts can be implemented. The time base can be programmed to wake up periodically from Standby mode.

A precision internal low frequency 128KHz crystal LSI can be selected as the auto-wakeup count time base.

2.4 Register Description

Table 2-2 PWR-related registers list

Name	Access address	Description	Reset value
R32_PWR_CTLR	0x40007000	Power control register	0x00000000
R32_PWR_CSR	0x40007004	Power control/status register	0x00000000
R32_PWR_AWUCSR	0x40007008	Auto-wakeup control/status register	0x00000000
R32_PWR_AWUWR	0x4000700C	Auto-wakeup window comparison value register	0x0000003f
R32_PWR_AWUPSC	0x40007010	Auto-wakeup crossover factor register	0x00000000

2.4.1 Power Control Register (PWR_CTLR)

Offset address: 0x00

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

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

Bit	Name	Access	Description	Reset value
[31:8]	Reserved	RO	Reserved	0
[7:5]	PLS[2:0]	RW	PVD voltage monitoring threshold setting. See the Electrical Characteristics section of the datasheet for detailed instructions. 000: 2.85V rising edge/2.7V falling edge. 001: 3.05V rising edge/2.9V falling edge. 010: 3.3V rising edge/3.15V falling edge. 011: 3.5V rising edge/3.3V falling edge. 100: 3.7V rising edge/3.5V falling edge. 101: 3.9V rising edge/3.7V falling edge. 110: 4.1V rising edge/3.9V falling edge. 111: 4.4V rising edge/4.2V falling edge.	0
4	PVDE	RW	Power supply voltage monitoring function enable flag bit 1: Enable the power supply voltage monitoring function. 0: Disable the power supply voltage monitoring function.	0
[3:2]	Reserved	RO	Reserved	0
1	PDDS	RW	Standby/ Sleep mode selection bit in power-down deep sleep scenario. 1: Enter Standby mode. 0: Enter Sleep mode.	0
0	Reserved	RO	Reserved	0

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

2.4.2 Power Control/Status Register (PWR_CSR)

Offset address: 0x04

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

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

Bit	Name	Access	Description	Reset value
[31:3]	Reserved	RO	Reserved	0
2	PVD0	RO	PVD output status flag bit. This bit is valid when PVDE=1 of PWR_CTLR register. 1: VDD and VDDA are below the PVD threshold set by PLS[2:0]. 0: VDD and VDDA are above the PVD threshold set by PLS[2:0].	0
[1:0]	Reserved	RO	Reserved	0

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

2.4.3 Auto-wakeup Control/Status Register (PWR_AWUCSR)

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														AWUEN	Reserved

Bit	Name	Access	Description	Reset value
[31:2]	Reserved	RO	Reserved	0
1	AWUEN	RW	Enable Automatic wake-up 1: Turn on auto-wakeup; 0: Invalid.	0
0	Reserved	RO	Reserved	0

2.4.4 Auto-wakeup Window Comparison Value Register (PWR_AWUWR)

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											AWUUWR[5:0]				

Bit	Name	Access	Description	Reset value
[31:6]	Reserved	RO	Reserved	0
[5:0]	AWUWR[5:0]	RW	AWU window value, which is used to compare with the recursive counter value and generate a wake-up signal when the counter value is equal to the window value.	0x3f

2.4.5 Auto-wakeup Crossover Factor Register (PWR_ AWUPSC)

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												AWUPSC[3:0]			

Bit	Name	Access	Description	Reset value
[31:4]	Reserved	RO	Reserved	0
[3:0]	AWUPSC[3:0]	RW	Counting time base 0000: Prescaler off. 0001: Prescaler off. 0010: Divided by 2. 0011: Divided by 4. 0100: Divided by 8. 0101: Divided by 16. 0110: Divided by 32. 0111: Divided by 64. 1000: Divided by 128. 1001: Divided by 256. 1010: Divided by 512. 1011: Divided by 1024. 1100: Divided by 2048. 1101: Divided by 4096. 1110: Divided by 10240. 1111: Divided by 61440.	0

Chapter 3 Reset and Clock Control (RCC)

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

3.1 Main Features

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

3.2 Reset

The controller provides 2 forms of reset: power Reset and system Reset.

3.2.1 Power Reset

When a power Reset occurs, it will reset all registers.
A power Reset is generated when the following event occurs:

- Power-up/power-down reset (POR/PDR)

3.2.2 System Reset

When a system Reset occurs, it will reset the reset flag in addition to the control/status register RCC_RSTSCKR and all the registers. The source of the reset event is identified by looking at the reset status flag bit in the RCC_RSTSCKR register.

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

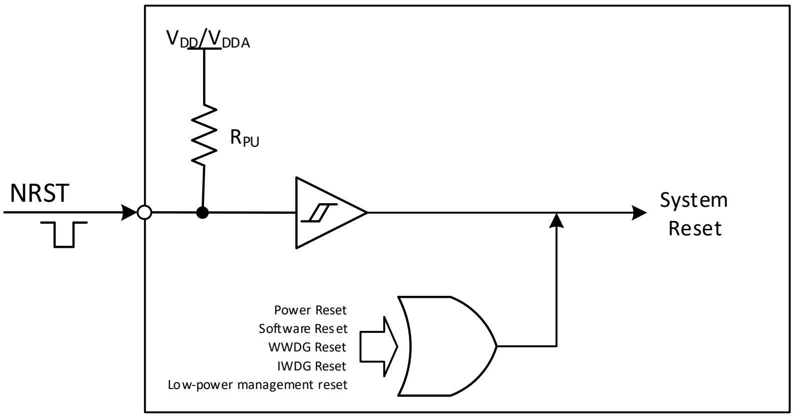
- Low signal on NRST pin (external reset)
- Window watchdog count termination (WWDG reset)
- Independent watchdog count termination (IWDG reset)
- Software reset (SW reset)
- Low-power management reset

Window/Independent Watchdog Reset: Generated by the window/independent watchdog peripheral timer count cycle overflow trigger, see its corresponding section for detailed description.

Software reset: The CH32V003 product resets the system via the RSTSYS position 1 of the interrupt configuration register PFIC_CFGR in the programmable interrupt controller PFIC or the SYSRST position 1 of the configuration register PFIC_SCTLR to reset the system cabinet, refer to the corresponding chapter for details.

Low Power Management Reset: Standby mode reset will be enabled by setting the STANDBY_RST position 1 in the user select byte. This will perform a system reset instead of entering standby mode after the process of entering standby mode is executed.

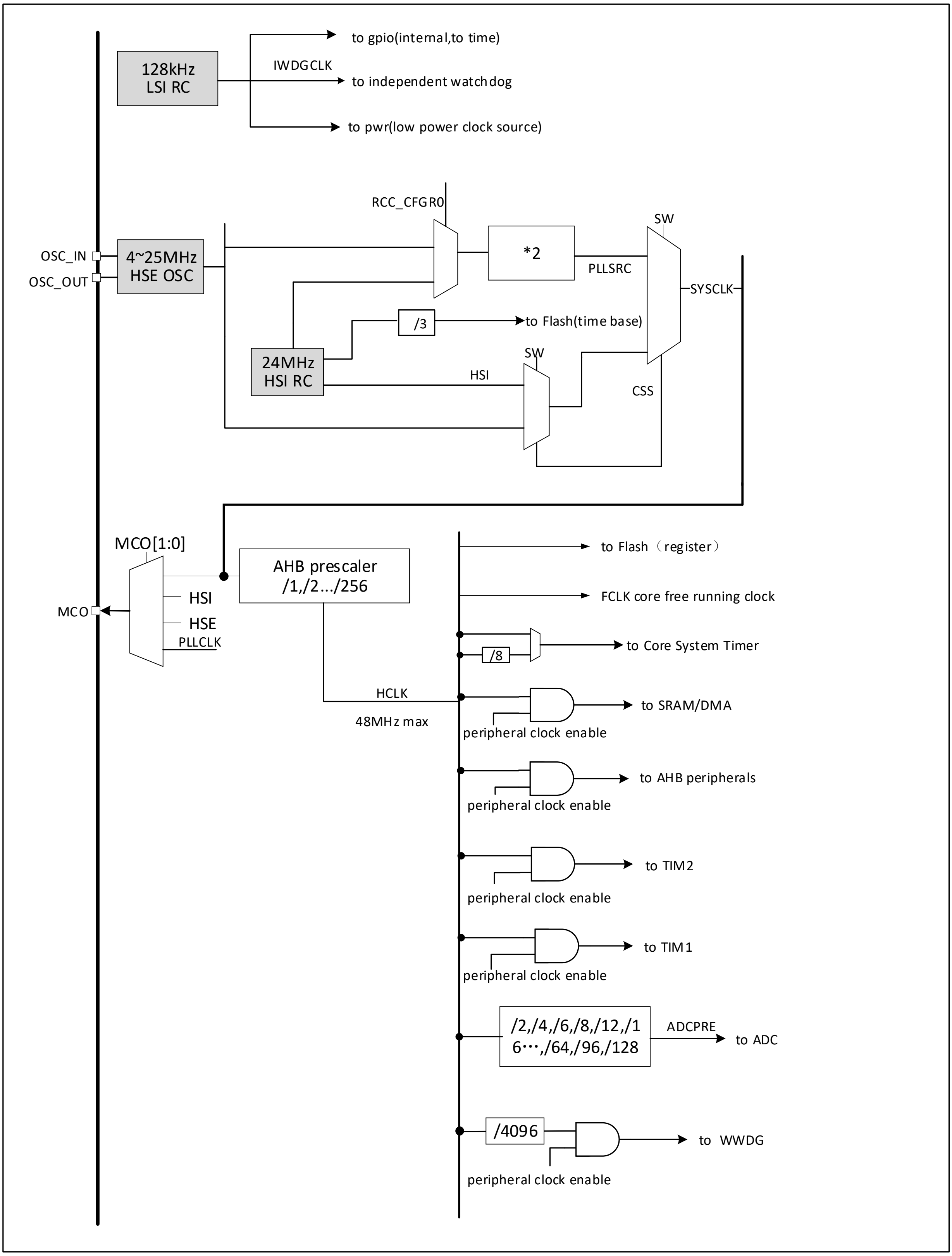
Figure 3-1 System reset structure



3.3 Clock

3.3.1 System Clock Structure

Figure 3-2 CH32V003 clock tree block diagram



3.3.2 High-speed Clock (HSI/HSE)

HSI is a high-speed clock signal generated by the system's internal 24MHz RC oscillator. HSI RC oscillator can provide system clock without any external devices. It has a short start-up time. HSI is enabled and disabled by setting the HSION bit in the RCC_CTLR register, and the HSIRDY bit indicates whether the HSI RC oscillator is stable or not. The system defaults HSION and HSIRDY to 1 (it is recommended not to turn them off). If the HSIRDYIE bit in the RCC_INTR register is set, the corresponding interrupt will be generated.

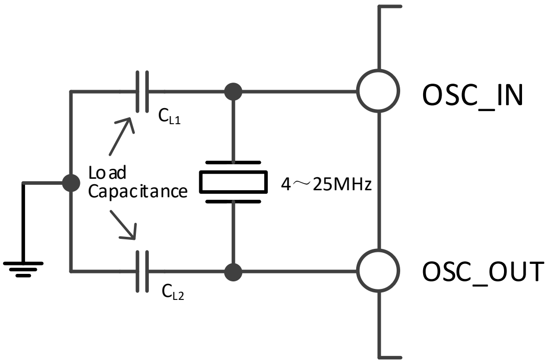
- Factory calibration: The difference of manufacturing process will cause different RC oscillation frequency for each chip, so HSI calibration is performed for each chip before it is shipped. After system reset, the factory calibration value is loaded into HSICAL[7:0] of the RCC_CTLR register.
- User tuning: Based on different voltages or ambient temperatures, the application can adjust the HSI frequency by using the HSITRIM[4:0] bits in the RCC_CTLR register.

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

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

- External Crystal/Ceramic Resonator (HSE Crystal): An external 4-25MHz oscillator provides a more accurate clock source for the system. Further information can be found in the Electrical Characteristics section of the datasheet. The HSE crystal can be turned on and off by setting the HSEON bit in the RCC_CTLR register. The HSERDY bit indicates whether the HSE crystal oscillation is stable or not, and the hardware feeds the clock into the system only after HSERDY position 1. If the HSERDYIE bit of the RCC_INTR register is set, the corresponding interrupt will be generated.

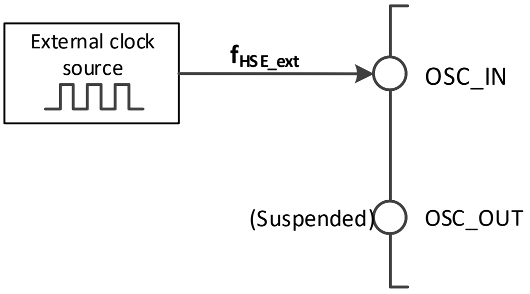
Figure 3-3 High-speed external crystal circuit



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

- External High-speed Clock Source (HSE Bypass): This mode feeds the clock source directly from the external to the OSC_IN pin, with the OSC_OUT pin dangling. The maximum frequency supported is 25MHz. The application needs to set the HSEBYP bit to turn on the HSE bypass function with the HSEON bit at 0, and then set the HSEON bit again.

Figure 3-4 High-speed clock source circuit



3.3.3 Low-speed Clock (LSI)

The LSI is a low-speed clock signal generated by the system's internal RC oscillator of approximately 128 KHz. It can be kept running in shutdown and standby modes and provides the clock reference for the RTC clock, independent watchdog and wake-up unit. Further information can be found in the Electrical Characteristics section of the datasheet. the LSI can be enabled and disabled by setting the LSION bit in the RCC_RSTSCKR register and then detecting whether the LSI RC oscillation is stable by interrogating the LSIRDY bit, and the hardware feeds the clock in only after LSIRDY position 1. If the LSIRDYIE bit in the RCC_INTR register is set, the corresponding interrupt will be generated.

3.3.4 PLL Clock

By configuring the RCC_CFGR0 register and the extended register EXTEND_CTR, the internal PLL clock can select 2 clock sources, these settings must be done before PLL is turned on, once PLL is started these parameters cannot be changed. Set the PLLON bit in the RCC_CTLR register to be enabled and disabled, the PLLRDY bit to indicate whether the PLL clock is stable, and the hardware to feed the clock into the system only after PLL position 1. If the PLLRDYIE bit of the RCC_INTR register is set, the corresponding interrupt will be generated.

PLL clock source:

- HSI clock
- HSE Clock

3.3.5 Bus/Peripheral Clock

3.3.5.1 System Clock (SYSCLK)

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

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

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

3.3.5.2 AHB Bus Peripheral Clock (HCLK)

The AHB, APB1, and APB2 bus clocks can be configured by configuring the HPRE[3:0] bits of the RCC_CFGR0 register. The bus clock determines the peripheral interface access clock reference that is mounted below them. Applications can adjust different values to reduce the power consumption when some of the peripherals are operating.

The various bits in the RCC_APB1PRSTR and RCC_APB2PRSTR registers can reset the different peripheral modules to their initial state.

Each bit in the RCC_AHBPCENR, RCC_APB1PCENR, and RCC_APB2PCENR registers can be used to individually turn on or off the communication clock interface for different peripheral modules. When using a peripheral, you first need to turn on its clock enable bit in order to access its registers.

3.3.5.3 Independent Watchdog Clock

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

3.3.5.4 Microcontroller Clock Output (MCO)

The microcontroller allows outputting clock signals to the MCO pins. The following 4 clock signals can be selected as MCO clock outputs by configuring the multiplexed push-pull output mode in the corresponding GPIO port registers by configuring the MCO[2:0] bits of the RCC_CFGR0 register.

- System clock (SYSCLK) output
- HSI clock output
- HSE clock output
- PLL clock output after 2X frequency

3.3.6 Clock Security System

The clock safety system is an operational protection mechanism for the controller that switches to the HSI clock in the event of an HSE clock transmit failure and generates an interrupt notification to allow the application software to complete a rescue operation.

The clock security system is activated by setting CSSON position 1 of the RCC_CTLR register. At this point, the clock monitor will be enabled after the HSE oscillator start (HSERDY=1) delay and will be turned off after the HSE clock is turned off. Once the HSE clock fails during system operation, the HSE oscillator will be turned off, the clock failure event will be sent to the brake input of the advanced-control timer (TIM1) and a clock safety interrupt will be generated with CSSF position 1 and the application enters the NMI non-maskable interrupt. By setting the CSSC bit, the CSSF bit flag can be cleared and the NMI interrupt pending bit can be undone.

If the current HSE is used as the system clock, or if the current HSE is used as the PLL input clock and the PLL is used as the system clock, the clock safety system will automatically switch the system clock to the HSI oscillator and turn off the HSE oscillator and PLL in case of HSE failure.

3.4 Register Description

Table 3-1 RCC-related registers list

Name	Access address	Description	Reset value
R32_RCC_CTLR	0x40021000	Clock control register	0x0000xx83
R32_RCC_CFGR0	0x40021004	Clock configuration register 0	0x00000020
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	0x00000004
R32_RCC_APB2PCENR	0x40021018	APB2 peripheral clock enable register	0x00000000
R32_RCC_APB1PCENR	0x4002101C	APB1 peripheral clock enable register	0x00000000
R32_RCC_RSTSCKR	0x40021024	Control/status register	0x0C000000

3.4.1 Clock Control Register (RCC_CTLR)

Offset address: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						PLLRDY	PLLON	Reserved				CSSON	HSEBYP	HSERDY	HSEON

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

Bit	Name	Access	Description	Reset value
[31:26]	Reserved	RO	Reserved	0
25	PLLRDY	RO	PLL clock-ready lock flag bit. 1: PLL clock lock. 0: PLL clock is not locked.	0

24	PLLON	RW	PLL clock enable control bit. 1: Enables the PLL clock. 0: Turn off the PLL clock. *Note: After entering Standby low-power mode, this bit is cleared by hardware to 0.*	0
[23:20]	Reserved	RO	Reserved	0
19	CSSON	RW	Clock security system enable control bit. 1: Enable the clock security system. When HSE is ready (HSERDY set to 1), the hardware turns on the clock monitoring function of HSE and triggers CSSF flag and NMI interrupt when HSE is found to be abnormal; when HSE is not ready, the hardware turns off the clock monitoring function of HSE. 0: Turns off the clock security system.	0
18	HSEBYP	RW	External high-speed crystal bypass control bit. 1: Bypass external high-speed crystal/ceramic resonators (using an external clock source). 0: No bypass of high-speed external crystal/ceramic resonators. *Note: This bit needs to be written with HSEON at 0.*	0
17	HSERDY	RO	External high-speed crystal oscillation stabilization ready flag bit (set by hardware). 1: Stable external high-speed crystal oscillation. 0: External high-speed crystal oscillation is not stabilized. *Note: After the HSEON bit is cleared to 0, it takes 6 HSE cycles for this bit to clear to 0.*	0
16	HSEON	RW	External high-speed crystal oscillation enable control bit. 1: Enables the HSE oscillator. 0: Turn off the HSE oscillator. *Note: This bit is cleared to 0 by hardware after entering Standby low-power mode.*	0
[15:8]	HSICAL	RO	Internal high-speed clock calibration values, which are automatically initialized at system startup.	xxh
[7:3]	HSITRIM	RW	Internal high-speed clock adjustment value. The user can enter an adjustment value to superimpose on the HSICAL[7:0] value to adjust the frequency of the internal HSI RC oscillator based on voltage and temperature variations. The default value is 16, which can adjust the HSI to 24MHz ±1%; the change of HSICAL is adjusted about 60KHz per step.	10000
2	Reserved	RO	Reserved	0
1	HSIRDY	RO	Internal high-speed clock (24MHz) Stable Ready flag bit (set by hardware). 1: The internal high-speed clock (24MHz) is stable; 0: The internal high-speed clock (24MHz) is not stable. *Note: After the HSION bit is cleared to 0, it takes 6 HSI cycles for the bit to be cleared to 0.*	1
0	HSION	RW	Internal high-speed clock (24MHz) enable control bit. 1: Enable the HSI oscillator. 0: Disable the HSI oscillator. *Note: This bit is set to 1 by hardware to start the internal 24MHz RC oscillator when returning from standby mode or when the external oscillator HSE used as the system clock fails.*	1

3.4.2 Clock Configuration Register0 (RCC_CFGR0)

Offset address: 0x04

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

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

Bit	Name	Access	Description	Reset value
[31:27]	Reserved	RO	Reserved	0
[26:24]	MCO[2:0]	RW	Microcontroller MCO pin clock output control. 0xx: no clock output. 100: System clock (SYSCLK) output. 101: Internal 24 MHz RC oscillator clock (HSI) output. 110: External oscillator clock (HSE) output. 111: PLL clock output.	0
[23:17]	Reserved	RO	Reserved	0
16	PLLSRC	RW	Input clock source for PLL (write only when PLL is off). 1: HSE is fed into PLL without dividing the frequency. 0: HSI is not divided and sent to PLL.	0
[15:11]	ADCPRE[4:0]	RW	ADC clock source prescaler control {13:11,15:14}. 000xx: AHBCLK divided by 2 as ADC clock. 010xx: AHBCLK divided by 4 as ADC clock. 100xx: AHBCLK divided by 6 as ADC clock. 110xx: AHBCLK divided by 8 as ADC clock. 00100: AHBCLK divided by 4 as ADC clock. 01100: AHBCLK divided by 8 as ADC clock. 10100: AHBCLK divided by 12 as ADC clock. 11100: AHBCLK divided by 16 as ADC clock. 00101: AHBCLK divided by 8 as ADC clock. 01101: AHBCLK divided by 16 as ADC clock. 10101: AHBCLK divided by 24 as ADC clock. 11101: AHBCLK divided by 32 as ADC clock. 00110: AHBCLK divided by 16 as ADC clock. 01110: AHBCLK divided by 32 as ADC clock. 10110: AHBCLK divided by 48 as ADC clock. 11110: AHBCLK divided by 64 as ADC clock. 00111: AHBCLK divided by 32 as ADC clock. 01111: AHBCLK divided by 64 as ADC clock. 10111: AHBCLK divided by 96 as ADC clock. 11111: AHBCLK divided by 128 as ADC clock. *Note: The ADC clock should not exceed a maximum of 24MHz.*	0
[10:8]	Reserved	RW	Reserved	0
[7:4]	HPRE[3:0]	RW	AHB clock source prescaler control. 0000: Prescaler off. 0001: SYSCLK divided by 2. 0010: SYSCLK divided by 3. 0011: SYSCLK divided by 4. 0100: SYSCLK divided by 5. 0101: SYSCLK divided by 6. 0110: SYSCLK divided by 7. 0111: SYSCLK divided by 8. 1000: SYSCLK divided by 2. 1001: SYSCLK divided by 4. 1010: SYSCLK divided by 8.	0010

			1011: SYSCLK divided by 16. 1100: SYSCLK divided by 32. 1101: SYSCLK divided by 64. 1110: SYSCLK divided by 128. 1111: SYSCLK divided by 256. *Note: When the prescaler factor of the AHB clock source is greater than 1, the prefetch buffer must be turned on.*	
[3:2]	SWS[1:0]	RO	System clock (SYSCLK) status (hardware set). 00: the system clock source is HSI. 01: The system clock source is HSE. 10: The system clock source is a PLL. 11: Not available.	0
[1:0]	SW[1:0]	RW	Select the system clock source. 00: HSI as system clock. 01: HSE as system clock. 10: PLL output as system clock. 11: Not available. *Note: With Clock Safe enabled (CSSON=1), HSI is forced by hardware to be selected as the system clock when returning from Standby and Stop mode or when the external oscillator HSE used as the system clock fails.*	0

3.4.3 Clock Interrupt Register (RCC_INTR)

Offset address: 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								CSSC	Reserved		PLLRDYC	HSERDYC	HSIRDYC	Reserved	LSIRDYC

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			PLLRDYIE	HSERDYIE	HSIRDYIE	Reserved	LSIRDYIE	CSSF	Reserved		PLLRDYF	HSERDYF	HSIRDYF	Reserved	LSIRDYF

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 the CSSF interrupt flag. 0: No action.	0
[22:21]	Reserved	RO	Reserved	0
20	PLLRDYC	WO	Clear the PLL-ready interrupt flag bit. 1: Clear the PLLRDYF interrupt flag. 0: No action.	0
19	HSERDYC	WO	Clear the HSE oscillator ready interrupt flag bit. 1: Clear the HSERDYF interrupt flag. 0: No action.	0
18	HSIRDYC	WO	Clear the HSI oscillator ready interrupt flag bit. 1: Clear the HSIRDYF interrupt flag. 0: No action.	0
17	Reserved	RO	Reserved	0
16	LSIRDYC	WO	Clear the LSI oscillator ready interrupt flag bit. 1: Clear the LSIRDYF interrupt flag. 0: No action.	0
[15:13]	Reserved	RO	Reserved	0
12	PLLRDYIE	RW	PLL-ready interrupt enable bit. 1: Enable the PLL-ready interrupt.	0

			0: Disable the PLL-ready interrupt.	
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	Reserved	RO	Reserved	0
8	LSIRDYIE	RW	LSI-ready interrupt enable bit. 1: Enable LSI-ready interrupt. 0: Disable LSI-ready interrupt.	0
7	CSSF	RO	Clock security system interrupt flag bit. 1: HSE clock failure, which generates a clock safety interrupt CSSI. 0: No clock security system interrupt. Hardware set, software write CSSC bit 1 cleared.	0
[6:5]	Reserved	RO	Reserved	0
4	PLLRDYF	RO	PLL clock-ready lockout interrupt flag. 1: PLL clock lock generating interrupt. 0: No PLL clock lock interrupt. Hardware set, software write PLLRDYC bit 1 cleared.	0
3	HSERDYF	RO	HSE clock-ready interrupt flag. 1: HSE clock-ready interrupt generation. 0: No HSE clock-ready interrupt. Hardware set, software write HSERDYC bit 1 cleared.	0
2	HSIRDYF	RO	HSI clock-ready interrupt flag. 1: HSI clock-ready interrupt generation. 0: No HSI clock-ready interrupt. Hardware set, software write HSIRDYC bit 1 cleared.	0
1	Reserved	RO	Reserved	0
0	LSIRDYF	RO	LSI clock-ready interrupt flag. 1: LSI clock-ready interrupt generation. 0: No LSI clock-ready interrupt. Hardware set, software write LSIRDYC bit 1 cleared.	0

3.4.4 APB2 Peripheral Reset Register (RCC_APB2PRSTR)

Offset address: 0x0C

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">Reserved</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Reserved</td><td>USART1 RST</td><td>Reserved</td><td>SPI1 RST</td><td>TIM1 RST</td><td>Reserved</td><td>ADC1 RST</td><td colspan="3">Reserved</td><td>IOPD RST</td><td>IOPC RST</td><td>Reserved</td><td>IOPA RST</td><td>Reserved</td><td>AFIO RST</td></tr>
</table>

Bit	Name	Access	Description	Reset value
[31:15]	Reserved	RO	Reserved	0
14	USART1RST	RW	USART1 interface reset control. 1: Reset module; 0: No effect.	0
13	Reserved	RO	Reserved	0
12	SPI1RST	RW	SPI1 interface reset control. 1: Reset module; 0: No effect.	0
11	TIM1RST	RW	TIM1 module reset control. 1: Reset module; 0: No effect.	0
10	Reserved	RO	Reserved	0
9	ADC1RST	RW	ADC1 module reset control.	0

			1: Reset module; 0: No effect.	
[8:6]	Reserved	RO	Reserved	0
5	IOPDRST	RW	PD port module reset control for I/O. 1: Reset module; 0: No effect.	0
4	IOPCRST	RW	PC port module reset control for I/O. 1: Reset module; 0: No effect.	0
3	Reserved	RO	Reserved	0
2	IOPARST	RW	PA port module reset control for I/O. 1: Reset module; 0: No effect.	0
1	Reserved	RO	Reserved	0
0	AFIORST	RW	I/O auxiliary function module reset control. 1: Reset module; 0: No effect.	0

3.4.5 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			PWRRST	Reserved						I2C1RST	Reserved				

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

Bit	Name	Access	Description	Reset value
[31:29]	Reserved	RO	Reserved	0
28	PWRRST	RW	Power interface module reset control. 1: Reset module; 0: No effect.	0
[27:22]	Reserved	RO	Reserved	0
21	I2C1RST	RW	I2C 1 interface reset control. 1: Reset module; 0: No effect.	0
[20:12]	Reserved	RO	Reserved	0
11	WWDGRST	RW	Window watchdog reset control. 1: Reset module; 0: No effect.	0
[10:1]	Reserved	RO	Reserved	0
0	TIM2RST	RW	Timer 2 module reset control. 1: Reset module; 0: No effect.	0

3.4.6 AHB Peripheral Clock Enable Register (RCC_AHBPCENR)

Offset address: 0x14

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													SRAMEN	Reserved	DMA1EN

Bit	Name	Access	Description	Reset value
[31:3]	Reserved	RO	Reserved	0
2	SRAMEN	RW	SRAM interface module clock enable bit. 1: SRAM interface module clock on during Sleep mode. 0: The SRAM interface module clock is turned off in Sleep mode.	1

1	Reserved	RO	Reserved	0
0	DMA1EN	RW	DMA1 module clock enable bit. 1: Module clock is on; 0: Module clock is off.	0

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

3.4.7 APB2 Peripheral Clock Enable Register (RCC_APB2PCENR)

Offset address: 0x18

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

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

Bit	Name	Access	Description	Reset value
[31:15]	Reserved	RO	Reserved	0
14	USART1EN	RW	USART1 interface clock enable bit. 1: Module clock is on; 0: Module clock is off.	0
13	Reserved	RO	Reserved	0
12	SPI1EN	RW	SPI1 interface clock enable bit. 1: Module clock is on; 0: Module clock is off.	0
11	TIM1EN	RW	TIM1 module clock enable bit. 1: Module clock is on; 0: Module clock is off.	0
10	Reserved	RO	Reserved	0
9	ADC1EN	RW	ADC1 module clock enable bit. 1: Module clock is on; 0: Module clock is off.	0
[8:6]	Reserved	RO	Reserved	0
5	IOPDEN	RW	PD port module clock enable bit for I/O. 1: Module clock is on; 0: Module clock is off.	0
4	IOPCEN	RW	PC port module clock enable bit for I/O. 1: Module clock is on; 0: Module clock is off.	0
3	Reserved	RO	Reserved	0
2	IOPAEN	RW	PA port module clock enable bit for I/O. 1: Module clock is on; 0: Module clock is off.	0
1	Reserved	RO	Reserved	0
0	AFIOEN	RW	I/O auxiliary function module clock enable bit. 1: Module clock is on; 0: Module clock is off.	0

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

3.4.8 APB1 Peripheral Clock Enable Register (RCC_APB1PCENR)

Offset address: 0x1C

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

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

Bit	Name	Access	Description	Reset value
[31:29]	Reserved	RO	Reserved	0
28	PWREN	RW	Power interface module clock enable bit. 1: Module clock is on; 0: Module clock is off.	0
[27:22]	Reserved	RO	Reserved	0
21	I2C1EN	RW	I2C 1 interface clock enable bit. 1: Module clock is on; 0: Module clock is off.	0
[20:12]	Reserved	RO	Reserved	0
11	WWDGEN	RW	Window watchdog clock enable bit. 1: Module clock is on; 0: Module clock is off.	0
[10:1]	Reserved	RO	Reserved	0
0	TIM2EN	RW	Timer 2 module clock enable bit. 1: Module clock is on; 0: Module clock is off.	0

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

3.4.9 Control/Status Register (RCC_RSTSCKR)

Offset address: 0x24

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
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: Occurrence of low-power resets. 0: No low-power reset occurs. Set to 1 by hardware when a low-power management reset occurs; cleared by software writing of the RMVF bit.	0
30	WWDGRSTF	RO	Window watchdog reset flag. 1: Occurrence of a window watchdog reset. 0: No window watchdog reset occurs. Set to 1 by hardware when a window watchdog reset occurs; cleared by software writing of the RMVF bit.	0
29	IWDGRSTF	RO	Independent watchdog reset flag. 1: Occurrence of an independent watchdog reset. 0: No independent watchdog reset occurs. Set to 1 by hardware when an independent watchdog reset occurs; cleared by software writing of the RMVF bit.	0
28	SFTRSTF	RO	Software reset flag. 1: Software reset occurs. 0: No software reset occurs. Set to 1 by hardware when a software reset occurs; software write RMVF bit cleared.	0
27	PORRSTF	RO	Power-up/power-down reset flag. 1: Power-up/power-down reset occurs. 0: No power-up/power-down reset occurs. Set to 1 by hardware when power-up/power-down reset occurs; cleared by software writing of RMVF bit.	1

26	PINRSTF	RO	External manual reset (NRST pin) flag. 1: Occurrence of NRST pin reset. 0: No NRST pin reset occurs. Set to 1 by hardware when NRST pin reset occurs; cleared by software writing of RMVF bit.	0
25	Reserved	RO	Reserved	0
24	RMVF	RW	Clear reset flag control. 1: Clear the reset flag. 0: No effect.	0
[23:2]	Reserved	RO	Reserved。	0
1	LSIRDY	RO	Internal Low Speed Clock (LSI) Stable Ready flag bit (set by hardware). 1: Stable internal low-speed clock (128KHz). 0: The internal low-speed clock (128KHz) is not stable. *Note: After the LSION bit is cleared to 0, the bit requires 3 LSI cycles to clear 0.*	0
0	LSION	RW	Internal low-speed clock (LSI) enable control bit. 1: Enable the LSI (128KHz) oscillator. 0: Disable the LSI (128KHz) oscillator.	0

Note: Except for the reset flag which can only be cleared by power-on reset, others are cleared by system Reset.

Chapter 4 Independent Watchdog (IWDG)

The system is equipped with an independent watchdog (IWDG) to detect logic errors and software faults caused by external environmental disturbances. the IWDG clock source is derived from the LSI and can run independently of the main program, making it suitable for applications requiring low accuracy.

4.1 Main Features

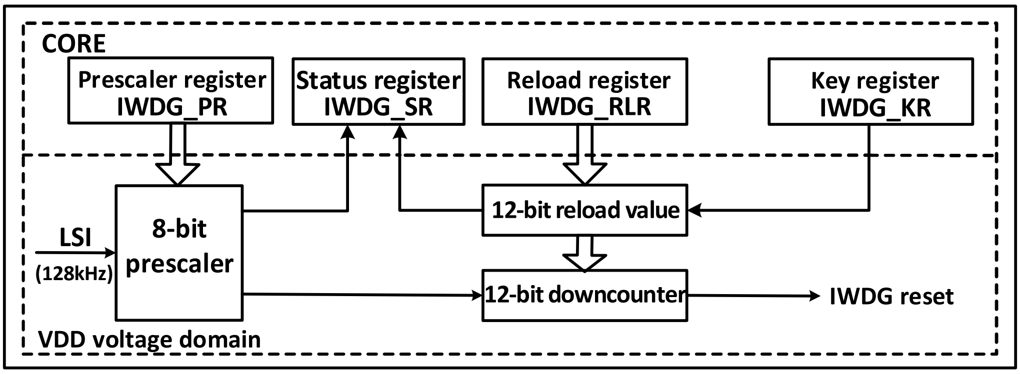
- 12-bit self-subtracting counter
- Clock source LSI divider, can run in low-power mode
- Reset condition: Counter value is reduced to 0

4.2 Function Description

4.2.1 Principle and Application

The independent watchdog is clocked from the LSI clock and its function remains functional during shutdown and Standby modes. When the watchdog counter self-decreases to 0, a system Reset will be generated, so the timeout is (reload value + 1) clock.

Figure 4-1 Block diagram of the structure of the independent watchdog



- Enable independent watchdog

After a system reset, the watchdog is off and writing 0xCCCC to the IWDG_CTLR register turns the watchdog on, after which it cannot be turned off again unless a reset occurs.

If the hardware independent watchdog enable bit (IWDG_SW) is turned on at the user-option bytes, IWDG will be fixed on after a microcontroller reset.

- Watchdog configuration

The watchdog is internally a 12-bit counter that runs decreasingly. When the counter value decreases to 0, a system Reset will occur. To turn on the IWDG function, the following actions need to be performed.

1) Counting time base: IWDG clock source LSI, set the LSI crossover value clock as the counting time base of IWDG through the IWDG_PSCR register. The operation method first writes 0x5555 to the IWDG_CTLR register, and then modifies the crossover value in the IWDG_PSCR register. the PVU bit in the IWDG_STATR register indicates the update status of the crossover value, and the crossover value can be modified and read out only when the update is completed.
2) Reload value: Used to update the current value of the counter in the standalone watchdog and the counter is decremented by this value. The RVU bit in the IWDG_STATR register indicates the update status of the reload value, and the IWDG_RLDR register can be modified and read out only when the update is completed.
3) Watchdog enable: write 0xCCCC to the IWDG_CTLR register to enable the watchdog function.

4) Feed the dog: i.e., flush the current counter value before the watchdog counter decrements to 0 to prevent a system reset from occurring. Write 0xAAAA to the IWDG_CTLR register to allow the hardware to update the IWDG_RLDR register value to the watchdog counter. This action needs to be executed regularly after the watchdog function is turned on, otherwise a watchdog reset action will occur.

4.2.2 Debug Mode

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

4.3 Register Description

Table 4-1 IWDG-related registers list

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

4.3.1 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	Operate the key value lock. 00xAAAA: Feed the dog. Loading of the IWDG_RLDR register value into the independent watchdog counter. 0x5555: Allows modification of the R16_IWDG_PSCR and R16_IWDG_ RLDR registers. 0xCCCC: Start the watchdog, but not if the hardware watchdog is enabled (user-option bytes configuration).	0

4.3.2 Prescaler Register (IWDG_PSCR)

Offset address: 0x04

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

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

4.3.3 Reload Register (IWDG_RLDR)

Offset address: 0x08

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

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

Note: This register will be reset in Standby mode.

4.3.4 Status Register (IWDG_STATR)

Offset address: 0x0C

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

Bit	Name	Access	Description	Reset value
[15:2]	Reserved	RO	Reserved	0
1	RVU	RO	Reload value update flag bit. Hardware set or clear 0. 1: Reload value update is in progress. 0: End of reload update (up to 5 LSI cycles). *Note: The reload value register IWDG_RLDR can only be accessed read or write after the RVU bit is cleared to 0.*	0
0	PVU	RO	Clock division factor update flag bit. Hardware set or clear 0. 1: Clock division value update is in progress. 0: End of clock division value update (up to 5 LSI cycles). *Note: The crossover factor register IWDG_PSCR can only be accessed read or write after the PVU bit is cleared to 0.*	0

Note: After the prescaler or reload value is updated, it is not necessary to wait for the RVU or PVU to reset, and the following code can continue to be executed. (This write operation will continue to be executed to completion even in low-power mode.)

Chapter 5 Window Watchdog (WWDG)

A Window Watchdog is generally used to monitor system operation for software faults such as external disturbances, unforeseen logic errors, and other conditions. It requires a counter refresh (dog feeding) within a specific window time (with upper and lower limits), otherwise earlier or later than this window time the watchdog circuit will generate a system Reset.

5.1 Main Features

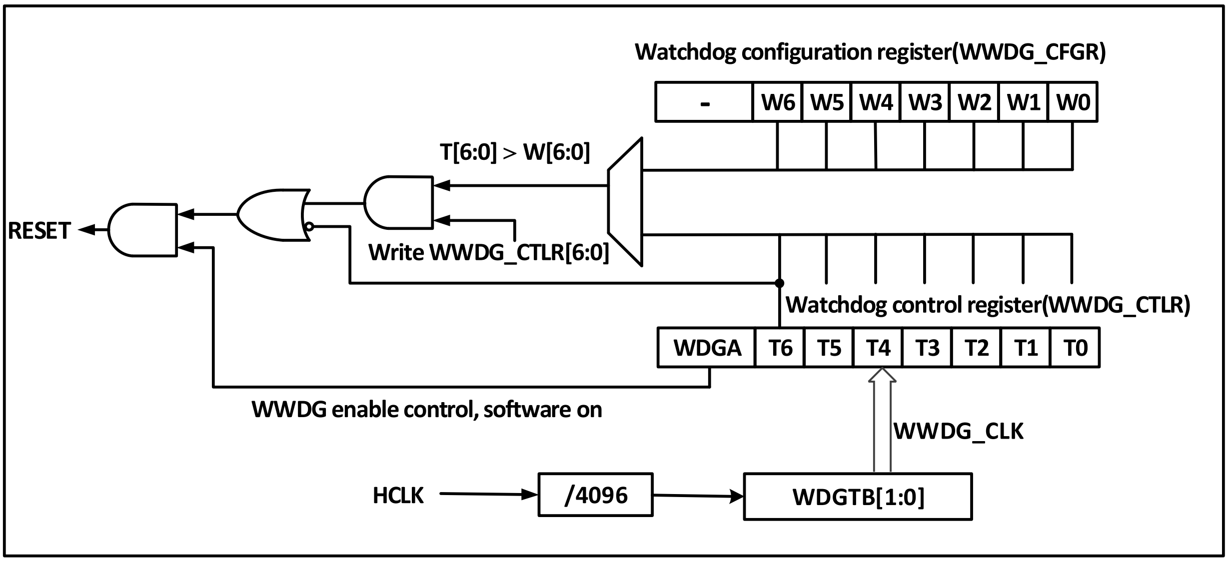
- Programmable 7-bit downcounter
- Biconditional reset: the downcounter value is less than 0x40, or the counter value is reloaded outside the window time
- Wake Up Early Notification (EWI) function for timely dog feeding action to prevent system Reset

5.2 Function Description

5.2.1 Principle and Application

The window watchdog operation is based on a 7-bit downcounter, which is mounted under the AHB bus and counts the dividing frequency of the time base WWDG_CLK source (HCLK/4096) clock with the dividing factor set in the WDGTB[1:0] field in the configuration register WWDG_CFGR. The downcounter is in the free-running state, and the counter keeps cycling downcount regardless of whether the watchdog function is on or not. As shown in Figure 5-1, the block diagram of the internal structure of the window watchdog.

Figure 5-1 Block diagram of Window Watchdog structure



- Enable Window Watchdog

After a system Reset, the watchdog is off. Setting the WDGA bit of the WWDG_CTLR register enables the watchdog, and then it cannot be turned off again unless a reset occurs.

Note: The watchdog function can be stopped indirectly by setting the RCC_APB1PCENR register to turn off the clock source of WWDG and suspend the WWDG_CLK count, or by setting the RCC_APB1PRSTR register to reset the WWDG module, which is equivalent to the role of reset.

- Watchdog Configuration

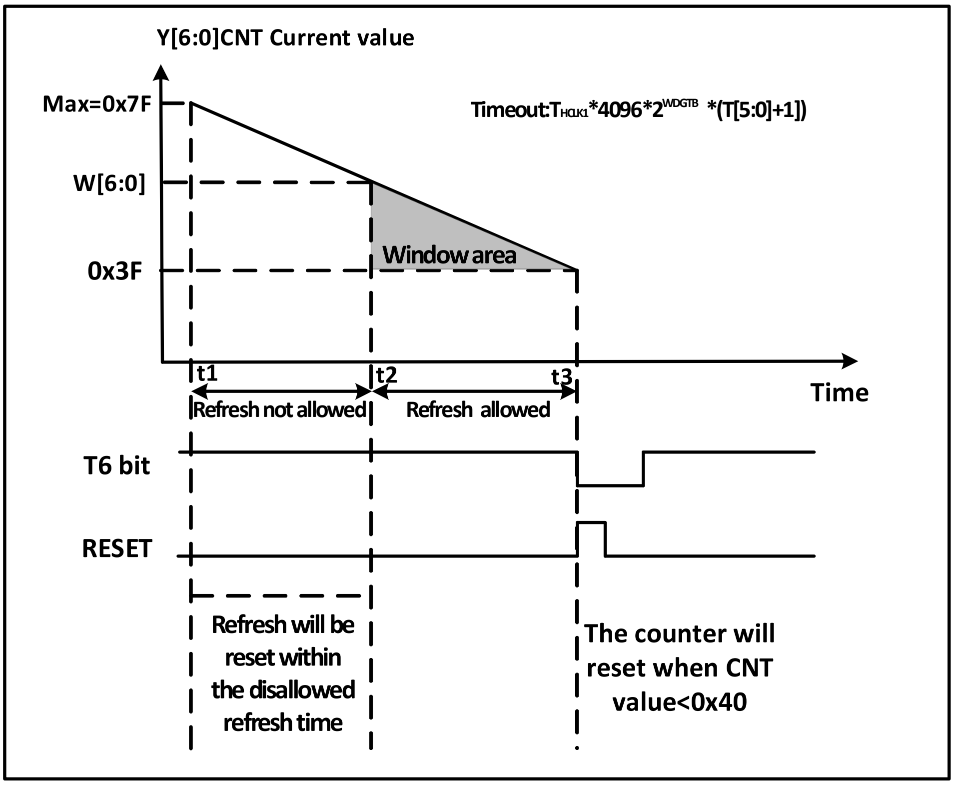
The watchdog is internally a 7-bit counter that runs in a continuous decreasing cycle and supports read and write access. To use the watchdog reset function, the following actions need to be performed.

1) Counting time base: via the WDGTB[1:0] bit field of the WWDG_CFGR register, note that the WWDG module clock of the RCC unit should be turned on.
2) Window counter: Set the W[6:0] bit field of WWDG_CFGR register, this counter is used by hardware as a comparison with the current counter, the value is configured by user software and will not change. It is used as the upper limit value of the window time.
3) Watchdog enable: WDG_CTLR register WDGA bit software set to 1, to turn on the watchdog function, you can system reset.
4) Feed the dog: i.e., refresh the current counter value and configure the T[6:0] bit field of the WWDG_CTLR register. This action needs to be executed within the periodic window time after the watchdog function is turned on, otherwise a watchdog reset action will occur.

- Dog feeding window time

As shown in Figure 5-2, the gray area is the monitoring window area of the window watchdog, whose upper time t2 corresponds to the point in time when the current counter value reaches the window value W[6:0]; its lower time t3 corresponds to the point in time when the current counter value reaches 0x3F. This area time t2<t<t3 can be fed with a dog operation (write T[6:0]) to refresh the current counter value.

- Figure 5-2 Counting mode of Window Watchdog



- Watchdog reset

1) When the value of T[6:0] counter changes from 0x40 to 0x3F due to no timely dog feeding operation, a "window watchdog reset" will occur and a system reset will be generated. That is, the T6-bit is detected as 0 by the hardware and a system reset will occur.
   Note: The application can write T6-bit to 0 by software to achieve system Reset, which is equivalent to software reset function.
2) When the counter refresh action is executed within the disallowed dog feeding time, i.e., the write T[6:0] bit field is operated within $t_1 \leqslant t \leqslant t_2$ time, a "window watchdog reset" will occur and a system Reset will be generated.

- Wake up in advance

To prevent the system Reset caused by not refreshing the counter in time, the watchdog module provides an early wakeup interrupt (EWI) notification. When the counter self-decreases to 0x40, an early wake-up signal is generated and the EWIF flag is set to 1. If the EWI bit is set, a window watchdog interrupt will be triggered at the same time. At this time, there is 1 counter clock cycle (self-decrement to 0x3F) before the hardware reset, and the application can perform the dog feeding operation instantly within this time.

5.2.2 Debug Mode

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

5.3 Register Description

Table 5-1 WWDG-related registers list

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

5.3.1 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: Turn on the watchdog function (which generates a reset signal). 0: Disable the watchdog function. Software write 1 is on, but only allows hardware to clear 0 after reset.	0
[6:0]	T[6:0]	RW	The 7-bit self-decrement counter decrements by 1 every $4096*2^{WDGTB}$ HCLK cycles. A watchdog reset is generated when the counter decrements from 0x40 to 0x3F, i.e., when T6 jumps to 0.	7Fh

5.3.2 Configuration Register (WWDG_CFGR)

Offset address: 0x04

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

Bit	Name	Access	Description	Reset value
[15:10]	Reserved	RO	Reserved	0
9	EWI	RW1	Early wakeup interrupt enable bit. If this position is 1, an interrupt is generated when the counter value reaches 0x40. This bit can only be invited to 0 by hardware after a reset.	0
[8:7]	WDGTB[1:0]	RW	Window watchdog clock division selection. 00: Divided by 1, counting time base = HCLK/4096. 01: Divided by 2, counting time base = HCLK /4096/2.	0

			10: Divided by 4, counting time base = HCLK /4096/4. 11: Divided by 8, counting time base = HCLK /4096/8.	
[6:0]	W[6:0]	RW	Window watchdog 7-bit window value. Used to compare with the counter value. The feed dog operation can only be performed when the counter value is less than the window value and greater than 0x3F.	7Fh

5.3.3 Status Register (WWDG_STATR)

Offset address: 0x08

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

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

Chapter 6 Interrupt and Events (PFIC)

The CH32V003 series has a built-in Programmable Fast Interrupt Controller (PFIC) that supports up to 255 interrupt vectors. The current system manages 23 peripheral interrupt channels and 4 core interrupt channels, the others are reserved.

6.1 Main features

6.1.1 PFIC

- 23 peripheral interrupts, each interrupt request has independent trigger and mask control bits, with dedicated status bits
- Programmable multi-level interrupt nesting, maximum nesting depth 2 levels, hardware stack depth 2 levels
- Fast interrupt entry and exit mechanism, hardware automatic stacking and recovery
- Vector Table Free (VTF) interrupt response mechanism, 2-way programmable direct access to interrupt vector addresses

6.2 System Timer

- CH32V003 Series

The core comes with a 32-bit add counter (SysTick) that supports HCLK or HCLK/8 as a time base with high priority and can be used as a time reference after calibration.

6.3 Vector Table of Interrupts and Exceptions

Table 6-1 CH32V003 series vector table

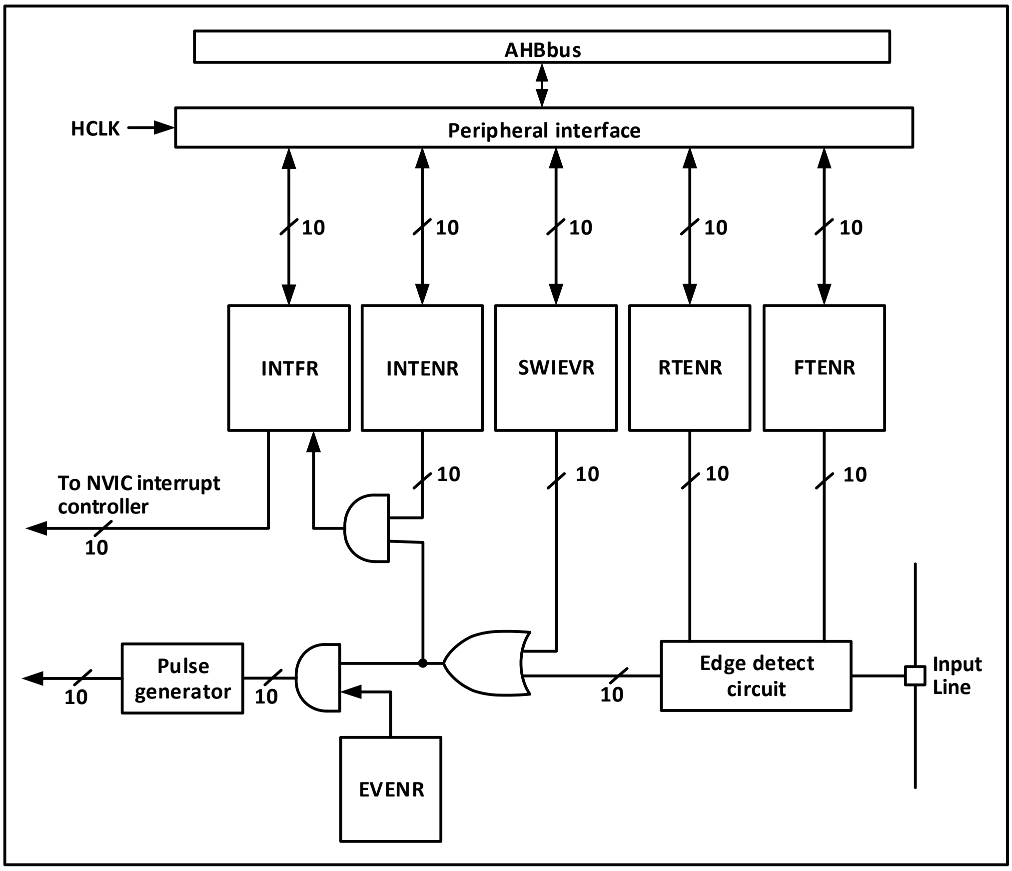
No.	Priority	Type	Name	Description	Entrance address
0	-	-	-	-	0x00000000
1	-	-	-	-	0x00000004
2	-2	fixed	NMI	Non-maskable interrupts	0x00000008
3	-1	fixed	HardFault	Abnormal interruptions	0x0000000C
4-11	-	-	-	Reserved	0x00000010- 0x0000002C
12	0	programmable	SysTick	System timer interrupt	0x00000030
13	-	-	-	Reserved	0x00000034
14	1	programmable	SW	Software interrupt	0x00000038
15	-	-	-	Reserved	0x0000003C
16	2	programmable	WWDG	Window timer interrupt	0x00000040
17	3	programmable	PVD	Supply voltage detection interrupt (EXTI)	0x00000044
18	4	programmable	FLASH	Flash global interrupt	0x00000048
19	5	programmable	RCC	Reset and clock control interrupts	0x0000004C
20	6	programmable	EXTI7_0	EXTI line 0-7 interrupt	0x00000050
21	7	programmable	AWU	Wake-up interrupt	0x00000054
22	8	programmable	DMA1_CH1	DMA1 channel 1 global interrupt	0x00000058
23	9	programmable	DMA1_CH2	DMA1 channel 2 global interrupt	0x0000005C
24	10	programmable	DMA1_CH3	DMA1 channel 3 global interrupt	0x00000060
25	11	programmable	DMA1_CH4	DMA1 channel 4 global interrupt	0x00000064
26	12	programmable	DMA1_CH5	DMA1 channel 5 global interrupt	0x00000068
27	13	programmable	DMA1_CH6	DMA1 channel 6 global interrupt	0x0000006C
28	14	programmable	DMA1_CH7	DMA1 channel 7 global interrupt	0x00000070

29	15	programmable	ADC	ADC global Interrupt	0x00000074
30	16	programmable	I2C1_EV	I2C1 event interrupt	0x00000078
31	17	programmable	I2C1_ER	I2C1 error interrupt	0x0000007C
32	18	programmable	USART1	USART1 global interrupt	0x00000080
33	19	programmable	SPI1	SPI1 global Interrupt	0x00000084
34	20	programmable	TIM1BRK	TIM1 brake interrupt	0x00000088
35	21	programmable	TIM1UP	TIM1 update interrupt	0x0000008C
36	22	programmable	TIM1TRG	TIM1 triggers an interrupt	0x00000090
37	23	programmable	TIM1CC	TIM1 captures the compare interrupt	0x00000094
38	24	programmable	TIM2	TIM2 global interrupt	0x00000098

6.4 External Interrupt and Event Controller (EXTI)

6.4.1 Overview

Figure 6-1 External interrupt (EXTI) interface block diagram



As can be seen from Figure 6-1, the trigger source of the external interrupt can be either a software interrupt (SWIEVR) or an actual external interrupt channel. The signal of the external interrupt channel will be screened by the edge detect circuit first. Whenever one of the software interrupt or external interrupt signals is generated, it will be output to two with-gate circuits, event enable and interrupt enable, through the or-gate circuit in the figure, as long as an interrupt is enabled or an event is enabled, an interrupt or an event will be generated. six registers of EXTI are accessed by the processor through the AHB interface.

6.4.2 Wake-up Event

The system can wake up the Sleep mode caused by the WFE command through a wake-up event. The wake-up event is generated by either of the following two configurations.

- Enabling an interrupt in a peripheral register, but not enabling this interrupt in the PFIC of the core, and enabling the SEVONPEND bit in the core at the same time. Embodied in EXTI, it is to enable an EXTI interrupt, but not to enable the EXTI interrupt in PFIC, and to enable the SEVONPEND bit at the same time. When the CPU wakes up from WFE, it needs to clear the EXTI interrupt flag bit and the PFIC pending bit.

- Enabling an EXTI channel as an event channel eliminates the need for the CPU to clear the interrupt flag bit and the PFIC pending bit after waking up from the WFE.

6.4.3 Description

Using an external interrupt requires configuring the corresponding external interrupt channel, i.e. selecting the corresponding trigger edge and enabling the corresponding 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 be set. The flag bit can be cleared by writing 1 to the flag bit.

Steps for using external hardware interrupts.

1) Configuration of GPIO operations.
2) Configure the interrupt enable bit (EXTI_INTENR) for the corresponding external interrupt channel.
3) Configuring the trigger edge (EXTI_RTENR or EXTI_FTENR) to select rising edge trigger, falling edge trigger or double edge trigger.
4) Configure EXTI interrupts in the core's PFIC to ensure they can respond correctly.

Steps for using external hardware events.

1) Configuration of GPIO operations.
2) Configure the event enable bit (EXTI_EVENR) for the corresponding external interrupt channel.
3) Configure the trigger edge (EXTI_RTENR or EXTI_FTENR) to select rising edge trigger, falling edge trigger, or double edge trigger.

Using the software interrupt/event steps.

1) Enabling external interrupts (EXTI_INTENR) or external events (EXTI_EVENR).
2) If using interrupt service functions, the EXTI interrupt needs to be set in the core's PFIC.
3) Set the software interrupt trigger (EXTI_SWIEVR), that is, an interrupt will be generated.

6.4.4 External Event Mapping

Table 6-2 EXTI Interrupt Mapping

External interrupt/ Event lines	Mapping Event Description
EXTI0~EXTI7	Px0 to Px7 (x=A/C/D), any IO port can enable external interrupt/event function, configured by AFIO_EXTICRx register.
EXTI8	PVD event: voltage monitoring threshold value exceeded
EXTI9	Auto-wakeup events

6.5 Register Description

6.5.1 EXTI Registers

Table 6-3 EXTI-related registers list

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 register	0x0000XXXX

6.5.1.1 Interrupt Enable Register (EXTI_INTENR)

Offset address: 0x00

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						MR9	MR8	MR7	MR6	MR5	MR4	MR3	MR2	MR1	MR0

Bit	Name	Access	Description	Reset value
[31:10]	Reserved	RO	Reserved	0
[9:0]	MRx	RW	Enable the interrupt request signal for external interrupt channel x. 1: Enables interrupts for this channel. 0: Mask interrupts for this channel.	0

6.5.1.2 Event Enable Register (EXTI_EVENR)

Offset address: 0x04

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						MR9	MR8	MR7	MR6	MR5	MR4	MR3	MR2	MR1	MR0

Bit	Name	Access	Description	Reset value
[31:10]	Reserved	RO	Reserved	0
[9:0]	MRx	RW	Enable the event request signal for external interrupt channel x. 1: Event enabling this channel. 0: Block the events of this channel.	0

6.5.1.3 Rising Edge Trigger Enable Register (EXTI_RTENR)

Offset address: 0x08

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						TR9	TR8	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0

Bit	Name	Access	Description	Reset value
[31:10]	Reserved	RO	Reserved	0
[9:0]	TRx	RW	Enable rising edge triggering of external interrupt channel x. 1: Enable rising edge triggering of this channel. 0: Disable rising edge triggering for this channel.	0

6.5.1.4 Falling Edge Trigger Enable Register (EXTI_FTENR)

Offset address: 0x0C

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

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

Reserved	TR9	TR8	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0

Bit	Name	Access	Description	Reset value
[31:10]	Reserved	RO	Reserved	0
[9:0]	TRx	RW	Enable falling edge triggering of external interrupt channel x. 0: Disable falling edge triggering for this channel. 1: Enable falling edge triggering for this channel.	0

6.5.1.5 Software Interrupt Event Register (EXTI_SWIEVR)

Offset address: 0x10

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						SWIER9	SWIER8	SWIER7	SWIER6	SWIER5	SWIER4	SWIER3	SWIER2	SWIER1	SWIER0

Bit	Name	Access	Description	Reset value
[31:10]	Reserved	RO	Reserved	0
[9:0]	SWIERx	RW	A software interrupt is set on the corresponding externally triggered interrupt channel. Setting it here causes the interrupt flag bit (EXTI_INTFR) to correspond to the position bit, and if interrupt enable (EXTI_INTENR) or event enable (EXTI_EVENR) is on, then an interrupt or event will be generated.	0

6.5.1.6 Interrupt Flag Register (EXTI_INTFR)

Offset address: 0x14

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						IF9	IF8	IF7	IF6	IF5	IF4	IF3	IF2	IF1	IF0

Bit	Name	Access	Description	Reset value
[31:10]	Reserved	RO	Reserved	0
[9:0]	IFx	W1	The interrupt flag bit, this location bit flag indicates that the corresponding external interrupt has occurred. A write of 1 clears this bit.	X

6.5.2 PFIC Registers

Table 6-4 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 pending status register 1	0x00000000
R32_PFIC_IPR2	0xE000E024	PFIC interrupt pending status register 2	0x00000000
R32_PFIC_ITHRESDR	0xE000E040	PFIC interrupt priority threshold configuration register	0x00000000
R32_PFIC_CFGR	0xE000E048	PFIC interrupt configuration register	0x00000000

R32_PFIC_GISR	0xE000E04C	PFIC interrupt global status register	0x00000000
R32_PFIC_VTFIDR	0xE000E050	PFIC VTF interrupt ID configuration register	0x00000000
R32_PFIC_VTFADDRR0	0xE000E060	PFIC VTF interrupt 0 offset address register	0x00000000
R32_PFIC_VTFADDRR1	0xE000E064	PFIC VTF interrupt 1 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 clear register 1	0x00000000
R32_PFIC_IRER2	0xE000E184	PFIC interrupt enable clear register 2	0x00000000
R32_PFIC_IPSR1	0xE000E200	PFIC interrupt pending setting register 1	0x00000000
R32_PFIC_IPSR2	0xE000E204	PFIC interrupt pending setting register 2	0x00000000
R32_PFIC_IPRR1	0xE000E280	PFIC interrupt pending clear register 1	0x00000000
R32_PFIC_IPRR2	0xE000E284	PFIC interrupt pending clear register 2	0x00000000
R32_PFIC_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

Note:

1. The default value of PFIC_ISR1 register is 0xC, that is, NMI and exception are always enabled by default.

2. NMI and EXC support interrupt pending clear and setting operation, but not interrupt enable clear and setting operation.

6.5.2.1 PFIC Interrupt Enable Status Register 1 (PFIC_ISR1)

Offset address: 0x00

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	INTENSTA14	Reserved	INTENSTA12	Reserved								INTENSTA3	INTENSTA2	Reserved	

Bit	Name	Access	Description	Reset value
[31:16]	INTENSTA	RO	16#-31# Interrupt current enable status. 1: The current numbered interrupt is enabled. 0: The current numbered interrupt is not enabled.	0
15	Reserved	RO	Reserved	0
14	INTENSTA	RO	14# Interrupt current enable status. 1: The current numbered interrupt is enabled. 0: The current numbered interrupt is not enabled.	0
13	Reserved	RO	Reserved	0
12	INTENSTA	RO	12# Interrupt current enable status. 1: The current numbered interrupt is enabled. 0: The current numbered interrupt is not enabled.	0
[11:4]	Reserved	RO	Reserved	0

[3:2]	INTENSTA	RO	2#-3# interrupt current enable state. 1: The current numbered interrupt is enabled. 0: The current numbered interrupt is not enabled.	0
[1:0]	Reserved	RO	Reserved	0

6.5.2.2 PFIC Interrupt Enable Status Register 2 (PFIC_ISR2)

Offset address: 0x04

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

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

Bit	Name	Access	Description	Reset value
[31:7]	Reserved	RO	Reserved	0
[6:0]	INTENSTA	RO	32#-38# Interrupt current enable state. 1: The current numbered interrupt is enabled. 0: The current numbered interrupt is not enabled.	0

6.5.2.3 PFIC Interrupt Pending Status Register 1 (PFIC_IPR1)

Offset address: 0x20

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	PENDSTA14	Reserved	PENDSTA12	Reserved								PENDSTA3	PENDSTA2	Reserved	

Bit	Name	Access	Description	Reset value
[31:16]	PENDSTA	RO	1216#-31# interrupt the current pending status. 1: The current number break is pending. 0: The current number break is not pending.	0
15	Reserved	RO	Reserved	0
14	PENDSTA		14# Interrupt the current pending state. 1: The current number break is pending. 0: The current number break is not pending.	0
13	Reserved	RO	Reserved	0
12	PENDSTA		12# Interrupt the current pending state. 1: The current number break is pending. 0: The current number break is not pending.	0
[11:4]	Reserved	RO	Reserved	0
[3:2]	PENDSTA	RO	2#-3# interrupt the current pending state. 1: The current number break is pending. 0: The current number break is not pending.	0
[1:0]	Reserved	RO	Reserved	0

6.5.2.4 PFIC Interrupt Pending Status Register 2 (PFIC_IPR2)

Offset address: 0x24

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									PENDSTA[38:32]						

Bit	Name	Access	Description	Reset value
[31:7]	Reserved	RO	Reserved	0
[6:0]	PENDSTA	RO	32#-38# Interrupt current pending status. 1: The current number break is pending. 0: The current number break is not pending.	0

6.5.2.5 PFIC Interrupt Priority Threshold Configuration Register (PFIC_ITHRESDR)

Offset address: 0x40

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

Bit	Name	Access	Description	Reset value
[31:8]	Reserved	RO	Reserved	0
[7:0]	THRESHOLD [7:0]	RW	Interrupt priority threshold setting value. The interrupt priority value lower than the current setting value, when hung, does not perform interrupt service; this register is 0 means the threshold register function is invalid. [7:6]: priority threshold. [5:0]: reserved, fixed to 0, write invalid.	0

6.5.2.6 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								RESETSYS	Reserved						

Bit	Name	Access	Description	Reset value
[31:16]	KEYCODE[15:0]	WO	Corresponding to different target control bits, the corresponding security access identification data needs to be written simultaneously in order to be modified, and the readout data is fixed to 0. KEY1 = 0xFA05. KEY2 = 0xBCAF. KEY3 = 0xBEEF.	0
[15:8]	Reserved	RO	Reserved	0
7	RESETSYS	WO	System reset (simultaneous writing to KEY3). Auto clear 0. Writing 1 is valid, writing 0 is invalid. *Note: Same function as the PFIC_SCTLR register SYSRESET bit.*	0
[6:0]	Reserved	RO	Reserved	0

6.5.2.7 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	Are there any interrupts currently on hold. 1: Yes; 0: No.	0
8	GACTSTA	RO	Are there any interrupts currently being executed. 1: Yes; 0: No.	0
[7:0]	NESTSTA[7:0]	RO	Current interrupt nesting status, currently supports a maximum of 2 levels of nesting and a maximum hardware stack depth of 2 levels. 0x03: Level 2 interrupt in progress. 0x01: Level 1 interrupt in progress. Other: no interrupt occurred.	0

6.5.2.8 PFIC VTF Interrupt ID Configuration Register (PFIC_VTFIDR)

Offset address: 0x50

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VTFID1								VTFID0							

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:8]	VTFID1	RW	Configure the interrupt number of VTF interrupt 1.	0
[7:0]	VTFID0	RW	Configure the interrupt number of VTF interrupt 0.	0

6.5.2.9 PFIC VTF Interrupt 0 Address Register (PFIC_VTFADDRR0)

Offset address: 0x60

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

Bit	Name	Access	Description	Reset value
[31:1]	ADDR0[31:1]	RW	VTF interrupt 0 service program address bit[31:1], bit0 is 0.	0
0	VTF0EN	RW	VTF interrupt 0 enable bit. 1: enable VTF interrupt 0 channel; 0: off.	0

6.5.2.10 PFIC VTF Interrupt 1 Address Register (PFIC_VTFADDRR1)

Offset address: 0x64

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR1[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR1[15:1]															VTF1EN

Bit	Name	Access	Description	Reset value
[31:1]	ADDR1[31:1]	RW	VTF interrupt 1 service program address bit[31:1], bit0 is 0.	0
0	VTF1EN	RW	VTF interrupt 1 enable bit. 1: VTF interrupt 1 channel is enabled; 0: Off.	0

6.5.2.11 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
Reserved	INTEN14	Reserved	INTEN12	Reserved											

Bit	Name	Access	Description	Reset value
[31:16]	INTEN	WO	16#-31# interrupt enable control. 1: Current number interrupt enable. 0: No effect.	0
15	Reserved	RO	Reserved	0
14	INTEN	WO	14# Interrupt enable control. 1: Current number interrupt enable. 0: No effect.	0
13	Reserved	RO	Reserved	0
12	INTEN	WO	12# Interrupt enable control. 1: Current number interrupt enable. 0: No effect.	0
[11:0]	Reserved	RO	Reserved	0

6.5.2.12 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															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									INTEN[38:32]						

Bit	Name	Access	Description	Reset value
[31:7]	Reserved	RO	Reserved	0
[6:0]	INTEN	WO	32#-38# interrupt enable control. 1: current number interrupt enable. 0: No effect.	0

6.5.2.13 PFIC Interrupt Enable Clear Register 1 (PFIC_IRER1)

Offset address: 0x180

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	INTRSET14	Reserved	INTRSET12	Reserved											

Bit	Name	Access	Description	Reset value
[31:16]	INTRSET	WO	16#-31# interrupt shutdown control. 1: current number interrupt off. 0: No effect.	0
15	Reserved	RO	Reserved	0
14	INTRSET	WO	14# Interrupt off control. 1: current number interrupt off. 0: No effect.	0
13	Reserved	RO	Reserved	0
12	INTRSET	WO	12# Interrupt off control. 1: current number interrupt off. 0: No effect.	0
[11:0]	Reserved	RO	Reserved	0

6.5.2.14 PFIC Interrupt Enable Clear Register 2 (PFIC_IRER2)

Offset address: 0x184

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									INTRSET[38:32]						

Bit	Name	Access	Description	Reset value
[31:7]	Reserved	RO	Reserved	0
[6:0]	INTRSET32_38	WO	32#-38# interrupt shutdown control. 1: current number interrupt off. 0: No effect.	0

6.5.2.15 PFIC Interrupt Pending Setup 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
Reserved	PENDSET14	Reserved	PENDSET12	Reserved								PENDSET3	PENDSET2	Reserved	

Bit	Name	Access	Description	Reset value
[31:16]	PENDSET	WO	16#-31# interrupt pending setting. 1: Current numbered interrupt hang. 0: No effect.	0
15	Reserved	RO	Reserved	0
14	PENDSET	WO	14# Interrupt hang setting.	0

			1: current numbered interrupt hang. 0: No effect.	
13	Reserved	RO	Reserved	0
12	PENDSET	WO	12# Interrupt hang setting. 1: current numbered interrupt hang. 0: No effect.	0
[11:4]	Reserved	RO	Reserved	0
[3:2]	PENDSET	WO	2#-3# interrupt pending setting. 1: current number break hang. 0: No effect.	0
[1:0]	Reserved	RO	Reserved	0

6.5.2.16 PFIC Interrupt Pending Setup Register 2 (PFIC_IPSR2)

Offset address: 0x204

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">Reserved</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="9">Reserved</td><td colspan="7">PENDSET[38:32]</td></tr>
</table>

Bit	Name	Access	Description	Reset value
[31:7]	Reserved	RO	Reserved	0
[6:0]	PENDSET	WO	32#-38# interrupt pending setting. 1: current number break hang. 0: No effect.	0

6.5.2.17 PFIC Interrupt Pending Clear Register 1 (PFIC_IPRR1)

Offset address: 0x280

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">PENDRST[31:16]</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Reserved</td><td>PENDRST14</td><td>Reserved</td><td>PENDRST12</td><td colspan="8">Reserved</td><td>PENDRST3</td><td>PENDRST2</td><td colspan="2">Reserved</td></tr>
</table>

Bit	Name	Access	Description	Reset value
[31:16]	PENDRST	WO	16#-31# interrupt hang clear. 1: The current numbered interrupt clears the pending state. 0: No effect.	0
15	Reserved	RO	Reserved	0
14	PENDRST	WO	14# Interrupt hang clear. 1: The current numbered interrupt clears the pending state. 0: No effect.	0
13	Reserved	RO	Reserved	0
12	PENDRST	WO	12# Interrupt hang clear. 1: The current numbered interrupt clears the pending state. 0: No effect.	0
[11:4]	Reserved	RO	Reserved	0
[3:2]	PENDRST	WO	2#-3# interrupt hang clear. 1: The current numbered interrupt clears	0

			the pending state. 0: No effect.	
[1:0]	Reserved	RO	Reserved	0

6.5.2.18 PFIC Interrupt Pending Clear Register 2 (PFIC_IPRR2)

Offset address: 0x284

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									PENDRST[38:32]						

Bit	Name	Access	Description	Reset value
[31:7]	Reserved	RO	Reserved	0
[6:0]	PENDRST	WO	32#-38# interrupt hang clear. 1: The current numbered interrupt clears the pending state. 0: No effect.	0

6.5.2.19 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
Reserved	IACTS14	Reserved	IACTS12	Reserved								IACTS3	IACTS2	Reserved	

Bit	Name	Access	Description	Reset value
[31:16]	IACTS	RO	16#-31# Interrupt execution status. 1: current number interruption in execution. 0: The current number interrupt is not executed.	0
15	Reserved	RO	Reserved	0
14	IACTS	RO	14# Interrupt execution status. 1: current number interruption in execution. 0: The current number interrupt is not executed.	0
13	Reserved	RO	Reserved	0
12	IACTS	RO	12# Interrupt execution status. 1: current number interruption in execution. 0: The current number interrupt is not executed.	0
[11:4]	Reserved	RO	Reserved	0
[3:2]	IACTS	RO	2#-3# interrupt execution status. 1: current number interruption in execution. 0: The current number interrupt is not executed.	0
[1:0]	Reserved	RO	Reserved	0

6.5.2.20 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															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									IACTS [38:32]						

Bit	Name	Access	Description	Reset value
[31:7]	Reserved	RO	Reserved	0
[6:0]	IACTS	RO	32#-38# Interrupt execution status. 1: Current number interruption in execution. 0: The current number interrupt is not executed.	0

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

Offset address: 0x400-0x4FF

The controller supports 256 interrupts (0-255), each using 8 bits to set the control priority.

	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. [7:6:4]: priority control bits. If no nesting is configured, no preemption bits. Bit7 is preempted if 2 levels of nesting are configured. [5:0]: reserved, fixed to 0, write invalid.	0

6.5.2.22 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
SYSRESET	Reserved														

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

Bit	Name	Access	Description	Reset value
31	SYSRESET	WO	System reset, clear 0 automatically. write 1 valid, write 0 invalid, same effect as PFIC_CFGR register.	0
[30:6]	Reserved	RO	Reserved	0

5	SETEVENT	WO	Set the event to wake up the WFE case.	0
4	SEVONPEND	RW	When an event occurs or interrupts a pending state, the system can be woken up from after the WFE instruction, or if the WFE instruction is not executed, the system will be woken up immediately after the next execution of the instruction. 1: enabled events and all interrupts (including unenabled interrupts) can wake up the system. 0: Only enabled events and enabled interrupts can wake up the system.	0
3	WFITOWFE	RW	Execute the WFI command as if it were a WFE. 1: treat the subsequent WFI instruction as a WFE instruction. 0: No effect.	0
2	SLEEPDEEP	RW	Low-power mode of the control system. 1： deep sleep 0： sleep	0
1	SLEEPONEXIT	RW	System status after 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

6.5.3 Dedicated CSR Registers

A number of Control and Status Registers (CSRs) are defined in the RISC-V architecture to configure or identify or record the operational status. The CSR registers are internal to the core and use a dedicated 12-bit address space; the CH32V003 chip adds a number of vendor-defined registers in addition to the standard registers defined in the RISC-V privileged architecture document, which need to be accessed using the csr instruction.

Note: These registers are labeled "MRW, MRO, MRW1" and require the system to be in machine mode to access them.

6.5.3.1 Interrupt System Control Register (INTSYSCR)

CSR address: 0x804

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

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

Bit	Name	Access	Description	Reset value
[31:2]	Reserved	MRO	Reserved	0
1	INESTEN	MRW	Interrupt nesting enable. 0: interrupt nesting function off. 1: Interrupt nesting function is enabled.	0
0	HWSTKEN	MRW	Hardware stack enable. 0: hardware stacking function off. 1: Hardware stacking function is enabled.	0

6.5.3.2 Exception Entry Base Address Register (MTVEC)

CSR address: 0x305

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">BASEADDR[31:16]</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">BASEADDR[15:2]</td><td>MODE1</td><td>MODE0</td></tr>
</table>

Bit	Name	Access	Description	Reset value
[31:2]	BASEADDR[31:2]	MRW	Interrupt vector table base address.	0
1	MODE1	MRW	Interrupt vector table identifies patterns. 0: identification by jump instruction, limited range, support for non-jump instructions. 1: Identify by absolute address, support full range, but must jump.	0
0	MODE0	MRW	Interrupt or exception entry address mode selection. 0: use of a unified entry address. 1: Address offset based on interrupt number *4.	0

6.5.4 STK Register Description

Table 6-5 STK-related registers list

Name	Access address	Description	Reset value
R32_STK_CTLR	0xE000F000	System count control register	0x00000000
R32_STK_SR	0xE000F004	System count status register	0x00000000
R32_STK_CNTL	0xE000F008	System counter register	0x00000000
R32_STK_CMPLR	0xE000F010	Counting comparison register	0x00000000

6.5.4.1 System Count Control Register (STK_CTLR)

Offset address: 0x00

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>SWIE</td><td colspan="15">Reserved</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="12">Reserved</td><td>STRE</td><td>STCLK</td><td>STIE</td><td>STE</td></tr>
</table>

Bit	Name	Access	Description	Reset value
31	SWIE	RW	Software interrupt trigger enable (SWI). 1: Triggering software interrupts. 0: Turn off the trigger. After entering software interrupt, software clear 0 is required, otherwise it is continuously triggered.	0
[30:4]	Reserved	RO	Reserved	
3	STRE	RW	Auto-reload count enable bit. 1: Re-counting from 0 after counting up to the comparison value. 0: Count up to the comparison value and continue counting up, count down to 0 and start counting down again from the maximum value.	
2	STCLK	RW	Counter clock source selection bit. 1: HCLK for time base.	

			0: HCLK/8 for time base.	
1	STIE	RW	Counter interrupt enable control bit. 1: Enable counter interrupt. 0: Disable counter interrupt.	
0	STE	RW	System counter enable control bit. 1: Turn on the system counter STK. 0: Turn off the system counter STK and the counter stops counting.	0

6.5.4.2 System Count Status Register (STK_SR)

Offset address: 0x04

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

Bit	Name	Access	Description	Reset value
[31:1]	Reserved	RO	Reserved	0
0	CNTIF	RW0	Count value comparison flag, write 0 to clear, write 1 to invalidate. 1: Up count reaches the comparison value. 0: The comparison value is not reached.	0

6.5.4.3 System Counter Register (STK_CNTL)

Offset address: 0x08

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

Bit	Name	Access	Description	Reset value
[31:0]	CNT[31:0]	RW	The current counter count value is 32 bits.	0

6.5.4.4 Counting Comparison Register (STK_CMPLR)

Offset address: 0x10

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

Bit	Name	Access	Description	Reset value
[31:0]	CMP[31:0]	RW	Set the comparison counter value to 32 bits.	0

Chapter 7 GPIO and Alternate Function (GPIO/AFIO)

The GPIO port can be configured for multiple input or output modes, with built-in pull-up or pull-down resistors that can be turned off, and can be configured for push-pull or open-drain functions. the GPIO port can also be multiplexed for other functions.

7.1 Main Features

Each pin of the port can be configured to one of the following multiple modes.

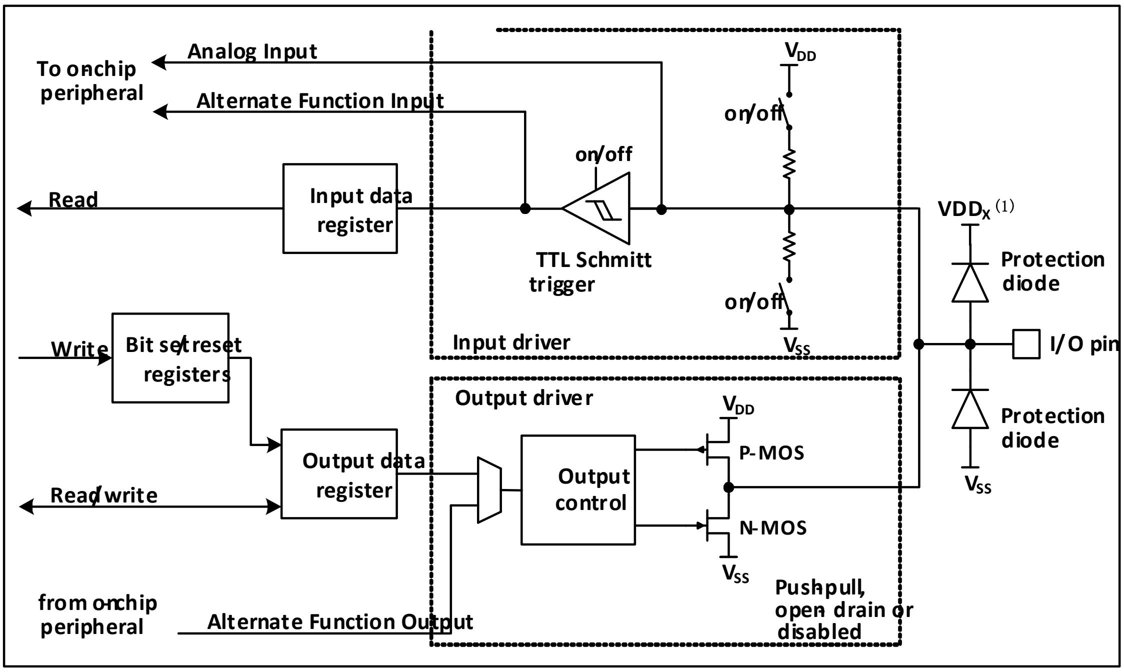
- Floating input
- Pull-up input
- Dropdown input
- Analog input
- Open drain output
- Push-pull output
- Multiplexing the inputs and outputs of functions

Many pins have multiplexing capabilities, and many other peripherals map their output and input channels to these pins. The specific usage of these multiplexed pins needs to be referred to the individual peripherals, and the content of whether these pins are multiplexed and remapped is explained in this chapter.

7.2 Function Description

7.2.1 Overview

Figure 7-1 GPIO module basic structure block diagram



Note: (1) VDDx is VDD when GPIO is normal IO, and VDDx is VDD_FT when GPIO is FT IO.

As shown in Figure 7-1 IO port structure, each pin has two protection diodes inside the chip, and the IO port can be divided into input and output driver modules internally. Among them, the input driver has a weak pull-up and pull-down resistor optional, which can be connected to AD and other analog input peripherals; if the input is to a digital peripheral, it needs to go through a TTL Schmitt trigger and then connect to GPIO input registers or other multiplexed peripherals. The output driver has a pair of MOS tubes, and the IO port can be configured as open-drain or push-pull output by configuring whether the upper and lower MOS tubes are enabled or not; the output driver can also be configured internally to control the output by GPIO or by other multiplexed peripherals.

7.2.2 GPIO Initialization Function

Just after reset, the GPIO ports run in the initial state, when most IO ports are running in the floating input state, but there are also peripheral related pins such as HSE that are running on the peripheral multiplexing function. Please refer to the chapter related to pin description for the specific initialization function.

7.2.3 External Interrupts

All GPIO ports can be configured with external interrupt input channels, but an external interrupt input channel can only be mapped to at most one GPIO pin, and the serial number of the external interrupt channel must be the same as the bit number of the GPIO port, for example, PA1 (or PC1, PD1, etc.) can only be mapped to EXTI1, and EXTI1 can only accept one of PA1, PC1 or PD1, etc. The mapping of both parties is one-to-one.

7.2.4 Multiplexing Functions

It is important to note that using the multiplexing function.

- To use the multiplexing function in the input direction, the port must be configured in multiplexed input mode, and the pull-down settings can be set according to actual needs.
- Using the multiplexing function in the output direction, the port must be configured in multiplexed output mode, push-pull or open-drain can be set according to the actual situation.
- For bidirectional multiplexing, the port must be configured in multiplexed output mode, when the driver is configured in floating input mode

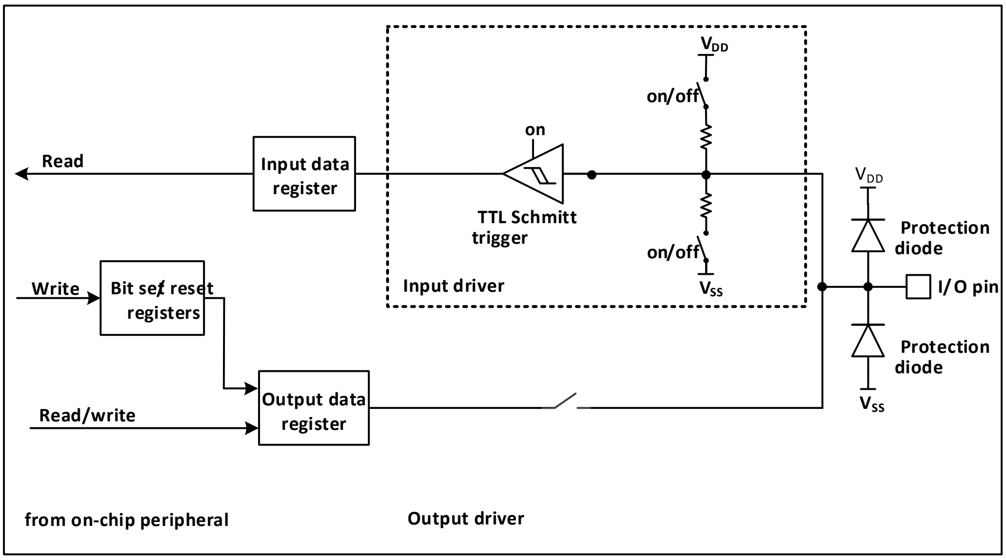
The same IO port may have multiple peripherals multiplexed to this pin, so in order to maximize the space for each peripheral, the multiplexed pins of peripherals can be remapped to other pins in addition to the default multiplexed pins, avoiding the occupied pins.

7.2.5 Locking Mechanism

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

7.2.6 Input Configuration

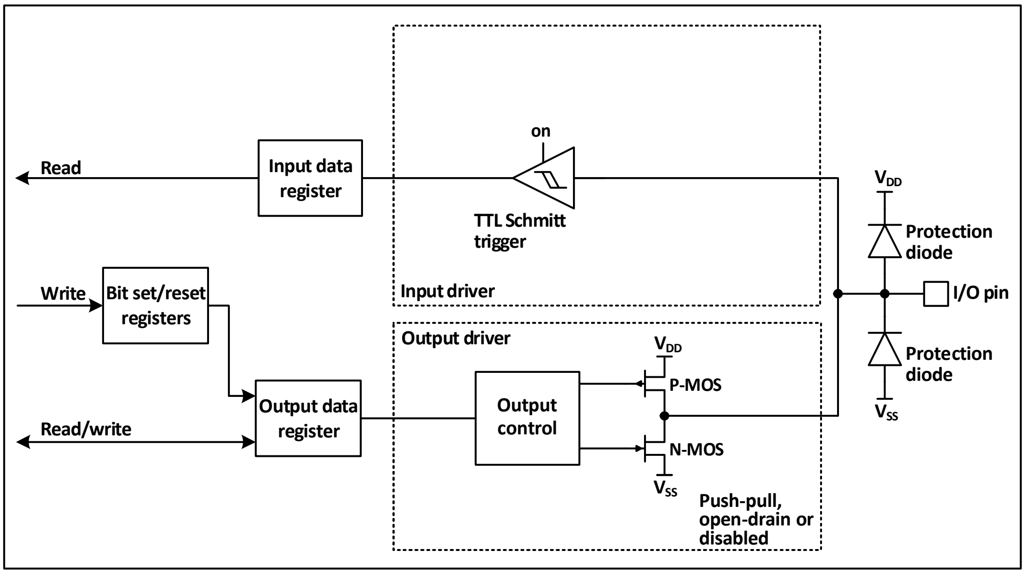
Figure 7-2 GPIO module input configuration structure block diagram



When the IO port is configured in input mode, the output driver is disconnected, the input pull-up and pull-down are selectable, and no multiplexed functions or analog inputs are connected. The data on each IO port is sampled into the input data register at each AHB clock, and the level status of the corresponding pin is obtained by reading the corresponding bit of the input data register.

7.2.7 Output Configuration

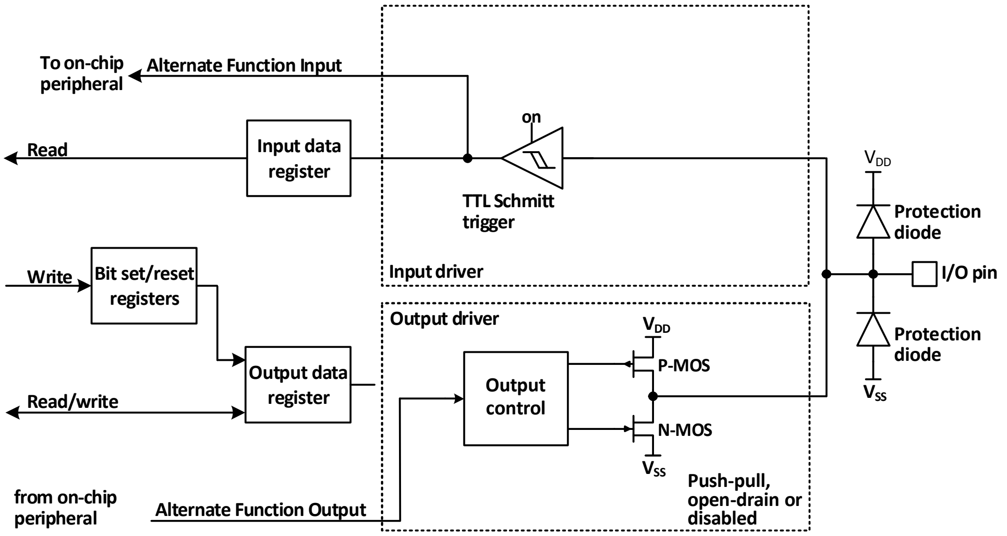
Figure 7-3 GPIO module output configuration structure block diagram



When the IO port is configured to output mode, the pair of MOS in the output driver can be configured to push-pull or open-drain mode as needed, without using the multiplexing function. The pull-up and pull-down resistors of the input driver are disabled, the TTL Schmitt trigger is activated, and the levels appearing on the IO pins will be sampled into the input data registers at each AHB clock, so reading the input data registers will give the IO status, and in push-pull output mode, access to the output data registers will give the last written value.

7.2.8 Multiplexing Function Configuration

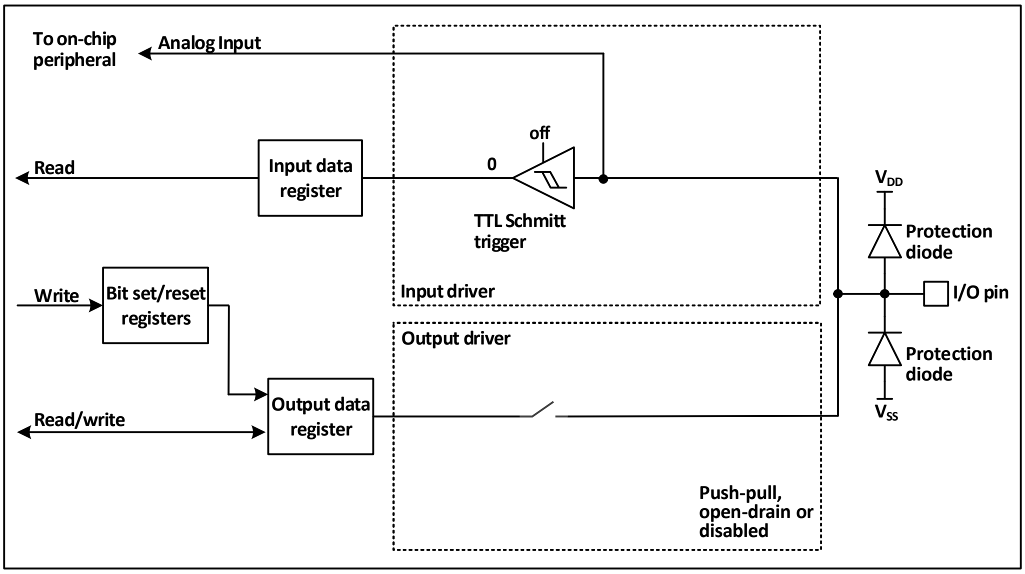
Figure 7-4 The structure of GPIO module when it is multiplexed by other peripherals



When multiplexing is enabled, the output driver is enabled and can be configured to open-drain or push-pull mode as desired, the Schmitt trigger is turned on, the input and output lines of the multiplexing function are connected, but the output data registers are disconnected, and the levels appearing on the IO pins will be sampled into the input data registers at each AHB clock. In open-drain mode, reading the input data register will give the current status of the IO port; in push-pull mode, reading the output data register will give the last written value.

7.2.9 Analog Input Configuration

Figure 7-5 The configuration structure when the GPIO module is used as an analog input



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

7.2.10 GPIO Settings for Peripherals

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

Table 7-1 Advanced-control timer (TIM1)

TIM1 pins	Configuration	GPIO configuration
TIM1_CHx	Input capture channel x	Floating input
	Output comparison channel x	Push-pull multiplexed output
TIM1_CHxN	Complementary output channels x	Push-pull multiplexed output
TIM1_BKIN	Brake input	Floating input
TIM1_ETR	Externally triggered clock input	Floating input

Table 7-2 General-purpose timer (TIM2)

TIM2 pins	Configuration	GPIO configuration
TIM2_CHx	Input capture channel x	Floating input
	Output comparison channel x	Push-pull multiplexed output
TIM2_ETR	Externally triggered clock input	Floating input

Table 7-3 Universal synchronous asynchronous serial transceiver (USART)

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

Table 7-4 Serial peripheral interface (SPI) modules

SPI pins	Configuration	GPIO configuration
SPIx_SCK	Master mode	Push-pull multiplexed output
	Slave mode	Floating input

SPIx_MOSI	Full-duplex Master mode	Push-pull multiplexed output
	Full-duplex Slave mode	Floating input or pull-up input
	Simple bi-directional data line/Master mode	Push-pull multiplexed output
	Simple bi-directional data line/Slave mode	Not used
SPIx_MISO	Full-duplex Master mode	Floating input or pull-up input
	Full-duplex Slave mode	Push-pull multiplexed output
	Simple bi-directional data line/Master mode	Not used
	Simple bi-directional data line/Slave mode	Push-pull multiplexed output
SPIx_NSS	Hardware Master or Slave mode	Float, pull-up or pull-down input
	Hardware Master mode/NSS output enable mode	Push-pull multiplexed output
	Software mode	Not used

Table 7-5 Internal integrated bus (I2C) module

I2C pins	Configuration	GPIO configuration
I2C_SCL	I2C clock	Open-drain multiplexed output
I2C_SDA	I2C data	Open-drain multiplexed output

Table 7-6 Analog-to-digital converters (ADCs)

ADC pin	GPIO configuration
ADC	Analog input

Table 7-7 Other I/O function settings

Pins	Configuration features	GPIO configuration
MCO	Clock output	Push-pull multiplexed output
EXTI	External interrupt input	Float, pull-up or pull-down input
OPA	Operational Amplifier Input	Floating input

7.2.11 Alternate Function Remapping GPIO Configuration

7.2.11.1 Timer Alternate Function Remapping

Table 7-8 TIM1 alternate function remapping

Alternate function	TIM1_RM=00 Default mapping	TIM1_RM=01 Partial mapping	TIM1_RM=10 Partial mapping	TIM1_RM=11 Full mapping
TIM1_ETR	PC5	PC5	PD4	PC2
TIM1_CH1	PD2	PC6	PD2	PC4
TIM1_CH2	PA1	PC7	PA1	PC7
TIM1_CH3	PC3	PC0	PC3	PC5
TIM1_CH4	PC4	PD3	PC4	PD4
TIM1_BKIN	PC2	PC1	PC2	PC1
TIM1_CH1N	PD0	PC3	PD0	PC3
TIM1_CH2N	PA2	PC4	PA2	PD2
TIM1_CH3N	PD1	PD1	PD1	PC6

Note: For the mapping function of TIM1_CH1 in the table, the condition is TIM1_1_RM=0. When TIM1_1_RM=1, TIM1_CHI is mapped to LSI (for LSI calibration).

Table 7-9 TIM2 alternate function remapping

Alternate function	TIM2_RM=00 Default mapping	TIM2_RM=01 Partial mapping	TIM2_RM=10 Partial mapping	TIM2_RM=11 Full mapping
TIM2_ETR	PD4	PC5	PC1	PC1
TIM2_CH1	PD4	PC5	PC1	PC1
TIM2_CH2	PD3	PC2	PD3	PC7
TIM2_CH3	PC0	PD2	PC0	PD6
TIM2_CH4	PD7	PC1	PD7	PD5

7.2.11.2 USART Alternate Function Remapping

Table 7-10 USART1 alternate function remapping

Alternate function	USART1_RM=00 Default mapping	USART1_RM=01 Partial mapping	USART1_RM=10 Partial mapping	USART1_RM=11 Full mapping
USART1_CK	PD4	PD7	PD7	PC5
USART1_TX	PD5	PD0	PD6	PC0
USART1_RX	PD6	PD1	PD5	PC1
USART1_CTS	PD3	PC3	PC6	PC6
USART1_RTS	PC2	PC2	PC7	PC7

7.2.11.3 SPI Alternate Function Remapping

Table 7-11 SPI alternate function remapping

Alternate function	SPI1_RM=0 Default mapping	SPI1_RM=1 Remapping
SPI1_NSS	PC1	PC0
SPI1_SCK	PC5	PC5
SPI1_MISO	PC7	PC7
SPI1_MOSI	PC6	PC6

7.2.11.4 I2C Alternate Function Remapping

Table 7-12 I2C alternate function remapping

Alternate function	I2C1_RM=00 Default mapping	I2C1_RM=01 Remapping	I2C1_RM=1x Remapping
I2C1_SCL	PC2	PD1	PC5
I2C1_SDA	PC1	PD0	PC6

7.2.11.4 ADC Alternate Function Remapping

Table 7-13 ADC external trigger injection conversion alternate function remapping

Alternate function	ADC_ETRGINJ_RM=0 Default mapping	ADC_ETRGINJ_RM=1 Remapping
ADC external trigger injection conversion	ADC externally triggered injection conversion connected to PD1	ADC externally triggered injection conversion connected to PA2

Table 7-13 ADC external trigger rule conversion alternate function remapping

Alternate function	ADC_ETRGREG_RM=0 Default mapping	ADC_ETRGREG_RM=1 Remapping
ADC external trigger rule conversion	ADC externally triggered rule conversion connected to PD3	ADC externally triggered rule conversion connected to PC2

7.3 Register Description

7.3.1 GPIO Register Description

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

Table 7-8 GPIO-related registers list

Name	Access address	Description	Reset value
R32_GPIOA_CFGLR	0x40010800	PA port configuration register low	0x44444444
R32_GPIOC_CFGLR	0x40011000	PC port configuration register low	0x44444444
R32_GPIOD_CFGLR	0x40011400	PD port configuration register low	0x44444444
R32_GPIOA_INDR	0x40010808	PA 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_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_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_GPIOC_BCR	0x40011014	PC port reset register	0x00000000
R32_GPIOD_BCR	0x40011414	PD port reset register	0x00000000
R32_GPIOA_LCKR	0x40010818	PA port configuration lock register	0x00000000
R32_GPIOC_LCKR	0x40011018	PC port configuration lock register	0x00000000
R32_GPIOD_LCKR	0x40011418	PD port configuration lock register	0x00000000

7.3.1.1 Port Configuration Register Low (GPIOx_CFGLR) (x=A/C/D)

Offset address: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNF7[1:0]		MODE7[1:0]		CNF6[1:0]		MODE6[1:0]		CNF5[1:0]		MODE5[1:0]		CNF4[1:0]		MODE4[1:0]	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNF3[1:0]		MODE3[1:0]		CNF2[1:0]		MODE2[1:0]		CNF1[1:0]		MODE1[1:0]		CNF0[1:0]		MODE0[1:0]	

Bit	Name	Access	Description	Reset value
[31:30] [27:26] [23:22] [19:18] [15:14] [11:10] [7:6] [3:2]	CNFy[1:0]	RW	(y=0-7), the configuration bits for port x, by which the corresponding port is configured. When in input mode (MODE=00b). 00: Analog input mode. 01: Floating input mode. 10: With pull-up and pull-down mode. 11: Reserved. In output mode (MODE>00b). 00: Universal push-pull output mode. 01: Universal open-drain output mode. 10: Multiplexed 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[1:0]	RW	(y=0-7), port x mode selection, configure the corresponding port by these bits. 00: Input mode. 01: Output mode, maximum speed 10MHz; 10: Output mode, maximum speed 2MHz. 11: Output mode, maximum speed 50MHz.	00b

7.3.1.2 Port Input Register (GPIOx_INDR) (x=A/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**
Reserved								IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0

Bit	Name	Access	Description	Reset value
[31:8]	Reserved	RO	Reserved	0
[7:0]	IDRy	RO	(y=0-7), the port input data. These bits are read-only and can only be read out in 16-bit form. The value read is the high and low state of the corresponding bit.	X

7.3.1.3 Port Output Register (GPIOx_OUTDR) (x=A/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**
Reserved								ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0

Bit	Name	Access	Description	Reset value
[31:8]	Reserved	RO	Reserved	0
[7:0]	ODRy	RW	For output modes. (y=0-7), the data output by the port. These data can only be operated in 16-bit form. the I/O port outputs the values of these registers externally. For modes with drop-down inputs. 0: Drop-down input. 1: Pull-up input.	0

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

Offset address: 0x10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved								BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0

Bit	Name	Access	Description	Reset value
[31:24]	Reserved	R0	Reserved	0

[23:16]	BRy	WO	(y=0-7), the corresponding OUTDR bits are cleared for these location bits, and writing 0 has no effect. These bits can only be accessed in 16-bit form. If both BR and BS bits are set, the BS bit takes effect.	0
[15:8]	Reserved	RO	Reserved	0
[7:0]	BSy	WO	(y=0-7), for which the location bits will make the corresponding OUTDR location bits, writing 0 has no effect. These bits can only be accessed in 16-bit form. If both BR and BS bits are set, the BS bit takes effect.	0

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

Offset address: 0x14

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0

Bit	Name	Access	Description	Reset value
[31:8]	Reserved	RO	Reserved	0
[7:0]	BRy	WO	(y=0-7), the corresponding OUTDR bits are cleared for these location bits, and writing 0 has no effect. These bits can only be accessed in 16-bit form.	0

7.3.1.6 Port Configuration Lock Register (GPIOx_LCKR) (x=A/C/D)

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							LCKK	LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0

Bit	Name	Access	Description	Reset value
[31:9]	Reserved	RO	Reserved	0
8	LCKK	RW	The lock key, which can be written in a specific sequence to achieve locking, but which can be read out at any time. It reads 0 to indicate that no locking is in effect, and reads 1 to indicate that locking is in effect. The write sequence for the lock key is: write 1 - write 0 - write 1 - read 0 - read 1. The last step is not necessary, but can be used to confirm that the lock key is active. Any error while writing the sequence will not enable the activation of the lock and the value of LCK[7:0] cannot be changed while the sequence is being written. After the lock is in effect, the port configuration can only be changed after the next reset.	0
[7:0]	LCKy	RW	(y=0-7), these bits are 1 to indicate locking the configuration of the corresponding port. These bits can only be changed before the LCKK is	0

			unlocked. The locked configuration refers to the configuration registers GPIOx_CFGLR and GPIOx_CFGHR.	

Note: After the LOCK sequence is executed for the corresponding port bit, the configuration of the port bit will not be changed again until the next system reset.

7.3.2 AFIO Register Description

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

Table 7-9 List of AFIO-related registers

Name	Access address	Description	Reset value
R32_AFIO_PCFR1	0x40010004	Remap Register 1	0x00000000
R32_AFIO_EXTICR	0x40010008	External interrupt configuration register 1	0x00000000

7.3.2.1 Remap Register 1 (AFIO_PCFR1)

Offset address: 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					SWCFG[2:0]			TIM1_RM	I2C1RM1	USART1RM1	Reserved		ADC1_ETRGREG_RM	ADC1_ETRGINJ_RM	Reserved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PA12_RM	Reserved					TIM2_RM [1:0]		TIM1_RM [1:0]		Reserved			USART1_RM	I2C1_RM	SPI1_RM

Bit	Name	Access	Description	Reset value
[31:27]	Reserved	RO	Reserved	0
[26:24]	SWCFG[2:0]	RW	These bits are used to configure the I/O ports for SW function and trace function. SWD (SDI) is the debug interface to access the core. It is always used as a SWD port after system reset. 0xx: SWD (SDI) enabled. 100: Turn off SWD (SDI), which functions as a GPIO. Others: Invalid.	0
23	TIM1_IREMAP	RW	Control timer 1 channel 1 selection 0: Select external pins 1: Select internal LSI clock	0
22	I2C1REMAP1	RW	I2C1 remapping high bit (used in conjunction with AFIO_PCFR1 register bit1 I2C1_RM [22,1]). 00: default mapping (SCL/PC2, SDA/PC1). 01: Remapping (SCL/ PD1, SDA/ PD0). 1X: Remapping (SCL/PC5, SDA/PC6)	0
21	USART1_RM1	RW	USART1 mapping configuration high (used in conjunction with AFIO PCFR1 register bit2 USART1RM [21,2]). 00: default mapping (CK/PD4, TX/PD5, RX/PD6, CTS/PD3, RTS/PC2). 01: Remapping (CK/PD7, TX/PD0, RX/PD1, CTS/PC3, RTS/PC2, SW_RX/PD0). 10: Remapping (CK/PD7, TX/PD6, RX/PD5, CTS/PC6, RTS/PC7, SW_RX/PD6).	0

			11: Remapping (CK/PC5, TX/PC0, RX/PC1, CTS/PC6, RTS/PC7, SW_RX/PC0).	
[20:19]	Reserved	RO	Reserved	0
18	ADC_ETRGREG_RM	RW	Remap bit for ADC external trigger rule conversion. 0: ADC external trigger rule conversion connected to PD3. 1: ADC external trigger rule conversion connected to PC2.	0
17	ADC_ETRGINJ_RM	RW	Remap bit for ADC external trigger rule conversion. 0: ADC external trigger rule conversion connected to PD3. 1: ADC external trigger rule conversion connected to PC2.	0
16	Reserved	RO	Reserved	0
15	PA12_RM	RW	Pin PA1 & PA2 remapping bit, this bit can be read or written by user. It controls the proper function of PA1 and PA2 (set to 1 when connected to an external crystal pin) 0: Pin is used as GPIO and multiplexed function 1: No functional role for pins	0
[14:10]	Reserved	RO	Reserved	0
[9:8]	TIM2_RM[1:0]	RW	Remap bits for timer 2. These bits can be read and written by the user. It controls the mapping of Timer 2's channels 1 through 4 and external trigger (ETR) on the GPIO ports. 00: Default mapping (CH1/ETR/PD4, CH2/PD3, CH3/PC0, CH4/PD7). 01: Partial mapping (CH1/ETR/PC5, CH2/PC2, CH3/PD2, CH4/PC1). 10: Partial mapping (CH1/ETR/PC1, CH2/PD3, CH3/PC0, CH4/PD7). 11: Complete mapping (CH1/ETR/PC1, CH2/PC7, CH3/PD6, CH4/PD5).	0
[7:6]	TIM1_RM[1:0]	RW	Remap bits for timer 1. These bits can be read and written by the user. It controls the mapping of channels 1 to 4, 1N to 3N, external trigger (ETR) and brake input (BKIN) of timer 1 to the GPIO ports. 00: Default mapping (ETR/PC5, CH1/PD2, CH2/PA1, CH3/PC3, CH4/PC4, BKIN/PC2, CH1N/PD0, CH2N/PA2, CH3N/PD1). 01: Partial mapping (ETR/PC5, CH1/PC6, CH2/PC7, CH3/PC0, CH4/PD3, BKIN/PC1, CH1N/PC3, CH2N/PC4, CH3N/PD1). 10: Partial mapping (ETR/PD4, CH1/PD2, CH2/PA1, CH3/PC3, CH4/PC4, BKIN/PC2, CH1N/PD0, CH2N/PA2, CH3N/PD1). 11: Complete mapping (ETR/PC2, CH1/PC4, CH2/PC7, CH3/PC5, CH4/PD4, BKIN/PC1, CH1N/PC3, CH2N/PD2, CH3N/PC6).	0
[5:3]	Reserved	RO	Reserved	0
2	USART1_RM	RW	USART1 mapping configuration low bit (used in conjunction with AFIO PCFR1 register bit21 USART1REMAP1 [21,2]). 00: Default mapping (CK/PD4, TX/PD5, RX/PD6, CTS/PD3, RTS/PC2). 01: Remapping (CK/PD7, TX/PD0, RX/PD1,	0

			CTS/PC3, RTS/PC2, SW_RX/PD0). 10: Remapping (CK/PD7, TX/PD6, RX/PD5, CTS/PC6, RTS/PC7, SW_RX/PD6). 11: Remapping (CK/PC5, TX/PC0, RX/PC1, CTS/PC6, RTS/PC7, SW_RX/PC0).	
1	I2C1_RM	RW	I2C1 remapping low bit (used in conjunction with AFIO_PCFR1 register bit22 I2C1_RM1 [22,1]). 00: Default mapping (SCL/PC2, SDA/PC1). 01: Remapping (SCL/ PD1, SDA/ PD0). 1X: Remapping (SCL/PC5, SDA/PC6)	0
0	SPI1_RM	RW	Remapping of SPI1. This bit can be read or written by the user. It controls the mapping of SPI1's NSS, SCK, MISO, and MOSI multiplexing functions to the GPIO ports. 0: Default mapping (NSS/PC1, CK/PC5, MISO/PC7, MOSI/PC6). 1: Remapping (NSS/PC0, CK/PC5, MISO/PC7, MOSI/PC6).	0

7.3.2.2 External Interrupt Configuration Register 1 (AFIO_EXTICR)

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
EXTI7[1:0]		EXTI6[1:0]		EXTI5[1:0]		EXTI4[1:0]		EXTI3[1:0]		EXTI2[1:0]		EXTI1[1:0]		EXTI0[1:0]	

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:14] [13:12] [11:10] [9:8] [7:6] [5:4] [3:2] [1:0]	EXTIx[1:0]	RW	(x=0-7), external interrupt input pin configuration bit. Used to determine to which port pins the external interrupt pins are mapped. 00: xth pin of the PA pin. 10: xth pin of the PC pin. 11: xth pin of the PD pin.	0

Chapter 8 Direct Memory Access Control (DMA)

Direct Memory Access Controller (DMA) provides a high-speed data transfer method between peripherals and memory or between memory and memory without CPU intervention, and data can be moved quickly through DMA to save CPU resources for other operations.

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

8.1 Main Features

- Multiple independently configurable channels
- Each channel is directly connected to a dedicated hardware DMA request and supports software triggering
- Buffer management with loop support
- Request priority between multiple channels can be set by software programming (very high, high, medium and low) and priority setting is determined by the channel number when equal (the lower the channel number the higher the priority)
- Supports peripheral-to-memory, memory-to-peripheral, and memory-to-memory transfers
- Flash memory, SRAM, peripheral SRAM and AHB peripherals can be used as access sources and targets
- Programmable number of data transfer bytes: up to 65535

8.2 Function Description

8.2.1 DMA Channel Processing

1) Arbitration priority

DMA requests generated by multiple independent channels are fed to the DMA controller via a logical or structure, and only one channel request is currently responded to. An arbiter inside the module selects the peripheral/memory access to be initiated based on the priority of the channel request.

In software management, the application can configure the priority level for each channel independently by setting the PL[1:0] bits of the DMA_CFGRx register, including four levels: highest, high, medium and low. When the software setting levels are the same between channels, the module will be selected according to a fixed hardware priority, with the lower channel number having a higher priority than the higher one.

2) DMA configuration

When the DMA controller receives a request signal, it accesses the requested peripheral or memory and establishes a data transfer between the peripheral or memory and the memory. It consists of the following 3 main operation steps.

(1) Fetch data from the memory address indicated by the Peripheral Data Register or the Current Peripheral/Memory Address Register. The start address for the first transfer is the peripheral base address or memory address specified by the DMA_PADDRx or DMA_MADDRx registers.

(2) Store data to the memory address indicated by the Peripheral Data Register or the Current Peripheral/Memory Address Register, and the start address for the first transfer is the peripheral base address or memory address specified by the DMA_PADDRx or DMA_MADDRx registers.

(3) Performs a decrement operation of the value in the DMA_CNTRx register, which indicates the number of operations currently outstanding for transfer.

Each channel includes 3 types of 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 a peripheral request signal. After configuring this mode (MEM2MEM=1), the channel is turned on (EN=1) to start data transfer. This mode does not support cyclic *mode.*

The configuration process is as follows.

1) Set the first address of the peripheral register or the memory data address in the memory-to-memory mode (MEM2MEM=1) in the DMA_PADDRx register. This address will be the source or destination address for data transfer when a DMA request occurs.
2) Set the memory data address in the DMA_MADDRx register. When a DMA request occurs, the transferred data will be read from or written to this address.
3) Set the amount of data to be transferred in the DMA_CNTRx register. This value is decremented after each data transfer.
4) Set the priority of the channel in the PL[1:0] bits of the DMA_CFGRx register.
5) Set the direction of data transfer, cyclic mode, incremental mode for peripheral and memory, data width for peripheral and memory, transfer halfway, transfer complete, and transfer error interrupt enable bits in the DMA_CFGRx register.
6) Set the ENABLE bit of the DMA_CCRx register to start channel x.

Note: The DMA_PADDRx/DMA_MADDRx/DMA_CNTRx registers and the direction of data transfer (DIR), cyclic mode (location), and incremental mode of peripherals and memory (MINC/PINC) control bits in the DMA_CFGRx register can be configured to write only when the DMA channel is turned off.

3) Circular mode

Setting CIRC position 1 of the DMA_CFGRx register enables the cyclic mode function for channel data transfers. In cyclic mode, when the number of data transfers becomes 0, the contents of the DMA_CNTRx register are automatically reloaded to its initial value, and the internal peripheral and memory address registers are reloaded to the initial address values set by the DMA_PADDRx and DMA_MADDRx registers, and DMA operation will continue until the channel is turned off or the DMA mode is turned off.

4) DMA processing status

- Transfer half: It corresponds to the hardware setting of HTIFx bit in DMA_INTFR register. The DMA transfer bytes half flag will be generated when the number of DMA transfers is reduced to less than half of the initial set value, and an interrupt will be generated if HTIE is set in the DMA_CCRx register. The hardware uses this flag to alert the application that it can prepare for a new round of data transfers.
- Transfer completion: corresponds to the hardware setting of the TCIFx bit in the DMA_INTFR register. When the number of DMA transfer bytes decreases to 0, the DMA transfer completion flag will be generated, and if TCIE is set in the DMA_CCRx register, an interrupt will be generated.
- Transfer error: corresponds to a hardware set of the TEIFx bit in the DMA_INTFR register. Reading and writing a reserved address area will generate a DMA transfer error. At the same time the module hardware will automatically clear the EN bit of the DMA_CCRx register corresponding to the channel where the error occurred, and the channel is turned off. If TEIE is set in the DMA_CCRx register, an interrupt will be generated.

When the application queries the DMA channel status, it can first access the GIFx bit of the DMA_INTFR register to determine which channel is currently experiencing a DMA event, and then process the specific DMA event content for that channel.

8.2.2 Programmable Total Data Transfer Size/Data Bit Width/Alignment

The total size of the data to be transferred per DMA channel round is programmable up to 65535 times, and the number of pending transfer bytes is indicated in the DMA_CNTRx register. At EN=0, the set value is written, and at EN=1 when the DMA transfer channel is turned on, this register becomes a read-only attribute with a decreasing value after each transfer.

The transferred data fetch values of peripherals and memories support the address pointer auto-increment function with programmable pointer increments. The first transmitted data address they access is stored in the

DMA_PADDRx and DMA_MADDRx registers.By setting the PINC bit or MINC position 1 of the DMA_CFGRx register, the peripheral address self-increment mode or memory address self-increment mode can be enabled, respectively. PSIZE[1:0] sets the peripheral address fetch data size and address selfincrement size. MSIZE[1:0] sets the memory address to take the data size and address self-increasing small, including three choices: 8-bit, 16-bit, 32-bit. The specific data transfer methods are listed in the following table.

Table 8-1 DMA transfer with different data bit widths (PINC=MINC=1)

Source bit width	Objectives bit width	Transmission number	Source: address/data	Target: address/data	Transfer operations
8	8	4	0x00/B0 0x01/B1 0x02/B2 0x03/B3	0x00/B0 0x01/B1 0x02/B2 0x03/B3	● The source address increment is aligned with the data bit width set at the source and takes a value equal to the data bit width at the source ● The target address increment is aligned with the bit width of the target setup data and takes a value equal to the target data bit width ● DMA transfer of data sent to the target based on the principle: the high bit of the data size is not enough to make up 0, the high bit of the data size overflow is removed ● Storage data mode: small-end mode, low address stores low bytes, high address stores high bytes
8	16	4	0x00/B0 0x01/B1 0x02/B2 0x03/B3	0x00/00B0 0x02/00B1 0x04/00B2 0x06/00B3	
8	32	4	0x00/B0 0x01/B1 0x02/B2 0x03/B3	0x00/000000B0 0x04/000000B1 0x08/000000B2 0x0C/000000B3	
16	8	4	0x00/B1B0 0x02/B3B2 0x04/B5B4 0x06/B7B6	0x00/B0 0x01/B2 0x02/B4 0x03/B6	
16	16	4	0x00/B1B0 0x02/B3B2 0x04/B5B4 0x06/B7B6	0x00/B1B0 0x02/B3B2 0x04/B5B4 0x06/B7B6	
16	32	4	0x00/B1B0 0x02/B3B2 0x04/B5B4 0x06/B7B6	0x00/0000B1B0 0x04/0000B3B2 0x08/0000B5B4 0x0C/0000B7B6	
32	8	4	0x00/B3B2B1B0 0x04/B7B6B5B4 0x08/BBBAB9B8 0x0C/BFBEBDBC	0x00/B0 0x01/B4 0x02/B8 0x03/BC	
32	16	4	0x00/B3B2B1B0 0x04/B7B6B5B4 0x08/BBBAB9B8 0x0C/BFBEBDBC	0x00/B1B0 0x02/B5B4 0x04/B9B8 0x06/BDBC	
32	32	4	0x00/B3B2B1B0 0x04/B7B6B5B4 0x08/BBBAB9B8 0x0C/BFBEBDBC	0x00/B3B2B1B0 0x04/B7B6B5B4 0x08/BBBAB9B8 0x0C/BFBEBDBC	

8.2.3 DMA Request Mapping

The DMA controller provides seven channels, each corresponding to multiple peripheral requests. By setting the corresponding DMA control bits in the corresponding peripheral registers, the DMA function of each peripheral can be turned on or off independently, and the specific correspondence is as follows.

Figure 8-1 DMA1 request image

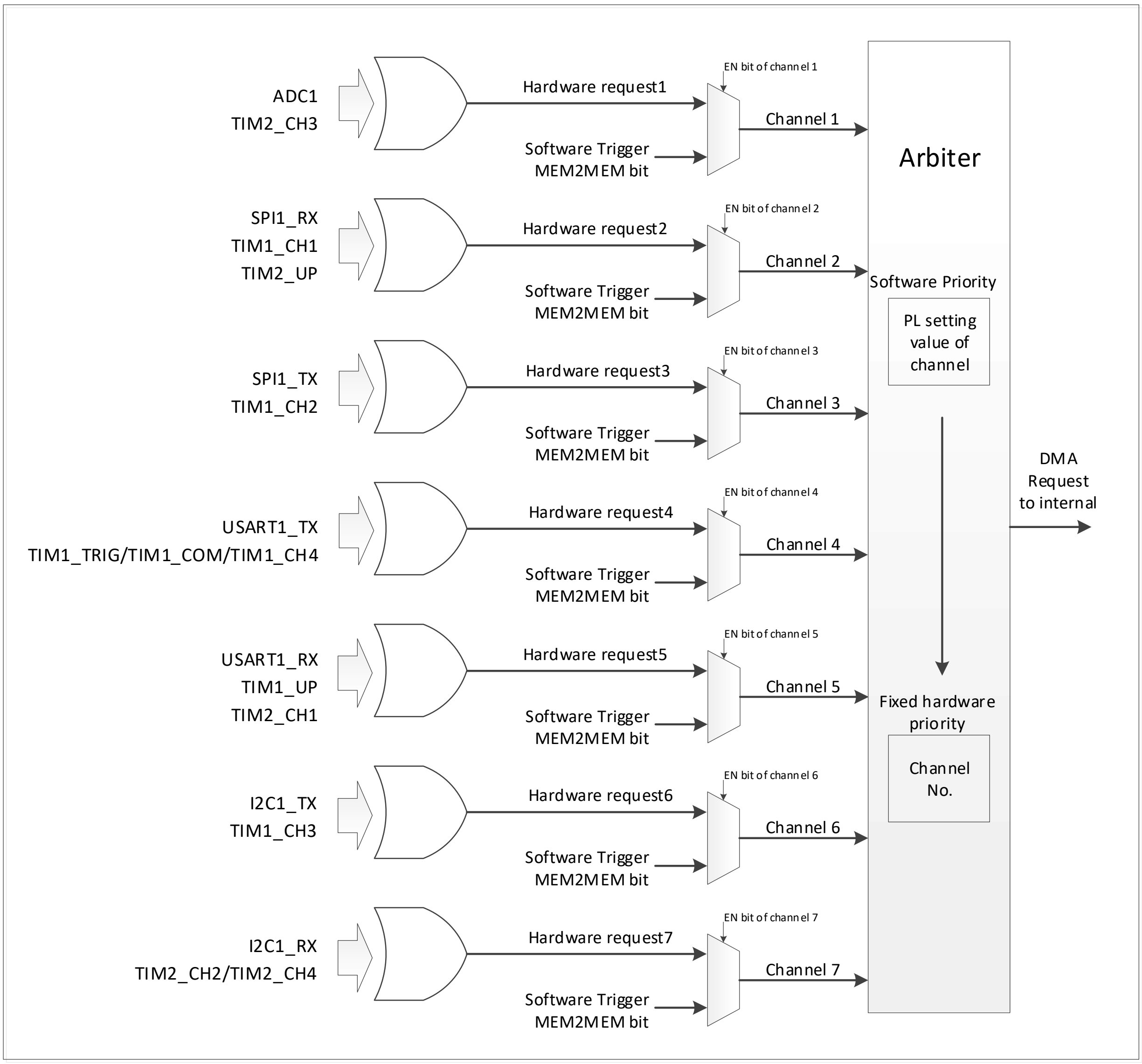


Table 8-2 DMA1 peripheral mapping table for each channel

Peripherals	Channel1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7
ADC1	ADC1						
SPI1		SPI1_RX	SPI1_TX				
USART1				USART1_TX	USART1_RX		
I2C1						I2C1_TX	I2C1_RX
TIM1		TIM1_CH1	TIM1_CH2	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_UP	TIM1_CH3	
TIM2	TIM2_CH3	TIM2_UP			TIM2_CH1		TIM2_CH2 TIM2_CH4

8.3 Register Description

Table 8-3 DMA-related registers list

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

8.3.1 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 for channel x (x=1/2/3/4/5/6/7). 1: A transmission error occurred on channel x. 0: No transmission error on channel x. Hardware set, software write CTEIFx bit to clear this flag.	0
26/22/18/14/10/6/2	HTIFx	RO	Transmission halfway flag for channel x (x=1/2/3/4/5/6/7). 1: a transmission over half event is generated on channel x. 0: No transmission over half on channel x. Hardware set, software write CHTIFx bit to clear this flag.	0
25/21/17/13/9/5/1	TCIFx	RO	Transmission completion flag for channel x (x=1/2/3/4/5/6/7). 1: a transmission completion event is generated on channel x. 0: No transmission completion event on channel x. Hardware set, software write CTCIFx bit to clear this flag.	0
24/20/16/12/8/4/0	GIFx	RO	Global interrupt flag for 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 occurred on channel x. Hardware set, software write CGIFx bit to clear this flag.	0

8.3.2 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 for channel x (x=1/2/3/4/5/6/7). 1: Clear the TEIFx flag in the DMA_INTFR register. 0: No effect.	0
26/22/18/14/10/6/2	CHTIFx	WO	Clear the transmission halfway flag for 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 for 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 for channel x (x=1/2/3/4/5/6/7). 1: Clear the TEIFx/HTIFx/TCIFx/ GIFx flags in the DMA_INTFR register. 0: No effect.	0

8.3.3 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	MEM2MEM	PL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN

Bit	Name	Access	Description	Reset value
[31:15]	Reserved	RO	Reserved	0
14	MEM2MEM	RW	Memory-to-memory mode enable. 1: Enable memory-to-memory data transfer mode. 0: Disable memory-to-memory data transfer mode.	0
[13:12]	PL	RW	Channel priority setting. 00: low; 01: medium. 10: High; 11:Very high.	0
[11:10]	MSIZE	RW	Memory address data width setting. 00: 8 bits; 01: 16 bits. 10: 32 bits; 11: Reserved.	0
[9:8]	PSIZE	RW	Peripheral address data width setting. 00: 8 bits; 01: 16 bits.	0

			10: 32 bits; 11: Reserved.	
7	MINC	RW	Memory address incremental incremental mode enable. 1: Enable incremental memory address increment operation. 0: Memory address remains unchanged operation.	0
6	PINC	RW	Peripheral address incremental incremental mode enable. 1: Enable incremental incremental operation of the peripheral address. 0: Peripheral address remains unchanged operation.	0
5	CIRC	RW	DMA channel cyclic mode enable. 1: Enables cyclic operation. 0: Perform a single operation.	0
4	DIR	RW	Data transfer direction. 1: Read from memory. 0: Read from peripheral.	0
3	TEIE	RW	Transmission error interrupt enable control. 1: Enable transmission error interrupt. 0: Disable transmission error interrupt.	0
2	HTIE	RW	Transmission over half interrupt enable control. 1: Enable the transmission over half interrupt. 0: Disable the transmission over half interrupt.	0
1	TCIE	RW	Transmission completion interrupt enable control. 1: Enable the transmission completion interrupt. 0: Disable the transmission completion interrupt.	0
0	EN	RW	Channel enable control. 1: Channel on; 0: Channel off. When a DMA transfer error occurs, the hardware automatically clears this bit to 0 and shuts down the channel.	0

8.3.4 DMA Channel x Number of Data Register (DMA_CNTRx) (x=1/2/3/4/5/6/7)

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

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

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:0]	NDT[15:0]	RW	Number of data transfers, range 0-65535. This register can only be written when the channel is not operating (EN=0 for DMA_CFGRx). After the channel is turned on this register becomes read-only and indicates the number of remaining pending transfer (the register content is decremented after each DMA transfer). When the channel is in 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 pending transfer. When the register content is 0, no data transmission will occur regardless of whether the channel is on or off.

8.3.5 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[31:0]	RW	Peripheral base address, which serves as the source or destination address for peripheral data transfer. When PSIZE[1:0]='01' (16 bits), the module automatically ignores bit0 and the operation address is automatically 2-byte aligned; when PSIZE[1:0]='10' (32 bits), the module automatically ignores bit[1:0] and the operation address is automatically 4-byte aligned.	0

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

8.3.6 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[31:0]	RW	The memory data address, which serves as the source or destination address for data transfers. When MSIZE[1:0]='01' (16 bits), the module automatically ignores bit0, and the operation address is automatically 2-byte aligned; when MSIZE[1:0]='10' (32 bits), the module automatically ignores bit[1:0], and the operation address is automatically 4-byte aligned.	0

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

Chapter 9 Analog-to-digital Converter (ADC)

The ADC module contains a 10-bit successive approximation type analog-to-digital converter with up to 24MHz input clock. It supports 8 external channels and 2 internal signal source sampling sources. Single conversion and continuous conversion of channels, automatic scan mode between channels, intermittent mode, external trigger mode, double sampling, trigger delay, etc. can be accomplished. The channel voltage can be monitored to see if it is within the threshold range by using the analog watchdog function.

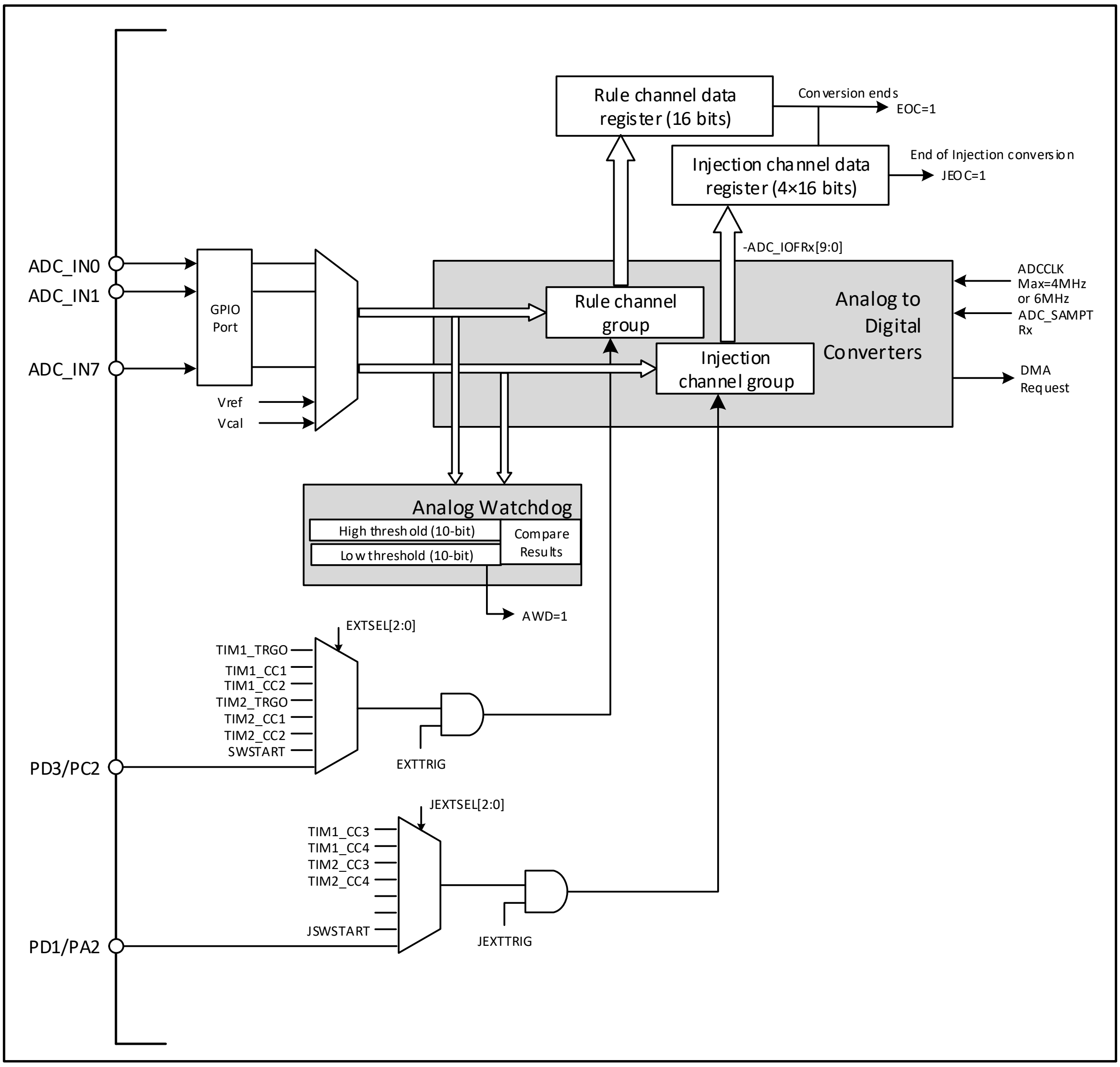
9.1 Main Features

- 10-bit resolution
- Supports 8 external channels and 2 internal signal sources for sampling
- Multiple sampling conversion methods for multiple channels: single, continuous, scan, trigger, intermittent, etc.
- Data alignment modes: left-aligned, right-aligned
- Sampling time can be programmed separately by channel
- Both rule conversion and injection conversion support external triggering
- Analog watchdog to monitor channel voltage, self-calibration function
- ADC channel input range: $0 \leqslant V_{IN} \leqslant V_{DDA}$
- Trigger delay

9.2 Functional Description

9.2.1 Module Structure

Figure 9-1 ADC module block diagram



9.2.2 ADC Configuration

1) Module power-up

An ADON bit of 1 in the ADC_CTLR2 register indicates that the ADC module is powered up. When the ADC module enters the power-up state (ADON=1) from the power-down mode (ADON=0), a delay period t_{STAB} is required for the module stabilization time. After that, the ADON bit is written to 1 again and is used as the start signal for software to start the ADC conversion. By clearing the ADON bit to 0, the current conversion can be terminated and the ADC module placed in power-down mode, a state in which the ADC consumes almost no power.

2) Sampling clock

The register operation of the module is based on the AHBCLK (AHB bus) clock, and the clock reference of its conversion unit, ADCCLK, is configured by the ADCPRE field of the RCC_CFGR0 register to divide the frequency. Refer to datasheet *CH32V003DS0* for detailed information.

3) Channel configuration

The ADC module provides 10 channel sampling sources, including 8 external channels and 2 internal channels. They can be configured into two types of conversion groups: regular groups and injection groups. to achieve a group conversion consisting of a series of conversions in any order on any number of channels.

Conversion group.

- Rule group: consists of up to 16 conversions. The rule channels and their conversion order are set in the ADC_RSQRx register. The total number of conversions in the rule group should be written to L[3:0] in the ADC_RSQR1 register.
- Injection group: consists of up to 4 conversions. The injection channels and the order of their conversions are set in the ADC_ISQR register. The total number of conversions in the injection group should be written in JL[1:0] of the ADC_ISQR register.

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

2 internal channels.

- Vref internal reference voltage: connected to ADC_IN8 channel.
- Vcal internal calibration voltage: connected to ADC_IN9 channel, 2 steps selectable.

4) Calibration

The ADC has a built-in self-calibration mode. A calibration session significantly reduces accuracy errors due to variations in the internal capacitor banks. During calibration, an error correction code is calculated on each capacitor, which is used to eliminate the errors generated on each capacitor in subsequent conversions.

Initialize the calibration register by writing RSTCAL position 1 of ADC_CTLR2 register and wait for RSTCAL hardware to clear 0 to indicate the completion of initialization. Set the CAL bit to start the calibration function. Once the calibration is finished, the hardware will automatically clear the CAL bit and store the calibration code into ADC_RDATAR. After that, the normal conversion function can be started. It is recommended to perform an ADC calibration when the ADC module is powered up.

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

5) Programmable sampling time

The ADC uses several ADCCLK cycles to sample the input voltage. The number of sampling cycles for a channel can be changed using the SMPx[2:0] bits in the ADC_SAMPTR1 and ADC_SAMPTR2 registers. Each channel can be sampled separately using a different time.

The total conversion time is calculated as follows.

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

The ADC's rule channel conversion supports the DMA function. The value of the rule channel conversion is stored in a data-only register, ADC_RDATAR. To prevent the data in ADC_RDATAR register from being fetched in time when multiple rule channels are converted in succession, the DMA function of ADC can be enabled. The hardware will generate a DMA request at the end of the conversion of a rule channel (EOC set) and transfer the converted data from the ADC_RDATAR register to the user-specified destination address.

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

Note: Injection group conversion does not support DMA function.

6) Data alignment

The ALIGN bit in the ADC_CTLR2 register selects the alignment of the ADC converted data storage. 10-bit data supports left-aligned and right-aligned modes.

The data register ADC_RDATAR of the rule group channel holds the actual converted 10-bit digital value; while the data register ADC_IDATARx of the injection group channel is the actual converted data minus the value written after the offset defined in the ADC_IOFRx register, there will be positive and negative cases, so there are sign bits (SIGNB).

Table 9-1 Data left alignment

Rule group data register

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

Inject group data register

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

Table 9-2 Data right alignment

Rule group data register

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

Inject group data register

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

9.2.3 External Trigger Source

The ADC conversion start event can be triggered by an external event. If the EXTTRIG or JEXTTRIG bits of the ADC_CTLR2 register are set, the conversion of a rule group or injection group channel can be triggered by an external event, respectively. In this case, the configuration of EXTSEL[2:0] and JEXTSEL[2:0] bits determines the external event source for the rule group and injection group.

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

Table 9-3 External trigger sources for rule group channels

EXTSEL[2:0]	Trigger source	Type
000	TRGO event of timer 1	Internal signal from on-chip timer
001	CC1 event of timer 1	
010	CC2 event of timer 1	
011	TRGO event of timer 2	
100	CC1 event of timer 2	
101	CC2 event of timer 2	
110	PD3/PC2 events	From external pins
111	SWSTART software trigger	Software control bits

Table 9-4 External trigger sources for injection group channels

JEXTSEL[2:0]	Trigger source	Type
000	CC3 event of timer 1	Internal signal from on-chip timer
001	CC4 event of timer 1	
010	CC3 event of timer 2	
011	CC4 event of timer 2	
100	-	
101	-	
110	PD1/PA2	From external pins
111	JSWSTART software trigger	Software control bits

9.2.4 Conversion Mode

Table 9-5 Conversion mode combinations

ADC_CTLR1 and ADC_CTLR2 register control bits					ADC conversion mode
CONT	SCAN	RDISCEN/IDISCEN	JAUTO	Start event	
0	0	0	0	ADON position 1	Single single-channel mode: A rule channel performs a single conversion.
				External trigger method	Single single-channel mode: A single conversion is performed on one of the rule channels or injection channels.

	1	0	0	ADON position 1 or external trigger method	Single scan mode: performs a single conversion of all selected rule group channels (ADC_RSQRx) or all injection group channels (ADC_ISQR) one by one in sequence. Trigger injection method: When the rule group channel conversion process can be inserted into the injection group channel all conversion, and then continue the rule group channel conversion afterwards; but the rule group channel conversion will not be inserted when converting the injection group channel.
			1	ADON position 1 or external trigger method	Single scan mode: performs a single conversion of all selected rule group channels (ADC_RSQRx) or all injection group channels (ADC_ISQR) one by one in sequence. Automatic injection method: After the rule group channel is converted, the injection group channel is automatically converted.
	0	1 (RDISCEN and IDISCEN cannot be 1 at the same time)	0	External trigger method	Single intermittent mode: Each time an event is started, a short sequence (DISCNUM[2:0] defined number) of channel number transitions is executed and cannot be restarted until all selected channel transitions are completed. *Note: The IDISCEN and RDISCEN control bits are selected for the rule group and injection group respectively, and the intermittent mode cannot be configured for the rule group and injection group at the same time.*
			1	-	Disable this mode.
	1	1	X	-	No such mode.
1	0	0	0	ADON position 1 or external trigger method	Continuous single channel/scan mode: repeat a new round of transitions at the end of each round until CONT clears 0 to terminate.
	1	0	0		
			1		

Note: The external trigger events for rule groups and injection groups are different, and the 'ACON' bit can only initiate rule group channel conversion, so the initiation events for rule group and injection group channel conversion are independent.

1) Single single-channel conversion mode

In this mode, only one conversion is executed for the current 1 channel. This mode performs conversion for the channel that is sorted 1st in the rule group or injection group, where it is initiated by setting ADON position 1 of the ADC_CTLR2 register (for rule channels only) or can be initiated by external trigger (for rule channels or injection channels). Once the conversion of the selected channel is completed it will.

If the conversion is for a rule group channel, the conversion data is stored in the 16-bit ADC_RDATAR register, the EOC flag is set, and an ADC interrupt is triggered if the EOCIE bit is set.

If the conversion is for an injection group channel, the conversion data is stored in the 16-bit ADC_IDATAR1 register, the EOC and JEOC flags are set, and an ADC interrupt is triggered if the JEOCIE or EOCIE bit is set.

2) Single scan mode conversion

The ADC scan mode is entered 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 injection channels) one by one, and the next channel in the same group is converted automatically when the current channel conversion is finished.

In the scan mode, there is a subdivision into triggered injection mode and automatic injection mode depending on the status of the JAUTO bit.

- Trigger injection

JAUTO bit is 0. When the trigger event of injection group channel conversion occurs during the scanning of rule group channels, the current conversion is reset and the sequence of injection channels is performed in a single scan, and the last interrupted rule group channel conversion is resumed after all selected injection group channel scanning conversions are completed.

If a rule channel start event occurs while the injection group channel sequence is currently being scanned, the injection group conversion is not interrupted, but the rule sequence conversion is executed again after the injection sequence conversion is completed.

Note: When using triggered injection conversions, you must ensure that the interval between triggered events is longer than the injection sequence. For example, if the overall time to complete the conversion of the injection sequence takes 28 ADCCLK, then the minimum value of the event interval to trigger the injection channel is 29 ADCCLK.

- Auto-injection

The JAUTO bit is set to 1, and conversion of the selected channel of the injection group is performed automatically after scanning all the channels selected by the rule group for conversion. This approach can be used to convert up to 20 conversion sequences in the ADC_RSQRx and ADC_ISQR registers.

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

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

3) Single intermittent mode conversion

The intermittent mode of the rule group or injection group is entered 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 perform a short sequence of scan transitions.

The length of the short sequence n (n<=8) is defined in DISCNUM[2:0] of ADC_CTLR1 register, when RDISCEN is 1, it is the interrupted mode of the rule group, and the total length to be converted is defined in L[3:0] of ADC_RSQR1 register; when IDISCEN is 1, it is the interrupted mode of the injection group, and the total length to be converted is defined in JL[1:0] of ADC_ISQR register. It is not possible to set both the rule group and the injection group to intermittent mode.

Example of rule group intermittent mode.

RDISCEN=1, DISCNUM[2:0]=3, L[3:0]=8, channels to be converted = 1, 3, 2, 5, 8, 4, 10, 6
The 1st external trigger: conversion sequence is: 1, 3, 2
The 2nd external trigger: conversion sequence is: 5, 8, 4
The 3rd external trigger: conversion sequence is: 10, 6, while generating EOC events
The 4th external trigger: conversion sequence is: 1, 3, 2

Examples of intermittent patterns injected into groups.

IDISCEN=1, DISCNUM[2:0]=1, JL[1:0]=3, channel to be converted=1, 3, 2
The 1st external trigger: conversion sequence is: 1
The 2nd external trigger: the conversion sequence is: 3
The 3rd external trigger: conversion sequence is: 2, generating both EOC and JEOC events
The 4th external trigger: conversion sequence is: 1

Note: 1. When converting a rule group or injection group in intermittent mode, the conversion sequence does not automatically start from the beginning when it ends. When all subgroups have been converted, the next trigger event starts the conversion of the first subgroup.

2. You cannot use auto-injection (JAUTO=1) and intermittent mode at the same time.

3. You cannot set intermittent mode for both rule groups and injection groups, and intermittent mode can only be used for a group of conversions.

4) Continuous conversion

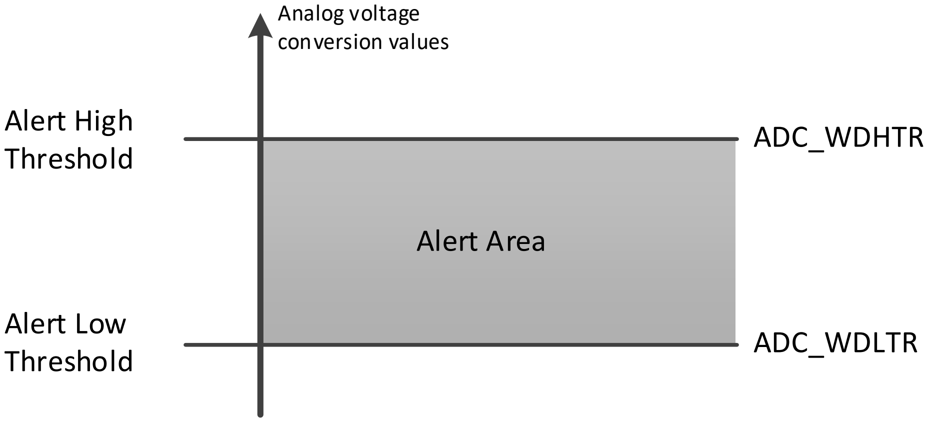
The continuous conversion mode of the ADC is entered by setting the CONT bit of the ADC_CTLR2 register to 1. This mode starts another conversion as soon as the previous ADC conversion is finished, and the conversion does not stop at the last channel of the selection group, but continues again from the first channel of the selection group.

Start events include external trigger events and ADON positions.1 Combined with several conversions in the previous single mode, they also include continuous single-channel conversions, and continuous scan mode (triggered injection or automatic injection) conversions.

9.2.5 Analog Watchdog

The AWD analog watchdog status bit is set if the analog voltage being converted by the ADC is below the low threshold or above the high threshold. The threshold settings are located in the lowest 10 valid bits of the ADC_WDHTR and ADC_WDLTR registers. The AWDIE bit of the ADC_CTLR1 register is set to allow the corresponding interrupt to be generated.

Figure 9-4 Analog watchdog threshold area



Configure the AWDSGL, AWDEN, JAWDEN and AWDCH[4:0] bits of the ADC_CTLR1 register to select the channel for analog watchdog alerting, as related in the following table.

Table 9-6 Analog Watchdog channel selection

Simulation of watchdog alert channel	ADC_CTLR1 register control bit			
	AWDSGL	AWDEN	JAWDEN	AWDCH[4:0]
No vigilance	Ignore	0	0	Ignore
All injection channels	0	0	1	Ignore
All rule channels	0	1	0	Ignore
All injection and rule channels	0	1	1	Ignore
Single injection channel	1	0	1	Determine the channel number
Single rule channel	1	1	0	Determine the channel number
Single injection and rule channel	1	1	1	Determine the channel number

9.3 Register Description

Table 9-7 ADC-related registers list

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 register 1	0x00000000
R32_ADC_SAMPTR2	0x40012410	ADC sample time register 2	0x00000000
R32_ADC_IOFR1	0x40012414	ADC injected channel data offset register 1	0x00000000
R32_ADC_IOFR2	0x40012418	ADC injected channel data offset register 2	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 sequence register 1	0x00000000
R32_ADC_RSQR2	0x40012430	ADC regular sequence register 2	0x00000000
R32_ADC_RSQR3	0x40012434	ADC regular sequence register 3	0x00000000
R32_ADC_ISQR	0x40012438	ADC injected 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
R32_ADC_DLYR	0x40012450	ADC delayed data register	0x00000000

9.3.1 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											STRT	JSTRT	JEOC	EOC	AWD

Bit	Name	Access	Description	Reset value
[31:5]	Reserved	RO	Reserved	0
4	STRT	RW0	Rule channel transition start state. 1: Rule channel conversion has started. 0: Rule channel conversion is not started. This bit is set to 1 by hardware and cleared to 0 by software (write 1 is not valid).	0
3	JSTRT	RW0	Injection channel conversion start state. 1: Injection channel conversion has started. 0: Injection channel conversion has not started. This bit is set to 1 by hardware and cleared to 0 by software (write 1 is not valid).	0
2	JEOC	RW0	Injection into the end state of the channel group conversion. 1: Conversion complete. 0: The conversion is not completed. This bit is set to 1 by hardware (all injected channels are converted) and cleared to 0 by software (write 1 is invalid).	0
1	EOC	RW0	Conversion end state. 1: Conversion complete.	0

			0: The conversion is not completed. This bit is set to 1 by hardware (end of rule or injection channel group conversion), cleared by software to 0 (write 1 is invalid) or when reading ADC_RDATAR.	
0	AWD	RW0	Analog watchdog flag bit. 1: Occurrence of simulated watchdog events. 0: No simulated watchdog event occurred. This bit is set to 1 by hardware (conversion value is out of range of ADC_WDHTR and ADC_WDLTR registers) and cleared to 0 by software (write 1 is not valid).	0

9.3.2 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					CALVOL [1:0]		Reserved	AWDEN	JAWDEN	Reserved					

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

Bit	Name	Access	Description	Reset value
[31:27]	Reserved	RO	Reserved	0
[26:25]	CALVOL[1:0]	RW	Calibration voltage selection 01: Calibration voltage 2/4 AVDD 10: Calibration voltage 3/4 AVDD Other: Invalid	01
24	Reserved	RO	Reserved	0
23	AWDEN	RW	Analog watchdog function enable bit on the rule channel. 1: Enable the analog watchdog on the rule channel. 0: Disable the analog watchdog on the rule channel.	0
22	JAWDEN	RW	Analog watchdog function enable bit on the injection channel. 1: Enable the analog watchdog on the injection channel. 0: Disable the analog watchdog on the injection channel.	0
[21:16]	Reserved	RO	Reserved	0
[15:13]	DISCNUM[2:0]	RW	Number of rule channels to be converted after external triggering in intermittent mode. 000: 1 channel. ... 111: 8 channels.	0
12	JDISCEN	RW	Inject the intermittent mode enable bit on the channel. 1: Enable intermittent mode on the injection channel. 0: Disable intermittent mode on the injection channel.	0
11	DISCEN	RW	Intermittent mode enable bit on rule channel. 1: Enables intermittent mode on the rule channel. 0: Disable intermittent mode on the rule channel.	0
10	JAUTO	RW	After the opening of the rule channel is completed, the injection channel group enable bit is automatically switched. 1: Enable automatic injection channel group switching. 0: Disable automatic injection channel group conversion. *Note: This mode requires disabling the external trigger function of the injection channel.*	0
9	AWDSGL	RW	In scan mode, use the analog watchdog enable bit on a	0

			single channel. 1: Use an analog watchdog on a single channel (AWDCH[4:0] selection). 0: Use analog watchdog on all channels.	
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	JEOCIE	RW	Inject the channel group end-of-conversion interrupt enable bit. 1: Enable the injection of the channel group conversion completion interrupt (IEOC JEOC flag). 0: Disable the injection channel group conversion completion interrupt.	0
6	AWDIE	RW	Analog watchdog interrupt enable bit. 1: Enable the analog watchdog interrupt. 0: Disable the analog watchdog interrupt. *NOTE: In scan mode, this interrupt will abort the scan if it occurs.*	0
5	EOCIE	RW	End of conversion (rule or injection channel group) interrupt enable bit. 1: Enable the end-of-conversion interrupt (EOC flag). 0: Disable the end-of-conversion interrupt.	0
[4:0]	AWDCH[4:0]	RW	Analog watchdog channel selection bits. 00000: Analog input channel 0. 00001: Analog input channel 1. ... 01111: Analog input channel 15.	0

9.3.3 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									SWSTART	JSWSTART	EXTTRIG	EXTSEL[2:0]			Reserved

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

Bit	Name	Access	Description	Reset value
[31:23]	Reserved	RO	Reserved	0
22	SWSTART	RW	To start a rule channel conversion, you need to set the software trigger to. 1: Initiate rule channel conversion. 0: Reset state. This bit is set by software and cleared to 0 by hardware when conversion starts.	0
21	JSWSTART	RW	To initiate an injection channel transition, set the software to trigger: 1: initiates an injection channel transition; 0: reset state. This bit is set by software and is cleared to 0 by hardware or 0 by software when the conversion starts.	0
20	EXTTRIG	RW	External trigger transition mode enable for the rule channel.	0

			1: Use of external events to initiate conversions. 0: Turn off the external event activation function.	
[19:17]	EXTSEL[2:0]	RW	External trigger event selection for initiating rule channel conversion. 000: TRGO event for timer 1. 001: CC1 event of timer 1. 010: CC2 event of timer 1. 011: TRGO event of timer 2. 100: CC1 event of timer 2. 101: CC2 event of timer 2. 110: PD3/PC2 events. 111: SWSTART software trigger.	0
16	Reserved	RO	Reserved	0
15	JEXTTRIG	RW	External trigger transition mode enable for the injected channel. 1: Use of external events to initiate conversions. 0: Turn off the external event activation function.	0
[14:12]	JEXTSEL[2:0]	RW	External trigger event selection for initiating injection channel conversion. 000: CC3 event of timer 1. 001: CC4 event of timer 1. 010: CC3 event of timer 2. 011: CC4 event of timer 2. 100. 101. 110: PD1/PA2 111: JSWSTART software trigger.	0
11	ALIGN	RW	Data alignment. 1: left-aligned; 0: right-aligned.	0
[10:9]	Reserved	RO	Reserved	0
8	DMA	RW	Direct Memory Access (DMA) mode enable. 1: Enable DMA mode. 0: Disable DMA mode.	0
[7:4]	Reserved	RO	Reserved	0
3	RSTCAL	RW	Reset calibration, this bit is set by software and cleared by hardware after the reset is completed. 1: Initialization of the calibration registers. 0: Calibration register is initialized. *Note: If RSTCAL is set while conversion is in progress, additional cycles are required to clear the calibration register.*	0
2	CAL	RW	A/D calibration, this bit is set by software and cleared to 0 by hardware at the end of calibration. 1: Start of calibration. 0: Calibration is complete.	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	On/off A/D converter 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: Turn on the ADC and start the conversion. 0: Turn off ADC conversion/calibration and enter power-down mode. *Note: A conversion is initiated when only ADON is changed in the register, and no new conversion is initiated*	0

			if there are any other bits sent for change.	

9.3.4 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														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
[31:18]	Reserved	RO	Reserved	0
[17:0]	SMPx[2:0]	RW	SMPx[2:0]: sample time configuration for channel x. 000: 3 cycles; 001: 9 cycles. 010: 15 cycles; 011: 30 cycles. 100: 43 cycles; 101:57 cycles. 110: 73 cycles; 111: 241 cycles. These bits are used to independently select the sample time for each channel, and the channel configuration value must remain constant during the sample cycle.	0

9.3.5 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[2:0]	RW	SMPx[2:0]: sample time configuration for channel x. 000: 3 cycles; 001: 9 cycles. 010: 15 cycles; 011: 30 cycles. 100: 43 cycles; 101:57 cycles. 110: 73 cycles; 111: 241 cycles. These bits are used to independently select the sample time for each channel, and the channel configuration value must remain constant during the sample cycle.	

9.3.6 ADC Injected Channel Data Offset Register x (ADC_IOFRx) (x=1/2/3/4)

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

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

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

Bit	Name	Access	Description	Reset value

[31:10]	Reserved	RO	Reserved	0
[9:0]	JOFFSETx[11:0]	RW	The data offset value of the injected channel x. When converting the injected channels, this value defines the value used to subtract from the original conversion data. The result of the conversion can be read out in the ADC_IDATARx register.	0

9.3.7 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[9:0]									

Bit	Name	Access	Description	Reset value
[31:10]	Reserved	RO	Reserved	0
[9:0]	HT[9:0]	RW	Analog watchdog high threshold setting value.	0

Note: You can change the values of WDHTR and WDLTR during the conversion process, but they will take effect at the next conversion.

9.3.8 ADC Watchdog Low Threshold Register (ADC_WDLTR)

Offset address: 0x28

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

Bit	Name	Access	Description	Reset value
[31:10]	Reserved	RO	Reserved	0
[9:0]	LT[9:0]	RW	Analog watchdog low threshold setting value.	0

Note: You can change the values of WDHTR and WDLTR during the conversion process, but they will take effect at the next conversion.

9.3.9 ADC Regular 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								L[3:0]				RSQ16[4:1]			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SQ16[0]	SQ15[4:0]					SQ14[4:0]					SQ13[4:0]				

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

9.3.10 ADC Regular Sequence Register 2(ADC_RSQR2)

Offset address: 0x30

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="2">Reserved</td><td colspan="5">SQ12[4:0]</td><td colspan="5">SQ11[4:0]</td><td colspan="4">SQ10[4:1]</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>SQ10[0]</td><td colspan="5">SQ9[4:0]</td><td colspan="5">SQ8[4:0]</td><td colspan="5">SQ7[4:0]</td></tr>
</table>

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

9.3.11 ADC Regular Sequence Register 3(ADC_RSQR3)

Offset address: 0x34

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

Bit	Name	Access	Description	Reset value
[31:30]	Reserved	RO	Reserved	0
[29:25]	SQ6[4:0]	RW	The number of the 6th conversion channel in the rule sequence (0-9).	0
[24:20]	SQ5[4:0]	RW	The number of the 5th conversion channel in the rule sequence (0-9).	0

[19:15]	SQ4[4:0]	RW	The number of the 4th conversion channel in the rule sequence (0-9).	0
[14:10]	SQ3[4:0]	RW	The number of the 3th conversion channel in the rule sequence (0-9).	0
[9:5]	SQ2[4:0]	RW	The number of the 2th conversion channel in the rule sequence (0-9).	0
[4:0]	SQ1[4:0]	RW	The number of the 1th conversion channel in the rule sequence (0-9).	0

9.3.12 ADC Injected Sequence Register (ADC_ISQR)

Offset address: 0x38

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										JL[1:0]		JSQ4[4:1]			

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

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

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

9.3.13 ADC Injected Data Register (ADC_IDATARx) (x=1/2/3/4)

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

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

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

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

9.3.14 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
DATA[31:16]															

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[15:0]	RO	Rule channel conversion data (data left-aligned or right-aligned)	0

9.3.15 ADC Delayed Data Register (ADC_DLYR)

Offset address: 0x50

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

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

Bit	Name	Access	Description	Reset value
[31:10]	Reserved	RO	Reserved	0
9	DLYSRC	RW	External trigger source delay selection 0: Rule channel external trigger delay 1: Injection channel external trigger delay	0
[8:0]	DLYVLU[8:0]	RW	External trigger delay data, delay time configuration, unit: ADC clock cycle	0

Chapter 10 Advanced-control Timer (ADTM)

The Advanced-control timer Module contains a powerful 16-bit auto-reload timer, TIM1, which can be used to measure pulse width or generate pulses, PWM waves, etc. It is used in motor control, power supply, etc.

10.1 Main Features

The main features of the advanced-control timer TIM1 include.

- 16-bit auto-reload counter supporting incremental counting mode, decremental counting mode and incremental and decremental counting mode.
- 16-bit prescaler with dynamically adjustable crossover coefficients from 1 to 65536.
- Support for four independent comparison capture channels.
- Each comparison capture channel supports multiple operating modes, such as: input capture, output comparison, PWM generation and single pulse output.
- Complementary outputs supporting programmable dead time.
- Support for external signals to control the timer.
- Support for updating the timer after a defined period using a repeat counter.
- Support for resetting the timer or placing it in the OK state using the brake signal.
- Support for the use of DMA in multiple modes.
- Support for incremental encoders.
- Support cascading and synchronization between timers.

10.2 Principle and Structure

This section deals with the internal construction of advanced-control timers.

10.2.1 Overview

As shown in Figure 10-1, the structure of the advanced-control timer can be roughly divided into three parts, namely the input clock part, the core counter part and the compare capture channel part.

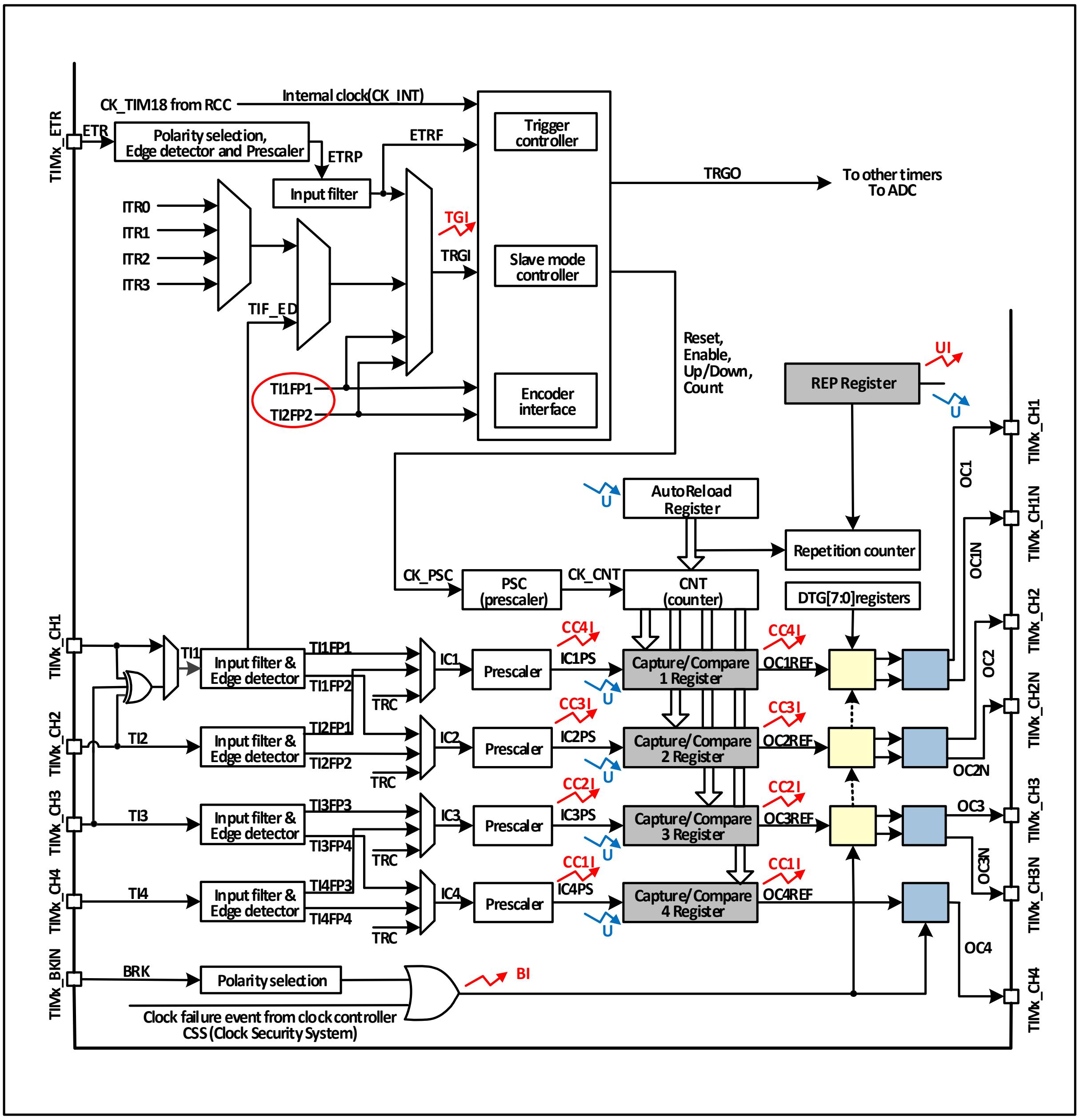
The advanced-control timer can be clocked from the AHB bus clock (CK_INT), from an external clock input pin (TIMx_ETR), from other timers with clock output (ITRx), or from the input of the compare capture channel (TIMx_CHx). These input clock signals become the CK_PSC clock after various set filtering and dividing operations and are output to the core counter section. In addition, these complex clock sources can also be output as TRGO to other peripherals such as timers and ADCs.

The core of the advanced-control timer is a 16-bit counter (CNT), and the CK_PSC is divided by a prescaler (PSC) to become the CK_CNT and output to the CNT. An auxiliary counter counts the number of times the ATRLR reloads the initial value for the CNT and generates a specific event when the count reaches the number set in the Repeat Count Register (RPTCR).

The advanced-control timer has four sets of compare capture channels, each of which can input pulses from exclusive pins or output waveforms to the pins, i.e., the compare capture channels support both input and output modes. The input of each channel of the compare capture register supports filtering, dividing and edge detection operations, and supports mutual triggering between channels, as well as providing a clock for the core counter

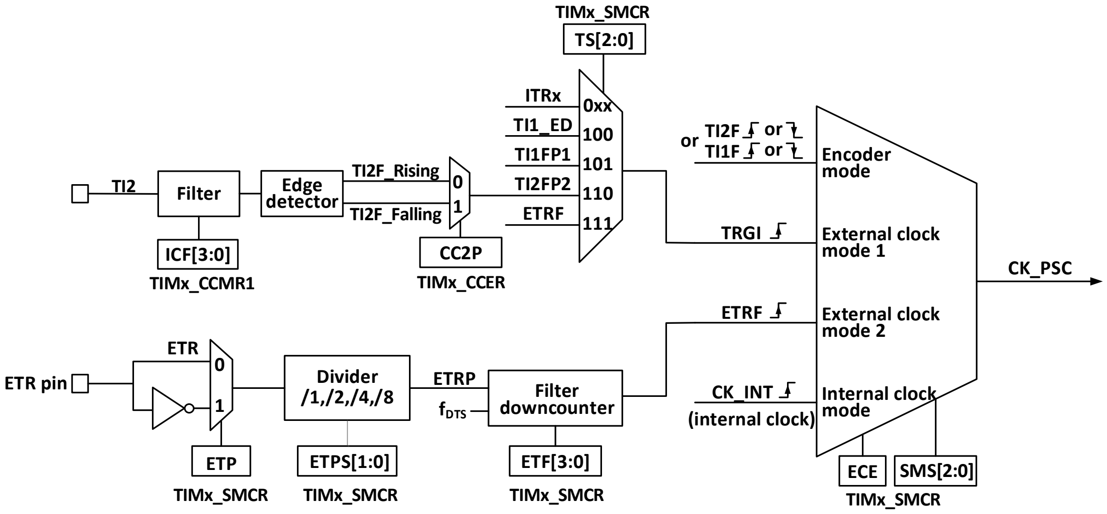
CNT. Each compare capture channel has a set of compare capture registers (CHxCVR) that support comparison with the main counter (CNT) to output pulses.

Figure 10-1 Block diagram of advanced-control timer structure



10.2.2 Clock Input

Figure 10-2 Block diagram of CK_PSC source for advanced-control timer



The advanced-control timer CK_PSC has many clock sources and can be divided into 4 categories.

1) The advanced-control timer CK_PSC has many clock sources and can be divided into 4 categories.
2) Internal AHB clock input route: CK_INT.
3) Route from the comparison capture channel pin (TIMx_CHx): TIMx_CHx → TIx → TIxFPx, this route is also used in encoder mode.
4) Inputs from other internal timers: ITRx.

The actual operation can be divided into 4 categories by determining the choice of input pulse for the SMS of the CK_PSC source.

1) Selection of the internal clock source (CK_INT).
2) External clock source mode 1.
3) External clock source mode 2.
4) Encoder mode.

All 4 clock source sources mentioned above can be selected by these 4 operations.

10.2.2.1 Internal clock source (CK_INT)

If the SMS field is held at 000b to start the advanced-control timer, then it is the internal clock source (CK_INT) that is selected as the clock. At this point CK_INT is CK_PSC.

10.2.2.2 External Clock Source Mode 1

When the SMS domain is set to 111b, external clock source mode 1 is enabled. When external clock source 1 is enabled, TRGI is selected as the source of CK_PSC. it is worth noting that the source of TRGI also needs to be selected by configuring the TS domain. the TS domain can select the following types of pulses as clock sources.

1) Internal trigger (ITRx, x is 0,1,2,3).
2) Comparison of the signal after capturing channel 1 through the edge detector (TI1F_ED).
3) Comparison of signals TI1FP1, TI2FP2 of the capture channel.
4) The signal ETRF from the external clock pin input.

10.2.2.3 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 position is set, the external clock source mode 2 is used. when using the external clock source mode 2, ETRF is selected as CK_PSC. the ETR pin becomes ETRP after passing through the optional inverter (ETP), divider (ETPS), and then ETRF after passing through the filter (ETF).
With the ECE position bit and the SMS set to 111b, this is equivalent to the TS selecting ETRF as an input.

10.2.2.4 Encoder Mode

Setting the SMS to 001b, 010b, 011b will enable the encoder mode. Enabling encoder mode allows you to select a specific level in TI1FP1 and TI2FP2 to signal the output with another jump edge as the signal. This mode is used when an external encoder is used. Refer to Section 10.3.9 for specific functions.

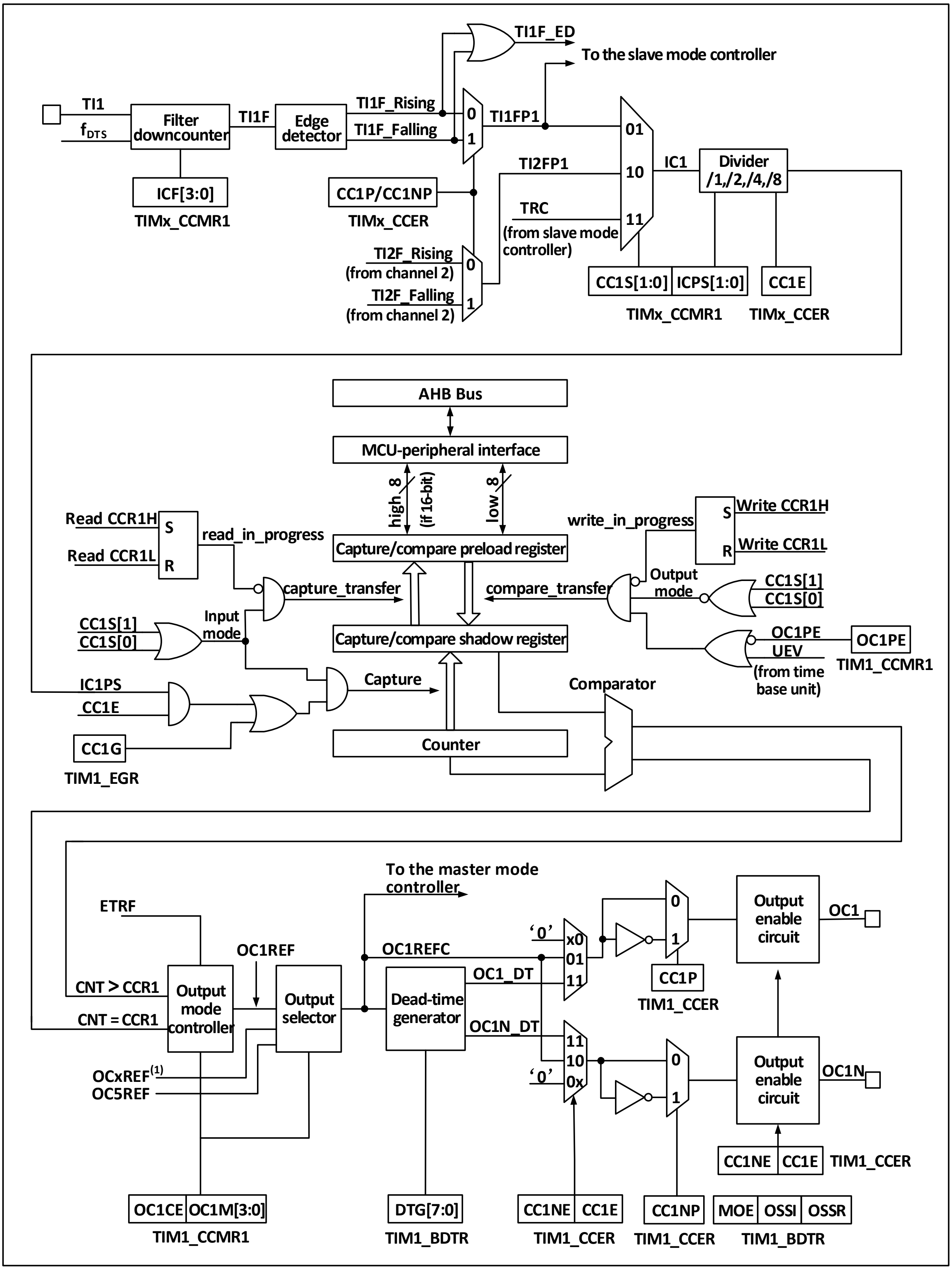
10.2.3 Counters and Peripherals

CK_PSC is input to the prescaler (PSC) for dividing. the PSC is 16-bit and the actual dividing factor is equal to the value of R16_TIMx_PSC + 1. CK_PSC goes through the PSC and becomes CK_INT. changing the value of R16_TIM1_PSC does not take effect in real time, but is updated to the PSC after an update event. the update event includes a UG bit clear and reset. The core of the timer is a 16-bit counter (CNT). CK_CNT is eventually fed to the CNT, which supports incremental count mode, decremental count mode, and incremental and decremental count modes, and has an Automatic Reload Register (ATRLR) that reloads the initial value for the CNT at the end of each count cycle. There is also an auxiliary counter that keeps track of the number of times the ATRLR reloads the initial value for the CNT and can generate a specific event when the number of times set in the Repeat Count Register (RPTCR) is reached.

10.2.4 Compare/capture Channels and Perimeters

The core of the timer is the compare/capture register, which is complemented by digital filtering, frequency division and inter-channel multiplexing in the peripheral input section, comparator and output control in the output section.

Figure 10-3 Block diagram of the structure of the comparison capture channel



The structure block diagram of the comparison capture channel is shown in Figure 10-3. The signal is input from the channel x pin and optionally made as TIx (the source of TI1 can be more than just CH1, see the structure block diagram of timer 10-1), TI1 is passed through the filter (ICF[3:0]) to generate TI1F, and then divided into TI1F_Rising and TI1F_Falling through the edge detector, these two signals are selected (CC1P) to generate TI1FP1, TI1FP1 and TI2FP1 from channel 2 are sent together to CC1S to select to become IC1, which is sent to the comparison capture register after ICPS dividing.

The compare capture register consists of a preload register and a shadow register, and the read/write process operates only on the preload register. In capture mode, the capture occurs on the shadow register and is then copied to the preload register; in compare mode, the contents of the preload register are copied to the shadow register, and then the contents of the shadow register are compared to the core counter (CNT).

10.3 Functionality and Implementation

The implementation of the complex functions of the advanced-control timer are all achieved by the operation of the timer's compare capture channel, clock input circuit and counter and peripheral parts. The clock input to the timer can come from multiple clock sources, including the input to the compare capture channel. The operation of the compare capture channel and clock source selection directly determines its function. The compare capture channel is bidirectional and can operate in both input and output modes.

10.3.1 Input Capture Mode

The input capture mode is one of the basic functions of the timer. The principle of input capture mode is that a capture event occurs when a determined edge on the ICxPS signal is detected, and the current value of the counter is latched into the compare capture register (R16_TIMx_CHCTLRx). When a capture event occurs, CCxIF (in R16_TIMx_INTFR) is set, and if an interrupt or DMA is enabled, the corresponding interrupt or DMA is also generated. if CCxIF is already set when a capture event occurs, the CCxOF bit is set. CCxIF can be cleared by software, or by hardware by reading the compare capture register. CCxOF is cleared by software.
An example of channel 1 to illustrate the steps to use the input capture mode is as follows.

1) Configure the CCxS domain to select the source of the ICx signal. For example, set to 10b and select TI1FP1 as the source of IC1 instead of using the default setting, where the CCxS domain defaults to making the comparison capture module the output channel.
2) Configure the ICxF domain to set the digital filter for the TI signal. The digital filter will sample the signal at a determined frequency, a determined number of times, and then output a hop. This sampling frequency and number of times is determined by ICxF.
3) Configure the CCxP bit to set the polarity of the TIxFPx. For example, keeping the CC1P bit low and selecting rising edge jumps.
4) Configure the ICxPS domain to set the ICx signal to be the crossover factor between ICxPS. For example, keeping ICxPS at 00b, without crossover.
5) Configure the CCxE bit to allow capturing the value of the core counter (CNT) into the compare capture register. Set the CC1E bit.
6) Configure the CCxIE and CCxDE bits as needed to determine whether to allow enable interrupts or DMA.

This completes the comparison capture channel configuration.
When a captured pulse is input to TI1, the value of the core counter (CNT) is recorded in the compare capture register, CC1IF is set, and the CCIOF bit is set when CC1IF has been set before. If the CC1IE bit is set, then an interrupt is generated; if CC1DE is set, a DMA request is generated. An input capture event can be generated by software by writing the event generation register (TIMx_SWEVGR).

10.3.2 Compare Output Modes

The compare output mode is one of the basic functions of the timer. The principle of the compare output mode is to output a specific change or waveform when the value of the core counter (CNT) agrees with the value of the compare capture register. the OCxM field (in R16_TIMx_CHCTLRx) and the CCxP bit (in R16_TIMx_CCER) determine whether the output is a definite high or low level or a level flip. The CCxIF bit is also set when a compare coherent event is generated. If the CCxIE bit is pre-set, an interrupt will be generated; if the CCxDE bit is pre-set, a DMA request will be generated.
To configure to compare output modes, proceed as follows.

1) Configuring the clock source and auto-reload value of the core counter (CNT).
2) Setting the count value to be compared to the comparison capture register (R16_TIMx_CHxCVR).
3) If an interrupt needs to be generated, set the CCxIE bit.

4) Keeping OCxPE at 0 to disable the preload register of the compare register.
5) Setting the output mode, setting the OCxM field and the CCxP bit.
6) Enable the output, setting the CCxE bit.
7) Set the CEN bit to start the timer.

10.3.3 Forced Output Mode

The output pattern of the timer's compare capture channel can be forced by software to output a determined level without relying on comparison of the compare capture register's shadow register with the core counter.
This is done by setting OCxM to 100b, which forces OCxREF to low, or by setting OCxM to 101b, which forces OCxREF to high.
Note that by forcing OCxM to 100b or 101b, the comparison process of the internal core counters and compare capture registers is still going on, the corresponding flags are still set, and interrupts and DMA requests are still being generated.

10.3.4 PWM Input Mode

The PWM input mode is used to measure the duty cycle and frequency of PWM and is a special case of the input capture mode. The operation is the same as 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 as trigger input and SMS is set to reset mode.
For example, to measure the period and frequency of the PWM wave input from TI1, the following operations are required.

1) Set TI1 (TI1FP1) to be the input of IC1 signal. Set CC1S to 01b.
2) Set TI1FP1 to rising edge active. Holding CC1P at 0.
3) Set TI1 (TI1FP2) as the input of IC2 signal. Set CC2S to 10b.
4) Select TI1FP2 to set to falling edge active. Set CC2P to 1.
5) Select TI1FP1 as the source of the clock source. set TS to 101b.
6) Set the SMS to reset mode, i.e. 100b.
7) Enables input capture. cc1e and cc2e are set.

Thus the value of compare capture register 1 is the period of the PWM, and the value of compare capture register 2 is its duty cycle.

10.3.5 PWM Output Mode

PWM output mode is one of the basic functions of the timer. PWM output mode is most commonly used to determine the PWM frequency using the reload value and the duty cycle using the capture comparison register. Set 110b or 111b in the OCxM field to use PWM mode 1 or mode 2, set the OCxPE bit to enable the preload register, and finally set the ARPE bit to enable automatic reload of the preload register. Since the value of the preload register can only be sent to the shadow register when an update event occurs, the UG bit needs to be set to initialize all registers before the core counter starts counting. In PWM mode, the core counter and the compare capture register are always comparing, and depending on the CMS bit, the timer is able to output edge-aligned or center-aligned PWM signals.

- Edge alignment

When edge alignment is used, the core counter is incremented or decremented, and in the PWM mode 1 scenario, OCxREF is high when the core counter value is greater than the compare capture register, and low when the core counter value is less than the compare capture register (e.g., when the core counter grows to the value of R16_TIMx_ATRLR and reverts to all zeros).

- Central alignment

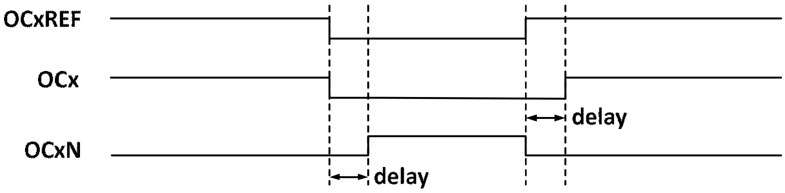
When using the central alignment modes, the core counter runs in alternating incremental and decremental count modes, and OCxREF makes rising and falling jumps when the values of the core counter and the compare capture register match. However, the comparison flags are set at different times in the three central alignment modes. When using the central alignment modes, it is best to generate a software update flag (set the UG bit) before starting the core counter.

10.3.6 Complementary Outputs and Dead Zones

The comparison capture channel generally has two output pins (comparison capture channel 4 has only one output pin) and can output two complementary signals (OCx and OCxN). OCx and OCxN can be independently set for polarity via the CCxP and CCxNP bits, independently set for output enable via CCxE and CCxNE, and independently set for output enable via the MOE, OIS, OISN, OSSI, and OSSR bits for deadband and other controls. Enabling the OCx and OCxN outputs simultaneously will insert a deadband, and each channel has a 10-bit deadband generator. OCx and OCxN are generated by the OCxREF association. If both OCx and OCxN are high active, then OCx is the same as OCxREF except that the rising edge of OCx is equivalent to OCxREF with a delay, and OCxN is the opposite of OCxREF in that its rising edge will have a delay relative to the falling edge of the reference signal. If the delay is greater than the effective output width, the corresponding pulse will not be generated.

The relationship between OCx and OCxN and OCxREF is illustrated in Figure 10-4, which shows the dead zone.

Figure 10-4 Complementary outputs and deadband



10.3.7 Brake Signal

When the brake signal is generated, the output enable signal and invalid level are modified according to the MOE, OIS, OISN, OSSI, and OSSR bits. However, OCx and OCxN will not be at the active level at any time. The source of the brake event can come from the brake input pin or it can be a clock failure event which is generated by the CSS (Clock Safety System).

After system reset, the brake function is disabled by default (MOE bit is low), and setting the BKE bit enables the brake function. The polarity of the input brake signal can be set by setting BKP, and the BKE and BKP signals can be written at the same time, and there is a delay of one AHB clock before the actual writing, so you need to wait for one AHB cycle to read the written value correctly.

At the presence of the selected level on the brake pin the system will generate the following actions.

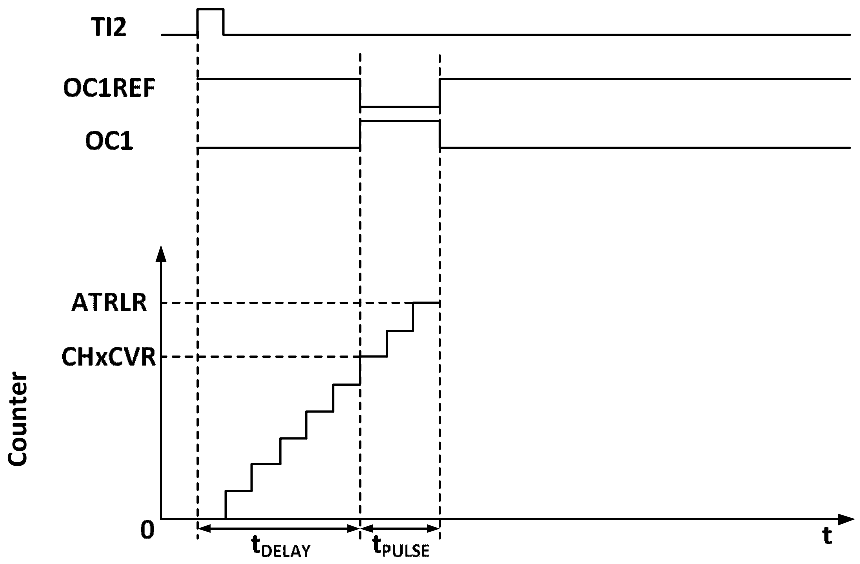
1) The MOE bit is cleared asynchronously, setting the output to an invalid, idle or reset state, depending on the setting of the SOOI bit.
2) After the MOE has been cleared, each output channel outputs a level determined by OISx.
3) When using complementary outputs: the outputs are placed in a null state, depending on the polarity.
4) If the BIE is set, an interrupt is generated when the BIF is set; if the BDE bit is set, a DMA request is generated.
5) If the AOE is set, the MOE bit is automatically set at the next update event UEV.

10.3.8 Single Pulse Mode

Single pulse mode can be used to allow the microcontroller to respond to a specific event by causing it to generate a pulse after a delay, with the delay and width of the pulse programmable. Placing the OPM bit allows the core counter to stop when the next update event UEV is generated (counter flips to 0).

As shown in Figure 10-5, a positive pulse of length Tpulse needs to be generated on OC1 after a delay Tdelay at the beginning of a rising edge detected on the TI2 input pin.

Figure 10-5 Generation of single pulse



1) Set TI2 to trigger. Setting the CC2S field to 01b to map TI2FP2 to TI2; setting the CC2P bit to 0b to set TI2FP2 as rising edge detection; setting the TS field to 110b to set TI2FP2 as trigger source; setting the SMS field to 110b to set TI2FP2 to be 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 Compare Capture Register.

10.3.9 Encoder Mode

The encoder mode is a typical application of the timer and can be used to access the biphasic output of the encoder. The counting direction of the core counter is synchronized with the direction of the encoder's rotation axis, and each pulse output from the encoder will cause the core counter to add or subtract one. To use the encoder, set the SMS field to 001b (count only on TI2 edge), 010b (count only on TI1 edge) or 011b (count on both TI1 and TI2 edges), connect the encoder to the input of comparison capture channels 1 and 2, and set a value for the reload value register, which can be set to a larger value. When in encoder mode, the internal compare capture register, prescaler, repeat count register, etc. of the timer are working normally. The following table shows the relationship between the counting direction and the encoder signal.

Table 10-1 Relationship between counting direction and encoder signal of timer encoder mode

Counting effective edges	The level of relative signals	TI1FP1 signal edge		TI2FP2 signal	
		Rising edge	Falling edge	Rising edge	Falling edge
Counting at TI1 edge only	high	Downward counting	Upward counting	No count	
	low	Upward counting	Downward counting		
Counting at TI2 edge only	high	No count		Upward counting	Downward counting
	low			Downward counting	Upward counting
Double edge counting at TI1 and TI2	high	Downward counting	Upward counting	Upward counting	Downward counting
	low	Upward counting	Downward counting	Downward counting	Upward counting

10.3.10 Timer Synchronization Mode

Timers are capable of outputting clock pulses (TRGO) and receiving inputs from other timers (ITRx). The source of ITRx (TRGO from other timers) is different for different timers. The timer internal trigger connections are shown in Table 10-2.

Table 10-2 TIMx internal trigger connections

Slave timer	ITR0(TS=000)	ITR1(TS=001)	ITR2(TS=010)	ITR3(TS=011)
TIM1		TIM2		
TIM2	TIM1			

10.3.11 Debug Mode

When the system enters debug mode, the timer continues to run or stops according to the settings of the DBG module.

10.4 Register Description

Table 10-3 TIM1-related registers list

Name	Access address	Description	Reset value
R16_TIM1_CTLR1	0x40012C00	Control register 1	0x0000
R16_TIM1_CTLR2	0x40012C04	Control register 2	0x0000
R16_TIM1_SMCFGR	0x40012C08	Slave mode control register	0x0000
R16_TIM1_DMAINTENR	0x40012C0C	DMA/interrupt enable register	0x0000
R16_TIM1_INTFR	0x40012C10	Interrupt status register	0x0000
R16_TIM1_SWEVGR	0x40012C14	Event generation register	0x0000
R16_TIM1_CHCTLR1	0x40012C18	Compare/capture control register 1	0x0000
R16_TIM1_CHCTLR2	0x40012C1C	Compare/capture control register 2	0x0000
R16_TIM1_CCER	0x40012C20	Compare/capture enable register	0x0000
R16_TIM1_CNT	0x40012C24	Counters	0x0000
R16_TIM1_PSC	0x40012C28	Counting clock prescaler	0x0000
R16_TIM1_ATRLR	0x40012C2C	Auto-reload value register	0x0000
R16_TIM1_RPTCR	0x40012C30	Recurring count value register	0x0000
R16_TIM1_CH1CVR	0x40012C34	Compare/capture register 1	0x0000
R16_TIM1_CH2CVR	0x40012C38	Compare/capture register 2	0x0000
R16_TIM1_CH3CVR	0x40012C3C	Compare/capture register 3	0x0000
R16_TIM1_CH4CVR	0x40012C40	Compare/capture register 4	0x0000
R16_TIM1_BDTR	0x40012C44	Brake and deadband registers	0x0000
R16_TIM1_DMACFGR	0x40012C48	DMA control register	0x0000
R16_TIM1_DMAADR	0x40012C4C	DMA address register for continuous mode	0x0000

10.4.1 Control Register 1 (TIM1_CTLR1)

Offset address: 0x00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPLVL	CAPOV	Reserved				CKD[1:0]		ARPE	CMS[1;0]		DIR	OPM	URS	UDIS	CEN

Bit	Name	Access	Description	Reset value
15	CAPLVL	RW	In double-edge capture mode, the capture level indication is enabled. 0: Disable the indication function 1: Enable the indication function. *Note: When enabled, [16] of CHxCVR indicates the level corresponding to the capture value.*	0
14	CAPOV	RW	Capture value mode configuration. 0: The capture value is the value of the actual counter	0

			1: The CHxCVR value is 0xFFFF when a counter overflow is generated before capture.	
[13:10]	Reserved	RO	Reserved	0
[9:8]	CKD[1:0]	RW	These 2 bits define the division ratio between the timer clock (CK_INT) frequency, the dead time and the sampling clock used by the dead time generator and the digital filter (ETR,TIx). 00: Tdts=Tck_int 01: Tdts = 2 x Tck_int 10: Tdts = 4 x Tck_int 11: Reserved.	0
7	ARPE	RW	Auto-reload preload enable bit. 1: Enables the Automatic Reload Value Register (ATRLR). 0: Auto Reload Value Register (ATRLR) is disabled.	0
[6:5]	CMS[1:0]	RW	Central alignment mode selection. 00: Edge-aligned mode. The counter counts up or down based on the direction bit (DIR). 01: Central alignment mode 1. The counter counts up and down alternately. The output compare interrupt flag bit of the channel configured as output (CCxS=00 in the CHCTLRx register) is set only when the counter counts down. 10: Central alignment mode 2. The counter counts up and down alternately. The output compare interrupt flag bit of the channel configured as output (CCxS=00 in the CHCTLRx register) is set only when the counter counts up. 11: Central alignment mode 3. The counter counts up and down alternately. The output compare interrupt flag bit of the channel configured as output (CCxS=00 in the CHCTLRx register) is set when the counter counts both up and down. *Note: When the counter is enabled (CEN=1), the transition from edge-aligned mode to center-aligned mode is not allowed.*	0
4	DIR	RW	Counting direction. 0: the counter's counting mode is incremental. 1: The counting mode of the counter is decimal counting. *Note: This bit is not valid when the counter is configured in central alignment mode or encoder mode.*	0
3	OPM	RW	Single pulse mode. 1: The counter stops when the next update event (clearing the CEN bit) occurs. 0: The counter does not stop when the next update event occurs.	0
2	URS	RW	Update request source, by which the software selects the source of the UEV event. 1: If an update interrupt or DMA request is enabled, only an update interrupt or DMA request is generated if the counter overflows/underflows. 0: If an update interrupt or DMA request is enabled, an update interrupt or DMA request is generated by any of the following events. -Counter overflow/underflow -Setting the UG position - Updates generated by the slave mode controller	0
1	UDIS	RW	Disable updates, the software allows/disables the	0

			generation of UEV events by means of this bit. 1: UEV is disabled. no update event is generated and the registers (ARR, PSC, CCRx) keep their values. If the UG bit is set or a hardware reset is issued from the mode controller, the counters and prescaler are reinitialized. 0: UEV is allowed. update (UEV) events are generated by any of the following events: -Counter overflow/underflow -Setting the UG position - Updates generated by the slave mode controller Registers with caches are loaded with their preloaded values.	
0	CEN	RW	Enables the counter. 1: Enable the counter. 0: Disable the counter. *Note: The external clock, gated mode and encoder mode will not work until the CEN bit is set in software. Trigger mode can automatically set the CEN bit in hardware.*	0

10.4.2 Control Register 2 (TIM1_CTLR2)

Offset address: 0x04

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	MMS[2:0]			CCDS	CCUS	Reserved	CCPC

Bit	Name	Access	Description	Reset value
15	Reserved	RO	Reserved	0
14	OIS4	RW	Output idle state 4. 1: When MOE=0, if OC4N is implemented, OC4=1 after deadband; 0: When MOE=0, if OC4N is implemented, OC4=0 after deadband. *Note: This bit cannot be modified after LOCK (TIMx_BDTR register) level 1, 2 or 3 has been set.*	0
13	OIS3N	RW	Output idle state 3. 1: when MOE = 0, OC3N = 1 after dead zone. 0: When MOE = 0, OC3N = 0 after dead zone. *Note: This bit cannot be modified after the LOCK (TIMx_BDTR register) level 1, 2 or 3 has been set.*	0
12	OIS3	RW	Output idle state 3, see OIS4.	0
11	OIS2N	RW	Output idle state 2, see OIS3N.	0
10	OIS2	RW	Output idle state 2, see OIS4.	0
9	OIS1N	RW	Output idle state 1, see OIS3N.	0
8	OIS1	RW	Output idle state 1, see OIS4.	0
7	TI1S	RW	TI1 selection. 1: TIMx_CH1, TIMx_CH2 and TIMx_CH3 pins connected to TI1 input after heterodyning. 0: TIMx_CH1 pin is connected directly to TI1 input.	0
[6:4]	MMS[2:0]	RW	Master mode selection: These 3 bits are used to select the synchronization information (TRGO) sent to the slave timer in master mode. The possible combinations are as follows. 000: The UG bit of the Reset-TIMx_EGR register is used as the trigger output (TRGO). In the case of a reset generated by a trigger input (from a mode controller in	0

			reset mode), there is a delay in the signal on TRGO relative to the actual reset. 001: Enable - The counter enable signal CNT_EN is used as a trigger output (TRGO). Sometimes it is necessary to start multiple timers at the same time or to control the enable from timers over a period of time. The counter enable signal is generated by the logical or of the trigger input signal in CEN control bit and gated mode. When the counter enable signal is controlled by a trigger input, there is a delay on TRGO unless master/slave mode is selected (see the description of the MSM bit in the TIMx_SMCR register). 010: Update - The update event is selected as a trigger input (TRGO). For example, the clock of a master timer may be used as a prescaler for a slave timer. 011: comparison pulse - on the occurrence of a capture or a successful comparison, when the CC1IF flag is to be set (even if it is already high), the trigger output sends a positive pulse (TRGO). 100: The comparison-OC1REF signal is used as a trigger output (TRGO). 101: The comparison-OC2REF signal is used as a trigger output (TRGO). 110: The comparison-OC3REF signal is used as a trigger output (TRGO). 111: The compare-OC4REF signal is used as the trigger output (TRGO).	
3	CCDS	RW	Capture the DMA selection for comparison. 1: Sending a DMA request for CHxCVR when an update event occurs. 0: Generate a DMA request for CHxCVR when CHxCVR occurs.	0
2	CCUS	RW	Compare capture control update selection bits. 1: if CCPC is set, they can be updated by setting the COM bit or a rising edge on TRGI. 0: If the CCPC is set, they can only be updated by setting the COM bit. *Note: This bit only works for channels with complementary outputs.*	0
1	Reserved	RO	Reserved	0
0	CCPC	RW	Compare capture preload control bits. 1: the CCxE, CCxNE and OCxM bits are preloaded and when this bit is set they are only updated when the COM bit is set. 0: CCxE, CCxNE and OCxM bits are not preloaded. *Note: This bit only works for channels with complementary outputs.*	0

10.4.3 Slave Mode Control Register (TIM1_SMCFGR)

Offset address: 0x08

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

Bit	Name	Access	Description	Reset value
15	ETP	RO	ETR trigger polarity selection, this bit selects whether to input ETR directly or to input the inverse of ETR.	0

			1: Invert ETR, low or falling edge active; 0: ETR, active high or rising edge.	
14	ECE	RW	External clock mode 2 enable selection. 1: Enables external clock mode 2. 0: Disable external clock mode 2. *Note 1: Slave mode can be used simultaneously with external clock mode 2: reset mode, gated mode and trigger mode; however, TRGI cannot be connected to ETRF in this case (TS bit cannot be '111').* *Note 2: When both external clock mode 1 and external clock mode 2 are enabled, the external clock input is ETRF.*	0
[13:12]	ETPS[1:0]	RW	The external trigger signal (ETRP) divides the frequency of this signal, which cannot exceed a maximum of 1/4 of the TIMxCLK frequency, and can be downconverted through this domain. 00: Prescaler off. 01: ETRP frequency divided by 2. 10: ETRP frequency divided by 4. 11: ETRP frequency divided by 8.	0
[11:8]	ETF[3:0]	RW	Externally triggered filtering, in fact, the digital filter is an event counter, which uses a certain sampling frequency to record up to N events and then produces a jump in the output. 0001: sampling frequency Fsampling=Fck_int, N=2. 0010: sampling frequency Fsampling=Fck_int, N=4. 0011: Sampling frequency Fsampling=Fck_int, N=8. 0100: sampling frequency Fsampling = Fdts/2, N = 6. 0101: sampling frequency Fsampling = Fdts/2, N = 8. 0110: sampling frequency Fsampling = Fdts/4, N = 6. 0111: sampling frequency Fsampling = Fdts/4, N = 8. 1000: sampling frequency Fsampling = Fdts/8, N = 6. 1001: sampling frequency Fsampling = Fdts/8, N = 8. 1010: sampling frequency Fsampling = Fdts/16, N = 5. 1011: sampling frequency Fsampling = Fdts/16, N = 6. 1100: sampling frequency Fsampling = Fdts/16, N = 8. 1101: sampling frequency Fsampling = Fdts/32, N = 5. 1110: sampling frequency Fsampling = Fdts/32, N = 6. 1111: Sampling frequency Fsampling=Fdts/32, N=8.	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 useful when the synchronization of several timers to a single external event is required. 0: Does not function.	0
[6:4]	TS[2:0]	RW	Trigger selection field, these 3 bits select the trigger input source used to synchronize the counter. 000: Internal trigger 0 (ITR0). 001: Internal trigger 1 (ITR1). 010: Internal trigger 2 (ITR2). 011: Internal trigger 3 (ITR3). 100: Edge detector of TI1 (TI1F_ED). 101: Filtered timer input 1 (TI1FP1). 110: Filtered timer input 2 (TI2FP2). 111: External trigger input (ETRF). The above only changes when SMS is 0. *Note: See Table 10-2 for details.*	0
3	Reserved	RO	Reserved	0

[2:0]	SMS[2:0]	RW	Input mode selection field. Selects the clock and trigger mode of the core counter. 000: driven by the internal clock CK_INT. 001: encoder mode 1, where the core counter increments or decrements the count at the edge of TI2FP2 depending on the level of TI1FP1. 010: encoder mode 2, where the core counter increments or decrements the count at the edge of TI1FP1, depending on the level of TI2FP2. 011: encoder mode 3, where the core counter increments and decrements the count on the edges of TI1FP1 and TI2FP2 depending on the input level of another signal; 100: reset mode, where the rising edge of the trigger input (TRGI) will initialize the counter and generate a signal to update the registers. 101: Gated mode, when the trigger input (TRGI) is high, the counter clock is turned on; at the trigger input becomes low, the counter is stopped, and the counter starts and stops are controlled. 110: trigger mode, where the counter is started on the rising edge of the trigger input TRGI and only the start of the counter is controlled. 111: External clock mode 1, rising edge of the selected trigger input (TRGI) drives the counter.	0

10.4.4 DMA/Interrupt Enable Register (TIM1_DMAINTENR)

Offset address: 0x0C

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

Bit	Name	Access	Description	Reset value
15	Reserved	RO	Reserved	0
14	TDE	RW	Trigger the DMA request enable bit. 1: Allow DMA requests to be triggered. 0: Triggering of DMA requests is disabled.	0
13	COMDE	RW	DMA request enable bit of COM. 1: Allow DMA requests for COM. 0: DMA request for COM is disabled.	0
12	CC4DE	RW	Compare the DMA request enable bit of capture channel 4. 1: Allow comparison of DMA requests for capture channel 4. 0: Disable comparison of DMA requests for capture channel 4.	0b
11	CC3DE	RW	Compare the DMA request enable bit of capture channel 3. 1: Allow comparison of DMA requests for capture channel 3. 0: Disable comparison of DMA requests for capture channel 3.	0
10	CC2DE	RW	Compare the DMA request enable bit of capture channel 2. 1: Allow comparison of DMA requests for capture channel 2. 0: Disable comparison of DMA requests for capture	0

			channel 2.	
9	CC1DE	RW	Compare the DMA request enable bit of capture channel 1. 1: Allow comparison of DMA requests for capture channel 1. 0: Disable comparison of DMA requests for capture channel 1.	0
8	UDE	RW	Updated DMA request enable bit. 1: DMA requests that allow updates. 0: DMA requests for updates are disabled.	0b
7	BIE	RW	Brake interrupt enable bit. 1: Allow brakes to be interrupted. 0: Brake interruption is prohibited.	0
6	TIE	RW	Trigger the interrupt enable bit. 1: Enable triggering of interrupts. 0: Trigger interrupt is disabled.	0
5	COMIE	RW	COM interrupt allow bit. 1: Allow COM interrupts. 0: COM interrupt is disabled.	0
4	CC4IE	RW	Compare capture channel 4 interrupt enable bit. 1: Allow comparison of capture channel 4 interrupts. 0: Disable compare capture channel 4 interrupt.	0
3	CC3IE	RW	Compare capture channel 3 interrupt enable bit. 1: Allow comparison of capture channel 3 interrupts. 0: Disable compare capture channel 3 interrupt.	0
2	CC2IE	RW	Compare capture channel 2 interrupt enable bit. 1: Allow comparison of capture channel 2 interrupts. 0: Disable compare capture channel 2 interrupt.	0
1	CC1IE	RW	Compare capture channel 1 interrupt enable bit. 1: Allow comparison of capture channel 1 interrupts. 0: Disable compare capture channel 1 interrupt.	0
0	UIE	RW	Update the interrupt enable bit. 1: Allow updates to be interrupted. 0: Disable update interruption.	0

10.4.5 Interrupt Status Register (TIM1_INTFR)

Offset address: 0x10

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

Bit	Name	Access	Description	Reset value
[15:13]	Reserved	RO	Reserved	0
12	CC4OF	RW0	Compare capture channel 4 to repeat capture flag bits.	0
11	CC3OF	RW0	Compare capture channel 3 to repeat capture flag bits.	0
10	CC2OF	RW0	Compare capture channel 2 to repeat capture flag bits.	0
9	CC1OF	RW0	The compare capture channel 1 repeat capture flag bit is used only when the compare capture channel is configured for input capture mode. This flag is set by hardware and a software write of 0 clears this bit. 1: the value of the counter is captured into the capture comparison register when the status of CC1IF has been set. 0: No duplicate captures are generated.	0b
8	Reserved	RO	Reserved	0
7	BIF	RW0	The brake interrupt flag bit, once the brake input is	0

			valid, by hardware for this position bit, can be cleared by software. 1: A set valid level is detected on the brake pin input. 0: No braking event is generated.	
6	TIF	RW0	Trigger interrupt flag bit, when a trigger event occurs by hardware to this location bit, by software to clear. Trigger events include the detection of a valid edge at the TRGI input from a mode other than gated, or any edge in gated mode. 1: Trigger event generation. 0: No trigger event is generated.	0
5	COMIF	RW0	COM interrupt flag bit, this bit is set by hardware and cleared by software once a COM event is generated. com events including CCxE, CCxNE, OCxM are updated. 1: COM event generation. 0: No COM event is generated.	0
4	CC4IF	RW0	Compare capture channel 4 interrupt flag bits.	0
3	CC3IF	RW0	Compare capture channel 3 interrupt flag bits.	0
2	CC2IF	RW0	Compare capture channel 2 interrupt flag bits.	0
1	CC1IF	RW0	Compare capture channel 1 interrupt flag bits. If the compare capture channel is configured in output mode. This bit is set by hardware when the counter value matches the comparison value, except in centrosymmetric mode. This bit is cleared by software. 1: The value of the core counter matches the value of compare capture register 1; 0: No match occurs. If compare capture channel 1 is configured as input mode. This bit is set by hardware when a capture event occurs, and it is cleared by software or by reading the compare capture register. 1: the counter value has been captured compare capture register 1. 0: No input capture is generated.	0
0	UIF	RW0	Update interrupt flag bit, this bit is set by hardware when an update event is generated and cleared by software. 1: Update interrupt generation. 0: No update event is generated. The following scenarios generate update events. If UDIS = 0, when the repeat counter value overflows or underflows. If URS = 0, UDIS = 0, when the UG bit is set, or when the counter core counter is reinitialized by software. If URS = 0, UDIS = 0, when the counter CNT is reinitialized by a trigger event.	0

10.4.6 Event Generation Register (TIM1_SWEVGR)

Offset address: 0x14

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

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved	0

7	BG	WO	The brake event generation bit, which is set and cleared by software, is used to generate a brake event. 1: Generate a brake event. At this point, MOE=0, BIF=1, if the corresponding interrupt and DMA are enabled, the corresponding interrupt and DMA are generated. 0: No action.	0
6	TG	WO	The trigger event generation bit, which is set by software and cleared by hardware, is used to generate a trigger event. 1: Generate a trigger event, TIF is set, and the corresponding interrupts and DMAs are generated if enabled. 0: No action.	0
5	COMG	WO	Compare capture control update generation bit. Generates a compare capture control update event. This bit is set by software and automatically cleared by hardware. 1: when CCPC = 1, allow updating of CCxE, CCxNE, OCxM bits. 0: No action. *Note: This bit is only valid for channels with complementary outputs (channels 1, 2, 3).*	0
4	CC4G	WO	Compare capture event generation bit 4. generates compare capture event 4.	0
3	CC3G	WO	Compare capture event generation bit 3. generates compare capture event 3.	0
2	CC2G	WO	Compare capture event generation bit 2. generates compare capture event 2.	0
1	CC1G	WO	Compare capture event generation bit 1. generates compare capture event 1. This bit is set by software and cleared by hardware. It is used to generate a compare capture event. 1: Generate a compare capture event on compare capture channel 1. If compare capture channel 1 is configured as output. Set the CC1IF bit. Generate the corresponding interrupts and DMAs if they are enabled. If compare capture channel 1 is configured as input. The current core counter value is captured to compare capture register 1; set the CC1IF bit to generate the corresponding interrupts and DMAs if they are enabled; if CC1IF is already set, set the CC1OF bit. 0: No action.	0
0	UG	WO	Update event generation bit to generate an update event. This bit is set by software and is automatically cleared by hardware. 1: Initialize the counter and generate an update event. 0: No action. *Note: The prescaler counter is also cleared to zero, but the prescaler factor remains unchanged. The core counter is cleared if in centrosymmetric mode or incremental counting mode; if in decremental counting mode, the core counter takes the value of the reload value register.*	0

10.4.7 Compare/Capture Control Register 1 (TIM1_CHCTLR1)

Offset address: 0x18

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]		OC1CE	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
IC2F[3:0]				IC2PSC[1:0]				IC1F[3:0]				IC1PSC[1:0]			

Compare mode (pin direction is output).

Bit	Name	Access	Description	Reset value
15	OC2CE	RW	Compare capture channel 2 clear enable bit. 1: Clear OC2REF bit zero once ETRF input is detected high; 0: OC2REF is not affected by ETRF input.	0
[14:12]	OC2M[2:0]	RW	Compare Capture Channel 2 mode setting field. The 3 bits define the action of the output reference signal OC2REF, which determines the values of OC2, OC2N. OC2REF is active high, while the active levels of OC2 and OC2N depend on the CC2P, CC2NP bits. 000: Freeze. Comparison of the value of the capture register with the value of the comparison between the core counters does not work for OC2REF. 001: force to set to valid level. Forcing OC2REF high when the core counter has the same value as the comparison capture register 1. 010: Force to set to invalid level. Forcing OC2REF low when the value of the core counter is the same as the comparison capture register 1. 011: Flip. Flips the level of OC2REF when the core counter is the same as the value of compare capture register 1. 100: Forced to invalid level. Forces OC2REF to low. 101: Forced to valid level. Force OC2REF to high. 110: PWM mode 1: When counting up, channel 2 is invalid level once the core counter is greater than the value of the compare capture register, otherwise it is valid level; when counting down, channel 2 is valid level once the core counter is greater than the value of the compare capture register, otherwise it is invalid level. 111: PWM mode 2: When counting up, channel 2 is valid level once the core counter is greater than the value of the compare capture register, otherwise it is invalid level; when counting down, channel 2 is invalid level once the core counter is greater than the value of the compare capture register, otherwise it is valid level (OC2REF=1). *Note: This bit cannot be modified once the LOCK level is set to 3 and CC2S=00b. In PWM mode 1 or PWM mode 2, the OC2REF level is changed only when the comparison result is changed or when switching from freeze mode to PWM mode in the output comparison mode.*	0
11	OC2PE	RW	Compare Capture Register 1 preload enable bit.	0

			1: Enable the preload function of compare capture register 1, read and write operations only operate on the preload registers, the preload value of compare capture register 1 is loaded into the current shadow register when the update event comes; 0: Disable the preload function of compare capture register 1, compare capture register 1 can be written at any time, and the newly written value takes effect immediately. *Note: Once the LOCK level is set to 3 and CC2S=00, this bit cannot be modified; PWM mode can be used only in single pulse mode (OPM=1) without confirming the pre-load register, otherwise its action is not determined.*	
10	OC2FE	RW	Compare Capture Channel 2 fast enable bit, this bit is used to speed up the response of the compare capture channel output to a trigger input event. 1: The active edge of the input to the flipflop acts as if a comparison match has occurred. Therefore, the OC is set to the comparison level independent of the comparison result. The delay between the valid edge of the sample trigger and the output of the compare capture channel 2 is reduced to 3 clock cycles. 0: Based on the value of the counter and compare capture register 1, compare capture channel 2 operates normally, even if the flip-flop is open. The minimum delay to activate the compare capture channel 2 output is 5 clock cycles when the input of the flipflop has a valid edge. OC2FE only works when the channel is configured to PWM1 or PWM2 mode.	0
[9:8]	CC2S[1:0]	RW	Compare capture channel 2 input selection fields. 00: comparison capture channel 2 is configured as an output. 01: comparison capture channel 2 is configured as an input and IC2 is mapped on TI2. 10: comparison capture channel 2 is configured as an input and IC2 is mapped on TI1. 11: Compare Capture Channel 2 is configured as an input and IC2 is mapped on TRC. This mode works only when the internal trigger input is selected (by the TS bit). *Note: Compare Capture Channel 2 is writable only when the channel is off (when CC2E is zero).*	0
7	OC1CE	RW	Compare capture channel 1 clear enable bit.	0
[6:4]	OC1M[2:0]	RW	Compare capture channel 1 mode setting field.	0
3	OC1PE	RW	Compare capture register 1 preload enable bit.	0
2	OC1FE	RW	Compare capture channel 1 fast enable bit.	0
[1:0]	CC1S[2:0]	RW	Compare capture channel 1 input selection fields.	0

Capture mode (pin direction is input).

Bit	Name	Access	Description	Reset value
[15:12]	IC2F[3:0]	RW	The input capture filter 2 configuration field, these bits set the sampling frequency of the TI1 input and the digital filter length. The digital filter consists of an event counter, which records N events and then generates a jump in the output. 0000: no filter, sampled at fDTS.	0

			1000: sampling frequency Fsampling = Fdts/8, N = 6. 0001: sampling frequency Fsampling=Fck_int, N=2. 1001: sampling frequency Fsampling = Fdts/8, N = 8. 0010: sampling frequency Fsampling=Fck_int, N=4. 1010: sampling frequency Fsampling = Fdts/16, N = 5. 0011: sampling frequency Fsampling=f=Fck_int, N=8. 1011: sampling frequency Fsampling = Fdts/16, N = 6. 0100: sampling frequency Fsampling = Fdts/2, N = 6. 1100: sampling frequency Fsampling = Fdts/16, N = 8. 0101: sampling frequency Fsampling = Fdts/2, N = 8. 1101: sampling frequency Fsampling = Fdts/32, N = 5. 0110: sampling frequency Fsampling = Fdts/4, N = 6. 1110: sampling frequency Fsampling = Fdts/32, N = 6. 0111: sampling frequency Fsampling = Fdts/4, N = 8. 1111: Sampling frequency Fsampling=Fdts/32, N=8.	
[11:10]	IC2PSC[1:0]	RW	Compare capture channel 2 prescaler configuration field, these 2 bits define the prescaler coefficient for compare capture channel 2. Once CC1E = 0, the prescaler is reset. 00: without prescaler, one capture is triggered for each edge detected on the capture input. 01: capture triggered every 2 events. 10: capture triggered every 4 events. 11: Capture is triggered every 8 events.	0
[9:8]	CC2S[1:0]	RW	Compare the capture channel 2 input selection field, these 2 bits define the direction of the channel (input/output), and the selection of the input pin. 00: comparative capture channel 1 channel is configured as an output. 01: comparison capture channel 1 channel is configured as an input and IC1 is mapped on TI1. 10: comparison capture channel 1 channel is configured as an input and IC1 is mapped on TI2. 11: The compare capture channel 1 channel is configured as an input and IC1 is mapped on TRC. This mode works only when the internal trigger input is selected (by the TS bit). *Note: CC1S is writable only when the channel is off (CC1E is 0).*	0
[7:4]	IC1F[3:0]	RW	Input capture filter 1 configuration field.	0
[3:2]	IC1PSC[1:0]	RW	Compare the capture channel 1 prescale configuration field.	0
[1:0]	CC1S[1:0]	RW	Compare capture channel 1 input selection fields.	0

10.4.8 Compare/Capture Control Register 2 (TIM1_CHCTLR2)

Offset address: 0x1C

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

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

Compare mode (pin direction is output).

Bit	Name	Access	Description	Reset value
15	OC4CE	RW	Compare capture channel 4 clear enable bit.	0
[14:12]	OC4M[2:0]	RW	Compare the Capture Channel 4 mode setting field.	0
11	OC4PE	RW	Compare Capture Register 4 preload enable bit.	0
10	OC4FE	RW	Compare capture channel 4 fast enable bit.	0
[9:8]	CC4S[1:0]	RW	Compare capture channel 4 input selection fields.	0
7	OC3CE	RW	Compare capture channel 3 clear enable bit.	0
[6:4]	OC3M[2:0]	RW	Compare capture channel 3 mode setting field.	0
3	OC3PE	RW	Compare Capture Register 3 preload enable bit.	0
2	OC3FE	RW	Compare capture channel 3 fast enable bit.	0
[1:0]	CC3S[1:0]	RW	Compare capture channel 3 input selection fields.	0

Capture mode (pin direction is input).

Bit	Name	Access	Description	Reset value
[15:12]	IC4F[3:0]	RW	Input capture filter 4 configuration field.	0
[11:10]	IC4PSC[1:0]	RW	Compare the capture channel 4 prescaler configuration field.	0
[9:8]	CC4S[1:0]	RW	Compare capture channel 4 input selection fields.	0
[7:4]	IC3F[3:0]	RW	Input capture filter 3 configuration field.	0
[3:2]	IC3PSC[1:0]	RW	Compare capture channel 3 prescaler configuration fields.	0
[1:0]	CC3S[1:0]	RW	Compare capture channel 3 input selection fields.	0

10.4.9 Compare/Capture Enable Register 2 (TIM1_CCER)

Offset address: 0x20

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

Bit	Name	Access	Description	Reset value
[15:14]	Reserved	RO	Reserved	0
13	CC4P	RW	Compare the capture channel 4 output polarity setting bit.	0
12	CC4E	RW	Compare capture channel 4 output enable bit.	0
11	CC3NP	RW	Compare capture channel 3 complementary output polarity setting bit.	0
10	CC3NE	RW	Compare capture channel 3 complementary output enable bits.	0
9	CC3P	RW	Compare the capture channel 3 output polarity setting bit.	0
8	CC3E	RW	Compare the capture channel 3 output enable bit.	0
7	CC2NP	RW	Compare capture channel 2 complementary output polarity setting bit.	0
6	CC2NE	RW	Compare capture channel 2 complementary output enable bits.	0
5	CC2P	RW	Compare the capture channel 2 output polarity setting bit.	0
4	CC2E	RW	Compare the capture channel 2 output enable bit.	0
3	CC1NP	RW	Compare capture channel 1 complementary output polarity setting bit.	0
2	CC1NE	RW	Compare capture channel 1 complementary output enable bit.	0

1	CC1P	RW	Compare capture channel 1 output polarity setting bit. CC1 channel configured as output. 1: OC1 active low. 0: OC1 active high. CC1 channel configured as input: This bit selects whether IC1 or the inverted signal of IC1 is used as the trigger or capture signal. 1: Inverted: capture occurs on the falling edge of IC1; when used as an external trigger, IC1 is inverted. 0: Non-inverted: capture occurs on the rising edge of IC1; when used as an external trigger, IC1 is not inverted. *Note: Once the LOCK level (LOCK bit in TIMx_BDTR register) is set to 3 or 2, this bit cannot be modified.*	0
0	CC1E	RW	Compare capture channel 1 output enable bit. The CC1 channel is configured as output: 1: ON. the OC1 signal is output to the corresponding output pin, and its output level depends on the values of the MOE, OSSI, OSSR, OIS1, OIS1N, and CC1NE bits. 0: off. OC1 disables output, so the output level of OC1 depends on the values of the MOE, OSSI, OSSR, OIS1, OIS1N, and CC1NE bits. The CC1 channel is configured as an input: This bit determines whether the counter value can be captured into the TIMx_CCR1 register. 1: capture enable. 0: capture disable.	0

10.4.10 Counter for Advanced-control Timer (TIM1_CNT)

Offset address: 0x24

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

Bit	Name	Access	Description	Reset value
[15:0]	CNT[15:0]	RW	The real-time value of the timer's counter.	0

10.4.11 Counting Clock Prescaler (TIM1_PSC)

Offset address: 0x28

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

Bit	Name	Access	Description	Reset value
[15:0]	PSC[15:0]	RW	The dividing factor of the prescaler of the timer; the clock frequency of the counter is equal to the input frequency of the divider/(PSC+1).	0

10.4.12 Auto-reload Value Register (TIM1_ATRLR)

Offset address: 0x2C

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

ATRLR[15:0]

Bit	Name	Access	Description	Reset value
[15:0]	ATRLR[15:0]	RW	The value of this field will be loaded into the counter, see section 10.2.3 for when the ATRLR acts and updates; the counter stops when the ATRLR is empty.	0

10.4.13 Repeat Count Value Register (TIM1_RPTCR)

Offset address: 0x30

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

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved	0
[7:0]	RPTCR	RW	The value of the repeat counter.	0

10.4.14 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[15:0]	RW	Compare the value of capture register channel 1.	0

10.4.15 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[15:0]	RW	Compare the value of capture register channel 2.	0

10.4.16 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[15:0]	RW	Compare the value of capture register channel 3.	0

10.4.17 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[15:0]	RW	Compare the value of capture register channel 4.	0

10.4.18 Brake and Deadband Register (TIM1_BDTR)

Offset address: 0x44

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

Bit	Name	Access	Description	Reset value
15	MOE	RW	Main output enable bit. Once the brake signal is active, it will be cleared asynchronously. 1: Allow OCx and OCxN to be set as outputs. 0: Disable the output of OCx and OCxN or force to idle state.	0
14	AOE	RW	Auto output enable. 1: the MOE can be set by software or set in the next update event. 0: MOE can only be set by software.	0
13	BKP	RW	The brake input polarity setting bit. 1: brake input active high. 0: Brake input is active low. *Note: When LOCK level 1 is set, this bit cannot be modified. A write to this bit requires an AHB clock before it can take effect.*	0
12	BKE	RW	Brake function enable bit. 1: Turn on the brake input. 0: Brake input is disabled. *Note: When LOCK level 1 is set, this bit cannot be modified. A write to this bitrequires an AHB clock before it can take effect.*	0
11	OSSR	RW	1: when the timer is not working, once CCxE=1 or CCxNE=1, first turn on OC/OCN and outputinvalid level, then set OCx, OCxN enable output signal=1. 0: When the timer is not operating, OC/OCN output is disabled. *Note: When LOCK level 1 is set, this bit cannot be modified.*	0
10	OSSI	RW	1: when the timer is not operating, once CCxE = 1 or CCxNE = 1, OC/OCN first outputs its idle level, then OCx, OCxN enable output signal = 1. 0: When the timer is not operating, OC/OCN output is disabled. *Note: When LOCK level 1 is set, this bit cannot be modified.*	0
[9:8]	LOCK[1:0]	RW	Lock the function setting field. 00: Disable the locking function. 01: lock level 1, no DTG, BKE, BKP, AOE, OISx and OISxN bits can be written. 10: Lock level 2, where the bits in lock level 1 cannot be written, nor the CC polarity bits, nor the OSSR and OSSI bits. 11: Lock level 3, cannot write to the bits in lock level 2,	0

			and cannot write to the CC control bits. *Note: After system reset, the LOCK bit can only be written once and cannot be modified again until reset.*	
[7:0]	DTG[7:0]	RW	Deadband setting bits that define the duration of the deadband between complementary outputs. Assume that DT denotes its duration. DTG[7:5]=0xx=>DT=DTG[7:0]*Tdtg, Tdtg=TDTS； DTG[7:5]=10x=>DT=(64+DTG[5:0])*Tdtg, Tdtg=2*TDTS; DTG[7:5]=110=>DT=(32+DTG[4:0])*Tdtg, Tdtg =8×TDTS; DTG[7:5]=111=>DT=(32+DTG[4:0])*Tdtg, Tdtg =16*TDTS.	0

10.4.19 DMA Control Register (TIM1_DMACFGR)

Offset address: 0x48

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

Bit	Name	Access	Description	Reset value
[15:13]	Reserved	RO	Reserved	0
[12:8]	DBL[4:0]	RW	The length of the DMA continuous transmission, the actual value of which is the value of this field + 1.	0
[7:5]	Reserved	RO	Reserved	0
[4:0]	DBA[4:0]	RW	These bits define the offset of the DMA in continuous mode from the address where control register 1 is located.	0

10.4.20 DMA Address Register for Continuous Mode (TIM1_DMAADR)

Offset address: 0x4C

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

Bit	Name	Access	Description	Reset value
[15:0]	DMAB[15:0]	RW	The address of the DMA in continuous mode.	0

Chapter 11 General-purpose Timer (GPTM)

The general-purpose timer module contains a 16-bit auto-reloadable timer, TIM2, for measuring pulse width or generating pulses of a specific frequency, PWM waves, etc. It can be used in automation control, power supply, etc.

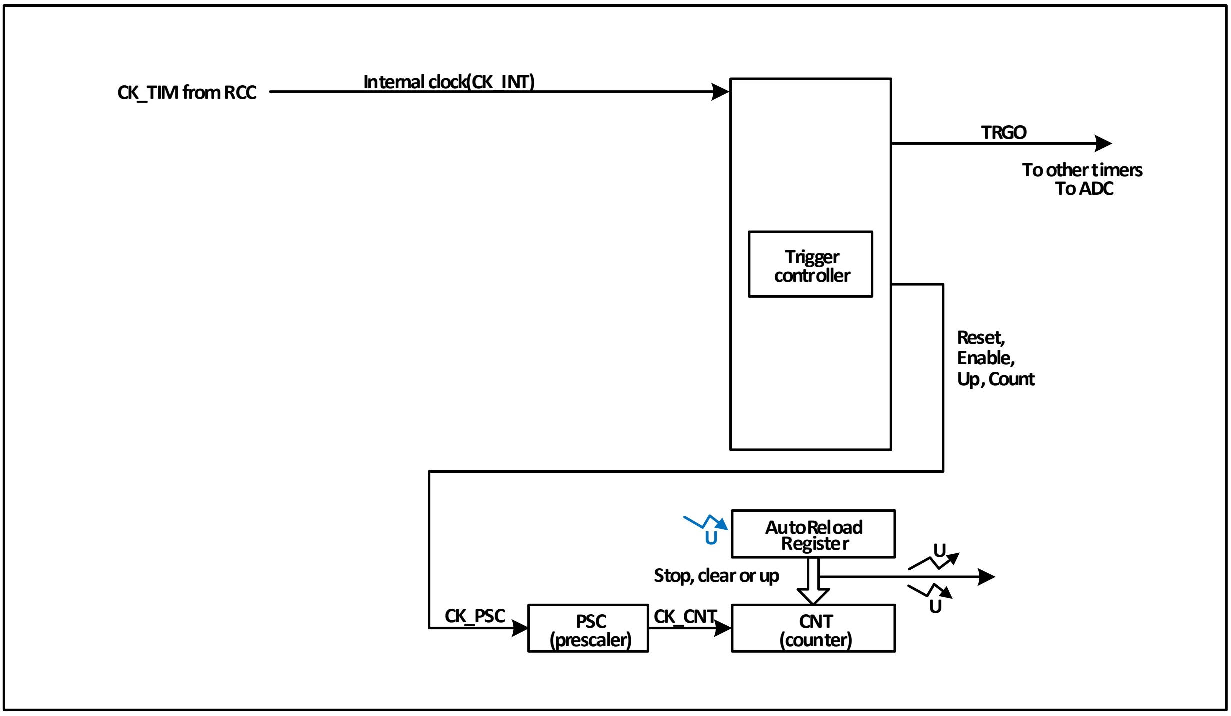
11.1 Main Features

The main features of the general-purpose timer include.

- 16-bit auto-reload counter, supports incremental counting mode, decremental counting mode and incremental and decremental counting mode
- 16-bit prescaler with dynamically adjustable crossover factor from 1 to 65536
- Support four independent comparison capture channels
- Each comparison capture channel supports multiple operating modes, such as: input capture, output comparison, PWM generation, and single pulse output
- Support external signal control timer
- Support DMA in multiple modes
- Support incremental coding, cascading and synchronization between timers

11.2 Principle and Structure

Figure 11-1 Block diagram of the structure of the general-purpose timer



11.2.1 Overview

As shown in Figure 11-1, the structure of the general-purpose timer can be roughly divided into three parts, namely the input clock part, the core counter part and the compare capture channel part.

The clock for the general purpose timer can come from the AHB bus clock (CK_INT), from the external clock input pin (TIMx_ETR), from other timers with clock output (ITRx), and from the input of the compare capture channel (TIMx_CHx). These input clock signals become CK_PSC clocks after various set filtering and dividing operations, etc., and are output to the core counter section. In addition, these complex clock sources can also be output as TRGO to other peripherals such as timers and ADCs.

The core of the general-purpose timer is a 16-bit counter (CNT). cK_PSC is divided by a prescaler (PSC) to become cK_CNT and then finally fed to the CNT, which supports incremental counting mode, decremental counting mode, and incremental and decremental counting mode, and has an auto-reload register (ATRLR) to reload the initialization value for the CNT at the end of each counting cycle.

The general-purpose timer has four sets of compare capture channels, each of which can input pulses from exclusive pins or output waveforms to pins, i.e., the compare capture channels support both input and output modes. The input of each channel of the compare capture register supports filtering, dividing, edge detection, and other operations, and supports mutual triggering between channels, and can also provide clock for the core counter CNT. Each comparison capture channel has a set of comparison capture registers (CHxCVR) that support comparison with the main counter (CNT) to output pulses.

11.2.2 Difference between General-purpose Timer and Advanced-control Timer

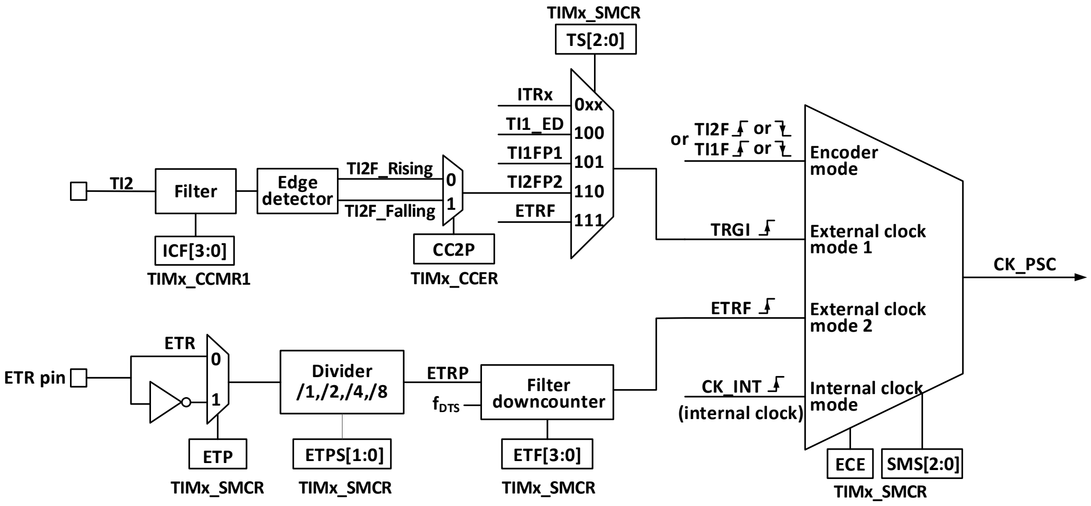
Compared to advanced-control timers, general purpose timers lack the following features.

1) The general-purpose timer lacks a repeat count register for counting the count cycles of the core counter.
2) The comparison capture channel of the general-purpose timer lacks deadband generation and has no complementary output.
3) The general-purpose timer does not have a brake signal mechanism.

11.2.3 Clock Input

This section discusses the source of CK_PSC. The clock source portion of the overall block diagram of the general-purpose timer is captured here.

Figure 11-2 General-Purpose Timer CK_PSC Source Block Diagram



The optional input clocks can be divided into 4 categories.

1) Route of the external clock pin (ETR) input: ETR → ETRP → ETRF.
2) Internal AHB clock input route: CK_INT.
3) Route from the comparison 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 3 categories by determining the choice of input pulse for the SMS of the CK_PSC source.

1) Selection of the internal clock source (CK_INT).
2) External clock source mode 1.

3) External clock source mode 2.
4) Encoder mode.

All 4 clock source sources mentioned above can be selected by these 4 operations.

11.2.3.1 Internal Clock Source (CK_INT)

If the general-purpose timer is started when the SMS field is held at 000b, then it is the internal clock source (CK_INT) that is selected as the clock. At this point CK_INT is CK_PSC.

11.2.3.2 External Clock Source Mode 1

When the SMS domain is set to 111b, external clock source mode 1 is enabled. When external clock source 1 is enabled, TRGI is selected as the source for CK_PSC. it is worth noting that the user also needs to select the source for TRGI by configuring the TS domain. the TS domain can select the following types of pulses as clock sources.

1) Internal trigger (ITRx, x is 0,1,2,3).
2) Comparison of the signal after capturing channel 1 through the edge detector (TI1F_ED).
3) Comparison of signals TI1FP1, TI2FP2 of the capture channel.
4) The signal ETRF from the external clock pin input.

11.2.3.3 External Clock Source Mode 2

Use external trigger mode 2 to count on every rising or falling edge of the external clock pin input. When the ECE position is set, the external clock source mode 2 is used. when using the external clock source mode 2, ETRF is selected as CK_PSC. the ETR pin becomes ETRP after passing through the optional inverter (ETP), divider (ETPS), and then ETRF after passing through the filter (ETF).

With the ECE position bit and the SMS set to 111b, then it is equivalent to the TS selecting ETRF as the input.

11.2.3.4 Encoder Mode

Setting the SMS to 001b, 010b, 011b will enable the encoder mode. Enabling encoder mode allows you to select a specific level in TI1FP1 and TI2FP2 to signal the output with another jump edge as the signal. This mode is used when an external encoder is used. Refer to Section 10.3.9 for specific functions.

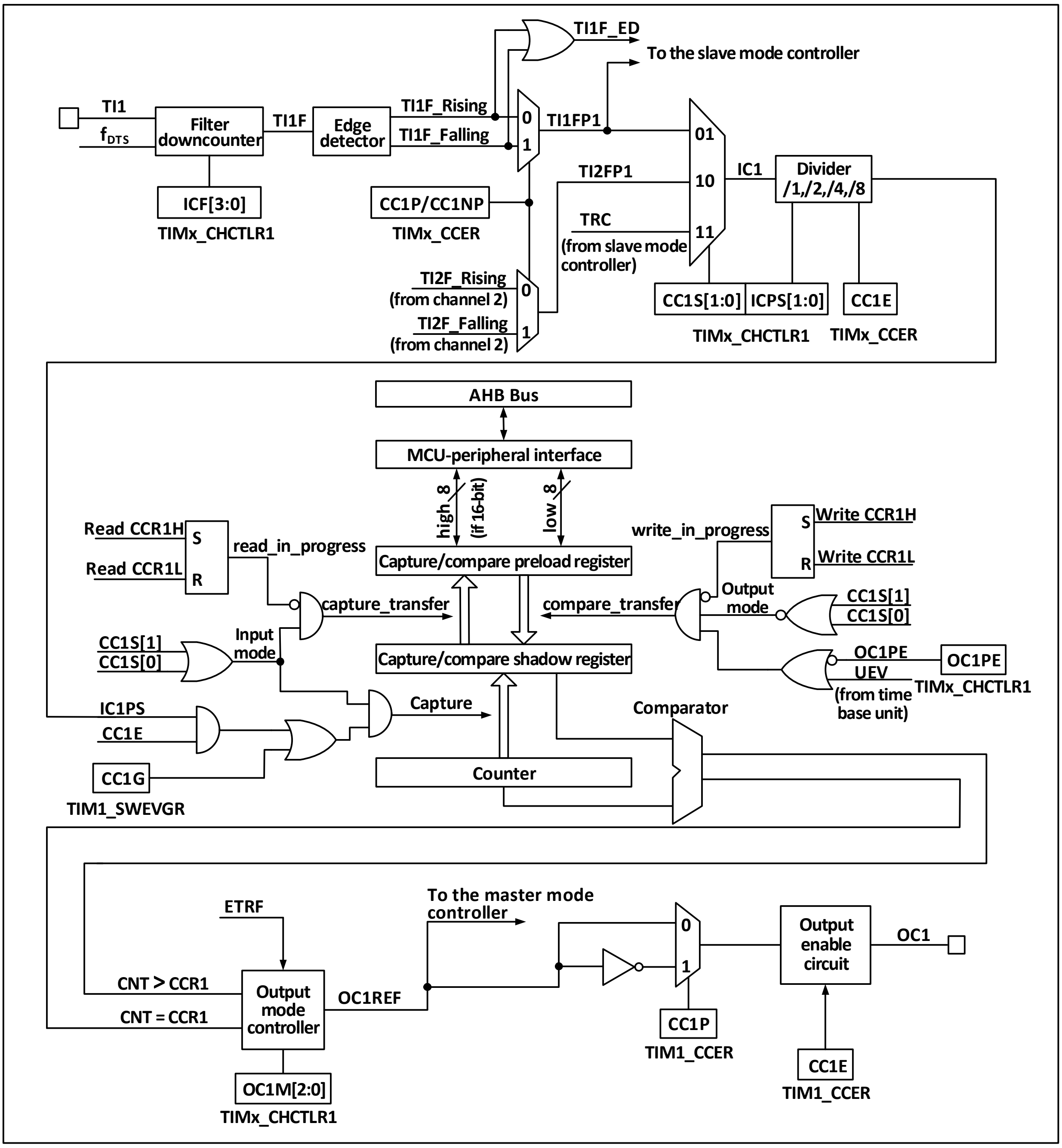
11.2.4 Counters and Peripherals

CK_PSC is input to the prescaler (PSC) for dividing. the PSC is 16-bit and the actual dividing factor is equal to the value of R16_TIMx_PSC + 1. CK_PSC goes through the PSC and becomes CK_INT. changing the value of R16_TIM1_PSC does not take effect in real time, but is updated to the PSC after an update event. the update event includes a UG bit clear and reset.

11.2.5 Compare/capture Channels

The core of the compare/capture channel, which is the core of the timer to achieve complex functions, is the compare/capture register, supplemented by digital filtering, frequency division and inter-channel multiplexing in the peripheral input section, and comparator and output control in the output section. The structure block diagram of the compare/capture channel is shown in Figure 11-3.

Figure 11-3 Block diagram of the structure of the comparison capture channel



The signal is input from the channel x pin and optionally made as TIx (the source of TI1 can be more than CH1, see block diagram 10-1 of the timer), TI1 is passed through the filter (ICF[3:0]) to generate TI1F, and then divided into TI1F_Rising and TI1F_Falling through the edge detector, these two signals are selected (CC1P) to generate TI1FP1, TI1FP1 and TI2FP1 from channel 2 are sent together to CC1S to select to become IC1, which is sent to the comparison capture register after ICPS dividing.

The compare capture register consists of a preload register and a shadow register, and the read/write process operates only on the preload register. In capture mode, the capture occurs on the shadow register and is then copied to the preload register; in compare mode, the contents of the preload register are copied to the shadow register, and then the contents of the shadow register are compared to the core counter (CNT).

11.3 Functionality and Implementation

The complex functions of a general-purpose timer are implemented by manipulating the timer's compare capture channel, clock input circuitry, and counter and peripheral components. The clock input to the timer can be derived from multiple clock sources including the input to the compare capture channel. The operation of the compare capture host channel and clock source selection directly determines its function. The compare capture channel is bidirectional and can operate in both input and output modes.

11.3.1 Input Capture Mode

The input capture mode is one of the basic functions of the timer. The principle of input capture mode is that when a determined edge on the ICxPS signal is detected, a capture event is generated and the current value of the counter is latched into the compare capture register (R16_TIMx_CHCTLRx). The CCxIF (in R16_TIMx_INTFR) is set when a capture event occurs, and the corresponding interrupt or DMA is generated if enabled. If the CCxIF is already set when a capture event occurs, the CCxOF bit is set. the CCxIF can be cleared by software, or by hardware by reading the compare capture register. CCxOF is cleared by software.

An example of channel 1 to illustrate the steps to use the input capture mode is as follows.

1) Configure the CCxS domain to select the source of the ICx signal. For example, set it to 10b and select TI1FP1 as the source of IC1, not using the default setting, the CCxS domain defaults to making the comparison capture module the output channel.
2) Configure the ICxF domain to set the digital filter for the TI signal. The digital filter will sample the signal at a determined frequency, a determined number of times, and then output a hop. This sampling frequency and number of times is determined by ICxF.
3) Configure the CCxP bit to set the polarity of the TIxFPx. For example, keeping the CC1P bit low and selecting rising edge jumps.
4) Configure the ICxPS domain to set the ICx signal to be the crossover factor between ICxPS. For example, keeping ICxPS at 00b, without crossover.
5) Configure the CCxE bit to allow capturing the value of the core counter (CNT) into the compare capture register. Set the CC1E bit.
6) Configure the CCxIE and CCxDE bits as needed to determine whether to allow enable interrupts or DMA.

This completes the comparison capture channel configuration.

When a captured pulse is input to TI1, the value of the core counter (CNT) is recorded in the compare capture register, CC1IF is set, and the CCIOF bit is set when CC1IF has been set before. If the CC1IE bit is set, then an interrupt is generated; if CC1DE is set, a DMA request is generated. An input capture event can be generated by software by way of writing the event generation register (R16_TIMx_SWEVGR).

11.3.2 Compare Output Mode

The compare output mode is one of the basic functions of the timer. The principle of the compare output mode is to output a specific change or waveform when the value of the core counter (CNT) agrees with the value of the compare capture register. the OCxM field (in R16_TIMx_CHCTLRx) and the CCxP bit (in R16_TIMx_CCER) determine whether the output is a definite high or low level or a level flip. The CCxIF bit is also set when a compare coherent event is generated. If the CCxIE bit is pre-set, an interrupt will be generated; if the CCxDE bit is pre-set, a DMA request will be generated.

To configure to compare output modes, proceed as follows.

1) Configuring the clock source and auto-reload value of the core counter (CNT).
2) Set the count value to be compared to the comparison capture register (R16_TIMx_CHxCVR).
3) Set the CCxIE bit if an interrupt needs to be generated.
4) Keep OCxPE at 0 to disable the preload register for the compare capture register.
5) Setting the output mode, setting the OCxM field and the CCxP bit.
6) Enable the output, setting the CCxE bit.
7) Setting the CEN bit to start the timer.

11.3.3 Forced Output Mode

The output pattern of the timer's compare capture channel can be forced by software to output a determined level without relying on comparison of the compare capture register's shadow register with the core counter.

This is done by setting OCxM to 100b, which forces OCxREF to low, or by setting OCxM to 101b, which forces OCxREF to high.

Note that by forcing OCxM to 100b or 101b, the comparison process between the internal main counter and the compare capture register is still going on, the corresponding flags are still set, and interrupts and DMA requests are still being generated.

11.3.4 PWM Input Mode

The PWM input mode is used to measure the duty cycle and frequency of PWM and is a special case of the input capture mode. The operation is the same as 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 as trigger input and SMS is set to reset mode.

For example, to measure the period and frequency of the PWM wave input from TI1, the following operations are required.

1) Set TI1 (TI1FP1) to be the input of IC1 signal. Set CC1S to 01b.
2) Set TI1FP1 to rising edge active. Holding CC1P at 0.
3) Set TI1 (TI1FP2) as the input of IC2 signal. Set CC2S to 10b.
4) Select TI1FP2 to set to falling edge active. Set CC2P to 1.
5) Select TI1FP1 as the source of the clock source. set TS to 101b.
6) Set the SMS to reset mode, i.e. 100b.
7) Enables input capture. cc1e and cc2e are set.

11.3.5 PWM Output Mode

PWM output mode is one of the basic functions of the timer. PWM output mode is most commonly used to determine the PWM frequency using the reload value and the duty cycle using the capture comparison register. Set 110b or 111b in the OCxM field to use PWM mode 1 or mode 2, set the OCxPE bit to enable the preload register, and finally set the ARPE bit to enable the automatic reload of the preload register. The value of the preload register can only be sent to the shadow register when an update event occurs, so the UG bit needs to be set to initialize all registers before the core counter starts counting. In PWM mode, the core counter and the compare capture register are always comparing, and depending on the CMS bit, the timer is able to output edge-aligned or center-aligned PWM signals.

- Edge alignment

When using edge alignment, the core counter is incremented or decremented, and in the PWM mode 1 scenario, OCxREF rises to high when the core counter value is greater than the compare capture register; when the core counter value is less than the compare captureregister (for example, when the core counter grows to the value of R16_TIMx_ATRLR and reverts to full 0), OCxREF falls to low.

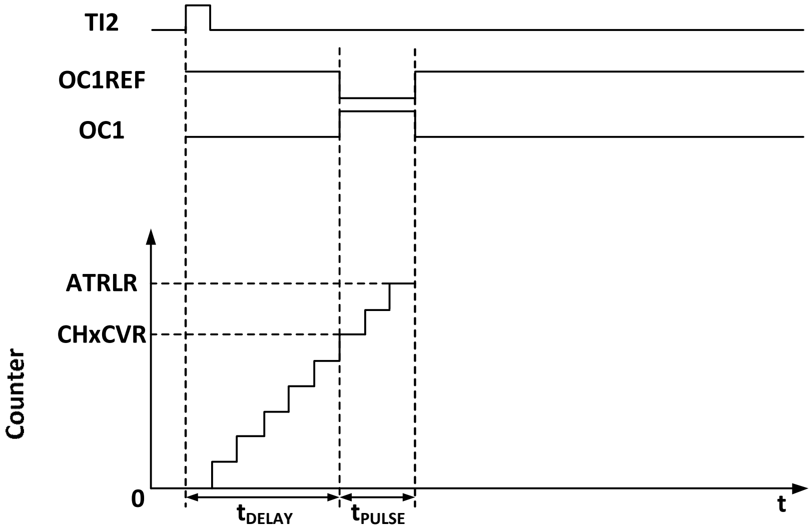
- Central alignment

When using the central alignment modes, the core counter runs in alternating incremental and decremental count modes, and OCxREF performs rising and falling jumps when the values of the core counter and the compare capture register match. However, the comparison flags are set at different times in the three central alignment modes. When using the central alignment modes, it is best to generate a software update flag (set the UG bit) before starting the core counter.

11.3.6 Single Pulse Mode

The single pulse mode can respond to a specific event by generating a pulse after a delay, with programmable delay and pulse width. Setting the OPM bit stops the core counter when the next update event UEV is generated (counter flips to 0).

Figure 11-4 Event generation and impulse response



As shown in Figure 11-4, a positive pulse of length Tpulse needs to be generated on OC1 after a delay Tdelay at the beginning of a rising edge detected on the TI2 input pin.

1) Set TI2 to trigger. Setting the CC2S field to 01b to map TI2FP2 to TI2; setting the CC2P bit to 0b to set TI2FP2 as rising edge detection; setting the TS field to 110b to set TI2FP2 as trigger source; setting the SMS field to 110b to set TI2FP2 to be used to start the counter.
2) Tdelay is defined by the Compare Capture Register and Tpulse is determined by the value of the Auto Reload Value Register and the Compare Capture Register.

11.3.7 Encoder Mode

The encoder mode is a typical application of the timer and can be used to access the biphasic output of the encoder. The counting direction of the core counter is synchronized with the direction of the encoder's rotation axis, and each pulse output from the encoderwill add or subtract one from the core counter. To use the encoder, set the SMS field to 001b (count only on TI2 edge), 010b (count only on TI1 edge) or 011b (count on both TI1 and TI2 edges), connect the encoder to the input of the comparison capture channels 1 and 2, and set a reload value counter value, which can be set to a larger value. When in encoder mode, the internal compare capture register, prescaler, repeat count register, etc. of the timer are working normally. The following table shows the relationship between the counting direction and the encoder signal.

Table 11-1 Relationship between counting direction and encoder signal of timer encoder mode

Counting effective edges	The level of relative signals	TI1FP1 signal edge		TI2FP2 signal edge	
		Rising edge	Falling edge	Rising edge	Falling edge
Counting at TI1 edge only	High	Downward counting	Upward counting	No count	
	Low	Upward counting	Downward counting		
Counting at TI2 edge only	High	No count		Upward counting	Downward counting
	Low			Downward counting	Upward counting
Double edge counting at TI1 and TI2	High	Downward counting	Upward counting	Upward counting	Downward counting
	Low	Upward counting	Downward counting	Downward counting	Upward counting

11.3.8 Timer Synchronization Mode

Timers are capable of outputting clock pulses (TRGO) and also receiving inputs from other timers (ITRx). The source of ITRx (TRGO from other timers) is different for different timers. The timer internal trigger connections are shown in Table 11-2.

Table 11-2 GTPM internal trigger connection

From timer	ITR0(TS=000)	ITR1(TS=001)	ITR2(TS=010)	ITR3(TS=011)
TIM2	TIM1			
TIM1		TIM2		

11.3.9 Debug Mode

When the system enters the debug mode, the timer can be controlled to continue running or stop according to the setting of DBG module.

11.4 Register Description

Table 11-3 TIM2-related registers list

Name	Offset address	Description	Reset value
R16_TIM2_CTLR1	0x40000000	TIM2 control register1	0x0000
R16_TIM2_CTLR2	0x40000004	TIM2 control register2	0x0000
R16_TIM2_SMCFGR	0x40000008	TIM2 Slave mode control register	0x0000
R16_TIM2_DMAINTENR	0x4000000C	TIM2 DMA/interrupt enable register	0x0000
R16_TIM2_INTFR	0x40000010	TIM2 interrupt status register	0x0000
R16_TIM2_SWEVGR	0x40000014	TIM2 event generation register	0x0000
R16_TIM2_CHCTLR1	0x40000018	TIM2 compare/capture control register1	0x0000
R16_TIM2_CHCTLR2	0x4000001C	TIM2 compare/capture control register2	0x0000
R16_TIM2_CCER	0x40000020	TIM2 compare/capture enable register	0x0000
R16_TIM2_CNT	0x40000024	TIM2 counter	0x0000
R16_TIM2_PSC	0x40000028	TIM2 count clock prescaler	0x0000
R16_TIM2_ATRLR	0x4000002C	TIM2 auto-reload register	0x0000
R16_TIM2_CH1CVR	0x40000034	TIM2 compare/capture register1	0x0000
R16_TIM2_CH2CVR	0x40000038	TIM2 compare/capture register2	0x0000
R16_TIM2_CH3CVR	0x4000003C	TIM2 compare/capture register3	0x0000
R16_TIM2_CH4CVR	0x40000040	TIM2 compare/capture register4	0x0000
R16_TIM2_DMACFGR	0x40000048	TIM2 DMA control register	0x0000
R16_TIM2_DMAADR	0x4000004C	TIM2 DMA address register in continuous mode	0x0000

11.4.1 Control Register 1 (TIM2_CTLR1)

Offset address: 0x00

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>CAPLVL</td><td>CAPOV</td><td colspan="4">Reserved</td><td colspan="2">CKD[1:0]</td><td>ARPE</td><td colspan="2">CMS[1:0]</td><td>DIR</td><td>OPM</td><td>URS</td><td>UDIS</td><td>CEN</td></tr>
</table>

Bit	Name	Access	Description	Reset value
15	CAPLVL	RW	In double-edge capture mode, the capture level indication is enabled.	0

			0: Turn off the indication function 1: Enables the indication function. *Note: When enabled, [17] of CHxCVR indicates the level corresponding to the capture value.*	
14	CAPOV	RW	Capture value mode configuration. 0: The capture value is the actual counter value 1: The CHxCVR value is 0xFFFF when a counter overflow is generated before capture.	0
[13:10]	Reserved	RO	Reserved	0
[9:8]	CKD[1:0]	RW	These 2 bits define the division ratio between the timer clock (CK_INT) frequency, the sampling clock used by the digital filter. 00: Tdts=Tck_int; 01: Tdts= 2xTck_int; 10: Tdts= 4xTck_int; 11：Reserved.	0
7	ARPE	RW	Auto-reload preload enable bit. 1: Enables the Auto-reload value register (ATRLR). 0: Auto-reload value register (ATRLR) is disabled.	0
[6:5]	CMS[1:0]	RW	Central alignment mode selection. 00: Edge-aligned mode. The counter counts up or down based on the direction bit (DIR). 01: Central alignment mode 1. The counter counts up and down alternately. The output compare interrupt flag bit of the channel configured as output (CCxS=00 in the CHCTLRx register) is set only when the counter counts down. 10: Central alignment mode 2. The counter counts up and down alternately. The output compare interrupt flag bit of the channel configured as output (CCxS=00 in the CHCTLRx register) is set only when the counter counts up. 11: Central alignment mode 3. The counter counts up and down alternately. The output compare interrupt flag bit of the channel configured as output (CCxS=00 in the CHCTLRx register) is set when the counter counts both up and down. *Note: When the counter is enabled (CEN=1), the transition from edge-aligned mode to center-aligned mode is not allowed.*	0
4	DIR	RW	Counting direction. 0: the counter's counting mode is incremental. 1: The counting mode of the counter is decimal counting. *Note: This bit is not valid when the counter is configured in central alignment mode or encoder mode.*	0
3	OPM	RW	Single pulse mode. 1: the counter stops when the next update event (clearing the CEN bit) occurs. 0: The counter does not stop when the next update event occurs.	0
2	URS	RW	Update request source, by which the software selects the source of the UEV event. 1: if an update interrupt or DMA request is enabled, only an update interrupt or DMA request is generated if the counter overflows/underflows. 0: If an update interrupt or DMA request is enabled, the update interrupt or DMA request is generated by any of the following events.	0

			-Counter overflow/underflow -Setting the UG position -Updates generated from the mode controller	
1	UDIS	RW	Disable updates, the software allows/disables the generation of UEV events via this bit. 1: UEV is disabled. no update event is generated and the registers (ATRLR, PSC, CHCTLRx) maintain their values. If the UG bitis set or a hardware reset is issued from the mode controller, the counter and prescaler are reinitialized. 0: UEV is allowed. update (UEV) events are generated by any of the following events: - Counter overflow/underflow -Setting the UG position -Updates generated from the mode controller registers with caches are loaded with their preloaded values.	0
0	CEN	RW	Enable the counter (Counter enable). 1: Enables the counter. 0: Disable the counter. *Note: The external clock, gated mode and encoder mode will not work until the CEN bit is set in software. Trigger mode can automatically set the CEN bit in hardware.*	0

11.4.2 Control Register 2 (TIM2_CTLR2)

Offset address: 0x04

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

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved	0
7	TI1S	RW	TI1 selection. 1: TIMx_CH1, TIMx_CH2 and TIMx_CH3 pins connected to TI1 input after heterodyning. 0: TIMx_CH1 pin is connected directly to TI1 input.	0
[6:4]	MMS[2:0]	RW	Master mode selection: These 3 bits are used to select the synchronization information (TRGO) sent to the slave timer in master mode. The possible combinations are as follows. 000: The Reset-UG bit is used as a trigger output (TRGO). If the reset is generated by a trigger input (from a mode controller in reset mode), there is a delay in the signal on TRGO relative to the actual reset. 001: Enable - The counter enable signal CNT_EN is used as a trigger output (TRGO). Sometimes it is necessary to start multiple timers at the same time or to control the enable from timers over a period of time. The counter enable signal is generated by the logical or of the trigger input signal in CEN control bit and gated mode. When the counter enable signal is controlled by a trigger input, there is a delay on TRGO unless master/slave mode is selected (see the description of the MSM bit in the TIMx_SMCFGR register). 010: The update event is selected as a trigger input (TRGO). For example, the clock of a master timer may be used as a prescaler for a slave timer.	0

			011: comparison pulse that triggers the output to send a positive pulse (TRGO) when a capture or a successful comparison occurs, when the CC1IF flag is to be set (even if itis already high). 100: The OC1REF signal is used as a trigger output (TRGO. 101: The OC2REF signal is used as a trigger output (TRGO). 110: The OC3REF signal is used as a trigger output (TRGO). 111: The OC4REF signal is used as a trigger output (TRGO).	
3	CCDS	RW	1: Sending a DMA request for CHxCVR when an update event occurs. 0: Generate a DMA request for CHxCVR when CHxCVR occurs.	0
[2:0]	Reserved	RO	Reserved	0

11.4.3 Slave Mode Control Register (TIM2_SMCFGR)

Offset address: 0x08

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

Bit	Name	Access	Description	Reset value
15	ETP	RO	ETR trigger polarity selection, this bit selects whether to input ETR directly or to input the inverse of ETR. 1: Invert ETR, active low or falling edge. 0: ETR, active high or rising edge.	0
14	ECE	RW	External clock mode 2 enabled selection. 1: Enables external clock mode 2. 0: Disable external clock mode 2. *Note 1: Slave mode can be used simultaneously with external clock mode 2: reset mode, gated mode and trigger mode; however, TRGI cannot be connected to ETRF in this case (TS bit cannot be 111b).* *Note 2: When both external clock mode 1 and external clock mode 2 are enabled, the external clock input is ETRF.*	0
[13:12]	ETPS[1:0]	RW	The external trigger signal (ETRP) is divided into frequencies, and the maximum frequency of this signal cannot exceed is 1/4 of the TIMxCLK frequency, which can be downconverted by this domain. 00: Prescaler off. 01: ETRP frequency divided by 2. 10: ETRP frequency divided by 4. 11: ETRP frequency divided by 8.	0
[11:8]	ETF[3:0]	RW	Externally triggered filtering, in fact, the digital filter is an event counter that uses a certain sampling frequency to generate a jump in the output after N events are recorded. 0000: no filter, sampled in Fdts. 0001: sampling frequency Fsampling=Fck_int,N=2. 0010: sampling frequency Fsampling=Fck_int, N=4. 0011: Sampling frequency Fsampling=Fck_int, N=8. 0100: sampling frequency Fsampling = Fdts/2, N = 6. 0101: sampling frequency Fsampling = Fdts/2, N = 8.	0

			0110: sampling frequency Fsampling = Fdts/4, N = 6. 0111: sampling frequency Fsampling = Fdts/4, N = 8. 1000: sampling frequency Fsampling = Fdts/8, N = 6. 1001: sampling frequency Fsampling = Fdts/8, N = 8. 1010: sampling frequency Fsampling = Fdts/16, N = 5. 1011: sampling frequency Fsampling = Fdts/16, N = 6. 1100: sampling frequency Fsampling = Fdts/16, N = 8. 1101: sampling frequency Fsampling = Fdts/32, N = 5. 1110: sampling frequency Fsampling = Fdts/32, N = 6. 1111: Sampling frequency Fsampling=Fdts/32, N=8.	
7	MSM	RW	Master/slave mode selection. 1: The event on the trigger input (TRGI) isdelayed to allow perfect synchronization between the current timer (via TRGO) and its slave timer. This is useful when the synchronization of several timers to a single external event is required. 0: Does not function.	0
[6:4]	TS[2:0]	RW	Trigger select 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: Filtered timer input 1 (TI1FP1). 010: Internal trigger 2 (ITR2). 110: Filtered timer input 2 (TI2FP2). 011: Internal trigger 3 (ITR3). 111: External trigger input (ETRF). The above only changes when SMS is 0.	0
3	Reserved	RO	Reserved	0
[2:0]	SMS[2:0]	RW	Input mode selection field. Selects the clock and trigger mode of the core counter. 000: driven by the internal clock CK_INT. 001: encoder mode 1, where the core counter increments or decrements the count at the edge of TI2FP2 depending on the level of TI1FP1. 010: encoder mode 2, where the core counter increments or decrements the count at the edge of TI1FP1, depending on the level of TI2FP2. 011: encoder mode 3, where the core counter increments and decrements the count on the edges of TI1FP1 and TI2FP2 depending on the input level of another signal; 100: reset mode, where the rising edge of the trigger input (TRGI) will initialize the counter and generate a signal to update the registers. 101: Gated mode, when the trigger input (TRGI) is high, the counter clock is turned on; at the trigger input becomes low, the counter is stopped, and the counter starts and stops are controlled. 110: Trigger mode, where the counter is started on the rising edge of the trigger input TRGI and only the start of the counter is controlled. 111: External clock mode 1, rising edge of the selected trigger input (TRGI) drives the counter.	0

11.4.4 TIM2 DMA/Interrupt Enable Register (TIM2_DMAINTENR)

Offset address: 0x0C

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

Reserved	TDE	Reserved	CC4DE	CC3DE	CC2DE	CC1DE	UDE	Reserved	TIE	Reserved	CC4IE	CC3IE	CC2IE	CC1IE	UIE

Bit	Name	Access	Description	Reset value
15	Reserved	RO	Reserved	0
14	TDE	RW	Trigger the DMA request enable bit. 1: Allowing DMA requests to be triggered. 0: Triggering of DMA requests is prohibited.	0
13	Reserved	RO	Reserved	0
12	CC4DE	RW	Compare the DMA request enable bit of capture channel 4. 1: Allows comparison of DMA requests for capture channel 4. 0: Disable comparison of DMA requests for capture channel 4.	0
11	CC3DE	RW	Compare the DMA request enable bit of capture channel 3. 1: Allows comparison of DMA requests for capture channel 3. 0: Disable comparison of DMA requests for capture channel 3.	0
10	CC2DE	RW	Compare the DMA request enable bit of capture channel 2. 1: allows comparison of DMA requests for capture channel 2. 0: Disable comparison of DMA requests for capture channel 2.	0
9	CC1DE	RW	Compare the DMA request enable bit of capture channel 1. 1: allows comparison of DMA requests for capture channel 1. 0: Disable comparison of DMA requests for capture channel 1.	0
8	UDE	RW	Updated DMA request enable bit. 1: DMA requests that allow updates. 0: DMA requests for updates are disabled.	0
7	Reserved	RO	Reserved	0
6	TIE	RW	Trigger the interrupt enable bit. 1: Enables triggering of interrupts. 0: Trigger interrupt is disabled.	0
5	Reserved	RO	Reserved	0
4	CC4IE	RW	Compare capture channel 4 interrupt enable bit. 1: Allows comparison of capture channel 4 interrupts. 0: Disable compare capture channel 4 interrupt.	0
3	CC3IE	RW	Compare capture channel 3 interrupt enable bit. 1: Allows comparison of capture channel 3 interrupts. 0: Disable compare capture channel 3 interrupt.	0
2	CC2IE	RW	Compare capture channel 2 interrupt enable bit. 1: allows comparison of capture channel 2 interrupts. 0: Disable compare capture channel 2 interrupt.	0
1	CC1IE	RW	Compare capture channel 1 interrupt enable bit. 1: allows comparison of capture channel 1 interrupts. 0: Disable compare capture channel 1 interrupt.	0
0	UIE	RW	Update the interrupt enable bit. 1: Allowing updates to be interrupted. 0: Disable update interruption.	0

11.4.5 Interrupt Status Register (TIM2_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		TIF	Reserved	CC4IF	CC3IF	CC2IF	CC1IF	UIF

Bit	Name	Access	Description	Reset value
[15:13]	Reserved	RO	Reserved	0
12	CC4OF	WO	Compare capture channel 4 to repeat capture flag bits.	0
11	CC3OF	WO	Compare capture channel 3 to repeat capture flag bits.	0
10	CC2OF	WO	Compare capture channel 2 to repeat capture flag bits.	0
9	CC1OF	WO	The compare capture channel 1 repeat capture flag bit is used only when the compare capture channel is configured for input capture mode. This flag is set by hardware and a software write of 0 clears this bit. 1: the value of the counter is captured into the capture comparison register when the status of CC1IF has been set. 0: No duplicate captures are generated.	0
[8:7]	Reserved	RO	Reserved	0
6	TIF	WO	Trigger interrupt flag bit, when a trigger event occurs by hardware to this location bit, by software to clear. Trigger events include the detection of a valid edge at the TRGI input from a mode other than gated, or any edge in gated mode. 1: Trigger event generation. 0: No trigger event is generated.	0
5	Reserved	RO	Reserved	0
4	CC4IF	WO	Compare capture channel 4 interrupt flag bits.	0
3	CC3IF	WO	Compare capture channel 3 interrupt flag bits.	0
2	CC2IF	WO	Compare capture channel 2 interrupt flag bits.	0
1	CC1IF	WO	Compare capture channel 1 interrupt flag bits. If the compare capture channel is configured in output mode, this bit is set by hardware when the counter value matches the compare value, except in centrosymmetric mode. This bit is cleared by software. 1: The value of the core counter matches the value of compare capture register 1; 0: No match occurs. If compare capture channel 1 is configured in input mode, this bit is set by hardware when a capture event occurs and it is cleared by software or by reading the compare capture register. 1: the counter value has been captured compare capture register 1. 0: No input capture is generated.	0
0	UIF	WO	Update interrupt flag bit, this bit is set by hardware when an update event is generated and cleared by software. 1: Update interrupt generation. 0: No update event is generated. The following scenarios generate update events. If UDIS = 0, when the repeat counter value overflows or underflows. If URS = 0, UDIS = 0, when the UG bit is set, or when the counter core counter is reinitialized by software. If URS = 0, UDIS = 0, when the counter CNT is	0

			reinitialized by a trigger event.	

11.4.6 TIM2 Event Generation Register (TIM2_SWEVGR)

Offset address: 0x14

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

Bit	Name	Access	Description	Reset value
[15:7]	Reserved	RO	Reserved	0
6	TG	WO	The trigger event generation bit, which is set by software and cleared by hardware, is used to generate a trigger event. 1: Generate a trigger event, TIF is set, and the corresponding interrupts and DMAs are generated if enabled. 0: No action.	0
5	Reserved	RO	Reserved	0
4	CC4G	WO	Compare capture event generation bit 4. Generate Compare Capture Event 4.	0
3	CC3G	WO	Compare capture event generation bit 3. Generate Compare Capture Event 3.	0
2	CC2G	WO	Compare capture event generation bit 2. Generate Compare Capture Event 2.	0
1	CC1G	WO	Compare capture event generation bit 1. Generate Compare Capture Event 1. This bit is set by software and cleared by hardware. It is used to generate a compare capture event. 1: Generate a compare capture event on compare capture channel 1. If compare capture channel 1 is configured as output: set the CC1IF bit. Generate the corresponding interrupts and DMAs if they are enabled. If compare capture channel 1 is configured as input: the current core counter value is captured to compare capture register 1; set the CC1IF bit and generate the corresponding interrupts and DMAs if they are enabled. If CC1IF is already set, set the CC1OF bit. 0: No action.	0
0	UG	WO	Update event generation bit to generate an update event. This bit is set by software and is automatically cleared by hardware. 1: Initialize the counter and generate an update event. 0: No action. *Note: The prescaler counter is also cleared to zero, but the prescaler factor remains unchanged. The core counter is cleared if in centrosymmetric mode or incremental counting mode; if in decremental counting mode, the core counter takes the value of the reload value register.*	0

11.4.7 Compare/Capture Control Register 1 (TIM2_CHCTLR1)

Offset address: 0x18

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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	Compare capture channel 2 clear enable bit. 1: Clear the OC2REF bit zero once the ETRF input is detected high. 0: OC2REF is not affected by the ETRF input.	0
[14:12]	OC2M[2:0]	RW	Compare the Capture Channel 2 mode setting field. The 3 bits define the action of the output reference signal OC2REF, which determines the values of OC2, OC2N. OC2REF is active high, while the active levels of OC2 and OC2N depend on the CC2P, CC2NP bits. 000: Freeze. Comparison of the value of the capture register with the value of the comparison between the core counters does not work for OC2REF. 001: force to set to valid level. Forcing OC1REF high when the core counter has the same value as the comparison capture register 1. 010: Force to set to invalid level. Forcing OC2REF low when the value of the core counter is the same as the comparison capture register 1. 011: Flip. Flips the level of OC2REF when the core counter is the same as the value of compare capture register 1. 100: Forced to invalid level. Forces OC2REF to low. 101: Force to valid level. Force OC2REF to high. 110: PWM mode 1: When counting up, channel 2 is invalid level once the core counter is greater than the value of the compare capture register, otherwise it is valid level; when counting down, channel 2 is valid level once the core counter is greater than the value of the compare capture register, otherwise it is invalid level. 111: PWM mode 2: When counting up, channel 2 is valid level once the core counter is greater than the value of the compare capture register, otherwise it is invalid level; when counting down, channel 2 is invalid level once the core counter is greater than the value of the compare capture register, otherwise it is valid level (OC2REF=1). *Note: This bit cannot be modified once the LOCK level is set to 3 and CC2S=00b. In PWM mode 1 or PWM mode 2, the OC2REF level is changed only when the comparison result is changed or when switching from freeze mode to PWM mode in the output comparison mode.*	0
11	OC2PE	RW	Compare Capture Register 1 preload enable bit. 1: Enable the preload function of the compare capture register 1, the read and write operations operate only on the preload register, and the preload value of the compare capture register 1 is loaded into the current shadow register when the update event comes. 0: Disable the preload function of compare capture register 1. The compare capture register 1 can be	0

			written at any time, and the newly written value takes effect immediately. *Note: Once the LOCK level is set to 3 and CC2S=00, this bit cannot be modified. PWM mode can be used only in single pulse mode (OPM=1) without confirming the pre-load register, otherwise its action is not determined.*	
10	OC2FE	RW	Compare Capture Channel 2 fast enable bit, this bit is used to speed up the response of the compare capture channel output to trigger input events. 1: The active edge of the input to the flipflop acts as if a comparison match has occurred. Therefore, the OC is set to the comparison level independent of the comparison result. The delay between the valid edge of the sample trigger and the output of the compare capture channel 2 is reduced to 3 clock cycles. 0: Based on the value of the counter and compare capture register 1, compare capture channel 2 operates normally, even if the flip-flop is open. The minimum delay to activate the compare capture channel 2 output is 5 clock cycles when the input of the flipflop has a valid edge. OC2FE only works when the channel is configured to PWM1 or PWM2 mode.	0
[9:8]	CC2S[1:0]	RW	Compare capture channel 2 input selection fields. 00: comparison capture channel 2 is configured as an output. 01: comparison capture channel 2 is configured as an input and IC2 is mapped on TI2. 10: comparison capture channel 2 is configured as an input and IC2 is mapped onTI1. 11: Compare Capture Channel 2 is configured as an input and IC2 is mapped on TRC. This mode works only when the internal trigger input is selected (by the TS bit). *Note: Compare Capture Channel 2 is writable only when the channel is off (when CC2E is zero).*	0
7	OC1CE	RW	Compare capture channel 1 clear enable bit.	0
[6:4]	OC1M[2:0]	RW	Compare capture channel 1 mode setting field.	0
3	OC1PE	RW	Compare capture register 1 preload enable bit.	0
2	OC1FE	RW	Compare capture channel 1 fast enable bit.	0
[1:0]	CC1S[1:0]	RW	Compare capture channel 1 input selection fields.	0

Capture mode (pin direction is input).

Bit	Name	Access	Description	Reset value
[15:12]	IC2F[3:0]	RW	The input capture filter 2 configuration field, these bits set the sampling frequency of the TI1 input and the digital filter length. The digital filter consists of an event counter, which records N events and then generates a jump in the output. 0000: no filter, sampled at fDTS. 1000: sampling frequency Fsampling = Fdts/8, N = 6. 0001: sampling frequency Fsampling=Fck_int, N=2. 1001: sampling frequency Fsampling = Fdts/8, N = 8. 0010: sampling frequency Fsampling=Fck_int, N=4. 1010: sampling frequency Fsampling = Fdts/16, N = 5. 0011: sampling frequency Fsampling=f=Fck_int, N=8. 1011: sampling frequency Fsampling = Fdts/16, N = 6.	0

			0100: sampling frequency Fsampling = Fdts/2, N = 6. 1100: sampling frequency Fsampling = Fdts/16, N = 8. 0101: sampling frequency Fsampling = Fdts/2, N = 8. 1101: sampling frequency Fsampling = Fdts/32, N = 5. 0110: sampling frequency Fsampling = Fdts/4, N = 6. 1110: sampling frequency Fsampling = Fdts/32, N = 6. 0111: sampling frequency Fsampling = Fdts/4, N = 8. 1111: Sampling frequency Fsampling=Fdts/32, N=8.	
[11:10]	IC2PSC[1:0]	RW	Compare capture channel 2 prescaler configuration field, these 2 bits define the prescaler coefficient for compare capture channel 2. Once CC1E = 0, the prescaler is reset. 00: without prescaler, one capture is triggered for each edge detected on the capture input. 01: capture triggered every 2 events. 10: capture triggered every 4 events. 11: Capture is triggered every 8 events.	0
[9:8]	CC2S[1:0]	RW	Compare the capture channel 2 input selection field, these 2 bits define the direction of the channel (input/output), and the selection of the input pin. 00: Comparative capture channel 1 channel is configured as an output. 01: Comparison capture channel 1 channel is configured as an input and IC1 is mapped on TI1. 10: Comparison capture channel 1 channel is configured as an input and IC1 is mapped on TI2. 11: The compare capture channel 1 channel is configured as an input and IC1 is mapped on TRC. This mode works only when the internal trigger input is selected (by the TS bit). *Note: CC1S is writable only when the channel is off (CC1E is 0).*	0
[7:4]	IC1F[3:0]	RW	Input capture filter 1 configuration field.	0
[3:2]	IC1PSC[1:0]	RW	Compare the capture channel 1 prescaler configuration field.	0
[1:0]	CC1S[1:0]	RW	Compare capture channel 1 input selection fields.	0

11.4.8 Compare/Capture Control Register 2 (TIM2_CHCTLR2)

Offset address: 0x1C

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

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

Comparison mode (pin direction is output).

Bit	Name	Access	Description	Reset value
15	OC4CE	RW	Compare capture channel 4 clear enable bit.	0
[14:12]	OC4M[2:0]	RW	Compare the capture channel 4 mode setting field.	0
11	OC4PE	RW	Compare Capture Register 4 preload enable bit.	0
10	OC4FE	RW	Compare capture channel 4 fast enable bit.	0
[9:8]	CC4S[1:0]	RW	Compare capture channel 4 input selection fields.	0

7	OC3CE	RW	Compare capture channel 3 clear enable bit.	0
[6:4]	OC3M[2:0]	RW	Compare the capture channel 3 mode setting field.	0
3	OC3PE	RW	Compare Capture Register 3 preload enable bit.	0
2	OC3FE	RW	Compare capture channel 3 fast enable bit.	0
[1:0]	CC3S[1:0]	RW	Compare capture channel 3 input selection fields.	0

Capture mode (pin direction is input).

Bit	Name	Access	Description	Reset value
[15:12]	IC4F[3:0]	RW	Input capture filter 4 configuration field.	0
[11:10]	IC4PSC[1:0]	RW	Compare the capture channel 4 prescaler configuration field.	0
[9:8]	CC4S[1:0]	RW	Compare capture channel 4 input selection fields.	0
[7:4]	IC3F[3:0]	RW	Input capture filter 3 configuration field.	0
[3:2]	IC3PSC[1:0]	RW	Compare the capture channel 3 prescaler configuration field.	0
[1:0]	CC3S[1:0]	RW	Compare capture channel 3 input selection fields.	0

11.4.9 Compare/Capture Enable Register (TIM2_CCER)

Offset address: 0x20

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

Bit	Name	Access	Description	Reset value
[15:14]	Reserved	RO	Reserved	0
13	CC4P	RW	Compare the capture channel 4 output polarity setting bit.	0
12	CC4E	RW	Compare capture channel 4 output enable bit.	0
[11:10]	Reserved	RO	Reserved	0
9	CC3P	RW	Compare the capture channel 3 output polarity setting bit.	0
8	CC3E	RW	Compare capture channel 3 output enable bit.	0
[7:6]	Reserved	RO	Reserved	0
5	CC2P	RW	Compare the capture channel 2 output polarity setting bit.	0
4	CC2E	RW	Compare capture channel 2 output enable bit.	0
[3:2]	Reserved	RO	Reserved	0
1	CC1P	RW	Compare the capture channel 1 output polarity setting bit. CC1 channel configured as output: 1: OC1 active low. 0: OC1 active high. CC1 channel configured as input: This bit selects whether IC1 or the inverted signal of IC1 is used as the trigger or capture signal. 1: Inverted: capture occurs on the falling edge of IC1; when used as an external trigger, IC1 is inverted. 0: Non-inverted: capture occurs on the rising edge of IC1; when used as an external trigger, IC1 is not inverted. *Note: Once the LOCK level (LOCK bit in TIMx_BDTR register) is set to 3 or 2, this bit cannot be modified.*	0
0	CC1E	RW	Compare capture channel 1 output enable bit. The CC1 channel is configured as output:	0

			1: ON. the OC1 signal is output to the corresponding output pin, and its output level depends on the values of the MOE, OSSI, OSSR, OIS1, OIS1N, and CC1NE bits. 0: off. OC1 disables output, so the output level of OC1 depends on the values of the MOE, OSSI, OSSR, OIS1, OIS1N, and CC1NE bits. The CC1 channel is configured as an input: This bit determines whether the counter value can be captured into the TIMx_CCR1 register. 1: capture enable. 0: capture disable.	

11.4.10 Counter for General-purpose Timer (TIM2_CNT)

Offset address: 0x24

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

Bit	Name	Access	Description	Reset value
[15:0]	CNT[15:0]	RW	The real-time value of the timer's counter.	0

11.4.11 Counting Clock Prescaler (TIM2_PSC)

Offset address: 0x28

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

Bit	Name	Access	Description	Reset value
[15:0]	PSC[15:0]	RW	The dividing factor of the prescaler of the timer; the clock frequency of the counter is equal to the input frequency of the divider/(PSC+1).	0

11.4.12 Auto-reload Value Register (TIM2_ATRLR)

Offset address: 0x28

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[15:0]	RW	The value of ATRLR[15:0] will be loaded into the counter, read section 10.2.4 for when ATRLR acts and updates; the counter stops when ATRLR is empty.	0xFFFF

11.4.13 Compare/capture Register 1 (TIM2_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[15:0]	RW	Compare the value of capture register channel 1.	0

11.4.14 Compare/capture Register 2 (TIM2_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[15:0]	RW	Compare the value of capture register channel 2.	0

11.4.15 Compare/capture Register 3 (TIM2_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[15:0]	RW	Compare the value of capture register channel 3.	0

11.4.16 Compare/capture Register 4 (TIM2_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[15:0]	RW	Compare the value of capture register channel 4.	0

11.4.17 DMA Control Register (TIM2_DMACFGR)

Offset address: 0x48

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

Bit	Name	Access	Description	Reset value
[15:13]	Reserved	RO	Reserved	0
[12:8]	DBL[4:0]	RW	The length of the DMA continuous transmission, the actual value of which is the value of this field + 1.	0
[7:5]	Reserved	RO	Reserved	0
[4:0]	DBA[4:0]	RW	These bits define the offset of the DMA in continuous mode from the address where control register 1 is located.	0

11.4.18 DMA Address Register for Continuous Mode (TIM2_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[15:0]	RW	The address of the DMA in continuous mode.	0

Chapter 12 Universal Synchronous Asynchronous Receiver Transmitter (USART)

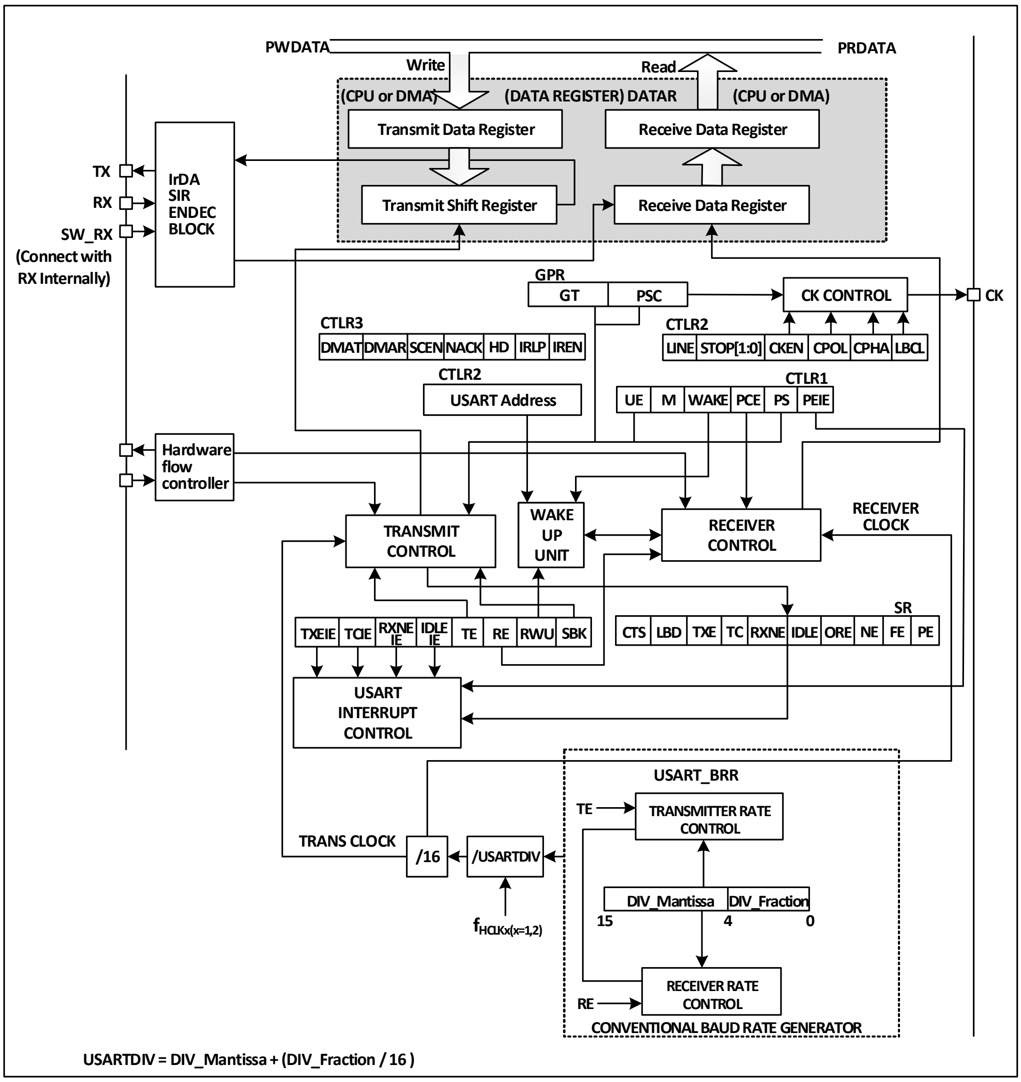
The module contains one Universal Synchronous Asynchronous Transceiver USART1.

12.1 Main Features

- Full-duplex or half-duplex synchronous or asynchronous communication
- NRZ data format
- Fractional baud rate generator, up to 3Mbps
- Programmable data length
- Configurable stop bits
- Support LIN, IrDA encoders, smart cards
- DMA support
- Multiple interrupt sources

12.2 Overview

Figure 12-1 Block diagram of a general-purpose synchronous/asynchronous transceiver



When TE (transmit enable bit) is set, the data in the transmit shift register is output on the TX pin and the clock is output on the CK pin. When transmitting, the first bit shifted out is the least significant bit and each data frame starts with a low start bit, then the transmitter sends an 8- or 9-bit data word depending on the setting on the M (word length) bit, and finally a configurable number of stop bits. If equipped with a parity check bit, the last bit of the data word is the check bit. After the TE is set an idle frame is sent, which is 10 or 11 bits high and contains the stop bit. The disconnect frame is 10 or 11 bits low followed by the stop bit.

12.3 Baud Rate Generator

The baud rate of the transceiver = HCLK/(16*USARTDIV), HCLK is the clock of AHB. The value of USARTDIV is determined by the two fields DIV_M and DIV_F in USART_BRR, which is calculated by the formula The formula is as follows.

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

It is important to note that the bit rate generated by the baud rate generator may not always generate exactly the baud rate required by the user, and there may be deviations. In addition to taking as close a value as possible, a way to reduce the deviation is to increase the AHB clock. For example, if you set the baud rate to 115200bps, the value of USARTDIV is set to 39.0625, which will give you a baud rate of exactly 115200bps at the highest frequency, but if you need a baud rate of 921600bps, the calculated USARTDIV is 4.88, but the closest value filled in USART_BRR is actually only 4.875. 4.875, the actual baud rate is 923076bps, which is 0.16% error. When the serial waveform sent by the sender is transmitted to the receiver, the baud rate of the receiver and the sender is subject to some error. The error mainly comes from three aspects: the actual baud rate of the receiver and the sender is not the same; the receiver and the sender's clock has errors; the waveform in the line generated by the change. Peripheral module receiver is a certain receiving tolerance, when the sum of the above three aspects of the total deviation is less than the module's tolerance limit, the total deviation does not affect the transmission and reception. The tolerance limitof the module is affected by whether to use fractional baud rate and M-bit (data field word length), using fractional baud rate and using 9-bit data field length will reduce the tolerance limit, but not less than 3%.

12.4 Synchronous Mode

Synchronous mode allows the system to output a clock signal when using the USART module. When synchronous mode is enabled to send data externally, the CK pin will output the clock externally at the same time.

The way to turn on the synchronous mode is to the CLKEN position bit in control register 2 (R16_USARTx_CTLR2), but also need to turn off the LIN mode, smart card mode, infrared mode and half duplex mode, i.e. ensure that the SCEN, HDSEL and IREN bits are in reset, these three in control register 3 (R16_USARTx_CTLR3).

The key point of using synchronous mode is the clock output control. There are several points to note.

a) The USART module synchronization mode works only in the main mode, i.e. the CK pin outputs only the clock and does not receive inputs.

   Outputs a clock signal only when data is output on the TX pin.

   The LBCL bit determines whether the clock is output when the last data bit is sent, the CPOL bit determines the polarity of the clock, and the CPHA determines the phase of the clock. These three bits are in control register 2 (R16_USARTx_CTLR2), which needs to be set when TE and RE are not enabled, see Figure 12-2 for the differences.

   The receiver will only sample at the output clock in synchronous mode, requiring a certain amount of signal build time and hold time from the device, as shown in Figure 12-3.

Figure 12-2 USART clock timing example (M=0)

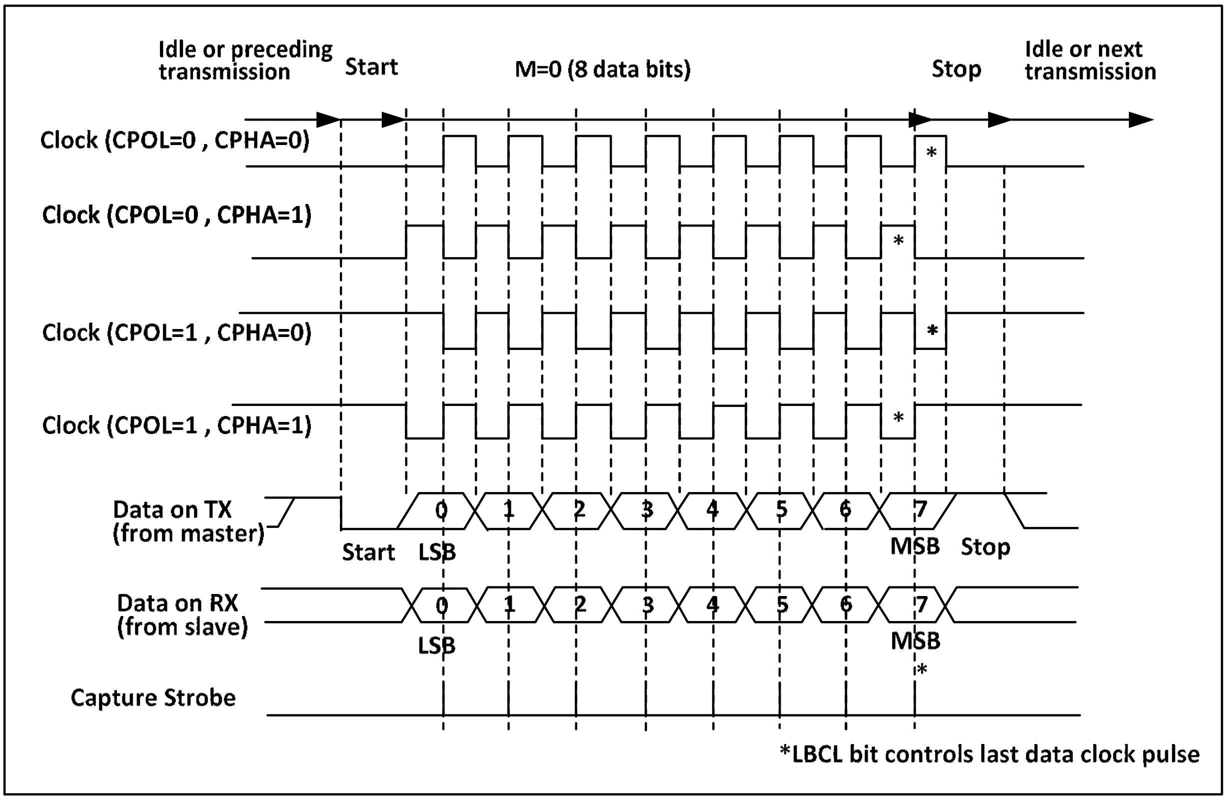
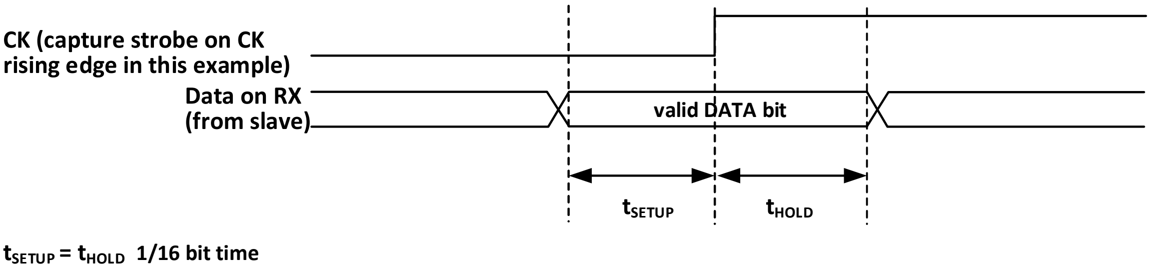


Figure 12-3 Data Sample Hold Time



12.5 1-wire Half-duplex Mode

Half-duplex mode supports the use of a single pin (TX pin only) for receive and transmit, with the TX and RX pins connected internally on the chip.

The way to turn on the half-duplex mode is to set the HDSEL position bit in control register 3 (R16_USARTx_CTLR3), but it is also necessary to turn off the LIN mode, smart card mode, IR mode and synchronous mode, i.e. to ensure that the SCEN, CLKEN and IREN bits are in reset, which are in control registers 2 and 3 (R16_USARTx_CTLR2 and R16_USARTx_CTLR3).

After setting to half duplex mode, you need to set the IO port of TX to open-drain output high mode. With TE set, the data will be sent out as soon as it is written to the data register. Special attention should be paid to the fact that the half-duplex mode may cause bus conflicts when multiple devices use a single bus to send and receive, which needs to be avoided by the user with software itself.

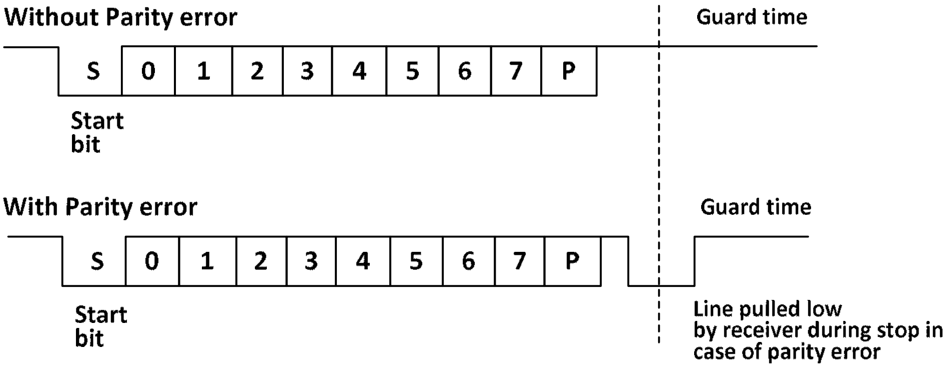
12.6 Smart Card

Smart card mode supports ISO7816-3 protocol access to smart card controllers.

The smart card mode is turned on by setting the SCEN position bit in control register 3

(R16_USARTx_CTLR3), but it is also necessary to turn off LIN mode, half duplex mode and IR mode, i.e. to ensure that the LINEN, HDSEL and IREN bits are in reset, but CLKEN can be turned on to output the clock, these bits are in control registers 2 and 3 (R16_USARTx_CTLR2 and R16_USARTx_CTLR3).
To support smart card mode, USART should be set to 8 bits of data plus 1 bit of parity, and its stop bit is recommended to be configured to 1.5 bits for both transmit and receive. Smart card mode is a 1-wire half-duplex protocol that uses the TX line for data communication and should be configured as an open-drain output plus a pull. When the receiver receives a frame of data and detects a parity error, it sends a NACK signal, i.e., it actively pulls the TX down by one cycle during the stop bit, and the sender detects the NACK signal, which generates a frame error whereby the application can retransmit. Figure 17-4 shows the waveforms on the TX pin in the correct case and in the case of a parity error. the TC flag (transmit complete flag) of the USART can delay the GT (protection time) generation by one clock, and the receiver will not recognize the NACK signal it sets as the start bit.

Figure 12-4 (Un)Occurrence of parity error diagram



In smart card mode, the waveform output from the CK pin when enabled has nothing to do with communication; it simply clocks the smart card with the value of the AHB clock followed by a five-bit settable clock division (twice the value of the PSC, up to 62 divisions).

12.7 IrDA

The USART module supports control of IrDA infrared transceivers for physical layer communication. The LINEN, STOP, CLKEN, SCEN and HDSEL bits must be cleared to use IrDA. NRZ (non-return to zero) coding is used between the USART module and the SIR physical layer (infrared transceiver) and is supported up to 115200 bps rates.
IrDA is a half-duplex protocol, if UASRT is sending data to SIR physical layer, then IrDA decoder will ignore the newly sent IR signal, if USART is receiving data from SIR, then SIR will not accept the signal from USART. the level logic of USART to SIR and SIRto USART is different. In SIR receive logic, the high level is 1 and the low level is 0, but in SIR send logic, the high level is 0 and the low level is 1.

12.8 DMA

The USART module supports DMA function, which can be used to achieve fast and continuous sending and receiving. When DMA is enabled, the DMA writes data from the set memory space to the transmit buffer when TXE is set. When using DMA to receive, each time RXNE is set, DMA transfers the data in the receive buffer to a specific memory space.

12.9 Interruptions

The USART module supports a variety of interrupt sources, including transmit data register empty (TXE), CTS, transmit complete (TC), receive data ready (RXNE), dataoverflow (ORE), line idle (IDLE), parity error (PE), disconnect flag (LBD), noise (NE), overflow for multi-buffered communication (ORT), and frame error (FE), among others.

Table 12-1 Relationship between interrupts and corresponding enable bits

Interrupt source	Enable bit
Transmit data register empty (TXE)	TXEIE
Allowed to send (CTS)	CTSIE
Transmission complete（TC）	TCIE
Received data ready to be read（RXNE）	RXNEIE
Overrun error detected（ORE）	
Idle line detected（IDLE）	IDLEIE
Parity error（PE）	PEIE
Break flag（LBD）	LBDIE
Noise flag（NE）	EIE
Overflow of multi-buffered communication (ORE)	
Frame error for multibuffered communication (FE)	

12.10 Register Description

Table 12-2 USART-related registers list

Name	Offset address	Description	Reset value
R32_USART_STATR	0x40013800	UASRT status register	0x000000C0
R32_USART_DATAR	0x40013804	UASRT data register	0x000000XX
R32_USART_BRR	0x40013808	UASRT baud rate register	0x00000000
R32_USART_CTLR1	0x4001380C	UASRT control register 1	0x00000000
R32_USART_CTLR2	0x40013810	UASRT control register 2	0x00000000
R32_USART_CTLR3	0x40013814	UASRT control register 3	0x00000000
R32_USART_GPR	0x40013818	UASRT protection time and prescaler register	0x00000000

12.10.1 USART Status Register (USART_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						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 state change flag. If the CTSE bit is set, this bit will be set high by hardware when the nCTS output state changes. It is cleared to zero by software. If the CTSIE bit is already set, an interrupt will be generated. 1: the presence of changes on the nCTS state line. 0: No change on the nCTS state line.	0
8	LBD	RW0	LIN Break detection flag. This bit is set by hardware when a LIN Break is detected. It is cleared by software.	0

			If LBDIE is already set, an interrupt will be generated. 1: LIN Break detected. 0: No detection of pending LIN Break.	
7	TXE	RO	Send data register empty flag. This bit is set by hardware when the data in the TDR register is transferred to the shift register by hardware. If TXEIE is already set, an interrupt will be generated to perform a write operation to the data register and this bit will be reset. 1: the data has been transferred to the shift register. 0: The data has not been transferred to the shift register.	1
6	TC	RW0	Send completion flag. When a frame containing data is sent and TXE is set, the hardware will set this bit, and if TCIE is set, an interrupt will be generated, and the software will clear this bit by reading it and then writing to the data register. It is also possible to write 0 directly to clear this bit. 1: Sending completed. 0: Sending is not yet complete.	1
5	RXNE	RW0	Read data register non-empty flag, this bit is set by hardware when data in the shift register is transferred to the data register. If RXNEIE is already set, a corresponding interrupt is also generated. A read operation of the data register clears this bit. It is also possible to clear the bit by writing a 0 directly. 1: Data received and able to be read out. 0: The data has not been received.	0
4	IDLE	RO	Bus idle flag. When the bus is idle, this bit will be set by hardware. If IDLEIE is already set, the corresponding interrupt will be generated. The operation of reading the status register and then reading the data register will clear this bit. 1: The bus is idle. 0: No bus idle is detected. *Note: This bit will not be set again until RXNE is set.*	0
3	ORE	RO	Overload error flag. This bit will be set when there is data in the receive shift register that needs to be transferred to the data register, but there is still data in the receive field of the data register that has not been read out. If RXNEIE is set, the corresponding interrupt will also be generated. 1: Occurrence of an overload error. 0: No overload error. *Note: In case of an overload error, the value of the data register is not lost, but the value of the shift register is overwritten. If the EIEable bit is set, the ORE flag position bit generates an interrupt in multi-buffer communication mode.*	0
2	NE	RO	Noise error flag. It is set by hardware when the noise error flag is detected. The operation of reading the status register and then reading the data register resets this bit. 1: Noise detected. 0: No noise is detected. *Note: This bit does not generate an interrupt. If the EIE bit is set, the FE flag position*	0

			bit generates an interrupt in multi-buffer communication mode.	
1	FE	RO	Frame error flag. This bit will be set by hardware when a synchronization error, excessive noise or disconnect character is detected. Reading this bit and then reading the data register operation will reset this bit. 1: Frame error detected. 0: No frame error detected. *Note: This bit will not generate an interrupt. If the EIE bit is set, the FE flag position bit will generate an interrupt in multi-buffer communication mode.*	0
0	PE	RO	Checksum error flag. In receive mode, hardware sets this bit if a parity check error is generated. A read of this bit and then a read of the data register operation resets this bit. Before clearing this bit, software must wait for the RXNE flag bit to be set. If the PEIE has been set previously, then this bit being set generates a corresponding interrupt. 1: A parity error. 0: No inspection error.	0

12.10.2 USART Data Register (USART_DATAR)

Offset address: 0x04

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

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

Bit	Name	Access	Description	Reset value
[31:9]	Reserved	RO	Reserved	0
[8:0]	DR[8:0]	RW	Data register. This register is actually the receive data register (RDR) and send register (TDR) two registers composed of DR read and write operation start is read receive register (RDR) and write send register (TDR) respectively.	X

12.10.3 USART Baud Rate Register (USART_BRR)

Offset address: 0x08

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

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

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:4]	DIV_ Mantissa[11:0]	RW	These 12 bits define the integer part of the dividing factor of the frequency divider.	0
[3:0]	DIV_ Fraction[3:0]	RW	These 4 bits define the fractional part of the dividing factor of the frequency divider.	0

12.10.4 USART Control Register 1 (USART_CTLR1)

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, both the USART divider and the output stop working after the current byte transfer is completed.	0
12	M	RW	Word long bit. 1: 9 data bits; 0: 8 data bits.	0
11	WAKE	RW	Wake-up bit. This bit determines the method of waking up the USART. 1: Address marker; 0: Bus idle.	0
10	PCE	RW	The parity bit is enabled. For the receiver, it is the parity check of the data; for the sender, it is the insertion of the parity bit. Once this bit is set, the parity bit enable will take effect only after the current byte transmission is completed.	0
9	PS	RW	Parity selection. 0 means even parity, 1 means odd parity. When this bit is set, the parity bit enable will take effect only after the current byte transmission is completed.	0
8	PEIE	RW	Parity check interrupt enable bit. This bit indicates that parity check error interrupts are allowed.	0
7	TXEIE	RW	TXE interrupt enable. This bit indicates that a TXE interrupt is allowed to be generated.	0
6	TCIE	RW	Transmit completion interrupt enable. This bit indicates that the transmit completion interrupt is allowed to be generated.	0
5	RXNEIE	RW	RXNE interrupt enable. This bit indicates that a RXNE interrupt is allowed to be generated.	0
4	IDLEIE	RW	IDLE interrupt enable. This bit allows IDLE interrupt to be generated.	0
3	TE	RW	Transmitter enable. Setting this bit will enable the transmitter.	0
2	RE	RW	Receiver enable. Setting this bit enables the receiver, which starts detecting the start bit on the RX pin.	0
1	RWU	RW	Receiver wakeup. This bit determines whether to place the USART in silent mode. 1: The receiver is in silent mode. 0: The receiver is in normal operation mode. *Note 1: Before setting the RWU bit, the USART needs to receive a data byte first, otherwise it cannot be woken up by bus idle in silent mode.* *Note 2: When configured as address mark wake-up, the RWU bit cannot be modified by software when RXNE is set.*	0
0	SBK	RW	Send break bit. Set this bit to send break character. It is reset by hardware on the stop bit of the break frame.	0

			1: Send; 0: Do not send.	

12.10.5 USART Control Register 2 (USART_CTLR2)

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[1:0]		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, set to enable LIN mode. The LIN mode enables the capability to send LIN Synch Breaks using the SBK bit in the USART_CR1 register, and to detect LIN Sync breaks.	0
[13:12]	STOP[1:0]	RW	STOP bits. These bits are used for programming the stop bits. 00: 1 Stop bit 01: 0.5 Stop bit 10: 2 Stop bits 11: 1.5 Stop bit	0
11	CLKEN	RW	Clock enable. This bit allows the user to enable the CK pin. 0: CK pin disabled 1: CK pin enabled	0
10	CPOL	RW	Clock polarity This bit allows the user to select the polarity of the clock output on the CK pin in synchronous mode. It works in conjunction with the CPHA bit to produce the desired clock/data relationship 0: Steady low value on CK pin outside transmission window. 1: Steady high value on CK pin outside transmission window. *Note: This bit cannot be modified after enabling transmit.*	0
9	CPHA	RW	Clock phase This bit allows the user to select the phase of the clock output on the CK pin in synchronous mode. It works in conjunction with the CPOL bit to produce the desired clock/data relationship 0: The first clock transition is the first data capture edge. 1: The second clock transition is the first data capture edge. *Note: This bit cannot be modified after enabling transmit.*	0
8	LBCL	RW	Last bit clock pulse This bit allows the user to select whether the clock pulse associated with the last data bit transmitted (MSB) has to be output on the CK pin in synchronous mode. 0: The clock pulse of the last data bit is not output to the CK pin	0

			1: The clock pulse of the last data bit is output to the CK pin *Note: This bit cannot be modified after enabling transmit.*	
7	Reserved	RW	Reserved	0
6	LBDIE	RW	LIN Break detection interrupt enable, this position bit enables interrupts caused by LBD.	0
5	LBDL	RW	LIN Break detection length, this bit is used to select whether the Break character detection is 11 bits or 10 bits. 1: 11-bit Break character detection. 0: 10-bit Break character detection.	0
4	Reserved	RW	Reserved	0
[3:0]	ADD[3:0]	RW	Address of the USART node, this bit-field gives the address of the USART node. This is used in multiprocessor communication during mute mode, for wake up with address mark detection.	0

12.10.6 USART Control Register 3 (USART_CTLR3)

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	CTS interrupt enable bit, when this bit is set, an interrupt will be generated when CTS is set.	0
9	CTSE	RW	CTS enable bit, setting this bit will enable CTS flow control.	0
8	RTSE	RW	RTS enable bit, setting this bit will enable RTS flow control.	0
7	DMAT	RW	DMA transmit enable bit. This bit 1 uses DMA when transmitting.	0
6	DMAR	RW	DMA receive enable bit. This position 1 uses DMA on receive.	0
5	SCEN	RW	Smartcard mode enable bit, set to 1 to enable smart card mode.	0
4	NACK	RW	Smartcard NACK enable bit, set this bit to send NACK in case of check error.	0
3	HDSEL	RW	Half-duplex selection bit, set this bit to select half-duplex mode.	0
2	IRLP	RW	IrDA low-power bit, set this bit to enable low-power mode when IrDA is selected.	0
1	IREN	RW	IrDA enable bit, set this bit to enable infrared mode.	0
0	EIE	RW	Error interrupt enable bit, when set, generates an interrupt if FE, ORE or NE is set provided that DMAR is set.	0

12.10.7 USART Guard Time and Prescaler Register (USART_GPR)

Offset address: 0x18

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

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:8]	GT[7:0]	RW	Guard time value. This bit-field gives the Guard time value in terms of number of baud clocks. This is used in Smartcard mode. The Transmission Complete flag is set after this guard time value.	0
[7:0]	PSC[7:0]	RW	Prescaler value field. In IrDA Low-power mode, the source clock is divided by this value (all 8 bits valid), with a value of 0 indicating retention. In normal IrDA mode, this bit can only be set to 1. In smartcard mode, the source clock is divided by twice this value (valid in the lower 5 bits) to clock the smart card, with a value of 0 indicating retention.	0

Chapter 13 Inter-integrated Circuit (I2C) interface

The Internal Integrated Circuit Bus (I2C) is widely used for communication between microcontrollers and sensors and other off-chip modules, it supports multi-master and multi-slave modes, and can communicate at 100KHz (standard) and 400KHz (fast) using only two lines (SDA and SCL). Timing and DMA, with CRC checksum function.

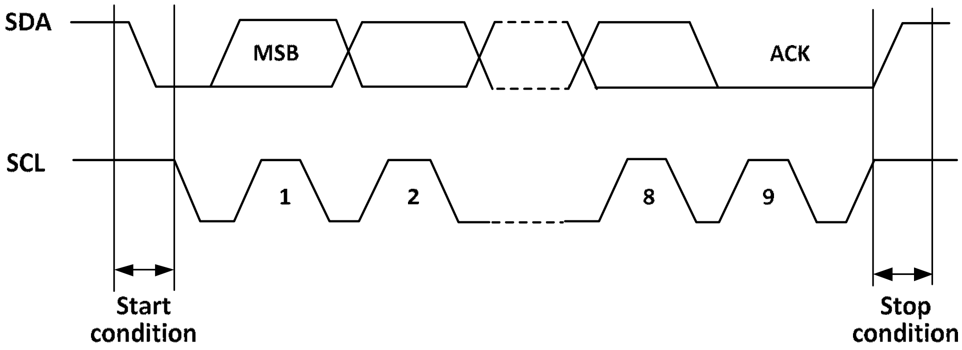
13.1 Main Features

- Support master and slave modes
- Support 7-bit or 10-bit addresses
- Slave devices support dual 7-bit addresses
- Support two speed modes: 100KHz and 400KHz
- Multiple status modes, multiple error flags
- Support extended clock function
- 2 interrupt vectors
- DMA support
- Support PEC
- SMBus compatible

13.2 Overview

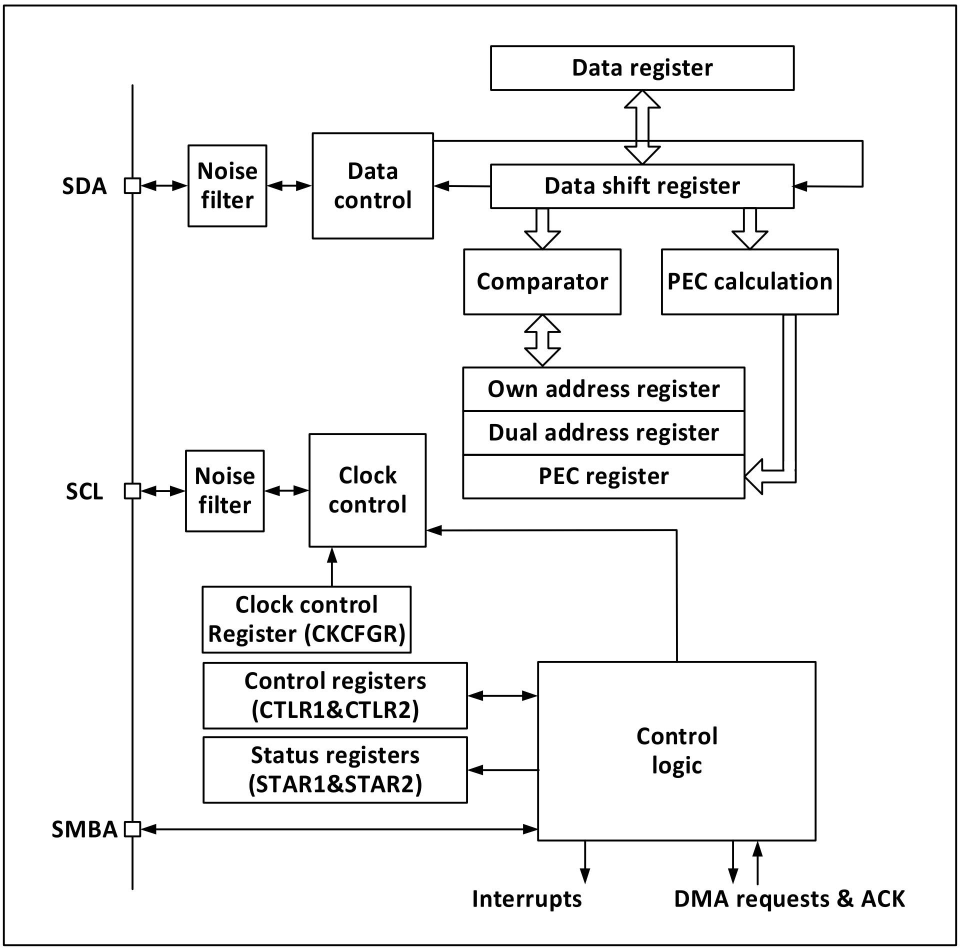
I2C is a half-duplex bus that can only operate in one of the following four modes at the same time: master device transmit mode, master device receive mode, slave device transmit mode and slave device receive mode. the I2C module works in slave mode by default and automatically switches to master mode when a start condition is generated and to slave mode when arbitration is lost or a stop signal is generated. the I2C module supports multi-master functionality. When working in master mode, the I2C module actively emits data and addresses. Both data and address are transmitted in 8-bit units, with the high bit before and the low bit after. After the start event is a one-byte (in 7-bit address mode) or two-byte (in 10-bit address mode) address, and for every 8-bit data or address sent by the host, the slave needs to reply with an answer ACK, which pulls the SDA bus low, as shown in Figure 13-1.

Figure 13-1 I2C Timing Diagram



In order to work properly the I2C must be fed with the correct clock, which is a minimum of 2MHz in standard mode and 4MHz in fast mode.

Figure 13-2 shows the functional block diagram of the I2C module.



13.3 Master Mode

In master mode, the I2C module dominates the data transfer and outputs the clock signal, and the data transfer starts with a start event and ends with an end event. The steps to use master mode communication are.

Setting the correct clock in control register 2 (R16_I2Cx_CTLR2) and clock control register (R16_I2Cx_CKCFGR).

Setting the appropriate rising edge in the rising edge register (R16_I2Cx_RTR).

Setting the PE bit in the control register (R16_I2Cx_CTLR1) to start the peripheral.

Set the START bit in the control register (R16_I2Cx_CTLR1) to generate the start event.

After setting the START bit, the I2C module will automatically switch to the main 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 and if the ITEVTEN bit (in R16_I2Cx_CTLR2) is set, an interrupt will be generated. The status register 1 (R16_I2Cx_STAR1) should be read at this time and the SB bit will be cleared automatically after writing from the address to the data register.

If the 10-bit address mode is used, then the write data register sends the header sequence (the header sequence is 11110xx0b, where the xx bits are the top two bits of the 10-bit address). After sending the header sequence, the ADD10 bit of the status register will be set, and if the ITEVTEN bit has been set, an interrupt will be generated, at this time the R16_I2Cx_STAR1 register should be read and the ADD10 bit cleared after writing the second address byte to the data register.

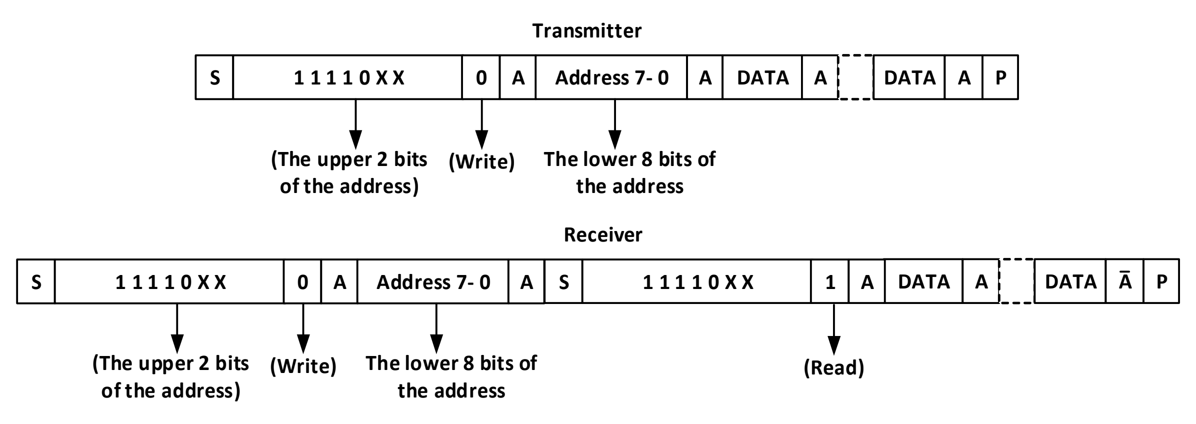
Then write the data register to send the second address byte, after sending the second address byte, the ADDR bit of the status register will be set, if the ITEVTEN bit is already set, an interrupt will be generated, at this time the R16_I2Cx_STAR1 register should be read and then read the R16_I2Cx_STAR2 register once to clear the ADDR bit;

If the 7-bit address mode, then write data register to send address byte, after sending address byte, ADDR bit of status register will be set, if ITEVTEN bit has been set, then interrupt will be generated, at this time, R16_I2Cx_STAR1 register should be read and then R16_I2Cx_STAR2 register should be read once to clear ADDR bit;

In 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, 0 means the master device writes data to the slave device, 1 means the master device reads information to the slave device.

In 10-bit address mode, as shown in Figure 13-3, in the send address phase, the first byte is 11110xx0, xx is the highest 2 bits of the 10-bit address, and the second byte is the lower 8 bits of the 10-bit address. If subsequently enter the master device transmit mode, continue to send data; if subsequently ready to enter the master device receive mode, you need to re-send a start condition, follow to send a byte as 11110xx1, and then enter the master device receive mode.

Figure 13-3 Schematic diagram of master sending and receiving data at 10-bit address

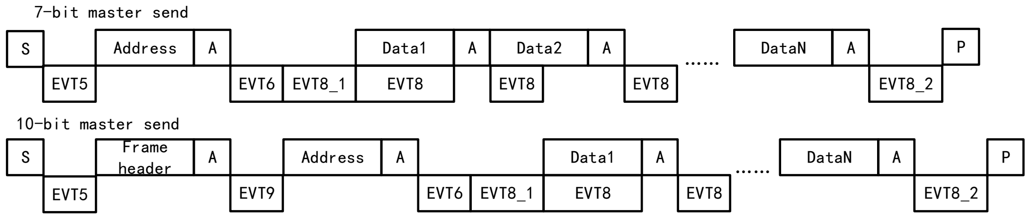


Master transmit mode:

The master device's internal shift register sends data from the data register to the SDA line. When the master device receives an ACK, TxE in status register 1 (R16_I2Cx_STAR1) is set, and an interrupt is also generated if ITEVTEN and ITBUFEN are set. Writing data to the data register will clear the TxE bit.

If the TxE bit is set and no new data was written to the data register before the last data was sent, then the BTF bit will be set and SCL will remain low until it is cleared, and writing data to the data register after reading R16_I2Cx_STAR1 will clear the BTF bit.

Figure 13-4 Master transmitter transmission sequence diagram



Description: S=Start (start condition), Sr=repeated start condition, P=Stop (stop condition), A=response, NA=non-response, EVTx=event (interrupt generated when ITEVFEN=1)

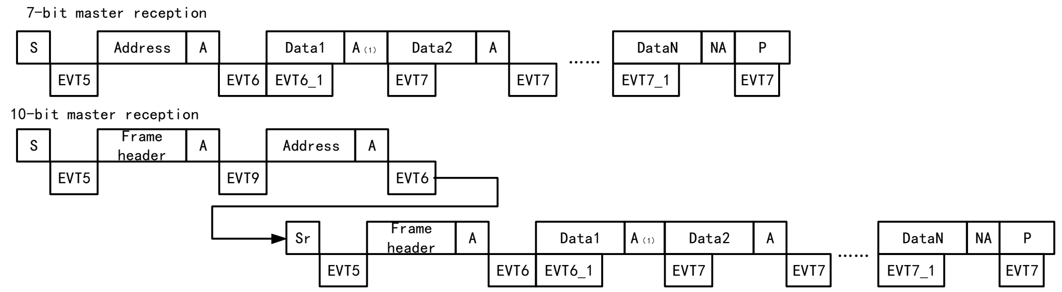
EVT5: SB=1, reading SR1 and then writing the address to the DR register will clear the event.
EVT6;ADDR=1,reading SR1 then reading SR2 will clear the event.
EVT8_1: TxE=1, shift register empty, data register empty, write DR register.
EVT8: TxE=1,shift register is not empty,data register is empty,writing DR register will clear the event.
EVT8_2: TxE=1, BTF=1, request to set the stop bit. the TxE and BTF bits are cleared by hardware when the stop condition is generated.
EVT9: ADDR10=1, reading SR1 and then writing to DR register will clear the event.

Note: 1: EVT5, EVT6, EVT9, EVT8_1 and EVT8_2 events elongate the SCL low until the end of the corresponding software sequence.
2: The software sequence of EVT8 must be completed before the end of the current byte transfer.

Master receive mode:

The I2C module will receive data from the SDA line and write it into the data register via a shift register. After each byte, if the ACK bit is set, then the I2C module will send an answer low, and the RxNE bit will be set, and an interrupt will be generated if ITEVTEN and ITBUFEN are set. If RxNE is set and the original data is not read before the new data is received, the BTF bit will be set and SCL will remain low until the BTF is cleared, and reading R16_I2Cx_STAR1 and then reading the data register will clear the BTF bit.

Figure 13-5 Receiver transmission sequence diagram



Description: S=Start (start condition), Sr=repeated start condition, P=Stop (stop condition), A=response, NA=non-response, EVTx=event (interrupt generated when ITEVFEN=1)

EVT5: SB=1, reading SR1 and then writing the address to DR register will clear the event.
EVT6: ADDR=1,reading SR1 and then reading SR2 will erase this event. In 10-bit master receive mode, START=1 of CR2 should be set after this event.
EVT6_1: There is no corresponding event flag and it is only suitable for receiving 1 byte. Exactly after EVT6 (i.e. after ADDR is cleared), the response and stop condition generation bits should be cleared.
EVT7: RxNE=1, read DR register to clear the event.
EVT7_1: RxNE=1, read the DR register to clear this event. Set ACK=0 and STOP request.
EVT9: ADDR10=1, reading SR1 and then writing to DR register will clear this event.

When the master device ends sending data, it will actively send an end event, i.e. set the STOP bit, and the I2C will switch to slave mode. In receive mode, the master device needs to NAK at the answer position of the last data bit, and after receiving NACK, the slave device releases control of the SCL and SDA lines; the master device can then send a stop/restart condition. Note that the I2C module will automatically switch to slave mode after the stop condition is generated.

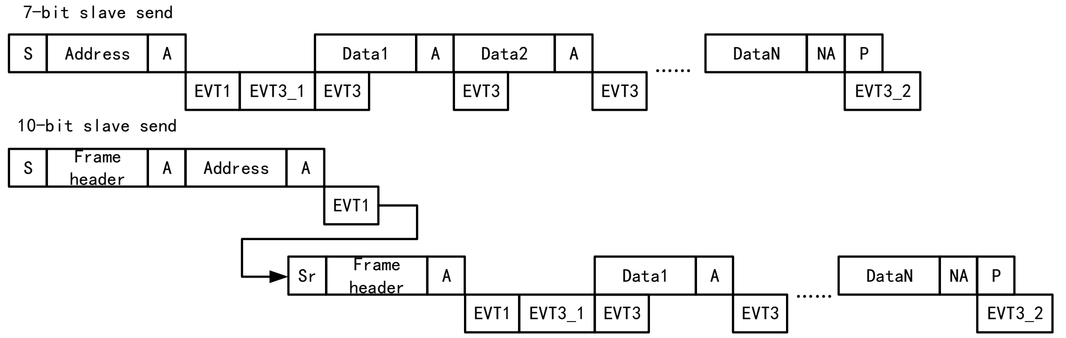
13.4 Slave Mode

When in slave mode, the I2C module recognizes its own address and the broadcast call address. The software can control whether the recognition of the broadcast call address is enabled or disabled. Once a start event is detected, the I2C module compares the SDA data through the shift register with its own address (number of bits depends on ENDUAL and ADDMODE) or the broadcast address (when ENGC is set), if there is a mismatch it will be ignored until a new start event is generated. If it matches the header sequence, an ACK signal is generated and the address of the second byte is waited for; if the address of the second byte also matches or the full segment address matches in the case of a 7- bit address, then:
first an ACK answer is generated;
the ADDR bit is set, and if the ITEVTEN bit is already set, then a corresponding interrupt is also generated;
if the dual address mode is used (ENDUAL bit is set), the DUALF bit also needs to be read to determine which address the host is evoking.
The slave mode is receive mode by default. In case 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 go to transmitter mode and the TRA bit will indicate whether it is currently receiver or transmitter mode.

Slave transmit mode:
After clearing the ADDR bit, the I2C module sends bytes from the data register to the SDA line via a shift register. After an answer ACK is received, the TxE bit is set and an interrupt is generated if ITEVTEN and ITBUFEN are set. If TxE is set but no new data is written to the data register before the end of the next data send, the BTF bit will be set. SCL will remain low until the BTF is cleared. Reading status register 1

(R16_I2Cx_STAR1) and then writing data to the data register will clear the BTF bit.

Figure 13-6 Slave transmitter transmission sequence diagram



Description: S=Start (start condition), Sr=repeated start condition, P=Stop (stop condition), A=response, NA=non-response, EVTx=event (interrupt is generated when ITEVFEN=1)

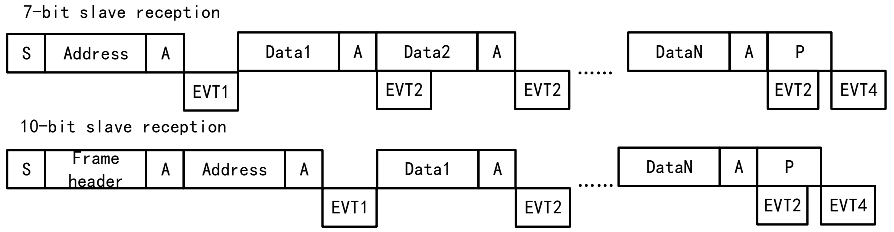
EVT1;ADDR=1,read SR1 then read SR2 will eliminate the event.
EVT3_1: TxE=1, shift register empty, data register empty, write DR.
EVT3: TxE=1,shift register is not empty,data register is empty,writing DR will clear the event.
EVT3_2: AF=1, write '0' in AF bit of SR1 register to clear AF bit.

Note: 1: EVT1 and EVT3_1 events elongate SCL low until the end of the corresponding software sequence.
2: The software sequence of EVT3 must be completed before the end of the current byte transfer.

Slave receive mode:

After ADDR is cleared, the I2C module stores the data on SDA into the data register via the shift register. After each byte is received, the I2C module sets an ACK bit and sets the RxNE bit, and generates an interrupt if ITEVTEN and ITBUFEN are set. If RxNE is set and the old data is not read before the new data is received, then BTF is set. SCL will remain low until the BTF bit is cleared. Reading status register 1 (R16_I2Cx_STAR1) and reading the data in the data register will clear the BTF bit.

Figure 13-7 Receiver transmission sequence diagram



Description: S=Start (start condition), Sr=repeated start condition, P=Stop (stop condition), A=response, NA=non-response, EVTx=event (interrupt generated when ITEVFEN=1)
EVT1;ADDR=1,reading SR1 then SR2 will erase the event.
EVT2: RxNE=1, reading DR will clear the event.
EVT4: STOPF=1,reading SR1 and then writing CR1 register will clear the event.

Note: 1: EVT1 event elongates SCL low until the end of the corresponding software sequence.
2: The software sequence of EVT2 must be completed before the end of the current byte transfer.

The master device will generate a stop condition after the last data byte is transferred. When the I2C module detects a stop event, it will set the STOPF bit, and if the ITEVFEN bit is set, it will also generate an interrupt. The user needs to read the status register (R16_I2Cx_STAR1) and then write the control register (e.g. reset control word SWRST) to clear it. (See EVT4 in the figure above).

13.5 Error Conditions

13.5.1 Bus Error (BERR)

A bus error will be generated when the I2C module detects an external start or stop event during address or data transfer. When a bus error is generated, the BERR bit is set and an interrupt is generated if ITERREN is set. In slave mode, the data is discarded and the hardware releases the bus. If it is a start signal, the hardware assumes it is a restart signal and starts waiting for an address or stop signal; if it is a stop signal, it operates ahead of normal stop conditions. In master mode, the hardware does not release the bus while not affecting the current transfer, and it is up to the user code to decide whether to abort the transfer.

13.5.2 Acknowledge Failure (AF)

An answer error will be generated when the I2C module detects a byte and then no answer. When an answer error is generated: AF will be set and an interrupt will be generated if ITERREN is set; when an AF error is encountered, the hardware must release the bus if the I2C module is working in slave mode and the software must generate a stop event if it is in master mode.

13.5.3 Arbitration Lost (ARLO)

An arbitration lost error is generated when the I2C module detects an arbitration lost. When an arbitration loss error is generated: the ARLO bit is set and an interrupt is generated if ITERREN is set; the I2C module switches to slave mode and no longer responds to transfers initiated against its slave address unless a new start event is initiated by the host; the hardware releases the bus.

13.5.4 Overrun/underrun Error (OVR)

- **Overrun error**

In Slave mode, if the clock extension is disabled and the I2C module is receiving data, an overrun error will occur if a byte of data has been received but the last received data has not been read out. When an overrun error occurs, the last received byte will be discarded and the sender should retransmit the last sent byte.

- **Underrun error**

In Slave mode, if the clock is forbidden to extend and the I2C module is sending data, an underrun error will occur if new data has not been written to the data register before the next byte of the clock comes. In case of an underrun error, the data in the previous data register will be sent twice, and if an underrun error occurs, then the receiver should discard the data received repeatedly. In order not to generate an underrun error, the I2C module should write the data to the data register before the first rising edge of the next byte.

13.6 Clock Extension

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

- In transmit mode, if TxE is set and BTF is set, SCL will always be low, always waiting for the user to read the status register and write the data to be sent to the data register.
- In receive mode, if RxNE is set and BTF is set, SCL will remain low after data is received 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.

13.7 SMBus

SMBus is also a 2-wire interface, which is generally used between system and power management. SMBus and I2C have many similarities, for example, SMBus uses the same 7-bit address mode as I2C, and the following are common to SMBus and I2C.

1) Master-slave communication mode, where the host provides the clock and supports multiple masters and slaves.
2) 2-wire communication architecture, with an optional warning line for SMBus.
3) Both support 7-bit address format.

There are also differences between SMBus and I2C.

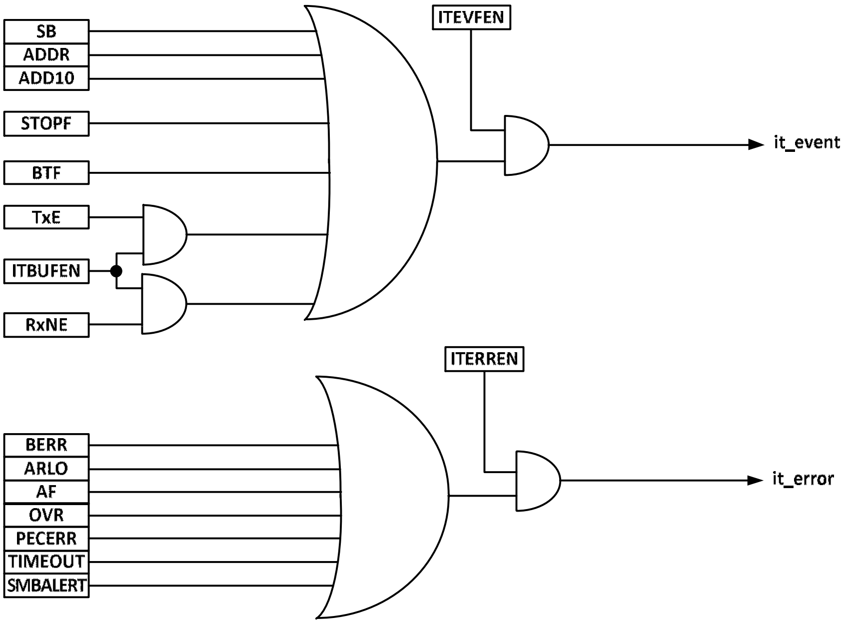
1) I2C supports speeds up to 400 KHz, while SMBus supports up to 100 KHz, and SMBus has a minimum speed limit of 10 KHz.
2) A timeout will be reported when the SMBus clock is low for more than 35mS, but there is no such limit for I2C.
3) SMBus has a fixed logic level, while I2C does not, depending on VDD.
4) SMBus has a bus protocol, while I2C does not.

SMBus also includes device identification, address resolution protocols, unique device identifiers, SMBus reminders and various bus protocols as described in the SMBus specification version 2.0. When using SMBus, only the SMBus bit of the control register needs to be set, and the SMBTYPE bit and ENAARP bit need to be configured as needed.

13.8 Interruptions

Each I2C module has two interrupt vectors, event interrupts and error interrupts. Both interrupts support the interrupt sources in Figure 13-4.

Figure 13-4 I2C Interrupt Request



13.9 DMA

DMA can be used to send and receive bulk data. The ITBUFEN bit of the control register cannot be set when using DMA.

- Transmission using DMA

DMA mode can be activated by setting the DMAEN bit of the CTLR2 register. As long as the TxE bit is set, data will be loaded by DMA from the set memory into the data register of the I2C. The following settings are required to allocate channels for I2C.

1) Set the I2Cx_DATAR register address to the DMA_PADDRx register and the memory address in the DMA_MADDRx register so that after each TxE event, data will be sent from memory to the

I2Cx_DATAR register.

2) Set the required number of bytes to be transferred in the DMA_CNTRx register. This value will be decremented after each TxE event.
3) Configure the channel priority using the PL[0:1] bits in the DMA_CFGRx register.
4) Set the DIR bit in the DMA_CFGRx register and depending on the application requirements can be configured to issue an interrupt request when the entire transfer is half or fully completed.
5) Activate the channel by setting the EN bit on the DMA_CFGRx register.

When the number of data transfer bytes set in the DMA controller has been completed, the DMA controller sends an end of transfer EOT/ EOT_1 signal to the I2C interface. A DMA interrupt will be generated if the interrupt is allowed.

- Reception using DMA

DMA receive mode can be performed after setting DMAEN in the CTLR2 register. When using DMA receive, DMA transfers the data in the data register to the preset memory area. The following steps are required to allocate channels for I2C.

1) Set the I2Cx_DATAR register address to the DMA_PADDRx register and the memory address in the DMA_MADDRx register so that after each RxNE event, data will be written to memory from the I2Cx_DATAR register.
2) Set the required number of bytes to be transferred in the DMA_CNTRx register. This value will be decremented after each RxNE event.
3) Configure the channel priority with PL[0:1] in the DMA_CFGRx register.
4) The DIR bit in the DMA_CFGRx register is cleared, and depending on the application requirements, an interrupt request can be set to be issued when the data transfer is half or fully completed.
5) Set the EN bit in the DMA_CFGRx register to activate the channel.

When the number of data transfers set in the DMA controller has been completed, the DMA controller sends an end of transfer EOT/EOT_1 signal to the I2C interface. A DMA interrupt will be generated if the interrupt is allowed.

13.10 Packet Error Checking

Packet Error Checksum (PEC) is an additional CRC8 checksum step to provide transmission reliability, calculated for each bit of serial data using the following polynomial.

$$C=X^8+X^2+X+1$$

The PEC calculation is activated by the ENPEC bit in the control register and is performed on all information bytes, including address and read/write bits. In transmitting, enabling PEC adds a byte of CRC8 calculation result after the last byte of data; while in receiving mode, in the last byte is considered as CRC8 check result, and if it does not match with the internal calculation result, it will reply a NAK, and in case of the main receiver, regardless of the correct check result.

13.11 Register Description

Table 13-1 I2C-related registers list

Name	Offset address	Description	Reset value
R16_I2C_CTLR1	0x40005400	I2C control register 1	0x0000
R16_I2C_CTLR2	0x40005404	I2C control register 2	0x0000
R16_I2C_OADDR1	0x40005408	I2C address register 1	0x0000
R16_I2C_OADDR2	0x4000540C	I2C address register 2	0x0000
R16_I2C_DATAR	0x40005410	I2C data register	0x0000
R16_I2C_STAR1	0x40005414	I2C status register 1	0x0000
R16_I2C_STAR2	0x40005418	I2C status register 2	0x0000
R16_I2C_CKCFGR	0x4000541C	I2C clock register	0x0000

13.11.1 I2C Control Register 1(I2C1_CTLR1)

Offset address: 0x00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWRST	Reserved		PEC	POS	ACK	STOP	START	NOSTRETCH	ENGC	ENPEC	Reserved				PE

Bit	Name	Access	Description	Reset value
15	SWRST	RW	Software reset, setting this bit by user code will reset the I2C peripheral. Make sure the pins of the I2C bus are released and the bus is idle before the reset. *Note: This bit resets the I2C module when no stop condition is detected on the bus but the busy bit is 1.*	0
[14:13]	Reserved	RO	Reserved	0
12	PEC	RW	Packet error checking bit, set this bit to enable packet error detection. The user code can set or clear this bit; the hardware clears this bit when the PEC is transmitted, when a start or end signal is generated, or when the PE bit is cleared to 0. 1: With PEC. 0: Without PEC. *Note: The PEC is invalidated when arbitration is lost.*	0
11	POS	RW	ACK and PEC position setting bits, which can be set or cleared by user code and can be cleared by hardware after the PE has been cleared. 1: 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 the PEC. 0: The ACK bit controls the ACK or NAK of the byte currently being accepted in the shift register. the PEC bit indicates that the byte in the shift register before the current bit is PEC. *Note: The POS bit is used in 2-byte data reception as follows: it must be configured before reception. In order to NACK the 2nd byte, the ACK bit must be cleared immediately after clearing the ADDR bit; in order to detect the PEC of the second byte, the PEC bit must be set after the ADDR event and after configuring the POS bit.*	0
10	ACK	RW	Acknowledge enable, This bit is set and cleared by software and cleared by hardware when PE=0. 1：Acknowledge returned after a byte is received. 0：No acknowledge returned.	0
9	STOP	RW	Stop generation bit. This bit is set and cleared by software, cleared by hardware when a Stop condition is detected, set by hardware when a timeout error is detected. In Master mode: 1：Stop generation after the current byte transfer or after the current Start condition is sent. 0：No Stop generation. In Slave mode: 1：Release the SCL and SDA lines after the current byte transfer.	0

			0：No Stop generation.	
8	START	RW	Start generation. This bit is set and cleared by software and cleared by hardware when start is sent or PE=0. In Master mode: 1：Repeated start generation 0：No Start generation In Slave mode: 1：Start generation when the bus is free 0：No Start generation	0
7	NOSTRETCH	RW	Clock stretching disable bit. This bit is used to disable clock stretching in slave mode when ADDR or BTF flag is set, until it is reset by software. 1：Clock stretching disabled. 0：Clock stretching enabled.	0
6	ENGC	RW	General call enable bit. Set this bit to enable broadcast call and answer broadcast address 00h.	0
5	ENPEC	RW	PEC enable bit, set this bit to enable PEC calculation.	0
[4:1]	Reserved	RO	Reserved	0
0	PE	RW	I2C peripheral enable bit. 1: Enable the I2C module. 0: Disable the I2C module.	0

13.11.2 I2C Control Register 2(I2C1_CTLR2)

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	DMA last transfer bit. 1：Next DMA EOT is the last transfer. 0：Next DMA EOT is not the last transfer. *Note: This bit is used in master receiver mode to permit the generation of a NACK on the last received data.*	0
11	DMAEN	RW	DMA requests enable bit. Set this bit to allow DMA request when TxE or RxEN is set.	0
10	ITBUFEN	RW	Buffer interrupt enable bit. 1：When TxE or RxEN is set, event interrupt is generated. 0：When TxE or RxEN is set, no interrupt is generated.	0
9	ITEVTEN	RW	Event interrupt enable bit. Set this bit to enable event interrupt. This interrupt will be generated under the following conditions. SB=1 (Master mode). ADDR=1 (Master-slave mode). ADDR10 = 1 (Master mode). STOPF=1 (Slave mode). BTF = 1, but no TxE or RxEN events. TxE event to 1 if ITBUFEN = 1. RxNE event to 1if ITBUFEN = 1.	0

8	ITERREN	RW	Error interrupt enable bit. Set to allow error interrupts. The interrupt will be generated under the following conditions. BERR=1; ARLO=1; AF=1; OVR=1; PECERR=1. TIMEOUT=1; SMBAlert=1.	0
[7:6]	Reserved	RO	Reserved	0
[5:0]	FREQ[5:0]	RW	The I2C module clock frequency field, which must be entered at the correct clock frequency to produce the correct timing, allows a range between 8-48MHz. It must be set between 001000b and 110000b in MHz.	0

13.11.3 I2C Own Address Register 1(I2C1_OAR1)

Offset address: 0x08

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDMODE	Reserved					ADD[9:8]		ADD[7:1]							ADD0

Bit	Name	Access	Description	Reset value
15	ADDMODE	RW	Address mode. 1: 10-bit slave address (does not respond to 7-bit addresses). 0: 7-bit slave address (does not respond to 10-bit address)	0
[14:10]	Reserved	RO	Reserved	0
[9:8]	ADD[9:8]	RW	Interface address, bits 9-8 when using a 10-bit address, ignored when using a 7-bit address.	0
[7:1]	ADD[7:1]	RW	Interface address, bits 7-1.	0
0	ADD0	RW	Interface address, bit 0 when using a 10-bit address, ignored when using a 7-bit address.	0

13.11.4 I2C Own Address Register 2(I2C1_OAR2)

Offset address: 0x0C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								ADD2[7:1]							ENDUAL

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved	0
[7:1]	ADD2[7:1]	RW	Interface address, bits 7-1 of the address in dual address mode.	0
0	ENDUAL	RW	Dual address mode enable bit, set this bit to allow ADD2 to be recognized as well.	0

13.11.5 I2C Data Register (I2C_DATAR)

Offset address: 0x10

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

Bit	Name	Access	Description	Reset value
15:8	Reserved	RO	Reserved	0
7:0	DR[7:0]	RW	Data register, this field is used to store the received data or to store the data used to send to the bus.	0

13.11.6 I2C Status Register 1(I2C_STAR1)

Offset address: 0x14

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			PECERR	OVR	AF	ARLO	BERR	TxE	RxNE	Reserved	STOPF	ADD10	BTF	ADDR	SB

Bit	Name	Access	Description	Reset value
[15:13]	Reserved	RO	Reserved	0
12	PECERR	RW0	The PEC error flag bit occurs on reception, and this bit can be reset by a user write of 0 or by hardware when PE goes low. 1: There is a PEC error and the PEC is received and NAK is returned. 0: No PEC error.	0
11	OVR	RW0	Overrun and underrun flag bits. 1: There are overrun and underrun events occurring: when NOSTRETCH=1, when a new byte is received in receive mode, the content in the data register has not been read out, then the newly received byte will be lost; when in send mode, no new data is written to the data register, and the same byte will be sent twice. 0: No overrun or underrun events.	0
10	AF	RW0	Acknowledge failure bit. Cleared by software writing 0, or by hardware when PE=0. 1：Acknowledge failure. 0：No acknowledge failure.	0
9	ARLO	RW0	Arbitration lost bit. Cleared by software writing 0, or by hardware when PE=0. 1：Arbitration Lost detected. 0：No Arbitration Lost detected.	0
8	BERR	RW0	The bus error flag bit, which can be reset by a user write of 0, or by hardware when PE goes low. 1: Error in start or stop condition; 0: Normal.	0
7	TxE	RO	Data register empty bit. Cleared by software writing to the DR register or by hardware after a start or a stop condition or when PE=0. 1：Data register empty. 0：Data register not empty.	0
6	RxNE	RO	Data register not empty bit. Cleared by software reading or writing the DR register or by hardware when PE=0. 1：Data register not empty. 0：Data register empty.	0
5	Reserved	RO	Reserved	0
4	STOPF	RO	Stop detection bit. Cleared by software reading the SR1 register followed by a write in the CR1 register, or by hardware when PE=0 1：Set by hardware when a Stop condition is detected on the bus by the slave after an	0

			acknowledge (if ACK=1). 0：No Stop condition detected.	
3	ADD10	RO	10-bit header sent bit. Cleared by software reading the SR1 register followed by a write in the DR register of the second address byte, or by hardware when PE=0. 1：Master has sent first address byte. 0：No ADD10 event occurred.	0
2	BTF	RO	Byte transfer finished bit. Cleared by software reading SR1 followed by either a read or write in the DR register or by hardware after a start or a stop condition in transmission or when PE=0. 1：Data byte transfer succeeded. When NOSTRETCH=0: when sending, when a new data is sent and the data register has not yet been written with new data; when receiving, when a new byte is received but the data register has not yet been read. 0：Data byte transfer not done.	0
1	ADDR	RW0	Address sent /matched bit. This bit is cleared by software reading SR1 register followed reading SR2, or by hardware when PE=0. In Master mode: 1：End of address transmission. For 10-bit addressing, the bit is set after the ACK of the 2nd byte. For 7-bit addressing, the bit is set after the ACK of the byte. 0：No end of address transmission. In Slave mode: 1：Received address matched. 0：Address mismatched or not received.	0
0	SB	RO	Start bit. Cleared by software by reading the SR1 register followed by writing the DR register, or by hardware when PE=0 1：Start condition generated. 0：No Start condition.	0

13.11.7 I2C Status Register 2(I2C_STAR2)

Offset address: 0x18

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PEC[7:0]								DUALF	Reserved		GENCALL	Reserved	TRA	BUSY	MSL

Bit	Name	Access	Description	Reset value
[15:8]	PEC[7:0]	RO	Packet error checking bit. When PEC is enabled (ENPEC is set), this field holds the value of PEC.	0
7	DUALF	RO	Dual flag. Cleared by hardware after a Stop condition or repeated Start condition, or when PE=0. 1：Received address matched with OAR2. 0：Received address matched with OAR1.	0
[6:5]	Reserved	RO	Reserved	0
4	GENCALL	RO	General call address bit. Cleared by hardware after a Stop condition or repeated Start condition, or when PE=0.	0

			1：General Call Address received when ENGC=1. 0：No General Call.	
3	Reserved	RO	Reserved	0
2	TRA	RO	Transmitter/receiver bit. It is cleared by hardware after detection of Stop condition (STOPF=1), repeated Start condition, loss of bus arbitration (ARLO=1), or when PE=0. 1：Data bytes transmitted. 0：Data bytes received. This bit is set depending on the R/W bit of the address byte.	0
1	BUSY	RO	Bus busy bit. Cleared by hardware on detection of a Stop condition. This information is still updated when the interface is disabled (PE=0). 1：Communication ongoing on the bus: low level present in SDA or SCL. 0：No communication on the bus.	0
0	MSL	RO	Master/slave bit. Set by hardware as soon as the interface is in Master mode (SB=1). Cleared by hardware after detecting a Stop condition on the bus or a loss of arbitration (ARLO=1), or by hardware when PE=0.	0

13.11.8 I2C Clock Register (I2C1_CKCFGR)

Offset address: 0x1C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F/S	DUTY	Reserved		CCR[11:0]											

Bit	Name	Access	Description	Reset value
15	F/S	RW	Master mode selection bit. 1：Fm mode I2C. 0：Sm mode I2C	0
14	DUTY	RW	Duty cycle of high-level time over low-level time in Fm. 1：36%； 0：33.3%。	0
[13:12]	Reserved	RO	Reserved	0
[11:0]	CCR[11:0]	RW	Clock control register in Fm/Sm mode	0

Chapter 14 Serial Peripheral Interface (SPI)

SPI supports data interaction in a 3-wire synchronous serial mode, plus a chip selector line to support hardware switching between Master and Slave modes, and supports communication on a single data line.

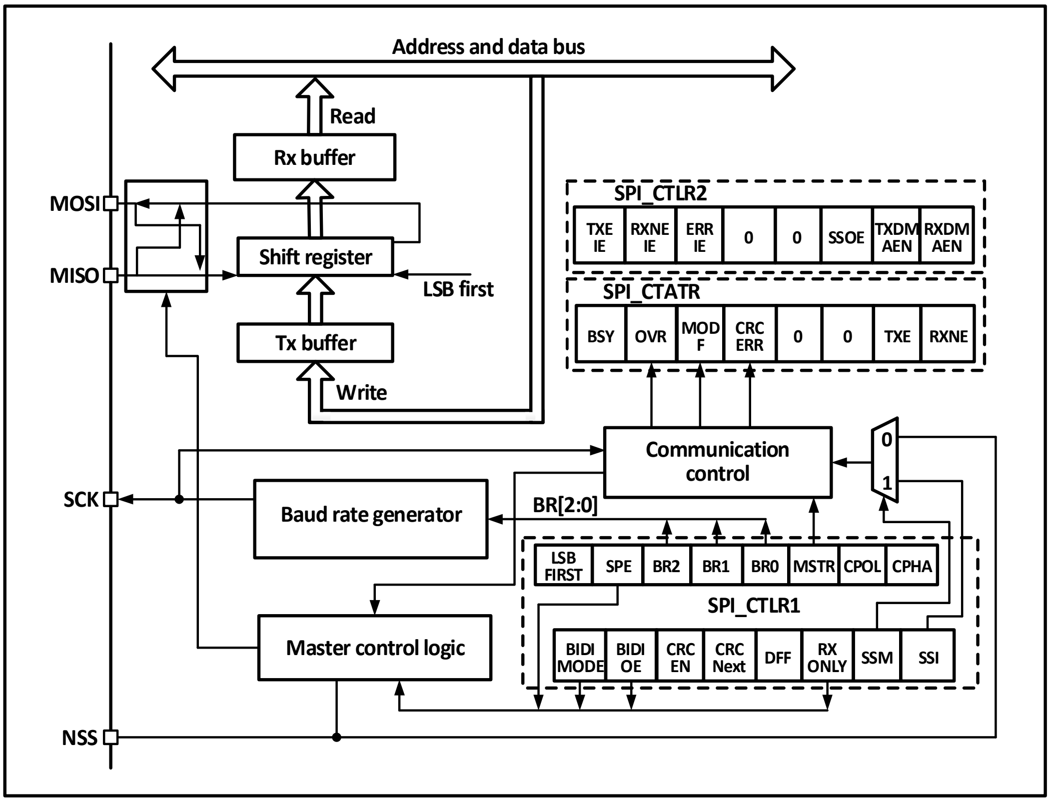
14.1 Main Features

- Support full-duplex synchronous serial mode
- Support single-line half-duplex mode
- Support Master mode and Slave mode, Multi-slave mode
- Support 8-bit or 16-bit data structures
- Maximum clock frequency supports up to half of F_{HCLK}
- Data order supports MSB or LSB first
- Support hardware or software control of NSS pins
- Hardware CRC checksum support for sending and receiving
- Transceiver buffers support DMA transfers
- Support modification of clock phase and polarity

14.2 Function Description

14.2.1 Overview

Figure 14-1 SPI structure block diagram



As can be seen from Figure 14-1, the four main SPI-related pins are MISO, M0SI, SCK and NSS. The MISO pin is the data input pin when the SPI module is operating in Master mode and the data output pin when it is operating in Slave mode. the MOSI pin is the data output pin when it is operating in Master mode and the data input pin when it is operating in Slave mode. the SCK is the clock pin, the clock signal is always output by the host and the slave receives the clock signal and synchronizes the data sending and receiving. the NSS pin is the chip select pin with the following usage.

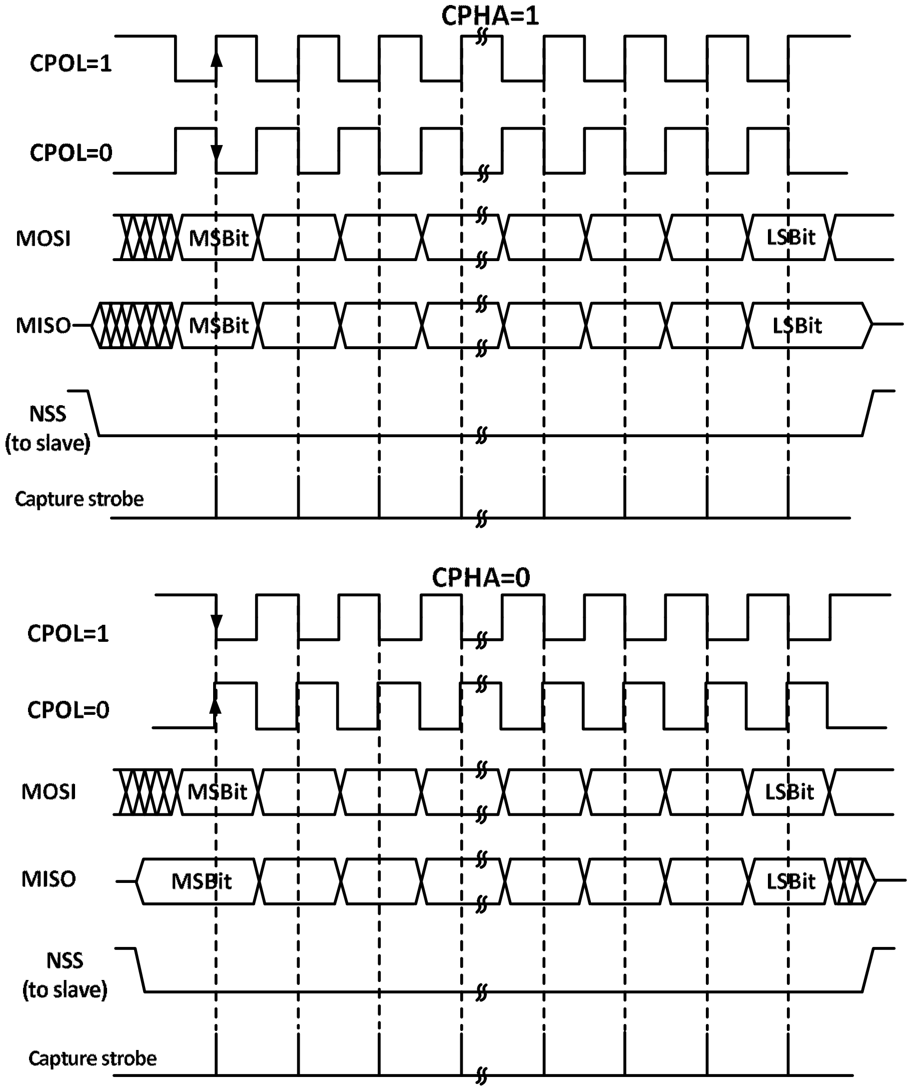
1) NSS controlled by software: When SSM is set and the internal NSS signal is output high or low as

determined by SSI, this case is generally used in SPI Master mode.

2) NSS is controlled by hardware: When the NSS output is enabled, i.e., when SSOE is set, the NSS pin will be actively pulled down when the SPI host sends outputs outward, and if it does not succeed in pulling down the NSS pin, which indicates that there is another master device on the main line that is communicating, a hardware error will be generated; SSOE is not set, it can be used in multi-master mode, and if it is pulled down it will be forced to enter the slave mode, and the MSTR bit will be cleared automatically.

CPHA is set to indicate that the module samples data on the second edge of the clock and the data is latched, while CPHA is not set to indicate that the SPI module samples data on the first edge of the clock and the data is latched, and CPOL indicates whether the clock is held high or low when there is no data. See Figure 14-2 below for details.

Figure 14-2 SPI Mode



The host and device need to be set to the same SPI mode, and the SPE bit needs to be cleared before configuring the SPI mode. the DEF bit determines whether the individual data length of the SP is 8 bits or 16 bits. LSBFIRST controls whether a single data word is preceded by the high bit or the low bit.

14.2.2 Master Mode

The serial clock is generated by SCK when the SPI module is operating in master mode. The following steps are performed to configure into master mode.

Configure the BR[2:0] field of the control register to determine the clock.
Configure the CPOL and CPHA bits to determine the SPI mode.
Configure DEF to determine the data word length.
Configure LSBFIRST to determine the frame format;
Configure the NSS pin, for example by setting the SSOE bit and letting the hardware set the NSS. it is also possible to set the SSM bit and set the SSI bit high.
To set the MSTR bit and the SPE bit, you need to make sure that the NSS is already high at this time.
When you need to transmit data you just need to write the data to be transmitted to the data register. SPI will transmit the data from the transmit buffer to the shift register in parallel and then transmit the data from the shift register according to the setting of LSBFIRST. When the data is already in the shift register, the TXE flag will be set. If the TXEIE is already set, then an interrupt will be generated. If the TXE flag position bit needs to be filled with data into the data register to maintain the complete data flow.
When the receiver receives data, when the last sample clock edge of the data word comes, the data is transferred from the shift register to the receive buffer in parallel, the RXNE bit is set, and an interrupt is generated if the RXNEIE bit was previously set. At this time, the data register should be read as soon as possible to take away the data.

14.2.3 Slave Mode

When the SPI module is operating in slave mode, SCK is used to receive the clock from the host and its own baud rate setting is invalid. To configure into slave mode, proceed as follows.
Configure the DEF bit to set the data bit length.
Configure the CPOL and CPHA bits to match the host mode.
Configure LSBFIRST to match the host data frame format;
The NSS pin needs to be held low in hardware management mode, if NSS is set to software management (SSM set), then keep SSI unset.
Clear the MSTR bit and set the SPE bit to enable SPI mode. In transmitting, when the first slave receive sample edge appears in SCK, the slave starts to transmit. The process of sending is to move the data in the transmit buffer t the transmit shift register. When the data in the transmit buffer is moved to the shift register, the TXE flag will be set, and if the TXEIE bit was set before, then an interrupt will be generated.
During reception, after the last clock sample edge, the RXNE bit is set, the bytes received by the shift register are transferred to the receive buffer, and the read operation of the read data register can obtain the data in the receive buffer. If RXNEIE is set before RXNE is set, then an interrupt is generated.

14.2.4 Simplex Mode

The SPI interface can operate in half-duplex mode, where the master device uses the MOSI pin and the slave device uses the MISO pin for communication. When using half-duplex communication, you need to set BIDIMODE and use BIDIOE to control the transmission direction.
Setting the RXONLY bit in normal full-duplex mode sets the SPI module to receive-only simplex mode, releasing a data pin after RXONLY is set. The SPI can also be set to transmit only mode by ignoring the received data.

14.2.5 CRC

The SPI module uses CRC checksum to ensure the reliability of full-duplex communication, and separate CRC calculators are used for data sending and receiving. the polynomial for CRC calculation is determined by the polynomial register, and different calculations are used for 8-bit data width and 16-bit data width, respectively. Setting the CRCEN bit will enable CRC checksum and at the same time will reset the CRC calculator. After the last data byte is sent, setting the CRCNEXT bit will send the TXCRCR calculator calculation after the current byte is sent, while the CRCERR bit will be set if the last received receive shift register value does not match the locally calculated RXCRCR calculation. Using the CRC checksum requires setting the polynomial calculator and setting the CRCEN bit when configuring the SPI operating mode, and setting the CRCNEXT bit on the last word or half-word to send the CRC and perform the receive CRC checksum. Note that the polynomial for the CRC calculation should be unified for both sending and receiving.

14.2.6 DMA

The SPI module supports the use of DMA to speed up data communication, either by using DMA to fill the transmit buffer or by using DMA to pick up data from the receive buffer in a timely manner. DMA will pick up or send data in a timely manner using RXNE and TXE as signals. DMA can also operate in simplex or CRC mode.

14.2.7 Errors

- Master mode fault (MODF)

When the SPI is operating in NSS pin hardware management mode, an external pull-down of the NSS pin occurs; or in NSS pin software management mode, the SSI bit is cleared; or the SPE bit is cleared, causing the SPI to be shut down; or the MSTR bit is cleared and the SPI enters slave mode. If the ERRIE bit is already set, an interrupt is also generated. Steps to clear the MODF bit: First perform a read or write operation to R16_SPI1_STATR, and then write R16_SPI1_CTLR1.

- Overrun condition

If the host sends data and there is unread data in the receive buffer of the slave device, an overflow error occurs, the OVR bit is set, and an interrupt is also generated if ERRIE is set. Sending an overflow error should restart the current transmission. Reading the data register and then reading the status register will eliminate this bit.

- CRC error

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

14.2.8 Interrupts

The SPI module supports five interrupt sources, among which the TXE and RXNE events are set when the TXEIE and RXNEIE bits are set respectively. In addition to the above three errors will also generate interrupts, namely MODF, OVR and CRCERR, after enabling the ERRIE bit, these three errors will also generate error interrupts.

14.3 Register Description

Table 14-1 SPI-related registers list

Name	Access address	Description	Reset value
R16_SPI_CTLR1	0x40013000	SPI Control register1	0x0000
R16_SPI_CTLR2	0x40013004	SPI Control register2	0x0000
R16_SPI_STATR	0x40013008	SPI Status register	0x0002
R16_SPI_DATAR	0x4001300C	SPI Data register	0x0000
R16_SPI_CRCR	0x40013010	SPI Polynomial register	0x0007
R16_SPI_RCRCR	0x40013014	SPI Receive CRC register	0x0000
R16_SPI_TCRCR	0x40013018	SPI Transmit CRC register	0x0000
R16_SPI_HSCR	0x40013024	SPI High-speed control register	0x0000

14.3.1 SPI Control Register 1 (SPI_CTLR1)

Offset address: 0x00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIDIMODE	BIDIOE	CRCEN	CRCNEXT	DFF	RXONLY	SSM	SSI	LSBFIRST	SPE	BR[2:0]			MSTR	CPOL	CPHA

Bit	Name	Access	Description	Reset value
15	BIDIMODE	RW	Bidirectional data mode enable bit.	0

			1: Selection of 1-line bidirectional mode. 0: Selection of 2-line bi-directional mode.	
14	BIDIOE	RW	Output enable in bidirectional mode bit, used in conjunction with BIDImode. 1: Enable output, transmit only. 0: Disable output, receive only.	0
13	CRCEN	RW	Hardware CRC checksum enable bit, this bit can only be written when SPE is 0. This bit can only be used in full-duplex mode. 1: Initiate CRC calculation. 0: CRC calculation is disabled.	0
12	CRCNEXT	RW	After the next data transfer, send the value of the CRC register. This should be set immediately after the last data is written to the data register. 1: Sending CRC checksum results. 0: Continue to send data from the data register.	0
11	DFF	RW	Data frame format bit, this bit can only be written when SPE is 0. 1: Sending and receiving using 16-bit data length. 0: Use 8-bit data length for sending and receiving.	0
10	RXONLY	RW	The receive-only bit in 2-wire mode is used in conjunction with BIDIMODE. Setting this bit allows the device to receive only and not transmit. 1: Receive only, simplex mode. 0: Full-duplex mode.	0
9	SSM	RW	Software slave management bit, this bit determines whether the level of the NSS pin is controlled by hardware or software. 1: Software control of the NSS pins. 0: Hardware control NSS pins.	0
8	SSI	RW	Internal slave select bit, with SSM set, this bit determines the level of the NSS pin. 1: NSS is high. 0: NSS is low.	0
7	LSBFIRST	RW	Frame format control bit. It is not possible to modify this bit during communication. 1: LSB is transmitted first; 0: MSB is transmitted first. Note: LSB is only supported by SPI as host.	0
6	SPE	RW	SPI enable bit. 1: Enable SPI. 0: Disable SPI.	0
[5:3]	BR[2:0]	RW	Baud rate setting field, this field cannot be modified during communication. 000: F_{HCLK} /2; 001: F_{HCLK} /4. 010: F_{HCLK} /8; 011: F_{HCLK} /16. 100: F_{HCLK} /32; 101: F_{HCLK} /64. 110: F_{HCLK} /128; 111: F_{HCLK} /256.	0
2	MSTR	RW	Master-slave setting bit, this bit cannot be modified during communication. 1: Configured as a master device. 0: Configured as a slave device.	0b
1	CPOL	RW	Clock polarity selection bit, this bit cannot be modified during communication. 1: SCK is held high in idle state. 0: SCK is held low in idle state.	0
0	CPHA	RW	Clock phase setting bit, this bit cannot be modified during communication. 1: Data sampling starts from the second clock	0

			edge. 0: Data sampling starts from the first clock edge.	

14.3.2 SPI Control Register 2 (SPI_CTLR2)

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	Tx buffer empty interrupt enable bit. Setting this bit allows an interrupt to be generated when TXE is set.	0
6	RXNEIE	RW	RX buffer not empty interrupt enable bit. Used to generate an interrupt request when the RXNE flag is set.	0
5	ERRIE	RW	Error interrupt enable bit. Setting this bit allows interrupts to be generated when errors (CRCERR, OVR, MODF) are generated.	0
[4:3]	Reserved	RO	Reserved	0
2	SSOE	RW	SS output enable bit. Disabling SS output can work in multi-master mode. 1: Enable the SS output. 0: Disable SS output in Master mode.	0
1	TXDMAEN	RW	Tx buffer DMA enable bit. 1：Enable Tx buffer DMA. 0：Disable Tx buffer DMA.	0
0	RXDMAEN	RW	Rx buffer DMA enable bit. 1：Enable Rx buffer DMA. 0：Disable Rx buffer DMA.	0

14.3.3 SPI Status Register (SPI_STATR)

Offset address: 0x08

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

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved	0
7	BSY	RO	Busy flag. This flag is set and cleared by hardware. 1：SPI is busy in communication or Tx buffer is not empty. 0：SPI (or I2S) not busy.	0
6	OVR	RWO	Overrun flag. This flag is set by hardware and reset by a software sequence. 1：Overrun occurred. 0：No overrun occurred.	0

5	MODF	RO	Mode fault. This flag is set by hardware and reset by a software sequence. 1：Mode fault occurred. 0：No mode fault occurred.	0
4	CRCERR	RW0	CRC error flag. This flag is set by hardware and reset by a software sequence. 1：CRC value received does not match the SPI_RXCRCR value. 0：CRC value received matches the SPI_RXCRCR value.	0
3	UDR	R0	Underrun flag. This flag is set by hardware and reset by a software sequence. 1：Underrun occurred. 0：No underrun occurred.	0
2	CHSID	RO	Channel side. This flag is set by hardware and reset by a software sequence. 1：Channel Right has to be transmitted or has been received. 0：Channel Left has to be transmitted or has been received.	0
1	TXE	RO	Transmit buffer empty. 1：Tx buffer empty. 0：Tx buffer not empty.	1
0	RXNE	RO	Receive buffer not empty. 1：Rx buffer not empty. 0：Rx buffer empty. Note: Read DATAR and auto-zero.	0

14.3.4 SPI Data Register (SPI_DATAR)

Offset address: 0x0C

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

Bit	Name	Access	Description	Reset value
[15:0]	DR[15:0]	RW	Data register. The data registers are used to store the received data or pre-store the data to be sent out, so the reading and writing of the data registers actually correspond to the operation of different areas, where the read pairs use the receive buffer and the write pairs correspond to the send buffer. Data can be received and sent in 8 or 16 bits, and it is necessary to determine how many bits of data to use before transmission. When using 8 bits for data transmission, only the lower 8 bits of the data registers are used, and the higher 8 bits are forced to 0 for reception. using a 16-bit data structure causes all 16 bits of the data registers to be used.	0

14.3.5 SPI1 Polynomial Register (SPI_CRCR)

Offset address: 0x10

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

Bit	Name	Access	Description	Reset value
[15:0]	CRCPOLY[15:0]	RW	CRC polynomial. This register contains the polynomial for the CRC calculation.	7

14.3.6 SPI1 Receive CRC Register (SPI_RCRCR)

Offset address: 0x14

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

Bit	Name	Access	Description	Reset value
[15:0]	RXCRC[15:0]	RO	Rx CRC. Store the result of the calculated CRC checksum of the received byte. Setting CRCEN resets this register. The calculation method uses the polynomial used in CRCPOLY. 8-bit mode only the lower 8 bits are involved in the calculation, 16-bit mode all 16 bits are involved in the calculation. It is necessary to read this register when BSY is 0.	0

14.3.7 SPI1 Transmit CRC Register (SPI_TCRCR)

Offset address: 0x18

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

Bit	Name	Access	Description	Reset value
[15:0]	TXCRC[15:0]	RO	Tx CRC. Store the result of the calculated CRC checksum of the bytes that have been sent out. Setting CRCEN resets this register. The calculation method uses the polynomial used in CRCPOLY. 8-bit mode only the lower 8 bits are involved in the calculation, while in 16-bit mode all 16 bits are involved. It is necessary to read this register when BSY is 0.	0

14.3.8 SPI High-speed Control Register (SPI_HSCR)

Offset address: 0x24

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

Bit	Name	Access	Description
[15:1]	Reserved	RO	Reserved
0	HSRXEN	WO	Read enable in SPI high-speed mode (CLK greater than or equal to 36 MHz). This mode is valid only when the clock is divided by 2 (i.e., BR = 000 in the CTLR1 register). This bit is not readable. 1: Enable high-speed read mode. 0: Disable high-speed read mode.

Chapter 15 Electronic Signature (ESIG)

The electronic signature contains the chip identification information: the flash memory area capacity and a unique identifier. It is burned into the system storage area of the memory module by the manufacturer at the factory and can be read by SWD (SDI) or application code.

15.1 Functional Description

Flash capacity: Indicates the current size of the chip that can be used by user applications.

Unique identification: 96-bit binary code, unique to any microcontroller, the user can only read access cannot be modified. This unique identification information can be used as a microcontroller (product) security password, encryption and decryption keys, product serial numbers, etc., to improve system security mechanisms or to indicate the identity information.

All the above can be read accessed by 8/16/32 bit by the user.

15.2 Register Description

Table 15-1 ESIG-related registers list

Name	Access Address	Description	Reset value
R16_ESIG_FLACAP	0x1FFFF7E0	Flash capacity register	0xXXXX
R32_ESIG_UNIID1	0x1FFFF7E8	UID register 1	0xXXXXXXXX
R32_ESIG_UNIID2	0x1FFFF7EC	UID register 2	0xXXXXXXXX
R32_ESIG_UNIID3	0x1FFFF7F0	UID register 3	0xXXXXXXXX

15.2.1 Flash capacity register (ESIG_FLACAP)

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

Bit	Name	Access	Description	Reset value
[15:0]	F_SIZE[15:0]	RO	Flash capacity in Kbyte. Example: 0x0080 = 128 K bytes	X

15.2.2 UID Register (ESIG_UNIID1)

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

Bit	Name	Access	Description	Reset value
[31:0]	U_ID[31:0]	RO	The 0-31 digits of UID.	X

15.2.3 UID Register (ESIG_UNIID2)

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

Bit	Name	Access	Description	Reset value
[31:0]	U_ID[63:32]	RO	The 32-63 digits of UID.	X

15.2.4 UID Register (ESIG_UNIID3)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
U_ID[95:80]															

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

Bit	Name	Access	Description	Reset value
[31:0]	U_ID[95:64]	RO	The 64-95 digits of UID.	X

Chapter 16 Flash Memory and User Option Bytes

16.1 Flash Memory Organization

The internal flash memory of the chip is organized as follows.

Table 16-1 Flash Memory Organization

Block	Name	Address Range	Size(byte)
Main memory	Page 0	0x0800 0000 – 0x0800 003F	64
	Page 1	0x0800 0040 – 0x0800 007F	64
	Page 2	0x0800 0080 – 0x0800 00BF	64
	Page 3	0x0800 00C0 – 0x0800 00FF	64
	…	…	…
	Page 256	0x0800 3FC0 – 0x0800 3FFF	64
Information block	Launcher code	0x1FFF F000 – 0x1FFF F77F	2K-128
	User option bytes	0x1FFF F800 – 0x1FFF F83F	64

Notes: The above main memory area is used for user's application storage and is write-protected in 1K byte (16 pages) units; except for the "vendor configuration word" area which is factory locked and inaccessible to the user, the other areas are user-operable under certain conditions.

16.2 Flash Memory Programming and Security

16.2.1 Two Programming/Erasing Methods

- Standard programming: This mode is the default programming mode (compatible mode). In this mode, the CPU performs programming in single 2-byte mode and performs erase and whole erase operation in single 1K byte.
- Fast programming: This method uses page operation (recommended). After a specific sequence of unlocking, it performs a single 64-byte programming and 64-byte erasing, 1K-byte erasing (standard 1K whole chip erasing is also applicable to fast programming).

16.2.2 Security - Prevent Illegal Access (read, write, erase)

- Page write protection
- Read protection

When the chip is in the read-protected state.

1) Main memory pages 0-32 (2K bytes) are automatically write-protected state, not controlled by FLASH_WPR register; unread-protected state, all main memory pages are controlled by FLASH_WPR register.
2) The system boot code area, SDI mode, and RAM area are not erasable or programmable for main memory, except for whole chip erasure. User-option bytes areas can be erased or programmed. If an attempt is made to unprotect the read (program the user word), the chip will automatically erase the entire user area.

Note: The internal RC oscillator (HSI) must be turned on when performing a program/erase operation of the flash memory.

16.3 Register Description

Table 16-2 FLASH-related registers list

Name	Access address	Description	Reset value
R32_FLASH_ACTLR	0x40022000	Control register	0x00000000
R32_FLASH_KEYR	0x40022004	FPEC key register	X

R32_FLASH_OBKEYR	0x40022008	OBKEY register	X
R32_FLASH_STATR	0x4002200C	Status register	0x00008000
R32_FLASH_CTLR	0x40022010	Configuration register	0x00008080
R32_FLASH_ADDR	0x40022014	Address register	X
R32_FLASH_OBR	0x4002201C	Option byte register	0x03FFFFFE
R32_FLASH_WPR	0x40022020	Write protection register	0xFFFFFFF
R32_FLASH_MODEKEYR	0x40022024	Extended key register	X
R32_FLASH_BOOT_MODEKEYR	0x40022028	Unlock BOOT key register	X

16.3.1 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														LATENCY	

Bit	Name	Access	Description	Reset value
[31:2]	Reserved	RO	Reserved	0
[1:0]	LATENCY[1:0]	RW	Number of FLASH wait states 00:0 wait (recommended 0=<SYSCLK=<24MHz) 01:1 wait (recommended 24=<SYSCLK=<48MHz) 10: 2 wait (recommended 24=<SYSCLK=<48MHz) Other: Invalid	0

16.3.2 FPEC Key Register (FLASH_KEYR)

Offset address: 0x04

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

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

Bit	Name	Access	Description	Reset value
[31:0]	KEYR[31:0]	WO	FPEC keys for entering FPEC unlocking keys include. RDPRT key = 0x000000A5. KEY1 = 0x45670123. KEY2 = 0xCDEF89AB.	X

16.3.3 OBKEY Register (FLASH_OBTKEYR)

Offset address: 0x08

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

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

Bit	Name	Access	Description	Reset value
[31:0]	OBKEYR[31:0]	WO	Option bytes key for entering the option bytes key to release OPTWRE.	X

16.3.4 Status Register (FLASH_STATR)

Offset address: 0x0C

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LOCK	MODE	Reserved								EOP	WRPRTERR	Reserved			BSY

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
15	LOCK	RW	BOOT Lock 1: Locked, cannot perform a write operation to the FLASH_STATA[14] field. 0: Unlocked, can perform write operation to FLASH_STATR[14] field. *Note: Write 1 is set, write 0 is invalid.*	1
14	MODE	RW	Control the switch between user area and BOOT area 1：After software reset, you can switch to the BOOT area. 0: After software reset, you can switch to the user area.	0
[13:6]	Reserved	RO	Reserved	0
5	EOP	RW1	Indicates the end of the operation, and write 1 clears 0. The hardware is set each time it is successfully erased or programmed.	0
4	WRPRTERR	RW1	Indicates a write protection error, write 1 clear. The hardware will set the address if it is programmed for write protection.	0
[3:1]	Reserved	RO	Reserved	0
0	BUSY	RO	Indicates busy status. 1: Indicates that a flash operation is in progress. 0: End of operation.	0

Note: When performing the programming operation, you need to make sure the STRT bit of FLASH_CTLR register is 0.

16.3.5 Configuration Register (FLASH_ CTLR)

Offset address: 0x10

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLOCK	Reserved		EOPIE	Reserved	ERRIE	OBWRE	Reserved	LOCK	STRT	OBER	OBPG	Reserved	MER	SER	PG

Bit	Name	Access	Description	Reset value
[31:20]	Reserved	RO	Reserved	0
19	BUFRST	RW	BUF reset operation	0
18	BUFLOAD	RW	Cache data into BUF	0
17	FTER	RW	Performs a fast page (64Byte) erase operation.	0
16	FTPG	RW	Performs quick page programming operations.	0
15	FLOCK	RW1	Fast programming lock. Write '1' only. When this bit is '1' it indicates that fast programming/erase mode is not available. Hardware clears this bit to '0' after the correct unlock sequence is detected. The software is set to 1 and re-locked.	1
[14:13]	Reserved	RO	Reserved	0
12	EOPIE	RW	Operation completion interrupt control (EOP set in FLASH_STATR register). 1: Allow generation of interrupts. 0: Interrupt generation is disabled.	0
11	Reserved	RO	Reserved	0
10	ERRIE	RW	Error status interrupt control (PGERR/WRPRTERR set in FLASH_STATR register). 1: Allow generation of interrupts. 0: Interrupt generation is disabled.	0
9	OBWRE	RW0	User selects word lock, software clears 0. 1: Indicates that the user option bytes can be programmed for operation. It needs to be set by hardware after writing the correct sequence in FLASH_OBKEYR register. 0: Re-lock the user selection word after the software is cleared.	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 unwritable. Hardware clears this bit to '0' after the correct unlock sequence is detected. After an unsuccessful unlock operation, the bit will not be changed again until the next system reset.	1
6	STRT	RW1	Start. Set 1 to start an erase action and the hardware automatically clears 0 (BSY becomes '0').	0
5	OBER	RW	Perform user-option bytes erasure	0
4	OBPG	RW	Perform user-option bytes programming	0
3	Reserved	RO	Reserved	0
2	MER	RW	Performs a full-erase operation (erases the entire user area).	0
1	PER	RW	Perform sector erase (1K)	0
0	PG	RW	Performs standard programming operations.	0

16.3.6 Address Register (FLASH_ ADDR)

Offset address: 0x14

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

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

Bit	Name	Access	Description	Reset value
[31:0]	FAR[31:0]	WO	The flash memory address, when programming, is the programmed address, and when erasing, is the start address of the erase. When the BSY bit in FLASH_SR register is '1', this register cannot be written.	0

16.3.7 Option Byte Register (FLASH_OBR)

Offset address: 0x14

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						DATA1								DATA0	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA0						2’b11		Reserved	CFGRSTT		STANDYRST	STOPRST	IWDGSW	RDPRT	OBERR

Bit	Name	Access	Description	Reset value
[31:26]	Reserved	RO	Reserved	0
[25:18]	DATA1[7:0]		Data byte 1	X
[17:10]	DATA0[7:0]		Data byte 0	X
[9:8]			2’b11	
7	Reserved	RO	Reserved	X
[6:5]	CFGRSTT	RO	Configuration word reset delay time	X
4	STANDY_RST	RO	System reset control in Standby mode.	X
3	Reserved	RO	Reserved	X
2	IWDG_SW	RO	Independent Watchdog (IWDG) hardware enable bit.	1
1	RDPRT	RO	Read protection status. 1: Indicates that the flash memory is currently read protected.	1
0	OBERR	RO	Wrong choice of words. 1: Indicates that the selection word and its inverse code do not match.	0

Note: USER and RDPRT are loaded from the user-option bytes area after a system reset.

16.3.8 Write Protect Register (FLASH_WPR)

Offset address: 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
WPR[15:0]															

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	X
[15:0]	WPR[15:0]	RO	Flash memory write protect state. 1: Write protection failure. 0: Write protection is valid. Each bit represents 1K bytes (16 pages) of storage write protection status.	X

Note: WPR is loaded from the user-option bytes area after a system reset.

16.3.9 Extended Key Register (FLASH_MODEKEYR)

Offset address: 0x24

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

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

Bit	Name	Access	Description	Reset value
[31:0]	MODEKEYR [31:0]	WO	Enter the following sequence to unlock the fast programming/erase mode. KEY1 = 0x45670123. KEY2 = 0xCDEF89AB.	X

16.3.10 BOOT Key Register (FLASH_BOOT_MODEKEYP)

Offset address: 0x28

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

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

Bit	Name	Access	Description	Reset value
[31:0]	MODEKEYR [31:0]	WO	Enter the following sequence to unlock the BOOT area KEY1 = 0x45670123. KEY2 = 0xCDEF89AB.	X

16.4 Flash Memory Operation Flow

16.4.1 Read Operations

With direct addressing in the general address space, any read operation of 8/16/32-bit data can access the contents of the flash module and get the corresponding data.

16.4.2 Unlocking the Flash Memory

After a system reset, the flash controller (FPEC) and FLASH_CTLR registers are locked and inaccessible. The flash controller module can be unlocked by writing a sequence to the FLASH_KEYR register.

Unlock sequence.

1) Write KEY1 = 0x45670123 to the FLASH_KEYR register (step 1 must be KEY1).
2) Write KEY2 = 0xCDEF89AB to FLASH_KEYR register (step 2 must be KEY2).

The above operations must be executed sequentially and consecutively, otherwise they are error operations and will lock the FPEC module and FLASH_CTLR registers and generate bus errors until the next system reset.

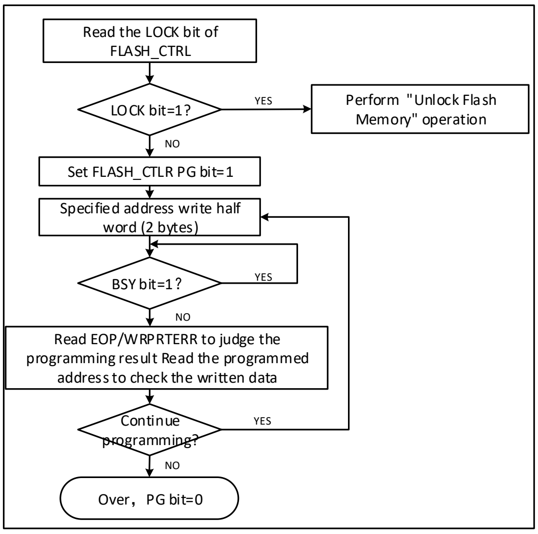
The flash memory controller (FPEC) and FLASH_CTLR registers can be locked again by setting the "LOCK" bit of the FLASH_CTLR register to 1.

16.4.3 Main Memory Standard Programming

Standard programming can be written 2 bytes at a time. When the PG bit of FLASH_CTLR register is '1', each half-word (2 bytes) written to the flash address will initiate programming once, and writing any non-half-word data will cause the FPEC to generate a bus error. During programming, the BSY bit is '1', and at the end of programming, the BSY bit is '0' and the EOP bit is '1'.

Note: When the BSY bit is '1', it will prohibit to perform write operation to any register.

Figure 16-1 FLASH Programming

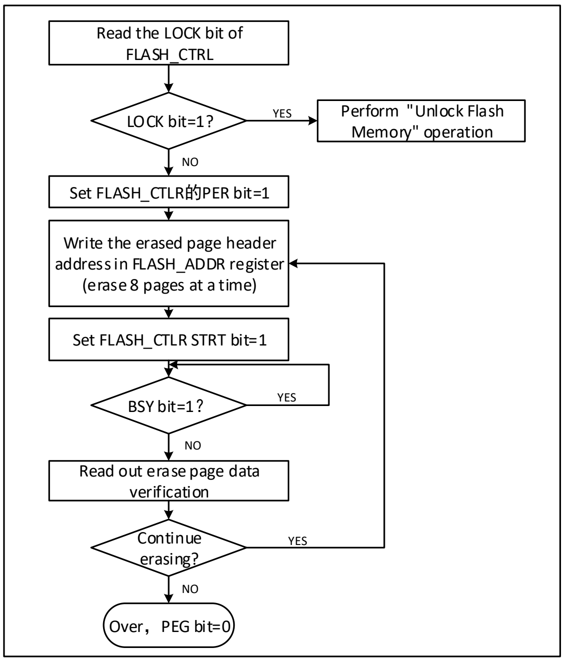


1) Check the FLASH_CTLR register LOCK, if it is 1, you need to execute the "Unlock Flash" operation.
2) Set the PG bit of FLASH_CTLR register to '1' to enable the standard programming mode.
3) Write the half word to be programmed to the specified flash address (even address).
4) Wait for the BSY bit to become '0' or the EOP bit of FLASH_STATR register to be '1' to indicate the end of programming, and clear the EOP bit to 0.
5) Query the FLASH_STATR register to see if there is an error or read the programmed address data checksum.
6) Continue programming you can repeat steps 3-5 and end programming to clear the PG bit to 0.

16.4.4 Main Memory Standard Erase

Flash memory can be erased by standard page (1K bytes) or by whole chip.

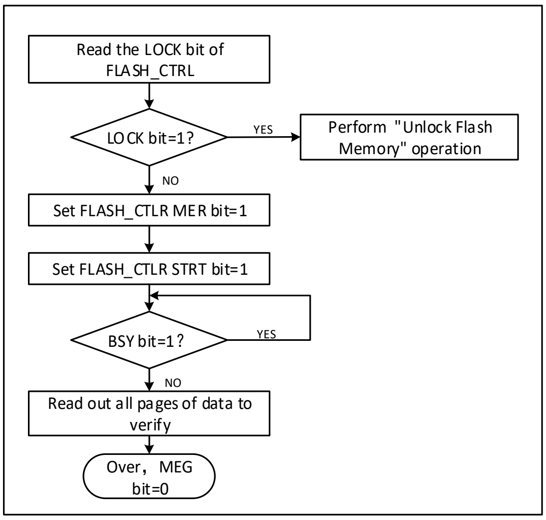
Figure 16-2 FLASH Page Erase



1) Check the LOCK bit of FLASH_CTLR register, if it is 1, you need to execute the "Unlock Flash" operation.

2) Set the PER bit of FLASH_CTLR register to '1' to enable the standard page erase mode.
3) Write the page header address of the selected erase to FLASH_ADDR register.
4) Set the STAT bit of FLASH_CTLR register to '1' to initiate an erase action.
5) Wait for the BYS bit to become '0' or the EOP bit of FLASH_STATR register to be '1' to indicate the end of erase, and clear the EOP bit to 0.
6) Read the data of the erased page for verification.
7) Continue the standard page erase can repeat steps 3-5 and end the erase to clear the PEG bit to 0.

Figure 16-3 FLASH whole chip erase



1) Check the LOCK bit of FLASH_CTLR register, if it is 1, you need to execute the "Unlock Flash" operation.
2) Set the MER bit of FLASH_CTLR register to '1' to enable the whole chip erase mode.
3) Set the STAT bit of FLASH_CTLR register to '1' to start the erase action.
4) Wait for the BYS bit to become '0' or the EOP bit of FLASH_STATR register to be '1' to indicate the end of erase, and clear the EOP bit to 0.
5) Read the data of the erased page for verification.
6) Clear the MER bit to 0.

16.4.5 Fast Programming Mode Unlocking

Fast programming mode operation can be unlocked by writing a sequence to the FLASH_MODEKEYR register. After unlocking, the FLOCK bit of FLASH_CTLR register will be cleared to 0, indicating that fast erase and programming operations can be performed. The FLASH_CTLR register is locked again by software setting the "FLOCK" bit to 1.

Unlock sequence.

1) Write KEY1 = 0x45670123 to the FLASH_MODEKEYR register.
2) Write KEY2 = 0xCDEF89AB to FLASH_MODEKEYR register.

The above operations must be performed sequentially and consecutively, otherwise they are wrong operations will be locked and cannot be unlocked again until the next system reset.

Note: Quick programming operation requires unlocking the "LOCK" and "FLOCK" layers.

16.4.6 Main Memory Fast Programming

Fast programming by page (64 bytes).

1) Check the LOCK bit of FLASH_CTLR register, if it is 1, you need to execute the "Unlock Flash" operation.
2) Check the BSY bit of the FLASH_STATR register to confirm that there are no other programming operations in progress.

3) Check the FLASH_CTLR register FLOCK bit, if it is 1, you need to execute the "fast programming mode unlock" operation.
4) Set the FTPG bit of FLASH_CTLR register to enable the fast programming mode function.
5) Set the BUFRST bit of FLASH_CTLR register to perform the operation of clearing the internal 64-byte buffer.
6) Wait for the BYS bit to become '0' or the EOP bit of FLASH_STATR register to be '1' to indicate the end of clearing, and clear the EOP bit to 0.
7) Start writing 4 bytes of data to the specified address (4 bytes/operation), then set the BUFLOAD bit of FLASH_CTLR register and execute loading to the buffer.
8) Wait for the BYS bit to become '0' or the EOP bit of FLASH_STATR register to be '1' to indicate the end of loading, and clear the EOP bit to 0.
9) Repeat steps 7-8 a total of 16 times to load all 64 bytes of data into the buffer (the main 16 rounds of operation addresses should be consecutive).
10) Write the first address of the fast programming page to the FLASH_ADDR register.
11) Set the STAT bit of FLASH_CTLR register to '1' to start a fast page programming action.
12) Wait for the BYS bit to become '0' or the EOP bit of FLASH_STATR register to be '1' to indicate the end of programming, and clear the EOP bit to 0.
13) Query FLASH_STATR register to see if there is an error, or read the programmed address data checksum.
14) Continue the Quick Page programming can repeat steps 5-13 and end the programming to clear the FTPG bit to 0.

16.4.7 Main Memory Fast Erase

Fast Erase erases by page (64 bytes).

1) Check the LOCK bit of FLASH_CTLR register, if it is 1, you need to execute the "Unlock Flash" operation.
2) Check the FLASH_CTLR register FLOCK bit, if it is 1, you need to execute the "fast programming mode unlock" operation.
3) Check the BSY bit of the FLASH_STATR register to confirm that there are no other programming operations in progress.
4) Set the FTER bit of FLASH_CTLR register to '1' to enable the fast page erase (64 bytes) mode function.
5) Write the first address of the fast erase page to the FLASH_ADDR register.
6) Set the STAT bit of FLASH_CTLR register to '1' to initiate a fast page erase (64 bytes) action.
7) Wait for the BSY bit to become '0' or the EOP bit of FLASH_STATR register to be '1' to indicate the end of erase, and clear the EOP bit to 0.
8) Query FLASH_STATR register to see if there is an error, or read the erase page address data checksum.
9) Continue fast page erase can repeat steps 5-8, end erase will FTER bit clear 0.

16.5 User Option Bytes

The user-option bytes is solidified in FLASH and will be reloaded into the corresponding register after system reset, and can be erased and programmed by the user at will. The user option bytes information block has a total of 8 bytes (4 bytes for write protection, 1 byte for read protection, 1 byte for configuration options, and 2 bytes for storing user data), and each bit has its inverse code bit for checksum during loading. The following describes the structure and meaning of the option bytes information.

Table 16-3 32-bit selection word format division

[31:24]	[23:16]	[15:8]	[7:0]
Option bytes byte 1 inverse code	Option bytes byte 1	Option bytes byte 0 inverse code	Option bytes byte 0

Table 16-4 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	Reserved	Reserved	Reserved	Reserved

Name/Byte			Description	Reset value
RDPR			Read protection control bit to configure whether the code in the flash memory can be read out. 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 out. Other values: indicates code read protection status, not readable, pages 0-31 (4K) will be automatically write protected and not controlled by WRPR0.	0x01
USER	[7:6]	Reserved	Reserved (must be 1)	11b
	5	START_MODE	Power-on startup mode 1: Boot from BOOT 0: Boot from user area *Note: This function is not applicable to products with 0 in the penultimate digit of the lot number.*	1
	[4:3]	RST_MODE[1:0]	PD7 multiplexed as external pin reset. 00: Ignoring pin states within 128us after turning on the multiplexing function. 01: Ignoring pin states within 1ms after turning on the multiplexing function. 10: Ignoring pin states within 12ms after turning on the multiplexing function. 11: Multiplexing function off, PD7 for I/O function.	10
	2	STANDYRST	System reset control in Standby mode: 1: Not enabled, does not reset when entering Standby mode system; 0: Enabled, generates a system reset when entering Standby mode.	1
	1	Reserved	Reserved	1
	0	IWDGSW	Independent Watchdog (IWDG) hardware enable configuration. 1: IWDG is enabled by software and disabled from being enabled by hardware. 0: IWDG is turned on by hardware itself (since the clock for IWDG is provided by LSI, it is automatically turned on by LSI). *Note: The core stops in debug mode and the watchdog hardware enable will be disabled.*	1
Data0–Data1			Store 2 bytes of user data.	FFFFh
WRPR0 - WRPR3			Write-protect control bits. Each bit is used to control the write-protect status of 1 sector (1K bytes/sector) in main memory. 1: Disable write protection. 0: Enable Write protection. 2 bytes are used to protect a total of 16K bytes of main memory. WRP0: Sector 0-7 storage write protection control. WRP1: Sector 8-15 storage write protection control. WRP2: Reserved.	FFFFh

	WRP3: Reserved.	

16.5.1 User Option Bytes Unlocking

The user option bytes operation can be unlocked by writing a sequence to the FLASH_OBKEYR register. After unlocking, the OBWRE bit of FLASH_CTLR register will be set to 1, indicating that the user option bytes can be erased and programmed. It can be locked again by clearing the "OBWRE" bit of FLASH_CTLR register to 0 by software.

Unlock sequence.

1) Write KEY1 = 0x45670123 to FLASH_OBKEYR register.
2) Write KEY2 = 0xCDEF89AB to FLASH_OBKEYR register.

Note: User-option bytes operation requires unlocking the "LOCK" and "OBWRE" layers.

16.5.2 User Option Bytes Programming

Only the standard programming method is supported, writing half-words (2 bytes) at a time. In practice, when programming the user-option bytes, FPEC uses only the low byte in the half-word and automatically calculates the high byte (the high byte is the inverse of the low byte) and then starts the programming operation, which will ensure that the byte in the user option bytes and its inverse code are always correct.

1) Check the LOCK bit of FLASH_CTLR register, if it is 1, you need to execute the "Unlock Flash" operation.
2) Check the BSY bit of the FLASH_STATR register to confirm that there are no other programming operations in progress.
3) Set the OBPG bit of FLASH_CTLR register to '1', after that set the STAT bit of FLASH_CTLR register to '1' to turn on the user option bytes programming.
4) Set the OBPG bit of FLASH_CTLR register to '1'.
5) Write the half word (2 bytes) to be programmed to the specified address.
6) Wait for the BYS bit to become '0' or the EOP bit of FLASH_STATR register to be '1' to indicate the end of programming, and clear the EOP bit to 0.
7) Read the programmed address data checksum.
8) Continue programming you can repeat steps 5-7 and end programming to clear the OBPG bit to 0.

Note: When "Read Protected" in the modified selection word becomes "Unprotected", a whole-slice erase of the main memory will be performed automatically. If the selection other than "read protected" is modified, the whole erase operation will not occur.

16.5.3 User Option Bytes Erasure

Directly erase the entire 64-byte user option bytes area.

1) Check the LOCK bit of FLASH_CTLR register, if it is 1, you need to execute the "Unlock Flash" operation.
2) Check the BSY bit of the FLASH_STATR register to confirm that there is no programming operation in progress.
3) Check the OBWRE bit of FLASH_CTLR register, if it is 0, it is necessary to execute the operation of "user option bytes unlock".
4) Set the OBER bit of FLASH_CTLR register to '1', after that set the STAT bit of FLASH_CTLR register to '1' to enable the user option bytes erase.
5) Wait for the BYS bit to become '0' or the EOP bit of FLASH_STATR register to be '1' to indicate the end of erase, and clear the EOP bit to 0
6) Read and erase the address data checksum.
7) End to clear the OBER bit to 0.

16.5.4 Unprotecting Reads

Whether the flash memory is read protected or not is determined by the user option bytes. Read the

FLASH_OBR register, when the RDPRT bit is '1' indicates that the flash memory is currently in the read-protected state, and the flash memory is operationally protected by a series of security guards for the read-protected state. The process of unprotecting the read protection is as follows.

1) Erase the entire user-option bytes area, at which point the read protection field RDPR, at which point the read protection remains in effect.
2) User-option bytes programming and writes the correct RDPR code 0xA5 to unprotect the flash memory from reads. (This step will first cause the system to automatically perform an entire erase operation on the flash memory)
3) Perform a power-on reset to reload the selection byte (including the new RDPR code), at which point the read protection is removed.

Chapter 17 Extended configuration

17.1 Extended Configuration

The system provides the EXTEND extended configuration unit (EXTEND_CTR register). This unit uses the AHB clock and performs a reset action only at system reset. It mainly includes the following extended control bit functions.

1) Adjusting the built-in voltage: The LDOTRIM field selects the default value, which can be modified when adjusting performance and power consumption.
2) Lock-up function monitoring: The LKUPEN field is enabled, which will open the Lock-up situation monitoring of the system. Once the Lock-up situation occurs, the system will perform a software reset and set the LKUPRESET field to 1. After reading, you can write 1 to clear this flag.
3) Configure the op-amp: set OPA_EN to enable the OPA, configure OPA_PSEL to select the positive input pin of the OPA, and configure OPA_NSEL to select the negative input pin of the OPA.

17.2 Register Description

Table 17-1 EXTEND-related registers list

Name	Access address	Description	Reset value
R32_EXTEN_CTR	0x40023800	Configure extended control registers	0x00000A00

17.2.1 Configure Extended Control Register (EXTEND_CTR)

Offset address: 0x00

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="13">Reserved</td><td>OPA_PSEL</td><td>OPA_NSEL</td><td>OPA_EN</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="5">Reserved</td><td>LDO_TRIM</td><td colspan="2">Reserved</td><td>LKUPRST</td><td>LKUPEN</td><td colspan="6">Reserved</td></tr>
</table>

Bit	Name	Access	Description	Reset value
[31:19]	Reserved	RO	Reserved	0
18	OPA_PSEL	RW	OPA positive end channel selection 0: Positive end channel 0 1: Positive end channel 1	0
17	OPA_NSEL	RW	OPA negative end channel selection 0: Negative end channel 0 1: Negative end channel 1	0
16	OPA_EN	RW	OPA Enable 1: Turn on enable 0: Turn off OPA	0
[15:11]	Reserved	RO	Reserved	0
10	LDOTRIM	RW	Core voltage modes. 0: Normal voltage mode 1: Boost voltage mode	0
[9:8]	Reserved	RO	Reserved	0
7	LKUPRST	RW1	LOCKUP reset flag. 1: occurrence of LOCKUP resulting in system reset, write 1 cleared.0: Normal.	0

6	LKUPEN	RW	LOCKUP monitoring function. 1: Enabled, performs a reset and sets LOCKUP_RESET when a lock-up occurs on the system. 0: Not enabled.	0
[5:0]	Reserved	RO	Reserved	0

Chapter 18 Debug Support (DBG)

18.1 Main Features

This register allows the MCU to be configured in the debug state. It includes:

- Independent Watchdog (IWDG) enabled counters
- Window Watchdog (WWDG) enabled counters
- Timer1 enabled counters
- Timer2 enabled counters

18.2 Register Description

18.2.1 Debug MCU Configuration Register (DBGMCU_CR)

Address: 0x7C0(CSR)

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		TIM2_STOP	TIM1_STOP	Reserved		WWDG_STOP	IWDG_STOP	Reserved					STANDBY	STOP	SLEEP

Bit	Name	Access	Description	Reset value
[31:14]	Reserved	RW	Reserved	0
13	TIM2_STOP	RW	Timer 2 debug stop bit. The counter stops when the core enters the debug state. 1: Timer 2's counter stops working. 0: Timer 2's counter is still working normally.	0
12	TIM1_STOP	RW	Timer 2 debug stop bit. The counter stops when the core enters the debug state. 1: Timer 2's counter stops working. 0: Timer 2's counter is still working normally.	0
[10:11]	Reserved	RW	Reserved	0
9	WWDG_STOP	RW	WWDG debug stop bit. The debug WWDG stops working when the core enters the debug state. 1: WWDG counter stops working. 0: WWDG counter is still working normally.	0
8	IWDG_STOP	RW	IWDG debug stop bit. The debug IWDG stops working when the core enters the debug state. 1: IWDG counter stops working. 0: IWDG counter is still working normally.	0
[7:3]	Reserved	RW	Reserved	0
2	STANDBY	RW	Debug the standby mode bits. 1: (FCLK on, HCLK on) The digital circuitry section is not powered down, and the FCLK and HCLK clocks are clocked by the internal RL oscillator. Alternatively, the microcontroller exits STANDBY mode and reset by generating a system reset is the same. 0: (FCLK off, HCLK off) The entire digital circuitry section is powered down. From the software point of view, exiting STANDBY mode is the same as a reset (except that some status bits indicate that the microcontroller has just exited from	0

			STANDBY state).	
1	STOP	RW	Debug stop mode bits. 1: (FCLK on, HCLK on) When in Stop mode, the FCLK and HCLK clocks are provided by the internal RC oscillator. When exiting stop mode, software must reconfigure the clock system to start the PLL, crystal, etc. (same operation as when configuring this bit to 0). 0: (FCLK off, HCLK off) When in STOP mode, the clock controller disables all clocks (including HCLK and FCLK). When exiting from STOP mode, the clock configuration is the same as after reset (the microcontroller is clocked by an 8 MHz internal RC oscillator (HIS)). Therefore, the software must reconfigure the clock control system to start the PLL, crystal, etc.	0
0	SLEEP	RW	Debug sleep mode bits. 1: (FCLK on, HCLK on) In Sleep mode, both FCLK and HCLK clocks are provided by the originally configured system clock. 0: (FCLK on, HCLK off) In Sleep mode, FCLK is provided by the originally configured system clock, and HCLK is off. Since Sleep mode does not reset the configured clock system, the software does not need to reconfigure the clock system when exiting from sleep mode.	0