CH32H417, H416, H415 Series Application Manual

V1.5

illustrate

The CH32H417 is an interconnect-oriented general-purpose microcontroller based on a dual-core design of the RISC-V5F and RISC-V3F chipsets. The CH32H417 integrates...

Includes USB 3.2 Gen1 controller and transceiver, 100Mbps Ethernet MAC and PHY, SerDes high-speed isolated transceiver, Type-C/PD controller.

Controller and PHY, providing SD/EMMC controller, 500MBytes universal high-speed interface UHSIF, DVP digital image interface, single-wire protocol master

Interface SWPMI, Programmable Protocol I/O Controller PIOC, Flexible Memory Controller FMC, DFSDM, LTDC, GPHA, DMA Controller,

It features multiple timers, 8 serial ports, I3C, 4 I2C, 2 QSPI, 4 SPI, 2 I2S, 3 CAN, and other peripheral resources.

It features a 5MHz dual 12-bit ADC unit, a 20MHz 10-bit high-speed HSADC unit, a 16-channel Touchkey, dual DAC units, and 3...

It includes analog resources such as operational amplifiers (OPA) and voltage comparators (CMP), supports 10M/100M Ethernet communication, and supports USB 2.0 and USB 3.0.

It supports USB Host and USB Device functions, Type-C and PDUSB fast charging, and SerDes high-speed isolation and long-distance charging.

Off-line transmission supports dual-core division of labor to improve network protocol processing efficiency and communication response speed.

This manual provides detailed usage information for the CH32H417, H416, and H415 chips, tailored to user application development.

For device characteristics of the CH32H417 series chips, please refer to the CH32H417DS0 datasheet.

RISC-V kernel version overview

Features / kernel Version	Instruction set	hardware stack series	Interruption Nested series	fast Interruption aisle number	Flowing water Wire	Vector table pattern	Extend instruction	instruction cache	Memory Protect	Interface call try
Barley V3F IMABCF-X Barley V5F		2	2	4	3 Address or command support - Standard [1] Single/Dual Wire					
IMABCF-X		8	8	4	7-9 Address or command support supports standard single/dual lines					

Note [1]: Only chips with a non-zero fifth digit in the batch number support memory protection.

For information about the RISC-V core, please refer to Chapter 4, Central Processing Unit (CPU).

Abbreviated descriptions of bit attributes in registers:

Register bit attributes	Attribute Description
RF	A read-only property that reads a fixed value.
RO	The read-only attribute is changed by the hardware.
RZ is a read-only attribute; the bit is automatically cleared to 0 after a read	
WO	operation. It is a write-only attribute (unreadable; the read value is uncertain).
WA	Write-only properties; can be written in safe mode.
WZ	is a write-only property; after a write operation, the bits are automatically cleared
RW	to 0. It is readable and writable.
RWA	is readable and writable in safe mode.
RW1	is readable; writing 1 is valid, writing 0 is invalid. RW0 is readable;
	writing 0 is valid, writing 1 is invalid.
RW1T	is readable; writing 0 is invalid, and writing 1 flips the value. RW1Z is readable.
	Writing 1 clears this bit, and writing 0 has no effect.
SC	Automatically clear.

Chapter 1 Memory and Bus Architecture

1.1 Bus Architecture

The CH32H417 series chip features an internally integrated dual-core architecture, employing the Qingke V3F and V5F designs. Its architecture integrates the core and arbitration unit...

The components, including the memory, DMA module, and SRAM storage, interact through multiple buses. A general-purpose DMA controller is integrated into the design to reduce CPU load and improve performance.

High access efficiency, coupled with data protection mechanisms and automatic clock switching protection, enhances system stability. Its system block diagram is shown below.

picture.

Figure 1-1-1 CH32H417 System Block Diagram

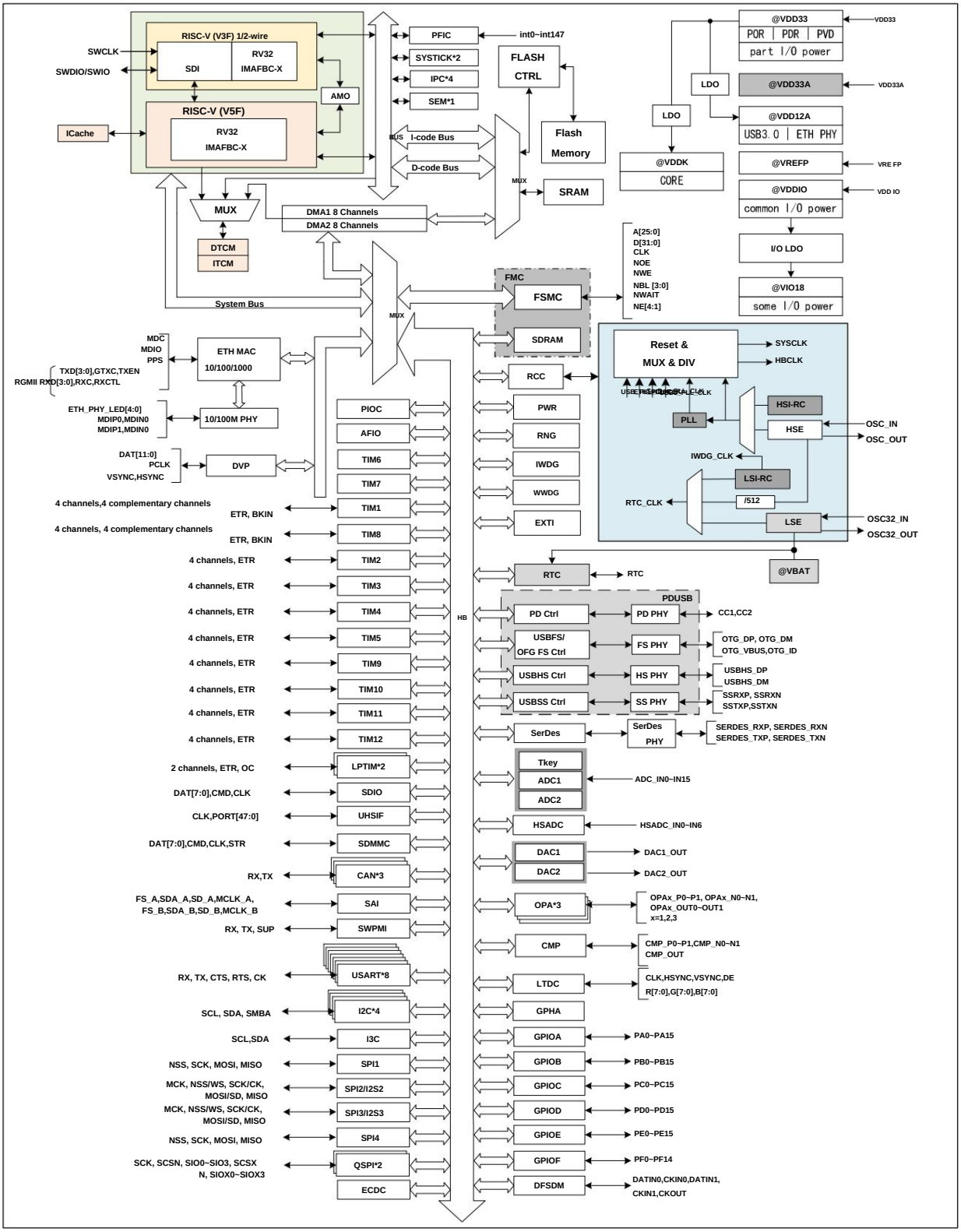


Figure 1-1-2 CH32H416 System Block Diagram

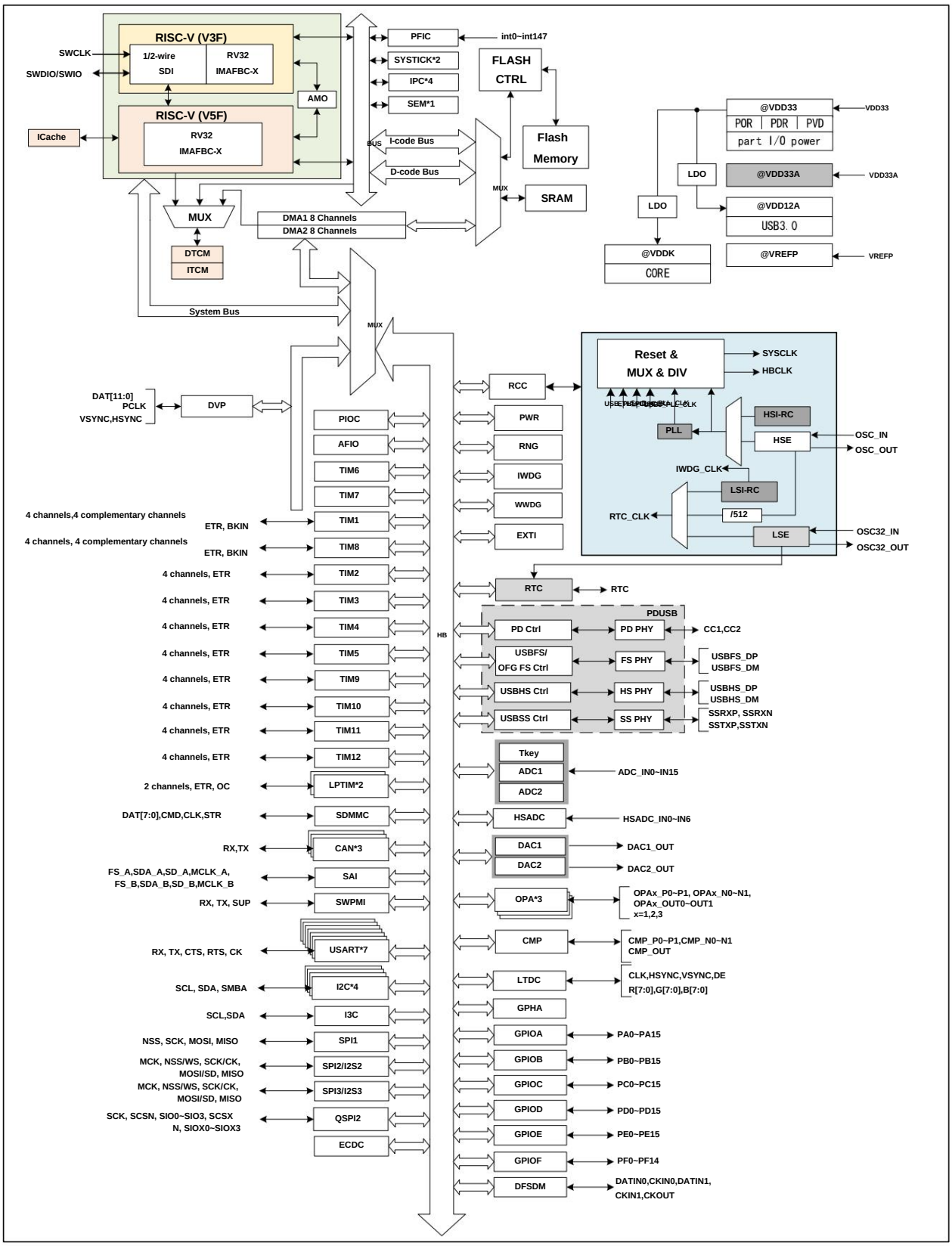
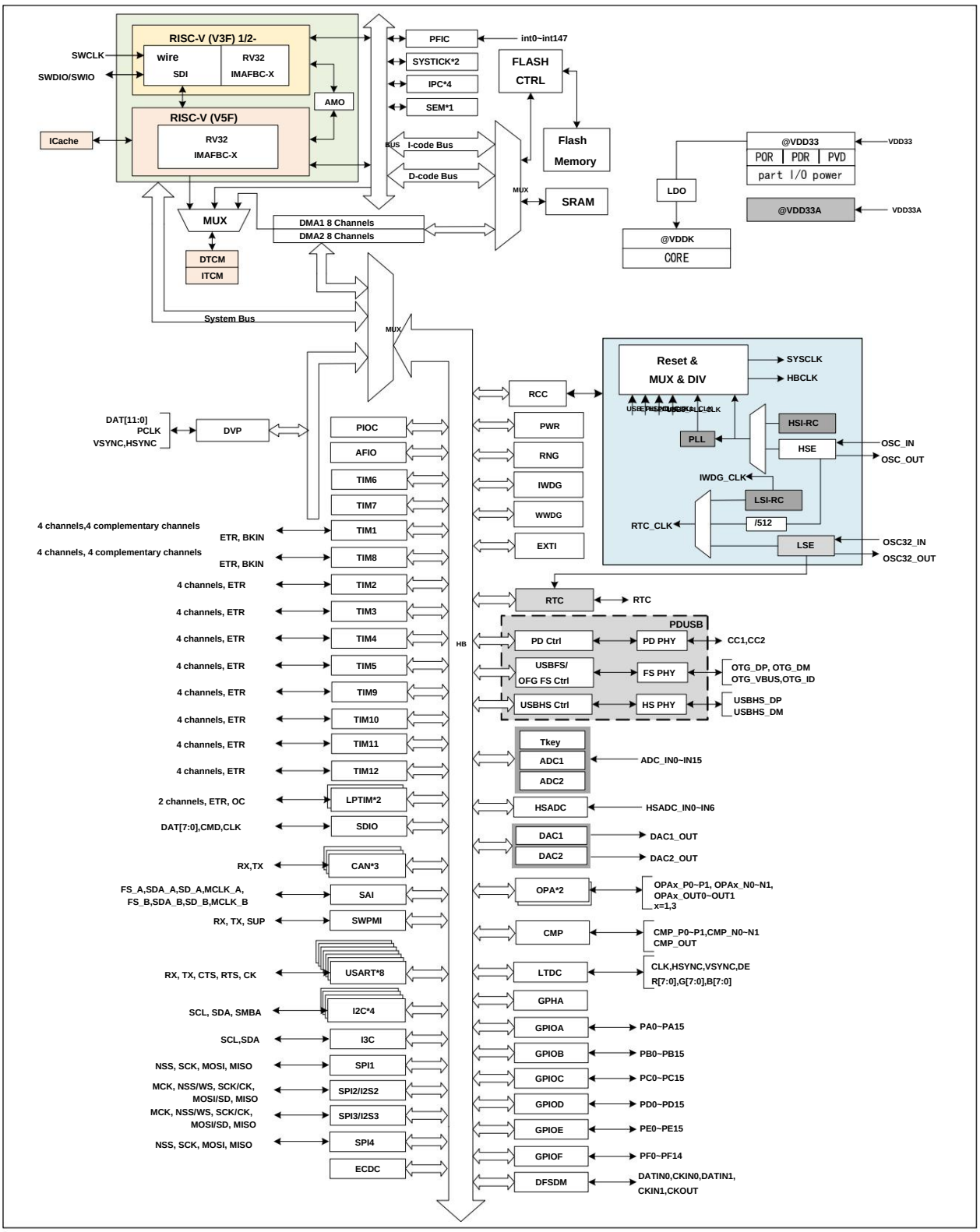


Figure 1-1-3 CH32H415 System Block Diagram



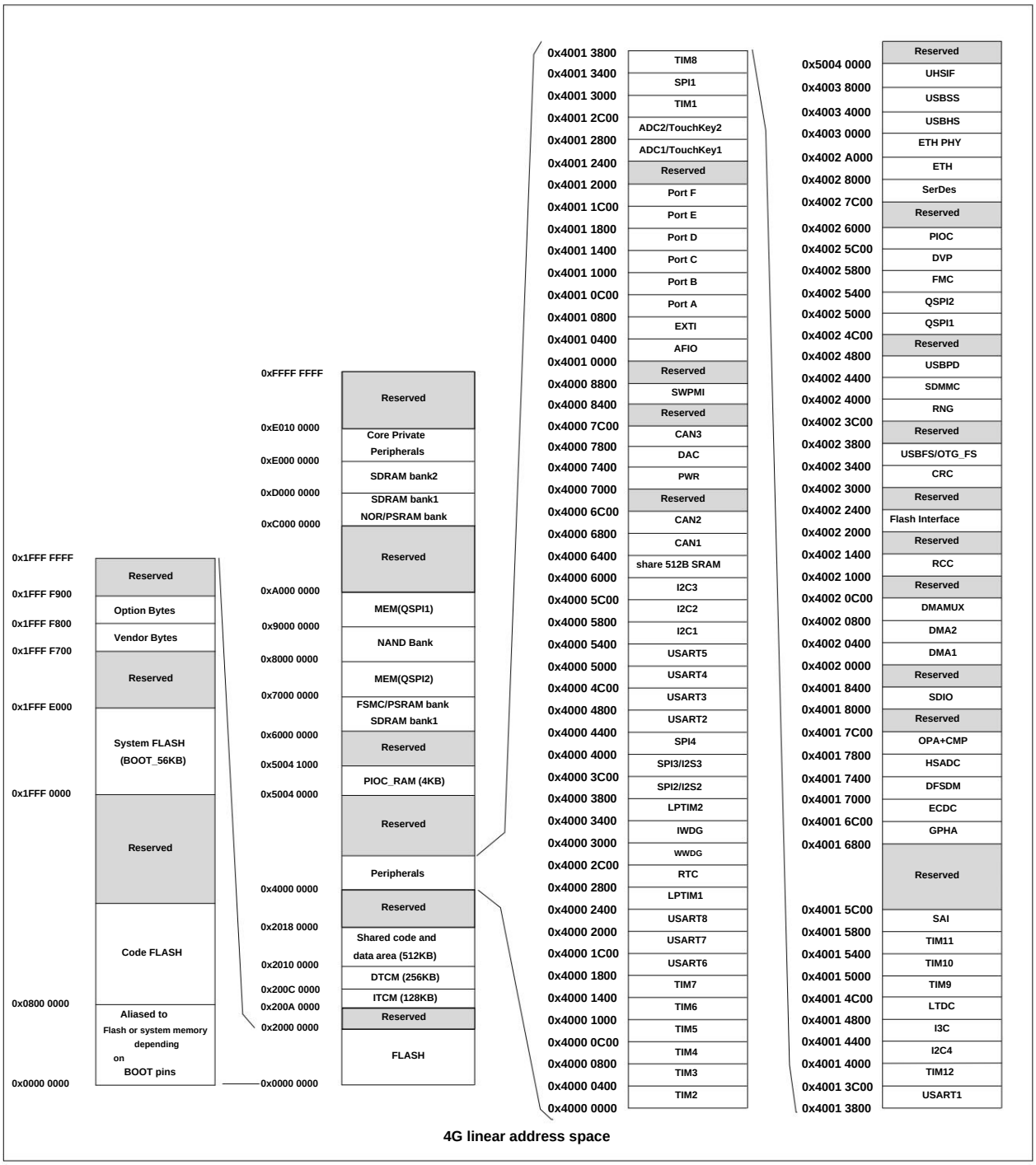
The system includes: a Flash access prefetch mechanism to accelerate code execution; a general-purpose DMA controller to reduce CPU load and improve efficiency; a clock tree hierarchical management system to reduce the overall power consumption of peripherals; and data protection mechanisms to increase system stability. The instruction bus (I-Code) connects the kernel and the FLASH instruction interface, where prefetching is performed . The data bus (D-Code) connects the kernel and the FLASH data interface, used for constant loading and debugging . The system bus connects the kernel and the bus matrix, coordinating access between the kernel, DMA, SRAM, and

The DMA bus connects the HB master interface to the bus matrix and is responsible for accessing FLASH data, SRAM, and peripherals. The bus matrix coordinates access between the system bus, data bus, DMA bus, and SRAM.

1.2 Memory Image

The CH32H417 series chips all include program memory, data memory, core registers, and peripheral registers, etc., all of which are... A 4GB linear addressing space. The system stores data in little-endian format, meaning the low byte is stored at the low address and the high byte is stored at the high address.

Figure 1-2 Storage Image



1.2.1 The memory allocation

includes a built-in SRAM of 896KB, partly used for the data area and partly for the zero-wait code area. Data is lost upon power failure.

The SRAM is divided into three blocks: a 128KB ITCM core 1 tightly coupled zero-wait code area, a 256KB DTCM core 1 tightly coupled zero-wait data area, and a...

The kernel contains a data area, the remaining 512KB of shared code and data. Additionally, kernel 1 has a 32KB instruction cache.

The 512KB shared area can be configured as the zero-wait code and data area for the RISC-V3F. It is recommended to configure it in 128KB units as needed.

distribute.

As an option, the 128KB ITCM and 256KB DTCM, totaling 384KB, can both be configured as the RISC-V5F code area.

The DTCM, as a code section, will add one clock cycle wait when a jump occurs.

The RISC-V3F can wait to access the ITCM or DTCM by pressing the HCLK clock for 2 seconds.

The RISC-V5F and RISC-V3F can access the 512KB shared area with zero wait time at the HCLK clock.

It has a built-in 256-byte non-volatile configuration information storage area for manufacturer configuration word storage. It is fixed before leaving the factory and cannot be modified by the user.

It has a built-in 256-byte user-defined information storage area for storing user-selected words.

The sizes of the user area and BOOT area are read through the DBMODE bit of the R32_FLASH_CFGR0 register:

When DBMODE=1, the chip has a built-in 960KB non-zero wait time program flash memory area (Code FLASH), i.e., the user area.

For user application and constant data storage, the equivalent frequency is approximately 25MHz. It has a built-in system FLASH memory of up to 56KB.

The BOOT area is used to store the system bootloader and contains a built-in bootloader.

When DBMODE=0, the chip has a built-in 480KB non-zero wait time program flash memory area (Code FLASH), i.e., the user area.

It stores user applications and constant data at an equivalent frequency of approximately 12.5MHz. It has a built-in system flash memory (BOOT area) of up to 28KB for

storing the system bootloader and includes a built-in bootloader.

Chapter 2 Power Control (PWR)

2.1 Overview

The CH32H417 chip system operates at a voltage VDD33 ranging from 2.4 to 3.6V, and its built-in voltage regulator provides the operating power required by the core.

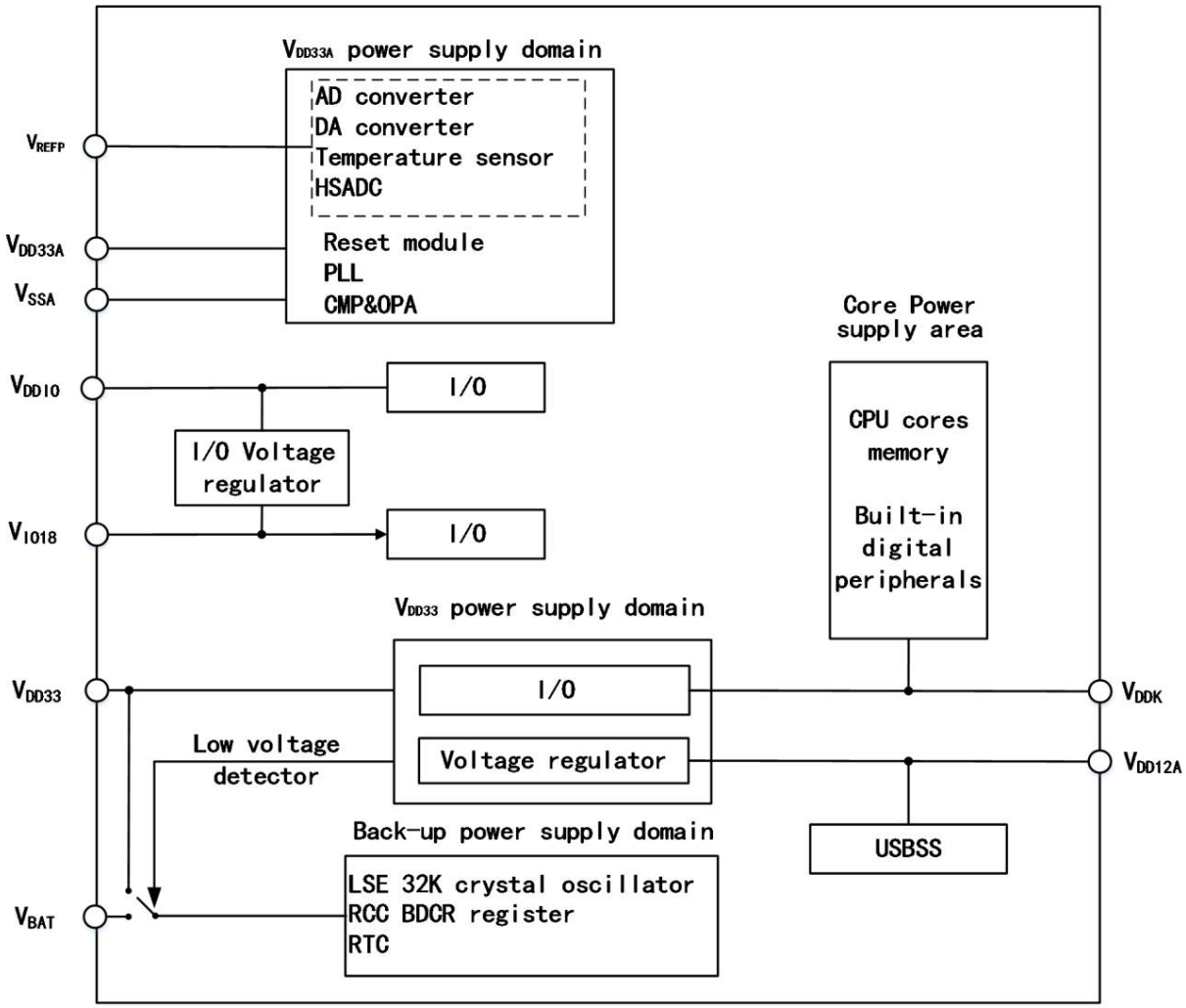
When the main power supply VDD33 fails, backup power sources such as batteries can power the real-time clock (RTC) via the VBAT pin. If no backup power is required...

For the source, it is recommended to connect VDD33 directly to the VBAT pin.

The VDD33A and VSSA pins are dedicated to powering analog-related circuitry in the system, including ADCs, DACs, temperature sensors, etc. VREFP is used as a...

For analog circuits such as ADCs, the reference point for the upper voltage limit is partially packaged as an independent pin, and if not packaged, it is equal to **VDD33A inside the chip .**

Figure 2-1 Power supply structure block diagram



2.2 Power Management

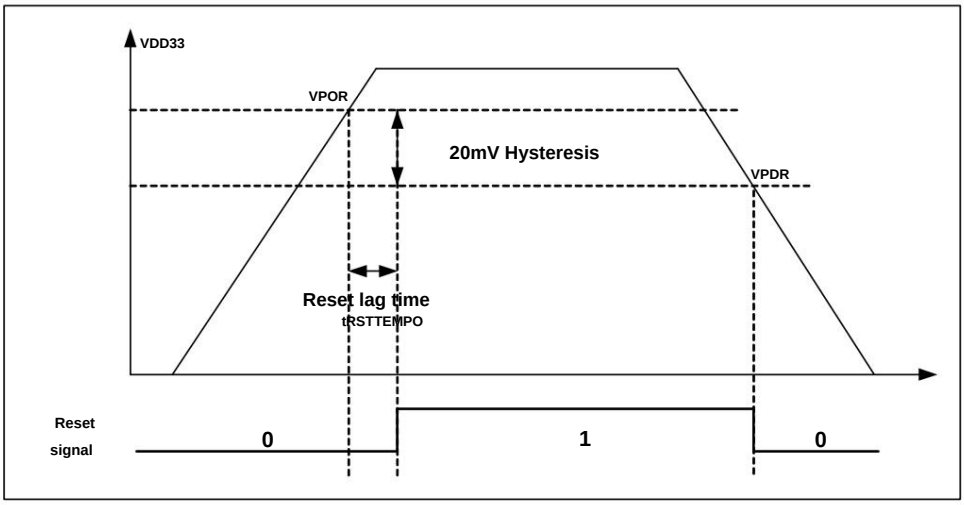
2.2.1 Power-on reset and power-off reset

The system integrates power-on reset (POR) and power-down reset (PDR) circuits. When the chip supply voltage VDD33 is lower than the corresponding threshold voltage,

The system is reset by the relevant circuitry, requiring no external reset circuitry. Please refer to the parameters for the power-on threshold voltage **VPOR and power-down threshold voltage VPDR.**

Refer to the corresponding data manual.

Figure 2-2 Schematic diagram of POR and PDR operation



2.2.2 Programmable Voltage Monitor (PVD) The

programmable voltage monitor (PVD) is primarily used to monitor changes in the system's main power supply, and interacts with the power control register PWR_CTLR.

By comparing the threshold voltage set by PLS[2:0], a status flag can be generated so that the system can be notified in time to perform operations such as data saving before power failure.

The specific configuration is as follows:

1) Set the PLS[2:0] field of the PWR_CTLR register and select the voltage threshold to be monitored.

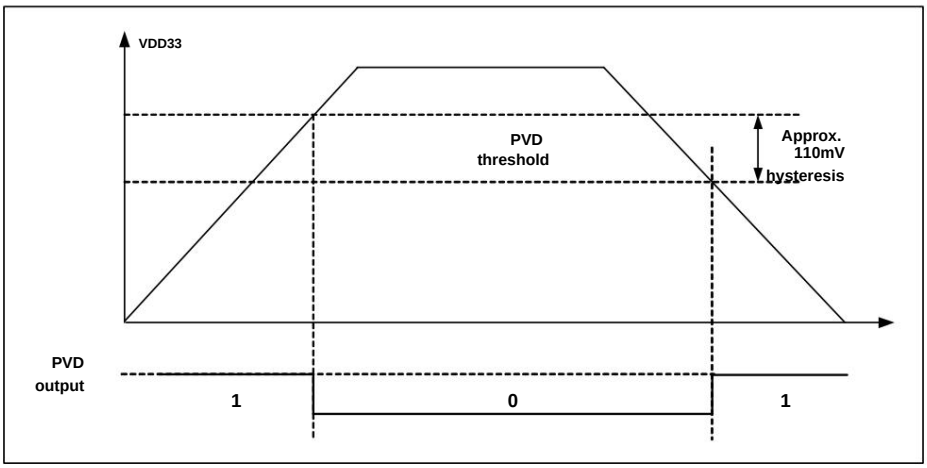
2) Set the PVDE bit in the PWR_CTLR register to enable the PVD function.

3) By reading the PVDO bit of the PWR_CSR status register, the relationship between the current system main power supply and the PLS[2:0] setting thresholds can be obtained, and corresponding actions can be performed.

Soft processing. When the VDD33 voltage is higher than the threshold set by PLS[2:0], the PVDO position is 0; when the VDD33 voltage is lower than the threshold set by PLS[2:0],

PVDO bit 1.

Figure 2-3 Schematic diagram of PVD operation



2.3 Low Power Mode

After the system reset, the microcontroller returns to normal operating mode (running mode). At this point, you can reduce the system clock speed or disable...

Save system power by not using external clocks or reducing the clock speed of operating external devices. If the system is not needed, it can be configured to enter low-power mode.

The system can exit a certain state by triggering a specific event.

Microcontrollers currently offer two low-power modes, distinguished by differences in the operation of the processor, peripherals, voltage regulators, etc.:

Sleep mode: The kernel stops running, but all peripherals (including kernel-private peripherals) continue to run.

Stop Mode: Stop mode only takes effect when both V3F and V5F are in stop mode. It stops all clocks; the system resumes operation after wake-up.

run.

Table 2-1 Overview of Low Power Modes

model	Enter	wake-up source	Impact on clock voltage regulator	
sleep	WFI	Arbitrary interrupt wake-up	Kernel clock off, Other clocks are unaffected.	normal
	WFE wake-up event wake-up			
stop	SLEEPDEEP (set to 1) WFI or WFE	Any external interrupt/event, RST External reset signal, IWDG reset Bit.	Both V3F and V5F are advanced. Enter Stop Mode: Off Close HSE, HSI, PLL and peripherals bell.	Normal: LPDS=0 or Low power consumption: LPDS=1

2.3.1 Low-power configuration options :

WFI and WFE modes

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

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

1) Configure an external or internal EXTI line to event mode; in this case, there is no need to configure an interrupt controller.

2) Interrupt response and interrupt exit flags are maintained as events in the interrupt controller;

3) Alternatively, configure the SEVONPEND bit to enable peripheral interrupts, but disable interrupts in the interrupt controller. After system wake-up, it will require... The interrupt pending bit needs to be cleared.

4) Debug request from the debugger.

SLEEPONEXIT

Enabled: After executing the WFI or WFE instruction, the microcontroller ensures that all pending interrupt services exit before entering low-power mode.

Disabled: After executing the WFI or WFE instruction, the microcontroller immediately enters low-power mode.

SEVONPEND

Enabled: All interrupt or wake-up events can wake up the low-power mode entered by executing WFE.

Disabled: Only interrupts or wake-up events enabled in the interrupt controller can wake up the low-power system that has been entered by executing WFE.

2.3.2 Sleep Mode (SLEEP) In this mode, all

IO pins maintain their running state, and all peripheral clocks function normally, thus entering sleep mode.

Before entering sleep mode, try to turn off unnecessary peripheral clocks to reduce power consumption. This mode has the shortest wake-up time.

Enter: Configure the kernel register control bit SLEEPDEEP=0, LPDS determines the internal voltage regulator state, and executes WFI or WFE (optional). SEVONPEND and SLEEPONEXIT.

Exit: Any interrupt or wake-up event.

2.3.3 Stop Mode (STOP) Stop mode

combines peripheral clock control with the deep sleep mode (SLEEPDEEP) of both the V3F and V5F kernels.

This mechanism allows the voltage regulator to operate in a lower power consumption state. Both V3F and V5F enter this mode: high-frequency clock (HSE/HSI/PLL). The domain is closed, but the SRAM and register contents are preserved, and the I/O pin states are maintained. The system can continue operating after waking up in this mode; HSI is the default. System clock.

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

Modules that can operate in stop mode: Independent Watchdog Timer (IWDG), Real-Time Clock (RTC), Low-Frequency Clock (LSI/LSE).

Enter: Configure kernel register control bit SLEEPDEEP=1, optional LPDS bit, execute WFI or WFE, optional SEVONPEND and SLEEPONEXIT.

Exit: Any external interrupt/event, external reset signal on RST, IWDG reset.

In stop mode, the LPDS bit is selectable. LPDS=0, the voltage regulator operates in normal mode; LPDS=1, the voltage regulator operates...

In low power mode.

2.3.4 RTC Automatic Wake-up

The RTC can automatically wake up without external interrupts. By programming the time base, it can periodically wake up from the stop mode.

Wake-up under certain conditions.

A precise external 32.768kHz low-frequency crystal oscillator (LSE) can be selected as the RTC clock source, or an internal LSI oscillator can be selected.

RTC clock source, LSI has worse accuracy and power consumption than LSE.

The RTC alarm event can wake the MCU from stop mode. To achieve this, external interrupt line 17 needs to be configured, and...

RTC is set to generate alarm events.

2.4 Register Description

Table 2-2 List of PWR Related Registers

name	Access Address Description: 0x40007000 Power	Reset value
R32_PWR_CTLR	Control Register; 0x40007004 Power	0x00000000
R32_PWR_CSR	Control/Status Register; 0x5003C000 System	0x00000X00
R32_SYS_CFGR0	Configuration Register 0	0xXXXXXXXX

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		VSEL_VIO18[2:0]			VIO_SW_CR	DBP	PLS[2:0]			PVDE	Reserved			LPDS

Bit	Name access		describe	Reset value
[31:13] Reserved		RO is reserved.		0
[12:10]	VSEL_VIO18[2:0]	RW	VIO18 power adjustment position: 000: 1.2V; 001: 1.8V; 010: 2.5V; 011: 3.3V; 100: VIO18 is powered off and is discharging at 10mA; 101: VIO18 is powered off, and the output is floating. 110: VIO18 is powered off, and the output is floating. 111: VIO18 is powered off, and the output is floating.	0
9	VIO_SW_CR	RW	VIO18 power adjustment mode selection bit: 1: Configure via VSEL_VIO18 software; 0: XO External pull-down resistor hardware configuration. Note: The initial power-on values of VIO18 are configured by hardware, i.e., via external XO. The pull-down resistor determines: Float: 1.8V; Pull-down 330K: 2.5V;	0

			Pull-down 100K: 1.2V. Note 1. For chips with externally packaged VIO18, users should 3.3V external demand for timepieces use shorting VDDIO VIO18, and after power-on, it needs to be set in software via VSEL_VIO18=111b. 2. For chips that do not externally encapsulate VIO18, VDDIO and VIO18 are shorted internally , while Furthermore, after power-on, it needs to be turned off in the software via VSEL_VIO18=111b. Turn off the LDO to reduce power consumption.	
8	DBP	RW	Write enable for the backup area: In the reset state, all registers in the backup area are write-protected. This bit must be set to 1 to allow write access to these registers. 1: Enable access to registers in the backup region; 0: Access to registers in the backup region is prohibited.	0
[7:5] PLS[2:0]		RW	PVD voltage monitoring threshold setting. See the electrical characteristics section of the datasheet for detailed instructions. Sexual aspect. 000: Rising edge 2.54V / Falling edge 2.44V; 001: Rising edge 2.6V / Falling edge 2.49V; 010: Rising edge 2.7V / Falling edge 2.59V; 011: Rising edge 2.8V / Falling edge 2.69V; 100: Rising edge 2.9V / Falling edge 2.79V; 101: Rising edge 3.0V / Falling edge 2.89V; 110: Rising edge 3.1V / Falling edge 2.99V; 111: Rising edge 3.2V / Falling edge 3.09V.	0
4	PVDE	RW	Power supply voltage monitoring function enable bit: 1. Enable power supply voltage monitoring function; 0: Disable power supply voltage monitoring function.	0
[3:1] Reserved		RO is reserved.		0
0	LPDS	RW	In stop mode, this is the voltage regulator operating mode selection bit. 1: The voltage regulator operates in low-power mode; 0: The voltage regulator is operating in normal mode.	0

2.4.2 Power Control/Status Register (PWR_CSR)

Offset Address: 0x04

31	30 29 28	27 26 25 24 23 22 21	2019	18	17 16
Reserved					

15	14 13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					VIO18_SR [1:0]		Reserved							PVDO

Bit	Name access		describe	Reset value
[31:10] Reserved		RO is reserved.		0
[9:8] VIO18_SR[1:0]		R0	VIO18 power initial status indicator bit: 00: 1.2V; 01: 2.5V; 1x: 1.8V.	x

			Note: The initial power-on value of VIO18 is determined by VIO_SW_CR and VSEL_VIO18[2:0]. Decide.	
[7:1] Reserved		RO is reserved.		0
0	PVDO	RO	PVD output status flag. When PVDE in the PWR_CTLR register is 1... At that time, this bit is valid. 1: VDD33 is lower than the PVD falling edge threshold set by PLS[2:0]; 0: VDD33 is higher than the PVD rising edge threshold set by PLS[2:0].	0

2.4.3 System Configuration Register 0 (SYS_CFGR0)

Access address: 0x5003C000

31	30 29 28	27	26	25 24 23 22 21	2019	18	17 16
Reserved							

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

Bit	Name access		describe	Reset value
[31:20] Reserved		RW is reserved and must retain its original value		x
[6:4] LDO_VDDK[2:0]		RW	when written. Adjusting the digital core voltage value, LDO_VDDK voltage adjustment bits: 011: Rated voltage; 101: Improved voltage optimization performance, only applicable to TA ÿ 70ÿ and heat dissipation. Excellent for commercial applications; Other: Reserved. When an external power supply is used in the application, it is recommended to set 001b or 000b.	011b
[3:0] Reserved		RW is reserved; the original value must be preserved when writing.		x

Chapter 3 Reset and Clock Control (RCC)

Based on the division of power supply zones and considerations for peripheral power consumption management in the application, the controller provides different reset methods and configurable features.
The clock tree structure. This section describes the scope of each clock in the system.

3.1 Key Features

Multiple reset methods

Multiple clock sources, bus clock management

Built -in external crystal oscillation monitoring and clock safety system

Independent clock management for each peripheral: reset, enable, disable

Supports internal clock output

3.2 Reset

The controller provides three reset methods: power reset, system reset, and backup area reset.

3.2.1 Power Reset When a

power reset occurs, all registers except the backup area (which is powered by VBAT) will be reset.

Its conditions for generation include:

ÿ Power-on/Power-off Reset (POR/PDR Reset)

3.2.2 System Reset When a

system reset occurs, all registers except the reset flag in the control/status register RCC_RSTSCKR and the backup area will be reset.
Registers. Identify the source of the reset event by examining the reset status flags in the RCC_RSTSCKR register.

Its conditions for generation include:

A low-level signal on the RST pin (external reset)

ÿ Window watchdog timer terminated (WWDG reset)

Independent watchdog timer terminated (IWDG reset)

Software reset (SW reset)

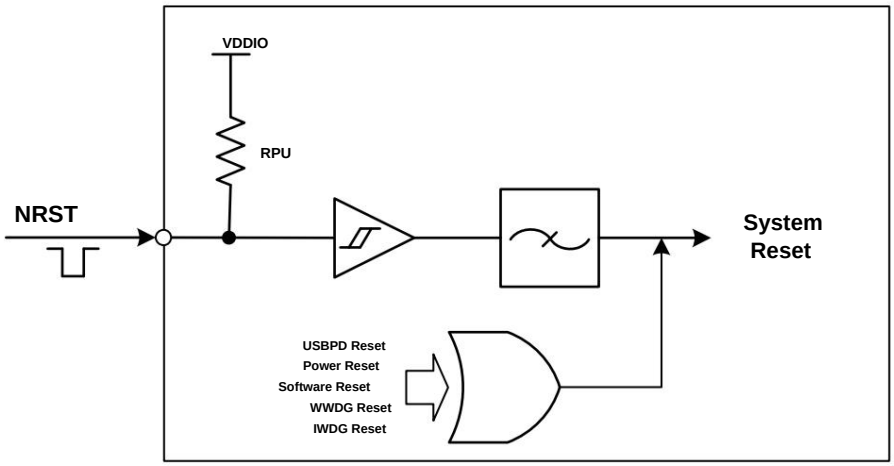
USBPD Reset

Window/Standalone Watchdog Reset: Triggered by the overflow of the window/standalone watchdog peripheral timer count cycle. See its description for details.
The corresponding chapter.

Software reset: Reset the system by setting the SYSRST bit to 1 in the interrupt configuration register PFIC_CFGR in the programmable interrupt controller (PFIC).
Reset the system by setting the SYSRST bit to 1 in the PFIC_SCTLR register or configuration register. Refer to the corresponding chapter for details.

USBPD Reset: When PD_RST_EN is 1, the CH32H417 supports resets generated by the USB PD signal frame Hard Reset; if
If IE_RX_RESET is also 1, then resets generated by the Cable Reset signal frame are also supported. USB PD does not have a reset flag, but the generated reset...
The effect is the same as software reset.

Figure 3-1 System Reset Structure



3.2.3 Backup Region Reset When a backup

region reset occurs, only the backup region registers are reset, including the RCC_BDCTLR register (RTC enable and LSE oscillation).

(Apparatus). Its production conditions include:

ÿ **Under** the premise that both **VDD33 and VBAT are powered down, the problem is caused by powering on VDD33 or** VBAT.

ÿ **The BDRST bit of the RCC_BDCTLR register is set to 1.**

3.3 Clock

3.3.1 System Clock Structure Diagram

3-2 Clock Tree Diagram (Applicable only to chips with the fifth bit of the batch number being 0)

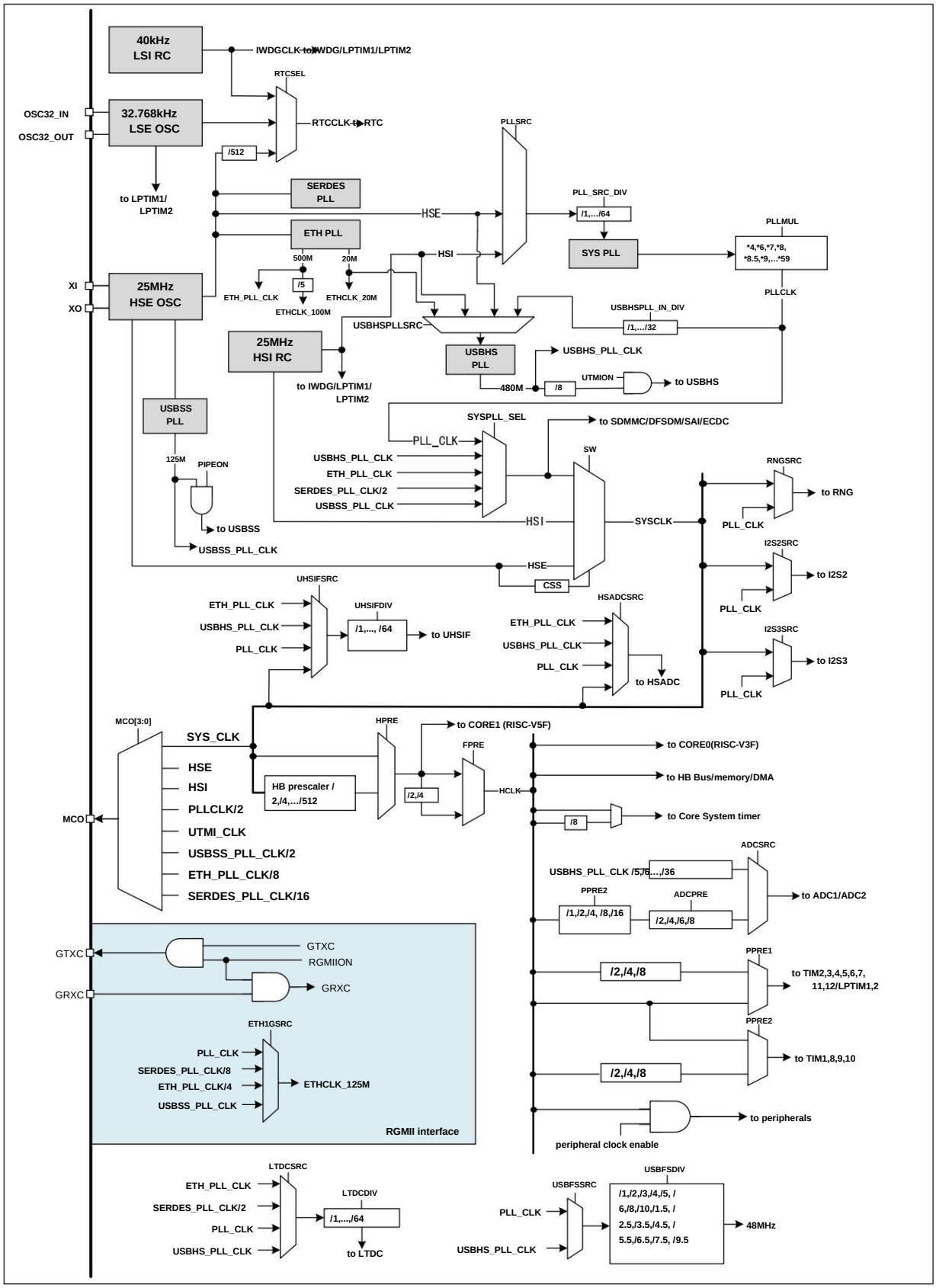
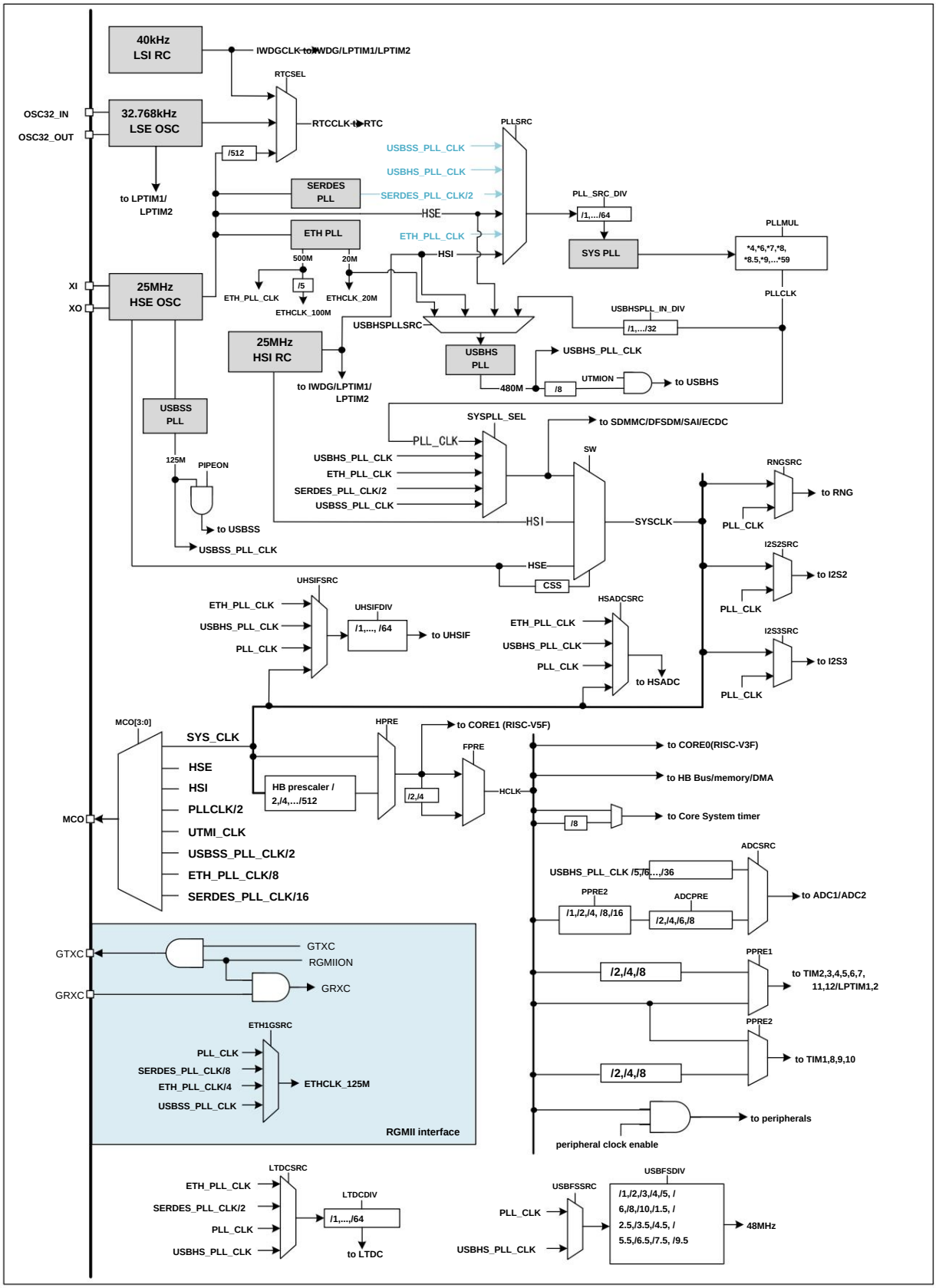


Figure 3-3 Clock tree diagram (applicable to chips with the fifth digit of the batch number being greater than 0)



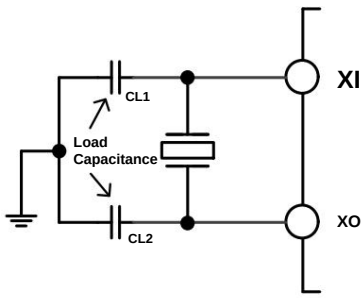
3.3.2 High-speed clock (HSI/HSE)

HSI is a high-speed clock signal generated by an internal 25MHz RC oscillator. The HSI RC oscillator can provide the system clock without any external components. It has a short startup time but relatively poor clock frequency accuracy. 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. HSION and HSIRDY are set to 1 by default (it is recommended not to disable them). Setting the HSIRDYIE bit in the RCC_INTR register will generate a corresponding interrupt. • Factory calibration: Differences in manufacturing processes can result in different RC oscillation frequencies for each chip. Therefore, HSI calibration is performed on each chip before it leaves the factory. After system reset, the factory calibration value is loaded into HSIAL[7:0] in the RCC_CTLR register. • User adjustment: Based on different voltages or ambient temperatures, the application can adjust the HSI frequency using the HSITRIM[4:0] bits in the RCC_CTLR register.

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

HSE is an external high-speed clock signal, generated by an external crystal/ceramic resonator or fed into the system via an external high-speed clock. External crystal/ceramic resonator (HSE crystal): An external oscillator provides a more accurate clock source for the system. Further information can be found in the electrical characteristics section of the datasheet. The HSE crystal can be enabled and disabled by setting the HSEON bit in the RCC_CTLR register. The HSERDY bit indicates whether the HSE crystal oscillation is stable; the hardware only sends the clock to the system after the HSERDY bit is set to 1. If the HSERDYIE bit in the RCC_INTR register is set, a corresponding interrupt will be generated. Figure 3-4

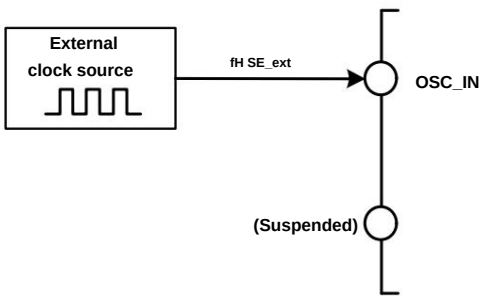
High-speed external crystal circuit.



Note: The load capacitor should be placed as close as possible to the oscillator pins, and the capacitance value should be selected according to the crystal manufacturer's specifications.

External high-speed clock source (HSE bypass): In this mode, an external clock source is directly fed to the XI pin, and the XO pin is left floating. A maximum frequency of 25MHz is supported. The application must enable the HSE bypass function by setting the HSEBYP bit when the HSEON bit is 0, and then setting the HSEON bit again.

Figure 3-5 High-speed clock source circuit



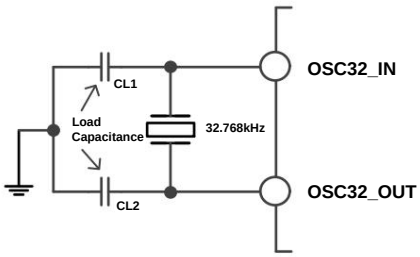
3.3.3 Low-speed clock (LSI/LSE)

LSI is a low-speed clock signal generated by the system's internal RC oscillator. It can continue operating in stop mode, providing a clock reference for the RTC clock, independent watchdog timer, and wake-up unit. Further information can be found in the electrical characteristics section of the datasheet. LSI can be enabled and disabled by setting the LSION bit in the RCC_RSTSCKR register, and the stability of the LSI RC oscillation is checked by polling the LSIrdy bit. The hardware only sends the clock signal after the LSIrdy bit is set to 1. Setting the LSIrdyie bit in the RCC_INTR register will generate a corresponding interrupt.

LSE is an external low-speed clock signal, generated by an external crystal/ceramic resonator or fed into the system via an external low-speed clock. It is for RTC.

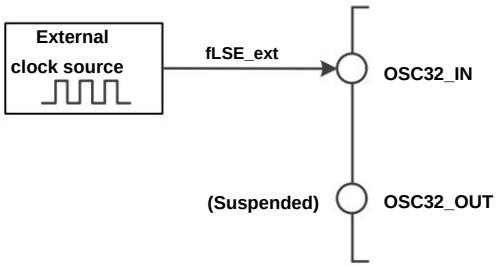
The clock or other timing function provides a low-power and accurate clock source. External crystal/

ceramic resonator (LSE crystal): An external 32.768kHz low-speed oscillator. The LSE is enabled and disabled by setting the LSEON bit in the RCC_BDCTLR register. The LSERDY bit indicates whether the LSE crystal oscillation is stable; the hardware sends the clock to the system only after the LSERDY bit is set to 1. If the LSERDYIE bit in the RCC_INTR register is set, a corresponding interrupt will be generated. Figure 3-6 Low-speed external crystal circuit.



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

Figure 3-7 Low-speed clock source circuit



3.3.4 PLL Clock: By configuring

the RCC_CFGR0 and RCC_PLLCFGR registers, the internal PLL clock can select from 6 clock sources and frequency multiplications.

These parameters must be set before each PLL is started, and once the PLL is started, they cannot be changed.

The PLLON bit in the RCC_CTLR register is enabled and disabled, and the PLLRDY bit indicates whether the PLL clock is stable. The hardware...

The clock is sent to the system only after PLLRDY is set to position 1.

The USBHS_PLLON bit in the RCC_CTLR register is enabled and disabled, while the USBHS_PLLRDY bit indicates the USBHS_PLL clock.

To determine if the system is stable, the hardware sends the clock signal to the system only after the USBHS_PLLRDY bit is set to 1.

The USBSS_PLLON bit in the RCC_CTLR register is enabled and disabled, while the USBSS_PLLRDY bit indicates the USBSS_PLL clock.

To determine if the system is stable, the hardware sends the clock signal to the system only after setting the USBSS_PLLRDY bit to 1.

The ETH_PLLON bit in the RCC_CTLR register is enabled or disabled, and the ETH_PLLRDY bit indicates whether the ETH_PLL clock is stable. The hardware sends the clock to the system only after the ETH_PLLRDY bit is set to 1. The SERDES_PLLON bit in the

RCC_CTLR register is enabled or disabled, and the SERDES_PLLRDY bit indicates whether the SERDES_PLL clock is stable.

To determine if the clock is stable, the hardware sends the clock to the system only after setting SERDES_PLLRDY to 1.

If the PLLRDYIE, ETHPLLRDYIE, or SERDESPLLRDYIE bits of the RCC_INTR register are set, the corresponding output will be generated.

It should be interrupted.

PLL clock sources: ÿ HSI

clock input ÿ HSE clock input ÿ

USBHS_PLL (480MHz) clock

input ÿ ETH_PLL (500MHz) clock input ÿ USBSS_PLL (125MHz)

clock input ÿ SERDES_PLL clock divided by 2

3.3.5 Bus/Peripheral

Clocks 3.3.5.1 System Clock (SYSCLK)

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

HSI as the system clock

HSE as the system clock

ÿ PLL clock as system clock

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

It will happen later.

3.3.5.2 HB Bus Peripheral Clock (HCLK)

By configuring the HPRE[3:0], PPRE1[2:0], and PPRE2[2:0] bits of the RCC_CFGR0 register, the timing of the HB bus can be configured.

These bus clocks determine the clock reference for peripheral interfaces connected to them. Applications can adjust different values to...

Reduce the power consumption of some peripherals when they are working.

Different peripheral modules can be reset by using the bits in the RCC_HBRSTR, RCC_HB1PRSTR, and RCC_HB2PRSTR registers.

It returns to its initial state.

Different peripherals can be individually enabled or disabled using individual bits in the RCC_HBPCENR, RCC_HB1PCENR, and RCC_HB2PCENR registers.

Module communication clock interface. When using a peripheral, you must first enable its clock bit before you can access its registers.

3.3.5.3 RTC Clock (RTCCLK) The

RTCCLK clock source can be determined by setting the RTCSEL[1:0] bits of the RCC_BDCTLR register, which can be HSE divided, LSE, or LSI.

Provided by clock. Before modifying this bit, ensure that the DBP bit in the power control register (PWR_CTLR) is set to 1. This is only possible after a backup area reset.

Reset this bit.

ÿ LSE as RTC clock: Since the LSE is in the backup domain and powered by VBAT, it can operate even if the VDD33 power supply is cut off , as long as VBAT maintains power .

The RTC continues to operate.

ÿ LSI as RTC clock: If the VDD33 power supply is cut off, automatic RTC wake-up cannot be guaranteed.

ÿ The HSE clock, divided by 512, is used as the RTC clock: if the VDD33 power supply is cut off or the internal voltage regulator is turned off (power supply in the 1.8V range).

If the circuit is cut off, the RTC state is uncertain.

3.3.5.4 Independent Watchdog Clock

If the independent watchdog timer has been configured by hardware or started by software, the LSI oscillator will be forced on and cannot be turned off.

After the LSI oscillator stabilizes, the clock is supplied to the IWDG.

3.3.5.5 Clock Output (MCO)

The microcontroller allows outputting a clock signal to the MCO pin. Configure the multiplexed push-pull output mode in the corresponding GPIO port register.

By configuring the MCO[3:0] bits of the RCC_CFGR0 register, the following 8 clock signals can be selected as the MCO clock output:

System clock (SYSCLK) output

HSI clock output

HSE clock output

ÿ PLL clock frequency divided by 2 output

ÿ UTMI clock output

ÿ USBSS_PLL clock divided by 2 output

ÿ ETH_PLL clock output after 8-fold frequency division

SERDES_PLL clock output after being divided by 16

3.3.5.6 USB Clock

The USBHS clock source is derived from the USBHSPLL clock, selected by configuring the USBHSPLLSRC bit in the RCC_PLLCFGR2 register.
USBHS_PLL clock source; if the input clock is selected as the divided SYS_PLL_CLK, the division factor can be obtained through USBHSPLL_IN_DIV[4:0].
Bit selection; the clock frequency is selected by configuring the USBHSPLL_REFSEL[1:0] bits. The clock frequency must match the USBHS_PLL input frequency.
The clock remains consistent; the USBHS_PLL clock is enabled by configuring the USBHS_PLLON bit in the RCC_CTLR register.

The USBFS/OTG_FS 48MHz clock source is either a PLL clock via a configurable divider or a USBFS_PLL clock.
Selectable by configuring the USBFSSRC bit in the RCC_CFGR2 register. Enabled by configuring the OTGFSEN bit in the RCC_HBPCENR register.
Enables USBOTG_FS clock.

The USBSS clock source comes from a USBSS_PLL clock, which outputs a 125MHz clock to USBSS. This is achieved by configuring the RCC_CTLR register.
Set the USBSS_PLLON bit of the device to enable the USBSS_PLL clock.

3.3.5.7 ETH Clock

Refer to section 31.4.3 for ETH clock configuration.

3.3.5.8 I2S and RNG Clock

The clock sources for I2S and RNG are PLL_CLK or the system clock (SYSCLK). I2S2, I2S3, and RNG can be configured separately.
The I2S2SRC, I2S3SRC, and RNGSRC bits of register RCC_CFGR2 select the clock source.

3.3.6 Clock Safety System The clock

safety system is an operational protection mechanism for the controller. It can switch to HSI in the event of an HSE clock transmission failure.
The clock is activated, and an interrupt notification is generated, allowing the application software to complete the rescue operation.

The clock safety system is activated by setting the CSSON bit in the RCC_CTLR register to 1. At this time, the clock monitor will oscillate at HSE.
The device is enabled after a startup delay (HSERDY=1) and disabled after the HSE clock is turned off. If the HSE clock fails during system operation...
If a fault occurs, the HSE oscillator will be shut down, and a clock failure event will be sent to the brake input of the advanced timers (TIM1 and TIM8), generating...
A clock-safe interrupt occurs, CSSF is set to 1, and the application enters an NMI non-maskable interrupt. CSSF can be cleared by setting the CSSC bit.
This bit flag can cancel the NMI interrupt suspension bit.

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...
The system will automatically switch the system clock to the HSI oscillator and shut down the HSE oscillator and PLL in the event of an HSE failure.

If an HSE failure event occurs while CSSON is enabled, it does not affect HSE and PLL enabling; the software can also resolve this within the NMI interrupt.
Process accordingly. If this is done before activating the clock safety system (CSSON=1), set the CSS_HSE_DIS bit in the RCC_CTLR register to 1.
If an HSE failure event occurs while CSSON is enabled, the hardware will disable HSE and all PLLs that use HSE.

3.3.7 RTC calibration This

function requires the PC13 pin to be configured as a general I/O port.

ÿPulse output

Configure the ASOE bit of the RCC_BDCTLR register to enable RTC pulse output, and set the ASOS bit to select between second pulse output and alarm output.
Clock pulse output.

ÿ RTC calibration

After configuring the CCO bit in the RCC_BDCTLR register, the internal RTC clock will be divided by 64 and output to the PC13 pin.
Through actual testing, the software was used to adjust the clock and calibrate the RTC by modifying the RTCCAL[6:0] bits.

3.3.8 BKP Interface Reset The BKP

area can be independently powered by VBAT even when the VDD33 main power supply is down . During the application code-controlled BKP area register reset, ASOS...
The BDRST bit and ASOE bit are reset under the software configuration of the RCC_BDCTLR register, and are not affected by the RCC peripheral interface control BKPRST bit.
ring.

3.4 Register Description

Table 3-1 List of RCC-related registers

name	Access Address Description: 0x40021000 Clock	Reset value
R32_RCC_CTLR	Control Register; 0x40021004 Clock	0x0000xx83
R32_RCC_CFGR0	Configuration Register 0; 0x40021008 PLL	0x00000000
R32_RCC_PLLCFGR	Configuration Register; 0x4002100C	0x00000004
R32_RCC_INTR	Clock Interrupt Register; 0x40021010 HB2	0x00000000
R32_RCC_HB2PRSTR	Peripheral Reset Register; 0x40021014 HB1	0x00000000
R32_RCC_HB1PRSTR	Peripheral Reset Register; 0x40021018 HB	0x00000000
R32_RCC_HBPCENR	Peripheral Clock Enable Register; 0x4002101C HB2	0x00000000
R32_RCC_HB2PCENR	Peripheral Clock Enable Register; 0x40021020 HB1	0x00000000
R32_RCC_HB1PCENR	Peripheral Clock Enable Register; 0x40021024	0x00000000
R32_RCC_BDCTLR	Backup Domain Control Register; 0x40021028	0x00000000
R32_RCC_RSTSCKR	Control/Status Register; 0x4002102C HB	0x08000000
R32_RCC_HBRSTR	Peripheral Reset Register; 0x40021030 Clock	0x00000000
R32_RCC_CFGR2	Configuration Register 2; 0x40021034 PLL	0x00000000
R32_RCC_PLLCFGR2	Configuration Register 2	0x00080020

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
CSS_HSE_DIS	Reserved	SERDES_PLLRDY	SERDES_PLLON	ETH_PLLRDY	ETH_PLLON	PLLRDY	PLLON	USBSS_PLLRDY	USBSS_PLLON	USBHS_PLLRDY	USBHS_PLLON	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			Reset value
31	CSS_HSE_DIS	RW	Description of the configuration for HSE failure events when the clock safety system is enabled: 1: When an HSE failure event occurs while CSSON is enabled, the hardware is shut down. HSE and all PLLs that use HSE; 0: When an HSE failure event occurs while CSSON is enabled, it does not affect HSE. The enabling of the PLL can be handled by the software within the NMI interrupt. Note: This bit can only be written when CSSON is 0.	0
30	Reserved	RO is reserved.		0
29	SERDES_PLLRDY	RO	SERDES_PLL lock flag: 1: SERDES_PLL clock lock; 0: SERDES_PLL clock is not locked.	0
28	SERDES_PLLON	RW	SERDES_PLL clock enable control bit: 1: Enable SERDES_PLL clock;	0

			0: Turn off the SERDES_PLL clock. The hardware is automatically reset after entering deep sleep.	
27	ETH_PLLRDY	RO	ETH_PLL lock flag: 1: ETH_PLL clock lock; 0: ETH_PLL clock is not locked.	0
26	ETH_PLLON	RW	ETH_PLL enable control bit: 1: Enable the ETH_PLL clock; 0: Turn off the ETH_PLL clock. The hardware is automatically reset after entering deep sleep.	0
25	PLLRDY	RO	PLL clock ready lock flag (set by hardware): 1: PLL clock lock; 0: PLL clock not locked.	0
twenty four	PLLON	RW	PLL clock enable control bit: 1: Enable PLL clock; 0: Turn off the PLL clock. Note: This bit is cleared to 0 by hardware after entering stop mode.	0
twenty three	USBSS_PLLRDY	RO	USBSS_PLL clock ready lock flag: 1: USBSS_PLL clock lock; 0: USBSS_PLL clock is not locked.	0
twenty two	USBSS_PLLON	RW	USBSS_PLL clock enable control bit: 1: Enable USBSS_PLL clock; 0: Disable USBSS_PLL clock. The hardware is automatically reset after entering deep sleep.	0
twenty one	USBHS_PLLRDY	RO	USBHS_PLL clock ready lock flag: 1: USBHS_PLL clock lock; 0: USBHS_PLL clock is not locked.	0
20	USBHS_PLLON	RW	USBHS_PLL clock enable control bit: 1: Enable USBHS_PLL clock; 0: Disable USBHS_PLL clock. The hardware is automatically reset after entering deep sleep.	0
19	CSSON	RW	Clock security system enable control bit: 1: Enable clock safety system. When HSE is ready (HSERDY set to 1), The hardware enabled clock monitoring for HSE and detected an abnormal HSE trigger. CSSF flag and NMI interrupt; hardware shutdown when HSE is not ready. Clock monitoring function for HSE. 0: Disable the clock security system.	0
18	HSEBYP	RW	External high-speed crystal bypass control bit: 1: Bypass external high-speed crystal/ceramic resonator (using external clock source); 0: High-speed external crystal/ceramic resonator without bypass. Note: This field must be written when HSEON is 0.	0
17	HSERDY	RO	External high-speed crystal oscillation stable and ready flag (set by hardware): 1: External high-speed crystal oscillation is stable; 0: The external high-speed crystal oscillation is not stable. Note: After clearing the position, this position needs to be cleared in one cycle: HSEON 0 6	0
16	HSEON	RW	External high-speed crystal oscillation enable control bit:	0

			1: Enable the HSE oscillator; 0: Turn off the HSE oscillator. Note: This bit is cleared to 0 by hardware after entering the stop low power mode.	
[15:8] HSICAL[7:0]		RO	internal high-speed clock calibration value, automatically initialized during system startup.	x
[7:3] HSITRIM[4:0]		RW	Internal high-speed clock adjustment value: The user can input an adjustment value to be added to the HSICAL[7:0] value. The frequency of the internal HSI RC oscillator is adjusted according to changes in voltage and temperature. Rate. The default value is 16, which allows you to adjust the HSI to 25MHz ± 0.25%; per step The HSICAL frequency is adjusted by approximately 50kHz.	10000b
2	Reserved	RO	reserved.	0
1	HSIRDY	RO	Internal high-speed clock (25MHz) stable and ready flag (set by hardware): 0: The internal high-speed clock (25MHz) is not stable; 1: Stable internal high-speed clock (25MHz). Note: After clearing the bit, this bit needs to be cleared in one cycle: HSION 0 6 HSI	1
0	HSION	RW	Internal high-speed clock (25MHz) enable control bit: 1: Enable HSI oscillator; 0: Turn off the HSI oscillator. Note: When returning from stop mode or using an external oscillator as the system clock. HSE 1.2MHz oscillator In the event of a fault, this bit is set by hardware to initiate internal [processing/operation]. RC	1

3.4.2 Clock Configuration Register 0 (RCC_CFGR0) Offset

Address: 0x04

31	30	29 28	27 26 25 24 23 22 21	2019	18	17 16
ADCSRC	ADC_DUTY_SEL	Reserved	MCO[3:0]	UTMION / PIPEON / RGMIION	Reserved	FPRE[1:0]

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

Bit	Name access		describe	Reset value
31	ADCSRC	RW	ADC input clock source selection: 1: USBHS_PLL (480MHz); 0: HCLK.	0
30	ADC_DUTY_SEL	RW	ADC clock duty cycle selection: 1: The ADC clock duty cycle is 75%; 0: The ADC clock duty cycle is 50%. Note: This bit is only valid when the input clock source is selected. ADC HCLK	0
[29:28] Reserved		RO	is reserved.	0
[27:24] MCO[3:0]		RW	Microcontroller MCO pin clock output control: 00xx: No clock output; 0100: System clock (SYSCLK) output;	0

			0101: Internal 25MHz RC oscillator clock (HSI) output; 0110: External Oscillator Clock (HSE) Output; 0111: Output after PLL clock frequency division by 2; 1000: UTMI clock output; 1001: Output of USBSS_PLL clock after being divided by 2; 1010: Output of ETH_PLL clock after being divided by 8; 1011: SERDES_PLL clock output after being divided by 16; Other: Reserved.	
twenty three	UTMION	RW	UTMI clock gating enable: 1: Enable; 0: Off.	0
twenty two	PIPEON	RW	PIPE clock gating enable: 1: Enable; 0: Off.	0
twenty one	RGMIION	RW	Whether the 1000M Ethernet RGMII interface is enabled and clock enabled: 1: Enable the 1000M Ethernet RGMII interface and enable the clock; 0: Disable and turn off the clock.	0
[20:18] Reserved		RO is reserved.		0
[17:16] FPRE[1:0]		RW	HCLK input clock source prescaler control: 00: No frequency division; 01:2 frequency division; 1x: 4 frequency division.	0
[15:14] ADCPRE[1:0]		RW	ADC clock source selection USBHS_PLL prescaler control high 2 bits (in conjunction with...) RCC_CFGR0 register bits [13:11] and PPRE2 [2:0] are used): 00000: 5-fold frequency division; 00001: 6-division frequency; 00010: 7-division frequency; 00011: 8-frequency divider; ... 11110:35 frequency division; 11111:36 frequency division. ADC clock source selection HCLK prescaler control: 00:2 frequency division; 01:4 frequency division; 10:6 frequency division; 11:8 frequency division. Note: The ADC clock should not exceed 80MHz.	0
[13:11] PPRE2[2:0]		RW	When applied to an ADC, the USBHS_PLL prescaler is selected as the clock source. Controls the lower 3 bits (in conjunction with RCC_CFGR0 register bits [15:14]) ADCPRE[1:0] is used). When applied to an ADC, the clock source is selected using HCLK prescaler control: 0xx: No frequency division;	0

			100:2 frequency division; 101:4 frequency division; 110:8 frequency division; 111:16 frequency division. When applied to timers TIM1/8/9/10, the clock source prescaler control is as follows: Other: No frequency division; 101:2 frequency division; 110:4 frequency division; 111:8 frequency division.	
[10:8] PPRE1[2:0]		RW	Operating on TIM2/3/4/5/6/7/11/12 and LPTIM1/2, clock source Prescaler control: Other: No frequency division; 101:2 frequency division; 110:4 frequency division; 111:8 frequency division.	0
[7:4] HPRE[3:0]		RW	HB clock source prescaler control: 0xxx: SYSCLK is not divided; 1000: SYSCLK 2 divider; 1001: SYSCLK divided by 4; 1010: SYSCLK 8 divider; 1011: SYSCLK divided by 16; 1100: SYSCLK 64 divider; 1101: SYSCLK 128 divider; 1110: SYSCLK 256 divider; 1111: SYSCLK 512 frequency divider.	0
[3:2] SWS[1:0]		RO	System clock (SYSCLK) status (hardware set): 00: The system clock source is HSI; 01: The system clock source is HSE; 10: The system clock source is a PLL; 11: Unavailable.	0
[1:0] SW[1:0]		RW	Select system clock source: 00: HSI serves as the system clock; 01: HSE serves as the system clock; 10: The PLL output is used as the system clock; 11: Unavailable. Note: With clock safety enabled (CSSON=1), when returning from stop mode or using an external oscillator as the system clock. At that time, the hardware forces the selection of HSI as the system clock.	0

3.4.3 PLL Configuration Register (RCC_PLLCFGR) Offset

Address: 0x08

31	30 29 28	27 26 25 24 23 22 21 2019 18 17 16
SYSPLL	SYSPLL_SEL[2:0]	Reserved

L_GATE		

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

Bit	Name access		describe	Reset value
31	SYSPLL_GATE	This bit must be set before the RW system clock switches to the		0
[30:28]	SYSPLL_SEL[2:0]	RW	PLL. The system clock switching to PLL selection is only enabled when SYSPLL_GATE is 0. Allow writing: 0xx: PLL_CLK; 100: USBHS_PLL (480MHz); 101: ETH_PLL (500MHz); 110: SERDES_PLL/2; 111: USBSS_PLL (125MHz).	0
[27:14] Reserved		RO is reserved.		0
[13:8]	PLL_SRC_DIV[5:0]	RW	The PLL input clock source divider can only be written to when PLLON is 0: 000000: 1 frequency divider; 000001: Frequency division by 2; 000010: 3 frequency divider; … 111111:64 frequency division.	0
[7:5] PLLSRC[2:0]		RW	The PLL input clock source can only be written to when PLLON is 0: 000: HSI; 001: HSE; 100: USBHS_PLL (480MHz); 101: ETH_PLL (500MHz); 110: USBSS_PLL (125MHz); 111: SERDES_PLL/2. Note: The input reference clock is derived from the clock source mentioned above.	0
[4:0] PLLMUL[4:0]		RW	host PLL PLL_SRC_DIV divides the frequency, and PLLMUL multiplies it. The PLL clock multiplication factor can only be written when PLLON is 0: 00000: PLL 4x frequency multiplier output; 00001: PLL 6x frequency multiplier output; 00010: PLL 7x frequency multiplier output; 00011: PLL 8x frequency multiplier output; 00100: PLL 8.5x frequency multiplier output; 00101: PLL 9x frequency multiplier output; 00110: PLL 9.5x frequency multiplier output; 00111: PLL 10x frequency multiplier output; 01000: PLL 10.5x frequency multiplier output; 01001: PLL 11x frequency multiplier output; 01010: PLL 11.5x frequency multiplier output; 01011: PLL 12x frequency multiplier output; 01100: PLL 12.5x frequency multiplier output;	00100b

			01101: PLL 13x frequency multiplier output; 01110: PLL 14x frequency multiplier output; 01111: PLL 15x frequency multiplier output; 10000: PLL 16x frequency multiplier output; 10001: PLL 17x frequency multiplier output; 10010: PLL 18x frequency multiplier output; 10011: PLL 19x frequency multiplier output; 10100: PLL 20x frequency multiplier output; 10101: PLL 22x frequency multiplier output; 10110: PLL 24x multiplier output; 10111: PLL 26x frequency multiplier output; 11000: PLL 28x multiplier output; 11001: PLL 30x frequency multiplier output; 11010: PLL 32x frequency multiplier output; 11011: PLL 34x frequency multiplier output; 11100: PLL 36x frequency multiplier output; 11101: PLL 38x frequency multiplier output; 11110: PLL 40x frequency multiplier output; 11111: PLL 59x frequency multiplier output.	

3.4.4 Clock Interrupt Register (RCC_INTR) Offset

Address: 0x0C

31	30 29 28	27 26 25 24 23 22 21	2019	18	17 16
Reserved			CSSC	SERDESPLLRDYC	ETHPLLRDYC	PLLRDYC	HSERDYC	HSIRDYC	LSERDYC	LSIRDYC

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

Bit	Name access		describe	Reset value
[31:24] Reserved		RO reserved.		0
twenty three	CSSC	WO	Clear the Clock Safety System Interrupt Flag (CSSF): 1: Clear the CSSF break flag; 0: No action.	0
twenty two	SERDESPLLRDYC	WO	Clear the SERDES_PLL ready interrupt flag: 1: Clear the SERDESRDYF interrupt flag; 0: No action.	0
twenty one	ETHPLLRDYC	WO	Clear the ETH_PLL ready interrupt flag: 1: Clear the ETHRDYF interrupt flag; 0: No action.	0
20	PLLRDYC	WO	Clear the PLL ready interrupt flag: 1: Clear the PLLRDYF interrupt flag;	0

			0: No action.	
19	HSERDYC	WO	Clear the HSE oscillator ready interrupt flag: 1: Clear the HSERDYF interrupt flag; 0: No action.	0
18	HSIRDYC	WO	Clear the HSI oscillator ready interrupt flag: 1: Clear the HSIRDYF interrupt flag; 0: No action.	0
17	LSERDYC	WO	Clear the LSE oscillator ready interrupt flag: 1: Clear the LSERDYF interrupt flag; 0: No action.	0
16	LSIRDYC	WO	Clear the LSI oscillator ready interrupt flag: 1: Clear the LSIRDYF interrupt flag; 0: No action.	0
15	Reserved	RO is reserved.		0
14	SERDESPLLRDYIE RW		SERDES_PLL Ready interrupt enable bit: 1: Enable SERDES_PLL ready interrupt; 0: Disable SERDES_PLL ready interrupts.	0
13	ETHPLLRDYIE	RW	ETH_PLL Ready interrupt enable bit: 1: Enable ETH_PLL ready interrupt; 0: Disable ETH_PLL ready interrupt.	0
12	PLLRDYIE	RW	PLL ready interrupt enable bit: 1: Enable PLL ready interrupt; 0: Disable PLL ready interrupt.	0
11	HSERDYIE	RW	HSE Ready Interrupt Enable Bit: 1: Enable HSE ready interrupt; 0: HSE ready interrupt disabled.	0
10	HSIRDYIE	RW	HSI Ready Interrupt Enable Bit: 1: Enable HSI ready interrupt; 0: Disable HSI ready interrupt.	0
9	LSERDYIE	RW	LSE (Ready Interrupt Enable) bit: 1: Enable LSE ready interrupt; 0: Disable LSE ready interrupt.	0
8	LSIRDYIE	RW	LSI Ready Interrupt Enable Bit: 1: Enable LSI ready interrupt; 0: Disable LSI ready interrupt.	0
7	CSSF	RO	Clock safety system interrupt flag: 1: The HSE clock failed, generating a Clock Safety Interrupt (CSSI). 0: Clockless safety system interrupt. Hardware sets the bit, software writes CSSC bit 1 and clears it.	0
6	SERDESPLLRDYF	RO	SERDES_PLL Clock Ready Lock Interrupt Flag: 1: SERDES_PLL clock lock generates an interrupt; 0: No SERDES_PLL clock lock interrupt. Hardware sets the bit, software writes SERDESRDYC bit 1 to clear it.	0
5	ETHPLLRDYF	RO	ETH_PLL clock ready interrupt flag: 1: An interrupt was generated by ETH_PLL clock locking;	0

			0: No ETH_PLL clock lock interrupt. Hardware sets the bit, software writes ETHPLLRDYC bit 1 to clear it.	
4	PLLRDYF	RO	PLL clock ready to lock interrupt flag: 1: PLL clock lock generates an interrupt; 0: No PLL clock lock interrupt. Hardware sets the bit, software writes PLLRDYC bit 1 to clear it.	0
3	HSERDYF	RO	HSE clock ready interrupt flag: 1: An interrupt is generated when the HSE clock is ready; 0: No HSE clock ready interrupt. Hardware sets the bit, software writes HSERDYC bit 1 to clear it.	0
2	HSIRDYF	RO	HSI Clock Ready Interrupt Flag: 1: An interrupt is generated when the HSI clock is ready; 0: No HSI clock ready interrupt. Hardware sets the bit, software writes HSIRDYC bit 1 to clear it.	0
1	LSERDYF	RO	LSE clock ready interrupt flag: 1: An interrupt is generated when the LSE clock is ready; 0: No LSE clock ready interrupt. Hardware sets the bit, software writes LSERDYC bit 1 to clear it.	0
0	LSIRDYF	RO	LSI Clock Ready Interrupt Flag: 1: An interrupt is generated when the LSI clock is ready; 0: No LSI clock ready interrupt. Hardware sets the bit, software writes LSIRDYC bit 1 to clear it.	0

3.4.5 HB2 Peripheral Reset Register (RCC_HB2PRSTR) Offset Address:

0x10

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
I3C RST	LTDCRST	Reserved		GPHA RST	ECDCRST	DFSDM RST	Reserved	OPCMRST	TIM12 RST	TIM11 RST	TIM10 RST	TIM9 RST	SDIO RST	Reserved	SAI RST

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2C4 RST	USART1 RST	TIM8 RST	SPI1 RST	TIM1 RST	ADC2 RST	ADC1 RST	Reserved	IOPF RST	IOPE RST	IOPD RST	IOPC RST	IOPB RST	IOPA RST	HSADC RST	AFIO RST

Bit	Name access		describe	Reset value
31	I3CRST	RW	I3C module reset control: 1: Reset module; 0: No effect.	0
30	LTDCRST	RW	LTDC module reset control: 1: Reset module; 0: No effect.	0
[29:28] Reserved		R0 reserved.		0
27	GPHARST	RW	GPHA module reset control: 1: Reset module; 0: No effect.	0
26	ECDCRST	RW	ECDC module reset control: 1: Reset module; 0: No effect.	0
25	DFSDMRST	RW DFSDM module reset control:		0

			1: Reset module; 0: No effect.	
twenty four	Reserved	R0 reserved.		0
twenty three	OPCMRST	RW	OPCM module reset control: 1: Reset module; 0: No effect.	0
twenty two	TIM12RST	RW	TIM12 module reset control: 1: Reset module; 0: No effect.	0
twenty one	TIM11RST	RW	TIM11 module reset control: 1: Reset module; 0: No effect.	0
20	TIM10RST	RW	TIM10 module reset control: 1: Reset module; 0: No effect.	0
19	TIM9RST	RW	TIM9 module reset control: 1: Reset module; 0: No effect.	0
18	SDIORST	RW	SDIO module reset control: 1: Reset module; 0: No effect.	0
17	Reserved	R0 reserved.		0
16	SAIRST	RW	SAI module reset control: 1: Reset module; 0: No effect.	0
15	I2C4RST	RW	I2C4 module reset control: 1: Reset module; 0: No effect.	0
14	USART1RST	RW	USART1 module reset control: 1: Reset module; 0: No effect.	0
13	TIM8RST	RW	TIM8 module reset control: 1: Reset module; 0: No effect.	0
12	SPI1RST	RW	SPI1 module reset control: 1: Reset module; 0: No effect.	0
11	TIM1RST	RW	TIM1 module reset control: 1: Reset module; 0: No effect.	0
10	ADC2RST	RW	ADC2 module reset control: 1: Reset module; 0: No effect.	0
9	ADC1RST	RW	ADC1 module reset control: 1: Reset module; 0: No effect.	0
8	Reserved	R0 reserved.		0
7	IOPFRST	RW	PF port module reset control for IO: 1: Reset module; 0: No function.	0
6	IOPERST	RW	IO PE port module reset control: 1: Reset module; 0: No effect.	0
5	IOPDRST	RW	PD port module reset control for I/O: 1: Reset module; 0: No function.	0
4	IOPCRST	RW	PC port module reset control for IO: 1: Reset module; 0: No function.	0
3	IOPBRST	RW	IO PB port module reset control: 1: Reset module; 0: No function.	0
2	IOPARST	RW	IO PA port module reset control: 1: Reset module; 0: No function.	0

1	HSADCRST	RW	HSADC module reset control: 1: Reset module; 0: No effect.	0
0	AFIORST	RW	IO auxiliary function module reset control: 1: Reset module; 0: No effect.	0

3.4.6 HB1 Peripheral Reset Register (RCC_HB1PRSTR) Offset Address:

0x14

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
SWPMI RST	I2C3 RST	DAC RST	PWR RST	BKP RST	CAN2 RST	CAN1 RST	CAN3 RST	Reser ved	I2C2 RST	I2C1 RST	USART5 RST	USART4 RST	USART3 RST	USART2 RST	SPI4 RST

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3 RST	SPI2 RST	QSPI 2 RST	QSPI1 RST	WWDG RST	LPTI M2 RST	LPTI M1 RST	USAR T8RS T	USAR T7RS T	USAR T6RS T	TIM7 RST	TIM6 RST	TIM5 RST	TIM4 RST	TIM3 RST	TIM2 RST

Bit	Name access		describe	Reset value
31	SWPMIRST	RW	SWPMI module reset control: 1: Reset module; 0: No effect.	0
30	I2C3RST	RW	I2C3 module reset control: 1: Reset module; 0: No effect.	0
29	DACRST	RW	DAC module reset control: 1: Reset module; 0: No effect.	0
28	PWRRST	RW	Power interface module reset control: 1: Reset module; 0: No effect.	0
27	BKPRST	RW	Backup unit reset control: 1: Reset module; 0: No effect.	0
26	CAN2RST	RW	CAN2 module reset control: 1: Reset module; 0: No effect.	0
25	CAN1RST	RW	CAN1 module reset control: 1: Reset module; 0: No effect.	0
twenty four	CAN3RST	RW	CAN3 module reset control: 1: Reset module; 0: No effect.	0
twenty three	Reserved	RO reserved.		0
twenty two	I2C2RST	RW	I2C2 module reset control: 1: Reset module; 0: No effect.	0
twenty one	I2C1RST	RW	I2C1 module reset control: 1: Reset module; 0: No effect.	0
20	USART5RST	RW	USART5 module reset control: 1: Reset module; 0: No effect.	0
19	USART4RST	RW	USART4 module reset control: 1: Reset module; 0: No effect.	0
18	USART3RST	RW	USART3 module reset control: 1: Reset module; 0: No effect.	0

17	USART2RST	RW	USART2 module reset control: 1: Reset module; 0: No effect.	0
16	SPI4RST	RW	SPI4 module reset control: 1: Reset module; 0: No effect.	0
15	SPI3RST	RW	SPI3 module reset control: 1: Reset module; 0: No effect.	0
14	SPI2RST	RW	SPI2 module reset control: 1: Reset module; 0: No effect.	0
13	QSPI2RST	RW	QSPI2 module reset control: 1: Reset module; 0: No effect.	0
12	QSPI1RST	RW	QSPI1 module reset control: 1: Reset module; 0: No effect.	0
11	WWDGRST	RW	Window watchdog reset control: 1: Reset module; 0: No effect.	0
10	LPTIM2RST	RW	LPTIM2 module reset control: 1: Reset module; 0: No effect.	0
9	LPTIM1RST	RW	LPTIM1 module reset control: 1: Reset module; 0: No effect.	0
8	USART8RST	RW	USART8 module reset control: 1: Reset module; 0: No effect.	0
7	USART7RST	RW	USART7 module reset control: 1: Reset module; 0: No effect.	0
6	USART6RST	RW	USART6 module reset control: 1: Reset module; 0: No effect.	0
5	TIM7RST	RW	Timer 7 module reset control: 1: Reset module; 0: No effect.	0
4	TIM6RST	RW	Timer 6 module reset control: 1: Reset module; 0: No effect.	0
3	TIM5RST	RW	Timer 5 module reset control: 1: Reset module; 0: No effect.	0
2	TIM4RST	RW	Timer 4 module reset control: 1: Reset module; 0: No effect.	0
1	TIM3RST	RW	Timer 3 module reset control: 1: Reset module; 0: No effect.	0
0	TIM2RST	RW	Timer 2 module reset control: 1: Reset module; 0: No effect.	0

3.4.7 HB Peripheral Clock Enable Register (RCC_HBPCENR) Offset

Address: 0x18

31	30 29 28	27 26 25 24 23 22 21	2019	18	17	16
Reserved			PIOCEN	Reserved	SERDESEN	USBPDEN	UHSIFEN	USBOTGEN	Reserved

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

Reserved	ETHMACEN	DVPEN	USBSSEN	USBHSEN	SDMMCEN	RNGEN	FMCEN	Reserved	CRCEN	Reserved	DMA2EN	DMA1EN

Bit	Name access		describe	Reset value
[31:23] Reserved		RO is reserved.		0
twenty two	PIOCEN	RW	PIOC module clock enable bit: 1: Module clock on; 0: Module clock off.	0
twenty one	Reserved	RO is reserved.		0
20	SERDESEN	RW	SERDES module clock enable bit: 1: Module clock on; 0: Module clock off.	0
19	USBPDEN	RW	USBPD module clock enable bit: 1: Module clock on; 0: Module clock off.	0
18	UHSIFEN	RW	HSPP module clock enable bit: 1: Module clock on; 0: Module clock off.	0
17	OTGFSEN	RW	USBOTG_FS module clock enable bit: 1: Module clock on; 0: Module clock off.	0
[16:15] Reserved		RO reserved.		0
14	ETHMACEN	RW	Ethernet MAC clock enabled: 1: Ethernet MAC clock enabled; 0: Ethernet MAC clock is off.	0
13	DVPEN	RW	DVP module clock enable bit: 1: Module clock on; 0: Module clock off.	0
12	USBSSEN	RW	USBSS module clock enable bit: 1: Module clock on; 0: Module clock off.	0
11	USBHSEN	RW	USBHS module clock enable bit: 1: Module clock on; 0: Module clock off.	0
10	SDMMCEN	RW	SDMMC module clock enable bit: 1: Module clock on; 0: Module clock off.	0
9	RNGEN	RW	RNG module clock enable bit: 1: Module clock on; 0: Module clock off.	0
8	FMCEN	RW	FMC module clock enable bit: 1: Module clock on; 0: Module clock off.	0
7	Reserved	RO is reserved.		0
6	CRCEN	RW	CRC module clock enable bit: 1: Module clock on; 0: Module clock off.	0
[5:2] Reserved		RO is reserved.		0
1	DMA2EN	RW	DMA2 module clock enable bit: 1: Module clock on; 0: Module clock off.	0
0	DMA1EN	RW	DMA1 module clock enable bit: 1: Module clock on; 0: Module clock off.	0

3.4.8 HB2 Peripheral Clock Enable Register (RCC_HB2PCENR) Offset

Address: 0x1C

31 30 29 28 27 26 25 24 23 22 21 2019 18 17 16

I3C EN	LTDC EN	Reserved	GPHA EN	ECDC EN	DFSDM EN	Reserved	OPCM EN	TIM12 EN	TIM11 EN	TIM10 EN	TIM9 EN	SDIO EN	Reserved	SAI EN
15	14	13 12	11	10	9	8	7	6	5	4	3	2	1	0
I2C4 EN	USART1 EN	TIM8 EN	SPI1 EN	TIM1 EN	ADC2 EN	ADC1 EN	Reserved	IOPF EN	IOPE EN	IOPD EN	IOPC EN	IOPB EN	IOPA EN	HSADC EN	AFIO EN

Bit	Name access		describe	Reset value
31	I3CEN	RW	I3C module clock enable bit: 1: Module clock on; 0: Module clock off.	0
30	LTDCEN	RW	LTDC module clock enable bit: 1: Module clock on; 0: Module clock off.	0
[29:28] Reserved		RO is reserved.		0
27	GPHAEN	RW	GPHA module clock enable bit: 1: Module clock on; 0: Module clock off.	0
26	ECDCEN	RW	ECDC module clock enable bit: 1: Module clock on; 0: Module clock off.	0
25	DFSDMEN	RW	DFSDM module clock enable bit: 1: Module clock on; 0: Module clock off.	0
twenty four	Reserved	RO is reserved.		0
twenty three	OPCMEN	RW	OPCM module clock enable bit: 1: Module clock on; 0: Module clock off.	0
twenty two	TIM12EN	RW	TIM12 module clock enable bit: 1: Module clock on; 0: Module clock off.	0
twenty one	TIM11EN	RW	TIM11 module clock enable bit: 1: Module clock on; 0: Module clock off.	0
20	TIM10EN	RW	TIM10 module clock enable bit: 1: Module clock on; 0: Module clock off.	0
19	TIM9EN	RW	TIM9 module clock enable bit: 1: Module clock on; 0: Module clock off.	0
18	SDIOEN	RW	SDIO module clock enable bit: 1: Module clock on; 0: Module clock off.	0
17	Reserved	RO is reserved.		0
16	SAIEN	RW	SAI module clock enable bit: 1: Module clock on; 0: Module clock off.	0
15	I2C4EN	RW	I2C4 module clock enable bit: 1: Module clock on; 0: Module clock off.	0
14	USART1EN	RW	USART1 module clock enable bit: 1: Module clock on; 0: Module clock off.	0
13	TIM8EN	RW	TIM8 module clock enable bit: 1: Module clock on; 0: Module clock off.	0
12	SPI1EN	RW	SPI1 module clock enable bit: 1: Module clock on; 0: Module clock off.	0
11	TIM1EN	RW	TIM1 module clock enable bit: 1: Module clock on; 0: Module clock off.	0

10	ADC2EN	RW	ADC2 module clock enable bit: 1: Module clock on; 0: Module clock off.	0
9	ADC1EN	RW	ADC1 module clock enable bit: 1: Module clock on; 0: Module clock off.	0
8	Reserved	RO is reserved.		0
7	IOPFEN	RW	PF port module clock enable bit for IO: 1: Module clock on; 0: Module clock off.	0
6	IOPEEN	RW	IO PE port module clock enable bit: 1: Module clock on; 0: Module clock off.	0
5	IOPDEN	RW	PD port module clock enable bit for IO: 1: Module clock on; 0: Module clock off.	0
4	IOPCEN	RW	PC port module clock enable bit for IO: 1: Module clock on; 0: Module clock off.	0
3	IOPBEN	RW	PB port module clock enable bit for IO: 1: Module clock on; 0: Module clock off.	0
2	IOPAEN	RW	IO PA port module clock enable bit: 1: Module clock on; 0: Module clock off.	0
1	HSADCEN	RW	HSADC module clock enable bit: 1: Module clock on; 0: Module clock off.	0
0	AFIOEN	RW	IO Auxiliary Function Module Clock Enable Bit: 1: Module clock on; 0: Module clock off.	0

3.4.9 HB1 Peripheral Clock Enable Register (RCC_HB1PCENR) Offset

Address: 0x20

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
SWPMI EN	I2C3 EN	DAC EN	PWR EN	BKP EN	CAN2 EN	CAN1 EN	CAN3 EN	Reser ved	I2C2 EN	I2C1 EN	USART5 EN	USART4 EN	USART3 EN	USART2 EN	SPI4 EN

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3 EN	SPI2 EN	QSPI2 EN	QSPI1 EN	WWDG EN	LPTIM 2EN	LPTIM 1EN	USART 8EN	USART 7EN	USART 6EN	TIM7 EN	TIM6 EN	TIM5 EN	TIM4 EN	TIM3 EN	TIM2 EN

Bit	Name access		describe	Reset value
31	SWPMIEN	RW	SWPMI module clock enable bit: 1: Module clock on; 0: Module clock off.	0
30	I2C3EN	RW	I2C3 module clock enable bit: 1: Module clock on; 0: Module clock off.	0
29	DACEN	RW	DAC module clock enable bit: 1: Module clock on; 0: Module clock off.	0
28	PWREN	RW	Power interface module clock enable bit: 1: Module clock on; 0: Module clock off.	0
27	BKPEN	RW	Backup unit clock enable bit: 1: Module clock on; 0: Module clock off.	0
26	CAN2EN	RW	CAN2 module clock enable bit:	0

			1: Module clock on; 0: Module clock off.	
25	CAN1EN	RW	CAN1 module clock enable bit: 1: Module clock on; 0: Module clock off.	0
twenty four	CAN3EN	RW	CAN3 module clock enable bit: 1: Module clock on; 0: Module clock off.	0
twenty three	Reserved	RO is reserved.		0
twenty two	I2C2EN	RW	I2C2 module clock enable bit: 1: Module clock on; 0: Module clock off.	0
twenty one	I2C1EN	RW	I2C1 module clock enable bit: 1: Module clock on; 0: Module clock off.	0
20	USART5EN	RW	USART5 module clock enable bit: 1: Module clock on; 0: Module clock off.	0
19	USART4EN	RW	USART4 module clock enable bit: 1: Module clock on; 0: Module clock off.	0
18	USART3EN	RW	USART3 module clock enable bit: 1: Module clock on; 0: Module clock off.	0
17	USART2EN	RW	USART2 module clock enable bit: 1: Module clock on; 0: Module clock off.	0
16	SPI4EN	RW	SPI4 module clock enable bit: 1: Module clock on; 0: Module clock off.	0
15	SPI3EN	RW	SPI3 module clock enable bit: 1: Module clock on; 0: Module clock off.	0
14	SPI2EN	RW	SPI2 module clock enable bit: 1: Module clock on; 0: Module clock off.	0
13	QSPI2EN	RW	QSPI2 module clock enable bit: 1: Module clock on; 0: Module clock off.	0
12	QSPI1EN	RW	QSPI1 module clock enable bit: 1: Module clock on; 0: Module clock off.	0
11	WWDGEN	RW	Window watchdog clock enable bit: 1: Module clock on; 0: Module clock off.	0
10	LPTIM2EN	RW	LPTIM2 module clock enable bit: 1: Module clock on; 0: Module clock off.	0
9	LPTIM1EN	RW	LPTIM1 module clock enable bit: 1: Module clock on; 0: Module clock off.	0
8	USART8EN	RW	USART8 module clock enable bit: 1: Module clock on; 0: Module clock off.	0
7	USART7EN	RW	USART7 module clock enable bit: 1: Module clock on; 0: Module clock off.	0
6	USART6EN	RW	USART6 module clock enable bit: 1: Module clock on; 0: Module clock off.	0
5	TIM7EN	RW	Timer 7 module clock enable bit: 1: Module clock on; 0: Module clock off.	0
4	TIM6EN	RW	Timer 6 module clock enable bit: 1: Module clock on; 0: Module clock off.	0

3	TIM5EN	RW	Timer 5 module clock enable bit: 1: Module clock on; 0: Module clock off.	0
2	TIM4EN	RW	Timer 4 module clock enable bit: 1: Module clock on; 0: Module clock off.	0
1	TIM3EN	RW	Timer 3 module clock enable bit: 1: Module clock on; 0: Module clock off.	0
0	TIM2EN	RW	Timer 2 module clock enable bit: 1: Module clock on; 0: Module clock off.	0

3.4.10 Backup Domain Control Register (RCC_BDCTLR) Offset

Address: 0x24

31	30 29 28	27 26 25 24 23 22 21	2019	18	17 16
Reserved					BDRST

15	14 13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTCCAL[6:0]						RTCEN	RTCSEL [1:0]		ASOS	ASOE	CCO	LSEBYP	LSEREDY	LSEON

Bit	Name access		describe	Reset value
[31:17] Reserved		RO reserved.		0
16	BDRST	RW	Backup domain software reset control: 1: Reset the entire backup domain. 0: Cancel reset.	0
[15:9] RTCCAL[6:0]		RW	RTC calibration value register; the value of this register represents the value every 2^{20} that time How many clock pulses are skipped. This function is used to calibrate the RTC Clocks. RTC clocks can be slowed down by 0 to 121 ppm.	0
8	RTCEN	RW	RTC clock enable control: 1: Enable RTC clock; 0: Turn off the RTC clock.	0
[7:6] RTCSEL[1:0]		RW	RTC clock source selection: 00: No clock; 01: The LSE oscillator is used as the RTC clock; 10: LSI oscillator as RTC clock; 11: The HSE oscillator is divided by 512 to serve as the RTC clock.	0
5	ASOS	RW	PC13 pin alarm/second pulse output selection: 1: Output a second pulse; 0: Output alarm clock pulse. Note: This bit will only be reset by the backup domain reset (BDRST).	0
4	ASOE	RW	PC13 pin enable pulse output bit: 1: Enable output of alarm clock pulse or second pulse; 0: Disable the output of alarm clock pulse or second pulse. Note: This bit will only be reset by Backup Domain Reset (BDRST).	0
3	CCO	RW	Calibration clock output selection bit: 1: The PC13 pin outputs an RTC clock divided by 64;	0

			0: Do not output calibration clock. Note: This bit is cleared when the VDD33 power supply is disconnected.	
2	LSEBYP	RW	External low-speed crystal (LSE) bypass control bit: 1: Bypass external low-speed crystal/ceramic resonator (using external clock source); 0: No bypass for low-speed external crystal/ceramic resonators. Note: This field must be written when LSEON is 0.	0
1	LSERDY	RO	External low-speed crystal oscillation stable and ready flag (set by hardware): 1: External low-speed crystal oscillation is stable; 0: The external low-speed crystal oscillation is unstable. Note: After clearing LSEON in place, this bit needs to be cleared in one cycle: LSEON 0 6	0
0	LSEON	RW	External low-speed crystal oscillation enable control bit: 1: Enable the LSE oscillator; 0: Turn off the LSE oscillator.	0

Note: The LSEON, LSEBYP, CCO, ASOE, ASOS, RTCSEL, RTCEN and... in the backup domain control register (RCC_BDCTLR)

The RTCCAL bits are in the backup domain. Therefore, these bits are write-protected after reset, and can only be accessed in the power control register (PWR_CR). DBP

These bits can only be modified after their location is determined. These bits can be cleared by a backup domain reset. Any internal or external reset will not affect this.

Some positions.

3.4.11 Control/Status Register (RCC_RSTSCKR) Offset Address:

0x28

31	30 29 28	27 26 25 24 23 22 21	2019	18	17 16
LKUP RSTF	WWDG RSTF	IWDG RSTF	SFT RSTF	POR RSTF	PIN RSTF	Reser ved	RMVF	Reserved		

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

Bit	Name access		describe	Reset value
31	LKUPRSTF	RO	LOCKUP reset flag: 1: A LOCKUP event caused the system to reset; 0: Normal. When a LOCKUP reset occurs, the bit is set to 1 by hardware; it is cleared by writing the RMVF bit in software.	0
30	WWDGRSTF	RO	Window watchdog reset flag: 1: A window watchdog reset occurred; 0: No window watchdog reset occurred. When a window watchdog reset occurs, the hardware sets the bit to 1; the software writes the RMVF bit to clear it. remove.	0
29	IWDGRSTF	RO	Independent watchdog reset flag: 1: An independent watchdog reset occurred; 0: No independent watchdog reset occurred. When a standalone watchdog reset occurs, the bit is set to 1 by hardware; it is cleared by writing the RMVF bit in software. remove.	0
28	SFTRSTF	RO	Software reset flag: 1: A software reset occurred;	0

			0: No software reset occurred. When a software reset occurs, the hardware sets the bit to 1; the software writes the RMVF bit to clear it.	
27	PORRSTF	RO	Power-on/Power-off reset flag: 1: Power-on/power-off reset occurs; 0: No power-on/power-off reset occurred. During power-on/power-off reset, the bit is set to 1 by hardware; the RMVF bit is cleared by software writing. remove.	1
26	PINRSTF	RO	External manual reset (NRST pin) flag: 1: An NRST pin reset occurs; 0: No NRST pin reset occurred. The NRST pin is set to 1 by hardware when a reset occurs; the RMVF bit is cleared by software. remove.	0
25	Reserved	RO reserved.		0
twenty four	RMVF	W0	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 and ready flag (set by hardware): 1: Stable internal low-speed clock (40kHz); 0: The internal low-speed clock (40kHz) is not stable. Note: After clearing the bit, this bit needs to be cleared for one cycle. LSION 0 3 LSI	0
0	LSION	RW	Internal Low Speed Clock (LSI) Enable Control Bit: 1: Enable LSI (40kHz) oscillator; 0: Turn off the LSI (40kHz) oscillator.	0

BIT1

Note: Except for the power-on reset which clears the reset flag, other write-clear reset flags can be cleared.

3.4.12 HB Peripheral Reset Register (RCC_HBRSTR) Offset

Address: 0x2C

31	30 29 28	27 26 25 24 23 22 21	2019	18	17 16
Reserved			PIOC RST	Reser ved	SERDE SRST	USBPD RST	UHSIF RST	OTGFS RST	Reser ved

15	14 13	12	11	10	9	8	7 6 5 4 3 2	1	0
Reser ved	ETHMA CRST	DVP RST	USBSS RST	USBHS RST	SDMMC RST	RNG RST	FMC RST	Reserved	DMA2 RST	DMA1 RST

Bit	Name access		describe	Reset value
[31:22] Reserved		RO is reserved.		0
twenty two	PIOCRST	RW	PIOC module reset control: 1: Reset module; 0: No effect.	0
twenty one	Reserved	RO reserved.		0
20	SERDESRST	RW	SERDES module reset control: 1: Reset module; 0: No effect.	0
19	USBPDRST	RW	USBPD module reset control: 1: Reset module; 0: No effect.	0

18	UHSIFRST	RW	UHSIF module reset control: 1: Reset module; 0: No effect.	0
17	OTGFSRST	RW	USBOTG_FS module reset control: 1: Reset module; 0: No effect.	0
[16:15] Reserved		RO reserved.		0
14	ETHMACRST	RW	Ethernet MAC module reset control: 1: Reset module; 0: No effect.	0
13	DVPRST	RO	DVP module reset control: 1: Reset module; 0: No effect.	0
12	USBSSRST	RW	USBSS module reset control: 1: Reset module; 0: No effect.	0
11	USBHSRST	RW	USBHS module reset control: 1: Reset module; 0: No effect.	0
10	SDMMCRST	RW	SDMMC module reset control: 1: Reset module; 0: No effect.	0
9	RNGRST	RW	RNG module reset control: 1: Reset module; 0: No effect.	0
8	FMCRST	RW	FMC module reset control: 1: Reset module; 0: No effect.	0
[7:2] Reserved		RO reserved.		0
1	DMA2RST	RW	DMA2 module reset control: 1: Reset module; 0: No effect.	0
0	DMA1RST	RW	DMA1 module reset control: 1: Reset module; 0: No effect.	0

3.4.13 Clock Configuration Register 2 (RCC_CFGR2) Offset

Address: 0x30

31 30 29 28	27 26 25 24 23 22 21	2019	18	17 16
ETH1GSRC [1:0]	HSADCSRC [1:0]	Reserved	I2S3SRC	I2S2SRC	RNGSRC	Reserved	USBFSSRC	USBFSDIV[3:0]

15 14 13	12 11 10	9 8	7 6 5 4 3 2 1 0
LTDCSRC [1:0]	LTDCDIV[5:0]	UHSIFSRC [1:0]	UHSIFDIV[5:0]

Bit	Name access			Reset value
[31:30] ETH1GSRC[1:0]		RW	Description of Gigabit Ethernet 125M clock source selection: 00: PLL clock (PLL_CLK); 01: USBSS_PLL clock; 10: Output of ETH_PLL clock after being divided by 4; 11: SERDES_PLL is the output after the clock is divided by 8.	0
[29:28] HSADCSRC[1:0]		RW	HSADC clock source selection: 00: System clock (SYSCLK); 01: PLL clock (PLL_CLK);	0

			10: USBHS_PLL clock (480MHz); 11: ETH_PLL clock (500MHz).	
[27:26] Reserved		RO	is reserved.	0
25	I2S3SRC	RW	I2S3 clock source: 1: PLL clock (PLL_CLK); 0: System clock (SYSCLK).	0
twenty four	I2S2SRC	RW	I2S2 clock source: 1: PLL clock (PLL_CLK); 0: System clock (SYSCLK).	0
twenty three	RNGSRC	RW	RNG clock source selection: 1: PLL clock (PLL_CLK); 0: System clock (SYSCLK).	0
[22:21] Reserved		RO	is reserved.	
20	USBFSSRC	RW	USBFS 48M clock source selection: 1: USBHS_PLL clock; 0: PLL clock.	0
[19:16] USBFSDIV[3:0]		RW	USBFS 48M clock source prescaler: 0000: No frequency division; 0001: Frequency division by 2; 0010: 3-way divider; 0011: 4-way divider; 0100: 5-way divider; 0101: 6-way divider; 0110: 8-way divider; 0111: 10-way divider; 1000: 1.5 divider; 1001: 2.5 divider; 1010: 3.5 divider; 1011: 4.5 divider; 1100: 5.5 divider; 1101: 6.5 divider; 1110: 7.5 divider; 1111: 9.5 divider.	0
[15:14] LTDCSRC[1:0]		RW	LTDC clock source selection: 00: PLL clock (PLL_CLK); 01: SERDES_PLL clock divided by 2; 10: ETH_PLL clock; 11: USBHS_PLL clock.	0
[13:8] LTDCDIV[5:0]		RW	LTDC clock source prescaler control: 000000: No frequency division; 000001: Frequency division by 2; 000010: 3 frequency divider; 000011: 4 frequency dividers; 000100: 5-fold frequency division; ... 111110:63 frequency division; 111111:64 frequency division.	0
[7:6] UHSIFSRC[1:0]		RW	UHSIF clock source selection: 00: System clock (SYSCLK); 01: PLL clock (PLL_CLK); 10: USBHS_PLL clock (480MHz); 11: ETH_PLL clock (500MHz).	0
[5:0] UHSIFDIV[5:0]		RW	UHSIF clock source prescaler control:	0

			000000: No frequency division; 000001: Frequency division by 2; 000010: 3 frequency divider; 000011: 4 frequency dividers; 000100: 5-fold frequency division; ... 111110:63 frequency division; 111111:64 frequency division.	

3.4.14 PLL Configuration Register 2 (RCC_PLLCFGR2) Offset

Address: 0x34

31	30 29 28	27 26 25 24 23 22	twenty one	2019	18	17 16
Reserved					SERDESPLL_MUL[3:0]	

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

Bit	name	access	describe	Reset value
[31:20] Reserved		RO is reserved.		0
[19:16]	SERDESPLL_MUL[3:0]	RW	Writes are only allowed when the , Only when SERDES_PLLON clock multiplier is 0. 0000: 25x multiplier output; 0001: 28x multiplier output; 0010: 30x frequency multiplier output; 0011: 32x frequency multiplier output; 0100: 35x frequency multiplier output; 0101: 38x frequency multiplier output; 0110: 40x frequency multiplier output; 0111: 45x frequency multiplier output; 1000: 50x multiplier output; 1001: 56x multiplier output; 1010: 60x multiplier output; 1011: 64x multiplier output; 1100: 70x multiplier output; 1101: 76x multiplier output; 1110: 80x frequency multiplier output; 1111: 90x frequency multiplier output.	1000b
[15:13] Reserved		RO is reserved.		0
[12:8]	USBHSPLL_IN_DIV[4:0]	RW	The frequency division setting for the USBHS_PLL input clock source SYS_PLL. Writes are only allowed when USBHS_PLLON is 0: 00000: No frequency division; 00001: Frequency division by 2; ... 11111:32 frequency division.	0
7	Reserved	RO is reserved.		0
[6:4]	USBSSPLL_REFSEL[1:0]	RW	USBSS_PLL Reference Clock Frequency Selection, Only when Writes are only allowed when USBSS_PLLON is 0: 000: 20MHz; 001: 24MHz; 010: 25MHz; 011: 30MHz;	010b

			100:32MHz; 101: 40MHz; 110: 60MHz; 111: 80MHz.	
[3:2]	USBHSPLL_REFSEL[1:0]	RW	USBHS_PLL Reference Clock Frequency Selection, Only when Write operations are only allowed when USBHS_PLLON is 0: 00: 25MHz; 01: 20MHz; 10:24MHz; 11:32MHz.	0
[1:0]	USBHSPLLSRC[1:0]	RW	USBHS_PLL clock source selection: 00: HSE; 01: HSI; 10: ETHCLK_20M; 11: The frequency division coefficients of SYS_PLL_CLK after frequency division are referenced. RCC_PLLCFGR2[12:8].	0

Chapter 4 Central Processing Unit (CPU)

The CH32H417 series integrates a dual-core architecture, with both RISC-V5F and RISC-V3F dual cores supporting 32-bit I, C, M, A, B, F, and... Extended instruction set.

The RISC-V3F core (C0/core 0) internally employs a sequential single-fetch architecture. The instruction fetch mode is configurable, supporting both pipelining and non-pipelined modes. Threaded mode reduces core power consumption in non-pipelined mode.

The RISC-V5F core (C1/core 1) employs an out-of-order dual-issue architecture and integrates multiple branch predictors, supporting instructions... Caching enables V5 to achieve superior performance. Furthermore, RISC-V5 implements the WCH-X extended instruction set, resulting in improved code density and computational performance. All of them have improved.

The proprietary peripheral includes system timers, inter-core communication modules, and hardware semaphore modules; it also supports external debuggers, which support single/dual-line connections. Mode communication, single-line communication speed is adjustable in four levels.

4.1 Main Features

RISC-V3F

Sequential single-shot architecture

ÿ 3-stage production line

Low clock speed, low power consumption, with a focus on control.

Provides a non-maskable interrupt NMI . 4-channel

physical memory protection (PMP).

ÿ 4-channel hardware breakpoint

ÿ Level 2 interrupt nesting

ÿ Fast Programmable Interrupt Controller (PFIC)

Custom extension instructions

Hardware stack

RISC-V5F

Out -of-order multi-launch architecture

Built -in 32KB instruction cache

High clock speed, high performance, focusing on computation.

Provides a non-maskable interrupt NMI . 4-channel

physical memory protection (PMP).

ÿ 4-channel hardware breakpoint

ÿ 8-level interrupt nesting

ÿ Fast Programmable Interrupt Controller (PFIC)

Custom extension instructions

Hardware stack

4.1.1 V5F Kernel Memory Operation Model

The V5F kernel is an out-of-order superscalar processor that reorders memory accesses.

V5F memory types are divided into two main categories:

ÿ Normal

CPU access to this type of memory is reordered to improve performance.

ÿ Device and Strongly-ordered

CPU access to this type of memory will be executed strictly in program order.

Table 4-1 lists the default attributes and types of memory regions. A Memory Protection Unit (PMP) can handle read (R), write (W), and execute operations. The attribute of (X) is turned off.

Table 4-1 Default Attributes and Types of Memory Regions

	CODE	ITCM	DTCM	SRAM	PERI	MEMO	MEM1	DEV0	DEV1	PPB	RSV
genus sex	RWX	RWX	RWX	RWX	RW	RWX	RWX	RW	RW	RW	RW
total Wire	HB	TCM_BUS	TCM_BUS	HB	HB	HB	HB	HB	HB	HB	HB
kind type	Normal	Normal	Normal	Normal	Device	Device	Device	Normal	Device	Device	Device

Note: (1) TCM_BUS refers to the custom bus of this chip that is adapted to the TCM memory.

(2) PPB area access has a fixed priority, with the V5 kernel having a higher priority than the V3 kernel. Attention should be paid to the impact of bandwidth limitations on performance.

The address ranges of each memory region are shown in Table 4-2:

Table 4-2 Address Ranges of Memory Regions

memory area	Address range
CODE	0x00000000 - 0x1FFFFFFF
ITCM	0x200A0000 - 0x200BFFFF
DTCM	0x200C0000 - 0x200FFFFF
SRAM	0x20100000 - 0x3FFFFFFF
PERI	0x40000000 - 0x5FFFFFFF
MEM0	0x60000000 - 0x7FFFFFFF
MEM1	0x80000000 - 0x9FFFFFFF
DEV0	0xA0000000 - 0xBFFFFFFF
DEV1	0xC0000000 - 0xDFFFFFFF
PPB	0xE0000000 - 0xE00FFFFF
RSV	Other addresses

In an out-of-order kernel, the actual memory access order is not always consistent with the memory access order in the program. To improve access efficiency, the kernel... The kernel will employ methods such as reordering, buffering, and multiple bus interfaces to speed up memory access without violating data dependencies.

For two unrelated memory access instructions A1 and A2 in the program, assume that instruction A1 is executed before instruction A2, and there is no direct instruction between them. The command/data synchronization instruction determines the actual request order of A1/A2, which depends on the memory type (Normal, Device) accessed by A1/A2 and its... Instruction type (Load, Store). Possible scenarios are shown in Table 4-3.

Table 4-3 Memory types and instruction types accessed by A1/A2

A1 \ A2	store device	store normal		load device	Load normal
store device	order	re-order		order	re-order
store normal	re-order	re-order		re-order	re-order
load device	order	order		order	re-order
Load normal	order	order		re-order	re-order

Note: The order and reordering mentioned above are all based on... CPU The top-level interface bus operation sequence is used as the standard. For all out-of-order accesses, insertion can be used.

fence Command order is forced.

4.2 Privilege Register (CSR)

Both the RISC-V3F and RISC-V5F cores conform to the RISC-V standard privileged architecture, supporting M and U modes, debug mode, etc. The kernel defines some standard registers and user-defined registers.

4.2.1 General Privilege Register

The RISC-V architecture defines several control and status registers (CSRs) for configuration.

It can be used to identify or record the running status. The CSR register is an internal kernel register that uses a dedicated 12-bit address space.

Note: Registers with the attributes "MRW, MRO, MRW1" can only be accessed in machine mode.

Table 4-4 General Privilege Register List

name	CSR Address Description 0xF11 Chip	Reset value
MVENDORID	Manufacturer ID Register	0x00000000
MARCHID	0xF12 Chip Structure ID Register	V3F:0xDC68D86D V5F:0xDC68D8AE
MIMPID	0xF13 Kernel ID Register	V3F:0xDC688002 V5F:0xDC688001
MHARTID	0xF14 Kernel ID Register	V3F:0x00000000 V5F:0x00000001
MSTATUS	0x300 Machine Status Register; 0x301	0x00000000
MISA	Machine Instruction Set Register; 0x305	0x40901027
MTVEC	Machine Mode Exception Base Address Register; 0x340	0x00000000
MSCRATCH	Machine Mode Temporary Register; 0x341	X
MEPC	Machine Exception Program Counter Register; 0x342	X
MCAUSE	Machine Exception Register; 0x343	0xX00000XX
MTVAL	Machine Trap Value Register; 0x7A4 Trigger	X
TINFO	Information Register; 0x7B1 Debug PC	0x00000004
DPC	Address Register; 0x7B2 Debug Mode	X
DSCRATCH0	Temporary Register 0; 0x7B3 Debug Mode	X
DSCRATCH1	Temporary Register 1; 0x7C0 Debug Register 0;	X
DBGMCU_0	0x001 Floating-Point Accumulated	0x00000000
FFLAGS	Exception Flag Register; 0x002 Floating-Point	0x00000000
FRM	Rounding Mode Register; 0x003 Floating-Point	0x00000000
FCSR	Control and Status Register; 0x800 User Access	0x00000000
UACCES_MSTATUS	Machine Status Register; 0x804 Hardware Stack	0x00000000
HW_POPDM_CTLR	Control Register; 0x7A0 Trigger Channel	0x00000004
TSELECT	Selection Register; 0x7A1 Trace Trigger Data	0x0000000X
TDATA1	Register; 0x7A2 Save Trigger Data Register	0x23E00048
TDATA2		X

4.2.1.1 Chip Manufacturer ID Register (MVENDORID)

CSR Address: 0xF11

Bit	Name	Access description: MRO chip	Reset value
[31:0]	mvendorid [31:0]	manufacturer ID, fixed value is 0.	0

4.2.1.2 Chip Structure ID Register (MARCHID) CSR

Address: 0xF12

Bit	name	access	describe	Reset value
[31:0]	marchid[31:0]	MRO	chip structure ID, a fixed value that cannot be modified.	V3F: 0xDC68D86D V5F: 0xDC68D8AE

4.2.1.3 Machine-implemented ID Register

(MIMPID) CSR Address: 0xF13

Bit	name	access	describe	Reset value
[31:0]	mimpid[31:0]	The MRO machine implements an ID, which is a fixed value and cannot be modified.		V3F: 0xDC688002 V5F: 0xDC688001

4.2.1.4 Kernel ID Register (MHARTID) CSR

Address: 0xF14

Bit	name	access	describe	Reset value
[31:0]	mhartid[31:0]	MRO	Kernel ID indicates the kernel's position within the chip architecture. Nuclear ID number.	V3F: 0 V5F: 1

4.2.1.5 Machine Status Register (MSTATUS)

CSR Address: 0x300

Bit	name	access	describe	Reset value
31	SD	MRO	is used to mark whether a floating-point instruction is in a dirty state.	0
[30:15]	Reserved	MRO	reserved.	0
[14:13]	FS[1:0]	MRW	Floating-point instruction set status; a non-zero value allows single precision. Floating-point instruction (F) execution: 00: OFF; 01: Initial; 10: Clean; 11: Dirty.	0
[12:11]	MPP[1:0]	MRW	Save/restore the machine state before the error occurred. In case of an anomaly, update to the machine state before the anomaly: 00: When an exit error occurs, the machine state is set to U mode; 01: Reserved; 10: Retained; 11: When an exit exception occurs, the machine state is set to M mode.	0
[10:8]	Reserved	MRO	reserved.	0
7	MPIE	MRW	Saves the interrupt enable state when entering an interrupt: 1: Enable global interrupts after exiting the interrupt; 0: Disable global interrupts after exiting the interrupt.	0
[6:4]	Reserved	MRO	retained.	0
3	MIE	MRW	Global interrupt enabled, updated to MPIE upon interrupt exit. value:	0

			1: Global interrupt enable, allowing the processor to respond to interrupts. ask; 0: Global interrupts are disabled, disallowing processor responses. Disconnect the request.	
[2:0]	Reserved	MRO retained.		0

4.2.1.6 Machine Instruction Set Register (MISA) CSR

Address: 0x301

Bit	name	access		Reset value
[31:30] XLEN[1:0]		MRO	Describe the machine's basic integer bit width and basic instruction bit width. Information, 32-bit wide kernel, fixed value of 1.	0x1
[29:24] Reserved		MRO retained.		0
twenty three	X	MRO This kernel supports the execution of custom instruction sets.		1
[22:21] Reserved		MRO retained.		0
20	U	MRO supports user mode operation.		1
[19:13] Reserved		MRO retained.		0
12	M	MRO This kernel supports the execution of the multiplication instruction set.		1
[11:9] Reserved		MRO retained.		0
8	I	MRO This kernel supports the execution of basic integer instruction sets.		1
[7:6]	Reserved	MRO retained.		0
5	F	MRO This kernel supports the execution of single-precision floating-point instruction sets.		1
[4:3]	Reserved	MRO retained.		0
2	C	MRO This kernel supports the execution of compression instruction sets.		1
1	B	MRO This kernel supports the execution of bit manipulation instruction sets.		1
0	A	MRO This kernel supports the execution of atomic instruction sets.		1

4.2.1.7 Machine Mode Exception Base Address Register (MTVEC)

CSR Address: 0x305

Bit	name	access	describe	Reset value
[31:10] mtvec_base[21:0]		MRW	The high-order bits of the base address to jump to when a processor exception occurs. 1kB alignment, actual 32-bit base address BASE = {mtvec_base,10'h0}.	0
[9:2]	Reserved	MRO reserved.		0
1	mode1	RW	Vector table pattern 1: 1: The interrupt vector table stores jump addresses; 0: The interrupt vector table stores jump instructions.	0
0	mode0	RW	Vector table mode 0: 1: All interrupts/exceptions have independent entry points and jump addresses. The ID is BASE + 4 * IRQ / EXC. 0: All interrupts/exceptions have a unified entry point, which jumps to... Abnormal jump base address.	0

4.2.1.8 Machine Mode Temporary Register (MSCRATCH)

CSR Address: 0x340

Bit		access	describe	Reset value
Name[31:0] mscratch[31:0]		MRW is a machine-mode temporary register.		x

4.2.1.9 Machine Exception Program Counter Register (MEPC)

CSR Address: 0x341

Bit	name	access	Describe	Reset value
[31:0] mepc[31:0]		MRW	the PC address that is saved and restored when exiting an interrupt/exception. mepc[0] is fixed to 0.	x

4.2.1.10 Machine Exception Register (MCAUSE) CSR

Address: 0x342

Bit	name	access		Reset value
31	Interrupt	RW	Description of interrupt/exception flags: 1: The exception/interrupt was generated externally; 0: The exception/interrupt is generated by the kernel.	x
[30:8] Reserved		RO is reserved.		0
[7:0]	Exception Code[7:0]	RW stores the source information that caused the exception or interruption.		x

Interruption exception source code:

Interrupt/Exception Exception Code		Interrupt/Exception Source
1	2	Retain NMI
1	3	machine mode software interrupt
1	4	User-mode timer interrupt
1	5	Monitor mode timer interrupt
1	6	reserve
1	7	Machine mode timer interrupt
1	8	User-mode external interrupt
1	11	Machine Mode External Interrupt
1	12	System timer interrupt
1	ÿ16	Reserved platform interrupt
0	0	ID; instruction address not aligned.
0	1	Instruction access error
0	2	illegal command
0	3	breakpoint
0	4	Reading unaligned addresses
0	5	Read access error
0	6	Write/atomic operation address unaligned
0	7	Write/atomic operation access error
0	8	User-mode ecall,
0	11	Machine-mode ecall,
0	ÿ16	Reserved

4.2.1.11 Machine Trap Value Register (MTVAL)

CSR Address: 0x343

Bit		access	describe	Reset value
Name[31:0] mtval[31:0]		RW	stores additional information about the trap.	X

Additional information stored in the `mtval` register within the `trap`:

Exception types	Mcause	Save information
Instruction address	0	Instruction PC
unaligned, instruction	1	(Instruction
access failed,	2	PC) - Instruction Machine Code
illegal instruction.	4	Read data address
Read address	5	Read data address
unaligned, read access	6	Write data address
failed, write/atomic access failed, write/atomic address unaligned.	7	Write data address

4.2.1.12 Trigger Information Register (TINFO)

CSR Address: 0x7A4

Bit	name	access	describe	Reset value
[31:0] tinfo[31:0]		MRO	The trigger information register corresponds to the TDATA1 register. The type value, where type is 2, is the second bit of tinfo. Set to 1.	0x4

4.2.1.13 Debug PC Address Register (DPC) CSR Address:

Bit 0x7B1

		access	describe	Reset value
Name[31:0] dpc[31:0]		RW	saves the PC address restored when exiting debug mode.	X

The addresses where dpc is saved for jumps to debug mode caused by different debug request sources are shown in the table below:

Triggers	Save address
ebreak	The instruction address of Ebreak.
single step	Save the address of the next instruction to be executed in non-debug mode (V5F only).
Trigger module tdata1.timing=0: The address of the instruction that triggers the breakpoint (V5F only).	
The address of the next instruction to be executed when the request is paused and entered debug mode.	

4.2.1.14 Debug Mode Temporary Register 0 (DSCRATCH0) CSR Address:

0x7B2

Bit	name	access	describe	Reset value
[31:0] Dscratch0[31:0]		RW	Debug Mode Temporary Register 0.	X

4.2.1.15 Debug Mode Temporary Register 1 (DSCRATCH1) CSR Address:

0x7B3

Bit	name	access	describe	Reset value
[31:0] Dscratch1[31:0]		RW	debug mode temporary register 1.	X

4.2.1.16 Debug Register 0 (DBGMCU_0)

CSR Address: 0x7C0

Bit	name	access	The	Reset value
[31:8]	dbg_mode_stop	MRW	registered module stops working when entering debug mode: 1: Registered peripherals stop when the kernel enters debug mode. Stop working; 0: Registered peripherals are active when the kernel enters debug mode. Regular work.	0
[7:0]	Reserved	MRO	retained.	0

4.2.1.17 Floating-point Accumulated Exception Flag Register (FFLAGS)

CSR Address: 0x001

Bit	name	access	describe	Reset value
[31:5]	Reserved	RO	is reserved.	0
[4:0]	FFlags[4:0]	RW	Cumulative anomaly flags are used to indicate anomalies since the last reset. To (reset or software clear) in floating-point instructions An anomaly occurred.	0

name bit		description
NV	4	illegal operation
DZ	3	Divisor is zero
OF	2	Overflowing upwards
UF	1	Downflow
NX	0	Inaccurate

4.2.1.18 Floating-Point Rounding Mode Register (FRM)

CSR Address: 0x002

Bit	name	access	describe	Reset value
[31:3]	Reserved	RO	is reserved.	0
[2:0]	Frm[2:0]	RW	Floating-point rounding mode register: 000: Round to the nearest value (round to even values when the value is in the middle). (rounding in the direction of the number). 001: Round to zero; 010: Round down; 011: Round up; 100: Round to the nearest value (when it is the middle value, round to the nearest value). (large-scale rounding) 101, 110: Retained; 111: Dynamic rounding, based on the FRM bit configuration in the instruction. Select the rounding mode for the value.	0

4.2.1.19 Floating-Point Control and Status Register (FCSR)

CSR Address: 0x003

Bit	name	access	describe	Reset value

[31:8] Reserved		RO is reserved.	0
[7:5]	Frm[2:0]	RW is the floating-point rounding mode register.	0
[4:0]	FFlags[4:0]	RW cumulative anomaly marker.	0

4.2.1.20 User access to the machine status register (UACCES_MSTATUS)

CSR address: 0x800

bits	name	access	describe	Reset value
31	SD	RO is used to mark whether a floating-point instruction is in a dirty state.		0
[30:15] Reserved		RO is reserved.		0
[14:13] FS[1:0]		RO	Floating-point instruction set status; single precision is allowed when the value is not 0. Floating-point instruction (F) execution: 00: OFF; 01: Initial; 10: Clean; 11: Dirty.	0
[12:11] MPP[1:0]		RO	Save/restore the machine state before the error occurred. In case of an anomaly, update to the machine state before the anomaly: 00: When an exit error occurs, the machine state is set to U mode; 01: Reserved; 10: Retained; 11: When an exit exception occurs, the machine state is set to M mode.	0
[10:8] Reserved		RO is reserved.		0
7	MPIE	RW	Saves the interrupt enable state when entering an interrupt: 1: Enable global interrupts after exiting the interrupt; 0: Disable global interrupts after exiting the interrupt.	0
[6:4]	Reserved	RO is reserved.		0
3	MIE	RW	Global interrupts are enabled; when exiting an interrupt, it is updated to MPIE. value: 1: Global interrupt enable, allowing the processor to respond to interrupts. ask; 0: Global interrupts are disabled, disallowing processor responses. Disconnect the request.	0
[2:0]	Reserved	RO is reserved.		0

Note: This register is used by the kernel to configure the MSTATUS register when running in user mode. The CPU_RUN_CTLR register bit When 1 is usta
is enabled, this address becomes available, and its register settings are the same as those of the MSTATUS (0x300) register.

4.2.1.20 Hardware Push Control Register (HW_POPDM_CTLR) CSR

Address: 0x804

Bit	name	access		Reset value
31	lock	RW	Describes the user mode lock flag; once locked, user mode is not... The configuration of this group of registers can be rewritten: 1: Locked, configurable only in machine mode; 0: Unlocked, configurable machine/user mode.	0
[30:6] Reserved		RO is reserved.		0

5	hw_pop_off	RW	One-time hardware pop-off shutdown, automatically after exiting the interrupt. Reset: 1: Disable hardware stack popping the next time the interrupt is exited; 0: Hardware popping is not disabled.	0
4	Reserved	RO is reserved.		0
[3:2]	preempt[1:0]	RW	Preemption priority bit-width configuration register, used to configure Interrupt priority preemption priority bit width: 00: Preemption bit width is 0, no interrupt of any priority. Nested laws; 01: Preemption bit width is 1, i.e., interrupt priority register Bit [7] represents the preemption priority; 10: Preemption bit width is 2, i.e., interrupt priority register Bits [7:6] represent preemption priority; 11: Preemption bit width is 3, i.e., interrupt priority register. The [7:5] bits are the preemption priority.	0x1
1	nest_en	RW	Interrupt nesting enable register: 1: Allow nested interrupts 0: Nested interruptions are not allowed.	0
0	hw_stk_en	RW	Hardware stack protection enabled: 1. Hardware stack protection is used when entering or exiting interrupts; 0: Hardware stack protection is not used when entering or exiting an interrupt.	0

4.2.1.22 Trigger Channel Selection Register (TSELECT)

CSR Address: 0x7A0

Bit	name	access	describe	Reset value
[31:2] Reserved		MRO retained.		0
[1:0]	mscratch[1:0]	MRW trigger channel selection register.		x

Note: (1) Kernel 1 All implement four-channel triggers, configured by selecting the trigger channel through the TSELECT register, and triggered by TSELECT.

The least significant bit is active, and the remaining bits are 0.

(2) The kernel's TSELECT register is only applicable to chips whose fifth bit of the batch number is not equal.

4.2.1.23 Trace Trigger Data Register (TDATA1) CSR Address:

0x7A1

Bit	Name	access	describe	Reset value
[31:28] Type [3:0]		MRO trigger type definition.		0x2
27	dmode	MRO	Both machine mode and debug mode allow modification of the trigger phase. Turn off the	0
[26:21] maskmax		MRO	register. When macth=1, the maximum allowed exponentiation for matching is... scope.	0x1F
[20:19] Reserved		MRO retained.		0
18	timing	MRO is triggered before the execution of instructions/operations.		0
[17:13] Reserved		MRO retained.		0
12	action	MRW sets the processing mode to be used when triggered:		0

			1: Upon triggering, enter debug mode, save and exit dpc. The address of the instruction that needs to be restored during debugging; 0: When triggered, a process is generated that directs the thread to M mode. The anomaly.	
[11:8] Reserved		MRO retained.		0
7	match	MRO	Matching strategy configuration: Allows comparisons of addresses whose values are equal to the value of tdata2. hair.	0
6	m	MRO	Trigger enable in M mode: 1: Enable; 0: Off.	1
[5:4]	Reserved	MRO retained.		0
3	u	MRO	Trigger enable in U mode: 1: Enable; 0: Off.	1
2	execute	MRW	Triggered based on the virtual address of the executed instruction. 1: Address of the trigger comparison instruction; 0: The trigger does not compare instruction addresses.	0
1	store	MRW	Triggered based on the virtual address of the stored data 1: Trigger compares and writes data addresses; 0: The trigger does not compare the write data address.	0
0	load	MRW	Triggered based on the virtual address of the read data 1: Trigger compares the read data address; 0: The trigger does not compare the read data address.	0

Note: (1) The processor trigger configuration conforms to the RISC-V debugging standard, and the trigger configuration type is 2, based on the lowest virtual address accessed.

Bit matching is performed on the TDATA1 register only when the fifth bit of the batch number is 0. The chip.

4.2.1.24 Save the trigger data register (TDATA2) CSR address:

0x7A2

Bit	name	access	describe	Reset value
[31:0] tdata2[31:0]		MRW	Used to save trigger data (command PC or data location) site).	x

Note: The kernel TDATA2 register is only applicable to chips where the fifth digit of the batch

4.2.2 RISC-V3F Kernel Custom Registers Table 4-5 RISC-

V3F Kernel Custom Register List

name	CSR Address Description: 0xBC0 Processor	Reset value
CPU_RUN_CTLR	Execution Control Register; 0xBC1 Interrupt	0x12370000
INEST_CTLR	Nesting Control Register; 0xBC8 Machine	0x00000000
MIE	Interrupt Register; 0x7B0 Debug Control	0x00000000
DCSR	and Status Register	0x40000000

4.2.2.1 Processor Run Control Register (CPU_RUN_CTLR)

CSR Address: 0xBC0

Bit	name	access	This describes	Reset value
[31:28]	fadd_clkdiv[3:0]	MRW	the frequency divider for floating-point addition instructions. The default value is 1, meaning that the frequency divider for floating-point addition instructions is 1. Let the frequency be 2 times the main frequency.	0x1
[27:24]	fmul_clkdiv[3:0]	MRW	The frequency divider for floating-point multiplication instructions, with a default value of 2, i.e., the frequency divider for floating-point multiplication instructions. Let the frequency be 3 times the main frequency.	0x2
[23:20]	fmac_clkdiv[3:0]	MRW	The frequency divider for floating-point multiplication and addition instructions has a default value of 3, which is the frequency divider for floating-point multiplication. The instruction frequency is divided by 4 of the main frequency.	0x3
[19:16]	fdiv_clkdiv[3:0]	MRW	The floating-point division instruction prescaler has a default value of 4, meaning that the floating-point division instruction prescaler is... Let the frequency be 8 times the main frequency.	0x7
7	nmi_ie	MRW	Non-maskable interrupt enable: 1: When an exception causes a nested overflow, an NMI interrupt is entered; 0: When an exception causes a nested overflow, the NMI interrupt is not triggered.	0
6	int_fence	MRW	Fence interrupt enable: 1: Clear interrupt requests when executing the fence instruction; 0: Do not clear interrupt requests when executing the fence instruction.	0
5	non_usta	MRW	Enable the UACCES_MSTATUS register: 1: Bits 3 and 7 of CSR address 0x800 are mapped to... Bit MIE of the MSTATUS register and bit MSTATUS register MPIE; 0: CSR address 0x800 is a read-only register, and the return value is... The value of MSTATUS.	0
[4:2]	Reserved	MRO	reserved.	0
[1:0]	pipe_acc[1:0]	MRW	Instruction fetch mode: 00: Prefetching disabled. Instruction prefetching is disabled to avoid invalid commands. In an instruction fetch operation, at most one valid instruction exists on the CPU pipeline. This mode consumes the least power, but performance drops by about 2 to 3 times. 01: Prefetch Enabled. Instruction prefetching is enabled, and the CPU will continuously... Access to the instruction memory continues until the internal instruction buffer is ready. The number of instructions to be executed exceeds a certain limit, or the instruction buffer is full. When the CPU is full, instruction fetching is paused. This mode has high power consumption but high performance. Prediction failure will introduce redundant instruction fetch operations, and in some cases, The execution unit will introduce an additional 0 to 2 cycles of bubbles, most of which (The performance degradation of the sub-process is not significant). Other: Reserved.	0

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

Floating-point arithmetic instructions are multi-cycle instructions. The maximum operating frequency for different types of floating-point instructions is shown in the table above. (Depending on the processor...)

To determine the operating frequency, users need to configure the floating-point frequency division factor so that the divided frequency does not exceed the maximum operating frequency shown in the table above.

fxxx freq(max) ! **kernel clock speed /** fxxx clkdiv

4.2.2.2 Interrupt Nesting Control Register (INEST_CTLR)

CSR Address: 0xBC1

Bit	name	access	describe	Reset value
31	Reserved	RO is reserved.		0
30	nest_ovr	MRW	Interrupt/exception nesting overflow flag; when the nesting depth exceeds the maximum... When the depth is large, it is set to 1; writing 1 clears it to zero. Note: Interrupt overflow will only cause instruction execution during an interrupt. At this time, the exception of the NMI interrupt service routine. The program entered normally, but a stack overflow occurred, so it cannot be terminated due to this exception. and NMI interrupt exit.	0
[29:12] Reserved		MRO reserved.		0
[11:8] nest_sta[3:0]		MRO	Nested status flags: 0000: No interruption; 0001: Level 1 interrupt; 0011: Second-level interrupt nesting; 0111: Three-level interrupt nesting, overflow; Other: Reserved.	0
[7:2] Reserved		MRO reserved.		0
[1:0] nest_max[1:0]		MRW	Maximum nesting level: 00: Nesting is disabled (nesting functionality is turned off); 01: Second-level nesting (Enable nesting functionality); 10: Unable to write; 11: Unable to write. Note: (1) Writing 11b to this field sets the register. could be 01b; (2) Write 11b At that time, reading this register will give you the highest nesting level of the chip.	0

4.2.2.3 Machine Interrupt Register (MIE)

CSR Address: 0xBC8

Bit	name	access	describe	Reset value
[31:5] Reserved		RO is reserved. The		0
[4:0] nest_mie[4:0]		RW	machine interrupt register stores each level of interrupts in a multi-level interrupt nesting system. Mie information: status.mie = nest_mie[0]; status.mpie=nest_mie[1].	0

4.2.2.4 Debug Control and Status Register (DCSR)

CSR Address: 0x7B0

Bit		access	describe	Reset value
Name[31:28] exdebug[3:0]		MRO has an external debugger that conforms to the debugger protocol.		0x4
[27:16] Reserved		MRO reserved. In		0
15	ebreakm	MRW	machine mode, the ebreak command enters debug mode. Enable:	0

			1: Run the ebreak instruction in machine mode to process... The device enters debug mode; 0: The ebreak instruction is executed in machine mode to process... The device entered an abnormal state.	
[14:13] Reserved		MRO retained.		0
12	ebreaku	MRW	The `ebreak` command in user mode enters debug mode. Enable: 1: Run the ebreak instruction in user mode to process... The device enters debug mode; 0: The ebreak instruction is executed in user mode to process... The device entered an abnormal state.	0
[11:10] Reserved		MRO reserved.		0
9	stoptime	MRW	Timer pause enabled: 1: The counter stops counting in debug mode; 0: The counter works normally in debug mode.	0
[8:6]	dcause[2:0]	MRW	The debug source register is used to mark the entry point for debugging. Specific events of the pattern 000: Reset value; 001: The `ebreak` instruction in the program triggers debugging (preferably) Priority 3); 010: The trigger module caused a breakpoint exception (preferred). Level 4 (V5F only); 011: The debugger generates a pause request (priority 1); 100: Pause in single-step mode (priority 0, lowest) (V5F only); 101: Not achieved; 110: Not achieved; 111: Reserved.	0
[5:4]	Reserved	MRO retained.		0
3	nmip	MRO kernel NMI interrupt suspension flag.		0
2	Reserved	MRO retained.		0
[1:0]	prv[1:0]	MRW	Record the machine state before entering debug mode: 00: Enter debug mode from user mode and restore. The processor reverts to user mode; 01, 10: Reserved; 11: Enter debug mode from machine mode and restore. The processor reverts to machine mode.	0

4.2.3 RISC-V5F Kernel Custom Registers Table 4-6 RISC-

V5F Kernel Custom Register List

name	CSR Address Description: 0x320 Counter Mask		Reset value
MCOUNT_INHIBIT	Register; 0x7B0 Debug Control and Status		0x00000000
DCSR	Register; 0xB00 Machine Cycle Counter Register		0x40000000
MCYCLE			0x00000000

MINSTRET	0xB02 Machine Instruction Counter Register; 0xC00		0x00000000
UCYCLE	User Mode Cycle Counter Register; 0xC02 User Mode		0x00000000
UINSTRET	Instruction Counter Register; 0xBC0 Processor Execution		0x00000000
CPU_RUN_CTLR	Control Register; 0xBC1 Interrupt Nesting Control		0x12370300
INEST_CTLR	Register; 0xBC2 Cache Policy Control Register;		0x00000000
CACHE_STRTG_CTLR	0xBC3 Cache Policy PMP Overwrite Register;		0x0F000003
CACHE_PMP_OVR	0xBC4 Hardware Stack Address Register; 0xBC5		0x00000000
HW_POPDM_ADDR	Memory Configuration Register		0x20060000
MEMARY_CFGR			0x600F0FFF
TCM_RRDUTY_CFGR	0xBC6	TCM Priority Access Duration Configuration Register	0x00000000
MIE	0xBC8 Machine Interrupt Register; 0xBD0		0x00000000
OPCACHE_CTLR	Cache Operation Register; 0xFC0 Memory		0x00000000
MEMINFO	Information Register		0x12220030

4.2.3.1 Counter Mask Register (MCOUNT_INHIBIT) CSR Address: 0x320

Bit	name	access	describe	Reset value
[31:3] Reserved		MRO reserved.		0
2	IR	MRW	Instruction counter mask register: 1: Disable instruction counters; stop minstret and unstret. count; 0: Instruction counters are not masked; minstret and unstret are positive. Constant count.	0
1	Reserved	MRO reserved.		0
0	CY	MRW	Cycle counter mask register: 1: Disable command counters, mcycle and ucycle stop counters. number; 0: The cycle counter is not disabled; mcycle and ucycle are normal. count.	0

4.2.3.2 Debug Control and Status Register (DCSR) CSR

Address: 0x7B0

Bit		access	describe	Reset value
Name[31:28] exdebug[3:0]		MRO has an external debugger that conforms to the debugger protocol.		0x4
[27:16] Reserved		MRO reserved. In		0
15	ebreakm	MRW	machine mode, the ebreak command enters debug mode. Enable: 1: Run the ebreak instruction in machine mode to process... The device enters debug mode; 0: The ebreak instruction is executed in machine mode to process... The device entered an abnormal state.	0
[14:13] Reserved		MRO reserved.		0
12	ebreaku	MRW	The `ebreak` command in user mode enters debug mode. Enable:	0

			1: Run the ebreak instruction in user mode to process... The device enters debug mode; 0: The ebreak instruction is executed in user mode to process... The device entered an abnormal state.	
11	stepie	MRW	Single-step mode interrupt enable: 1: Allow interrupt responses in single-step mode; 0: Disable interrupt response in single-step mode.	0
10	Reserved	MRO reserved.		0
9	stoptime	MRW	Timer pause enabled: 1: The counter stops counting in debug mode; 0: The counter works normally in debug mode.	0
[8:6]	dcause[2:0]	MRW	The debug source register is used to mark the entry point for debugging. Specific events of the pattern 000: Reset value; 001: The `ebreak` instruction in the program triggers debugging (preferably) Priority 3); 010: The trigger module caused a breakpoint exception (preferred). Level 4 (V5F only); 011: The debugger generates a pause request (priority 1); 100: Pause in single-step mode (priority 0, lowest) (V5F only); 101: Not achieved; 110: Not achieved; 111: Reserved.	0
[5:4]	Reserved	MRO retained.		0
3	nmip	MRO kernel NMI interrupt suspension flag.		0
2	step	MRW	Single-step mode: 1: Single-step mode; 0: Non-single-step mode.	0
[1:0]	prv[1:0]	MRW	Record the machine state before entering debug mode: 00: Enter debug mode from user mode and restore. The processor reverts to user mode; 01, 10: Reserved; 11: Enter debug mode from machine mode and restore. The processor reverts to machine mode.	0

4.2.3.3 Machine Cycle Counter Register (MCYCLE) CSR Address:

0xB00 Name

Bit		access	describe	Reset value
[31:0] mcyle	[31:0]	MRW	A machine cycle counter is used to count the clock cycles of a machine. During this period, the CY bit of the MCOUNT_INHIBIT register is 0 when counting begins. In debug mode, the counter stops counting when it stops.	0

4.2.3.4 Machine Instruction Counter Register (MINSTRET) CSR

Address: 0xB02

Bit	name	access	describe	Reset value
[31:0]	minstret[31:0]	MRW	An instruction counter is used to count the number of instructions that the machine has correctly executed. When the IR bit of the MCOUNT_INHIBIT register is 0, the timer is active. The counter stops counting when it stops in debug mode.	0

4.2.3.5 User Mode Cycle Counter Register (UCYCLE) CSR Address: 0xC00

Bit Name

		access		Reset value
[31:0]	ucyle[31:0]	RO	Describes a user-mode cycle counter used to count the time the machine is working. When the clock cycle is complete and the CY bit of the MCOUNT_INHIBIT register is 0. The counter stops counting when it stops in debug mode.	0

Note: User-mode cycle counter, with the same function as the mcycle register, is used to read the number of machine cycles in user mode.

4.2.3.6 User Mode Instruction Counter Register (UINSTRET) CSR Address:

0xC02

Bit	name	access	describe	Reset value
[31:0]	uinstret[31:0]	RO	An instruction counter is used to count the number of instructions that the machine has correctly executed. When the IR bit of the MCOUNT_INHIBIT register is 0, the timer is active. The counter stops counting when it stops in debug mode.	0

Note: User-mode instruction counter, with the same function as the minstret register, is used to read the number of instructions correctly executed by the machine in user mode.

4.2.3.7 Processor Run Control Register (CPU_RUN_CTLR) CSR Address:

0xBC0

Bit	name	access	Describes	Reset value
[31:28]	fadd_clkdiv[3:0]	MRW	the floating-point instruction prescaler, with a default value of 1, i.e., floating-point instruction prescaler. The frequency of the dot-add instruction is divided by 2 of the main frequency.	0x1
[27:24]	fmul_clkdiv[3:0]	MRW	The floating-point multiplication factor is the instruction prescaler, with a default value of 2, i.e., floating-point multiplication. The frequency of the dot product instruction is 3 times the main frequency.	0x2
[23:20]	fmac_clkdiv[3:0]	MRW	The frequency divider for floating-point multiply-accumulate instructions, with a default value of 3. The frequency of the floating-point multiply-accumulate instruction is 4 times the main frequency.	0x3
[19:16]	fdiv_clkdiv[3:0]	MRW	The floating-point divider instruction frequency prescaler has a default value of 4, meaning it is a floating-point divider. The frequency of the division instruction is 8 times the main frequency.	0x7
15	nlp_en	MRW	Next-line branch prediction enabled: 1: Enable; 0: Off.	0
14	ghr_en	MRW	Global history register function enabled: 1: Enable the global history register for branches; 0: Branch global history register is disabled.	0
[13:11]	Reserved	MRO	retained.	0
10	lsu_dual	MRW	Parallel issuance of memory access instructions is enabled; setting it to 1 allows issuance. The emitter unit issues memory access instructions in parallel, improving memory access efficiency. Set the launch speed.	0
[9:8]	Reserved	MRO	retained.	0
7	nmi_ie	MRW	Non-Maskable Interrupt Enable:	0

			1: When an exception causes a nested overflow, it enters the NMI. Break; 0: When an exception causes a nested overflow, NMI is not entered. Interrupted.	
6	int_fence	MRW	Fence interrupt enable: 1: Clear interrupt requests when executing the fence instruction; 0: Do not clear interrupt requests when executing the fence instruction.	0
5	non_usta	MRW	Enable the UACCES_MSTATUS register: 1: Bits 3 and 7 of CSR address 0x800 respectively map to The MIE and MSTATUS bits of the MSTATUS register are mapped to the MSTATUS register. The register's MPIE bit; 0: CSR address 0x800 is a read-only register, return. The value is the value of MSTATUS.	0
[4:0]	Reserved	MRO retained.		0

Kernel fadd_freq(max) fmul_freq(max)			fmac_freq(max)	fdiv_freq(max)
V5F	128MHz	96MHz	76MHz	38MHz

Floating-point arithmetic instructions are multi-cycle instructions. The maximum operating frequency for different types of floating-point instructions is shown in the table above. (Depending on the processor...)

To determine the operating frequency, users need to configure the floating-point frequency division factor so that the divided frequency does not exceed the maximum operating frequency shown in the table above.

fxxx freq(max) ! **kernel clock speed /** fxxx clkdiv

4.2.3.8 Interrupt Nesting Control Register (INEST_CTLR) CSR Address:

0xBC1

Bit	name	access		Reset value
31	lsu_nmi_sta	MRW	Describes the memory access unit generating a non-maskable interrupt flag (usually a write interrupt). (Error returned by line), write 1 to reset.	0
30	nest_ovr_sta	MRW	Interrupt/exception nesting overflow flag, when the nesting depth exceeds... It is set to 1 at maximum depth, and cleared to zero when written to 1. Note: Interrupt overflow will only occur when the maximum number of interrupts allowed by the kernel is executed. When the series service function generates an instruction exception and an NMI interrupt, The exception and NMI interrupt proceed normally at this point, but the CPU... Stack overflow. Do not exit from this exception or NMI interrupt.	0
[29:16] Reserved		MRO reserved.		0
[15:8] nest_sta		MRO	Nested status flags: [0x00]: No interruption; [0x01]: Level 1 interrupt; [0x03]: Level 2 interrupt nesting; [0x07]: Three-level interrupt nesting; [0x0F]: Level 4 interrupt nesting; [0x1F]: Level 5 interrupt nesting; [0x3F]: Level 6 interrupt nesting; [0x7F]: Seventh-level interrupt nesting; [0xFF]: Eight-level interrupt nesting;	0

			Other: Reserved.	
[7:3] Reserved		MRO reserved.		0
[2:0] nest_max		MRW	Maximum nesting level: 000: Nesting is prohibited (nesting functionality is disabled); 001: Allows two levels of nesting; 010: Allows three levels of nesting; 011: Allows four levels of nesting; 100: Allows five levels of nesting; 101: Allows six levels of nesting; 110: Allows seven levels of nesting; 111: Allows eight levels of nesting;	0

4.2.3.9 Cache Policy Control Register (CACHE_STRTG_CTLR) CSR Address:

0xBC2

Bitname		access	describe	Reset value
[31:28] Reserved		RO is reserved.		0
27	ic_mem1_strtg	MRW	Instruction cache for the 0x80000000-0x9fffffff region is... able: 1: Allow caching of instructions within this region; 0: Caching of instructions within this region is prohibited.	1
26	ic_mem0_strtg	MRW	Instruction cache for the 0x60000000-0x7fffffff region Yes.	1
25	ic_sram_strtg	MRW	The instruction cache for the 0x20000000-0x3fffffff region is enabled. Yes.	1
twenty four	ic_code_strtg	MRW	The instruction cache for the 0x00000000-0x1fffffff region is enabled. able.	1
[23:2] Reserved		RO is reserved.		0
1	ic_disable	The MRW instruction cache disable flag is set to 0 to enable instruction cache functionality.		
0	Reserved	1 indicates RO is reserved.		1

4.2.3.10 Caching Policy PMP Overwrite Register (CACHE_PMP_OVR)

CSR Address: 0xBC3

Bit Name [31:13] Reserved		access	describe	Reset value
ic_pmp3cache_strtg		MRO retained.		0
12	Instruction cache policy register for MRW PMP channel 3 address area. 0 [11:9] Reserved ic_pmp2cache_strtg Instruction			
cache policy register for		MRO retained.		0
8	MRW PMP channel 2 address area. 0 [7:5] Reserved ic_pmp1cache_strtg Instruction cache policy register for MRW PMP			
channel 1 address area.		MRO retained.		0
4	0 [3:1] Reserved			
		MRO retained.		0
0	ic_pmp0cache_strtg MRW		Instruction cache policy register in the address region of PMP channel 0: 1: When the instruction address is protected by PMP channel 0, the current instruction... Make it cacheable;	0

			0: When the instruction address is protected by PMP channel 0, the current instruction... Make it uncacheable.	

Note: Caching strategy PMP Overwrite register, when the instruction or data address is different from the received register. PMP

When the address for channel control matches, the policy in the cache policy control register (0xBC2) is not executed; instead, the policy in this register is executed. PMP Channel caching strategy

4.2.3.11 Hardware Push Address Register (HW_POPDM_ADDR) CSR Address:

0xBC4

	name	Access the fixed starting address of the MRO DTCM	Reset value
Position [31:20]		region.	0x200
[19:9]	hw_stk_vector	MRW DTCM region block address, 512-bit aligned.	0x600
[8:0]		MRO is fixed at 0.	0

Note: The hardware stack address register points DTCM Within the region, when the hardware push function is enabled and an interrupt/exception occurs, the stack value is quickly pushed onto the machine.

to the specified area, usually pointing to the bottom of the area. DTCM DTCM

4.2.3.12 Memory Configuration Register (MEMARY_CFGR)

CSR Address: 0xBC5

Bit	name	access		Reset value
[31:20]	MEM0_addr_hi[11:0]	MRW	Describes the branch predictor address matching register, configured for external memory. The high 12 bits of the address information are used when the high bits of the external memory address do not match. At that time, the prediction function of some branches is limited. Note: Do not configure it as a high address of internal memory (0x000, 0x200, 0x201).	0x600
[19:18]	dtcm_rr_mode[1:0]	MRW	DTCM memory access priority mode: 00: Fixed priority, other requests (DMA, C0_lsu, C0_ifu) Requests above C1; 01: Fixed priority, C1 request takes precedence over other requests (DMA). C0_lsu, C0_ifu); 10: Polling priority, a single channel can only continuously receive polling data for a maximum period of time. Priority is granted; after a timeout, the system switches to another channel. Priority is then granted when the system gains priority. Priority is given when one channel has no operational request while another channel does. Switch to another channel and re-time; see priority duration configuration. The TCM_RRDUTY_CFGR register. 11: Forced polling priority, one channel is forced to obtain priority for a certain period of time. Operation priority, switching to another channel after timeout, priority duration configuration See the TCM_RRDUTY_CFGR register.	0x3
[17:16]	itcm_rr_mode[1:0]	MRW	ITCM memory access priority mode: 00: Fixed priority, other requests (DMA, C0_lsu, C0_ifu) Requests above C1 01: Fixed priority, C1 request takes precedence over other requests (DMA). (C0_lsu, C0_ifu) 10: Polling priority, a single channel can only continuously receive polling data for a maximum period of time. Priority is granted; after a timeout, the system switches to another channel. Priority is then granted when the system gains priority. Priority is given when one channel has no operational request while another channel does.	0x3

			Switch to another channel and re-time; see priority duration configuration. The TCM_RRDUTY_CFGR register. 11: Forced polling priority, one channel is forced to obtain priority for a certain period of time. Operation priority, switching to another channel after timeout, priority duration configuration See the TCM_RRDUTY_CFGR register.	
[15:12] Reserved		RO is reserved.		0
[11:10] C0_ifu_dtcm_mode		MRW	C0_IFU configures access permissions for DTCM memory: C0_ifu_dtcm_mode[1]ÿ 1: Reading enables; 0: Reading is prohibited. C0_ifu_dtcm_mode[0]ÿ 1: Write the enable function; 0: Writing is prohibited.	0x3
[9:8] C0_ifu_itcm_mode		MRW	C0_IFU configures access permissions for the ITCM memory: C0_ifu_itcm_mode[1]ÿ 1: Reading enables; 0: Reading is prohibited. C0_ifu_itcm_mode[0]ÿ 1: Write the enable function; 0: Writing is prohibited.	0x3
[7:6] C0_lsu_dtcm_mode		MRW	C0_LSU configures access permissions for DTCM memory: C0_lsu_dtcm_mode[1]ÿ 1: Reading enables; 0: Reading is prohibited. C0_lsu_dtcm_mode[0]ÿ 1: Write the enable function; 0: Writing is prohibited.	0x3
[5:4] C0_lsu_itcm_mode		MRW	C0_LSU configures access permissions for the ITCM memory: C0_lsu_itcm_mode[1]ÿ 1: Reading enables; 0: Reading is prohibited. C0_lsu_itcm_mode[0]ÿ 1: Write the enable function; 0: Writing is prohibited.	0x3
[3:2] DMA_dtcm_mode		MRW	DMA access permission configuration for DTCM memory: DMA_dtcm_mode[1]: 1: Reading enables; 0: Reading is prohibited. DMA_dtcm_mode[0]: 1: Write the enable function; 0: Writing is prohibited.	0x3
[1:0] DMA_itcm_mode		MRW	DMA access permission configuration for ITCM memory: DMA_itcm_mode[1]: 1: Reading enables; 0: Reading is prohibited.	0x3

			DMA_itcm_mode[0]: 1: Write the enable function; 0: Writing is prohibited.	

4.2.3.13 TCM Preferred Access Duration Configuration Register (TCM_RRDUTY_CFGR) CSR

Address: 0xBC6 Bit

Name		access		Reset value
[31:24]	C1_dtcm_rr_cycle [7:0]	MRW	Description of a high-priority continuous clock cycle request from C1 to dtcm.	0
[23:16]	Cx_dtcm_rr_cycle [7:0]	MRW	Configuration. High-priority requests for DTCM from non-C1 (C0, DMA) requests. Persistent clock cycle	0
[15:8]	C1_itcm_rr_cycle [7:0]	MRW	configuration. A high-priority persistent clock cycle is requested from C1 for itcm.	0
[7:0]	Cx_itcm_rr_cycle [7:0]	MRW	Configuration. Prioritize ITC requests from non-C1 (C0, DMA) sources. The continuous clock cycle configuration.	0

Note: TCM area access priority duration configuration, when TCM_CFGR is configured as 10b itcm_rr_mode and 11b dtcm_rr_mode are used to control the duration of the high priority of the corresponding channel.

It takes effect

4.2.3.14 Machine Interrupt Register (MIE) CSR

Address: 0xBC8

Bitname		access	describe	Reset value
[31:9] Reserved		RO is reserved. The		0
[8:0] nest_mie		RW	machine interrupt register stores the interrupts from each level in a multi-level interrupt nesting system. Broken MIE information: MSTATUS register mie bit = nest_mie[0]; The mpie bit of the MSTATUS register is nest_mie[1].	0

4.2.3.15 Cache Operation Register (OPCACHE_CTLR)

CSR Address: 0xBD0

Bit	name	access	describe	Reset value
[31:5] Vaddr[27:0]		WO	Operation address or index information.	0
[4:3] Reserved		RO is reserved.		0
2	Idx_mode	WO	Index pattern. 1: Perform the operation using the vaddr value as the address information; 0: Execute the operation starting at the index information in the vaddr value. do.	0
[1:0] Opcode[1:0]		WO	00: Icache invalidate; Other: Invalid.	0

4.2.3.16 Memory Information Register (MEMINFO) CSR

Address: 0xFC0 Bit

Name		access	describe	Reset value
[31:30] Reserved		RO is reserved.		0
[29:26] DTCM datasize[3:0]		MRO	data memory capacity information:	0100b

			64KB*2DTCM datasize=256KB	
[25:24] DTCM LineSize[1:0]		MRO	Data memory row length information: 00: 8 bytes; 01: 16 bytes; 10:32 bytes; 11:64 bytes.	10b
[23:22] Reserved		RO is reserved.		0
[21:18] ITCM datasize[3:0]		MRO	Instruction memory capacity information: 64KB * 2ITCM datasize = 128KB.	0010b
[17:16] ITCM LineSize[1:0]		MRO	Instruction memory line length information: 00: 8 bytes; 01: 16 bytes; 10:32 bytes; 11:64 bytes.	10b
[15:7] Reserved		RO is reserved.		0
[6:5] Icache_way[1:0]		MRO	Instruction cache path count: 00: 1-way; 01: 2-way; 10: 4-way; 11:8-way.	01b
[4:2]	Icache_datasize [2:0]	MRO	Instruction cache capacity information: 000: NONE; 001: 4KB; 010: 8KB; 011: 16KB; 100:32KB; 101: 64KB; 110: 128KB; 111: 256KB.	100b
[1:0]	Icache linesize [1:0]	MRO	Instruction cache line length information: 00: 8 bytes; 01: 16 bytes; 10: 32 bytes; 11: 64 bytes.	0

4.3 Physical Memory Protection Unit (PMP)

To enhance system security, the RISC-V architecture defines a set of physical address access restrictions, which can be set for physical memory within a given region. Its read, write, and execute attributes are protected by a minimum area length of 4 bytes. The PMP unit remains active in user mode and can be disabled in machine mode. If the current memory limit is violated, a system exception (EXC) will occur.

The PMP unit of core 1 (RISC-V5F) contains four 8-bit configuration registers (CSR address 0x3A0) and four address registers. The device (CSR address is 0x3B0ÿ0x3B3) needs to be accessed using the csr instruction and must be done in machine mode.

The PMP unit of kernel 0 (RISC-V3F) contains four 8-bit configuration registers (CSR address 0x3A0) and four address registers. The device (CSR address is 0x3B0ÿ0x3B3) needs to be accessed using the csr instruction and must be done in machine mode.

0 PMP 0

Note: The kernel functionality is only applicable to chips whose fifth digit of the batch number is not a specific number.

4.3.1 PMP Register

Table 4-7 PMP Register List

name	CSR Address	describe	Reset value
PMPCFG0	0x3A0	PMP Configuration Register	0x00000000
PMPADDR0	0x3B0	PMP Address 0 Register	0x00000000
PMPADDR1	0x3B1	PMP Address 1 Register	0x00000000
PMPADDR2	0x3B2	PMP Address 2 Register	0x00000000
PMPADDR3	0x3B3	PMP Address 3 Register	0x00000000

4.3.1.1 PMP Configuration Register (PMPCFG0)

CSR Address: 0x3A0

<table>
<tr><th>Bit</th><th>Name Access</th><th>describe</th><th>Reset value</th></tr>
<tr><td>[31:24] pmp3cfg[7:0]</td><td>[23:16]</td><td>See pmp0cfg for MRW.</td><td>0</td></tr>
<tr><td>pmp2cfg[7:0]</td><td>[15:8] pmp1cfg[7:0]</td><td>See pmp0cfg for MRW.</td><td>0</td></tr>
<tr><td></td><td></td><td>See pmp0cfg for MRW.</td><td>0</td></tr>
<tr><td>[7:0] pmp0cfg[7:0]</td><td></td><td>MRW</td><td>
<table>
<tr><th>Bit</th><th>name</th><th>describe</th></tr>
<tr><td>7</td><td>L</td><td>Lock enabled; can be unlocked in machine mode.
0: Not locked;
1: Lock the relevant registers.</td></tr>
<tr><td>6</td><td colspan="2">- reserve.</td></tr>
<tr><td>5</td><td>IC_Str</td><td>When an address matches, the instruction caching strategy is executed.
Policy value and CACHE_PMP_OVR register correspond.</td></tr>
<tr><td colspan="3">[4:3] A. Address alignment and protection zone range selection.</td></tr>
<tr><td>2</td><td colspan="2">X is an executable attribute.</td></tr>
<tr><td>1</td><td colspan="2">W is a writable property.</td></tr>
<tr><td>0</td><td colspan="2">R can read attributes.</td></tr>
</table>
</td><td>0</td></tr>
</table>

Specifically, address alignment and protection region selection involve memory protection for regions where A_ADDR ÿ region < B_ADDR (requirements). Both A_ADDR and B_ADDR are 4-byte aligned.

1. If B_ADDR – A_ADDR == 22, then use the NA4 method;
2. If B_ADDR – A_ADDR == 2 (G+2), Gÿ1, and A_ADDR is 2 (G+2) aligned, then the NAPOT method is used;
3. Otherwise, use the TOR method.

A Value	Name	describe
00b	OFF	No area to protect
01b	TOR	Top alignment area protection: region = BUS_ADDR >> 2; Under pmp0cfg, 0 ÿ region < pmpaddr0; Under pmp1cfg, pmpaddr0 ÿ region < pmpaddr1; Under pmp2cfg, pmpaddr1 ÿ region < pmpaddr2; In pmp3cfg, pmpaddr2 ÿ region < pmpaddr3.

		pmpaddri-1 = A_ADDR >> 2; pmpaddri = B_ADDR >> 2.
10b NA4		Fixed 4-byte area protection. pmp0cfgÿpmp3cfg correspond to pmpaddr0ÿpmpaddr3 as the starting address. pmpaddri = A_ADDR >> 2.
11b NAPOT		Protect 2 (G+2) regions, Gÿ1, at which point A_ADDR is 2 (G+2) aligned. pmpaddri = ((A_ADDR >> 2) | (2 (G-1) -1).

4.3.1.2 PMP Address 0 Register (PMPADDR0) CSR Address:

0x3B0 Name

Bit		The MRW PMP access description sets bits	Reset value
[31:0] ADDR0[31:0]		[33:2] at address 0, but the high 2 bits are actually unused. 0	

4.3.1.3 PMP Address 1 Register (PMPADDR1) CSR Address:

0x3B1

Bit	name	Access description reset value	
[31:0] ADDR1[31:0]		MRW PMP sets bits [33:2] at address 1, but the high 2 bits are actually unused. 0	

4.3.1.4 PMP Address 2 Register (PMPADDR2) CSR Address:

0x3B2 Name

Bit		The access description states that MRW PMP sets	Reset value
[31:0] ADDR2[31:0]		bits [33:2] at address 2, but the high 2 bits are actually unused. 0	

4.3.1.5 PMP Address 3 Register (PMPADDR3) CSR Address:

0x3B3

Bit	name	Access description reset value	
[31:0] ADDR3[31:0]		MRW PMP sets bits [33:2] at address 3, but the high 2 bits are actually unused. 0	

4.4 Inter-core communication (IPC)

The inter-core communication function provides four communication channels, each with 8 status bits. The status bits are enabled independently, and there are two interrupt modes. (32 bits * 4)

Communication data register.

Two channel interruption methods: receiver interruption and transmitter interruption.

Each channel has 8 status bits, each status bit is enabled independently, and the enable is automatically updated.

ÿ Receive and transmit interrupts are enabled independently, and the interrupt kernel is independently configurable.

ÿ Four sets of inter-core information registers.

4.4.1 IPC Function Description

This module provides two interrupt types:

ÿ Receiver interruption

Transmitter interruption

The IPC channel interrupt is enabled by the interrupt enable bit (RX_IERx/TX_IERx), the status enable bit (IPC_ENAx[7:0]), and the status bit.

(IPC_STSx[7:0]) Joint control.

When the

receive interrupt status bit is 1, the corresponding status enable bit is 1, and the corresponding bit in the IPC_ISR register is 1; if the channel receive interrupt is enabled...

When valid (RX_IERx = 1), an interrupt request can be generated to the RX_CID kernel configured for the channel, and the corresponding bit in the IPC_ISM register is set to 1.

When a transmit

interrupt occurs, if the status bit is 0, the corresponding status enable bit is 1, and the corresponding bit in the IPC_ISR register is 1; if the channel transmit interrupt is enabled...

When valid (TX_IERx = 1), an interrupt request can be generated to the TX_CID kernel configured for the channel, and the corresponding bit in the IPC_ISM register is set to 1.

When the AUTOENx bit is configured for automatic update, the IPC_ENA register is updated synchronously when the sending end writes to or sets the IPC_STS register.

The values are the same; if the channel receive interrupt enable is active, a receive interrupt can be immediately generated at the receiver. The message is cleared after the receive interrupt response is completed.

The IPC_STS flag does not affect the IPC_ENA register. If the channel transmit interrupt is valid, a transmit interrupt can be generated immediately at the transmitting end.

The sending end disables IPC_ENA enable and clears the sending end's interrupt flag during the interrupt.

When the AUTOENx bit is configured to not update automatically, the order in which the sender writes the status enable bit first and then writes IPC_STS may introduce problems.

Additional sender interruption.

4.4.2 IPC Register

Table 4-8 List of IPC-Related Registers

name	Access address	describe	Reset value
R32_IPC_CTLR	0xE000D000	IPC Control Register	0x00000000
R32_IPC_ISR	0xE000D004	IPC Interrupt Status Register	0x00000000
R32_IPC_ISM	0xE000D008	IPC ISM Channel Interrupt Mask Register	0x00000000
R32_IPC_ENA	0xE000D010	IPC status bit interrupt enable register	0x00000000
R32_IPC_STS	0xE000D014	IPC Channel Status Register	0xXXXXXXXX
R32_IPC_SET	0xE000D018	IPC Status Flag Position Register	0x00000000
R32_IPC_CLR	0xE000D01C	IPC status flag reset register	0x00000000
R32_IPC_MSG0	0xE000D020	IPC Information Register 0	0xXXXXXXXX
R32_IPC_MSG1	0xE000D024	IPC Information Register 1	0xXXXXXXXX
R32_IPC_MSG2	0xE000D028	IPC Information Register 2	0xXXXXXXXX
R32_IPC_MSG3	0xE000D02C	IPC Information Register 3	0xXXXXXXXX

4.4.2.1 IPC Control Register (IPC_CTLR) Offset Address:

0x00 Bit Name

		access		Reset value
31	LOCK3	RW	Channel 3 is configured to lock; writing a 1 will only clear the value by resetting. The other control registers of the rear channel 3 cannot be modified. 1: Channel 3 control lock; 0: Channel 3 is unlocked.	0
30	AUTOEN3	RW	Channel 3 status bit enables automatic update; when set to 1, it is active at the sending end. When writing IPC_STS, IPC_ENA is automatically updated to the same value. When the sender writes IPC_SET and sets IPC_STS, the IPC_ENA phase is... The position should be in place. 1: The IPC_ENA register is automatically updated; 0: Automatic update of the IPC_ENA register is disabled.	0

29	RX_IER3	RW	Channel 3 receiver interrupt enable: when set to 1, channel 3 can generate... Live reception interrupt: 1: Channel 3 receiver interrupt enabled; 0: Channel 3 receiver interrupt shielding.	0
28	TX_IER3	RW	Channel 3 transmitter interrupt enable: when set to 1, channel 3 can generate... Data transmission interrupted: 1: Channel 3 transmitter interrupt enabled; 0: Channel 3 transmitter interruption mask.	0
27	Reserved	RO is reserved.		0
26	RX_CID3	RW	Receiver kernel ID, configured to respond to receive interrupts: 1: Generate a receive interrupt request to kernel 1; 0: Generate a receive interrupt request to kernel 0.	0
25	Reserved	RO is reserved. The		0
twenty four	TX_CID3	RW	sending kernel ID, which configures the kernel to respond to transmission interrupts. 1: Send an interrupt request to kernel 1; 0: Send an interrupt request to kernel 0.	0
twenty three	LOCK2	RW Channel 2 configuration locked.		0
twenty two	AUTOEN2	RW Channel 2 status bit is enabled for automatic update.		0
twenty one	RX_IER2	RW	Channel 2 receiver interrupt is enabled; setting it to 1 allows Channel 2 to generate... The live reception was interrupted.	0
20	TX_IER2	RW	Channel 2 transmitter interrupt enable: when set to 1, channel 2 can generate... The transmission was interrupted.	0
19	Reserved	RO is reserved.		0
18	RX_CID2	RW: Receiver kernel ID, configures the kernel that responds to receive interrupts. 0 RO:		
17	Reserved	Reserved.		0
16	TX_CID2	RW: Sender kernel ID, configures the kernel that responds to transmit interrupts. 0 RW:		
15	LOCK1	Channel 1, configures locking.		0
14	AUTOEN1	RW Channel 1 status bit is enabled for automatic update.		0
13	RX_IER1	RW	Channel 1 receiver interrupt is enabled; setting it to 1 allows Channel 1 to generate... The live reception was interrupted.	0
12	TX_IER1	RW	Channel 1 transmitter interrupt enable: when set to 1, Channel 1 can generate... The transmission was interrupted.	0
11	Reserved	RO is reserved.		0
10	RX_CID1	RW: Receiver kernel ID, configures the kernel that responds to receive interrupts. 0 RO:		
9	Reserved	Reserved.		0
8	TX_CID1	RW: Sender kernel ID, configures the kernel that responds to transmit interrupts. 0 RW:		
7	LOCK0	Channel 0, configures locking.		0
6	AUTOEN0	RW channel 0 status bit is enabled for automatic update.		0
5	RX_IER0	RW	Channel 0 receiver interrupt is enabled; when set to 1, channel 0 can generate... The live reception was interrupted.	0
4	TX_IER0	RW	Channel 0 transmitter interrupt enabled; when set to 1, channel 0 can generate... The transmission was interrupted.	0
3	Reserved	RO is reserved.		0
2	RX_CID0	RW is the receiver kernel ID, configured to respond to receive interrupts.		0

1	Reserved	RO is reserved.	0
0	TX_CID0	RW is the sender kernel ID, configuring the kernel that responds to send interrupts. 0	

4.4.2.2 IPC Interrupt Status Register (IPC_ISR) Offset Address:

0x04 Bit Name

		access	This	Reset value
[31:24] IPC_ISR3[7:0]		RO	describes the interrupt status register for IPC channel 3. A non-zero value indicates that the IPC... Channel 3 is interrupted. When IPC_ENA is enabled, the IPC_STS register value is 0. A transmitter interrupt is generated when the IPC_STS register value is 1. The receiver was interrupted.	0
[23:16] IPC_ISR2[7:0]		RO	IPC Channel 2 Interrupt Status Register.	0
[15:8] IPC_ISR1[7:0]		RO	IPC Channel 1 Interrupt Status Register.	0
[7:0] IPC_ISR0[7:0]		RO	IPC Channel 0 Interrupt Status Register.	0

4.4.2.3 IPC ISM Channel Interrupt Mask Register (IPC_ISM) Offset Address: 0x08

Name

Bit		access	This	Reset value
[31:24] IPC_ISM3[7:0]		RO	describes the IPC channel 3 interrupt mask register; a non-zero value indicates that IPC... Channel 3 generates an interrupt request. When RX_IER is enabled, IPC_STS is 1, and IPC_ENA is 1. A bit with a value of 1 in RX_CID generates a receiver interrupt. When TX_IER is enabled, IPC_STS is 0 and IPC_ENA is 0. A bit with a value of 1 in TX_CID generates a transmitter interrupt.	0
[23:16] IPC_ISM2[7:0]		RO	IPC Channel 2 Interrupt Mask Register.	0
[15:8] IPC_ISM1[7:0]		RO	IPC Channel 1 Interrupt Mask Register.	0
[7:0] IPC_ISM0[7:0]		RO	IPC Channel 0 Interrupt Mask Register.	0

4.4.2.4 IPC Status Bit Interrupt Enable Register (IPC_ENA) Offset Address:

0x10 Bit Name

		access	This	Reset value
[31:24] IPC_ENA3[7:0]		RW	describes the interrupt enable register for IPC channel 3; when its value is 1... Enable the interrupt function for the corresponding bit. The bit TX_CID with an IPC_STS value of 0 and an IPC_ENA value of 1. An interrupt is generated at the transmitting end. The bit RX_CID with IPC_STS value of 1 and IPC_ENA value of 1. An interrupt was generated at the receiving end.	0
[23:16] IPC_ENA2[7:0]		RW	IPC Channel 2 Flag Interrupt Enable Register.	0
[15:8] IPC_ENA1[7:0]		RW	IPC Channel 1 Flag Interrupt Enable Register.	0
[7:0] IPC_ENA0[7:0]		RW	IPC Channel 0 Flag Interrupt Enable Register.	0

4.4.2.5 IPC Channel Status Register (IPC_STS) Offset Address:

0x14 Bit Name

		access	describe	Reset value

[31:24] IPC_STS3[7:0]	RW	IPC Channel 3 Status Register: A bit with an IPC_STS value of 1 can generate a receive interrupt; A bit with an IPC_STS value of 0 can generate a transmit interrupt.	X
[23:16] IPC_STS2[7:0]	RW	IPC Channel 2 Status Register.	X
[15:8] IPC_STS1[7:0]	RW	IPC Channel 1 Status Register.	X
[7:0] IPC_STS0[7:0]	RW	IPC Channel 0 Status Register.	X

4.4.2.6 IPC Status Flag Set Register (IPC_SET) Offset Address: 0x18

Bit	name	access	describe	Reset value
[31:24] IPC_SET3[7:0]		WO	IPC Channel 3 Status Flag Set Register, write 1 and then set to 0. The corresponding bit in the IPC_STS register, if the channel's AUTOEN... If the bit is enabled, the corresponding bit in the IPC_ENA register is also set.	0
[23:16] IPC_SET2[7:0]		WO	IPC Channel 2 Status Flags Position Register.	0
[15:8] IPC_SET1[7:0]		WO	IPC Channel 1 Status Flags Position Register.	0
[7:0] IPC_SET0[7:0]		WO	IPC Channel 0 Status Flags Set Bit Register.	0

4.4.2.7 IPC Status Flag Reset Register (IPC_CLR) Offset Address: 0x1C

Bit	name	access	describe	Reset value
[31:24] IPC_CLR3[7:0]		WO	Clear the status flag register for IPC channel 3 by writing 1 and then resetting. The corresponding bit in the IPC_STS register.	0
[23:16] IPC_CLR2[7:0]		WO	Clear the status flag register for IPC channel 2 by writing 1 and then resetting. The corresponding bit in the IPC_STS register.	0
[15:8] IPC_CLR1[7:0]		WO	Clear the status flag register for IPC channel 1 by writing 1 and then resetting it. The corresponding bit in the IPC_STS register.	0
[7:0] IPC_CLR0[7:0]		WO	Clear the IPC channel 0 status flag register by writing 1, then reset. The corresponding bit in the IPC_STS register.	0

4.4.2.8 IPC Information Register (IPC_MSGx) (x=0/1/2/3) Offset Address: 0x20+0x4*n

Name

Bit		access	Describes	Reset value
[31:0] IPC_MSGx[31:0]		RW	the IPC information register, used to store and transmit information required for inter-core communication. The message to be conveyed.	X

4.5 Hardware Semaphores (HSEM)

The hardware semaphore feature provides 32 hardware semaphore channels. Semaphores can be used to manage communication between different processes running on a multi-core system.

Step-by-step operation, and provides a non-blocking single-instruction locking mode. Features include the following:

An interrupt is generated when the semaphore channel is released .

ÿSingle -channel semaphore unlocking: Unlocking a channel requires matching the kernel operation with the CID and writing data with the PID value.

ÿ Batch unlocking of semaphore channels: Batch unlocking via key value, CID, and PID matching.

Two semaphore channel locking methods:

- 2-step locking: By writing the PID lock to the SEM channel, the CID is automatically updated to the kernel ID value that acquired the lock, and then read...

The data was checked.

- 1-step locking: The lock is acquired by reading the SEM channel, and the CID is automatically updated to the kernel ID value of the acquired lock.

4.5.1 HSEM Function Description 4.5.1.1

Locking Methods This module

provides two locking methods:

ÿ 2-step (write) lock

ÿ 1-step (read) lock: A 0 value in the HSEM

channel LOCK register indicates the channel is idle; at this time, the CID and PID values for that channel are also 0. When the channel LOCK register...

When the register value is 1, it indicates that the channel is locked. The CID value is updated to the kernel ID of the locked channel, and the PID register value is the bus write value.

When writing to a locked channel, the CID is updated to the kernel ID of the locked channel, and the PID value is updated to the written data. When reading from a locked channel, the CID is updated...

The new kernel ID is used to lock the channel, and the PID value remains 0. Any PID value can be written when writing to the lock.

2-step (write) lock process

The 2-step locking process locks the channel by writing lock information to the SEM_RX register, and then checks if the returned data read matches the written data.

The lookup was successful.

Write the PID to the SEM_RX register, set LOCK = 1, and lock successfully when the channel is idle. Update the PID, CID, and LOCK values.

ÿ Read data check: the read data PID matches the write data, the CID matches the kernel ID, and a LOCK value of 1 indicates that the operation was successful.

Otherwise , repeat the above steps.

The HSEM channel can only be locked when idle.

Re-locking an already locked channel will have no effect.

1-step (read) lock process

The 1-step lock process checks for successful locking by reading the SEM_RLRX register and comparing the return value.

Read the SEM_RLRX register to obtain the return value.

If the read data is 0, the lock was successful; a non-zero value indicates the lock failed, meaning the channel has already been locked by another process.

The SEM_RLRX register value for the operation is updated to LOCK = 1, CID = the kernel ID of the operation, and PID = 0.

Otherwise , repeat the above steps.

ÿChannel unlocking process:

Write to unlock:

Write the PID to the SEM_RX register corresponding to the locked SEM channel, and set LOCK=0. If the channel is locked by the kernel, write the PID and...

If the channel PIDs are equal, the channel is unlocked, and PID, CID, and LOCK are reset to zero. An interrupt request can be generated when the channel interrupt is enabled.

Match unlock:

Write a key value to the SEM_CLR register to configure the matching rules. Channels matching the rules can be unlocked in batches. When a channel...

When the enable function is disabled, an interrupt request can be generated.

4.5.1.2 HSEM Interruption An

interrupt can occur when the HSEM channel is unlocked. HSEM interrupts have the following characteristics:

Each HSEM channel can be enabled independently.

Each HSEM channel has an independent interrupt flag.

Each HSEM channel interruption flag can be cleared individually.

Each kernel has its own independent interrupt enable, interrupt flag, and clear bit.

When channel interrupts are enabled (HSEM_IER), an interrupt request can be generated when the channel is unlocked by another kernel, corresponding to the channel interrupt.

Set the flag register (HSEM_ISR) to 1.

Write 1 to the corresponding bit of the channel interrupt flag (HSEM_ISR) to clear the interrupt flag.

The Channel Interrupt Mask Register (HSEM_ISM) can be used to obtain the source channel that generated the interrupt, with channel interrupt enabled and the interrupt flag set to 1.

Only channels that meet certain conditions can generate an interrupt request, and the corresponding HSEM_ISM bit value is 1.

4.5.2 HSEM Register

Table 4-9 List of HSEM-related registers

name	Access address description		Reset value
R32_HSEM_RX0	0xE000C000	HSEM Lock Channel Register 0	0x00000000
R32_HSEM_RX1	0xE000C004	HSEM Lock Channel Register 1	0x00000000
R32_HSEM_RX2	0xE000C008	HSEM Lock Channel Register 2	0x00000000
R32_HSEM_RX3	0xE000C00C	HSEM Lock Channel Register 3	0x00000000
R32_HSEM_RX4	0xE000C010	HSEM Lock Channel Register 4	0x00000000
R32_HSEM_RX5	0xE000C014	HSEM Lock Channel Register 5	0x00000000
R32_HSEM_RX6	0xE000C018	HSEM Lock Channel Register 6	0x00000000
R32_HSEM_RX7	0xE000C01C	HSEM Lock Channel Register 7	0x00000000
R32_HSEM_RX8	0xE000C020	HSEM Locked Channel Register 8	0x00000000
R32_HSEM_RX9	0xE000C024	HSEM Lock Channel Register 9	0x00000000
R32_HSEM_RX10	0xE000C028	HSEM Lock Channel Register 10	0x00000000
R32_HSEM_RX11	0xE000C02C	HSEM Lock Channel Register 11	0x00000000
R32_HSEM_RX12	0xE000C030	HSEM Lock Channel Register 12	0x00000000
R32_HSEM_RX13	0xE000C034	HSEM Lock Channel Register 13	0x00000000
R32_HSEM_RX14	0xE000C038	HSEM Lock Channel Register 14	0x00000000
R32_HSEM_RX15	0xE000C03C	HSEM Lock Channel Register 15	0x00000000
R32_HSEM_RX16	0xE000C040	HSEM Lock Channel Register 16	0x00000000
R32_HSEM_RX17	0xE000C044	HSEM Lock Channel Register 17	0x00000000
R32_HSEM_RX18	0xE000C048	HSEM Lock Channel Register 18	0x00000000
R32_HSEM_RX19	0xE000C04C	HSEM Lock Channel Register 19	0x00000000
R32_HSEM_RX20	0xE000C050	HSEM Lock Channel Register 20	0x00000000
R32_HSEM_RX21	0xE000C054	HSEM Lock Channel Register 21	0x00000000
R32_HSEM_RX22	0xE000C058	HSEM Lock Channel Register 22	0x00000000
R32_HSEM_RX23	0xE000C05C	HSEM Lock Channel Register 23	0x00000000
R32_HSEM_RX24	0xE000C060	HSEM Lock Channel Register 24	0x00000000
R32_HSEM_RX25	0xE000C064	HSEM Lock Channel Register 25	0x00000000
R32_HSEM_RX26	0xE000C068	HSEM Lock Channel Register 26	0x00000000
R32_HSEM_RX27	0xE000C06C	HSEM Lock Channel Register 27	0x00000000
R32_HSEM_RX28	0xE000C070	HSEM Locked Channel Register 28	0x00000000
R32_HSEM_RX29	0xE000C074	HSEM Lock Channel Register 29	0x00000000
R32_HSEM_RX30	0xE000C078	HSEM Lock Channel Register 30	0x00000000

R32_HSEM_RX31	0xE000C07C	HSEM Lock Channel Register 31	0x00000000
R32_HSEM_RLRX0	0xE000C100	HSEM Read Lock Register 0	0x00000000
R32_HSEM_RLRX1	0xE000C104	HSEM Read Lock Register 1	0x00000000
R32_HSEM_RLRX2	0xE000C108	HSEM Read Lock Register 2	0x00000000
R32_HSEM_RLRX3	0xE000C10C	HSEM Read Lock Register 3	0x00000000
R32_HSEM_RLRX4	0xE000C110	HSEM Read Lock Register 4	0x00000000
R32_HSEM_RLRX5	0xE000C114	HSEM Read Lock Register 5	0x00000000
R32_HSEM_RLRX6	0xE000C118	HSEM Read Lock Register 6	0x00000000
R32_HSEM_RLRX7	0xE000C11C	HSEM Read Lock Register 7	0x00000000
R32_HSEM_RLRX8	0xE000C120	HSEM Read Lock Register 8	0x00000000
R32_HSEM_RLRX9	0xE000C124	HSEM Read Lock Register 9	0x00000000
R32_HSEM_RLRX10	0xE000C128	HSEM Read Lock Register 10	0x00000000
R32_HSEM_RLRX11	0xE000C12C	HSEM Read Lock Register 11	0x00000000
R32_HSEM_RLRX12	0xE000C130	HSEM Read Lock Register 12	0x00000000
R32_HSEM_RLRX13	0xE000C134	HSEM Read Lock Register 13	0x00000000
R32_HSEM_RLRX14	0xE000C138	HSEM Read Lock Register 14	0x00000000
R32_HSEM_RLRX15	0xE000C13C	HSEM Read Lock Register 15	0x00000000
R32_HSEM_RLRX16	0xE000C140	HSEM Read Lock Register 16	0x00000000
R32_HSEM_RLRX17	0xE000C144	HSEM Read Lock Register 17	0x00000000
R32_HSEM_RLRX18	0xE000C148	HSEM Read Lock Register 18	0x00000000
R32_HSEM_RLRX19	0xE000C14C	HSEM Read Lock Register 19	0x00000000
R32_HSEM_RLRX20	0xE000C150	HSEM Read Lock Register 20	0x00000000
R32_HSEM_RLRX21	0xE000C154	HSEM Read Lock Register 21	0x00000000
R32_HSEM_RLRX22	0xE000C158	HSEM Read Lock Register 22	0x00000000
R32_HSEM_RLRX23	0xE000C15C	HSEM Read Lock Register 23	0x00000000
R32_HSEM_RLRX24	0xE000C160	HSEM Read Lock Register 24	0x00000000
R32_HSEM_RLRX25	0xE000C164	HSEM Read Lock Register 25	0x00000000
R32_HSEM_RLRX26	0xE000C168	HSEM Read Lock Register 26	0x00000000
R32_HSEM_RLRX27	0xE000C16C	HSEM Read Lock Register 27	0x00000000
R32_HSEM_RLRX28	0xE000C170	HSEM Read Lock Register 28	0x00000000
R32_HSEM_RLRX29	0xE000C174	HSEM Read Lock Register 29	0x00000000
R32_HSEM_RLRX30	0xE000C178	HSEM Read Lock Register 30	0x00000000
R32_HSEM_RLRX31	0xE000C17C	HSEM Read Lock Register 31	0x00000000
R32_HSEM_LSE	0xE000C200	HSEM Lock Status Register	0x00000000
R32_HSEM_CLR	0xE000C208	HSEM unlock register	0x00000000
R32_HSEM_KEY	0xE000C20C	HSEM Unlock Key Register	0x5AA50003
R32_HSEM_IER	0xE000C300	HSEM interrupt enable register	0x00000000
R32_HSEM_ISR	0xE000C308	HSEM Interrupt Status Register	0x00000000
R32_HSEM_ISM	0xE000C310	HSEM Interrupt Mask Register	0x00000000
R32_HSEM_LSM	0xE000C318	HSEM Lock State Mask Register	0x00000000

Note: Registers marked "kernel private registers" are multiple sets of registers in the peripheral device that have the same function, independent data, and whose data cannot be accessed or modified by other kernels. Only the kernel has the same access address for them.

4.5.2.1 HSEM Lock Channel Register y (HSEM_RXy) (y=0-31)

Offset address: 0x000 + 4y bit

name		access	describe	Reset value
31	LOCK	RW	Channel lock status indicator: 1: Channel locked; 0: Channel is idle.	0
[30:10] Reserved		RO reserved.		0
[9:8] CID		RW	Kernel ID for locked channel: 1: The channel is locked by kernel 1; 0: Channel is idle or locked by kernel 0.	0
[7:0] PID		RW	Process ID for locking the channel: The reset value or channel is loaded to 0 after release; write lock is engaged during locking. Updated to the bus data being written.	0

4.5.2.2 HSEM Read Lock Register y (HSEM_RLRXy) (y=0-31) Offset Address: 0x100 + 4y

Bit Name

		access	describe	Reset value
31	LOCK	RO	Channel locked status indicator: 1: Channel locked; 0: Channel is idle.	0
[30:10] Reserved		RO reserved.		0
[9:8] CID		RO	Kernel ID for locked channel: 1: The channel is locked by kernel 1; 0: Channel is idle or locked by kernel 0.	0
[7:0] PID		RO is the process ID for locking the channel; it is 0 when reading the lock.		0

4.5.2.3 HSEM Lock Status Register (HSEM_LSE) Offset Address:

0x200 Bit Name

		access		Reset value
[31:0] HSEM_LSE		RO	Describes the HSEM channel lock status register, used to obtain the status of 32 channels. Locked status: 1: Channel locked; 0: Channel is idle.	0

4.5.2.4 HSEM Unlock Register (HSEM_CLR) Offset Address:

0x208 Bit Name

		access	The	Reset value
[31:16] CLR_KEY		WO	description unlocks keywords; operations are performed when the written data matches the matching keywords. If valid, then invalid. The key value is 0x5AA5.	0
[15:14] Reserved		RO is reserved.		0
13	CID_MASK	WO	CID Matching Masking: 1: Block CID matching; all CID values are unlocked. 0: Do not block CID matching; lock the channel CID value. Unlock when the match_CID values are the same.	0
12	PID_MASK	WO	PID matching masking:	0

			1: Disable PID matching; all PID values are unlocked. 0: Do not mask PID matching; the locked channel PID value is matched with... Unlock when the match_PID value is the same.	
[11:10] Reserved		RO is reserved.		0
[9:8] match_CID		WO	Used to match the unlocked CID value with the CID of each locked channel. Value	0
[7:0] match_PID		WO	comparison. The PID value used to match the unlocking value with the PID value of each locking channel. Value comparison.	0

4.5.2.5 HSEM Unlock Key Register (HSEM_KEY)

Offset address: 0x20C Bit

name		Access description reset value		
[31:16] KEY_VALUE		RO (Unlock Keyword) is used for keyword comparison of unlocking methods. 0x5AA5		
[15:2] Reserved		RO is reserved. The		0
1	auto_2step_clr	RW	corresponding channel is automatically cleared if locking the channel using the 2-step method fails. Channel interrupt status bit enabled: 1: Automatically clear the interrupt status bit of the corresponding channel when locking fails; 0: Locking failure does not affect the corresponding channel interrupt status bit.	1
0	auto_1step_clr	RW	Automatically clear the corresponding channel when locking the channel using the 1-step method fails. Channel interrupt status bit enabled: 1: Automatically clear the interrupt status bit of the corresponding channel when locking fails; 0: Locking failure does not affect the corresponding channel interrupt status bit.	1

4.5.2.6 HSEM Interrupt Enable Register (HSEM_IER) Offset Address:

0x300 Bit Name

		access		Reset value
[31:0] HSEM_IER		RW	Describes the HSEM channel interrupt enable register, a kernel-private register. Setting it to 1 allows the generation of intermediate signals when the corresponding channel is unlocked by other kernels. An interrupt can be generated when used in conjunction with the HSEM_ISR register.	0

4.5.2.7 HSEM Interrupt Status Register (HSEM_ISR) Offset Address: 0x308

Bit Name

		access		Reset value
[31:0] HSEM_ISR		RW1Z	Describes the HSEM channel interrupt status register, a kernel-private register. When a locked channel is unlocked by another kernel, the corresponding bit is set. This bit, when used in conjunction with the HSEM_IER register, can generate an interrupt.	0

4.5.2.8 HSEM Interrupt Mask Register (HSEM_ISM) Offset Address: 0x310

Bit	name	access	describe	Reset value
[31:0] HSEM_ISM		RO	HSEM channel interrupt mask register, a kernel-private register. The value equals HSEM_ISR & HSEM_IER, and a non-zero value indicates that... This indicates that an SEM interrupt was generated, and the corresponding bit value of the channel that generated the interrupt. The value is 1.	0

4.5.2.9 HSEM Lock State Mask Register (HSEM_LSM)

Offset Address:

Bit	0x318 Name	access	describe	Reset value
[31:0]	HSEM_LSM	RO	HSEM channel lock status register, used to acquire HSEM channel status. Is it locked by this kernel? 1: The channel is locked by the kernel. 0: The channel is idle or locked by another kernel.	0

4.6 System Counter (SysTick)

The kernel comes with two 32-bit up/down counters (SysTick), which can independently generate interrupt signals and can be configured to generate interrupt signals for specified kernel nodes.

Disconnect request. The system counter counts at the same frequency as HCLK (V3F kernel frequency), and the RTC count is HCLK/8.

Dual 32-bit timers

ÿ Counting up and down

ÿThe main frequency clock and RTC clock count can be configured.

Interrupt requests are configurable in the kernel .

Automatic reloading is configurable.

4.6.1 RISC-V-SysTick Register Table 4-10 STK

Related Register List

name	Access Address Description: 0xE000F000 System	Reset value
R32_STK_CTLR_0	Counter Control Register 0; 0xE000F004 System Counter	0x00000000
R32_STK_ISR	Interrupt Status Register; 0xE000F008 System Counter 0 Count	0x00000000
R32_STK_CNT_0	Register; 0xE000F010 System Counter 0 Compare Register;	0x00000000
R32_STK_CMP_0	0xE000F080 System Counter Control Register 1; 0xE000F088	0x00000000
R32_STK_CTLR_1	System Counter 1 Count Register; 0xE000F090 System	0x00000000
R32_STK_CNT_1	Counter 1 Compare Register	0x00000000
R32_STK_CMP_1		0x00000000

4.6.1.1 System Counter Control Register 0 (STK_CTLR_0)

Offset Address:

Bit	0x00 Name	access	describe	Reset value
[31:7]	Reserved	RO is reserved.		0
6	CID_0	RW	Counter 0 is allocated to a kernel register for configuring counter 0. Generate an interrupt request to the specified kernel, and enable it at counter 0. When the configuration is valid, it will be automatically updated to the kernel ID of the operation.	0
5	Reserved	RO is reserved.		0
4	DOWN_MODE_0	RW	Counting mode: 1: Count down; 0: Count upwards.	0
3	AUTO_RELOAD_0	RW	Automatic reload counter enable bit: 1: Automatic reload;	0

			0: Do not reload automatically.	
2	NO_RTC_0	RW	Counter clock source selection bit: 1: Use HCLK as the time base; 0: Use HCLK/8 as the time base.	0
1	IE_0	RW	Interrupt enable control bit for counter 0: 1: Enable counter interrupt; 0: Disable counter interrupt.	0
0	EN_0	RW	System counter 0 enable control bit: 1: Start the system counter STK, and after setting it to 1, only CID and the operator... When performing kernel matching, the TIM_CTL_0 configuration can be modified; 0: Disable system counter STK, and the counter stops counting.	0

4.6.1.2 System Counter Interrupt Status Register (STK_ISR) Offset Address:

0x04

Bit	name	access	describe	Reset value
[31:2] Reserved		RO is reserved.		0
1	STK_ISR1	RW	Systick1 Interrupt Flag: 1: There is an unhandled interrupt request for Timer 1; 0: Timer 1 has no interrupt request.	0
0	STK_ISR0	RW	Systick0 Interrupt Flag: 1: There is an unhandled interrupt request in Timer 0; 0: No interrupt request is available for Timer 0.	0

4.6.1.3 System Counter 0 Count Register (STK_CNT_0) Offset Address: 0x08

Bit	name	access	describe	Reset value
[31:0] CNT_0[31:0]		RW	Systick0 is a counter register; after counting is enabled, it accumulates according to the configuration. Add/remove and automatic reload.	0

4.6.1.4 System Counter 0 Comparator Register (STK_CMP_0) Offset Address:

0x10

Bit	name	access	describe	Reset value
[31:0] CMP_0[31:0]		RW	Systick0 compare register is used for comparison during up-counting. Automatic reload when counting down.	0

4.6.1.5 System Counter Control Register 1 (STK_CTLR_1) Offset Address:

0x80 Bit Name [31:7]

Reserved		access	describe	Reset value
		RO is reserved.		0
6	CID_1	RW	Counter 1 is allocated to a kernel register for configuring counter 1. An interrupt request is generated to the specified kernel, and the interrupt is enabled by counter 1. When the configuration is valid, it will be automatically updated to the kernel ID of the operation.	0
5	Reserved	RO is reserved.		0
4	DOWN_MODE_1	RW counting mode:		0

			1: Count down; 0: Count upwards.	
3	AUTO_RELOAD_1	RW	Automatic reload counter enable bit: 1: Automatic reload; 0: Do not reload automatically.	0
2	NO_RTC_1	RW	Counter clock source selection bit: 1: Use HCLK as the time base; 0: Use HCLK/8 as the time base.	0
1	IE_1	RW	Interrupt enable control bit for counter 1: 1: Enable counter interrupt; 0: Disable counter interrupt.	0
0	EN_1	RW	System counter 1 enable control bit: 1: Start the system counter STK, and after setting it to 1, only CID and the operator... When performing kernel matching, the TIM_CTL_0 configuration can be modified; 0: Disable system counter STK, and the counter stops counting.	0

4.6.1.6 System Counter 1 Count Register (STK_CNT_1) Offset Address: 0x88 Name

Bit		access	describe	Reset value
[31:0] CNT_1[31:0]		RW	Systick1 is a counter register; once the counter is enabled, it accumulates data according to the configuration. Add/remove and automatic reload.	0

4.6.1.7 System Counter 1 Comparator Register (STK_CMP_1) Offset Address: 0x90

Bit	name	access	describe	Reset value
[31:0] CMP_1[31:0]		RW	Systick1 is a compare register used for comparing sums during up-counting. Automatic reload when counting down.	0

4.7 Interrupts and Events (PFIC)

Built-in Programmable Fast Interrupt Controller (PFIC), supporting up to 256

There are several interrupt vectors, with each of the RISC-V3F and RISC-V5F cores managing 32 core-private interrupts, and sharing management of other interrupts.

By configuring the interrupt allocation register, interrupts are assigned to one of the kernel responses. Some peripherals, such as IPC and SysTick, in interrupts lower 32 require...

The registers in the configured peripherals are assigned to a specific kernel, and an interrupt can only generate an interrupt request to one kernel at a time.

The interrupt control platform can control the kernel's sleep state and wake-up. When the kernel executes the WFI instruction, the kernel enters a low-power state.

The interrupt control platform will set the kernel sleep flag and monitor interrupts or events. When an interrupt or event occurs, the interrupt control platform will wake up the kernel.

Nuclear execution interrupts and other procedures.

4.7.1 Key Features of the PFIC Controller : ÿ

Supports up to 256 interrupt arbitration channels, with private interrupts in the lower 32 bits.

Shared interrupt allocation

Supports Hardware Interrupt Stack (HPE) without instruction overhead.

Provides 4-channel meter-free interrupt (VTF)

Supports low-power sleep and wake-up management .

The RISC-V3F kernel supports up to 2 levels of configurable interrupt nesting, while the RISC-V5F kernel supports up to 8 levels of configurable interrupt nesting .

4.7.2 Vector Table for Interrupts and Exceptions

Table 4-11 CH32H417 Series Chip Vector Table

Numbering	priority	type	-	6	name	describe	Offset address
0	Fixed				RESET	Reset interrupt,	0x000
1	-	-			-	non-	0x004
2	-5	Fixed	-4	Fixed	NMI	maskable interrupt,	0x008
3					HardFault	exception	0x00C
4	-	-			-		0x010
5	-3	fixed			Ecall-M	interrupt, machine mode	0x014
6-7	-	-			-		0x018-0x01C
8	-2	Fixed	-1	Fixed	Ecall-U	callback interrupt, user mode	0x020
9					Breakpoint	callback interrupt,	0x024
10-11	-	-			-		0x028-0x02C
12	0	Programmable	SysTick0	1		breakpoint callback interrupt,	0x030
13	Programmable	SysTick1	2			system timer interrupt 0,	0x034
14	Programmable				SW	system timer	0x038
15	-	-			-		0x03C
16	3	Programmable			IPC_CH0	interrupt 1, software	0x040
17	4	Programmable			IPC_CH1	interrupt, inter-core	0x044
18	5	Programmable			IPC_CH2	communication interrupt	0x048
19	6	Programmable			IPC_CH3	0, inter-core	0x04C
20-27	-	-			-		0x050-0x06C
28	7	Programmable			HSEM	communication interrupt	0x070
29-31	-	-			-	1, inter-	0x074-0x07C
32	8	Programmable			WWDG	core communication interrupt 2, inter-core communication interrupt 3, hardware semaph	0x080
33	9	Programmable	EXTI15_8	10		EXTI line [15:8] interrupt	0x084
34	Programmable				FLASH	flash global	0x088
35	11	Programmable			RCC	interrupt reset and clock control interrupt	0x08C
36	12	Programmable	EXTI7_0	13		EXTI line [7:0] interrupted	0x090
37	Programmable				SPI1	SPI1 Global Interrupt	0x094
38	14	Programmable	DMA1_CH2	15		DMA1 Channel 2 Global Interrupt	0x098
39	Programmable	DMA1_CH3	16			DMA1 Channel 3 Global Interrupt	0x09C
40	Programmable	DMA1_CH4	17			DMA1 Channel 4 Global Interrupt	0x0A0
41	Programmable	DMA1_CH5	18			DMA1 Channel 5 Global Interrupt	0x0A4
42	Programmable	DMA1_CH6	19			DMA1 Channel 6 Global Interrupt	0x0A8
43	Programmable	DMA1_CH7	20			DMA1 Channel 7 Global Interrupt	0x0AC
44	Programmable	DMA1_CH8	21			DMA1 Channel 8 Global Interrupt	0x0B0
45	Programmable				USART2	USART2 Global Interrupt	0x0B4
46	22	Programmable			I2C1_EV	I^2C1 event interruption	0x0B8
47	23	Programmable			I2C1_ER	I^2C1 error interrupt	0x0BC
48	24	Programmable			USART1	USART1 Global Interrupt	0x0C0
49	25	Programmable			SPI2	SPI2 Global Interrupt	0x0C4
50	26	Programmable			SPI3	SPI3 Global Interrupt	0x0C8

51	27 Programmable 28	SPI4	SPI4 Global Interrupt	0x0CC
52	Programmable 29	I2C2_EV	I²C2 event interruption	0x0D0
53	Programmable 30	I2C2_ER	I²C2 error interruption	0x0D4
54	Programmable 31	USBPD	USBPD Global Interrupt	0x0D8
55	Programmable USBPDWakeUP 32 Programmable		USBPD wake-up interrupt	0x0DC
56	33 Programmable	USBHS	USBHS Global Interrupt	0x0E0
57	DMA1_CH1 34 Programmable CAN1_SCE 35		DMA1 Channel 1 Global Interrupt	0x0E4
58	Programmable CAN1_TX 36 Programmable		CAN1_SCE Global Interrupt	0x0E8
59	CAN1_RX0 37 Programmable CAN1_RX1 38		CAN1_TX Global Interrupt	0x0EC
60	Programmable 39 Programmable USBSS_LINK		CAN1_RX0 Global Interrupt	0x0F0
61	40 Programmable USBHSWakeUP 41		CAN1_RX1 Global Interrupt	0x0F4
62	Programmable	USBSS	USBSS Global Interrupt	0x0F8
63	USBSSWakeUP 42 Programmable RTCAlarm		USBSS LINK interrupted	0x0FC
64	43 Programmable 44 Programmable		USBHS wake-up interrupt	0x100
65	USBFSWakeUP 45 Programmable		USBSS wake-up interrupt	0x104
66			RTC alarm interruption (EXTI)	0x108
67		USBFS	USBFS Global Interrupt	0x10C
68			USBFS wake-up interrupt	0x110
69		ADC1_2	ADC1_2 Global Interrupt	0x114
70	46 Programmable TIM1_BRK 47 Programmable		TIM1 Brake Interruption	0x118
71	TIM1_UP 48 Programmable TIM1_TRG_COM		TIM1 update interrupted	0x11C
72	49 Programmable TIM1_CC 50 Programmable		TIM1 Triggering and Communication Interruption	0x120
73	51 Programmable 52 Programmable 53		TIM1 capture comparison interrupt	0x124
74	Programmable 54	TIM2	TIM2 Global Interrupt	0x128
75	Programmable 55	TIM3	TIM3 Global Interrupt	0x12C
76	Programmable 56	TIM4	TIM4 Global Interrupt	0x130
77	Programmable 57	TIM5	TIM5 Global Interrupt	0x134
78	Programmable 58	I2C3_EV	I²C3 event interruption	0x138
79	Programmable 59	I2C3_ER	I²C3 Error Interruption	0x13C
80	Programmable 60	I2C4_EV	I²C4 event interruption	0x140
81	Programmable 61	I2C4_ER	I²C4 Error Interruption	0x144
82	Programmable 62	QSPI1	QSPI1 Global Interrupt	0x148
83	Programmable	SERDES	SERDES Global Interrupts	0x14C
84	TIM8_BRK 63	USART3	USART3 Global Interrupt	0x150
85	Programmable	USART4	USART4 Global Interrupt	0x154
86	TIM8_UP 64 Programmable TIM8_TRG_COM		TIM8 Brake Interruption	0x158
87	65 Programmable TIM8_CC 66 Programmable		TIM8 update interrupted	0x15C
88	67 Programmable 68 Programmable 69		TIM8 Triggering and Communication Interruption	0x160
89	Programmable 70 Programmable		TIM8 capture comparison interrupt	0x164
90		TIM9	TIM9 Global Interrupt	0x168
91		TIM10	TIM10 Global Interrupt	0x16C
92		TIM11	TIM11 Global Interrupt	0x170
93		TIM12	TIM12 Global Interrupt	0x174
94		FMC	FMC Global Interrupt	0x178

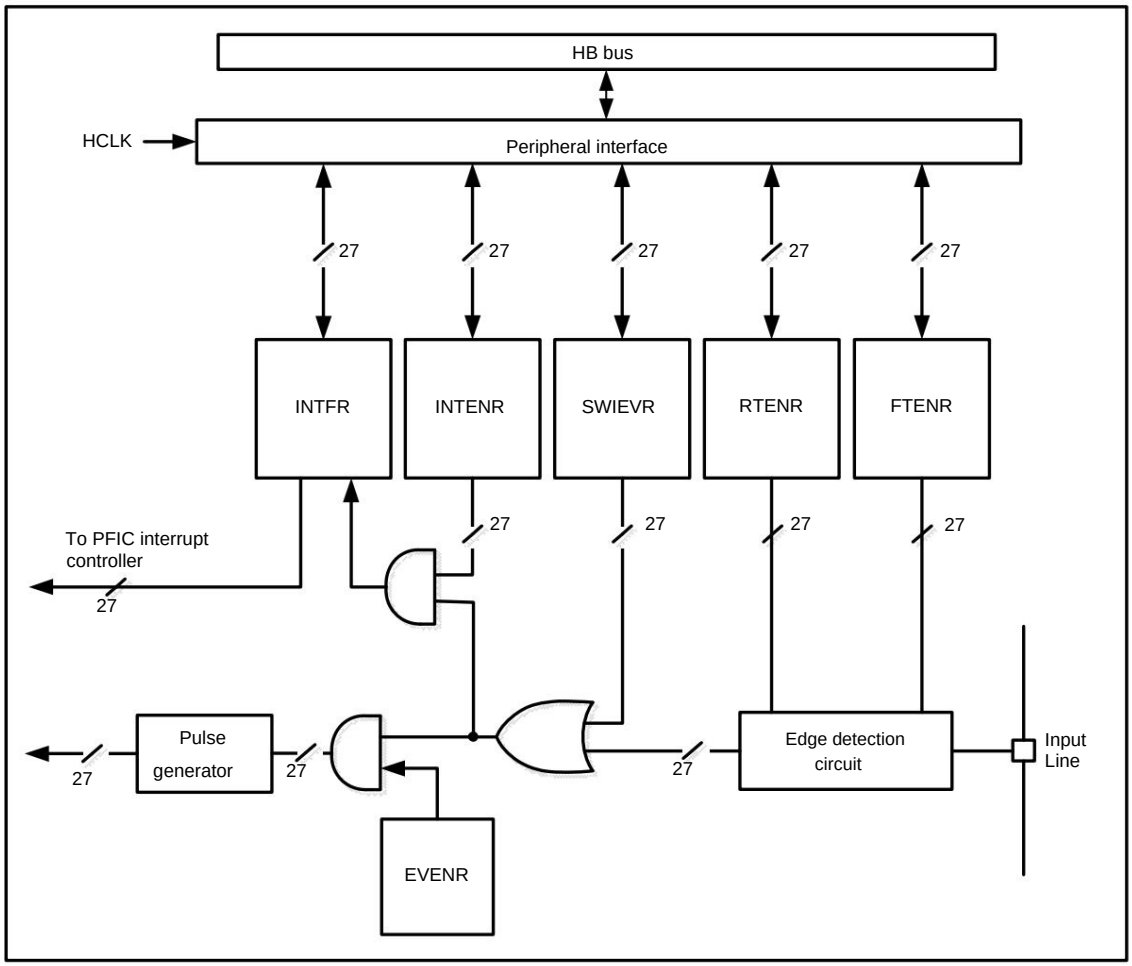
95	71 Programmable	SDMMC	SDMMC Global Interrupt	0x17C
96	72 Programmable	LPTIM1	LPTIM1 Global Interrupt	0x180
97	73 Programmable	LPTIM2	LPTIM2 Global Interrupt	0x184
98	74 Programmable	USART5	USART5 Global Interrupt	0x188
99	75 Programmable	USART6	USART6 Global Interrupt	0x18C
100	76 Programmable	TIM6	TIM6 Global Interrupt	0x190
101	77 Programmable	TIM7	TIM7 Global Interrupt	0x194
102	78 Programmable DMA2_CH1 79		DMA2 Channel 1 Global Interrupt	0x198
103	Programmable DMA2_CH2 80		DMA2 Channel 2 Global Interrupt	0x19C
104	Programmable DMA2_CH3 81		DMA2 Channel 3 Global Interrupt	0x1A0
105	Programmable DMA2_CH4 82		DMA2 Channel 4 Global Interrupt	0x1A4
106	Programmable DMA2_CH5 83		DMA2 Channel 5 Global Interrupt	0x1A8
107	Programmable DMA2_CH6 84		DMA2 Channel 6 Global Interrupt	0x1AC
108	Programmable DMA2_CH7 85		DMA2 Channel 7 Global Interrupt	0x1B0
109	Programmable DMA2_CH8 86		DMA2 Channel 8 Global Interrupt	0x1B4
110	Programmable 87	ETH	ETH Global Interruption	0x1B8
111	Programmable ETH_WKUP 88		ETH wake-up interrupt	0x1BC
112	Programmable CAN2_SCE 89		CAN2_SCE Global Interrupt	0x1C0
113	Programmable CAN2_TX 90		CAN2_TX Global Interrupt	0x1C4
114	Programmable CAN2_RX0 91		CAN2_RX0 Global Interrupt	0x1C8
115	Programmable CAN2_RX1 92		CAN2_RX1 Global Interrupt	0x1CC
116	Programmable 93	USART7	USART7 Global Interrupt	0x1D0
117	Programmable 94	USART8	USART8 Global Interrupt	0x1D4
118	Programmable 95	I3C_EV	I^3C event interruption	0x1D8
119	Programmable 96	I3C_ER	I^3C Error Interruption	0x1DC
120	Programmable 97	DVP	DVP Global Interrupt	0x1E0
121	Programmable 98	ECDC	ECDC Global Interrupt	0x1E4
122	Programmable 99	PIOC	PIOC Global Interrupt	0x1E8
123	Programmable 100	SAI	SAI Global Interrupt	0x1EC
124	Programmable 101	LTDC	LTDC Global Interrupt	0x1F0
125	Programmable	GPHA	GPHA Global Interrupt	0x1F4
126	- -	-	Retention	0x1F8
127	102 Programmable	DFSDM0	DFSDM0 Global Interrupt	0x1FC
128	103 Programmable	DFSDM1	DFSDM1 Global	0x200
129	- -	-		0x204
130	- -	-	Interrupt Reservation	0x208
131	104 Programmable	SWPMI	SWPMI Global	0x20C
132	- -	-		0x210
133	- -	-	Interrupt Reservation	0x214
134	105 Programmable	QSPI2	QSPI2 Global Interrupt	0x218
135	106 Programmable SWPMIWakeUP 107		SWPMI wake-up interrupt	0x21C
136	Programmable CAN3_SCE 108		CAN3_SCE Global Interrupt	0x220
137	Programmable CAN3_TX 109		CAN3_TX Global Interrupt	0x224
138	Programmable CAN3_RX0		CAN3_RX0 Global Interrupt	0x228

139	110 Programmable CAN3_RX1 111 Programmable		CAN3_RX1 Global Interrupt	0x22C
140	LPTIM2 WakeUP 112 Programmable LPTIM1		LPTIM2 wake-up interrupt	0x230
141	WakeUP 113 Programmable I3C WakeUP 114		LPTIM1 wake-up interrupt	0x234
142	Programmable 115 Programmable 116		I^3C wake-up interrupt	0x238
143	Programmable 117	RTC	RTC Global Interrupt	0x23C
144	Programmable 118	HSADC	HSADC Global Interrupt	0x240
145	Programmable 119	UHSIF	UHSIF Global Interrupt	0x244
146	Programmable USART	RNG	RNG Global Interrupt	0x248
147	WakeUP	SDIO	SDIO Global Interrupt	0x24C
148			USART wake-up interrupt	0x250

4.7.3 External Interrupts and Event Controllers

(EXTI) 4.7.3.1 Overview

Figure 4-1 External Interrupt (EXTI) Interface Block Diagram



As shown in Figure 4-1, the trigger source for an external interrupt can be either a software interrupt (SWIEVR) or an actual external interrupt.

In this case, the signal from the external interrupt channel will first be filtered by the edge detection circuit. This is as long as the signal is generated in the software...

If either an interrupt signal or an external interrupt signal is received, it will be output through the OR gate circuit shown in the diagram to the two AND gate circuits for event enable and interrupt enable. As long as there is an interrupt signal...

An interrupt or event is generated when an interrupt is enabled or an event is enabled. The six registers of EXTI are accessed by the processor through the HB interface.

4.7.3.2 Wake-up Event Description The

system can wake up from sleep mode caused by WFE commands through wake-up events. Wake-up events are generated through the following three configurations:

Enable an interrupt in the peripheral device's register, but not in the kernel's PFIC, while simultaneously enabling it in the kernel.

The SEVONPEND bit. In EXTI, this means enabling the EXTI interrupt, but not enabling the EXTI interrupt in the PFIC, and simultaneously enabling...

The SEVONPEND bit. When the CPU wakes up from WFE, the EXTI interrupt flag and PFIC pending bit need to be cleared.

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

A write event is generated when another running kernel writes to the SENDEVENT register.

4.7.3.3 This section explains

that using external interrupts requires configuring the corresponding external interrupt channel, i.e., selecting the appropriate trigger edge and enabling the corresponding interrupt. When the external interrupt channel...

When the set trigger edge occurs, an interrupt request will be generated, and the corresponding interrupt flag will be set. Writing 1 to the flag will...

Clear this flag.

Steps for using external hardware interrupts:

1) Configure GPIO operations;

2) Configure the interrupt enable bit (EXTI_INTENR) for the corresponding external interrupt channel;

3) Configure the trigger edge (EXTI_RTENR or EXTI_FTENR), selecting rising edge trigger, falling edge trigger, or double edge trigger;

4) Configure the EXTI interrupt in the kernel's PFIC to ensure that it can respond correctly.

Steps for using external hardware events:

1) Configure 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), and select rising edge trigger, falling edge trigger, or double edge trigger.

Using software interrupt/event steps:

1) Enable external interrupts (EXTI_INTENR) or external events (EXTI_EVENR);

2) If using interrupt service routines, the EXTI interrupt needs to be configured in the kernel's PFIC;

3) Set the software interrupt trigger (EXTI_SWIEVR) to generate an interrupt.

4.7.3.4 External Event Mapping

Table 4-12 EXTI Interrupt Mapping

External interrupt/event line	Mapping event description
EXTI0ÿEXTI15	Any one of the I/O ports Px0 to Px15 (x = A/B/C/D/E) can be enabled. External interrupt/event functionality is used, configured via the AFIO_EXTICRx register.
EXTI16	USBHS Wake-up Event
EXTI17	RTC alarm clock incident
EXTI18	SWPMI wake-up events
EXTI19	USBFSOTG Wake-up Event
EXTI20	USBPD Wake-up Event
EXTI21	ETH wake-up event
EXTI22	USBSS wake-up event
EXTI23	LPTIM1 wake-up event
EXTI24	LPTIM2 wake-up event
EXTI25	I3C wake-up event
EXTI26	USART wake-up event

4.7.4 EXTI Register Description

Table 4-13 List of EXTI-related registers

name	Access Address Description: 0x40010400	Reset value
R32_EXTI_INTENR	Interrupt Enable Register; 0x40010404	0x00000000
R32_EXTI_EVENR	Event Enable Register; 0x40010408	0x00000000
R32_EXTI_RTENR	Rising Edge Trigger Enable Register; 0x4001040C	0x00000000
R32_EXTI_FTENR	Falling Edge Trigger Enable Register; 0x40010410	0x00000000
R32_EXTI_SWIEVR	Soft Interrupt Event Register; 0x40010414	0x00000000
R32_EXTI_INTFR	Interrupt Flag Register	0x0000XXXX

4.7.4.1 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		MR26 MR25 MR24 MR23 MR22 MR21 MR20 MR19	MR18	MR17	MR16		

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

Bit	name	access	describe	Reset value
[31:27] Reserved		RO is reserved.		0
[26:0] MRx		RW	Enable the interrupt request signal for external interrupt channel x: 1: Enable interruption for this channel; 0: Disables interrupts on this channel.	0

4.7.4.2 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		MR26 MR25 MR24 MR23 MR22 MR21 MR20 MR19	MR18	MR17	MR16		

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

Bitname		access	describe	Reset value
[31:27] Reserved		RO is reserved.		0
[26:0] MRx		RW	Enable the event request signal for external interrupt channel x: 1: Events that enable this channel; 0: Block events for this channel.	0

4.7.4.3 Rising edge triggered enable register (EXTI_RTENR)

offset address: 0x08

31	30 29 28	27 26 25 24 23 22	20	19	18	17	16
Reserved		TR26 TR25 TR24 TR23 TR22 TR21 TR20 TR19	TR18	TR17	TR16		

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

TR15	TR14	TR13	TR12	TR11	TR10	TR9	TR8	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0						

Bit	name	access	describe	Reset value
[31:27] Reserved		RO is reserved.		0
[26:0] TRx		RW	Enable rising edge triggering of external interrupt channel x: 1: Enable rising edge triggering for this channel; 0: Disable rising edge triggering for this channel.	0

4.7.4.4 Falling edge triggered enable register (EXTI_FTENR)

offset address: 0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					TR26	TR25	TR24	TR23	TR22	TR21	TR20	TR19	TR18	TR17	TR16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TR15	TR14	TR13	TR12	TR11	TR10	TR9	TR8	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0

Bit	name	access	describe	Reset value
[31:27] Reserved		RO Retention		0
[26:0] TRx		RW	Enable falling edge triggering of external interrupt channel x: 1: Enable falling edge triggering for this channel; 0: Disable falling edge triggering for this channel.	0

4.7.4.5 Offset address of the software interrupt event register (EXTI_SWIEVR) : 0x10

31	30	29	28	27	26	25	24	23	22	21	2019		18	17	16
Reserved					SWIER 26	SWIER 25	SWIER twenty four	SWIER twenty three	SWIER twenty two	SWIER twenty one	SWIER 20	SWIER 19	SWIER 18	SWIER 17	SWIER 16

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

Bit	name	access	describe	Reset value
[31:27] Reserved		RO is reserved.		0
[26:0] SWIERx		RW	A soft interrupt is set on the corresponding external trigger interrupt channel. The device is interrupted. Setting this bit will cause the interrupt flag to be set. The corresponding bit (EXTI_INTFR) is set if interrupt is enabled. (EXTI_INTENR) or event enable (EXTI_EVENR) on If it is started, then an interrupt or event will occur.	0

4.7.4.6 Interrupt Flag Register (EXTI_INTFR)

Offset address: 0x14

31	30	29	28	27	26	25	24	23	22	21	2019		18	17	16

Reserved															
IF26 IF25 IF24 IF23 IF22 IF21 IF20 IF19 IF18 IF17 IF16															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IF15 IF14 IF13 IF12 IF11 IF11 IF10 IF9 IF8 IF7 IF6 IF5 IF5 IF4 IF3 IF2 IF1															IF0

Bit	name	access	describe	Reset value
[31:27] Reserved		RO	Reserved.	0
[26:0] IFx		RW1Z	Interrupt flag, this bit indicates that a corresponding interrupt has occurred. External interrupt. Writing 1 will clear this bit.	X

4.7.5 PFIC Register

Table 4-14 List of PFIC Related Registers

name	Access address	describe	Reset value
R32_PFIC_ISR1	0xE000E000	PFIC Interrupt Enable Status Register 1	0x0000032C
R32_PFIC_ISR2	0xE000E004	PFIC Interrupt Enable Status Register 2	0x00000000
R32_PFIC_ISR3	0xE000E008	PFIC Interrupt Enable Status Register 3	0x00000000
R32_PFIC_ISR4	0xE000E00C	PFIC Interrupt Enable Status Register 4	0x00000000
R32_PFIC_ISR5	0xE000E010	PFIC Interrupt Enable Status Register 5	0x00000000
R32_PFIC_IPR1	0xE000E020	PFIC Interrupt Suspension Status Register 1	0x00000000
R32_PFIC_IPR2	0xE000E024	PFIC Interrupt Suspension Status Register 2	0x00000000
R32_PFIC_IPR3	0xE000E028	PFIC Interrupt Suspension Status Register 3	0x00000000
R32_PFIC_IPR4	0xE000E02C	PFIC Interrupt Suspension Status Register 4	0x00000000
R32_PFIC_IPR5	0xE000E030	PFIC Interrupt Suspension Status Register 5	0x00000000
R32_PFIC_ITHRESDR	0xE000E040	PFIC interrupt priority threshold configuration register	0x00000000
R32_PFIC_CFGR	0xE000E048	PFIC Interrupt Configuration Register	0x00000000
R32_PFIC_GISR	0xE000E04C	PFIC Interrupt Global Status Register	0x00000000
R32_PFIC_VTFIDR	0xE000E050	PFIC VTF Interrupt ID Configuration Register	0xXXXXXXXX
R32_PFIC_VTFADDRR0	0xE000E060	PFIC VTF Interrupt 0 Offset Address Register	0xXXXXXXXX
R32_PFIC_VTFADDRR1	0xE000E064	PFIC VTF Interrupt 1 Offset Address Register	0xXXXXXXXX
R32_PFIC_VTFADDRR2	0xE000E068	PFIC VTF Interrupt 2 Offset Address Register	0xXXXXXXXX
R32_PFIC_VTFADDRR3	0xE000E06C	PFIC VTF Interrupt 3 Offset Address Register	0xXXXXXXXX
R32_PFIC_IENR1	0xE000E100	PFIC Interrupt Enable Setting Register 1	0x00000000
R32_PFIC_IENR2	0xE000E104	PFIC Interrupt Enable Setting Register 2	0x00000000
R32_PFIC_IENR3	0xE000E108	PFIC Interrupt Enable Setting Register 3	0x00000000
R32_PFIC_IENR4	0xE000E10C	PFIC Interrupt Enable Setting Register 4	0x00000000
R32_PFIC_IENR5	0xE000E110	PFIC Interrupt Enable Setting Register 5	0x00000000
R32_PFIC_IRER1	0xE000E180	PFIC interrupt enable clear register 1	0x00000000
R32_PFIC_IRER2	0xE000E184	PFIC interrupt enable clear register 2	0x00000000
R32_PFIC_IRER3	0xE000E188	PFIC interrupt enable clear register 3	0x00000000
R32_PFIC_IRER4	0xE000E18C	PFIC interrupt enable clear register 4	0x00000000
R32_PFIC_IRER5	0xE000E190	PFIC interrupt enable clear register 5	0x00000000
R32_PFIC_IPSR1	0xE000E200	PFIC Interrupt Suspension Setting Register 1	0x00000000
R32_PFIC_IPSR2	0xE000E204	PFIC Interrupt Suspension Setting Register 2	0x00000000
R32_PFIC_IPSR3	0xE000E208	PFIC Interrupt Suspension Setting Register 3	0x00000000

R32_PFIC_IPSR4	0xE000E20C	PFIC Interrupt Suspension Setting Register 4	0x00000000
R32_PFIC_IPSR5	0xE000E210	PFIC Interrupt Suspension Setting Register 5	0x00000000
R32_PFIC_IPRR1	0xE000E280	PFIC interrupt pending clear register 1	0x00000000
R32_PFIC_IPRR2	0xE000E284	PFIC interrupt pending clear register 2	0x00000000
R32_PFIC_IPRR3	0xE000E288	PFIC interrupt suspend clear register 3	0x00000000
R32_PFIC_IPRR4	0xE000E28C	PFIC interrupt pending clear register 4	0x00000000
R32_PFIC_IPRR5	0xE000E290	PFIC interrupt pending clear register 5	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_IACTR3	0xE000E308	PFIC Interrupt Activation Status Register 3	0x00000000
R32_PFIC_IACTR4	0xE000E30C	PFIC Interrupt Activation Status Register 4	0x00000000
R32_PFIC_IACTR5	0xE000E310	PFIC Interrupt Activation Status Register 5	0x00000000
R32_PFIC_IPRIORx	0xE000E400	PFIC Interrupt Priority Configuration Register	0x00000000
R32_PFIC_IALLOCRx	0xE000E600	PFIC Interrupt Allocation Register	0xXXXXXXXX
R32_PFIC_IAUTR1	0xE000E700	PFIC Interrupt Handler Register 1	0xFFFFFFFF
R32_PFIC_IAUTR2	0xE000E704	PFIC Interrupt Handler Register 2	0xXXXXXXXX
R32_PFIC_IAUTR3	0xE000E708	PFIC Interrupt Handler Register 3	0xXXXXXXXX
R32_PFIC_IAUTR4	0xE000E70C	PFIC Interrupt Handler Register 4	0xXXXXXXXX
R32_PFIC_IAUTR5	0xE000E710	PFIC Interrupt Handler Register 5	0xXXXXXXXX
R32_PFIC_WAKEIP0	0xE000E720	PFIC Wake-up Instruction Pointer Register 0	0x00000001
R32_PFIC_WAKEIP1	0xE000E724	PFIC Wake-up Instruction Pointer Register 1	0x00000000
R32_PFIC_CSTAR0	0xE000E780	PFIC kernel status register 0	0x00000000
R32_PFIC_CSTAR1	0xE000E784	PFIC kernel status register 1	0x00000000
R32_PFIC_EENR	0xE000EC80	PFIC Event Enable Register	0xFFFFFFFF
R32_PFIC_EPR	0xE000EC84	PFIC Event Suspension Register	0x00000000
R32_PFIC_EWUPR	0xE000EC88	PFIC Event Wake-up Register	0x00000000
R32_PFIC_SCTLR	0xE000ED10	PFIC System Control Register	0x00000000

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

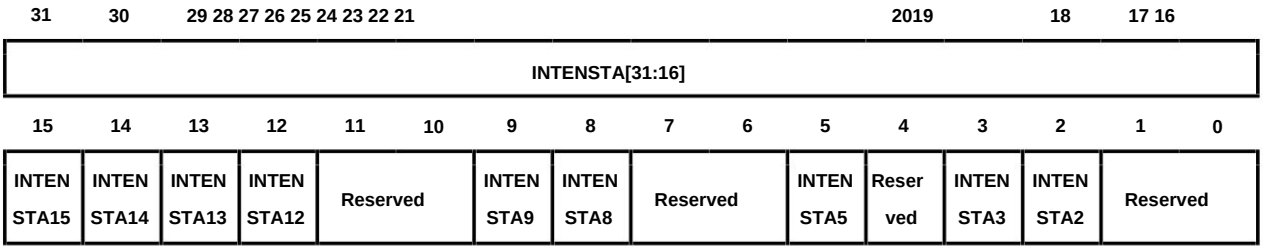
2. ECALL-M, ECALL-U, BREAKPOINT EXC EXC bit3 3. NMI EXC Support interrupt suspension clearing and setting This is a case where both are true, and the state is represented by the state bit.

operations, but do not support interrupt enable clearing and setting operations.

4. ECALL-M, ECALL-U, and BREAKPOINT do not support interruption suspension clearing and setting, or interruption enable clearing and setting operations.

4.7.5.1 PFIC Interrupt Enable Status Register 1 (PFIC_ISR1) Offset

Address: 0x00



Bit	name	access		Reset value
[31:12]	INTENSTA	RO	Description of the current enabled state of interrupts 12#-31#. 1: Interruption for the current number is enabled;	0

			0: The current number is interrupted and not enabled.	
[11:10] Reserved		RO is reserved.		0
[9:8]	INTENSTA	RO	8#-9# interrupt the current enabled state. 1: Interruption for the current number is enabled; 0: The current number is interrupted and not enabled.	0x3
[7:6]	Reserved	RO is reserved.		0
5	INTENSTA	RO	#5 interrupts the current enabled state. 1: Interruption for the current number is enabled; 0: The current number is interrupted and not enabled.	1
4	Reserved	RO is reserved.		0
[3:2]	INTENSTA	RO	2#-3# interrupt the current enabled state. 1: Interruption for the current number is enabled; 0: The current number is interrupted and not enabled.	0x3
[1:0]	Reserved	RO is reserved.		0

Note: This register is a kernel-private register, meaning that there are multiple sets of registers in the peripheral device that have the same function, independent data, and whose data cannot be accessed or modified by other kernels. Only the kernel has the same access address for it.

4.7.5.2 PFIC Interrupt Enable Status Register 2 (PFIC_ISR2) Offset

Address: 0x04

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
INTENSTA[63:48]														

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

Bit	name	access		Reset value
[31:0]	INTENSTA	RO	Description of the current enabled state of interrupts 32#-63#. 1: Interruption for the current number is enabled; 0: The current number is interrupted and not enabled.	0

Note: This register is a kernel-private register, meaning that there are multiple sets of registers in the peripheral device that have the same function, independent data, and whose data cannot be accessed or modified by other kernels. Only the kernel has the same access address for it.

4.7.5.3 PFIC Interrupt Enable Status Register 3 (PFIC_ISR3) Offset

Address: 0x08

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
INTENSTA[95:80]														

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

Bit	name	access		Reset value
[31:0]	INTENSTA	RO	Description of the current enabled state of interrupts 64#-95#. 1: Interruption for the current number is enabled; 0: The current number is interrupted and not enabled.	0

Note: This register is a kernel-private register, meaning that there are multiple sets of registers in the peripheral device that have the same function, independent data, and whose data cannot be accessed or modified by other kernels. Only the kernel has the same access address for it.

4.7.5.4 PFIC Interrupt Enable Status Register 4 (PFIC_ISR4) Offset

Address: 0x0C

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
INTENSTA[127:112]														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTENSTA[111:96]															

Bit	name	access		Reset value
[31:0]	INTENSTA	RO	Description of the current enabled state of interrupts 96#-127#. 1: Interruption for the current number is enabled; 0: The current number is interrupted and not enabled.	0

Note: This register is a kernel-private register, meaning that there are multiple sets of registers in the peripheral device that have the same function, independent data, and whose data cannot be accessed or modified by other kernels. Only the kernel has the same access address for it.

4.7.5.5 PFIC Interrupt Enable Status Register 5 (PFIC_ISR5) Offset

Address: 0x10

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved											INTENSTA[148:144]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTENSTA[143:128]															

Bitname		access	describe	Reset value
[31:21] Reserved		RO is reserved.		0
[20:0]	INTENSTA	RO	128#-148# interrupt the current enabled state. 1: Interruption for the current number is enabled; 0: The current number is interrupted and not enabled.	0

Note: This register is a kernel-private register, meaning that there are multiple sets of registers in the peripheral device that have the same function, independent data, and whose data cannot be accessed or modified by other kernels. Only the kernel has the same access address for it.

4.7.5.6 PFIC Interrupt Suspension Status Register 1 (PFIC_IPR1)

Offset Address: 0x20

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
PENDSTA [31:16]														

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

Bit	name	access	describe	Reset value

[31:12] PENDSTA	RO	12#-31# Interrupt the current suspended state. 1: The current number is interrupted and has been suspended; 0: The current number is interrupted and not suspended.	0
[11:10] Reserved	RO is reserved.		0
[9:8] PENDSTA	RO	8#-9# interrupt the current suspended state. 1: The current number is interrupted and has been suspended; 0: The current number is interrupted and not suspended.	0
[7:6] Reserved	RO is reserved. 5#		0
5 PENDSTA	RO	Interrupts the current suspended state. 1: The current number is interrupted and has been suspended; 0: The current number is interrupted and not suspended.	0
4 Reserved	RO is reserved.		0
[3:2] PENDSTA	RO	2#-3# interrupt the current suspended state. 1: The current number is interrupted and has been suspended; 0: The current number is interrupted and not suspended.	0
[1:0] Reserved	RO is reserved.		0

Note: This register is a kernel-private register, meaning that there are multiple sets of registers in the peripheral device that have the same function, independent data, and whose data cannot be accessed or modified by other kernels. Only the kernel has the same access address for it.

4.7.5.7 PFIC Interrupt Suspension Status Register 2 (PFIC_IPR2)

Offset Address: 0x24

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
PENDSTA [63:48]														

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

Bit	name	access	describe	Reset value
[31:0] PENDSTA		RO	32#-63# interrupt the current suspended state. 1: The current number is interrupted and has been suspended; 0: The current number is interrupted and not suspended.	0

4.7.5.8 PFIC Interrupt Suspension Status Register 3 (PFIC_IPR3)

Offset Address: 0x28

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
PENDSTA [95:80]														

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

Bit	name	access		Reset value
[31:0] PENDSTA		RO	Description of the current suspended state of interrupts 64#-95#. 1: The current number is interrupted and has been suspended; 0: The current number is interrupted and not suspended.	0

4.7.5.9 PFIC Interrupt Suspension Status Register 4 (PFIC_IPR4)

Offset Address: 0x2C

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
PENDSTA[127:112]														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PENDSTA [111:96]															

Bit	name	access		Reset value
[31:0]	PENDSTA	RO	Description of the current suspended state of interrupts 96#-127#. 1: The current number is interrupted and has been suspended; 0: The current number is interrupted and not suspended.	0

4.7.5.10 PFIC Interrupt Suspension Status Register 5 (PFIC_IPR5)

Offset Address: 0x30

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved											INTENSTA[148:144]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTENSTA[143:128]															

Bit	name	access	describe	Reset value
[31:21]	Reserved	RO Retention		0
[20:0]	PENDSTA	RO	128#-148# interrupt the current suspended state. 1: The current number is interrupted and has been suspended; 0: The current number is interrupted and not suspended.	0

4.7.5.11 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	describe	Reset value
[31:8]	Reserved	RO is reserved.		0
[7:0]	THRESHOLD	RW	Interrupt priority threshold setting value. Interrupts with a priority lower than the set threshold are suspended. Interrupt service is not executed when this register is 0; when this register is 0, it indicates that interrupt service is not executed. The threshold register function is invalid. [7:4]: Priority threshold; [3:0]: Reserved, fixed at 0.	0

Note: This register is a kernel-private register, meaning that there are multiple sets of registers in the peripheral device that have the same function, independent data, and whose data cannot be accessed or modified by other kernels. Only the kernel has the same access address for it.

4.7.5.12 PFIC Interrupt Configuration Register (PFIC_CFGR)

Offset Address: 0x48

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

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

Bit	name	access	The	Reset value
[31:16]	KEYCODE[15:0]	WO	description key controls the keyword; it is valid when writing to beef, otherwise the operation has no effect. effect.	0
[15:8]	Reserved	RO is reserved.		0
7	SYSRST	WO	The reset register is written to 0xBEEF0080 during system reset. The other values are invalid. Note: This has the same function as the SYSRST bit in the register. PFIC_SCTLR.	0
[6:0]	Reserved	RO is reserved.		0

Note: This register is a kernel-private register, meaning that there are multiple sets of registers in the peripheral device that have the same function, independent data, and whose data cannot be accessed or modified by other kernels. Only the kernel has the same access address for it.

4.7.5.13 PFIC Interrupt Global Status Register (PFIC_GISR)

Offset Address: 0x4C

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
Reserved					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EX_STATE [1:0]		LOCK _UP	DBG_ MODE	GLOB L_IE	Rese rved	GPEN DSTA	GACT STA	NESTSTA [7:0]							

Bit	name	access	describe	Reset value
[31:16]	Reserved	RO (Reserved).		0
[15:14]	EX_STATE[1:0]	RO	Kernel status register, used to obtain the kernel's running status. state 00: Running; 01: Light sleep; 10: Deep sleep; 11: Deep sleep (locked).	0
13	LOCK_UP	RO	The kernel lock status register is used to query whether the kernel is locked. It is in a locked state. 1: Lock; 0: Not locked.	0
12	DBG_MODE	RO	The kernel debug mode register is used to check if the kernel is in debug mode. In debug mode 1: The kernel is in debug mode;	0

			0: The kernel is in non-debug mode.	
11	GLOBL_IE	RO	The kernel global interrupt enable register is used to query the kernel. Is global interrupt enabled? 1: Enable kernel global interrupts; 0: Kernel global interrupt enable is disabled.	0
10	Reserved	RO is reserved.		0
9	GPENDSTA	RO	Is there any interruption currently pending? 1: Yes; 0: No.	0
8	GACTSTA	RO	Are there any interrupts being executed at present? 1: Yes; 0: No.	0
[7:0]	NESTSTA	RO	The current interrupt nesting state is used to query kernel interrupts. Nested state. Same as the INEST_CTLR register in the CSR register. The register nest_sta is a bit.	0

Note: This register is a kernel-private register, meaning that there are multiple sets of registers in the peripheral device that have the same function, independent data, and whose data cannot be accessed or modified by other kernels. Only the kernel has the same access address for it.

4.7.5.14 PFIC VTF Interrupt ID Configuration Register (PFIC_VTFIDR)

Offset address: 0x50

31	30 29 28 27 26	25 24 23	twenty two	twenty one	20	19	18	17	16
VTFID3					VTFID2				

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

Bit	name	access	describe	Reset value
[31:24]	VTFID3	RW	configures the interrupt number for VTF interrupt 3.	X
[23:16]	VTFID2	RW	configures the interrupt number for VTF interrupt 2.	X
[15:8]	VTFID1	RW	configures the interrupt number for VTF interrupt 1.	X
[7:0]	VTFID0	RW	configures the interrupt number for VTF interrupt 0.	X

Note: This register is a kernel-private register, meaning that there are multiple sets of registers in the peripheral device that have the same function, independent data, and whose data cannot be accessed or modified by other kernels. Only the kernel has the same access address for it.

4.7.5.15 PFIC VTF Interrupt 0 Address Register (PFIC_VTFADDRR0)

Offset address: 0x60

31	30	29 28 27 26 25 24 23 22 21	2019	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	describe	Reset value
[31:1]	ADDR0	RW	VTF Interrupt 0 Service Routine Address bits [31:1].	X
0	VTF0EN	RW	VTF Interrupt 0 Enable Bit: 1: Enable VTF interrupt channel 0;	0

			0: Off.	

Note: This register is a kernel-private register, meaning that there are multiple sets of registers in the peripheral device that have the same function, independent data, and whose data cannot be accessed or modified by other kernels. Only the kernel has the same access address for it.

4.7.5.16 PFIC VTF Interrupt 1 Address Register (PFIC_VTFADDRR1) Offset

Address: 0x64

31	30	29	28	27	26	25	24	23	22	21	2019	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

	name	access	describe	Reset value
Position [31:1]	ADDR1	RW	VTF Interrupt 1 Service Routine Address bit[31:1].	X
0	VTF1EN	RW	VTF Interrupt 1 Enable Bit: 1: Enable VTF interrupt channel 1; 0: Off.	0

Note: This register is a kernel-private register, meaning that there are multiple sets of registers in the peripheral device that have the same function, independent data, and whose data cannot be accessed or modified by other kernels. Only the kernel has the same access address for it.

4.7.5.17 PFIC VTF Interrupt 2 Address Register (PFIC_VTFADDRR2) Offset

Address: 0x68

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
ADDR2[31:16]														

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

Bit	name	access	describe	Reset value
[31:1]	ADDR2	RW	VTF Interrupt 2 Service Routine Address bit[31:1].	X
0	VTF2EN	RW	VTF Interrupt 2 Enable Bit: 1: Enable VTF interrupt channel 2; 0: Off.	0

Note: This register is a kernel-private register, meaning that there are multiple sets of registers in the peripheral device that have the same function, independent data, and whose data cannot be accessed or modified by other kernels. Only the kernel has the same access address for it.

4.7.5.18 PFIC VTF Interrupt 3 Address Register (PFIC_VTFADDRR3) Offset

Address: 0x6C

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
ADDR3[31:16]														

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

Bit	name	access	describe	Reset value
[31:1]	ADDR3	RW	VTF interrupt 3 service routine address bits[31:1].	X
0	VTF3EN	RW	VTF Interrupt 3 Enable Bit: 1: Enable VTF interrupt channel 3; 0: Off.	0

Note: This register is a kernel-private register, meaning that there are multiple sets of registers in the peripheral device that have the same function, independent data, and whose data cannot be accessed or modified by other kernels. Only the kernel has the same access address for it.

4.7.5.19 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
INTEN15	INTEN14	INTEN13	INTEN12	Reserved

Bit	name	access	Describe	Reset value
[31:12]	INTEN	WO	the interrupt enable control for #12-#31. 1: Interruption enabled for the current number; 0: No effect.	0
[11:0] Reserved		RO	is reserved.	0

Note: This register is a kernel-private register, meaning that there are multiple sets of registers in the peripheral device that have the same function, independent data, and whose data cannot be accessed or modified by other kernels. Only the kernel has the same access address for it.

4.7.5.20 PFIC interrupt enable setting register 2 (PFIC_IENR2) offset

address: 0x104

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
INTEN[63:48]					

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

Bit	name	access		Reset value
[31:0]	INTEN	WO	Describe the interrupt enable control for 32#-63#. 1: Interruption enabled for the current number; 0: No effect.	0

Note: This register is a kernel-private register, meaning that there are multiple sets of registers in the peripheral device that have the same function, independent data, and whose data cannot be accessed or modified by other kernels. Only the kernel has the same access address for it.

4.7.5.21 PFIC interrupt enable setting register 3 (PFIC_IENR3) offset

address: 0x108

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
INTEN[95:80]					

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

Bit	name	access		Reset value
[31:0]	INTEN	WO	Describe the interrupt enable control for 64#-95#. 1: Interruption enabled for the current number; 0: No effect.	0

Note: This register is a kernel-private register, meaning that there are multiple sets of registers in the peripheral device that have the same function, independent data, and whose data cannot be accessed or modified by other kernels. Only the kernel has the same access address for it.

4.7.5.22 PFIC interrupt enable setting register 4 (PFIC_IENR4) offset

address: 0x10C

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
INTEN[127:112]														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTEN[111:96]															

Bit	name	access		Reset value
[31:0]	INTEN	WO	Describe the interrupt enable control for 96#-127#. 1: Interruption enabled for the current number; 0: No effect.	0

Note: This register is a kernel-private register, meaning that there are multiple sets of registers in the peripheral device that have the same function, independent data, and whose data cannot be accessed or modified by other kernels. Only the kernel has the same access address for it.

4.7.5.23 PFIC interrupt enable setting register 5 (PFIC_IENR5) offset

address: 0x110

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved											INTEN[148:144]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTEN[143:128]															

Bit	name	access	describe	Reset value
[31:21] Reserved		RO is reserved.		0
[20:0]	INTEN	WO	Interrupt enable control for 128#-148#. 1: Interruption enabled for the current number; 0: No effect.	0

Note: This register is a kernel-private register, meaning that there are multiple sets of registers in the peripheral device that have the same function, independent data, and whose data cannot be accessed or modified by other kernels. Only the kernel has the same access address for it.

4.7.5.24 PFIC interrupt enable clear register 1 (PFIC_IRER1) offset

address: 0x180

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTRSET15	INTRSET14	INTRSET13	INTRSET12	Reserved											

Bit	name	access		Reset value
[31:12]	INTRSET	WO	Description of interrupt shutdown control for 12#-31#. 1: The current number is interrupted and closed; 0: No effect.	0
[11:0]	Reserved	RO	is reserved.	0

Note: This register is a kernel-private register, meaning that there are multiple sets of registers in the peripheral device that have the same function, independent data, and whose data cannot be accessed or modified by other kernels. Only the kernel has the same access address for it.

4.7.5.25 PFIC interrupt enable clear register 2 (PFIC_IRER2) offset

address: 0x184

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
INTRSET[63:48]					

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

Bit	name	access	describe	Reset value
[31:0]	INTRSET	WO	Interrupt shutdown control for 32#-63#. 1: The current number is interrupted and closed; 0: No effect.	0

Note: This register is a kernel-private register, meaning that there are multiple sets of registers in the peripheral device that have the same function, independent data, and whose data cannot be accessed or modified by other kernels. Only the kernel has the same access address for it.

4.7.5.26 PFIC interrupt enable clear register 3 (PFIC_IRER3) offset

address: 0x188

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
INTRSET[95:80]					

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

Bit	name	access		Reset value
[31:0]	INTRSET	WO	Description of interrupt shutdown control for 64#-95#. 1: The current number is interrupted and closed; 0: No effect.	0

Note: This register is a kernel-private register, meaning that there are multiple sets of registers in the peripheral device that have the same function, independent data, and whose data cannot be accessed or modified by other kernels. Only the kernel has the same access address for it.

4.7.5.27 PFIC interrupt enable clear register 4 (PFIC_IRER4) offset address: 0x18C

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
INTRSET[127:112]														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTRSET[111:96]															

Bit	name	access		Reset value
[31:0]	INTRSET	WO	Description of interrupt shutdown control for 96#-127#. 1: The current number is interrupted and closed; 0: No effect.	0

Note: This register is a kernel-private register, meaning that there are multiple sets of registers in the peripheral device that have the same function, independent data, and whose data cannot be accessed or modified by other kernels. Only the kernel has the same access address for it.

4.7.5.28 PFIC interrupt enable clear register 5 (PFIC_IRER5) offset address: 0x190

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved											INTRSET[148:144]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTRSET[143:128]															

Bitname		access	describe	Reset value
[31:21] Reserved		RO is reserved.		0
[20:0]	INTRSET	WO	Interrupts 128#-148# disable control. 1: The current number is interrupted and closed; 0: No effect.	0

Note: This register is a kernel-private register, meaning that there are multiple sets of registers in the peripheral device that have the same function, independent data, and whose data cannot be accessed or modified by other kernels. Only the kernel has the same access address for it.

4.7.5.29 PFIC Interrupt Suspension Setting Register 1 (PFIC_IPSR1) Offset Address: 0x200

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PENDSET15	PENDSET14	PENDSET13	PENDSET12	Reserved		PENDSET9	PENDSET8	Reserved		PENDSET5	Reserved	PENDSET3	PENDSET2	Reserved	

Bit	name	access	The	Reset value
[31:12] PENDSET		WO	interrupt suspension settings for 12#-31# are described, while 13# and 15# are reserved. 1: The current number is interrupted and suspended;	0

			0: No effect.	
[11:10] Reserved		RO reserved.		0
[9:8]	PENDSET	RO	8#-9# Interrupt suspension settings. 1: The current number is interrupted and suspended; 0: No effect.	0
[7:6]	Reserved	RO reserved. 5#		0
5	PENDSET	RO	Interrupt suspension setting. 1: The current number is interrupted and suspended; 0: No effect.	0
4	Reserved	RO is reserved.		0
[3:2]	PENDSET	WO	2#-3# interrupt suspension settings. 1: The current number is interrupted and suspended; 0: No effect.	0
[1:0]	Reserved	RO is reserved.		0

Note: This register is a kernel-private register, meaning that there are multiple sets of registers in the peripheral device that have the same function, independent data, and whose data cannot be accessed or modified by other kernels. Only the kernel has the same access address for it.

4.7.5.30 PFIC Interrupt Suspension Setting Register 2 (PFIC_IPSR2)

Offset Address: 0x204

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
PENDSET[63:48]														

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

Bit	name	access	Describe	Reset value
[31:0]	PENDSET	WO	the interrupt suspension settings for 32#-63#. 1: The current number is interrupted and suspended; 0: No effect.	0

4.7.5.31 PFIC Interrupt Suspension Setting Register 3 (PFIC_IPSR3)

Offset Address: 0x208

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
PENDSET[95:80]														

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

Bit	name	access	describe	Reset value
[31:0]	PENDSET	WO	64#-95# Interrupt suspension settings. 1: The current number is interrupted and suspended; 0: No effect.	0

4.7.5.32 PFIC Interrupt Suspension Setting Register 4 (PFIC_IPSR4)

Offset address: 0x20C

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
PENDSET[127:112]														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PENDSET[111:96]															

Bit	name	access	Describe	Reset value
[31:0]	PENDSET	WO	the interrupt suspension settings for 96#-127#. 1: The current number is interrupted and suspended; 0: No effect.	0

4.7.5.33 PFIC Interrupt Suspension Setting Register 5 (PFIC_IPSR5)

Offset Address: 0x210

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved											PENDSET[148:144]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PENDSET[143:128]															

Bit	name	access	describe	Reset value
[31:21]	Reserved	RO	is reserved.	0
[20:0]	PENDSET	WO	Interrupt suspension settings for 128#-148#. 1: The current number is interrupted and suspended; 0: No effect.	0

4.7.5.34 PFIC Interrupt Suspension Clear Register 1 (PFIC_IPRR1)

Offset Address: 0x280

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

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

Bit	name	access		Reset value
[31:12]	PENDRST	WO	Description: 12#-31# interrupted and suspended, cleared; 13# and 15# retained. 1: The current number is interrupted and the suspended status is cleared; 0: No effect.	0
[11:10]	Reserved	RO	is reserved. 8#-9#	0
[9:8]	PENDRST	RO	interrupted and suspended, then cleared. 1: The current number is interrupted and the suspended status is cleared; 0: No effect.	0

[7:6]	Reserved	RO is reserved.		0
5	PENDRST	RO	5# Interruption suspension cleared. 1: The current number is interrupted and the suspended status is cleared; 0: No effect.	0
4	Reserved	RO is reserved. 2#-3#		0
[3:2]	PENDRST	WO	interrupted and suspended, then cleared. 1: The current number is interrupted and the suspended status is cleared; 0: No effect.	0
[1:0]	Reserved	RO is reserved.		0

Note: This register is a kernel-private register, meaning that there are multiple sets of registers in the peripheral device that have the same function, independent data, and whose data cannot be accessed or modified by other kernels. Only the kernel has the same access address for it.

4.7.5.35 PFIC Interrupt Suspension Clear Register 2 (PFIC_IPRR2)

Offset Address: 0x284

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
PENDRST[63:48]														

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

Bit	name	access		Reset value
[31:0]	PENDRST	WO	Description of interruption suspension clearing for 32#-63#. 1: The current number is interrupted and the suspended status is cleared; 0: No effect.	0

4.7.5.36 PFIC Interrupt Suspension Clear Register 3 (PFIC_IPRR3)

Offset Address: 0x288

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
PENDRST[95:80]														

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

Bit	name	access	Description	Reset value
[31:0]	PENDRST	WO	of interruption suspension and clearing for 64#-95#. 1: The current number is interrupted and the suspended status is cleared; 0: No effect.	0

4.7.5.37 PFIC Interrupt Suspension Clear Register 4 (PFIC_IPRR4)

Offset Address: 0x28C

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
PENDRST[127:112]														

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

PENDRST[111:96]

Bit	name	access	describe	Reset value
[31:0] PENDSET		WO	96#-127# Interruption suspension cleared. 1: The current number is interrupted and the suspended status is cleared; 0: No effect.	0

4.7.5.38 PFIC Interrupt Suspension Clear Register 5 (PFIC_IPRR5)

Offset Address: 0x290

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
Reserved			PENDRST[148:144]		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PENDRST[143:128]															

Bitname		access	describe	Reset value
[31:21] Reserved		RO is reserved.		0
[20:0] PENDSET		WO	128#-148# interrupted and suspended, then cleared. 1: The current number is interrupted and the suspended status is cleared; 0: No effect.	0

4.7.5.39 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
IACTS15	IACTS14	IACTS13	IACTS12	Reserved								IACTS3	IACTS2	Reserved	

Bit	name	access	Describe	Reset value
[31:12]	IACTS	RO	the execution status of interrupts 12#-31#, while 13# and 15# are reserved. 1: The current number is currently being interrupted during execution; 0: The current number was interrupted and not executed.	0
[11:4] Reserved		RO is reserved.		0
[3:2]	IACTS	RO	2#-3# are in an interrupted execution state. 1: The current number is currently being interrupted during execution; 0: The current number was interrupted and not executed.	0
[1:0]	Reserved	RO is reserved.		0

Note: This register is a kernel-private register, meaning that there are multiple sets of registers in the peripheral device that have the same function, independent data, and whose data cannot be accessed or modified by other kernels. Only the kernel has the same access address for it.

4.7.5.40 PFIC Interrupt Activation Status Register 2 (PFIC_IACTR2)

Offset Address: 0x304

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
IACTS[63:48]														

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

Bit	name	access	Describe	Reset value
[31:0]	IACTS	RO	the execution status of interrupts 32#-63#. 1: The current number is currently being interrupted during execution; 0: The current number was interrupted and not executed.	0

4.7.5.41 PFIC Interrupt Activation Status Register 3 (PFIC_IACTR3)

Offset Address: 0x308

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
IACTS[95:80]														

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

Bit	name	access	describe	Reset value
[31:0]	IACTS	RO	64#-95# are in an interrupted execution state. 1: The current number is currently being interrupted during execution; 0: The current number was interrupted and not executed.	0

4.7.5.42 PFIC Interrupt Activation Status Register 4 (PFIC_IACTR4)

Offset Address: 0x30C

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
IACTS[127:112]														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IACTS[111:96]															

Bit	name	access	Describe	Reset value
[31:0]	IACTS	WO	the execution status of interrupts 96#-127#. 1: The current number is currently being interrupted during execution; 0: The current number was interrupted and not executed.	0

4.7.5.43 PFIC Interrupt Activation Status Register 5 (PFIC_IACTR5)

Offset Address: 0x310

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved												IACTS[148:144]		

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

IACTS[143:128]

Bit	name	access	describe	Reset value
[31:21] Reserved		RO is reserved.		0
[20:0]	IACTS	WO	128#-148# are in an interrupted execution state. 1: The current number is currently being interrupted during execution; 0: The current number was interrupted and not executed.	0

4.7.5.44 PFIC Interrupt Priority Configuration Register (PFIC_IPRIORx) (x=0-63) Offset address:

0x400–0x4FF The controller

supports 256 interrupts (0-255), and each interrupt uses 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_ÿ4x+3ÿ	PRIO_ÿ4x+2ÿ	PRIO_ÿ4x+1ÿ	PRIO_ÿ4xÿ
…	…	…	…	…
IPRIOR0	PRIO_3	PRIO_2	PRIO_1	PRIO_0

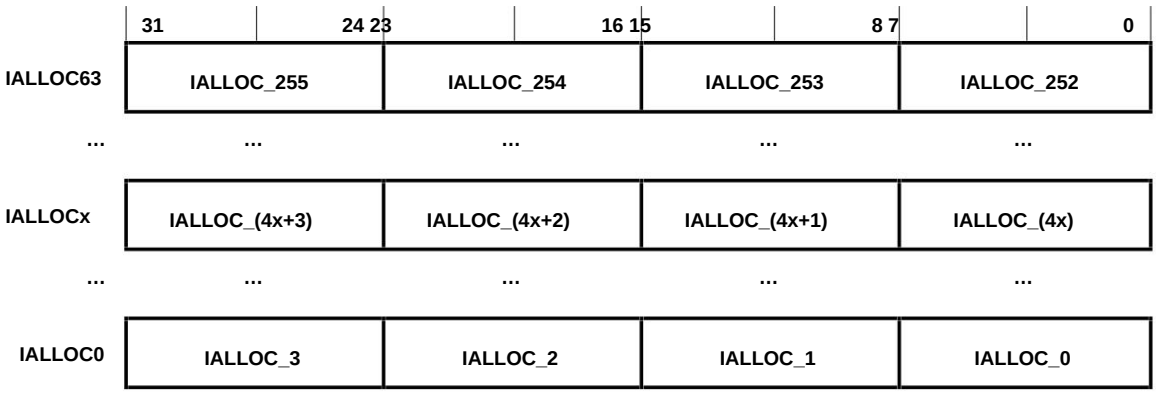
Bit	name	access	describe	Reset value
[2047:2040] IP_255		RW is the same as IP_0.		0
…	…	… …		…
[31:24]	IP_3	RW is the same as IP_0.		0
[23:16]	IP_2	RW is the same as IP_0.		0
[15:8]	IP_1	RW is the same as IP_0.		0
[7:0]	IP_0	RW	Interrupt priority configuration for number 0: [7:4]: Priority control bit; the smaller the value, the higher the priority. The higher the level, the more the preemption bit is controlled by the CSR register. preempt (control). When the preempt register value is 0, there is no preemption. When the preemept register value is 1, bit 7 is preemptive. Bit. When the preempt register value is 2, bits 7-6 are preemptive. Placeholder. When the preempt register value is 3, bits 7-5 are preemptive. Placeholder. The V3F kernel supports a maximum of 2 levels of nesting. When the nesting is full... Unable to continue the assault. The V5F kernel supports a maximum of 8 levels of nesting. It can be configured, but cannot preempt when the nesting is full. The [3:0] bits are fixed at 0.	0

Note: This register is a kernel-private register, meaning that multiple sets of peripherals may exist with the same function, but their data is independent and cannot be shared with other kernels.

Registers accessed or modified by the kernel are only accessible at the same address as the kernel.

4.7.5.45 PFIC Interrupt Allocation Register (PFIC_IALLOCRx) (x=0-63) Offset

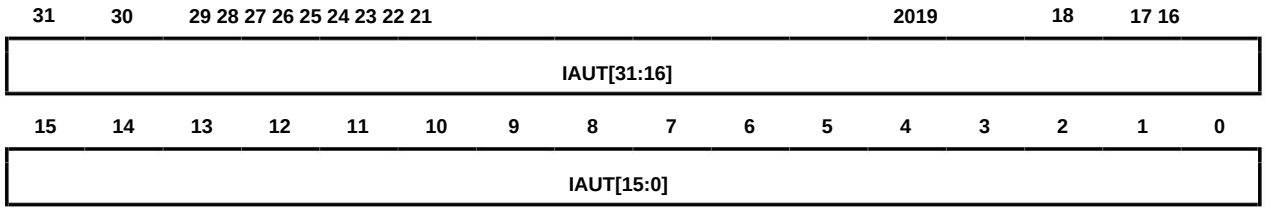
Address: 0x600-0x6FF



Bit	name	access	describe	Reset value
[2047:2040]	IALLOC_255	RW	is the same as IALLOC_32.	X
...	...	...	...	...
[263:256]	IALLOC_33	RW	is described as IALLOC_32.	X
[255:248]	IALLOC_32	RW	Interrupt allocation register number 32 exists in this system. Two kernels, only bit 0 is configurable; when this interrupt... When the ENA is enabled by a kernel, this register automatically... The kernel ID is dynamically updated to operate on the ENA register. value. 1: This interrupt is assigned to C1 for processing; 0: This interrupt is assigned to C0 for processing.	X
[247:240]	IALLOC_31	RO	is described as IALLOC_0.	X
...	...	...	...	...
[31:24]	IALLOC_3	RO	is the same as IALLOC_0.	X
[23:16]	IALLOC_2	RO	is described as IALLOC_0.	X
[15:8]	IALLOC_1	RO	is described as IALLOC_0.	X
[7:0]	IALLOC_0	RO	Interrupt allocation number 0: This system has two kernels, but only the 0th kernel is configurable. The C0 kernel read value is 0, and the C1 kernel read value is 1. 1: This interrupt is assigned to C1 for processing; 0: This interrupt is assigned to C0 for processing.	X

4.7.5.46 PFIC Interrupt Handler Register 1 (PFIC_IAUTR1) Offset

Address: 0x700



Bit	name	access	describe	Reset value
[31:0]	IAUT	RO	Interrupt 31#-0# allocates all registers for polling. Whether an interrupt is assigned to a response from this kernel. 1: Interrupts are assigned to this kernel; 0: Interrupts have not been assigned to this kernel.	0xFFFFFFFF

Note: This register is a kernel-private register, meaning that there are multiple sets of registers in the peripheral device that have the same function, independent data, and whose data cannot be accessed or modified by other kernels. Only the kernel has the same access address for it.

4.7.5.47 PFIC Interrupt Handler Register 2 (PFIC_IAUTR2) Offset

Address: 0x704

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
IAUT[63:48]														

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

Bit	name	access	Describe	Reset value
[31:0]	IAUT	RO	the allocation of all registers for interrupts 32#-63#, used for checking... Check if an interrupt is assigned to a response from this kernel. 1: Interrupts are assigned to this kernel; 0: Interrupts have not been assigned to this kernel.	X

Note: This register is a kernel-private register, meaning that there are multiple sets of registers in the peripheral device that have the same function, independent data, and whose data cannot be accessed or modified by other kernels. Only the kernel has the same access address for it.

4.7.5.48 PFIC Interrupt Handler Register 3 (PFIC_IAUTR3) Offset

Address: 0x708

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
IAUT[95:80]														

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

Bit	name	access		Reset value
[31:0]	IAUT	RO	Description of interrupts 64#-95#: All registers are allocated for querying. Check if an interrupt is assigned to a response from this kernel. 1: Interrupts are assigned to this kernel; 0: Interrupts have not been assigned to this kernel.	X

Note: This register is a kernel-private register, meaning that there are multiple sets of registers in the peripheral device that have the same function, independent data, and whose data cannot be accessed or modified by other kernels. Only the kernel has the same access address for it.

4.7.5.49 PFIC Interrupt Handler Register 4 (PFIC_IAUTR4) Offset

Address: 0x70C

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16

IAUT[127:112]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IAUT[111:96]															

Bit	name	access	The	Reset value
[31:0]	IAUT	RO	description describes the allocation of all registers for interrupts 96#-127#, used for querying. Check if an interrupt is assigned to a response from this kernel. 1: Interrupts are assigned to this kernel; 0: Interrupts have not been assigned to this kernel.	X

Note: This register is a kernel-private register, meaning that there are multiple sets of registers in the peripheral device that have the same function, independent data, and whose data cannot be accessed or modified by other kernels. Only the kernel has the same access address for it.

4.7.5.50 PFIC Interrupt Handler Register 5 (PFIC_IAUTR5) Offset

Address: 0x710

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
Reserved			IAUT[148:144]		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IAUT[143:128]															

Bit	name	access	describe	Reset value
[31:21] Reserved		RO is reserved.		0
[20:0]	IAUT	RO	All registers are allocated for interrupts 128#-148# for use in... Check if an interrupt is assigned to a response from this kernel. 1: Interrupts are assigned to this kernel; 0: Interrupts have not been assigned to this kernel.	X

Note: This register is a kernel-private register, meaning that there are multiple sets of registers in the peripheral device that have the same function, independent data, and whose data cannot be accessed or modified by other kernels. Only the kernel has the same access address for it.

4.7.5.51 PFIC Wake-up Instruction Pointer Register 0 (PFIC_WAKEIP0) Offset

Address: 0x720

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
IP_RELOAD0[30:15]					

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

Bit	name	access		Reset value
[31:1]	IP_RELOAD0	RW	Describes the PC address register during kernel C0 wake-up, used for... Reloading of PC value after electrical wake-up and reset wake-up.	0
0	SHUTDOWN0	RW	Kernel C0 deep sleep (lock) cancel register, using The internal system enters a deep sleep (locked) state after power-on.	1

			The core is unlocked, allowing it to be woken up by time: 1: Lock; 0: Unlock.	

4.7.5.52 PFIC Wake-up Instruction Pointer Register 1 (PFIC_WAKEIP1)

Offset Address: 0x724

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
IP_RELOAD1[30:15]														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IP_RELOAD1[14:0]															SHUTDOWN1

Bit	name	access	describe	Reset value
[31:1]	IP_RELOAD1[30:0]	RW	Kernel C1 wake-up PC address register, used for power-on Wake-up and reset: Reload PC value after wake-up.	0
0	SHUTDOWN1	RW	Kernel C1 sleep (lock) cancel register, used for The kernel enters a deep sleep (locked) state after power-on. Cancel the lock and allow wake-up by time: 1: Lock; 0: Unlock.	0

4.7.5.53 PFIC kernel status register 0 (PFIC_CSTAR0) offset

address: 0x780

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

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

Bit	name	access	describe	Reset value
[31:16] Reserved		RO is reserved. The		0
[15:14] CPU_ex_state_0		RO	kernel C0 status register is used to obtain kernel C0 status information. Operating status 00: Running; 01: Light sleep; 10: Deep sleep; 11: Deep sleep (locked).	0
13	CPU_lock_up_0	RO	The kernel C0 lock status register is used to query the internal... Is core C0 in a locked state? 1: Lock; 0: Not locked.	0

12	CPU_dbg_mode_0	RO	The kernel C0 debug mode register is used to query the kernel. Is core C0 in debug mode? 1: The kernel is in debug mode; 0: The kernel is in non-debug mode.	0
11	CPU_globl_ie_0	RO	The kernel C0 global interrupt enable register is used to check... Inquire whether global interrupts are enabled in kernel C0: 1: Enable kernel global interrupts; 0: Kernel global interrupt enable is disabled.	0
10	Reserved	RO is reserved.		0
9	CPU_irq_pend_0	RO	The kernel C0 interrupt pending flag register is used to check... Check kernel C0 for any unhandled interrupts: 1. There are unhandled interrupts in the kernel; 0: There are no unhandled interrupts in the kernel.	0
8	CPU_irq_active_0	RO	The kernel C0 interrupt active flag register is used to check... Query kernel C0 to see if it is handling an interrupt. 1: The kernel is processing an interrupt; 0: The kernel did not handle the interrupt.	0
[7:0]	CPU_nest_sta_0	RO	The kernel C0 interrupt nesting status register is used for querying... Nested state of kernel C0 interrupts. Same as CSR register. The nest_sta bit of the INEST_CTLR register.	0

4.7.5.54 PFIC kernel status register 1 (PFIC_CSTAR1) offset address:

0x784

31 30	29	28 27 26 25	24 23 22 21	2019	18	17 16
Reserved						

15 14 13	12	11	10	9	8	7 6 5 4 3 2 1 0
CPU_ex_state_1	CPU_lock_up_1	CPU_dbg_mode_1	CPU_globl_ie_1	Reserved	CPU_irq_pend_1	CPU_irq_active_1	CPU_nest_sta_1

Bitname		access	describe	Reset value
[31:16] Reserved		RO Reserved.		0
[15:14] CPU_ex_state_1		RO	Kernel C1 Status Register, used to obtain kernel C1 status information. Operating status: 00: Running; 01: Light sleep; 10: Deep sleep; 11: Deep sleep (locked).	0
13	CPU_lock_up_1	RO	Kernel C1 Lock Status Register, used to query internal... Is core C1 in a locked state? 1: Lock; 0: Not locked.	0
12	CPU_dbg_mode_1	RO	The kernel C1 debug mode register is used to query the internal... Is core C1 in debug mode?	0

			1: The kernel is in debug mode; 0: The kernel is in non-debug mode.	
11	CPU_globl_ie_1	RO	The kernel C1 global interrupt enable register is used to check... Inquire whether global interrupts in kernel C1 are enabled: 1: Enable kernel global interrupts; 0: Kernel global interrupt enable is disabled.	0
10	Reserved	RO is reserved.		0
9	CPU_irq_pend_1	RO	The kernel C1 interrupt pending flag register is used to check... Check kernel C1 for any unhandled interrupts: 1. There are unhandled interrupts in the kernel; 0: There are no unhandled interrupts in the kernel.	0
8	CPU_irq_active_1	RO	The kernel C1 interrupt active flag register is used to check... Query kernel C1 to see if it is handling an interrupt. 1: The kernel is processing an interrupt; 0: The kernel did not handle the interrupt.	0
[7:0]	CPU_nest_sta_1	RO	The kernel C1 interrupt nesting status register is used for querying... Nested state of kernel C1 interrupts. Same as CSR register. The nest_sta bit of the INEST_CTLR register.	0

4.7.5.55 PFIC Event Enable Register (PFIC_EENR)

Offset address: 0xC80

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
EVENTEN [31:16]														

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

Bit	name	access	describe	Reset value
[31:0] EVENTEN		RW	Event wake-up enable (31#-0#): 1: Allow the corresponding event to wake the kernel from sleep; 0: Blocks the corresponding event; this event is not allowed to occur. Wake up the kernel. Supports up to 32 event sources, where 0-7 represent electrical levels. Trigger (active high), 8-31 are pulse triggers: 0#: The kernel pauses requests; 1#: Kernel interrupt request; 7#-2#: Reserved; #8: Kernel enters interrupt; 9#: Kernel exit interrupt; 10#: New interrupt pending; (For shared interrupts, (Regardless of whether the interrupt is routed to the local kernel) #11: Kernel send event; (local kernel send) event) #12: Kernel receive event; (send events from other kernels)	0xFFFFFFFF

			event) 13#: System received event; (System configured event, For example, GPIO toggle events...) 31#-14#: Reserved.	

Note: This register is a kernel-private register, meaning that there are multiple sets of registers in the peripheral device that have the same function, independent data, and whose data cannot be accessed or modified by other kernels. Only the kernel has the same access address for it.

4.7.5.56 PFIC Event Suspension Register (PFIC_EPR) Offset

Address: 0xC84

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
EVENT_PEND[31:16]														

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

Bit	name	access	describe	Reset value
[31:8]	EVENT_PEND	RW1Z	Events 31#-8# are in a suspended state; writing 1 will clear the status. 1: The corresponding event occurred but was not handled; 0: The corresponding event did not occur.	0
[7:0]	EVENT_PEND	RO	Events 7#-0# are in a suspended state and cannot be cleared: 1: The corresponding event occurred but was not handled; 0: The corresponding event did not occur.	0

Note: This register is a kernel-private register, meaning that there are multiple sets of registers in the peripheral device that have the same function, independent data, and whose data cannot be accessed or modified by other kernels. Only the kernel has the same access address for it.

4.7.5.57 PFIC Event Wake-up Register (PFIC_EWUPR) Offset

Address: 0xC88

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
EVENT_WUP[31:16]														

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

Bit	name	access	describe	Reset value
[31:0]	EVENT_WUP	RO	Event wake-up register 31#-0#: 1: The corresponding event will wake the kernel from sleep; 0: The corresponding event is not the event that wakes up the kernel this time.	0

Note: This register is a kernel-private register, meaning that there are multiple sets of registers in the peripheral device that have the same function, independent data, and whose data cannot be accessed or modified by other kernels. Only the kernel has the same access address for it.

4.7.5.58 PFIC System Control Register (PFIC_SCTLR) Offset

Address: 0xD10

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16

SYSRST	Reserved	HART_ID

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

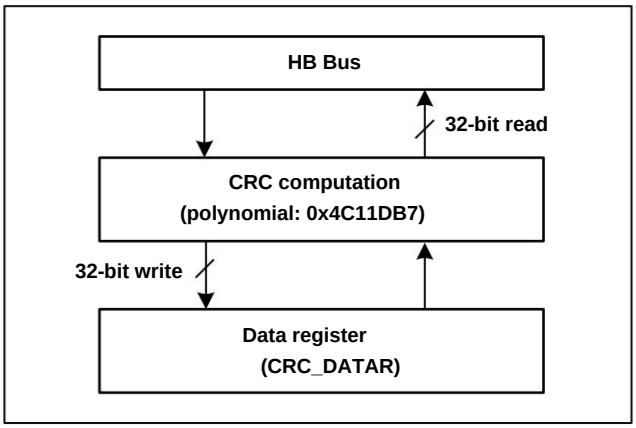
Bit	name	access	This	Reset value
31	SYSRST	RW1	describes a system reset that automatically clears the cache to 0. Writing 1 indicates validity, writing 0 indicates no validity. The effect is the same as that of the PFIC_CFGR register.	0
[30:24] Reserved		RO is reserved.		0
[23:16] HART_ID		RO	The kernel ID register is used to retrieve the read register. Kernel ID: 1: This kernel is C1; 0: This kernel is C0.	0
[15:6] Reserved		RO is reserved. A		0
5	SENDEVENT	WO	write event is generated, which can be used to wake up the system. Other kernels.	0
4	SEVONPEND	RW	When an event or interrupt occurs and the system is in a suspended state, it can... Wake up the system from the WFE command if WFE is not executed. The instruction will wake up immediately after the instruction is executed again. system. 1: Enabled events and all interrupts (including those not enabled) Interrupts can wake up the system. 0: Only enabled events and enabled interrupts can be invoked. Awake.	0
3	WFITOWFE	RW	The WFI command is executed as WFE, where WFE is a custom command. After this register is set to 1, the kernel executes the WFI instruction. After the command is given, it enters a low-power state and waits for external events to occur. The kernel can be awakened after birth. 1: The WFI command waits for an event to wake up the kernel; 0: The WFI instruction waits for an interrupt to wake up the kernel.	0
2	SLEEPDEEP	RW	Low-power mode of the control system: 1: deepsleep 0: sleep Controls the system	0
1	SLEEPONEXIT	RW	state after leaving the interrupt service routine: 1: The system enters low-power mode; 0: The system enters the main program.	0
0	Reserved	RO is reserved.		0

Note: This register is a kernel-private register, meaning that there are multiple sets of registers in the peripheral device that have the same function, independent data, and whose data cannot be accessed or modified by other kernels. Only the kernel has the same access address for it.

Chapter 5 Cyclic Redundancy Check (CRC)

The Cyclic Redundancy Check (CRC) calculation unit calculates the CRC result for any 32-bit data based on a fixed generator polynomial.
Generally used in data storage and data communication to verify data integrity. The system provides a hardware CRC calculation unit, which can greatly save costs.
Improve the efficiency of CPU and RAM resources.

Figure 5-1 CRC Structure Diagram



5.1 Main Features

Using CRC32 polynomial (0x4C11DB7): X 32+X26+X23+X22+X16+X12+X11+X10+X8 +X7 +X5 +X4 +X2 +X+1;
The same 32-bit register is used for both data input and CRC32 calculation output.
ÿSingle conversion time: 4 HB clock cycles (HCLK)

5.2 Functional Description

ÿ CRC cell reset

To begin a CRC calculation for a new data group, the CRC calculation unit needs to be reset. This is done by setting the RST value in the control register CRC_CTLR.
Writing a 1 to the bit will reset the data register, restoring it to its initial value of 0xFFFFFFFF.

ÿ CRC calculation

The CRC unit is calculated by combining the result of the previous CRC calculation with the CRC result of the newly added data. The CRC_DATAR data register...
A write operation will send new data to the hardware computing unit; a read operation will obtain the latest CRC calculation value.
The calculation will interrupt the system's write operation, so new values can be written continuously.
Note: The CRC unit is calculated for the entire 32 bit data, not byte by byte.

Independent data buffer

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

5.3 Register Description

Table 5-1 List of CRC-related registers

name	Access address	describe	Reset value
R32_CRC_DATAR	0x40023000 Data Register; 0x40023004		0xFFFFFFFF
R8_CRC_IDATAR	Independent Data Buffer		0x00

R32_CRC_CTLR	0x40023008 Control Register	0x00000000

5.3.1 Data Register (CRC_DATAR) Offset

Address: 0x00

31	30 29	28	27 26 25 24 23 22 21	2019	18	17 16
DR[31:16]						

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

Bit	name	Access description RW: Writes raw data;	Reset value
[31:0]	DR[31:0]	reads calculation results.	0xFFFFFFFF

5.3.2 Independent data buffer (CRC_IDATAR)

offset address: 0x04

Bit	name	access	describe	Reset value
[7:0]	IDR[7:0]	RW	An 8-bit general-purpose register, which can be used as a data buffer, this register... The register is not affected by the RST field of the control register.	0

5.3.3 Control Register (CRC_CTLR) Offset

Address: 0x08

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

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

Bitname		access	describe	Reset value
[31:1]	Reserved	RO	is reserved.	0
0	RST	WO	CRC calculation unit reset control, write 1 to execute, hardware automatic. After clearing to zero and completing the process, the data register will be 0xFFFFFFFF.	0

Chapter 6 Real-Time Clock (RTC)

The Real-Time Clock (RTC) is a standalone timer module with a programmable counter up to 32 bits, which can be used in conjunction with software.

It implements a real-time clock function and allows modification of the counter value to reconfigure the system's current time and date. The RTC module is available in backup mode.

The electrical region is unaffected by system reset.

6.1 Main Features

ÿ The maximum prescaler coefficient is 220.

ÿ 32-bit programmable counter

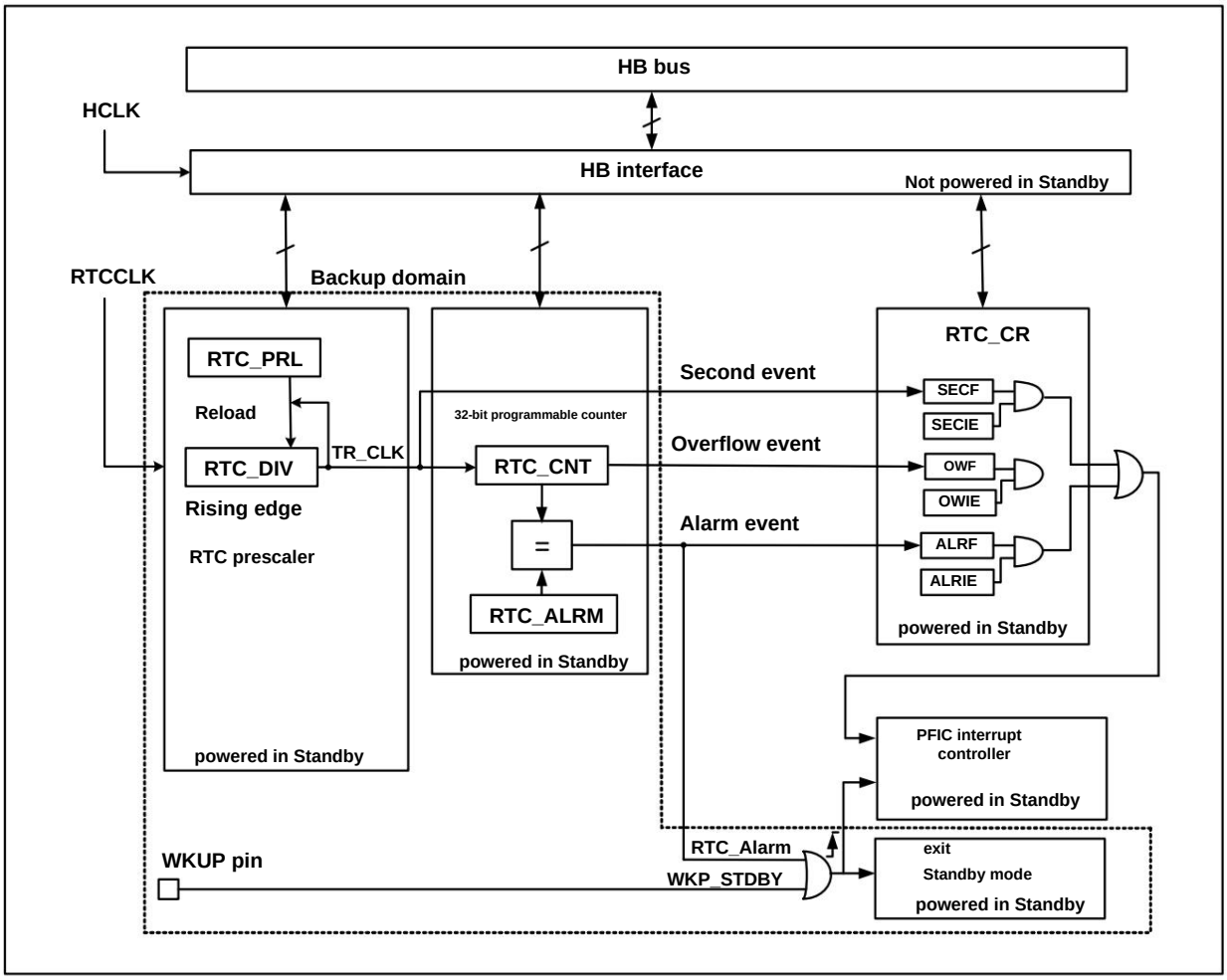
Multiple clock sources, interrupts

Independent reset

6.2 Functional Description

6.2.1 Overview

Figure 6-1 RTC Structure Diagram



As shown in Figure 6-1, the RTC module mainly consists of three parts: the HB bus interface, the frequency divider and counter, and the control and status registers.

The frequency divider and counter sections are located in the backup area and can be powered by VBAT. After RTCCLK is input to the frequency divider (RTC_DIV), it is divided into...

TR_CLK. It's worth noting that the frequency divider (RTC_DIV) internally uses a decrementing counter; when it overflows, it outputs TR_CLK.

Then, the preset value is retrieved from the reload value register (RTC_PSCR) and reloaded into the frequency divider. Reading the frequency divider actually reads its real-time value.

(Read only) The frequency divider should be written to the reload value register (RTC_PSCR). Normally, the period of TR_CLK is set to 1.
Seconds, TR_CLK will trigger a seconds event and simultaneously increment the main counter (RTC_CNT) by 1; when the main counter increments to the alarm register...
When the values match, an alarm event is triggered; when the main counter increments to overflow, an overflow event is triggered. All three events can trigger an alarm.
An interrupt is triggered, and the corresponding interrupt enable bit is controlled.

6.2.2 Reset

Due to the special purpose of real-time clocks, their four registers in the backup domain are: prescaler, prescaler reload value, main counter, and alarm.
The clock can only be reset via the backup domain reset signal; refer to the backup domain reset section of the RCC documentation. The real-time clock control register is subject to system reset.
Position or power reset control.

6.2.3 Special Register Read/Write Operations

Due to the special purpose of the real-time clock, the RTC and HB buses are independent. The HB bus's reading of the RTC is not necessarily real-time.
Reading the RTC registers must be done after HB startup and after one rising edge of the RTC has passed. This can occur during system reset and power-on.
After reset and wake-up from stop mode, a convenient approach is to wait for the RSF bit of the control register (CTLR) to be set high before writing to the RTC.
The operator must wait for the previous write operation to complete and must enter configuration mode. The specific steps are as follows:

1) Check the RTOFF bit until it becomes 1;

2) Set the CNF bit to enter configuration mode;

3) Perform write operations on one or more RTC registers;

4) Clear the CNF bit, exit configuration mode, and the HB interface begins writing to the RTC register;

5) Check the RTOFF bit until it becomes 1, then the writing is complete.

6.3 Register Description

Table 6-1 List of RTC-related registers

name	Access address	describe	Reset value
R16_RTC_CTLRH	0x40002800	RTC control register high bit	0x0000
R16_RTC_CTLRL	0x40002804	RTC control register low byte	0x0020
R16_RTC_PSCRH	0x40002808; prescaler	reload value register high byte 0x4000280C;	0x0000
R16_RTC_PSCRL	prescaler reload value	register low byte 0x40002810; divider	0xXXXX
R16_RTC_DIVH	register high byte 0x40002814; divider register low		0xXXXX
R16_RTC_DIVL	byte		0xXXXX
R16_RTC_CNTH	0x40002818	RTC counter high bit	0xXXXX
R16_RTC_CNTL	0x4000281C	RTC counter low byte	0xXXXX
R16_RTC_ALRMH	0x40002820 Alarm register high byte 0x40002824		0xXXXX
R16_RTC_ALRML	Alarm register low byte		0xXXXX

6.3.1 RTC Control Register High Byte (RTC_CTLRH) Offset Address:

0x00 Name

Bit		access	describe	Reset value
[15:3] Reserved		RO is reserved.		0
2	OWIE	RW	Overflow interrupt enable bit: 1: Enable; 0: Off.	0
1	ALRIE	RW alarm interrupt enable bit:		0

			1: Enable; 0: Off.	
0	SECIE	RW	Second interrupt enable bit: 1: Enable; 0: Off.	0

6.3.2 Offset address of the low byte of the RTC control register

(RTC_CTLRL) : 0x04 Bit

name [15:6] Reserved		access	describe	Reset value
		RO is reserved.		0
5	RTOFF	RO	The RTC operation status indicator bit indicates the last RTC operation. The execution status of the operation; operations on the RTC must wait for this bit. The value is 1. 1: The last operation on the RTC has been completed; 0: The last operation on the RTC is still in progress.	1
4	CNF	RW	Configure the flag bit; setting this bit to 1 will enter configuration mode, thereby... Allow input to the counter (R16_RTC_CNTx) and alarm register (R16_RTC_ALRMx) and prescaler reload value register (R16_RTC_PSCRx) Write value. Only write 1 to this bit and... Write operations will only be performed after the data is reset to zero by the software. 1: Enter configuration mode; 0: Exit configuration mode and begin updating the RTC registers.	0
3	RSF	RW0	The register synchronization flag is used in the prescaler of the RTC module. (PSCRx), Alarm Clock (ALRMx), Counter (CNTx) Before reading or writing these registers, it must be ensured that this bit is already... These registers have been set by the hardware to confirm that they have been... Step; when reading or writing these registers, either HB is reset or After the HB clock stops, the first step should be to reset this bit. 1: The registers have been synchronized; 0: The registers are not synchronized.	0
2	OWF	RW0	Counter overflow flag: This bit is used when a 32-bit counter overflows. Set by hardware. If the OWIE bit is set, a... An overflow interrupt. This bit can only be cleared by software, not by other means. Item placement.	0
1	ALRF	RW0	The alarm clock indicator appears when the counter value reaches the alarm clock register. The value of (ALRMx) will be set by the hardware if the alarm clock... Setting the interrupt enable bit (ALRIE) will also generate an alarm. Interruption. This bit can only be cleared by software, not set by software.	0
0	SECF	RW0	Second event flag, which is generated every second after the clock is divided by the prescaler. A falling edge will cause the counter to increment by one, and simultaneously generate... When a second event occurs, this bit will be set; if a second interrupt is triggered... Enable (SECIE is set), and a second will also be generated. Interruption. This bit can only be cleared by software, not set by software.	0

6.3.3 Prescaler Reload Value Register High Bit Offset Address (RTC_PSCRH):

0x08 Bit Name [15:4]

Reserved [3:0] PRL [19:16]	access	describe	Reset value
	RO is reserved.		0
	WO reinstallation value is high.		0

6.3.4 Offset address of the low byte of the prescaler reload value register

(RTC_PSCRL) : 0x0C

Bit	name	access	describe	Reset value
[15:0] PRL[15:0]		WO	The reload value is the low-order bit. The actual frequency division coefficient is... (PRL[19:0]+1), for example, if the RTC input frequency is 32768Hz, then setting this value to 0x7fff will allow for frequency division. Output a signal with a 1-second cycle.	X

6.3.5 Offset address of the high-order bit (RTC_DIVH) of the frequency divider

register: 0x10 Bit name	access	describe	Reset value
[15:4] Reserved [3:0] DIV	RO is reserved.		0
[19:16]	RO divider register high bit.		X

6.3.6 Offset address of the low byte (RTC_DIVL) of the frequency divider

Bit	register: 0x14 Name	access	This	Reset value
[15:0] DIV[15:0]		RO	describes the low-order bits of the divider register. DIV is actually a decrementing counter. The counter, RTC_CLK, will activate the DIV counter with each clock cycle. Decrement by 1, and upon overflow, a TR_CLK will be output, along with a value from PSCR. Reload value. A DIV can only be read; what is read is the current value. The remaining value of the frequency counter.	X

6.3.7 RTC counter high bit (RTC_CNTH) offset address:

0x18 Name

Bit		access	describe	Reset value
[15:0] CNT[31:16]		RW is the high bit of the counter.		X

6.3.8 Offset address of the low byte of the RTC counter

(RTC_CNTL) : 0x1C

Bit	name	access	The	Reset value
[15:0] CNT[15:0]		RW	low-order bit of the counter is the core component of the RTC timer, defined by TRCLK. (The period is typically set to 1 second) This provides a clock. It is accessed by reading... Use CNT[31:0] to calculate the current time. Writing this value requires... To enter configuration mode.	X

6.3.9 Alarm Register High Byte (RTC_ALRMH) Offset

Address: 0x20

Bit	name	access	describe	Reset value
[15:0]	ALR[31:16]	RW	is the high bit of the alarm register.	X

6.3.10 Alarm Register Low Byte (RTC_ALRML) Offset Address:

0x24 Name

Bit		access		Reset value
[15:0]	ALR[15:0]	RW	Describes the low-order bits of the alarm register. When the alarm register ALRM[31:0]... A value is generated when it matches the value of the counter CNT[31:0]. Alarm clock event. Changing this value requires entering configuration mode.	X

Chapter 7 Independent Watchdog (IWDG)

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

7.1 Main Features

ÿ 12-bit decrementing counter

The clock source is an LSI divider, which allows it to operate in low-power mode.

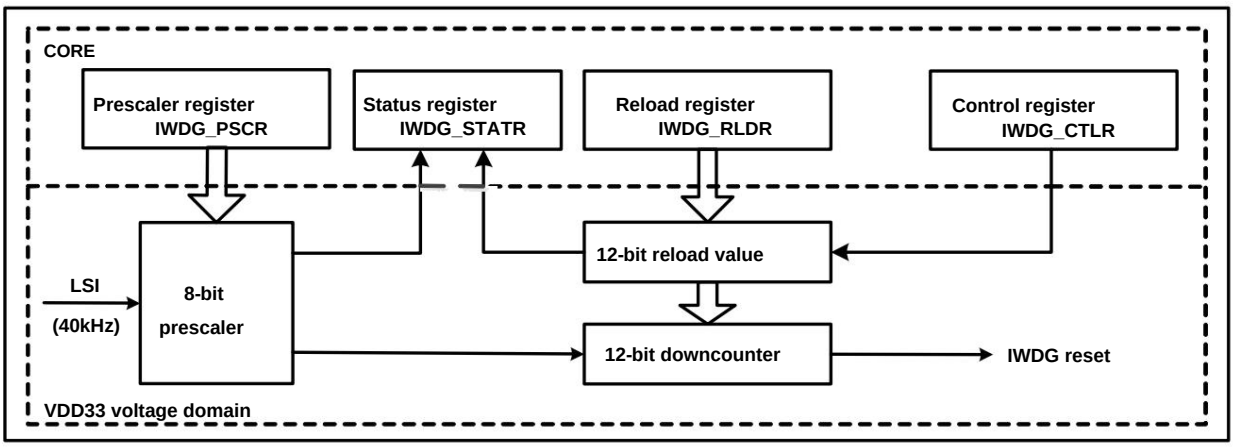
Reset condition: The counter value decreases to 0.

7.2 Functional Description

7.2.1 Principle and Usage: The

independent watchdog timer uses an LSI clock as its clock source, and its function continues to operate normally in stop mode. When the watchdog counter decrements to 0...
When this happens, a system reset will occur, so the timeout period is (reload value + 1) clock cycles.

Figure 7-1 Structural diagram of an independent watchdog



ÿActivate independent watchdog

After a system reset, the watchdog timer is disabled. Writing 0xCCCC to the IWDG_CTLR register enables the watchdog timer; afterwards, it cannot be used again.
It is turned off unless a reset occurs.

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

Watchdog configuration

Internally, the watchdog timer is a decrementing 12-bit counter. When the counter reaches zero, a system reset occurs. Enable IWDG.
This function requires the following steps:

1) Counting Time Base: The IWDG clock source is the LSI. The LSI divider value is set through the IWDG_PSCR register to serve as the counting time for the IWDG.

The basic procedure is to first write 0x5555 to the IWDG_CTLR register, and then modify the divider value in the IWDG_PSCR register. IWDG_STATR

The PVU bit in the register indicates the frequency divider update status. The frequency divider can only be modified and read after the update is complete.

out.

2) Reload Value: Used to update the current value of the counter in the standalone watchdog timer, and the counter is decremented by this value. The operation method is to first...

Write 0x5555 to the IWDG_CTLR register, then modify the IWDG_RLDR register to set the target reload value. IWDG_STATR register.

The RVU bit in the register indicates the reload value update status; the IWDG_RLDR register can only be modified after the update is complete.

And read it aloud.

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

4) Feed the watchdog: This involves refreshing the current watchdog counter value before it decrements to 0 to prevent a system reset. This is done by updating the IWDG_CTLR register.

Write 0xAAAA to instruct the hardware to update the IWDG_RLDR register value to the watchdog counter. This action must be performed after the watchdog function is enabled.

It must be executed on a scheduled basis; otherwise, a watchdog reset will occur.

7.2.2 Debug Mode When the

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

7.3 Register Description

Table 7-1 List of IWDG Related Registers

name	Access address description		Reset value
R16_IWDG_CTLR	0x40003000	IWDG Control Register	0x0000
R16_IWDG_PSCR	0x40003004	IWDG Frequency Division Factor Register	0x0000
R16_IWDG_RLDR	0x40003008	IWDG Reload Value Register	0x0FFF
R16_IWDG_STATR	0x4000300C	IWDG Status Register	0x0000

7.3.1 IWDG Control Register (IWDG_CTLR) Offset Address: 0x00

Name

Bit		access	describe	Reset value
[15:0] KEY[15:0]		WO	Operate key-value locks. 0xAAAA: Feed the dog. Load the IWDG_RLDR register value into the unique... The watchdog counter is in operation; 0x5555: Allows modification of R16_IWDG_PSCR and R16_IWDG_ RLDR register; 0xCCCC: Enable watchdog timer if hardware watchdog timer is enabled. (User-selected word configuration) is not subject to this restriction.	0

7.3.2 IWDG Divider Register (IWDG_PSCR) Offset Address: 0x04 Name

Bit		access	describe	Reset value
[15:3] Reserved		RO is reserved.		0
[2:0] PR[2:0]		RW	IWDG is the clock divider factor. Before modifying this field, you must write it to the KEY. 0x5555. 000: 4-division; 001: 8 frequency division; 010: 16-division; 100: 011:32 frequency division; 64-division; 110: 256- 101:128 frequency division; division; IWDG 111:256 frequency division. counting time base = LSI / division coefficient. Note: Before reading this field value, ensure that the IWDG_STATR register is empty. The bit must be 0; otherwise, the read value is invalid.	000b
	RVU			

7.3.3 IWDG Reload Value Register (IWDG_RLDR) Offset Address: 0x08

Bit	name	access	describe	Reset value
[15:12] Reserved		RO Reserved.		0
[11:0] RL[11:0]		RW	Counter reload value. Write to KEY before modifying this field. 0x5555. When 0xAAAA is written to KEY, the value of this field will be hard-coded. The item is loaded into the counter, and then the counter starts from this value. Decreasing count. Note: Before reading or writing this field value, ensure that the IWDG_STATR register is empty. The position in the middle is RVU 0, otherwise reading or writing this field is invalid.	0xFFF

7.3.4 IWDG Status Register (IWDG_STATR) Offset Address: 0x0C Bit

Name [15:2] Reserved

		access	describe	Reset value
		RO is reserved.		0
1	RVU	RO	Reload value update flag. Hardware set or clear to 0. 1: Reload value update is in progress; 0: Reload update complete (maximum 5 LSI cycles). Note: The reload value register IWDG_RLDR is only valid when it is in place. RVU can only be read and written after it has been cleared. 0	0
0	PVU	RO	Clock divider update flag. Hardware set or clear to 0. 1: Clock divider value update in progress; 0: Clock divider value update complete (maximum 5 LSI cycles). Note: The frequency division factor register IWDG_PSCR is only available when in place. PVUs can only be read and written after they are cleared.	0

Note 0: After the prescaler or reload value update, there is no need to wait or reset; you can continue executing the following code. (Even at low power mode, this write operation will still be executed and completed. In RVU/PVU

Chapter 8 Window Watchdog (WWDG)

A watchdog timer is typically used to monitor software malfunctions in a system, such as external interference or unforeseen logical errors.
The counter needs to be refreshed (fed) within a specific window of time (with upper and lower limits); otherwise, it will refresh earlier or later than this window.
The watchdog circuit will trigger a system reset.

8.1 Main Features

ÿ Programmable 7-bit decrementing counter

ÿ Dual-condition reset: The current counter value is less than 0x40, or the counter value is reloaded outside the window time.

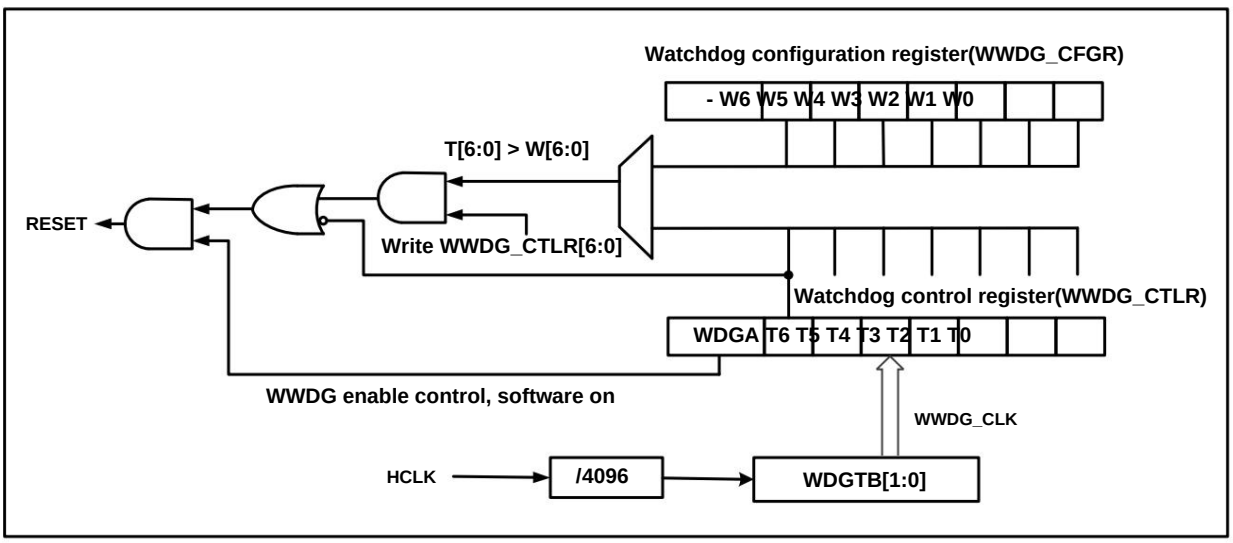
• Early Wake-Up Notification (EWI) function is used to prevent system reset by promptly feeding the dog.

8.2 Functional Description

8.2.1 Principles and Usage The

window watchdog operates based on a 7-bit decrementing counter, which is mounted on the HB bus, with the counting time base WWDG_CLK.
The clock frequency (HCLK/4096) is divided, and the division factor is set in the WDGTB[1:0] fields of the configuration register WWDG_CFGR. Decrement counter.
In free-running mode, the counter continuously decrements regardless of whether the watchdog function is enabled. As shown in Figure 8-1, this is a window-watching operation.
Internal structure diagram of a dog.

Figure 8-1 Window Watchdog Structure Diagram



Start the window watchdog

After a system reset, the watchdog timer is off. Setting the WDGA bit in the WWDG_CTLR register enables the watchdog timer.
It can no longer be turned off unless a reset occurs.
Note: The gatekeeper can be indirectly stopped by pausing the WWDG_CLK count and disabling the clock source by setting the RCC_HB1PCENR register.
The
WWDG dongle function, or the module reset function by setting the RCC_HB1PRSTR register, is equivalent to a reset.

Watchdog configuration

Internally, the watchdog timer is a continuously decrementing 7-bit counter that supports read and write access. To use the watchdog reset function, you need to...
Perform the following operations:
1) Counting time base: through the WDGTB[1:0] bit fields of the WWDG_CFGR register, note that the WWDG module clock of the RCC unit must be enabled.
2) Window Counter: Sets the W[6:0] bit fields of the WWDG_CFGR register. This counter is used by hardware for comparison with the current counter.

The value is configured by the user software and will not change. It serves as the upper limit for the window time.

3) Watchdog Enable: Set the WDGA bit of the WWDG_CTLR register to 1 via software to enable the watchdog function, which can reset the system.

4) Feed the watchdog: This refreshes the current counter value and configures the T[6:0] bit fields of the WWDG_CTLR register. This action requires the watchdog function to be enabled.

After activation, it will be executed within a periodic window; otherwise, a watchdog reset will occur.

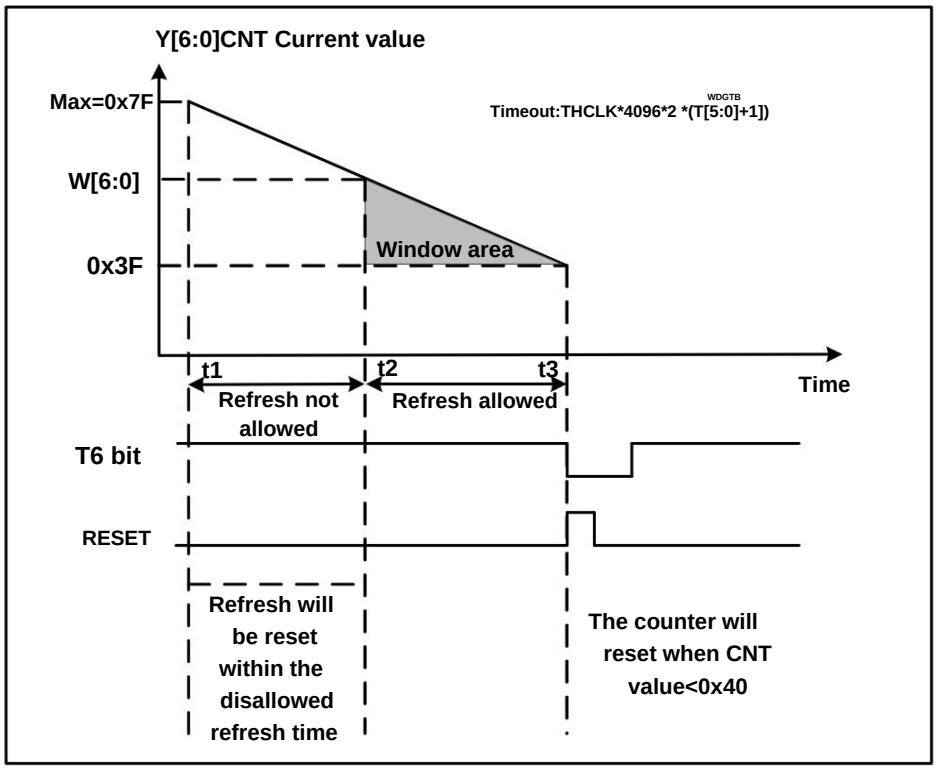
Dog feeding window time

As shown in Figure 8-2, the gray area represents the monitoring window area of the window watchdog, and its upper limit time t2 corresponds to the current counter value reaching...

The window value W[6:0] is the time point; its lower limit time t3 corresponds to the time point when the current counter value reaches 0x3F. Within this time range, t2 < t < t3.

You can perform a dog-feeding operation (write T[6:0]) to refresh the current counter value.

Figure 8-2 Counting mode of the window watchdog



Watchdog reset

1) If the watchdog timer is not fed in time, causing the value of the T[6:0] counter to change from 0x40 to 0x3F, a "window watchdog reset" message will appear.

A system reset will occur. That is, when the T6-bit is detected as 0 by the hardware, a system reset will happen.

Note: The application can reset the system by writing T6-bit to 0 via software, which is equivalent to a software reset function.

2) When the counter refresh action is performed within the time during which dog feeding is not allowed, i.e., when writing to the T[6:0] bit field is performed within the time t1ýtýt2, the output will be...

The "window watchdog reset" now triggers a system reset.

ÿ Early wake-up

To prevent system reset due to failure to refresh the counter in a timely manner, the watchdog module provides an Early Wake-up Interrupt (EWI) notification.

When the counter decrements 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 will be triggered simultaneously.

Watchdog interrupt. At this point, there is one counter clock cycle remaining (decreasing to 0x3F) before the hardware reset. The application can then proceed immediately within this time.

Perform the dog feeding operation.

8.2.2 Debug Mode When the

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

8.3 Register Description

Table 8-1 List of WWDG Related Registers

name	Access address	describe	Reset value
R16_WWDG_CTLR	0x40002C00	WWDG Control Register	0x007F
R16_WWDG_CFGR	0x40002C04	WWDG Configuration Register	0x007F
R16_WWDG_STATR	0x40002C08	WWDG Status Register	0x0000

8.3.1 WWDG Control Register (WWDG_CTLR) Offset Address:

0x00 Name

Bit		access	describe	Reset value
[15:8] Reserved		RO is reserved.		0
7	WDGA	RW1	Window watchdog reset enable bit. 1: Enable watchdog timer (which can generate a reset signal); 0: Disable watchdog functionality. Software write 1 is enabled, but hardware clearing to 0 is only allowed after a reset.	0
[6:0] T[6:0]		RW	A 7-bit decrementing counter, decrementing every 4096*2WDGTB HCLK cycles. Decrement by 1. When the counter decreases from 0x40 to 0x3F, i.e., T6... When the value jumps to 0, a watchdog reset is triggered.	0x7F

8.3.2 WWDG Configuration Register (WWDG_CFGR) Offset

Address: 0x04 Bit Name

[15:10] Reserved		access	describe	Reset value
		RO is reserved.		0
9	EWI	RW1	Early wake-up interrupt enable bit. If this position is 1, then a value of 0x40 will be generated. Interrupt. This bit can only be cleared by hardware after a reset.	0
[8:7] WDGTB[1:0]		RW	Window watchdog clock divider selection: 00:1 frequency division, counting time base = HCLK/4096; 01:2 frequency division, counting time base = HCLK/4096/2; Frequency division by 10:4, counting time base = HCLK/4096/4; 11:8 frequency division, counting time base = HCLK/4096/8.	0
[6:0] W[6:0]		RW	The window watchdog timer uses a 7-bit window value. This value is used to perform operations with the counter value. Comparison. The dog-feeding operation can only be performed when the counter value is less than the window value. And it is performed when it is greater than 0x3F.	0x7F

8.3.3 WWDG Status Register (WWDG_STATR) Offset Address: 0x08

Bit	name	access	describe	Reset value
[15:1] Reserved		WO is reserved.		0
0	EWIF	RW0	Early wake-up interrupt flag. When the counter reaches 0x40, this bit will be set by the hardware. It must be cleared to 0 via software; user-defined settings are ineffective. Even with EWI If not set, this bit will still be set when the event occurs.	0

			Bit.	

Chapter 9 GPIO and its Multiplexing Functions (GPIO/AFIO)

The chip has 6 built-in GPIO ports (PA0ÿPA15, PB0ÿPB15, PC0ÿPC15, PD0ÿPD15, PE0ÿPE15, PF0ÿPF14), totaling 95 GPIO pins. Most pins can be configured by software as output (push-pull or open-drain), input (with or without pull-up or pull-down), or multiplexed peripheral function ports.

Pins PA9-PA12, PC13-PC15, and PE3-PE6 are powered by VDD33, rated at 3.3V. When VDD33 is powered down, pins PC13-PC15 automatically switch to VBAT power. Pins

PA0-PA8, PB2-PB7, PB15, PC4-PC5, PD8, PE2, and PF6-PF10 are powered by VDDIO, rated at 3.3V.

It supports 1.8V, 2.5V, and 3.3V power supplies.

High-speed pins PA13ÿPA15, PB0ÿPB1, PB10ÿPB14, PC0ÿPC3, PC6ÿPC12, PD0ÿPD7, PD9ÿPD15.

PE0ÿPE1, PE7ÿPE15, PF0ÿPF5, and PF11ÿPF14 are powered by VIO18 and have built-in I/O pin voltage regulators, supporting 1.2V.

It supports 1.8V, 2.5V, and 3.3V voltages and dynamic power supply voltage switching. It also supports configuring the default voltage after power-on via the XO pin. For specific configuration information, please refer to the [12:9] fields of the PWR_CTLR register.

9.1 Main Features

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

- ÿ Floating input ÿ
- Pull-up input ÿ
- Pull-down input ÿ
- Analog input
- Open -drain
- output ; push-pull
- output ; multiplexed inputs and outputs

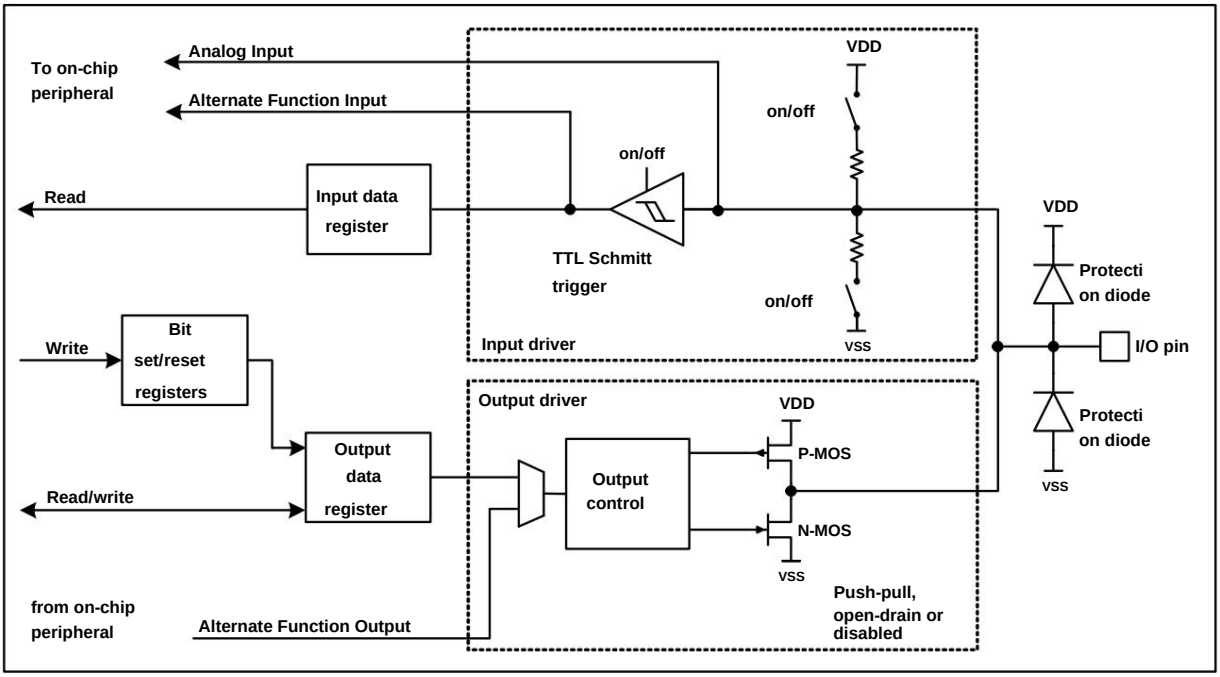
Many pins have multiplexing capabilities; many other peripherals map their output and input channels to these pins.

The specific usage of the pins needs to be referred to the respective peripherals, and whether these pins are multiplexed and remapped is explained in this chapter.

9.2 Functional Description

9.2.1 Overview

Figure 9-1 Basic structural block diagram of GPIO module



Note: VDD is the power supply voltage. The voltage type depends on the specific pin. Please refer to Chapter [Chapter Number] for details.

As shown in Figure 9-1, the I/O port structure has two protection diodes inside the chip for each pin. The I/O port can be internally divided into input and output.

Output driver module. The input driver has optional weak pull-up/pull-down resistors, allowing connection to analog input peripherals such as AD converters; if the input is to a digital...

Peripherals require a TTL Schmitt trigger before being connected to the GPIO input register or other multiplexed peripherals. The output driver...

There is a pair of MOSFETs; the I/O port can be configured as open-drain or push-pull output by configuring the enable/disable of the upper and lower MOSFETs; the output driver is internal.

It can also be configured to allow the output to be controlled by GPIO or by other multiplexed peripherals.

9.2.2 GPIO Initialization Function: Immediately

after reset, the GPIO ports operate in the initial state. At this time, most IO ports are operating in a floating input state, but some, such as HSE, are not.

The peripheral-related pins operate on peripheral multiplexing functions. For specific initialization functions, please refer to the relevant sections of the pin descriptions.

9.2.3 External Interrupts All

GPIO ports can be configured with external interrupt input channels, but each external interrupt input channel can only be mapped to one device.

On the GPIO pin, the external interrupt channel number must match the GPIO port number, such as PA1 (or PB1, PC1, PD1, ...).

PE1, etc., can only be mapped to EXTI1, and EXTI1 can only accept mappings to one of PA1, PB1, PC1, PD1, or PE1.

All relationships are one-to-one.

9.2.4 Multiplexing Function Multiplexers and Mapping

Multiplexing Functions

The I/O pins are connected to onboard peripherals/modules via a multiplexer that allows only one peripheral to be multiplexed at a time (AF).

Connect to the I/O pin. This ensures that peripherals sharing the same I/O pin will not conflict.

Each I/O pin has a multiplexer with up to 16 multiplexed function inputs (AF0 to AF15), which can be used via...

The GPIOx_AFLR and GPIOx_AFHR registers configure these inputs:

After reset, the multiplexer is selected as multiplexing function 0 (AF0). In multiplexing mode, this is achieved through the GPIOx_CFGLR/GPIOx_CFGHR registers.

Configure I/O.

The CH32H417DS0 datasheet details the specific multiplexing function assignments for each pin.

Multiplexing Function

Configuration : To use the input direction multiplexing function, the port must be configured in multiplexed input mode. The pull-up/down settings can be configured according to actual needs.

ÿ To use the output direction multiplexing function, 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, in which case the driver is configured in floating input mode.

Multiple peripherals may reuse the same I/O pin. Therefore, to ensure that each peripheral has maximum utilization, the peripheral's...

In addition to the default multiplexed pins, multiplexed pins can also be remapped to other pins to avoid being occupied.

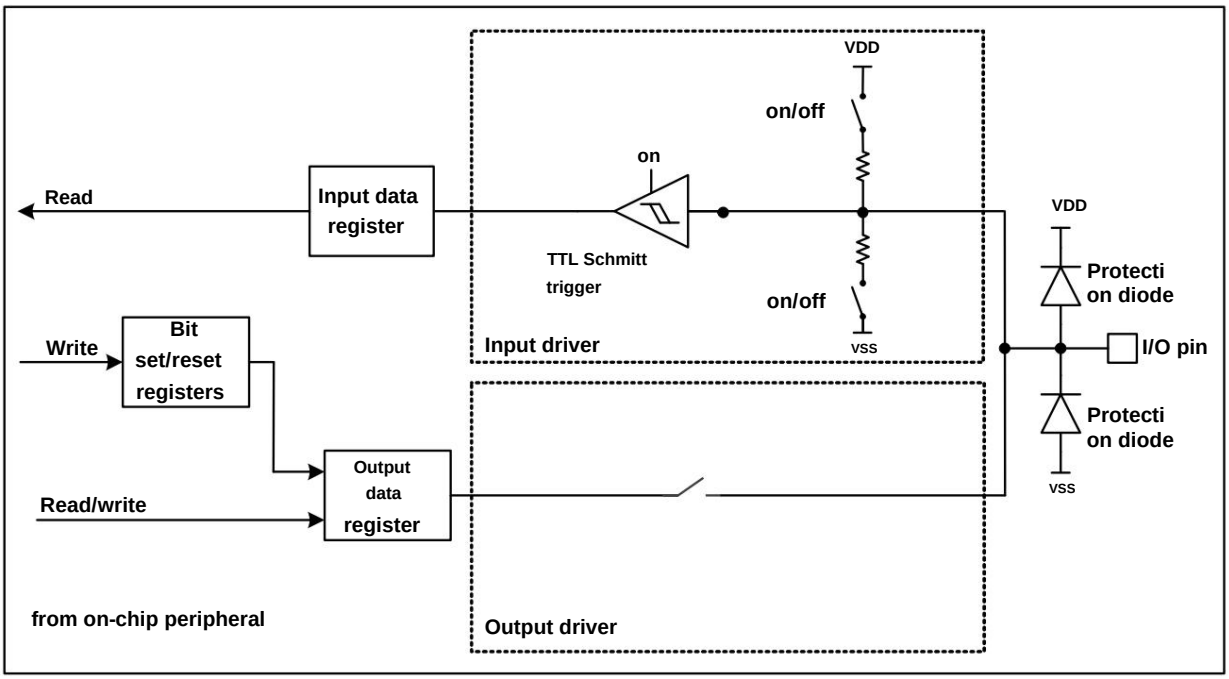
9.2.5 Locking Mechanism The

locking mechanism can lock the configuration of an I/O port. After a specific write sequence, the selected I/O pin configuration will be locked, and the following...

One cannot be changed before a reset.

9.2.6 Input Configuration

Figure 9-2 GPIO Module Input Configuration Structure Diagram



Note: VDD is the power supply voltage. The voltage type depends on the specific pin. Please refer to section [link/details] for details.

9

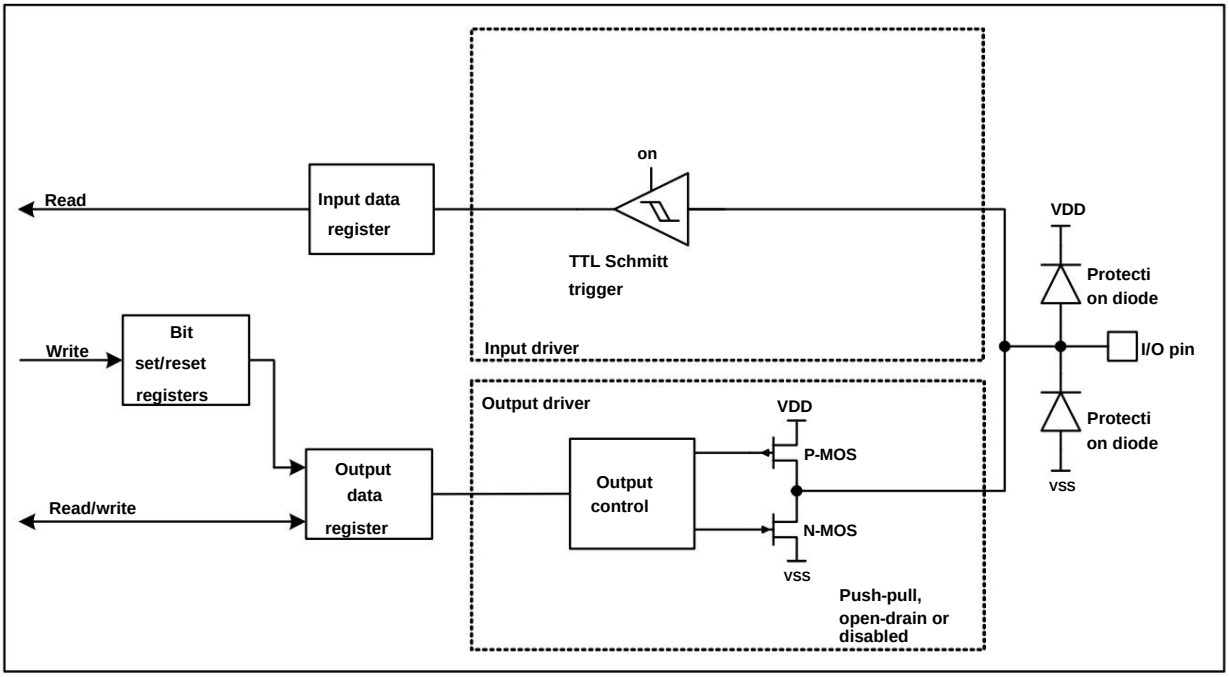
When the I/O port is configured as input mode, the output driver is disconnected, the input pull-up/pull-down sliders are selectable, and the multiplexing function and analog input are not connected.

Data from each I/O port is sampled into the input data register every HB clock cycle. Reading the corresponding bit from the input data register yields the data.

The voltage level of the corresponding pin.

9.2.7 Output Configuration

Figure 9-3 Block diagram of GPIO module output configuration structure



Note: VDD is the power supply voltage. The voltage type depends on the specific pin. Please refer to Chapter [Chapter Number] for details.

When the I/O port is configured in output mode, a pair of MOSFETs in the output driver can be configured in push-pull or open-drain mode as needed.

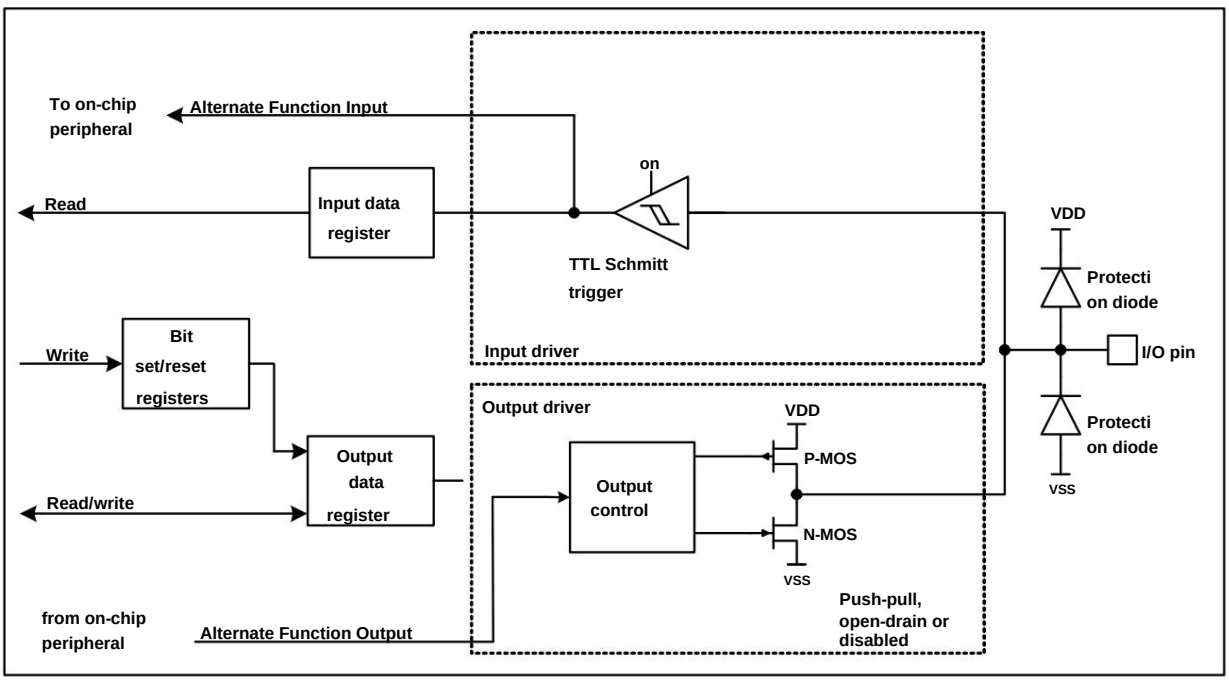
Use multiplexing functionality. The pull-up/pull-down resistors of the input drive are disabled, the TTL Schmitt trigger is activated, and the level appearing on the IO pin will be...

The input data register is sampled every HB clock cycle, so reading the input data register will provide the I/O status, which is then displayed in the push-pull output.

In this mode, accessing the output data register will retrieve the value of the last write.

9.2.8 Reuse Function Configuration

Figure 9-4 Block diagram of a GPIO module when it is reused by other peripherals

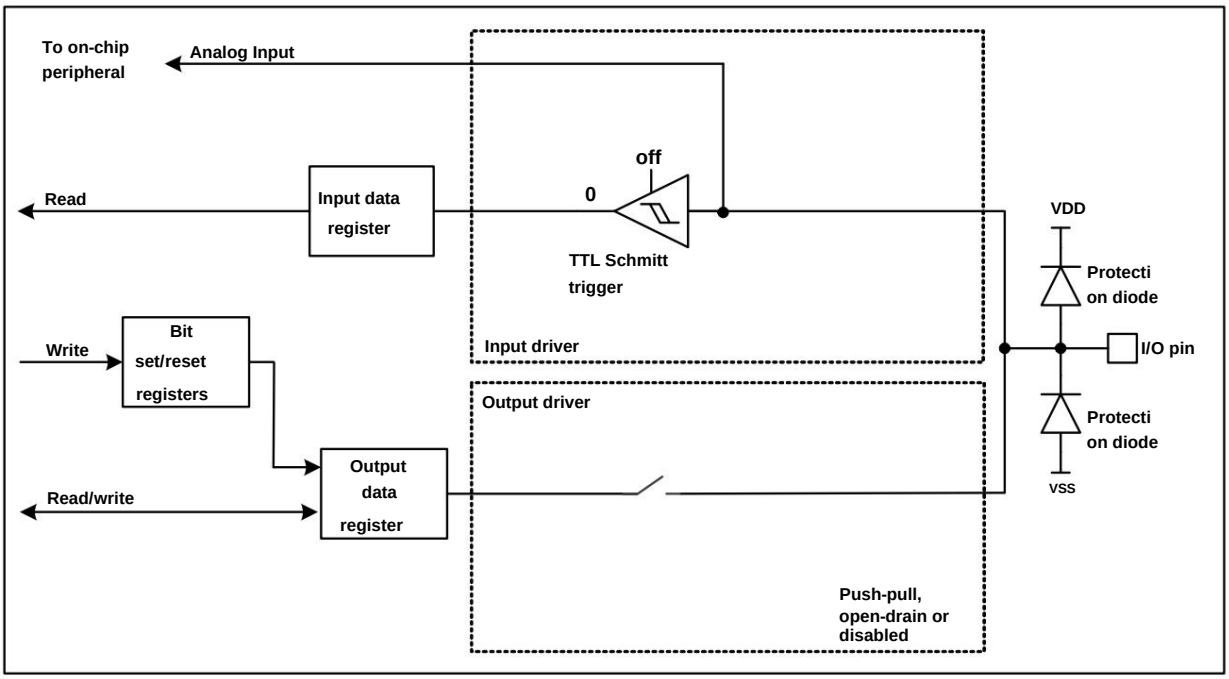


Note: VDD is the power supply voltage. The voltage type depends on the specific pin. Please refer to Chapter [Chapter Number] for details.

When multiplexing is enabled, the output driver is activated and can be configured as needed into open-drain or push-pull mode, and the Schmitt trigger also...
When enabled, the input and output lines of the multiplexed function are connected, but the output data register is disconnected, resulting in a voltage level appearing on the IO pin.
The input data register will be sampled every HB clock cycle. In open-drain mode, reading the input data register will provide the I/O port information.
Previous state; in push-pull mode, reading the output data register will yield the value of the last write.

9.2.9 Analog Input Configuration

Figure 9-5 Configuration block diagram of the GPIO module as an analog input



Note: VDD is the power supply voltage. The voltage type depends on the specific pin. Please refer to Chapter [Chapter Number] for details.

When analog input is enabled, the output buffer is disconnected, and the input to the Schmitt trigger in the input driver is disabled to prevent I/O.
With the power consumption at the port disabled by pull-up and pull-down resistors, the input data register will always return 0.

9.2.10 Peripheral GPIO Settings

The following table recommends the corresponding GPIO port configurations for the pins of each peripheral.

Table 9-1 Advanced Timers (TIM1/8)

TIMx	Configuration	GPIO Configuration
TIMx_CHx	Input capture channel x	Floating input
	Output compare channel	Push-pull multiplexed output
TIMx_CHxN	x Complementary output	Push-pull multiplexed output
TIMx_BKIN	channel x	Floating input
TIMx_BKIN2	Brake input	Floating input
TIMx_ETR	Brake input External trigger clock input	Floating input

Table 9-2 General Purpose Timers (TIM2/3/4/5/9/10/11/12)

TIMx pin	Configuration	GPIO Configuration

TIMx_CHx	Input capture channel x	Floating input
	Output compare channel	Push-pull multiplexed output
TIMx_ETR	x External trigger clock input	Floating input

Table 9-3 Low-power timers (LPTIM1/2)

LPTIMx pin	Configuration	GPIO Configuration
LPTIMx_CHx	Input capture channel x	Floating input
	Output compare channel	Push-pull multiplexed output
LPTIMx_ETR	x External trigger clock input	Floating input
LPTIMx_OC	PWM output	Push-pull multiplexed output

Table 9-4 General Purpose Synchronous/Asynchronous Serial Transceivers (USART1/2/3/4/5/6/7/8)

USARTx pin		GPIO Configuration
USARTx_TX	Configure full-	Push-pull multiplexed output
	duplex mode, half-duplex	Open-drain multiplexed output
USARTx_RX	mode, synchronous	Floating input or input with pull-up bar
	mode, hardware flow	Unused
USARTx_CK	control.	Push-pull multiplexed output
USARTx_RTS		Push-pull multiplexed output
USARTx_CTS		Floating input or input with pull-up bar

Table 9-5 Serial Peripheral Interface (SPI1/2/3/4) Modules

SPIx pin	Configuration	GPIO Configuration
SPIx_SCK	Master	Push-pull multiplexed output
	mode,	Floating input
SPIx_MOSI	Slave mode, Full-	Push-pull multiplexed output
	duplex, Master mode,	Floating input or input with pull-up bar
	Full-duplex, Slave mode, Simple bidirectional	Push-pull multiplexed output
	data cable / Master mode, Simple bidirectional	Unused
SPIx_MISO	data cable / Slave	Floating input or input with pull-up bar
	mode, Full-duplex,	Push-pull multiplexed output
	Master mode, Full-duplex, Slave mode,	Unused
	Simple bidirectional data cable / Master	Push-pull multiplexed output
SPIx_NSS	mode, Simple bidirectional	Floating, pull-up, or pull-down input
	data cable / Slave mode, Hardware master	Push-pull multiplexed output
	or slave mode, Hardware master mode / NSS output enabled, Software mode	Unused

Table 9-6 Built-in Audio Bus (I2S2/3) Module

I2Sx pin	Configuration	GPIO Configuration
I2Sx_WS	Master	Push-pull multiplexed output
	mode slave	Floating input
I2Sx_CK	mode	Push-pull multiplexed output
	master mode slave mode	Floating input

I2Sx_SD	Transmitter	Push-pull multiplexed output
	receiver	Floating, pull-up, or pull-down input
I2Sx_MCK	master	Push-pull multiplexed output
	mode slave mode	Unused

Table 9-7 Internal Integrated Bus (I2C1/2/3/4) Modules

I²Cx pin	Configuration	GPIO Configuration
I²Cx_SCL	I²C clock	Open-drain multiplexed output
I²Cx_SDA	I²C data	Open-drain multiplexed output

Table 9-8 I3 C Bus (I 3C) Module

I³C pin	Configuration	GPIO Configuration
I³C_SCL	I³C clock	Push-pull multiplexed output
I³C_SDA	I³C data	Push-pull multiplexed output

Table 9-9 Controller Area Network (CAN1/2/3) Modules

CANx pin	GPIO Configuration
CANx_TX	Push-pull multiplexed output
CANx_RX	Floating input or pull-up input

Table 9-10 USB PD Controller

USBPD pin	GPIO Configuration
USBPD_CC1/USBPD_CC2	Push-pull multiplexed output

Table 9-11 USB 2.0 High-Speed Host Device (USBHS) Controller

USBHS pin	GPIO Configuration
USBHS_DM/USBHS_DP	After enabling the USB module, the multiplexed I/O port will automatically connect to the internal USBHS transceiver. instrument

Table 9-12 USB Host Device (USBFS) Controller

USBFS pin	GPIO Configuration
USBFS_DM/USBFS_DP	Once the USB module is enabled, the multiplexed I/O port will automatically connect to the internal USBFS transceiver. instrument

Table 9-13 USB OTG_FS Controller

USB OTG_FS pin	GPIO Configuration
OTG_FS_VBUS	Analog Input
OTG_FS_ID	Pull-up input
OTG_FS_DM	Automatic control by USB power failure
OTG_FS_DP	Automatic control by USB power failure

Table 9-14 SDIO Controller Module

SDIO pin	Configuration	GPIO Configuration

CK	clock	Main mode: Push-pull multiplexed output From mode: Floating input
CMD		Push-pull multiplexed output
D[7:0]	Command Data	Push-pull multiplexed output

Table 9-15 SD/EMMC Controller Module

SD/EMMC pin	Configuration	GPIO Configuration
CK	clock	Main mode: Push-pull multiplexed output From mode: Push-Pull Multiplexed Output
CMD		Push-pull multiplexed output
D[7:0]		Push-pull multiplexed output
STR	Command data return clock signal, used by HS400 model	Push-pull multiplexed output

Table 9-16 Digital Filters for ÿÿ Modulator (DFSDM) Modules

DFSDM pin	Configuration	GPIO Configuration
DFSDM_DATINx	Serial data input, serial	Floating input
DFSDM_CKINx	clock input, internal	Floating input
DFSDM_CKOUT	clock output	Push-pull multiplexed output

Table 9-17 Serial-to-Parallel Conversion Controller and Transceiver (SerDes) Module

SerDes pins	Configuration	GPIO Configuration
SerDes_TXP	Differential transmission	Floating input
SerDes_TXN		Floating input
SerDes_RXP	Differential reception	Floating input
SerDes_RXN		Floating input

Table 9-18 Flexible Memory (FMC/SDRAM) Controller

FMC pin	GPIO Configuration
FMC_A[25:0] FMC_D[31:0]	Push-pull multiplexed output
FMC_CK	Push-pull multiplexed output
FMC_NOE FMC_NWE	Push-pull multiplexed output
FMC_NE1 FMC_NCE2	Push-pull multiplexed output
FMC_NWAIT	Floating input or input with pull-up bar
FMC_NBL[1:0]	Push-pull multiplexed output
FMC_SDCKE[1:0]	Push-pull multiplexed output
FMC_SDNE[1:0]	Push-pull multiplexed output
FMC_BA[1:0]	Push-pull multiplexed output
FMC_NRAS	Push-pull multiplexed output
FMC_NCAS	Push-pull multiplexed output

FMC_SDNWE	Push-pull multiplexed output
FMC_DQM[3:0]	Push-pull multiplexed output

Table 9-19 Analog-to-Digital Converters (ADCs) and Digital-to-Analog Converters (DACs)

ADC/DAC pins	GPIO Configuration
ADC/DAC	Analog Input

Table 9-20 High-Speed Analog-to-Digital Converters (HSADCs)

HSADC pin	GPIO Configuration
HSADC	Analog Input

Table 9-21 Digital Image Interface (DVP) Module

DVP pin		GPIO Configuration
PCLK	Configure pixel clock	Floating input or input with pull-up bar
DATA[11:0]	input, pixel data input,	Floating input or input with pull-up bar
VSYNC	frame synchronization	Floating input or input with pull-up bar
HSYNC	signal input, and line synchronization signal input.	Floating input or input with pull-up bar

Table 9-22 Serial Audio Interface (SAI) Module

SAI pin	Configuration	GPIO Configuration
SAI_SD	SD data line input/output bit	Multiplexed push-pull output
SAI_SCK	clock input/output	Multiplexed push-pull output
SAI_FS	FS signal input/output;	Multiplexed push-pull output
SAI_MCLK	external decoder master clock output	Multiplexed push-pull output

Table 9-23 QSPI Interface (QSPI) Module

QSPI pin	GPIO Configuration
QSPI_SCK	Multiplexed push-pull output
QSPI_SCSN	Multiplexed push-pull output
QSPI_SIO[3:0]	Multiplexed push-pull output
QSPI_SCSXN	Multiplexed push-pull output
QSPI_SIOX[3:0]	Multiplexed push-pull output

Table 9-24 Single-Wire Protocol Main Interface (SWPMI) Module

SWPMI pin		GPIO Configuration
SWP_RX	Configuration (Non-single-wire mode) Receive	Floating input
SWP_TX	signal input (Non-single-wire mode) Transmit	Multiplexed push-pull output
SWP_SUP	signal output (Non-single-wire mode)	Multiplexed push-pull output
SWPMI_IO	Suspend signal output (single-wire mode) Input/Output	Multiplexed push-pull output

Table 9-25 LCD-TFT Display Controller (LTDC) Module

LTDC pin		GPIO Configuration
LCD_CLK	Configure clock output	Multiplexed push-pull output

LCD_VSYNC	Vertical synchronization	Multiplexed push-pull output
LCD_HSYNC	output, horizontal	Multiplexed push-pull output
LCD_DE	synchronization output,	Multiplexed push-pull output
LCD_R[7:0]	non-data enabled output data: 8-bit red	Multiplexed push-pull output
LCD_G[7:0]	data, 8-bit green data, 8-bit blue data.	Multiplexed push-pull output
LCD_B[7:0]		Multiplexed push-pull output

Note: LCD-TFT controller pins must be configured via user program. Unused pins can be used for other functions.

For high-bit (RGB888) output, if a pixel depth lower than 24 bits is used, for example,

the LCD-TFT output bit or bit display, then control the display data cable must be connected to RGB

When the LCD-TFT controller is connected to an RGB 565-bit display, the R[4:0], G[5:0], and B[4:0] data line pins of the LCD display...

Must go LCD_R[7:3], LCD_G[7:2], and LCD_B[7:3] of the LCD-TFT controller.

24 LTDC 8bpp RGB666 data cable MSB. For example, RGB The 16-18

16

Table 9-26 Other I/O Function Settings

pin	Configuration Function	GPIO Configuration
(PC13) RTC	RTC output	Automatic hardware setup
MCO	Clock output	Push-pull multiplexed output
EXTI	external interrupt input	Floating, pull-up, or pull-down input

9.2.11 Multiplexing Function Remapping GPIO Settings

9.2.11.1 OSC32_IN/OSC32_OUT as GPIO ports PC14/PC15

When LSEON=0, the LSE oscillator pins OSC32_IN/OSC32_OUT can be used as GPIOs PC14/PC15 respectively.

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

9.2.11.2 XI/XO pins as GPIO ports PD0/PD1

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

9.2.11.3 ADC Multiplexing Function Remapping

Table 9-27 ADC1 External Trigger Injection Conversion Multiplexing Function Remapping

Reuse function	ADC1_ETRGINJ_RM=0 Default mapping	ADC1_ETRGINJ_RM=1 remapping
ADC1 External Trigger Injection Conversion	ADC1 external trigger injection conversion and EXTI15 connected	ADC1 external trigger injection conversion and TIM8_CH4 connected

Table 9-28 ADC1 External Trigger Rule Conversion Multiplexing Function Remapping

Reuse function	ADC1_ETRGREG_RM=0 Default mapping	ADC1_ETRGREG_RM=1 remapping
ADC1 External Trigger Rule Conversion	ADC1 external trigger rule conversion and EXTI11 connected	ADC1 external trigger rule conversion and TIM8_TRGO connected

Table 9-29 ADC2 External Trigger Injection Conversion Multiplexing Function Remapping

Reuse function	ADC2_ETRGINJ_RM=0 Default mapping	ADC2_ETRGINJ_RM=1 remapping

ADC2 external trigger injection conversion	ADC2 external trigger injection conversion and EXTI15 connected	ADC2 external trigger injection conversion and TIM8_CH4 connected

Table 9-30 ADC2 External Trigger Rule Conversion Multiplexing Function Remapping

Reuse function	ADC2_ETRGREG_RM=0 Default mapping	ADC2_ETRGREG_RM=1 remapping
ADC2 External Trigger Rule Conversion	ADC2 external trigger rule conversion and EXTI11 connected	ADC2 external trigger rule conversion and TIM8_TRGO connected

9.2.11.4 TIM2 Internal Trigger 1 Multiplexing Function Remapping

Table 9-31 TIM2 ITR1 Multiplexing Function Remapping

Reuse function	TIM2ITR1_RM=0	TIM2ITR1_RM=1
TIM2 Internal Trigger 1	Internally connect RIM2_ITR1 PTP output to Ethernet	internally connects RIM2_ITR1 SOF to full-speed USB OTG Output

9.2.11.5 SDMMC Multiplexing Function Remapping

Table 9-32 SDMMC Multiplexing Function Remapping

Reuse function	SDMMC_RM=00 Default Mapping	SDMMC_RM=01 remapping	SDMMC_RM=1x remapping
STS/CMD	PD2	PD12	PC10
SDCK/SLVCK	PC12	PD11	PC12
STR	PD3	PD10	PC11
D0	PC8	PB13	PD0
D1	PC9	PC9	PD1
D2	PC10	PB10	PD2
D3	PC11	PB11	PD3
D4	PA14	PA14	PD4
D5	PA15	PA15	PD5
D6	PC6	PC6	PD6
D7	PC7	PC7	PD7

9.2.11.6 UHSIF Multiplexing Function Remapping

Table 9-33 UHSIF PORT Multiplexing Function Remapping

Reuse function	UHSIF_PORT_RM=00 Default mapping	UHSIF_PORT_RM=01 Remapping	UHSIF_PORT_RM=1x Remapping
PORT0	PF12	PF12	PC1
PORT1	PF13	PF13	PC2
PORT2	PE7	PE7	PC3
PORT3	PE8	PC1	PB0
PORT4	PE9	PC2	PB1

PORT5	PE10	PC3	PE10
PORT6	PE11	PB0	PE11
PORT7	PE12	PB1	PE12
PORT8	PE13		
PORT9	PE14		
PORT10	PE15		
PORT11	PB10		
PORT12	PB11		
PORT13	PB12		
PORT14	PB13		
PORT15	PB14		
PORT16	PD10		
PORT17	PD11		
PORT18	PD12		
PORT19	PD13		
PORT20	PD14		
PORT21	PD15		
PORT22	PF0		
PORT23	PF1		
PORT24	PF2		
PORT25	PC6		
PORT26	PC7		
PORT27	PC8		
PORT28	PC9		
PORT29	PA13		
PORT30	PA14		
PORT31	PA15		
PORT32	PC10		
PORT33	PC11		
PORT34	PC12		
PORT35	PD0		
PORT36	PD1		
PORT37	PD2		
PORT38	PD3		
PORT39	PD4		
PORT40	PD5		
PORT41	PD6		
PORT42	PD7		
PORT43	PF3		
PORT44	PF4		

PORT45	PF5
PORT46	PE0
PORT47	PE1

Table 9-34 UHSIF CLK Multiplexing Function Remapping

Reuse function	UHSIF_CLK_RM=00 Default mapping	UHSIF_CLK_RM=01 Remapping	UHSIF_CLK_RM=10 Remapping	UHSIF_CLK_RM=11 Remapping
UHSIF_CLK	PF11	PC0	PF14	PD9

9.3 Register Description

9.3.1 GPIO Register Description

Unless otherwise specified, GPIO registers must be operated in word mode (32-bit operation of these registers).

Table 10-35 List of GPIO Related Registers

name	Access address	describe	Reset value
R32_GPIOA_CFGLR	0x40010800	PA Port Configuration Register Low Bit	0x44444444
R32_GPIOB_CFGLR	0x40010C00	PB port configuration register low bits	0x44444444
R32_GPIOC_CFGLR	0x40011000	PC Port Configuration Register Low Bit	0x44444444
R32_GPIOD_CFGLR	0x40011400	PD Port Configuration Register Low Bit	0x44444444
R32_GPIOE_CFGLR	0x40011800	PE port configuration register low bit	0x44444444
R32_GPIOF_CFGLR	0x40011C00	PF Port Configuration Register Low Bit	0x44444444
R32_GPIOA_CFGHR	0x40010804	PA Port Configuration Register High Bit	0x44444444
R32_GPIOB_CFGHR	0x40010C04	PB port configuration register high bit	0x44444444
R32_GPIOC_CFGHR	0x40011004	PC Port Configuration Register High Bit	0x44444444
R32_GPIOD_CFGHR	0x40011404	PD Port Configuration Register High Bit	0x44444444
R32_GPIOE_CFGHR	0x40011804	PE port configuration register high bit	0x44444444
R32_GPIOF_CFGHR	0x40011C04	PF Port Configuration Register High Bit	0x44444444
R32_GPIOA_INDR	0x40010808	PA Port Input Data Register	0x0000XXXX
R32_GPIOB_INDR	0x40010C08	PB port input data register	0x0000XXXX
R32_GPIOC_INDR	0x40011008	PC Port Input Data Register	0x0000XXXX
R32_GPIOD_INDR	0x40011408	PD port input data register	0x0000XXXX
R32_GPIOE_INDR	0x40011808	PE port input data register	0x0000XXXX
R32_GPIOF_INDR	0x40011C08	PF Port Input Data Register	0x0000XXXX
R32_GPIOA_OUTDR	0x4001080C	PA Port Output Data Register	0x00000000
R32_GPIOB_OUTDR	0x40010C0C	PB port output data register	0x00000000
R32_GPIOC_OUTDR	0x4001100C	PC Port Output Data Register	0x00000000
R32_GPIOD_OUTDR	0x4001140C	PD port output data register	0x00000000
R32_GPIOE_OUTDR	0x4001180C	PE port output data register	0x00000000
R32_GPIOF_OUTDR	0x40011C0C	PF Port Output Data Register	0x00000000
R32_GPIOA_BSHR	0x40010810	PA Port Set/Reset Register	0x00000000
R32_GPIOB_BSHR	0x40010C10	PB Port Set/Reset Register	0x00000000
R32_GPIOC_BSHR	0x40011010	PC Port Set/Reset Register	0x00000000
R32_GPIOD_BSHR	0x40011410	PD Port Set/Reset Register	0x00000000

R32_GPIOE_BSHR	0x40011810	PE Port Set/Reset Register	0x00000000
R32_GPIOF_BSHR	0x40011C10	PF Port Set/Reset Register	0x00000000
R32_GPIOA_BCR	0x40010814	PA Port Reset Register	0x00000000
R32_GPIOB_BCR	0x40010C14	PB port reset register	0x00000000
R32_GPIOC_BCR	0x40011014	PC Port Reset Register	0x00000000
R32_GPIOD_BCR	0x40011414	PD Port Reset Register	0x00000000
R32_GPIOE_BCR	0x40011814	PE port reset register	0x00000000
R32_GPIOF_BCR	0x40011C14	PF Port Reset Register	0x00000000
R32_GPIOA_LCKR	0x40010818	PA Port Lockout Configuration Register	0x00000000
R32_GPIOB_LCKR	0x40010C18	PB Port Lockout Configuration Register	0x00000000
R32_GPIOC_LCKR	0x40011018	PC Port Lockout Configuration Register	0x00000000
R32_GPIOD_LCKR	0x40011418	PD Port Lockout Configuration Register	0x00000000
R32_GPIOE_LCKR	0x40011818	PE Port Lockout Configuration Register	0x00000000
R32_GPIOF_LCKR	0x40011C18	PF Port Lockout Configuration Register	0x00000000
R32_GPIOA_SPEED	0x4001081C	PA Port Speed Register	0x00000000
R32_GPIOB_SPEED	0x40010C1C	PB port speed register	0x00000000
R32_GPIOC_SPEED	0x4001101C	PC Port Speed Register	0x00000000
R32_GPIOD_SPEED	0x4001141C	PD Port Speed Register	0x00000000
R32_GPIOE_SPEED	0x4001181C	PE port speed register	0x00000000
R32_GPIOF_SPEED	0x40011C1C	PF Port Speed Register	0x00000000

9.3.1.1 GPIO Configuration Register Low Bit (GPIOx_CFGLR) (x=A/B/C/D/E/F) Offset

Address: 0x00

31 30 29 28	27 26 25 24 23 22 21	2019	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	describe	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, configured through these bits. The corresponding port. In input mode (MODE=00b): 00: Analog input mode; 01: Floating input mode; 10: Includes pull-up and drop-down modes. 11: Retained. In output mode (MODE>00b): 00: General push-pull output mode; 01: General open-drain output mode; 10: Multiplexing function push-pull output mode; 11: Open-drain output mode for multiplexing function.	01b
[29:28] [25:24]	MODEy[1:0]	RW	(y=0-7), port x mode selection, configured through these bits. The corresponding port.	00b

[21:20] [17:16] [13:12] [9:8] [5:4] [1:0]			00: Input mode; 01: Output mode; 10: Output Mode; 11: Output mode.	

9.3.1.2 GPIO Configuration Register High Bit (GPIOx_CFGHR) (x=A/B/C/D/E/F) Offset

Address: 0x04

31	30	29	28	27	26	25	twenty four	twenty three	twenty two	twenty one	20	19	18	17	16
CNF15[1:0]		MODE15[1:0]		CNF14[1:0]		MODE14[1:0]		CNF13[1:0]		MODE13[1:0]		CNF12[1:0]		MODE12[1:0]	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNF11[1:0]		MODE11[1:0]		CNF10[1:0]		MODE10[1:0]		CNF9[1:0]		MODE9[1:0]		CNF8[1:0]		MODE8[1:0]	

Bit	name	access		Reset value
[31:30] [27:26] [23:22] [19:18] [15:14] [11:10] [7:6] [3:2]	CNFy[1:0]	RW	Description (y=8-15), configuration bits for port x, configured through these bits. Configure the corresponding port. In input mode (MODE=00b): 00: Analog input mode; 01: Floating input mode; 10: Includes pull-up and drop-down modes. 11: Retained. In output mode (MODE>00b): 00: General push-pull output mode; 01: General open-drain output mode; 10: Multiplexing function push-pull output mode; 11: Open-drain output mode for multiplexing function.	01b
[29:28] [25:24] [21:20] [17:16] [13:12] [9:8] [5:4] [1:0]	MODEy[1:0]	RW	(y=8-15), the mode bits of port x, configured through these bits. Configure the corresponding port. 00: Input mode; 01: Output mode; 10: Output Mode; 11: Output mode.	00b

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

Offset address: 0x08

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0

Bit	name	access	describe	Reset value
[31:16] Reserved		RO is		0
[15:0] IDRy		RO	reserved. (y=0-15), port input data. These bits are read-only and only... It can be read in 16-bit format. The read value is the corresponding bit. High and low status.	X

9.3.1.4 Port Output Register (GPIOx_OUTDR) (x=A/B/C/D/E/F) Offset Address: 0x0C

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0

Bit	name	access	describe	Reset value
[31:16] Reserved		RO is		0
[15:0] ODRy		RW	reserved. For output modes: (y=0-15), the data output from the port. This data can only... Operate in 16-bit format. The I/O ports output these registers externally. The value of the instrument. For input modes with drop-down menus: 1: Pull up the input bar; 0: Drop-down input.	0

9.3.1.5 Port Reset/Set Register (GPIOx_BSHR) (x=A/B/C/D/E/F) Offset Address: 0x10

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0

Bit	name	access		Reset value
[31:16] BRy		WO	Description (y=0-15), these bit positions will clear the corresponding OUTDR. Writing 0 to these bits has no effect. These bits can only be in 16-bit form. Access method. If both the BR and BS bits are set, then the BS bit... kick in.	0
[15:0] BSy		WO	(y=0-15), these bit positions will cause the corresponding OUTDR These bits are set; writing 0 has no effect. These bits can only be used in 16-bit increments. Access is in the form of [format]. If both the BR and BS bits are set, then... The BS bit comes into play.	0

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

Offset address: 0x14

31	30 29 28 27 26	25 24 23 22 21	2019	18	17 16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0

Bit	name	access	describe	Reset value
[31:16] Reserved		RO is reserved.		0
[15:0] BRy		WO	(y=0-15), these bit positions will clear the corresponding OUTDR. Writing 0 to these bits has no effect. These bits can only be in 16-bit form. Access in this manner.	0

9.3.1.7 Configure the lock register (GPIOx_LCKR) (x=A/B/C/D/E/F) offset address:

0x18

31	30	29	28	27	26	25	twenty four	twenty three	twenty two	twenty one	20	19	18	17	16
Reserved															LCKK

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

Bit	name	access	describe	Reset value
[31:17] Reserved		RO Retention		0
16	LCKK	RW	A lock key can be implemented by writing a specific sequence of keys. It is fixed, but it can be read at any time. A reading of 0 indicates that it is not... The lock is active; reading 1 indicates that the lock is active. 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 it can be used to confirm the lock key. It has been activated. No error during sequence writing will cause the activation to lock, and The value of LCK[15:0] cannot be changed while writing to the sequence. (Lock) Once the setting takes effect, the port can only be changed after the next reset. Configuration.	0
[15:0] LCKy		RW	(y=0-15), when these bits are 1, it indicates that the corresponding port is locked. The configuration. These bits can only be changed before LCKK is locked. Lock The specified configuration refers to the configuration registers GPIOx_CFGLR and GPIOx_CFGHR.	0

Note: Once a sequence has been executed on the corresponding port, the port configuration cannot be changed until the next system reset. LOCK

9.3.1.8 GPIO Port Speed Register (GPIOx_SPEED) (x=A/B/C/D/E/F) Offset Address: 0x1C

31 30 29 28	27 26 25 24 23 22 21	2019	18	17	16

SPEED15 [1:0]	SPEED14 [1:0]	SPEED13 [1:0]	SPEED12 [1:0]	SPEED11 [1:0]	SPEED10 [1:0]	SPEED9 [1:0]	SPEED8 [1:0]

15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
SPEED7 [1:0]	SPEED6 [1:0]	SPEED5 [1:0]	SPEED4 [1:0]	SPEED3 [1:0]	SPEED2 [1:0]	SPEED1 [1:0]	SPEED0 [1:0]

Bit	name	access	describe	Reset value
[31:0]	SPEEDy[1:0]	RW	(y=0-15), port x speed selection, through these bit configurations Configure the corresponding port. 00: 10MHz; 01: 50MHz; 10:100MHz; 11: 180MHz.	00b

9.3.2 AFIO Registers Unless

otherwise specified, AFIO registers must be operated in word mode (operate these registers in 32-bit mode).

Table 10-36 List of AFIO Related Registers

name	Access address	describe	Reset value
R32_AFIO_PCFR1	0x40010000	Remapping Register 1	0x00000000
R32_AFIO_GPIOA_AFLR	0x40010004	PA Port Multiplexing Function Register Low Bit	0x00000000
R32_AFIO_GPIOA_AFHR	0x40010008	PA Port Multiplexing Function Register High Bit	0x00000000
R32_AFIO_GPIOB_AFLR	0x4001000C	PB Port Multiplexing Function Register Low Byte	0x00000000
R32_AFIO_GPIOB_AFHR	0x40010010	PB Port Multiplexing Function Register High Byte	0x00000000
R32_AFIO_GPIOC_AFLR	0x40010014	PC Port Multiplexing Function Register Low Bit	0x00000000
R32_AFIO_GPIOC_AFHR	0x40010018	PC Port Multiplexing Function Register High Bit	0x00000000
R32_AFIO_GPIOD_AFLR	0x4001001C	PD Port Multiplexing Function Register Low Bit	0x00000000
R32_AFIO_GPIOD_AFHR	0x40010020	PD Port Multiplexing Function Register High Bit	0x00000000
R32_AFIO_GPIOE_AFLR	0x40010024	PE Port Multiplexing Function Register Low Bit	0x00000000
R32_AFIO_GPIOE_AFHR	0x40010028	PE Port Multiplexing Function Register High Bit	0x00000000
R32_AFIO_GPIOF_AFLR	0x4001002C	PF Port Multiplexing Function Register Low Byte	0x00000000
R32_AFIO_GPIOF_AFHR	0x40010030	PF Port Multiplexing Function Register	0x00000000
R32_AFIO_EXTICR1	High Byte 0x4001003C	External Interrupt Configuration	0x00000000
R32_AFIO_EXTICR2	Register 1 0x40010040	External Interrupt Configuration Register 2	0x00000000

9.3.2.1 Remapping Register 1 (AFIO_PCFR1) Offset

Address: 0x00

31 30 29 28 27	26 25	twenty four	twenty three	twenty two	twenty one	20	19	18	17	16
Reserved	SW_CFG[2:0]			Reserved		USBPD_CC_HVT	Reserved	VDD33_IO_HSLV	VDDIO_IO_HSLV	VIO18_IO_HSLV

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			TIM2ITR1_RM	SDMMC_RM[1:0]		UHSIF_PORT_RM[1:0]		UHSIF_CLK_RM[1:0]		Reserved	ADC2_ETRGINJ_RM	ADC2_ETRGREG_RM	ADC1_ETRGINJ_RM	ADC1_ETRGREG_RM	PD0PD1_RM

Bit	name	access	describe	Reset value
[31:27] Reserved		RO is reserved.		0
[26:24] SW_CFG[2:0]		RW	These bits are used to configure the I/O ports for SW and tracing functions. SWD (SDI) is the debug interface for accessing the kernel. It always appears after a system reset. As an SWD port. 0xx: Enable SWD (SDI); 100: Disable SWD (SDI) and use it as a GPIO function; Other: Invalid.	000b
[23:21] Reserved		RO is reserved.		0
20	USBPD_CC_HVT	RW	CC pin input channel threshold adjustment. USBPD CC pin high threshold input mode enabled: 1: High threshold input can reduce power consumption during PD communication; 0: Normal GPIO threshold input.	0
19	Reserved	RO is reserved.		0
18	VDD33_IO_HSLV	RW	VDD33 IO speed configuration at low voltage: 1: When the I/O voltage is less than 2.7V, the I/O speed can be enhanced; 0: When IO operates at low voltage, the speed will not be enhanced.	0
17	VDDIO_IO_HSLV	RW	VDDIO IO speed configuration at low voltage: 1: When the I/O voltage is less than 2.7V, the I/O speed can be enhanced; 0: When IO operates at low voltage, the speed will not be enhanced.	0
16	VIO18_IO_HSLV	RW	VIO18 IO speed configuration at low voltage: 1: When the I/O voltage is less than 2.7V, the I/O speed can be enhanced; 0: When IO operates at low voltage, the speed will not be enhanced.	0
[15:13] Reserved		RO is reserved.		0
[12] TIM2ITR1_RM		RW	TIM2 internally triggers 1 remapping. 1: Internally connect TIM2_ITR1 to the SOF of full-speed USB OTG. Output; 0: Internally connects TIM2_ITR1 to the PTP output of Ethernet.	0
[11:10] SDMMC_RM[1:0]		RW	SDMMC remapping: 00: Default mapping (STS/CMD/PD2, SDCK/SLVCK/PC12, STR/PD3, D0/PC8, D1/PC9, D2/PC10, D3/PC11, D4/PA14, D5/PA15, D6/PC6, D7/PC7); 01: Remapping (STS/CMD/PD12, SDCK/SLVCK/PD11, STR/PD10, D0/PB13, D1/PC9, D2/PB10, D3/PB11, D4/PA14, D5/PA15, D6/PC6, D7/PC7); 1x: Remapping (STS/CMD/PC10, SDCK/SLVCK/PC12, STR/PC11, D0/PD0, D1/PD1, D2/PD2, D3/PD3, D4/PD4, D5/PPD5, D6/PD6, D7/PD7).	00b
[9:8]	UHSIF_PORT_RM[1:0]	RW	UHSIF PORT remapping: 00: Default mapping (PORT0/PF12, , PORT1/PF13 , PORT2/PE7, PORT3/PE8, PORT4/PE9, PORT5/PE10, ...) PORT6/PE11 PORT8/PE13 PORT7/PE12 , , PORT9/PE14 , PORT10/PE15 , PORT11/PB10 , PORT12/PB11 , PORT13/PB12 , PORT14/PB13 , PORT15/PB14 , PORT16/PD10 , PORT17/PD11 ,	00b

			PORT18/PD12 , PORT19/PD13 , PORT20/PD14 , PORT21/PPD15 , PORT22/PF0 , PORT23/PF1 , PORT24/PF2 , PORT25/PC6 , PORT26/PC7 , PORT27/PC8 , PORT28/PC9 , PORT29/PA13 , PORT30/PA14 , PORT31/PA15 , PORT32/PC10 , PORT33/PC11 , PORT34/PC12 , PORT35/PD0 , PORT36/PD1 , PORT37/PD2 , PORT38/PD3 , PORT39/PD4 , PORT40/PD5 , PORT41/PD6 , PORT42/PD7 , PORT43/PF3 , PORT44/PF4 , PORT45/PF5, PORT46/PE0, PORT47/PE1); 01: Remapping (PORT0/PF12, PORT1/PF13, PORT2/PE7, PORT3/PC1, PORT4/PC2, PORT5/PC3, PORT6/PB0, PORT7/PB1 , PORT8/PE13 , PORT9/PE14 , PORT10/PE15 , PORT11/PB10 , PORT12/PB11 , PORT13/PB12 , PORT14/PB13 , PORT15/PB14 , PORT16/PD10 , PORT17/PD11 , PORT18/PD12 , PORT19/PD13 , PORT20/PD14 , PORT21/PPD15 , PORT22/PF0 , PORT23/PF1 , PORT24/PF2 , PORT25/PC6 , PORT26/PC7 , PORT27/PC8 , PORT28/PC9 , PORT29/PA13 , PORT30/PA14 , PORT31/PA15 , PORT32/PC10 , PORT33/PC11 , PORT34/PC12 , PORT35/PD0 , PORT36/PD1 , PORT37/PD2 , PORT38/PD3 , PORT39/PD4 , PORT40/PD5 , PORT41/PD6 , PORT42/PD7 , PORT43/PF3 , PORT44/PF4 , PORT45/PF5 , PORT46/PE0, PORT47/PE1). 1x: Remapping (PORT0/PC1, PORT1/PC2, PORT2/PC3, PORT3/PB0, PORT4/PB1, PORT5/PE10, PORT6/PE11, PORT7/PE12 , PORT8/PE13 , PORT9/PE14 , PORT10/PE15 , PORT11/PB10 , PORT12/PB11 , PORT13/PB12 , PORT14/PB13 , PORT15/PB14 , PORT16/PD10 , PORT17/PD11 , PORT18/PD12 , PORT19/PD13 , PORT20/PD14 , PORT21/PPD15 , PORT22/PF0 , PORT23/PF1 , PORT24/PF2 , PORT25/PC6 , PORT26/PC7 , PORT27/PC8 , PORT28/PC9 , PORT29/PA13 , PORT30/PA14 , PORT31/PA15 , PORT32/PC10 , PORT33/PC11 , PORT34/PC12 , PORT35/PD0 , PORT36/PD1 , PORT37/PD2 , PORT38/PD3 , PORT39/PD4 , PORT40/PD5 , PORT41/PD6 , PORT42/PD7 , PORT43/PF3 , PORT44/PF4 , PORT45/PF5 , PORT46/PE0, PORT47/PE1). Note: For chips packaged with pins, it is recommended to use [special type of chip] configuration. 88 1xb The recommended pin configuration for the chip is [missing information].	

[7:6]	UHSIF_CLK_RM[1:0]	RW	UHSIF CLK remapping: 00: Default mapping (CLK/PF11); 01: Remapping (CLK/PC0); 10: Remapping (CLK/PF14); 11: Remapping (CLK/PD9).	00b
5	Reserved	RO is reserved.		0
4	ADC2_ETRGINJ_RM	RW	ADC2 externally triggered injection of the remapped bit of the conversion. 1: The ADC2 external trigger injection conversion is connected to TIM8_CH4; 0: ADC2 external trigger injection conversion is connected to EXTI15.	0
3	ADC2_ETRGREG_RM	RW	ADC2 external trigger rule conversion remapping bit. 1: The ADC2 external trigger rule conversion is connected to TIM8_TRGO; 0: ADC2 external trigger rule conversion is connected to EXTI11.	0
2	ADC1_ETRGINJ_RM	RW	ADC1 externally triggered injection of the remapped bit of the conversion. 1: The ADC1 external trigger injection conversion is connected to TIM8_CH4; 0: ADC1 external trigger injection conversion is connected to EXTI15.	0
1	ADC1_ETRGREG_RM	RW	ADC1 is the remapping bit for external trigger rule conversion. 1: The ADC1 external trigger rule conversion is connected to TIM8_TRGO; 0: ADC1 external trigger rule conversion is connected to EXTI11.	0
0	PD0PD1_RM	RW	Pin PD0 & PD1 remapping bit, which can be read and written by the user. It controls... Whether the GPIO functions of PD0 and PD1 are remapped, i.e. PD0 & PD1 are mapped to XI & XO. 1: The pin is used as a GPIO port; 0: The pin is used as a crystal oscillator pin.	0

9.3.3.2 **Offset address of the low byte of the port multiplexing function register (GPIOx_AFLR) (x=A/B/C/D/E/F) :** 0x04, 0x0C, 0x14, 0x1C, 0x24, 0x2C

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

Bit	name	access	describe	Reset value
[31:28] [27:24] [23:20] [19:16] [15:12] [11:8] [7:4] [3:0]	AFRy[3:0]	RW	(y=0-7), selection of multiplexing function for port x and pin y: 0000:AF0; 0001: AF1; 0010: AF2; 0011: AF3; 0100: AF4; 0101: AF5; 0110: AF6; 0111: AF7; 1000: AF8; 1001:AF9; 1010: AF10; 1011: AF11; 1100: AF12; 1101: AF13; 1110:AF14; 1111:AF15.	0

9.3.3.3 High-order bits of the port multiplexing function register (GPIOx_AFHR) (x=A/B/C/D/E/F)

Offset addresses: 0x08, 0x10, 0x18, 0x20, 0x28, 0x30

31	30	29	28	27	26	25	twenty four	twenty three	twenty two	twenty one	20	19	18	17	16
AFR15[3:0]				AFR14[3:0]								AFR13[3:0]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFR11[3:0]				AFR10[3:0]				AFR9[3:0]				AFR8[3:0]			

Bit	name	access	describe	Reset value
[31:28] [27:24] [23:20] [19:16] [15:12] [11:8] [7:4] [3:0]	AFRy[3:0]	RW	(y=8-15), selection of multiplexing function for port x and pin y: 0000:AF0; 0001: AF1; 0010: AF2; 0011: AF3; 0100: AF4; 0101: AF5; 0110: AF6; 0111: AF7; 1000: AF8; 1001:AF9; 1010: AF10; 1011: AF11; 1100: AF12; 1101: AF13; 1110:AF14; 1111:AF15.	0

9.3.2.4 External Interrupt Configuration Register 1 (AFIO_EXTICR1)

Offset Address: 0x3C

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
EXTI7[3:0]				EXTI6[3:0]							EXTI5[3:0]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI3[3:0]				EXTI2[3:0]				EXTI1[3:0]				EXTI0[3:0]			

Bit	name	access		Reset value
[31:28] [27:24] [23:20] [19:16] [15:12] [11:8] [7:4] [3:0]	EXTIx[3:0]	RW	Description (x=0-7), External Interrupt Input Pin Configuration Bits. Used to determine... Determine which port pin the external interrupt pin is mapped to: 0000: The xth pin of the PA pin; 0001: The xth pin of the PB pin; 0010: The xth pin of the PC pin; 0011: The xth pin of the PD pin; 0100: The xth pin of the PE pin; 0101: The xth pin of the PF pin; 0110: CMP output; Other: Reserved.	0000b

9.3.2.5 External Interrupt Configuration Register 2

(AFIO_EXTICR2) Offset Address: 0x40

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
EXTI15 [3:0]				EXTI14[3:0]							EXTI13[3:0]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

EXTI11[3:0]	EXTI10[3:0]	EXTI9[3:0]	EXTI8[3:0]

Bit	name	access	describe	Reset value
[31:28] [27:24] [23:20] [19:16] [15:12] [11:8] [7:4] [3:0]	EXTIx[3:0]	RW	(x=8-15), External interrupt input pin configuration bits. Used for... Determine which port pin the external interrupt pin is mapped to: 0000: The xth pin of the PA pin; 0001: The xth pin of the PB pin; 0010: The xth pin of the PC pin; 0011: The xth pin of the PD pin; 0100: The xth pin of the PE pin; 0101: The xth pin of the PF pin; 0110: CMP output; Other: Reserved.	0000b

Chapter 10 Direct Memory Access Control (DMA)

Direct Memory Access (DMA) controllers provide high-speed data transfer between peripherals and memory, or between memory devices.

In this way, without CPU intervention, data can be moved quickly via DMA to save CPU resources for other operations.

The DMA controller supports reconfiguration of DMA request lines between peripherals and the DMA. An arbitrator is also included to coordinate priority among channels.

class.

10.1 Key Features

Multiple independent configurable channels

Supports reconfiguration of DMA request lines between peripherals and DMA, and supports software triggering.

ÿ Supports cyclic buffer management

The request priority among multiple channels can be set via software programming (highest, high, medium, and low). When the priority settings are equal ...

The priority is determined by the channel number (the lower the channel number, the higher the priority).

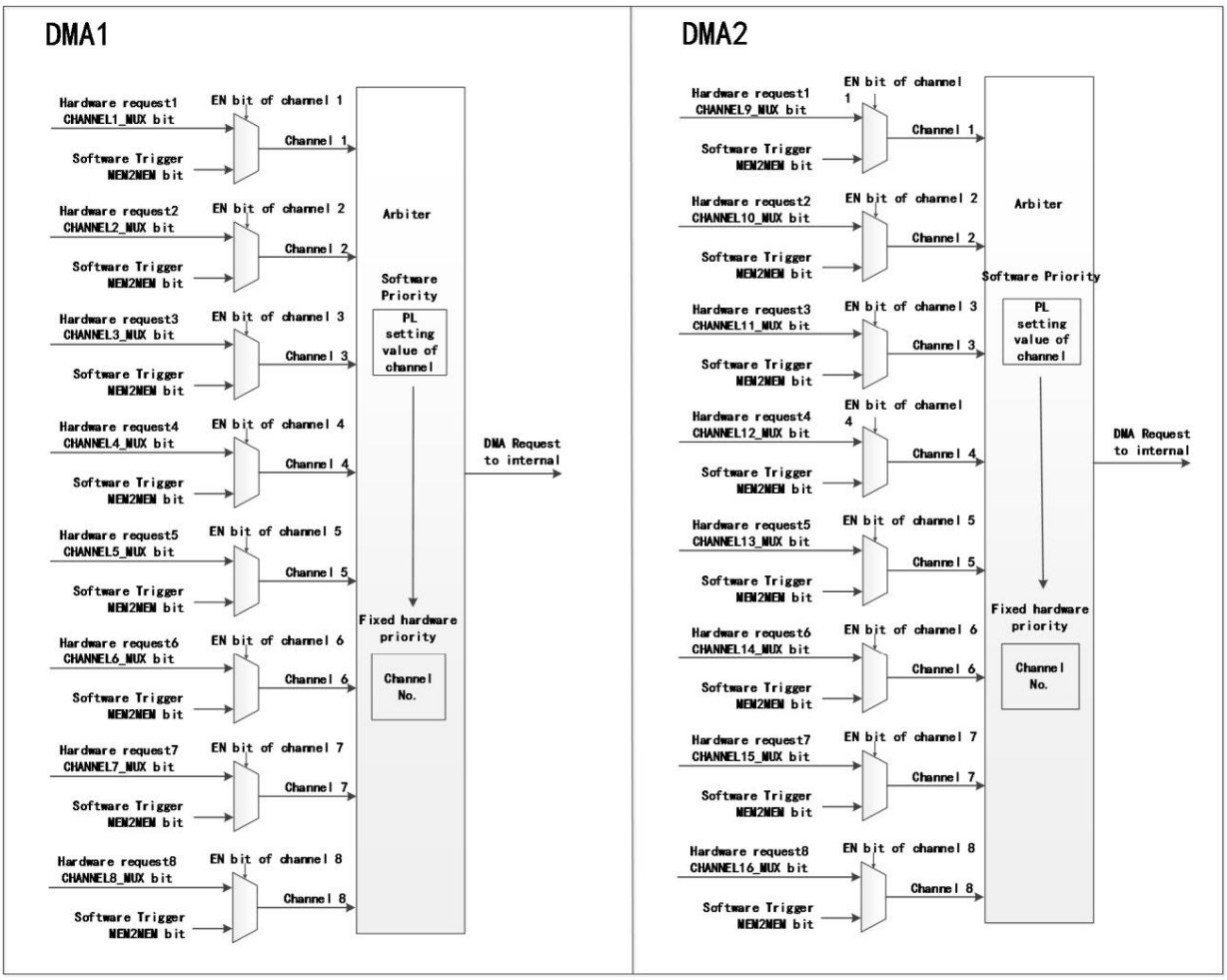
Supports data transfer between peripherals and memory, memory and peripherals, and memory and memory.

Flash memory, SRAM, peripheral SRAM, and HB peripherals can all be used as sources and destinations for access.

Programmable data transfer bytes: Maximum 65535

10.2 Functional Description

Figure 10-1 Schematic diagram of DMA channel operation



10.2.1 DMA Channel Processing 1)

Arbitration Priority

Multiple independent DMA requests from multiple channels are input to the DMA controller via a logic OR structure. Only one channel's request will be processed at any given time
A response is received. The arbitrator inside the module selects which peripheral/memory access to initiate based on the priority of the channel request.

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

2) DMA Configuration:

When the DMA controller receives a request signal, it accesses the requesting peripheral or memory, and establishes the connection between the peripheral or memory and the storage.
Data transfer between devices. This mainly includes the following three steps:

1) Retrieving data from the peripheral data register or the memory address indicated by the current peripheral/memory address register; the initial transfer...
The starting address is the peripheral base address or memory address specified by the DMA_PADDRx or DMA_MADDRx register.

2) Store data in the peripheral data register or the memory address indicated by the current peripheral/memory address register, during the first transfer.
The starting address is the peripheral base address or memory address specified by the DMA_PADDRx or DMA_MADDRx register. 3) Perform a decrement operation on the value in the DMA_CNTRx register, which indicates the number of operations that have not yet been completed.

Each channel includes 3 DMA data transfer methods:

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

Note: Memory-to-memory mode does not require an external request signal. After configuring this mode (MEM2MEM=1), data transmission can be started by opening the channel (EN=1). This mode does not support loop mode.

The configuration process is as follows:

1) Set the starting address of the peripheral register or the memory-to-memory mode (MEM2MEM=1) in the DMA_PADDRx register.
Data address. This address will be the source or destination address for data transfer when a DMA request occurs.

2) Select the memory address using the FLAG_CUR_MEM bit and set the memory data address in the DMA_MADDRx register. DMA occurs.
When a request is made, the data to be transmitted 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 channel priority in the PL[1:0] bits of the DMA_CFGRx register. 5) Set the data transfer direction, loop mode, incremental mode for peripherals and memory, data width for peripherals and memory, halfway through transfer, transfer complete, and transfer error interrupt enable bits in the DMA_CFGRx register. 6) Set the EN bit of the DMA_CFGRx register to start channel x.

Note: The data transfer direction (DIR), loop mode (position), and incremental mode (MINC/PINC) of peripherals and memory in the DMA_PADDRx/DMA_MADDRx/DMA_CNTRx registers and the DMA_CFGRx register can only be configured for DMA writing when the channel is closed.

3) Loop mode

Setting the CIRC bit of the DMA_CFGRx register to 1 enables the cyclic mode function for channel data transfer. In cyclic mode, when the number of data transfers becomes 0, the contents of the DMA_CNTRx register are automatically reloaded to their initial value, and the internal peripheral and memory address registers are also reloaded to the initial address values set by the DMA_PADDRx and DMA_MADDRx registers. DMA operations will continue until the channel is closed or DMA mode is disabled.

4) DMA processing status

ÿ Halfway through transfer: The HTIFx bit in the corresponding DMA_INTFR register is set by hardware. This occurs when the number of bytes transferred by the DMA is reduced to the initial setting.

A value below half will trigger a DMA transfer completion flag. If HTIE is set in the DMA_CFGRx register, a partial transfer will occur.

Disconnect. This flag indicates to the application that the hardware is preparing for a new round of data transfer.

• Transfer complete: The TCIFx bit in the corresponding DMA_INTFR register is set by hardware. A completion message will appear when the number of bytes transferred by the DMA reaches 0.

The DMA transfer completion flag is generated. If TCIE is set in the DMA_CFGRx register, an interrupt will be generated.

• Transfer error: The TEIFx bit in the corresponding DMA_INTFR register is set by hardware. Reading or writing to a reserved address area will result in a transfer error.

DMA transfer error. The module hardware will automatically clear the EN bit of the DMA_CFGRx register corresponding to the channel where the error occurred.

The channel is closed. An interrupt will be generated if TEIE is set in the DMA_CFGRx register.

When an application queries the DMA channel status, it can first access the GIFx bit of the DMA_INTFR register to determine which channel is currently active.

A DMA event occurred on the channel, and then the specific DMA event content of that channel was processed.

10.2.2 Programmable total data transfer size/data bit width/alignment

The total amount of data transferred per channel in a round of DMA is programmable, with a maximum of 65535 transfers. The DMA_CNTRx register indicates the data to be transferred.

Number of bytes. When EN=0, the setting value is written; when EN=1 and the DMA transfer channel is enabled, this register becomes read-only.

The value decreases after transmission.

The data transfer values between peripherals and memory support automatic address pointer increment, with programmable pointer increments. The first address they access...

The data address to be transferred is stored in the DMA_PADDRx and DMA_MADDRx registers, which is determined by setting the PINC bit of the DMA_CFGRx register.

Alternatively, setting MINC to position 1 allows you to enable either the peripheral address auto-increment mode or the memory address auto-increment mode, respectively. PSIZE[1:0] sets the peripheral address.

The data size and address increment/decrement are set by MSIZE[1:0], which sets the memory address and allows you to retrieve the data size and address increment/decrement. There are four options:

8-bit, 16-bit, 32-bit, and 256-bit. The specific data transfer methods are shown in the table below:

Table 10-1 DMA Transfer under Different Data Bit Widths (PINC=MINC=1)

Source Bit width	Target Bit width	transmission number	Source: address/data	Destination: address/data	Transmission operation
8	8	4	0x00/B0 0x01/B1 0x02/B2 0x03/B3	0x00/B0 0x01/B1 0x02/B2 0x03/B3	Source address increment and source setting Set data bit width alignment, value retrieval Size equal to the source data bit width ÿ Target address increment and target setting Set data bit width alignment and retrieve values. Size equal to target data bit width ÿ DMA transfer of data sent to the target end Based on the principle: Insufficient data size High-order bits are padded with 0s; data size overflows. Remove the position Data storage method: Little-endian In this format, the low address stores the low byte. The high address stores the high byte.
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	
8	256	4	0x00/B0 0x01/B1 0x02/B2 0x03/B3	0x00/00…0000B0 0x20/00…0000B1 0x40/00…0000B2 0x60/00…0000B3	
16	8	4	0x00/B1B0 0x02/B3B2 0x04/B5B4 0x06/B7B6	0x00/B0 0x01/B2 0x02/B4 0x03/B6	
16	16	4	0x00/B1B0 0x02/B3B2	0x00/B1B0 0x02/B3B2	

			0x04/B5B4 0x06/B7B6	0x04/B5B4 0x06/B7B6	
16	32	4	0x00/B1B0 0x02/B3B2 0x04/B5B4 0x06/B7B6	0x00/0000B1B0 0x04/0000B3B2 0x08/0000B5B4 0x0C/0000B7B6	
16	256	4	0x00/B1B0 0x02/B3B2 0x04/B5B4 0x06/B7B6	0x00/00…00B1B0 0x20/00…00B3B2 0x40/00…00B5B4 0x60/00…00B7B6	
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	
32	256	4	0x00/B3B2B1B0 0x04/B7B6B5B4 0x08/BBBAB9B8 0x0C/BFBEBDBC	0x00/00…B3B2B1B0 0x20/00…B7B6B5B4 0x40/00…BBBAB9B8 0x60/00…BFBEBDBC	
256	8	4	0x00/2F…13121110 0x20/4F…33323130 0x40/6F…53525150 0x60/8F…73727170	0x00/10 0x01/30 0x02/50 0x03/70	
256	16	4	0x00/2F…13121110 0x20/4F…33323130 0x40/6F…53525150 0x60/8F…73727170	0x00/1110 0x02/3130 0x04/5150 0x06/7170	
256	32	4	0x00/2F…13121110 0x20/4F…33323130 0x40/6F…53525150 0x60/8F…73727170	0x00/13121110 0x04/33323130 0x08/53525150 0x0C/73727170	
256 256		4	0x00/2F…13121110 0x20/4F…33323130 0x40/6F…53525150 0x60/8F…73727170	0x00/2F…13121110 0x20/4F…33323130 0x40/6F…53525150 0x60/8F…73727170	

10.2.3 DMA Request Multiplexer

The DMAMUX request multiplexer can reconfigure (route) DMA request lines between the chip's peripherals and the DMA controller. This routing function...

This is ensured by a programmable multi-channel DMA request line multiplexer. Each channel can select one DMA request line without restriction, or...

The device selects a DMA request line in a manner synchronized with events from its DMAMUX synchronization input.

The DMA controller provides 16 channels, of which DMAMUX channels 1 to 8 are connected to DMA1 channels 1 to 8, and DMAMUX channels 9 to...

16 is connected to DMA2 channels 1 to 8.

Table 10-2 DMA Multiplexer Input to Resource Allocation Table

DMA request for input peripheral		DMA request for input peripheral		DMA request for input peripheral	
1	TIM1_CH1	42	TIM9_CH4	83	I3C_TX
2	TIM1_CH2	43	TIM9_UP	84	I3X_RX
3	TIM1_CH3	44	TIM9_TRIG	85	USART1_TX
4	TIM1_CH4	45	TIM10_CH1	86	USART1_RX
5	TIM1_UP	46	TIM10_CH2	87	USART2_TX
6	TIM1_COM	47	TIM10_CH3	88	USART2_RX
7	TIM1_TRIG	48	TIM10_CH4	89	USART3_TX
8	TIM2_CH1	49	TIM10_UP	90	USART3_RX
9	TIM2_CH2	50	TIM10_TRIG	91	USART4_TX
10	TIM2_CH3	51	TIM11_CH1	92	USART4_RX
11	TIM2_CH4	52	TIM11_CH2	93	USART5_TX
12	TIM2_UP	53	TIM11_CH3	94	USART5_RX
13	TIM2_TRIG	54	TIM11_CH4	95	USART6_TX
14	TIM3_CH1	55	TIM11_UP	96	USART6_RX
15	TIM3_CH2	56	TIM11_TRIG	97	USART7_TX
16	TIM3_CH3	57	TIM12_CH1	98	USART7_RX
17	TIM3_CH4	58	TIM12_CH2	99	USART8_TX
18	TIM3_UP	59	TIM12_CH3	100	USART8_RX
19	TIM3_TRIG	60	TIM12_CH4	101	SWPMI_TX
20	TIM4_CH1	61	TIM12_UP	102	SWPMI_RX
twenty one	TIM4_CH2	62	TIM12_TRIG	103	DAC1
twenty two	TIM4_CH3	63	SPI1_TX	104	DAC2
twenty three	TIM4_CH4	64	SPI1_RX	105	reserve
twenty four	TIM4_UP	65	SPI2_TX	106	reserve
25	TIM4_TRIG	66	SPI2_RX	107	DFSDM_DMA0
26	TIM5_CH1	67	SPI3_TX	108	DFSDM_DMA1
27	TIM5_CH2	68	SPI3_RX	109	Reserved
28	TIM5_CH3	69	SPI4_TX	110	reserve
29	TIM5_CH4	70	SPI4_RX	111	SDIO
30	TIM5_UP	71	QSPI1_DMA	112	SAI_A_TX
31	TIM5_TRIG	72	QSPI2_DMA	113	SAI_A_RX
32	TIM8_CH1	73	I2C1_TX	114	SAI_B_TX
33	TIM8_CH2	74	I2C1_RX	115	SAI_B_RX
34	TIM8_CH3	75	I2C2_TX	116	Reserved
35	TIM8_CH4	76	I2C2_RX	117	reserve
36	TIM8_UP	77	I2C3_TX	118	reserve
37	TIM8_COM	78	I2C3_RX	119	reserve
38	TIM8_TRIG	79	I2C4_TX	120	ADC1

39	TIM9_CH1	80	I2C4_RX	121	ADC2
40	TIM9_CH2	81	I3C_RS	122	TIM6_UP
41	TIM9_CH3	82	I3C_TC	123	TIM7_UP

10.2.4 Double-Buffered Mode

Double-buffered mode is used for DMA1 and DMA2 channels. Besides having two memory pointers, the channel operation in double-buffered mode...

The formula remains the same as that of the single-buffered mode.

By setting the DOUBLE_MODE bit in the DMA_CFGRx register to 1, double-buffered mode is enabled, automatically entering loop mode, and...

After completing one cycle, the memory address is automatically switched (alternating between DMA_MADDRx and DMA_M1ADDRx).

At the end of each loop, the DMA controller swaps the memory target from one memory target to another, thus allowing the software to process...

While using one memory region, DMA transfers can also fill or use a second memory region. Dual-buffered channels can operate bidirectionally, such as...

As shown in the table below.

Table 10-3 Source and Destination Address Registers in Double-Buffered Mode

DMA_CFGRx.DIR	direction	Source address	Target address
0	Read from peripherals	DMA_PADDRx	DMA_MADDRx/DMA_M1ADDRx
1	Read DMA_MADDRx/DMA_M1ADDRx from memory		DMA_PADDRx

Note: When double buffering mode is enabled, memory to memory mode is disabled.

10.3 Register Description

Table 10-4 List of DMA1 Related Registers

name	Access Address Description: 0x40020000 DMA1	Reset value
R32_DMA1_INTFR	Interrupt Status Register; 0x40020004 DMA1 Interrupt	0x00000000
R32_DMA1_INTFCR	Flag Clear Register; 0x40020008 DMA1 Channel 1	0x00000000
R32_DMA1_CFGR1	Configuration Register; 0x4002000C DMA1 Channel 1	0x00000000
R32_DMA1_CNTR1	Data Transfer Count Register; 0x40020010 DMA1 Channel 1 Peripheral	0x00000000
R32_DMA1_PADDR1	Address Register; 0x40020014 DMA1 Channel 1 Memory	0x00000000
R32_DMA1_MADDR1	Address Register; 0x40020018 DMA1 Channel 1 Memory Address	0x00000000
R32_DMA1_M1ADDR1	Register 1; 0x4002001C DMA1 Channel 2 Configuration Register;	0x00000000
R32_DMA1_CFGR2	0x40020020 DMA1 Channel 2 Data Transfer Count	0x00000000
R32_DMA1_CNTR2	Register; 0x40020024 DMA1 Channel 2 Peripheral Address Register;	0x00000000
R32_DMA1_PADDR2	0x40020028 DMA1 Channel 2 Memory Address Register;	0x00000000
R32_DMA1_MADDR2	0x4002002C DMA1 Channel 2 Memory Address Register 1.	0x00000000
R32_DMA1_M1ADDR2	0x40020030 DMA1 Channel 3 Configuration Register; 0x40020034	0x00000000
R32_DMA1_CFGR3	DMA1 Channel 3 Data Transfer Count Register;	0x00000000
R32_DMA1_CNTR3	0x40020038 DMA1 Channel 3 Peripheral Address Register;	0x00000000
R32_DMA1_PADDR3	0x4002003C DMA1 Channel 3 Memory Address Register;	0x00000000
R32_DMA1_MADDR3	0x40020040 DMA1 Channel 3 Memory Address Register 1;	0x00000000
R32_DMA1_M1ADDR3	0x40020044 DMA1 Channel 4 Configuration Register; 0x40020048	0x00000000
R32_DMA1_CFGR4	DMA1 Channel 4 Data Transfer Count Register;	0x00000000
R32_DMA1_CNTR4	0x4002004C DMA1 Channel 4 Peripheral Address Register;	0x00000000
R32_DMA1_PADDR4	0x40020050 DMA1 Channel 4 Memory Address Register;	0x00000000
R32_DMA1_MADDR4	0x40020054 DMA1 Channel 4 Memory Address Register 1	0x00000000
R32_DMA1_M1ADDR4		0x00000000

R32_DMA1_CFGR5	0x40020058	DMA1 Channel 5 Configuration Register;	0x00000000
R32_DMA1_CNTR5	0x4002005C	DMA1 Channel 5 Data Transfer Count Register;	0x00000000
R32_DMA1_PADDR5	0x40020060	DMA1 Channel 5 Peripheral Address Register;	0x00000000
R32_DMA1_MADDR5	0x40020064	DMA1 Channel 5 Memory Address Register;	0x00000000
R32_DMA1_M1ADDR5	0x40020068	DMA1 Channel 5 Memory Address Register 1;	0x00000000
R32_DMA1_CFGR6	0x4002006C	DMA1 Channel 6 Configuration Register;	0x00000000
R32_DMA1_CNTR6	0x40020070	DMA1 Channel 6 Data Transfer Count Register;	0x00000000
R32_DMA1_PADDR6	0x40020074	DMA1 Channel 6 Peripheral Address Register;	0x00000000
R32_DMA1_MADDR6	0x40020078	DMA1 Channel 6 Memory Address Register;	0x00000000
R32_DMA1_M1ADDR6	0x4002007C	DMA1 Channel 6 Memory Address Register 1;	0x00000000
R32_DMA1_CFGR7	0x40020080	DMA1 Channel 7 Configuration Register;	0x00000000
R32_DMA1_CNTR7	0x40020084	DMA1 Channel 7 Data Transfer Count Register	0x00000000
R32_DMA1_PADDR7	0x40020088	DMA1 Channel 7 Peripheral Address Register	0x00000000
R32_DMA1_MADDR7	0x4002008C	DMA1 Channel 7 Memory Address Register	0x00000000
R32_DMA1_M1ADDR7	0x40020090	DMA1 Channel 7 Memory Address Register 1	0x00000000
R32_DMA1_CFGR8	0x40020094	DMA1 Channel 8 Configuration Register	0x00000000
R32_DMA1_CNTR8	0x40020098	DMA1 Channel 8 Data Transfer Count Register	0x00000000
R32_DMA1_PADDR8	0x4002009C	DMA1 Channel 8 Peripheral Address Register	0x00000000
R32_DMA1_MADDR8	0x400200A0	DMA1 Channel 8 Memory Address Register	0x00000000
R32_DMA1_M1ADDR8	0x400200A4	DMA1 Channel 8 Memory Address Register 1	0x00000000

Table 10-5 List of DMA2 Related Registers

name	Access address	description	Reset value
R32_DMA2_INTFR	0x40020400	DMA2 Interrupt Status Register;	0x00000000
R32_DMA2_INTFCR	0x40020404	DMA2 Interrupt Flag Clear Register;	0x00000000
R32_DMA2_CFGR1	0x40020408	DMA2 Channel 1 Configuration Register;	0x00000000
R32_DMA2_CNTR1	0x4002040C	DMA2 Channel 1 Data Transfer Count Register;	0x00000000
R32_DMA2_PADDR1	0x40020410	DMA2 Channel 1 Peripheral Address Register;	0x00000000
R32_DMA2_MADDR1	0x40020414	DMA2 Channel 1 Memory Address Register;	0x00000000
R32_DMA2_M1ADDR1	0x40020418	DMA2 Channel 1 Memory Address Register 1;	0x00000000
R32_DMA2_CFGR2	0x4002041C	DMA2 Channel 2 Configuration Register;	0x00000000
R32_DMA2_CNTR2	0x40020420	DMA2 Channel 2 Data Transfer Count Register;	0x00000000
R32_DMA2_PADDR2	0x40020424	DMA2 Channel 2 Peripheral Address Register;	0x00000000
R32_DMA2_MADDR2	0x40020428	DMA2 Channel 2 Memory Address Register;	0x00000000
R32_DMA2_M1ADDR2	0x4002042C	DMA2 Channel 2 Memory Address Register 1.	0x00000000
R32_DMA2_CFGR3	0x40020430	DMA2 Channel 3 Configuration Register;	0x00000000
R32_DMA2_CNTR3	0x40020434	DMA2 Channel 3 Data Transfer Count Register;	0x00000000
R32_DMA2_PADDR3	0x40020438	DMA2 Channel 3 Peripheral Address Register;	0x00000000
R32_DMA2_MADDR3	0x4002043C	DMA2 Channel 3 Memory Address Register;	0x00000000
R32_DMA2_M1ADDR3	0x40020440	DMA2 Channel 3 Memory Address Register 1;	0x00000000
R32_DMA2_CFGR4	0x40020444	DMA2 Channel 4 Configuration Register;	0x00000000
R32_DMA2_CNTR4	0x40020448	DMA2 Channel 4 Data Transfer Count Register;	0x00000000
R32_DMA2_PADDR4	0x4002044C	DMA2 Channel 4 Peripheral Address Register;	0x00000000
R32_DMA2_MADDR4	0x40020450	DMA2 Channel 4 Memory Address Register	0x00000000

R32_DMA2_M1ADDR4	0x40020454 DMA2 Channel 4 Memory Address Register 1	0x00000000
R32_DMA2_CFGR5	0x40020458 DMA2 Channel 5 Configuration	0x00000000
R32_DMA2_CNTR5	Register 0x4002045C DMA2 Channel 5 Transfer Count	0x00000000
R32_DMA2_PADDR5	Register 0x40020460 DMA2 Channel 5 Peripheral	0x00000000
R32_DMA2_MADDR5	Address Register 0x40020464 DMA2 Channel 5 Memory	0x00000000
R32_DMA2_M1ADDR5	Address Register 0x40020468 DMA2 Channel 5 Memory	0x00000000
R32_DMA2_CFGR6	Address Register 1 0x4002046C DMA2 Channel	0x00000000
R32_DMA2_CNTR6	6 Configuration Register 0x40020470 DMA2 Channel 6	0x00000000
R32_DMA2_PADDR6	Transfer Count Register 0x40020474 DMA2 Channel 6	0x00000000
R32_DMA2_MADDR6	Peripheral Address Register 0x40020478 DMA2 Channel	0x00000000
R32_DMA2_M1ADDR6	6 Memory Address Register 0x4002047C DMA2 Channel 6	0x00000000
R32_DMA2_CFGR7	Memory Address Register 1 0x40020480 DMA2	0x00000000
R32_DMA2_CNTR7	Channel 7 Configuration register 0x40020484 DMA2 Channel	0x00000000
R32_DMA2_PADDR7	7 Data Transfer Count Register 0x40020488 DMA2	0x00000000
R32_DMA2_MADDR7	Channel 7 Peripheral Address Register 0x4002048C DMA2	0x00000000
R32_DMA2_M1ADDR7	Channel 7 Memory Address Register 0x40020490 DMA2	0x00000000
R32_DMA2_CFGR8	Channel 7 Memory Address Register 1 0x40020494	0x00000000
R32_DMA2_CNTR8	DMA2 Channel 8 Configuration register 0x40020498 DMA2	0x00000000
R32_DMA2_PADDR8	Channel 8 Data Transfer Count Register 0x4002049C	0x00000000
R32_DMA2_MADDR8	DMA2 Channel 8 Peripheral Address Register 0x400204A0	0x00000000
R32_DMA2_M1ADDR8	DMA2 Channel 8 Memory Address Register 0x400204A4 DMA2 Channel 8 Memory Address Register 1	0x00000000

Table 10-6 DMAMUX Register List

name	Access address	describe	Reset value
R32_DMAMUX1_4_CFGR	0x40020800	DMA request multiplexer channel 1-4 configuration register	0x00000000
R32_DMAMUX5_8_CFGR	0x40020804	DMA request multiplexer channel 5-8 configuration register	0x00000000
R32_DMAMUX9_12_CFGR	0x40020808	DMA request multiplexer channel 9-12 configuration register	0x00000000
R32_DMAMUX13_16_CFGR	0x4002080C	DMA request multiplexer channel 13-16 configuration register	0x00000000

10.3.1 DMAx Interrupt Status Register (DMAx_INTFR) (x=1/2)

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TEIF8	HTIF8	TCIF8	GIF8	TEIF7	HTIF7	TCIF7	GIF7	TEIF6	HTIF6	TCIF6	GIF6	TEIF5	HTIF5	TCIF5	GIF5

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEIF4	HTIF4	TCIF4	GIF4	TEIF3	HTIF3	TCIF3	GIF3	TEIF2	HTIF2	TCIF2	GIF2	TEIF1	HTIF1	TCIF1	GIF1

Bit	Name access			Reset value
31/27/23/19/15/11/7/3	TEIFx	RO	Description of transmission error flags for channel x (x=1/2/3/4/5/6/7/8): 1: A transmission error occurred on channel x; 0: No transmission errors on channel x. The hardware sets the flag, and the software writes the CTEIFx bit to clear this flag.	0
30/26/22/18/14/10/	HTIFx	RO	The transmission halfway mark for channel x (x=1/2/3/4/5/6/7/8): 1: A half-transmission event occurred on channel x;	0

6/2			0: No more than half of the data has been transmitted on channel x. The hardware sets this flag, while the software writes the CHTIFx bit to clear it.	
29/25/21/17/13/9/5/1	TCIFx	RO	Transmission completion flags for channel x (x=1/2/3/4/5/6/7/8): 1: A transmission completion event was generated on channel x; 0: No transmission completion event occurred on channel x. The hardware sets the flag, while the software writes the CTCIFx bit to clear it.	0
28/24/20/16/12/8/4/0	GIFx	RO	Global interrupt flags for channel x (x=1/2/3/4/5/6/7/8): 1: TEIFx, HTIFx, or TCIFx were generated on channel x; 0: No TEIFx, HTIFx, or TCIFx occurred on channel x. The hardware sets the flag, while the software writes the CGIFx bit to clear it.	0

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

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

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
CTEIF8	CHTIF8	CTCIF8	CGIF8	CTEIF7	CHTIF7	CTCIF7	CGIF7	CTEIF6	CHTIF6	CTCIF6	CGIF6	CTEIF5	CHTIF5	CTCIF5	CGIF5

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTEIF4	CHTIF4	CTCIF4	CGIF4	CTEIF3	CHTIF3	CTCIF3	CGIF3	CTEIF2	CHTIF2	CTCIF2	CGIF2	CTEIF1	CHTIF1	CTCIF1	CGIF1

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

10.3.3 DMAy Channel x Configuration Register (DMAy_CFGRx) (x=1/2/3/4/5/6/7/8, y=1/2)

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

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														FLAG_CUR_MEM

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

Bit	Name access		describe	Reset value
[31:17] Reserved		RO is reserved.		0
16	FLAG_CUR_MEM	RW	Memory address selection settings; the hardware will automatically switch in double-buffered mode. 1: M1ADDR (addressed using the DMA_M1ADDRx pointer); 0: MADDR (addressed using DMA_MADDRx pointer). Note: This bit is only used in double-buffered mode. After the channel is enabled, this bit acts as a status flag to indicate the current target storage. district.	0
15	DOUBLE_MODE	RW	Double buffering mode enabled: 1: Enabling double buffering mode will automatically enter loop mode, completing one cycle. It will switch memory addresses; 0: Disable double buffering mode.	0
14	MEM2MEM	RW	Memory-to-memory mode enabled: 1: Enable memory-to-memory data transfer mode; 0: Non-memory to memory data transfer.	0
[13:12] PL[1:0]		RW	Channel priority settings: 00: Low; 01: Medium; 10: high; 11: highest.	00b
[11:10] MSIZE[1:0]		RW	Memory address data width settings: 00: 8 bits; 01: 16 bits; 10: 32 bits; 11: 256 bits.	00b
[9:8] PSIZE[1:0]		RW	Peripheral address data width settings: 00: 8 bits; 01: 16 bits; 10: 32 bits; 11: 256 bits.	00b
7	MINC	RW	Memory address increment mode enabled: 1: Enable memory address increment operation; 0: Operation where the memory address remains unchanged.	0
6	PINC	RW	Peripheral address increment mode enabled: 1: Enable peripheral address increment operation; 0: Operation where the peripheral address remains unchanged.	0
5	CIRC	RW	DMA channel loop mode enabled: 1: Enable loop operation; 0: Perform a single operation.	0
4	DIR	RW	Data transmission direction: 1: Read from memory; 0: Read from peripheral device.	0
3	TEIE	RW	Transmission error interrupt enable control: 1: Enable transmission error interruption; 0: Prevent transmission errors from interrupting the transmission.	0
2	HTIE	RW	Transmission halfway interrupt enable control: 1: Enable interrupt when half of the transmission has occurred; 0: Do not interrupt transmission when half of the transmission has occurred.	0
1	TCIE	RW	Transmission completion interrupt enable control: 1: Enable transmission completion interrupt; 0: Disallow interruption upon completion of transmission.	0

0	EN	RW	Channel enable control: 1: Channel enabled; 0: Channel closed. When a DMA transfer error occurs, the hardware automatically clears this bit to 0, disabling the channel. road.	0

10.3.4 DMAy Channel x Data Transfer Count Register (DMAy_CNTRx) (x=1/2/3/4/5/6/7/8,

Offset

address: 0x0C + (x-1)*20 + (y-1)*0x400 (y=1/2)

31	30 29 28	27 26 25 24 23 22 21	2019	18	17 16
Reserved					

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

Bit	Name access		describe	Reset value
[31:16] Reserved		RO is reserved.		0
[15:0] NDT[15:0]		RW	Number of data transfers, ranging from 0 to 65535. Indicates the remaining number of data points to be transferred (register contents in each DMA transfer). (Then decreasing). When the channel is in loop mode, the register contents will be automatically incremented. The value is set to the previously configured value.	0

Note: Indicates the current number of data to be transferred. This register can only be modified when EN=1, and is not writable. When the register content is empty, there is no... EN=0 No data transmission will occur regardless of whether the channel is open or not.

10.3.5 DMAy Channel x Peripheral Address Register (DMAy_PADDRx) (x=1/2/3/4/5/6/7/8, y=1/2) Offset Address: 0x10 + (x-1)*20 + (y-1)*0x400

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

Bit	Name access		Describes	Reset value
[31:0] PA[31:0]		RW	the peripheral base address, which serves as the source or destination address for peripheral data transmission. When PSIZE[1:0]='01' (16 bits), the module automatically ignores bit 0. The operation address is automatically aligned to 2 bytes; when PSIZE[1:0]='10' (32 (bits), the module automatically ignores bits [1:0], and the operation address is automatically 4 bytes. Alignment. When PSIZE[1:0]='11' (256-bit), the module automatically ignores... Omitted bits[4:0], the operation address is automatically aligned to 32 bytes.	0

10.3.6 DMAy Channel x Memory Address Register (DMAy_MADDRx) (x=1/2/3/4/5/6/7/8, y=1/2) Offset Address: 0x14 + (x-1)*20 + (y-1)*0x400

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

Bit	Name access		describe	Reset value

[31:0] MA[31:0]		RW	The memory data address serves as the source or destination address for data transfer. When MSIZE[1:0]='01' (16 bits), the module automatically ignores bit 0. The operation address is automatically aligned to 2 bytes; when MSIZE[1:0]='10' (32 bytes), the address is automatically aligned to 2 bytes.	0
			(bits), the module automatically ignores bits [1:0], and the operation address is automatically 4 bytes. Alignment. When MSIZE[1:0]='11' (256 bits), the module automatically ignores... Omitted bits[4:0], the operation address is automatically aligned to 32 bytes.	

10.3.7 DMAy Channel x Memory Address Register 1 (DMAy_M1ADDRx) (x=1/2/3/4/5/6/7/8, y=1/2) Offset Address: 0x18 + (x-1)*20 + (y-1)*0x400

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

M1A[31:0]

Bit	Name access			Reset value
[31:0] M1A[31:0]		RW	Describes the memory data address, serving as the source or destination address for data transfer. When MSIZE[1:0]='01' (16 bits), the module automatically ignores bit 0. The operation address is automatically aligned to 2 bytes; when MSIZE[1:0]='10' (32 bytes), the address is automatically aligned to 2 bytes.	0
			(bits), the module automatically ignores bits [1:0], and the operation address is automatically 4 bytes. Alignment. When MSIZE[1:0]='11' (256 bits), the module automatically ignores... Omitted bits[4:0], the operation address is automatically aligned to 32 bytes.	

10.3.8 DMA Request Multiplexer Channels 1-4 Configuration Registers (DMAMUX1_4_CFGR)

Offset address: 0x00

31 30 29 28 27 26 25 24 23 22 21 2019 18 17 16

Reserved	CHANNEL4_MUX[6:0]	Reserved	CHANNEL3_MUX[6:0]

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

Reserved	CHANNEL2_MUX[6:0]	Reserved	CHANNEL1_MUX[6:0]

Bit	name	access	describe	Reset value
31	Reserved	RO	is reserved.	0
[30:24]	CHANNEL4_MUX[6:0]	RW	DMAMUX Multiplexer channel 4 Input to resource allocation.	0
23	Reserved	RO	is reserved.	0
[22:16]	CHANNEL3_MUX[6:0]	RW	DMAMUX Multiplexer channel 3 Input to resource allocation.	0
15	Reserved	RO	is reserved.	0
[14:8]	CHANNEL2_MUX[6:0]	RW	DMAMUX Multiplexer channel 2 input to resource allocation.	0
7	Reserved	RO	is reserved.	0
[6:0]	CHANNEL1_MUX[6:0]	RW	DMAMUX multiplexer channel 1 input to resource allocation: 0000000: DMA request input 1; 0000001: DMA request input 2; 0000010: DMA request input 3; ... 1111010: DMA request input 123;	0

			Other: Reserved.	

10.3.9 DMA Request Multiplexer Channels 5-8 Configuration Registers (DMAMUX5_8_CFGR)

Offset address: 0x04

31	30 29 28	27 26 25 24 23 22 21	2019	18	17 16
Reserved	CHANNEL8_MUX[6:0]		Reserved	CHANNEL7_MUX[6:0]	

15	14 13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	CHANNEL6_MUX[6:0]						Reserved	CHANNEL5_MUX[6:0]						

Bit	name	access	describe	Reset value
31	Reserved	RO	is reserved.	0
[30:24]	CHANNEL8_MUX[6:0]	RW	DMAMUX Multiplexer channel 8 Input to resource allocation.	0
23	Reserved	RO	is reserved.	0
[22:16]	CHANNEL7_MUX[6:0]	RW	DMAMUX Multiplexer channel 7 Input to resource allocation.	0
15	Reserved	RO	is reserved.	0
[14:8]	CHANNEL6_MUX[6:0]	RW	DMAMUX Multiplexer channel 6 Input to resource allocation.	0
7	Reserved	RO	is reserved.	0
[6:0]	CHANNEL5_MUX[6:0]	RW	DMAMUX multiplexer channel 5 input to resource allocation: 0000000: DMA request input 1; 0000001: DMA request input 2; 0000010: DMA request input 3; ... 1111010: DMA request input 123; Other: Reserved.	0

10.3.10 DMA Request Multiplexer Channels 9-12 Configuration Register (DMAMUX9_12_CFGR)

Offset Address: 0x08

31	30 29 28	27 26 25 24 23 22 21	2019	18	17 16
Reserved	CHANNEL12_MUX[6:0]		Reserved	CHANNEL11_MUX[6:0]	

15	14 13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	CHANNEL10_MUX[6:0]						Reserved	CHANNEL9_MUX[6:0]						

Bit	name	access	describe	Reset value
31	Reserved	RO	is reserved.	0
[30:24]	CHANNEL12_MUX[6:0]	RW	DMAMUX Multiplexer channel 12 input to resource allocation.	0
23	Reserved	RO	is reserved.	0
[22:16]	CHANNEL11_MUX[6:0]	RW	DMAMUX Multiplexer channel 11 Input to resource allocation.	0
15	Reserved	RO	is reserved.	0

[14:8]	CHANNEL10_MUX[6:0]	RW	DMAMUX Multiplexer channel 10 Input to resource allocation.	0
7	Reserved	RO	is reserved.	0
[6:0]	CHANNEL9_MUX[6:0]	RW	DMAMUX multiplexer channel 9 input to resource allocation: 0000000: DMA request input 1; 0000001: DMA request input 2; 0000010: DMA request input 3; ... 1111010: DMA request input 123; Other: Reserved.	0

10.3.11 DMA Request Multiplexer Channels 13-16 Configuration Register (DMAMUX13_16_CFGR)

Offset Address: 0x0C

31	30 29 28	27 26 25 24 23 22 21	2019	18	17 16
Reserved	CHANNEL16_MUX[6:0]	Reserved	CHANNEL15_MUX[6:0]		

15	14 13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	CHANNEL14_MUX[6:0]						Reserved	CHANNEL13_MUX[6:0]						

Bit	name	access	describe	Reset value
31	Reserved	RO	is reserved.	0
[30:24]	CHANNEL16_MUX[6:0]	RW	DMAMUX Multiplexer channel 16 Input to resource allocation.	0
23	Reserved	RO	is reserved.	0
[22:16]	CHANNEL15_MUX[6:0]	RW	DMAMUX Multiplexer channel 15 Input to resource allocation.	0
15	Reserved	RO	is reserved.	0
[14:8]	CHANNEL14_MUX[6:0]	RW	DMAMUX Multiplexer channel 14 Input to resource allocation.	0
7	Reserved	RO	is reserved.	0
[6:0]	CHANNEL13_MUX[6:0]	RW	DMAMUX multiplexer channel 13 input to resource allocation: 0000000: DMA request input 1; 0000001: DMA request input 2; 0000010: DMA request input 3; ... 1111010: DMA request input 123; Other: Reserved.	0

Chapter 11 Analog-to-Digital Conversion (ADC)

The ADC module contains two 12-bit analog-to-digital converters with a maximum input clock speed of 80MHz. It supports 16 external channels and two... Internal signal source sampling source. Capable of single-channel and continuous channel conversion, automatic channel scanning mode, intermittent mode, and external triggering. Features include mode selection and dual sampling. A simulated watchdog function can be used to monitor whether the channel voltage is within the threshold range.

11.1 Key Features

ÿ 12-bit resolution

Supports sampling from 16 external channels and 2 internal signal sources.

Multiple sampling conversion modes for multiple channels: single, continuous, scan, trigger, intermittent, etc.

Data alignment modes: left alignment, right alignment

Sampling time can be programmed separately for each channel .

Both rule transformation and injection transformation support external triggering.

ÿSimulated watchdog monitoring channel voltage

Dual Mode

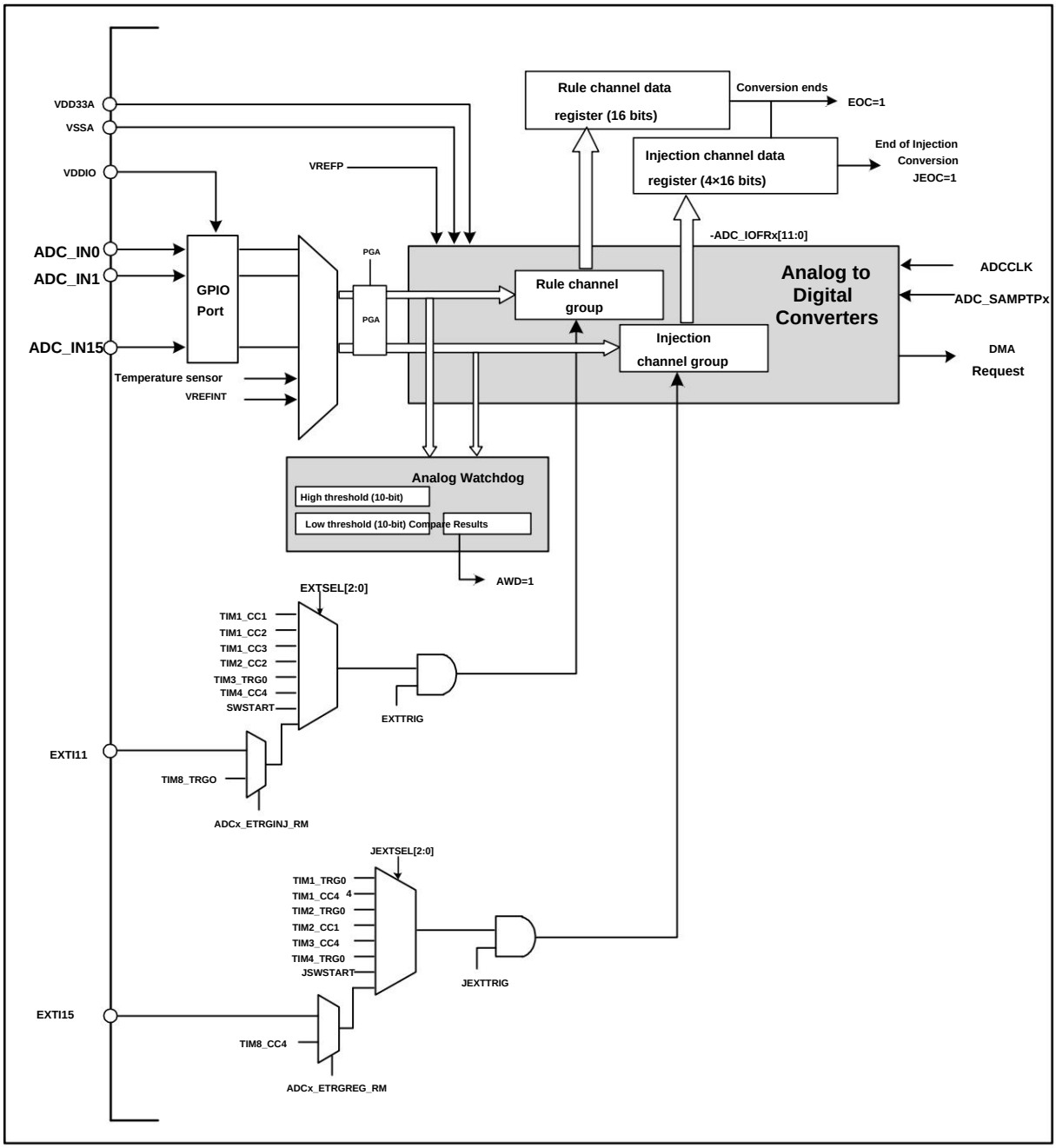
ÿ ADC channel input range: VSSÿVINÿVDDIO

ÿ The input gain is adjustable, enabling small signal amplification and sampling.

11.2 Functional Description

11.2.1 Module Structure

Figure 11-1 ADC Module Block Diagram



11.2.2 ADC Configuration

1) Module Power-On

A 1 value in the ADON bit of the ADCx_CTLR2 register indicates that the ADC module is powered on. This occurs when the ADC module transitions from power-off mode (ADON=0) to power-on mode.

After power-on (ADON=1), a delay of tSTAB is required for module stabilization. Then, the ADON bit is written back to 1 for...

This serves as the start signal for software-initiated ADC conversion. Clearing the ADON bit to 0 terminates the current conversion and puts the ADC module into a stop state.

In electric mode, the ADC consumes almost no power.

2) Sampling clock

The module's register operations are based on the HCLK (HB bus) clock, and its conversion unit's clock reference ADCCLK is synchronized with HCLK.
The ADCPRE[1:0] field of the RCC_CFGR0 register is configured with a frequency divider, which cannot exceed 80MHz.

3) Channel Configuration

The ADC module provides 18 sampling channels, including 16 external channels and 2 internal channels. They can be configured to two conversion modes.
Within the group switching mechanism: rule groups and injection groups. This allows for group switching that involves performing a series of transformations in any order across any number of channels.

Conversion group:

ÿ Rule group: Consists of up to 16 conversions. The rule channels and their conversion order are set in the ADCx_RSQRx register.

The total number of conversions in the rule group should be written into L[3:0] of the ADCx_RSQR1 register.

Injection group: Consists of up to 4 conversions. The injection channels and their conversion order are set in the ADCx_ISQR register. Note

The total number of conversions in the input group should be written into JL[1:0] of the ADCx_ISQR register.

Note: If the ADCx_RSQRx or ADCx_ISQRregister is modified during conversion, the current conversion is terminated, and a new conversion is initiated.
The number will be sent to the newly selected group for conversion.ADC

2 internal passages:

Temperature sensor: Connected to the ADC_IN16 channel, used to measure the temperature (TA) around the device.

ÿ VREFINT internal reference voltage: Connect to ADC_IN17 channel.

4) Calibration

is achieved by writing the RSTCAL bit of the ADCx_CTLR2 register to 1 to initialize the calibration register, waiting for the RSTCAL bit to be cleared to 0 by the hardware to indicate initialization.
Calibration complete. Set the CAL bit to initiate the calibration function. Once calibration is complete, the hardware will automatically clear the CAL bit and store the calibration code.
In ADCx_RDATAR. After that, normal conversion functions can begin. It is recommended to perform an ADC calibration once when the ADC module is powered on.

The calibration voltage value can be configured by writing to the CAL_VOL[1:0] bits of the ADCx_CTLR2 register; setting the CAL_AUTO bit to 1 enables the hardware.
Automatic calibration function.

Note: Before starting calibration, it must ADC ADCthe module has been powered on (ADON=1) for at least two clock cycles.
be ensured that the ADC sampling time is at least 1µs during calibration.

5) Programmable sampling time

The ADC samples the input voltage using a number of ADCCLK cycles. The number of sampling cycles for a channel can be determined by ADCx_SAMPTR1 and...
Change the SMPx[2:0] bits in the ADCx_SAMPTR2 register. Each channel can be sampled at a different time.
The total conversion time is calculated as follows:

TCONV = Sampling time + 12.5TADCCLK

The ADC's regular channel conversion supports DMA functionality. The value of the regular channel conversion is stored in a single data register, ADCx_RDATAR.
In order to prevent the data in the ADCx_RDATAR register from not being retrieved in time when switching between multiple rule channels consecutively, the ADC's...
DMA functionality. The hardware generates a DMA request at the end of the conversion of a regular channel (EOC set) and transfers the converted data from ADCx_RDATAR.
The register is transferred to the user-specified destination address.

After configuring the channels of the DMA controller module, write the DMA bit in the ADCx_CTLR2 register to 1 to enable the ADC's DMA function.
able.
Note: Injection group conversion is not supported.DMA

6) Data alignment

The ALIGN bit in the ADCx_CTLR2 register selects the alignment of the data stored after ADC conversion. 12-bit data supports left alignment and...
Right alignment mode.

The data register ADCx_RDATAR of the regular group channel stores the actual 12-bit digital value of the conversion; while the data of the injected group channel...
The ADCx_IDATARx register contains the actual converted data minus the offset defined in the ADCx_IOFRx register, and this value will persist.

There are positive and negative cases, so there is a sign bit (SIGNB).

Figure 11-2 Data left aligned

Rule Group Data Register

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

Injection group data register

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

Figure 11-3 Data right alignment

Rule Group Data Register

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

Injection group data register

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

11.2.3 External Trigger Source

The ADC conversion start event can be triggered by an external event. This is if the EXTTRIG or JEXTTRIG setting of the ADCx_CTLR2 register is configured.

The bits can be used to trigger the conversion of the rule group or injection group channels through external events. At this time, the EXTSEL[2:0] and JEXTSEL[2:0] bits...

The configuration determines the external event sources for rule groups and injection groups.

Note: When the external trigger signal is selected as ADC When a rule or injection transformation occurs, only its rising edge can initiate the transformation.

Table 11-1 External Trigger Sources for Rule Group Channels

EXTSEL[2:0]	Trigger source	type
000	CC1 event of Timer 1	Internal signals from on-chip timers
001	CC2 event of Timer 1	
010	CC3 event of Timer 1	
011	CC2 event of Timer 2	
100	TRGO event of Timer 3	
101	CC4 event of Timer 4	
110	EXTI line 11/TIM8_TRGO from external pin/internal timer signal	
111	SWSTART Position 1 Software Trigger	Software control bit

Table 11-2 External Trigger Sources for Injection Group Channels

JEXTSEL[2:0]	Trigger source	type
000	TRGO event of Timer 1	Internal signals from on-chip timers
001	CC4 event of Timer 1	
010	TRGO event of Timer 2	
011	CC1 event of Timer 2	
100	CC4 event of Timer 3	
101	TRGO event of Timer 4	
110	EXTI line 15/TIM8_CC4 from external pin/internal timer signal	
111	JSWSTART Location 1 Software Trigger	Software control bit

11.2.4 Switching Mode

Table 11-3 Conversion Mode Combinations

ADCx_CTLR1 and ADCx_CTLR2 register control bits						ADC conversion mode
CONT	SCAN	DISCEN/JDISCEN	JAUTO	Startup Event		

0	0	0	0	ADON Location 1 External trigger Way	Single-transfer single-channel mode: A single transfer is performed on a specific channel. Change. Single-time single-channel mode: a specific rule channel or injection channel. One channel performs a single conversion.
	1	0	0	ADON Location 1 or external Triggering method	Single scan mode: sequentially scans all selected rule groups. Channel (ADCx_RSQRx) or all injected group channels (ADCx_ISQR) performs a single conversion one by one. Trigger injection method: During the channel conversion process of the rule group, it is possible to... Insert all transformations into the injection group channel, and then continue with the regularization. Then the group channel is converted; however, no insertion will occur when the conversion is injected into the group channel. Enter the rule group channel conversion.
			1	ADON Location 1 or external Triggering method	Single scan mode: sequentially scans all selected rule groups. Channel (ADCx_RSQRx) or all injected group channels (ADCx_ISQR) performs a single conversion one by one. Automatic injection method: After the channel conversion of the rule group is completed, The injection group channel is automatically switched. Note: External trigger signals for the injection channel are not allowed during the conversion process.
	0	1 (DISCEN and) JDISCEN cannot be the same (Time is 1)	0	External trigger Way	Single-interrupted mode: Each time an event is initiated, a short interrupt is executed. Number of channels in the sequence (DISCNUM[2:0] defines the number of channels) The conversion will continue until all selected channels have been converted before starting over. start. Note: When this mode control bit is selected in the rule group and injection group, respectively... for DISCEN and JDISCEN cannot configure interrupt mode for both rule groups and injection groups at the same time; interrupt mode can only be used for one set of transformations.
			1	- Disable this mode.	
	1	1	X	- This mode is not available.	
1	0	0	0	ADON Location 1 or external Triggering method	Continuous single-channel/scan mode: Repeats a new cycle after each round. The conversion cycle continues until CONT is cleared to 0.
	1	0	0		
			1		

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

1) Single-channel conversion mode

In this mode, the transformation is performed only once for the current channel. This mode performs the transformation on the first-ranked channel in the rule group or injection group. The switch can be initiated by setting the ADON bit of the ADCx_CTLR2 register to 1 (only applicable to regular channels) or by external triggering. Dynamic (applicable to regular channels or injection channels). Once the selected channel's conversion is complete, it will:

If the channel being converted is a regular group, the converted data is stored in the 16-bit ADCx_RDATAR register, and the EOC flag is set. If the EOCIE bit is set, an ADC interrupt will be triggered.

If the conversion is applied to an injected group channel, the conversion data is stored in the 16-bit ADCx_IDATAR1 register, and the EOC and JEOC labels are displayed. If the JEOCIE or EOCIE bit is set, an ADC interrupt will be triggered.

2) Single scan mode switching is

achieved by setting the SCAN bit of the ADCx_CTLR1 register to 1 to enter ADC scan mode. This mode is used to scan a group of analog channels.

Perform a single turn on each of the channels selected by the ADCx_RSQRx register (for regular channels) or ADCx_ISQR (for injected channels). When the current channel transition ends, the next channel in the same group is automatically transitioned.

In scanning mode, depending on the state of the JAUTO bit, it is further divided into triggered injection mode and automatic injection mode.

Triggered Injection

When the JAUTO bit is 0, if a trigger event for injecting a channel transformation occurs during the scanning of a rule group channel, the current transformation is reset. The injection channel sequence is performed in a single scan manner. After all selected injection group channels have been scanned and converted, the last scan is restored. Interrupted rule group channel conversion.

If a regular channel initiation event occurs while the injection group channel sequence is being scanned, the injection group conversion will not be interrupted, but... The conversion of the regular sequence is performed after the conversion of the injected sequence is completed.

Note: When using triggered injection mode, ensure that the interval between trigger events must be longer than the injection sequence. For example, the total time to complete the transformation of the injection sequence should be [time required]. is 28 for ADCCLKs to 29 ADCCLKs, then the minimum event interval for triggering the injection channel

Automatic injection

When the JAUTO bit is 1, the injection of the selected channel's transformation is automatically performed after all channel transformations selected in the rule group have been completed. This method can be used to convert up to 20 conversion sequences in the ADCx_RSQRx and ADCx_ISQR registers.

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

Note: When the clock prescaler coefficient (ADCPRE[1:0]) is 0, when switching from a regular transition to an injection sequence or from an injection sequence... When switching from ADC 4 to 8 to a regular sequence, an ADCCLK interval will be automatically inserted; the delay of the ADCCLK interval will be[1] considered. ADC 2 Clock prescaler coefficient is At that time, there is[2] indivual

3) Single-transition interrupt mode

switching is achieved by setting the DISCEN or JDISCEN bit of the ADCx_CTLR1 register to 1 to enter the interrupt mode of the rule group or injection group. In contrast to the scanning mode that scans a complete set of channels, this mode divides a set of channels into multiple short sequences, which are executed each time an external event is triggered. A scan transformation of a short sequence.

The length n (n<=8) of the short sequence is defined in DISCNUM[2:0] of the ADCx_CTLR1 register. When DISCEN is 1, it is a regular sequence. The discontinuous mode of the group, the total length to be converted is defined in L[3:0] of the ADCx_RSQR1 register; when JDISCEN is 1, it is injected. The discontinuous mode of the group, the total length to be converted is defined in JL[1:0] of the ADCx_ISQR register. The rule group and the injection group cannot be set simultaneously. Set to intermittent mode.

Example of a discontinuous pattern in a rule group:

DISCEN=1, DISCNUM[2:0]=3, L[3:0]=8, Channels to be converted=1, 3, 2, 5, 8, 4, 10, 6 First external trigger: Conversion sequence: 1, 3, 2 Second external trigger: Conversion sequence: 5, 8, 4 Third external trigger: Conversion sequence: 10, 6, simultaneously generating an EOC event.

Fourth external trigger: The conversion sequence is: 1, 3, 2. Example of intermittent injection group pattern:

JDISCEN=1, JL[1:0]=3, Channels to be converted=1,3,2 First external trigger: Conversion sequence: 1 Second external trigger: Conversion sequence: 3 Third external trigger: Conversion sequence: 2, simultaneously generating EOC and JEOC events.

Fourth external trigger: The conversion sequence is: 1

Note: 1. When transforming a rule group or injection group in intermittent mode, the transformation sequence does not automatically restart from the beginning after completion. The transformation of the first subgroup will begin only after all subgroups have been transformed. 2. Automatic injection (JAUTO=1) and intermittent mode cannot be used simultaneously. 3. Intermittent mode cannot be set for both rule groups and injection groups simultaneously; intermittent mode can only be used for one set of transformations.

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

4) Continuous conversion

In continuous conversion mode, another conversion starts immediately after the previous ADC conversion ends; the conversion will not occur at the end of the selection group. Instead of stopping on one channel, the transition continues from the first channel of the selection group. The start events for this mode include external trigger events and... ADON position 1, after setting to start, CONT position 1 must be set.

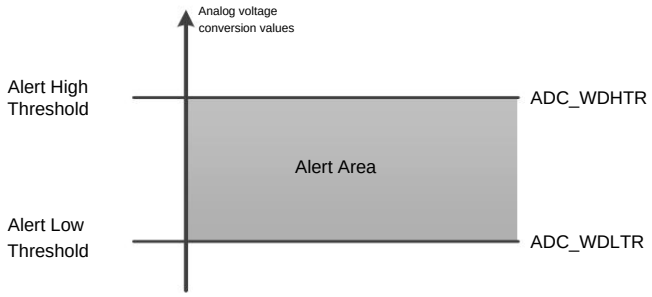
If a regular channel is converted, the converted data is stored in the ADCx_RDATAR register, and the conversion end flag EOC is set. If the EOCIE bit is set, an interrupt will be generated.

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

11.2.5 Analog Watchdog: If the

analog voltage converted by the ADC is below a low threshold or above a high threshold, the AWD analog watchdog status bit is set. Threshold Setting Located in the lowest 12 significant bits of the ADCx_WDHTR and ADCx_WDLTR registers. This is achieved by setting the AWDIE of the ADCx_CTLR1 register. The bit is used to allow the corresponding interrupt to be generated.

Figure 11-4 Simulated watchdog threshold region



Configure the AWDSGL, AWDEN, JAWDEN , and AWDCH[4:0] bits of the ADCx_CTLR1 register to select the analog watchdog alert. The specific relationships are shown in the table below:

Table 11-4 Simulated Watchdog Channel Selection

Simulated watchdog guard lane	ADCx_CTLR1 register control bits			
	AWDSGL	AWDEN	JAWDEN	AWDCH[4:0] Ignore
Not alerting	neglect	0	0	
all injection channels,	0	0	1	neglect
all rule channels, all	0	1	0	neglect
injections and rule channels,	0	1	1	neglect
single injection	1	0	1	Determine the channel number
channel, single rule	1	1	0	Determine the channel number
channel, single injection and rule channel	1	1	1	Determine the channel number

11.2.6 The temperature sensor chip

has a built-in temperature sensor connected to the ADC_INT16 channel. The ADC converts the voltage output by the sensor into a digital value for feedback. For the chip's internal temperature, a sampling time of 17.1 μs is recommended. The voltage output by the temperature sensor changes linearly with temperature, but this is due to manufacturing discrepancies. The slope and offset of the linear change curve differ, so internal temperature sensors are more suitable for detecting temperature changes than external sensors. Measure the absolute temperature. If precise temperature measurement is required, an external temperature sensor should be used.

The ADC's internal sampling channel can be woken up by setting the TSVREFE bit in the ADCx_CTLR2 register to 1, either through software startup or external triggering. Initiate the temperature sensor channel conversion of the ADC and read the data result (mV). The conversion formula between the digital value and temperature (°C) is as follows:

Temperature (°C) = ((VSENSE - V25) / Avg_Slope) + 25

V25: Voltage value of the temperature sensor at 25ÿ

Avg_Slope: The average slope (mV/ÿ) of the temperature versus VSENSE curve.

Refer to the actual values of V25 and Avg_Slope in the Electrical Characteristics section of the datasheet .

Note: Powering on the internal temperature sensor (changing the TSVREFE bit from 1) requires a setup time, and powering on the ADC module also requires a setup time.

setup time (ADON 1), so to shorten the waiting time, you can simultaneously set the bit from 0 ADON TSVREFE And position.

11.2.7 Dual ADC Mode In a chip with

two ADC modules, the two ADCs can be used together to achieve dual ADC mode. In dual ADC mode,

ADC1 is the master ADC, and ADC2 is the slave ADC. The mode is selected by configuring DUALMOD[3:0] in ADC1_CTLR1, thus enabling the operation of ADC1 and ADC2.

Alternate triggering or synchronous triggering switching.

Note: In dual-mode, when selecting external event triggering, the user must enable external triggering for both master and slave, and the master must be enabled.

ADC Set as the corresponding trigger, from ADC ADC is set to software triggering to prevent unnecessary triggering from affecting the slave. ADC Perform the conversion.

The following possible modes can be achieved through configuration:

Independent mode

ÿSynchronous injection mode

Synchronization rule mode

ÿ Rapid Alternation Mode

Slow Alternating Mode

Alternating trigger mode

ÿSynchronous rule mode + synchronous injection mode

Synchronization rule mode + alternating trigger mode

ÿSynchronous injection mode + fast alternation mode

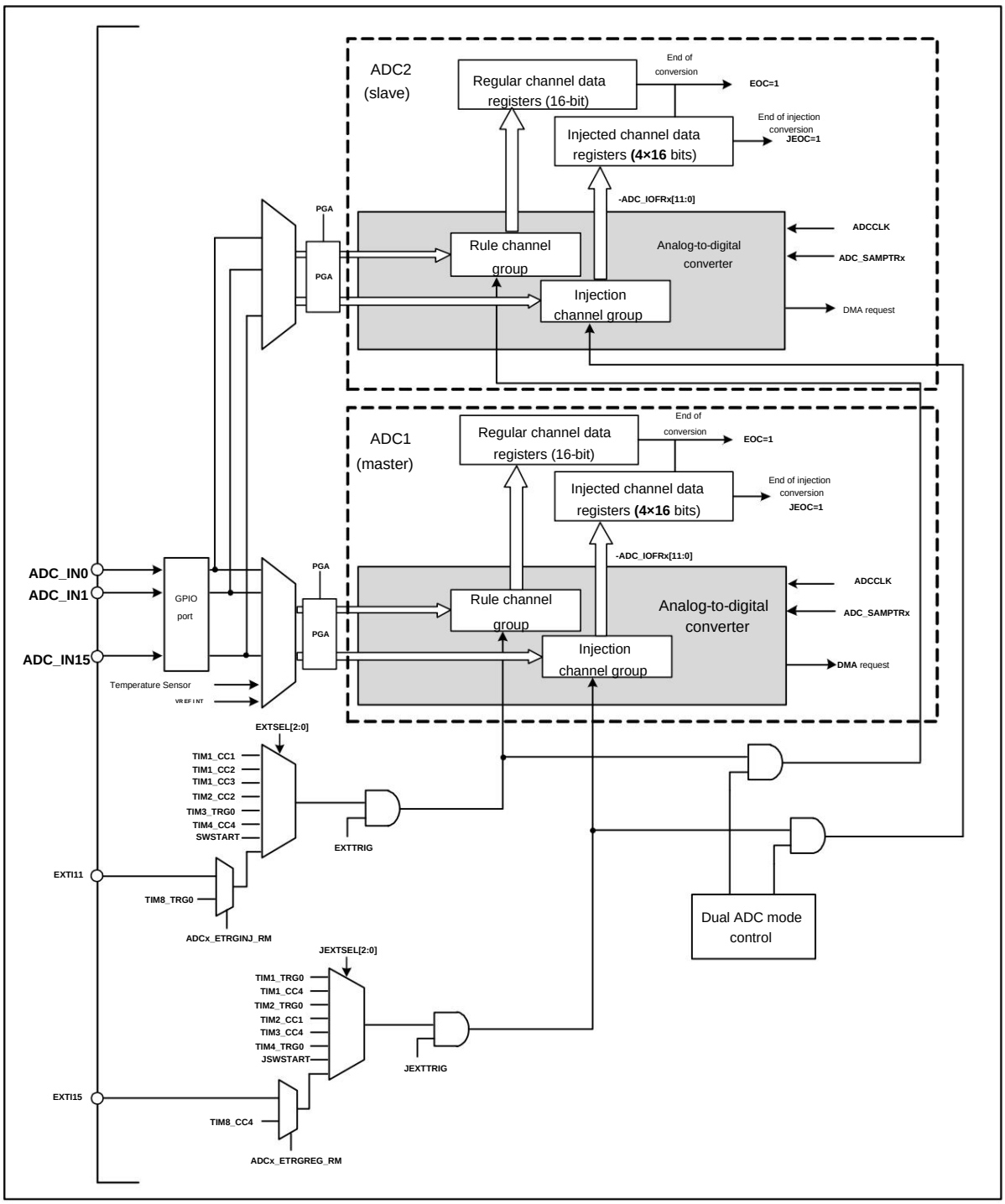
ÿSynchronous injection mode + slow alternation mode

Note: 1. In dual mode, an enable bit is required so that the master data register can read the converted data from the slave.

ADC DMA 2. Only has the function. The converted data can only be processed through dual modes, utilizing

the function. ADC1 DMA ADC2 ADC ADC1 DMA transfer.

Figure 11-5 Block diagram of dual ADCs



1) Independent Mode

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

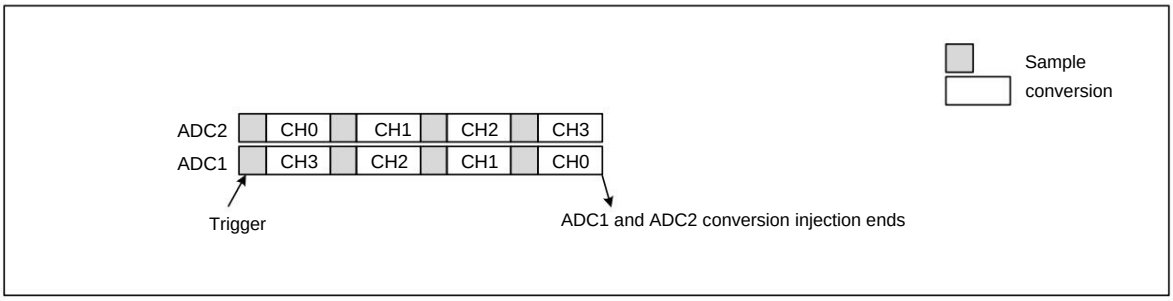
2) Synchronous Injection Mode :

This mode is used to switch an injection channel group, setting JEXTSEL[2:0] in ADC1_CTLR2 to select the trigger source, while it...

It will also be used to synchronously trigger ADC2. Upon completion of the conversion, the converted data is stored in ADCx_IDATARx of ADCx, and if enabled...

Any ADC interrupt will generate a JEOC interrupt after the conversion is completed.

Figure 11-6 4-channel synchronous injection conversion



Note: 1. At the same time ADC1 and ADC2

and ADC2 The conversion channels should not overlap. 2. In synchronous mode, ADC1 There should be transformation sequences of the same duration or the longer of the two transformation sequences.

The time interval is less than the trigger time interval to ensure that both sequences can be converted completely each time a trigger is triggered.

3) Synchronization Rule Mode :

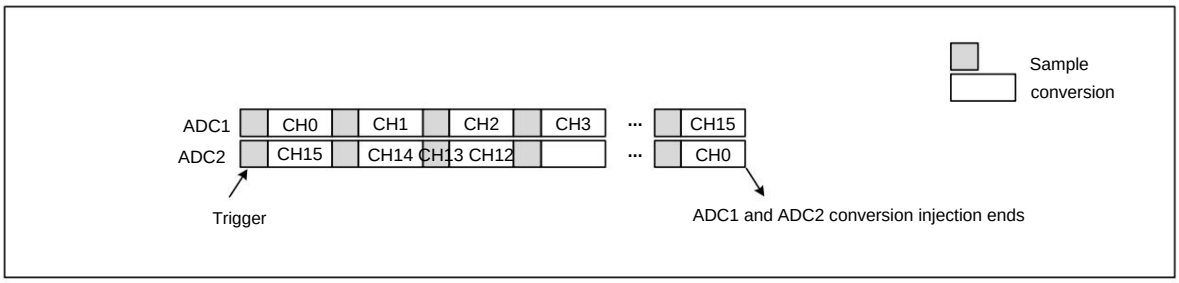
This mode is used to convert rule channel sequences. It sets EXTSEL[2:0] in ADC1_CTLR2 to select the trigger source, and it also...

Used to synchronously trigger ADC2. Upon completion of the conversion, a 32-bit DMA transfer request will be generated, transferring the contents of the data register ADC1_RDATAR.

The data is transferred to SRAM, with the high 16 bits containing ADC2 conversion data and the low 16 bits containing ADC1 conversion data. If any ADC interrupt is enabled,

This will generate an EOC interrupt.

Figure 11-7 16-Channel Synchronization Rule Conversion



Note: 1. At the same time ADC1 and ADC2

and ADC2 1. The conversion channels should not overlap. 2. In synchronous mode, ADC1 There should be transformation sequences of the same duration or the longer of the two transformation sequences.

The time interval is less than the trigger time interval to ensure that both sequences can be converted completely each time a trigger is triggered.

4) Fast Alternating Mode: This

mode is only applicable to regular channels (often only one channel). Set EXTSEL[2:0] in ADC1_CTLR2 to select the trigger.

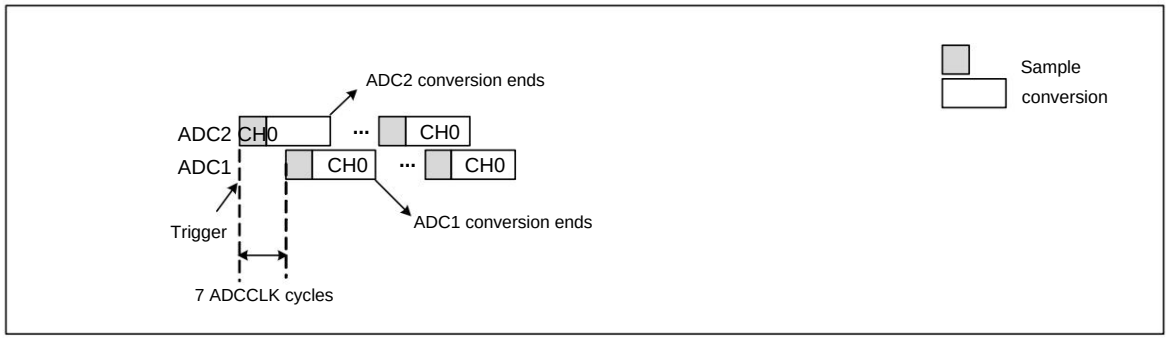
When the source is triggered, ADC2 will immediately start conversion, while ADC1 will start conversion after a delay of 7 ADC clock cycles. If both are enabled...

In continuous mode (CONT set), the two ADCs will continuously alternate switching on the regular channel. If interrupts are enabled, ADC1 will generate an EOC.

If DMA is enabled at the same time, a 32-bit DMA transfer request will be generated, transferring the contents of the data register ADC1_RDATAR to...

In the SRAM, the high 16 bits contain ADC2 conversion data, and the low 16 bits contain ADC1 conversion data.

Figure 11-8 Single-channel fast alternating continuous conversion



The problem of combining ADC1 and ADC2

Note: The sampling time should be less than one clock cycle to avoid sampling cycle overlap when sampling the same channel.

5) Slow Alternating Mode: This

mode is only applicable to regular channels and can only be used on one channel at a time. Set EXTSEL[2:0] in ADC1_CTLR2 to select the trigger source.

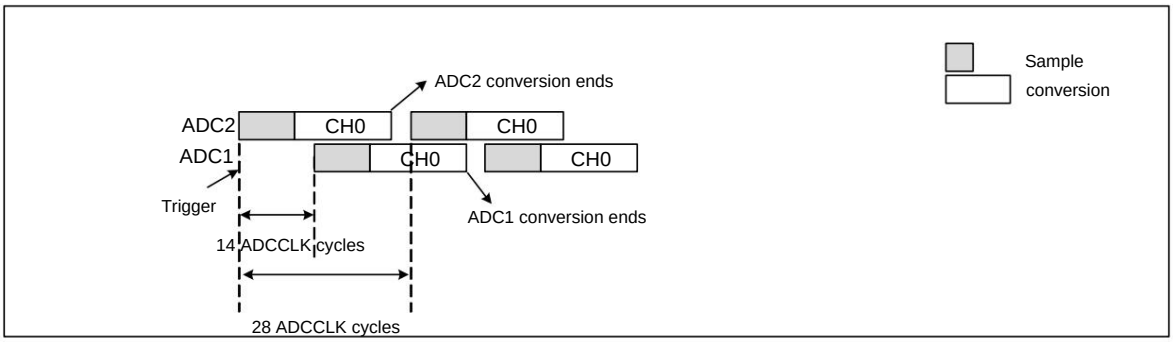
After the trigger is generated, ADC2 will start conversion immediately, while ADC1 will start conversion after a delay of 14 ADC clock cycles, followed by another 14 ADC clock cycles.

After one clock cycle, ADC2 restarts, and this cycle repeats. If interrupts are enabled, ADC1 will generate an EOC interrupt; if DMA is enabled simultaneously, ...

This will generate a 32-bit DMA transfer request, transferring the contents of the data register ADC1_RDATAR to SRAM, with the high 16 bits...

Contains ADC2 conversion data; the lower 16 bits contain ADC1 conversion data.

Figure 11-9 Single-channel slow alternation conversion



A new

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

2.28 ADC2 ADC2 2. No need to set bits conversion will automatically begin after 14 ADC clock cycles;

CONT

6) Alternating Trigger Mode:

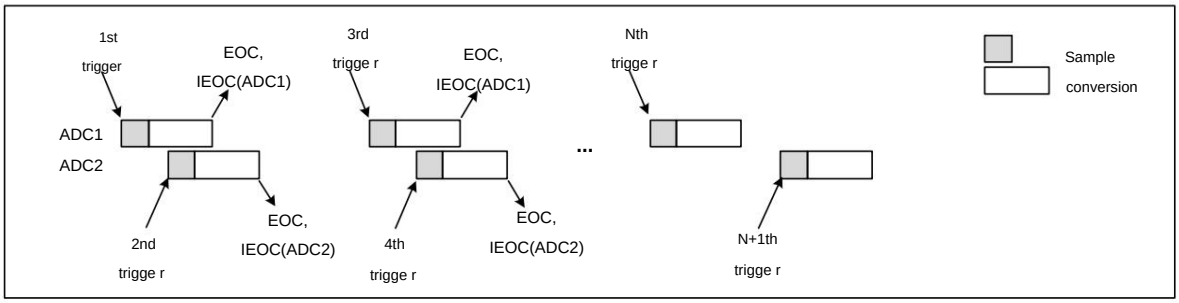
This mode is only applicable to the injection channel group. Set JEXTSEL[2:0] in ADC1_CTLR2 to select the trigger source. When the first trigger occurs...

When the event occurs, all injection channels on ADC1 are switched; when the second event occurs, all injection channels on ADC2 are switched.

The conversion process repeats sequentially. If interrupts are enabled, a JEOC interrupt will be generated when all injection channels of ADC1 have completed their conversions, and all injection channels of ADC2 will be converted.

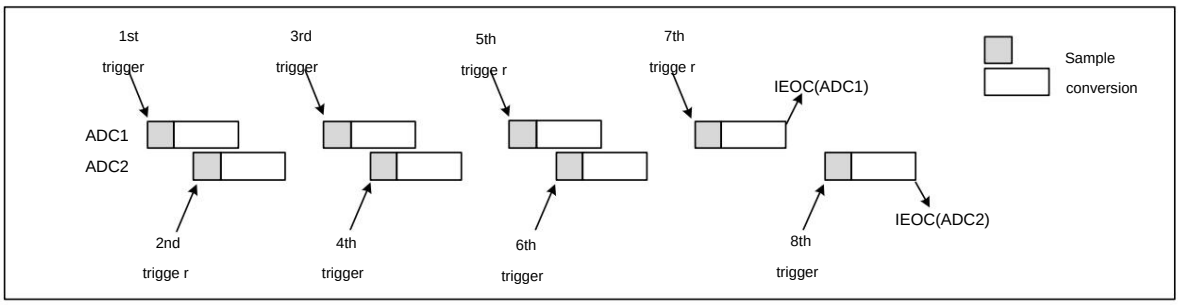
A JEOC interrupt is generated after the input channel conversion is completed.

Figure 11-10 Each ADC injection channel group alternately triggers the conversion.



If the injection discontinuity mode is used simultaneously, the first injection channel on ADC1 will be converted when the first trigger event occurs, and the second injection channel will be converted when the first event occurs. When the secondary trigger event occurs, the first injection channel on ADC2 is converted. This process continues in a loop. Simultaneously, if interrupts are enabled, then when... A JEOC interrupt is generated when all injected channels of ADC1 have been converted, and a JEOC interrupt is generated when all injected channels of ADC2 have been converted.

Figure 11-11 Intermittent mode: Each ADC injection channel alternately triggers conversion.



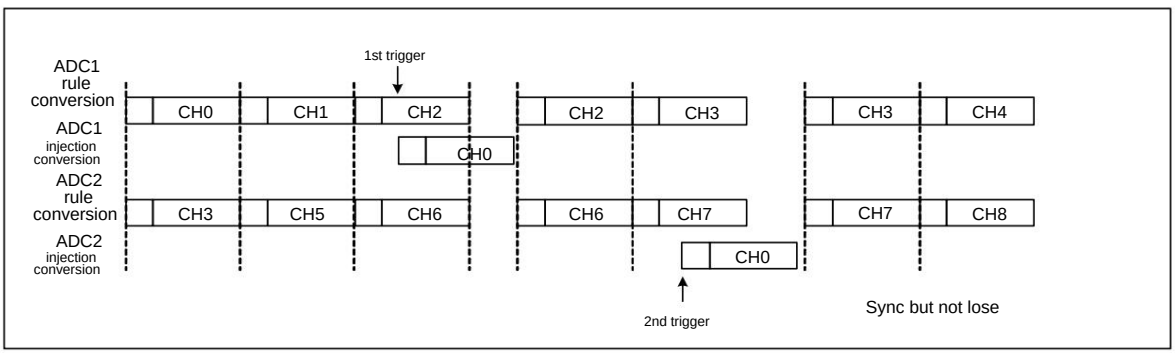
7) Synchronous rule mode + synchronous injection mode

In this mode, the synchronous transformation of the rule group can be interrupted to initiate the synchronous transformation of the injection group. However, this mode requires ensuring the accuracy of the transformation tool. There are sequences with the same duration, or the trigger interval is guaranteed to be longer than the duration of the longer sequence of the two sequences.

8) Synchronous rule mode + alternating trigger mode

In this mode, the synchronous transition of the rule group can be interrupted to initiate the alternating triggering transition of the injection group. When an injection event occurs, the transition... The trigger conversion starts immediately. If a synchronization rule is being converted, all ADC rule conversions are stopped, and the conversion is resumed after the injection conversion is complete. Step-by-step recovery.

Figure 11-12 Alternating Trigger Injection Channel Switching in Synchronous Rule Mode

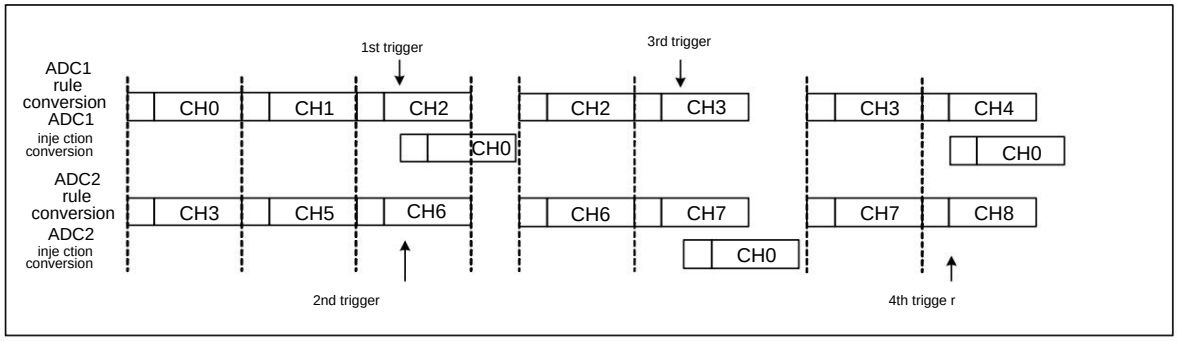


Note: In this mode, it is necessary to ensure that the sequences have the same duration, or that the trigger interval is longer than the duration of the longer sequence of the two sequences.

If the injection trigger event occurs during an injection transformation that interrupts the rule transformation, then this trigger event will be ignored, for example, as follows:

The diagram illustrates the second triggering scenario.

Figure 11-13 The triggering event occurs during the injection conversion.

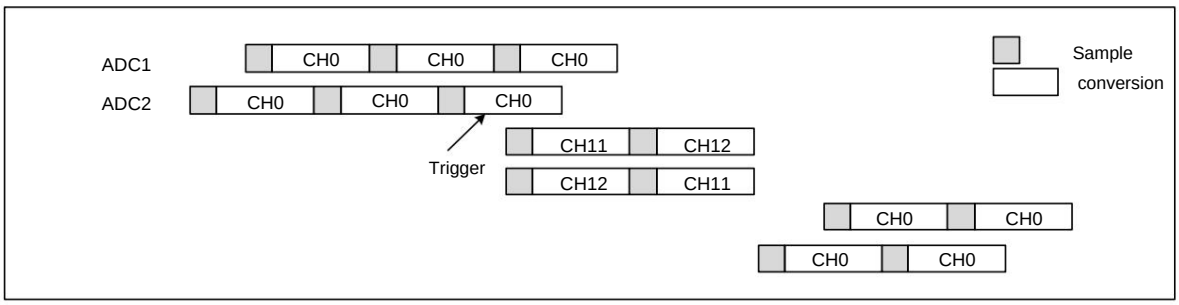


9) Synchronous Injection Mode + Alternating

Mode: In this mode, injection transitions can interrupt alternating transitions. When an injection event occurs, alternating transitions are interrupted, and injection transitions begin.

After the injection conversion is completed, the alternation conversion is resumed.

Figure 11-14 Trigger injection group switching under alternating switching



Note: When the pre-frequency division coefficient is The ADC alternates for 4 clock When the alternating conversion is restored, the sampling interval is no longer uniform. One ADC is changed to one sum 7 6 cycles.

11.2.8 DFSDM Conversion The ADC

conversion result can be directly transmitted to a digital filter (DFSDM) with a ÿÿ modulator. In this case, ADCx_AUX[31]

The bit must be set to 1, and the DMA bit must be cleared to 0. The ADC will transfer the 12 least significant bits of the regular data register data to the DFSDM.

Once the transfer is complete, DFSDM will reset the EOC flag.

11.2.9 ADC Low Power Mode

The ADC operating mode can be configured via the ADC_LP bit in the ADCx_CTLR2 register. The default value of 0 results in higher power consumption and is unsuitable for certain applications.

It supports sampling rates of 6 MHz and above, and is only applicable to VDD33A above 3V; when set to 1, it is a low-power mode with lower power consumption, suitable for 2 MHz.

The following sampling rates.

11.3 Register Description

Table 11-5 List of ADC1 Related Registers

name	Access address	describe	Reset value
R32_ADC1_STATR	0x40012400 ADC1 Status Register; 0x40012404 ADC1		0x00000000
R32_ADC1_CTLR1	Control Register 1; 0x40012408 ADC1 Control Register		0x00000000
R32_ADC1_CTLR2	2; 0x4001240C ADC1 Sampling Time Configuration		0x000000D0
R32_ADC1_SAMPTR1	Register 1		0x00000000

R32_ADC1_SAMPTR2	0x40012410 ADC1 Sampling Time Configuration Register	0x00000000
R32_ADC1_IOFR1	2 0x40012414 ADC1 Injection Channel Data Offset Register 1	0x00000000
R32_ADC1_IOFR2	0x40012418 ADC1 Injection Channel Data Offset Register 2	0x00000000
R32_ADC1_IOFR3	0x4001241C ADC1 Injection Channel Data Offset Register 3	0x00000000
R32_ADC1_IOFR4	0x40012420 ADC1 Injection Channel Data Offset Register 4	0x00000000
R32_ADC1_WDHTR	0x40012424 ADC1 Watchdog High Threshold Register	0x00000FFF
R32_ADC1_WDLTR	0x40012428 ADC1 Watchdog Low Threshold Register	0x00000000
R32_ADC1_RSQR1	0x4001242C ADC1 Regular Channel Sequence Register	0x00000000
R32_ADC1_RSQR2	1 0x40012430 ADC1 Regular Channel Sequence Register	0x00000000
R32_ADC1_RSQR3	2 0x40012434 ADC1 Regular Channel Sequence Register	0x00000000
R32_ADC1_ISQR	3 0x40012438 ADC1 Injection Channel Sequence	0x00000000
R32_ADC1_IDATAR1	Register 0x4001243C ADC1 Injection Data Register	0x00000000
R32_ADC1_IDATAR2	1 0x40012440 ADC1 Injected Data Register 2	0x00000000
R32_ADC1_IDATAR3	0x40012444 ADC1 Injected Data Register 3	0x00000000
R32_ADC1_IDATAR4	0x40012448 ADC1 Injected Data Register 4	0x00000000
R32_ADC1_RDATAR	0x4001244C ADC1 Regular Data Register	0x00000000
R32_ADC1_AUX	0x40012454 ADC1 Sampling Time Register	0x00000000

Table 11-6 List of ADC2 Related Registers

name	Access Address Description: 0x40012800 ADC2	Reset value
R32_ADC2_STATR	Status Register; 0x40012804 ADC2 Control	0x00000000
R32_ADC2_CTLR1	Register 1; 0x40012808 ADC2 Control	0x00000000
R32_ADC2_CTLR2	Register 2; 0x4001280C ADC2 Sampling	0x000000D0
R32_ADC2_SAMPTR1	Time Configuration Register 1; 0x40012810 ADC2	0x00000000
R32_ADC2_SAMPTR2	Sampling Time Configuration Register 2; 0x40012814	0x00000000
R32_ADC2_IOFR1	ADC2 Injection Channel Data Offset Register 1; 0x40012818	0x00000000
R32_ADC2_IOFR2	ADC2 Injection Channel Data Offset Register 2; 0x4001281C	0x00000000
R32_ADC2_IOFR3	ADC2 Injection Channel Data Offset Register 3; 0x40012820	0x00000000
R32_ADC2_IOFR4	ADC2 Injection Channel Data Offset Register 4; 0x40012824	0x00000000
R32_ADC2_WDHTR	ADC2 Watchdog High Threshold Register; 0x40012828	0x00000FFF
R32_ADC2_WDLTR	ADC2 Watchdog Low Threshold Register; 0x4001282C	0x00000000
R32_ADC2_RSQR1	ADC2 Regular Channel Sequence Register 1 0x40012830	0x00000000
R32_ADC2_RSQR2	ADC2 Regular Channel Sequence Register 2 0x40012834	0x00000000
R32_ADC2_RSQR3	ADC2 Regular Channel Sequence Register 3 0x40012838	0x00000000
R32_ADC2_ISQR	ADC2 Injection Channel Sequence Register 0x4001283C	0x00000000
R32_ADC2_IDATAR1	ADC2 Injection Data Register 1 0x40012840 ADC2	0x00000000
R32_ADC2_IDATAR2	Injection Data Register 2 0x40012844 ADC2	0x00000000
R32_ADC2_IDATAR3	Injection Data Register 3 0x40012848 ADC2	0x00000000
R32_ADC2_IDATAR4	Injection Data Register 4 0x4001284C ADC2 Regular	0x00000000
R32_ADC2_RDATAR	Data Register 0x40012854 ADC2 Sampling Time	0x00000000
R32_ADC2_AUX	Register 0x40012858 ADC2_TOUCHKEY	0x00000000
R32_ADC2_DRV	Multiplexer Mask Register	0x00000000

11.3.1 ADCx Status Register (ADCx_STATR) (x=1/2)

Offset address: 0x00

31	30 29 28	27 26 25 24 23 22 21	2019	18	17 16
Reserved					

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

Bit	Name access		describe	Reset value
[31:16] Reserved		RO is reserved.		0
15	ADCRSTF	RO	ADC reset status indication: 1: Reset complete; 0: Resetting.	0
[14:5] Reserved		RO reserved.		0
4	STRT	RW0	Rule channel conversion start status: 1: The rule channel conversion has begun; 0: Rule channel conversion has not started. This bit is set to 1 by hardware and cleared to 0 by software (writing 1 is invalid).	0
3	JSTRT	RW0	Injection channel conversion start state: 1: Injection channel conversion has begun; 0: Injection channel conversion has not started. This bit is set to 1 by hardware and cleared to 0 by software (writing 1 is invalid).	0
2	JEOC	RW0	Injection channel group conversion completion status: 1: Conversion complete; 0: Conversion incomplete. This bit is set to 1 by hardware (after all injected channels have been converted) and cleared to 0 by software. (Writing 1 is invalid).	0
1	EOC	RW0	Transition completion status: 1: Conversion complete; 0: Conversion incomplete. This bit is set to 1 by hardware (when rule or injection channel group conversion ends), and by software. Clear the value to 0 (writing 1 is invalid) or clear it when reading ADCx_RDATAR.	0
0	AWD	RW0	Simulated watchdog marker position: 1: A simulated watchdog event occurs; 0: No simulated watchdog event occurred. This bit is set to 1 by hardware (conversion value exceeds ADCx_WDHTR and The ADCx_WDLTR register range is cleared to 0 by software (writing 1 is invalid).	0

11.3.2 ADCx Control Register 1 (ADCx_CTLR1) (x=1/2)

Offset address: 0x04

31	30 29	28	27 26 25	twenty four	twenty three	twenty two	21 20 19	18	17 16
SW_PRE	ANA_RST	Reserved	PGA[1:0]	BUFEN	TKITUNE	TKENABLE	AWDEN	JAWDEN	Reserved	DUALMOD[3:0]	

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

DISCNUM[2:0]	JDISC EN	DISC EN	JAUTO	AWD SGL	SCAN	JEOC IE	AWDIE	EOC IE	AWDCH[4:0]

Bit	Name access		describe	Reset value
31	SW_PRE	RW	Channel pre-switching off: 1: Channel pre-switching is disabled; 0: Channel pre-switching enabled.	0
30	ANA_RST	RW	Analog module reset enable: 1: Enable; 0: Off.	0
29	Reserved	RO is reserved.		0
[28:27] PGA[1:0]		RW	ADC channel gain configuration: 00: x1; 01: x4; 10: x16; 11:x64. Note: The input gain is adjustable, enabling small signal amplification and sampling. ADC_Buffer must be enabled to use this function.	00b
26	BUFEN	RW	ADC BUFFER Enable: 1: Enable input buffer; 0: Turn off the input buffer.	0
25	TKITUNE	RW	TKEY module charging current configuration: 1: Charging current halved; 0: The charging current is 35uA.	0
twenty four	TKENABLE	RW	TKEY module enable control, including TKEY_F and TKEY_V units: 1: Enable the TKEY module; 0: Disable the TKEY module.	0
twenty three	AWDEN	RW	Simulate the watchdog function enable bit on the rule channel: 1: Enable simulated watchdog on the rule channel; 0: Disable simulated watchdog on the rules channel.	0
twenty two	JAWDEN	RW	Simulate the watchdog function enable bit on the injection channel: 1: Enable the simulated watchdog on the injection channel; 0: Disable the simulated watchdog on the injection channel.	0
[21:20] Reserved		RO is reserved.		0
[19:16] DUALMOD[3:0]		RW	Dual mode selection: 0000: Standalone mode; 0001: Synchronization rules + synchronous injection mode;	0000b

			0010: Synchronization rules + alternating trigger mode; 0011: Synchronous injection + fast alternation mode; 0100: Synchronous injection + slow alternating mode; 0101: Synchronous injection mode; 0110: Synchronization rule mode; 0111: Rapid Alternation Mode; 1000: Slow alternation mode; 1001: Alternating trigger mode. Note: These bits are reserved in ADC2. Any modification to these configuration bits should be made with dual mode disabled.	
[15:13]	DISCNUM[2:0]	RW	In intermittent mode, the number of rule channels to be converted after an external trigger: 000: 1 channel; … 111: 8 channels.	000b
12	JDISCEN	RW	Discontinuous mode enable bit on the injection channel: 1: Enable discontinuous mode on the injection channel; 0: Disables the intermittent mode on the injection channel.	0
11	DISCEN	RW	Discontinuous mode enable bit on the regular channel: 1: Enable discontinuous mode on the rule channel; 0: Disable intermittent mode on the rule channel.	0
10	JAUTO	RW	After the rule channel is enabled, the injection channel group enable bit will be automatically switched: 1: Enable automatic injection channel group switching; 0: Disable automatic injection channel group switching. Note: This mode requires disabling external triggering of the injection channel.	0
9	AWDSGL	RW	In scan mode, use an analog watchdog enable bit on a single channel: 1: Use an analog watchdog on a single channel (AWDCH[4:0] selected); 0: Use a simulated watchdog on all channels.	0
8	SCAN	RW	Scan mode enable bit: 1: Enable scan mode (continuous conversion ADCx_IOFRx and ADCx_RSQRx) (Select all channels); 0: Turn off scan mode.	0
7	JEOCIE	RW	Injection channel group conversion end interrupt enable bit: 1: Enable the interrupt for injection channel group conversion completion (JEOC flag); 0: The injection channel group conversion is interrupted.	0
6	AWDIE	RW	Simulated watchdog interrupt enable bit: 1: Enable simulated watchdog interrupt; 0: Disable simulated watchdog interrupt. Note: In scan mode, this interruption will abort the scan.	0
5	EOCIE	RW	End of conversion (rule or injection channel group) interrupt enable bit: 1: Enable End of Conversion Interrupt (EOC Flag); 0: Disable conversion end interrupt.	0
[4:0]	AWDCH[4:0]	RW	Simulated watchdog channel selection position: 00000: Analog input channel 0; 00001: Analog input channel 1; …	00000b

			10000: Analog input channel 16; 10001: Analog input channel 17.	

11.3.3 ADCx Control Register 2 (ADCx_CTLR2) (x=1/2)

Offset address: 0x08

31	30 29 28	27 26 25 24 23 22 21	2019	18	17 16
Reserved			TS VREFE	SW START	JSW START	EXT TRIG	EXTSEL[2:0]	Reserved

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

Bit	Name access		describe	Reset value
[31:24] Reserved		RO reserved.		0
twenty three	TSVREFE	RW	Temperature sensor and internal voltage (VREFINT) channel enable bit: 1: Enable the temperature sensor and VREFINT channel; 0: Disable temperature sensor and VREFINT channel. Note: This bit applies only to ADC1.	0
twenty two	SWSTART	RW	To initiate a rule-based channel conversion, software triggering needs to be configured: 1: Initiate rule channel conversion; 0: Reset state. This bit is set by software and cleared to 0 by hardware after the conversion begins.	0
twenty one	JSWSTART	RW	Initiating an injection channel conversion requires configuring software triggering: 1: Initiate injection channel conversion; 0: Reset state. This bit is set by software and cleared to 0 by hardware or software after the conversion begins.	0
20	EXTTRIG	RW	External trigger conversion mode enabled for rule channels: 1: Use an external event to initiate the conversion; 0: Disable external event startup functionality.	0
[19:17] EXTSEL[2:0]		RW	External trigger event selection for initiating rule channel conversion: 000: CC1 event of Timer 1; 001: CC2 event of Timer 1; 010: CC3 event of Timer 1; 011: CC2 event of Timer 2; 100: TRGO event of Timer 3; 101: CC4 event of Timer 4; 110: TRGO event of EXTI line 11/timer 8; 111: SWSTART software trigger.	000b
16	Reserved	RO reserved.		0
15	JEXTTRIG	RW	External trigger switching mode enabled for injection channel: 1: Use an external event to initiate the conversion; 0: Disable external event startup functionality.	0
[14:12] JEXTSEL[2:0]		RW	External trigger event selection to initiate injection channel conversion: 000: TRGO event for Timer 1;	000b

			001: CC4 event of Timer 1; 010: TRGO event of Timer 2; 011: CC1 event of Timer 2; 100: CC4 event of Timer 3; 101: TRGO event of Timer 4; 110: CC4 event on EXTI line 15/Timer 8; 111: JSWSTART software trigger.	
11	ALIGN	RW	Data alignment: 1: Left alignment; 0: Right alignment.	0
[10:9] Reserved		RO reserved.		0
8	DMA	RW	Direct Memory Access (DMA) mode enabled: 1: Enable DMA mode; 0: DMA mode is off.	0
7	ADC_LP	RW	ADC low-power mode control bit: 1: Low power mode, with lower power consumption, suitable for sampling rates below 2MHz; 0: High power consumption, suitable for sampling rates of 6MHz and above, and only applicable to... VDD33A is higher than 3V.	1
6	CAL_AUTO	RW	Hardware automatic calibration function enabled: 1: The ADC result will be automatically subtracted from the offset value; 0: The ADC result will not be subtracted from the offset value.	1
[5:4] CAL_VOL[1:0]		RW	Calibration voltage configuration: 00: 0.3*Vref (recommended); 01: 0.5*Vref; 10: 0.75*Vref; 11: Suspended.	01b
3	RSTCAL	RW	Reset calibration: This bit is set by software and cleared to 0 by hardware after the reset is complete. 1: Initialize the calibration register; 0: The calibration register has been initialized. Note: If RSTCAL is set during conversion, clear the calibration register. The device requires additional cycles.	0
2	CAL	RW	A/D calibration: This bit is set by software and cleared to 0 by hardware when calibration is complete. 1: Begin calibration; 0: Calibration 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-off mode; When this bit is 1, writing 1 will initiate the conversion; if there are other... If the intended position changes, a new conversion will not be initiated. 1: Enable ADC and start conversion; 0: Turn off ADC conversion/calibration and enter power-off mode.	0

11.3.4 ADCx Sampling Time Configuration Register 1 (ADCx_SAMPTR1) (x=1/2)

Offset address: 0x0C

31	30 29	28	27 26 25 24 23 22 21	2019	18	17 16
Reserved				SMP17[2:0]	SMP16[2:0]	SMP15[2:1]

15	14 13	12	11 10	9	8	7	6	5	4	3	2	1	0
SMP15[0]	SMP14[2:0]		SMP13[2:0]		SMP12[2:0]			SMP11[2:0]			SMP10[2:0]		

Bit	Name access		describe	Reset value
[31:24] Reserved		RO is reserved.		0
[23:0] SMPx[2:0]		RW	SMPx[2:0]: Sampling time configuration for channel x: 000: 1.5 cycles; 001: 7.5 cycles; 010: 13.5 cycles; 011: 28.5 cycles; 100: 41.5 cycles; 101: 55.5 cycles; 110: 71.5 cycles; 111: 239.5 cycles; SMPx[2:0]: When the ADC_SMP_SELx bit is set, the sampling time of channel x is... Room configuration: 000: 1.5 cycles; 001: 7.5 cycles; 010: 13.5 cycles; 011: 28.5 cycles; 100: 2.5 cycles; 101: 3.5 cycles; 110: 4.5 cycles; 111: 5.5 cycles; These bits are used to independently select the sampling time for each channel during sampling. The channel configuration value must remain unchanged during the cycle. Note: When the sampling rate is higher than a certain value, it is not recommended that the sampling time be less than one week. 1MHz 3.5 periods	000b

11.3.5 ADCx Sampling Time Configuration Register 2 (ADCx_SAMPTR2) (x=1/2)

Offset address: 0x10

31	30 29	28	27 26 25 24 23 22 21	2019	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		describe	Reset value
[31:30] Reserved		RO is reserved.		0
[29:0] SMPx[2:0]		RW	SMPx[2:0]: Sampling time configuration for channel x: 000: 1.5 cycles; 001: 7.5 cycles; 010: 13.5 cycles; 011: 28.5 cycles; 100: 41.5 cycles; 101: 55.5 cycles; 110: 71.5 cycles; 111: 239.5 cycles; SMPx[2:0]: When the ADC_SMP_SELx bit is set, the sampling time of channel x is... Room configuration: 000: 1.5 cycles; 001: 7.5 cycles;	000b

			010: 13.5 cycles; 011: 28.5 cycles; 100: 2.5 cycles; 101: 3.5 cycles; 110: 4.5 cycles; 111: 5.5 cycles; These bits are used to independently select the sampling time for each channel during sampling. The channel configuration value must remain unchanged during the cycle. Note: When the sampling rate is higher than a certain value, it is not recommended that the sampling time be less than one week. 1MHz 3.5 periods	

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

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

31	30 29	28	27 26 25 24 23 22 21	2019	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	describe	Reset value
[31:12]	Reserved	RO is reserved.	0
[11:0]	JOFFSETx[11:0] RW	The data offset value of the injected channel x. When converting the injection channel, this value defines the parameters used to convert the original data. The value subtracted from the input. The result of the conversion can be stored in the ADCx_IDATARx register. Read from the device.	0

11.3.7 ADCx Watchdog High Threshold Register (ADCx_WDHTR) (x=1/2) Offset address: 0x24

31	30 29	28	27 26 25 24 23 22 21	2019	18	17 16
Reserved						

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

Bit	Name access	describe	Reset value
[31:12]	Reserved	RO is reserved.	0
[11:0]	HT[11:0]	RW simulates a high threshold setting for a watchdog.	0xFFF

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

11.3.8 ADCx Watchdog Low Threshold Register (ADCx_WDLTR) (x=1/2) Offset address: 0x28

31	30 29	28	27 26 25 24 23 22 21	2019	18	17 16
Reserved						

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

Bit	Name access	describe	Reset value
[31:12] Reserved		RO is reserved.	0
[11:0] LT[11:0]		RW simulates a watchdog low threshold setting.	0

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

11.3.9 ADCx Regular Channel Sequence Register 1 (ADCx_RSQR1) (x=1/2) Offset

Address: 0x2C

31	30 29 28	27 26 25 24 23 22 21	2019	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	describe	Reset value
[31:24] Reserved		RO is reserved.	0
[23:20] L[3:0]	RW	The number of channels to be converted in the regular channel conversion sequence: 0000-1111: 1-16 transformations.	0
[19:15] SQ16[4:0]		The number (0-17) of the 16th conversion channel in the RW rule sequence.	0
[14:10] SQ15[4:0]		The number (0-17) of the 15th conversion channel in the RW rule sequence.	0
[9:5] SQ14[4:0]		The number (0-17) of the 14th conversion channel in the RW rule sequence.	0
[4:0] SQ13[4:0]		The number (0-17) of the 13th conversion channel in the RW rule sequence.	0

11.3.10 ADCx Regular Channel Sequence Register 2 (ADCx_RSQR2) (x=1/2) Offset

address: 0x30

31	30 29 28	27 26 25 24 23 22 21	2019	18	17 16
Reserved	SQ12[4:0]		SQ11[4:0]	SQ10[4:1]	

15	14 13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ10[0]	SQ9[4:0]				SQ8[4:0]					SQ7[4:0]				

Bit	Name access	describe	Reset value
[31:30] Reserved		RO is reserved.	0
[29:25] SQ12[4:0]		The number (0-17) of the 12th conversion channel in the RW rule sequence.	0
[24:20] SQ11[4:0]		The number (0-17) of the 11th conversion channel in the RW rule sequence.	0
[19:15] SQ10[4:0]		The number (0-17) of the 10th conversion channel in the RW rule sequence.	0
[14:10] SQ9[4:0]		The number (0-17) of the 9th conversion channel in the RW rule sequence.	0
[9:5] SQ8[4:0]		The number (0-17) of the 8th conversion channel in the RW rule sequence.	0
[4:0] SQ7[4:0]		The number (0-17) of the 7th conversion channel in the RW rule sequence.	0

11.3.11 ADCx Regular Channel Sequence Register 3 (ADCx_RSQR3) (x=1/2) Offset

address: 0x34

31	30 29 28	27 26 25 24 23 22 21	2019	18	17 16
Reserved	SQ6[4:0]		SQ5[4:0]	SQ4[4:1]	

15	14 13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ4[0]	SQ3[4:0]				SQ2[4:0]					SQ1[4:0]				

Bit	Name access	describe	Reset value
[31:30] Reserved	RO is reserved.		0
[29:25] SQ6[4:0]	The number (0-17) of the 6th conversion channel in the RW rule sequence.		0
[24:20] SQ5[4:0]	The number (0-17) of the 5th conversion channel in the RW rule sequence.		0
[19:15] SQ4[4:0]	The number (0-17) of the 4th conversion channel in the RW rule sequence.		0
[14:10] SQ3[4:0]	The number (0-17) of the 3rd conversion channel in the RW rule sequence.		0
[9:5] SQ2[4:0]	The number (0-17) of the second conversion channel in the RW rule sequence.		0
[4:0] SQ1[4:0]	The number (0-17) of the first conversion channel in the RW rule sequence.		0

11.3.12 ADCx Injection Channel Sequence Register (ADCx_ISQR) (x=1/2)

Offset Address: 0x38

31	30 29 28	27 26 25 24 23 22 21	2019	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	describe	Reset value
[31:22] Reserved	RO is reserved.		0
[21:20] JL[1:0]	RW	Number of channels to be converted in the injection 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 (0-17). Note: The software writes and assigns channel numbers (0-17) to the sequence to be converted.	0
[14:10] JSQ3[4:0]	The number (0-17) of the third conversion channel in the RW injection sequence.		0
[9:5] JSQ2[4:0]	The number (0-17) of the second conversion channel in the RW injection sequence.		0
[4:0] JSQ1[4:0]	The number (0-17) of the first conversion channel in the RW injection sequence.		0

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

For example, when JL[1:0]=3 (the sequencer contains 4 JSQ3[4:0] and JSQ4[4:0]); During the second injection conversion, the ADC will convert channels in the following order: JSQ1[4:0], JSQ2[4:0], ...

contains 4 JSQ3[4:0] and JSQ4[4:0]); During the second injection conversion, the ADC will convert channels in the following order: JSQ2[4:0], JSQ3[4:0]

When JL[1:0]=1 (there is a second injection conversion in the sequencer), the order of the ADC conversion channels is: first JSQ3[4:0], then JSQ4[4:0];

When JL[1:0]=0 (there is 1 in the sequencer) If ADCx_ISQR[21:0]=10 00111 00011 00111 00010, the ADC will switch channels in the following order: JSQ2[4:0], JSQ3[4:0] and JSQ4[4:0], it means that the scan conversion is performed in the following channel order: 7, 3, 7. During the second injection conversion, the ADC will only convert the JSQ4[4:0] channel.

11.3.13 Inject ADCy into data register x (ADCy_IDATARx) (y=1/2, x=1/2/3/4)

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

31	30 29 28	27 26 25 24 23 22 21	2019	18	17 16
Reserved					

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

Bit	Name access		describe	Reset value
[31:16] Reserved		RO is reserved.		0
[15:0] JDATA[15:0]		RO injection channel transforms data (data left-aligned or right-aligned).		0

11.3.14 ADCx Regular Data Register (ADCx_RDATAR) (x=1/2) Offset Address: 0x4C

31	30 29 28	27 26 25 24 23 22 21	2019	18	17 16
ADC2DATA[15:0]					

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

Bit	Name access		describe	Reset value
[31:16] ADC2DATA[15:0]	RO		Data converted by ADC2: In ADC1: In dual mode, these bits contain the rules for ADC2 conversion. Then the channel data. In ADC2: These bits are not used.	0
[15:0] DATA[15:0]		RO rule channel transformation data (data left-aligned or right-aligned).		0

11.3.15 ADCx Sampling Time Register (ADCx_AUX) (x=1/2) Offset Address: 0x54

31	30 29 28	27 26 25 24 23 22 21	2019	18	17 16
ADC_TO_DFSDM	Reserved				ADC_SMP_SEL17 ADC_SMP_SEL16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC_SMP_SEL15	ADC_SMP_SEL14	ADC_SMP_SEL13	ADC_SMP_SEL12	ADC_SMP_SEL11	ADC_SMP_SEL10	ADC_SMP_SEL9	ADC_SMP_SEL8	ADC_SMP_SEL7	ADC_SMP_SEL6	ADC_SMP_SEL5	ADC_SMP_SEL4	ADC_SMP_SEL3	ADC_SMP_SEL2	ADC_SMP_SEL1	ADC_SMP_SEL0

Bit	Name access		describe	Reset value
31	ADC_TO_DFSDM	RW	Enable the function of sending ADC module sampled data to DFSDM module: 1: Enable; 0: Off.	0
[30:18] Reserved		R0 is reserved.		0
[17:0] ADC_SMP_SELx		RW	x = 0-17, the sampling time of channel x can be enabled (only when SMPx is 0). 100-111): 1: SMPx=100: Sampling time 2.5 cycles; SMPx=101: Sampling time 3.5 cycles;	0

			SMPx=110: Sampling time 4.5 cycles; SMPx=111: Sampling time 5.5 cycles. 0: The sampling time is the corresponding SMPx configuration period.	

11.3.16 ADCx_TOUCHKEY Multiplexer Register (ADCx_DRV) (x=2) Offset address:

0x58

31	30 29 28	27 26 25 24 23 22 21	2019	18	17 16
Reserved					TKEY_DRV_EN

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

Bit	Name access		describe	Reset value
[31:17] Reserved		R0 is reserved.		0
16	TKEY_DRV_EN	RW	TOUCHKEY Multi-channel Masking Enable: 1: Enable; 0: Prohibited.	0
[15:0] TKEY_DRV_OUTEN		RW	TOUCHKEY Multi-channel masking and enabling: 1: Enable; 0: Prohibited.	0

Chapter 12 High-Speed Analog-to-Digital Conversion (HSADC)

The chip integrates a 10-bit high-speed analog-to-digital converter (HSADC) and provides 7 external sampling channels with a sampling rate of
With a capacity of up to 20 Msps, it can perform continuous conversions and supports DMA operations.

12.1 Main Features

Supports sampling of 7 external channels .

Supports receiving FIFO ;

Supports burst mode.

Supports DMA functionality

Supports Ping -Pong Storage Mode

Supports continuous conversion mode

Provides various transmission interruption flags and statuses .

Provides 1 interrupt vector

12.2 Functional Description

12.2.1 Powering on the HSADC module

A EN bit of the R32_HSADC_CFGR register being 1 indicates that the ADC module is powered on. When the ADC module enters power-on mode from power-off mode (EN=0), it indicates that the module is powered on.
After the electrical state (EN=1), a delay of tSTAB is required for the module to stabilize. Afterwards, the START bit is written to 1 for use as a soft state.
The EN bit is the start signal that initiates the ADC conversion. Clearing the EN bit to 0 will terminate the current conversion and put the ADC module into power-down mode.
In this state, the ADC consumes almost no power.

12.2.2 HSADC Clock

The HSADC clock source can be from the system clock (SYSCLK), PLL clock, USBHS_PLL clock, or ETH_PLL clock, and can be accessed via...
The RCC_CFGR2 register HSADCSRC[1:0] bits are selected, and the R32_HSADC_CFGR register CLKDIV[5:0] bits are also selected.
The HSADC clock is divided.

The total conversion time for the sampling period is calculated as follows:

TCONV = Sampling time + 4THSADCCLK ;

The sampling time is fixed at 1 THSADCCLK.

12.2.3 HSADC Channel Configuration

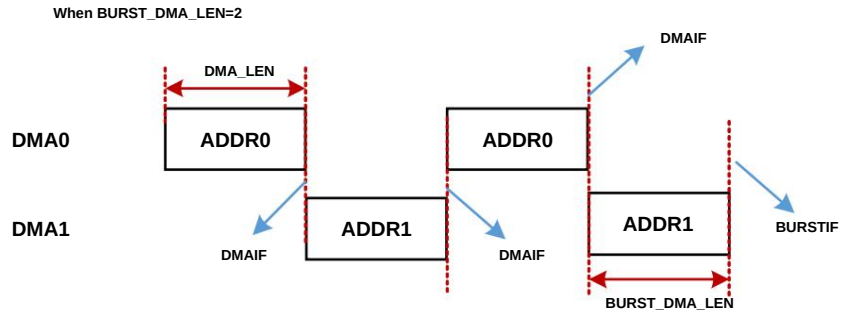
The HSADC module provides 7 external channel sampling sources.

12.2.4 HSADC DMA Function

HSADC channel conversion supports DMA functionality. The channel conversion value is stored in a single data register, R32_HSADC_DATAR.
If the data in the R32_HSADC_DATAR register is not retrieved in time, the ADC's DMA function can be enabled (if ping-pong transfer is not enabled).
Output (DMA0 can only be set). The hardware will generate a DMA request at the end of the channel conversion (EOC set) and transfer the converted data from...
The R32_HSADC_DATAR register is transferred to the user-specified destination address.

Supports burst transfer mode. During a burst transfer, the PPMODE bit is set to enable ping-pong storage mode. DMA transfers will terminate after reaching the transfer length.
Alternate between destination address ADDR0/1; enable abort of a burst transmission by setting BURST_END bit to 1 in the R32_HSADC_CTLR register.
The transmission will stop once the set burst length is reached. The DMA transmission length of the final burst can be adjusted as needed in R32_HSADC_CTLR2.
Set the BURST_DMA_LEN bit in the register.

The following diagram illustrates the burst transfer function using BURST_DMA_LEN=2 as an example:



12.2.5 HSADC Conversion Mode

HSADC only supports continuous conversion. In continuous conversion mode, the next conversion starts immediately after the previous ADC conversion ends.

The mode start event is initiated by setting the START bit to 1 in the R32_HSADC_CTLR1 register. Once the selected channel's conversion is complete, the mode will switch to the next active mode.

The swapped data is stored in the 10-bit R32_HSADC_DATAR register, and the EOC flag is set. If the EOCIE bit is set, it will trigger...

ADC interrupt.

12.3 Register Description

Table 12-1 List of HSADC Channel-Related Registers

name	Access Address Description: 0x40017400	Reset value
R32_HSADC_CFGR	HSADC Configuration Register; 0x40017404	0x00002400
R32_HSADC_CTLR1	HSADC Control Register 1; 0x40017408	0x00000000
R32_HSADC_CTLR2	HSADC Control Register 2; 0x4001740C	0x00000000
R32_HSADC_STATR	HSADC Status Register; 0x40017410 HSADC	0x00000000
R32_HSADC_DATAR	Data Register; 0x40017414 HSADC DMA	0x00000000
R32_HSADC_ADDR0	Receive Address Register 0; 0x40017418 HSADC DMA	0x00000000
R32_HSADC_ADDR1	Receive Address Register 1	0x00000000

12.3.1 HSADC Configuration Register (R32_HSADC_CFGR)

Offset Address: 0x00

31	30	29 28 27 26 25 24	23 22 21	2019	18	17 16
DMA_LEN[15:0]						

15	14	13 12 11 10 9 8	7	6 5	4 3 2	1	0
BURST_EN	PPMODE	CLKDIV[5:0]	WIDTH	SETUP[1:0]	CHSEL[2:0]	DMAEN	EN

Bit	name	access	describe	Reset value
[31:16]	DMA_LEN[15:0]	RW	DMA transfer length configuration	0
15	BURST_EN	RW	bits. Burst transfer enabled: 1: Enable; 0: Off.	0
14	PPMODE	RW	Ping-Pong Storage Mode Enabled: 1: Enable;	0

			0: Off.	
[13:8]	CLKDIV[5:0]	RW	High-speed ADC clock divider configuration bit: 000000: No frequency division; 000001: Frequency division by 2; 000010: 3 frequency divider; ... 111111:64 frequency division.	0x9
7	WIDTH	RW	Storage bit width configuration: 1: 8 bits; 0: 16 bits.	0
[6:5]	SETUP[1:0]	RW	Initial conversion setup time configuration bits: 00: 8 cycles; 01: 9 cycles; 10: 10 cycles; 11: 11 cycles.	0
[4:2]	CHSEL[2:0]	RW	Channel selection bit: 000: Input channel 0; ... 110: Input channel 6; 111: Reserved.	0
1	DMAEN	RW	Direct Memory Access (DMA) mode enabled: 1: Enable DMA mode; 0: DMA mode is off.	0
0	EN	RW	High-speed ADC Enable: 1: Enable; 0: Off.	0

12.3.2 HSADC Control Register 1 (R32_HSADC_CTLR1) Offset

Address: 0x04

31	30	29 28 27 26	25 24 23 22 21	2019	18	17 16
Reserved						

15	14	13	12 11	10	9	8	7	6	5	4	3	2	1	0
Reserved				BURSTIE	DMAIE	EOCIE	Reserved						BURST_END	START

Bit	name	access	describe	Reset value
[31:11]	Reserved	RO	reserved.	0
10	BURSTIE	RW	Burst transmission completion interrupt enabled: 1: Enable; 0: Off.	0
9	DMAIE	RW	DMA transfer completion interrupt enabled: 1: Enable;	0

			0: Off.	
8	EOCIE	RW	Interrupt enable upon completion of conversion: 1: Enable; 0: Off.	0
[7:2] Reserved		RO is reserved.		0
1	BURST_END	RW aborts a burst transmission.		0
0	START	WO	Initiates high-speed ADC conversion: 1: Turn on; 0: Off.	0

12.3.3 HSADC Control Register 2 (R32_HSADC_CTLR2) Offset Address:

0x08

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
BURST_LEN[15:0]														

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

Bit	name	access	describe	Reset value
[31:16]	BURST_LEN[15:0]	RW	burst transmission length configuration.	0
[15:0]	BURST_DMA_LEN[15:0]	RW	Burst transfer final DMA transfer length configuration bits. When the amount of data to be transmitted is not aligned with the DMA transfer length, the DMA transfer... Input length configuration bits.	0

12.3.4 HSADC Status Register (R32_HSADC_STATR) Offset Address:

0x0C

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		FIFO_CNT[2:0]			FIFO_OV	FIFO_FULL	FIFO_RDY	Reserved			PP_ADDR	RXNE	BURSTIF	DMAIF	EOCIF

Bit	name	access	describe	Reset value
[31:14]	Reserved	RO is reserved.		0
[13:11]	FIFO_CNT[2:0]	RO receives the current FIFO count value.		0
10	FIFO_OV	RO receives the FIFO overflow status flag.		0
9	FIFO_FULL	RO receives the FIFO full status flag.		0
8	FIFO_RDY	RO receives the FIFO non-empty status flag.		0
[7:5]	Reserved	RO reserved.		0
4	PP_ADDR	RO	Ping-pong storage mode cache address indicator bits: 1: Data is stored in DMA receive address 1; 0: Data is stored in DMA receive address 0.	0

3	RXNE	RO	The data register is not empty when the conversion is complete and the data is stored. In the register, this bit is set; a read operation on the data register can... This bit is cleared.	0
2	BURSTIF	RW1	Burst transmission completion interruption flag.	0
1	DMAIF	RW1	DMA transfer completed interrupt flag.	0
0	EOCIF	RW1	conversion complete interrupt flag.	0

12.3.5 HSADC Data Register (R32_HSADC_DATAR) Offset Address:

0x10

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

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

Bit	name	access	describe	Reset value
[31:10]	Reserved	RO	is reserved.	0
[9:0]	DR[9:0]	RW	is the conversion data register.	0

12.3.6 HSADC DMA Receive Address Register 0 (R32_HSADC_ADDR0) Offset

Address: 0x14

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
DMA_ADDR0[31:16]														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMA_ADDR0[15:0]															

Bit	name	access	describe	Reset value
[31:0]	DMA_ADDR0[31:0]	RW	DMA transfer address 0 configuration bit.	0

12.3.7 HSADC DMA Receive Address Register 1 (R32_HSADC_ADDR1) Offset

Address: 0x18

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
DMA_ADDR1[31:16]														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMA_ADDR1[15:0]															

Bit	name	access	describe	Reset value
31:0	DMA_ADDR1[31:0]	RW	DMA transfer address 1 configuration bit.	0

Chapter 13 Touch Key Detection (TKEY)

The touch detection control (TKEY) unit, utilizing the voltage conversion function of the ADC module, converts capacitance into voltage for sampling.

This achieves touch button detection functionality. The detection channel reuses 16 external channels of the ADC, and is implemented through the single-conversion mode of the ADC module.

Touch button detection is now in progress.

13.1 TKEY Function Description

ÿ TKEY enabled

The TKEY detection process requires the cooperation of the ADC module, so the ADC module must be powered on when using the TKEY function.

(ADON=1), then set the TKENABLE bit in the ADC_CTLR1 register to 1 to enable the TKEY unit function, and it can be accessed via TKITUNE.

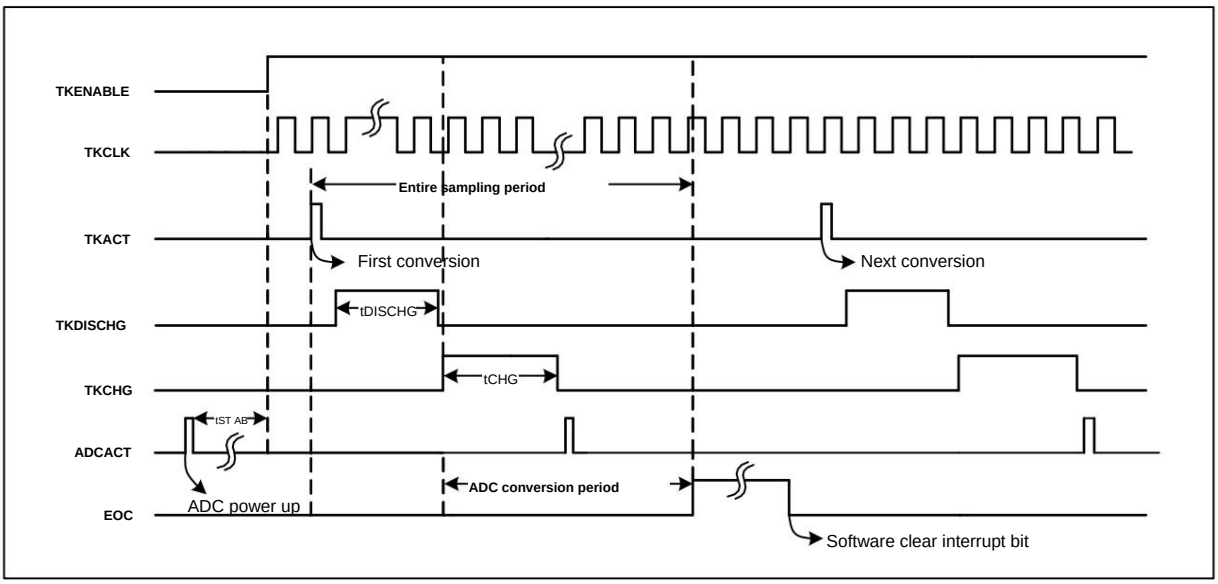
The charging current of the TKEY module is adjusted.

TKEY only supports single-channel conversion mode. Configure the channel to be converted as the first channel in the ADC module's rule group sequence.

Initiate the conversion (write to the TKEY_ACT_DCG register).

Note: Channel configuration conversion functionality can still be retained even if no conversion is performed.TKEY ADC

Figure 13-1 TKEY Timing Diagram



Programmable sampling time

The TKEY cell transition requires first discharging using several HCLK clock cycles (tDISCHG), and then discharging using several HCLK cycles.

The channel is charged and voltage is sampled during the period (tCHG). The number of charging cycles is the value of TKEY_CHGOFFSET. Each channel can be charged separately.

The sampling voltage is adjusted according to the same charging cycle.

13.2 TKEY Operation Steps

TKEY detection is an extended function under the ADC module. Its working principle is to enable hardware channel detection through "touch" and "non-touch" methods.

The sensed capacitance changes, and then the capacitance change is converted into a voltage change through a settable number of charge/discharge cycles. Finally, the voltage is...

The ADC module converts the data into digital values.

During sampling, the ADC needs to be configured to operate in single-pass, single-channel mode, and a single pass is initiated by the "write operation" of the TKEY_ACT register.

The specific process for replacement is as follows:

1) Initialize the ADC function, configure the ADC module as a single-conversion module, set the ACON bit to 1, and wake up the ADC module. Set ADC_CTLR1...

Set the TKENABLE bit in the register to 1 to enable the TKEY cell.

2) Set the channel to be converted, and write the channel number into the first conversion position in the ADC rule group sequence (ADC_RSQR3[4:0]).

L[3:0] is 1.

3) Write to the TKEY_CHGOFFSET register to set the channel's charging time (valid for the lower 16 bits) to adjust the charging time.

4) Write to the TKEY_ACT_DCG register, set the discharge time (lower 16 bits are valid), and start a sampling and conversion of TKEY.

5) Wait for the EOC conversion end flag in the ADC status register to be set to 1, and read the ADC_DR register to obtain the conversion value.

6) If a next conversion is required, repeat steps 2-5. If the channel charging sampling time does not need to be modified, steps 3 or 4 can be omitted.

13.3 TKEY Register Description

Table 13-1 List of TKEY1 Related Registers

name	Access address	describe	Reset value
R32_TKEY1_CHGOFFSET	0x4001243C	TKEY Charging Time Register	0x00000000
R32_TKEY1_ACT_DCG	0x4001244C	TKEY Startup and Discharge Time Register	0x00000000
R32_TKEY1_DR	0x4001244C	TKEY Data Register	0x00000000

Table 13-2 List of TKEY2 Related Registers

name	Access address	describe	Reset value
R32_TKEY2_CHGOFFSET	0x4001283C	TKEY Charging Time Register	0x00000000
R32_TKEY2_ACT_DCG	0x4001284C	TKEY Startup and Discharge Time Register	0x00000000
R32_TKEY2_DR	0x4001284C	TKEY Data Register	0x00000000

13.3.1 TKEYx Charging Time Register (TKEYx_CHGOFFSET) (x=1/2) Offset Address: 0x3C

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

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

Bit	name	access	describe	Reset value
[31:16]	Reserved	RO	is reserved.	0
[15:0]	TKCGOFFSET[15:0]	WO	TKEY Charging time configuration value. Total charging time TCHG = TKCGOFFSET	0

Note: This register mapping module's injected data register 1 (ADC_IDATAR1). Therefore, when this address register is "written",... When the ADC is in "operation", it is executed as the charging time (TKEY_CHGOFFSET), when performing a "read operation", it is executed as the module's injection time. TKEY ADC executes according to register 1 (ADC_IDATAR1).

13.3.2 TKEYx Start-up and Discharge Time Register (TKEYx_ACT_DCG) (x=1/2)

Offset address: 0x4C

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

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

Bit	name	access	describe	Reset value
[31:16]	Reserved	RO	is reserved.	0
[15:0]	TKACT_DCG[15:0]		Write the discharge time and initiate a TKEY channel check.	0

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

13.3.3 TKEYx Data Register (TKEYx_DR) (x=1/2)

Offset address: 0x4C

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

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

Bit	name	access	describe	Reset value
[31:16]	Reserved	RO	is reserved.	0
[15:0]	DATA[15:0]	RO	-converted data.	0

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

Chapter 14 Advanced Timers (ADTM)

The advanced timer module includes two 16-bit auto-reload increment/decrement counters (TIM1 and TIM8) with a 16-bit programmable prescaler. In addition to full general-purpose timer functions, it can be viewed as a three-phase PWM generator with six channels, featuring complementary PWM output with dead-time insertion, allowing the timer to be updated after a specified number of counter cycles for repeated counting, braking functions, etc. It can be used to measure pulse width or generate pulses, PWM waves, etc., and is applicable to motor control, power supply, and other fields.

Many functions of advanced timers are the same as those of general-purpose timers, and their internal structure is also the same. Therefore, advanced timers can be used with timers... The linking function works in conjunction with other TIM timers, providing synchronization or event linking capabilities.

14.1 Main Features

Key features of the advanced timer (TIM1/8) include:

ÿ 16-bit auto-reload counter, supporting up-counting, down-counting, and up-down-counting modes. ÿ 16-bit prescaler, dynamically adjustable division factor from 1 to 65536. ÿ Supports four independent compare-capture channels .

ÿ Each compare-capture channel supports multiple operating modes, such as input capture, output compare, PWM generation, and single-pulse output. ÿ Supports complementary output with programmable dead time. ÿ Supports external signal control of the timer . ÿ Supports updating the timer after a defined period using a repeating counter . ÿ Supports resetting or setting the timer to a defined state using a brake signal.

ÿ Supports DMA in multiple modes . ÿ Supports incremental encoders . ÿ Supports cascading and synchronization between timers.

14.2 Principles and Structure

This section mainly discusses the internal structure of advanced timers.

14.2.1 Overview As

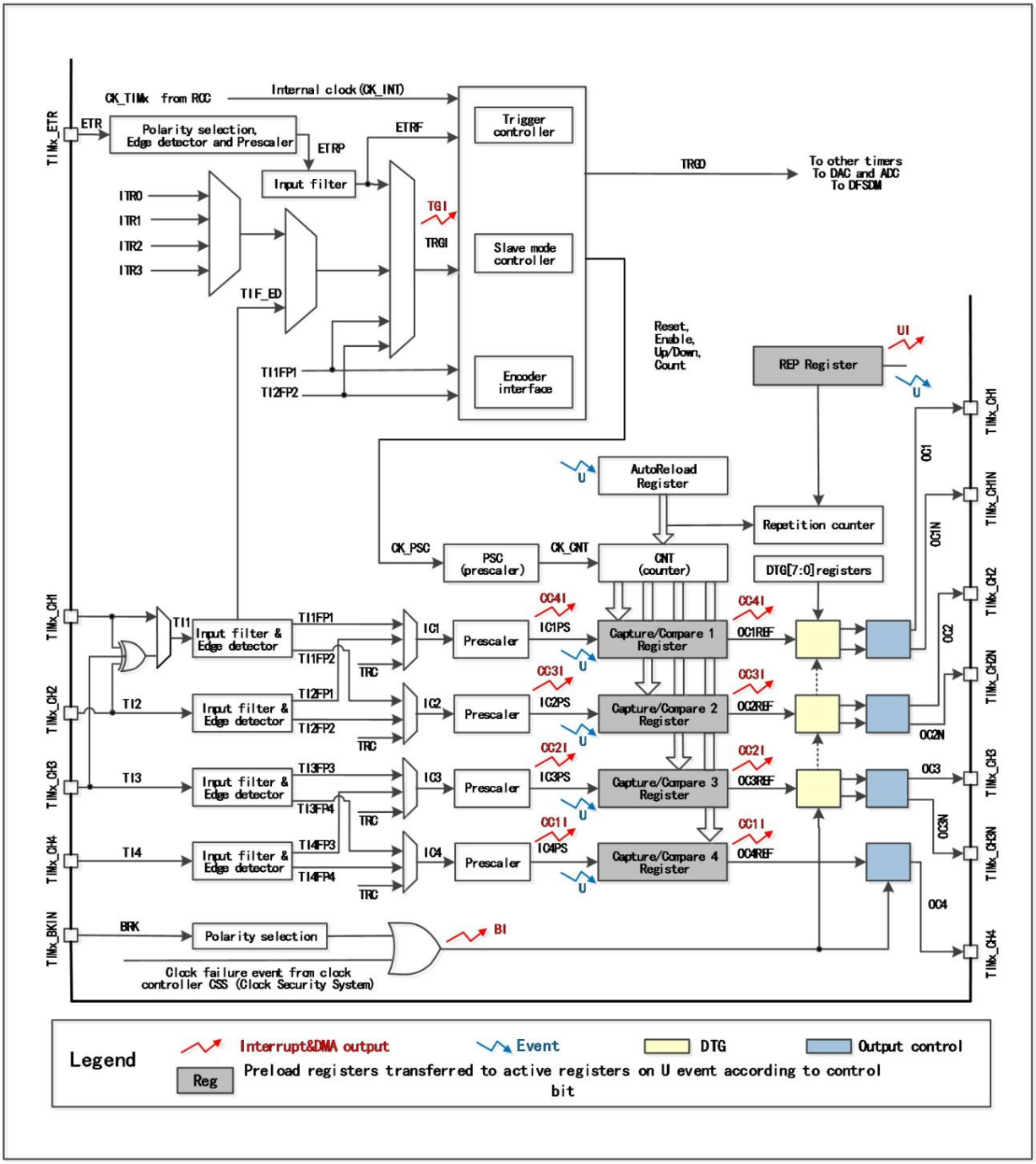
shown in Figure 14-1, the structure of an advanced timer can be roughly divided into three parts: the input clock section, the core counter section, and the comparison catcher. Access to the channel section.

The advanced timer's clock can originate from the HB bus clock (CK_INT), an external clock input pin (TIMx_ETR), other timers with clock output capabilities (ITRx), or the input of the compare/capture channel (TIMx_CHx). These input clock signals undergo various filtering and frequency division operations to become the CK_PSC clock, which is then output to the core counter section. Furthermore, these complex clock sources can also be used as TRGO outputs to other peripherals such as timers, ADCs, and DACs.

At the core of the advanced timer is a 16-bit counter (CNT). CK_PSC is divided by the prescaler (PSC) to become CK_CNT and output to CNT. CNT supports up-counting, down-counting, and up-down-counting modes, and has an automatic reload register (ATRLR) that reloads the initial value of CNT at the end of each counting cycle. An auxiliary counter also counts the number of times the ATRLR reloads the initial value of CNT; when this count reaches the number set in the repeat count register (RPTCR), a specific event is generated.

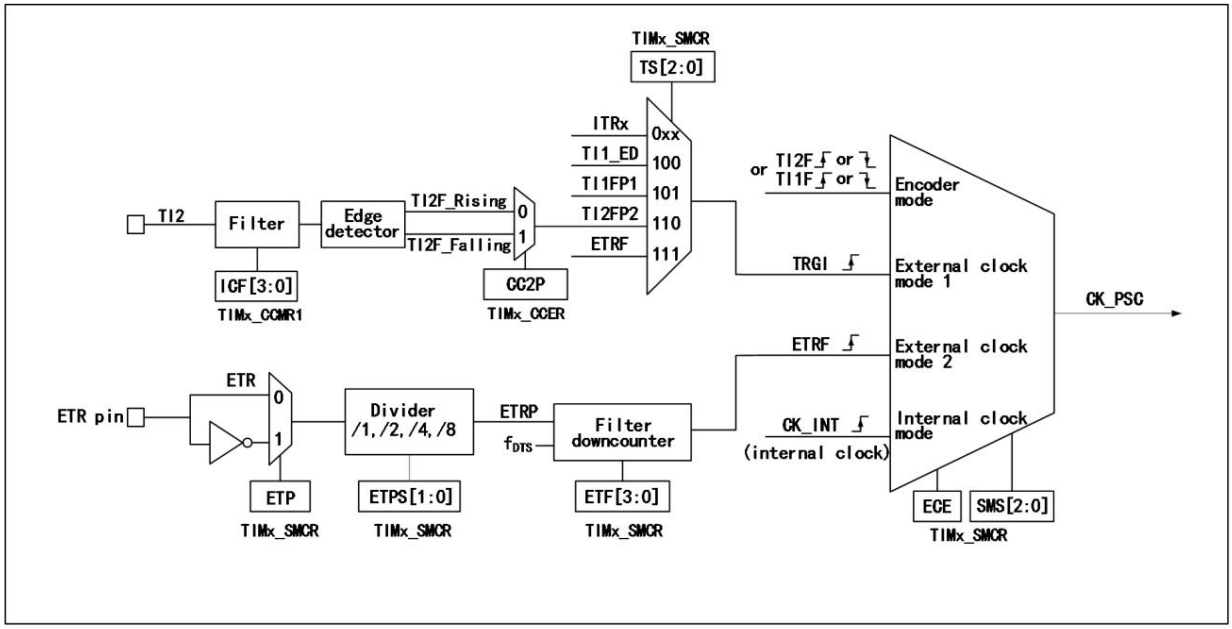
The advanced timer has four compare-capture channels, each capable of inputting pulses from a dedicated pin and outputting waveforms to the pin, meaning the compare-capture channels support both input and output modes. The inputs of each compare-capture register channel support filtering, frequency division, and edge detection, and also support inter-channel triggering, while providing a clock for the core counter (CNT). Each compare-capture channel has its own compare-capture register (CHxCVR), supporting comparison with the main counter (CNT) to output pulses.

Figure 14-1 Block diagram of advanced timer



14.2.2 Clock Input

Figure 14-2 Block diagram of CK_PSC source for advanced timers



The clock source for the advanced timer CK_PSC is diverse and can be categorized into four types:

1) The path of the external clock pin (ETR) input clock: ETRÿETRPÿETRF;

2) Internal HB clock input path: CK_INT;

3) Route from the compare capture channel pin (TIMx_CHx): TIMx_CHxÿTIxÿTIxFPx. This route is also used for the encoder module.

Mode;

4) Input from other internal timers: ITRx;

The actual operation can be divided into 4 categories by determining the input pulse selection of the SMS source from CK_PSC:

1) Select the internal clock source (CK_INT);

2) External clock source mode 1;

3) External clock source mode 2;

4) Encoder mode;

All four clock sources mentioned above can be selected through these four operations.

14.2.2.1 Internal Clock Source (CK_INT)

If the advanced timer is started while the SMS field is kept at 000b, then the internal clock source (CK_INT) is selected as the clock.

CK_INT is the same as CK_PSC.

14.2.2.2 External Clock Source Mode 1

External clock source mode 1 is enabled when the SMS field is set to 111b. When external clock source mode 1 is enabled, TRGI is selected.

As the source of CK_PSC, it's worth noting that the source of TRGI also needs to be selected through the TS field configuration. The TS field can select the following pulse types.

As a source of clocks:

1) Internal trigger (ITRx, x is 0, 1, 2, 3);

2) Compare the signal (TI1F_ED) after passing through the edge detector in capture channel 1;

3) Compare the signals TI1FP1 and TI2FP2 of the capture channel;

4) The signal ETRF from the external clock pin input.

14.2.2.3 External Clock Source Mode 2

Using external trigger mode 2, it can count on each rising or falling edge of the external clock pin input. Setting the ECE bit will...

External clock source mode 2 is used. When using external clock source mode 2, ETRF is selected as CK_PSC. The ETR pin is optionally inverted.

After passing through an ETP (Electronic Frequency Converter), an ETP (Electronic Frequency Divider), and an ETP (Electronic Frequency Divider), it becomes an ETP (Electronic Frequency Divider), and after passing through an ETF (Electronic Frequency Filter), it becomes an ETP (Electronic Frequency Filter).

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

14.2.2.4 Encoder Mode: Setting SMS to

001b, 010b, or 011b will enable encoder mode. Enabling encoder mode can be selected on TI1FP1 and TI2FP2.

This mode outputs a signal at a specific voltage level using another transition edge as the signal. This mode is used when an external encoder is connected.

For specific functions, please refer to section 14.3.9.

14.2.3 Counters and Peripherals

The CK_PSC input is fed to the prescaler (PSC) for frequency division. The PSC is 16-bit, and the actual division factor is equivalent to R16_TIMx_PSC.

The value is incremented by 1. CK_PSC becomes CK_CNT after passing through PSC. Changing the value of R16_TIM1_PSC does not take effect immediately, but will occur during the update event.

The update is then sent to the PSC. Update events include clearing the UG bit and resetting. The core of the timer is a 16-bit counter (CNT), CK_CNT being the most...

The final input will be given to CNT, which supports increment, decrement, and increment/decrement counting modes, and has an auto-reload value register.

(ATRLR) reloads the initial value of CNT at the end of each counting cycle. An auxiliary counter also records the ATRLR value.

The number of times the initial value of CNT is reloaded, when the number set in the Repetition Count Register (RPTCR) is reached, can trigger a specific event.

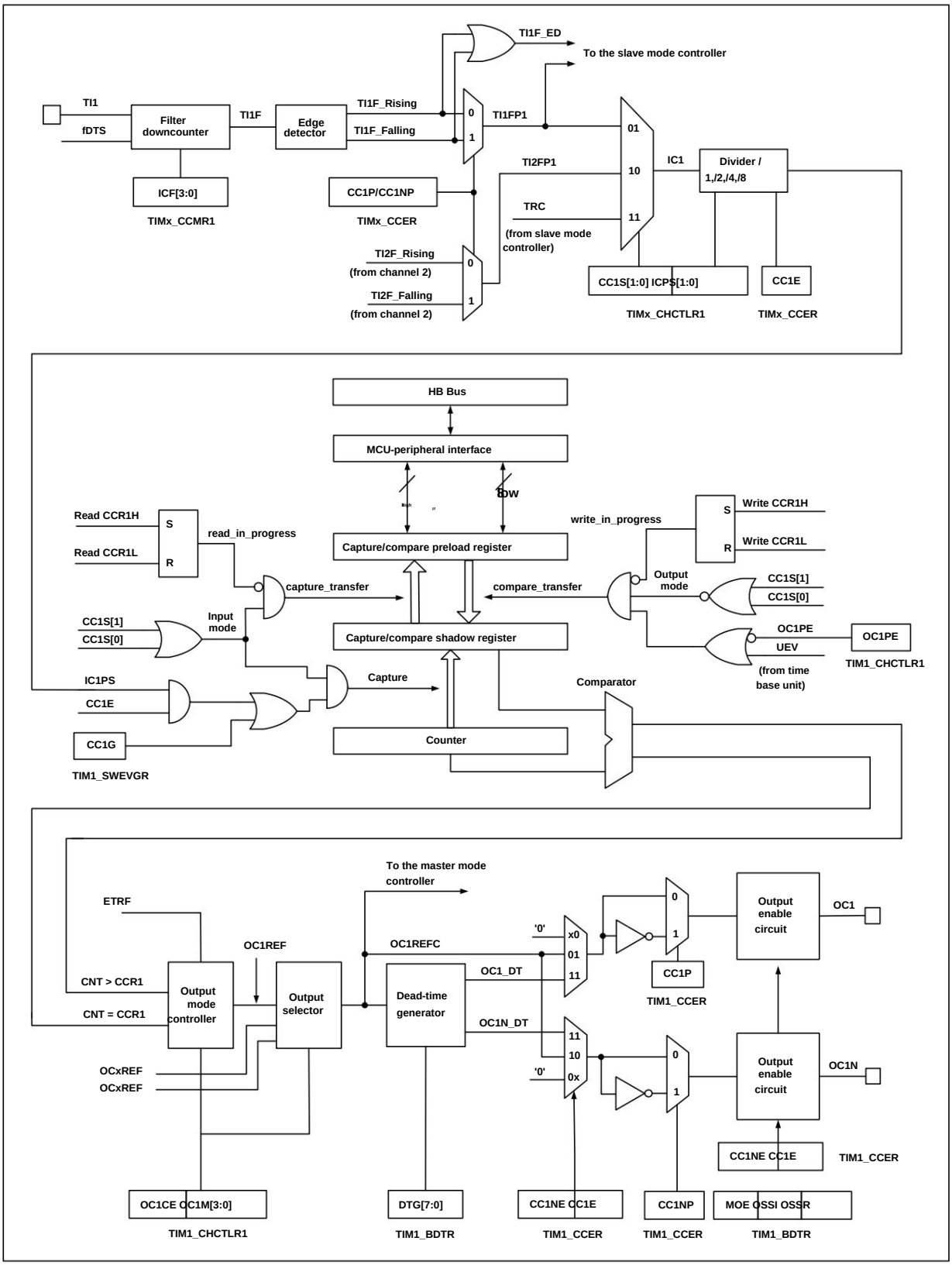
Item.

14.2.4 Compare the capture channel and surrounding area

The compare-capture channel is a key component for timers to implement complex functions. Its core is the compare-capture register, supplemented by external input sections.

It consists of digital filtering, frequency division and channel multiplexing, comparators in the output section and output control.

Figure 14-3 Block diagram of the comparison capture channel



The block diagram of the comparison capture channel is shown in Figure 14-3. The signal input from the channel x pin can be selected as TIx (the source of TI1). The source can be more than just CH1 (see the timer block diagram 14-1). Taking TI1 as an example: TI1 is filtered (ICF[3:0]) to generate TI1F, and then divided into TI1F_Rising and TI1F_Falling by the edge detector. These two signals are selected (CC1P) to generate TI1FP1. TI1FP1 and TI2FP1 from channel 2 are sent to CC1S to be selected as IC1. After being divided by ICPS, they are sent to the compare capture register.

The compare-capture register consists of a preload register and a shadow register. Read and write operations only manipulate the preload register. In capture mode, the capture occurs in 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 with the core counter (CNT).

14.3 Functions and Implementation

The implementation of complex functions in advanced timers involves operations on the timer's compare-capture channel, clock input circuitry, counter, and peripheral components. The timer's clock input can come from multiple clock sources, including the compare-capture channel input. The operation of selecting the compare-capture channel and clock source directly determines its functionality. The compare-capture channel is bidirectional and can operate in both input and output mod

14.3.1 Input Capture Mode

Input capture mode is one of the basic functions of a timer. The principle of input capture mode is that when a defined edge is detected on the ICxPS signal, a capture event occurs, 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. If interrupts or DMA are enabled, corresponding interrupts or DMA will also be generated. If CCxIF is already set when a capture event occurs, then the CCxOF bit will be set. CCxIF can be cleared by software or by hardware by reading the compare-capture register. CCxOF is cleared by software. The steps for using input capture mode are illustrated below using channel 1 as an

example:

1) Configure the CC1S domain and select the source of the IC1 signal. For example, set it to 10b and select TI1FP1 as the source of IC1. (Note: This option cannot be used.)

By default, the CC1S domain uses the compare capture module as the output channel.

2) Configure the IC1F field; these bits set the sampling frequency of the TI1 input and the length of the digital filter. The digital filter is controlled by an event counter.

The system consists of a counter that records N events before generating an output transition. 3) Configure the CC1P bit to set the polarity of TI1FP1. For example, keeping the CC1P bit low allows for a rising edge transition. 4) Configure the IC1PS field to set the IC1 signal as the division factor between IC1PS. For example, keeping IC1PS at 00b allows for no division. 5) Configure the CC1E bit to allow capturing the core counter (CNT) value into the compare capture register. 6) Configure the CC1IE and CC1DE bits depending on whether an interrupt or DMA is needed. When a

captured pulse is input to TI1, the core counter (CNT) value is recorded in the compare capture register, and CC1IF is set. If CC1IF was previously set, CC10F will also be set. If CC1IE is set, an interrupt will be generated; if CC1DE is set, a DMA request will be generated. An input capture event can be generated by software by writing to the event generation register (TIMx_SWEVGR).

14.3.2 Comparison Output Mode

Compare-to-output mode is one of the fundamental functions of a timer. The principle of compare-to-output mode is to output a specific change or waveform when the value of the core counter (CNT) matches the value of the compare-to-capture register. The OCxM field (in R16_TIMx_CHCTLRx) and the CCxP bit (in R16_TIMx_CCER) determine whether the output is a defined high/low level or a level toggle. A compare-to-capture event will also set... The CCxIF bit, if the CCxIE bit is preset, will generate an interrupt; if the CCxDE bit is preset, it will generate a DMA request. The steps to configure for compare output

mode are as follows: 1) Configure the clock source and auto-reload value for the core counter (CNT); 2) Set the count value to be compared to the compare capture register (R16_TIMx_CHxCVR); 3) If an interrupt is needed, set the CCxIE bit; 4) Keep OCxPE at 0 and disable the preload register of the compare register; 5) Set the output mode by setting the OCxM field and the CCxP bit; 6) Enable output by setting the CCxE bit; 7) Set the CEN bit to start the timer.

14.3.3 Forced Output Mode

The output mode of the timer's compare-capture channel can be forced to a predetermined level by software, without depending on the compare-capture register. Comparison of shadow registers and core counters.

The specific method is to set OCxM to 100b, which forces OCxREF to low; or to set OCxM to 101b, which forces OCxREF to high. It's important to note that forcing OCxM to 100b or 101b

will cause issues with the comparison between the internal core counter and the compare-capture register.

The process is still in progress, the corresponding flags are still being set, and interrupts and DMA requests are still being generated.

14.3.4 PWM Input Mode The PWM

input mode is used to measure the duty cycle and frequency of the PWM signal and is a special case of the input capture mode. Except for the following differences, the operation is the same as the input capture mode: PWM occupies two compare capture channels, and the input polarities of the two channels are set to opposite, one signal is set as the trigger input, and SMS is set to reset mode.

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

1) Set TI1 (TI1FP1) as the input to the IC1 signal. Set CC1S to 01b; 2) Set TI1FP1 to rising edge valid. Keep
CC1P at 0; 3) Set TI1 (TI1FP2) as the input to the IC2 signal. Set CC2S to 10b; 4) Set
TI1FP2 to falling edge valid. Set CC2P to 1; 5) Select TI1FP1 as the clock source. Set TS to 101b; 6) Set
SMS to reset mode, i.e., 100b; 7) Enable input capture. Set CC1E and CC2E;

The value of capture register 1 is the PWM period, and the value of capture register 2 is the duty cycle.

14.3.5 PWM Output Mode

PWM output mode is one of the basic functions of a timer. The most common method for PWM output mode is to use a reload value to determine the PWM frequency and a capture-compare register to determine the duty cycle. Setting 110b or 111b in the OCxM field enables PWM mode 1 or mode 2.
The OCxPE bit enables the preload register, and finally, the ARPE bit enables automatic reloading of the preload register. Since the value of the preload register is only 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 constantly compared; depending on the CMS bit, the timer can output edge-aligned or center-aligned PWM signals.
Edge Alignment

When using edge alignment, the core counter increments or decrements. In PWM mode 1, OCxREF is high when the core counter value is greater than the compare capture register value; it is low when the core counter value is less than the compare capture register value (e.g., when the core counter increments to the value of R16_TIMx_ATRLR and then returns to all zeros). In center alignment mode, the core
counter operates in

alternating increment and decrement mode. OCxREF rises and falls when the core counter and compare capture register values are the same. However, the timing of the compare flag setting differs across the three center alignment modes. When using center alignment mode, it is best to generate a software update flag (set the UG bit) before starting the core counter.

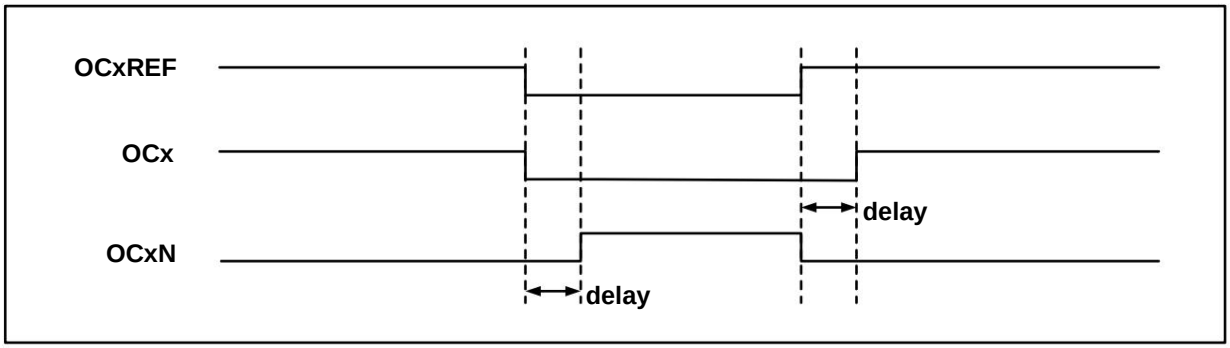
14.3.6 Complementary Output and Dead-

Time Comparison Capture Channels typically have two output pins (comparison capture channel 4 has only one output pin), capable of outputting two complementary signals (OCx and OCxN). The polarity of OCx and OCxN can be independently set via the CCxP and CCxNP bits, and the output enable can be independently set via the CCxE and CCxNE bits. Dead-time and other controls are performed via the MOE, OIS, OISN, OSSI, and OSSR bits. Enabling both OCx and OCxN outputs simultaneously inserts a dead timer; each channel has a 10-bit dead-time generator. If a braking circuit is present, the MOE b
OCx and OCxN are generated in association with OCxREF. If both OCx and OCxN are highly active, then OCx is the same as OCxREF, except that OCx...

The rising edge of OCxREF has a delay relative to the falling edge of the reference signal. OCxN is the opposite of OCxREF; its rising edge has a delay relative to the falling edge of the reference signal. If the delay is greater than the effective output width, no corresponding pulse will be generated.

Figure 14-4 shows the relationship between OCx and OCxN and OCxREF, and also shows the dead zone.

Figure 14-4 Complementary Output and Dead Zone



14.3.7 Braking Signal When a

braking signal is generated, the output enable and disable levels are modified according to the MOE, OIS, OISN, OSSI, and OSSR bits. However, OCx and OCxN will not be active at all times. The braking event source can originate from the braking input pin or a clock failure event, which is generated by the CSS (Clock Safety System). After system reset, the braking function is disabled by default (MOE bit low). Setting the BKE bit enables the

braking function. The polarity of the input braking signal can be set by setting BKP. The BKE and BKP signals can be written simultaneously. There is an HB clock delay before the actual writing, so it takes one HB cycle to correctly read the written value. When the selected level appears on the braking pin, the system will perform the following actions:

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

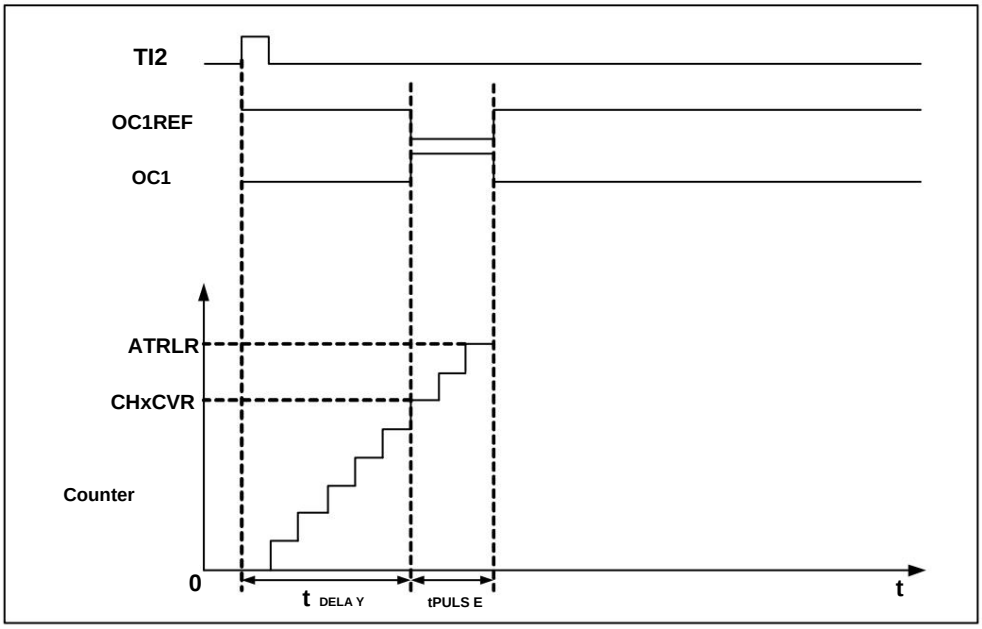
14.3.8 Single-Pulse Mode Single-

pulse mode allows the microcontroller to respond to a specific event by generating a pulse after a delay. The delay and pulse width are programmable. Setting the OPM bit stops the core counter when the next update event UEV (the counter toggles to 0) is generated. As shown in Figure 14-5, a rising edge needs to be detected on the

TI2 input pin, and after a delay of Tdelay, a pulse needs to be generated on OC1.

A positive pulse of length Tpulse:

Figure 14-5 Generation of a single pulse



1) Set TI2 as the trigger. Set the CC2S field to 01b, mapping TI2FP2 to TI2; set the CC2P bit to 0b, and set TI2FP2 to rising.

Edge detection; set the TS field to 110b, and set TI2FP2 as the trigger source; set the SMS field to 110b, and use TI2FP2 to start the counter;

2) Tdelay is determined by the value of the compare-capture register, and Tpulse is determined by the value of the auto-reload value register and the value of the compare-capture register.

Sure.

14.3.9 Encoder Mode Encoder

mode is a typical application of timers. It can be used to connect to the two-phase output of an encoder and to control the counting direction of the core counter.

Synchronized with the encoder's shaft direction, each pulse output by the encoder increments or decrements the core counter by one. Steps for using the encoder.

To: Set the SMS field to 001b (counting only on TI2 edges), 010b (counting only on TI1 edges), or 011b (counting on both TI1 and TI2 edges).

(Edge counting), connect the encoder to the input terminals of compare capture channels 1 and 2, and set a value for the reload value register. This value can be set...

It's a bit larger. In encoder mode, the timer's internal compare-capture register, prescaler, repeat count register, etc., all function normally.

The following table shows the relationship between the counting direction and the encoder signal.

Table 14-1 Relationship between counting direction and encoder signal in timer encoder mode

Counting effective edge	relative signal level	TI1FP1 signal edge		TI2FP2 signal	
		Rising edge	Falling edge	Rising edge	Falling edge
Counting only on the TI1 edge	High down count	up count	Low up count	Not counted	
	down count				
Counting only on TI2 edges	high	Not counted		Count up	Count down
	Low down count,	up count			
Counting on both sides of TI1 and TI2	High down count	Up count	Up count	Down count	
	Low up count	down count	down count	up count	

14.3.10 Timer Synchronization Mode: The

timer can output a clock pulse (TRGO) and also receive inputs from other timers (ITRx). Different timers have different ITRx values.

The source (TRGO of other timers) is different. The internal trigger connection of the timer is shown in Table 14-2.

Table 14-2 TIMx Internal Trigger Connections

From timer	ITR0	ITR1	ITR2	ITR3

	(TS=000)	(TS=001)	(TS=010)	(TS=011)
TIM1	TIM5	TIM2	TIM3	TIM4
TIM8	TIM1	TIM2	TIM4	TIM5

14.3.11 Bilateral Edge Capture Mode

The pulse measurement can be enabled by enabling dual-edge capture on the corresponding channel through the CAP_ED_CHx bit of the TIMx_AUX register.

For example: to capture the pulse width via channel 2, in the capture function configuration, select the source of the IC2 signal (TIMx_CHCTLR1). The CC2S bit of the register is 11b), enabling the double-edge capture function of channel 2 (the CAP_ED_CH2 bit of the TIMx_AUX register is 1). The dual-edge capture configuration is now complete.

The high and low level width values of the captured pulse can be read through the CH2CVR bit of the TIMx_CH2CVR register. Enabling the CAPLVL bit allows... The level corresponding to the captured value is indicated by bit[16] of the TIMx_CH2CVR register.

14.3.12 Dead-time asymmetry is addressed by

configuring the DT_VLU2 bits (the configuration value must be greater than 0 and less than TIMx_BDTR) based on complementary output configuration and dead-time control. The number of Tdtg values set in registers DTG[7:0]), configuring the DTP_MODE bit or DTN_MODE bit to select the dead value configured by DT_VLU2. The dead time occurs at the rising or falling edge of OCXREF. This method achieves dead time asymmetry.

14.3.13 Debug Mode When

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

14.4 Register Description

Table 14-3 List of TIM1 Related Registers

name	Access address	describe	Reset value
R16_TIM1_CTLR1	0x40012C00	Control Register 1	0x0000
R16_TIM1_CTLR2	0x40012C04	Control Register 2	0x0000
R16_TIM1_SMCFGR	0x40012C08	Slave Mode Control Register	0x0000
R16_TIM1_DMAINTENR	0x40012C0C	DMA/Interrupt Enable Register	0x0000
R16_TIM1_INTFR	0x40012C10	Interrupt Status Register	0x0000
R16_TIM1_SWEVGR	0x40012C14	Event Generation Register	0x0000
R16_TIM1_CHCTLR1	0x40012C18	Compare/Capture Control Register 1	0x0000
R16_TIM1_CHCTLR2	0x40012C1C	Compare/Capture Control Register 2	0x0000
R16_TIM1_CCER	0x40012C20	Compare/Capture Enable Register	0x0000
R16_TIM1_CNT	0x40012C24	Counter	0x0000
R16_TIM1_PSC	0x40012C28	Count Clock Prescaler 0x40012C2C	0x0000
R16_TIM1_ATRLR	Auto Reload Value Register 0x40012C30		0xFFFF
R16_TIM1_RPTCR	Repeat Count Value Register 0x40012C34		0x0000
R32_TIM1_CH1CVR	Compare/Capture Register 1 0x40012C38		0x0000
R32_TIM1_CH2CVR	Compare/Capture Register 2 0x40012C3C		0x0000
R32_TIM1_CH3CVR	Compare/Capture Register 3 0x40012C40		0x0000
R32_TIM1_CH4CVR	Compare/Capture Register 4 0x40012C44 Brake		0x0000
R16_TIM1_BDTR	and Dead Zone Register		0x0000
R16_TIM1_DMACFGR	0x40012C48	DMA Control Register	0x0000
R16_TIM1_DMAADR	0x40012C4C: DMA Address Register in Continuous Mode		0x0000

R16_TIM1_AUX	0x40012C50 Dual-edge capture register	0x0000

Table 14-4 List of TIM8 Related Registers

name	Access address	describe	Reset value
R16_TIM8_CTLR1	0x40013400 Control Register 1		0x0000
R16_TIM8_CTLR2	0x40013404 Control Register 2		0x0000
R16_TIM8_SMCFGR	0x40013408 Slave Mode Control Register		0x0000
R16_TIM8_DMAINTENR	0x4001340C DMA/Interrupt Enable Register		0x0000
R16_TIM8_INTFR	0x40013410 Interrupt Status Register		0x0000
R16_TIM8_SWEVGR	0x40013414 Event Generation Register		0x0000
R16_TIM8_CHCTLR1	0x40013418 Compare/Capture Control Register 1		0x0000
R16_TIM8_CHCTLR2	0x4001341C Compare/Capture Control Register 2		0x0000
R16_TIM8_CCER	0x40013420 Compare/Capture Enable Register		0x0000
R16_TIM8_CNT	0x40013424 Counter 0x40013428		0x0000
R16_TIM8_PSC	Count Clock Prescaler 0x4001342C Auto Reload		0x0000
R16_TIM8_ATRLR	Value Register 0x40013430 Repeat Count Value		0xFFFF
R16_TIM8_RPTCR	Register 0x40013434 Compare/Capture Register		0x0000
R32_TIM8_CH1CVR	1 0x40013438 Compare/Capture Register 2		0x0000
R32_TIM8_CH2CVR	0x4001343C Compare/Capture Register 3		0x0000
R32_TIM8_CH3CVR	0x40013440 Compare/Capture Register 4		0x0000
R32_TIM8_CH4CVR	0x40013444 Brake and Dead Zone Register		0x0000
R16_TIM8_BDTR			0x0000
R16_TIM8_DMACFGR	0x40013448	DMA Control Register	0x0000
R16_TIM8_DMAADR	0x4001344C DMA Address Register 0x40013450 Double		0x0000
R16_TIM8_AUX	Edge Capture Register		0x0000

14.4.1 Control Register 1 (TIMx_CTLR1) (x=1/8) Offset Address: 0x00 Name

Bit		access		Reset value
15	CAPLVL	RW	Describes the capture level indicator enable in dual-edge capture mode. 1: Enable indicator function; 0: Disable indicator function. Note: When enabled, CHxCVR's [16] indicates the level corresponding to the captured value.	0
14	CAPOV	RW	Capture value mode configuration. 1: When a counter overflows before capture, the CHxCVR value is... 0xFFFF; 0: The captured value is the actual counter value.	0
[13:10] Reserved		RO is reserved.		0
[9:8] CKD[1:0]		RW	These two bits are defined in the timer clock (CK_INT) frequency and dead time. The interval is used by the dead-time generator and digital filter (ETR, TIx). The frequency division ratio between sampling clocks: 00: Tdts = Tck_int; 01: Tdts = 2 x Tck_int; 10: Tdts = 4 x Tck_int;	00b

			11: Retained.	
7	ARPE	RW	Automatic reinstallation pre-installation enable bit: 1: Enable the Automatic Reload Value Register (ATRLR); 0: Disable automatic reload of value register (ATRLR).	0
[6:5] CMS[1:0]		RW	Center alignment mode selection: 00: Edge-aligned mode. The counter moves upwards based on the direction bit (DIR). Or count downwards. 01: Center Alignment Mode 1. The counter alternates between counting up and down. Number. The channel configured for output (CCxS=00 in the CHCTLRx register). The output compare interrupt flag is only triggered when the counter counts down. set up. 10: Center Alignment Mode 2. The counter alternates between counting up and down. Number. The channel configured for output (CCxS=00 in the CHCTLRx register). The output compare interrupt flag is only triggered when the counter is counting up. set up. 11: Center Alignment Mode 3. The counter alternates between counting up and down. Number. The channel configured for output (CCxS=00 in the CHCTLRx register). The output compares the interrupt flag bit, and the counter counts up and down. The time is set. Note: When the counter is enabled (CEN=1), switching from edge-aligned mode to center-aligned mode is not allowed.	00b
4	DIR	RW	Counter direction: 1: The counter's counting mode is down; 0: The counter is in increment mode. Note: This bit is invalid when the counter is configured in center-aligned mode or encoder mode.	0
3	OPM	RW	Single pulse mode: 1: The counter stops when the next update event occurs (CEN is cleared). Bit); 0: The counter does not stop when the next update event occurs.	0
2	URS	RW	Update request source; the software selects the source of the UEV event using this bit. 1: If update interrupts or DMA requests are enabled, only the counter will be active. An update interrupt or DMA request is only generated when there is an overflow or underflow. 0: If update interrupts or DMA requests are enabled, then either of the following will occur: The device generates an update interrupt or a DMA request. - Counter overflow/underflow -Set UG position - Updates generated from mode control.	0
1	UDIS	RW	Disabling updates allows/disallows the generation of UEV events via this bit. born. 1: Disable UEV. No update events are generated, and all registers (ARR, ...) are updated. PSC and CHxCVR retain their values. If the UG bit is set or A hardware reset is issued from the mode controller, then the counter and pre-... The frequency divider was reinitialized; 0: UEV enabled. The update (UEV) event is generated by any of the following events.	0

			born. - Counter overflow/underflow -Set UG position -Updates generated from mode control The cached registers are loaded with their preloaded values.	
0	CEN	RW	Enable counter: 1: Enable counter. Note: This is in the software setting. CEN After this step, the external clock, gating mode, and encoder mode can operate. The trigger mode can be automatically set via hardware. CEN Bit 0: Counter disabled.	0

14.4.2 Control Register 2 (TIMx_CTLR2) (x=1/8) Offset Address: 0x04

Bit	name	access	describe	Reset value
15	Reserved	RO is reserved.		0
14	OIS4	RW	Output idle state 4: 1: When MOE=0, if OC4N is implemented, then OC4=1 after the dead zone; 0: When MOE=0, if OC4N is implemented, then OC4=0 after the dead zone. Note: LOCK (TIMx_BDTR register) level 1 has been set. After that, this bit cannot be modified.	0
13	OIS3N	RW	Output idle state 3: 1: When MOE=0, OC3N=1 after the dead zone; 0: When MOE=0, OC3N=0 after the dead zone. Note: LOCK (TIMx_BDTR register) level 1 has been set. After that, this bit cannot be modified.	0
12	OIS3	RW outputs idle state 3, see OIS4.		0
11	OIS2N	RW outputs idle state 2, see OIS3N.		0
10	OIS2	RW outputs idle state 2, see OIS4.		0
9	OIS1N	RW outputs idle state 1, see OIS3N.		0
8	OIS1	RW outputs idle state 1, see OIS4.		0
7	TI1S	RW	TI1 Selection: 1: The TIMx_CH1, TIMx_CH2, and TIMx_CH3 pins are XORed together. Connect to TI1 input; 0: The TIMx_CH1 pin is directly connected to the TI1 input.	0
[6:4] MMS[2:0]		RW	Master mode selection: These 3 bits are used to select whether to send to the slave in master mode. Timer synchronization information (TRGO). Possible combinations are as follows: 000: Reset – The UG bit of the TIMx_EGR register is used as a trigger. Output (TRGO). If it is a reset generated by a trigger input (from...	000b
			(When the mode controller is in reset mode), the signal on TRGO is relative to... There will be a delay in 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 scheduled commands at the same time.	

			A timer or counter enables a function to run for a specified period of time. The power signal is triggered by the CEN control bit and the gated mode trigger input. The logical OR of the signal is generated. When the counter enable signal is triggered... There will be a delay on TRGO during input unless master/slave is selected. Mode (see description of the MSM bit in the TIMx_SMCR register); 010: Update – The update event is selected as the trigger output (TRGO). Example For example, the clock of a master timer can be used as a slave timer. Prescaler; 011: Comparison Pulse – Announced upon successful capture or comparison. When you want to set the CC1IF flag (even if it is already high). The trigger output sends out a positive pulse (TRGO); 100: Compare – The OC1REF signal is used as a trigger output. (TRGO); 101: Comparison – The OC2REF signal is used as a trigger output. (TRGO); 110: Compare – The OC3REF signal is used as a trigger output. (TRGO); 111: Comparison – The OC4REF signal is used as a trigger output. (TRGO).	
3	CCDS	RW	DMA selection for capture comparison: 1: When an update event occurs, send a DMA request for CHxCVR; 0: When CHxCVR occurs, a DMA request for CHxCVR is generated.	0
2	CCUS	RW	Compare capture control update selection bit: 1. If the CCPC bit is set, you can adjust the settings by setting the COM bit or the TRGI bit. Update them on a rising edge; 0: If CCPC is set, they can only be updated by setting the COM bit. Note: This bit only applies to channels with complementary outputs.	0
1	Reserved	RO reserved.		0
0	CCPC	RW	Compare captured preload control bits: 1: The CCxE, CCxNE, and OCxM bits are preloaded; setting this bit... After that, they are only updated after the COM bit is set; 0: The CCxE, CCxNE, and OCxM bits are not preloaded. Note: This bit only applies to channels with complementary outputs.	0

14.4.3 From the Mode Control Register (TIMx_SMCFGR) (x=1/8) Offset Address: 0x08 Name

Bit		access	The	Reset value
15	ETP	RO	description of ETR triggers polarity selection; this bit selects whether to directly input ETR or... Invert the input ETR. 1: Invert ETR; it is active low or on the falling edge. 0: ETR, high level or rising edge valid.	0
14	ECE	RW	External clock mode 2 enabled: 1: Enable external clock mode 2; 0: Disable external clock mode 2. Note 1: Slave mode can be used simultaneously with2external clock mode:	0

			Bit mode, gated mode, and trigger mode; however, at this time..TRGI can be connected to ETRF (TS bit cannot be '111'). Note 2: External clock mode and external clock mode are used simultaneously. When 12 is available, the input of the external clock is ETRF.	
[13:12] ETPS[1:0]		RW	External trigger signal (ETRP) frequency division, the maximum frequency of this signal is not It can exceed 1/4 of the TIMxCLK frequency, and this domain can be used to reduce [frequency]. frequency: 00: Disable prescaler; 01: ETRP frequency divided by 2; 10: ETRP frequency divided by 4; 11: ETRP frequency divided by 8.	00b
[11:8] ETF [3:0]		RW	Externally triggered filtering, in fact, a digital filter is an event meter. A counter that records N events using a certain sampling frequency. This will then result in an output jump. 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; 0110: Sampling frequency Fsampling = Fdts/4, N = 6; 0111: Sampling frequency Fsampling = Fdts/4, N = 8; 1000: Sampling frequency Fsampling = Fdts/8, N = 6; 1001: Sampling frequency Fsampling = Fdts/8, N = 8; 1010: Sampling frequency Fsampling = Fdts/16, N = 5; 1011: Sampling frequency Fsampling = Fdts/16, N = 6; 1100: Sampling frequency Fsampling = Fdts/16, N = 8; 1101: Sampling frequency Fsampling = Fdts/32, N = 5; 1110: Sampling frequency Fsampling = Fdts/32, N = 6; 1111: Sampling frequency Fsampling=Fdts/32, N=8.	0000b
7	MSM	RW	Master/Slave Mode Selection: 1: Events on the trigger input (TRGI) are delayed to allow for... Perfect connection between the current timer (via TRGO) and its slave timer Synchronization. This requires synchronizing several timers to a single external timer. It is very useful in departmental events; 0: Not effective.	0
[6:4] TS[2:0]		RW	Trigger selection field: These 3 bits are used to select the trigger input of the synchronization counter. Source: 000: Internal Trigger 0 (ITR0); 001: Internal Trigger 1 (ITR1); 010: Internal Trigger 2 (ITR2); 011: Internal Trigger 3 (ITR3); 100: TI1 edge detector (TI1F_ED); 101: Filtered timer input 1 (TI1FP1); 110: Filtered timer input 2 (TI2FP2);	000b

			111: External Trigger Input (ETRF); The above only changes when SMS is 0. Note: See Table 14-2 for details.	
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. Mode. 000: Driven by the internal clock CK_INT; 001: Encoder Mode 1, based on the TI1FP1 level, core calculation... The counter increments or decrements on the edge of the TI2FP2; 010: Encoder mode 2, based on the TI2FP2 level, core calculation... The counter increments or decrements on the edge of TI1FP1; 011: Encoder mode 3, based on the input level of another signal, The core counter increments and decrements on the edges of TI1FP1 and TI2FP2; 100: Reset mode, the rising edge of the trigger input (TRGI) will initialize The counter is updated, and a signal to update the register is generated; 101: Gated mode, counting occurs when the trigger input (TRGI) is high. The counter's clock is on; when the trigger input goes low, the counter stops. The start and stop of the counter are controlled; 110: Trigger mode, the counter is triggered on the rising edge of the trigger input TRGI. Only the start-up of the counter is controlled; 111: External clock mode 1, selected trigger input (TRGI) The rising edge drives the counter.	000b

14.4.4 DMA/Interrupt Enable Register (TIMx_DMAINTENR) (x=1/8) Offset Address: 0x0C

Bit	name	access	describe	Reset value
15	Reserved	RO	reserved.	0
14	TDE	RW	Trigger DMA request enable bit: 1: Allow DMA requests to be triggered; 0: Disable DMA requests.	0
13	COMDE	RW	COM DMA request enable bit: 1: Allow COM DMA requests; 0: Disable DMA requests for COM.	0
12	CC4DE	RW	Compare the DMA request enable bits of capture channel 4: 1: Allow comparison of DMA requests for capturing channel 4; 0: Disable comparison of DMA requests for capture channel 4.	0
11	CC3DE	RW	Compare the DMA request enable bits of capture channel 3: 1: Allow comparison of DMA requests for capturing channel 3; 0: Disable comparison of DMA requests for capture channel 3.	0
10	CC2DE	RW	Compare the DMA request enable bits of capture channel 2: 1: Allow comparison of DMA requests for capturing channel 2; 0: Disable comparison of DMA requests for capture channel 2.	0
9	CC1DE	RW	Compare the DMA request enable bits of capture channel 1: 1: Allow comparison of DMA requests for capturing channel 1; 0: Disable comparison of DMA requests for capturing channel 1.	0

8	UDE	RW	Updated DMA request enable bit: 1: Allow DMA requests for updates; 0: DMA requests for updates are prohibited.	0
7	BIE	RW	Brake interruption enable bit: 1: Allow braking interruption; 0: Braking interruption is prohibited.	0
6	TIE	RW	Trigger interrupt enable bit: 1: Enable interrupt triggering; 0: Interrupt triggering is disabled.	0
5	COMIE	RW	COM interrupt enable bit: 1: Enable COM interrupts; 0: Disable COM interrupts.	0
4	CC4IE	RW	Compare the interrupt enable bit of capture channel 4: 1: Enable comparison capture channel 4 interrupt; 0: Disable comparison capture channel 4 interruption.	0
3	CC3IE	RW	Compare the interrupt enable bit of capture channel 3: 1: Enable comparison capture channel 3 interrupt; 0: Disable comparison capture channel 3 interruption.	0
2	CC2IE	RW	Compare the interrupt enable bit of capture channel 2: 1: Enable comparison capture channel 2 interrupt; 0: Disable comparison capture channel 2 interruption.	0
1	CC1IE	RW	Compare the interrupt enable bit of capture channel 1: 1: Enable comparison capture channel 1 interrupt; 0: Disable comparison capture channel 1 interruption.	0
0	UIE	RW	Update the interrupt enable bit: 1: Allow update interruptions; 0: Prevent update interruption.	0

14.4.5 Interrupt Status Register (TIMx_INTFR) (x=1/8)

Offset address: 0x10

Bit	name	access	describe	Reset value
[15:13] Reserved		RO is reserved.		0
12	CC4OF	RW0 compares the capture channel 4 duplicate capture flag.		0
11	CC3OF	RW0 compares the capture channel 3 duplicate capture flag.		0
10	CC2OF	RW0 compares the capture channel 2 duplicate capture flag. It		0
9	CC1OF	RW0	also compares the capture channel 1 duplicate capture flag; this is only used for comparing captures. When the channel is configured for input capture mode. This flag is set by hardware. Writing 0 to the software will clear this bit. 1: When the counter value is captured into the capture compare register, CC1IF The status has been set; 0: No duplicate captures were generated.	0
8	Reserved	RO is reserved.		0
7	BIF	RW0	This is the brake interruption flag, which is activated by hardware once the brake input is valid. The position can be cleared by the software.	0

			1: A set valid level was detected on the brake pin input; 0: No braking event occurs.	
6	TIF	RW0	The trigger interrupt flag is set by hardware when a trigger event occurs. The position bit is cleared by software. Triggering events include those from gating mode. In other modes, a valid edge is detected at the TRGI input. Or any edge in gating mode. 1: Trigger event generated; 0: No trigger event is generated.	0
5	COMIF	RW0	The COM interrupt flag is set by the hardware once a COM event occurs. Set, cleared by software. COM events include CCxE, CCxNE, and OCxM. Updated. 1: A COM event is generated; 0: No COM event was generated.	0
4	CC4IF	RW0	compares the interrupt flag of capture channel 4.	0
3	CC3IF	RW0	compares the interrupt flag of capture channel 3.	0
2	CC2IF	RW0	compares the interrupt flag of capture channel 2.	0
1	CC1IF	RW0	Compares the interrupt flag of capture channel 1. If the compare capture channel is configured for output mode: This bit is set by hardware when the counter value matches the comparison value, but... Except in centrally symmetric 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 occurred. If the comparison 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. Alternatively, the capture register can be cleared by reading the compare register. 1: The counter value has been captured by comparing capture register 1; 0: No input capture was generated.	0
0	UIF	RW0	Update interrupt flag; this bit is set by hardware when an update event occurs. The bit is cleared by the software. 1: An update interruption occurred; 0: No update event occurred. The following situations will trigger an update event: If UDIS=0, when the repeat counter value overflows or underflows; If URS=0 and UDIS=0, when the UG bit is set, or when the software is used to... When the counter core is reinitialized; If URS=0 and UDIS=0, the event is retried when the counter CNT is triggered. During initialization.	0

14.4.6 Event Generation Register (TIMx_SWEVGR) (x=1/8)

Offset address: 0x14 bit

name		access	describe	Reset value
[15:8] Reserved		RO	Reserved.	0
7	BG	WO	This is the brake event generation bit, set and cleared by software to generate... A braking incident occurred.	0

			1: A braking event is generated. At this time, MOE=0 and BIF=1. If enabled... The corresponding interrupts and DMA will then generate the corresponding interrupts and DMA. 0: No action.	
6	TG	WO	The event trigger bit is set by software and cleared by hardware. This generates a triggering event. 1: A trigger event is generated, and the TIF bit is set. If the corresponding bit is enabled...	0
			Interrupts and DMA will generate corresponding interrupts and DMA operations; 0: No action.	
5	COMG	WO	Compare capture control update generation bits. Generate compare capture control update. Event. This bit is set by software and automatically cleared by hardware. 1: When CCPC=1, updating the CCxE, CCxNE, and OCxM bits is allowed; 0: No action. Note: This bit only applies to channels with complementary outputs (channels 1, 2, and 3).	0
			efficient.	
4	CC4G	WO	Compare capture event generates bit 4. Compare capture event 4 is generated.	0
3	CC3G	WO	Compare capture event generates bit 3. A comparison capture event 3 is generated.	0
2	CC2G	WO	Compare capture event generation bit 2. Generates comparison capture event	0
1	CC1G	WO	2. Compare capture event generation bit 1, generates comparison capture event 1. This bit is set by software and cleared by hardware. It is used to generate a comparison. Capture events. 1: Generate a compare capture event on compare capture channel 1: If the comparison capture channel 1 is configured as output: Set the CC1IF bit. If the corresponding interrupt and DMA are enabled, a corresponding interrupt will be generated. Corresponding interrupts and DMA; If the comparison capture channel 1 is configured as the input: The current core counter value is captured in compare capture register 1; Setting the CC1IF bit will generate an interrupt if the corresponding interrupt and DMA are enabled. The corresponding interrupts and DMA. If CC1IF is already set, then CC1OF is set. Bit. 0: No action.	0
0	UG	WO	Update event generation bit: This bit is set by the software to generate an update event. Automatically reset by hardware. 1: Initialize the counter and generate an update event; 0: No action. Note: The prescaler counter is also cleared, but the prescaler coefficient remains unchanged. In center-symmetric mode or increment mode, the core counter is cleared; in decrement mode, the core counter takes the value from the reload register.	0

14.4.7 Compare/Capture Control Register 1 (TIMx_CHCTLR1) (x=1/8) Offset Address: 0x18. The channel

can be used for input

(capture mode) or output (compare mode), and the channel direction is defined by the corresponding CCxS bits. This register...

The functions of other bits differ between input and output modes. OCxx describes the channel's function in output mode, and ICxx describes the channel's function in output mode.

Functionality 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):

Bitname		access	describe	Reset value
15	OC2CE	RW	Compare capture channel 2 to clear enable bit: 1. Clear the OC2REF bit once an ETRF input high level is detected. zero; 0: OC2REF is not affected by ETRF input.	0
[14:12]	OC2M[2:0]	RW	Compare capture channel 2 mode setting field: These 3 bits define the action of the output reference signal OC2REF, and OC2REF determines the values of OC2 and OC2N. OC2REF is a high-level signal. The effective levels of OC2 and OC2N depend on CC2P and CC2NP, respectively. Bit. 000: Freeze. Compare the value of the capture register with the core counter. The comparison value has no effect on OC2REF; 001: Forced to active level. When the core counter is compared with the capture level. If the value of register 2 is the same, force OC2REF to go high; 010: Forced to invalid level. When the value of the core counter is compared with... If the capture register 2 is the same, force OC2REF to go low; 011: Toggle. When the core counter is compared with the value of compare capture register 2. When they are the same, toggle the level of OC2REF; 100: Force to invalid level. Force OC2REF to low; 101: Force to active level. Force OC2REF to high; 110: PWM Mode 1: When counting up, once the core counter... Channel 2 is active when the value is less than the value in the compare-capture register. Otherwise, it is an invalid level; during down-counting, once the core counter... Channel 2 is at an invalid level when the value is greater than that of the comparison capture register. (OC2REF=0), otherwise it is an active level (OC2REF=1); 111: PWM Mode 2: When counting up, once the core counter... Channel 2 is at an invalid level when the value is less than the value in the compare capture register. Otherwise, it is an active level; during down-counting, once the core counter... Channel 2 is active when the value is greater than the value in the compare-capture register. (OC2REF=1), otherwise it is an invalid level (OC2REF=0); Note: Once the level is set and LOCK 3 can be modified), this bit will not... In PWM mode. 1 or PWM 2 model In this case, only when the comparison result changes or the output comparison mode is switched from frozen mode will the comparison be allowed to proceed. The OC2REF level only changes when switching modes.	000b
11	OC2PE	RW	Compare capture register 2 preload enable bit: 1: Enable the preload function of Compare-Capture Register 2 for read and write operations. Operate only on the preload register, compare the preload of capture register 2. The value is loaded into the current shadow register when the update event arrives. middle; 0: Disables the preload function of compare capture register 2, allowing writing at any time.	0

			Enter the comparison capture register 2, and the newly written value takes effect immediately. use. Note: Once the level is set... LOCK And 3 If CC2S=00, this bit cannot be modified; it can only be used in single-pulse mode (PWM Pattern, otherwise its movement register. The answer is uncertain.	
10	OC2FE	RW	Compare-capture channel 2 fast enable bit, this bit is used to speed up the comparison capture. Obtain the channel output in response to the triggered input event. 1: The effective edge of the input to the flip-flop acts as if a time difference has occurred. A better match. Therefore, OC is set to the comparison level and compared with the comparison result. Irrelevant. The valid edge of the sampling trigger and the comparison capture channel 2 output. The delay between clock cycles is reduced to 3 clock cycles; 0: Compare the capture value with the counter and the value of the comparison capture register 1.	0
[9:8] CC2S[1:0]		RW	Channel 2 operates normally, even if the trigger is on. When the trigger... When the input has a valid edge, the compare capture channel 2 is activated and the output is captured. The minimum delay is 5 clock cycles. Note: OC2FE only applies when the channel is configured in PWM1 and PWM2 are working. Compare capture channel 2 input selection field: 00: Compare capture channel 2 is configured as output; 01: Compare capture channel 2 is configured as an input, IC2 is mapped to TI2. superior; 10: Compare capture channel 2 is configured as an input, IC2 is mapped to TI1. superior; 11: Compare capture channel 2 is configured as an input, IC2 is mapped to TRC. Above. This mode only works when the internal trigger input is selected (by...). TS bit selection). Note: Comparison capture channels only occurs when the channel is off (CC2E is zero). Only number 2 is writable.	00b
7	OC1CE	RW compares and captures channel 1 to clear the enable bit.		0
[6:4] OC1M[2:0]		RW compares the capture channel 1 mode setting field.		0
3	OC1PE	RW compares the capture register 1 preload enable bit.		0
2	OC1FE	RW compares and captures the channel 1 fast enable bit.		0
[1:0] CC1S[1:0]		RW compares the capture channel 1 input selection field.		0

Capture mode (pin direction is input):

Bitname		access	The	Reset value
[15:12] IC2F[3:0]		RW	description of the input capture filter 2 configuration field shows how these bits set the TI1 input. The sampling frequency and digital filter length. A digital filter consists of a... The event counter consists of events that record N events and then generate a new one. The output jump. 0000: No filter, sampled in fDTS; 1000: Sampling frequency Fsampling = Fdts/8, N = 6; 0001: Sampling frequency Fsampling=Fck_int, N=2; 1001: Sampling frequency Fsampling = Fdts/8, N = 8;	0000b

			0010: Sampling frequency Fsampling=Fck_int, N=4; 1010: Sampling frequency Fsampling = Fdts/16, N = 5; 0011: Sampling frequency Fsampling=f=Fck_int, N=8; 1011: Sampling frequency Fsampling = Fdts/16, N = 6; 0100: Sampling frequency Fsampling = Fdts/2, N = 6; 1100: Sampling frequency Fsampling = Fdts/16, N = 8; 0101: Sampling frequency Fsampling = Fdts/2, N = 8; 1101: Sampling frequency Fsampling = Fdts/32, N = 5; 0110: Sampling frequency Fsampling = Fdts/4, N = 6; 1110: Sampling frequency Fsampling = Fdts/32, N = 6; 0111: Sampling frequency Fsampling = Fdts/4, N = 8; 1111: Sampling frequency Fsampling=Fdts/32, N=8.	
[11:10] IC2PSC[1:0]		RW	Compare the prescaler configuration field of capture channel 2; these 2 bits define the comparison... Capture the prescaler coefficients for channel 2. Once CC1E=0, the prescaler... Device reset. 00: No prescaler, captures every edge detected on the input port. Each step triggers a capture; 01: A capture is triggered every 2 events; 10: A capture is triggered every 4 events; 11: A capture is triggered once every 8 events.	00b
[9:8] CC2S[1:0]		RW	Compare the capture channel 2 input selection field; these two bits define the channel's... Input/output, and selection of input pins. 00: Compare capture channel 2. Channel is configured as output. 01: Compare and capture channel 2. Channel is configured as input, IC1 mapping. At TI1; 10: Compare and capture channel 2. Channel is configured as input, IC1 mapping. At TI2; 11: Compare and capture channel 2. Channel is configured as input, IC1 mapping. On TRC. This mode only works when the internal trigger input is selected. Time (selected by the TS bit). Note: CC2S is only writable when the channel is closed (CC2E is 0).	00b
[7:4] IC1F[3:0]		RW Input Capture Filter 1 Configuration Domain.		0
[3:2] IC1PSC[1:0]		RW compares the capture channel 1 prescaler configuration field.		0
[1:0] CC1S[1:0]		RW compares the capture channel 1 input selection field.		0

14.4.8 Compare/Capture Control Register 2 (TIMx_CHCTLR2) (x=1/8) Offset Address: 0x1C.

The channel can be

used for input (capture mode) or output (compare mode), and the channel direction is defined by the corresponding CCxS bits. This register...

The functions of other bits differ between input and output modes. OCxx describes the channel's function in output mode, and ICxx describes the channel's function in output mode.

Functionality 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):

Bitname		access	describe	Reset value
15	OC4CE	RW	compares and captures channel 4, clearing the enable bit.	0
[14:12]	OC4M[2:0]	RW	compares the capture channel 4 mode setting field.	0
11	OC4PE	RW	compares the capture register 4 preload enable bit.	0
10	OC4FE	RW	compares the capture channel 4 fast enable bit.	0
[9:8]	CC4S[1:0]	RW	compares the capture channel 4 input selection field.	0
7	OC3CE	RW	compares and captures channel 3 to clear the enable bit.	0
[6:4]	OC3M[2:0]	RW	compares the capture channel 3 mode setting field.	0
3	OC3PE	RW	compares capture register 3 preload enable bit.	0
2	OC3FE	RW	compares the capture channel 3 fast enable bit.	0
[1:0]	CC3S[1:0]	RW	compares the capture channel 3 input selection field.	0

Capture mode (pin direction is input):

Bitname		Access	description	Reset value
[15:12]	IC4F[3:0]	RW	Input Capture Filter 4 Configuration field.	0
[11:10]	IC4PSC[1:0]	RW	compares and captures the prescaler configuration field of channel 4.	0
[9:8]	CC4S[1:0]	RW	compares the capture channel 4 input selection field.	0
[7:4]	IC3F[3:0]	RW	Input Capture Filter 3 Configuration Domain.	0
[3:2]	IC3PSC[1:0]	RW	compares the capture channel 3 prescaler configuration field.	0
[1:0]	CC3S[1:0]	RW	compares the capture channel 3 input selection field.	0

14.4.9 Compare/Capture Enable Register (TIMx_CCER) (x=1/8) Offset Address: 0x20 Name

Bit		access	describe	Reset value
[15:14]	Reserved	RO	is reserved.	0
13	CC4P	RW	compares the capture channel 4 output polarity setting bit.	0
12	CC4E	RW	compares and captures the output enable bit of channel 4.	0
11	CC3NP	RW	compares the complementary output polarity setting bit of capture channel 3.	0
10	CC3NE	RW	Compare capture channel 3 complementary output enable bit.	0
9	CC3P	RW	compares the capture channel 3 output polarity setting bit.	0
8	CC3E	RW	compares and captures the output enable bit of channel 3.	0
7	CC2NP	RW	compares the complementary output polarity setting bit of capture channel 2.	0
6	CC2NE	RW	Compare capture channel 2 complementary output enable bit.	0
5	CC2P	RW	compares the capture channel 2 output polarity setting bit.	0
4	CC2E	RW	compares and captures the channel 2 output enable bit.	0
3	CC1NP	RW	compares the complementary output polarity setting bit of capture channel 1.	0
2	CC1NE	RW	Compare-capture channel 1 complementary output	0
1	CC1P	RW	enable bit. Compare-capture channel 1 output polarity setting bit. Channel CC1 is configured as an output: 1: OC1 is active low; 0: OC1 is active high. CC1 channel configured as input: This bit selects whether IC1 or its inverted signal is used as the trigger or catch.	0

			Signal received. 1: Inverting: Capture occurs at the falling edge of IC1; when used as an external contact. When the transmitter is activated, IC1 is inverted; 0: Non-inverting: Capture occurs on the rising edge of IC1; when used as an external... When the trigger is activated, IC1 is not inverted. Note: Once the lock (LOCK LOCK in the TIMx_BDTR register) is If a bit is set to 3 or 2, then that bit cannot be modified.	
0	CC1E	RW	Compare the capture channel 1 output enable bit. Channel CC1 is configured as an output: 1: Enable. The OC1 signal is output to the corresponding output pin, and its output... The voltage level depends on MOE, OSSI, OSSR, OIS1, OIS1N, and CC1NE. The value of the bit; 0: Off. OC1 disables output; therefore, the output level of OC1 depends on... At the MOE, OSSI, OSSR, OIS1, OIS1N and CC1NE bits value. CC1 channel configured as input: This bit determines whether the counter value can be captured into TIMx_CH1CVR. register. 1: Capture Enable; 0: Capture prohibited.	0

14.4.10 Advanced Timer Counter (TIMx_CNT) (x=1/8) Offset Address: 0x24 Bit Name [15:0] CNT[15:0]

		Access the real-time value of the counter that describes		Reset value
		the RW timer.		0

14.4.11 Counting Clock Prescaler (TIMx_PSC) (x=1/8)

Offset address: 0x28 Name

Bit		access	describe	Reset value
[15:0] PSC[15:0]		RW	The division factor of the timer's prescaler; the clock frequency of the counter. It equals the input frequency of the frequency divider / (PSC+1).	0

14.4.12 Automatic Reload Value Register (TIMx_ATRLR) (x=1/8) Offset Address: 0x2C

Bit	name	access	describe	Reset value
[15:0] ARR[15:0]		RW	The value of this field will be loaded into a counter, indicating when the ATRLR will act and update. See section 14.2.3; the counter stops when ATRLR is empty.	FFFFh

14.4.13 Repeat Count Register (TIMx_RPTCR) (x=1/8)

Offset address: 0x30

Bit	name	access	describe	Reset value
[15:8] Reserved		RO is reserved.		0
[7:0] REP[7:0]		RW is the value of the repeat counter.		0

14.4.14 Compare/Capture Register 1 (TIMx_CH1CVR) (x=1/8)

Offset Address: 0x34

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LEVEL1

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

Bitname	access	describe	Reset value
[31:17] Reserved	RO is reserved.		0
16 LEVEL1	RO is the level indicator bit corresponding to the captured value.		0
[15:0] CH1CVR[15:0]	RW compares the value of capture register channel 1.		0

14.4.15 Compare/Capture Register 2 (TIMx_CH2CVR) (x=1/8)

Offset Address: 0x38

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LEVEL2

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

Bitname	access	describe	Reset value
[31:17] Reserved	RO is reserved.		0
16 LEVEL2	RO is the level indicator bit corresponding to the captured value.		0
[15:0] CH2CVR[15:0]	RW compares the value of capture register channel 2.		0

14.4.16 Compare/Capture Register 3 (TIMx_CH3CVR) (x=1/8)

Offset Address: 0x3C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LEVEL3

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

Bit	name	access	describe	Reset value
[31:17] Reserved		RO is reserved.		0
16	LEVEL3	RO is the level indicator bit corresponding to the captured value.		0
[15:0] CH3CVR[15:0]		RW compares the value of capture register channel 3.		0

14.4.17 Compare/Capture Register 4 (TIMx_CH4CVR) (x=1/8)

Offset Address: 0x40

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

Reserved	LEVEL4

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

Bit	name	access	describe	Reset value
[31:17] Reserved		RO is reserved.		0
16	LEVEL4	RO is the level indicator bit corresponding to the captured value.		0
[15:0] CH4CVR[15:0]		RW compares the value of capture register channel 4.		0

14.4.18 Brake and Dead Zone Register (TIMx_BDTR) (x=1/8)

Offset address: 0x44

Bit	Name	access	This	Reset value
15	MOE	RW	describes the main output enable bit. It will be asynchronously cleared once the brake signal is active. 1: Allow OCx and OCxN to be set as outputs; 0: Disable the output of OCx and OCxN or force them to idle.	0
14	AOE	RW	Automatic output enable: 1: MOE can be set by software or in the next update event. Set; 0: MOE can only be set by software.	0
13	BKP	RW	Brake input polarity setting position: 1: Brake input is active high; 0: Brake input is active low. Note: When a level is set LOCK 1 Afterwards, this bit cannot be modified. Writing to this bit requires a... HB	0
12	BKE	RW	Brake function enable position: 1: Activate brake input; 0: Brake input is prohibited. Note: When a level is set LOCK 1 Afterwards, this bit cannot be modified. Writing to this bit requires a... HB	0
11	OSSR	RW	1: When the timer is not working, once CCxE=1 or CCxNE=1, First, enable OC/OCN and output an invalid level, then set OCx, OCxN enable output signal = 1; 0: When the timer is not working, OC/OCN output is disabled. Note: This bit cannot be modified once a level is set. LOCK 1	0
10	OSSI	RW	1: When the timer is not working, once CCxE=1 or CCxNE=1, OC/OCN first outputs its idle level, then OCx and OCxN are enabled. Output signal = 1; 0: When the timer is not working, OC/OCN output is disabled. Note: When a level is set LOCK 1 After that, this position cannot be modified.	0
[9:8] LOCK[1:0]		RW	Lock function settings field: 00: Disable the lock function; 01: Lock level 1, DTG, BKE, BKP, AOE, and OISx cannot be written. and OISxN bits;	00b

			10: Lock level 2, cannot write to any bits in lock level 1. It is also not possible to write the CC polarity bit, OSSR, or OSSI bits; 11: Locking level 3, cannot write to any bits in locking level 2. It also cannot be written to the CC control bit. Note: After a system reset, the bit can only be written once and cannot be written again. modifies until reset.	
[7:0] DTG[7:0]		RW	Dead-time setting bits, these bits define the dead-time duration between complementary outputs. Continued time. Assume DT represents its duration: DTG[7:5]=0xx=>DT=DTG[7:0]*Tdtg, Tdtg =TDTS; DTG[7:5]=10x=>DT=(64+DTG[5:0])*Tdtg, Tdtg= 2*TDTS; DTG[7:5]=110=>DT=(32+DTG[4:0])*Tdtg, Tdtg =8 ×TDTS; DTG[7:5]=111=>DT=(32+DTG[4:0])*Tdtg, Tdtg =16 *TDTS. Note: Dān level (in the register) If the TIMx_BDTR LOCK[1:0] bits are set to 01b, 10b, or 11b, then it cannot be modified. These positions.	0

14.4.19 DMA Control Register (TIMx_DMACFGR) (x=1/8) Offset Address: 0x48

Name

Bit		access	describe	Reset value
[15:13] Reserved		RO is reserved.		0
[12:8] DBL[4:0]		RW	The length of a continuous DMA transfer; the actual value is the value of this field + 1. When reading or writing to the TIMx_DMAADR register, the timer then... (Perform a single continuous transmission), that is: define the number of transmissions, transmission It can be a half-word (two-byte) or a byte: 00000: 1 transmission; 00001: 2 transmissions; 00010: 3 transmissions; ... ... 10001: 18 transmissions. Suppose,we perform the following transmission: DBL=7 , DBA=TIM2_CTLR1. If DBL=7, DBA=TIM2_CTLR1 indicates the location of the data to be transmitted. The address for transmission is given by the following formula: (TIMx_CTLR1) (address) + DBA + (DMA index), where DMA index = DBL, The value of (TIMx_CTLR1 address) + DBA plus 7 gives the result. The address where data will be written or read, so that data transmission will It occurred in 7 days starting from address (TIMx_CR1 address) + DBA. Registers. Depending on the DMA data length setting, the following may occur: Condition: 1. If the data is set to half-word (16 bits), then the data will be transmitted.	0

			The output is sent to all 7 registers. 2. If the data is set to bytes, the data will still be transmitted to all 7 bytes. Registers: The first register contains the first MSB byte, the second... The two registers contain the first LSB byte, and so on. Therefore For timers, the user must specify the data width to be transferred by DMA.	
[7:5] Reserved		RO is reserved.		0
[4:0] DBA[4:0]		RW	These bits define the DMA's access from control register 1 in continuous mode. Offset at the address. 00000: TIMx_CTLR1; 00001: TIMx_CTLR2; 00010: TIMx_SMCFGR; ...	0

14.4.20 DMA Address Register in Continuous Mode (TIMx_DMAADR) (x=1/8)

Offset address: 0x4C

Bit	name	access	describe	Reset value
[15:0] DMAB[15:0]		RW	In continuous mode, the address of the DMA. Reading or writing to the TIMx_DMAADR register will result in access to the following addresses Access operations to the register: TIMx_CTLR1 address + DBA + DMA The index, where "TIMx_CTLR1 address" is the control register 1 The address where (TIMx_CTLR1) is located; "DBA" is the base address defined in the TIMx_DMACFGR register; "DMA index" is determined by... The offset for DMA auto-control depends on the TIMx_DMACFGR register. The DBL defined in the register.	0

14.4.21 Dual-edge capture register (TIMx_AUX) (x=1/8)

Offset address: 0x50

Bit	Name	access	When	Reset value
[15:8] DT_VLU2		RW	describing the implementation of dead-time asymmetry functionality, the time should be set to be less than... The dead time duration of DTG needs to be determined in conjunction with DTP_MODE and Use the DTN_MODE bit. Note: The dead time duration is set by multiplying the set value by the middle value. settings Tdtg, which needs to be greater than or less than the set dead zone .	0
7	DTN_MODE	RW	The dead time duration set by enabling DT_VLU2 occurs during The falling edge of OCXREF. 1: The dead time duration set by DT_VLU2 occurs in OCXREF. The falling edge; 0: Functionality disabled, OCXREF falling edge dead time duration Set the time for the existing DTG.	0
6	DTP_MODE	RW	The dead time duration set by enabling DT_VLU2 occurs during The rising edge of OCXREF. 1: The dead time duration set by DT_VLU2 occurs in OCXREF. The rising edge;	0

			0: Functionality disabled, OCXREF rising edge dead time duration Set the time for the existing DTG.	
[5:3] BK_SEL		RW	Brake source selection position: 000: BKIN; 001: BKIN2; 01x: BKIN or BKIN2; 100: DFSDM_BK[0]; 101: DFSDM_BK[1]; Other: Reserved.	0
2	CAP_ED_CH4	RW	Double-edge capture enable for channel 4: 1: Enable double-edge capture of channel 4; 0: Disable double-edge capture enable for channel 4.	0
1	CAP_ED_CH3	RW	Double-edge capture enable for channel 3: 1: Enable double-edge capture of channel 3; 0: Disable double-edge capture enable for channel 3.	0
0	CAP_ED_CH2	RW	Double-edge capture enable for channel 2: 1: Enable double-edge capture of channel 2; 0: Disable double-edge capture enable for channel 2.	0

Chapter 15 General Purpose Timer (GPTM)

The general-purpose timer module includes four 16-bit auto-reload timers (TIM2, TIM3, TIM4, and TIM5) and four 32-bit timers.

Automatically reloadable timers (TIM9, TIM10, TIM11, and TIM12) are used to measure pulse width or generate pulses of a specific frequency.

PWM waves, etc. These can be used in fields such as automation control and power supplies.

15.1 Main Features

The main features of a 16-bit general-purpose timer include:

ÿ 16-bit auto-reload counter, supporting increment, decrement, and increment/decrement modes.

ÿ 16-bit prescaler, with a dynamically adjustable division factor from 1 to 65536.

Supports four independent compare-capture channels .

Each compare-capture channel supports multiple operating modes, such as: input capture, output compare, PWM generation, and single-pulse output.

Supports external signal control of timers

• Supports DMA use in multiple modes. • Supports

incremental encoding, cascading and synchronization between timers.

The main features of a 32-bit general-purpose timer include:

ÿ 32-bit auto-reload counter, supporting increment, decrement, and increment/decrement modes.

ÿ 16-bit prescaler, with a dynamically adjustable division factor from 1 to 65536.

Supports four independent compare-capture channels .

Each compare-capture channel supports multiple operating modes, such as: input capture, output compare, PWM generation, and single-pulse output.

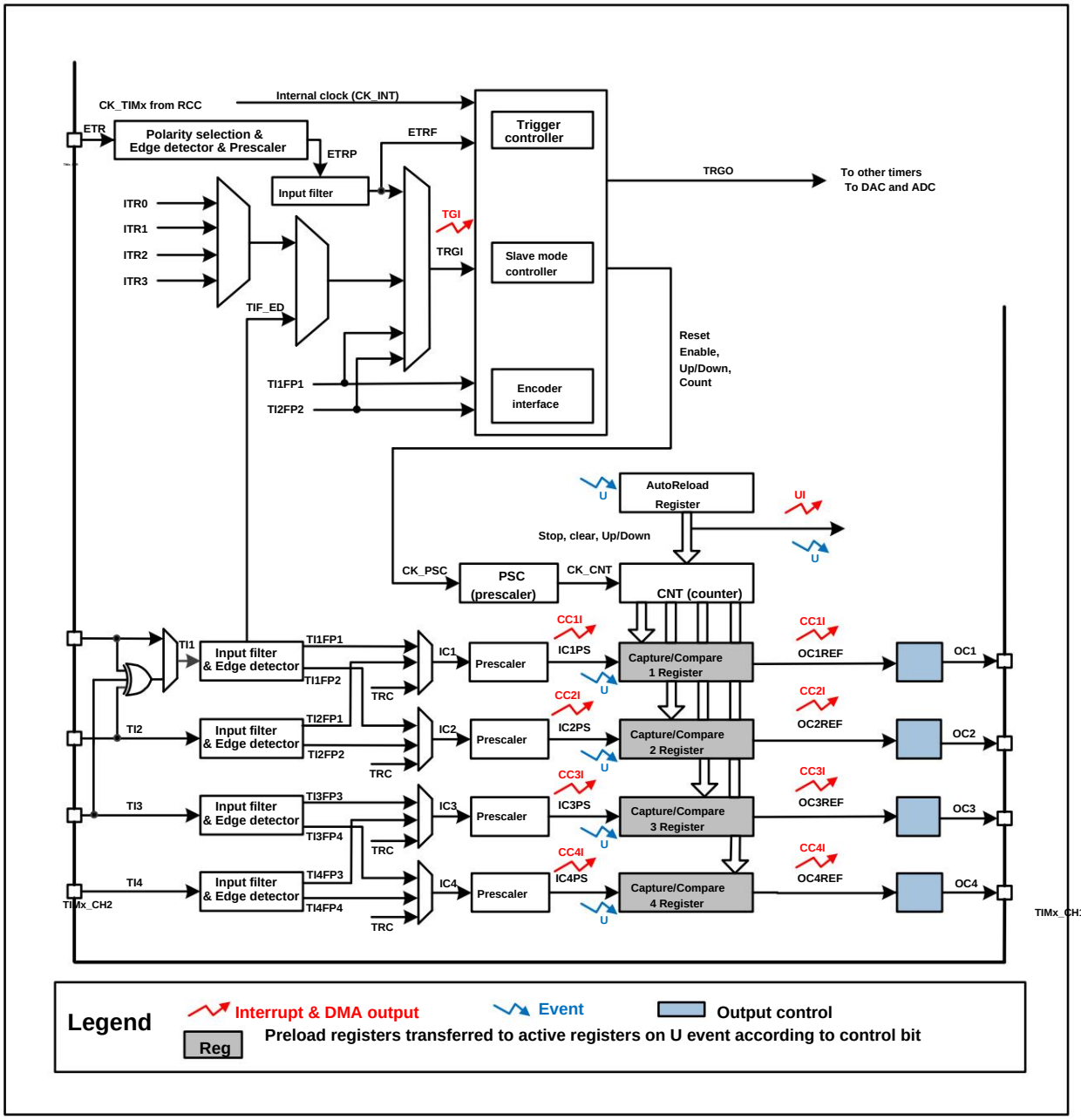
Supports external signal control of timers

• Supports DMA use in multiple modes. • Supports

incremental encoding, cascading and synchronization between timers.

15.2 Principles and Structure

Figure 15-1 Block diagram of a general-purpose timer



15.2.1 Overview As

shown in Figure 15-1, the structure of a general-purpose timer can be roughly divided into three parts: the input clock section, the core counter section, and... Compare the captured channel portions.

The clock for the general-purpose timer can be derived from the HB bus clock (CK_INT) or from an external clock input pin (TIMx_ETR). It can come from other timers with clock output function (ITRx), or from the input of the compare capture channel (TIMx_CHx). These input clock signals, after undergoing various filtering and frequency division operations, become the CK_PSC clock, which is then output to the core counter section. In addition, these complex clock sources can also be used as TRGO outputs to other peripherals such as timers, ADCs, and DACs.

At the core of a general-purpose timer is a 16-bit or 32-bit counter (CNT). After being divided by a prescaler (PSC), CK_PSC... It becomes CK_CNT and then ultimately loses to CNT. CNT supports incrementing, decrementing, and increment/decrementing modes, and has an automatic reload function. The value register (ATRLR) reloads the initial value for CNT at the end of each counting cycle.

The general-purpose timer has four compare-capture channels, each of which can receive pulses from its dedicated pin.

The waveform is output to the pin, meaning the compare capture channel supports both input and output modes. Each channel's input to the compare capture register supports filtering, frequency division, edge detection, and other operations, as well as inter-channel triggering. It also provides 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.

15.2.2 Difference between General Purpose Timers and Advanced Timers

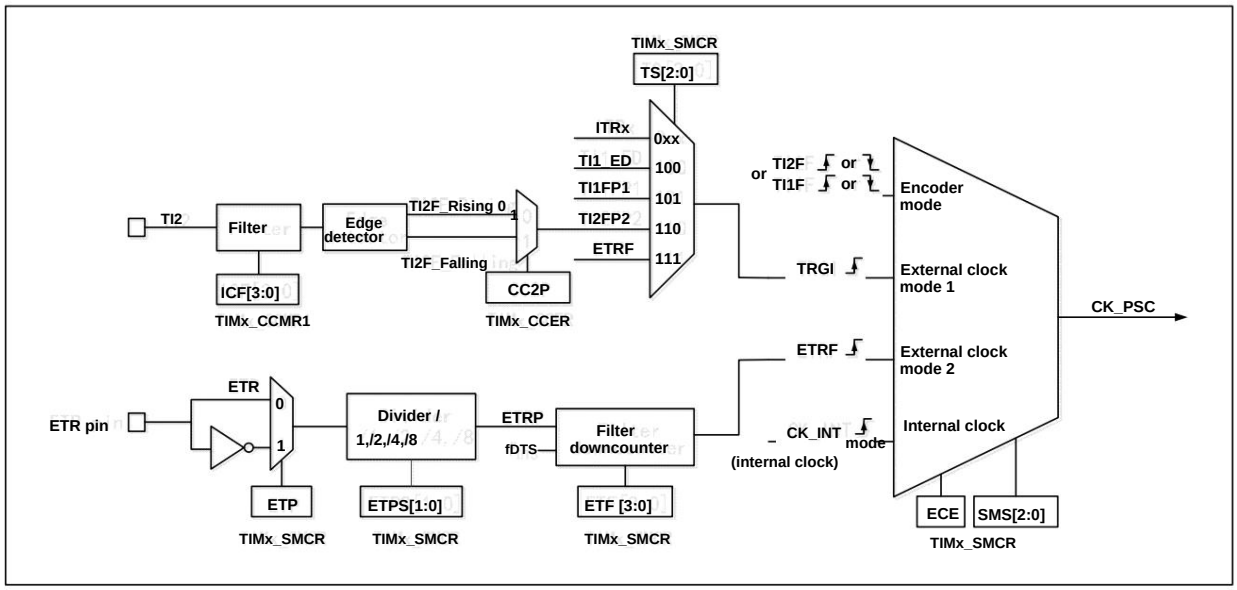
Compared to advanced timers, general-purpose timers lack the following features:

1) The general-purpose timer lacks a repeat counter register to count the count cycles of the core counter. 2) The general-purpose timer's compare-capture channel lacks dead-time generation and complementary output. 3) The general-purpose timer lacks a braking signal mechanism.

15.2.3 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 shown here.

Figure 15-2 Source block diagram of general-purpose timer CK_PSC



The optional input clocks can be categorized into four types: 1) External clock pin (ETR) input route: ETRÿETRPÿETRF; 2) Internal HB clock input route: CK_INT; 3) Route from the compare capture channel pin (TIMx_CHx): TIMx_CHxÿTIxÿTIxFPx, this route is also used for encoder modules.

Mode;

4) Inputs from other internal timers: ITRx. The actual operation can

be categorized into three types by determining the input pulse selection of the SMS from which CK_PSC originates:

1) Select internal clock source (CK_INT); 2) External clock source mode 1; 3) External clock source mode 2; 4) Encoder mode. All four clock source options

mentioned above can be selected through these four operations.

15.2.3.1 Internal Clock Source (CK_INT)

If the general-purpose timer is started while the SMS field is kept at 000b, then the internal clock source (CK_INT) is selected as the clock. In this case, CK_INT is CK_PSC.

15.2.3.2 External Clock Source Mode 1

Setting the SMS field to 111b enables External Clock Source Mode 1. When External Clock Source 1 is enabled, TRGI is selected.

For the source of CK_PSC, it's worth noting that users also need to select the TRGI source by configuring the TS field. The TS field can be selected from the following:

This pulse serves as a clock source:

1) Internal trigger (ITRx, x is 0, 1, 2, 3);

2) Compare the signal (TI1F_ED) after passing through the edge detector in capture channel 1;

3) Compare the signals TI1FP1 and TI2FP2 of the capture channel;

4) The signal ETRF from the external clock pin input.

15.2.3.3 External Clock Source Mode 2 uses External

Trigger Mode 2 to count on each rising or falling edge of the external clock pin input. Setting the ECE bit will...

External clock source mode 2 is used. When using external clock source mode 2, ETRF is selected as CK_PSC. The ETR pin is optionally inverted.

After passing through an ETP (Electronic Frequency Converter), an ETP (Electronic Frequency Divider), and an ETP (Electronic Frequency Divider), it becomes an ETP (Electronic Frequency Divider), and after passing through an ETF (Electronic Frequency Filter), it becomes an ETP (Electronic Frequency Filter).

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

15.2.3.4 Encoder Mode: Setting SMS to

001b, 010b, or 011b will enable encoder mode. Enabling encoder mode can be selected on TI1FP1 and TI2FP2.

This mode outputs a signal at a specific voltage level using another transition edge as the signal. This mode is used when an external encoder is connected.

For specific functions, please refer to section 15.3.7.

15.2.4 Counters and Peripherals

The CK_PSC input is fed to the prescaler (PSC) for frequency division. The PSC is 16-bit, and the actual division factor is equivalent to R16_TIMx_PSC.

The value is incremented by 1. CK_PSC becomes CK_CNT after passing through PSC. Changing the value of R16_TIM1_PSC does not take effect immediately, but will occur during the update event.

The update is then sent to the PSC. Update events include clearing and resetting the UG bit. There is also an Automatic Reload Value Register (ATRLR) that can be used to set the count.

The counter threshold is set in the Automatic Reload Value Register (ATRLR). When the CNT count value reaches the counter threshold set in the ATRLR, the CNT will be cleared to zero.

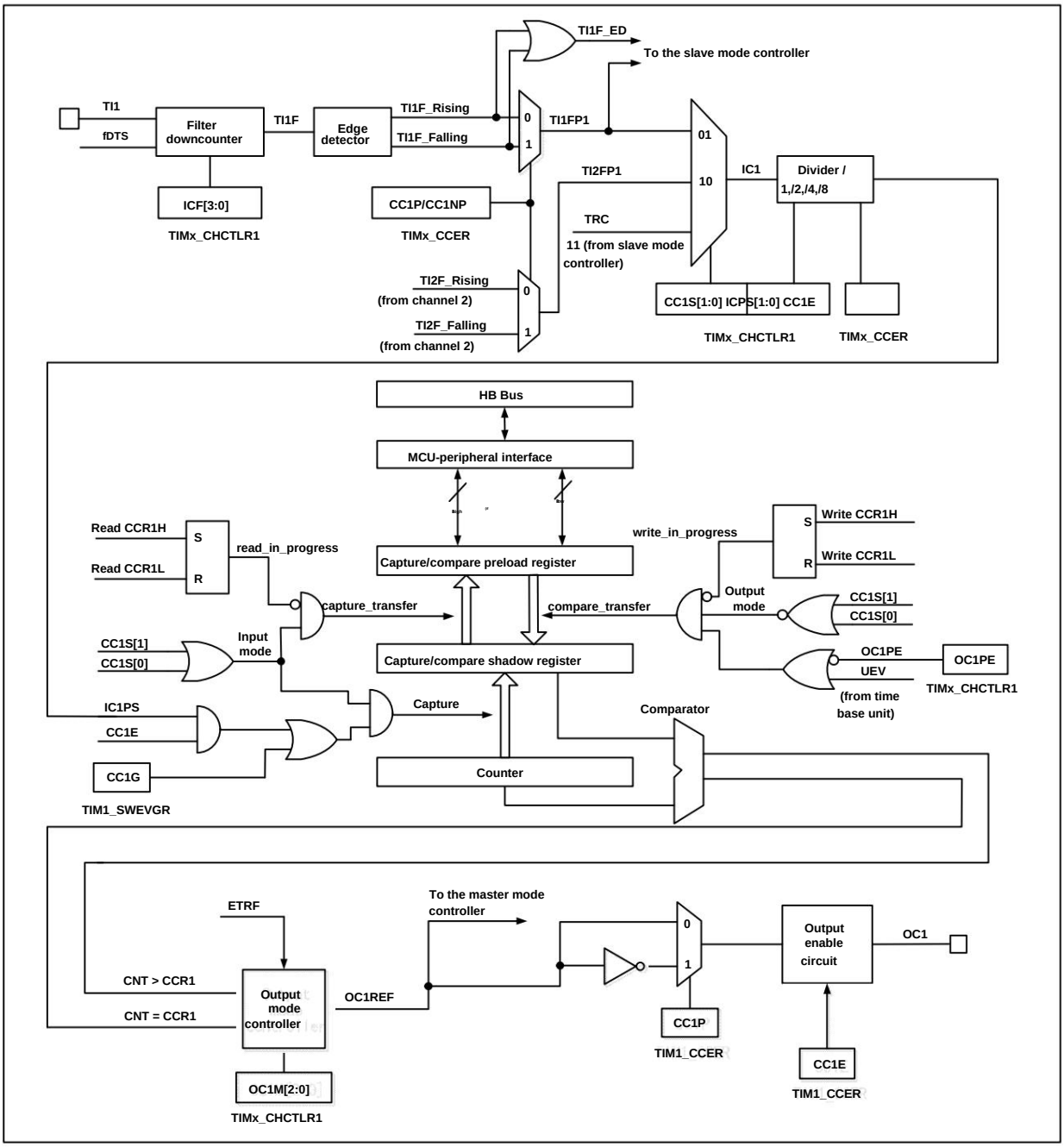
15.2.5 Compare capture channels

The compare-capture channel is the core of the timer's ability to implement complex functions. Its core is the compare-capture register, supplemented by external input sections.

The system consists of digital filtering, frequency division, channel multiplexing, a comparator in the output section, and output control. The block diagram of the comparison capture channel is shown in Figure 15.

As shown in Figure 3.

Figure 15-3 Block diagram of the comparison capture channel



After the signal is input from the channel x pin, it can be selected as TIx (the source of TI1 can be more than just CH1, see the timer block diagram 15-1). Taking TI1 as an example: TI1 is filtered (ICF[3:0]) to generate TI1F, and then divided into TI1F_Rising and TI1F_Falling by the edge detector. These two signals are selected (CC1P) to generate TI1FP1. TI1FP1 and TI2FP1 from channel 2 are sent to CC1S to be selected as IC1. After being divided by ICPS, they are sent to the compare capture register. The compare capture register consists of a preload register and a shadow register. The read and write

process only operates 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 with the core counter (CNT).

15.3 Functions and Implementation

The implementation of complex functions in general-purpose timers involves operating the timer's compare-capture channel, clock input circuit, counter, and peripheral components. The timer's clock input can come from multiple clock sources, including the input to the compare-capture channel. For the compare-capture channel...

The operation of acquiring the register channel and clock source selection directly determines its function. The compare-capture channel is bidirectional and can operate in both input and output modes.

15.3.1 Input Capture Mode

Input capture mode is one of the basic functions of a timer. The principle of input capture mode is that when a defined edge is detected on the ICxPS signal, a capture event is generated, 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. If interrupts or DMA are enabled, corresponding interrupts or DMA will also be generated. If CCxIF is already set when a capture event occurs, then the CCxOF bit will be set. CCxIF can be cleared by software or by hardware by reading the compare-capture register. CCxOF is cleared by software. The steps for using input capture mode are illustrated below using channel 1 as an example:

1) Configure the CC1S domain and select the source of the IC1 signal. For example, set it to 10b and select TI1FP1 as the source of IC1. (Note: This option cannot be used.)

By default, the CC1S domain uses the compare capture module as the output channel.

2) Configure the IC1F field; these bits set the sampling frequency of the TI1 input and the length of the digital filter. The digital filter is controlled by an event counter.

The system consists of a counter that records N events before generating an output transition. 3)

Configure the CC1P bit to set the polarity of TI1FP1. For example, keeping the CC1P bit low allows for a rising edge transition. 4) Configure the

IC1PS field to set the IC1 signal as the division factor between IC1PS. For example, keeping IC1PS at 00b allows for no division. 5) Configure the CC1E bit to allow capturing the core counter (CNT) value into the compare capture register. 6) Configure the CC1IE and CC1DE bits depending on whether an

interrupt or DMA is needed. When a captured pulse is input to TI1, the core counter (CNT) value is

recorded in the compare capture register, and CC1IF is set. If CC1IF was previously set, CC10F will also be set. If CC1IE is set, an interrupt will be generated; if CC1DE is set, a DMA request will be generated. An input capture event can be generated by software by writing to the event generation register (R16_TIMx_SWEVGR).

15.3.2 Comparison Output Mode

Compare-to-output mode is one of the fundamental functions of a timer. The principle of compare-to-output mode is to output a specific change or waveform when the value of the core counter (CNT) matches the value of the compare-to-capture register. The OCxM field (in R16_TIMx_CHCTLRx) and the CCxP bit (in R16_TIMx_CCER) determine whether

the output is a defined high/low level or a level toggle. A compare-to-capture event will also set...

The CCxIF bit, if the CCxIE bit is preset, will generate an interrupt; if the CCxDE bit is preset, it will generate a DMA request. The steps to configure for compare-output mode are as follows:

1) Configure the clock source and auto-reload value for the

core counter (CNT); 2) Set the count value to be compared to the compare-capture register

(R16_TIMx_CHxCVR); 3) If an interrupt is needed, set the CCxIE bit; 4) Keep OCxPE at 0 and disable the preload register of the

compare-capture register; 5) Set the output mode by setting the

OCxM field and the CCxP bit; 6) Enable output by setting the CCxE bit; 7) Set the CEN bit to start the timer;

15.3.3 Forced Output Mode

The output mode of the timer's compare-capture channel can be forced to a predetermined level by software, without depending on the compare-capture register. Comparison of shadow registers and core counters.

Specifically, setting OCxM to 100b forces OCxREF low; or setting OCxM to 101b forces OCxREF high. It's important to note that forcing OCxM to 100b or 101b alters the comparison process between

the internal main counter and the compare-capture register.

The process is still ongoing; the corresponding flags are still being set, and interrupts and DMA requests are still being generated.

15.3.4 PWM Input Mode

PWM input mode is used to measure the duty cycle and frequency of PWM signals and is a special case of input capture mode. Except for the following differences, the operation is the same as input capture mode: PWM uses two compare-capture channels with opposite input polarities, one signal is set as the trigger input, and SMS is set to reset mode.

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

1) Set TI1 (TI1FP1) as the input to the IC1 signal. Set CC1S to 01b; 2) Set TI1FP1 to rising edge valid. Keep CC1P at 0; 3) Set TI1 (TI1FP2) as the input to the IC2 signal. Set CC2S to 10b; 4) Set TI1FP2 to falling edge valid. Set CC2P to 1; 5) Select TI1FP1 as the clock source. Set TS to 101b; 6) Set SMS to reset mode, i.e., 100b; 7) Enable input capture. Set CC1E and CC2E.

15.3.5 PWM Output Mode PWM

output mode is one of the basic functions of a timer. The most common method for PWM output mode is to determine the PWM frequency using the reload value and the duty cycle using the capture-compare register. Set the OCxM field to 110b or 111b, 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 reloading of the preload register. The value of the preload register is only 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 capture-compare register are constantly compared. Depending on the CMS bit, the timer can output edge-aligned or center-aligned PWM signals. ÿ Edge Alignment

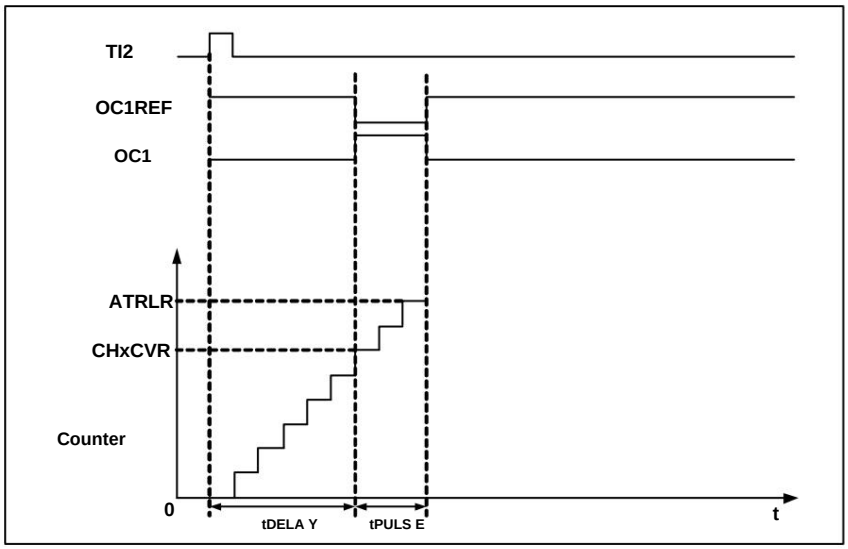
When using edge alignment, the core counter increments or decrements. In PWM mode 1, OCxREF rises high when the core counter value is greater than the compare-capture register value; and falls low when the core counter value is less than the compare-capture register value (e.g., when the core counter increments to the value of R16_TIMx_ATRLR and then returns to all zeros). ÿ Center alignment

When using center-aligned mode, the core counter operates in an alternating up and down counting mode, with OCxREF rising and falling when the core counter and compare capture register values match. However, the timing of setting the compare flag differs across the three center-aligned modes. When using center-aligned mode, it is best to generate a software update flag (set the UG bit) before starting the core counter.

15.3.6 Single Pulse Mode

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

Figure 15-4 Event Generation and Impulse Response



As shown in Figure 15-4, a rising edge needs to be detected on the TI2 input pin, and after a delay of Tdelay, a signal needs to be generated on OC1. Generate a positive pulse of length Tpulse:

1) Set TI2 as the trigger. Set the CC2S field to 01b, mapping TI2FP2 to TI2; set the CC2P bit to 0b, and set TI2FP2 to rising.

Edge detection; set the TS field to 110b, and set TI2FP2 as the trigger source; set the SMS field to 110b, and use TI2FP2 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 value of the compare capture register.

15.3.7 Encoder Mode Encoder

mode is a typical application of timers. It can be used to connect the two-phase output of an encoder and control the counting direction of the core counter. Synchronized with the encoder's shaft direction, each pulse output by the encoder increments or decrements the core counter by one. Steps for using the encoder. To: Set the SMS field to 001b (counting only on the TI2 edge), 010b (counting only on the TI1 edge), or 011b (counting on both TI1 and TI2 edges). (Dual-edge counting), connect the encoder to the input terminals of compare capture channels 1 and 2, and set a reload value counter value. This value can... Set it to a larger value. In encoder mode, the timer's internal compare-capture register, prescaler, and repeat count register all function normally.

The following table shows the relationship between the counting direction and the encoder signal.

Table 15-1 Relationship between counting direction and encoder signal in timer encoder mode

Counting effective edge	relative signal level	TI1FP1 signal edge		TI2FP2 signal	
		Rising edge	Falling edge	Rising edge	Falling edge
Counting only on the TI1 edge	High down count	up count	Low up count	down	Not counted
	count				
Counting only on TI2 edges	high	Not counted		Count up	Count down
	Low down count, up count				
Counting on both sides of TI1 and TI2	High down count	Up count	Up count	Down count	
	Low up count	down count	down count	up count	

15.3.8 Timer Synchronization Mode A

timer can output a clock pulse (TRGO) and also receive inputs from other timers (ITRx). Different timers have different ITRx values. The source (TRGO of other timers) is different. The internal trigger connection of the timer is shown in Table 15-2.

Table 15-2 GTPM Internal Trigger Connections

From timer	ITR0 (TS=000)	ITR1 (TS=001)	ITR2 (TS=010)	ITR3 (TS=011)
TIM2	TIM1	TIM8/USB/ETH	TIM3	TIM4
TIM3	TIM1	TIM2	TIM5	TIM4
TIM4	TIM1	TIM2	TIM3	TIM8
TIM5	TIM2	TIM3	TIM4	TIM8
TIM9	TIM1	TIM2	TIM10	TIM11
TIM10	TIM1	TIM8	TIM2	TIM9
TIM11	TIM1	TIM2	TIM9	TIM12
TIM12	TIM1	TIM2	TIM9	TIM11

15.3.9 Bilateral Edge Capture Mode

The pulse measurement can be enabled by enabling dual-edge capture on the corresponding channel through the CAP_ED_CHx bit of the TIMx_AUX register.

For example: to capture the pulse width via channel 2, in the capture function configuration, select the source of the IC2 signal (TIMx_CHCTLR1). The CC2S bit of the register is 11b), enabling the double-edge capture function of channel 2 (the CAP_ED_CH2 bit of the TIMx_AUX register is 1). The dual-edge capture configuration is now complete.

The high and low level width values of the captured pulse can be read through the CH2CVR bit of the TIMx_CH2CVR register. Enabling the CAPLVL bit allows... The level corresponding to the captured value is indicated by bit[16] of the TIMx_CH2CVR register.

Note: This function is only available for TIM2/TIM3/TIM4/TIM5.

15.3.10 Debug Mode When

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

15.4 Register Description

Table 15-3 List of TIM2 Related Registers

name	Offset address description		Reset value
R16_TIM2_CTLR1	0x40000000	TIM2 Control Register 1	0x0000
R16_TIM2_CTLR2	0x40000004	TIM2 Control Register 2	0x0000
R16_TIM2_SMCFGR	0x40000008	TIM2 from 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 Register 1	0x0000
R16_TIM2_CHCTLR2	0x4000001C	TIM2 Compare/Capture Control Register 2	0x0000
R16_TIM2_CCER	0x40000020	TIM2 Compare/Capture Enable Register	0x0000
R16_TIM2_CNT	0x40000024	TIM2 counter	0x0000
R16_TIM2_PSC	0x40000028	TIM2 Counting Clock Prescaler	0x0000
R16_TIM2_ATRLR	0x4000002C	TIM2 Auto Reload Value Register	0xFFFF
R32_TIM2_CH1CVR	0x40000034	TIM2 Compare/Capture Register 1	0x0000
R32_TIM2_CH2CVR	0x40000038	TIM2 Compare/Capture Register 2	0x0000
R32_TIM2_CH3CVR	0x4000003C	TIM2 Compare/Capture Register 3	0x0000
R32_TIM2_CH4CVR	0x40000040	TIM2 Compare/Capture Register 4	0x0000

R16_TIM2_DMACFGR	0x40000048	TIM2 DMA Control Register	0x0000
R16_TIM2_DMAADR	0x4000004C	TIM2 Continuous Mode DMA Address Register 0x0000	
R16_TIM2_AUX	0x40000050	TIM2 Dual Edge Capture Register	0x0000

Table 15-4 List of TIM3 Related Registers

name	Offset address	describe	Reset value
R16_TIM3_CTLR1	0x40000400	TIM3 Control Register 1	0x0000
R16_TIM3_CTLR2	0x40000404	TIM3 Control Register 2	0x0000
R16_TIM3_SMCFGR	0x40000408	TIM3 from Mode Control Register	0x0000
R16_TIM3_DMAINTENR	0x4000040C	TIM3 DMA/Interrupt Enable Register	0x0000
R16_TIM3_INTFR	0x40000410	TIM3 Interrupt Status Register	0x0000
R16_TIM3_SWEVGR	0x40000414	TIM3 Event Generation Register	0x0000
R16_TIM3_CHCTLR1	0x40000418	TIM3 Compare/Capture Control Register 1	0x0000
R16_TIM3_CHCTLR2	0x4000041C	TIM3 Compare/Capture Control Register 2	0x0000
R16_TIM3_CCER	0x40000420	TIM3 Compare/Capture Enable Register	0x0000
R16_TIM3_CNT	0x40000424	TIM3 counter	0x0000
R16_TIM3_PSC	0x40000428	TIM3 Counting Clock Prescaler	0x0000
R16_TIM3_ATRLR	0x4000042C	TIM3 Auto Reload Value Register	0xFFFF
R32_TIM3_CH1CVR	0x40000434	TIM3 Compare/Capture Register 1	0x0000
R32_TIM3_CH2CVR	0x40000438	TIM3 Compare/Capture Register 2	0x0000
R32_TIM3_CH3CVR	0x4000043C	TIM3 Compare/Capture Register 3	0x0000
R32_TIM3_CH4CVR	0x40000440	TIM3 Compare/Capture Register 4	0x0000
R16_TIM3_DMACFGR	0x40000448	TIM3 DMA Control Register	0x0000
R16_TIM3_DMAADR	0x4000044C	TIM3 Continuous Mode DMA Address Register 0x0000	
R16_TIM3_AUX	0x40000450	TIM3 Dual Edge Capture Register	0x0000

Table 15-5 List of TIM4 Related Registers

name	Offset address description		Reset value
R16_TIM4_CTLR1	0x40000800	TIM4 Control Register 1	0x0000
R16_TIM4_CTLR2	0x40000804	TIM4 Control Register 2	0x0000
R16_TIM4_SMCFGR	0x40000808	TIM4 from Mode Control Register	0x0000
R16_TIM4_DMAINTENR	0x4000080C	TIM4 DMA/Interrupt Enable Register	0x0000
R16_TIM4_INTFR	0x40000810	TIM4 Interrupt Status Register	0x0000
R16_TIM4_SWEVGR	0x40000814	TIM4 Event Generation Register	0x0000
R16_TIM4_CHCTLR1	0x40000818	TIM4 Compare/Capture Control Register 1	0x0000
R16_TIM4_CHCTLR2	0x4000081C	TIM4 Compare/Capture Control Register 2	0x0000
R16_TIM4_CCER	0x40000820	TIM4 Compare/Capture Enable Register	0x0000
R16_TIM4_CNT	0x40000824	TIM4 counter	0x0000
R16_TIM4_PSC	0x40000828	TIM4 Counting Clock Prescaler	0x0000
R16_TIM4_ATRLR	0x4000082C	TIM4 Auto Reload Value Register	0xFFFF
R32_TIM4_CH1CVR	0x40000834	TIM4 Compare/Capture Register 1	0x0000
R32_TIM4_CH2CVR	0x40000838	TIM4 Compare/Capture Register 2	0x0000
R32_TIM4_CH3CVR	0x4000083C	TIM4 Compare/Capture Register 3	0x0000
R32_TIM4_CH4CVR	0x40000840	TIM4 Compare/Capture Register 4	0x0000

R16_TIM4_DMACFGR	0x40000848	TIM4 DMA Control Register	0x0000
R16_TIM4_DMAADR	0x4000084C	TIM4 Continuous Mode DMA Address Register 0x0000	
R16_TIM4_AUX	0x40000850	TIM4 Dual-Edge Capture Register	0x0000

Table 15-6 List of TIM5 Related Registers

name	Offset address	describe	Reset value
R16_TIM5_CTLR1	0x40000C00	TIM5 Control Register 1	0x0000
R16_TIM5_CTLR2	0x40000C04	TIM5 Control Register 2	0x0000
R16_TIM5_SMCFGR	0x40000C08	TIM5 from Mode Control Register	0x0000
R16_TIM5_DMAINTENR	0x40000C0C	TIM5 DMA/Interrupt Enable Register	0x0000
R16_TIM5_INTFR	0x40000C10	TIM5 Interrupt Status Register	0x0000
R16_TIM5_SWEVGR	0x40000C14	TIM5 Event Generation Register	0x0000
R16_TIM5_CHCTLR1	0x40000C18	TIM5 Compare/Capture Control Register 1	0x0000
R16_TIM5_CHCTLR2	0x40000C1C	TIM5 Compare/Capture Control Register 2	0x0000
R16_TIM5_CCER	0x40000C20	TIM5 Compare/Capture Enable Register	0x0000
R16_TIM5_CNT	0x40000C24	TIM5 counter	0x0000
R16_TIM5_PSC	0x40000C28	TIM5 Counting Clock Prescaler	0x0000
R16_TIM5_ATRLR	0x40000C2C	TIM5 Auto Reload Value Register	0xFFFF
R32_TIM5_CH1CVR	0x40000C34	TIM5 Compare/Capture Register 1	0x0000
R32_TIM5_CH2CVR	0x40000C38	TIM5 Compare/Capture Register 2	0x0000
R32_TIM5_CH3CVR	0x40000C3C	TIM5 Compare/Capture Register 3	0x0000
R32_TIM5_CH4CVR	0x40000C40	TIM5 Compare/Capture Register 4	0x0000
R16_TIM5_DMACFGR	0x40000C48	TIM5 DMA Control Register	0x0000
R16_TIM5_DMAADR	0x40000C4C	TIM5 Continuous Mode DMA Address Register 0x0000	
R16_TIM5_AUX	0x40000C50	TIM5 Dual-Edge Capture Register	0x0000

Table 15-7 List of TIM9 Related Registers

name	Offset address description		Reset value
R16_TIM9_CTLR1	0x40014C00	TIM9 Control Register 1	0x0000
R16_TIM9_CTLR2	0x40014C04	TIM9 Control Register 2	0x0000
R16_TIM9_SMCFGR	0x40014C08	TIM9 from Mode Control Register	0x0000
R16_TIM9_DMAINTENR	0x40014C0C	TIM9 DMA/Interrupt Enable Register	0x0000
R16_TIM9_INTFR	0x40014C10	TIM9 Interrupt Status Register	0x0000
R16_TIM9_SWEVGR	0x40014C14	TIM9 Event Generation Register	0x0000
R16_TIM9_CHCTLR1	0x40014C18	TIM9 Compare/Capture Control Register 1	0x0000
R16_TIM9_CHCTLR2	0x40014C1C	TIM9 Compare/Capture Control Register 2	0x0000
R16_TIM9_CCER	0x40014C20	TIM9 Compare/Capture Enable Register	0x0000
R32_TIM9_CNT	0x40014C24	TIM9 counter	0x0000
R16_TIM9_PSC	0x40014C28	TIM9 Counting Clock Prescaler	0x0000
R32_TIM9_ATRLR	0x40014C2C	TIM9 Auto Reload Value Register	0x0000FFFF
R32_TIM9_CH1CVR	0x40014C34	TIM9 Compare/Capture Register 1	0x00000000
R32_TIM9_CH2CVR	0x40014C38	TIM9 Compare/Capture Register 2	0x00000000
R32_TIM9_CH3CVR	0x40014C3C	TIM9 Compare/Capture Register 3	0x00000000

R32_TIM9_CH4CVR	0x40014C40	TIM9 Compare/Capture Register 4	0x00000000
R16_TIM9_DMACFGR	0x40014C48	TIM9 DMA Control Register	0x0000
R16_TIM9_DMAADR	0x40014C4C	TIM9 Continuous Mode DMA Address Register	0x0000

Table 15-8 List of TIM10 Related Registers

name	Offset address	describe	Reset value
R16_TIM10_CTLR1	0x40015000	TIM10 Control Register 1	0x0000
R16_TIM10_CTLR2	0x40015004	TIM10 Control Register 2	0x0000
R16_TIM10_SMCFGR	0x40015008	TIM10 from Mode Control Register	0x0000
R16_TIM10_DMAINTENR	0x4001500C	TIM10 DMA/Interrupt Enable Register	0x0000
R16_TIM10_INTFR	0x40015010	TIM10 Interrupt Status Register	0x0000
R16_TIM10_SWEVGR	0x40015014	TIM10 Event Generation Register	0x0000
R16_TIM10_CHCTLR1	0x40015018	TIM10 Compare/Capture Control Register 1	0x0000
R16_TIM10_CHCTLR2	0x4001501C	TIM10 Compare/Capture Control Register 2	0x0000
R16_TIM10_CCER	0x40015020	TIM10 Compare/Capture Enable Register	0x0000
R32_TIM10_CNT	0x40015024	TIM10 counter	0x0000
R16_TIM10_PSC	0x40015028	TIM10 Counting Clock Prescaler	0x0000
R32_TIM10_ATRLR	0x4001502C	TIM10 Auto Reload Value Register	0x0000FFFF
R32_TIM10_CH1CVR	0x40015034	TIM10 Compare/Capture Register 1	0x00000000
R32_TIM10_CH2CVR	0x40015038	TIM10 Compare/Capture Register 2	0x00000000
R32_TIM10_CH3CVR	0x4001503C	TIM10 Compare/Capture Register 3	0x00000000
R32_TIM10_CH4CVR	0x40015040	TIM10 Compare/Capture Register 4	0x00000000
R16_TIM10_DMACFGR	0x40015048	TIM10 DMA Control Register	0x0000
R16_TIM10_DMAADR	0x4001504C	TIM10 Continuous Mode DMA Address Register	0x0000

Table 15-9 List of TIM11 Related Registers

name	Offset address	describe	Reset value
R16_TIM11_CTLR1	0x40015400	TIM11 Control Register 1	0x0000
R16_TIM11_CTLR2	0x40015404	TIM11 Control Register 2	0x0000
R16_TIM11_SMCFGR	0x40015408	TIM11 from Mode Control Register	0x0000
R16_TIM11_DMAINTENR	0x4001540C	TIM11 DMA/Interrupt Enable Register	0x0000
R16_TIM11_INTFR	0x40015410	TIM11 Interrupt Status Register	0x0000
R16_TIM11_SWEVGR	0x40015414	TIM11 Event Generation Register	0x0000
R16_TIM11_CHCTLR1	0x40015418	TIM11 Compare/Capture Control Register 1	0x0000
R16_TIM11_CHCTLR2	0x4001541C	TIM11 Compare/Capture Control Register 2	0x0000
R16_TIM11_CCER	0x40015420	TIM11 Compare/Capture Enable Register	0x0000
R32_TIM11_CNT	0x40015424	TIM11 Counter	0x0000
R16_TIM11_PSC	0x40015428	TIM11 Counting Clock Prescaler	0x0000
R32_TIM11_ATRLR	0x4001542C	TIM11 Auto Reload Value Register	0x0000FFFF
R32_TIM11_CH1CVR	0x40015434	TIM11 Compare/Capture Register 1	0x00000000
R32_TIM11_CH2CVR	0x40015438	TIM11 Compare/Capture Register 2	0x00000000
R32_TIM11_CH3CVR	0x4001543C	TIM11 Compare/Capture Register 3	0x00000000
R32_TIM11_CH4CVR	0x40015440	TIM11 Compare/Capture Register 4	0x00000000

R16_TIM11_DMACFGR	0x40015448	TIM11 DMA Control Register	0x0000
R16_TIM11_DMAADR	0x4001544C	TIM11 Continuous Mode DMA Address Register	0x0000

Table 15-10 List of TIM12 Related Registers

name	Offset address	describe	Reset value
R16_TIM12_CTLR1	0x40013C00	TIM12 Control Register 1	0x0000
R16_TIM12_CTLR2	0x40013C04	TIM12 Control Register 2	0x0000
R16_TIM12_SMCFGR	0x40013C08	TIM12 Slave Mode Control Register	0x0000
R16_TIM12_DMAINTENR	0x40013C0C	TIM12 DMA/Interrupt Enable Register	0x0000
R16_TIM12_INTFR	0x40013C10	TIM12 Interrupt Status Register	0x0000
R16_TIM12_SWEVGR	0x40013C14	TIM12 Event Generation Register	0x0000
R16_TIM12_CHCTLR1	0x40013C18	TIM12 Compare/Capture Control Register 1	0x0000
R16_TIM12_CHCTLR2	0x40013C1C	TIM12 Compare/Capture Control Register 2	0x0000
R16_TIM12_CCER	0x40013C20	TIM12 Compare/Capture Enable Register	0x0000
R32_TIM12_CNT	0x40013C24	TIM12 counter	0x0000
R16_TIM12_PSC	0x40013C28	TIM12 Counting Clock Prescaler	0x0000
R32_TIM12_ATRLR	0x40013C2C	TIM12 Auto Reload Value Register	0x0000FFFF
R32_TIM12_CH1CVR	0x40013C34	TIM12 Compare/Capture Register 1	0x00000000
R32_TIM12_CH2CVR	0x40013C38	TIM12 Compare/Capture Register 2	0x00000000
R32_TIM12_CH3CVR	0x40013C3C	TIM12 Compare/Capture Register 3	0x00000000
R32_TIM12_CH4CVR	0x40013C40	TIM12 Compare/Capture Register 4	0x00000000
R16_TIM12_DMACFGR	0x40013C48	TIM12 DMA Control Register	0x0000
R16_TIM12_DMAADR	0x40013C4C	TIM12 Continuous Mode DMA Address Register	0x0000

15.4.1 Control Register 1 (TIMx_CTLR1) (x=2/3/4/5/9/10/11/12)

Offset address: 0x00

Bit	name	access	describe	Reset value
15	CAPLVL	RW	In dual-edge capture mode, the capture level indicator is enabled. 1: Enable indicator function; 0: Disable indicator function. Note: When enabled, CHxCVR's [16] indicates the level corresponding to the captured value.	0
14	CAPOV	RW	Capture value mode configuration. 1: When a counter overflows before capture, the CHxCVR value is... 0xFFFF; 0: The captured value is the actual counter value.	0
[13:10] Reserved		RO is reserved.		0
[9:8] CKD[1:0]		RW	These two bits are defined in the timer clock (CK_INT) frequency and digital filter. The frequency division ratio between the sampling clocks used by the oscilloscope: 00: Tdts = Tck_int; 01: Tdts = 2xTck_int; 10: Tdts = 4xTck_int; 11: Retained.	00b
7	ARPE	RW	Automatic reinstallation pre-installation enable bit: 0: Disable automatic reload of value register (ATRLR);	0

			1: Enable the Automatic Reload Value Register (ATRLR).	
[6:5] CMS[1:0]		RW	Center alignment mode selection: 00: Edge-aligned mode. The counter moves upwards based on the direction bit (DIR). Or count downwards. 01: Center Alignment Mode 1. The counter alternates between counting up and down. Number. The channel configured for output (CCxS=00 in the CHCTLRx register). The output compare interrupt flag is only triggered when the counter counts down. set up. 10: Center Alignment Mode 2. The counter alternates between counting up and down. Number. The channel configured for output (CCxS=00 in the CHCTLRx register). The output compare interrupt flag is only triggered when the counter is counting up. set up. 11: Center Alignment Mode 3. The counter alternates between counting up and down. Number. The channel configured for output (CCxS=00 in the CHCTLRx register). The output compares the interrupt flag bit, and the counter counts up and down. The time is set. Note: When the counter is enabled (CEN=1), switching from edge-aligned mode to center-aligned mode is not allowed.	00b
4	DIR	RW	Counter direction: 1: The counter's counting mode is down; 0: The counter is in increment mode. Note: This bit is invalid when the counter is configured in center-aligned mode or encoder mode.	0
3	OPM	RW	Single pulse mode. 1: The count is updated when the next update event (clearing the CEN bit) occurs. The device stopped; 0: The counter does not stop when the next update event occurs.	0
2	URS	RW	Update request source; the software selects the source of the UEV event using this bit. 1: If update interrupts or DMA requests are enabled, only the counter will be active. An update interrupt or DMA request is only generated when there is an overflow or underflow. 0: If update interrupts or DMA requests are enabled, then either of the following will occur: The device generates an update interrupt or DMA request: - Counter overflow/underflow -Set UG position -Updates generated from the mode controller	0
1	UDIS	RW	Disabling updates allows/disallows the generation of UEV events via this bit. born. 1: Disable UEV. No update events are generated, and all registers (ATRLR, ...) are updated. PSC and CHCTLRx retain their values. If the UG bit is set or A hardware reset is issued from the mode controller, then the counter and pre-... The frequency divider was reinitialized; 0: UEV enabled. The update (UEV) event is generated by any of the following events. born: - Counter overflow/underflow -Set UG position	0

			- Updates generated from the mode controller have cached registers loaded. Enter their preloaded values.	
0	CEN	RW	Enable the counter: 1: Enable counter; 0: Counter disabled. Note: This is in the software settings.CEN CEN Bit After this step, the external clock, gating mode, and encoder mode can operate. The trigger mode can be automatically set via hardware.	0

15.4.2 Control Register 2 (TIMx_CTLR2) (x=2/3/4/5/9/10/11/12)

Offset address: 0x04

Bit	Name	access	describe	Reset value
[15:8] Reserved		RO is reserved.		0
7	TI1S	RW	TI1 Selection: 1: The TIMx_CH1, TIMx_CH2, and TIMx_CH3 pins are XORed together. Connect to TI1 input; 0: The TIMx_CH1 pin is directly connected to the TI1 input.	0
[6:4] MMS[2:0]		RW	Master mode selection: These 3 bits are used to select whether to send to the slave in master mode. Timer synchronization information (TRGO). Possible combinations are as follows: 000: Reset – The UG bit is used as a trigger output (TRGO). For example... The result is a reset triggered by the input (from the mode controller being in reset state). In this mode, the signal on TRGO will have a relative reset time to the actual reset. Delay; 001: Enable – The counter enable signal CNT_EN is used as a trigger. Output (TRGO). Sometimes it is necessary to start multiple scheduled commands at the same time. A timer or counter enables a function to run for a specified period of time. The power signal is triggered by the CEN control bit and the gated mode trigger input. The logical OR of the signal is generated. When the counter enable signal is triggered... There will be a delay on TRGO during input unless master/slave is selected. Mode (see description of the MSM bit in the TIMx_SMCFGR register); 010: The update event is selected as the trigger output (TRGO). For example, one The clock of a master timer can be used as a prescaler for a slave timer. Frequency converter; 011: Comparison pulse, occurring upon a successful capture or comparison. When the CC1IF flag is set (even if it is already high), triggering... The output sends out a positive pulse (TRGO); 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).	000b
3	CCDS	RW	1: When an update event occurs, send a DMA request for CHxCVR; 0: When CHxCVR occurs, a DMA request for CHxCVR is generated.	0
[2:1] Reserved		RO reserved.		0
0	CCPC	RW	Compare captured preload control bits: 1: The CCxE, CCxNE, and OCxM bits are preloaded; setting this bit...	0

			After that, they are only updated after the COM bit is set; 0: The CCxE, CCxNE, and OCxM bits are not preloaded. Note: This bit only applies to channels with complementary outputs.	

15.4.3 From the Mode Control Register (TIMx_SMCFGR) (x=2/3/4/5/9/10/11/12) Offset address: 0x08 Name

Bit		access	The	Reset value
15	ETP	RO	description of ETR triggers polarity selection; this bit selects whether to directly input ETR or... Invert the input ETR. 1: Invert ETR; it is active low or on the falling edge. 0: ETR, high level or rising edge valid.	0
14	ECE	RW	External clock mode 2 enabled: 1: Enable external clock mode 2; 0: Disable external clock mode 2. Note 1: The slave mode can be used with an external clock mode 2. Simultaneous use of: reset mode, gating mode, and trigger mode; however, at this time... TRGI must be It can connect to ETRF (TS bit cannot be 111b). Note 2: External clock mode and external clock mode are used simultaneously. When 12 is available, the input of the external clock is ETRF.	0
[13:12] ETPS[1:0]		RW	External trigger signal (ETRP) frequency division, the maximum frequency of this signal is not It can exceed 1/4 of the TIMxCLK frequency, and this domain can be used to... Reduce frequency. 00: Disable prescaler; 01: ETRP frequency divided by 2; 10: ETRP frequency divided by 4; 11: ETRP frequency divided by 8.	00b
[11:8] ETF [3:0]		RW	Externally triggered filtering, in fact, a digital filter is an event meter. A counter that records N events using a certain sampling frequency. This will then result in an output jump. 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; 0110: Sampling frequency Fsampling = Fdts/4, N = 6; 0111: Sampling frequency Fsampling = Fdts/4, N = 8; 1000: Sampling frequency Fsampling = Fdts/8, N = 6; 1001: Sampling frequency Fsampling = Fdts/8, N = 8; 1010: Sampling frequency Fsampling = Fdts/16, N = 5; 1011: Sampling frequency Fsampling = Fdts/16, N = 6; 1100: Sampling frequency Fsampling = Fdts/16, N = 8; 1101: Sampling frequency Fsampling = Fdts/32, N = 5; 1110: Sampling frequency Fsampling = Fdts/32, N = 6; 1111: Sampling frequency Fsampling=Fdts/32, N=8.	0000b

7	MSM	RW	Master/Slave Mode Selection: 1: Events on the trigger input (TRGI) are delayed to allow for... Perfect connection between the current timer (via TRGO) and its slave timer Synchronization. This requires synchronizing several timers to a single external timer. It is very useful in departmental events; 0: Not effective.	0
[6:4] TS[2:0]		RW	Trigger selection field: These 3 bits are used to select the trigger input of the synchronization counter. Entering the source. 000: Internal Trigger 0 (ITR0); 100: TI1 edge detector (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.	000b
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. Mode. 000: Driven by the internal clock CK_INT; 001: Encoder Mode 1, based on the TI1FP1 level, core calculation... The counter increments or decrements on the edge of the TI2FP2; 010: Encoder mode 2, based on the TI2FP2 level, core calculation... The counter increments or decrements on the edge of TI1FP1; 011: Encoder mode 3, based on the input level of another signal, The core counter increments and decrements on the edges of TI1FP1 and TI2FP2; 100: Reset mode, the rising edge of the trigger input (TRGI) will initialize The counter is updated, and a signal to update the register is generated; 101: Gated mode, counting occurs when the trigger input (TRGI) is high. The counter's clock is on; when the trigger input goes low, the counter stops. The start and stop of the counter are controlled; 110: Trigger mode, the counter is triggered on the rising edge of the trigger input TRGI. Only the start-up of the counter is controlled; 111: External clock mode 1, selected trigger input (TRGI) The rising edge drives the counter.	000b

15.4.4 DMA/Interrupt Enable Register (TIMx_DMAINTENR) (x=2/3/4/5/9/10/11/12) Offset Address: 0x0C

Bit	name	access	describe	Reset value
15	Reserved	RO	reserved.	0
14	TDE	RW	Trigger DMA request enable bit: 1: Allow DMA requests to be triggered; 0: Disable DMA requests.	0
13	Reserved	RO	is reserved.	0

12	CC4DE	RW	Compare the DMA request enable bits of capture channel 4: 1: Allow comparison of DMA requests for capturing channel 4; 0: Disable comparison of DMA requests for capture channel 4.	0
11	CC3DE	RW	Compare the DMA request enable bits of capture channel 3: 1: Allow comparison of DMA requests for capturing channel 3; 0: Disable comparison of DMA requests for capture channel 3.	0
10	CC2DE	RW	Compare the DMA request enable bits of capture channel 2: 1: Allow comparison of DMA requests for capturing channel 2; 0: Disable comparison of DMA requests for capture channel 2.	0
9	CC1DE	RW	Compare the DMA request enable bits of capture channel 1: 1: Allow comparison of DMA requests for capturing channel 1; 0: Disable comparison of DMA requests for capturing channel 1.	0
8	UDE	RW	Updated DMA request enable bit: 1: Allow DMA requests for updates; 0: DMA requests for updates are prohibited.	0
7	Reserved	RO reserved.		0
6	TIE	RW	Interrupt enable bit triggered: 1: Enable interrupt triggering; 0: Interrupt triggering is disabled.	0
5	Reserved	RO reserved.		0
4	CC4IE	RW	Compare the interrupt enable bit of capture channel 4: 1: Enable comparison capture channel 4 interrupt; 0: Disable comparison capture channel 4 interruption.	0
3	CC3IE	RW	Compare the interrupt enable bit of capture channel 3: 1: Enable comparison capture channel 3 interrupt; 0: Disable comparison capture channel 3 interruption.	0
2	CC2IE	RW	Compare the interrupt enable bit of capture channel 2: 1: Enable comparison capture channel 2 interrupt; 0: Disable comparison capture channel 2 interruption.	0
1	CC1IE	RW	Compare the interrupt enable bit of capture channel 1: 1: Enable comparison capture channel 1 interrupt; 0: Disable comparison capture channel 1 interruption.	0
0	UIE	RW	Update the interrupt enable bit: 1: Allow update interruptions; 0: Prevent update interruption.	0

15.4.5 Interrupt Status Register (R16_TIMx_INTFR) (x=2/3/4/5/9/10/11/12) Offset Address: 0x10 Name

Bit		access	describe	Reset value
[15:13] Reserved		RO is reserved.		0
12	CC4OF	RW0 compares the capture channel 4 duplicate capture flag.		0
11	CC3OF	RW0 compares the capture channel 3 duplicate capture flag.		0
10	CC2OF	RW0 compares the capture channel 2 duplicate capture flag.		0
9	CC1OF	RW0 compares the capture channel 1 duplicate capture flag; this is only used to compare capture 0.		

			When the channel is configured for input capture mode. This flag is set by hardware. This bit can be cleared by writing 0 to it in the software. 1: When the counter value is captured into the capture compare register, CC1IF The status has been set; 0: No duplicate captures were generated.	
[8:7] Reserved		RO Reserved.		0
6	TIF	RW0	Trigger interrupt flag, configured by hardware when a trigger event occurs. The position bit is cleared by software. Triggering events include those from gating mode. In other modes, a valid edge is detected at the TRGI input. Or any edge in gating mode. 1: Trigger event generated; 0: No trigger event is generated.	0
5	Reserved	RO is reserved.		0
4	CC4IF	RW0 compares the interrupt flag of capture channel 4.		0
3	CC3IF	RW0 compares the interrupt flag of capture channel 3.		0
2	CC2IF	RW0 compares the interrupt flag of capture channel		0
1	CC1IF	RW0	2. Compares the interrupt flag of capture channel 1. If the comparison capture channel is configured in output mode, when the counter value is compared with... This bit is set by hardware when the comparison values match, but in center-symmetric mode... Except for the one below. This bit is cleared by the software. 1: The value of the core counter matches the value of the compare capture register 1. 0: No match occurred. If the comparison capture channel 1 is configured as input mode, when a capture event occurs... When this occurs, the bit is set by hardware; it is cleared by software or by reading the ratio. Clear the capture register. 1: The counter value has been captured by comparing capture register 1. 0: No input capture occurred;	0
0	UIF	RW0	Update interrupt flag; this bit is set by hardware when an update event occurs. The bit is cleared by the software. 1: An update interruption occurred; 0: No update event occurred. The following situations will trigger an update event: If UDIS=0, when the repeat counter value overflows or underflows; If URS=0 and UDIS=0, when the UG bit is set, or when the software is used to... When the counter core is reinitialized; If URS=0 and UDIS=0, the event is retried when the counter CNT is triggered. During initialization;	0

15.4.6 Event Generation Register (TIMx_SWEVGR) (x=2/3/4/5/9/10/11/12) Offset Address: 0x14 Name

Bit		access	describe	Reset value
[15:7] Reserved		RO is reserved.		0
6	TG	WO	This is the trigger event generation bit, set by software and cleared by hardware. This generates a triggering event.	0

			1: A trigger event is generated, and the TIF bit is set. If the corresponding bit is enabled... Interrupts and DMA will generate corresponding interrupts and DMA operations; 0: No action.	
5	Reserved	RO is reserved.		0
4	CC4G	WO	Compare capture event generates bit 4. Compare capture event 4 is generated.	0
3	CC3G	WO	Compare capture event generates bit 3. A comparison capture event 3 is generated.	0
2	CC2G	WO	Compare capture event generates bit 2. Compare capture event 2 is generated.	0
1	CC1G	WO	The compare capture event generates bit 1, thus generating compare capture event 1. This bit... Set by software, cleared by hardware. Used to generate a compare capture. event. 1: Generate a compare capture event on compare capture channel 1: If the comparison capture channel 1 is configured as an output: set the CC1IF bit. If... If a corresponding interrupt and DMA are available, then the corresponding interrupt and DMA will be generated; if Compare capture channel 1 is configured as input: the current core counter value. Captured to Compare-Capture Register 1; Set CC1IF bit, if enabled If the corresponding interrupt and DMA are triggered, then the corresponding interrupt and DMA will be generated. If CC1IF is already set, then set the CC1OF bit. 0: No action;	0
0	UG	WO	Update event generation bit: This bit is set by the software to generate an update event. Automatically reset by hardware. 1: Initialize the counter and generate an update event; 0: No action. Note: The prescaler counter is also cleared, but the prescaler coefficient remains unchanged. In center-symmetric mode or increment mode, the core counter is cleared; in decrement mode, the core counter takes the value from the reload register.	0

15.4.7 Compare/Capture Control Register 1 (TIMx_CHCTLR1) (x=2/3/4/5/9/10/11/12) Offset Address: 0x18 The channel can be used for input (capture mode) or output (compare mode), and the direction of the channel is defined by the corresponding CCxS bits. This register...

The functions of other bits differ between input and output modes. OCxx describes the channel's function in output mode, and ICxx describes the channel's function in output mode. Functionality 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):

Bitname		access	describe	Reset value
15	OC2CE	RW	Compare capture channel 2 to clear enable bit: 1. Once a high level is detected at the ETRF input, clear the OC2REF bit; 0: OC2REF is not affected by ETRF input.	0
[14:12]	OC2M[2:0]	RW	compares the capture channel 2 mode setting field.	000b

			These 3 bits define the action of the output reference signal OC2REF, and OC2REF determines the values of OC2 and OC2N. OC2REF is a high-level signal. The effective levels of OC2 and OC2N depend on CC2P and CC2NP, respectively. Bit. 000: Freeze. Compare the value of the capture register with the core counter. The comparison value has no effect on OC2REF; 001: Forced to active level. When the core counter is compared with the capture level. If the value of register 2 is the same, force OC2REF to go high; 010: Forced to invalid level. When the value of the core counter is compared with... If the capture register 2 is the same, force OC2REF to go low; 011: Toggle. When the core counter is compared with the value of compare capture register 2. When they are the same, toggle the level of OC2REF; 100: Force to invalid level. Force OC2REF to low; 101: Force to active level. Force OC2REF to high; 110: PWM Mode 1: When counting up, once the core counter... Channel 2 is active when the value is less than the value in the compare-capture register. Otherwise, it is an invalid level; during down-counting, once the core counter... Channel 2 is at an invalid level when the value is greater than that of the comparison capture register. (OC2REF=0), otherwise it is an active level (OC2REF=1); 111: PWM Mode 2: When counting up, once the core counter... Channel 2 is at an invalid level when the value is less than the value in the compare capture register. Otherwise, it is an active level; during down-counting, once the core counter... Channel 2 is active when the value is greater than the value in the compare-capture register. (OC2REF=1), otherwise it is an invalid level (OC2REF=0). Note: Once the level is set and OC2S=00b, this bit will not... LOCK 3 can be modified. In PWM mode. 1 or PWM 2 model In this case, only when the comparison result changes or the output comparison mode is switched from frozen mode will the comparison be allowed to proceed. The OC2REF level only changes when switching modes. PWM .
11	OC2PE	RW	Compare capture register 2 preload enable bit: 1: Enable the preload function of Compare-Capture Register 2 for read and write operations. Operate only on the preload register, compare the preload of capture register 2. The value is loaded into the current shadow register when the update event arrives. middle; 0: Disables the preload function of compare capture register 2, allowing writing at any time. Enter the comparison capture register 2, and the newly written value takes effect immediately. use. Note: Once the level is set... LOCK And 3 If CC2S=00, this bit cannot be modified. It can only be used in single-pulse mode (PWM mode) without verification. PWM pattern, otherwise its movement register. The answer is uncertain.	0
10	OC2FE	RW	Compare-capture channel 2 fast enable bit, this bit is used to speed up the comparison capture. Obtain the channel output in response to the triggered input event. 1: The effective edge of the input to the flip-flop acts as if a time difference has occurred. A better match. Therefore, OC is set to the comparison level and compared with the comparison result. Irrelevant. The valid edge of the sampling trigger and the comparison capture channel 2 output.	0

			The delay between clock cycles is reduced to 3 clock cycles; 0: Compare the capture value with the counter and the value of the comparison capture register 1. Channel 2 operates normally, even if the trigger is on. When the trigger... When the input has a valid edge, the compare capture channel 2 is activated and the output is captured. The minimum delay is 5 clock cycles. Note: OC2FE only applies when the channel is configured in ON mode.PWM1 and PWM2 are working.	
[9:8] CC2S[1:0]		RW	Compare capture channel 2 input selection field: 00: Compare capture channel 2 is configured as output; 01: Compare capture channel 2 is configured as an input, IC2 is mapped to TI2. superior; 10: Compare capture channel 2 is configured as an input, IC2 is mapped to TI1. superior; 11: Compare capture channel 2 is configured as an input, IC2 is mapped to TRC. Above. This mode only works when the internal trigger input is selected (by...). TS bit selection). Note: Comparison capture channels only occurs when the channel is off (CC2E is zero). Only number 2 is writable.	00b
7	OC1CE	RW compares and captures channel 1 to clear the enable bit.		0
[6:4] OC1M[2:0]		RW compares the capture channel 1 mode setting field.		0
3	OC1PE	RW compares the capture register 1 preload enable bit.		0
2	OC1FE	RW compares and captures the channel 1 fast enable bit.		0
[1:0] CC1S[1:0]		RW compares the capture channel 1 input selection field.		0

Capture mode (pin direction is input):

Bit	name	access	describe	Reset value
[15:12] IC2F[3:0]		RW	Input capture filter 2 configuration field, these bits set the TI1 input The sampling frequency and digital filter length. A digital filter consists of a... The event counter consists of events that record N events and then generate a new one. The output jump. 0000: No filter, sampled in 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; 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;	0000b

			1111: Sampling frequency Fsampling=Fdts/32, N=8.	
[11:10] IC2PSC[1:0]		RW	Compare the prescaler configuration field of capture channel 2; these 2 bits define the comparison... Capture the prescaler coefficients for channel 2. Once CC1E=0, the prescaler... Device reset. 00: No prescaler, captures every edge detected on the input port. Each step triggers a capture; 01: A capture is triggered every 2 events; 10: A capture is triggered every 4 events; 11: A capture is triggered once every 8 events.	00b
[9:8] CC2S[1:0]		RW	Compare the capture channel 2 input selection field; these two bits define the channel's... Input/output, and selection of input pins. 00: Compare capture channel 1. Channel is configured as output. 01: Compare and capture channel 1. Channel is configured as input, IC1 mapping. At TI1; 10: Compare and capture channel 1. Channel is configured as input, IC1 mapping. At TI2; 11: Compare and capture channel 1. Channel is configured as input, IC1 mapping. On TRC. This mode only works when the internal trigger input is selected. Time (selected by the TS bit). Note: CC1S is only writable when the channel is closed (CC1E is 0).	00b
[7:4] IC1F[3:0]		RW	Input Capture Filter 1 Configuration Domain.	0
[3:2] IC1PSC[1:0]		RW	compares the capture channel 1 prescaler configuration field.	0
[1:0] CC1S[1:0]		RW	compares the capture channel 1 input selection field.	0

15.4.8 Compare/Capture Control Register 2 (TIMx_CHCTLR2) (x=2/3/4/5/9/10/11/12) Offset Address: 0x1C

The channel can be used for input (capture mode) or output (compare mode), and the direction of the channel is defined by the corresponding CCxS bits. This register...

The functions of other bits differ between input and output modes. OCxx describes the channel's function in output mode, and ICxx describes the channel's function in output mode.

Functionality 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):

Bitname		access	describe	Reset value
15	OC4CE	RW	compares and captures channel 4, clearing the enable bit.	0
[14:12]	OC4M[2:0]	RW	compares the capture channel 4 mode setting field.	0
11	OC4PE	RW	compares the capture register 4 preload enable bit.	0
10	OC4FE	RW	compares the capture channel 4 fast enable bit.	0
[9:8]	CC4S[1:0]	RW	compares the capture channel 4 input selection field.	0
7	OC3CE	RW	compares and captures channel 3 to clear the enable bit.	0
[6:4]	OC3M[2:0]	RW	compares the capture channel 3 mode setting field.	0
3	OC3PE	RW	compares capture register 3 preload enable bit.	0

2	OC3FE	RW	compares the capture channel 3 fast enable bit.	0
[1:0]	CC3S[1:0]	RW	compares the capture channel 3 input selection field.	0

Capture mode (pin direction is input):

Bit	name	access	describe	Reset value
[15:12]	IC4F[3:0]	RW	Input Capture Filter 4 Configuration Domain.	0
[11:10]	IC4PSC[1:0]	RW	compares and captures the prescaler configuration field of channel 4.	0
[9:8]	CC4S[1:0]	RW	compares the capture channel 4 input selection field.	0
[7:4]	IC3F[3:0]	RW	Input Capture Filter 3 Configuration Domain.	0
[3:2]	IC3PSC[1:0]	RW	compares the capture channel 3 prescaler configuration field.	0
[1:0]	CC3S[1:0]	RW	compares the capture channel 3 input selection field.	0

15.4.9 Compare/Capture Enable Register (TIMx_CCER) (x=2/3/4/5/9/10/11/12) Offset Address: 0x20

Bit	name	access	describe	Reset value
[15:14]	Reserved	RO	is reserved.	0
13	CC4P	RW	compares the capture channel 4 output polarity setting bit.	0
12	CC4E	RW	compares and captures the output enable bit of channel 4.	0
[11:10]	Reserved	RO	is reserved.	0
9	CC3P	RW	compares the capture channel 3 output polarity setting bit.	0
8	CC3E	RW	compares and captures the output enable bit of channel 3.	0
[7:6]	Reserved	RO	is reserved.	0
5	CC2P	RW	compares the capture channel 2 output polarity setting bit.	0
4	CC2E	RW	compares and captures the channel 2 output enable bit.	0
[3:2]	Reserved	RO	is reserved.	0
1	CC1P	RW	Compare the capture channel 1 output polarity setting bit. Channel CC1 is configured as an output: 1: OC1 is active low; 0: OC1 is active high. CC1 channel configured as input: This bit selects whether IC1 or its inverted signal is used as the trigger or catch Signal received. 1: Inverting: Capture occurs at the falling edge of IC1; when used as an external contact. When the transmitter is activated, IC1 is inverted; 0: Non-inverting: Capture occurs on the rising edge of IC1; when used as an external... When the trigger is activated, IC1 is not inverted.	0
0	CC1E	RW	Compare the capture channel 1 output enable bit. Channel CC1 is configured as an output: 1: Enable: The OC1 signal is output to the corresponding output pin; 0: Off: OC1 disables output. CC1 channel configured as input: This bit determines whether the counter value can be captured into TIMx_CH1CVR. register. 1: Capture Enable;	0

			0: Capture prohibited.	

15.4.10 General Purpose Timer Counter (TIMx_CNT) (x=2/3/4/5) Offset

Address: 0x24

Bit	Name	Access	the real-time value of the counter that describes	Reset value
[15:0]	CNT[15:0]	the RW	timer.	0

15.4.11 General Purpose Timer Counter (TIMx_CNT) (x=9/10/11/12) Offset

Address: 0x24

31	30	29	28	27	26	25	24	23	22	21	2019	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	the real-time value of the counter that describes	Reset value
[31:0]	CNT[31:0]	the RW	timer.	0

15.4.12 Counter Clock Prescaler (TIMx_PSC) (x=2/3/4/5/9/10/11/12) Offset Address: 0x28

Bit	name	access		Reset value
[15:0]	PSC[15:0]	RW	Describes the division factor of the timer's prescaler; the clock frequency of the counter. It equals the input frequency of the frequency divider / (PSC+1).	0

15.4.13 Automatic Reload Value Register (TIMx_ATRLR) (x=2/3/4/5) Offset

Address: 0x2C

Bit	Name	access	For a	Reset value
[15:0]	ARR[15:0]	RW	description of the counter threshold, when ATRLR acts, and its updates, please read section 15.2.4. Section; the counter stops when ATRLR is empty.	0xFFFF

15.4.14 Automatic Reload Value Register (TIMx_ATRLR) (x=9/10/11/12) Offset

address: 0x2C

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
ARR[31:16]														

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

Bit	name	access	describe	Reset value
[31:0]	ARR[31:0]	RW	For counter thresholds, ATRLR activation and update information, please refer to section 15.2.4. Section; the counter stops when ATRLR is empty.	0xFFFF FFFF

15.4.15 Compare/Capture Register 1 (TIMx_CH1CVR) (x=2/3/4/5)

Offset address: 0x34

31	30 29 28 27 26 25 24 23 22 21	20 19 18 17 16
	Reserved	LEVEL1

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

Bitname	access	describe	Reset value
[31:17] Reserved	RO is reserved.		0
16 LEVEL1	RO is the level indicator bit corresponding to the captured value.		0
[15:0] CH1CVR[15:0]	RW compares the value of capture register channel 1.		0

15.4.16 Compare/Capture Register 1 (TIMx_CH1CVR) (x=9/10/11/12)

Offset address: 0x34

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
CH1CVR[31:16]					

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

Bit	name	access	describe	Reset value
[31:0]	CH1CVR[31:0]	RW compares the value of capture register channel 1.		0

15.4.17 Compare/Capture Register 2 (TIMx_CH2CVR) (x=2/3/4/5)

Offset address: 0x38

31	30 29 28 27 26 25 24 23 22 21 20 19 18 17	16
	Reserved	LEVEL2

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

Bitname	access	describe	Reset value
[31:17] Reserved	RO is reserved.		0
16 LEVEL2	RO is the level indicator bit corresponding to the captured value.		0
[15:0] CH2CVR[15:0]	RW compares the value of capture register channel 2.		0

15.4.18 Compare/Capture Register 2 (TIMx_CH2CVR) (x=9/10/11/12)

Offset address: 0x38

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
CH2CVR[31:16]					

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

CH2CVR[15:0]

Bit	name	access	describe	Reset value
[31:0]	CH2CVR[31:0]	RW	compares the value of capture register channel 2.	0

15.4.19 Compare/Capture Register 3 (TIMx_CH3CVR) (x=2/3/4/5) Offset Address:

0x3C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LEVEL3

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

Bit	name	access	describe	Reset value
[31:17]	Reserved	RO	is reserved.	0
16	LEVEL3	RO	is the level indicator bit corresponding to the captured value.	0
[15:0]	CH3CVR[15:0]	RW	compares the value of capture register channel 3.	0

15.4.20 Compare/Capture Register 3 (TIMx_CH3CVR) (x=9/10/11/12)

Offset address: 0x3C

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
CH3CVR[31:16]														

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

Bit	name	access	describe	Reset value
[31:0]	CH3CVR[31:0]	RW	compares the value of capture register channel 3.	0

15.4.21 Compare/Capture Register 4 (TIMx_CH4CVR) (x=2/3/4/5) Offset address:

0x40

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LEVEL4

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

Bit	name	access	describe	Reset value
[31:17]	Reserved	RO	is reserved.	0
16	LEVEL4	RO	is the level indicator bit corresponding to the captured value.	0
[15:0]	CH4CVR[15:0]	RW	compares the value of capture register channel 4.	0

15.4.22 Compare/Capture Register 4 (TIMx_CH4CVR) (x=9/10/11/12)

Offset address: 0x40

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
CH4CVR[31:16]														

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

Bit	name	access	describe	Reset value
[31:0]	CH4CVR[31:0]	RW	compares the value of capture register channel 4.	0

15.4.23 DMA Control Register (TIMx_DMACFGR) (x=2/3/4/5/9/10/11/12)

Offset address: 0x48

Bit	name	access	describe	Reset value
[15:13]	Reserved	RO	is reserved.	0
[12:8]	DBL [4:0]		The length of continuous transmission in RW DMA, the actual value is the value of this field + 1.	0
[7:5]	Reserved	RO	is reserved.	0
[4:0]	DBA[4:0]	RW	These bits define the DMA's access from control register 1 in continuous mode. Offset at the address.	0

15.4.24 DMA Address Register in Continuous Mode (TIMx_DMAADR) (x=2/3/4/5/9/10/11/12)

Offset address: 0x4C

Bit	Name	access	describe	Reset value
[15:0]	DMAB[15:0]	RW	In continuous mode, the DMA address.	0

15.4.25 Double-edge capture register (TIMx_AUX) (x=2/3/4/5)

offset address:

0x50	bit name	access	describe	Reset value
[15:3]	Reserved	R0	is reserved.	0
2	CAP_ED_CH4	RW	Double-edge capture enabled for channel 4: 1: Enable double-edge capture of channel 4; 0: Disable double-edge capture enable for channel 4.	0
1	CAP_ED_CH3	RW	Double-edge capture enable for channel 3: 1: Enable double-edge capture of channel 3; 0: Disable double-edge capture enable for channel 3.	0
0	CAP_ED_CH2	RW	Double-edge capture enable for channel 2: 1: Enable double-edge capture of channel 2; 0: Disable double-edge capture enable for channel 2.	0

Chapter 16 Basic Timers (BCTM)

The basic timer module contains two 16-bit auto-reloadable timers (TIM6 and TIM7) for counting and updating events.

Generates interrupts or DMA requests. TIM6 and TIM7 support a 16-bit programmable prescaler. It can provide clocking for digital-to-analog converters (DACs), triggering interrupts or DMA requests.

The DAC's synchronization circuit. The basic timers are independent of each other and do not share any resources.

16.1 Main Features

The main features of a basic timer include:

ÿ 16-bit auto-reload counter, supports incrementing count mode

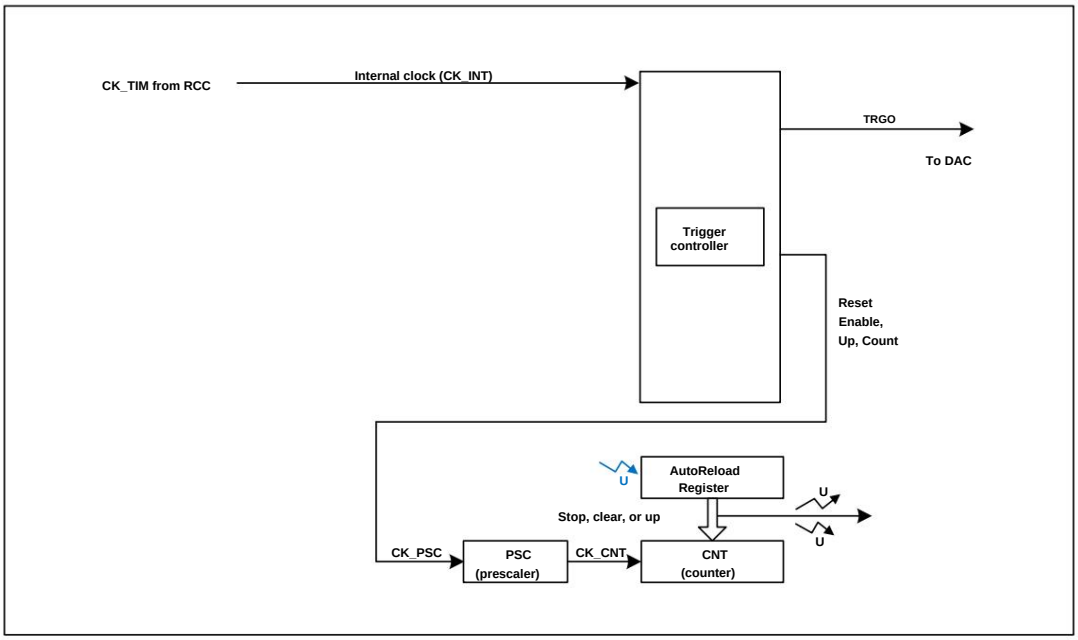
ÿ 16-bit prescaler, with a dynamically adjustable division factor from 1 to 65536.

Triggering DAC Synchronization Circuit

ÿ An interrupt or DMA request is generated during an update event .

16.2 Principles and Structure

Figure 16-1 Block diagram of a basic timer



16.2.1 Overview As

shown in Figure 16-1, the basic timer can be roughly divided into two parts: the input clock part and the core counter part.

The basic timer's clock comes from the HB bus clock (CK_INT). These input clock signals undergo various filtering settings.

After frequency equalization, it becomes the CK_PSC clock, which is output to the core counter section. Additionally, these complex clock sources can also be used as TRGO.

Output to DAC peripheral.

The core of the basic timer is a 16-bit counter (CNT). CK_PSC is divided by the prescaler (PSC) to become CK_CNT.

Finally, the value is sent to the CNT, which supports increment counting mode and has an automatic reload register (ATRLR) that returns a value at the end of each counting cycle.

Reload the initial values for CNT.

16.2.2 Difference between basic timers and general-purpose timers

Compared to general-purpose timers, basic timers lack the following features:

1) The basic timer lacks both count down and count up/down modes.

2) The basic timer lacks four independent compare-capture channels.

3) The basic timer does not support external signal control of the timer.

4) The basic timer does not support incremental encoding, cascading, or synchronization between timers.

16.2.3 Clock Input The

clock for the basic timer is provided by the internal clock CK_INT.

16.2.4 Counters and Peripherals

The CK_PSC input is fed to the prescaler (PSC) for frequency division. The PSC is 16-bit, and the actual division factor is equivalent to R16_TIMx_PSC.

The value is incremented by 1. CK_PSC becomes CK_CNT after passing through PSC. Changing the value of R16_TIM1_PSC does not take effect immediately, but will occur during the update event.

The update is then sent to the PSC. Update events include clearing and resetting the UG bit. There is also an Automatic Reload Value Register (ATRLR) that can be used to set the count.

The counter threshold is set in the Automatic Reload Value Register (ATRLR). When the CNT count value reaches the counter threshold set in the ATRLR, the CNT will be cleared to zero.

16.3 Debug Mode

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

16.4 Register Description

Table 16-1 List of TIM6 Related Registers

name	Offset address	describe	Reset value
R16_TIM6_CTLR1	0x40001000	TIM6 Control Register 1	0x0000
R16_TIM6_CTLR2	0x40001004	TIM6 Control Register 2	0x0000
R16_TIM6_DMAINTENR	0x4000100C	TIM6 DMA/Interrupt Enable Register	0x0000
R16_TIM6_INTFR	0x40001010	TIM6 Interrupt Status Register	0x0000
R16_TIM6_SWEVGR	0x40001014	TIM6 Event Generation Register	0x0000
R16_TIM6_CNT	0x40001024	TIM6 Counter	0x0000
R16_TIM6_PSC	0x40001028	TIM6 Counting Clock Prescaler	0x0000
R16_TIM6_ATRLR	0x4000102C	TIM6 Auto Reload Value Register	0xFFFF

Table 16-2 List of TIM7 Related Registers

name	Offset address description		Reset value
R16_TIM7_CTLR1	0x40001400	TIM7 Control Register 1	0x0000
R16_TIM7_CTLR2	0x40001404	TIM7 Control Register 2	0x0000
R16_TIM7_DMAINTENR	0x4000140C	TIM7 DMA/Interrupt Enable Register	0x0000
R16_TIM7_INTFR	0x40001410	TIM7 Interrupt Status Register	0x0000
R16_TIM7_CNT	0x40001424	TIM7 Counter	0x0000
R16_TIM7_PSC	0x40001428	TIM7 Counting Clock Prescaler	0x0000
R16_TIM7_ATRLR	0x4000142C	TIM7 Auto Reload Value Register	0xFFFF

16.4.1 Control Register 1 (TIMx_CTLR1) (x=6/7) Offset Address: 0x00

Name

Bit		access	describe	Reset value
[15:8] Reserved		RO is reserved.		0

7	ARPE	RW	Automatic reinstallation pre-installation enable bit: 1: Enable the Automatic Reload Value Register (ATRLR); 0: Disable automatic reload of value register (ATRLR).	0
[6:4] Reserved		RO is reserved.		0
3	OPM	RW	Single-pulse mode. 1: The count is updated when the next update event (clearing the CEN bit) occurs. The device stopped; 0: The counter does not stop when the next update event occurs.	0
2	URS	RW	Update request source; the software selects the source of the UEV event using this bit. 1: If update interrupts or DMA requests are enabled, only the counter will be active. An update interrupt or DMA request is only generated when there is an overflow or underflow. 0: If update interrupts or DMA requests are enabled, then either of the following will occur: The device generates an update interrupt or a DMA request. - Counter overflow/underflow -Set UG position - Updates generated from the mode controller.	0
1	UDIS	RW	Disabling updates allows/disallows the generation of UEV events via this bit. born. 1: Disable UEV. No update events are generated, and all registers (ATRLR, ...) are updated. PSC and CHCTLRx retain their values. If the UG bit is set or A hardware reset is issued from the mode controller, then the counter and pre-... The frequency divider was reinitialized. 0: UEV enabled. The update (UEV) event is generated by any of the following events. Student: - Counter overflow/underflow -Set UG position - Updates generated from the mode controller have cached registers loaded. Enter their preloaded values.	0
0	CEN	RW	Enable the counter: 1: Enable counter; 0: Counter disabled.	0

16.4.2 Control Register 2 (TIMx_CTLR2) (x=6/7) Offset Address: 0x04 Name

Bit		access	describe	Reset value
[15:7] Reserved		RO reserved.		0
[6:4] MMS[2:0]		RW	Master mode selection: These 3 bits are used to select whether to send to the slave in master mode. Timer synchronization information (TRGO). Possible combinations are as follows: 000: Reset – The UG bit is used as a trigger output (TRGO). For example... The result is a reset triggered by the input (from the mode controller being in reset state). In this mode, the signal on TRGO will have a relative reset time to the actual reset. Delay; 001: Enable – The counter enable signal CNT_EN is used as a trigger. Output (TRGO). Sometimes it is necessary to start multiple scheduled commands at the same time. A timer or counter enables a function to run for a specified period of time. The power signal is triggered by the CEN control bit and the gated mode trigger input.	0

			The logical OR of the signal is generated. When the counter enable signal is triggered... There will be a delay on TRGO during input unless master/slave is selected. Mode (see description of the MSM bit in the TIMx_SMCFGR register); 010: The update event is selected as the trigger output (TRGO). For example, one The clock of a master timer can be used as a prescaler for a slave timer. Frequency converter.	
[3:0] Reserved		RO is reserved.		0

16.4.3 DMA/Interrupt Enable Register (TIMx_DMAINTENR) (x=6/7) Offset Address: 0x0C Name

Bit		access	describe	Reset value
[15:9] Reserved		RO reserved.		0
8	UDE	RW	Updated DMA request enable bit: 1: Allow DMA requests for updates; 0: DMA requests for updates are prohibited.	0
[7:1] Reserved		RO is reserved.		0
0	UIE	RW	Update the interrupt enable bit: 1: Allow update interruptions; 0: Prevent update interruption.	0

16.4.4 Interrupt Status Register (R16_TIMx_INTFR) (x=6/7) Offset Address: 0x10

Bit	name	access	describe	Reset value
[15:1] Reserved		RO is reserved.		0
0	UIF	RW0	The update interrupt flag is set by hardware when an update event occurs. The bit is cleared by the software. 1: An update interruption occurred; The following situations will trigger an update event: If UDIS=0, when the repeat counter value overflows or underflows; If URS=0 and UDIS=0, when the UG bit is set, or when the software is used to... When the counter core is reinitialized; 0: No update event occurred.	0

16.4.5 Event Generation Register (TIMx_SWEVGR) (x=6/7)

Offset address: 0x14

Bit	name	access	describe	Reset value
[15:1] Reserved		RO Reserved.		0
0	UG	WO	Update event generation bit, used to generate an update event. This bit is set by software. Automatically reset by hardware. 1: Initialize the counter and generate an update event; 0: No action. Note: The prescaler counter is also cleared, but the prescaler coefficient remains unchanged. In increment mode, the core counter is cleared.	0

16.4.6 General Purpose Timer Counter (TIMx_CNT) (x=6/7)

Offset address: 0x24

Bit	name	access	describe	Reset value
[15:0]	CNT[15:0]	RW	The real-time value of the timer's counter.	0

16.4.7 Counting Clock Prescaler (TIMx_PSC) (x=6/7)

Offset address: 0x28

Bit	name	access		Reset value
[15:0]	PSC[15:0]	RW	Describes the division factor of the timer's prescaler; the clock frequency of the counter. It equals the input frequency of the frequency divider / (PSC+1).	0

16.4.8 Automatic Reload Value Register (TIMx_ATRLR) (x=6/7)

Offset address: 0x2C

Bit	name	access	describe	Reset value
[15:0]	ARR[15:0]	RW	For counter thresholds, ATRLR activation and update information, please refer to section 16.2.4. Section; the counter stops when ATRLR is empty.	0xFFFF

Chapter 17 Low-Power Timers (LPTIM)

The low-power timer module contains two 16-bit uplink counting timers. LPTIM has multiple selectable clock sources, enabling it to operate in all power modes. LPTIM can also operate without an internal clock source, allowing it to be used as a "pulse counter." Furthermore, LPTIM can wake the system from low-power mode, making it ideal for implementing "timeout functions" with extremely low power consumption.

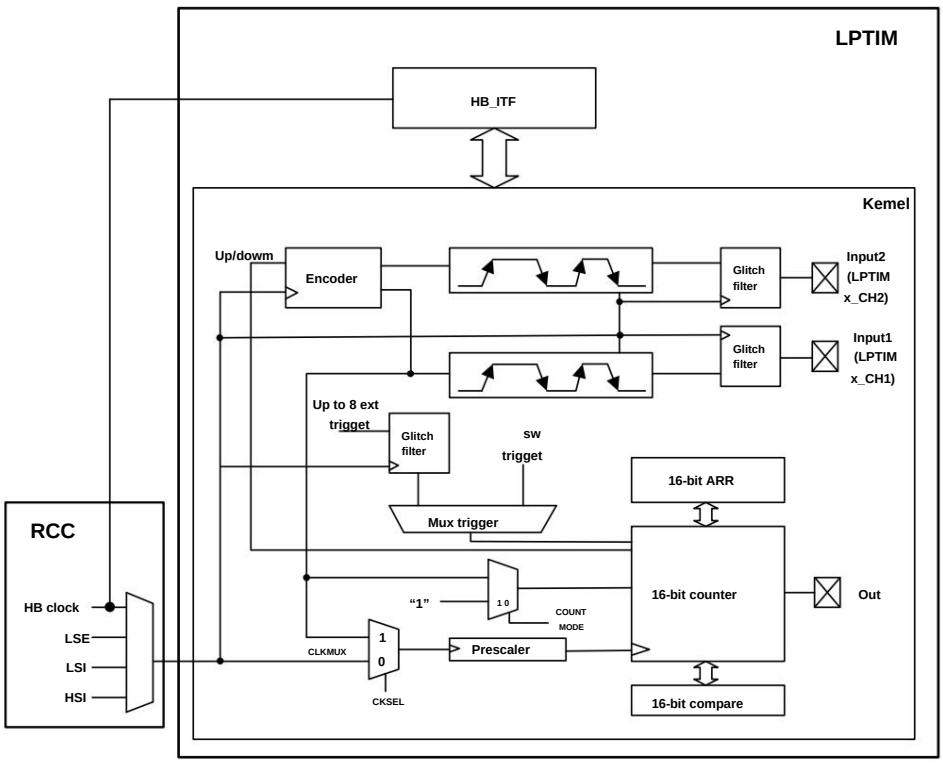
17.1 Main Features

ÿ 16-bit uplink counter ÿ 3-bit prescaler, supporting 8 division ratios (1, 2, 4, 8, 16, 32, 64, 128) ÿ Selectable clock sources Internal clock source: LSE, LSI, HSI, or HB clock External clock source: External clock on the LPTIM input ÿ 16-bit ARR auto-reload register ÿ 16-bit compare register ÿ Continuous/single trigger mode ÿ Selectable software/ hardware input triggers ÿ Programmable digital interference filter ÿ Configurable output PWM waveform ÿ Configurable I/ O polarity ÿ Encoder mode

17.2 LPTIM Function Description

17.2.1 LPTIM Block Diagram

Figure 17-1 Low-power timer block diagram



17.3 LPTIM Trigger Mapping

The information regarding the LPTIM external trigger connection is shown below:

Table 17-1 LPTIMx External Trigger Connections

TRIGSEL[1:0]	External trigger
LPTIMx_TRG_00	LPTIMx_ETR
LPTIMx_TRG_01	RTC_ALARM

17.3.1 LPTIM Reset and Clock

The LPTIM module reset is controlled by the LPTIMRST bit of the RCC_HB1PRSTR register. Setting it to 0 has no effect, while setting it to 1 resets the module.

The LPTIM module clock enable is controlled by the LPTIMEN bit in the RCC_HB1PCENR register. Setting it to 0 disables the module clock, and setting it to 1 disables it.

Module clock is on.

LPTIM's counting clock is provided by multiple selectable clock sources, which can be categorized as internal clock sources and external clock sources.

When using an internal clock source for counting, the internal clock source can be used to control HB via the CLKMX_SEL bit in the LPTIMx_CFGR register.

Four clock sources can be selected: LSI, LSE, and HSI. Additionally, LPTIM can use an external clock injected via the external input LPTIMx_CH1.

The signal is used for timing.

Table 17-2 LPTIM Internal Clock Sources

Register LPTIMx_CFGR[26:25]	Clock source
00	TIM_CLK (derived from HB_CLK)
01	HSI_CLK
10	LSE_CLK
11	LSI_CLK

When using an external clock source for counting, LPTIM may operate in one of the following two configurations:

The first configuration involves an external signal providing the clock to the LPTIM, while the internal clock signal is simultaneously supplied by a configurable internal clock source (HB, LSI).

Provided by LSE, HSI, etc.

The second configuration is that LPTIM is powered only by an external clock source through its external input channel 1. This configuration, upon entering low-power mode, allows all internal clock signals to be transmitted via an external clock source.

When the local clock source is off, it implements a pulse counter function or a timeout function.

Programming the CKSEL and COUNTMODE bits controls whether the LPTIM clock uses an external or internal clock source. When paired with...

When set to select an external clock source, the CKPOL bit is used to select the valid edge of the external clock signal. If the valid edge is set to rising edge and falling edge...

For a falling edge (double-edge), an internal clock signal must also be provided. In this case, the internal clock signal frequency should be higher than the external clock signal frequency.

The frequency is four times higher.

17.3.2 Filter LPTIM Input:

Whether external or internal, the LPTIM input is protected by a digital filter to prevent interference from any glitches and noise.

Disturbances propagate within LPTIM, thereby preventing accidental counts or triggers.

Before activating the digital filter, an internal clock source should be provided to LPTIM to ensure the filter operates normally.

The digital filters are divided into two groups:

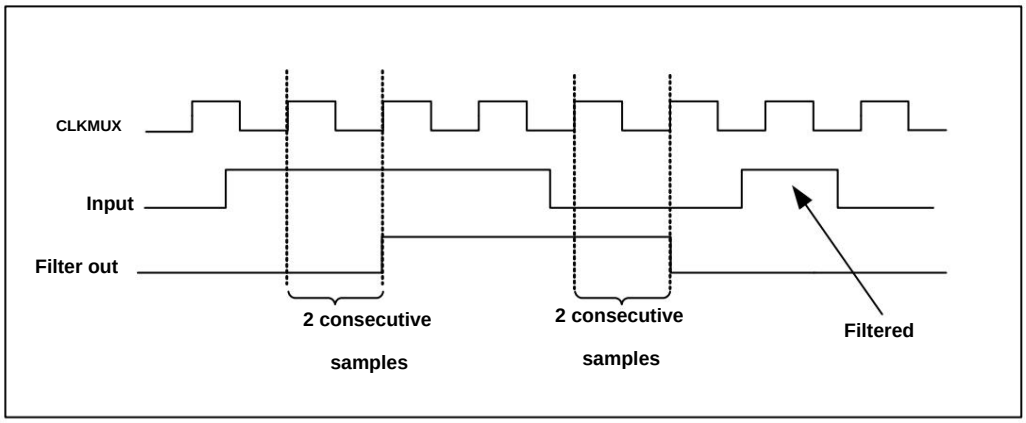
One set of digital filters protects the LPTIM external input, with the sensitivity of the digital filters controlled by the CKFLT bit; the other set of digital filters...

The device protects the internal trigger input of the LPTIM, and the sensitivity of the digital filter is controlled by the TRGFLT bit.

The sensitivity of the filter affects the number of consecutive samples detected on one of the LPTIM inputs.

Only when a signal level change is considered a valid switch can it be considered valid. The figure below shows an example of the interference filter behavior when programmed for two consecutive samples.

Figure 17-2 Timing block diagram of interference filter



Note: When the internal clock signal is not in use, the digital filter must be deactivated by clearing the CKFLT and TRGFLT bits. Currently, an external analog filter is used to avoid interference from external LPTIM inputs.

17.3.3 A configurable 2n

prescaler must precede the LPTIM 16-bit counter. The prescaler is controlled by the PRESC[2:0] 3-bit field.

Frequency division ratios, Table 17-3 lists all possible frequency division ratios.

Table 17-3 Division Ratio of Prescalers

PRESC[2:0]	Frequency division ratio
000	1
001	1/2
010	1/4
011	1/8
100	1/16
101	1/32
110	1/64
111	1/128

17.3.4 LPTIM Counter, a trigger multiplexer,

has two startup methods: one is software startup, and the other is startup upon detecting more than one valid edge in the trigger input.

Start after the trailing edge. The LPTIM trigger mode is controlled by TRIGEN[1:0], and the trigger source is controlled by the TRIGEL[1:0] bits.

Table 17-4 Triggering Methods

TRIGEN[1:0]	Triggering method
00	invalid
01	rising edge
10	Falling edge
11	bilateral

Table 17-5 Trigger Sources

TRIGSEL[1:0]	Trigger source
00	LPTIMx_ETR

01	RTC_ALARM is
10	invalid.
11	invalid

17.3.5 Operating Modes LPTIM

has two operating modes:

Continuous mode: The timer runs freely, starting from the trigger event and stopping only when the timer is disabled.

Single-trigger mode: The timer starts when a trigger event is triggered and stops when the ARR value is reached.

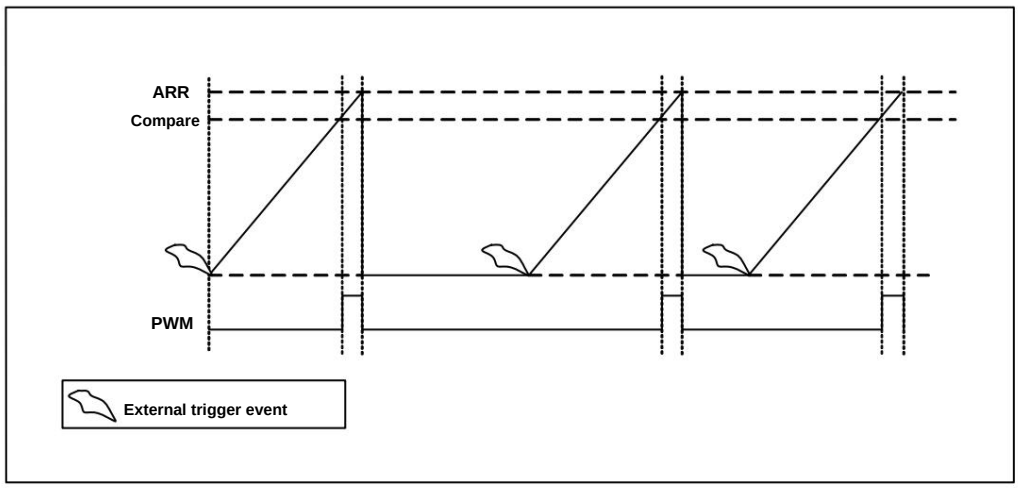
In single-trigger mode, to enable single-counting, the SNGSTRT bit must be set to 1, and a new trigger event will restart the timer.

Any triggering events that occur after the counter starts but before the counter reaches its ARR will be lost.

When using external triggers, each external trigger event that occurs after the SNGSTRT bit is set and after the counter register stops will trigger.

The device will start the counter for a new counting cycle.

Figure 17-3 LPTIM output waveform in single-count mode configuration

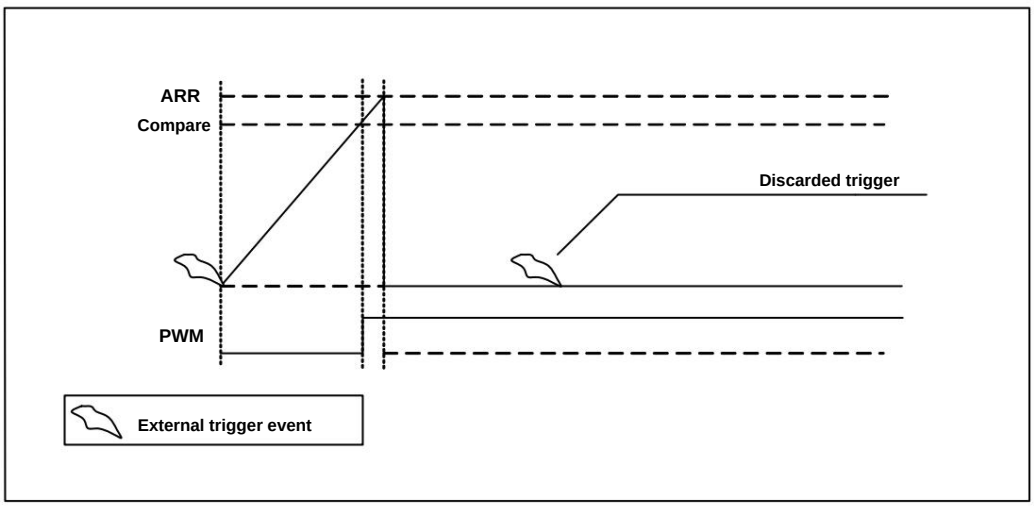


When activating single-trigger configuration mode, it should be noted that in single-trigger mode, when the WAVE bit in the LPTIMx_CFGR register...

When a segment is set, the one-time setting mode is activated. In this case, the counter only starts once after the first trigger, and any subsequent triggers will not start.

All triggered events were discarded.

Figure 17-4 LPTIM output waveform when single-count mode is configured and single-set mode is activated.

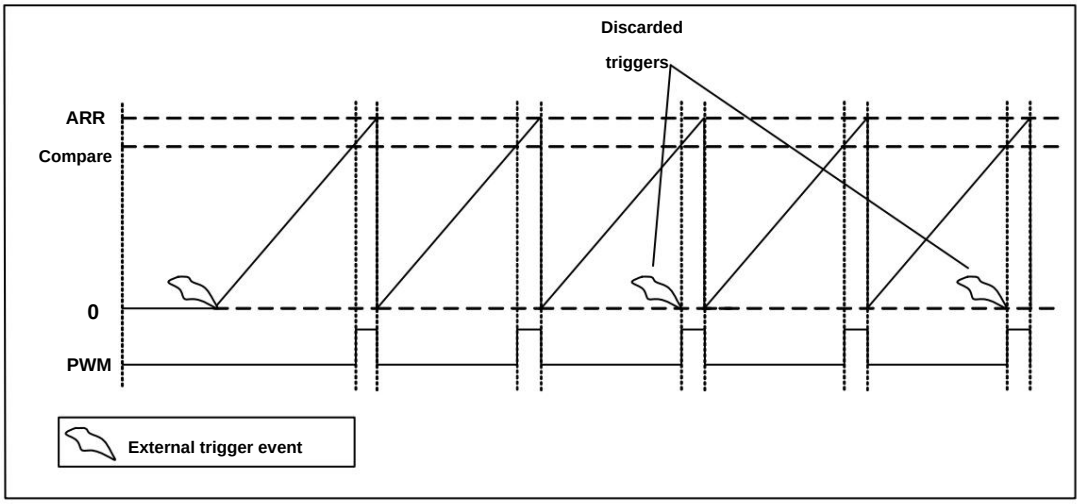


In continuous mode, to enable continuous counting, the CNTSTRT bit must be set. If an external trigger is selected, then the CNTSTRT bit must be set.

Subsequent external trigger events will start a counter for continuous counting. Any subsequent external trigger events will be discarded in the software.

When the device is running (TRIGEN=00), setting CNTSTRT will cause the startup counter to count continuously.

Figure 17-5 LPTIM output waveform in continuous counting mode configuration



The SNGSTRT and CNTSTRT bits can only be set when the timer is enabled (ENABLE bit set to 1). This changes the LPTIM counter mode. If continuous mode was previously selected, setting SNGSTRT will switch LPTIM to single-trigger mode, stopping the counter when it reaches the ARR value. If single-trigger mode was previously selected, setting CNTSTRT will switch LPTIM to continuous mode. The counter restarts immediately when it reaches the ARR value.

17.3.6 Timeout Function: Detecting

a valid edge on a selected trigger input can reset the counter. This function is controlled by the TIMOUT bit. The first trigger event starts the counter; any subsequent trigger events will reset the counter and restart the timer, enabling a low-power timeout function. If no trigger event occurs, the MCU is woken up by a compare-match event.

17.3.7 Waveform Generation Two

16-bit registers, LPTIMx_ARR and LPTIMx_CMP, are used to generate several different waveforms on the LPTIM output. The timer can generate the following waveforms: (1) PWM mode: The LPTIM output

is set once the counter value in LPTIMx_CNT exceeds the comparison value in LPTIMx_CMP. The LPTIM output is reset once the values in the LPTIMx_ARR and LPTIMx_CNT registers are equal. (2) Single pulse mode: The output waveform is similar to the PWM mode of the first pulse, and then

permanently resets. (3) One-time setting mode: The output waveform is similar to the single pulse mode, except that the output remains at

the last signal level (depending on the output).

(Configuration polarity).

The above mode requires that the value of the LPTIMx_ARR register be strictly greater than the value of the LPTIMx_CMP register.

The LPTIM output waveform can be configured via the WAVE bit as follows:

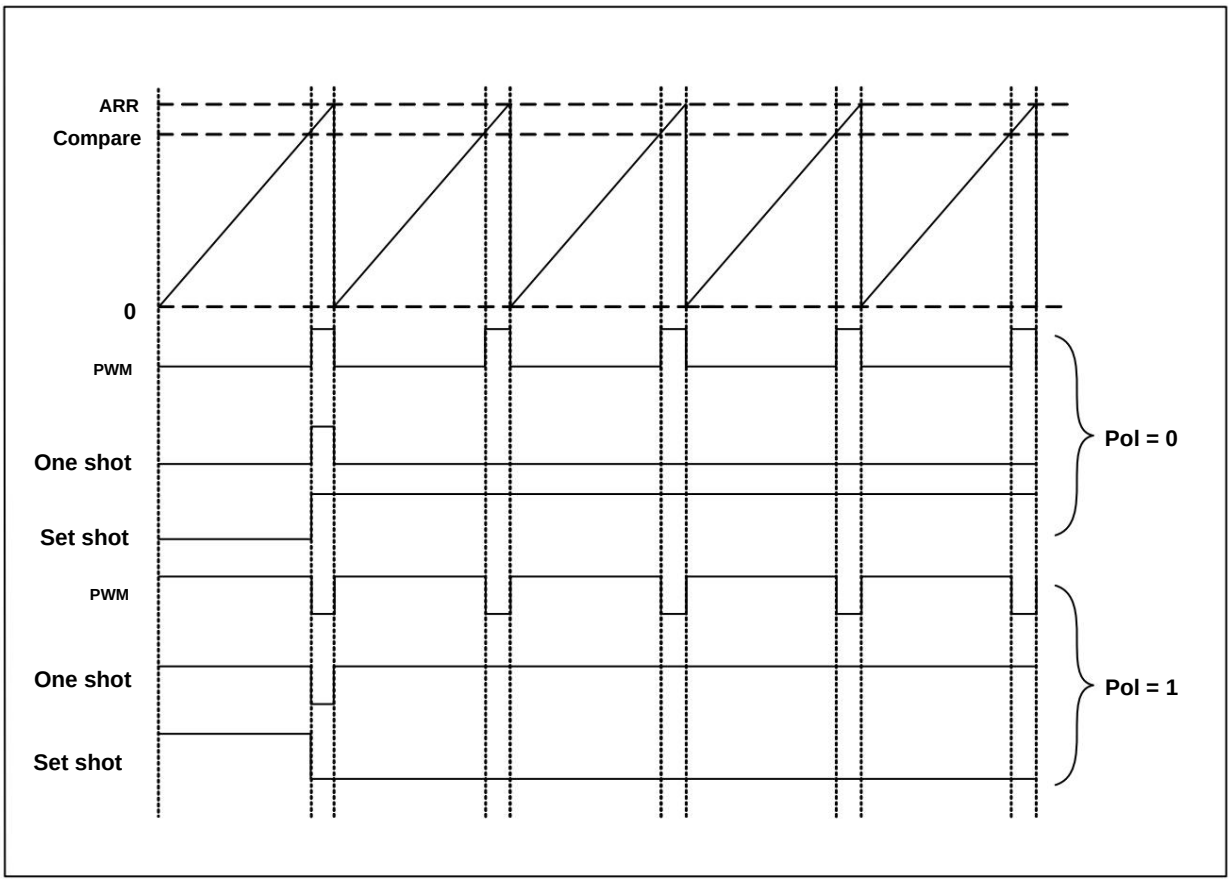
(1) Resetting the WAVE bit to 0 will force LPTIM to generate a PWM waveform or a single pulse waveform, depending on the bit set: CNTSTRT Or SNGSTRT. (2)

Setting the WAVE bit to 1 will force LPTIM to generate a setup mode once.

The WAVPOL bit controls the output polarity of LPTIM. Changes take effect immediately, so even after reconfiguring the polarity, or after enabling the timer, changes will not take effect. Before that, the output default value will also be changed immediately.

The generated signal has a frequency up to twice the LPTIM clock frequency. Figure 17-6 shows three possible frequencies that can be generated on the LPTIM output. Waveform. Furthermore, this figure also shows the effect of changing the polarity via the WAVPOL bit.

Figure 17-6 Generated Waveform



17.3.8 Register Updates The

LPTIMx_ARR and LPTIMx_CMP registers are updated immediately after an HB bus write operation, or if the timer has been updated.

Once started, it will be updated at the end of the current cycle.

The PRELOAD bit controls how the LPTIMx_ARR and LPTIMx_CMP registers are updated:

When the PRELOAD bit is set to 0, the LPTIMx_ARR and LPTIMx_CMP registers are updated immediately after any write access. When the PRELOAD bit is set to 1, the LPTIMx_ARR and LPTIMx_CMP registers are updated at the end of the current cycle if the timer has already started.

The LPTIM HB interface and LPTIM logic use different clocks, therefore there is a time lag between HB writes and the availability of these values to the counter comparator. There is a delay, during which any additional writes to these registers must be avoided.

The ARROK and CMPOK flags in the LPTIMx_ISR register indicate when the operation of the LPTIMx_ARR register is complete.

Write operations to the LPTIMx_CMP register.

After writing to the LPTIMx_ARR register or the LPTIMx_CMP register, the status will only be updated after the previous write operation is completed.

It can perform a new write operation on the same register.

Any consecutive writes before setting the ARROK or CMPOK flags will result in unpredictable results.

17.3.9 Counter Mode LPTIM The

counter can be used to count external events on LPTIM_CH1 or to count internal clock cycles.

The CKSEL and COUNTMODE bits control which source will be used to update the

counter. If LPTIM is configured to count external events on LPTIM_CH1, the count can be updated after a rising edge, falling edge, or both edges, depending on the value written to the CKPOL[1:0] bits. The following counting

modes can be selected based on the CKSEL and COUNTMODE values:

(1) CKSEL=0: LPTIM clock is counted by internal clock source.

When COUNTMODE=0, LPTIM is configured to be counted by an internal clock source, and the LPTIM counter is configured to follow each internal clock pulse.

Updated after the initial rush.

COUNTMODE=1 uses the internal clock provided to LPTIM to sample the data on LPTIM_CH1. Therefore, to ensure no data is missed...

Events: The frequency of signal changes on LPTIM_CH1 should not exceed the frequency of the LPTIM internal clock.

(2) CKSEL=1: When LPTIM is counted by an external clock source, the value of COUNTMODE is unaffected.

In this configuration, LPTIM does not require an internal clock source (unless the digital filter is enabled), and the signal injected on LPTIM_CH1 is used as LPTIM.

The system clock. This configuration applies to operating modes where the built-in oscillator is not enabled.

For this configuration, the LPTIM counter can be updated on either the rising or falling edge of the LPTIM_CH1 clock signal, but not on both edges.

Since the signal injected in LPTIM_CH1 is also used for LPTIM counting, there will be some initial delay before the counter increments (at startup).

(After enabling LPTIM). To be precise, after enabling LPTIM, the first five valid edges of LPTIM_CH1 outside LPTIM will be lost.

17.3.10 Timer Enable The ENABLE

bit in the LPTIMx_CR register is used to enable/disable the LPTIM logic. After setting the ENABLE bit, in the actual...

Two counter clock delays are required before LPTIM can be enabled.

17.3.11 Encoder Mode This mode

allows processing of signals from a quadrature encoder used to detect the position of rotating elements. The encoder interface mode is simply a...

An external clock with direction selection. This means the counter only counts continuously between 0 and the auto-reload value in the LPTIMx_ARR register.

Therefore, LPTIMx_ARR must be configured before startup to generate data based on two external input signals, LPTIMx_CH1 and LPTIMx_CH2.

The clock signal is used to time the LPTIM counter. The phase between these two signals determines the counting direction.

Encoder mode is only available when LPTIM is counted by an internal clock source. The signal frequencies on LPTIMx_CH1 and LPTIMx_CH2...

The frequency must not exceed 4 times the frequency of the LPTIM's internal clock. Meeting these conditions is essential to ensure the LPTIM operates normally.

Direction changes are signaled by two up and down flags in the LPTIMx_ISR register. Additionally, if enabled via the DOWNIE bit,

This allows for the generation of interrupts for events that change direction in both directions.

To activate encoder mode, the ENC bit must be set to 1, and LPTIM must first be configured to continuous mode.

When encoder mode is active, the LPTIM counter automatically modifies based on the speed and direction of the incremental encoder. Therefore, it...

The content always represents the encoder's position. The counting direction is indicated by up and down indicators, corresponding to the direction of rotation of the encoder rotor.

Based on the edge sensitivity configured in CKPOL[1:0] bits, the following possible combinations were obtained, where LPTIMx_CH1 and LPTIMx_CH2 are different.

Switching between time periods.

Table 17-6 Configuration of Edge Sensitivity Combinations

active edge (CKPOL[1:0])	Opposite signal electrical flat (LPTIMx_CH1) The corresponding input is LPTIMx_CH2, LPTIMx_CH2 The corresponding input is LPTIMx_CH1)	LPTIMx_CH1		LPTIMx_CH2	
		rise	decline	rise	decline
Rising edge (00)		Decreasing count not counted; increasing count not counted.			
	High	Increasing count not counted, decreasing count not counted.			
Falling edge (01)	low	Uncounted incrementing, uncounted decrementing			
	high	Uncounted decreasing Uncounted increasing			
Double edge (10)	low	Decreasing Increasing Increasing Decreasing			
	high low	Increasing Decreasing Decreasing Increasing			

11	invalid

17.3.12 Debug Mode When the

MCU enters debug mode (kernel stops), LPTIM continues to work normally.

17.3.13 LPTIM Low Power Mode

Table 17-7 Impact of Low Power Mode on LPTIM

Sleep	describe
mode	No impact; an LPTIM interrupt will cause the device to exit sleep mode.
Stop mode	LPTIM peripherals are active when timed by LSE or LSI. The interruption caused the device to exit stop mode.

17.4 LPTIM Interruption

An interrupt/wake-up will occur if the following events are enabled via the LPTIMx_IER register:

Comparison and matching

Automatic reload matching

External trigger events

Automatic reload register write complete

Comparison register write complete

ÿ Direction change (encoder mode)

Table 17-8 Interruption Events

Interruption event	describe
Comparison Matching	When the value of the counter register LPTIMx_CNT is compared with the value of the compare register LPTIMx_CMP An interrupt flag will be generated when the values are equal.
Automatic reload matching	When the value of the counter register LPTIMx_CNT is different from that of the auto-reload register An interrupt flag is generated when the values of LPTIMx_ARR are equal.
External trigger events	An interrupt flag is generated when an external trigger event is detected.
An interrupt flag is generated when the automatic reload register write operation to the LPTIMx_ARR register is complete.	
An interrupt flag is generated when the write operation to the LPTIMx_CMP register is complete.	
Counting direction changed	When used in encoder mode, two interrupt flags are embedded in the signal direction change. Change: Upward indicator: indicates a change in the counting direction; Downward indicator: Indicates a change in the counting direction.

17.5 Register Description

Table 17-9 List of LPTIM1 Related Registers

name	Access address	describe	Reset value
R32_LPTIM1_ISR	0x40002400	LPTIM1 Interrupt Status Register	0x00000000
R32_LPTIM1_ICR	0x40002404	LPTIM1 Interrupt Clear Register	0x00000000
R32_LPTIM1_IER	0x40002408	LPTIM1 Interrupt Enable Register	0x00000000
R32_LPTIM1_CFGR	0x4000240C	LPTIM1 Configuration Register	0x00000000
R32_LPTIM1_CR	0x40002410	LPTIM1 Control Register	0x00000000

R16_LPTIM1_CMR	0x40002414	LPTIM1 Comparator Register	0x00000000
R16_LPTIM1_ARR	0x40002418	LPTIM1 Automatic Reload Register	0x00000001
R16_LPTIM1_CNT	0x4000241C	LPTIM1 Count Register	0x00000000

Table 17-10 List of LPTIM2 Related Registers

name	Access address	describe	Reset value
R32_LPTIM2_ISR	0x40003400	LPTIM2 Interrupt Status Register	0x00000000
R32_LPTIM2_ICR	0x40003404	LPTIM2 Interrupt Clear Register	0x00000000
R32_LPTIM2_IER	0x40003408	LPTIM2 Interrupt Enable Register	0x00000000
R32_LPTIM2_CFGR	0x4000340C	LPTIM2 Configuration Register	0x00000000
R32_LPTIM2_CR	0x40003410	LPTIM2 control register	0x00000000
R16_LPTIM2_CMR	0x40003414	LPTIM2 Comparator Register	0x00000000
R16_LPTIM2_ARR	0x40003418	LPTIM2 Auto Reload Register	0x00000001
R16_LPTIM2_CNT	0x4000341C	LPTIM2 Count Register	0x00000000

17.5.1 Interrupt Status Register (LPTIMx_ISR) (x=1/2)

Offset address: 0x00

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								DIR_SYNC	DOWN	UP	ARROK	CMPOK	EXTTRIG	ARRM	CMPM

Bit	Name access		describe	Reset value
[31:8] Reserved		RO is reserved.		0
7	DIR_SYNC	RO	In encoder mode, the counter counting direction is: 1: Count down; 0: Count upwards. This bit is only valid in encoder mode, i.e., ENC bit is set to 1. It must be set before use. The counting direction function needs to be enabled by setting DIR_EXTEN to 1.	0
6	DOWN	RO	Count down: 1: The counting direction is from top to bottom; 0: Invalid. In encoder mode, this bit is set by hardware to notify the application. The direction of the counter changes from top to bottom. After reaching 1, it can be used to... Write 1 to the LPTIMx_ICR[6] bit to clear it.	0
5	UP	RO	Counting upwards: 1: The counting direction is from bottom to top; 0: Invalid. In encoder mode, this bit is set by hardware to notify the application. The direction of the counter changes from bottom to top. After reaching 1, it can be adjusted by moving... Write 1 to the LPTIMx_ICR[5] bit to clear it.	0
4	ARROK	RO automatic reload register data update successful:		0

			1: Data update successful; 0: Invalid. This bit is set by the hardware to notify the application of the HB bus pair. The write operation to LPTIMx_ARR has completed successfully, and the value is now 1. After that, you can proceed... Clear by writing 1 to the LPTIMx_ICR[4] bit.	
3	CMPOK	RO	Compare register data update successful: 1: Data update successful; 0: Invalid. This bit is set by the hardware to notify the application of the HB bus pair. The write operation to LPTIMx_CMR has completed successfully, and the value is now 1. You can then proceed... Clear by writing 1 to the LPTIMx_ICR[3] bit.	0
2	EXTTRIG	RO	Externally triggered edge events: 1: Valid edge input has occurred; 0: Invalid. This bit is set by the hardware to notify the application on the selected external trigger. A valid edge input has occurred, if due to the timer already started... If the trigger is ignored, this flag is not set. Setting it to 1 allows subsequent actions to proceed. Write 1 to the LPTIMx_ICR[2] bit to clear it.	0
1	ARRM	RO	Automatic reload register data matches the data in the LPTIMx_CNT register. match: 1: Match successful; 0: Invalid. This bit is set by hardware to notify the application of the LPTIMx_CNT register. After the value reaches the value of the LPTIMx_ARR register, which is 1, it can be used... Write 1 to the LPTIMx_ICR[1] bit to clear it.	0
0	CMPM	RO	Compare the data in the comparison register with the data in the LPTIMx_CNT register: 1: Match successful; 0: Invalid. This bit is set by hardware to notify the application of the LPTIMx_CNT register. After the value reaches the value of the LPTIMx_CMR register, which is 1, it can be used... Write 1 to bit LPTIMx_ICR[0] to clear it.	0

17.5.2 Interrupt Clear Register (LPTIMx_ICR) (x=1/2)

Offset address: 0x04

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
Reserved					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									DOWNCF	UPNCF	ARROKNCF	CMPOKNCF	EXTTRIGNCF	ARRMNCF	CMPMNCF

Bit	Name access		describe	Reset value
[31:7] Reserved		RO is reserved.		0
6	DOWNCF	RW1 clears the down flag.		x

			Note: Writing 0 is invalid; writing 1 clears the corresponding bit in the status register.	
5	UPCF	RW1	Clear the up flag. Note: Write 0 Invalid. Write the corresponding bit to the status register. 1 Clear 0	x
4	ARROKCF	RW1	Clear the automatic reload data update flag. Note: Writing 0 is invalid; writing 1 clears the corresponding bit in the status register.	x
3	CMPOKCF	RW1	Clear the comparator data update flag. Note: Writing 0 is invalid; writing 1 clears the corresponding bit in the status register.	x
2	EXTTRIGCF	RW1	Clear the edge event flag of the external trigger. Note: Write 0 Invalid. Write the corresponding bit to the status register. 1 Clear 0	x
1	ARRMCF	RW1	Clear the auto-reload register match flag. Note: Writing 0 is invalid; writing 1 clears the corresponding bit in the status register.	x
0	CMPMCF	RW1	Clear the match flag in the compare register. Note: Writing 0 is invalid; writing 1 clears the corresponding bit in the status register.	x

17.5.3 Interrupt Enable Register (LPTIMx_IER) (x=1/2)

Offset address: 0x08

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

15 14 13 12 11 10 9 8 7	6	5	4	3	2	1	0
Reserved	DOWNIE	UPIE	ARROKIE	CMPOKIE	EXTTRIGIE	ARRMIE	CMPMIE

Bit	Name access		describe	Reset value
[31:7] Reserved		RO reserved.		0
6	DOWNIE	RW	Downward interrupt enabled: 1: Turn on; 0: Off.	0
5	UPIE	RW	Upward interrupt enable: 1: Turn on; 0: Off.	0
4	ARROKIE	RW	Automatic reload register data update successful interrupt enabled: 1: Turn on; 0: Off.	0
3	CMPOKIE	RW	Compare register data update successful interrupt enabled: 1: Turn on; 0: Off.	0
2	EXTTRIGIE	RW	External trigger edge event input interrupt enable: 1: Turn on; 0: Off.	0
1	ARRMIE	RW	Automatic reload register data matching successful interrupt enabled: 1: Turn on; 0: Off.	0
0	CMPMIE	RW	Compare register data match successful interrupt enabled: 1: Turn on;	0

			0: Off.	

17.5.4 Configuration Register (LPTIMx_CFGR) (x=1/2) Offset Address:

0x0C

31	30 29	28	27 26 25 24 23 22 21	2019	18	17	16
Reserved			FORCE_PWM	CLKMX_SEL ENC	COUNT MODE	PRELOAD	WAVPOL	WAVE	TIMOUT	TRIGEN	Reserved

15	14 13	12	11 10 9	8	7 6	5	4 3	2 1	0
Reserved	TRIGSEL	Reserved	PRESC	Reserved	TRGFLT	Reserved	CKFLT	CKPOL CKSEL	

Bit	Name access		describe	Reset value
[31:28] Reserved		RO reserved.		0
27	FORCE_PWM	RW	Forced PWM output: 1: Force PWM output to high level; 0: Invalid.	0
[26:25] CLKMX_SEL[1:0]		RW	LPTIM counter internal clock source selection: 00: TIM_CLK (derived from HB_CLK); 01: HSI_CLK; 10: LSE_CLK; 11: LSI_CLK.	0
twenty four	ENC	RW	Encoder mode enabled: 1: Turn on; 0: Off.	0
twenty three	COUNTMODE	RW	Counter mode, select LPTIM to use which clock source for the counter. hour: 1: The counter increments after each valid pulse is input to the LPTIM external input; 0: The counter increments after each internal clock pulse.	0
twenty two	PRELOAD	RW	Register update mode controls LPTIMx_ARR and LPTIMx_CMP registers. Memory update mode: 1: The register is updated at the end of the current LPTIM cycle; 0: Update the register after each HB bus write access.	0
twenty one	WAVPOL	RW	PWM waveform polarity: 1: The output reflects the ratio between the LPTIMx_ARR and LPTIMx_CMP registers. Invert the result; 0: The output reflects the ratio between the LPTIMx_ARR and LPTIMx_CMP registers. Compare the results. Note: If the counter value is greater than the comparator value, the comparison result is 1; otherwise, it is 1. 0.	0
20	WAVE	RW	PWM waveform: 1: Set to single-pulse mode (one pulse waveform); 0: Turn off single-use mode.	0
19	TIMOUT	RW	Enables timeout function: 1: Trigger events that arrive when the timer has already started will be reset and restarted.	0

			counter; 0: Trigger events that arrive when the timer has already started are ignored.	
[18:17] TRIGEN[1:0]		RW	Trigger enable and polarity, controls whether the LPTIM counter is triggered externally. When the device starts, if the external trigger option is selected, the trigger active edge... There are three configurations for the relationship. 00: Software triggered (counter startup is initiated by software); 01: Triggered by rising edge; 10: Triggered by falling edge; 11: Double-edge triggering.	0
[16:15] Reserved		RO reserved.		0
[14:13] TRIGSEL[1:0]		RW	Trigger source selection: 00: LPTIM_ETR; 01: RTC_ALARM; 10: Invalid; 11: Invalid.	0
12	Reserved	RO Retention		0
[11:9] PRESC[2:0]		RW	Clock prescaler, configure the prescaler coefficient: 000:1 frequency division; 001: 2 frequency division; 010: 4 frequency division; 011: 8 frequency division; 100:16 frequency division; 101:32 frequency division; 110:64 frequency division; 111:128 frequency division.	0
8	Reserved	RO is reserved.		0
[7:6] TRGFLT[1:0]		RW	Configurable digital filter for the trigger: The TRGFLT value is set to be detected when a change occurs on the internal trigger. The number of consecutive equal samples is then considered as a valid level transition. An internal clock source is required to use this feature. 00: Any change to the trigger is considered a valid trigger; 01: The trigger activation level change must be stable for at least 2 clock cycles. Only then is it considered a valid trigger; 10: The trigger activation level change must be stable for at least 4 clock cycles. Only then is it considered a valid trigger; 11: The trigger activation level change must be stable for at least 8 clock cycles. Only then is it considered a valid trigger.	0
5	Reserved	RO reserved.		0
[4:3] CKFLT[1:0]		RW	Configurable digital filter for external clock: The CKFLT value is used to set the number of consecutive equal samples. When the external clock signal... When a signal level changes, such continuous samples should be detected before this signal is considered. Level changes are considered valid level transitions. An internal clock source is required. Use this feature. 00: Any change in the level of an external clock signal is considered a valid transition; 01: The external clock signal level must remain stable for at least two clock cycles.	0

			It is only considered valid after a certain period; 10: The external clock signal level must remain stable for at least 4 clock cycles. It is only considered valid after a certain period; 11: The external clock signal level must remain stable for at least 8 clock cycles. It is only considered valid after a certain period.	
[2:1] CKPOL[1:0]		RW	If LPTIM selects an external clock source for counting, the CKPOL bit is used to configure... Effective edge: 00: Rising edge used for counting; 01: Falling edge is used for counting; 10: Double-edge counting is used when both edges of the external clock signal are active. At the same time, LPTIM must also be powered by a frequency at least four times that of the external clock frequency. The timing is done using an internal clock source; 11: Invalid. If LPTIM is configured in encoder mode (ENC position 1), CKPOL The bit is used to select the encoder mode: 00: Encoder mode 1 activated; 01: Encoder Mode 2 activated; 10: Encoder Mode 3 activated; 11: Invalid.	0
0	CKSEL	RW	Clock selector, used to select the clock used by LPTIM: 1: LPTIM uses an external clock; 0: LPTIM uses an internal clock.	0

17.5.5 Control Register (LPTIMx_CR) (x=1/2) Offset Address: 0x10

31 30 29	28	27 26 25 24 23 22 21	2019	18	17	16
Reserved						

15 14 13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									DIR_EXTEN	OUTEN	CNTSTRT	SNGSTRT	ENABLE

Bit	Name access		describe	Reset value
[31:5] Reserved		RO reserved.		0
4	DIR_EXTEN	RW	External trigger count direction enabled: 1: Turn on; 0: Off. Note: Use after encoder mode is enabled.	0
3	OUTEN	RW	PWM output enable: 1: Turn on; 0: Off. Note: In non-encoder mode, this bit enables the output waveform.PWM ENABLE is only effective when both are enabled.	0
2	CNTSTRT	RW	Start in continuous mode: This bit is set by software and cleared by hardware, when the software is started.	0

			(TRIGEN==00), setting this bit enables LPTIM to start in continuous mode, such as... If the software startup is disabled (TRIGEN!=00), setting this bit will be used during detection. LPTIM starts in continuous mode immediately after the external trigger is reached. If in Setting this bit when LPTIM is in single-pulse counting mode will cause LPTIM to... The counter will not stop counting when it reaches the ARR value. Note: You can only write when ENABLE=1.	
1	SNGSTRT	RW	LPTIM starts in single-trigger mode: This bit is set by software and cleared by hardware, when the software is started. (TRIGEN==00) This bit sets LPTIM to start in single-pulse mode. If software startup is disabled (TRIGEN != 00), setting this bit will be checked. Upon detection of the external trigger, LPTIM is immediately started in single-pulse mode. If Setting this bit when LPTIM is in continuous counting mode will cause LPTIM to... Stop when the counter reaches the ARR value. Note: You can only write when ENABLE=1.	0
0	ENABLE	RW	Timer enabled; when the timer enable bit is cleared to 0, the internal timer will be reset. The logic is limited, and the LPT_CR[2:1] bits cannot be manipulated. 1: Enable low-power timer; 0: Low-power timer is disabled and internal logic is reset.	0

17.6.6 Comparison Register (LPTIMx_CMP) (x=1/2) Offset Address: 0x14

Bit	Name access		The	Reset value
[15:0] CMP		RW	timer describes the comparison value; when the counter value equals the value in the compare register, the timer... The comparison flag will be set to 1 if the corresponding enable signal is enabled before it is set to 1. The signal will generate an interrupt source and a low-power wake-up signal.	0

17.5.7 Automatic Reload Register (LPTIMx_ARR) (x=1/2)

Offset address: 0x18

Bit	Name access		The	Reset value
[15:0] ARR		RW	count value described by the reloaded count register, when counting upwards if the count... If the value equals the reloaded count value, the counter starts counting from 0; If the counter value is 0 when counting down, then start counting from the reloaded counter. The count begins.	1

17.5.8 Counter Register (LPTIMx_CNT) (x=1/2) Offset Address: 0x1C

Bit	Name access			Reset value
[15:0] COUNT		RO	Describes the current count value of the timer counter register: When LPTIM is running using an asynchronous clock, read the LPTIMx_CNT register. The instrument may return unreliable values, therefore, in such cases, it is necessary to... Perform two consecutive read accesses and verify that the two return values are the same. Same. It should be noted that for reliable LPTIMx_CNT register reads... For a single read operation, two consecutive reads must be performed and compared.	0

Chapter 18 Digital-to-Analog Conversion (DAC)

The digital-to-analog converter (DAC) module contains two 12-bit voltage-output digital-to-analog converters (DACs), converting two digital signals.

The signal is a two-channel analog voltage signal output, supporting independent or synchronous conversion of dual DAC channels, and supporting 12-bit data left-aligned or right-aligned.

Supports 12-bit or 8-bit data and external event-triggered conversion. Can generate triangular waves and noise. Supports DMA functionality.

18.1 Key Features

Two DAC converters, each with one output channel.

Triangular wave and noise waveform generator

ÿConfigurable 8-bit or 12-bit output

ÿ 12-bit data left-aligned or right-aligned

Dual DACs converting simultaneously or separately

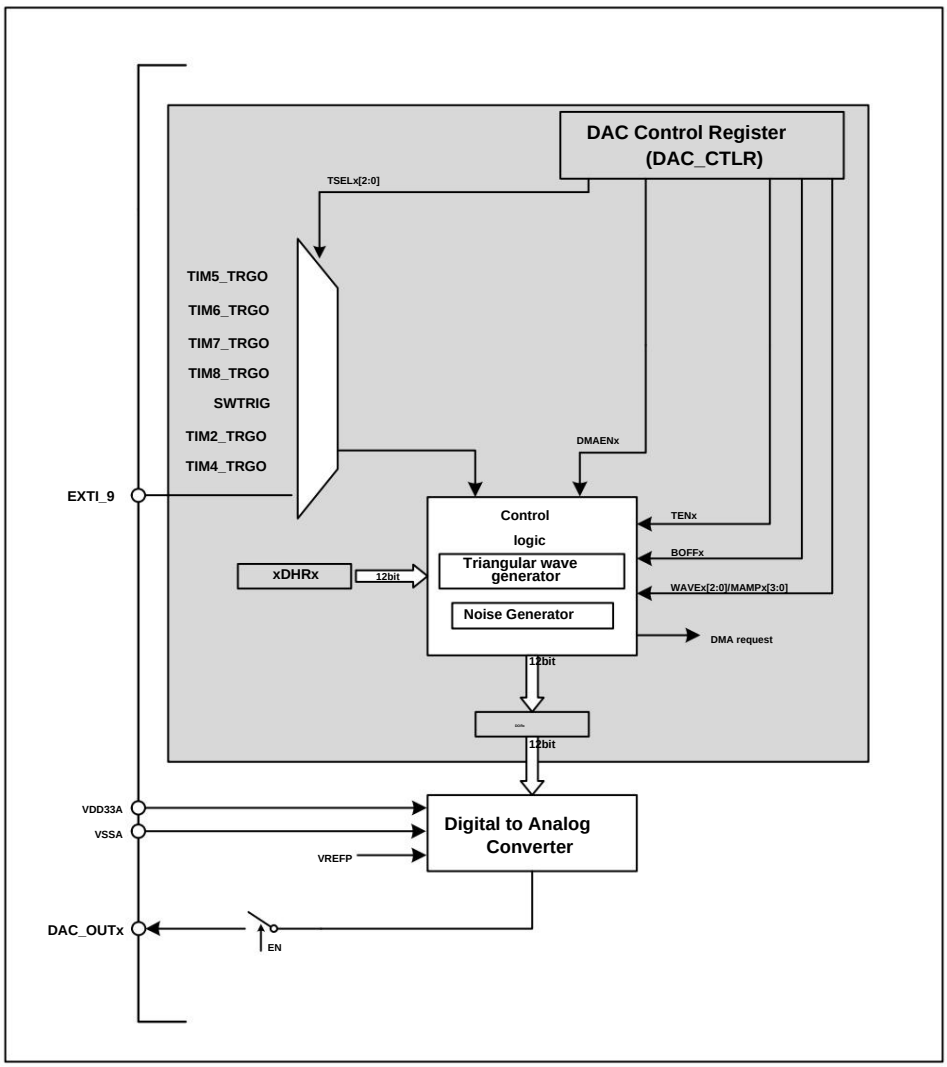
Supports DMA functionality

Multiple triggering events

18.2 Functional Description

18.2.1 DAC Module Structure

Figure 18-1 DAC Module Block Diagram



18.2.2 DAC Channel

Configuration 18.2.2.1 Enabling DAC Function:

Setting the ENx bit in the DAC_CTLR register to 1 enables analog power to DAC channel x. After a startup period, the DAC...

Channel x is now enabled. The DAC contains two analog output channels, which can output simultaneously or independently.

Note: To avoid parasitic interference and additional power consumption, the pins corresponding to the DAC channels should be set to analog input (AIN) mode in advance.

18.2.2.2 Enable output buffering:

The DAC integrates an output buffer, which can be used to reduce output impedance and increase drive capability to directly drive external loads. Each DAC channel...

The output buffer can be enabled or disabled by setting the BOFFx bit of the DAC_CTLR register.

18.2.2.3 Data format: In single

DAC channel mode, it includes 8-bit data right-aligned, 12-bit data left-aligned, and 12-bit data right-aligned.

When 8-bit data is right-aligned, writing data to DAC_R8BDHRx[7:0] will load the left-shifted number (after 1 HB clock cycle).

According to the data output register DAC_DORx[11:4].

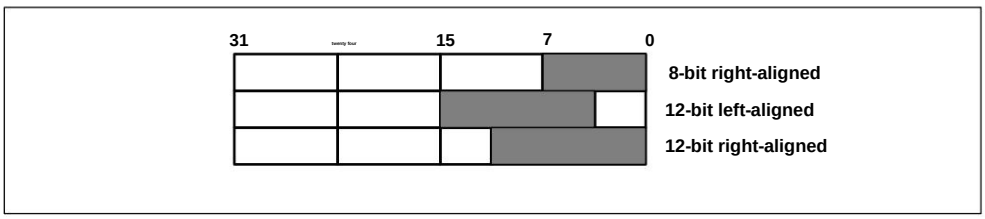
When 12-bit data is right-aligned, writing data to DAC_R12BDHRx[11:0] will load the right alignment (after 1 HB clock cycle).

All data is fed into the data output register DAC_DORx[11:0].

12. When the data is left-aligned, write the data to DAC_L12BDHRx[15:4]. After the module performs the corresponding shift, it will load (1 HB).

After clock cycle, left-align the data to the data output register DAC_DORx[11:0].

Figure 18-2 Single-channel data format



In dual DAC channel mode, there are also three modes: 8-bit data right alignment, 12-bit data left alignment, and 12-bit data right alignment.

When 8 bits of data are right-aligned, data is written to DAC_RD8BDHR[7:0], and the module will load bits [7:0] (after 1 HB clock cycle).

After shifting, bits [15:8] are moved to DAC_DOR1[11:4], and bits [15:8] are moved to DAC_DOR2[11:4].

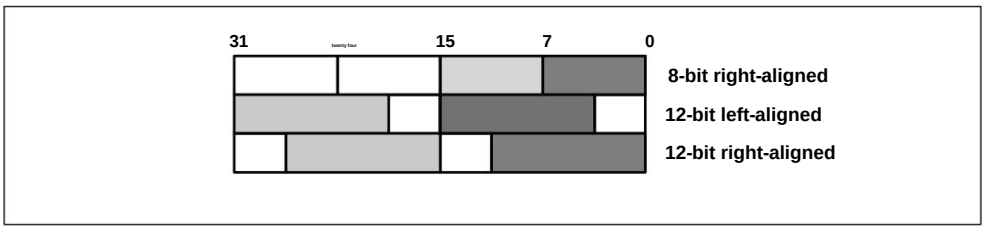
When data is left-aligned (12), write data to DAC_LD12BDHR[31:0], and the module will load bits [15:4] (after 1 HB clock cycle).

After the data is shifted to DAC_DOR1[11:0], the data in bits [31:20] is shifted to DAC_DOR2[11:0].

The 12-bit data is right-aligned and written to DAC_RD12BDHR[31:0]. The module will then load bits [11:0] (after 1 HB clock cycle).

Data is sent to DAC_DOR1[11:0], and bits [27:16] are sent to DAC_DOR2[11:0].

Figure 18-3 Dual-channel data format



18.2.2.4 DMA Function: The

DAC channel has DMA functionality. Setting the DMAENx bit of the DAC_CTLR register to 1 enables the DMA function for the corresponding channel. When

If a triggering event (excluding software triggers) occurs, a DMA request is generated, and the data in the DAC_DORx register will be updated.

18.2.2.5 Trigger Event Selection:

DAC conversion can be triggered by the following events: when the TENx bit of the DAC_CTLR register is set to 1, and TSELx[2:0] is configured.
The control bit selects a trigger event to trigger DAC conversion.

Table 18-1 Triggering Events

Trigger	type	TSELx[2:0]
source: Timer 6 TRGO event	Signals from the on-chip timer	000
Timer 8 TRGO Event Timer 7		001
TRGO Event Timer 5 TRGO		010
Event Timer 2 TRGO Event		011
Timer 4 TRGO Event		100
		101
EXTI Line 9	External pin	110
SWTRIG (Software Trigger)	software control bit	111

The DAC interface will detect the rising edge from the selected timer TRGO output or external interrupt line 9, 3 HB after triggering.
After a clock cycle, the DAC_DORx register is updated to the new value.

If software triggering is configured, once the SWTRIG bit is set to 1, a transition will be initiated, occurring one HB clock cycle after the trigger.
After the cycle, the DAC_DORx register is updated to the new value, and the hardware automatically clears the SWTRIG bit to 0.

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

18.2.3 DAC Conversion: Data

for the DAC channel comes from the DAC_DORx register, but data cannot be directly written to the DAC_DORx register. Any output to the DAC...
Data for channel x must be written to DAC_R12BDHR1, DAC_L12BDHR1, DAC_R12BDHR2, DAC_L12BDHR2, and DAC_RD12BDHR.
The values are stored in the DAC_LD12BDHR and DAC_RD8BDHR registers. The system's internal holding register DAC_DHRx retrieves these register values and stores them...
After a certain period of time, the data is sent to the DAC_DORx register.

In non-trigger mode, data written to register DAC_xDHRx will be moved into register DAC_DORx after 1 HB clock cycle.

When triggered by software, the DAC_DORx register is automatically updated one HB clock cycle after the rising edge of the event trigger.

Under hardware triggering (timer TRGO event or rising edge of external interrupt line 9), automatically after 3 HB clock cycles following the trigger event.
Update the DAC_DORx register.

After loading data into the DAC_DORx register, the output becomes valid after a time interval tSETTLING , the length of which depends on the power supply voltage.
The output load will vary depending on the analog output load.

The digital input is linearly converted to an analog voltage output by the DAC, ranging from 0 to VDD33A. The input on any DAC channel pin...
The output voltage satisfies the following relationship:

DAC output voltage = VDD33A * (DAC_DORx/4096).

18.2.4 DAC Triangle Wave Generator

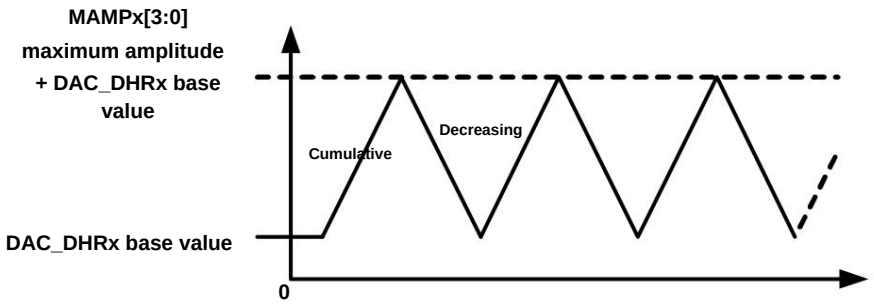
The module has a built-in triangular wave generator that can add a small-amplitude triangular wave to a reference signal. Set the WAVEx[1:0] bits.
Set the value to 10b to select the triangular wave generation function of the DAC. Set the MAMPx[3:0] bits of the DAC_CTLR register to select the amplitude of the triangular wave.

The system internally includes a triangular wave counter that starts at 0 and increments by 1 after 3 HB clock cycles following each triggered event.
The value of the counter is added to the value in the DAC_DHRx register, and after discarding the overflow bit, it is written to the DAC_DORx register. This is done before the value is passed to the DAC_DORx register.
When the value of the register is less than the maximum amplitude defined by bits MAMPx[3:0], the triangular wave counter increments gradually. Once the set maximum amplitude is reached...
When the value reaches 0, the counter starts to decrement, and after reaching 0, it starts to increment again, repeating this cycle. Setting WAVEx[1:0] to '00' will reset the triangle.
Wave generation.

Note: 1. To generate a triangular wave, it is necessary to enable [the function/equipment]. DAC Trigger, i.e., set the DAC_CTLR register. TENx 1. Position

2. The MAMPx[3:0] bits must be set before enabling DAC; otherwise, their values cannot be modified.

Figure 18-4 Generation of Triangular Waves



18.2.5 DAC Noise Generator

The module has a built-in noise generator that uses a linear feedback shift register (LFSR) to produce pseudo-noise with varying amplitude. Setting the WAVE[1:0] bits to 01b selects the DAC noise generation function. Setting the DAC_CTLR register...

Use the MAMPx[3:0] bits of the register to select the data of the LFSR that is masked.

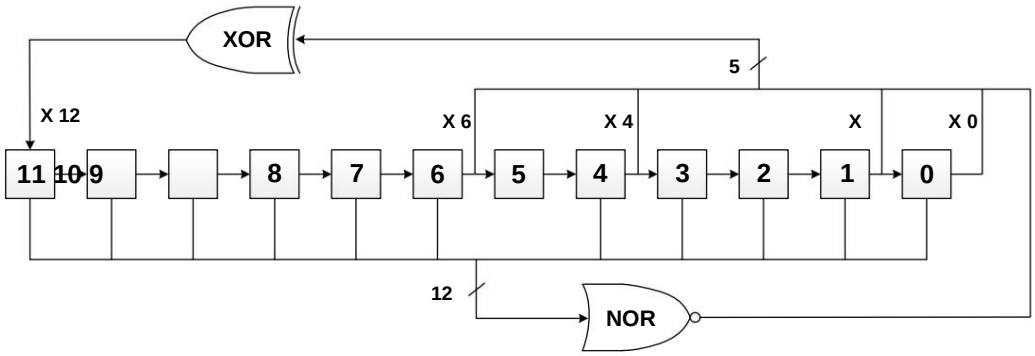
The LFSR register is preloaded with a value of 0xAAA. It is updated according to a specific algorithm 3 HB clock cycles after each triggered event. The value of this register. Setting the MAMPx[3:0] bits of the DAC_CR register can mask some or all of the LFSR data, thus obtaining...

The obtained LSFR value is added to the value of DAC_DHRx, and after removing the overflow bit, it is written to the DAC_DORx register. If the LFSR register... A value of 0x000 will inject 1 (anti-lock mechanism). Setting WAVEx[1:0] to 00b will reset the LFSR waveform generation algorithm.

Note: To generate noise, triggering must be enabled, i.e., the bit in the DAC_CTLR register must be set to 0.

DAC_TENx 1

Figure 18-5 LFSR Register Algorithm



18.3 Dual DAC Conversion

When simultaneous conversion by two DACs is required, the module integrates three dual DACs for more convenient and efficient operation. The data registers DAC_RD8BDHR, DAC_LD12BDHR, and DAC_RD12BDHR are used in this mode. Only one of these registers needs to be manipulated for further information. The conversion values of the two new DACs.

For dual-DAC conversion, in conjunction with other registers in the module, 11 different conversion modes can be achieved, with two channels to be converted. The value needs to be written to one of the three dual-channel data registers mentioned above.

18.3.1 Under different triggers, use the same LFSR setting:

TENx is set, TSELx has different values, WAVEx is 0b01, and MAMPx is the same LFSR mask value. When channel 1 is triggered... When the event occurs, add the LFSR1 counter value with the same mask to the Channel 1 data register DAC_DHR1 value, and delay for 3 HB clock cycles. The data is sent to DAC_DOR1 for conversion and LFSR1 is updated; when the channel 2 trigger event occurs, the channel 2 data register DAC_DHR2 is updated. The value is added to the LFSR2 count value with the same shielding, and after a delay of 3 HB clock cycles, it is sent to DAC_DOR2 for conversion and to update LFSR2.

18.3.2 Under different triggering conditions, use different LFSR

settings: TENx is set, TSELx has different values, WAVEx is 0b01, and MAMPx has different LFSR masking values. When channel 1 is triggered... When the event occurs, add the value of DAC_DHR1 in the channel 1 data register to the LFSR1 counter value set by MAMP1[3:0], and delay for 3 seconds.

After one HB clock cycle, the data is sent to DAC_DOR1 for conversion and to update LFSR1; when the channel 2 trigger event occurs, the channel 2 data is sent to...

The value of register DAC_DHR2 is added to the LFSR2 count value set by MAMP2[3:0], and after a delay of 3 HB clock cycles, it is sent to DAC_DOR2.

Used for conversion and updating LFSR2.

18.3.3 Generating the same triangular wave under different triggers

Set TENx to set, TSELx to different values, WAVEx to 0b1x, and MAMPx to the same triangular amplitude value. When channel 1 triggers an event...

When the event occurs, add the value of the Channel 1 data register DAC_DHR1 to the value of the triangular wave counter of the same amplitude set by MAMPx, and delay for 3 seconds.

After one HB clock cycle, the signal is sent to DAC_DOR1 for conversion and to update the channel 1 triangular wave counter; when the channel 2 trigger event occurs, it will...

The value of channel 2 data register DAC_DHR2 is added to the value of the triangular wave counter of the same amplitude set by MAMPx, and then sent after a delay of 3 HB clock cycles.

Give DAC_DOR2 for conversion and update the channel 2 triangular wave counter.

18.3.4 Different triangular waves are generated under different triggering conditions

Set TENx to set, TSELx to different values, WAVEx to 0b1x, and MAMPx to different triangular wave amplitude values. When channel 1 is triggered...

When the event occurs, add the value of the triangular wave counter set by MAMP1 to the value of the channel 1 data register DAC_DHR1, and delay for 3 HB.

The clock signal is sent to DAC_DOR1 for conversion and to update the channel 1 triangular wave counter; when the channel 2 trigger event occurs, channel 2...

The value of the data register DAC_DHR2 is added to the triangular wave counter value set by MAMP2, and after a delay of 3 HB clock cycles, it is sent to DAC_DOR2.

Used for conversion and updating the channel 2 triangular wave counter.

18.3.5 Waveform generator not used under different trigger conditions

Set TENx to the set position, and select different trigger sources for different TSELx values. When a trigger event occurs on channel 1, the channel 1 data register will be accessed.

The value of DAC_DHR1 is delayed by 3 HB clock cycles and then sent to DAC_DOR1 for conversion; when the channel 2 trigger event occurs, the channel 2 number is...

The value of register DAC_DHR2 is delayed by 3 HB clock cycles and then sent to DAC_DOR2 for conversion.

18.3.6 Both configurations use software triggering.

When writing the required conversion value to the dual-channel data registers, after one HB clock cycle, the values in DAC_DHR1 and DAC_DHR2 will be converted.

The data is sent to DAC_DOR1 and DAC_DOR2 respectively for conversion.

18.3.7 Under the same trigger, use the same LFSR settings: TENx is set,

TSELx has the same value, WAVEx is 0b01, and MAMPx has the same LFSR masking value. When the trigger event occurs...

After generation, the value of register DAC_DHR1 is added to the LFSR1 counter value with the same mask, and then sent to DAC_DOR1 after a delay of 3 HB clock cycles.

Used for conversion and updating LFSR1, while the value of register DAC_DHR2 is added to the LFSR2 count value with the same mask, delayed by 3 HB.

The clock signal is then sent to DAC_DOR2 for conversion and to update LFSR2.

18.3.8 Under the same trigger, use different LFSR settings: TENx is set,

TSELx has the same value, WAVEx is 0b01, and MAMPx has different LFSR masking values. When the trigger event occurs...

After generation, the value of register DAC_DHR1 is added to the LFSR1 counter value with different masking values, and then sent to DAC_DOR1 after a delay of 3 HB clock cycles.

Used for conversion and updating LFSR1, while adding the LFSR2 counter value with different mask values to the value of register DAC_DHR2, delayed by 3 seconds.

The HB clock signal is then sent to DAC_DOR2 for conversion and to update LFSR2.

18.3.9 The same triangular wave is generated under the same trigger.

Set TENx to the set value, TSELx to the same value, WAVEx to 0b1x, and MAMPx to the same triangular wave amplitude value. When the trigger event occurs...

Then, the value of register DAC_DHR1 is incremented by a counter value of the same triangular wave amplitude, and after a delay of 3 HB clock cycles, it is sent to DAC_DOR1 for use.

The converter converts and updates the channel 1 triangular wave counter, while simultaneously adding the same triangular wave amplitude value to the counter value in register DAC_DHR2, and delays...

The signal is sent to DAC_DOR2 after a delay of 3 HB clock cycles for conversion and to update the channel 2 triangular wave counter value.

18.3.10 Different triangular waves generated under the same trigger

Set TENx to the set bit, TSELx to the same value, WAVEx to 0b1x, and MAMPx to different triangular wave amplitude values. When a trigger event occurs... Then, the value of register DAC_DHR1 is added to the triangular wave amplitude counter value set by MAMP1[3:0], and sent to the register after a delay of 3 HB clock cycles. DAC_DOR1 is used for conversion and to update the channel 1 triangular wave counter. Simultaneously, the value of register DAC_DHR2 is incremented by the value set by MAMP2[3:0]. The triangular wave amplitude counter value is sent to DAC_DOR2 after a 3-HB clock delay for conversion and to update the triangular wave counter of channel 2. value.

18.3.11 Waveform generator not used under the same trigger condition

Set TENx to the set bit, and set TSELx to the same value. With this configuration, when a trigger event occurs, registers DAC_DHR1 and DAC_DHR2... The values are sent to DAC_DOR1 and DAC_DOR2 respectively after a delay of 3 HB clock cycles for DAC conversion.

18.4 Register Description

Table 18-2 List of DAC-related registers

name	Access	describe	Reset value
R32_DAC_CTLR	address 0x40007400 DAC configuration		0x00000000
R32_DAC_SWTR	register, 0x40007404 DAC software trigger register		0x00000000
R32_DAC_R12BDHR1	0x40007408	DAC Channel 1 Right-aligned 12-bit data storage register instrument	0x00000000
R32_DAC_L12BDHR1	0x4000740C	DAC Channel 1 Left-aligned 12-bit data storage register instrument	0x00000000
R32_DAC_R8BDHR1	0x40007410 DAC Channel 1 Right-aligned 8-bit data storage register 0x00000000		
R32_DAC_R12BDHR2	0x40007414	DAC Channel 2 Right-aligned 12-bit data storage register instrument	0x00000000
R32_DAC_L12BDHR2	0x40007418	DAC Channel 2 Left-aligned 12-bit data storage register instrument	0x00000000
R32_DAC_R8BDHR2	0x4000741C DAC Channel 2 Right-aligned 8-bit data storage register 0x00000000 0x40007420		
R32_DAC_RD12BDHR	Dual-channel Right-aligned 12-bit data storage register 0x00000000 0x40007424 Dual-channel		
R32_DAC_LD12BDHR	Left-aligned 12-bit data storage register 0x00000000 0x40007428 Dual-channel Right-aligned		
R32_DAC_RD8BDHR	8-bit data storage register 0x00000000 0x4000742C DAC Channel 1 Data Output Register		
R32_DAC_DOR1	0x40007430 DAC Channel 2 Data Output Register		0x00000000
R32_DAC_DOR2			0x00000000

18.4.1 DAC Configuration Register (DAC_CTLR) Offset

Address: 0x00

31	30 29	28	27 26 25 24 23 22 21	2019	18	17 16
Reserved		DMAEN2	MAMP2[3:0]	WAVE2[2:0]	TSEL2[2:0]	TEN2 BOFF2 EN2

15	14 13	12	11 10 9 8	7 6	5 4 3	2	1	0
Reserved		DMAEN1	MAMP1[3:0]	WAVE1[2:0]	TSEL1[2:0]	TEN1 BOFF1 EN1		

Bit	Name access	describe	Reset value
[31:29] Reserved		RO is reserved.	0
28	DMAEN2	DMA enable for RW DAC channel 2:	0

			1: Enable DMA function for DAC channel 2; 0: Disable DMA function for DAC channel 2.	
[27:24]	MAMP2[3:0]	RW	DAC Channel 2 Masking/Amplitude Settings: This software setting area is used to... In noise generation mode, select the LFSR data mask bit, and generate a triangular waveform. Select the waveform amplitude in the forming mode: 0000: LFSR not shielded, bit 0 / triangular wave amplitude value is 1; 0001: LFSR bit [1:0] not masked / triangular wave amplitude value is 3; 0010: LFSR bit [2:0] not masked / triangular wave amplitude value is 7; 0011: LFSR bit [3:0] not masked / triangular wave amplitude value is 15; 0100: LFSR bit [4:0] not masked / triangle wave amplitude value is 31; 0101: LFSR bit [5:0] not masked / triangular wave amplitude value is 63; 0110: LFSR bits [6:0] not masked / triangular wave amplitude value is 127; 0111: LFSR bits [7:0] are not masked / the triangular wave amplitude is 255; 1000: Unmasked LFSR bits [8:0] / Triangle wave amplitude value is 511; 1001: Unmasked LFSR bits [9:0] / Triangle wave amplitude value is 1023; 1010: Unmasked LFSR bits [10:0] / Triangle wave amplitude value is 2047; ÿ1011: LFSR bit [11:0] is not masked / the triangular wave amplitude is 4095.	0000b
[23:22]	WAVE2[1:0]	RW	Noise/triangle wave generation enable for DAC channel 2: 00: Turn off the waveform generator; 01: Enable noise waveform generator; 1x: Enables the triangular wave generator.	00b
[21:19]	TSEL2[2:0]	RW	DAC Channel 2 Trigger Event Selection Settings: 000: TIM6 TRGO incident; 001: TIM8 TRGO Incident; 010: TIM7 TRGO Incident 011: TIM5 TRGO Incident 100: TIM2 TRGO incident; 101: The TIM4 TRGO Incident; 110: External interrupt line 9; 111: Software triggered; Other: Reserved.	000b
18	TEN2	RW	DAC Channel 2 External Trigger Mode Enable: 1: Enable the trigger function of DAC channel 2 and write it to the DAC_xDHR register. The data is sent to the DAC_DOR2 register after 3 HB clock cycles. 0: Disable DAC channel 2 trigger function, write to DAC_xDHR register. The data is sent to the DAC_DOR2 register after 1 HB clock cycle. Note: If software triggering is selected, the data in DAC_xDHR will be sent to 1 indivual HB the DAC_DOR2 register after one clock cycle.	0
17	BOFF2	RW	DAC Channel 2 Output Buffer Off Control (Recommended to Enable): 1: Disable the output buffer of DAC channel 2; 0: Enable the output buffer of DAC channel 2.	0
16	EN2	RW	DAC Channel 2 Enable: 1: Enable DAC channel 2; 0: Disable DAC channel 2.	0
[15:13]	Reserved	RO is reserved.		0

12	DMAEN1	RW	DMA enable for DAC channel 1: 1: Enable DMA function for DAC channel 1; 0: Disable DMA function for DAC channel 1.	0
[11:8] MAMP1	[3:0]	RW	DAC Channel 1 Masking/Amplitude Settings: This software setting area is used to... In noise generation mode, select the LFSR data mask bit, and generate a triangular waveform. Select the waveform amplitude in the forming mode: 0000: LFSR not shielded, bit 0 / triangular wave amplitude value is 1; 0001: LFSR bit [1:0] not masked / triangular wave amplitude value is 3; 0010: LFSR bit [2:0] not masked / triangular wave amplitude value is 7; 0011: LFSR bit [3:0] not masked / triangular wave amplitude value is 15; 0100: LFSR bit [4:0] not masked / triangle wave amplitude value is 31; 0101: LFSR bit [5:0] not masked / triangular wave amplitude value is 63; 0110: LFSR bits [6:0] not masked / triangular wave amplitude value is 127; 0111: LFSR bits [7:0] are not masked / the triangular wave amplitude is 255; 1000: Unmasked LFSR bits [8:0] / Triangle wave amplitude value is 511; 1001: Unmasked LFSR bits [9:0] / Triangle wave amplitude value is 1023; 1010: Unmasked LFSR bits [10:0] / Triangle wave amplitude value is 2047; ÿ1011: LFSR bit [11:0] is not masked / the triangular wave amplitude is 4095.	0000b
[7:6] WAVE1	[1:0]	RW	Noise/triangle wave generation enabled for DAC channel 1. 00: Turn off the waveform generator; 01: Enable noise waveform generator; 1x: Enables the triangular wave generator.	00b
[5:3] TSEL1	[2:0]	RW	DAC Channel 1 Trigger Event Selection Settings: 000: TIM6 TRGO incident; 001: TIM8 TRGO Incident; 010: TIM7 TRGO Incident 011: TIM5 TRGO Incident 100: TIM2 TRGO incident; 101: The TIM4 TRGO incident; 110: External interrupt line 9; 111: Software triggered; Other: Reserved.	000b
2	TEN1	RW	DAC Channel 1 External Trigger Mode Enable: 1: Enable DAC channel 1 trigger function and write to the DAC_xDHR register. The data is sent to the DAC_DOR1 register after 3 HB clock cycles; 0: Disable DAC channel 1 trigger function, write to DAC_xDHR register. The data is sent to the DAC_DOR1 register after 1 HB clock cycle. Note: If software triggering is selected, only one data point is needed in DAC_xDHR. HB 1. The data is sent to the DAC_DOR1 register after one clock cycle.	0
1	BOFF1	RW	DAC Channel 1 Output Buffer Off Control (Recommended to Enable): 1: Disable the output buffer of DAC channel 1; 0: Enable DAC channel 1 output buffer.	0
0	EN1	RW	DAC Channel 1 Enable: 1: Enable DAC channel 1; 0: Disable DAC channel 1.	0

18.4.2 DAC Software Trigger Register (DAC_SWTR)

Offset Address: 0x04

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														SWTRIG2	SWTRIG1

Bit	name	access	describe	Reset value
[31:2]	Reserved	RO	is reserved.	0
1	SWTRIG2	WO	DAC Channel 2 Software Trigger Control Bit: 1: Enable DAC channel 2 software trigger; 0: Disable DAC channel 2 (software triggered). Note: Once the data in DAC_xDHR (1 clock cycle) is released... After the HB period, the bit is sent to the DAC_DOR2 register, which will clear the hardware. 0.	0
0	SWTRIG1	WO	DAC Channel 1 Software Trigger Control Bit: 1: Enable DAC channel 1 software trigger; 0: Disable DAC channel 1 (software triggered). Note: Once the data in DAC_xDHR (1 clock cycle) is released... After the HB period, the bit is sent to the DAC_DOR1 register, which will clear the hardware. 0.	0

18.4.3 Right-aligned 12-bit data storage register for DAC channel 1 (DAC_R12BDHR1)

Offset address: 0x08

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				DACC1DHR[11:0]											

Bit	name	access	describe	Reset value
[31:12]	Reserved	RO	is reserved.	0
[11:0]	DACC1DHR[11:0]	12-bit	right-aligned data for RW DAC channel 1.	0

18.4.4 DAC Channel 1 Left-Aligned 12-Bit Data Storage Register (DAC_L12BDHR1)

Offset address: 0x0C

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACC1DHR[11:0]												Reserved			

Bit	name	access	describe	Reset value
[31:16]	Reserved	RO	is reserved.	0
[15:4]	DACC1DHR[11:0]		12-bit left-aligned data for RW DAC channel 1.	0
[3:0]	Reserved	RO	is reserved.	0

18.4.5 Right-aligned 8-bit data storage register for DAC channel 1 (DAC_R8BDHR1)

Offset address: 0x10

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								DACC1DHR[7:0]							

Bit	name	access	describe	Reset value
[31:8]	Reserved	RO	is reserved.	0
[7:0]	DACC1DHR[7:0]		8-bit right-aligned data from RW DAC channel 1.	0

18.4.6 DAC Channel 2 Right-Aligned 12-Bit Data Storage Register (DAC_R12BDHR2)

Offset address: 0x14

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				DACC2DHR[11:0]											

Bitname		access	describe	Reset value
[31:12]	Reserved	RO	is reserved.	0
[11:0]	DACC2DHR[11:0]		12-bit right-aligned data from RW DAC channel 2.	0

18.4.7 DAC Channel 2 Left-Aligned 12-Bit Data Storage Register (DAC_L12BDHR2)

Offset address: 0x18

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACC2DHR[11:0]												Reserved			

Bit	name	access	describe	Reset value
[31:16]	Reserved	RO	is reserved.	0
[15:4]	DACC2DHR[11:0]		12-bit left-aligned data from RW DAC channel 2.	0
[3:0]	Reserved	RO	is reserved.	0

18.4.8 DAC Channel 2 Right-Aligned 8-Bit Data Storage Register (DAC_R8BDHR2)

Offset address: 0x1C

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								DACC2DHR[7:0]							

Bit	name	access	describe	Reset value
[31:8]	Reserved	RO	is reserved.	0
[7:0]	DACC2DHR[7:0]		8-bit right-aligned data from RW DAC channel 2.	0

18.4.9 DAC Dual-Channel Right-Aligned 12-Bit Data Storage Register (DAC_RD12BDHR)

Offset address: 0x20

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved											DACC2DHR[11:0]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											DACC1DHR[11:0]				

Bit	name	access	describe	Reset value
[31:28]	Reserved	RO	is reserved.	0
[27:16]	DACC2DHR[11:0]		12-bit right-aligned data from RW DAC channel 2.	0
[15:12]	Reserved	RO	is reserved.	0
[11:0]	DACC1DHR[11:0]		12-bit right-aligned data for RW DAC channel 1.	0

18.4.10 DAC Dual-Channel Left-Aligned 12-Bit Data Storage Register (DAC_LD12BDHR)

Offset address: 0x24

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
DACC2DHR[11:0]												Reserved		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACC1DHR[11:0]												Reserved			

Bit	name	access	describe	Reset value
[31:20]	DACC2DHR[11:0]		12-bit left-aligned data from RW DAC channel 2.	0
[19:16]	Reserved	RO	is reserved.	0
[15:4]	DACC1DHR[11:0]		12-bit left-aligned data for RW DAC channel 1.	0
[3:0]	Reserved	RO	is reserved.	0

18.4.11 DAC Dual-Channel Right-Aligned 8-Bit Data Storage Register (DAC_RD8BDHR)

Offset address: 0x28

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACC2DHR[7:0]								DACC1DHR[7:0]							

Bit	name	access	describe	Reset value
[31:16]	Reserved	RO	is reserved.	0
[15:8]	DACC2DHR[7:0]		8-bit right-aligned data from RW DAC channel 2.	0
[7:0]	DACC1DHR[7:0]		8-bit right-aligned data from RW DAC channel 1.	0

18.4.12 DAC Channel 1 Data Output Register (DAC_DOR1)

Offset address: 0x2C

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				DACC1DOR[11:0]											

Bit	name	access	describe	Reset value
[31:12]	Reserved	RO	is reserved.	0
[11:0]	DACC1DOR[11:0]	RO	DAC Channel 1 output data.	0

18.4.13 DAC Channel 2 Data Output Register (DAC_DOR2)

Offset address: 0x30

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				DACC2DOR[11:0]											

Bitname		Access	Description	Reset value
[31:12]	Reserved [11:0]	RO	is reserved.	0
	DACC2DOR[11:0]	RO	DAC Channel 2 Output Data.	0

Chapter 19 Universal Synchronous Asynchronous Receiver/Transmitter (USART)

This module contains 8 general-purpose synchronous and asynchronous transceivers (USART1/2/3/4/5/6/7/8).

19.1 Main Features

Full - duplex or half-duplex synchronous or asynchronous communication

ÿ NRZ data format

Fractional baud rate generator, up to 9 Mbps ;

Programmable data length.

ÿ Configurable stop bits

Supports LIN, IrDA encoders, and smart cards.

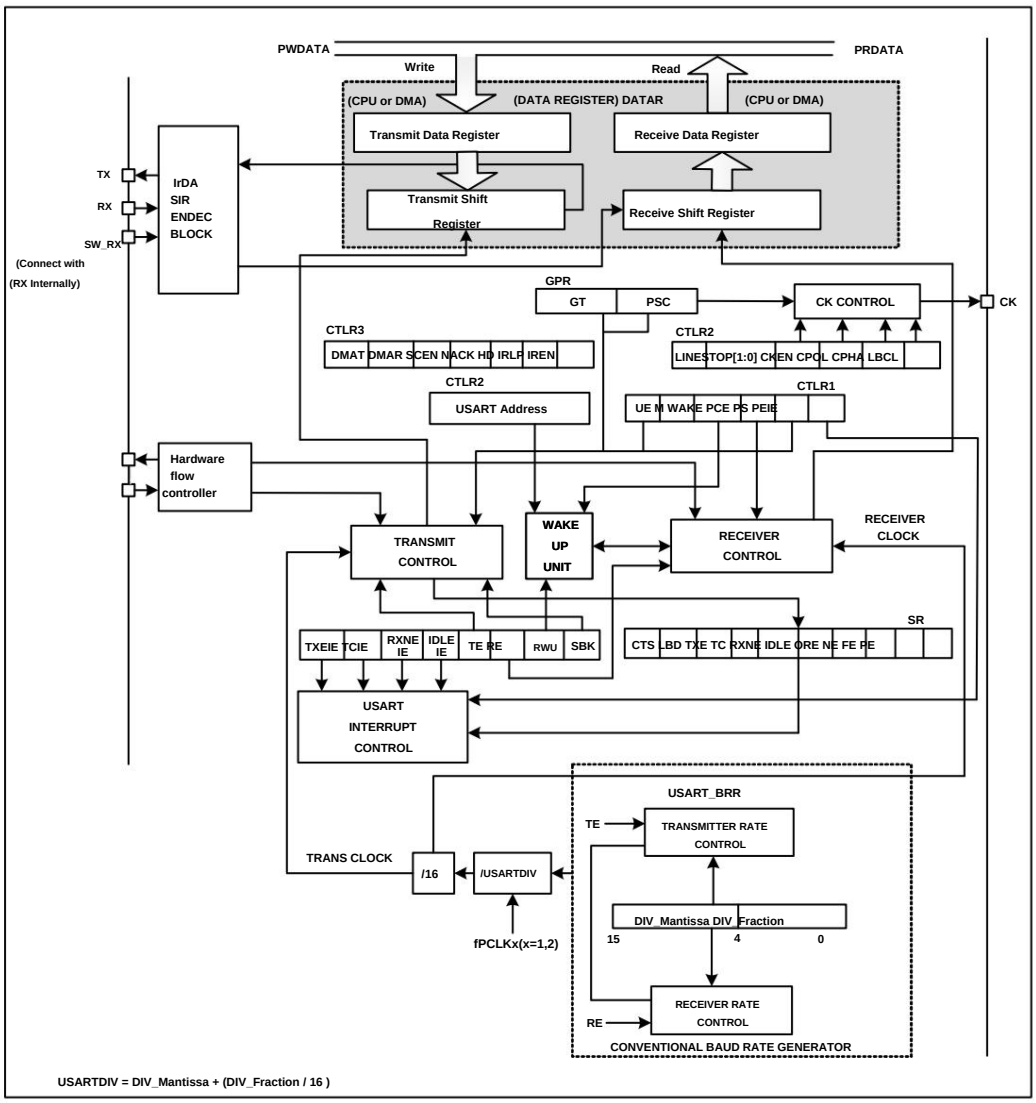
Supports DMA ÿ

Multiple interrupt sources

Supports wake-up function in low power mode

19.2 Overview

Figure 19-1 Block diagram of a general synchronous/asynchronous transceiver



When TE (Transmit Enable) is set, the data in the transmit shift register is output on the TX pin, and the clock is output on the CK pin. During transmission, the least significant bit is shifted out first. Each data frame begins with a low start bit, then the transmitter sends an eight- or nine-bit data word depending on the setting in the M (word length) bit, followed by a configurable number of stop bits. If a parity bit is included, the last bit of the data word is a parity bit. After TE is set, an idle frame is sent, which is 10 or 11 bits high and includes the stop bit. The break frame is 10 or 11 bits low, followed by the stop bit.

19.3 Baud Rate Generator

The transceiver's baud rate = HCLK / (16 * USARTDIV). The value of USARTDIV is based on DIV_M and DIV_F in USARTx_BRR.

The specific calculation formula, determined by the two domains, is as follows:

USARTDIV = DIV_M + (DIV_F/16) It's important to note

that the bit rate generated by the baud rate generator may not perfectly match the user's desired baud rate; there might be a discrepancy. Besides using a value as close as possible, another way to reduce this discrepancy is to increase the clock speed of HB. For example, if the baud rate is set to 9600bps, and USARTDIV is set to 78.125, a baud rate of 9600bps can be obtained at 12MHz. However, if you need a baud rate of 115200bps, the calculated USARTDIV is 6.51, but the closest value entered in USART_BRR can be is 6.5, resulting in an actual baud rate of 115384bps, an error of 0.16%. When the serial waveform sent by the sender reaches the receiver, there is a certain error between the baud rates of the receiver and the sender. This error mainly comes from three aspects:

the actual baud rates of the receiver and the sender are inconsistent; there are clock errors between the receiver and the sender; and changes in the waveform during transmission. The receiver of the peripheral module has a certain reception tolerance. When the sum of the deviations caused by the above three aspects is less than the module's tolerance limit, this total deviation will not affect transmission and reception. The module's tolerance limit is affected by whether fractional baud rate and M bits (data field word length) are used. Using fractional baud rate and a 9-bit data field length will reduce the tolerance limit, but not less than 3%.

19.4 Synchronization Mode

Synchronous mode allows the system to output a clock signal when using the USART module. When transmitting data in synchronous mode, CK...

The pin will simultaneously output a clock signal.

To enable synchronous mode, set the CLKEN bit in control register 2 (R16_USARTx_CTLR2). However, LIN mode, smart card mode, infrared mode, and half-duplex mode must be disabled simultaneously, meaning the SCEN, HDSEL, and IREN bits must be in the reset state. These three bits are located in control register 3 (R16_USARTx_CTLR3). The key to using synchronous mode lies in clock

output control. The following points should be noted: The USART module's synchronous mode only operates in

master mode, meaning the CK pin only outputs the clock and does not receive input; the clock signal is only output when data is

output on the TX pin; the LBCL bit determines whether a clock is

output when sending the last data bit, the CPOL bit determines the clock polarity, and the CPHA bit determines the clock phase. These three bits are in control register 2 (R16_USARTx_CTLR2), and they must be set when TE and RE are not enabled. See Figure 19-2 for specific differences. In synchronous mode, the receiver only samples when the clock is output, and the device needs to

maintain a certain signal setup time and hold time, as shown in Figure 19-3.

Figure 19-2 USART clock timing example (M=0)

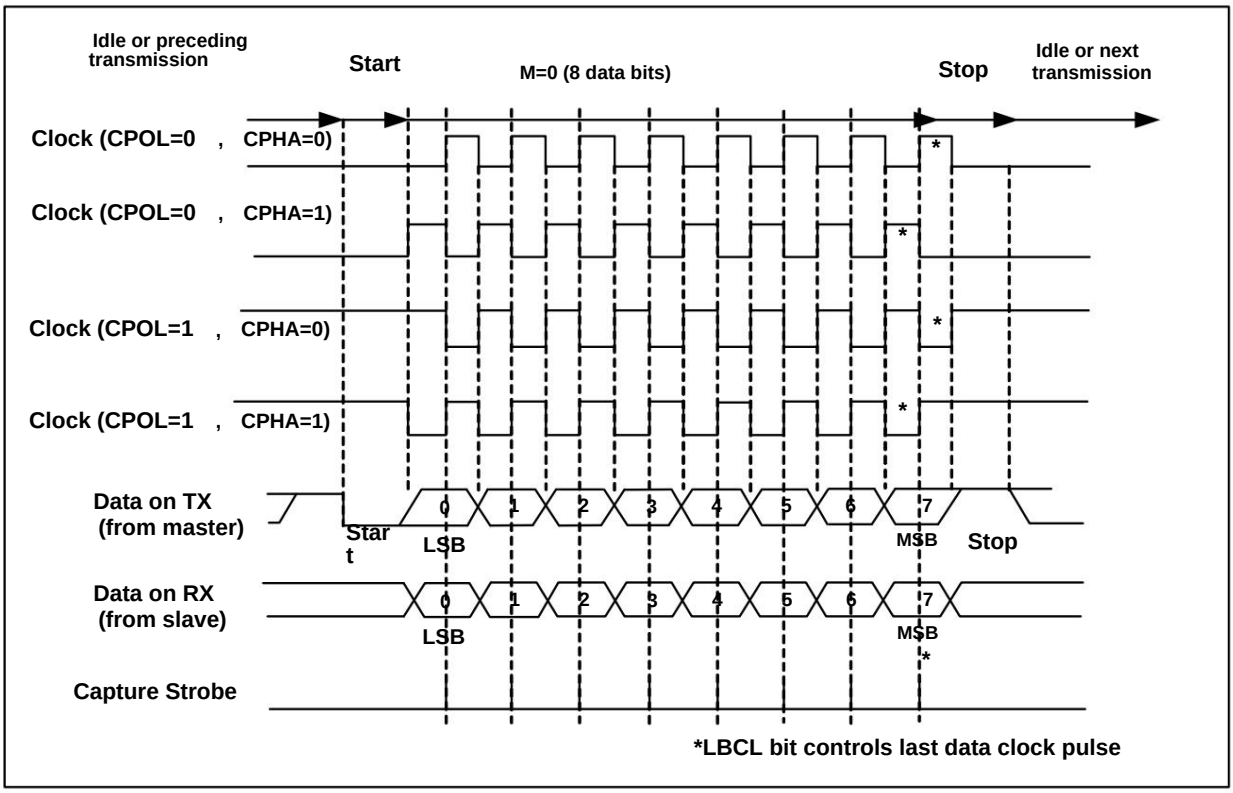
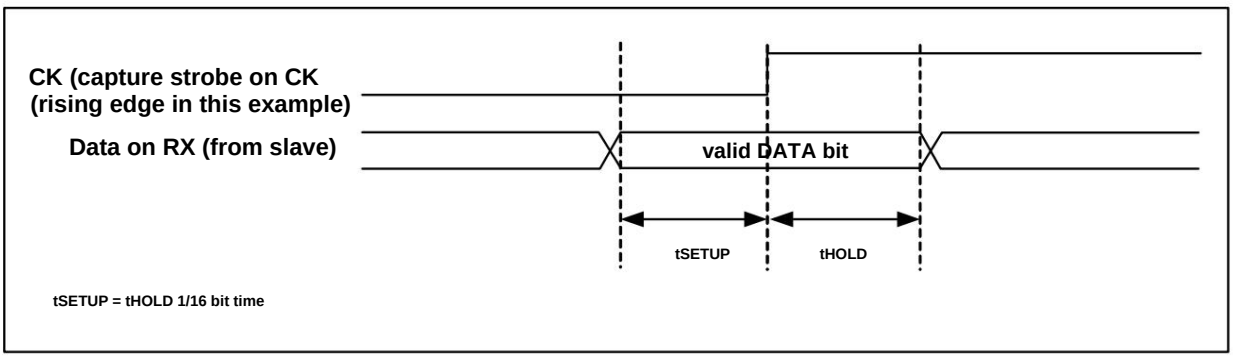


Figure 19-3 Data sampling and holding time



19.5 Single-line half-duplex mode

Half-duplex mode supports using a single pin (TX pin only) for both receiving and transmitting. The TX and RX pins are internally connected.

catch.

To enable half-duplex mode, set the HDSEL bit in control register 3 (R16_USARTx_CTLR3), but you also need to disable it.

Disable LIN mode, smart card mode, infrared mode, and synchronization mode, ensuring that the SCEN, CLKEN, and IREN bits are in the reset state.

The three bits are in control registers 2 and 3 (R16_USARTx_CTLR2 and R16_USARTx_CTLR3).

After setting to half-duplex mode, the TX I/O port needs to be set to open-drain high output mode. With TE set, as long as...

Data is sent after it is written to the data register. It's important to note that in half-duplex mode, multiple devices may use a single data transfer register.

Bus conflicts during wired transmission and reception need to be avoided by the user through software.

19.6 Smart Card

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

To enable smart card mode, set the SCEN bit in control register 3 (R16_USARTx_CTLR3), but you also need to disable it at the same time.

LIN mode, half-duplex mode, and infrared mode ensure that the LINEN, HDSEL, and IREN bits are in the reset state, but can be enabled.

CLKEN is used to output the clock, and these bits are in control registers 2 and 3 (R16_USARTx_CTLR2 and R16_USARTx_CTLR3).

To support smart card mode, the USART should be set to 8 data bits plus 1 parity bit, and its stop bit should ideally be configured to transmit.

Both sending and receiving are 1.5 bits. Smart card mode is a single-wire half-duplex protocol that uses the TX line for data communication and should be configured...

Set to open-drain output with pull-up. When the receiver receives a data frame and detects a parity error, it will send a NACK at the stop bit.

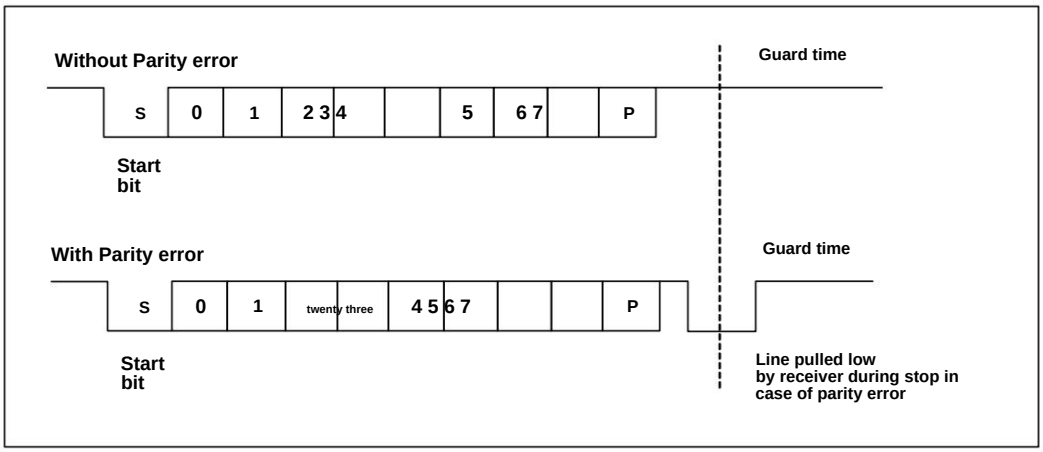
The signal, specifically the NACK signal, involves actively pulling TX low for one cycle during the stop bit period. When the sender detects the NACK signal, a frame error occurs, and the application...

This allows for retransmission. Figure 17-4 shows the waveforms on the TX pin under correct conditions and in the case of a parity error. USART

The TC flag (transmission complete flag) can be generated with a delay of GT (guard time) clock cycles, and the receiver will not set its own NACK signal.

The sign is recognized as the starting position.

Figure 19-4 Schematic diagram of (not) parity check error



In smart card mode, the waveform output after the CK pin is enabled is unrelated to communication; it merely provides the clock signal to the smart card.

It is the HB clock, which is then divided by a five-bit configurable clock (the division value is twice that of the PSC, with a maximum division of 62).

19.7 IrDA

The USART module supports controlling the IrDA infrared transceiver for physical layer communication. When using IrDA, LINEN, STOP, CLKEN, and CLKEN must be cleared.

SCEN and HDSEL bits. NRZ (Non-Return-to-Zero) encoding is used between the USART module and the SIR physical layer (infrared transceiver), supporting up to...

Up to 115200bps speed.

IrDA is a half-duplex protocol. If the UASRT is sending data to the SIR physical layer, the IrDA decoder will ignore new data.

If the USART is receiving data from the SIR, then the SIR will not receive the signal from the USART.

The logic levels used by the USART to send data to the SIR and by the SIR to send data to the USART are different. In the SIR's receiving logic, a high level is represented by 1, and a low level by 0.

0, but in the SIR transmission logic, a high level is 0 and a low level is 1.

19.8 DMA

The USART module supports DMA functionality, enabling fast continuous transmission and reception. When DMA is enabled, TXE is set, and DMA...

It will then write data from the designated memory space to the transmit buffer. When using DMA for reception, the DMA will write the receive data each time RXNE is set.

Data in the buffer is transferred to a specific memory space.

19.9 Interruption

The USART module supports multiple interrupt sources, including Transmit Data Register Empty (TXE), CTS, Transmit Complete (TC), and Receive Data.

Ready (RXNE), Data Overflow (ORE), Line Idle (IDLE), Parity Error (PE), Disconnect Flag (LBD), Noise

Sound (NE), overflow (ORE) in multibuffered communication, and frame error (FE), etc.

Table 19-1 Relationship between interrupts and their corresponding enable bits

Interruption source	Enable bit
Data register empty (TXE)	TXEIE
Transmit enabled	CTSIE
(CTS) Transmission	TCIE
complete (TC) Ready to receive data (RXNE)	RXNEIE
Data overflow (ORE)	
Line Idle (IDLE) Parity	IDLEIE
Error (PE) Disconnect Flag	PEIE
(LBD) Noise (NE)	LBDIE
	EIE
Overflow (ORE) in multibuffered	
communication; Frame error (FE) in multibuffered communication.	

19.10 Low-Power Wake-Up

Low-power wake-up is enabled by the LPWKUP_EN bit in the USARTx_CTLR3 register, and is controlled by the USARTx_STATR register. The LPWKUP_ACT_FLAG bit reads the indicator flag for the start of low-power wake-up; the clock source for low-power wake-up is stored via the USARTx_CTLR3 register. The LPWKUP_CK_SRC bit is used to select either LSE or LSI, and the LPWKUP_DLY_CFG bit can be used to configure the clock for low-power wake-up receive delay. Source cycle.

19.11 Register Description

Table 19-2 List of USART1 Related Registers

name	Access address	describe	Reset value
R32_USART1_STATR	0x40013800	UASRT1 Status Register	0x000000C0
R32_USART1_DATAR	0x40013804	UASRT1 Data Register	0x00000000
R32_USART1_BRR	0x40013808	UASRT1 Baud Rate Register	0x00000000
R32_USART1_CTLR1	0x4001380C	UASRT1 Control Register 1	0x00000000
R32_USART1_CTLR2	0x40013810	UASRT1 Control Register 2	0x00000000
R32_USART1_CTLR3	0x40013814	UASRT1 Control Register 3	0x00000000
R32_USART1_GPR	0x40013818	UASRT1 protection time and prescaler register	0x00000000
R32_USART1_CTLR4	0x4001381C	UASRT1 Control Register 4	0x00000000

Table 19-3 List of USART2 Related Registers

name	Access address	describe	Reset value
R32_USART2_STATR	0x40004400	UASRT2 Status Register	0x000000C0
R32_USART2_DATAR	0x40004404	UASRT2 Data Register	0x00000000
R32_USART2_BRR	0x40004408	UASRT2 Baud Rate Register	0x00000000
R32_USART2_CTLR1	0x4000440C	UASRT2 Control Register 1	0x00000000
R32_USART2_CTLR2	0x40004410	UASRT2 Control Register 2	0x00000000
R32_USART2_CTLR3	0x40004414	UASRT2 Control Register 3	0x00000000
R32_USART2_GPR	0x40004418	UASRT2 protection time and prescaler register	0x00000000

R32_USART2_CTLR4	0x4000441C	UASRT2 Control Register 4	0x00000000

Table 19-4 List of USART3 Related Registers

name	Access address	describe	Reset value
R32_USART3_STATR	0x40004800	UASRT3 Status Register	0x000000C0
R32_USART3_DATAR	0x40004804	UASRT3 Data Register	0x00000000
R32_USART3_BRR	0x40004808	UASRT3 Baud Rate Register	0x00000000
R32_USART3_CTLR1	0x4000480C	UASRT3 Control Register 1	0x00000000
R32_USART3_CTLR2	0x40004810	UASRT3 Control Register 2	0x00000000
R32_USART3_CTLR3	0x40004814	UASRT3 Control Register 3	0x00000000
R32_USART3_GPR	0x40004818	UASRT3 protection time and prescaler register	0x00000000
R32_USART3_CTLR4	0x4000481C	UASRT3 Control Register 4	0x00000000

Table 19-5 List of USART4 Related Registers

name	Access address	describe	Reset value
R32_USART4_STATR	0x40004C00	USART4 Status Register	0x000000C0
R32_USART4_DATAR	0x40004C04	USART4 Data Register	0x00000000
R32_USART4_BRR	0x40004C08	USART4 Baud Rate Register	0x00000000
R32_USART4_CTLR1	0x40004C0C	USART4 Control Register 1	0x00000000
R32_USART4_CTLR2	0x40004C10	USART4 Control Register 2	0x00000000
R32_USART4_CTLR3	0x40004C14	USART4 Control Register 3	0x00000000
R32_USART4_GPR	0x40004C18	USART4 Guard Time and Prescaler Register	0x00000000
R32_USART4_CTLR4	0x40004C1C	UASRT4 Control Register 4	0x00000000

Table 19-6 List of USART5 Related Registers

name	Access address	describe	Reset value
R32_USART5_STATR	0x40005000	USART5 Status Register	0x000000C0
R32_USART5_DATAR	0x40005004	USART5 Data Register	0x00000000
R32_USART5_BRR	0x40005008	USART5 Baud Rate Register	0x00000000
R32_USART5_CTLR1	0x4000500C	USART5 Control Register 1	0x00000000
R32_USART5_CTLR2	0x40005010	USART5 Control Register 2	0x00000000
R32_USART5_CTLR3	0x40005014	USART5 Control Register 3	0x00000000
R32_USART5_GPR	0x40005018	USART5 Guard Time and Prescaler Register	0x00000000
R32_USART5_CTLR4	0x4000501C	UASRT5 Control Register 4	0x00000000

Table 19-7 List of USART6 Related Registers

name	Access address	describe	Reset value
R32_USART6_STATR	0x40001800	USART6 Status Register	0x000000C0
R32_USART6_DATAR	0x40001804	USART6 Data Register	0x00000000
R32_USART6_BRR	0x40001808	USART6 Baud Rate Register	0x00000000
R32_USART6_CTLR1	0x4000180C	USART6 Control Register 1	0x00000000
R32_USART6_CTLR2	0x40001810	USART6 Control Register 2	0x00000000
R32_USART6_CTLR3	0x40001814	USART6 Control Register 3	0x00000000

R32_USART6_GPR	0x40001818	USART6 Guard Time and Prescaler Register 0x00000000	
R32_USART6_CTLR4	0x4000181C	UASRT6 Control Register 4	0x00000000

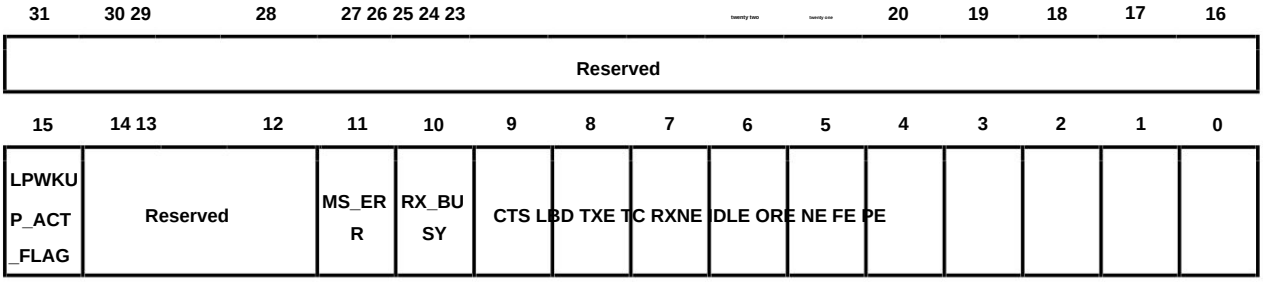
Table 19-8 List of USART7 Related Registers

name	Access address	describe	Reset value
R32_USART7_STATR	0x40001C00	USART7 Status Register	0x000000C0
R32_USART7_DATAR	0x40001C04	USART7 Data Register	0x00000000
R32_USART7_BRR	0x40001C08	USART7 Baud Rate Register	0x00000000
R32_USART7_CTLR1	0x40001C0C	USART7 Control Register 1	0x00000000
R32_USART7_CTLR2	0x40001C10	USART7 Control Register 2	0x00000000
R32_USART7_CTLR3	0x40001C14	USART7 Control Register 3	0x00000000
R32_USART7_GPR	0x40001C18	USART7 Guard Time and Prescaler Register 0x00000000	
R32_USART7_CTLR4	0x40001C1C	UASRT7 Control Register 4	0x00000000

Table 19-9 List of USART8 Related Registers

name	Access address	describe	Reset value
R32_USART8_STATR	0x40002000	USART8 Status Register	0x000000C0
R32_USART8_DATAR	0x40002004	USART8 Data Register	0x00000000
R32_USART8_BRR	0x40002008	USART8 Baud Rate Register	0x00000000
R32_USART8_CTLR1	0x4000200C	USART8 Control Register 1	0x00000000
R32_USART8_CTLR2	0x40002010	USART8 Control Register 2	0x00000000
R32_USART8_CTLR3	0x40002014	USART8 Control Register 3	0x00000000
R32_USART8_GPR	0x40002018	USART8 Guard Time and Prescaler Register 0x00000000	
R32_USART8_CTLR4	0x4000201C	UASRT8 Control Register 4	0x00000000

19.11.1 USART Status Register (USARTx_STATR) (x=1/2/3/4/5/6/7/8) Offset Address: 0x00

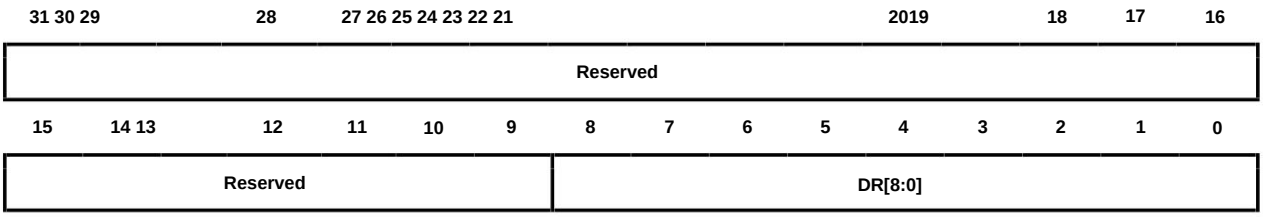


Bit	name	access	describe	Reset value
[31:16] Reserved		RO is reserved.		0
15	LPWKUP_ACT_FLAG	RW1Z	Low-power wake-up start indicator bit, write 1 to clear.	0
[14:12] Reserved		RO is reserved.		0
11	MS_ERR	RO	MARK or SPACE checksum error flag. In receive mode, If a MARK or SPACE checksum error occurs, the hardware will set a flag. This bit. Reading this bit and then reading the data register will reset this bit. This bit is set if MS_ERRIE was previously set. The bit will generate a corresponding interrupt.	0

10	RX_BUSY	RO	Receive status indicator bit: 1: This bit is set when the device is in receive mode; 0: In non-receiving state.	0
9	CTS	RW0	CTS status change flag. If the CTSE bit is set, when When the nCTS output state changes, this bit will be set high by hardware. Software reset. If the CTSIE bit is already set, it will generate... Life was interrupted. 1: There are changes on the nCTS state line; 0: There is no change on the nCTS status line.	0
8	LBD	RW0	LIN Break Detection Flag. When a LIN Break is detected, This bit is set by hardware. It is cleared by software. If LBDIE has already been... If the bit is set, an interrupt will be generated. 1: LIN Break detected; 0: No LIN Break detected.	0
7	TXE	RO	Send data register null flag. When the data in the TDR register... When the data is transferred to the shift register by the hardware, the bit is... Component set. If TXEIE is already set, it will generate... An interrupt occurs when a write operation is performed on the data register; this bit will be... Reset. 1: The data has been transferred to the shift register; 0: The data has not yet been transferred to the shift register.	1
6	TC	RW0	Transmission completion flag. This indicates that a frame containing data has been transmitted. Then, and if TXE is set, the hardware will set this bit. If TCIE is set, a corresponding interrupt will be generated, and the software will read... Writing to the data register after clearing this bit will clear it. Alternatively... Write 0 directly to clear this bit. 1: Sending complete; 0: Sending is not yet complete.	1
5	RXNE	RW0	The read data register is not empty when the number in the shift register is not empty. The data is transferred to the data register, and this bit is set by the hardware. If RXNEIE is already set, a corresponding bit will also be generated. The interrupt. A read operation on the data register can clear that bit. Alternatively, you can write 0 to clear the bit. 1: Data received and can be read; 0: Data not received yet.	0
4	IDLE	RO	Bus idle flag. This bit will be hard-coded when the bus is idle. Set the IDLEIE bit. If IDLEIE is already set, a new value will be generated. The corresponding interrupt. The operation of reading the status register and then reading the data register. This position will be cleared. 1: The bus is currently idle; 0: No bus idle was detected. Note: This bit will not be set again until it is set. RXNE	0
3	ORE	RO	Overload error flag. This flag is triggered when the receive shift register contains data. When it's time to switch to the data register, but the data register's receive... This bit will be set if there is still data to be read from the field.	0

			Setting RXNEIE will also generate a corresponding interrupt. 1: An overload error occurred; 0: No overload error. Note: In the event of an overload error, the value of the data register will not be lost, but the value of the shift register will be overwritten. If set... However, In multi-buffered communication mode, the ORE flag is a bit. setting the EIE bit will generate an interrupt.	
2	NE	RO	Noise error flag. When a noise error flag is detected, it is... Hardware set. After reading the status register, read the data register. The operation will reset this bit. 1: Noise detected; 0: No noise detected. Note: This bit will not generate an interrupt. If EIE is set in Bit, multi-buffered communication mode, the FE flag bit will generate an interrupt. Interrupted.	0
1	FE	RO	Frame error flag. This is triggered when a synchronization error or excessive noise is detected. Or, if a break character is encountered, this bit will be set by the hardware. Read this bit again. Reading the data register will reset this bit. 1: Frame error detected; 0: No frame error detected. Note: This bit will not generate an interrupt. If the bit is set... EIE multi-buffered communication mode, the FE flag set will generate Interrupted.	0
0	PE	RO	Check error flag. In receive mode, if a parity error occurs... Error detected; hardware sets this bit. Read this bit and then read the data register. Operations on the register will reset this bit. The software must clear this bit before... The RXNE flag must be set before PEIE is enabled. If a bit is set, then setting this bit will produce the corresponding result. Break. 1: A parity check error occurred; 0: No verification error.	0

19.11.2 USART Data Register (USARTx_DATAR) (x=1/2/3/4/5/6/7/8) Offset Address: 0x04



Bit	name	access	describe	Reset value
[31:9]	Reserved	RO	is reserved.	0
[8:0]	DR[8:0]	RW	This is the data register. It is actually used to receive data. The two registers are the Read Data Register (RDR) and the Transmit Data Register (TDR).	0

			The registers are composed of registers. The read and write operations of the DR start by reading the register and receiving the register, respectively. The Read-to-Date Register (RDR) and the Write-to-Send Register (TDR).	

19.11.3 USART Baud Rate Register (USARTx_BRR) (x=1/2/3/4/5/6/7/8) Offset Address: 0x08

31 30 29	28	27 26 25 24 23 22 21	2019	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	describe	Reset value
[31:16] Reserved		RO is reserved.		0
[15:4]	DIV_Mantissa [11:0]	The 12 bits RW define the integer part of the divider's division factor. 0		
[3:0] DIV_Fraction[3:0] RW These 4 bits define the fractional part of the divider's division factor.				0

19.11.4 USART Control Register 1 (USARTx_CTLR1) (x=1/2/3/4/5/6/7/8) Offset address: 0x0C

31 30 29	28	27 26 25 24 23 22 21	2019	18	17	16
Reserved						

15	14 13	12	11	10	9	8	7	6	5	4	3	2	1	0
M_EXT	UE	M WAKE PCE	PS PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	RWU	SBK			

Bit	name	access	describe	Reset value
[31:16] Reserved		RO reserved.		0
[15:14] M_EXT		RW	Word length extension bits: 00: Invalid. The M bits determine the data length. 01: 7 data bits; 10: 6 data bits; 11: 5 data bits.	0
13	UE	RW	USART enable bit. 1: Enable USART; 0: USART disabled. When this bit is cleared, after the current byte transmission is complete, The USART divider and output will stop working.	0
12	M	RW	The character length position. 1: 9 data bits; 0: 8 data bits.	0
11	WAKE	RW	Wake-up bit. This bit determines how the USART is woken up: 1: Address tag; 0: Bus idle.	0
10	PCE	RW parity bit enabled.		0

			1: Enable parity bit; 0: Check bit off. For the receiver, this involves performing parity checks on the data; For the sender, this involves inserting a checksum bit. Once this bit is set, The checksum is enabled only after the current byte transmission is complete. effect.	
9	PS	RW	Parity check selection: 1: Odd parity; 0: Even parity. After this bit is set, the calibration will only be performed after the current byte has been transmitted. Verification is required for the changes to take effect.	0
8	PEIE	RW	Parity check interrupt enable bit. 1: Allow parity check errors to cause interruptions; 0: Prevents parity check error interruption.	0
7	TXEIE	RW	Send buffer empty interrupt enabled: 1: Allow the generation of a transmit buffer empty interrupt; 0: Disables the generation of a send buffer empty interrupt.	0
6	TCIE	RW	Send completion interrupt enabled: 1: Allow transmission completion interruption; 0: Disables the transmission completion interrupt.	0
5	RXNEIE	RW	Receive buffer not empty interrupt enabled: 1: Allow interrupts that prevent the receive buffer from being empty; 0: Disables the generation of non-empty receive buffer interrupts.	0
4	IDLEIE	RW	Bus idle interrupt enabled: 1: Allow bus idle interrupts to be generated; 0: Disable bus idle interrupt generation.	0
3	TE	RW	Send enable. 1: Enable transmitter; 0: Turn off the transmitter.	0
2	RE	RW	Receive enable. 1: Enable the receiver; the receiver begins detecting signals on the RX pin. Start bit; 0: Turn off the receiver.	0
1	RWU	RW	Receive wake-up. This bit determines whether to put the USART into silent mode. Mode: 1: The receiver is in silent mode; 0: The receiver is in normal working mode. Note 1: Place RWU Before a bit is received, the USART needs to receive a data byte; otherwise, it cannot be idled by the bus in silent mode. wake; Note 2: When configured for address tag wake-up, in the case of being set When the RXNE bit is active, the bit cannot be modified by software. RWU	0
0	SBK	RW	Send frame disconnect character control bit. Set this bit to send a frame. Disconnect character. Reset by hardware when the stop bit of the disconnect frame is reached. 1: Send;	0

			0: Do not send.	

19.11.5 USART Control Register 2 (USARTx_CTLR2) (x=1/2/3/4/5/6/7/8) Offset address: 0x10

31	30 29	28	27 26 25 24 23 22 21	2019	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	describe	Reset value
[31:15] Reserved		RO	is reserved.	0
14	LINEN	RW	This is the LIN mode enable bit; setting it enables LIN mode. In LIN... In this mode, the SBK bit can be used to send the LIN synchronization disconnect symbol. The symbol, and the LIN synchronization disconnection symbol.	0
[13:12] STOP[1:0]		RW	Stop bit setting field. These two bits are used to set the stop bit. 00: 1 stop bit; 01: 0.5 stop bits; 10: 2 stop bits; 11: 1.5 stop bits.	00b
11	CLKEN	RW	Clock enable, enable the CK pin. 1: Enable; 0: Prohibited.	0
10	CPOL	RW	Clock polarity setting bit. This bit is used in synchronization mode. Select the polarity of the clock output on the SLCK pin, and the CPHA pin. They work together to generate the required clock/data sampling relationship. 1: The CK pin remains high when the bus is idle; 0: The CK pin remains low when the bus is idle. Note: This bit cannot be modified after sending is enabled.	0
9	CPHA	RW	Clock phase setting bit. This bit is used in synchronization mode. Select the phase of the clock output on the SLCK pin, and CPOL. The bits work together to generate the required clock/data sampling gate. Tie. 1: Data capture is performed on the second edge of the clock; 0: Data capture is performed on the first edge of the clock. Note: This bit cannot be modified after sending is enabled.	0
8	LBCL	RW	Last clock pulse control bit: In synchronization mode, this bit is used to control whether to trigger CK. The foot output corresponds to the last data byte sent. Clock pulse; 1: The clock pulse for the last bit of data is not output from CK; 0: The clock pulse for the last bit of data will be output from CK. Note: This bit cannot be modified after sending is enabled.	0

7	Reserved	RW	reserved.	0
6	LBDIE	RW	LIN Break detects interrupt enable: 1: Enable interrupts caused by LBD; 0: Disable interrupts caused by LBD.	0
5	LBDL	RW	LIN Break detection length; this bit is used to select whether it is 11 bits or... It is a 10-bit breakpoint detection. 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 field, used to set the USART node address of this device. Used in silent mode under multiprocessor communication. Address markers are used to wake up a USART device.	0

19.11.6 USART Control Register 3 (USARTx_CTLR3) (x=1/2/3/4/5/6/7/8) Offset address: 0x14

31 30 29	28	27 26 25 24 23	twenty two	twenty one	20	19	18	17	16
Reserved									

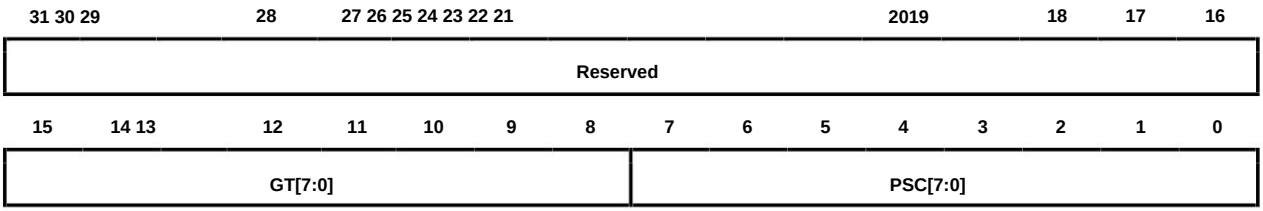
15 14 13	12	11	10	9	8	7	6	5	4	3	2	1	0
LPWKUP_DLY_CFG [2:0]	LPWKUP_CK_SRC	LPWKUP_EN	CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HDSEL	IRLP	IREN	EIE

Bit	name	access	describe	Reset value
[31:16]	Reserved	RO	reserved.	0
[15:13]	LPWKUP_DLY_CFG [2:0]	RW	Low-power wake-up receive delay configuration: 000: No delay; 001: Delay by 1 low-power wake-up clock source cycle; … 111: Delayed by 7 low-power wake-up clock cycles.	0
12	LPWKUP_CK_SRC	RW	Low-power wake-up clock source selection: 1: LSE; 0: LSI.	0
11	LPWKUP_EN	RW	Low-power wake-up enable: 1: Enable low-power wake-up; 0: Disable low-power wake-up.	0
10	CTSIE	RW	CTSIE interrupt enable bit: 1: An interrupt will be generated when CTS is set; 0: No interrupt is generated when CTS is set.	0
9	CTSE	RW	CTS Enable Bit: 1: Enable CTS flow control; 0: Disable CTS flow control.	0
8	RTSE	RW	RTS enable bit: 1: Enable RTS flow control;	0

			0: Turn off RTS flow control.	
7	DMAT	RW	DMA transmit enable bit: 1: Use DMA when sending; 0: DMA is not used during transmission.	0
6	DMAR	RW	DMA receive enable bit: 1: Use DMA during reception; 0: DMA is not used during reception.	0
5	SCEN	RW	Smart card mode enable bit: 1: Enable smart card mode; 0: Disable smart card mode.	0
4	NACK	RW	Smart card NACK enable bit: 1: When a verification error occurs, send a NACK; 0: Do not send NACK when a checksum error occurs.	0
3	HDSEL	RW	Half-duplex mode selection bit: 1: Enable half-duplex mode; 0: Disable half-duplex mode.	0
2	IRLP	RW	Infrared low power selection bit: 1: When selecting infrared, enable low power mode; 0: When selecting infrared, turn off low power mode.	0
1	IREN	RW	Infrared enable bit: 1: Enable infrared mode; 0: Infrared mode off.	0
0	EIE	RW	Error interrupt bit enable: 1: Enable error interrupt; 0: Disable error interruption. After setting this bit, provided that DMAR is set, if FE, An interrupt will be generated when ORE or NE is set.	0

19.11.7 USART Guard Time and Prescaler Register (USARTx_GPR) (x=1/2/3/4/5/6/7/8)

Offset address: 0x18



Bit	name	access	describe	Reset value
[31:16] Reserved		RO	Reserved.	0
[15:8] GT[7:0]		RW	Protects the time value range. This field specifies the time range in baud rate clocks. The protection time of the bit. In smart card mode, when the protection time... Only after the sending is completed will a sending completion flag be set.	0
[7:0] PSC[7:0]		RW	Prescaler value range. In infrared low-power mode, the source clock is controlled by this value (all 8). (Number valid) frequency division, a value of 0 indicates reservation;	0

			In normal infrared mode, this bit can only be set to 1; In smart card mode, the source clock is determined by this value (lower 5 bits are valid). The frequency is divided by twice to provide a clock for the smart card, with a value of 0. This is reserved.	

19.11.8 USART Control Register 4 (USARTx_CTLR4) (x=1/2/3/4/5/6/7/8) Offset address: 0x1C

31 30 29	28	27 26 25 24 23 22 21	2019	18	17	16
Reserved						

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

Bit	name	access	describe	Reset value
[31:4] Reserved		RO reserved.		0
[3:2] CHECK_SEL[1:0]		RW	Verification function selection bit: 0x: Disable MARK and SPACE verification; 10: Enable MARK verification; 11: Enable SPACE verification.	0
1	MS_ERRIE	RW	SPACE or MARK check enable bit: 1: Allow SPACE or MARK checksum errors to occur and cause an interrupt; 0: Prevent SPACE or MARK checksum error interrupts.	0
0	Reserved	RO is reserved.		0

Chapter 20 Serial-to-Parallel Conversion Controller and Transceiver (SerDes)

The module descriptions in this chapter apply only to the CH32H417 microcontroller product.

The system incorporates a set of SerDes modules that support signal isolation and long-distance transmission. These modules include two independent SerDes controllers.

The device supports 1.5Gbps high-speed differential signals (SERDES_RXP, SERDES_RXN, SERDES_TXP, SERDES_TXN pins) and can...

Long-distance data transmission is achieved through a differential peer-to-peer transmission medium in a fiber optic module or network cable.

The base address configurations for the two controllers are as follows:

SerDes Controller 1 Base Address: 0x40027C00

SerDes controller 2 base address: 0x40027C40

20.1 Main Features

Programmable data transmission and reception rates, supporting up to 1.5Gbps .

Supports both high-voltage signal isolation via network transformers and low-voltage isolation via capacitors.

Built -in 8-bit/10-bit encoding/decoding and CRC checksum, supporting serial number matching.

Built- in FIFO, supports double-buffered transmit and receive modes.

Supports DMA functionality; access addresses support byte alignment.

Provides various transmission interruption flags and statuses to promptly feed back information to the application layer .

Differential transceiver can directly drive fiber optic modules.

20.2 Functional Description

20.2.1 Sending Mode (Query Mode) 1. Set the

SERDESx_CTRL register

ALIGN_EN/PWR_UP/TX_PWR_UP/LINK_RESET/ALL_CLR is 1;

2. Wait for PLL lock (PLL_LOCK=1);

3. Configure the SERDESx_CTRL register

ALIGN_EN/PWR_UP/TX_PWR_UP/INT_BUSY/DMA_EN/TX_EN is 1;

4. Set LINK_INIT to 1, start sending the initialization packet, delay for 100ms (software delay), and then clear LINK_INIT;

5. Set the send DMA address (single-buffer configuration DMA0 is sufficient);

6. Configure SERDESx_DATA0 (single-buffered mode);

7. Configure SERDESx_TX_LEN;

8. Query if LINK_FREE is 1 (optional);

9. Set TX_VLD to 1 and start sending (the order of steps 5, 6, and 7 can be arbitrary);

10. If the interrupt flag for transmission completion is 1, stop transmission; if TX_VLD is 0, clear the interrupt flag.

Note: The software delay time in the steps is sufficient to ensure the receiver is ready to receive.

20.2.2 Receive Mode (Interrupt Mode) 1. Set the

SERDESx_CTRL register

PWR_UP/RX_PWR_UP/LINK_RESET/ALL_CLR is 1;

2. Wait for PLL lock (PLL_LOCK=1);

3. Configure the SERDESx_CTRL register

PWR_UP/TX_PWR_UP/DMA_EN/RX_EN is 1;

4. Set the receive DMA address (double buffering mode is recommended);

5. Enable interrupts RECV_DONE_IE/COMINIT_IE/FIFO_OV_IE;

6. After receiving data, query RECV_CRC_OK/RX_SEQ_MATCH. If double-buffered mode is used, query RX_SEQ_NUM for the lowest value.
The bit indicates whether the register used by the currently received packet is DATA0/DMA0/RX_LEN0 or DATA1/DMA1/RX_LEN1;
RX_SEQ_NUM[0] being 1 indicates that the current received packet uses DATA0/DMA0/RX_LEN0;
RX_SEQ_NUM[0] is 0, indicating that the current received packet uses DATA1/DMA1/RX_LEN1.

Note: RX_SEQ_NUM is automatically incremented by 1 after reception is complete, so the RX_SEQ_NUM value obtained after reception is completed is decremented by 1 to represent the current received packet.
of SEQ_NUM

20.2.3 Software Notes: 1. The internal
clock frequency SERDES_CLK of the controller is the PLL frequency divided by 40. For example, SERDES_CLK is 30MHz when the frequency is 1.2GHz.
2. The return values of LINK_RESET, RX_EN, and TX_VLD are synchronized via SERDES_CLK, so there will be a delay in reading them.
3. The length of the data packets sent or received does not include the 4-byte length of DATA0/DATA1;
4. After sending is complete, TX_VLD should be cleared first. Only after the return value of this bit is 0 can the sending completion interrupt flag be cleared; otherwise, asynchronous interruptions will occur.
question.

20.3 Register Description

Table 20-1 List of SERDES1 related registers

name	Access address	describe	Reset value
R32_SERDES1_CTRL	0x40027C00	SERDES1 control register	0x00041207
R32_SERDES1_INT_EN	0x40027C04	SERDES1 Interrupt Enable Register	0x00000000
R32_SERDES1_INT_FS	0x40027C08	SERDES1 Interrupt Flag/Status Register 0x00000000	
R32_SERDES1_RTX_CTRL	0x40027C0C	SERDES1 Transceiver Controller	0x00000000
R32_SERDES1_RX_LEN0	0x40027C10	SERDES1 Receive Length 0 Register	0x00000000
R32_SERDES1_DATA0	0x40027C14	SERDES1 Data 0 Buffer Register	0xXXXXXXXX
R32_SERDES1_DMA0	0x40027C18	The starting address register of DMA0 buffer is 0xXXXXXXXX.	
R32_SERDES1_RX_LEN1	0x40027C1C	SERDES1 Receive Length 1 Register	0x00000000
R32_SERDES1_DATA1	0x40027C20	SERDES1 Data 1 Buffer Register	0xXXXXXXXX
R32_SERDES1_DMA1	0x40027C24	The starting address register of DMA1 buffer is 0xXXXXXXXX.	

Table 20-2 List of SERDES2 related registers

name	Access address	describe	Reset value
R32_SERDES2_CTRL	0x40027C40	SERDES2 control register	0x00041207
R32_SERDES2_INT_EN	0x40027C44	SERDES2 Interrupt Enable Register	0x00000000
R32_SERDES2_INT_FS	0x40027C48	SERDES2 Interrupt Flag/Status Register 0x00000000	
R32_SERDES2_RTX_CTRL	0x40027C4C	SERDES2 transceiver controller	0x00000000
R32_SERDES2_RX_LEN0	0x40027C50	SERDES2 Receive Length 0 Register	0x00000000
R32_SERDES2_DATA0	0x40027C54	SERDES2 Data 0 Buffer Register	0xXXXXXXXX
R32_SERDES2_DMA0	0x40027C58	The starting address register of DMA0 buffer is 0xXXXXXXXX.	
R32_SERDES2_RX_LEN1	0x40027C5C	SERDES2 Receive Length 1 Register	0x00000000
R32_SERDES2_DATA1	0x40027C60	SERDES2 Data 1 Buffer Register	0xXXXXXXXX
R32_SERDES2_DMA1	0x40027C64	The starting address register of DMA1 buffer is 0xXXXXXXXX.	

Table 20-3 List of SERDES Related Registers

name	Access address description 0x5003C010 System	Reset value
R32_SYS_CFGR	Configuration Register	0x0054422B

20.3.1 SERDESx Control Register (R32_SERDESx_CTRL) (x=1/2) Offset Address: 0x00

31	30	29	28	27	26	25	twenty four	twenty three	twenty two	twenty one	20	19	18	17	16
Reserved													ALIGN _EN	CONT_ EN	Reser ved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_PW R_UP	RX_PW R_UP	Reserved						DMA_E N	TX_EN	RX_EN	RX_PO LARIT Y	INT_B USY_E N	RESET _PHY	RESET _LINK	CLR_A LL

Bit	name	access	describe	Reset value
[31:19] Reserved		RO is reserved.		0
18	ALIGN_EN	RW	An ALIGN signal is inserted during transmission to address frequency offset. efficient.	1
17	CONT_EN	RW	The SYNC message sent during idle periods will be replaced with CONT and scrambling data. Replace it to reduce EMI.	0
16	Reserved	RO reserved.		0
15	TX_PWR_UP	RW	Transmitter module power-on enable bit: 1: Enable; 0: Off.	0
14	RX_PWR_UP	RW	Receiver module power-on enable bit: 1: Enable; 0: Off.	0
[13:8] Reserved		RO is reserved.		0x12
7	DMA_EN	RW	DMA Enable: 1: Enable; 0: Off.	0
6	TX_EN	RW	Send Enable: 1: Enable; 0: Off.	0
5	RX_EN	RW	Receive Enable: 1: Enable; 0: Off.	0
4	RX_POLARITY	RW RXN/RXP signal switching.		0
3	INT_BUSY_EN	RW	SERDES will automatically pause until the interruption flag for transmission completion is cleared. Send enable bit: 1: Enable; 0: Off.	0
2	RESET_PHY	RW	SERDES Physical Layer Software Reset Control Bits: 1: Enable;	1

			0: Off.	
1	RESET_LINK	RW	SERDES Link Layer Software Reset Control Bits: 1: Enable; 0: Off.	1
0	CLR_ALL	RW	Clear the SERDES interrupt flag and FIFO: 1: Enable; 0: Off.	1

20.3.2 SERDESx Interrupt Enable Register (R32_SERDESx_INT_EN) (x=1/2)

Offset address: 0x04

31	30	29	28	27	26	25	twenty four	twenty three	twenty two	twenty one	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										COMINIT_IE	Reserved	FIFO_OV_IE	RECV_DONE_IE	TRAN_DONE_IE RECV_ERR_IE	PHYRDY_IE

Bitname		access	describe	Reset value
[31:6] Reserved		RO is reserved.		0
5	COMINIT_IE	RW	COMINIT receive interrupt enabled: 1: Enable; 0: Off.	0
4	Reserved	RO is reserved.		0
3	FIFO_OV_IE	RW	FIFO overflow interrupt enable: 1: Enable; 0: Off.	0
2	RECV_DONE_IE	RW	Receive completion interrupt enable: 1: Enable; 0: Off.	0
1	TRAN_DONE_IE RECV_ERR_IE	RW	Sending mode: Send completion interrupt enable: 1: Enable; 0: Off. Receive mode: Receive CRC error interrupt enabled: 1: Enable; 0: Off.	0
0	PHYRDY_IE	RW	Physical layer READY interrupt enable: 1: Enable; 0: Off.	0

20.3.3 SERDESx Interrupt Flag/Status Register (R32_SERDESx_INT_FS) (x=1/2) Offset Address: 0x08

31	30	29	28	27	26	25	twenty four	twenty three	twenty two	twenty one	20	19	18	17	16
TX_SEQ_NUM[3:0]				RX_SEQ_NUM[3:0]			Reserved			R_FIFO_RDY	LINK_FREE	PLL_LOCK	RECV_CRC_OK	RX_SEQ_MATCH	PHYRDY

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										COMINIT_IF	Reserved	FIFO_OV_IF	RECV_DONE_IF	TRAN_DONE_IF RECV_ERR_IF	PHYRDY_IF

Bit	name	Access description R0 LINK_INIT is automatically		Reset value
[31:28] TX_SEQ_NUM[3:0]		cleared when it is valid.		0
[27:24] RX_SEQ_NUM[3:0]		In RO receive mode, COMINT is automatically cleared upon receipt.		0
[23:22] Reserved		RO is reserved.		0
twenty one	R_FIFO_RDY	RO	SERDES receive FIFO data ready status: 1: The receive FIFO is not empty; 0: Receive FIFO empty.	0
20	LINK_FREE	RO SERDES: Link layer sends idle status bits.		0
19	PLL_LOCK	R0 is the PLL lock status bit. Transmission and reception will begin when this bit is 1.		0
18	RECV_CRC_OK	RO	Receive data frame check status bit, RECV_DONE interrupt check Measure this bit and update it after packet reception is complete.	0
17	RX_SEQ_MATCH	R0 receives the SEQ_NUM match status bit.		0
16	PHYRDY	RO Physical layer READY status bit.		0
[15:6] Reserved		RO is reserved.		0
5	COMINIT_IF	RW1Z COMINIT receive interrupt flag.		0
4	Reserved	RO is reserved.		0
3	FIFO_OV_IF	RW1Z FIFO overflow interrupt flag.		0
2	RECV_DONE_IF	RW1Z receive complete interrupt flag.		0
1	TRAN_DONE_IF RECV_ERR_IF	RW1Z	Transmit mode: transmit complete interrupt flag. Receive mode: Receive CRC error interrupt flag.	0
0	PHYRDY_IF	RW1Z Physical layer READY interrupt flag.		0

20.3.4 SERDESx Transceiver Controller (R32_SERDESx_RTX_CTRL) (x=1/2)

Offset address: 0x0C

31	30	29	28	27	26	25	twenty four	twenty three	twenty two	twenty one	20	19	18	17	16
Reserved													BUF_MODE	TX_VLD	LINK_INIT

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

Bit	name	access	describe	Reset value

[31:19] Reserved		RO is reserved.		0
18	BUF_MODE	RW	Double-buffered control bits: 1: Enable; 0: Off.	0
17	TX_VLD	RW	If the data packet is valid, it will be cleared by the software, and this bit will return a sum. PHY_CLK synchronization.	0
16	LINK_INIT	When the RW bit is 1, a LINK initialization packet is		0
[15:0]	SERDES_TX_LEN [15:0]	RW	sent. The number of data bytes sent is fixed at 0 for the lower 2 bits (4 bytes). (Alignment)	X

20.3.5 SERDESx Receive Length 0 Register (R32_SERDESx_RX_LEN0) (x=1/2) Offset Address: 0x10

31	30	29	28	27	26	25	twenty four	twenty three	twenty two	twenty one	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SERDES_RX_LEN0[15:0]															

Bitname		access	describe	Reset value
[31:16] Reserved		RO is reserved.		0
[15:0]	SERDES_RX_LEN0 [15:0]	RO	The length of the last received data packet, with the lowest 2 bits fixed. 0. In non-double buffered mode, this indicates the length of the currently received packet; In double-buffered mode, RX_SEQ_NUM is an even number of packet lengths.	0

20.3.6 SERDESx Data 0 Buffer Register (R32_SERDESx_DATA0) (x=1/2) Offset Address: 0x14

31	30	29	28	27	26	25	twenty four	twenty three	twenty two	twenty one	20	19	18	17	16
SERDES_DATA0[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SERDES_DATA0[15:0]															

Bit	name	access	describe	Reset value
[31:0]	SERDES_DATA0 [31:0]	RW	SERDES Data Buffer 0 Send Mode: LINK First 4 Bytes of Data Buffer for Sending and Receiving The high 4 bits of this register are replaced with TX_SEQ_NUM, the user can use the lower 28 bits as the default value. Define information. Receive mode: The first 4 bytes of the data packet are stored in the register. The data is stored in the register, and the subsequent data is stored in the DMA.	X

20.3.7 The starting address register of DMA0 buffer (R32_SERDESx_DMA0) (x=1/2) offset address: 0x18

31	30	29	28	27	26	25	twenty four	twenty three	twenty two	twenty one	20	19	18	17	16
SERDES_DMA0[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SERDES_DMA0[15:0]															

Bit	name	access		Reset value
[31:0]	SERDES_DMA0 [31:0]	RW	Describe the starting address of buffer 0. In receive mode, the software needs... Fix the lower 4 bits to 0 (16-byte aligned), and send the mode. The formula does not require 16-byte alignment. Note: 1. In single-buffer mode, only DMA0 is used; 2. In double buffer mode Use this method when TX_SEQ_NUM/RX_SEQ_NUM are even numbers. DMA0.	X

20.3.8 SERDESx Receive Length 1 Register (R32_SERDESx_RX_LEN1) (x=1/2) Offset Address: 0x1C

31	30	29	28	27	26	25	twenty four	twenty three	twenty two	twenty one	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SERDES_RX_LEN1[15:0]															

Bit	name	access	describe	Reset value
[31:16] Reserved		RO is reserved.		0
[15:0]	SERDES_RX_LEN1 [15:0]	RO	The length of the last received data packet, with the lowest 2 bits fixed. 0. RX_LEN1 is only valid in double-buffered mode. This indicates that RX_SEQ_NUM is an odd number of packet lengths.	0

20.3.9 SERDESx Data 1 Buffer Register (R32_SERDESx_DATA1) (x=1/2) Offset address: 0x20

31	30	29	28	27	26	25	twenty four	twenty three	twenty two	twenty one	20	19	18	17	16
SERDES_DATA1[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SERDES_DATA1[15:0]															

Bit	name	access	describe	Reset value
[31:0]	SERDES_DATA1 [31:0]	RW	SERDES Data Buffer 1 Sending Mode: LINK Storage of the first 4 bytes of data for transmission and reception. In this register, the high 4 bits are replaced with TX_SEQ_NUM. Users can use the lower 28 bits as custom information. Receive mode: The first 4 bytes of the data packet are stored in the register.	X

			The data is stored in the register, and the subsequent data is stored in the DMA. Note: In double-buffered mode, and RX_SEQ_NUM is odd. Use this buffer when counting.	

20.3.10 DMA1 buffer start address register (R32_SERDESx_DMA1) (x=1/2) offset address: 0x24

31	30	29	28	27	26	25	twenty four	twenty three	twenty two	twenty one	20	19	18	17	16
SERDES_DMA1[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SERDES_DMA1[15:0]															

Bit	name	access	describe	Reset value
[31:0]	SERDES_DMA1	RW	Buffer 0 starting address: In receive mode, the software needs to fix the lower 4 bits as 0 (16 Byte alignment); The sending mode does not require 16-byte alignment. Note buffer mode pair TX_SEQ_NUM/RX_SEQ_NUM Use when it is odd DMA.	X

20.3.11 System Configuration Register (R32_SYS_CFGR)

Access address: 0x5003C010

31	30	29	28	27	26	25	twenty four	twenty three	twenty two	twenty one	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		RX_FBCAP_ADJ [1:0]		RX_FBRES_ADJ[1:0]		Reserved		TX_DE_EMPHASIS[1:0]		TX_OUTPUT_SWING [1:0]		Reserved			

Bit	name	access	describe	Reset value
[31:14]	Reserved	RW	is reserved and must retain its original value when written.	0x151
[13:11]	RX_FBCAP_ADJ [1:0]	RW	Receiver feedback capacitor adjustment bit; refer to [reference needed] for detailed parameters. CH32H417DS0.	000b
[10:9]	RX_FBRES_ADJ [1:0]	RW	Receiver feedback resistor adjustment bit; see detailed parameters. CH32H417DS0.	01b
[8:7]	Reserved	RW	is reserved and must retain its original value when writing.	00b
[6:5]	TX_DE_EMPHASIS [1:0]	RW	Send the pre-emphasis adjustment bit; refer to the documentation for detailed parameters. CH32H417DS0.	01b
[4:3]	TX_OUTPUT_SWING [1:0]	RW	Send amplitude adjustment bit , Detailed parameter reference CH32H417DS0.	01b
[2:0]	Reserved	RW	is reserved; the original value must be preserved when writing.	011b

Note: For long-term use, it is recommended to adjust the value of this register to 0x542E7B.

Chapter 21 Two-Wire Communication Bus (I2C)

Internal integrated circuit bus (I2C) is widely used for communication between microcontrollers and sensors and other off-chip modules. It supports multiple...

In master-slave mode, communication can be achieved at two speeds: 100kHz (standard) and 400kHz (fast) using only two wires (SDA and SCL).

The I2C bus is also compatible with the SMBus protocol, supporting not only I2C timing but also arbitration, timing and DMA, and has CRC check functionality.

21.1 Main Features

Supports both master and slave modes .

Supports 7-bit or 10-bit addresses

ÿ The device supports dual 7-bit addresses

Supports two speed modes: 100kHz and 400kHz. Multiple status modes

and error flags are also included .

Supports extended clock functionality

ÿ 2 interrupt vectors

Supports DMA ;

Supports PEC ;

Compatible with SMBus

21.2 Overview

I2C is a half-duplex bus, meaning it can only operate in one of four modes at a time: master transmit mode, master receive mode, master transmit ...

The I2C module operates in slave mode by default, and will automatically switch to slave mode after the start condition is generated.

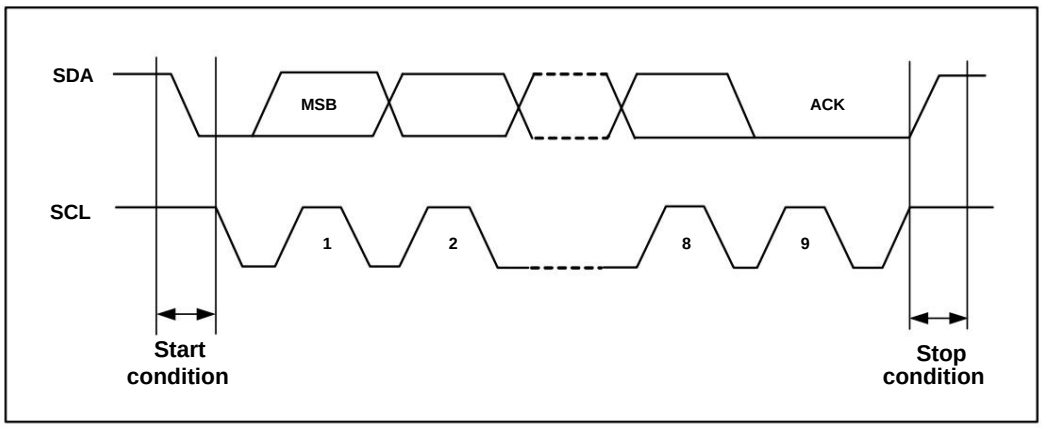
It switches to master mode, and switches to slave mode upon loss of arbitration or the generation of a stop signal. The I2C module supports multi-master functionality. It operates in...

In master mode, the I2C module actively sends data and address. Both data and address are transmitted in 8-bit units, with the most significant bit first and the least significant bit last.

Following the start event, the address is either one byte (in 7-bit address mode) or two bytes (in 10-bit address mode), and the host...

When sending 8 bits of data or address, the slave device needs to reply with an ACK, which involves pulling the SDA bus low, as shown in Figure 21-1.

Figure 21-1 I2C Timing Diagram

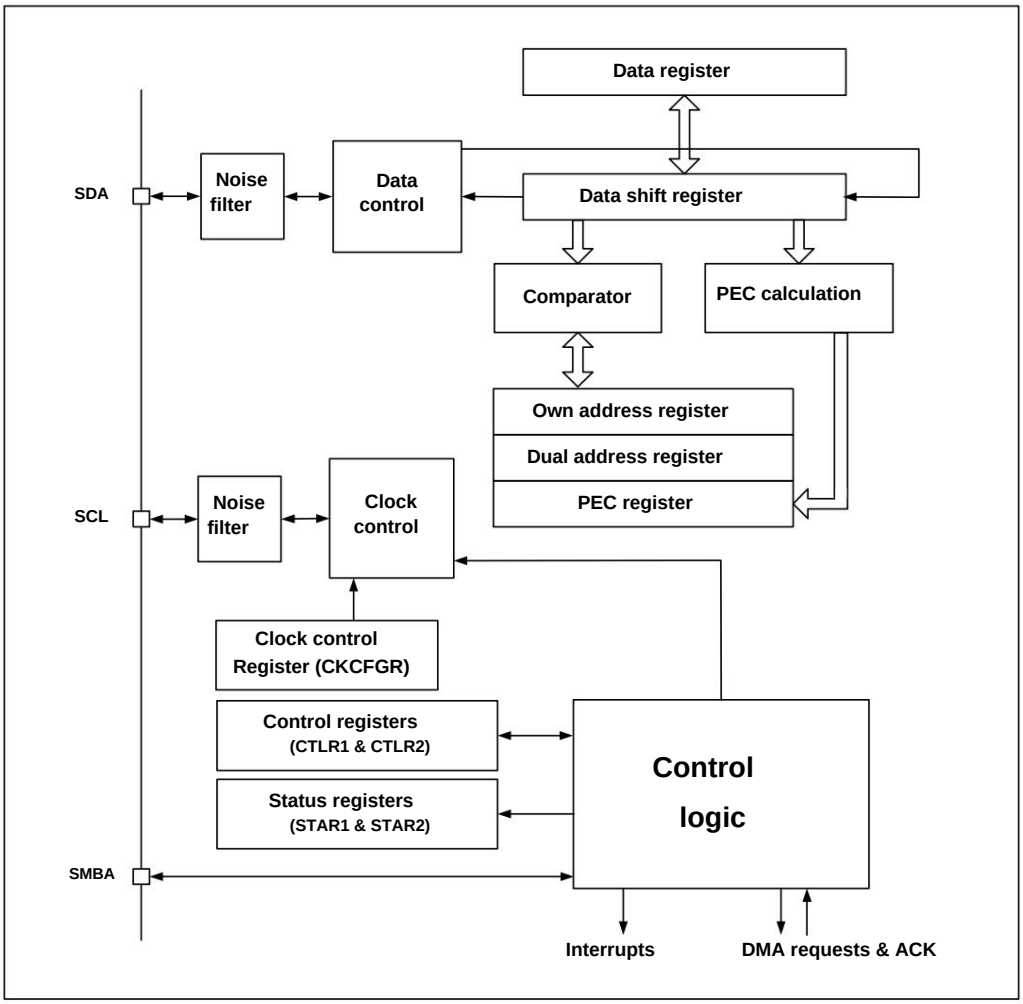


For proper operation, the I2C input must be supplied with the correct clock. In standard mode, the minimum input clock is 2MHz, and in fast mode...

The minimum input clock frequency is 4MHz.

Figure 21-2 is a functional block diagram of the I2C module.

Figure 21-2 I2C Functional Block Diagram



21.3 Main Mode

In master mode, the I2C module dominates data transmission and outputs a clock signal. Data transmission begins with a start event and ends with an end event.

The steps for using master mode communication are as follows:

Set the correct clock in Control Register 2 (R16_I2Cx_CTLR2) and Clock Control Register (R16_I2Cx_CKCFGR); Set the appropriate rising edge in Rising Edge Register (R16_I2Cx_RTR); Set the PE bit in Control Register (R16_I2Cx_CTLR1) to start the peripheral; Set the START bit in Control Register (R16_I2Cx_CTLR1) to generate a start event. After setting the START bit, the I2C module will automatically switch to master mode, the MSL bit will be set, and a start event will be generated. After the start event is generated, the SB bit will be set. If the ITEVTEN bit (in R16_I2Cx_CTLR2) is set, an interrupt will be generated. At this time, the Status Register 1 (R16_I2Cx_STAR1) should be read. After writing the slave address to the data register, the SB bit will be automatically cleared.

If using 10-bit address mode, then write the data register to send the header sequence (the header sequence is 11110xx0b, where xx bits are...). (These are the two highest bits of a 10-bit address).

After the header sequence is sent, the ADD10 bit of the status register will be set. If the ITEVTEN bit is already set, a middle... If the connection is broken, you should read the R16_I2Cx_STAR1 register, write the second address byte to the data register, and then clear the ADD10 bit.

Then, write to the data register and 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 point, the R16_I2Cx_STAR1 register should be read again. The R16_I2Cx_STAR2 register is used to clear the ADDR bit; if 7-bit address mode is used, the data register is written to send the address byte, and the status register is checked after sending the address byte.

The ADDR bit of the device will be set. If the ITEVTEN bit is already set, an interrupt will be generated, and the R16_I2Cx_STAR1 register should be read.

Read the R16_I2Cx_STAR2 register again to clear the 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, and the 8th bit determines the address.

The direction of subsequent messages is determined: 0 indicates that the master device is writing data to the slave device, and 1 indicates that the master device is reading information from the slave device.

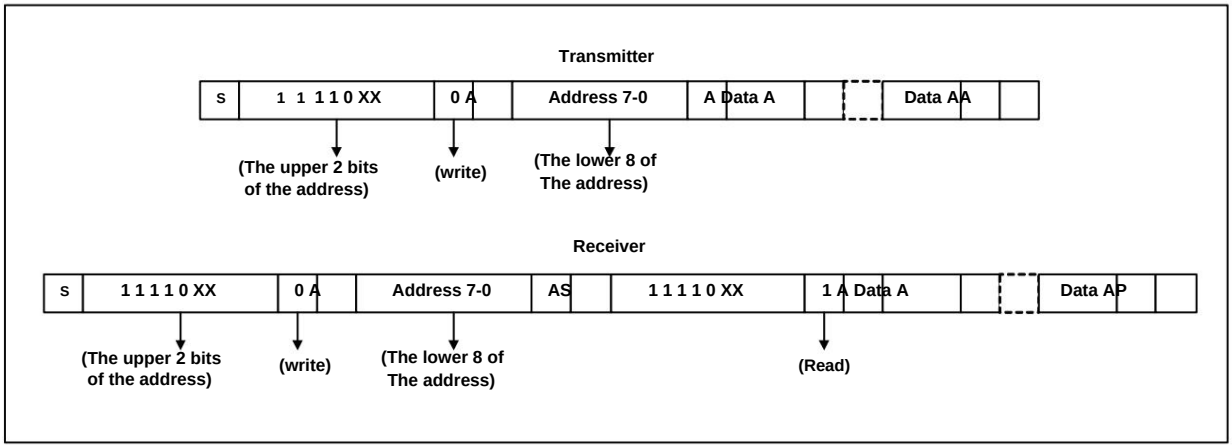
In 10-bit address mode, as shown in Figure 21-3, during the address transmission phase, the first byte is 11110xx0, where xx represents the 10-bit address.

The first byte is the highest 2 bits of the address, and the second byte is the lower 8 bits of the 10-bit address. If the device subsequently enters master transmit mode, it will continue to send data.

If you are preparing to enter master device receive mode, you need to resend a start condition, followed by sending a byte of 11110xx1.

Then it enters the master device receiving mode.

Figure 21-3 Schematic diagram of host data transmission and reception with 10-bit address



Master sending 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, the status register...

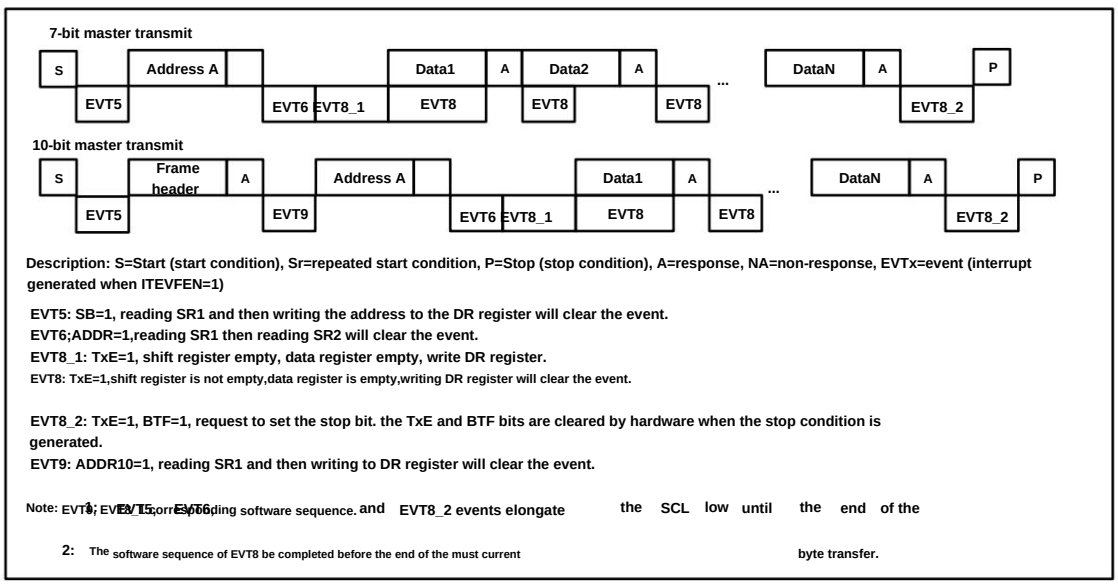
TxE of register 1 (R16_I2Cx_STAR1) is set. An interrupt will also be generated if ITEVTEN and ITBUFEN are set. Data is then transferred.

Writing data to the register will clear the TxE bit.

If the TxE bit is set and no new data has been written to the data register since the last data transmission, then the BTF bit will be set.

SCL will remain low until it is cleared. After reading R16_I2Cx_STAR1, writing data to the data register will clear the BTF bit.

Figure 21-4 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, EVT8_1, EVT8 and EVT8_2 events elongate the SCL low until the end of the corresponding software sequence.

2: The software sequence of EVT8 be completed before the end of the must current byte transfer.

Master Receive Mode:

The I2C module receives data from the SDA line and writes it into the data register via a shift register. After each byte, if the ACK bit...

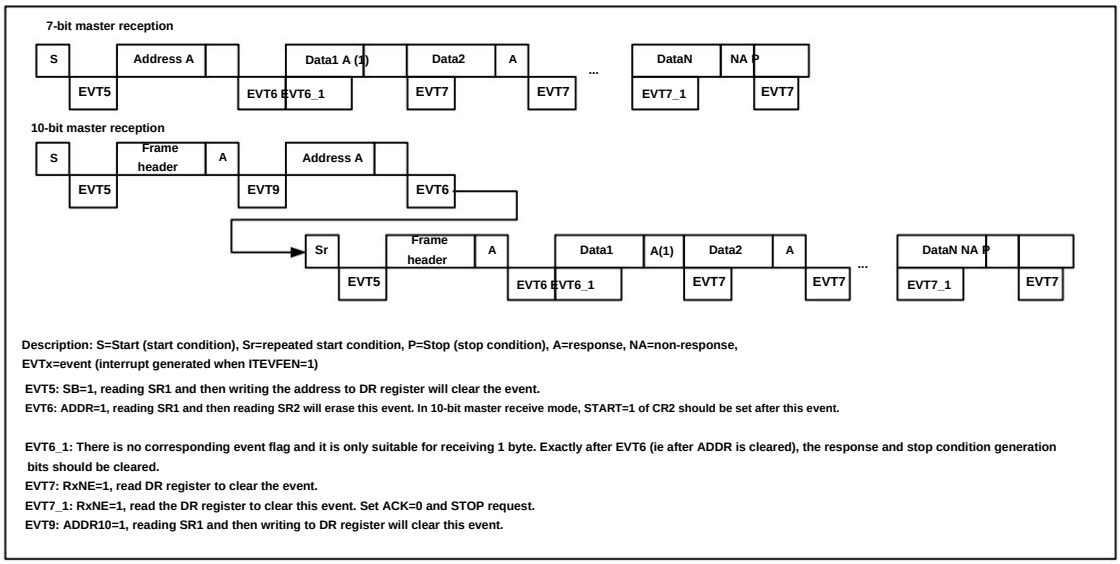
If set, the I2C module will send an acknowledge low level, and the RxNE bit will be set. If ITEVTEN and ITBUFEN are also set...

Setting this bit will also generate an interrupt. If RxNE is set and the existing data has not been read before new data is received, the BTF bit will be set.

This will be set; SCL will remain low until BTF is cleared. After reading R16_I2Cx_STAR1, reading the data register again will clear it.

Except for the BTF bit.

Figure 21-5 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 (ie 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 finishes transmitting data, it will proactively send a termination event, i.e., set the STOP bit, and the I2C will switch to slave mode. During reception...

In this mode, the master device needs to NAK at the acknowledgment position of the last data bit. After receiving NACK, the slave device releases the connection to the SCL and SDA lines.

The master device can then send a stop/restart condition. Note that after a stop condition is generated, the I2C module will automatically switch to [control/control].

From the pattern.

21.4 From the pattern

In slave mode, the I2C module can recognize its own address and the broadcast call address. Software can control whether the broadcast call address is enabled or disabled.

The I2C module identifies the SDA data and, once a start event is detected, moves the data through a shift register along with its own address (the number of bits depends on the address of the shift register).

The address is compared with either ENDUAL or ADDMODE, or the broadcast address (when ENGC is set). If they do not match, they will be ignored until a new start event is generated.

If the address matches the header sequence, an ACK signal is generated and the system waits for the address of the second byte; if the address of the second byte also matches...

If the entire address segment matches in the case of a match or a 7-bit address, then:

First, generate an ACK response;

If the ADDR bit is set, and the ITEVTEN bit is already set, a corresponding interrupt will be generated.

If dual-address mode is used (ENDUAL bit is set), the DUALF bit also needs to be read to determine which one the host is waking up.

address.

The default mode is receive mode, which is indicated by setting the last bit of the received header sequence to 1, or the last bit of the 7-bit address to 1 (depending on the context).

Upon receiving the header sequence (or a standard 7-bit address) for the first time, and upon receiving a repeated start condition, the I2C module will enter the sending phase.

In receiver mode, the TRA bit will indicate whether the current mode is receiver or transmitter.

From sending mode:

After clearing the ADDR bit, the I2C module sends bytes from the data register to the SDA line via a shift register. The slave device holds...

SCL is low until the ADDR bit is cleared and the data to be sent has been written to the data register. (See EVT1 and EVT3 in the diagram below).

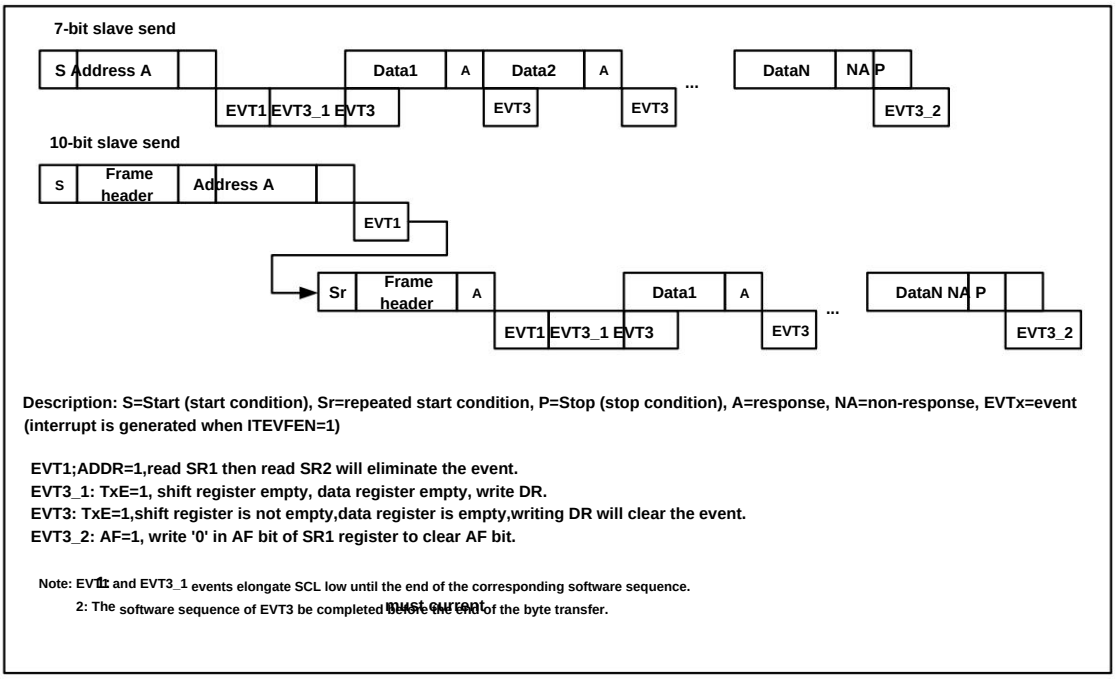
Upon receiving an ACK, the TxE bit will be set. If ITEVTEN and ITBUFEN are set, an interrupt will also be generated.

If TxE is set but no new data is written to the data register before the next data transmission ends, the BTF bit will be set.

Before BTF, SCL will remain low. After reading status register 1 (R16_I2Cx_STAR1), data will be written to the data register.

This will clear the BTF bit.

Figure 21-6 Transmission sequence diagram from the transmitter



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)

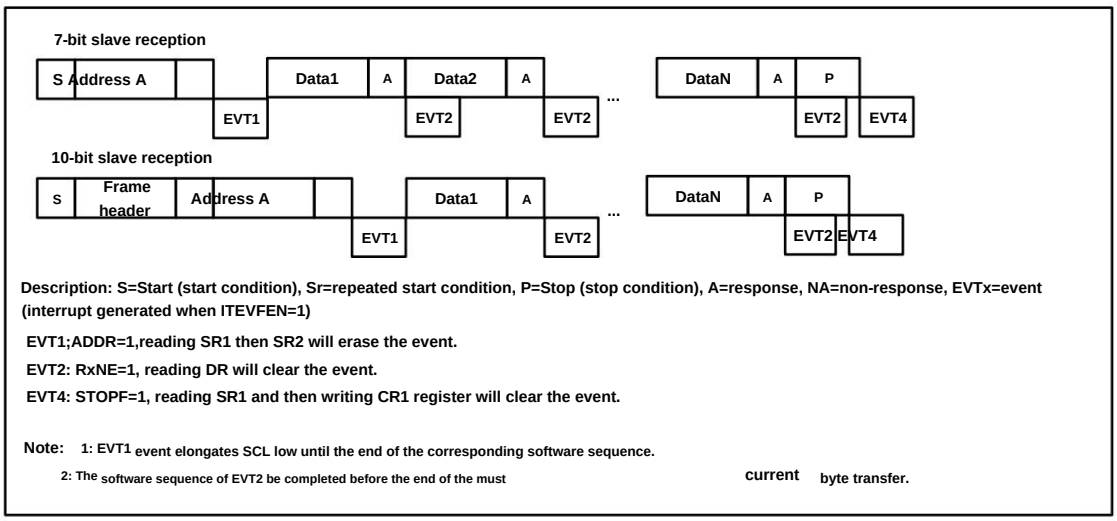
EVT1;ADDR=1,read SR1 then read SR2 will eliminate the event.
EVT3_1: TxE=1, shift register empty, data register empty, write DR.
EVT3: TxE=1,shift register is not empty,data register is empty,writing DR will clear the event.
EVT3_2: AF=1, write '0' in AF bit of SR1 register to clear AF bit.

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

From receive mode:

After ADDR is cleared, the I2C module stores the data on SDA into the data register via a shift register, upon receiving each data input. After each byte, the I2C module sets an ACK bit and the RxNE bit. If ITEVTEN and ITBUFEN are set, an additional ACK bit will be generated. Interruption. If RxNE is set and the old data has not been read before new data is received, then BTF will be set. (In clearing...) SCL will remain low until the BTF bit is reached. Reading status register 1 (R16_I2Cx_STAR1) and the data in the data register will... Clear the BTF bit.

Figure 21-7 Transmission sequence diagram from the receiver



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 be completed before the end of the must current byte transfer.

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

Error 21.5

21.5.1 Bus Error BERR

During address or data transmission, if the I2C module detects an external start or stop event, a bus error will be generated.

In the event of a bus error, the BERR bit is set, and if ITERREN is set, an interrupt will also be generated. In slave mode, data is discarded, and hard disk access is interrupted.

The device releases the bus. If it's a start signal, the hardware interprets it as a reset signal and begins waiting for an address or stop signal; if it's a stop signal,

Then, proceed according to the normal stop conditions. In main mode, the hardware will not release the bus, nor will it affect the current transmission; this is handled by the user code.

Decide whether to suspend the transmission.

21.5.2 Acknowledgment Error (AF)

An acknowledgment error will occur when the I2C module detects a byte but fails to acknowledge it. When an acknowledgment error occurs: AF will be set, such as...

Setting ITERREN will also generate an interrupt; if an AF error occurs and the I2C module is operating in slave mode, the hardware must release the master block.

If the line is in main mode, the software must generate a stop event.

21.5.3 Arbitration lost ARLO

When the I2C module detects a lost arbitration, it generates a lost arbitration error. When a lost arbitration error occurs: the ARLO bit is set, if...

Setting ITERREN will also generate an interrupt; the I2C module switches to slave mode and no longer responds to transfers initiated for its slave address.

Unless a new start event is initiated by the host, the hardware will release the bus.

21.5.4 Overload/Underload Error

OVR ÿ Overload Error:

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

However, if the previously received data has not yet been read, an overload error will occur. When an overload error occurs, the last received byte will be lost.

If the request is rejected, the sender should retransmit the last byte sent.

Underload error:

In slave mode, if clock extension is disabled, the I2C module is sending data, and if a new byte's clock signal arrives before the next byte's clock signal arrives...

If the data has not yet been written to the data register, an underload error will occur. When an underload error occurs, the data in the previous data register...

The data will be sent twice, and if an underrun error occurs, the receiver should discard the duplicate data. To prevent underrun errors...

Incorrect. The I2C module should write data to the data register before the first rising edge of the next byte.

21.6 Clock Extension

If clock extension is disabled, there is a possibility of overload/underload errors. However, if clock extension is enabled:

In transmit mode, if TxE and BTF are both set, SCL will remain low, waiting for the user to read the status register .

Write the data to be sent into the data register;

In receive mode, if RxNE and BTF are both set, SCL will remain low after receiving data until the user reads it.

Check the status register and read the data register;

Therefore, extending the enable clock can prevent overload/underload errors.

21.7 SMBus

SMBus is also a two-wire interface, typically used between the system and power management interfaces. SMBus and I2C share many similarities.

For example, SMBus uses the same 7-bit address mode as I2C. The following are the similarities between SMBus and I2C:

1) Master-slave communication mode, where the master provides the clock and supports multiple masters and multiple slaves;

2) Two-wire communication structure, where SMBus can optionally have one warning line;

3) Both support 7-bit address format.

SMBus and I2C also have differences:

1) I2C supports a maximum speed of 400kHz, while SMBus supports a maximum of 100kHz, and SMBus has a minimum speed limit of 10kHz.

system;

2) SMBus will report a timeout if its clock value is lower than 35ms, but I2C has no such limitation;

3) SMBus has a fixed logic level, while I2C does not, and depends on VDD33;

4) SMBus has a bus protocol, while I2C does not.

SMBus also includes device identification, address resolution protocol, unique device identifier, SMBus alerts, and various bus protocols, specifically...

Please refer to SMBus specification version 2.0. When using SMBus, simply set the SMBus bit in the control register and configure SMBTYPE as needed.

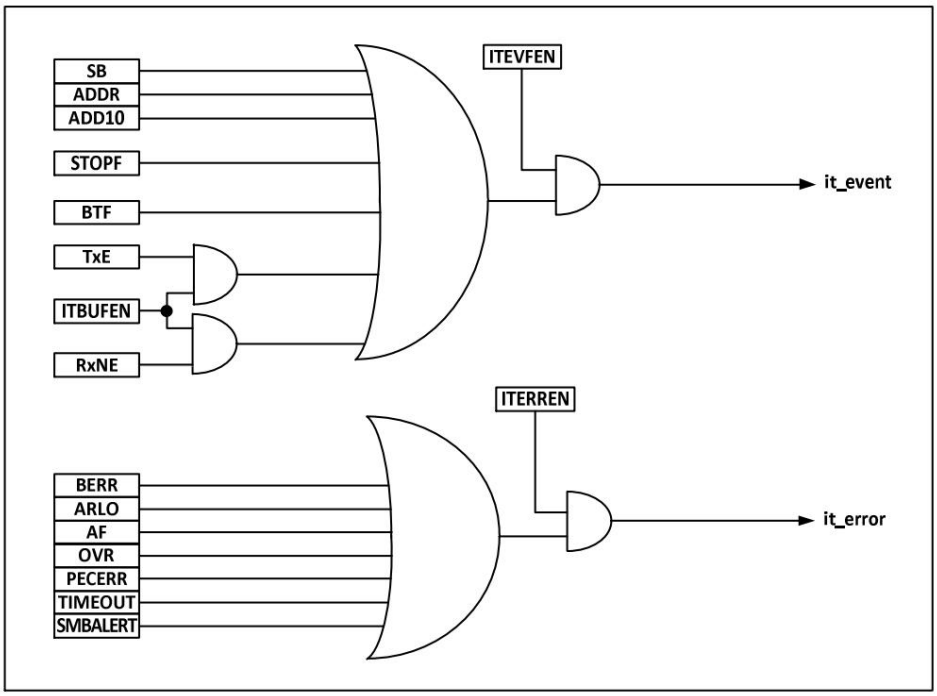
Bit and ENAARP bit.

21.8 Interruption

Each I2C module has two interrupt vectors: event interrupt and error interrupt. These two interrupts support the interrupts shown in Figure 21-8.

source.

Figure 21-8 I2C Interrupt Request



21.9 DMA

DMA can be used for sending and receiving bulk data. When using DMA, the ITBUFEN bit in the control register cannot be set.

• DMA transfer can be activated

by setting the DMAEN bit in the CTLR2 register. As long as the TxE bit is set, data will be transferred from the device via DMA.

The allocated memory is loaded into the I2C data register. The following settings are required to allocate a channel for I2C.

1) Set the I2Cx_DATAR register address in the DMA_PADDRx register and the memory address in the DMA_MADDRx register.

Thus, after each TxE event, data will be transferred from memory to the I2Cx_DATAR register.

2) Set the desired number of bytes to transfer 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 configure it to be set when half or all of the transfer is completed, depending on application requirements.

Issue an interrupt request.

5) Activate the channel by setting the EN bit on the DMA_CFGRx register.

When the number of bytes of data transfer set in the DMA controller has been completed, the DMA controller sends a transfer completion notification to the I2C interface.

The EOT/EOT_1 signal. If interrupts are enabled, a DMA interrupt will be generated.

To use DMA receive mode,

set the DMAEN flag in the CTLR2 register. When using DMA receive, the DMA will transfer the data from the data register...

Data is transferred to a predefined memory area. The following steps are required to allocate a channel for I2C.

1) Set the I2Cx_DATAR register address in the DMA_PADDRx register and the memory address in the DMA_MADDRx register.

Thus, after each RxNE event, data will be written from the I2Cx_DATAR register to memory.

2) Set the desired number of bytes to transfer in the DMA_CNTRx register. This value will be decremented after each RxNE event.

3) Configure the channel priority using PL[0:1] in the DMA_CFGRx register.

4) Clear the DIR bit in the DMA_CFGRx register. Depending on application requirements, this can be set to clear the DIR bit when data transfer is halfway or fully completed.

An interrupt request is issued.

5) Set the EN bit in the DMA_CFGRx register to activate the channel.

When the number of bytes of data transfer set in the DMA controller has been completed, the DMA controller sends a transfer completion notification to the I2C interface.

The EOT/EOT_1 signal. If interrupts are enabled, a DMA interrupt will be generated.

21.10 Packet verification error

Packet Error Checking (PEC) is a step that adds a CRC8 checksum to provide reliability for transmission. It uses the following polynomial to check each packet error.

Calculations are performed using a single serial data bit:

The PEC

calculation, C = X8 + X2 + X+1, is activated by the ENPEC bit in the control register and is performed on all information bytes, including address and read/write bits.

When transmitting, enabling PEC will append a CRC8 hash to the last byte of data; while in receive mode, it will append a CRC8 hash to the last byte of data.

The next byte is considered the CRC8 checksum result. If it does not match the internal calculation result, a NAK will be returned. If it is the main receiver...

The device will respond with a NAK regardless of whether the verification result is correct or not.

21.11 Debug Mode

Once the system enters debug mode, the I2CSMBus can be determined using the DBG_I2Cx_SMBUS_TIMEOUT bit of the DEBUG module.

The timeout control determines whether to continue working or stop.

21.12 Register Description

Table 21-1 List of I2C1 Related Registers

name	Access address description		Reset value
R16_I2C1_CTLR1	0x40005400	I2C1 Control Register 1	0x0000
R16_I2C1_CTLR2	0x40005404	I2C1 Control Register 2	0x0000
R16_I2C1_OADDR1	0x40005408	I2C1 Address Register 1	0x0000
R16_I2C1_OADDR2	0x4000540C	I2C1 Address Register 2	0x0000
R16_I2C1_DATAR	0x40005410	I2C1 Data Register	0x0000
R16_I2C1_STAR1	0x40005414	I2C1 Status Register 1	0x0000
R16_I2C1_STAR2	0x40005418	I2C1 Status Register 2	0x0000
R16_I2C1_CKCFGR	0x4000541C	I2C1 Clock Register	0x0000

R16_I2C1_RTR	0x40005420	I2C1 Rise Time Register	0x0002

Table 21-2 List of I2C2 Related Registers

name	Access address	describe	Reset value
R16_I2C2_CTLR1	0x40005800	I2C2 Control Register 1	0x0000
R16_I2C2_CTLR2	0x40005804	I2C2 Control Register 2	0x0000
R16_I2C2_OADDR1	0x40005808	I2C2 Address Register 1	0x0000
R16_I2C2_OADDR2	0x4000580C	I2C2 Address Register 2	0x0000
R16_I2C2_DATAR	0x40005810	I2C2 Data Register	0x0000
R16_I2C2_STAR1	0x40005814	I2C2 Status Register 1	0x0000
R16_I2C2_STAR2	0x40005818	I2C2 Status Register 2	0x0000
R16_I2C2_CKCFGR	0x4000581C	I2C2 Clock Register	0x0000
R16_I2C2_RTR	0x40005820	I2C2 Rise Time Register	0x0002

Table 21-3 List of I2C3 Related Registers

name	Access address description		Reset value
R16_I2C3_CTLR1	0x40005C00	I2C3 Control Register 1	0x0000
R16_I2C3_CTLR2	0x40005C04	I2C3 Control Register 2	0x0000
R16_I2C3_OADDR1	0x40005C08	I2C3 Address Register 1	0x0000
R16_I2C3_OADDR2	0x40005C0C	I2C3 Address Register 2	0x0000
R16_I2C3_DATAR	0x40005C10	I2C3 Data Register	0x0000
R16_I2C3_STAR1	0x40005C14	I2C3 Status Register 1	0x0000
R16_I2C3_STAR2	0x40005C18	I2C3 Status Register 2	0x0000
R16_I2C3_CKCFGR	0x40005C1C	I2C3 Clock Register	0x0000
R16_I2C3_RTR	0x40005C20	I2C3 Rise Time Register	0x0002

Table 21-4 List of I2C4 Relevant Registers

name	Access address description		Reset value
R16_I2C4_CTLR1	0x40014000	I2C4 Control Register 1	0x0000
R16_I2C4_CTLR2	0x40014004	I2C4 Control Register 2	0x0000
R16_I2C4_OADDR1	0x40014008	I2C4 Address Register 1	0x0000
R16_I2C4_OADDR2	0x4001400C	I2C4 Address Register 2	0x0000
R16_I2C4_DATAR	0x40014010	I2C4 Data Register	0x0000
R16_I2C4_STAR1	0x40014014	I2C4 Status Register 1	0x0000
R16_I2C4_STAR2	0x40014018	I2C4 Status Register 2	0x0000
R16_I2C4_CKCFGR	0x4001401C	I2C4 Clock Register	0x0000
R16_I2C4_RTR	0x40014020	I2C4 Rise Time Register	0x0002

21.12.1 I2C Control Register (I2Cx_CTLR1) (x=1/2/3/4) Offset Address: 0x00 Name

Bit		access	This	Reset value
15	SWRST	RW	describes a software reset; setting this bit in user code will reset the I2C peripheral. Before resetting, ensure that the I2C bus pins are released, at the bus location.	0

			In idle state. Note: This bit can be used even if no stop condition is detected on the bus. When busy is 1, reset the module.	
14	Reserved	RO	is reserved.	0
13	ALERT	RW	SMBus alert bit; user codes can set or clear this bit. This bit; when PE is set, this bit can be cleared by hardware. 1: Drive the SMBusALERT pin low to respond to the address. The header should immediately follow the ACK signal; 0: Release the SMBusALERT pin to make it high, responding to the address. The head should immediately follow the NACK signal.	0
12	PEC	RW	Packet error detection enable bit; setting this bit enables packet error detection. Error detection. User code can set this position to bit or clear it; when After the PEC is transmitted, a start or stop signal, or a PE bit, is generated. When clearing to 0, the hardware clears that bit; 1: Includes PEC; 0: Without PEC. Note: The PEC becomes invalid if the arbitration is lost.	0
11	POS	RW	The ACK and PEC position settings are configured so that they can be set by user code. Set or clear; after PE is cleared, it can be cleared by hardware. remove; 1: The ACK bit controls the next word received in the shift register. The ACK or NAK of the next segment. The PEC shift register receives the next segment. One byte is PEC; 0: The ACK bit controls the word currently being received in the shift register. The ACK or NAK of the section. The PEC bit indicates the current bit shift register. The bytes of the device are PEC. Note: The POS bit is used in byte data reception as follows: It must be configured before receiving. For the first character... cleared. ADDR ACK is used to detect the second Clear the bit immediately after it is used; byte of PEC; this must be done at the event occurrence. After ADDR is generated, the configuration bits are set.	0
10	ACK	RW	The acknowledge enable bit can be set or cleared by user code. Zero; when the PE bit is set, this bit can be cleared by hardware. 1: Return an acknowledgment after receiving a byte; 0: No response is required.	0
9	STOP	RW	Stop event generation bit, which can be set or disabled by user code. Reset, or cleared by hardware when a stop event is detected. Or, if a timeout error is detected, the hardware will set it. In main mode: 1: Produced after the current byte transmission or the current start condition is issued. Life-stopping event; 0: No stopping events are generated. From mode: 1: Release the SCL and SDA lines after the current byte is transferred; 0: No stopping events are generated.	0

8	START	RW	The start event generation bit can be set or disabled by user code. Zeroing occurs when the initial conditions are issued or when PE is zeroed, by the hard drive. Clear all items. In main mode: 1: Repeatedly generate the initial event; 0: No initiating event was generated. From mode: 1: A start event is generated when the bus is idle; 0: No initiating event was generated.	0
7	NOSTRETCH	RW	Clock extension disabled bit; this bit is used in ADDB or BTF settings. When the flag is set, clock delay from mode is disabled. It grows until it is cleared by the software. 1. Clock extension is prohibited; 0: Allows clock extension.	0
6	ENGC	RW	Broadcast call enable bit. Setting this bit enables broadcast calls and responses. Broadcast address 00h.	0
5	ENPEC	RW	PEC enable bit: 1: Enable PEC calculation; 0: Disable PEC calculation.	0
4	ENARP	RW	ARP enable bit: 1: Enable ARP; 0: Disable ARP. If SMBTYPE=0, the default location of the SMBus device is used. Address; if SMBTYPE=1, then the main address of SMBus is used.	0
3	SMBTYPE	RW	SMBus device type: 1: SMBus master device; 0: SMBus from the device.	0
2	Reserved	RO is reserved.		0
1	SMBUS	RW	SMBus mode selection bit: 1: Use SMBus mode; 0: Use I2C mode.	0
0	PE	RW	I2C peripheral enable bit. 1: Enable the I2C module; 0: Disable the I2C module.	0

21.12.2 I2C Control Register 2 (I2Cx_CTLR2) (x=1/2/3/4) Offset Address: 0x04 Bit Name [15:13]

Reserved

		access	describe	Reset value
		RO is reserved.		0
12	LAST	RW	DMA Last Transfer Setting Bit. 1: The next DMA EOT is the final transfer; 0: The next DMA EOT is not the final transfer. Note: This bit is used in master receive mode and can be generated during the last data reception. NAK.	0

11	DMAEN	RW	DMA request enable bit: 1: Allow DMA requests when TxE or RxNE is set; 0: DMA requests are disabled when TxE or RxNE is set.	0
10	ITBUFEN	RW	Buffer interrupt enable bit. 1: An event interrupt is generated when TxE or RxNE is set; 0: No interrupt is generated when TxE or RxNE is set.	0
9	ITEVTEN	RW	Event interrupt enable bit: 1: Enable event interruption; 0: Disable event interruption. This interrupt will occur under the following conditions: SB=1 (Main Mode); ADDR=1 (Master-Slave Mode); ADDR10=1 (Main Mode); STOPF=1 (from mode); BTF=1, but there are no TxE or RxNE events; If ITBUFEN=1, the TxE event is 1; If ITBUFEN=1, the RxNE event is 1.	0
8	ITERREN	RW	Error interrupt enable bit: 1: Allow error interruption; 0: Disable error interruption; This interrupt will occur under the following conditions; BERR=1; ARLO=1; AF=1; OVR=1; PECERR=1; TIMEOUT=1; SMBAlert=1.	0
[7:6] Reserved		RO is reserved.		0
[5:0] FREQ[5:0]		RW	The I2C module clock frequency domain must be entered correctly. The rate is sufficient to produce correct timing, with an allowable range of 4-60MHz. It must be set between 000100b and 111100b. The unit is MHz.	0

21.12.3 I2C Address Register 1 (I2Cx_OADDR1) (x=1/2/3/4) Offset Address: 0x08

Bit	name	access	describe	Reset value
15	ADDMODE	RW	Address pattern. 1: 10-bit slave address (does not respond to 7-bit address); 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. Ignore when using a 7-bit address.	0
[7:1] ADD[7:1]		RW interface address, bits 7-1. The		0
0	ADD0	RW	interface address, when using a 10-bit address, is bit 0. Ignore 7-bit addresses.	0

21.12.4 I2C Address Register 2 (I2Cx_OADDR2) (x=1/2/3/4) Offset Address: 0x0C

Bit	name	access	describe	Reset value
[15:8] Reserved		RO is 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; setting this bit allows ADD2 to also... It was identified.	0

21.12.5 I2C Data Register (I2Cx_DATAR) (x=1/2/3/4) Offset Address: 0x10 Bit Name [15:8]

Reserved

		access	describe	Reset value
		RO Reserved.		0
[7:0] DR[7:0]		RW	This field is used to store received data or stored data. This is used to send data to the bus.	0

21.12.6 I2C Status Register 1 (I2Cx_STAR1) (x=1/2/3/4) Offset Address: 0x14 Name

Bit		access	This	Reset value
15	SMBALERT	RW0	describes the SMBus warning bit, which can be reset by the user writing 0, or... It is reset by hardware when PE goes low. In SMBus host mode: 1: An SMBus warning was generated on the pin; 0: No SMBus warning. In SMBus slave mode: 1: Receive the SMBAlert response header sequence until... SMBAlert decreases; 0: No SMBAlert response header sequence was received.	0
14	TIMEOUT	RW0	Timeout or Tlow error flag, this bit can be written to 0 by the user. Reset, or hardware reset when PE goes low. 1: SCL has been low for 25ms, or the host has accumulated a low level. The clock expansion time exceeds 10ms, or the accumulation time from the device low level is... The time exceeded 25ms; 0: No timeout error. Note: When this bit is set in slave mode, the slave device will reset the communication and the hardware will release the bus; when this bit is set in master mode, the hardware will issue a stop condition.	0
13	Reserved	RO is reserved. The		0
12	PECERR	RW0	PEC error flag bit is set when a PEC error occurs during reception; this bit can be used by... Write 0 to reset, or be reset by hardware when PE goes low. 1: A PEC error has occurred. Upon receiving the PEC, a NAK response is returned. 0: No PEC error.	0
11	OVR	RW0	Overload and underload indicator: 1: Overload or underload events occur: When NOSTRETCH=1, When a new byte is received in receive mode, the data register... If the contents of the receiver have not yet been read, newly received bytes will be lost. Loss; in send mode, no new data is written to the data repository.	0

			The same byte will be sent twice; 0: No overload or underload events.	
10	AF	RW0	The response failure flag can be reset by the user writing 0. Or it can be reset by hardware when PE goes low. 1: Response error; 0: Normal response.	0
9	ARLO	RW0	The arbitration loss flag can be reset by writing 0 to it. Or it can be reset by hardware when PE goes low. 1: Arbitration loss was detected, and the module lost control of the bus; 0: Arbitration is normal.	0
8	BERR	RW0	Bus error flag, this bit can be reset by writing 0 to it. Or it can be reset by hardware when PE goes low. 1: An error occurred with the start or stop conditions; 0: Normal.	0
7	TxE	RO	When the data register is empty, write data to the data register. It can be cleared, or a start or stop bit can be generated, or when Once PE is 0, it will be automatically cleared by the hardware. 1: When sending data, the data transmission register is empty; 0: The data register is not empty.	0
6	RxNE	RO	The data register's not-empty flag is used for reading and writing data registers. This bit will be cleared by the operation, or by hardware when PE is 0. This position. 1: When receiving data, the data register is not empty; 0: Normal.	0
5	Reserved	RO is reserved.		0
4	STOPF	RO	The stop event flag is set after the user reads status register 1. A write operation to control register 1 will clear this bit, or when Once PE is 0, this bit is cleared by hardware. 1: After responding, the slave device detects a stop event on the bus. Item; 0: No stop event was detected.	0
3	ADD10	RO	The 10-bit address header sequence sends the flag bits, and the user reads the status register. After register 1 is accessed, any write operation to control register 1 will clear it. This bit is removed, or it is cleared by hardware when PE is 0. 1: In 10-bit address mode, the master device has already assigned the first... The address bytes are sent out; 0: None.	0
2	BTF	RO	End-of-byte transmission flag bit, user reads status register 1 Afterwards, reading or writing to the data register will clear this bit; during transmission In the middle, after initiating a start or stop event, or when PE is After 0, this bit is cleared by hardware. 1: End of byte transmission. When NOSTRETCH=0: During transmission, When new data is sent and the data register has not yet been written... When receiving new data, but... The data register has not yet been read;	0

			0: None.	
1	ADDR	RW0	Address sent/address match flag, user reads status. After register 1 is accessed, a read operation on status register 2 will clear it. This bit is cleared by hardware, except when PE is 0. Main mode: 1: Address transmission complete: In 10-bit address mode, when receiving... After the ACK bit in the second byte of the address, it is set; In 7-bit address mode, the bit is set upon receiving an ACK for the address; 0: Address transmission has not ended. From the pattern: 1: The received address matches; 0: Address mismatch or address not received.	0
0	SB	RO	Start bit sent flag, read status register 1 and then write data. The operation on the register will clear this bit, or when PE is 0. The hardware will clear this bit. 1: Start bit has been sent; 0: No start bit sent.	0

21.12.7 I2C Status Register 2 (I2Cx_STAR2) (x=1/2/3/4) Offset Address: 0x18 Name

Bit		access		Reset value
[15:8] PEC[7:0]		RO	Description packet error checking field, when PEC is enabled (ENPEC is set). This field stores the value of PEC.	0
7	DUALF	RO	Match the detection flag bit when a stop bit or start bit is generated. Alternatively, when PE=0, the hardware will clear this bit to zero. 1: The received address matches the content in OAR2; 0: The received address matches the content in OAR1.	0
6	SMBHOST	RO	SMBus host header flag bit, used to generate stop or start bits. When PE=0, the hardware will clear this bit. 1: When SMBTYPE=1 and ENARP=1, SMBus was received. Host address; 0: No SMBus host address received.	0
5	SMBDEFAULT	RO	The SMBus device defaults to the address flag, which is triggered when a stop bit is generated or... When the start bit is active, or when PE=0, the hardware will clear this bit to zero. 1: When ENARP=1, the default address of the SMBus device is received; 0: Address not received.	0
4	GENCALL	RO	Broadcast call address flag bit, when generating stop bit or start bit When PE=0, the hardware will clear this bit to zero. 1: When ENGC=1, the address where the broadcast call is received; 0: No broadcast call address received.	0
3	Reserved	RO	Reserved.	0
2	TRA	RO	Transmit/Receive Flags , When a stop event is detected (STOPF=1), repeated start condition, bus arbitration lost. When ARLO=1 or PE=0, the hardware will clear it to zero.	0

			1: Data has been sent; 0: Data received. This bit is determined by the R/W bits of the address byte.	
1	BUSY	RO	Bus busy flag, this bit will be activated when a stop bit is detected. Cleared. This information is still available when the interface is disabled (PE=0). Updated. 1: Bus busy: SDA or SCL is low; 0: Bus is idle and there is no communication.	0
0	MSL	RO	Master-slave mode indicator bit When the interface is in main mode (SB=1), the hardware sets this bit; when the bus detects a... If a stop bit is lost during arbitration, or if PE=0, the hardware will clear the cache. Except for this position.	0

21.12.8 I2C Clock Register (I2Cx_CKCFGR) (x=1/2/3/4) Offset Address: 0x1C

Bit	name	access	describe	Reset value
15	F/S	RW	Main mode selection bit. 1: Quick Mode; 0: Standard mode.	0
14	DUTY	RW	Duty cycle in fast mode: 1: T low level / T high level = 16/9; 0: T low level / T high level = 2.	0
[13:12] Reserved		RO is reserved.		0
[11:0] CCR[11:0]		The RW clock divider field determines the frequency waveform of the SCL clock.		0

21.12.9 I2C Rise Time Register (I2Cx_RTR) (x=1/2/3/4) Offset Address: 0x20 Bit Name [15:6]

Reserved

		access	describe	Reset value
		RO Reserved.		0
[5:0] TRISE[5:0]		RW	Maximum rise time domain. This bit sets the SCL in master mode. Rise time. The maximum rise time is equal to TRISE-1. Clock cycle. This bit can only be set when the PE (Pre-Execution Environment) is cleared. For example... The input clock period of the IF I2C module is 125ns, while TRISE If the value is 9, then the maximum rise time is (9-1). *125ns, which is 1000ns.	000010b

Chapter 22 I3C Bus (I3C)

The I3C bus is a two-wire serial, single-ended, multi-branch bus designed to improve upon the traditional I2C bus. The I3C interface is responsible for...

This device enables communication between itself and other devices connected to the I3C bus. It supports both master and slave modes.

When in use, it can enhance the functionality of the I2C interface while maintaining a certain degree of backward compatibility.

22.1 Main Features

Supports master and slave devices .

Supports MIPI I3C specification v1.1

Supports multi-host functionality

Supports DMA ÿ In-

band interrupt (IBI) function

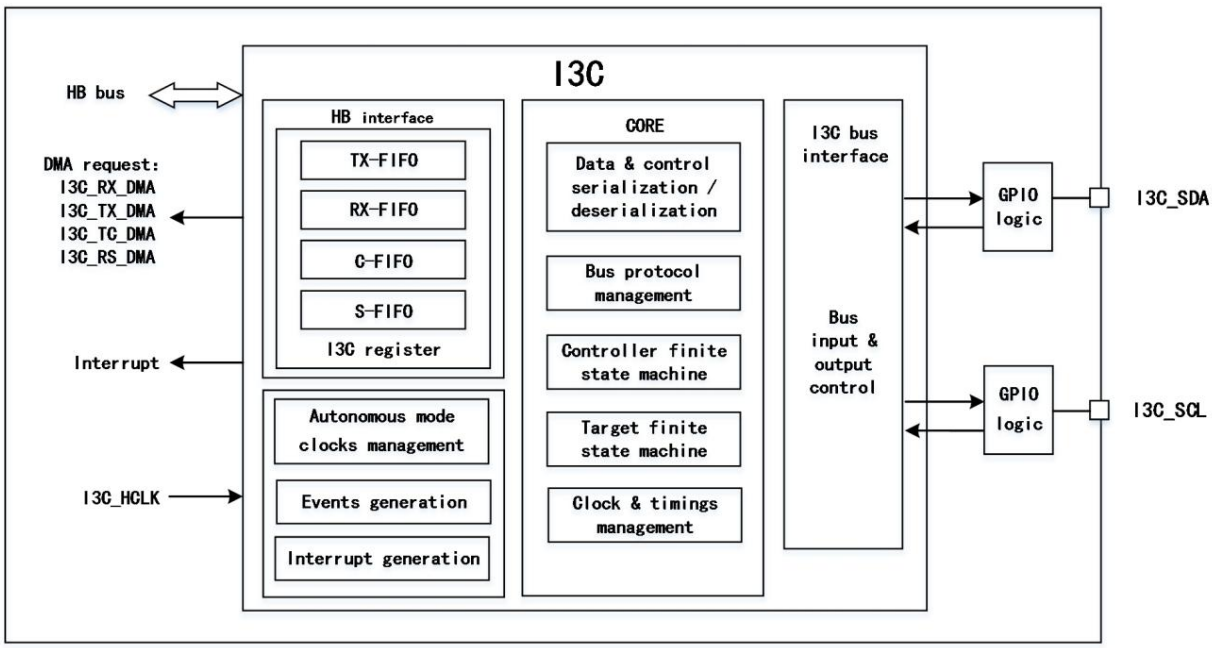
Built - in error detection and recovery

• I3C SCL bus clock up to 12.5MHz • Supports dynamic

address allocation, direct and broadcast Common Command Code (CCC) and private read/write transfers

22.2 Overview

Figure 22-1 Block diagram of I3C



22.3 Functional Description

22.3.1 I3C Peripheral Status

I3C peripherals can be used as either I3C masters or I3C slaves. In either case, the peripheral is in the following state:

one:

Prohibited status

Operating conditions: After the I3C peripheral is reset (I3C reset is performed by setting I3CRST bit 1 in the RCC module), the peripheral is in a disabled state.

Transition from disabled to idle state: When the software sets the EN bit of the R32_I3C_CFGR register to 1, the peripheral completes the internal configuration parameters.

After verification, the state switches from prohibited to idle.

ÿIdle state: In the

idle state, the software can perform partial updates to the I3C peripheral configuration.

Transition from idle to active state: 1)

Software control trigger: In

master mode, triggered when software starts frame transmission by setting the TSFSET bit of the R32_I3C_CFGR register to 1 via a write operation;

In slave mode, triggered when software issues an IBI/CR/HJ request; 2)

Hardware event trigger:

Triggered by the master device after receiving a start request from the slave device and the maximum TCAS time during

bus arbitration; Triggered by the slave device detecting a broadcast/direct CCC or private read/write via the SDA/SCL lines.

ÿ Operating State: In

operating state, the peripheral performs transfers on the bus.

When a requested transfer is completed in operating state, the software is notified via an event in the R32_I3C_EVR register, and the corresponding interrupt is enabled via the R32_I3C_INTENR register. After switching from operating state to idle state, the peripheral can still communicate on the bus and can undergo partial reconfiguration. 1) As a Master

Device: In operating state, the event/flag generated upon completion of a requested transfer can be Frame Complete (FCF), IBI/

Master device role/hot join request completed (IBIF/CRF/HJF) or transmission error (ERRF);

2) As a slave device: In the working state, the event/flag generated after the requested transmission is completed can be Dynamic Address Allocation Complete (DAUPDF), IBI Complete (IBIENDF), Master Role Acquisition Complete (CRUPDF), Broadcast/Direct CCC Complete (xxUPDF/RSTF/GETF/STAF), Private Read/Write Complete (FCF), or Transmission Error (ERRF).

22.3.2 I3C Master Device States and Programming Sequences

In this section, we provide the complete programming sequence for an I3C peripheral when it is used as a master device, covering state transitions, main subtasks, and conditions.

Master device initialization

When the master device is disabled (EN bit of R32_I3C_CFGR register is 0), the software must follow these steps for initialization.

change:

• Configure the R32_I3C_CFGR register, setting CRINIT to 1 to mark the master device role . • Clear

the HST_SIE_RST bit in the R32_I3C_RESET register . • Configure the I3C bus

timing: 1) Configure the

R32_I3C_TIMINGR0 register to set the SCL clock high-level time for the traditional I2C and I3C open-drain/push-pull phases.

Period and low-level time period

2) Configure the R32_I3C_TIMINGR1 register to set the SDA hold time (tHD_PP) and bus idle condition time during the push-pull phase.

(I3C tCAS, traditional I2C tBUF), I3C repeat start bit timing (tCASr, tCBSr), I3C stop timing (tCBP), SCL clock low-level maximum pause time (for ENTDAA CCC (tSTALLDAA), traditional I2C ACK/NACK, I3C transmission address phase, parity bit for write data transmission, traditional I2C write transmission ACK/NACK data phase, I3C read transmission transition bit, or traditional I2C write transmission ACK/NACK phase (tSTALL)). Peripherals used as master devices can adjust the SCL clock low-level pause time and tNEWCRLock according to their own needs to adapt to the master device role switching process (after the GETACCCR CCC command).

3) Configure the R32_I3C_TIMINGR2 register to set the pause time of the SCL clock low level (each stage can be individually enabled or disabled to adjust the pause time of the SCL clock low level (to meet the requirements of the addressed I3C slave device or traditional I2C slave device during SDA switching)

Configure the R32_I3C_DEVR0 register DA[6:0] bits to set its own dynamic address . Configure the R32_I3C_DEVRx register to set the management of any slave device x (x = 1/2/3/4) . Configure the R32_I3C_CFGR register to set the execution mode for frame transmission or target request transmission . Configure the R32_I3C_INTENR register to set the interrupt generation or polling mode for any event . Set the EN bit to 1 after initialization to enable the I3C peripheral.

To initiate frame transmission initiated by the master device, when the master device is enabled (EN bit of R32_I3C_CFGR register is 1), the software can initiate frame transmission using any of the following configuration methods: 1. Software trigger: Set the TSFSET bit of R32_I3C_CFGR register to 1 by performing a write operation;

– This operation will cause the hardware to set the CFNFF flag in the R32_I3C_EVR register to request the writing of the first control word to the R32_I3C_CTLR register. 2. No trigger: The first control word is written directly to the R32_I3C_CTLR register via software.

Regardless of the startup method, the I3C peripheral will then switch to working state. When the control word is not the last message of the I3C frame (i.e., the MEND bit of the R32_I3C_CTLR register is not set), and no transmission error occurs (i.e., the ERRF bit of the R32_I3C_EVR register is set to 1), the hardware will continuously request the next control word and continue frame transmission:

1) If the C-FIFO is in non-DMA mode (the CDMAEN bit of the R32_I3C_CFGR register is 0), the software will write the next control word after the CFNFF flag of the R32_I3C_EVR register is set or the corresponding interrupt is triggered (the CFNFIE bit of the R32_I3C_INTENR register is set to 1);

2) If the C-FIFO is in DMA mode (the CDMAEN bit of the R32_I3C_CFGR register is 1), then the corresponding I3C DMA... The request will be automatically processed by the allocated DMA channel and written to the next control word.

Initiating and receiving transmissions initiated by slave devices: When the master device is enabled (EN bit of R32_I3C_CFGR register is 1), the master device can initiate a transmission, and the slave device can also initiate a transmission by issuing a start request (driving SDA low), provided that the master device has allowed hot-join requests, IBI requests, or master role requests through the R32_I3C_DEVR0 register. In this case, even if the master device software does not plan to initiate a frame transmission, the hardware will automatically enter the working state (activating the SCL clock before the maximum tCAS time (which can be defined as 1ÿs, 100ÿs, 2ms, or 50ms, corresponding to bus working states 0, 1, 2, or 3, respectively)) to receive hot-join/in-band interrupt/master role requests from slave devices.

Perform frame transmission initiated by the master device

On the bus, the master device continuously performs frame transmissions until the last message transmission is complete (the FCF bit in the R32_I3C_EVR register is 1), or an error occurs during transmission (the ERRF bit in the R32_I3C_EVR register is 1), triggering the corresponding interrupt (if enabled). This process relies on the R32_I3C_CTLR and R32_I3C_TDW/TDR registers (written explicitly by software or automatically pushed by the allocated DMA channel) and the R32_I3C_RDW/RDR registers (read explicitly by software or by the allocated DMA channel). After the transmission is complete, the I3C master device returns to the idle state. When the S-FIFO is disabled (the SMODE bit in the R32_I3C_CFGR register is 0), the hardware updates the I3C status register (R32_I3C_STATR) for each message that completes without transmission errors, indicating that a message exchange has been completed on the I3C bus. The software can choose to read or ignore this register. When the S-FIFO of the R32_I3C_STATR register is disabled (the SMODE bit of the R32_I3C_CFGR register is 0), in addition to the completion of the last message transfer (the FCF bit of the R32_I3C_EVR register is 1) or a transfer error occurring (the ERRF bit of the R32_I3C_EVR register is 1) and generating a corresponding interrupt (if enabled), during direct CCC read or private read transfers, if the slave device prematurely terminates the read transfer, it will also notify the software by setting the R32_I3C_EVR register RXTGTENDF bit to 1 and generating a corresponding interrupt (if enabled). The software can then read the R32_I3C_STATR register to obtain more information about the transferred data.

If S-FIFO is enabled (R32_I3C_CFGR register SMODE bit set to 1), the status register R32_I3C_STATR must be read once after each message transmission, regardless of whether the slave device ends the read transmission early. This can be read by software (notified by R32_I3C_EVR register SFNEF bit being 1 and the corresponding interrupt (if enabled)) or via DMA (if R32_I3C_CFGR register SDMAEN bit set to 1). A frame transmission completion report is only submitted after reading the status of the last message (S-FIFO empty) (R32_I3C_EVR register FCF bit set to 1).

After returning to idle state, the

master device's transmission configuration software can update the configuration of I3C peripherals before the next transmission.

- Configure the R32_I3C_TIMINGR2 register to modify the SCL clock pause; -

Configure the R32_I3C_INTENR register to modify the interrupt/polling mode

strategy; - Configure the R32_I3C_CFGR registers TXTHRES, RXTHRES, TMODE, SMODE, TXDMAEN, CDMAEN, RXDMAEN, ...

The SDMAEN, EXITPTRN, RSTPTRN, and NOARBH bits are

used to modify/prepare the control word, status word, and read/write data for the next frame transmission via software

and/or DMA. Typically, after a broadcast/direct DISEC/ENEC CCC is sent

and completed: – The hot-join acknowledgment policy is modified via the HJACK bit

in the R32_I3C_CFGR register; – The IBI/CR acknowledgment policy for any slave device x is modified via the R32_I3C_DEVRx register.

22.3.3 I3C Slave State and Programming Sequence In

this section, we provide a complete programming sequence for an I3C peripheral used as a slave device, covering state transitions, main subtasks, and conditions.

Device initialization

When the slave device is disabled (EN bit of R32_I3C_CFGR register is 0), the software must be initialized as follows: ÿ Configure the

R32_I3C_CFGR register, setting CRINIT to 0 to set the I3C peripheral as a slave device. ÿ Clear the

DEV_SIE_RST bit in the R32_I3C_RESET register . ÿ Set the I3C bus

timing through I3C timing register 1 (R32_I3C_TIMINGR1): Write to AVAL[7:0] to set:

– Bus availability condition time (tAVAL), for IBI or master role requests – Bus idle

condition time (tIDLE), for hot-join requests – tNEWCRLock, for

master role switching processes (after GETACCCRCCC) ÿ Configure slave-

initiated requests: Write the following bits to the I3C's own device characteristic register (R32_I3C_DEVR0):

– IBIEN: In-band interrupt (also known as IBI) request enable/disable

– CREN: Master role request enable/disable

– HJEN: Enable/disable hot-join request ;

Initialize slave device characteristics and

functions ; Configure transfer execution mode:

In the R32_I3C_CFGR register, TXDMAEN and RXDMAEN enable/disable DMA modes for TX-FIFO and RX-FIFO, respectively.

TXTHRES and RXTHRES are the byte/word thresholds for TX-FIFO and RX-FIFO, respectively; setting SMODE to 0 disables S-

FIFO ; configure the R32_I3C_INTENR register to modify the interrupt generation or polling mode for

any event ; set EN to 1 after initialization to enable I3C peripherals.

Receive messages from the master device (broadcast CCC, direct read/write CCC, or private read/write).

When a slave device is in an idle state (EN bit of the R32_I3C_CFGR register is 1), it indicates that it is ready to receive communication messages from the master device on the I3C bus and is ready to switch to active

state. In active state, the slave device will first receive the broadcast ENTDAA CCC (possibly after receiving the optional broadcast ENEC/DISEC CCC), and then be assigned a dynamic address. The event DAUPF in the R32_I3C_EVR register is set to 1, and a relevant interrupt is generated (if enab

The device then returns to an idle state.

Afterwards, the idle slave device is ready to receive any other broadcast CCC messages, direct read/write CCC messages, or private read/write messages from the master device.

For each message received

without errors, the hardware reports the completion of a message exchange on the I3C bus by updating the I3C status register (R32_I3C_STATR). The software receives this notification through the corresponding flag in the I3C event register (R32_I3C_EVR) or the corresponding interrupt (if enabled) in the I3C interrupt enable register (R32_I3C_INTENR). The software must read the I3C status register (R32_I3C_STATR) if the following messages occur:

Private read: Get the number of bytes of data exchanged, because the master device may finish the transfer earlier than the slave device expects (if the R32_I3C_STATR register XDCNT[15:0] is less than the R32_I3C_TGTTDR register TGTTDCNT[15:0]).
At this point, the software must set TXFLUSH in the R32_I3C_CFGR register to 1 to clear the TX-FIFO. ÿ DEFTGTS CCC
or DEFGRPA CCC: Get the number of data bytes received in the RX-FIFO.

A transmission initiated by the slave

device is started when the slave device transitions from a disabled state to an idle state (the software writes 1 to the EN bit of the R32_I3C_CFGR register) and is simultaneously able to receive data.
When the master device broadcasts a CCC, the software can issue a hot-join request (by writing 1000b to bits MTYPE[3:0] in the R32_I3C_CTLR register) to apply for eligibility to join the next ENTDAACCC, provided that hot-join requests are enabled (bit HJEN in the R32_I3C_DEVR0 register is set to 1). After allocating

a dynamic address (bit DAUPF in the R32_I3C_EVR register is set to 1) (more commonly done simultaneously with frame transmissions from the master device), the software can issue an IBI (In-Band Interrupt Request) or a master role request to the master device by writing the relevant control word to the R32_I3C_CTLR register.

After the I3C peripheral (as a slave device) returns to an

idle state after updating its configuration, the software can update the I3C slave device's configuration before the next transfer.

- Configure the R32_I3C_INTENR register to modify the interrupt/polling mode policy. -

Configure the TXTHRES, RXTHRES, TXDMAEN, and RXDMAEN bits of the R32_I3C_CFGR register. - Modify/prepare

the I3CIBI payload data register (R32_I3C_IBIDR). If there is any payload (when bit BCR2 of the R32_I3C_BCR register is 1), it should be written to the I3C message control register [multiplexed] (R32_I3C_CTLR) when starting IBI transmission.
Perform this operation before MTYPE[3:0] bits are 1010b -

Modify/prepare the I3C slave device send configuration register (R32_I3C_TGTTDR), before receiving a private read or direct CCC from the master device.
Before reading (except GETSTATUS CCC) messages, disable or enable the TX-FIFO to preload a specified number of data bytes to be sent.

22.3.4 I3C Bus Transfer and Programming 22.3.4.1

I3C Command Sets (CCC) (As Master/Slave Device) Table 22-1 provides a list of

supported I3C command sets (e.g., CCC list, general command codes), and an overview of how the command set is handled when the peripheral is acting as a master or slave device. Table 22-1 List of Supported I3C

CCCs (As Master/Slave Device)

CCC Name CCC Value		read / Write	Yes/No definition byte Yes/No subcommand byte	Yes/No optional data bytes	use do host equipment	Used as a device to generate I3C_EVR event	When acting as a slave device: Specific operations
ENEC	0x00 Write		Undefined byte/ subcommand byte	There is one data byte (enabled slave device event byte).	XX, INTUPDF		Update and enable R32_I3C_DEVR0 Registers: HJEN

							CREN, IBIEN (If any)
DISEC	0x01			There is one data byte (Disabled from device events) byte)	XX,	INTUPDF	Update and disable R32_I3C_DEVR0 Registers: HJEN CREN, IBIEN (If any)
ENTASx (x=0…3))	0x02 … 0x05			No data bytes XX,		ASUPDF	renew R32_I3C_DEVR0 Registers AS[1:0] Bit
RSTDAA	0x06			-	XX,	DAUPDF	Will R32_I3C_DEVR0 Register DAVAL bit Clear
ENTDAA	0x07			-	XX,	DAUPDF	renew R32_I3C_DEVR0 Registers DA[6:0] Bit and DAVAL bit Set 1
DEFTGTS	0x08			There is [1+4x(1+n)]] data bytes (n - Number of devices)	X	X, DEFF	renew R32_I3C_RDR/R 32_I3C_RDWR Register
SETMWL	0x09			There are two data bytes	X	X, MWLUPDF	renew R32_I3C_MAXWLR register
SETMRL	0x0A			There are two or three data byte	X	X, MRLUPDF	renew R32_I3C_MAXRLR register
ENTTM	0x0B			There is a data byte X		-	
RSTACT	0x2A		Defining bytes (0x00, 0x01 or 0x02)	No data byte X		X was detected. Reset timing Post position RSTF	renew R32_I3C_DEVR0 register RSTACT[1:0] bits and RSTVAL bit Set to 1
DEFGRPA	0x2B		Undefined bytes/sub command bytes	There are multiple data bytes	X	X, GRPF	and update R32_I3C_RDR/RD WR register
RSTGRPA	0x2C			No data byte X		-	-
Direct CCC							Execute operation upon ACK (If I3C from the design) Backup address and R32_I3C_DEVR0 Registers DA[6:0]

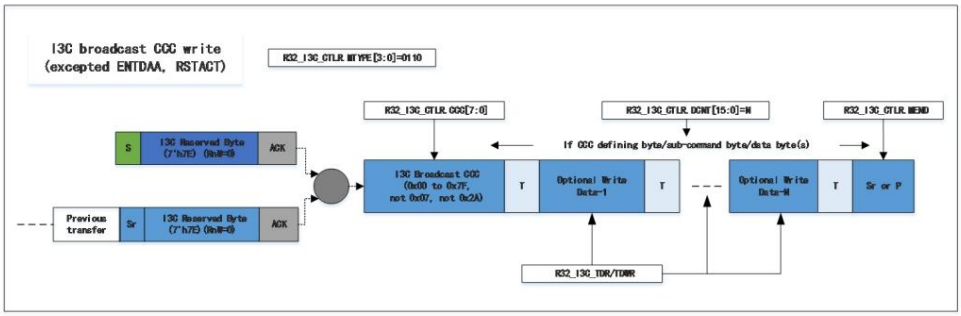
							Match and R32_I3C_DEVR0 register DAVAL=1) (No) Then it is NACK)
ENEC	0x80	Write	Undefined bytes/sub command bytes	There is one data byte (Enable device events) byte)	XX, INTUPDF		Update and enable R32_I3C_DEVR Register 0: HJEN, CREN, IBIEN (if) have)
DISEC	0x81			There is one data byte (Disabled from device events) byte)	XX, INTUPDF		Update and disable R32_I3C_DEVR Register 0: HJEN, CREN, IBIEN (if) have)
ENTASx (x=0…3))	0x82 … 0x85			No data bytes XX, ASUPDF			renew R32_I3C_DEVR0 Registers AS[1:0] Bit
SETDASA	0x87			No data byte X		-	-
SETNEWDA	0x88			There is a data byte XX, DAUPDF			renew R32_I3C_DEVR0 Registers DA[6:0] Bit, and DAVAL Set 1
SETMWL	0x89			There are two data bytes X X, MWLUPDF			renew R32_I3C_MAXWLR register
SETMRL	0x8A			There are two or three data byte	X X, MRLUPDF		renew R32_I3C_MAXRLR register
GETMWL	0x8B	read		There are two data bytes X		X, GETF	from R32_I3C_MAXWL register RMWL[15:0] return Data bytes
GETMRL	0x8C			There are two or three data byte	X	X, GETF	from R32_I3C_MAXRL R register MRL[15:0] returns Data bytes, such as R32_I3C_BCR Register BCR2=1 Then from R32_I3C_MAXRL R register IBIP[2:0] return The third byte

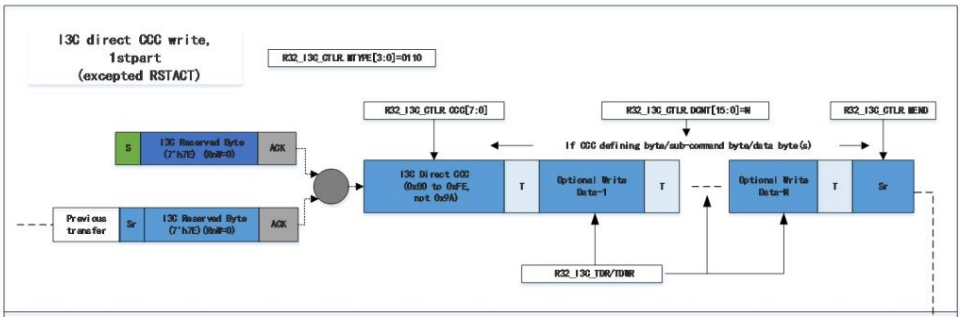
GETPID	0x8D			There are six data bytes X		X, GETF	from R32_I3C_EPIDR Register returns data byte
GETBCR	0x8E			There is one data byte	X	X, GETF	From R32_I3C_BCR register BCR[7:0] return Data bytes
GETDCR	0x8F				X	X, GETF	return R32_I3C_DCR Register BCR[7:0]
GETSTATUS 0x90			Words with/without definitions Section (TGTSTAT, PRECR)	There are two data bytes (Format 1 or Format 2) (With PRECR))	X	X, STAF (in format 1)) X, GETF (Format 2))	Return two data points byte
GETACCCR	0x91		Undefined bytes/sub command bytes	There is a data byte XX, CRUPDF			from R32_I3C_DEVR0 Registers DA[6:0] Return with odd/even Number of check bits According to ByteDance
GETMXDS	0x94		With/Without Semantic Bytes (WRRDTURN, CRHDLY)	There are two data bytes (Format 1) Five Data bytes (format) 2 or format 3 (available) When WRRDTURN) Or a data byte (Format 3 (with) CRHDLY	X	X, GETF	from R32_I3C_GETM XDSR register return Return data bytes
GETCAPS	0x95 Read		With/Without Semantic Bytes (TGTSTAT, CRCAPS	There are three data words Section (Format 1 or Format 2) Equation 2 (with TGTSTAT) (time) or two numbers According to bytes (format 2) (When CRCAPS is present)))	X	X, GETF	from R32_I3C_GETCA The PR register returned 3 Number of GETCAPx According to ByteDance or from R32_I3C_CRCAP Register R returns 2 CRCAPx data byte
RSTACT	0x9A	read / Write	Defining bytes (0x00, 0x01 or 0x02)	Defining bytes (0x00, 0x01 or 0x02)	X	X was detected. Reset timing Post position RSTF	Read: From R32_I3C_DEVR Register 0 RSTACT[1:0] Returned data bytes Write: Update R32_I3C_DEVR Register 0 RSTACT[1:0]

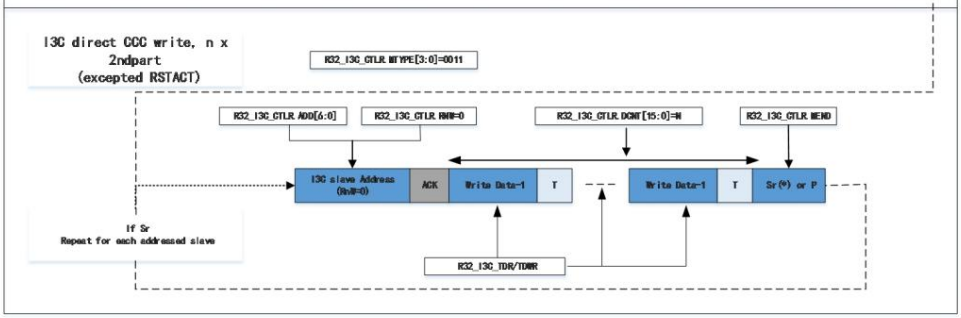
							and RSTVAL bit Set 1
SETGRPA	0x9B	Write	Undefined bytes/sub command bytes	Undefined bytes/sub-names Let byte	X	-	-
RSTGRPA	0x9C				X		

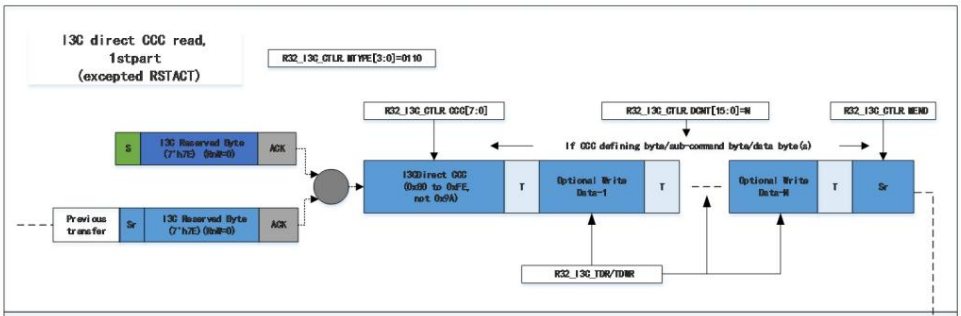
22.3.4.2 I3C Broadcast/Direct CCC Transmission (except ENTDAA and RSTACT) (as master device)

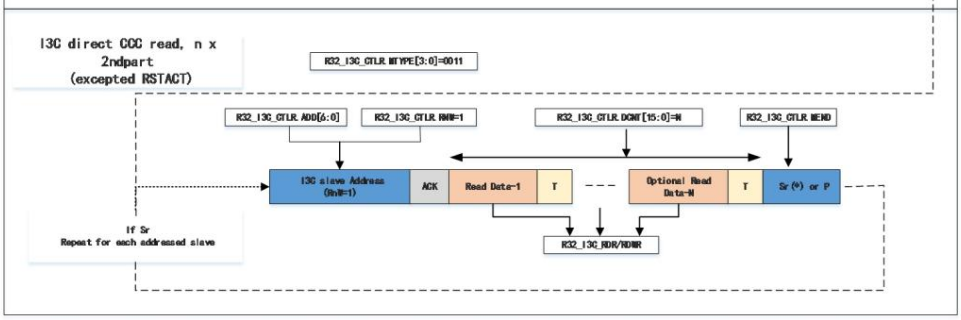
Figure 22-2 I3C CCC Message

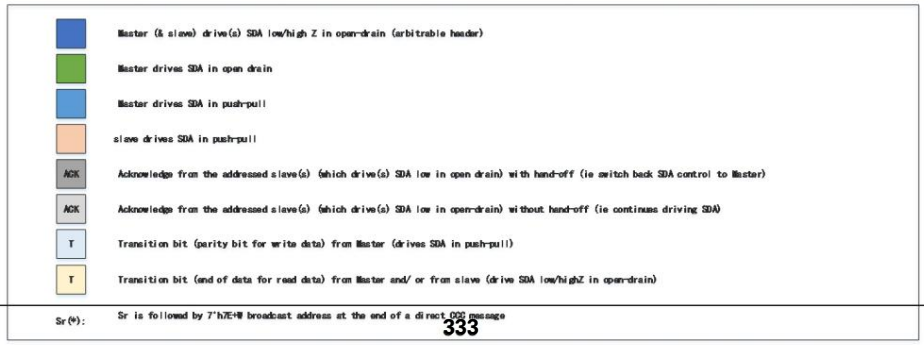






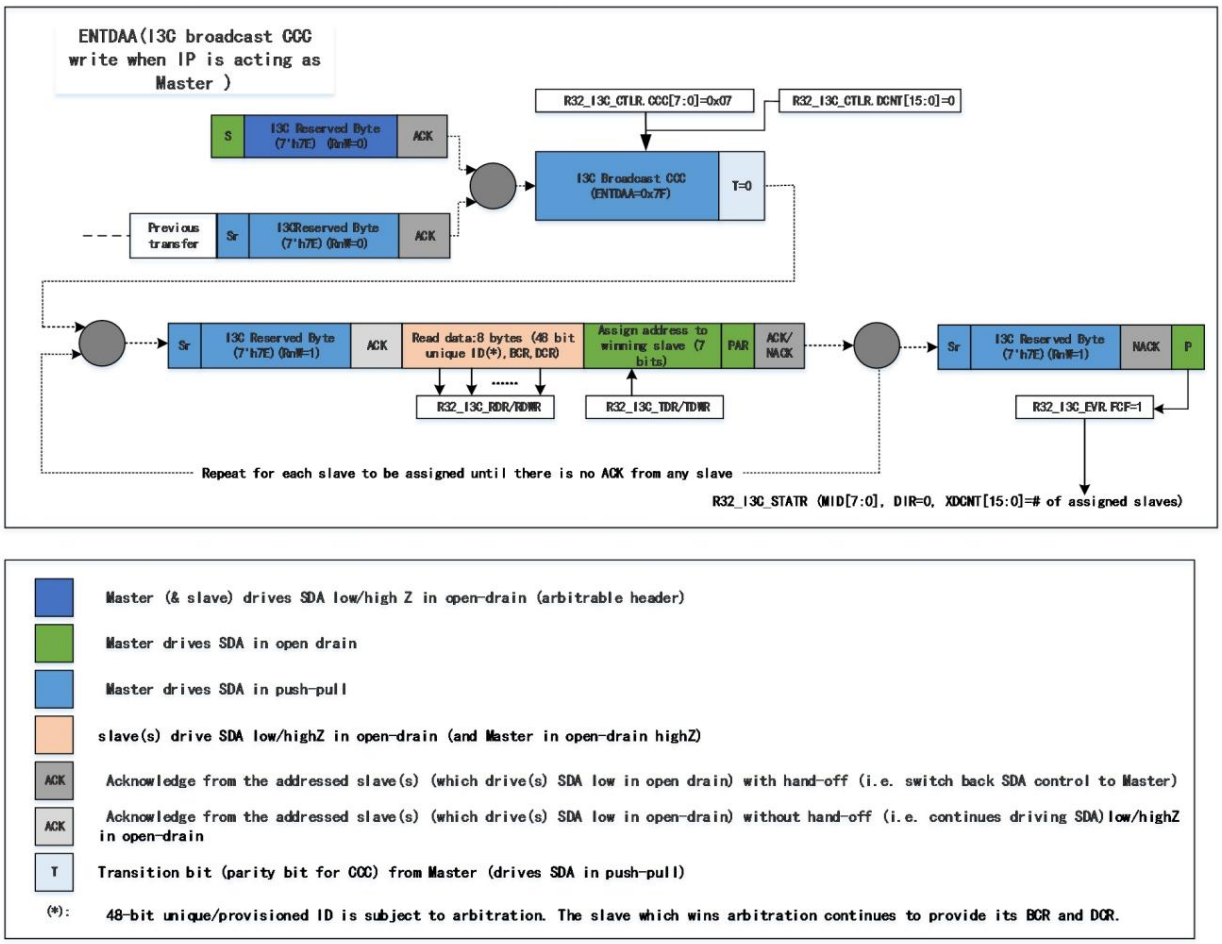






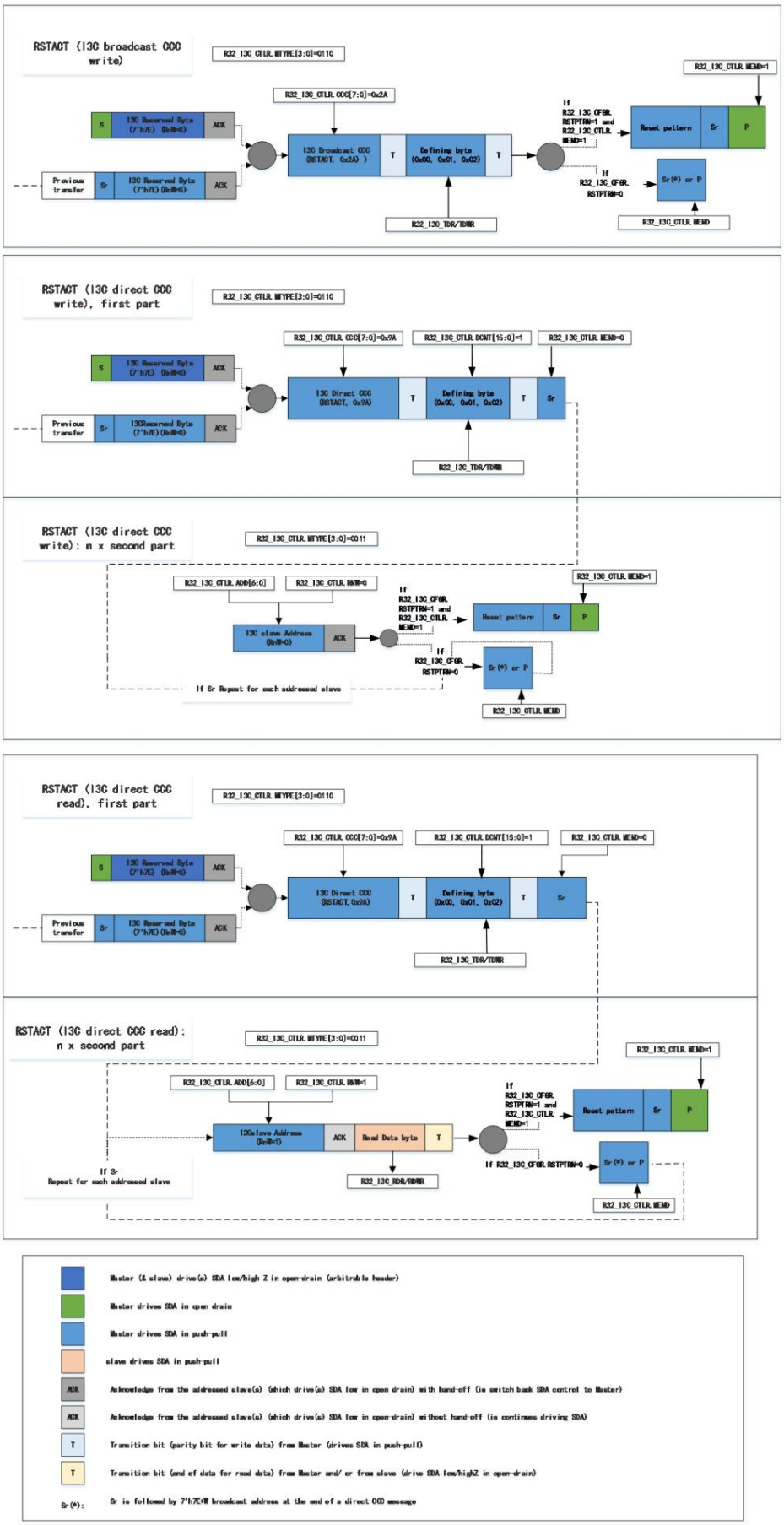
22.3.4.3 I3C Broadcast ENTDAA CCC Transmission (as Master Device)

Figure 22-3 I3C Broadcast ENTDAA CCC Transmission (as Master Device)



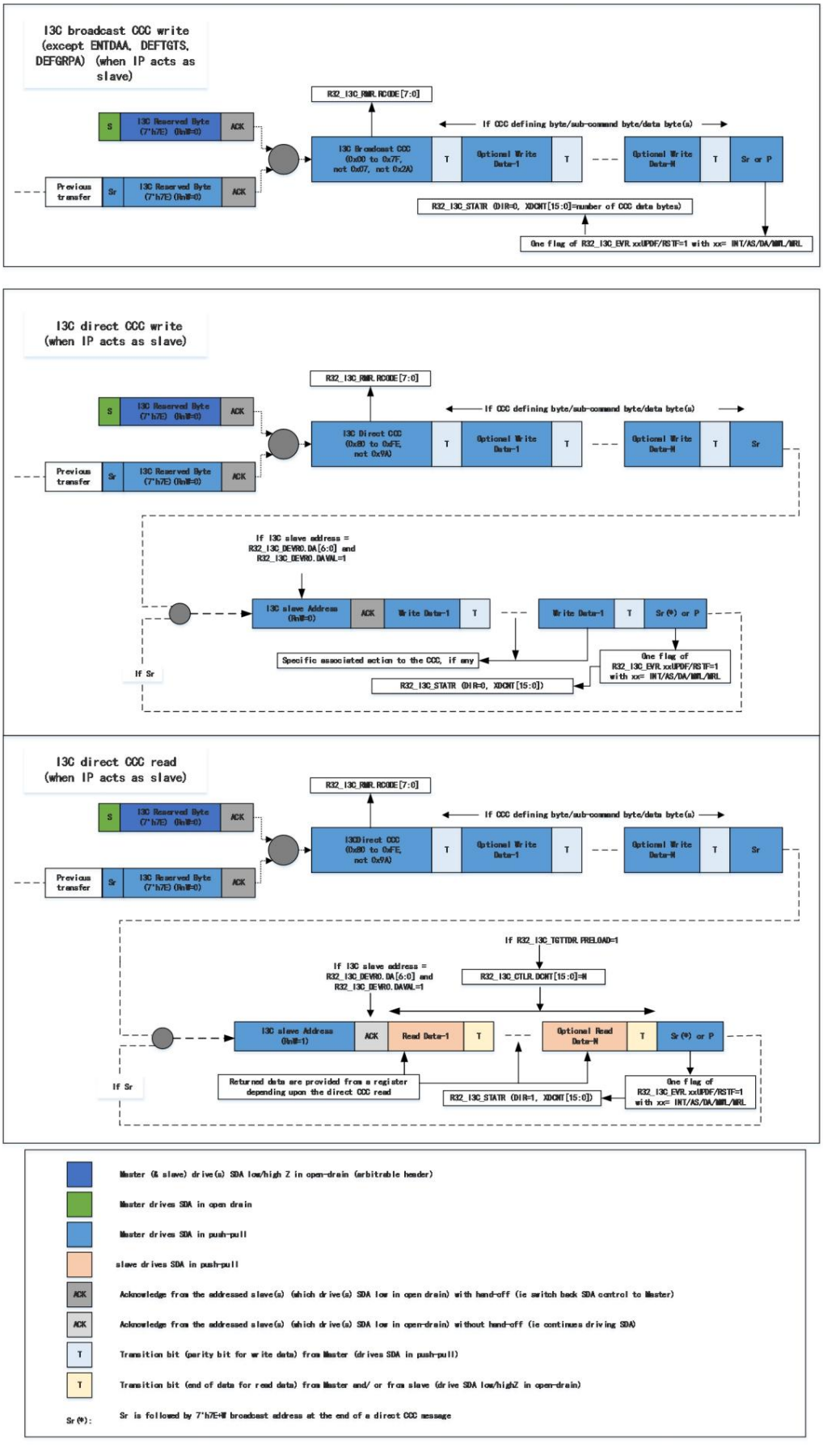
22.3.4.4 I3C Broadcast/Direct RSTACT CCC Transmission (as Master Device)

Figure 22-4 I3C Broadcast, Direct Read/Write RSTACT CCC (as Master Device)



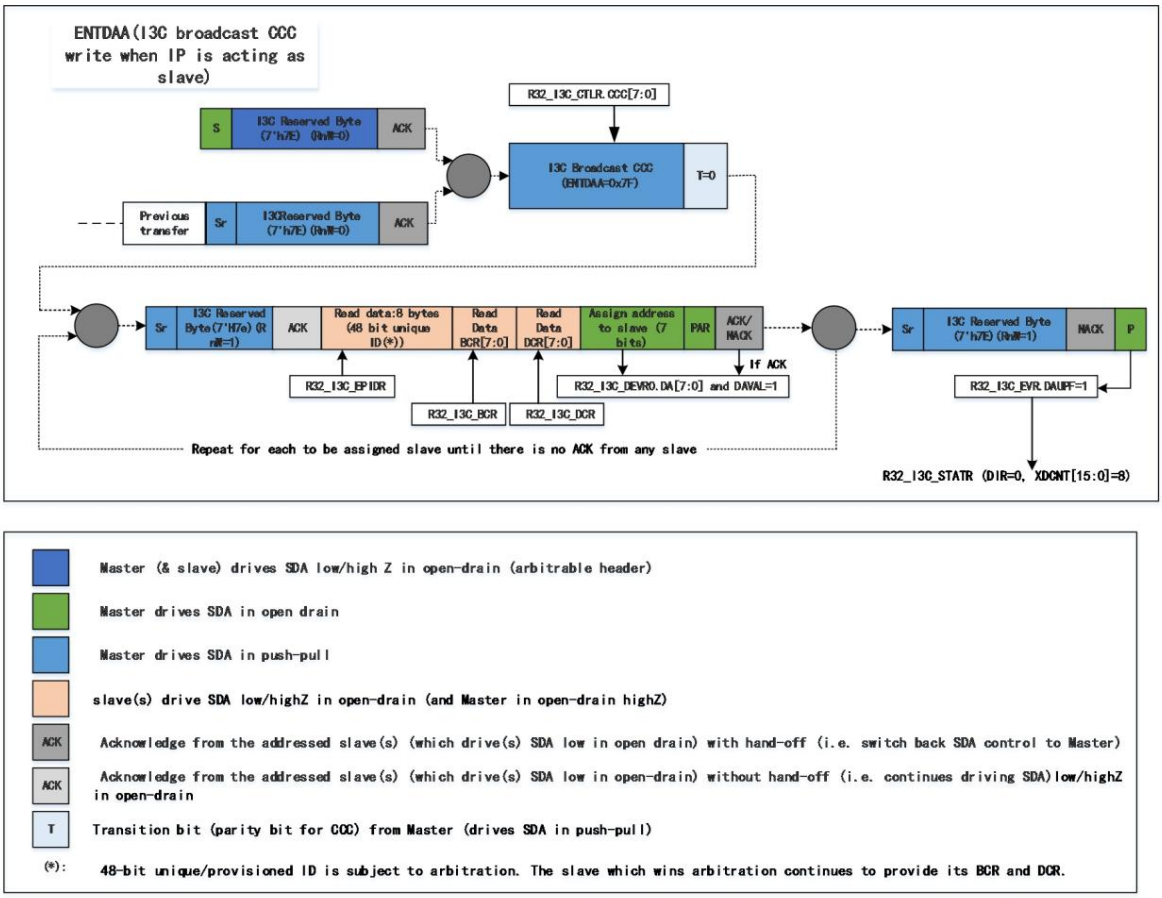
22.3.4.5 I3C Broadcast/Direct CCC Transmission (excluding ENTDAA, DEFTGTS, and DEFGRPA) (as a slave device)

Figure 22-5 I3C CCC Message (as a slave device)



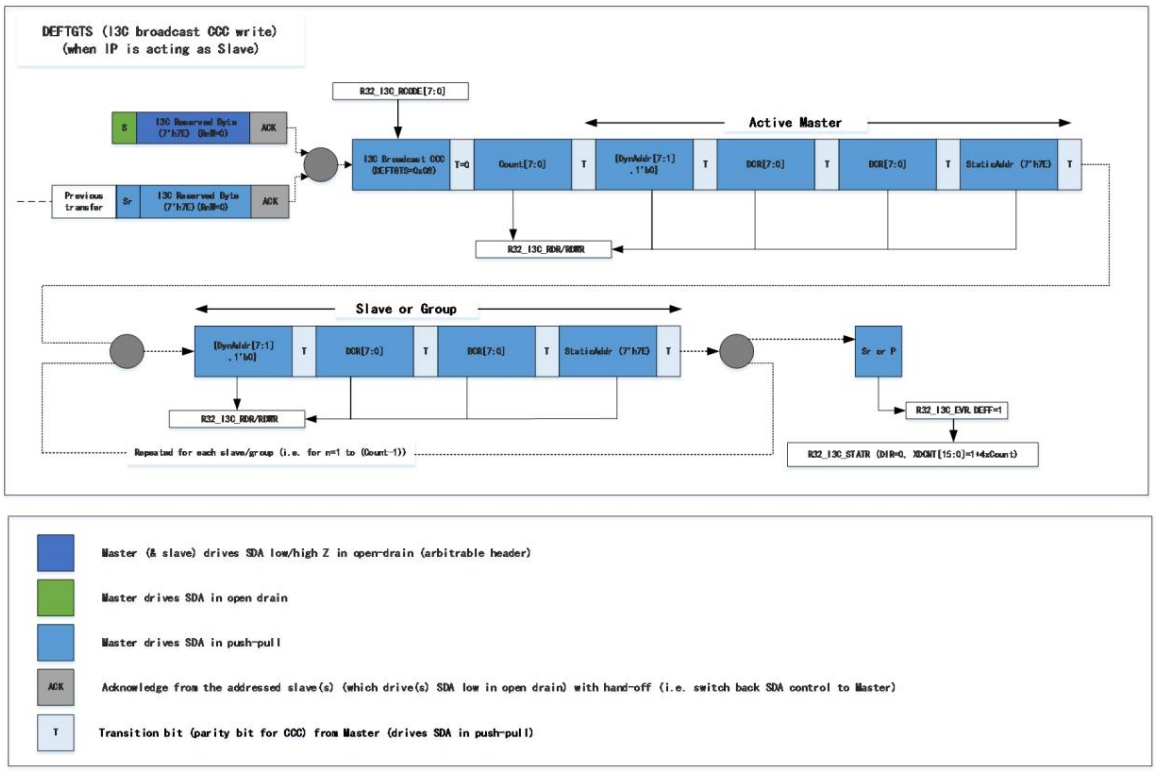
22.3.4.6 I3C Broadcast ENTDAA CCC Transmission (as a Slave Device)

Figure 22-6 I3C Broadcast ENTDAA CCC Transmission (as a Slave Device)



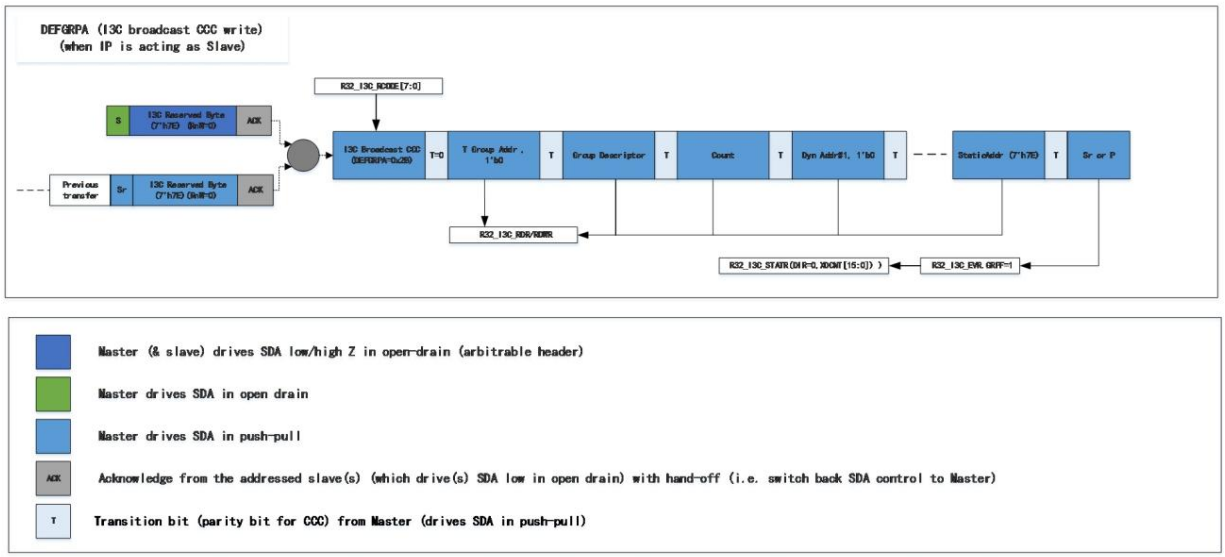
22.3.4.7 I3C Broadcast DEFTGTS CCC Transmission (as a Slave Device)

Figure 22-7 I3C Broadcast DEFTGTS CCC Transmission (as a Slave Device)



22.3.4.8 I3C Broadcast DEFGRPA CCC Transmission (as a Slave Device)

Figure 22-8 I3C Broadcast DEFGRPA CCC Transmission (as a Slave Device)



22.3.4.9 I3C Direct GETSTATUS CCC Transmission (as a Slave Device) When

the I3C is used as a slave device, the hardware will return two data bytes upon receiving GETSTATUS CCC, either in format 1 (no explanatory byte or explanatory byte TGTSTAT = 0x00) or format 2 (explanatory byte PRECR = 0x91).

On the I3C bus, the 2-byte STATUS[15:0] returned in format 1 is as follows:

STATUS[15:14] = 00 (Unused);

STATUS[13] = 1 (if a lost start bit is detected since the previous GETSTATUS CCC was received) or 0 (otherwise)

condition);

STATUS[12] = 1 (if an overflow/underflow error is detected since the previous GETSTATUS CCC was received) or 0 (otherwise);

STATUS[11] = 1 (if the settling time of SCL detected during SDR read transfer exceeds 125ÿs since the previous GETSTATUS CCC was received) or 0 (otherwise).

STATUS[10:8] = 000 to 110 (if a protocol error is detected since the previous GETSTATUS CCC was received (provided that...) STATUS[5] = 1), encoded value x = 0 to 6, corresponding to device error TEx) or 000 (other cases);

STATUS[7:6] = 00 (Ready to perform the handover procedure);

STATUS[5] = 1 (If a protocol error is detected since the previous GETSTATUS CCC was received) or 0 (Other); STATUS[4] = 0 (Reserved); STATUS[3:1] = 000 (Unused); STATUS[0] = 1 (If a pending interrupt exists (provided that IBI is configured in the R32_I3C_CTLR register, IBIEN = 1 and DAVAL = 1 in the R32_I3C_DEVR0 register, and IBI has not been acknowledged by the master device or disabled by DISEC) or 0 (Other).

On the I3C bus, the 2-byte STATUS[15:0] returned in format 2 is as follows:

STATUS[15:8] = 00000000 (Unused);

STATUS[7:2] = 00000 (unused);

STATUS[1] = 1 (if the software is still processing the received DEFTGTS or received DEFGRPACCC and has not cleared the relevant events (DEFF = 1 or GRPF = 1 in the R32_I3C_EVR register)); the master device must wait for the software to complete the above operations before issuing GETACCR CCC (otherwise it will not be acknowledged).

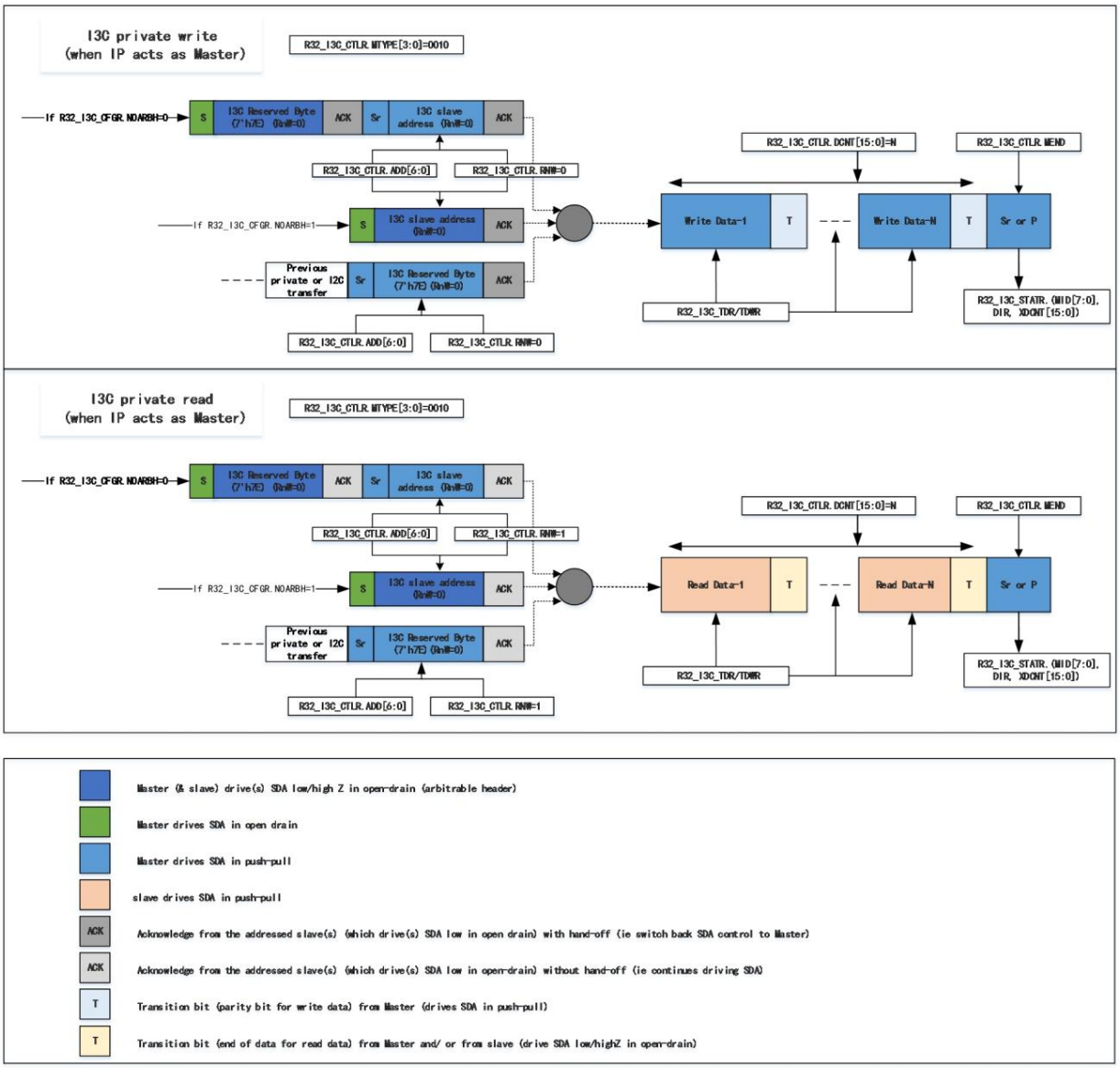
STATUS[0] = 1 (if DEFTGTS or DEFGRPA CCC may be missing). If a missing start bit is detected (WKPF = 1 in the R32_I3C_EVR register), this bit will be set to 1; if DEFF = 1 or GRPF = 1 in the R32_I3C_EVR register, this bit will be cleared to zero.

Once GETSTATUS CCC in format 1 completes, it will trigger a STAF = 1 in the R32_I3C_EVR register and a corresponding interrupt (such as...). If enabled, i.e., STAIE = 1 in the R32_I3C_INTENR register, a report will be generated.

When GETSTATUS CCC in format 2 completes, it will be detected by GETF=1 in the R32_I3C_EVR register and the corresponding interrupt (such as...). If enabled, i.e., GETIE = 1 in the R32_I3C_INTENR register, a report will be made.

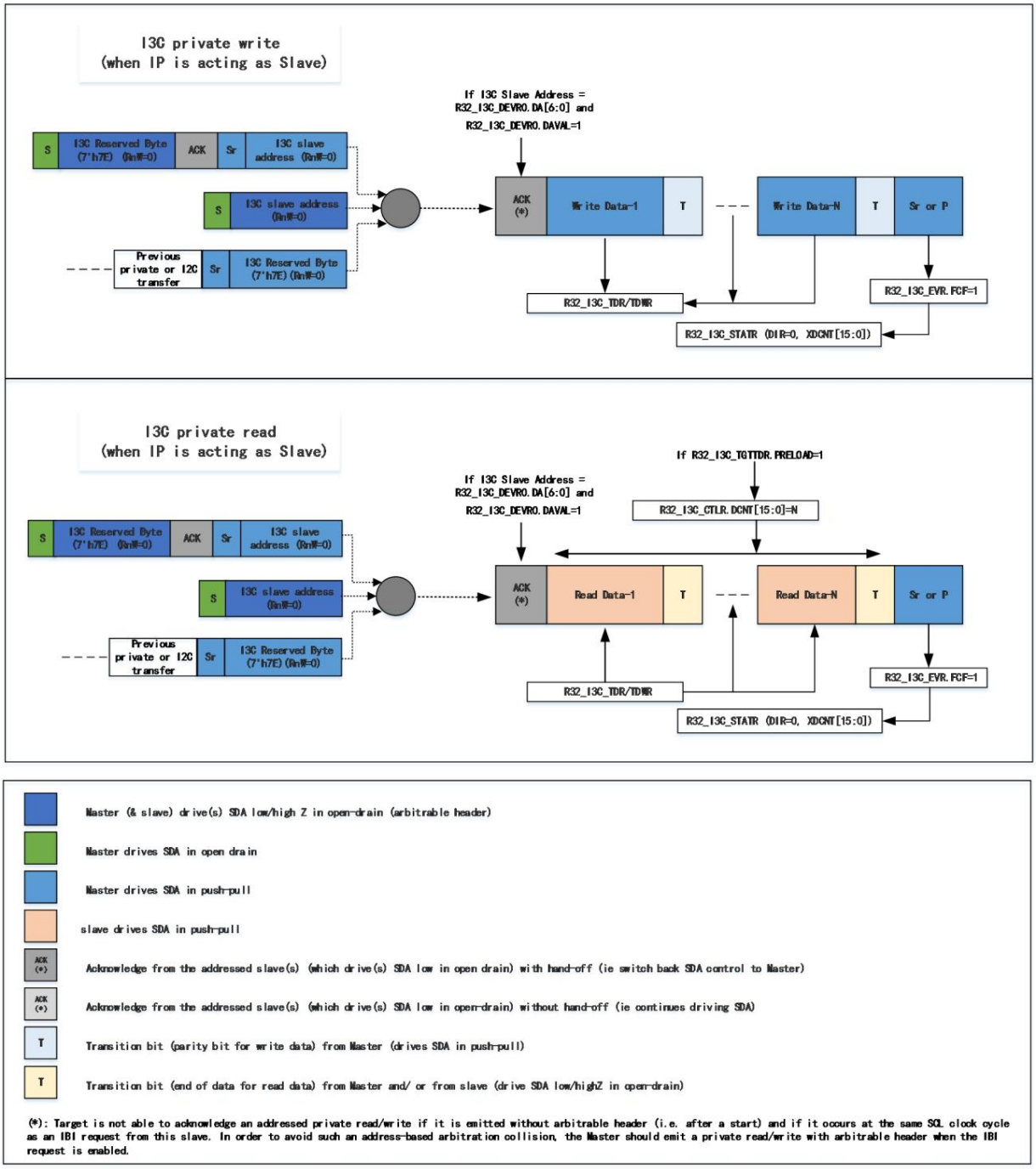
22.3.4.10 I3C Private Read/Write Transfer (as Master Device)

Figure 22-9 I3C Private Read/Write Messages (as Master Device)



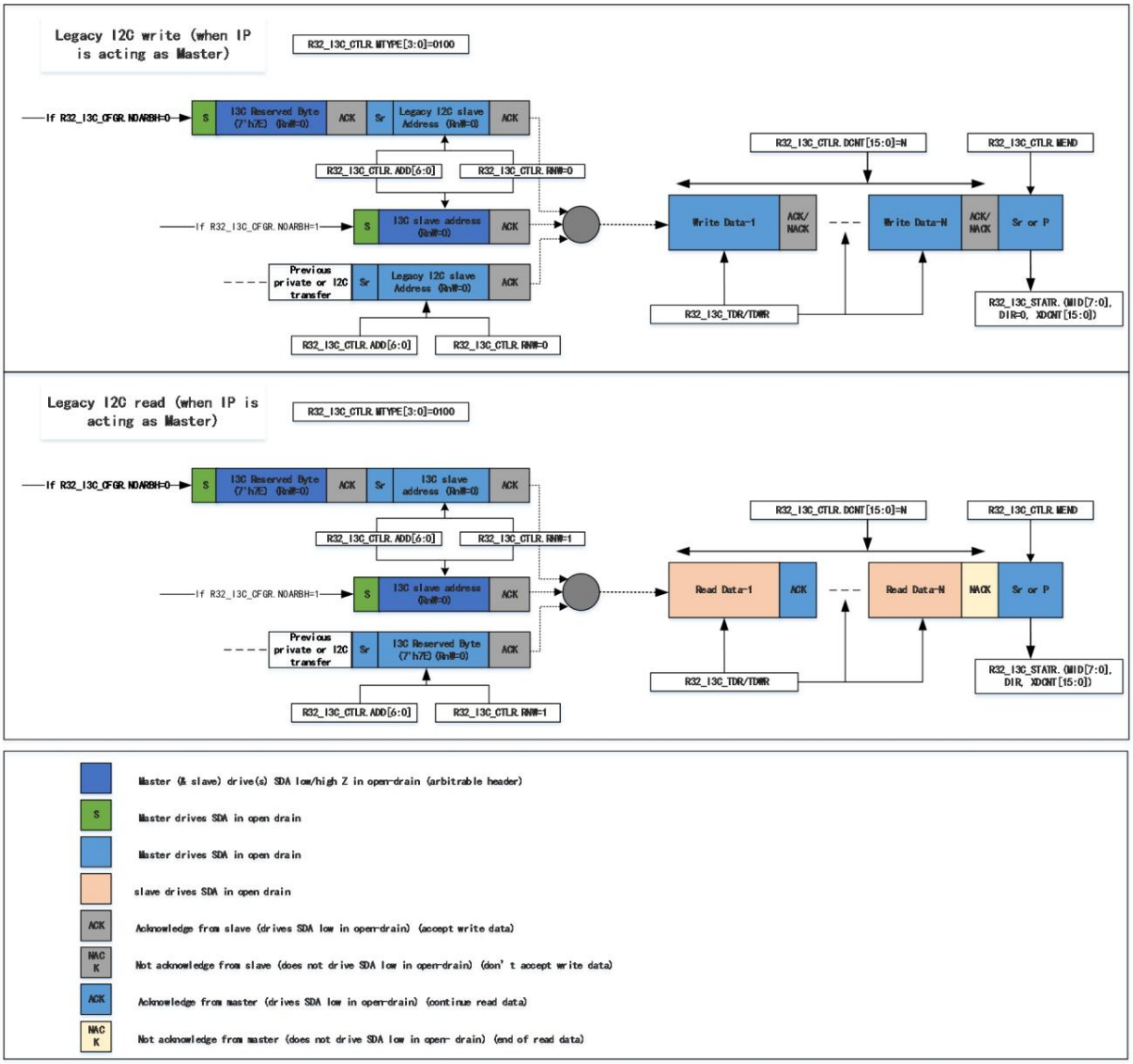
22.3.4.11 I3C Private Read/Write Transfer (as a Slave Device)

Figure 22-10 I3C Private Read/Write Transfer (as a Slave Device)



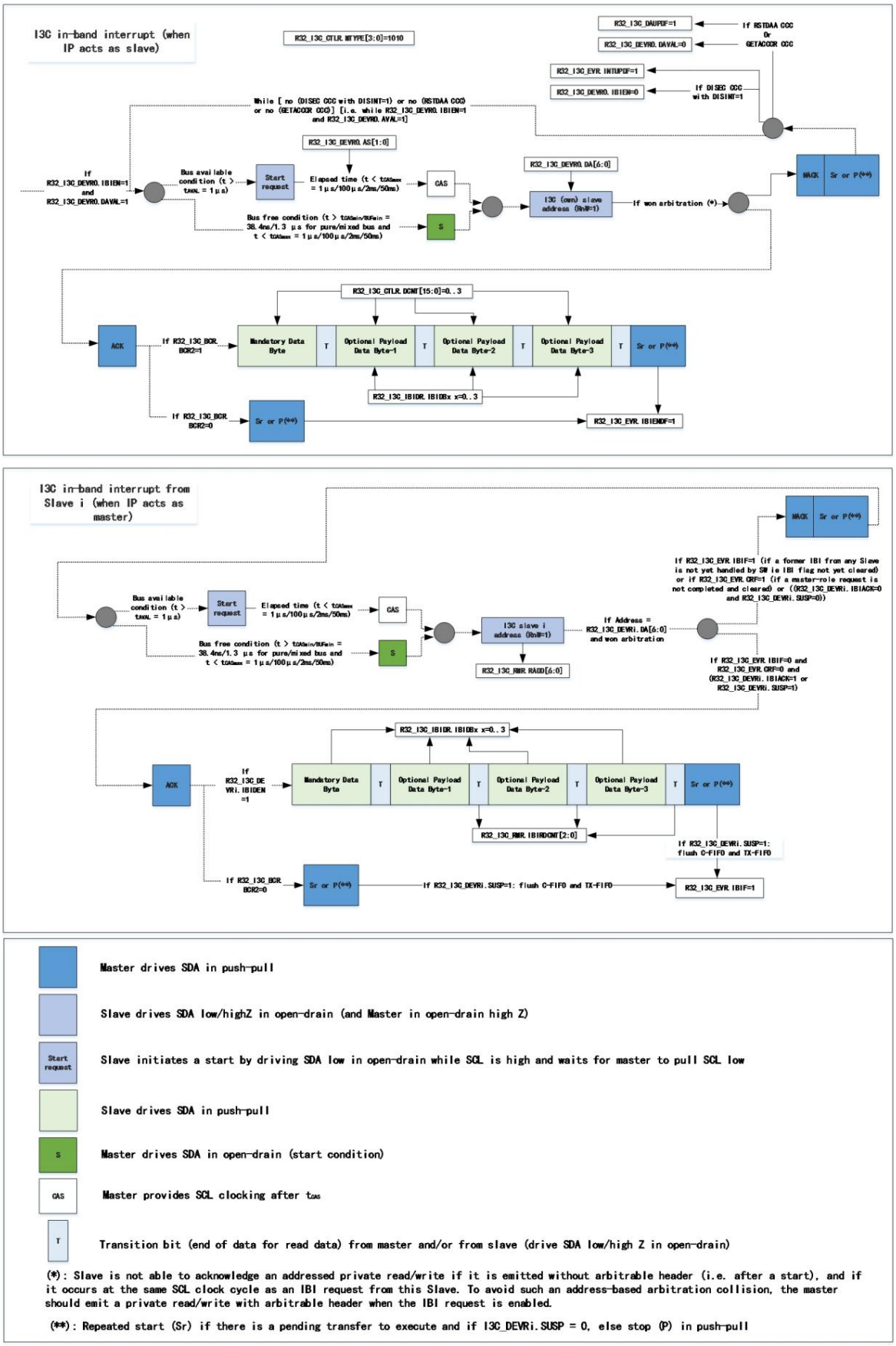
22.3.4.12 Traditional I2C Read/Write Transfer (as Master Device)

Figure 22-11 Traditional I2C read/write transfer (as master device)



22.3.4.13 I3C IBI Transmission (as Master/Slave Device)

Figure 22-12 I3C IBI Transmission (as Master/Slave Device)



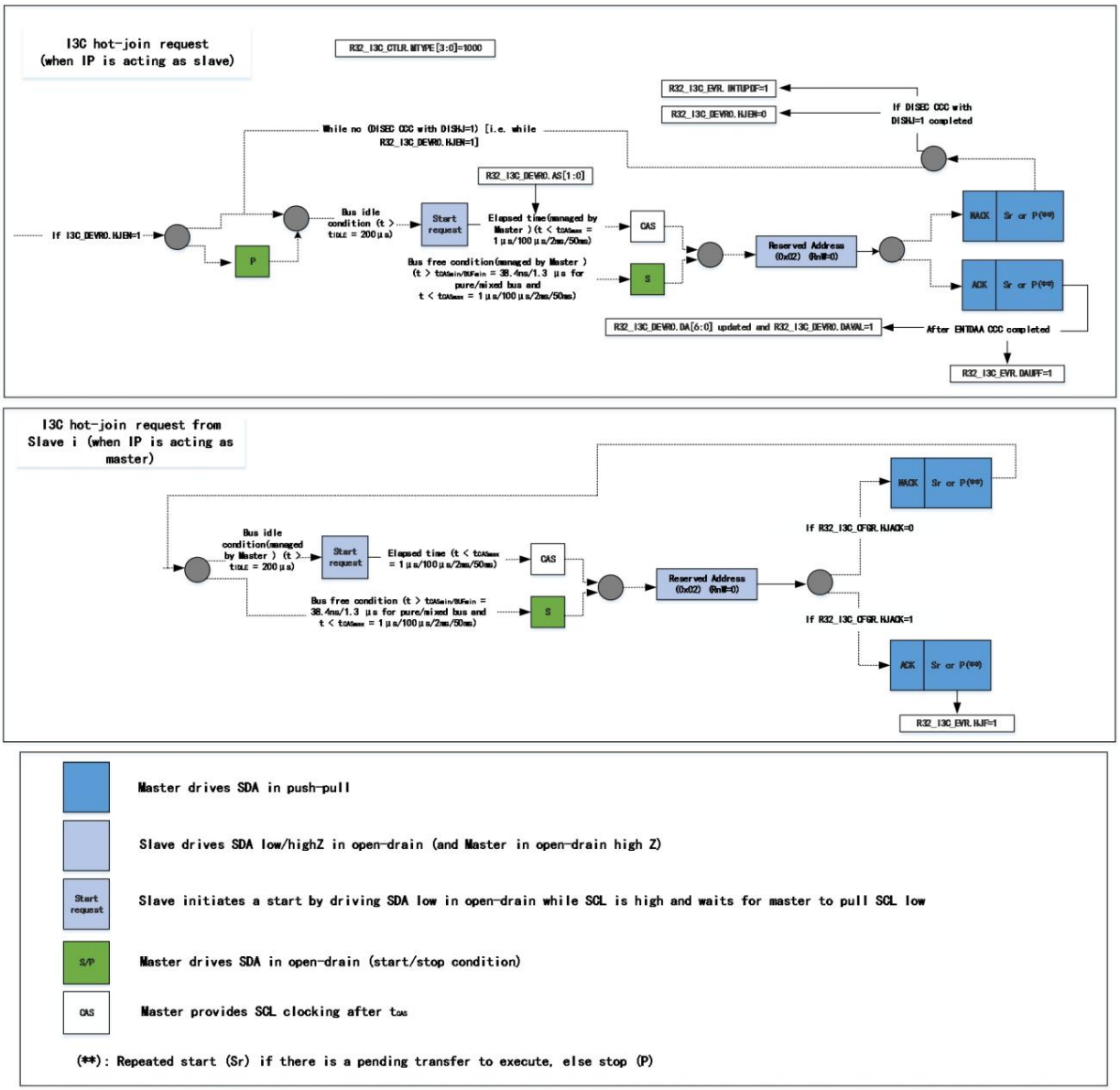
When the peripheral is used as the master device, the R32_I3C_IBIDR register is used to receive the IBI data payload. Therefore, data from the slave device...

IBI requests must not exceed a 4-byte data payload. If more information needs to be exchanged within the context of this in-band interrupt, the master...

Backup software must issue a private read control word.

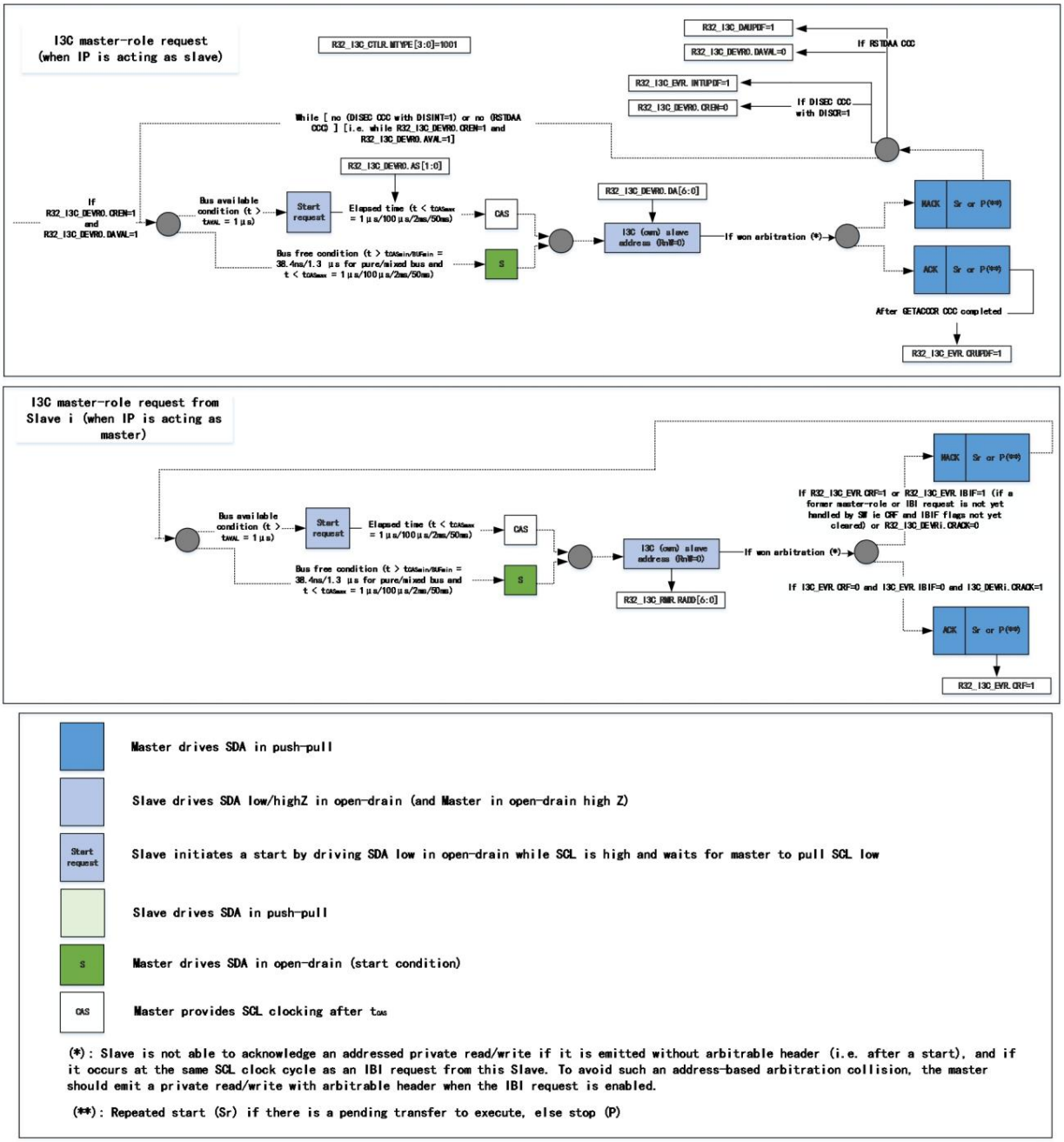
22.3.4.14 I3C Hot-Join Request Transmission (as Master/Slave Device)

Figure 22-13 I3C Hot-Join Request Transmission (as Master/Slave Device)



22.3.4.15 I3C Master Role Request Transmission (as Master/Slave)

Figure 22-14 I3C Master Role Request Transmission (As Master/Slave Device)



22.3.5 I3C FIFO Management

Table 22-2 I3C FIFO Implementation

FIFO	content	Unit size is used as master/slave device (basic principle)	
C-FIFO	32-bit control word	2 characters	The master device (a frame can be based on multiple control words; while) (This is not the case when used as a slave device)
S-FIFO	32-bit status word		Master device (slave device: register-only mode status) state)
Data sent by TX-FIFO		8 bytes	Master and slave devices
Data received by RX-FIFO			

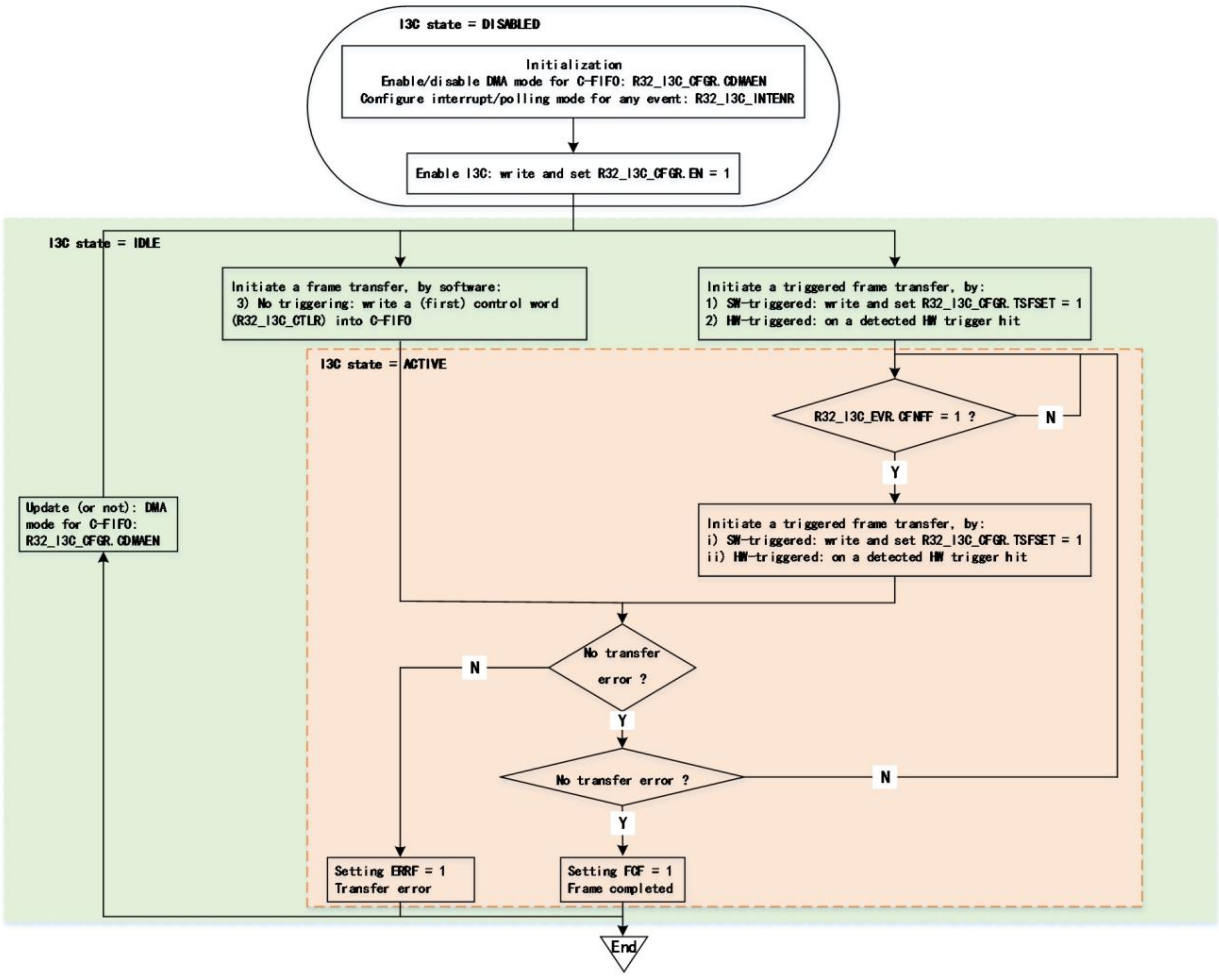
22.3.5.1 C-FIFO Management (as Master Device) When used as a

master device, CCC broadcasting, direct reading/writing, private reading/writing, traditional I2C reading/writing, and software-based reading/writing are all possible within the C-FIFO.

Used during error recovery.

Figure 22-15 illustrates how the C-FIFO is managed to queue control words on the I3C bus when an I3C peripheral is used as a master device.

Figure 22-15 C-FIFO Management (as Master Device)



First, the software must initialize C-FIFO management through the R32_I3C_CFGR register CDMAEN. The specific steps are as follows:

ÿ The software writes directly according to the control word (when CDMAEN=0):

– By polling mode (setting CFNFIE=0 in the R32_I3C_INTENR register): Explicitly writing to the R32_I3C_CTLR register

Before entering, wait for the hardware to request the next control word (CFNFF=1 in the R32_I3C_INTENR register).

– By enabling interrupt notification (when CFNFIE=1):

Write via the allocated DMA channel (when CDMAEN=1) to enable the corresponding I3C peripheral DMA request:

Depending on the module-level configuration, the DMA will automatically push/write control words from its memory source buffer to the R32_I3C_CTLR register.

This continues until the frame is complete (a stop bit is issued on the I3C bus after the last message of the frame), except in case of a transmission error.

• In any case where the C-FIFO is empty and a repeat start bit of a new control word must be issued, a C-FIFO underflow will be reported.

(ERRF in the R32_I3C_EVR register = 1 and COVR in the R32_I3C_STATER register = 1). If enabled.

If the ERRIE bit of the R32_I3C_INTENR register is checked, an interrupt will be generated.

When the I3C peripheral is not in operation, the DMA mode configuration managed by C-FIFO can be modified.

When used as a master device, if a transmission error occurs (ERRF=1 in the R32_I3C_EVR register), the hardware will automatically clear C-...

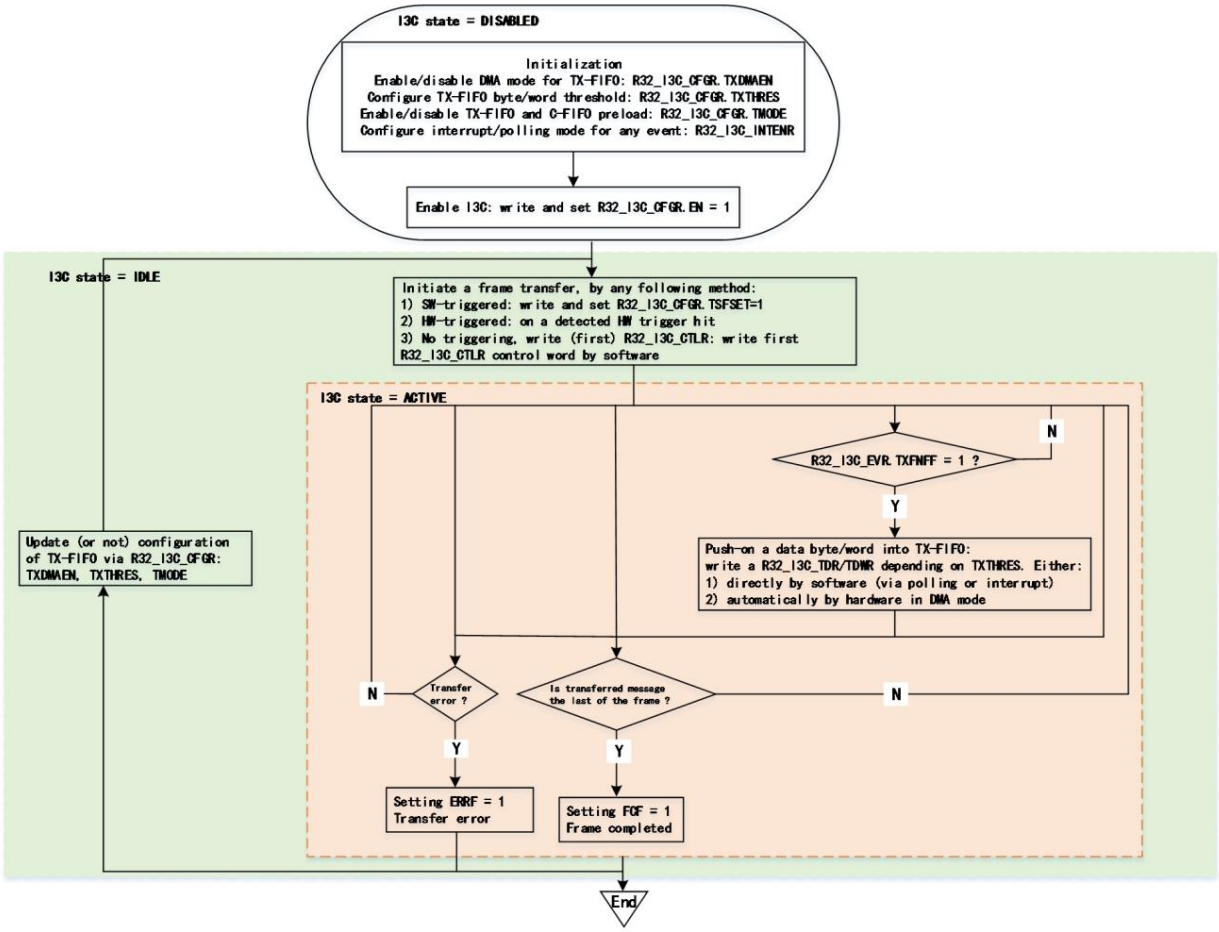
FIFO.

22.3.5.2 TX-FIFO Management (as Master Device)

When used as a master device, it can be broadcast or directly CCC (including ENTDAA and RSTACT) if a parsing byte/subcommand word exists. Bytes or data bytes, private writes, traditional I2C writes using TX-FIFO

Figure 22-16 illustrates how the TX-FIFO is managed to queue data bytes or words to be sent on the I3C bus when an I3C peripheral is used as a master device.

Figure 22-16 TX-FIFO Management (as Master Device)



First, TX-FIFO management must be initialized using the following bit fields in the R32_I3C_CFGR register: ÿ TXDMAEN: Enables/disables DMA mode for the TX-FIFO ÿ TXTHRES: Pushes data bytes or words into the TX-FIFO ÿ TMODE: Enables/ disables TX-FIFO and C-FIFO preloading Then, based on the TXDMAEN bit in the R32_I3C_CFGR register, there are two methods to write data to the TX-FIFO: ÿ Software writes directly byte/word (when TXDMAEN=0):

– By polling mode (R32_I3C_INTENR register TXFNFIE=0): Before explicitly writing to the R32_I3C_TDR or R32_I3C_TDWR register, wait for the hardware to request the next data byte/word (R32_I3C_INTENR register TXFNFF = 1), depending on the specific circumstances. The TXTHRES bit in the R32_I3C_CGFR register

– provides interrupt notification via an enabled interrupt (when TXFNFIE = 1).

Write via the allocated DMA channel (when TXDMAEN = 1) to enable the corresponding I3C peripheral DMA request:

Depending on the DMA module-level configuration, the DMA will automatically push/write data bytes/words from its memory source buffer. The R32_I3C_TDR or R32_I3C_TDWR register (depending on the TXTHRES bit of the R32_I3C_CGFR register) remains in the register until the frame is complete. (A stop bit is sent on the I3C bus after the last message of the frame), except in the event of a transmission error.

I3C messages begin with a start bit or repeat start bit and end with a stop bit or repeat start bit. At the message level, the last data byte/word to be sent is indicated by the R32_I3C_EVR register TXLASTF = 1. When an I3C frame contains multiple messages (starting with a repeat start bit...), the last data byte/word to be sent is indicated by the R32_I3C_EVR register TXLASTF = 1.

When the start-of-bit (NOT) is used, the software can use this event to update the corresponding pointer so that it points to the byte/word used to store the next message.
Buffer.

When a report frame is completed (R32_I3C_EVR register FCF = 1, and the corresponding interrupt is enabled), the TX-FIFO is empty. If
If the TX-FIFO is empty and a data byte must be sent on the I3C bus, an underflow of the TX-FIFO (in the R32_I3C_EVR register) will be reported.
ERRF=1 and R32_I3C_STATER register DOVR = 1. If the corresponding interrupt is enabled (R32_I3C_INTENR register ERRIE = 1), an interrupt will be generated. When
the I3C peripheral is not in operation, the configuration managed by TX-FIFO can be modified.

When used as a master device, the hardware will automatically clear the TX-FIFO if a transmission error (ERRF = 1) occurs.

No C-FIFO/TX-FIFO pre-loaded

The C-FIFO is 2 words in size, and the TX-FIFO is 8 bytes in size.

Without enabling the C-FIFO/TX-FIFO preloading function, i.e., with R32_I3C_CFGR register TMODE=0, once the C-FIFO...
Upon receiving the first control word, the I3C peripheral will immediately issue a start bit on the I3C bus. Subsequently, the peripheral will register R32_I3C_CTLR.
The device decodes the data and writes the next data byte or word as needed. Once a hardware request is detected, the next control signal is sent on the I3C bus.
When a repeat start bit needs to be issued on the I3C bus or a C-FIFO needs to acquire available space, the peripheral will immediately request that the repeat start bit be issued.
The control word is written to the C-FIFO until the last message (i.e., R32_I3C_CTLR register MEND = 1). Similarly, once detected...
The hardware requires sending another data byte or word on the I3C bus (e.g., issuing a repeat start bit on the I3C bus, TX-FIFO).
If the I3C message is not full, or if a data byte or word must be sent during the message, then the data byte/word must be written to the TX-FIFO immediately.

C-FIFO and TX-FIFO pre-loaded

When C-FIFO/TX-FIFO preloading is configured (R32_I3C_CFGR register TMODE = 1), a start bit is issued on the bus.
Previously, I3C peripherals needed to wait as long as possible for the C-FIFO and TX-FIFO to load, as follows:

Waiting for the first control word to be written to the C-FIFO; waiting for data bytes/words to be written to the TX-FIFO (if any), as indicated by the first control word.
Defined (if RNW=0 in the R32_I3C_CTLR register and DCNT[15:0]=0), the maximum write size of the TX-FIFO is: • If the TX-FIFO is not full and the first control word is not
the last message of the frame (MEND=0 in the R32_I3C_CTLR register):

-Wait for the second control word to be written to the C-FIFO, after which the C-FIFO will be full.

– If the TX-FIFO is not full, wait for data bytes/words to be written to the TX-FIFO (if any), as defined in the second control word.
(RNW=0 and DCNT[15:0] in the R32_I3C_CTLR register), the maximum write size is TX-FIFO.

Then, whenever a hardware requirement is detected to send the next control word on the I3C bus (if a repeating command must be sent on the I3C bus), the command will resume operation.
If the start bit is reached, a request will be made to write the control word to the C-FIFO immediately, until the last message (MEND=1 in the R32_I3C_CTLR register).

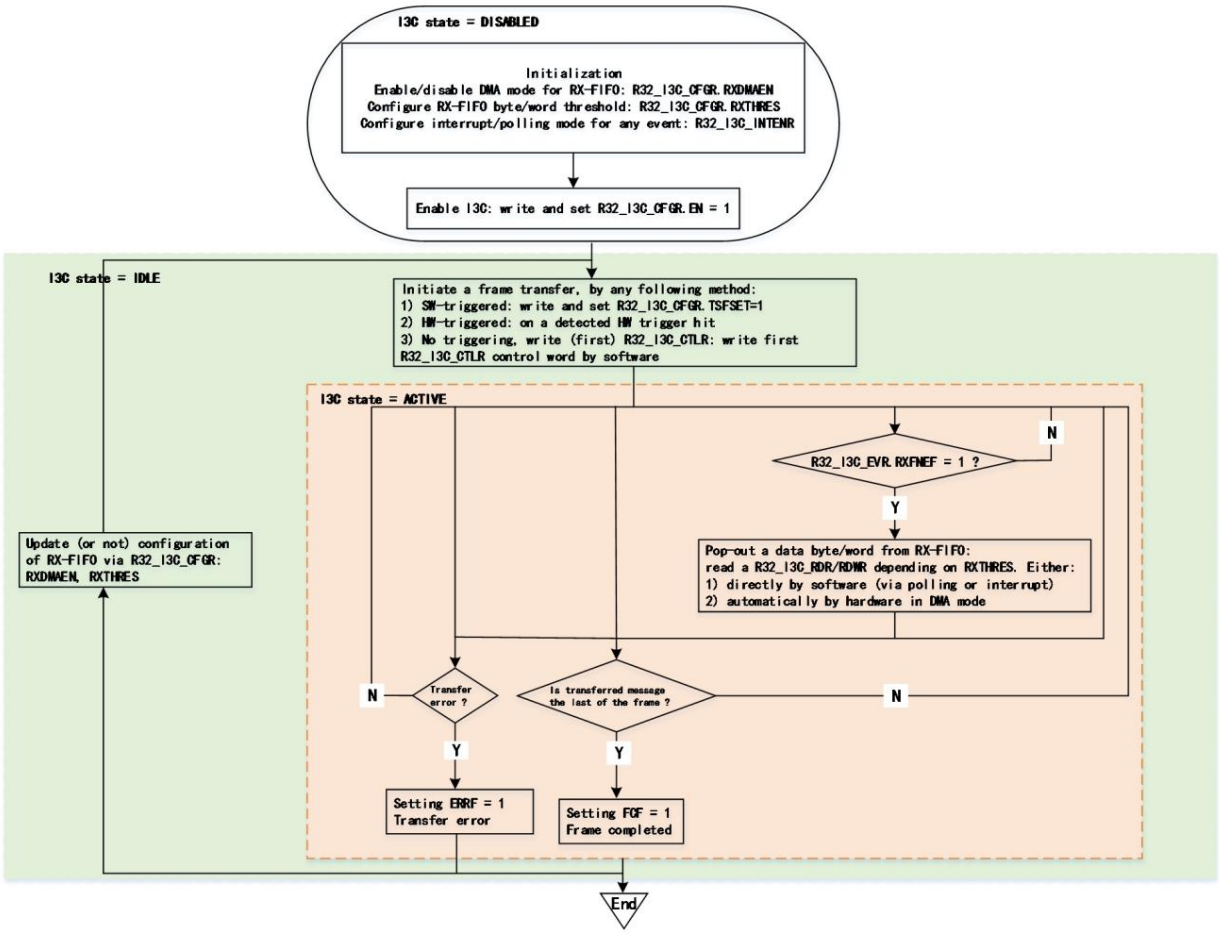
Similarly, as soon as a hardware requirement is detected to send the next data byte/word on the I3C bus (if it must be sent on the I3C bus),
If the start bit is repeated, the TX-FIFO is not full, or a data byte/word will be sent during this I3C message, the data byte/word must be immediately...
Words are written to the TX-FIFO.

22.3.5.3 RX-FIFO Management (as Master Device) When used as

a master device, it can perform broadcast ENTDAA CCC, direct CCC read (including direct RSTACT CCC read), private read, and traditional read.
RX-FIFO is used during I2C reads.

Figure 22-17 illustrates how the RX-FIFO is managed when an I3C peripheral is used as a master device to manage the data words to be received on the I3C bus.
Sections or words are queued and popped up.

Figure 22-17 RX-FIFO Management (as Master Device)



First, the software must initialize RX-FIFO management via the following bit fields of the R32_I3C_CFGR register: ÿ RXDMAEN: Enables/disables DMA mode for the RX-FIFO ÿ RXTHRES: Pops a data byte or word from the RX-FIFO

Then, the RX-FIFO is read according to the RXDMAEN bit: ÿ Software reads directly byte/word (RXDMAEN=0): – By polling mode (RXFNEIE=0 in the R32_I3C_INTENR register): Waits for hardware to request the next data byte/word before explicitly reading the R32_I3C_RDR or R32_I3C_RDWR register (RXFNEF=1 in the R32_I3C_INTENR register), depending on the RXTHRES bit in the R32_I3C_CGFR register – By enabled interrupt notification (if RXFNEIE=1 in the R32_I3C_INTENR register) ÿ By allocated DMA Channel read (if RXDMAEN=1) to enable the corresponding I3C peripheral DMA request: – Depending on the DMA module-level configuration, the DMA will automatically pop/read data bytes/words from the R32_I3C_RDR or R32_I3C_RDWR register (depending on the RXTHRES bit) and write them to its memory from the device buffer until the frame is complete (a stop bit is issued on the I3C bus after the last message of the frame), except in case of a transfer error.

I3C messages begin with a start bit or repeat start bit and end with a stop bit or repeat start bit. At the message level, the last data byte/word received from the I3C bus is indicated by IRXLASTF=1 in the I3C_EVR register. When an I3C frame contains multiple messages (separated by repeat start bits), the software can use this event to update the corresponding pointer to point to the location used to store the next message. A buffer for data bytes/words.

If the RX-FIFO is full and a data byte is received on the I3C bus, an RX-FIFO overflow is reported (ERRF=1 in the R32_I3C_EVR register and DOVR=1 in the R32_I3C_STATER register). An interrupt is generated if the corresponding interrupt is enabled (ERRIE=1 in the R32_I3C_INTENR register).

When the I3C peripheral is not in operation, the configuration managed by RX-FIFO can be modified.

Early termination of read transfers

from the device disables the S-FIFO (SMODE=0 in the R32_I3C_CFGR register), and will use the R32_I3C_EVR register...

RXTGTENDF=1 and the corresponding interrupt (if enabled) notify the software of an early termination of the read transfer. The software can then read the status register.

The register R32_I3C_STATR is used to check information related to the last message and to retrieve the data words received during an early-ended read transfer.

Number of sections (XDCNT[15:0] in the R32_I3C_STATR register).

In any case, if S-FIFO is enabled (SMODE=1 in the R32_I3C_CFGR register), software or DMA (specifically...)

The status register R32_I3C_STATR must be read once for each message, depending on the SDMAEN value in the R32_I3C_CFGR register.

The number of valid data bytes received in a read message that ends prematurely is determined by XDCNT[15:0] in the R32_I3C_STATR register (followed by ABT=1).

Report.

22.3.5.4 S-FIFO Management (as Master Device) When used

as a master device, the software can use the S-FIFO to read the I3C status register (R32_I3C_STATR) for each transmitted message.

Reading the status register when S-FIFO is disabled

When the SMODE bit of the R32_I3C_CFGR register is set to 0, the S-FIFO is disabled, and the status register is read in the same way as a normal register.

Same memory:

When transmitting new messages, the contents of the registers are overwritten by hardware.

R32_I3C_STATR contains the status of the last transmitted message .

The SCL clock will not stop if the status register is not read .

When SMODE=0, that is, when the slave device prematurely terminates the private read transfer under specific circumstances:

The software is notified by setting the R32_I3C_EVR register RXTGTENDF bit to 1 and the corresponding interrupt (if enabled).

Then , the software sets the CRXTGTENDF bit in the R32_I3C_CEVR register to 1 via a write operation to clear the event flag.

forward:

- No more data bytes are received on the I3C bus; instead, they are written to the R32_I3C_RDR/R32_I3C_RDWR registers by hardware.

– Cannot update R32_I3C_STATR – Pause

the SCL clock (if necessary)

Normally, when the FCF=1, ERRF=1, or RXTGTENDF=1 of the R32_I3C_EVR register, the R32_I3C_STATR register can be read.

Registers. After the slave device prematurely terminates the read transfer, the XDCNT[15:0] in the R32_I3C_STATR register can be read to obtain the private...

The number of valid data bytes received during read transmission.

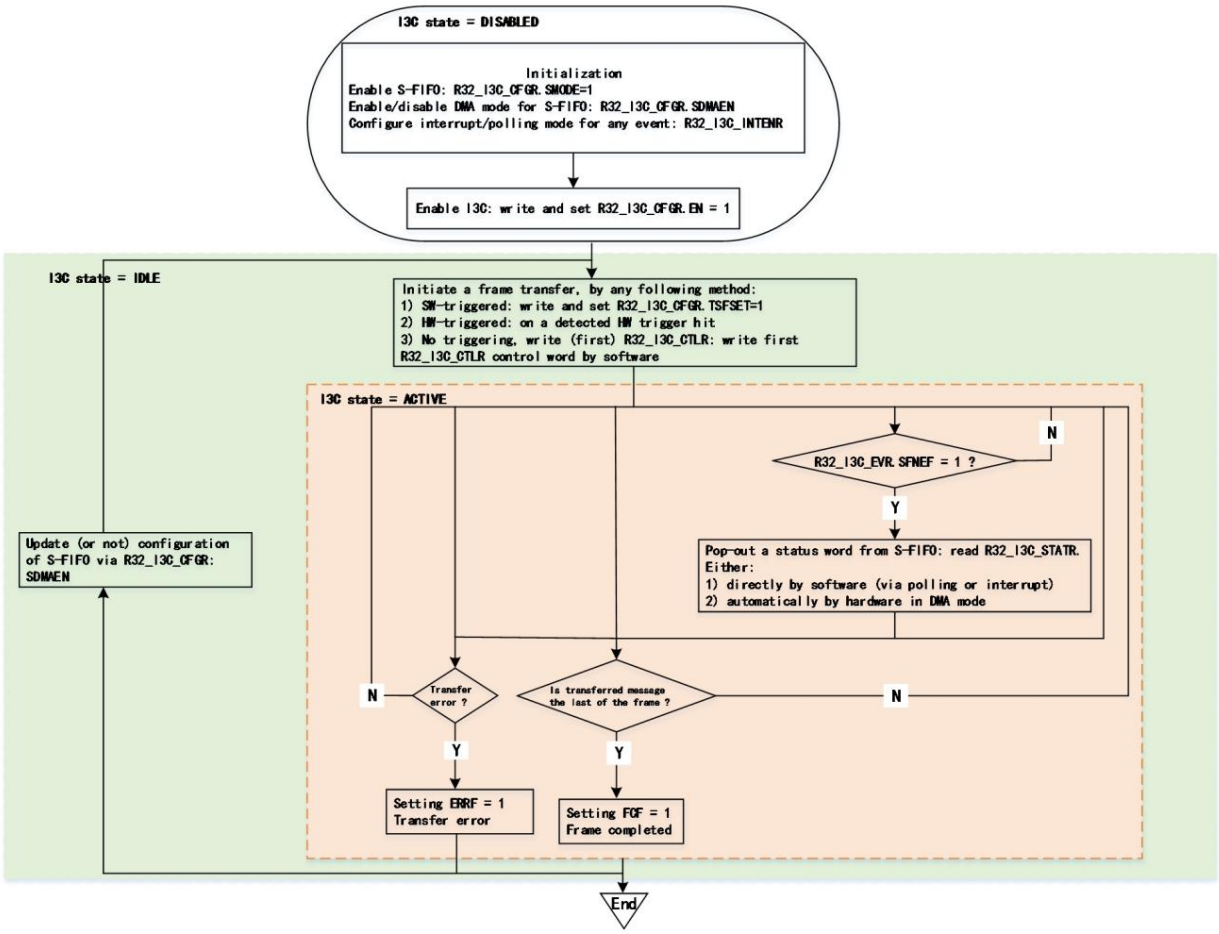
When S-FIFO is enabled, read the status register.

If the SMODE bit of the R32_I3C_CFGR register is set to 1, then S-FIFO is enabled.

Figure 22-18 illustrates how the S-FIFO is managed to process each message executed on the I3C bus when an I3C peripheral is used as a master device.

Status words are queued and popped up.

Figure 22-18 S-FIFO Management (as Master Device)



First, the software must initialize the SDMAEN bit field (enable/disable DMA mode of S-FIFO) in the R32_I3C_CFGR register. Initialize S-FIFO management. Then, read the S-FIFO based on the SDMAEN bit: ÿ Directly by software (if SDMAEN=0): – By polling mode (SFNEIE=0 in the R32_I3C_INTENR register): before explicitly reading the I3C_STATR register. Waiting for hardware to request the next status word (SFNEF=1 in the R32_I3C_INTENR register).

–By enabling interrupt notification (SFNEIE=1)

Read from the allocated DMA channel (if SDMAEN=1) to enable the corresponding I3C peripheral DMA request (i3c_rs_dma):

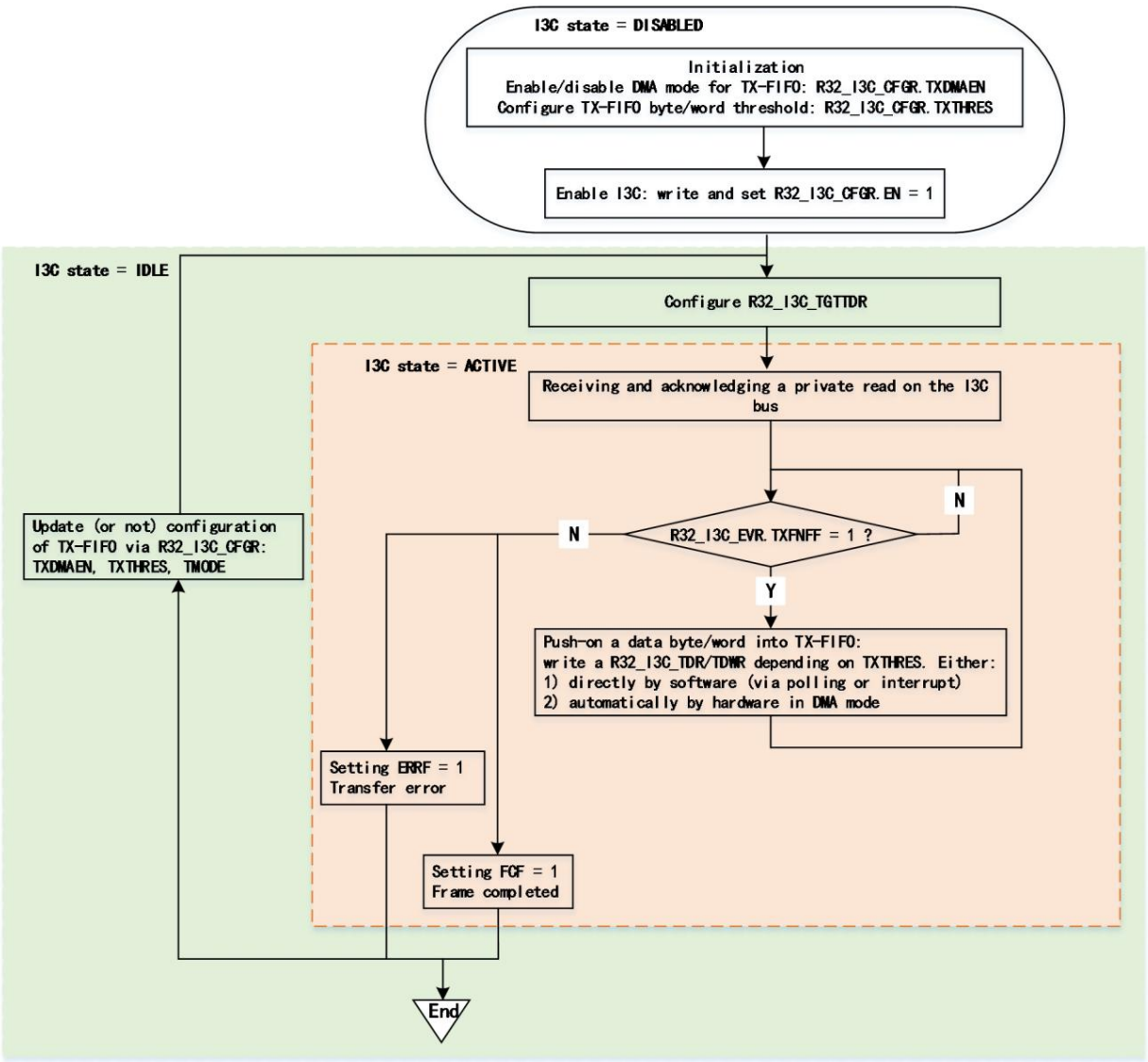
Depending on the DMA module-level configuration, the DMA automatically pops/reads status words from the R32_I3C_STATR register and writes them to its memory from the device buffer until the frame is complete (a stop bit is issued on the I3C bus after the last message of the frame), except in case of a transfer error. Every message status must be read; otherwise, when the S-FIFO is full and the next message status must be written, the hardware sets the overflow error flag to 1 (ERRF in the R32_I3C_EVR register = 1 and COVR in the R32_I3C_STATER register = 1). If the corresponding interrupt is enabled (ERRIE in the R32_I3C_INTENR register = 1), an interrupt is generated.

A frame completion will only be reported when the S-FIFO is empty (FCF=1 in the R32_I3C_EVR register).

The configuration managed by the S-FIFO can be modified when the I3C peripheral is not in operation.

22.3.5.5 TX-FIFO Management (as a Slave Device) When used as a slave device, the TX-FIFO can only be used during private read transfers. Figure 22-19 illustrates how the TX-FIFO is managed to queue and push data bytes or words to be sent on the I3C bus when an I3C peripheral is used as a slave device.

Figure 22-19 TX-FIFO Management (as a slave device)



First, the software must initialize TX-FIFO management and write the following bits into R32_I3C_CFGR:

ÿ TXDMAEN: Enables/disables DMA mode for TX-FIFO

ÿ TXTHRES: Pushes data bytes or words into the TX-FIFO

Then, before receiving private read transfers on the I3C bus, the software must configure the I3C slave device to send configuration registers.

(R32_I3C_TGTTDR) preloads a certain number of data bytes into the TX-FIFO (written in a single access to TGTTDCNT[15:0] and...

PRELOAD=1), thus preparing to send data bytes from the slave device on the I3C bus:

If PRELOAD=1 and TGTTDCNT[15:0]>TX-FIFO size, then TX-FIFO will be preloaded to the FIFO size.

If PRELOAD=1 and TGTTDCNT[15:0]ÿTX-FIFO size, then TX-FIFO will be preloaded into TGTTDCNT[15:0].

Preload TX-FIFO according to TXDMAEN bit:

• Preloaded directly by software by byte/word (TXDMAEN=0):

– By polling mode (R32_I3C_INTENR register TXFNFIE=0): Explicitly write R32_I3C_TDR or R32_I3C_TDWR

The register is waiting for a hardware request for the next data byte/word (R32_I3C_INTENR register TXFNFF=1), specifically whether it is a byte or...

The word depends on TXTHRES in the R32_I3C_CGFR register.

– By enabling interrupt notification (if TXFNFIE=1)

• Preload the allocated DMA channel (TXDMAEN=1) to enable the corresponding I3C peripheral DMA request (i3c_tx_dma):

– Depending on the DMA module-level configuration, the DMA will automatically push/write data bytes/words from its memory source buffer into or into the R32_I3C_TDR or R32_I3C_TDWR register (depending on TXTHRES) until the transfer is complete (R32_I3C_EVR register FCF=1), except in the event of a transfer error.

Then, when there is still data to be loaded into the TX-FIFO in TGTTDCNT[15:0] of the R32_I3C_TGTTDR register to continue the private read transfer (PRELOAD set to 1 and TGTTDCNT[15:0] > TX-FIFO size), if the master device has not yet completed the private read transfer, preloading is performed in the same way (directly through software or through the allocated DMA

channel): – If TXTHRES=0: When one byte is sent on the I3C bus, the next byte is preloaded into the TX-FIFO. – If

TXTHRES=1: When four bytes are sent on the I3C bus, the next word is preloaded into the TX-FIFO. The private read

transfer is complete when either the slave or master device terminates the data byte transfer first (FCF in the R32_I3C_EVR register=1). After the transmission

is complete, the software can: ÿ Read XDCNT[15:0] from the R32_I3C_STATR register: the number of

valid data bytes sent . ÿ Read TGTTDCNT[15:0] from the R32_I3C_TGTTDR register: the number of remaining bytes to be loaded and

sent on the I3C bus. ÿ Clear the TX-FIFO: whether to set TXFLUSH in the R32_I3C_CFGR register to 1 via a write operation and continue to the next step.

Secondary private read transfers depend on the user application.

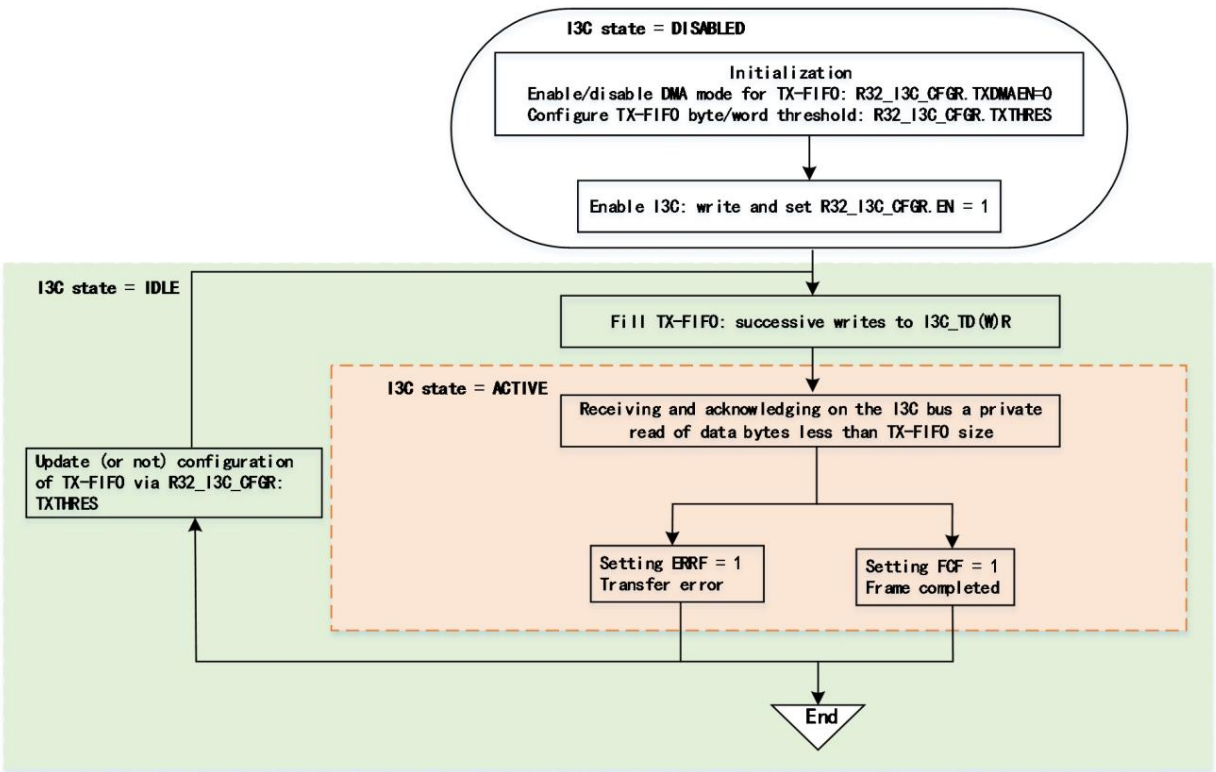
If the TX-FIFO is empty and a data byte must be sent on the I3C bus, an underflow of the TX-FIFO will be reported (R32_I3C_EVR register ERRF=1 and R32_I3C_STATER register DOVR=1). If the corresponding interrupt is enabled (R32_I3C_EVR register ERRIE=1), an interrupt will be generated. The

configuration for managing the TX-FIFO can be modified when the I3C peripheral is not in operation.

If the number of bytes is less than the TX-FIFO size, you can choose not to use R32_I3C_TGTTDR.

In addition to using the R32_I3C_TGTTDR register, as shown in Figure 22-20, when DMA is not used (TDMAEN=0), if the number of data bytes to be read on the I3C bus is less than the TX-FIFO size, the software can directly use the number of bytes it reads to prepare and fill the TX-FIFO by continuously writing to R32_I3C_TDR or R32_I3C_TDWR (depending on TXTHRES).

Figure 22-20 If the number of bytes read is less than the TX-FIFO size, then I3C_TGTTDR is not used, and the TX-FIFO is managed by software (as a slave device).



22.3.5.6 RX-FIFO Management (as a Slave Device) When

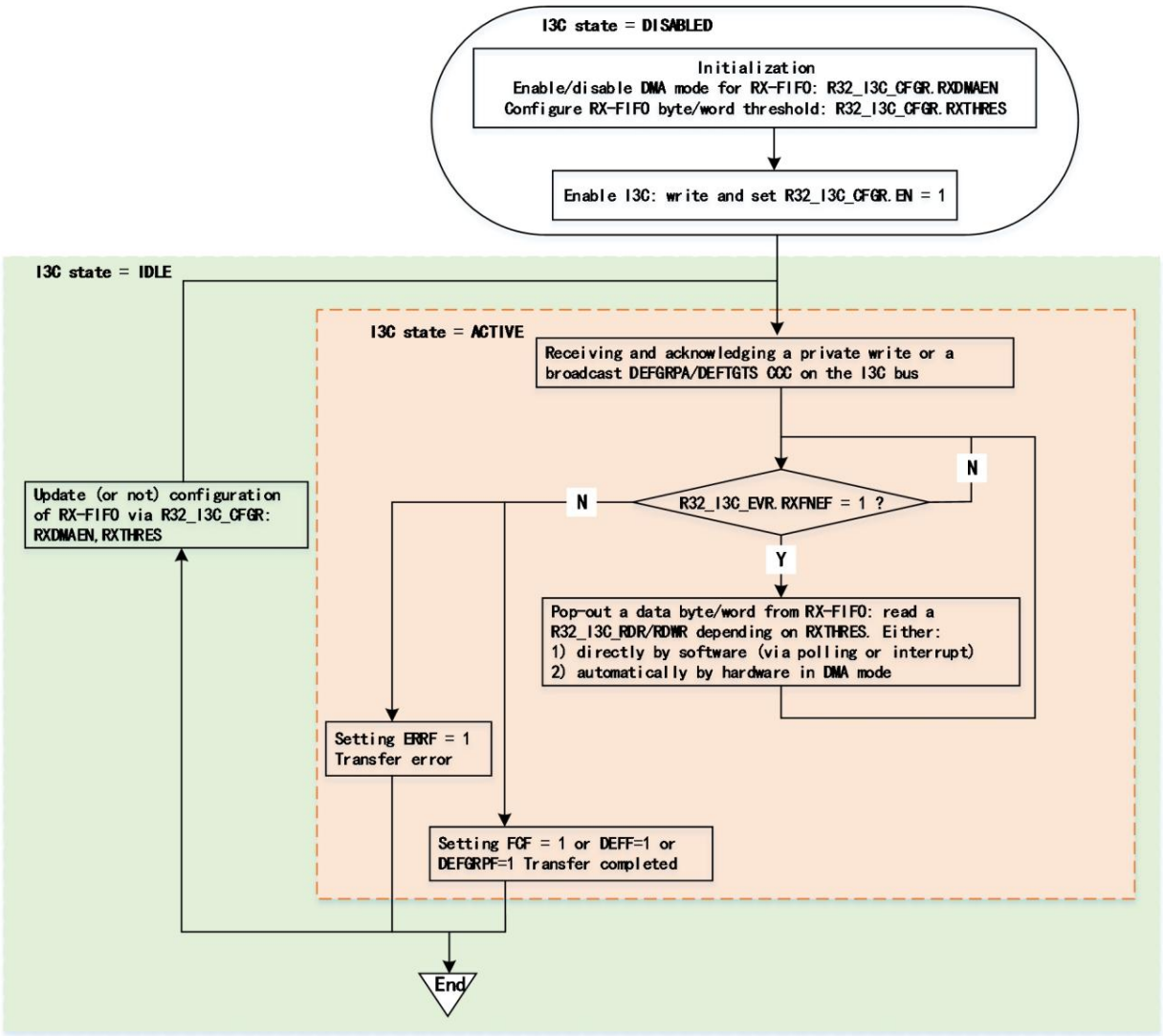
used as a slave device, it can broadcast DEFTGTS CCC, broadcast DEFGRPA CCC, and privately write received and acknowledged transmission periods.

Using RX-FIFO:

Figure 22-21 illustrates how the RX-FIFO is managed when an I3C peripheral is used as a slave device to manage data bytes received from the I3C bus.

The / character is used for queuing and popping.

Figure 22-21 RX-FIFO Management (as a slave device)



First, the software must initialize RX-FIFO management through the following R32_I3C_CFGR register bit fields:

ÿ RXDMAEN: Enables/disables DMA mode for the RX-FIFO;

ÿ RXTHRES: Pop data bytes or words from the RX-FIFO;

Then, read the RX-FIFO based on the RXDMAEN bit:

• Read directly by software byte/word (if RXDMAEN=0):

– By polling mode (RXFNEIE=0 in the R32_I3C_INTENR register): Explicitly read R32_I3C_RDR or

The R32_I3C_RDWR register waits for a hardware request for the next data byte/word (RXFNEF=1 in the R32_I3C_INTENR register).

Whether it's a byte or a word depends on RXTHRES in the R32_I3C_CGFR register –

notified via an enabled interrupt (if RXFNEIE=1).

Read from the allocated DMA channel (if RXDMAEN=1) to enable the corresponding I3C peripheral DMA request (I3C_RX_DMA):

Depending on the DMA module-level configuration, the DMA will automatically pop/read data from the R32_I3C_RDR or R32_I3C_RDWR registers.

Bytes/words (depending on RXTHRES) are written to their memory from the device buffer until the transfer is complete (R32_I3C_EVR).

The register has FCF=1 (private write), GRPF=1 (DEFGRPA CCC), or DEFF=1 (DEFTGTS CCC) in it, but a transmission error occurs.

Except for delays.

If the RX-FIFO is full and a new data byte is received on the I3C bus, an RX-FIFO overflow (R32_I3C_EVR register) will be reported.

If ERRF in the register is 1 and DOVR in the R32_I3C_STATER register is 1, then generate the corresponding interrupt (if enabled).

When the transfer is complete (FCF=1, GRPF=1, or DEFF=1 in the R32_I3C_EVR register):

ÿ RX-FIFO is empty

ÿ Before the software has finished processing the RX data buffer corresponding to the completed private write/DEFTGTS/DEFGRPA transfer (i.e., ...

The software has not yet cleared the corresponding flags by setting CFCF/CDEFF/CGRPF to 1 in the R32_I3C_CEVR register via a write operation.

Before (the next private write/DEFTGTS CCC/DEFGRPA CCC) is received, the hardware must transfer the data bytes.

Writing to the RX-FIFO will report an RX-FIFO overflow error (ERRF in the R32_I3C_EVR register = 1 and...).

Set the R32_I3C_STATER register DOVR=1 and generate the corresponding interrupt (if enabled).

When the I3C peripheral is not in operation, the configuration managed by RX-FIFO can be modified.

22.3.6 I3C Error Management

22.3.6.1 Master Device Error Management

Table 22-3 Main Equipment Error Management

Error Type	illustrate	Error detection	Main equipment operation	Report error
CE0	Illegal formatting of CCC (e.g., return) Data bytes few)	Direct CCC read transfer Reading from the device prematurely ended Retrieve data	Hardware sends stop bit	ERRF=1, PERR=1 and CODERR[3:0]=0000b
CE1	Monitoring errors	In traditional I2C read transmission Incorrect result detected at the end ACK	Hardware keeps SCL running The state lasted for nine hours. Clock cycles can be intersected Replace with another byte, then Then another NACK was sent. Followed by stop position	ERRF=1, PERR=1 and CODERR[3:0]=0001b
		I3C SDR Read Transfer Node Unable to generate duplicates during the bundle Start bit or stop bit	SCL remains running. The software can be used to... R32_I3C_CTLR register Device writes control word MTYPE[3:0]=0000 (News headline) Stop Stop SCL, then usually Must wait at least It takes 150µs to emit another A message	
		CE1 error detected After that, it was unable to generate a starter. Bit		
CE2	No broadcast address response (1111110b)	During message sending (Upgrade fault or reset) (excluding time-series messages) Header detected (1111110b+RnW=0) The new	Hardware issues HDR exit Timing, followed by stop bit	ERRF=1, PERR=1 and CODERR[3:0]=0010b
CE3	Master equipment switching failure defeat	master device was NACKed in SDA. After being low (through) Test header or from	Hardware sends start bit +1111110b+RnW=0, Followed by the slave device	ERRF=1, PERR=1 and CODERR[3:0]=0011b

		Device startup request (after the action) and in its work Delay for state definition The time has ended but has not yet been completed. SCL drive is low level	ACK/NACK, the last one is Stop bit	
-	In GETACCCR Returned in CCC With parity check The 7 bits are set from the design Incorrect backup address	Incorrect movement detected State address and/or parity correction Verification	Hardware can cancel GETACCCR CCC by issuing Restart +1111110b+RnW=0 Followed by a slave device ACK/NACK, finally It is a stop bit.	ERRF=1 and DERR=1
-	Addressed from The equipment was used for the first time in direct Connect to CCC for read transmission NACK	NACK detected	Hardware execution single re- The method is to issue a heavy The start bit + 7 bits are the same From device address +RnW=1.	-
-	Addressed from the device Preparing for the second time in direct Connect to CCC write transmission I3C private read/ Write transmission, traditional I2C transmission or direct Connect to CCC read transfer NACK	NACK detected	The hardware issues a stop bit. ERRF=1 and ANACK=1.	
-	Distributed/sent bands Parity bit Dynamic address first time In ENTDAA CCC NACK	NACK detected	Hardware performs a single retry. To configure a loop, the method is to issue... Repeat start position +1111110b+RnW=1, Followed by (highest priority) (Level) from device ACK and Read 8 bytes of data, then... It is the allocated address + parity. Check digit.	-
-	Distributed/sent bands Parity bit Dynamic address second Next in ENTDAA In CCC, data written	NACK detected	The hardware issues a stop bit. ERRF=1 and DNACK=1	
-	by NACK is traditional During I2C write transmission NACK			
-	Compared to the I3C bus Time sequence, not in time Write/Read Controller Production character, status character Sending data or Read data	SCL pause timeout, hardware issues stop bit. ERRF=1 and (COVR=1) (or DOVR=1)		

22.3.6.2 From Device Error Management

Table 22-4 From Device Error Management

Error Type	illustrate	Error detection master device operation		Report error
TE0	DAA-assigned broadcast address(1111110b+RnW=0) Invalid or 7-bit dynamic Address + RnW = 1 is invalid	At the start position or repeat Detected after the start bit Prohibited address hardware awaits HDR exit Time series		ERRF=1, PERR=1 and ODERR[3:0]=1000b
TE1	CCC code	CCC code detected There is a parity check error. error		ERRF=1, PERR=1 and ODERR[3:0]=1001b
TE2	Write data	Write messages privately in I3C Data words were detected in the middle. Parity check error in section error	Hardware waiting for stop bit Or repeat the start position	ERRF=1, PERR=1 and ODERR[3:0]=1010b
TE3	During dynamic address arbitration Allocated address	In ENTDAA CCC Dynamic allocation detected Parity check in the address mistake		ERRF=1, PERR=1 and ODERR[3:0]=1011b
TE4	During dynamic address arbitration Sr position missing 1111110b+RnW=1	Dynamic address interference detected Sr position missing during the layoff period few {1111110b+RnW=1}		ERRF=1, PERR=1 and ODERR[3:0]=1100b
TE5	What happens after CCC is detected Service	Direct read/write in CCC During transmission, no error was detected. Effective CCC direction (phase) For address phase provided (direction)		ERRF=1, PERR=1 and ODERR[3:0]=1101b
TE6	Monitoring errors	In SDR data read transfer During the period (in direct CCC) Read, private read, or IBI Detected during transmission The SDA is released by hardware using an outlier.		ERRF=1, PERR=1 and ODERR[3:0]=1110b
-	SDR data read transfer period SCL monitoring	In SDR data read transfer During the period (in direct CCC) Read, private read, or IBI Detected during transmission SCL's settling time exceeds 125µs	Wait for the stop bit or Repeat start position	ERRF=1 and STALL=1
-	The data to be sent has not yet been sent. Write/Unavailable (Compared to I3C) (Line timing)	In private read transfers, Software or DMA not in time Write the data to be sent	Hardware detected stop bit Or repeat the start position	ERRF=1 and DOVR=1
-	Unable to read the connection in time Received data (phase)	In private writing, DEFTGTS or		

	For I3C bus timing)	In DEFGRPA CCC, Software or DMA not in time Read the received data		

22.3.7 I3C Low Power Mode

In low-power mode, I3C is unaffected when the device is in sleep mode, and an I3C interrupt can bring the device out of sleep mode.

When in low-power mode, changes on the SCL and SDA buses will wake up the system and exit low-power mode.

22.3.8 I3C Interrupts

Table 22-5 I3C Interrupt Requests

Interruption event	Interruption Division	Used as Main Design Preparation	Used as From the design Preparation	Interrupt enable control	event flag	event clear control
Request control word	ask	×	-	CFNFIE	CFNFF	-
Status word		×	-	SFNEIE	SFNEF	-
available. Requesting		×	×	TXFNFIE	TXFNFF	-
data to be sent. Master		×	×	RXFNEIE	RXFNEF	-
device: Frame transmission complete. From device: Private transfer completed		×	×	FCIE	FCF	CFCF
The device prematurely terminates private read/transfer. Input (and R32_I3C_CFGR sent) (The SMODE bit of the register is 0)		×	-	RXTGTENDIE	RXTGTENDF	CRXTGTENDF
Received IBI request,		×	-	IBIIE	IBIF	CIBIF
received master device role		×	-	CRIE	CRF	CCRF
request, received hot-join request.		×	-	HJIE	HJF	CHJF
IBI request completed		-	×	IBIENDIE	IBIENDF	CIBIENDF
I3C bus start bit lost		-	×	WKPIE	WKPF	CWKPF
Received direct GETACCR		-	×	CRUPDIE	CRUPDF	CCRUPDF
CCC Received direct		-	×	STAIE	STAF	CSTAF
GETSTATUS CCC Received any direct GETxxx CCC (except GETSTATUS)		-	×	GETIE	GETF	CGETF
Dynamic address update (received broadcast) ENTDAA, RSTDAA (broadcast) or direct (followed by SETNEWDA)		-	×	DAUPDIE	DAUPDF	CDAUPDF
Received direct SETMWL		-	×	MWLUPDIE	MWLUPDF	CMWLUPDF
CCC Received direct		-	×	MRLUPDIE	MRLUPDF	CMRLUPDF
SETMRL CCC		-	×	RSTIE	RSTF	CRSTF
Detected reset timing bus status update (received) Direct/Broadcast ENTASx CCC)		-	×	RSTIE	ASUPDF	CASUPDF
Received broadcast/direct ENEC/DISEC CCC		-	×	INTUPDIE	INTUPDF	CINTUPDF
Received broadcast DEFTGTS CCC		-	×	DEFIE	DEFF	CDEFF
Received broadcast DEFGRPA		-	×	GRPIE	GRPF	CGRPF
CCC Error Occurred	Error ×		×	ERRIE	ERRF	CERRF

22.4 Register Description

Table 22-6 List of I3C Relevant Registers

name	Access Address Description:	Reset value
R32_I3C_CTLR	0x40014400 I3C Control Register;	0x00000000
R32_I3C_CFGR	0x40014404 I3C Configuration	0x00000000
R32_I3C_RDR	Register; 0x40014410 I3C Receive Data Byte	0x00000000
R32_I3C_RDWR	Register; 0x40014414 I3C Receive Data Word Register	0x00000000
R32_I3C_TDR	0x40014418 I3C Transmit Data Byte	0x00000000
R32_I3C_TDWR	Register 0x4001441C I3C Transmit Data Word	0x00000000
R32_I3C_IBIDR	Register 0x40014420 I3C IBI Payload Data Register 0x00000000 0x40014424	
R32_I3C_TGTTDR	I3C Slave Device Transmit Configuration Register	0x00000000
R32_I3C_RESET	0x4001442C I3C Reset Register	0x03000000
R32_I3C_STATR	0x40014430 I3C Status Register	0x00000000
R32_I3C_STATER	0x40014434 I3C Status Error Register	0x00000000
R32_I3C_RMR	0x40014440 I3C Receive Message Register	0x00000000
R32_I3C_EVR	0x40014450 I3C Event Register	0x00000003
R32_I3C_INTENR	0x40014454 I3C Interrupt Enable Register	0x00000000
R32_I3C_CEVR	0x40014458 I3C Clear Event Register	0x00000000
R32_I3C_DEVR0	0x40014460 I3C Device Register 0	0x00000000
R32_I3C_DEVR1	0x40014464 I3C Device Register 1	0x00000000
R32_I3C_DEVR2	0x40014468 I3C Device Register	0x00000000
R32_I3C_DEVR3	2 0x4001446C I3C Device Register 3	0x00000000
R32_I3C_DEVR4	0x40014470 I3C Device Register 4	0x00000000
R32_I3C_MAXRLR	0x40014490 I3C Maximum Read Length Register	0x00000000
R32_I3C_MAXWLR	0x40014494 I3C Maximum Write Length Register	0x00000000
R32_I3C_TIMINGR0	0x400144A0 I3C Timing Register 0	0x00000000
R32_I3C_TIMINGR1	0x400144A4 I3C Timing Register 1	0x00000000
R32_I3C_TIMINGR2	0x400144A8 I3C Timing Register 2	0x00000000
R32_I3C_BCR	0x400144C0 I3C Bus Feature Register;	0x00000000
R32_I3C_DCR	0x400144C4 I3C Device Feature Register	0x00000000
R32_I3C_GETCAPR	0x400144C8 I3C Get Function	0x00000000
R32_I3C_CRCAPR	Register 0x400144CC; I3C Master Role Function	0x00000000
R32_I3C_GETMDSR	Register 0x400144D0; I3C Get Maximum Data Rate Register 0x00000000	
R32_I3C_EPIDR	0x400144D4; I3C Extended Configuration ID Register	0x00000000

22.4.1 I3C Control Register (R32_I3C_CTLR) Offset

Address: 0x00

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
MEND	MTYPE[3:0]	Reserved	ADD[6:0]		RNW

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

Bit	name	access		Reset value
31	MEND	WO	Description of message end type/last message of frame: 1: This message from the master device ends with a stop bit and is the last bit of the frame. Next message; 0: This message from the master device is followed by a repeat start bit, and then... It can send another message. Note: This position applies only to the master device.	0
[30:27] MTYPE[3:0]		WO	Message type. When the I3C is used as the master device: 0000: The SCL clock is forcibly stopped until the next control word is executed; 0001: Header message; 0010: Private message; 0011: Direct Message (Part 2 of I3C SDR Direct CCC Command) point); 0100: Traditional I2C message; Other: Reserved. When I3C is used as a slave device: 1000: Hot-join request; 1001: Master role request; 1010: IBI (In-Band Interrupt) request; Other: Reserved.	0
[26:24] Reserved		RO	is reserved.	0
[23:17] ADD[6:0]		WO	7-bit I3C dynamic/I2C static slave address. Note: when This bit When used as a master device, if MTYPE[3:0] is 0010b, 0011b, 0100b field is used when using I3C or other methods.	0
16 RNW		WO	Read/non-write messages. 1: Read messages; 0: Write a message. Note: when This bit When used as a master device, if MTYPE[3:0] is 0010b, 0011b, 0100b field is used when using I3C or other methods.	0
[15:0] DCNT[15:0]		WO	The number of data to be transferred during a read/write message, in bytes. Bit: 0x0000: No data to transmit; 0x0001: 1 byte; 0x0002: 2 bytes; ... 0xFFFF: 64KB - 1 byte. When used as a master device, if MTYPE[3:0] is 0010b, I3C may use this bit field to set the crossover on the bus. 0011b 0100b Number of data bytes swapped. This applies to private or traditional read/write operations. I2C information is used, then this bit field must be non-empty. If I3C When used as a slave device, if MTYPE[3:0] is the 1010b and only any IBI data payload (I3C_BCR register)	0

			BCR2 1 sets if the bit is the data to be sent, then use that bit field. IBI the number of bytes in the data payload (1, 2, 3 or 4).	

22.4.2 I3C Control Register [Multiplexed] (R32_I3C_CTLR) Offset Address:

0x00

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
MEND	MTYPE[3:0]	Reserved	CCC[7:0]		

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

Bit	name	access		Reset value
31	MEND	WO	Description of message end type/last message of the frame: 1: This message from the master device ends with a stop bit and is the last bit of the frame. Next message; 0: This message from the master device is followed by a repeat start bit, and then... It can send another message. Note: This position applies only to the master device.	0
[30:27]	MTYPE[3:0]	WO	Message type. When the I3C is used as the master device: 0110: Broadcast/Direct CCC command; Bits [23:16] (CCC[7:0]) are the 8-bit CCC codes issued; Bits [15:0] (DCNT[15:0]) are CCC definition bytes, CCC subbytes. The number of command bytes or CCC data bytes. If bit[23] = CCC[7] = 1: the message is I3C SDR direct The first part of the CCC command transmits the direct CCC command (part one). The message (in minutes) is: – {S/S + 0b111_1110 +RnW = 0/Sr+*} + (straight) (CCC + T) + (8-bit data + T) * + Sr; – After S (start bit), register R32_I3C_CFGR The NOARBH bit determines whether to insert an arbitrable header (0b111_1110 + RnW = 0). – Sr+*: After Sr (start of repeat), the hardware will automatically... Insert (0b111_1110 + R/W). If bit[23] = CCC[7] = 0: the message is an I3C SDR broadcast. CCC command; the transmitted broadcast CCC command message is: – {S/S + 0b111_1110 + RnW = 0/Sr + *} + (Broadcast CCC + T) + (8- bit data + T)* + Sr/P – After S (start bit), determine whether to use based on the NOARBH bit. Insert arbitrable header (0b111_1110 + RnW = 0). – Sr+*: After Sr (start of repeat), the hardware will automatically... Insert (0b111_1110 + R/W). Other values: Reserved.	0
[26:24]	Reserved	RO	is reserved.	0

[23:16] CCC[7:0]		WO	8-bit CCC code: If bit[23] = CCC[7] = 1, then the message is I3C SDR. The first part of the CCC command. If bit[23] = CCC[7] = 0, then the message is I3C SDR. Broadcast CCC commands (including ENTDAA). Note: This position applies only to the master device.	0
[15:0] DCNT[15:0]		WO	To be used as a CCC interpretation byte, CCC subcommand byte, or CCC data. Count of CCC command-related data transmitted in bytes (in bytes) unit): Linear encoding can be up to 64KB - 1. 0x0000: No data to transmit; Note: This is a value that must be used when broadcasting ENTDAA. 0x0001: 1 byte; Note: This is the value that must be used when sending RSTACT directly or broadcasting. 0x0002: 2 bytes; ... 0xFFFF: 64KB - 1 byte.	0

22.4.3 I3C Configuration Register (R32_I3C_CFGR) Offset

Address: 0x04

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16	
Reserved	TSFSET	Reserved	CFLUSH	CDMAEN	TMODE	SMODE	SFLUSH	SDMAEN

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	txHRES	TXFLUSH	TXDMAEN	Reserved	RXTHRES	RXFLUSH	RXDMAEN	HJACK	Reserved		EXITPTRN	RSTPTRN	NOARBH	CRINIT	EN

Bit	name	access	describe	Reset value
31	Reserved	RO reserved.		0
30 TSFSET		W0	Frame transmission settings: 1: Set the CFNFF bit in hardware to start frame transmission; 0: No operation is performed. Note: (1) This bit applies only to the master device. (2) If this bit is not set to 1, frame transmission can be initiated via software. exist C-FIFO is empty (R32_I3C_EVR register bit is empty) CFEF Controller step Write directly to the one word register 1) (R32_I3C_CTLR). Then, if the first control word written is not marked as the end of a message (R32_I3C_CTLR) register MEND If the bit is 0, the hardware will set CFNFF to 1. Note: This position applies only to the master device.	0
[29:22] Reserved		RO is reserved.		0
twenty one	CFLUSH	W0	C-FIFO cleared: 1: Clear C-FIFOO;	0

			0: No operation is performed. Note: This position applies only to the master device.	
20	CDMAEN	RW	C-FIFO DMA Request Enable: 1: Enable DMA mode for C-FIFO; 0: Disable DMA mode for C-FIFO. Note: This position applies only to the master device.	0
19	TMODE	RW	Frames sent on the I3C bus are used to manage the C-FIFO and TX-FIFO: 1. Before initiating frame transmission, preload the C-FIFO and... TX-FIFO. 0: Do not preload C-FIFO and before initiating frame transmission. TX-FIFO. The frame begins when the first control word appears in the C-FIFO. The transfer will begin immediately. Note: This position applies only to the master device.	0
18	SMODE	RW	S-FIFO Enable/Status Receive Mode: When the I3C is used as the master device: 1: Enable S-FIFO; 0: Disable S-FIFO. This bit must be cleared when the I3C is used as a slave device.	0
17	SFLUSH	WO	S-FIFO cleared: 1: Clear the S-FIFO; 0: No operation is performed. Note: This position applies only to the master device.	0
16	SDMAEN	RW	S-FIFO DMA request to enable: 1: Enable reading the status register via DMA mode R32_I3C_STATER; 0: Disable reading the status register via DMA mode. R32_I3C_STATER. Note: This position applies only to the master device.	0
15	Reserved	RO	is reserved.	0
14	TXTHRES	RW	TX-FIFO threshold: 1:1 word/4 bytes threshold; 0:1 byte threshold.	0
13	TXFLUSH	WO	Clear TX-FIFO: 1: Clear TX-FIFO; 0: No operation is performed.	0
12	TXDMAEN	RW	TX-FIFO DMA request enable: 1: Enable DMA mode for TX-FIFO; 0: Disable DMA mode for TX-FIFO.	0
11	Reserved	RO	is reserved.	0
10	RXTHRES	RW	RX-FIFO threshold: 1:1 word/4 bytes threshold; 0:1 byte threshold.	0
9	RXFLUSH	WO	Clear RX-FIFO: 1: Clear RX-FIFO;	0

			0: No operation is performed.	
8	RXDMAEN	RW	RX-FIFO DMA request enable: 1: Enable DMA mode for RX-FIFO; 0: Disable DMA mode for RX-FIFO.	0
7	HJACK	RW	Hot-join request confirmation: 1: The hot-addition request has been confirmed; 0: Hot-join request not confirmed. Note: This position applies only to the master device.	0
[6:5] Reserved		RO is reserved.		0
4	EXITPTRN	RW	HDR exit timing enable: 1: Send message header (R32_I3C_CTLR register) After MTYPE[3:0] bits are 0001b), HDR exit timing is sent. Used to perform actions when the slave device does not respond to private messages or directly reads CCC. Perform main device error detection and upgrade processing. 0: Send message header (R32_I3C_CTLR register) After MTYPE[3:0] bits are set to 0001b, no HDR exit timing is sent. This is used to send headers, in case of suspicion after a master device role switch. Problem exists (when the new master device is not in the working state defined delay) Within a given timeframe, the host device is enabled by accessing the previous host device. (Color), which can be used to test ownership of the bus. Note: This position applies only to the master device.	0
3	RSTPTRN	RW	HDR reset timing enable: 1: In any frame sent (containing the RSTACT CCC command) HDR reset timing is inserted before each stop bit; 0: A standard stop bit is emitted at the end of the frame. Note: This position applies only to the master device.	0
2	NOARBH	RW	No arbitrator head after the start position: 1: No Arbitration Head - For traditional I2C messages or I3C SDR private read/write messages, The slave device address is sent directly after the start bit. - This is a more efficient option (when a 0x7E arbitrable header is issued). (When not in use), but can only be determined by the master device when the addressed slave device is located. This is only possible when the device cannot simultaneously issue an IBI or master role request. Use (to ensure that for IBI or MR, the master device is in open-drain mode) The address sent by the slave device can be sent after the start bit. There will be no misresolution or potential conflicts between the same address. 0: After the start bit, in a traditional I2C message or I3C SDR private message. Before reading/writing messages (by default), send an arbitrable header. (0b111_1110 + RnW = 0). Note: This position applies only to the master device.	0
1	CRINIT	RW	Label the master/slave device role: 1: Main device role; 0: From the device role.	0
0	EN	RW	I3C Enable: 1: Enable I3C;	0

			0: I3C is prohibited.	

22.4.4 I3C Receive Data Byte Register (R32_I3C_RDR) Offset

Address: 0x10

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								RDB0[7:0]							

Bit	name	access	describe	Reset value
[31:8]	Reserved	RO	is reserved.	0
[7:0]	RDB0[7:0]		8-bit receive data on the RO I3C bus.	0

22.4.5 I3C Receive Data Word Register (R32_I3C_RDWR) Offset

Address: 0x14

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
RDB3[7:0]								RDB2[7:0]						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDB1[7:0]								RDB0[7:0]							

Bit	name	access	describe	Reset value
[31:24]	RDB3[7:0]	RO	8-bit receive data (latest byte on the I3C bus).	0
[23:16]	RDB2[7:0]	RO	8-bit received data (byte following RDB1 on the I3C bus). 0	RO
[15:8]	RDB1[7:0]		8-bit received data (byte following RDB0 on the I3C bus). 0 RO	8-bit
[7:0]	RDB0[7:0]		received data (earliest byte on the I3C bus).	0

22.4.6 I3C Transmit Data Byte Register (R32_I3C_TDR) Offset

Address: 0x18

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TDB0[7:0]							

Bit	name	access	describe	Reset value
[31:8]	Reserved	RO	is reserved.	0
[7:0]	TDB0[7:0]		8-bit data transmission on the RO I3C bus.	0

22.4.7 I3C Transmit Data Word Register (R32_I3C_TDWR) Offset

Address: 0x1C

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16

TDB3[7:0]								TDB2[7:0]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDB1[7:0]								TDB0[7:0]							

Bit	name	access	describe	Reset value
[31:24]	TDB3[7:0]	WO	8-bit data transmission (latest byte on the I3C bus).	0
[23:16]	TDB2[7:0]	WO	8-bit data transmission (byte following TDB1 on the I3C bus). 0	WO
[15:8]	TDB1[7:0]		8-bit data transmission (byte following TDB0 on the I3C bus). 0	WO 8-bit
[7:0]	TDB0[7:0]		data transmission (earliest byte on the I3C bus).	0

22.4.8 I3C IBI Payload Data Register (R32_I3C_IBIDR) Offset Address: 0x20

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
IBIDB3[7:0]											IBIDB2[7:0]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IBIDB1[7:0]								IBIDB0[7:0]							

Bit	name	access	describe	Reset value
[31:24]	IBIDB3[7:0]	RW	8-bit IBI payload data (latest byte on the I3C bus). 0	8-bit IBI payload
[23:16]	IBIDB2[7:0]	RW	data (following the latest byte on the I3C bus). (bytes after IBIDB1[7:0]).	0
[15:8]	IBIDB1[7:0]	RW	8-bit IBI payload data (followed by the I3C bus) (bytes after IBIDB0[7:0]).	0
[7:0]	IBIDB0[7:0]	RW	8-bit IBI payload data (the earliest byte on the I3C bus, MDB[7:0] Forced data bytes).	0

22.4.9 I3C Slave Device Send Configuration Register (R32_I3C_TGTTDR)

Offset Address: 0x24

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														PRELOAD
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TGTTDCNT[15:0]															

Bit	name	access	describe	Reset value
[31:17]	Reserved	RO	is reserved.	0
16	PRELOAD	RW	TX-FIFO pre-loaded: 1: Perform TX-FIFO preloading; 0: Do not perform TX-FIFO preloading. Note: This bit applies only to slave devices.	0
[15:0]	TGTTDCNT[15:0]	RW	data transmission counter, in bytes.	0

			When PRELOAD is set to 1, this bit field must be accessed by software in the same location. Write this in the question to define the number of bytes to preload and send. This bit field is updated by hardware; reading this bit field will report what needs to be loaded. The number of bytes remaining in the TX-FIFO. Note: This bit applies only to slave devices.

22.4.10 I3C Reset Register (R32_I3C_RESET) Offset Address:

0x2C

31	30	29 28 27 26 25 24	23 22 21	2019	18	17 16
Reserved			TGT_SIE_RST	HST_SIE_RST	Reserved	

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

Bit	name	access	describe	Reset value
[31:26]	Reserved	RO	reserved.	0
25	TGT_SIE_RST	RW	Reset register from device: 1: The device's protocol layer is in a reset state; 0: The device's protocol layer is in an active state;	1
24	HST_SIE_RST	RW	Master device reset register: 1: The protocol layer of the master device is in a reset state; 0: The protocol layer of the master device is in working condition;	1
[23:0]	Reserved	RO	is reserved and must be kept at 0x1000 during initialization.	0

22.4.11 I3C Status Register (R32_I3C_STATR) Offset Address:

0x30

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
MID[7:0]			Reserved	DIR ABT	Reserved

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

Bit	name	access	describe	Reset value
[31:24]	MID[7:0]	RO	Message identifier/counter for a given frame: This bit field is used for identification when the I3C is used as the master device. The control word message referenced by the R32_I3C_STATR status register. (R32_I3C_CTLR). The first message of the frame is identified by MID[7:0] = 0. When a new message control word (R32_I3C_CTLR) appears on the I3C bus. Upon completion, this bit field is incremented (by hardware). This increments each time a new frame begins.	0

			All This will reset the bit field. Note: This position applies only to the master device.	
[23:19] Reserved		RO reserved.		0
18	DIR	RO	Message direction: 1: Read; 0: Write. Note: ENTDAA is considered a write command. CCC	0
17	ABT	RO	Early termination of private read messages from the device: 1: Complete the equipment in advance; 0: The equipment was not completed in advance. Note: This position applies only to the master device.	0
16 Reserved		RO reserved.		0
[15:0] XDCNT[15:0]		RO	Data counter: Condition: During dynamic address allocation (ENTDAA CCC) - When the I3C is used as the master device: Slave devices detected on the bus number; - When I3C is used as a slave device: Number of bytes sent. Condition: For other transmissions during the message period - Regardless of whether the I3C is used as a master or slave device: during the message... The number of bytes of data read from or sent on the I3C bus.	0

22.4.12 I3C Status Error Register (R32_I3C_STATER) Offset Address: 0x34

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					DERR	DNACK	ANACK	COVR	DOVR	STALL	PERR	CODERR[3:0]			

Bit	name	access	describe	Reset value
[31:11] Reserved		RO reserved.		0
10	DERR	RO	Data error: 1: When the received slave device address and/or parity bit do not match During the timing process, the master device detected a data error during the master device role switchover (GETACCCR CCC, formerly known as GETACCMST). The main device retains its main device role. 0: No error detected. Note: This position applies only to the master device.	0
9	DNACK	RO	Data unconfirmed: 1: The master device detects unacknowledged data from the slave device during the following process. byte: i) Traditional I2C write transfer; ii) A second attempt to send a dynamic address during the ENTDAA process.	0

			0: No error detected. Note: This position applies only to the master device.	
8	ANACK	RO	Address not confirmed: 1: The master device detects that the slave device has not acknowledged static status during the following process. /Dynamic address: i) Traditional I2C read/write transfer; ii) Direct CCC write transfer; iii) Second attempt at direct CCC read transfer; iv) Private read/write transfer; 0: No error detected. Note: This position applies only to the master device.	0
7	COVR	RO	C-FIFO underflow or S-FIFO overflow: 1: The main device detected: i) C-FIFO underflow: When the control FIFO is empty, a repeating command must be issued. Start bit; ii) S-FIFO overflow: S-FIFO is full, new messages have ended; 0: No error detected. Note: This position applies only to the master device.	0
6	DOVR	RO	RX-FIFO overflow or TX-FIFO underflow: 1: Regardless of whether it is used as a master or slave device, the hardware detects: i) TX-FIFO underflow: The TX-FIFO is empty and a write operation must be performed. Data bytes; ii) RX-FIFO Overflow: The RX-FIFO is full and a new one has been received. Data bytes; 0: No error detected.	0
5	STALL	RO	SCL pause error: 1: When the device detects stable SCL during I3C SDR data read transfers (during direct CCC reads, proprietary reads, or IB transfers). The interval exceeds 125ÿs; 0: No error detected. Note: This bit applies only to slave devices.	0
4	PERR	RO	Protocol error: 1: Regardless of whether it is used as a master or slave device, the hardware detects the cooperation. The proposal is incorrect; 0: No error detected.	0
[3:0] CODERR[3:0]		RO	0000: CE0 error (transaction after CCC sent): Master device Illegal CCC formatting detected; 0001: CE1 Error (Monitoring Error): The master device detected a total... The data sent online did not match expectations; 0010: CE2 Error (No response to broadcast address): Master device An unacknowledged broadcast address (0b111_1110) was detected. 0011: CE3 Error (Master device role switching failed): Master device The system detected that the new master device did not drive the main system after the master device role switch. Wire; 1000: TE0 Error (Invalid Broadcast Address)	0

			0b111_1110+W): An invalid broadcast address was detected from the device. 0b111_1110+W; 1001: TE1 Error (CCC Code): Device fails due to parity error. The check detected a parity error in the CCC code (relative to...). T position); 1010: TE2 Error (Write Data): Data loss due to parity error from device. The test detected a parity error in the written data (relative to T). Bit); 1011: TE3 Error (Land allocated during dynamic address arbitration) Address): Parity check performed by the device during dynamic address arbitration. A parity error was detected in the allocated address (relative to...). PAR position); 1100: TE4 Error (Missing Sr bit after dynamic address arbitration) 0b111_1110+R): From the device during dynamic address arbitration. After detecting the Sr bit, 0b111_1110 + R is missing; 1101: TE5 error (transaction after CCC detection): From the set The system detected an illegal formatting of the CCC. 1110: TE6 Error (Monitoring Error): Total errors detected from the device. The data sent online did not match expectations; Other values: Reserved.

22.4.13 I3C Receive Message Register (R32_I3C_RMR) Offset Address:

0x40

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
Reserved			RADD[6:0]		Reserved

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

Bit	name	access	describe	Reset value
[31:24]	Reserved	RO	reserved.	0
[23:17]	RADD[6:0]	RO	Received slave device address: When the I3C is configured as a master device, this bit field is used to record the established... Received from the device during the recognized IBI or master device role request Dynamic URL. Note: This position applies only to the master device.	0
16	Reserved	RO	reserved.	0
[15:8]	RCODE[7:0]	RO	Received CCC code: When the I3C is configured as a slave device, this bit field is used to record received data. The CCC code. Note: This bit applies only to slave devices.	0
[7:3]	Reserved	RO	is reserved.	0
[2:0]	IBIRDCNT[2:0]	RO	IBI received payload data count:	0

			When the I3C is configured as a master device, this bit field is used to record... Valid received data bytes in the R32_I3C_IBIDR register number. Note: This position applies only to the master device.	

22.4.14 I3C Event Register (R32_I3C_EVR) Offset Address: 0x50

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
GRPF	DEFF	INTUPDF	ASUPDF	RSTF	MRLUPDF	MWLUPDF	DAUPDF	STAF	GETF	WKPF	Reserved	HJF	CRUPDF	CRF	IBIENDF

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IBIF	Reserved			ERRF	RXTGTENDF	FCF	Reserved	RXLASTF	TXLASTF	RXFNEF	TXFNFF	SFNEF	CFNFF	TXFEF	CFEF

Bit	name	access	describe	Reset value
31	GRPF	RO	Group addressing flags: When the I3C is used as a slave device (typically supporting the master role), this The flag is set to 1 by hardware to indicate that a broadcast DEFGRPA CCC (Define Group Address List) has been received. The software can then store the connection. The received data is used when obtaining the master device role. This flag is cleared by sending a message to R32_I3C_CTLR via software. Write 1 to the CGRPF register. Note: This bit applies only to slave devices.	0
30	DEFF	RO	DEFTGTS logo: When the I3C is used as a slave device (typically supporting the master role), this The flag is set to 1 by hardware to indicate that a broadcast DEFTGTS CCC (defining a list of slave devices) has been received. The software can then store the connection. The received data is used when obtaining the master device role. This flag is cleared by writing 1 to the CDEFF bit of the R32_I3C_CEVR register via software. Note: This bit applies only to slave devices.	0
29	INTUPDF	RO	Interrupt/Master Device Role/Hot Join Update Flags: When the I3C is used as a slave device, this flag is set to 1 by hardware to indicate... This indicates that ENEC/DISEC CCC (enable/disable) has been received directly or via broadcast. (Stop slave device events), where slave device events are interrupts/IBI requests. Request, master device role request, or hot-join request. Then, the software must read the R32_I3C_DEVR0 register respectively. The IBIEN, CREN, or HJEN in the text. This flag is cleared by writing 1 to the CINTUPDF bit of the R32_I3C_CEVR register via software. Note: This bit applies only to slave devices.	0
28	ASUPDF	RO	operating status update indicator:	0

			When the I3C is used as a slave device, this flag is set to 1 by hardware to indicate... This indicates that ENTASx CCC (where x = 0...3) has been received directly or via broadcast. The software must then read the R32_I3C_DEVR0 register. Register AS[1:0]. This flag is cleared by writing 1 to the CASUPDF bit of the R32_I3C_CEVR register via software. Note: This bit applies only to slave devices.	
27	RSTF	RO	Reset timing flags: When the I3C is used as a slave device, this flag is set to 1 by hardware to indicate... The reset timing has been detected (occurring 14 times while SCL is low). SDA transition, followed by repeating start and stop bits. Then, when not in stop mode, the software must read The RSTACT[1:0] and RSTVAL in the I3C_DEVR0 register, Obtain the required reset level. – If RSTVAL = 1: When RSTF is set to 1 (and/or a phase is generated) When the corresponding interrupt (if enabled) occurs, register R32_I3C_DEVR0 is accessed. The RSTACT[1:0] field in the device indicates the reset operation to be performed by the software. (If any). – If RSTVAL = 0: When RSTF is set to 1 (and/or a phase is generated) When an interrupt occurs (if enabled), the software must be in the first check. After the detected reset timing, an I3C reset is issued, and in the second detection... After the reset timing is detected, a system reset is issued. When in stop mode, a corresponding interrupt can be used to wake it up. Device. The flag is cleared by sending a message to R32_I3C_CEVR via software. Write 1 to the corresponding CRSTF bit in the register. Note: This bit applies only to slave devices.	0
26	MRLUPDF	RO	Maximum read length update flag: When the I3C is used as a slave device, this flag is set to 1 by hardware to indicate... This indicates that a direct SETMRL CCC (set maximum read length) has been received. (degree). Then, the software must read the R32_I3C_MAXRLR register. Use the MRL[15:0] bits to obtain the maximum read length value. This flag is cleared by writing 1 to the CMRLUPDF bit of the R32_I3C_CEVR register via software. Note: This bit applies only to slave devices.	0
25	MWLUPDF	RO	Maximum write length update flag: When the I3C is used as a slave device, this flag is set to 1 by hardware to indicate... It indicates that a direct SETMWL CCC (set maximum write length) has been received. (degree). Then, the software must read the R32_I3C_MAXRLR register. Use the MWL[15:0] bits to get the maximum write length value. The flag is cleared by sending a message to R32_I3C_CEVR via software. Write 1 to the CMWLUPDF register. Note: This bit applies only to slave devices.	0
24	DAUPDF	RO	Dynamic address update flag: When the I3C is used as a slave device, this flag is set to 1 by hardware to indicate...	0

			This indicates that a dynamic address update has been received via broadcast ENTDAA, RSTDAA, and direct SETNEWDA CCC. Then, The software must read the R32_I3C_DEVR0 registers DA[6:0] and DAVAL is used to obtain dynamic address updates. The flag is cleared by sending a message to R32_I3C_CEVR via software. Write 1 to the CDAUPDF register. Note: This bit applies only to slave devices.	
23	STAF	RO	Get status flags: When the I3C is used as a slave device, this flag is set to 1 by hardware to indicate... This indicates that a direct GETSTATUS CCC (without definition) of format 1 has been received. (Byte or byte with interpretation TGTSTAT). The flag is cleared by sending a message to R32_I3C_CEVR via software. Write 1 to the CSTAF bit of the register. Note: This bit applies only to slave devices.	0
22	GETF	RO	Acquire Flag: When the I3C is used as a slave device, this flag is set to 1 by hardware to indicate... It indicates that GETSTATUS has been received in addition to format 1 (but includes) Format 2 (GETSTATUS) allows direct CCC for any type of retrieval. (GET*** CCC). This flag is cleared by writing 1 to the CGETF bit of the R32_I3C_CEVR register via software. Note: This bit applies only to slave devices.	0
twenty one	WKPF	RO	Wake-up/Loss Start Flag: When the I3C is used as a slave device, this flag is set to 1 by hardware to indicate... This indicates that a start bit has been detected (falling edge of SDA followed by falling edge of SCL). But on the next falling edge of SCL, the I3C core clock (still) is Gating. Therefore, the slave device may have lost I3C bus transactions. The corresponding interrupt can be used to bring the device from low power (sleep or stop). Mode wake-up. This flag is cleared by writing 1 to the CWKPF bit of the R32_I3C_CEVR register via software. Note: This bit applies only to slave devices.	0
20	Reserved	RO is retained.		0
19	HJF	RO	Hot-addition marking: When the I3C is used as the master device, this flag is set to 1 by hardware to indicate... This indicates that the heat application request has been received. This flag is cleared by writing 1 to the CHJF bit of the R32_I3C_CEVR register via software. Note: This position applies only to the master device.	0
18	CRUPDF	RO	Main device role update flag: When the I3C is used as a slave device, this flag is set to 1 by hardware to indicate... The I3C has now acquired the master device role after completing the master device role switchover process. Equipment role. This flag is cleared by writing 1 to the CCRUPDF bit of the R32_I3C_CEVR register via software.	0

			Note: This bit applies only to slave devices.	
17	CRF	RO	Master device role request flag: When the I3C is used as the master device, this flag is set to 1 by hardware to indicate... The hardware has been confirmed and the master device role request has been completed. Then, the software... The master device role switching process must issue GETACCCR CCC. (Acquire the host device role). This flag is cleared by writing 1 to the CCRF bit of the R32_I3C_CEVR register via software. Note: This position applies only to the master device.	0
16	IBIENDF	RO	IBI End Marker: When the I3C is used as a slave device, this flag is set to 1 by hardware to indicate... The system has received and completed the IBI transmission (the master device confirms the IBI and...). Read IBI data bytes (if any). The flag is cleared by sending a message to R32_I3C_CEVR via software. Write 1 to the CIBIENDF register. Note: This bit applies only to slave devices.	0
15	IBIF	RO	IBI logo: When the I3C is used as the master device, this flag is set to 1 by hardware to indicate... This indicates that the IBI request has been received. The flag is cleared by sending a message to R32_I3C_CEVR via software. Write 1 to the CIBIF bit of the register. Note: (1) This bit applies only to the master device; (2) This bit is not effective IBI when receiving requests without data from the slave device. For details on handling such packets, please refer to the EVT official website. Procedure.	0
[14:12] Reserved		RO reserved.		0
11	ERRF	RO	Symbol: This flag is set to 1 by the hardware to indicate that an error has occurred. Then, the software... The device must read R32_I3C_STATER to obtain the error type. The flag is cleared by sending a message to R32_I3C_CEVR via software. Write 1 to the CERRF bit of the register.	0
10	RXTGTENDF	RO	End-of-read flag initiated by the device: When the I3C is used as the master device, and only when S-FIFO is disabled. When the SMODE bit of the R32_I3C_CFGR register is 0, This flag will then be set to 1 by the hardware to indicate that the slave device has terminated in advance. The data is then read from the register. R32_I3C_STATR is used to check the information related to the last message. Information, and obtain the data words received during the prematurely terminated read transfer. Number of sections (R32_I3C_STATR register XDCNT). The flag is cleared by sending a message to R32_I3C_CEVR via software. Write 1 to the CRXTGTENDF register. Note: This position applies only to the master device.	0
9	FCF	RO	Frame completion flag: When the I3C is used as the master device, this flag is set to 1 by hardware to indicate...	0

			This indicates that a frame on the I3C bus has been completed (normally), for example, when a stop signal is issued. When stopped. When the I3C is used as a slave device, this flag is set to 1 by hardware to indicate... This is shown as a message sent on the I3C bus to or retrieved by the slave device. The address message has been completed (normally), for example, when the master device sends the next... When there is a stop bit or a repeat start bit. This flag is cleared by writing 1 to the CFCF bit of the R32_I3C_CEVR register via software.	
8	Reserved	RO is reserved.		0
7	RXLASTF	RO	Last read data byte/word flag: When the I3C is used as the master device, this flag is set to 1 by hardware to indicate... This indicates the last data word of the message that must be read from the RX-FIFO. Byte/word (depending on the RXTHRES bit of the I3C_CFGR register). This flag is set when the last data byte/word of the message is read. Reset hardware. Note: This position applies only to the master device.	0
6	TXLASTF	RO	Last written data byte/word flag: This flag is set to 1 by hardware to indicate that data must be written to the TX-FIFO. The last data byte/word of the message (depending on the specific data) The R32_I3C_CFGR register TXTHRES bit). When writing a message. The flag is cleared by hardware when the last data byte/word is received.	0
5	RXFNEF	RO	RX-FIFO Not Empty Flag: This flag is set to 1 or cleared by hardware to indicate whether it must/must be obtained from... RX-FIFO reads data bytes. Note: Before RX-FIFO Read data bytes (read and retrieve) R32_I3C_RDR using R32_I3C_RDWR (RXTHRES), the , Specifically depends on software must wait for RXFNEF to poll or via an enabled = 1 (through) interrupt.	0
4	TXFNFF	RO	TX-FIFO not full indicator: This flag is set to 1 or cleared by hardware to indicate whether it must be or must not be submitted to the relevant authority. Write data bytes to TX-FIFO. Note: (1) When writing to TX-FIFO (writing to R32_I3C_TDR or R32_I3C_TDWR, specifically depending on TXTHRES) before the software Must wait for TXFNFF 1 (either by polling or by enabling). (Broken). (2) When I3C When used as a slave device, if the software needs to write to R32_I3C_TDR/R32_I3C_TDWR using the TXFNFF flag, then TX-FIFO preloading must be enabled (I3C_TGTTDR). register PRELOAD Position1).	0
3	SFNEF	RO	S-FIFO Not Empty Flag: When the I3C is used as the master device, if the S-FIFO is enabled... (If the SMODE bit of the R32_I3C_CFGR register is 1), then the label The flag is set to 1 by hardware to indicate that the status must be read from the S-FIFO. This flag, which is cleared by hardware, indicates that no reading is being made from the SFIFO. Retrieve the status word.	0

			Note: This position applies only to the master device.	
2	CFNFF	RO	C-FIFO Not Full Indicator: When the I3C is used as the master device, this flag is set to 1 by hardware to indicate... This indicates that a control word must be written to the C-FIFO. This flag is cleared by hardware. Used to indicate that no control word should be written to the C-FIFO. Note: Before writing to the C-FIFO (writing to R32_I3C_CTLR), The software must wait for the = 1 (via polling or by enabling) CFNFF interrupt. Note: This position applies only to the master device.	0
1	TXFEF	RO	TX-FIFO is empty: This flag is set to 1 by hardware to indicate that the TX-FIFO is empty; This flag is cleared by hardware to indicate that the TX-FIFO is not empty.	1
0	CFEF	RO	C-FIFO is empty: When used as a master device: This flag is set to 1 by hardware to indicate that the C-FIFO is empty; This flag is cleared by hardware to indicate that the C-FIFO is not empty. When used as a slave device: This flag is set to 1 by hardware and is used in the R32_I3C_CTLR register. No control words included (no IBI/CR/HJ requests); This flag is cleared by hardware in the R32_I3C_CTLR register. It contains a control word (IBI/CR/HJ request to be processed). Note: When I3C used as the master device, if C-FIFO and TX-FIFO preload (R32_I3C_CFGR register bits) If TMODE is 1, the software must wait for = 1 and CFEF = 1, TXFEF before it can begin transmitting a new frame.	1

22.4.15 I3C Interrupt Enable Register (R32_I3C_INTENR) Offset Address: 0x54

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GRPIE	DEFIE	INTUPDIE	ASUPDIE	RSTIE	MRLUPDIE	MWLUPDIE	DAUPDIE	STAIE	GETIE	WKPIE	Reserved	HJIE	CRUPDIE	CRIE	IBIENDIE

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IBIIE	Reserved			ERRIE	RXTGTENDIE	FCIE	Reserved			RXFNEIE	TXFNEIE	SFNEIE	CFNFIE	Reserved	

Bit	name	access		Reset value
31	GRPIE	RW	Description of DEFGRPA CCC interrupt enable: 1: Enable; 0: Prohibited.	0

			Note: This bit applies only to slave devices.	
30	DEFIE	RW	DEFTGTS CCC Interrupt Enable: 1: Enable; 0: Prohibited. Note: This bit applies only to slave devices.	0
29	INTUPDIE	RW	ENEC/DISEC CCC Interrupt Enable: 1: Enable; 0: Prohibited. Note: This bit applies only to slave devices.	0
28	ASUPDIE	RW	ENTASx CCC Interrupt Enable: 1: Enable; 0: Prohibited. Note: This bit applies only to slave devices.	0
27	RSTIE	RW	Reset timing interrupt enable: 1: Enable; 0: Prohibited. Note: This bit applies only to slave devices.	0
26	MRLUPDIE	RW	SETMRL CCC Interrupt Enable: 1: Enable; 0: Prohibited. Note: This bit applies only to slave devices.	0
25	MWLUPDIE	RW	SETMWL CCC Interrupt Enable: 1: Enable; 0: Prohibited. Note: This bit applies only to slave devices.	0
24	DAUPDIE	RW	Interrupt enable for ENTDAA/RSTDAA/SETNEWDA CCC: 1: Enable; 0: Prohibited. Note: This bit applies only to slave devices.	0
23	STAIE	RW	Format 1 GETSTATUS CCC Interrupt Enable: 1: Enable; 0: Prohibited. Note: This bit applies only to slave devices.	0
22	GETIE	RW	GETxxx CCC interrupt enable (except for GETSTATUS in format 1) outside): 1: Enable; 0: Prohibited. Note: This bit applies only to slave devices.	0
twenty one	WKPIE	RW	Wake-up interrupt enable: 1: Enable; 0: Prohibited. Note: This bit applies only to slave devices.	0
20	Reserved	RO	retained.	0
19	HJIE	RW	Thermal interruption enable: 1: Enable;	0

			0: Prohibited. Note: This position applies only to the master device.	
18	CRUPDIE	RW	Master device role update interrupt enable: 1: Enable; 0: Prohibited. Note: This bit applies only to slave devices.	0
17	CRIE	RW	Master device role requests interrupt enable: 1: Enable; 0: Prohibited. Note: This position applies only to the master device.	0
16	IBIENDIE	RW	IBI interrupt enable termination: 1: Enable; 0: Prohibited. Note: This bit applies only to slave devices.	0
15	IBIIE	RW	IBI Request Interrupt Enable: 1: Enable; 0: Prohibited. Note: This position applies only to the master device.	0
[14:12]	Reserved	RO reserved.		0
11	ERRIE	RW	Error interrupt enabled: 1: Enable; 0: Prohibited.	0
10	RXTGTENDIE	RW	Enable read completion interrupt initiated by the device: 1: Enable; 0: Prohibited. Note: This position applies only to the master device.	0
9	FCIE	RW	Frame completion interrupt enabled: 1: Enable; 0: Prohibited.	0
[8:6]	Reserved	RO is reserved.		0
5	RXFNEIE	RW	RX-FIFO Non-Null Interrupt Enable: 1: Enable; 0: Prohibited.	0
4	RXFNEIE	RW	TX-FIFO not full interrupt enable: 1: Enable; 0: Prohibited.	0
3	SFNEIE	RW	S-FIFO Non-Null Interrupt Enable: 1: Enable; 0: Prohibited. Note: This position applies only to the master device.	0
2	CFNFIE	RW	C-FIFO not full interrupt enable: 1: Enable; 0: Prohibited. Note: This position applies only to the master device.	0
[1:0]	Reserved	RO is reserved.		0

22.4.16 I3C Clear Event Register (R32_I3C_CEVR)

Offset Address: 0x58

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
CGRPF	CDEFF	CINTUPDF		CASUPDF	CRSTF	CMRLUPDF	CMWLUPDF	CDAUPDF	CSTAF	CGETF	CWKPF	Reserved	CHJF	CCRUPDF	CCRF	CIBIENDF

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

Bit	name	access		Reset value
31	CGRPF	WO	Description of clearing DEFGRPA CCC flag: 1: Clear GRPF; 0: No effect. Note: This bit applies only to slave devices.	0
30	CDEFF	WO	Clear DEFTGTS CCC mark: 1: Clear the DEFF value; 0: No effect. Note: This bit applies only to slave devices.	0
29	CINTUPDF	WO	Clear ENEC/DISEC CCC mark: 1: Clear INTUPDF; 0: No effect. Note: This bit applies only to slave devices.	0
28	CASUPDF	WO	Clear ENTASx CCC indicator: 1: Clear ASUPDF; 0: No effect. Note: This bit applies only to slave devices.	0
27	CRSTF	WO	Clear reset timing flags: 1: Clear RSTF; 0: No effect. Note: This bit applies only to slave devices.	0
26	CMRLUPDF	WO	Clear SETMRL CCC flag: 1: Clear MRLUPDF; 0: No effect. Note: This bit applies only to slave devices.	0
25	CMWLUPDF	WO	Clear SETMWL CCC indicator: 1: Clear MWLUPDF; 0: No effect. Note: This bit applies only to slave devices.	0
24	CDAUPDF	WO	Clear ENTDAA/RSTDAA/SETNEWDA CCC indicator: 1: Clear DAUPDF;	0

			0: No effect. Note: This bit applies only to slave devices.	
23	CSTAF	WO	Clear format 1 GETSTATUS CCC flag: 1: Reset STAF; 0: No effect. Note: This bit applies only to slave devices.	0
22	CGETF	WO	Clear the GETxxx CCC flag (in Format 1, excluding GETSTATUS). outside): 1: Clear GETF to zero; 0: No effect. Note: This bit applies only to slave devices.	0
twenty one	CWKPF	WO	Reset wake-up flag: 1: Clear WKPF; 0: No effect. Note: This bit applies only to slave devices.	0
20	Reserved	RO is retained.		0
19	CHJF	WO	Zeroing hot-addition indicator: 1: Reset HJF to zero; 0: No effect. Note: This position applies only to the master device.	0
18	CCRUPDF	WO	Clear the main device role update flag: 1: Clear CRUPDF; 0: No effect. Note: This bit applies only to slave devices.	0
17	CCRF	WO	Clear the master device role request flag: 1: Clear the CRF; 0: No effect. Note: This position applies only to the master device.	0
16	CIBIENDF	WO	Clear IBI End Marker: 1: Clear IBIENDF; 0: No effect. Note: This bit applies only to slave devices.	0
15	CIBIF	WO	Clear IBI request flags: 1: Clear IBIF; 0: No effect. Note: This position applies only to the master device.	0
[14:12]	Reserved	RO reserved.		0
11	CERRF	WO	Zeroing error flag: 1: Clear ERRF; 0: No effect.	0
10	CRXTGTENDF	WO	Clear the read end flag initiated by the device: 1: Clear RXTGTENDF; 0: No effect. Note: This position applies only to the master device.	0
9	CFCF	WO Reset Frame Completed:		0

			1: Reset FCF to zero; 0: No effect.	
[8:0] Reserved		RO	is reserved.	0

22.4.17 I3C Device Register 0 (R32_I3C_DEVR0) Offset Address: 0x60

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
Reserved			RSTVAL	RSTACT[1:0]	AS[1:0]	HJEN	Reserved	CREN	IBIEN

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

Bit	name	access	describe	Reset value
[31:25] Reserved		RO	is reserved.	0
24	RSTVAL	RO	Reset operation is valid: This bit is set to 1 by hardware to indicate that the RSTACT[1:0] bit fields are already in use. Receive broadcast or directly write RSTACT CCC (from device reset operation) Update and take effect when (does). This bit is cleared by hardware when the device receives the start of a frame. When the device is not in stop mode: – If RSTVAL = 1: When the RSTF register in R32_I3C_EVR is set. When position 1 (and/or when generating the corresponding interrupt (if enabled)), The R32_I3C_DEVR0 register RSTACT[1:0] indicates that the software needs to execute... Reset operation for the row (if any). – If RSTVAL = 0: When RSTF is set to 1 (and/or a phase is generated) When an interrupt occurs (if enabled), the software must be in the first check. After the detected reset timing, an I3C reset is issued, and in the second detection... After the reset timing is detected, a system reset is issued. When in stop mode, a corresponding interrupt can be used to wake it up. Device. Note: This bit applies only to slave devices.	0
[23:22] RSTACT[1:0]		RO	Reset operation/level upon receiving reset timing: 00: No reset operation performed. 01: First-level reset: Application software must: a) By clearing the enable bit of the I3C configuration register (writing to...) EN = 0) performs a partial reset of the peripherals. This will reset the I3C main unit. It is part of the wire interface and I3C core, but does not modify the contents of the I3C HB register (except for the EN bit). b) Register the RCC (Reset and Clock Master) via a write operation The device's I3C reset control position 1 is used for peripherals (including all their registers). The register is completely reset. 10: Second-level reset: The application software must issue a warm reset, also known as...	0

			This is a system reset. This is similar to the function of a pin reset (NRST = 0). same: 11: No reset operation is performed. Note: This bit applies only to slave devices.	
[21:20]	AS[1:0]	RO	Work status: 00: Working status 0; 01: Working status 1; 10: Working status 2; 11: Working status 3. Note: This bit applies only to slave devices.	0
19	HJEN	RW	Hot-join request enabled: 1: Enable; 0: Prohibited. Note: This bit applies only to slave devices.	0
18	Reserved	RO reserved.		0
17	CREN	RW	Master role request to be enabled: 1: Enable; 0: Prohibited. Note: This bit applies only to slave devices.	0
16	IBIEN	RW	IBI Request Enablement: 1: Enable; 0: Prohibited. Note: This bit applies only to slave devices.	0
[15:8]	Reserved	RO is reserved.		0
[7:1]	DA[6:0]	RW	7-bit dynamic address: When the I3C is used as the master device, this bit field can be written by software. It defines its own dynamic address. When the I3C is used as a slave device, this bit field is determined by the hardware upon receiving a wide signal. Update when playing ENTDAA CCC or directly SETNEWDA CCC.	0
0	DAVAL	RW	Dynamic address valid: When the I3C is used as the master device, this bit can be written by software for use with... Verify its own dynamic address, for example, during master device role switching. Before. When the I3C is used as a slave device, this bit is determined by hardware in the acknowledgment broadcast. When setting ENTDAA CCC or directly SETNEWDA CCC to 1, ensure... Reset to zero when the broadcast RSTDAA CCC is recognized.	0

22.4.18 I3C Device Register x (R32_I3C_DEVRx) (x=1/2/3/4)

Offset address: 0x64, 0x68, 0x6C, 0x70

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
DIS		Reserved		SUSP	IBIDEN	CRACK	IBIACK

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

		erv ed

Bit	name	access	describe	Reset value
31	DIS	RO	DA[6:0] write forbidden: 1: Disable/lock write operations to DA[7:0] and IBIDEN; 0: Allows access to R32_I3C_DEVRx registers DA[7:0] and Write operations for IBIDEN. Note: This position applies only to the master device.	0
[30:20] Reserved		RO reserved.		0
19 SUSP		RW	Suspend/stop I3C transmission upon receiving IBI: 1: Upon receiving an IBI request from slave device x, stop I3C transmission and clear C-FIFO and TX-FIFO; 0: After confirming and completing the IBI request from slave device x, the status remains unclear. Empty C-FIFO and TX-FIFO, and determine whether a next one exists. Control word, sends out repeat start bit or stop bit. Note: This position applies only to the master device.	0
18	IBIDEN	RW	IBI Data Enablement: 1: Acknowledged IBI from slave device x followed by forced data bytes MDB[7:0]; 0: No data bytes following the acknowledged IBI from slave device x. Note: This position applies only to the master device.	0
17 CRACK		RW	Master device role request confirmation: 1: The master device role request from slave device x must be acknowledged (via ACK). (over 7-bit dynamic address DA[6:0]) The -DIS bit field is set to 1 by hardware to protect DA[6:0] from software interference. Modifications can be made, and the hardware can internally store the current DA[6:0]. To the kernel clock domain. After -ACK, the message continues to be transmitted as initially programmed. The device requests the master role flag (R32_I3C_EVR register). CRF position 1) and/or the corresponding interrupt (if enabled) obtains access. Know; in order to effectively grant the primary device role to the requesting secondary primary device. The device software must issue a GETACCCR (formerly known as...) GETACCMST), and then send the stop bit. -Regardless of the CRACK configuration of this device or other devices, its His primary device role requests will all be NACKed until the I3C_EVR is sent. Master Role Request Flag (CRF) and IBI Flag in the register All (IBIF) values are reset to zero. 0: A NACK request from the master device role of slave device x is required. Afterwards, the message continues to be transmitted according to the initial programming (from the device). NACK received; another master role request can be sent later. beg). Note: This position applies only to the master device.	0
16	IBIACK	RW	IBI requests confirmation: 1: The IBI request from slave device x must be acknowledged (via 7 bits).	0

			Dynamic address DA[6:0]) The -DIS bit field is set to 1 by hardware to protect DA[6:0] from software interference. Modifications can be made, and the hardware can internally store the current DA[6:0]. To the kernel clock domain. After -ACK, the master device records the IBI according to I3C_DEVRx.IBIDEN. Payload data (if any). - The software communicates via the IBI flag (IBIF=1) and/or the corresponding interrupt. (If enabled) Receive notification; - Regardless of the IBIACK configuration of this device or other devices. All other IBI requests will be NACKed until the IBI request flag is cleared. Both the IBIF and the Master Role Request Flag (CRF) are cleared. 0: A NACK must be sent after an IBI request from slave device x. The message continues to be transmitted as initially programmed (from the device already received). (Upon receiving the NACK, another IBI request can be sent later.) Note: This position applies only to the master device.	
[15:8] Reserved		RO	reserved.	0
[7:1] DA[6:0]		RW	I3C dynamic address allocated to slave device x: When the I3C is used as the master device, this bit field must be written by software. ENTDAA is used to store broadcasts acknowledged by the master device via slave device x. Or directly send the 7-bit dynamic address via SETNEWDA CCC. Note: This position applies only to the master device.	0
0	Reserved	RO	is reserved.	0

22.4.19 I3C Maximum Read Length Register (R32_I3C_MAXRLR) Offset Address:

0x90

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved												IBIP[2:0]		

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

Bit	name	access	describe	Reset value
[31:19] Reserved		RO	is reserved.	0
[18:16] IBIP[2:0]		RW	Maximum size of IBI payload data, in bytes: 000: No-load data size (only when R32_IC3_BCR register is active) (Allowed when BCR2 bit is 0) 001: 1 byte (forced data byte MDB[7:0]); 010: 2 bytes (including the first MDB[7:0]); 011: 3 bytes (including the first MDB[7:0]); Other: 4 bytes (including the first MDB[7:0]). Note: This bit applies only to slave devices.	0
[15:0] MRL[15:0]		RW	Maximum data read length: Send to the software via MRLUPF and the corresponding interrupt (if enabled). Notify MRL of an update.	0

			When the device receives GETMRL CCC, the hardware uses this bit. The field returns a value on the I3C bus. Note: This bit applies only to slave devices.	

22.4.20 I3C Maximum Write Length Register (R32_I3C_MAXWLR) Offset Address:

0x94

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

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

Bit	name	access	describe	Reset value
[31:16]	Reserved	RO	reserved.	0
[15:0]	MWL[15:0]	RW	Maximum data write length: This bit field is initially written by the software when EN = 0, and upon receiving... The SETMWL command is used for hardware updates. Through the R32_I3C_EVR register MWLUPF and the corresponding interrupt (If enabled) Notify the software of MWL updates. When a GETMWL CCC is received from the device, the hardware uses this bit. The field returns a value on the I3C bus. Note: This bit applies only to slave devices.	0

22.4.21 I3C Timing Register 0 (R32_I3C_TIMINGR0)

Offset address: 0xA0

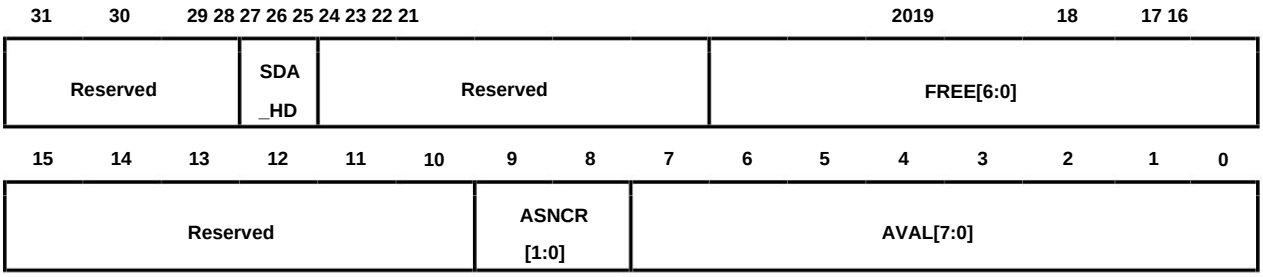
31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
SCLH_I2C[7:0]											SCLL_OD[7:0]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCLH_I3C[7:0]								SCLL_PP[7:0]							

Bit	name	access	describe	Reset value
[31:24]	SCLH_I2C[7:0]	RW	SCL high level duration, used for traditional I2C messages, within... The number of core clock cycles is in units: tSCLH_I2C = (SCLH_I2C + 1) x tHCLK , the duration	0
[23:16]	SCLL_OD[7:0]	RW	of the low level of SCL during the open-drain phase, used for traditional I2C messages and the I3C open-drain phase (address phase after the start bit, main...). The ACK phase during a device-initiated message and direct/private /IBI (T-bit phase during the payload), in kernel clock cycles. The number of periods is the unit: tSCLL_OD = (SCLL_OD + 1) x tHCLK	
[15:8]	SCLH_I3C[7:0]	RW	SCL high level duration, used for I3C messages (in push-pull and open). (Drain phase), measured in kernel clock cycles: tSCLH_I3C = (SCLH_I3C + 1) x tHCLK	

[7:0] SCLL_PP[7:0]	RW	The duration of the SCL low level during the I3C push-pull phase, in relation to the core clock... The number of cycles is the unit: tSCLL_PP = (SCLL_PP + 1) x tHCLK	

22.4.22 I3C Timing Register 1 (R32_I3C_TIMINGR1) Offset Address: 0xA4



Bit	name	access	describe	Reset value
[31:29]	Reserved	RO	is reserved.	0
28	SDA_HD	RW	SDA hold time, in kernel clock cycles. SDA hold time = (SDA_HD + 0.5) x tHCLK	0
[27:23]	Reserved	RO	is	0
[22:16]	FREE[6:0]	RW	reserved. Used to set certain MIPI timings (e.g., bus idle condition time). The number of kernel clock cycles. Note: This position applies only to the master device.	0
[15:10]	Reserved	RO	is retained.	0
[9:8]	ASNCR[1:0]	RW	Operating status of the new main equipment: The waiting time required before accessing the device as a new slave device. Please see AVAL[7:0]. Note: This position applies only to the master device.	0
[7:0]	AVAL[7:0]	RW	Used to set the number of kernel clock cycles in 1ÿs time units: The hardware then uses this time unit to build some internal timers. Corresponding to the following MIPI I3C timings: When I3C is used as a slave device: 1. For bus availability condition timing: after the stop bit and after the output. Before an IBI startup request or a master role request can be made, you must wait... A period of time (bus availability condition) (bus idle condition at least) (Continuous tAVAL). Please refer to the MIPI timing documentation for tAVAL = 1ÿs. The timing is defined as: tAVAL = (AVAL[7:0] + 2) x tHCLK 2. For bus idle condition time: when both SDA and SCL are high. After the level stabilizes and before a heat-injection event is issued, it is necessary to... A certain period of time (bus idle condition) must be waited for. Please refer to the MIPI v1.1 timing tIDLE = 200ÿs. This timing is defined as: tIDLE = (AVAL[7:0] + 2) x 200 x tHCLK When the I3C is used as a master device, its pause time must not exceed [a certain limit]. After the maximum pause time (pause the SCL clock to low level). Specifically as follows:	0

			1. For the first bit of an address allocated during dynamic address allocation: The pause time must not exceed the MIPI timing tSTALLDAA = 15ms. This timing is defined as: tSTALLDAAmax = (AVAL[7:0] + 1) x 15000 x tHCLK 2. For the ACK/NACK phase of I3C/I2C transmission, and writing data... Parity bits during transmission, I3C read transmission conversion bits: pause The clock time must not exceed MIPI timing tSTALL = 100ÿs. The timing sequence is defined as: tSTALLmax = (AVAL[7:0] + 1) x 100 x tHCLK Whether the I3C is used as a master device or (supports the master device role) During the switch from primary to secondary device role: 1. The new master device must wait for tNEWCRLock before it can complete. After GETACCR CCC, pull SDA low (send start bit). Then... Subsequently, the new master device can pull SCL low within tCAS to activate. SCL clock. This is used to test whether the new master device has been pulled low via SDA. Before gaining bus control, the active master must wait for the same The tNEWCRLock time, or at least 100 ÿs. The waiting time is taken as... It depends on the value of ANSCR[1:0] in the I3C_TIMINGR1 register: – ASNCR[1:0] = 00: tNEWCRLock = (AVAL[7:0] + 1) x tHCLK – ASNCR[1:0] = 01: tNEWCRLock = (AVAL[7:0] + 1) x 100 x tHCLK – ASNCR[1:0] = 10: tNEWCRLock = (AVAL[7:0] + 1) x 2000 x tHCLK – ASNCR[1:0] = 11: tNEWCRLock = (AVAL[7:0] + 1) x 50000 x tHCLK	

22.4.23 I3C Timing Register 2 (R32_I3C_TIMINGR2) Offset Address: 0xA8

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
STALL[7:0]								Reserved				STALLA	STALLC	STALLD	STALLT

Bit	name	access	describe	Reset value
[31:16]	Reserved	RO	Reserved.	0
[15:8]	STALL[7:0]	RW	Master device clock pause time, in kernel clock cycles: tSCLL_STALL = STALL x tHCLK	0
[7:4]	Reserved	RO	is reserved.	0
3	STALLA	RW	Master clock pause enable during the ACK phase: 1: Enable pause; 0: No pauses allowed.	0

2	STALLC	RW	Master clock pause enable during CCC PAR phase: 1: Enable pause; 0: No pauses allowed.	0
1	STALLD	RW	Master clock pause enable during the data PAR phase: 1: Enable pause; 0: No pauses allowed.	0
0	STALLT	RW	Data T-bit stage (and traditional I2C message reading data bytes) Master clock pause enable (ACK/NACK phase): 1: Enable pause; 0: No pauses allowed.	0

22.4.24 I3C Bus Feature Register (R32_I3C_BCR) Offset Address:

0xC0

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
Reserved					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									BCR6	Reserved			BCR2	Reserved	BCR0

Bit	name	access	describe	Reset value
[31:7] Reserved		RO is reserved.		0
6	BCR6	RW	Master device role supported: 1. Supports I3C master device role; 0: I3C Slave device (Master role not supported).	0
[5:3] Reserved		RO reserved.		0
2	BCR2	RW	In-band interrupt (IBI) payload: 1: The accepted IBI must be followed by at least one mandatory data byte (maximum). 4 data bytes); 0: No data bytes were received after the IBI.	0
1	Reserved	RO reserved.		0
0	BCR0	RW	Maximum data rate limit: 1: Restrictions; 0: No restrictions.	0

22.4.25 I3C Device Feature Register (R32_I3C_DCR) Offset Address:

0xC4

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
Reserved					

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

Bit	name	access	describe	Reset value
[31:8] Reserved		RO reserved.		0
[7:0] DCR	[7:0]	RW	Device characteristic ID: 0x00: General purpose device (applicable to v1.0 devices); Other values: IDs describing the I3C sensor/device type.	0

22.4.26 I3C Get Function Register (R32_I3C_GETCAPR)

Offset address: 0xC8

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

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

Bit	name	access	describe	Reset value
[31:15] Reserved		RO is reserved.		0
14	CAPPEND	RW	IBI MDB supports suspended read notifications: 1. When used as a slave device, I3C will send a read signal with an indication of pending read. Please provide the IBI of the known forced data byte value (MDB[7:5]=101). beg; 0: When used as a slave device, I3C will send a read request without indicating a pending read. The notification requires an IBI request for a mandatory data byte value.	0
[13:0] Reserved		RO is reserved.		0

22.4.27 I3C Master Role Function Register (R32_I3C_CRCAPR) Offset Address: 0xCC

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

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

Bit	name	access	describe	Reset value
[31:10] Reserved		RO is reserved.		0
9	CAPGRP	RW	Group management support: 1. Supports group address function (after becoming a master device); 0: Group address function is not supported. Note: This position applies only to the master device.	0
[8:4] Reserved		RO is reserved.		0
3	CAPDHOFF	RW	Delayed Master Role Switching:	0

			1. Additional time is needed to process the master device role switch; 0: No extra time is needed to handle the master device role switching.	
[2:0] Reserved		RO	is reserved.	0

22.4.28 I3C Get Maximum Data Rate Register (R32_I3C_GETMDSR) Offset Address:

0xD0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											TSCO	RDTURN[7:0]			

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

Bit	name	access	describe	Reset value
[31:25] Reserved		RO	reserved.	0
24 TSCO		RW	Clock to data transition time (tSCO): 1: tSCO > 12ns; 0: tSCO ÿ 12ns.	0
[23:16] RDTURN[7:0]		RW	Write the value of the selected byte to 3 bytes of MaxRdTurn (via The FMT[1:0] bit field returns this value, allowing you to toggle the maximum read speed. Encode the time to respond to GETMXDS CCC.	0
[15:10] Reserved		RO	is reserved.	0
[9:8] FMT[1:0]		RW	GETMXDS CCC format: 00: Format 1 (2 bytes: MaxWr (uninterpreted byte), MaxRd); 01: Format 2: (5 bytes: MaxWr (uninterpreted bytes) MaxRd, MaxRdTurn); - Returns a 3-byte MaxRdTurn, where MSB=0, and the middle byte. =0, LSB=RDTURN[7:0]; - Maximum read-through time is less than 256ÿs. 10: Format 2 (5 bytes: the middle byte of MaxWr (uninterpreted byte), MaxRd, and MaxRdTurn); - Returns a 3-byte MaxRdTurn, where MSB=0, and the middle byte. =RDTURN[7:0], LSB=0; - Maximum read turnaround time is between 256 and 65535 ÿs. 11: Format 2 (5 bytes: MaxWr (uninterpreted byte), MaxRd, (MSB of MaxRdTurn); - Returns a 3-byte MaxRdTurn, where MSB = RDTURN[7:0]. Intermediate byte = 0, LSB = 0; - Maximum read turnaround time is between 65535ÿs and 16s.	0
[7:2] Reserved		RO	reserved.	0
[1:0] HOFFAS[1:0]		RW	Main device switching operating status: 00: Working status 0 indicates that the I3C is in a state before and after becoming a master device. Initial operating status of the main device; 01: Working State 1 is the initial working state when the I3C becomes the master device.	0

			state; 10: Working State 2 is the initial working state when the I3C becomes the master device. state; 11: Working state 3 is the initial working state when the I3C becomes the master device. state.	

22.4.29 I3C Extended Configuration ID Register (R32_I3C_EPIDR)

Offset address: 0xD4

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
MIPIMID[14:0]					IDT SEL

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

Bit	name	Access	description RO 15-bit MIPI	Reset value
[31:17]	MIPIMID[14:0]	Manufacturer ID. Configure ID type		0x0104
16	IDTSEL	RO	selector: This bit field is set to 0. This bit field represents the 48-bit configuration ID [32]. Note: The configured bits [31:16] can be...	0
[15:12]	MIPIID[3:0]	RW	4-digit MIPI instance ID: This bit field is written by software for use on a single I3C bus. A specific number is set up and identifies each instance of this I3C IP. This bit field represents bits [15:12] of the 48-bit configuration ID. Note: The configured bits [11:0] can be...	0
[11:0]	Reserved	RO is reserved.		0

Chapter 23 Serial Peripheral Interface (SPI/I2S)

The chip has four built-in serial peripheral SPI interfaces (SPI1/2/3/4), and the SPI supports data exchange in a three-wire synchronous serial mode.

In addition, the chip select line supports hardware switching of master/slave mode and supports communication with a single data line.

I2S is also a three-wire synchronous serial interface communication protocol that supports four audio standards, including the Philips I2S standard and MSB alignment.

Standards, LSB alignment standards, and PCM standards.

23.1 Main Features

23.1.1 SPI Features ÿ Supports

full-duplex synchronous serial mode

Supports single - line half-duplex mode

Supports master and slave modes, and multiple slave modes.

Supports 8-bit or 16-bit data structures

The maximum clock frequency supported is up to half of FHCLK.

ÿ Data order supports MSB or LSB first

Supports hardware or software control of the NSS pin .

Hardware CRC check supported for both transmission and reception.

ÿ The transmit/receive buffer supports DMA transfer.

Supports modification of clock phase and polarity .

23.1.2 I2S Features ÿ Supports

simplex communication

Supports both master and slave modes .

Supports 16-bit, 24-bit, and 32-bit data formats .

• Supports audio sampling frequency range of 8kHz-562.2kHz • Supports

programmable clock polarity

Supports common I2S protocols: Philips standard, MSB aligned standard, LSB aligned standard, and PCM standard.

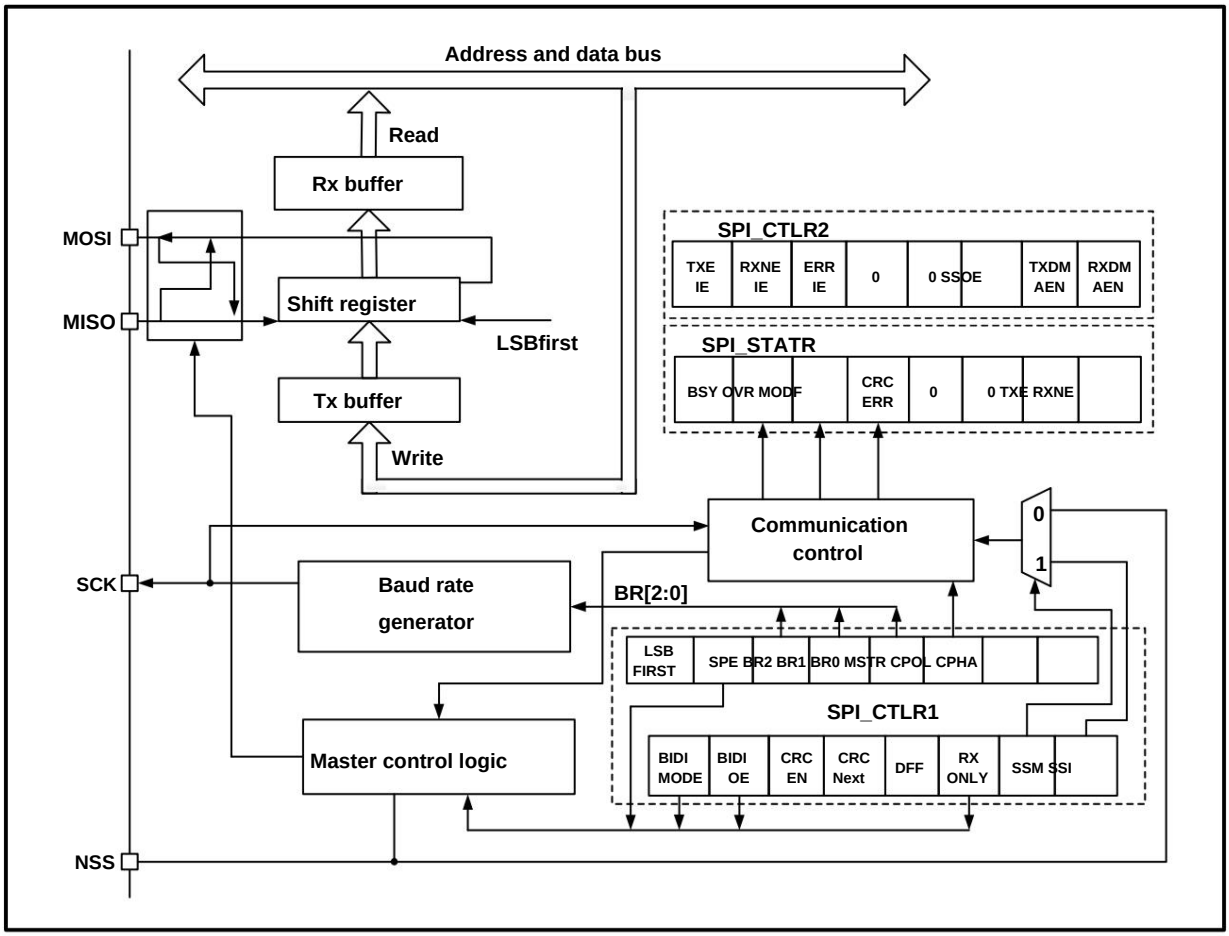
ÿ The transmit/receive buffer supports DMA transfer.

Supports outputting the master clock to external audio devices .

23.2 SPI Function Description

23.2.1 Overview

Figure 23-1 SPI Block Diagram



As shown in Figure 23-1, the main pins related to SPI are MISO, MOSI, SCK, and NSS. The MISO pin is located at...

When the SPI module is operating in master mode, it is a data input pin; when operating in slave mode, it is a data output pin. The MOSI pin is active.

In master mode, it is a data output pin; in slave mode, it is a data input pin. SCK is the clock pin; the clock signal is...

The clock signal is directly output from the master, and the slave receives the clock signal and synchronizes data transmission and reception. The NSS pin is the chip select pin, and has the following uses:

1) NSS is software controlled: In this case, SSM is set, and the internal NSS signal is determined by SSI to output high or low. This situation is generally used for

SPI master mode;

2) NSS is hardware controlled: When NSS output is enabled, i.e., when SSOE is set, it will be actively pulled low when the SPI master sends an output.

If the NSS pin cannot be successfully pulled low, it indicates that other master devices are communicating on the main line, which will generate a hardware error.

Error; if SSOE is not set, it can be used in multi-master mode. If it is pulled low, it will force entry into slave mode, and the MSTR bit will...

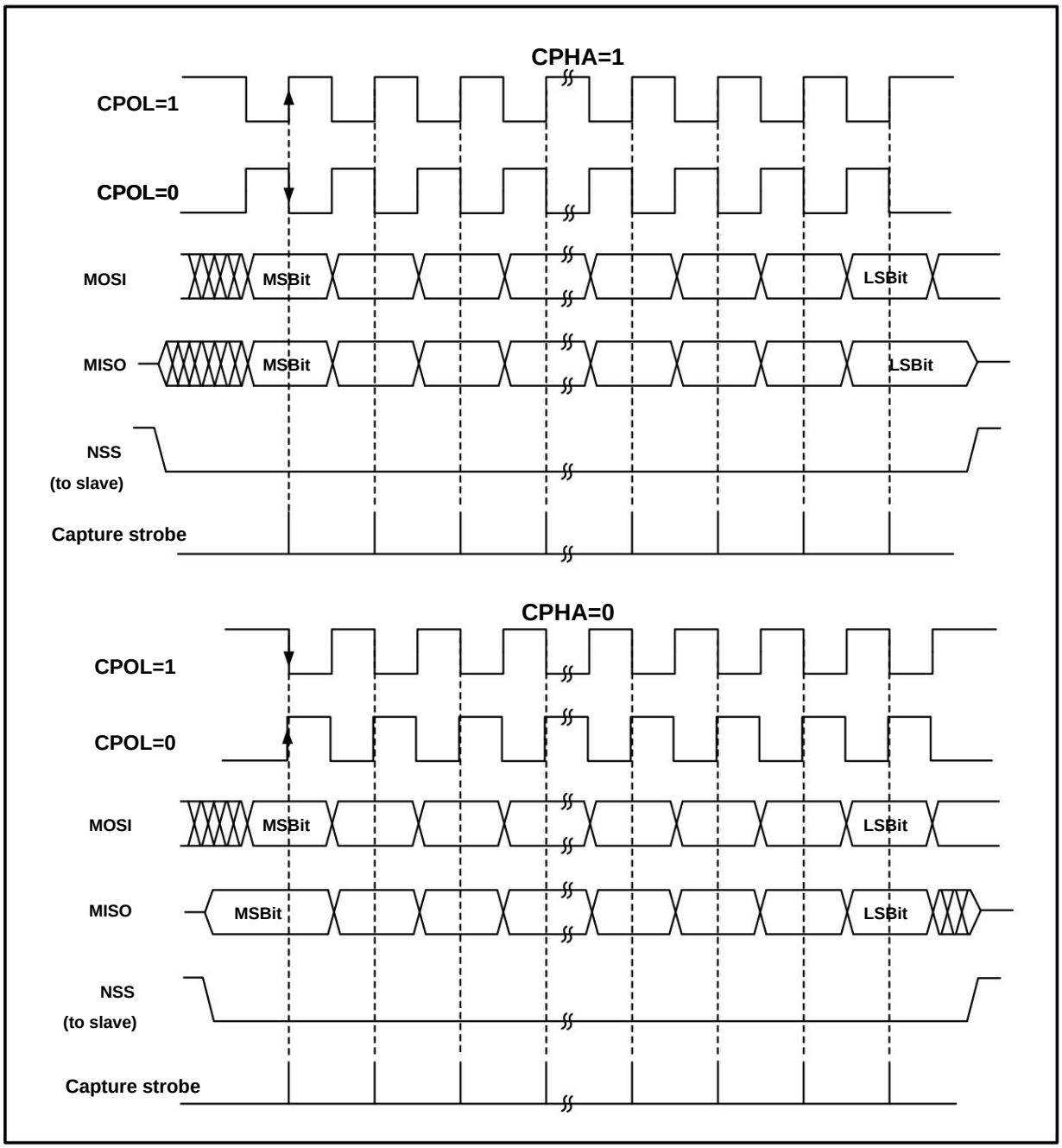
It was automatically deleted.

The SPI operating mode can be configured via CPHA and CPOL. Setting CPHA indicates that the module processes data on the second edge of the clock.

Sampling occurs when data is latched. CPHA not being set indicates that the SPI module samples on the first edge of the clock, and the data is latched. CPOL, on the other hand...

This indicates whether the clock remains high or low when there is no data. See Figure 23-2 below for details.

Figure 23-2 SPI Mode



The host and device need to be set to the same SPI mode. Before configuring the SPI mode, the SPE bit needs to be cleared. The DEF bit determines...
The length of a single data word in SP is either 8 bits or 16 bits. LSBFIRST controls whether a single data word is big-endian or little-endian.

23.2.2 Master Mode When

the SPI module is operating in master mode, the serial clock is generated by SCK. To configure master mode, follow these

steps: Configure the BR[2:0] fields 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 to allow

hardware to set NSS. Alternatively, set the SSM bit and set the SSI bit high;

Set the MSTR and SPE bits, ensuring that NSS is already high at this point. When data needs to be sent, simply write the data to be sent to the data register. The SPI will ser

The data is sent to the shift register, and then sent out from the shift register according to the LSBFIRST setting. When the data has arrived in the shift register, the TXE flag is set. If TXEIE has already been set, an interrupt will be generated. If the TXE flag is set, data needs to be filled into the data register to maintain a complete data stream. When the receiver receives data, when the last sampling clock edge

of the data word arrives, the data is transferred in parallel from the shift register to the receive buffer, and the RXNE bit is set. If the RXNEIE bit was set previously, an interrupt will also be generated. At this time, the data should be read from the data register as soon as possible.

23.2.3 Slave Mode When the

SPI module operates in slave mode, SCK is used to receive the clock signal from the master, and its own baud rate setting is invalid. The steps to configure it to slave mode are as follows:

Configure the DEF bit to set the data bit length;

Configure the CPOL and CPHA bits to match the master

mode; Configure LSBFIRST to match the master data frame

format; In hardware management mode, the NSS pin needs to be kept low. If NSS is set to software management (SSM set), then keep SSI unset; Clear the MSTR bit, set the SPE bit, and enable SPI

mode. During transmission, the slave starts transmitting when the first

slave receive sampling edge appears on SCK. The transmission process involves shifting the data in the transmit buffer into the transmit shift register. After the data in the transmit buffer is shifted into the shift register, the TXE flag is set. If the TXEIE bit was set previously, an interrupt will be generated.

During reception, after the last clock sampling edge, the RXNE bit is set, the byte received by the shift register is transferred to the receive buffer, and the read operation of the read data register can obtain the data in the receive buffer. If RXNEIE is already set before RXNE is set, an interrupt will be generated.

23.2.4 Simplex Mode The SPI

interface can operate in half-duplex mode, meaning the master device uses the MOSI pin and the slave device uses the MISO pin for communication. When using half-duplex communication, BIDIMODE needs to be set, and BIDIOE is used to control the transmission direction.

Setting the RXONLY bit in normal full-duplex mode will set the SPI module to receive-only simplex mode. Setting RXONLY will release a data pin; the released pin is different in master and slave modes. Alternatively, you can ignore received data and set the SPI to transmit-only mode.

23.2.5 CRC

The SPI module uses CRC checksum to ensure the reliability of full-duplex communication, with separate CRC calculators for data transmission and reception. The polynomial used for CRC calculation is determined by the polynomial register, and different calculation methods are used for 8-bit and 16-bit data widths. Setting the CRCEN

bit enables CRC checksum and resets the CRC calculator. After sending the last data byte, the CRCEN bit is set...

The CRCNEXT bit is sent after the current byte is transmitted, along with the result of the TXCRCR calculator. Simultaneously, if the value of the last received shift register does not match the locally calculated TXCRCR value, the CRCERR bit is set. To use CRC checksum, the polynomial calculator must be set and the CRCEN bit set when configuring the SPI operating mode, and the CRCNEXT bit must be set and sent in the last word or half-word.

Perform CRC calculation and receive CRC verification. Note that the CRC calculation polynomials for both the sender and receiver should be consistent.

23.2.6 DMA

The SPI module supports DMA to accelerate data communication. DMA can be used to fill the transmit buffer with data or to retrieve data from the receive buffer in a timely manner. DMA will use RXNE and TXE signals to retrieve or send data promptly. DMA can also operate in simplex or CRC-checked modes.

23.2.7 Errors ÿ

Master Mode Failure Error

When 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 SPI to be shut down; or the MSTR bit is cleared, causing SPI to enter slave mode. If the ERRIE bit is already set, an interrupt will also be generated. Steps to clear the MODF bit: First, perform a read or write operation on R16_SPI1_STATR, then write to R16_SPI1_CTLR1.

Overflow error

An overflow error occurs if the host sends data but there is still unread data in the device's receive buffer. The OVR bit is set, and an interrupt is generated if ERRIE is set. A transmit overflow error should restart the current transmission. Reading the data register and then the status register will clear this bit.

ÿ CRC error

A CRC check error will occur when the received CRC checksum and the value of RXCRCR do not match, and the CRCERR bit will be set.

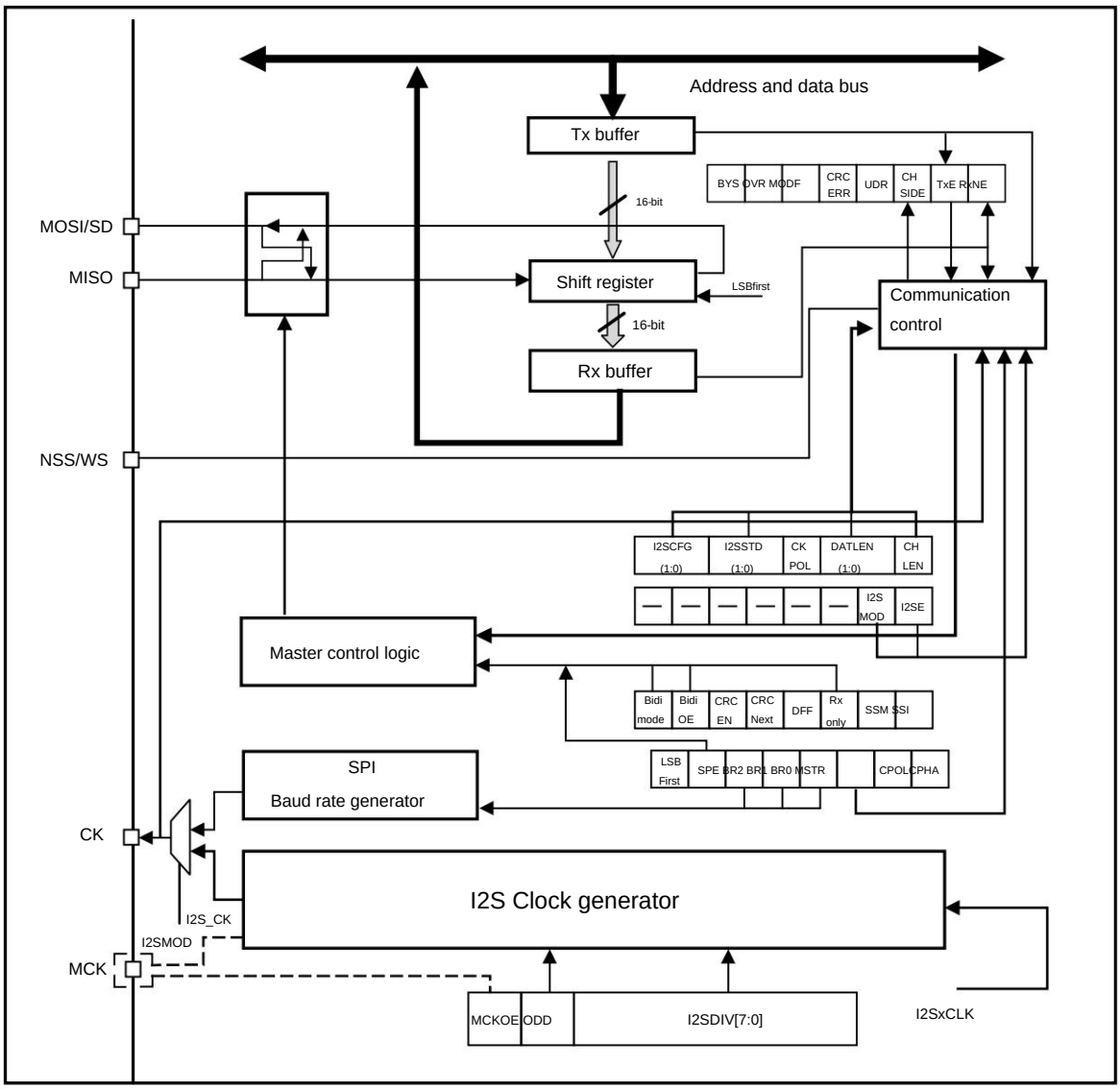
23.2.8 Interrupts

The SPI module supports five interrupt sources. The events of an empty transmit buffer and a non-empty receive buffer will set the TXE and RXNE bits respectively, generating an interrupt when the TXEIE and RXNEIE bits are set. In addition, the three errors mentioned above will also generate interrupts: MODF, OVR, and CRCERR. When the ERRIE bit is enabled, these three errors will also generate error interr

23.3 I2S Function Description

23.3.1 I2S Overview

Figure 23-3 I2S Structure Diagram



The I2S function is enabled by setting the I2SMOD bit in the I2SCFGR register. At this point, the SPI module can be used as an I2S audio source. Interface. I2S and SPI share 3 pins:

- SD: Serial data (mapped to MOSI pin), used to send and receive data from two time-division multiplexed channels;
- WS: Word Select (mapped to NSS pin), used as a data control signal output in master mode and as an input in slave mode;
- CK: Serial clock (mapped to the SCK pin), used as a clock signal output in master mode and as an input in slave mode.

When certain external audio devices require a master clock, an additional pin can be used to output the clock:

- MCK: Master clock (independent mapping). When I2S is configured in master mode and the MCKOE bit in the I2SPR register is 1, it acts as the input clock. An additional clock signal pin is provided. The frequency of the output clock signal is preset to 256 × Fs, where Fs is the sampling rate of the audio signal. frequency.

When set to master mode, I2S uses its own clock generator to generate the clock signal for communication. This clock generator is also the master clock. The clock source for the clock output. In I2S mode, there are two additional registers: one is the I2SPR register, which is related to the clock generator configuration. Another is the I2S general configuration register I2SCFGR (which can set the audio standard, slave/master mode, data format, packet frame, and clock frequency).

(Parameters such as sex). In I2S mode, register CTLR1 and all CRCR registers are not used. Similarly, in I2S mode, registers are not used.

The SSOE bit of register CTLR2, and the MODF and CRCERR bits of register STARTR. I2S uses the same register DATAR as SPI.

Used for 16-bit wide mode data transmission.

23.3.2 Supported Audio Protocols The three-

wire bus supports time-division multiplexing of audio data on two channels: the left channel and the right channel, but only one 16-bit register is used.

To send or receive data, the software must write the appropriate data based on the channel currently in transmission when writing data to the data register.

Similarly, when reading register data, the channel to which the received data belongs is determined by checking the CHSIDE bit of the STAR register.

The left channel always sends data before the right channel (the CHSIDE bit is meaningless under the PCM protocol). There are four available combinations of data and packet frames.

Data can be sent in the following four formats:

- Pack 16-bit data into a 16-bit frame
- Pack 16-bit data into a 32-bit frame
- Pack 24-bit data into a 32-bit frame
- Pack 32-bit data into a 32-bit frame

When using 16-bit data to expand to a 32-bit frame, the first 16 bits (MSB) are meaningful data, and the last 16 bits (LSB) are forced to be undefined.

0. This operation requires no software intervention and no DMA request (only one read or write operation is needed). 24-bit and 32-bit data frames require...

For the CPU to perform two read or write operations on the DATAR register, two DMA transfers are required when using DMA. For 24-bit data,

When extended to 32 bits, the lowest 8 bits are set to 0 by hardware. For all data formats and communication standards, the most significant bit (MSB) is always sent first.

The I2S interface supports four audio standards, which can be selected by setting the I2SSTD[1:0] bits and PCMSYNC bits of the I2SCFGR register.

23.3.2.1 I2S Philips Standard

Under this standard, pin WS is used to indicate which channel the data being transmitted belongs to. One pin before transmitting the first bit of data (MSB)...

This pin is active during the clock cycle. The sender changes the data on the falling edge of the clock signal (CK), and the receiver reads the data on the rising edge.

The WS signal also changes on the falling edge of the clock

signal. Figure 23-4 shows the Philips protocol waveform (16/32 full precision, CPOL=0).

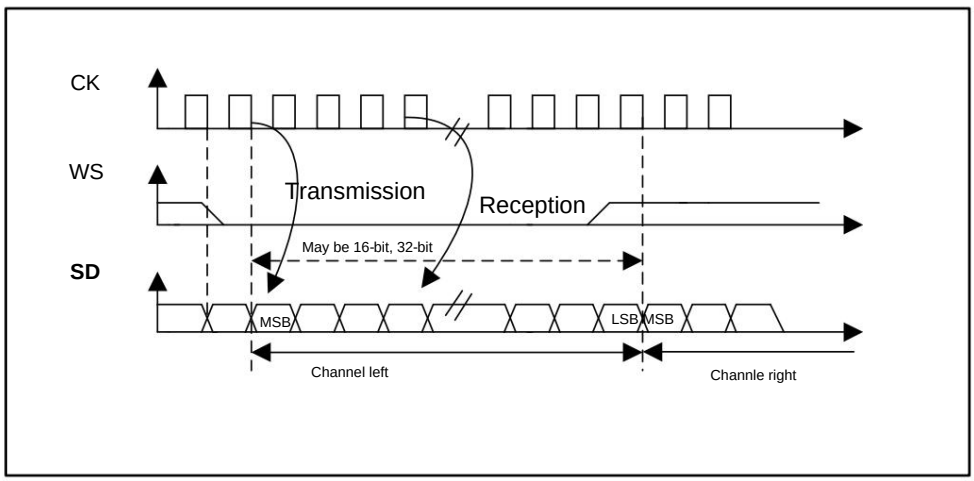
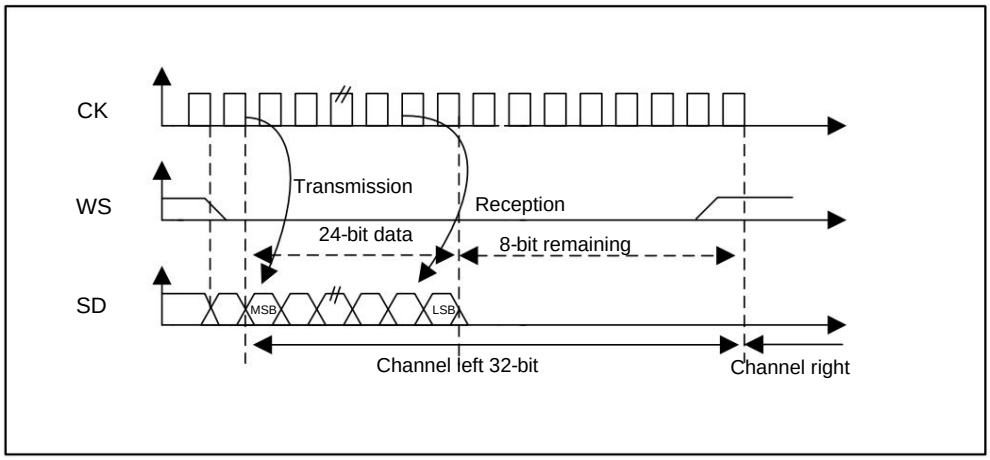
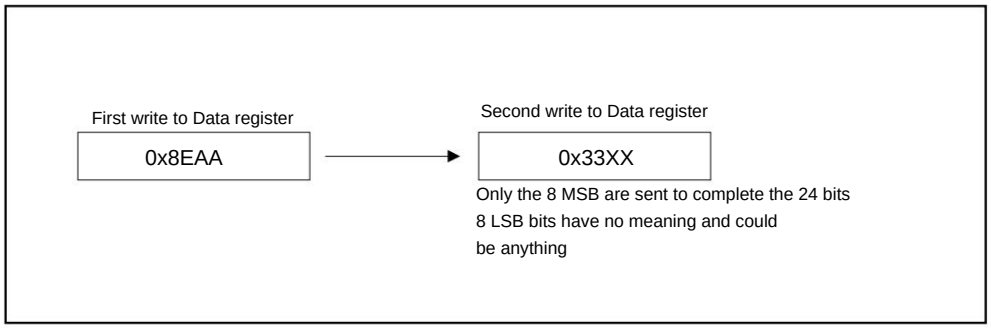


Figure 23-5 Philips protocol waveform (24-bit frame, CPOL=0)



This mode requires two read or write operations to the SPI_DATAR register. In transmit mode: if you need to send 0x8EAA33... (24-bit):



In receive mode: If 0x8EAA33 is received:

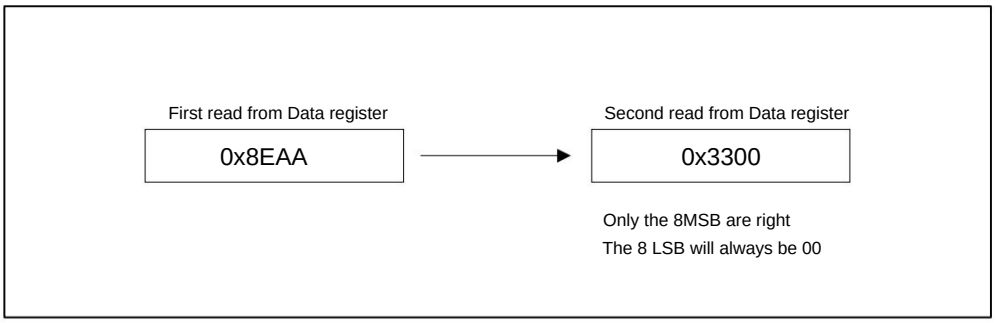
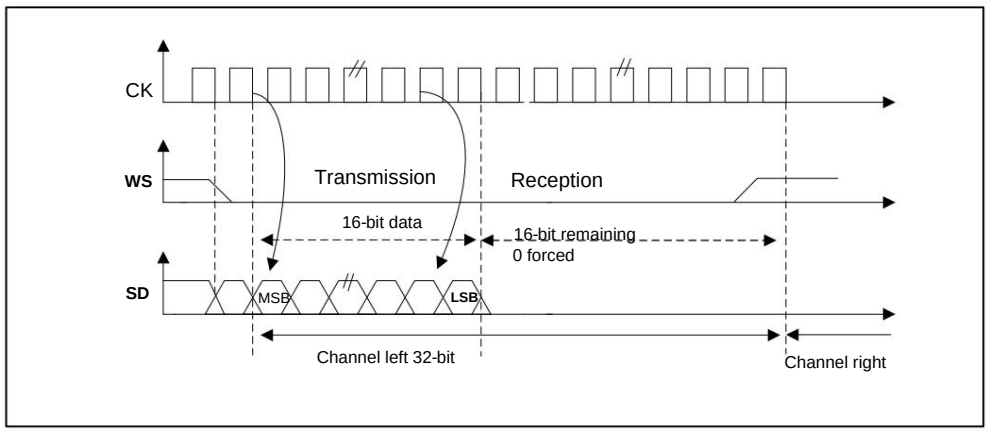


Figure 23-6 Philips Protocol Standard Waveform (16-bit extended to 32-bit packet frame, CPOL=0)



During the I2S configuration phase, if you choose to extend the 16-bit data to a 32-bit audio frame, you only need to access the DATAR register once. The lower 16 bits of the extended 32-bit data are set to 0x0000 by hardware. If the data to be transmitted or received is 0x76A3 (extended to 32 bits is...), then...

(0x76A30000), only one operation is needed with DATAR. During transmission, the MSB needs to be written to the DATAR register; a TXE flag of 1 indicates...

This indicates that new data can be written, and an interrupt can be generated if the corresponding interrupt is enabled. Transmission is handled by hardware, even before transmission occurs.

Sending the last 16 bits of 0x0000 will also set TXE and generate a corresponding interrupt. During reception, after each high 16-bit half-word (MSB) is received,

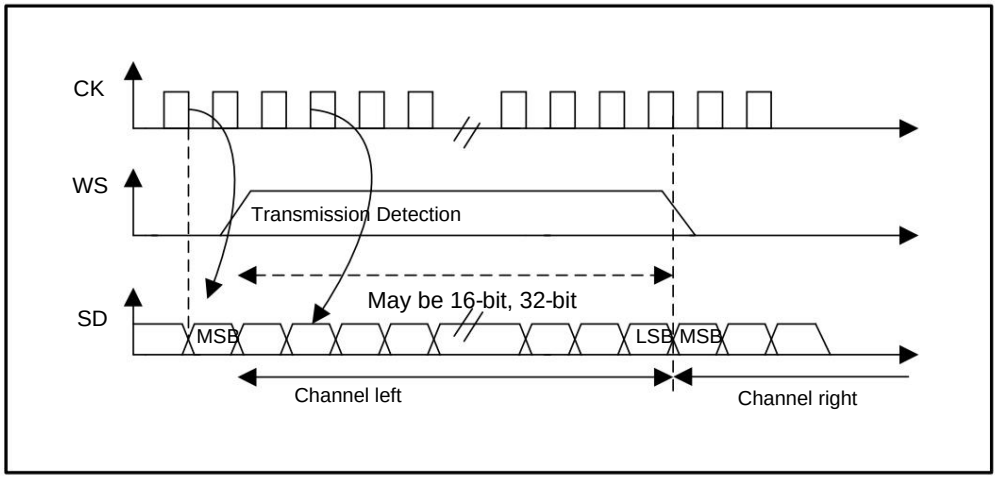
Setting the RXNE flag to 1 allows an interrupt to be generated if the corresponding interrupt is enabled. This provides more time between two reads and writes.

This can prevent overflowing or bubbling.

23.3.2.2 MSB Alignment Standard

Under this standard, the WS signal and the first data bit, i.e., the most significant bit (MSB), are generated simultaneously.

Figure 23-7 MSB Alignment 16-bit or 32-bit Full Precision (CPOL = 0)



The sender changes the data on the falling edge of the clock signal; the receiver reads the data on the rising edge.

Figure 23-8 MSB aligned to 24-bit data, CPOL = 0

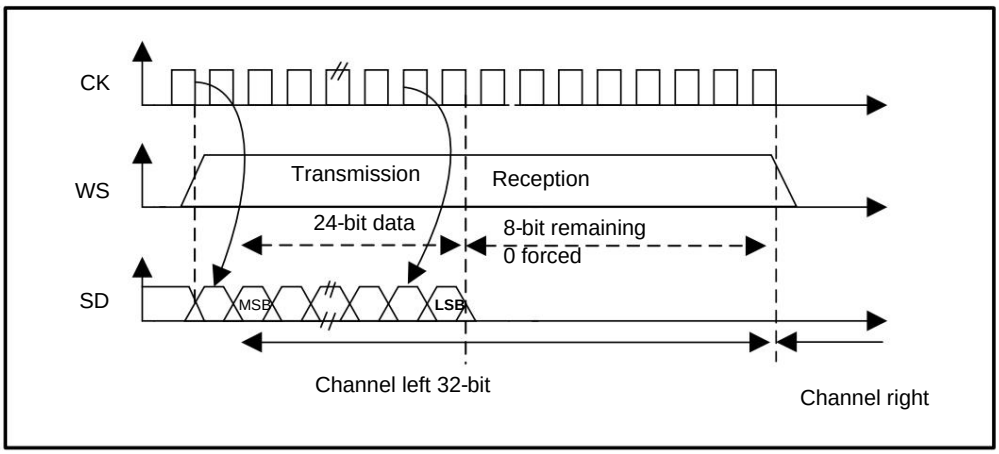
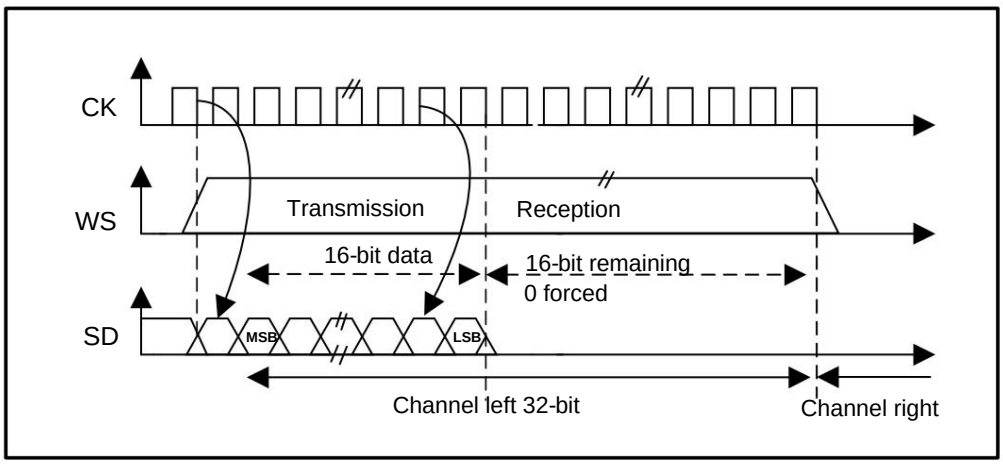


Figure 23-9 MSB aligned 16-bit data extended to a 32-bit packet frame, CPOL = 0



23.3.2.3 LSB Alignment Standard

This standard is similar to the MSB alignment standard (there is no difference in 16-bit or 32-bit full-precision frame formats).

Figure 23-10 LSB aligned to 16 or 32-bit full precision, CPOL = 0

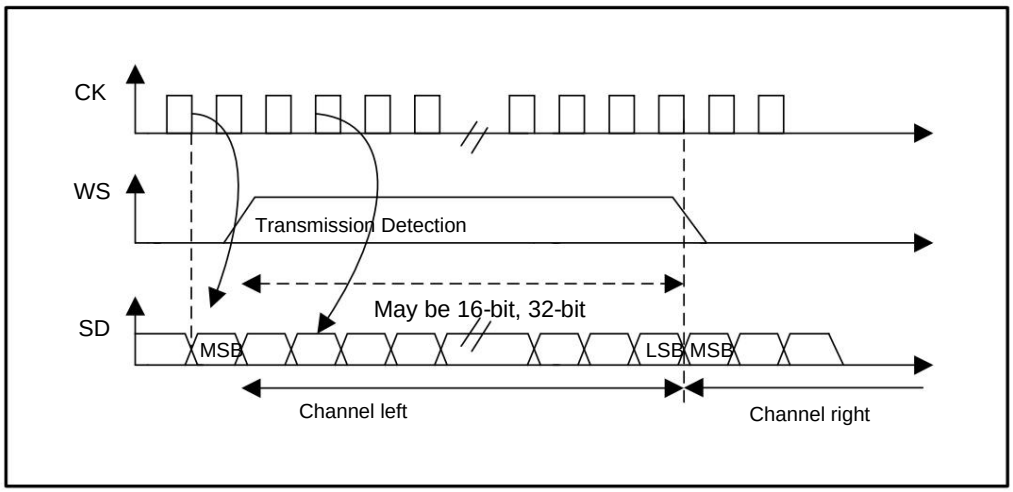
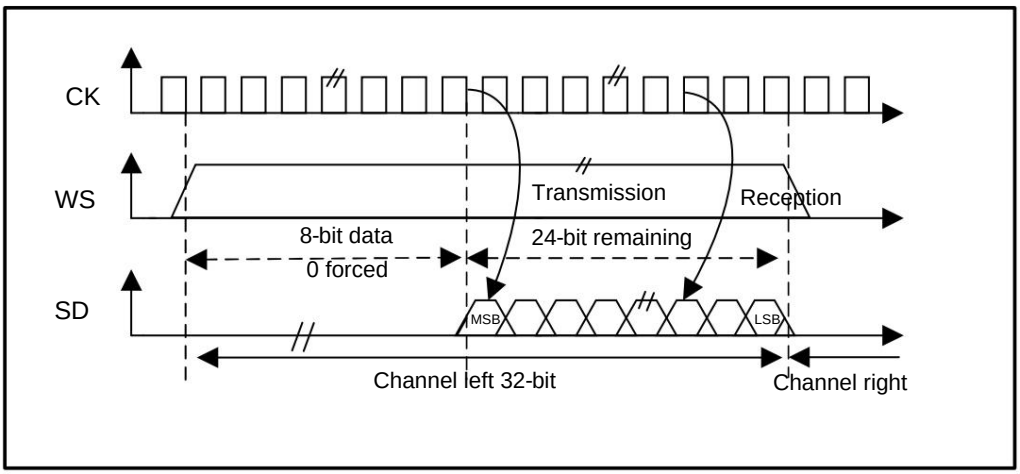
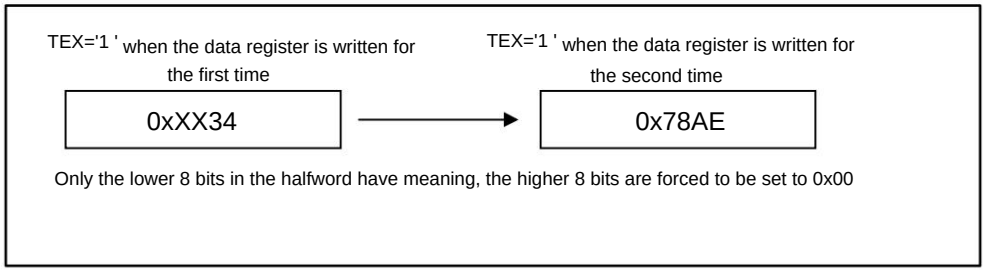


Figure 23-11 LSB aligned to 24-bit data, CPOL = 0



In transmit mode, to send the data 0x3478AE, the DATAR register needs to be written twice via software or DMA.



To receive data 0x3478AE in receive mode, the register needs to be modified during two consecutive RXNE events.

DATAR performs one read operation.

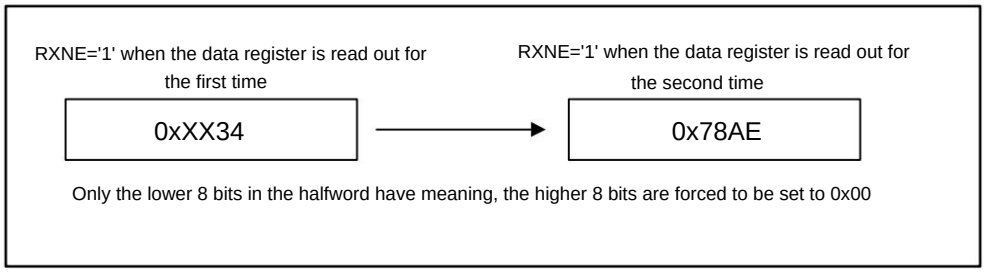
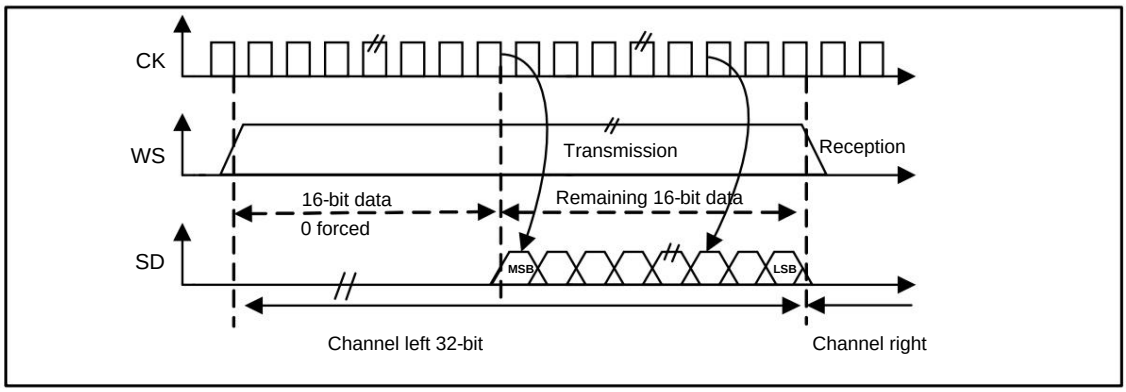


Figure 23-12 shows the LSB-aligned 16-bit data extended to a 32-bit packet frame, with CPOL = 0.



During the I2S configuration phase, if you choose to extend the 16-bit data to a 32-channel frame, you only need to access the DATAR register once. At this point,

The high half-word (16-bit MSB) after being extended to 32 bits is set to 0x0000 by hardware.

If the data to be transmitted or received is 0x76A3 (extended to 32 bits is 0x000076A3), only one DATAR operation is needed.

When sending, if TXE is 1, the user needs to write the data to be sent (i.e., 0x76A3). This is used to expand the 0x0000 portion to 32 bits.

The data is first sent out by the hardware; the next TXE event occurs once valid data begins to be sent from the SD pin. During reception, once...

A valid data (instead of the 0x0000 portion) is received, i.e., an RXNE event occurs. This allows for more time between the two read and write operations.

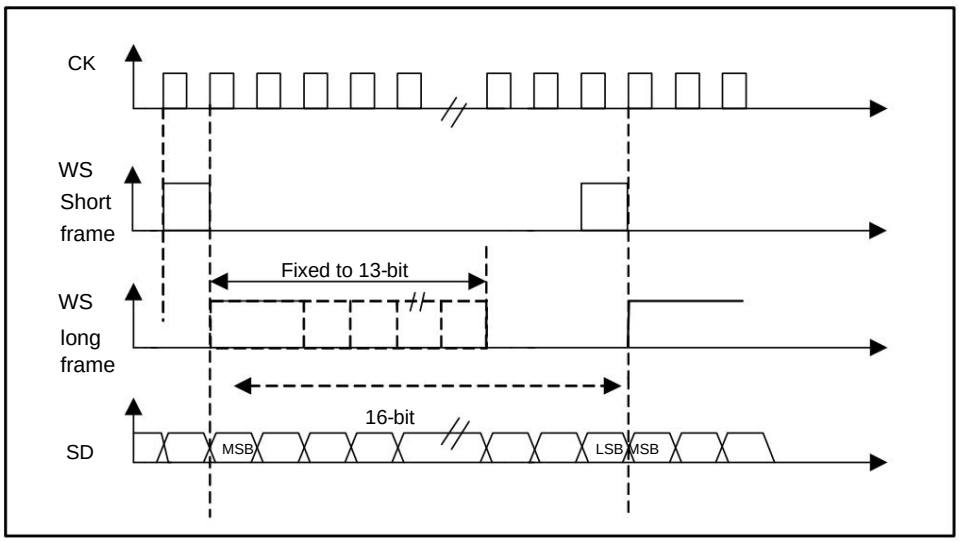
It can prevent overflowing or spilling.

23.3.2.4 The PCM standard

does not include channel selection information. The PCM standard offers two available frame structures: short frames or long frames, which can be selected through...

Select by setting the PCMSYNC bit of the I2SCFGR register.

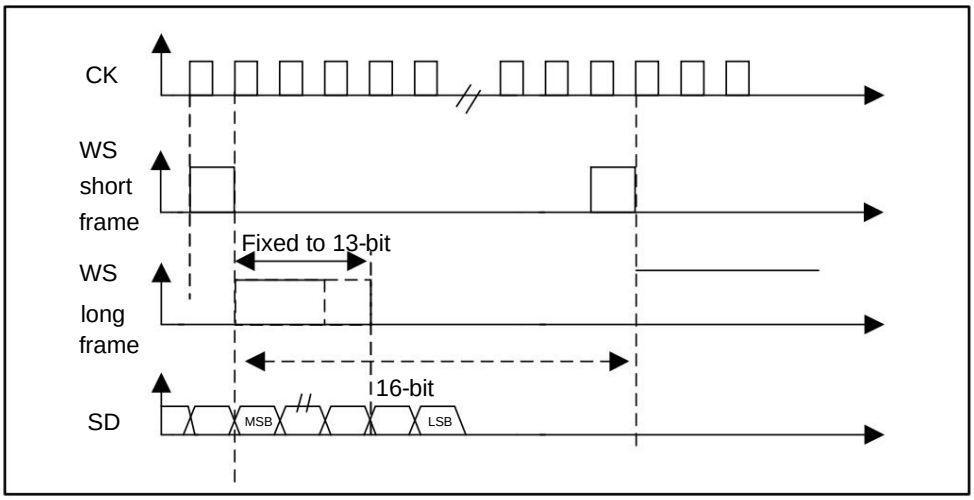
Figure 23-13 PCM standard waveform (16 bits)



For long frames, in master mode, the valid duration of the WS signal used for synchronization is fixed at 13 bits. For short frames, the WS signal used for synchronization...

The signal length is only 1 bit.

Figure 23-14 PCM standard waveform (16 bits)



Regardless of the mode (master or slave) or the synchronization method (short frame or long frame), the two consecutive frames of data and the two synchronization signals...

The time difference between them (even in slave mode) needs to be determined by setting the DATLEN and CHLEN bits of the I2SCFGR register.

23.3.3 The bit rate of the I2S clock

generator determines the data flow on the I2S data lines and the frequency of the I2S clock signal. I2S bit rate = per channel

Number of bits × Number of channels × Audio sampling frequency.

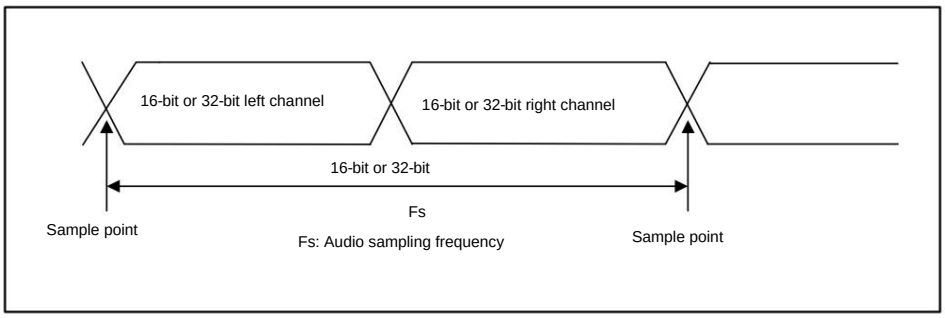
For an audio signal with left and right channels and 16 bits, the I2S bit rate is calculated as follows:

The I2S bit rate is calculated as follows: I2S bit

rate = 16 × 2 × Fs. If the packet length is 32 bits, the I2S bit rate is

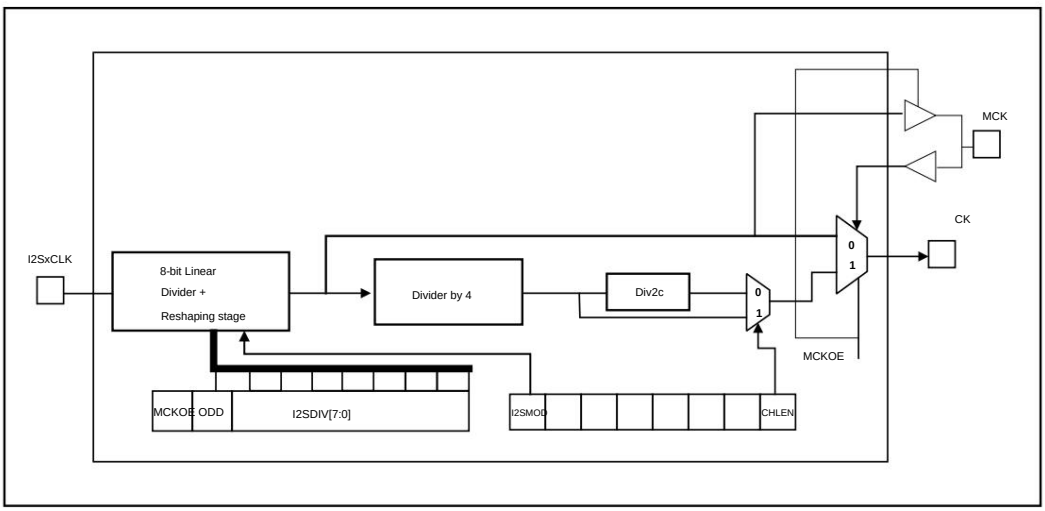
calculated as follows: I2S bit rate = 32 × 2 × Fs

Figure 23-15 Definition of audio sampling frequency



In main mode, the linear crossover needs to be set correctly in order to obtain the required audio frequencies.

Figure 23-16 I2S Clock Generator Structure



The clock source for I2SxCLK in the diagram is the system clock (i.e., the HSI, HSE, or PLL driving the HB clock). I2SxCLK can come from...
The SYSCLK, or PLL3 VCO (2xPLL3CLK) clock, can be selected via the I2S2SRC and I2S3SRC bits of the RCC_CFGR2 register.
The audio sampling frequency can be 96kHz, 48kHz, 44.1kHz, 32kHz, 22.05kHz, 16kHz, 11.025kHz, or 8kHz (or...
(Any value within this range). To obtain the desired frequency, the linear divider must be set according to the following formula:

When the master clock needs to be generated (the MCKOE bit in the SPI_I2SPR register is 1):

When the channel frame length is 16 bits, Fs = I2SxCLK / [(16*2) * ((2*I2SDIV) + ODD)*8] When the channel frame length is 32 bits, Fs =
I2SxCLK / [(32*2) * ((2*I2SDIV) + ODD)*4] When the master clock is off (MCKOE bit is '0'):

When the audio channel frame length is 16 bits, Fs = I2SxCLK / [(16*2) * ((2*I2SDIV) + ODD)] When the audio channel frame length is 32
bits, Fs = I2SxCLK / [(32*2) * ((2*I2SDIV) + ODD)]

23.3.4 I2S Master Mode Setting

When I2S operates in master mode, the serial clock is output from pin CK, and the word select signal is generated from pin WS. This can be achieved by configuring registers.
Use the MCKOE bit of the I2SPR to select whether to output the master clock (MCK).

23.3.4.1 Configuration Process ÿ

Set the I2SDIV[7:0] register of I2SPR to define the serial clock baud rate that matches the audio sampling frequency. Simultaneously, also set...
Define the ODD bit of the I2SPR register.

The CKPOL bit is set to define the communication clock level during idle periods. This is necessary if external DAC/ADC audio devices need to be supplied with this information.

The master clock MCK requires the MCKOE position in register I2SPR to be set.

• Set the I2SMOD bit of register I2SCFGR to 1 to activate the I2S function. Set the I2SSTD[1:0] and PCMSYNC bits to select the I2S standard to use, and set CHLEN to select the data bit width per channel. Also, set the I2SCFG[1:0] bits of register SPI_I2SCFGR to select the I2S master mode and direction (transmitter or receiver). • If needed, you can enable the required interrupt and DMA functions by setting register CR2.

• The I2SE bit of register I2SCFGR must be set to 1. • Pins WS and CK need to be configured in output mode. If the MCKOE bit of register SPI_I2SPR is 1, pin MCK must also be configured in output mode.

23.3.4.2 Transmission Procedure

The transmission procedure begins when one half-word (16 bits) of data is written to the transmit buffer. Assume the first data written to the transmit buffer corresponds to the left channel data. When data is moved from the transmit buffer to the shift register, the TXE flag is set to 1. At this point, the corresponding right channel data must be written to the transmit buffer. The CHSIDE flag indicates which channel the data to be transmitted corresponds to. The value of the CHSIDE flag is updated when TXE is 1, therefore it is meaningful when TXE is 1. A complete data frame is considered complete only after the data from the left channel, followed by the right channel, has been transmitted. Partial data frames, such as those containing only the left channel data, cannot be transmitted.

As the first bit of data is transmitted, half-word data is transferred in parallel to a 16-bit shift register. Subsequent bits are then transmitted sequentially from pin MOSI/SD in high-order order. Each time data is shifted from the transmit buffer to the shift register, the TXE flag is set to 1. If the TXEIE bit in register CR2 is 1, an interrupt is generated.

To ensure continuous audio data transmission, it is recommended to write the next audio data to be transmitted into the DATAR register before the current transmission is completed. Data. It is recommended that when you want to disable the I2S function, you wait for the flags TXE=1 and BSY=0 before clearing the I2SE bit to 0.

23.3.4.3 Receiving Procedure The

configuration steps for the receiving procedure are the same as the sending procedure, except for point 3 (see the aforementioned "Sending Procedure"). The main receiving mode needs to be selected by configuring I2SCFG[1:0]. Regardless of the data and channel length, audio data is always received in 16-bit packets. That is, after each time the receive buffer is filled, the RXNE flag is set to 1. If the RXNEIE bit in register CR2 is 1, an interrupt is generated. Depending on the configured data and channel length, receiving data from the left or right channel will require one or two processes to transfer the data to the receive buffer. The RXNE flag can be cleared by reading the DATAR register. CHSIDE is updated after each reception. Its value depends on the WS signal generated by the I2S unit. If new data is received before the previously received data has been read, an overflow occurs, the OVR flag is set to 1, and if the ERRIE bit in register CR2 is 1, an interrupt is generated, indicating an error has occurred. To disable I2S, special procedures are required to ensure the I2S module completes a transmission cycle without initiating a new data transmission. The procedure depends on the data configuration, channel length, and audio protocol mode: ÿ 16-bit data is expanded to a 32-bit channel length (DATLEN=00 and CHLEN=1), using LSB (low-order bit) alignment .

(I2SSTD=10) a) Wait for the penultimate (n-1) RXNE=1; b) Wait for 17 I2S clock cycles (using software delay); c) Turn off I2S (I2SE=0). ÿ 16-bit data extended to a 32-bit channel length (DATLEN=00 and CHLEN=1), using MSB (high-order) alignment, I2S or PCM mode (I2SSTD=00, I2SSTD=01, or I2SSTD=11 respectively): a) Wait for the last RXNE=1; b) Wait for 1 I2S clock cycle (using software delay); c) Turn off I2S (I2SE=0). ÿ For all other combinations of DATLEN and CHLEN, and any audio mode selected by I2SSTD, turn off I2S as follows: a) Wait for the penultimate (n-1) RXNE=1; b) Wait for one I2S clock cycle (using software delay); c) Turn off I2S (I2SE=0).

Note: The flag remains low throughout the transmission.BSY.

23.3.5 I2S Slave Mode In slave mode,

I2S can be configured for both send and receive modes. The configuration method for slave mode is basically the same as that for master mode.

The process is as follows. In slave mode, the I2S interface does not need to provide a clock. Both the clock signal and the WS signal are provided by the external master I2S device.

It is connected to the corresponding pin. Therefore, the user does not need to configure the clock.

The configuration steps are as follows:

ÿ Set the I2SMOD bit of the I2SCFGR register to activate the I2S function; set I2SSTD[1:0] to select the I2S standard to use;

Set DATLEN[1:0] to select the number of bits of data; set CHLEN to select the number of bits of data per channel. Set the I2SCFGR register.

I2SCFG[1:0] selects the data direction (sender or receiver) of the I2S from mode.

ÿ Configure register CR2 to enable the required interrupt and DMA functions as needed.

The I2SE bit of the I2SCFGR register must be set to 1.

23.3.5.1 Transmission Procedure

The transmission procedure begins when the external master device sends a clock signal and when the NSS_WS signal requests data transmission. Enable must be performed first.

The external master device can only begin communication after the slave device has written data to the I2S data register. This relates to MSB and LSB alignment in I2S.

The mode, the first data item written to the data register corresponds to the left channel data. When communication begins, data is transferred from the transmit buffer to...

The shift register is used, and then the TXE flag is set to 1; at this point, the data item corresponding to the right channel should be written to the I2S data register. Flag bit

CHSIDE indicates which channel the data to be transmitted corresponds to. Compared to the master mode transmission process, in slave mode, CHSIDE retrieves...

This depends on the WS signal from the external master I2S. This means that the slave I2S must prepare the clock signal before receiving it from the master.

A data to be sent. A WS signal of 1 indicates that the left channel will be sent first.

23.3.5.2 The receiving process

configuration steps are the same as the sending process, except for point 1. The primary receiving mode needs to be selected by configuring I2SCFG[1:0]. Regardless of...

Regardless of the data type and channel length, audio data is always received in 16-bit packets. This means that each time the receive buffer is filled, the RXNE flag is set to 1.

An interrupt is generated if the RXNEIE bit in the I2S_CTLR2 register is 1. Depending on the data and channel length settings, the left channel is received.

The right channel data may require one or two data transfers to the receive buffer. Each time data is received (to be read from the DATAR),...

After that, update CHSIDE, which corresponds to the WS signal generated by the I2S unit. Read the SPI_DATAR register and clear the RXNE bit.

An overflow occurs when new data is received before the previously received data has been read, and the OVR flag is set to 1; if

An interrupt is generated when the ERRIE bit in the I2S_CTLR2 register is 1, indicating an error. To disable the I2S function, you need to...

When RXNE=1 for the last time, the I2SE bit is cleared to 0.

23.3.6 Status Flags There are 3 status

flags for users to monitor the status of the I2S bus.

23.3.6.1 Status Flag (BSY)

The BSY flag is set and cleared by hardware (writing to this bit has no effect). This flag indicates the status of the I2S communication layer. When this bit is 1...

This indicates that I2S communication is in progress, with one exception: in master receive mode (I2SCFG=11), the BSY flag is always active during reception.

The value is low. Before the software shuts down the SPI module, the BSY flag can be used to detect whether the transmission has ended, thus avoiding damage to the last data.

This is a secondary transmission, therefore the following procedure must be strictly followed. When the transmission begins, the BSY flag is set to 1, unless the I2S module is in master mode.

Receive mode. This flag is cleared when the transmission ends or when the I2S module is turned off. In main transmit mode, this flag is cleared when communication is continuous.

In slave mode, the BSY flag remains high throughout the entire transmission; in slave mode, the BSY flag remains high for one I2S interval between each data item transmission.

It decreases within the clock cycle.

23.3.6.2 Transmit Buffer Flag (TXE)

A flag of 1 indicates that the transmit buffer is empty, and new data to be transmitted can be written to the transmit buffer. The transmit buffer already contains...

When data is available, the flag is cleared to 0. When I2S is disabled (I2SE bit is 0), the flag is also 0.

23.3.6.3 Receive Buffer Not Empty Flag (RXNE)

This flag being set to 1 indicates that there is valid received data in the receive buffer. It is cleared to 0 when reading the DATAR register.

23.3.6.4 Channel Marker (CHSIDE)

In transmit mode, this flag is refreshed when TXE is high, indicating the channel from which data is transmitted from the SD pin. If in

An underflow error occurred in send mode, and the value of this flag is invalid. I2S needs to be turned off and then on again before resuming communication.

In receive mode, this flag is refreshed when data is received in the DATAR register, indicating the channel on which the received data is located. Note that if...

An error has occurred; this flag is meaningless, and I2S needs to be turned off and then on again. Under the PCM standard, regardless of whether it's a short frame format or a long frame format...

This flag is meaningless. If the OVR or UDR flag in register STTRA is 1, and the ERRIE flag in register CR2 is 1,

This will generate an interrupt. (After the interrupt source has been cleared) The interrupt flag can be cleared by reading the START register.

23.3.7 Error Flags

23.3.7.1 Underflow Flag (UDR)

In transmit mode, if the new data has not yet been written to the DATAR register when the first clock edge of the data transmission arrives,

This flag will be set to 1. This flag is only valid after the I2SMOD bit in register I2SCFGR is set to 1. If register CR2's ERRIE...

An interrupt will be generated if the flag is set to 1. This flag can be cleared by reading the STAR register.

23.3.7.2 Overflow Flag (OVR)

If new data is received before the previously received data has been read, an overflow occurs, and this flag is set to 1.

If the ERRIE bit in register CTLR2 is 1, an interrupt is generated indicating an error has occurred. In this case, the contents of the receive buffer will not be refreshed from the previous data.

The transmitting device sends new data. A read operation on the DATAR register returns the last correctly received data. All others overflow.

After this event, all 16 bits of data sent by the transmitting device will be lost. This can be cleared by first reading the DATAR register and then reading the STATR register.

Flag position.

23.3.8 I2S Interrupts I2S has

four interrupt sources. The events of an empty transmit buffer and a non-empty receive buffer will set TXE and RXNE respectively.

An interrupt will be generated if both the TXEIE and RXNEIE bits are set. If the previously received data has not yet been read, and another interrupt occurs...

Upon receiving new data, an overflow occurs; if ERRIE is set, an overflow interrupt will be generated. In send mode, if data transmission...

When the first clock edge arrives, new data has not yet been written to the DATAR register. If ERRIE is set, an underflow interrupt will be generated.

23.3.9 DMA Function

DMA operates exactly the same in I2S mode as in SPI mode, except that the CRC function is unavailable. This is because in I2S mode...

There is no data transmission protection system.

23.4 Register Description

Table 23-1 List of SPI1 Related Registers

name	Access address	describe	Reset value
R16_SPI1_CTLR1	0x40013000	SPI1 Control Register 1	0x0000
R16_SPI1_CTLR2	0x40013004	SPI1 Control Register 2	0x0000
R16_SPI1_STATR	0x40013008	SPI1 Status Register	0x0002
R16_SPI1_DATAR	0x4001300C	SPI1 Data Register	0x0000

R16_SPI1_CRCR	0x40013010	SPI1 polynomial register	0x0007
R16_SPI1_RCRCR	0x40013014	SPI1 Receive CRC Register	0x0000
R16_SPI1_TCRCR	0x40013018	SPI1 transmits CRC register	0x0000
R16_SPI1_HSCR	0x40013024	SPI1 High-Speed Control Register	0x0000

Table 23-2 List of SPI2 Related Registers

name	Access address	describe	Reset value
R16_SPI2_CTLR1	0x40003800	SPI2 Control Register 1	0x0000
R16_SPI2_CTLR2	0x40003804	SPI2 Control Register 2	0x0000
R16_SPI2_STATR	0x40003808	SPI2 Status Register	0x0002
R16_SPI2_DATAR	0x4000380C	SPI2 Data Register	0x0000
R16_SPI2_CRCR	0x40003810	SPI2 polynomial register	0x0007
R16_SPI2_RCRCR	0x40003814	SPI2 Receive CRC Register	0x0000
R16_SPI2_TCRCR	0x40003818	SPI2 transmit CRC register	0x0000
R16_SPI2_I2S_CFGR	0x4000381C	SPI2_I2S Configuration Register	0x0000
R16_SPI2_I2SPR	0x40003820	SPI2_I2S prescaler register	0x0000
R16_SPI2_HSCR	0x40003824	SPI2 High-Speed Control Register	0x0000

Table 23-3 List of SPI3 Related Registers

name	Access address	describe	Reset value
R16_SPI3_CTLR1	0x40003C00	SPI3 Control Register 1	0x0000
R16_SPI3_CTLR2	0x40003C04	SPI3 Control Register 2	0x0000
R16_SPI3_STATR	0x40003C08	SPI3 Status Register	0x0002
R16_SPI3_DATAR	0x40003C0C	SPI3 Data Register	0x0000
R16_SPI3_CRCR	0x40003C10	SPI3 polynomial register	0x0007
R16_SPI3_RCRCR	0x40003C14	SPI3 Receive CRC Register	0x0000
R16_SPI3_TCRCR	0x40003C18	SPI3 transmit CRC register	0x0000
R16_SPI3_I2S_CFGR	0x40003C1C	SPI3_I2S Configuration Register	0x0000
R16_SPI3_I2SPR	0x40003C20	SPI3_I2S prescaler register	0x0000
R16_SPI3_HSCR	0x40003C24	SPI3 High-Speed Control Register	0x0000

Table 23-4 List of SPI4 Related Registers

name	Access address description		Reset value
R16_SPI4_CTLR1	0x40004000	SPI4 Control Register 1	0x0000
R16_SPI4_CTLR2	0x40004004	SPI4 Control Register 2	0x0000
R16_SPI4_STATR	0x40004008	SPI4 Status Register	0x0002
R16_SPI4_DATAR	0x4000400C	SPI4 Data Register	0x0000
R16_SPI4_CRCR	0x40004010	SPI4 polynomial register	0x0007
R16_SPI4_RCRCR	0x40004014	SPI4 Receive CRC Register	0x0000
R16_SPI4_TCRCR	0x40004018	SPI4 transmit CRC register	0x0000
R16_SPI4_HSCR	0x40004024	SPI4 High-Speed Control Register	0x0000

23.4.1 SPI Control Register 1 (SPIx_CTLR1) (x=1/2/3/4)

Offset address: 0x00

Bit	Name	access		Reset value
15	BIDIMODE	RW	Description of the one-way data mode enable bit: 1: Select single-line bidirectional mode; 0: Select dual-line bidirectional mode.	0
14	BIDIOE	RW	Single-line output enable bit, used in conjunction with BIDIMODE. 1: Enable output, send only; 0: Output is disabled, only receive.	0
13	CRCEN	RW	Hardware CRC check enable bit; this bit can only be used when SPE is 0. Write; this bit can only be used in full-duplex mode. 1: Start CRC calculation; 0: CRC calculation is disabled.	0
12	CRCNEXT	RW	After the next data transmission, send the CRC register. The value. This should be the last one written to the data register. Set the bit immediately after the data is received. 1: Send the CRC check result; 0: Continue sending data from the data register.	0
11	DFF	RW	Data frame length bit, this bit can only be written when SPE is 0. 1: Use 16-bit data length for sending and receiving; 0: Use 8 bits of data length for sending and receiving.	0
10	RXONLY	RW	In dual-wire mode, only one bit is received; this bit is used in conjunction with BIDIMODE. Use this bit. Setting this bit allows the device to only receive and not send. 1: Receive only, simplex mode; 0: Full-duplex mode.	0
9	SSM	RW	The chip select pin management bit determines the voltage level of the NSS pin. Hardware or software control. 1: Software control of the NSS pin; 0: Hardware control NSS pin.	0
8	SSI	RW	Chip select pin control bit; this bit is active when SSM is set. Determines the level of the NSS pin. 1: NSS is high level; 0: NSS is low.	0
7	LSBFIRST	RW	Frame format control bit. This bit cannot be modified during communication. 1: Send the LSB first; 0: Send the MSB first.	0
6	SPE	RW	SPI enable bit. 1: Enable SPI; 0: Disable SPI.	0
[5:3] BR[2:0]		RW	This is the baud rate setting field; it cannot be modified during communication. When the HSRXEN bit is 0, the SCK frequency is: 000: FHCLK/2; 001: FHCLK/4; 010: FHCLK/8; 011: FHCLK/16; 100: FHCLK/32; 101: FHCLK/64; 110: FHCLK/128; 111: FHCLK/256. When the HSRXEN bit is 1, the SCK frequency is:	000b

			000: FHCLK/2; 001: FHCLK/3; 010: FHCLK/4; 011: FHCLK/5; 100: FHCLK/6; 101: FHCLK/7; 110: FHCLK/8; 111:FHCLK/9.	
2	MSTR	RW	Master/slave setting bit, this bit cannot be modified during communication. 1: Configure the main device; 0: Configured as a slave device.	0
1	CPOL	RW	Clock polarity selection bit; this bit cannot be modified during communication. 1: When idle, SCK remains high; 0: SCK remains low during idle state.	0
0	CPHA	RW	Clock phase setting bit, which cannot be modified during communication. 1: Data sampling begins from the second clock edge; 0: Data sampling begins from the first clock edge.	0

23.4.2 SPI Control Register 2 (SPIx_CTLR2) (x=1/2/3/4) Offset Address: 0x04

Bit	name	access	describe	Reset value
[15:8] Reserved		RO reserved.		0
7	TXEIE	RW	Transmit buffer empty interrupt enable bit: 1: Enables an interrupt when TXE is set; 0: Disables interrupt generation when TXE is set.	0
6	RXNEIE	RW	Receive buffer not empty interrupt enable bit: 1: Enables an interrupt to be generated when RXNE is set; 0: Disables interrupt generation when RXNE is set.	0
5	ERRIE	RW	Error interrupt enable bit: 1: Allow error reporting (CRCERR, OVR, MODF) when errors occur. An interrupt was generated; 0: Disable error generation (CRCERR, OVR, MODF) An interrupt occurs.	0
[4:3] Reserved		RO is reserved.		0
2	SSOE	RW	Enable SS output. Disabling SS output allows operation in multi-master mode. The formula is as follows. 1: Enable SS output; 0: Disable SS output in main mode.	0
1	TXDMAEN	RW	DMA enable bit for transmit buffer: 1: Enable transmit buffer DMA; 0: Disable send buffer DMA.	0
0	RXDMAEN	RW	DMA enable bit for receive buffer: 1: Enable DMA for the receive buffer; 0: Disable receive buffer DMA.	0

23.4.3 SPI Status Register (SPIx_STATR) (x=1/2/3/4) Offset Address: 0x08 Name

Bit		access	describe	Reset value

[15:8] Reserved		RO	Reserved.	0
7	BSY	RO	Busy flag, which is set or reset by hardware. 1: SPI is communicating, or the transmit buffer is not empty; 0: SPI is not communicating.	0
6	OVR	RO	Overflow flag, which is set by hardware and reset by software. 1: An overflow error occurred; 0: No overflow error occurred.	0
5	MODF	RO	Mode error flag, which is set by hardware and reset by software. 1: A pattern error has occurred; 0: No pattern error occurred.	0
4	CRCERR	RW0	CRC error flag bit, which is set by hardware and reset by software. 1: The received CRC value is inconsistent with the RCRCR value; 0: The received CRC value is consistent with the RCRCR value.	0
3	UDR	RO	The underflow flag is set by hardware and reset by software. 1: Overflow occurred; 0: No underflow occurred.	0
2	CHSIDE	RO	The audio channel is set by hardware and reset by software. 1: Need to transmit or receive the right channel; 0: Left channel needs to be transmitted or received.	0
1	TXE	RO	Send buffer empty flag: 1: The send buffer is empty; 0: The send buffer is not empty.	1
0	RXNE	RO	Receive buffer not empty flag: 1: The receive buffer is not empty; 0: The receive buffer is empty. Note: Read DATAR, automatically resets to zero.	0

23.4.4 SPI Data Register (SPIx_DATAR) (x=1/2/3/4) Offset Address: 0x0C Name

Bit		access		Reset value
[15:0] DR[15:0]		RW	Describe the data register. The data register is used to store the received data. Data is stored or pre-stored for transmission, therefore data is stored. The read and write operations of the device actually correspond to different areas, among which Read from the corresponding receive buffer and write to the corresponding send buffer. Data The receiving and sending can be 8 bits or 16 bits, requiring... Determine the number of bits of data to use before transmission. Use 8 bits. When transmitting data bit by bit, only the lower 8 bits of the data register are used. When used, the high 8 bits are forced to 0 upon reception. Use 16 bits. The structure will allow all 16-bit data registers to be used.	0

23.4.5 SPI Polynomial Register (SPIx_CRCR) (x=1/2/3/4) Offset Address: 0x10 Name

Bit		access	describe	Reset value
[15:0] CRCPOLY[15:0]		RW	CRC Polynomial. This field defines the polynomial used for CRC calculation.	0x0007

23.4.6 SPI Receive CRC Register (SPIx_RCRCR) (x=1/2/3/4) Offset

Address: 0x14

Bit	Name	access	This	Reset value
[15:0]	RXCRC[15:0]	RO	describes the received CRC value. It stores the calculated received byte value. The result of the CRC check. Setting the CRCEN bit will reset the register. Memory. The calculation method uses the polynomials used in CRCPOLY. In 8-bit mode, only the lower 8 bits are used in the calculation; in 16-bit mode... All 16 bits will participate in the calculation. This is necessary when BSY is 0. Go and read this register.	0

23.4.7 Transmit CRC Register (SPIx_TCRCR) (x=1/2/3/4) Offset

Address: 0x18

Bit	Name	access	This	Reset value
[15:0]	TXCRC[15:0]	RO	describes the CRC value sent. It stores the calculated CRC value that has already been sent. The result of the CRC check of the bytes. Setting the CRCEN bit will reset... This register is used for calculation. The calculation method uses CRCPOLY, which is frequently used. Terms. In 8-bit mode, only the lower 8 bits are used in the calculation; in 16-bit mode... In this mode, all 16 bits will participate in the calculation. It is necessary to set BSY to... Read this register when the time is 0.	0

23.4.8 SPI_I2S Configuration Register (SPIx_I2S_CFGR) (x=2/3)

Offset Address:

Bit	0x1C Name	access	describe	Reset value
[15:12]	Reserved	RO	is reserved.	0
11	I2SMOD	RW	I2S mode selection; this bit is only active when SPI or I2S is disabled. It can only be set at certain times. 1: Select I2S mode; 0: Select SPI mode.	0
10	I2SE	RW	I2S is enabled and not used in SPI mode. 1: I2S Enable; 0: I2S is off.	0
[9:8]	I2SCFG[1:0]	RW	I2S mode selection; this bit is only active when I2S is disabled. Can be set: 00: Sent from device; 01: Received from the device; 10: Master device sends; 11: The main device accepts the request.	0
7	PCMSYNC	RW	PCM frame synchronization. This bit is only used when I2SSTD = 11 (using PCM). It is meaningful when it is a standard. 1: Long frame synchronization; 0: Short frame synchronization.	0
6	Reserved	RO	is reserved.	0
[5:4]	I2SSTD[1:0]	RW	I2S standard selection can only be set to 0 when I2S is disabled.	

			That position. 00: I2S Philips Standard; 01: High byte alignment standard (left alignment); 10: Low byte alignment standard (right alignment); 11: PCM Standard.	
3	CKPOL	RW	The static clock polarity, for proper operation, this bit only needs to be set to neutral. This can only be configured when I2S is turned off. 1: The I2S clock is high in the quiescent state; 0: The I2S clock is at rest when it is at a low level.	0
[2:1]	DATLEN[1:0]	RW	The length of the data to be transmitted; for correct operation, this bit is only required when... This can only be configured when I2S is turned off. 00: 16-bit data length; 01: 24-bit data length; 10: 32-bit data length; 11: Not allowed.	0
0	CHLEN	RW	Channel length; this bit is only written when DATLEN = 00. It only makes sense if it's done properly; otherwise, the channel length would be fixed at 32 by hardware. Bit. 1:32-bit width; 0: 16 bits wide.	0

23.4.9 SPI_I2S Prescaler Register (SPIx_I2SPR) (x=2/3) Offset Address: 0x20 Bit Name [15:10]

Reserved

		access	describe	Reset value
		RO Reserved.		0
9	MCKOE	RW	Master clock output enabled; this bit is only required for proper operation. This setting is only available when I2S is disabled. It is only applicable to I2S master devices. This bit is used in the mode. 1: Enable master device clock output; 0: Disable master device clock output.	0
8	ODD	RW	Odd-coefficient prescaler, for proper operation, this bit only needs to be turned off. This setting is only available when I2S is disabled. It is only applicable in I2S master mode. Use this bit. 1: Actual frequency division factor = (I²SDIV * 2) + 1; 0: Actual frequency division coefficient = I2SDIV * 2.	0
[7:0]	I2SDIV[7:0]	RW	I2S linear prescaler. For proper operation, this bit is only active when... This setting is only available when I2S is disabled. Only in I2S master mode. Use this bit below. Do not set I2SDIV[7:0] = 0 or I2SDIV[7:0] = 1.	0

23.4.10 SPI High-Speed Control Register (SPIx_HSCR) (x=1/2/3/4) Offset Address: 0x24 Name

Bit		access	describe	Reset value
[15:3]	Reserved	RO is reserved.		0

2	HSRXEN2	RW	SPI High-Speed Read Mode 2: This mode is only available when HSRXEN is enabled and SPI HCLK is greater than or equal to 0. Use this setting for speeds up to 120Mbps; it is not recommended for speeds below 120Mbps. To achieve higher host read speeds than simply enabling HSRXEN. 1: Enable high-speed read mode 2; 0: Disable high-speed read mode 2.	0
1	Reserved	RO reserved.		0
0	HSRXEN	WO	High-speed read mode enable bit: 1: Enable high-speed read mode; 0: High-speed read mode is off.	0

Chapter 24 USB PD Controller (USBPD)

24.1 Introduction to USB PD Controller

The chip integrates a USB Power Delivery controller and a PD physical layer transceiver PHY, supporting USB Type-C master-slave detection. The BMC encoding/decoding and CRC are supported, and the CC pin supports hardware edge control.

Supports USB PD2.0, PD3.0, and PD3.2 power delivery control; supports SPR and EPR; supports 100W or 240W fast charging. Supports PD Sink and PD Source, as well as DRP applications. Supports PDUSB, UFP, DFP, and DRD applications. use.

Supports PD packets such as SOP, SOP', and SOP"; supports hardware reset via USB PD reset signal frames; supports a maximum packet length of 510 bytes. DMA is supported.

24.2 Register Description

Table 24-1 List of USBPD Related Registers

name	Access address	describe	Reset value
R32_USBPD_CONFIG	0x40024400 PD Configuration Register;		0x00000002
R16_CONFIG	0x40024400 PD Interrupt Enable Register;		0x0002
R16_BMC_CLK_CNT	0x40024402 BMC Sampling Clock Counter;		0x0000
R32_USBPD_CONTROL	0x40024404 PD Control Register;		0x00000000
R16_CONTROL	0x40024404 PD Transceiver Control Register;		0x0000
R8_CONTROL	0x40024404 PD Transceiver Enable Register;		0x00
R8_TX_SEL	0x40024405 PD Transmit SOP Selection Register;		0x00
R16_BMC_TX_SZ	0x40024406 PD Transmit Length Register;		0x0000
R32_USBPD_STATUS	0x40024408 PD Status Register;		0x000000XX
R16_STATUS	0x40024408 PD Interrupt and Data Register;		0x00XX
R8_DATA_BUF	0x40024408 DMA Buffer Data Register;		0xXX
R8_STATUS	0x40024409 PD Interrupt Flag Register;		0x00
R16_BMC_BYTE_CNT	0x4002440A Byte Counter;		0x0000
R32_USBPD_PORT	0x4002440C Port Control Register;		0x00030003
R16_PORT_CC1	0x4002440C CC1 Port control register 0x4002440E		0x0003
R16_PORT_CC2	CC2 Port control register 0x40024410 DMA cache		0x0003
R32_USBPD_DMA	address register		0x00XXXXXX

24.2.1 PD Configuration Register (R32_USBPD_CONFIG) Offset

Address: 0x00

Bit	name
[31:16] R16_BMC_CLK_CNT [15:0]	
	R16_CONFIG

24.2.2 PD Interrupt Enable Register (R16_CONFIG) Offset Address:

0x00 Name

Bit		Access description RW: End of transmission interrupt		Reset value
15	IE_TX_END	enabled.		0

14	IE_RX_RESET	RW	receive reset interrupt enabled.	0
13	IE_RX_ACT	RW	receive complete interrupt enabled.	0
12	IE_RX_BYTE	RW	receive byte interrupt enabled.	0
11	IE_RX_BIT	RW	receive bit interrupt enabled.	0
10	IE_PD_IO	RW	PD IO interrupt enable.	0
9	Reserved	RO	reserved.	0
8	RX_MULTI_0	RO	Receive multiple consecutive 0 bits as an indicator: 1: Received multiple consecutive 0s; 0: No consecutive 0s were received.	0
7	CC_INC	RW	CC bias current adjustment position: 1: Bias current +10%; 0: Default gear.	0
6	CC_TR	RW	CC LV waveform Tr adjustment: 1: Tr increased by 5%; 0: Tr remains unchanged.	0
5	WAKE_POLAR	RW	PD port wake-up level: 1: High level is active; 0: Active low.	0
4	PD_RST_EN	RW	Enable PD mode reset command: 1: Reset; 0: Invalid.	0
3	PD_DMA_EN	RW	Enabling USBPD's DMA is required in normal transfer mode. Must be set to 1: 1: Enable DMA function and DMA interrupt; 0: DMA is off.	0
2	CC_SEL	RW	Select the current PD communication port: 1: Use the CC2 port for communication; 0: Use port CC1 for communication.	0
1	PD_ALL_CLR	RW	Clear all interrupt flags in PD mode: 1: Clear the interrupt flag; 0: Invalid.	1
0	PD_FILT_EN	RW	Control the input filter enable of the PD pin: 1: Enable filtering; 0: Filtering is off.	0

24.2.3 BMC Sampling Clock Counter (R16_BMC_CLK_CNT) Offset Address: 0x02

Bit Name [15:9]

Reserved	[8:0] BMC_CLK_CNT	access	describe	Reset value
		RO	is reserved.	0
		RW	BMC is a transmit or receive sampling clock counter.	0

Note: When transmitting, BMC_CLK_CNT = (HCLK/CC communication frequency)/2; when receiving, BMC_CLK_CNT = (HCLK/CC communication frequency)/4*3

24.2.4 PD Control Register (R32_USBPD_CONTROL)

Offset address: 0x04

Bit	name
[31:16] R16_BMC_TX_SZ [15:0]	
	R16_CONTROL

24.2.5 PD Transceiver Control Register (R16_CONTROL) Offset

Address: 0x04

	name
Position	R8_TX_SEL
[15:8] [7:0]	R8_CONTROL

24.2.6 PD Transceiver Enable Register (R8_CONTROL)

Offset address: 0x04

Bit	Name	access	describe	Reset value
7	BMC_BYTE_HI	RO	Indicates the nibble status during the current PD data transmission and reception: 1: Indicates that the high 4 bits are being processed; 0: Processing the lower 4 bits.	0
6	TX_BIT_BACK	RO	Indicates the bit state when the BMC transmits the encoding: 1: Indicates that BMC bytes are being sent; 0: Idle.	0
5	DATA_FLAG	RO is the valid flag for cached data.		0
[4:2] RX_STATE[2:0]		RO	PD Receive Status Identifier 000: Initial Receive State 001: Start Receive (SOP) 010: Receive Reset 011: Receive (SOP) 100: Receive End 101: Receive Not Used 110: Receive EOP 111: Receive byte.	0
1	BMC_START	RW BMC sends a start signal.		0
0	PD_TX_EN	RW	USBPD transmit/receive modes and transmit enable: 1: PD transmission enabled; 0: PD receiver enabled.	0

24.2.7 PD Send SOP Selection Register (R8_TX_SEL) Offset Address:

0x05

Bit	name	access	describe	Reset value
[7:6] TX_SEL4[1:0]		RW	K-CODE4 type selection in PD transmission mode: 00: SYNC2; 01: SYNC3; 1x: RST2.	0
[5:4] TX_SEL3[1:0]		RW	K-CODE3 type selection in PD transmission mode: 00: SYNC1; 01: SYNC3; 1x: RST1.	0
[3:2] TX_SEL2[1:0]		RW	K-CODE2 type selection in PD transmission mode: 00: SYNC1;	0

			01: SYNC3; 1x: RST1.	
1	Reserved	RO is reserved.		0
0	TX_SEL1	RW	K-CODE1 type selection in PD transmission mode: 1: RST1; 0: SYNC1.	0

24.2.8 PD Transmit Length Register (R16_BMC_TX_SZ) Offset Address:

0x06 Name

Bit		access	describe	Reset value
[15:9] Reserved		RO is reserved.		0
[8:0] BMC_TX_SZ		The total length sent in RW PD mode.		0

24.2.9 PD Status Register (R32_USBPD_STATUS) Offset Address: 0x08

Bit	name
[31:16] R16_BMC_BYTE_CNT [15:0]	
	R16_STATUS

24.2.10 PD Interrupt and Data Register (R16_STATUS) Offset Address:

0x08

Bit	name
[15:8]	R8_STATUS
[7:0]	R8_DATA_BUF

24.2.11 DMA Cache Data Register (R8_DATA_BUF) Offset Address: 0x08

Name [7:0] DATA_BUF

Bit		access	describe	Reset value
		RO DMA cache data.		X

24.2.12 PD Interrupt Flag Register (R8_STATUS) Offset Address:

0x09 Name

Bit		access	describe	Reset value
7	IF_TX_END	RW1Z transmission completion interruption flag: write 1 to clear 0, write 0 to invalidate.		0
6	IF_RX_RESET	The RW1Z receives the reset interrupt flag; writing 1 clears it to 0, and writing 0 invalidates it.		0
5	IF_RX_ACT	RW1Z receive complete interrupt flag: write 1 to clear 0, write 0 to invalidate.		0
4	IF_RX_BYTE	RW1Z	Receive byte or SOP interrupt flag; write 1 to clear 0, write 0 to clear 0. invalid.	0
3	IF_RX_BIT	RW1Z	Receive either a 1-bit or 5-bit interrupt flag; write 1 to clear 0. 0 is invalid.	0
2	BUF_ERR	RW1Z	BUFFER or DMA error interrupt flag: write 1 to clear 0. Writing 0 is invalid.	0
[1:0] BMC_AUX[1:0]		RO indicates the current PD status:		0

			During or after PD reception, the status is as follows: 00: The receiver is idle or no valid data packets have been received; 01: SOP received, i.e., SOP0; 10: Received SOP', i.e., SOP1 or Hard Reset; 11: Received SOP, i.e., SOP2 or Cable Reset. When PD sends, the status is as follows: 00: Sending CRC32[7:0]; 01: Sending CRC32[15:8]; 10: Sending CRC32 [23:16]; 11: Sending CRC32[31:24].	

24.2.13 Byte counter (R16_BMC_BYTE_CNT) offset address:

0x0A Name

Bit		access	describe	Reset value
[15:9] Reserved		RO is reserved.		0
[8:0] BMC_BYTE_CNT		RO byte counter.		0

24.2.14 Port Control Register (R32_USBPD_PORT) Offset Address:

0x0C

Bit	name
[31:16] R16_PORT_CC2	
[15:0]	R16_PORT_CC1

24.2.15 CC1 Port Control Register (R16_PORT_CC1) Offset Address:

0x0C

Bit	name	access	describe	Reset value
[15:8] Reserved		RO is reserved.		0
[7:5] CC1_CE		RW	Enable the CC1 port voltage comparator: 000: Off; 001: Reserved; 010: 0.22V; 011: 0.45V; 100: 0.55V; 101: 0.66V; 110: 0.95V; 111: 1.23V.	0
4	CC1_LVE	RW	CC1 port output low voltage enable: 1: Low voltage drive output; 0: Normal VDD33 voltage drive output.	0
[3:2] CC1_PU[1:0]		RW	CC1 port pull-up current selection: 00: Pull-up current is prohibited; 01: 330uA; 10: 180uA; 11:80uA.	0

1	CC1_PD	RW	Enable the pull-down resistor Rd at port CC1: 1: Enable the Rd pull-down resistor, approximately 5.1Kÿ; 0: Turn off the pull-down resistor. Note: There is still a weak pull-down of about 600kÿ after it is turned off.	1
0	PA_CC1_AI	RO	CC1 port comparator analog input.	1

24.2.16 CC2 Port Control Register (R16_PORT_CC2)

Offset address: 0x0E

Bit name	[15:8] Reserved	access	describe	Reset value
		RO	is reserved.	0
[7:5] CC2_CE		RW	Enable the CC2 port voltage comparator: 000: Off; 001: Reserved; 010: 0.22V; 011: 0.45V; 100: 0.55V; 101: 0.66V; 110: 0.95V; 111: 1.23V.	0
4	CC2_LVE	RW	CC2 port output low voltage enable: 1: Low voltage drive output; 0: Normal VDD33 voltage drive output.	0
[3:2] CC2_PU		RW	CC2 port pull-up current selection: 00: Pull-up current is prohibited; 01: 330uA; 10: 180uA; 11:80uA.	0
1	CC2_PD	RW	Enable the pull-down resistor Rd at port CC2: 1: Enable the Rd pull-down resistor, approximately 5.1Kÿ; 0: Turn off the pull-down resistor. Note: There is still a weak pull-down of about 600kÿ after it is turned off.	1
0	PA_CC2_AI	RO	CC2 port comparator analog input.	1

24.2.17 DMA Cache Address Register (R32_USBPD_DMA) Offset Address:

0x10 Name

Bit		access	describe	Reset value
[31:21] Reserved		RO	is reserved.	0
[20:0] USBPD_DMA_ADDR		RW	USBPD_DMA cache address. The lower 21 bits are valid, and the address must be 4-byte aligned.	X

Chapter 25 USB High-Speed Host/Device Controller (USBHS)

The USB 2.0 high-speed controller functions as both a host controller and a device controller, with a built-in 480Mbps USB-PHY physical layer. Transmitter. When used as a host controller, it supports low-speed, full-speed, and high-speed USB devices. When used as a device controller, it offers flexible... It can be set to low speed, full speed, or high speed mode to suit various applications.

25.1 USB Function Description

Supports USB 2.1, USB 2.0, USB 1.1, and USB 1.0 protocol specifications .

Supports both USB Host and USB Device functions .

Supports control transfer, batch transfer, interrupted transfer, and real-time/synchronous transfer.

Provides bus reset, suspend, wake-up, and resume functions .

• The host supports a USB hub . •

Non-zero endpoints support a maximum data packet size of 1024 bytes, with a built-in FIFO, interrupt support,

and DMA support . • Supports USART serial port or I2C pin mapping, also suitable for two-wire debugging.

25.2 Register Description

The USB-related registers are divided into two parts:

USB High-Speed Device Control Register

USB High-Speed Host Control Register

25.2.1 Device Register Description

Table 25-1 List of Device-Related Registers

name	Access Address Description: 0x40030000 USBHS	Reset value
R8_USB_CTRL	Control Register; 0x40030001 USBHS Mode	0x07
R8_USB_BASE_MODE	Control Register; 0x40030002 USBHS Interrupt	0x00
R8_USB_INT_EN	Enable Register; 0x40030003 USBHS Device Address	0x00
R8_USB_DEV_AD	Register; 0x40030004 USBHS Remote Wake-up	0x00
R8_USB_WAKE_CTRL	Register; 0x40030005 USBHS Test Mode Register;	0x00
R8_USB_TEST_MODE	0x40030006 USBHS Power Management Register;	0x00
R16_USB_LPM_DATA	0x40030008 USBHS Interrupt Flag Register;	0x8000
R8_USB_INT_FG	0x40030009 USBHS Interrupt Status Register;	0x00
R8_USB_INT_ST	0x4003000A USBHS Miscellaneous Status Register;	0x00
R8_USB_MIS_ST	0x4003000C USBHS Frame Number Register;	0x00
R16_USB_FRAME_NO	0x4003000E USBHS Bus Status Register;	0x0000
R16_USB_BUS	0x40030010 USBHS Endpoint Transmit Enable	0x0000
R16_UEP_TX_EN	Register; 0x40030012 USBHS Endpoint Receive Enable	0x0000
R16_UEP_RX_EN	Register.	0x0000
R16_UEP_T_TOG_AUTO	0x40030014 USBHS Endpoint Transmit Auto-Toggle Enable Register	0x0000
R16_UEP_R_TOG_AUTO	0x40030016 USBHS Endpoint Receive Auto-Toggle Enable Register	0x0000
R8_UEP_T_BURST	0x40030018 USBHS Endpoint Transmit Burst Register	0x00
R8_UEP_T_BURST_MODE	0x40030019 USBHS Endpoint Transmit Burst Mode Register 0x4003001A	0x00
R8_UEP_R_BURST	USBHS Endpoint Receive Burst Register	0x00

R8_UEP_R_RES_MODE	0x4003001B USBHS Endpoint Receive Reply Mode	0x00
R32_UEP_AF_MODE	Register; 0x4003001C USBHS Endpoint	0x00000000
R32_UEP0_DMA	Multiplexing Register; 0x40030020 Endpoint 0 Buffer	0x000XXXXX
R32_UEP1_RX_DMA	Start Address Register; 0x40030024 Endpoint 1 Receive Buffer Start Address	
R32_UEP2_RX_DMA	Register; 0x000XXXXX 0x40030028 Endpoint 2 Receive Buffer Start Address Register;	
R32_UEP3_RX_DMA	0x000XXXXX 0x4003002C Endpoint 3 Receive Buffer Start Address Register;	
R32_UEP4_RX_DMA	0x000XXXXX 0x40030030 Endpoint 4 Receive Buffer Start Address Register;	
R32_UEP5_RX_DMA	0x000XXXXX 0x40030034 Endpoint 5 Receive Buffer Start Address Register;	
R32_UEP6_RX_DMA	0x000XXXXX 0x40030038 Endpoint 6 Receive Buffer Start Address Register;	
R32_UEP7_RX_DMA	0x000XXXXX 0x4003003C Endpoint 7 Receive buffer start address register	
R32_UEP1_TX_DMA	0x000XXXXX 0x40030040 Endpoint 1 Transmit buffer start address register	
R32_UEP2_TX_DMA	0x000XXXXX 0x40030044 Endpoint 2 Transmit buffer start address register	
R32_UEP3_TX_DMA	0x000XXXXX 0x40030048 Endpoint 3 Transmit buffer start address register	
R32_UEP4_TX_DMA	0x000XXXXX 0x4003004C Endpoint 4 Transmit buffer start address register	
R32_UEP5_TX_DMA	0x000XXXXX 0x40030050 Endpoint 5 Transmit buffer start address register	
R32_UEP6_TX_DMA	0x000XXXXX 0x40030054 Endpoint 6 Transmit buffer start address register	
R32_UEP7_TX_DMA	0x000XXXXX 0x40030058 Endpoint 7 Transmit buffer start address register	
R32_UEP0_MAX_LEN	0x000XXXXX 0x4003005C Endpoint 0 Maximum	0x000000XX
R32_UEP1_MAX_LEN	length packet register 0x40030060 Endpoint 1	0x00000XXX
R32_UEP2_MAX_LEN	Maximum Length Packet Register 0x40030064	0x00000XXX
R32_UEP3_MAX_LEN	Endpoint 2 Maximum Length Packet Register	0x00000XXX
R32_UEP4_MAX_LEN	0x40030068 Endpoint 3 Maximum Length Packet	0x00000XXX
R32_UEP5_MAX_LEN	Register 0x4003006C Endpoint 4 Maximum	0x00000XXX
R32_UEP6_MAX_LEN	Length Packet Register 0x40030070 Endpoint 5	0x00000XXX
R32_UEP7_MAX_LEN	Maximum Length Packet Register 0x40030074	0x00000XXX
R16_UEP0_RX_LEN	Endpoint 6 Maximum Length Packet Register	0x00XX
R16_UEP1_RX_LEN	0x40030078 Endpoint 7 Maximum Length Packet	0xXXXX
R16_UEP1_R_SIZE	Register 0x4003007C Endpoint 0 Receive Length	0xXXXX
R16_UEP2_RX_LEN	Register 0x40030080 Endpoint 1 Single Receive	0xXXXX
R16_UEP2_R_SIZE	Length Register 0x40030082 Endpoint 1 Total Received	0xXXXX
R16_UEP3_RX_LEN	Data Length Register 0x40030084 Endpoint 2 Single	0xXXXX
R16_UEP3_R_SIZE	Receive Length Register 0x40030086 Endpoint 2 Total	0xXXXX
R16_UEP4_RX_LEN	Received Data Length Register 0x40030088 Endpoint	0xXXXX
R16_UEP4_R_SIZE	3 Single Receive Length Register 0x4003008A Endpoint	0xXXXX
R16_UEP5_RX_LEN	3 Total Received Data Length Register 0x4003008C	0xXXXX
R16_UEP5_R_SIZE	Endpoint 4 Single Received Length Register	0xXXXX
R16_UEP6_RX_LEN	0x4003008E Endpoint 4 Total Received Data Length	0xXXXX
R16_UEP6_R_SIZE	Register 0x40030090 Endpoint 5 Single Received	0xXXXX
R16_UEP7_RX_LEN	Length Register 0x40030092 Endpoint 5 Total	0xXXXX
R16_UEP7_R_SIZE	Received Data Length Register 0x40030094 Endpoint	0xXXXX
R16_UEP0_T_LEN	6 Single Received Length Register 0x40030096	0x00XX
R8_UEP0_TX_CTRL	Endpoint 6 Total Received Data Length	0x00
R8_UEP0_RX_CTRL	Register 0x40030098 Endpoint 7 Single	0x00
R16_UEP1_T_LEN	Received Length Register 0x4003009A Endpoint 7 Total Received Data Length Register 0x4003009C Endpoi	0x0000

R8_UEP1_TX_CTRL	0x400300A2 Endpoint 1 Transmit Control	0x00
R8_UEP1_RX_CTRL	Register 0x400300A3 Endpoint 1 Receive Control	0x00
R16_UEP2_T_LEN	Register 0x400300A4 Endpoint 2 Transmit	0x0000
R8_UEP2_TX_CTRL	Length Register 0x400300A6 Endpoint 2 Transmit	0x00
R8_UEP2_RX_CTRL	Control Register 0x400300A7 Endpoint 2 Receive	0x00
R16_UEP3_T_LEN	Control Register 0x400300A8 Endpoint 3	0x0000
R8_UEP3_TX_CTRL	Transmit Length Register 0x400300AA Endpoint	0x00
R8_UEP3_RX_CTRL	3 Transmit Control Register 0x400300AB	0x00
R16_UEP4_T_LEN	Endpoint 3 Receive Control Register 0x400300AC	0x0000
R8_UEP4_TX_CTRL	Endpoint 4 Transmit Length Register 0x400300AE	0x00
R8_UEP4_RX_CTRL	Endpoint 4 Transmit Control Register 0x400300AF	0x00
R16_UEP5_T_LEN	Endpoint 4 Receive Control Register 0x400300B0	0x0000
R8_UEP5_TX_CTRL	Endpoint 5 Transmit Length Register 0x400300B2	0x00
R8_UEP5_RX_CTRL	Endpoint 5 Transmit Control Register 0x400300B3	0x00
R16_UEP6_T_LEN	Endpoint 5 Receive Control Register 0x400300B4	0x0000
R8_UEP6_TX_CTRL	Endpoint 6 Transmit Length Register 0x400300B6	0x00
R8_UEP6_RX_CTRL	Endpoint 6 Transmit Control Register 0x400300B7	0x00
R16_UEP7_T_LEN	Endpoint 6 Receive Control Register 0x400300B8	0x0000
R8_UEP7_TX_CTRL	Endpoint 7 Transmit Length Register 0x400300BA	0x00
R8_UEP7_RX_CTRL	Endpoint 7 Transmit Control Register 0x400300BB	0x00
R16_UEP_T_ISO	Endpoint 7 Receive Control Register 0x400300BC USBHS	0x0000
R16_UEP_R_ISO	Endpoint Transmit Synchronization Mode Enable Register	0x0000
R32_UEP1_RX_FIFO	0x400300BE USBHS Endpoint Receive Synchronization	0x00000000
R32_UEP2_RX_FIFO	Mode Enable Register 0x400300C0 USBHS Endpoint 1	0x00000000
R32_UEP3_RX_FIFO	Receive FIFO Address 0x400300C4 USBHS Endpoint 2	0x00000000
R32_UEP4_RX_FIFO	Receive FIFO Address 0x400300C8 USBHS Endpoint 3	0x00000000
R32_UEP5_RX_FIFO	Receive FIFO Address 0x400300CC USBHS Endpoint 4	0x00000000
R32_UEP6_RX_FIFO	Receive FIFO Address 0x400300D0 USBHS Endpoint 5	0x00000000
R32_UEP7_RX_FIFO	Receive FIFO Address 0x400300D4 USBHS The receive	0x00000000
R32_UEP1_TX_FIFO	FIFO address of endpoint 6 is 0x400300D8 USBHS; the	0x00000000
R32_UEP2_TX_FIFO	receive FIFO address of endpoint 7 is 0x400300DC USBHS;	0x00000000
R32_UEP3_TX_FIFO	the receive FIFO address of endpoint 1 is 0x400300E0	0x00000000
R32_UEP4_TX_FIFO	USBHS; the transmit FIFO address of endpoint 2 is	0x00000000
R32_UEP5_TX_FIFO	0x400300E4 USBHS; the transmit FIFO address of	0x00000000
R32_UEP6_TX_FIFO	endpoint 3 is 0x400300E8 USBHS; the transmit FIFO	0x00000000
R32_UEP7_TX_FIFO	address of endpoint 4 is 0x400300EC USBHS; the transmit FIFO address of endpoint 5 is 0x400300F0 USBHS; the	0x00000000

25.2.1.1 USBHS Control Register (R8_USB_CTRL) Offset

Address: 0x00

Bit	name	access	describe	Reset value
7	RB_UD_LPM_EN	RW	LPM enabled: 1: Enable; 0: Prohibited.	0
6	Reserved	RO is reserved.		0

5	RB_UD_DEV_EN	RW	USB device enable: 1: Enable; 0: Prohibited.	0
4	RB_UD_DMA_EN	RW	DMA transfer enabled: 1: Enable; 0: Prohibited.	0
3	RB_UD_PHY_SUSPENDM	RW	USB PHY suspended: 1: Normal work; 0: Suspended.	0
2	RB_UD_CLR_ALL	RW	Clear all interrupt flags: 1: Clear the USB interrupt flag and FIFO. Software is required. Clear all items; 0: Do not clear.	1
1	RB_UD_RST_SIE	RW	USB protocol processor reset: 1: Force reset the USB protocol processor (SIE). This includes endpoint-related registers, which need to be cleared by software; 0: Do not reset.	1
0	RB_UD_RST_LINK	RW	LINK layer reset: 1: USB Link layer reset; 0: Do not reset.	1

25.2.1.2 USBHS Mode Control Register (R8_USB_BASE_MODE)

Offset address: 0x01

	name	access	describe	Reset value
bits [7:2]	Reserved	RO reserved.		0x0
[1:0]	RB_UD_SPEED_TYPE[1:0]	RW	Desired speed mode for the device: 00: Full-speed equipment; 01: High-speed equipment; 10: Low-speed equipment; 11: Retained.	0

25.2.1.3 USBHS Interrupt Enable Register (R8_USB_INT_EN) Offset Address: Bit

0x02

	name	access	describe	Reset value
7	RB_UDIE_FIFO_OVER	RW	FIFO overflow interrupt. 1: Enable interrupt; 0: Interruption is disabled.	0
6	RB_UDIE_LINK_RDY	RW	USB connection interruption enabled. 1: Enable interrupt; 0: Interruption is disabled.	0
5	RB_UDIE_SOF_ACT	RW	SOF packet reception interrupt enabled. 1: Enable interrupt; 0: Interruption is disabled.	0
4	RB_UDIE_TRANSFER	RW	USB transfer completion interrupt enabled.	0

			1: Enable interrupt; 0: Interruption is disabled.	
3	RB_UDIE_LPM_ACT	RW	LPM transmission completion interrupt enabled. 1: Enable interrupt; 0: Interruption is disabled.	0
2	RB_UDIE_BUS_SLEEP	RW	USB bus sleep interrupt enabled. 1: Enable interrupt; 0: Interruption is disabled.	0
1	RB_UDIE_SUSPEND	RW	USB bus pause interrupt enabled. 1: Enable interrupt; 0: Interruption is disabled.	0
0	RB_UDIE_BUS_RST	RW	USB bus reset interrupt enabled. 1: Enable interrupt; 0: Interruption is disabled.	0

25.2.1.4 USBHS Device Address Register (R8_USB_DEV_AD) Offset Address:

Bit 0x03

	name	access	describe	Reset value
7	Reserved	RO is reserved.		0
[6:0]	RB_UD_DEV_ADDR[6:0]	RW USB device address.		0

25.2.1.5 USBHS Remote Wake-up Register (R8_USB_WAKE_CTRL)

Offset address: 0x04

Bit	name	access	describe	Reset value
[7:1]	Reserved	RO is reserved.		0
0	RB_UD_REMOTE_WKUP	RW1Z remote wake-up and automatic hardware reset.		0

25.2.1.6 USBHS Test Mode Register (R8_USB_TEST_MODE)

Offset address: 0x05

bits	name	access	describe	Reset value
7	RB_UD_TEST_EN	RW test mode is enabled.		0
[6:4]	Reserved	RO is reserved.		0
3	RB_UD_TEST_SE0NAK	RW test mode, outputs SE0. Test		0
2	RB_UD_TEST_PKT	RW	mode, outputs one packet. The packet uses the data address and length of endpoint 4, TOG For DATA0.	0
1	RB_UD_TEST_K	RW test mode, output K.		0
0	RB_UD_TEST_J	RW test mode, output J.		0

25.2.1.7 USBHS Power Management Register (R16_USB_LPM_DATA)

Offset address: 0x06

bits	name	access	describe	Reset value
15	RB_UD_LPM_BUSY	RW power management busy.		1
[14:11] Reserved		RO is reserved.		0

[10:0]	RB_UD_LPM_DATA[10:0]	RO	power management data.	0

25.2.1.8 USB Interrupt Flag Register (R8_USB_INT_FG) Offset Address:

0x08

Bit	name	access	describe	Reset value
7	RB_UDIF_FIFO_OV	RW1Z	The FIFO overflow interrupt flag is cleared by writing 1. 1: FIFO overflow triggered; 0: No event.	0
6	RB_UDIF_LINK_RDY	RW1Z	USB connection interruption flag, write 1 to clear. 1: USB connection event triggered; 0: No event.	0
5	RB_UDIF_RX_SOF	RW1Z	The interrupt flag for receiving SOF packets is cleared by writing 1. 1: The event of receiving an SOF packet is triggered; 0: No event.	0
4	RB_UDIF_RTX_ACT	RO	USB transfer end interruption flag: The received data is cleared by RB_UEP_R_DONE; Sending is cleared by RB_UEP_T_DONE.	0
3	RB_UDIF_LPM_ACT	RW1Z	The LPM transmission end interrupt flag is cleared by writing 1. 1: LPM transmission end event triggered; 0: No event.	0
2	RB_UDIF_BUS_SLEEP	RW1Z	USB bus sleep interrupt flag, when LPM is enabled. When entering the L1 (Sleep) state, it will trigger a write operation (write 1). zero. 1: USB bus sleep event triggered; 0: No event.	0
1	RB_UDIF_SUSPEND	RW1Z	The USB bus suspend interrupt flag is cleared by writing 1. 1: USB suspend event triggered; 0: No event.	0
0	RB_UDIF_BUS_RST	RW1Z	USB bus reset interrupt flag bit, write 1 to clear it. 1: USB bus reset event triggered; 0: No event.	0

25.2.1.9 USB Interrupt Status Register (R8_USB_INT_ST) Offset Address:

0x09

	name	access	describe	Reset value
Position [7:5]	Reserved	RO	is reserved.	0
4	RB_UDIS_EP_DIR	RO	Endpoint data transmission direction: 1: Endpoint IN data; 0: Endpoint OUT/SETUP data.	0
3	Reserved	RO	is reserved.	0
[2:0]	RB_UDIS_EP_ID_MASK[2:0]	RO	is the endpoint number where data transmission occurred.	0

25.2.1.10 USB Miscellaneous Status Register (R8_USB_MIS_ST)

Offset Address:

0x0A bit	name	Access description: Is the RO host a high-	Reset value
7	RB_UDMS_HS_MOD	speed host?	0
[6:5]	Reserved	RO is reserved.	0
4	RB_UDMS_SUSP_REQ	RO USB suspend request.	0
3	RB_UDMS_SIE_FREE	RO USB is in idle state.	0
2	RB_UDMS_SLEEP	RO USB is in sleep mode.	0
1	RB_UDMS_SUSPEND	RO USB is in suspended state. 1: The USB bus is in a suspended state; it has been idle for some time. There is USB activity; 0: The USB bus is in a non-suspended state.	0
0	RB_UDMS_READY	RO USB connection status.	0

25.2.1.11 USB Frame Register (R16_USB_FRAME_NO)

Offset Address: 0x0C

Bit	name	access	describe	Reset value
[15:13]	RB_UD_MFRAME_NO[2:0]	The microframe number received by RO.		0
[12:11] Reserved		RO is reserved.		0
[10:0]	RB_UD_FRAME_NO[10:0]	The frame number received by RO.		0

25.2.1.12 USB Bus Status Register (R16_USB_BUS) Offset

Address: 0x0E

Bit	name	access	describe	Reset value
[15:4] Reserved		RO is reserved.		0
3	RB_USB_DM_ST	RO UDM status.		0
2	RB_USB_DP_ST	RO UDP status.		0
1	Reserved	RO is reserved.		0
0	RB_USB_WAKEUP	RO USB Wake-up (High Active).		0

25.2.1.13 USBHS Endpoint Transmit Enable Register (R16_UEP_TX_EN)

Offset address: 0x10

Bit	name	access		Reset value
[15:0]	RB_UEP_TX_EN[15:0]	RW	Description 0~15 Endpoint Send Enable: 1: Enable; 0: Prohibited.	0

25.2.1.14 USBHS Endpoint Receive Enable Register (R16_UEP_RX_EN)

Offset address: 0x12

bits	name	Access Description	Reset value
[15:0]	RB_UEP_RX_EN[15:0]	RW 0~15 Endpoint receive enable:	0

			1: Enable; 0: Prohibited.	

25.2.1.15 USBHS Endpoint Transmit Auto-Toggle Enable Register (R16_UEP_T_TOG_AUTO)

Offset address: 0x14

Bit	name	access	describe	Reset value
[15:8] Reserved		RO is reserved.		0
[7:0]	RB_UEP_T_TOG_AUTO[7:0]	RW	Automatic toggling of endpoint synchronization trigger bits (0-7) is enabled. 1: Automatically flips after data is successfully sent; 0: Manual control of the flipping. Note: The endpoint 0 only supports manual control of the flip.	0

25.2.1.16 USBHS Endpoint Receive Auto-Toggle Enable Register (R16_UEP_R_TOG_AUTO)

Offset address: 0x16

bits	name	access	describe	Reset value
[15:8] Reserved		RO retains the		0
[7:0]	RB_UEP_R_TOG_AUTO[7:0]	RW	auto-flip enable for the 0~7 endpoint synchronization trigger bits: 1: Automatically flips after successful data reception; 0: Manual control of the flipping. Note: The endpoint 0 only supports manual control of the flip.	0

25.2.1.17 USBHS Endpoint Transmit Burst Register (R8_UEP_T_BURST) Offset

Address: 0x18

Bit	name	access	describe	Reset value
[7:0]	RB_UEP_T_BURST_EN[7:0]	RW	Endpoints 0-7 burst send enable: 1: Enable; 0: Prohibited.	0

25.2.1.18 USBHS Endpoint Transmit Mode Register (R8_UEP_T_BURST_MODE) Offset

Address: Bit 0x19

	name	access		Reset value
[7:0]	RB_UEP_T_BURST_MODE [7:0]	RW	Description of endpoint burst sending mode 0~7: 1: In burst mode, zero-length data is not sent; 0: Sends 0-length data in burst mode.	0

25.2.1.19 USBHS Endpoint Receive Burst Register (R8_UEP_R_BURST) Offset

Address: Bit 0x1A

	name	access		Reset value
[7:0]	RB_UEP_R_BURST_EN[7:0]	RW	Description of endpoints 0-7 burst receive enable: 1: Enable; 0: Prohibited.	0

25.2.1.20 USBHS Endpoint Reply Mode Register (R8_UEP_R_RES_MODE) Offset

Address: 0x1B

Bit	name	access	describe	Reset value
[7:0]	RB_UEP_R_RES_MODE[7 :0]	RW	0~7 Endpoint receive return mode: 1: NYET; 0: ACK.	0

25.2.1.21 USBHS Endpoint Multiplexing Register (R32_UEP_AF_MODE)

Offset address: 0x1C

bit	name	access	describe	Reset value
[31:8] Reserved		RO reserved.		0
[7:1]	RB_UEP_T_AF[6:0]	RW	Endpoint multiplexing enable (1-7): 1: Reuse 9-15; 0: Reuse 1-7.	0
0	Reserved	RO is reserved.		0

25.2.1.22 Offset address of the start address register (R32_UEP0_DMA) for endpoint 0

buffer: 0x20 bits

	name	access	describe	Reset value
[31:24] Reserved		RO is reserved.		0
[23:0] UEP0_DMA[23:0]		The starting address of the buffer at RW endpoint 0.		X

25.2.1.23 Start address register of endpoint n receive buffer (R32_UEPn_RX_DMA) (n=1ÿ7)

Offset address: 0x24 + 4*(n-1)

Bit name		access	describe	Reset value
[31:24] Reserved		RO (Reserved). The		0
[23:0] UEPn_RX_DMA[23:0]		RW	starting address of the receive buffer for endpoint n. This address is required. Requires 4-byte alignment.	X

25.2.1.24 Start address register of endpoint n transmit buffer (R32_UEPn_TX_DMA) (n=1ÿ7)

Offset address: 0x40 + 4*(n-1)

Bit	name	access	describe	Reset value
[31:24] Reserved		RO (Reserved). The		0
[23:0] UEPn_TX_DMA[23:0]		RW	starting address of the send buffer for endpoint n. This address is required. Requires 4-byte alignment.	X

25.2.1.25 Endpoint n Maximum Length Packet Register (R32_UEPn_MAX_LEN) (n=0) Offset

Address: 0x5C bit

	name	access	describe	Reset value
[31:7] Reserved		RO is reserved.		0
[6:0]	UEPn_MAX_LEN[6:0]	RW endpoint n DMA maximum offset limit length.		X

25.2.1.26 Endpoint n Maximum Length Packet Register (R32_UEPn_MAX_LEN) (n=1ÿ7) Offset address:

0x60 + 4 (n-1)

Bit	name	access	describe	Reset value

[31:11] Reserved		RO is reserved.	0
[10:0] UEPn_MAX_LEN[10:0]		RW endpoint n DMA maximum offset limit length.	X

25.2.1.27 Endpoint 0 Receive Length Register (R16_UEP0_RX_LEN)

Offset address: 0x7C

bit	name	access	describe	Reset value
[15:7] Reserved		RO is reserved.		0
[6:0]	UEP0_RX_LEN[6:0]	RW endpoint 0 receives data length.		X

25.2.1.28 Endpoint n Single Receive Length Register (R16_UEPn_RX_LEN) (n=1ÿ7) Offset address: 0x80 + 4*(n-1)

Bit	name	Access description RW endpoint n Single	Reset value
[15:0] UEPn_RX_LEN[15:0]		data reception length.	X

25.2.1.29 Endpoint n Total Received Data Length Register (R16_UEPn_R_SIZE) (n=1ÿ7) Offset address: 0x82 + 4*(n-1)

Bit	name	access	describe	Reset value
[15:0] UEPn_R_SIZE[15:0]		RW endpoint n: Total length of received data.		X

25.2.1.30 Endpoint 0 Transmit Length Register (R16_UEP0_T_LEN)

Offset address: 0x9C

Bit	name	access	describe	Reset value
[15:7] Reserved		RO is reserved.		0
[6:0]	UEP0_T_LEN[6:0]	RW endpoint 0 transmits data length.		X

25.2.1.31 Endpoint 0 Transmit Control Register (R8_UEP0_TX_CTRL)

Offset address: 0x9E

bit	name	access	describe	Reset value
7	RB_UEP_T_DONE	RW0 endpoint 0 is the end-of-transmission flag; write 0 clears it.		0
6	RB_UEP_T_NAK_ACT	RW0 endpoint 0 sends NAK end flag, write 0 to clear. 0 RO reserved.		
[5:3]	Reserved			0
2	RB_UEP_T_TOG_MASK	RW	The expected synchronization trigger bit for endpoint 0: 1: DATA1; 0: DATA0.	0
[1:0]	RB_UEP_T_RES_MASK[1:0]	RW	Endpoint 0 controls the sent response: 00: Responds to NAK or Busy; 01: Response STALL or error; 10: ACK response; 11: Retained.	0

25.2.1.32 Endpoint 0 Receive Control Register (R8_UEP0_RX_CTRL)

Offset address: 0x9F

bit	name	access	describe	Reset value

7	RB_UEP_R_DONE	RW0	endpoint 0 receive end flag, write 0 clears.	0
6	RB_UEP_R_NAK_ACT	RW0	endpoint 0 receives NAK end flag; writing 0 clears it. 0 endpoint 0,	
5	RB_UEP_R_NAK_TOG	RO	for received NAK return data packets... type: 1: DATA1; 0: DATA0.	0
4	RB_UEP_R_TOG_MATCH	RO	The received synchronization trigger bit and the expected synchronization trigger Bit matching status: 1: Synchronization; 0: out of sync.	0
3	RB_UEP_R_SETUP_IS	Whether the data received by endpoint 0 (RO) is a SETUP		0
2	RB_UEP_R_TOG_MASK	RW	transaction. The expected synchronization trigger bit for endpoint 0: 1: DATA1; 0: DATA0.	0
[1:0]	RB_UEP_R_RES_MASK[1 :0]	RW	Endpoint 0's response control for received data: 00: Responds to NAK or Busy; 01: Response STALL or error; 10: ACK response; 11: Retained.	0

25.2.1.33 Endpoint n Transmit Length Register (R16_UEPn_T_LEN) (n=1ÿ7)

Offset address: 0xA0 + 4*(n-1)

Bit	name	Access	description	Reset value
[15:0]	UEPn_T_LEN[15:0]	RW	endpoint n data length sent.	0

25.2.1.34 Endpoint n Transmit Control Register (R8_UEPn_TX_CTRL) (n=1ÿ7) Offset address:

0xA2 + 4*(n-1)

Bit	name	access	describe	Reset value
7	RB_UEP_T_DONE	RW0	endpoint n is the end-of-transmission flag; writing 0 clears it.	0
6	RB_UEP_T_NAK_ACT	RW0	endpoint n sends NAK end flag, write 0 to clear. 0 RO reserved.	
[5:4]	Reserved			0
[3:2]	RB_UEP_T_TOG_MASK[1 :0]	RW	The expected synchronization trigger bit for endpoint n: 00: DATA0; 01: DATA1; 10: DATA2; 11: MDATA. Note: This setting is configurable in manual mode but not in automatic mode.	0
[1:0]	RB_UEP_T_RES_MASK[1 :0]	RW	Endpoint n controls the response sent: 00: Responds to NAK or Busy; 01: Response STALL or error; 10: ACK response; 11: Retained.	0

25.2.1.35 Endpoint n Receive Control Register (R8_UEPn_RX_CTRL) (n=1ÿ7)

Offset address: 0xA3 + 4*(n-1)

Bit	name	access	describe	Reset value
7	RB_UEP_R_DONE	RW0	endpoint n is the end-of-reception flag; writing 0 clears it.	0
6	RB_UEP_R_NAK_ACT	RW0	endpoint n receives the NAK end flag; writing 0 clears it. 0 endpoint	
5	RB_UEP_R_NAK_TOG	RO	n receives a NAK response from the data packet. type: 1: DATA1; 0: DATA0.	0
4	RB_UEP_R_TOG_MATCH	RO	The synchronization trigger bit received by endpoint n matches the expected synchronization trigger bit. Step trigger bit matching status: 1: Synchronization; 0: out of sync.	0
[3:2]	RB_UEP_R_TOG_MASK[1 :0]	RW	The expected synchronization trigger bit for endpoint n: 00: DATA0; 01: DATA1; 10: DATA2; 11: MDATA. Note: This setting is configurable in manual mode but not in automatic mode.	0
[1:0]	RB_UEP_R_RES_MASK[1 :0]	RW	Endpoint n controls the received response: 00: Responds to NAK or Busy; 01: Response STALL or error; 10: ACK response; 11: Retained.	0

25.2.1.36 USBHS Endpoint Transmit Synchronization Mode Enable Register

(R16_UEP_T_ISO) Offset Address: 0xBC

Bit	name	access	describe	Reset value
[15:9]	RB_UEP_T_FIFO_EN[6: 0]		Enable FIFO mode for TX at the RO endpoint.	0
8	Reserved		RO is reserved.	0
[7:1]	RB_UEPn_T_ISO_EN[6: 0]		RW upload endpoint (IN) synchronization mode enabled.	0
0	Reserved		RO is reserved.	0

25.2.1.37 USBHS Endpoint Receive Synchronization Mode Enable Register

(R16_UEP_R_ISO) Offset Address: 0xBE

Bit	name	access	describe	Reset value
[15:9]	RB_UEP_R_FIFO_EN[6: 0]		Enable FIFO mode for RX at the RO endpoint.	0
8	Reserved		RO is reserved.	0
[7:1]	RB_UEPn_R_ISO_EN[6 :0]		Enable synchronous mode for RW downlink endpoint (OUT).	0

0	Reserved	RO is reserved.	0

25.2.1.38 **The offset address of the receive FIFO address (R32_UEPn_RX_FIFO) of the USBHS endpoint n (n=1ÿ**

7) : 0xC0 + 4*(n-1)

Bit	Name	Access description: The end address of the FIFO	Reset value
[31:16]	RB_UEP_RX_FIFO_E	received by the RW endpoint.	0
[15:0]	RB_UEP_RX_FIFO_S	The starting address of the FIFO received by the RW endpoint.	0

25.2.1.39 Transmit FIFO address of USBHS endpoint n (R32_UEPn_TX_FIFO) (n=1ÿ7)

Offset address: 0xDC + 4*(n-1)

Bit	name	access	describe	Reset value
[31:16]	RB_UEP_TX_FIFO_E		The FIFO end address sent by the RW endpoint.	0
[15:0]	RB_UEP_TX_FIFO_S		The starting address of the FIFO sent by the RW endpoint.	0

25.2.2 Host Register Description

Table 25-2 List of Host-Related Registers

name	Access address description		Reset value
R8_UH_CFG	0x40030100	USBHS Host Configuration Register	0x07
R8_UH_INT_EN	0x40030102	USBHS Host Interrupt Enable Register	0x00
R8_UH_DEV_AD	0x40030103	USBHS Host Device Address Register	0x00
R32_UH_CONTROL	0x40030104	USBHS Host Control Register	0x00XXXXXX
R8_UH_INT_FLAG	0x40030108	USBHS Host Interrupt Flag Register	0x00
R8_UH_INT_ST	0x40030109	USBHS Host Interrupt Status Register	0xXX
R8_UH_MIS_ST	0x4003010A	USBHS Host Miscellaneous Status Register	0xXX
R32_UH_LPM_DATA	0x4003010C	USBHS Host Power Management Data Register	0x00000XXX
R32_UH_SPLIT_DATA	0x40030110	USBHS Host Split Data Register	0x000XXXXX
R32_UH_FRAME	0x40030114	USBHS Host Frame Register	0x00000000
R32_UH_TX_LEN	0x40030118	USBHS Host Transmit Length Register	0x00000XXX
R32_UH_RX_LEN	0x4003011C	USBHS Host Receive Length Register	0x00000XXX
R32_UH_RX_MAX_LEN	0x40030120	USBHS Host Receive Maximum Length Register	0x00000XXX
R32_UH_RX_DMA	0x40030124	DMA Receive Address Register	0x000XXXXX
R32_UH_TX_DMA	0x40030128	DMA transmit address register	0x000XXXXX
R32_UH_PORT_CTRL	0x4003012C	USBHS Host Port Control Register	0x0000X000
R8_UH_PORT_CFG	0x40030130	USBHS Host Port Configuration Register	0x00
R8_UH_PORT_INT_EN	0x40030132	USBHS Host Port Interrupt Enable Register	0x00
R8_UH_PORT_TEST_CT	0x40030133	USBHS Host Port Test Mode Register	0x00
R16_UH_PORT_ST	0x40030134	USBHS Host Port Status Register	0x0010
R8_UH_PORT_CHG	0x40030136	USBHS Host Port Status Change Register	0x00
R32_UH_BC_CTRL	0x4003013C	USBHS Host BC Charging Control Register	0x00000000

25.2.2.1 USBHS Host Configuration Register (R8_UH_CFG) Offset

Address: Bit 0x00

	name	access	describe	Reset value

7	RB_UH_LPM_EN	Enable RW LPM.	0
6	RB_UH_FORCE_FS	RW forces the use of USB_FS.	0
5	RB_UH_SOF_EN	RW enables SOF packet transmission.	0
4	RB_UH_DMA_EN	RW enables DMA transmission.	0
3	RB_UH_PHY_SUSPENDM	RW PHY suspended.	0
2	RB_UH_CLR_ALL	RW clears the interrupt flag and FIFO, which requires software reset.	1
1	RB_UH_RST_SIE	The RW USB protocol processor needs to be reset, requiring software clearing.	1
0	RB_UH_RST_LINK	RW USB connection control module reset.	1

25.2.2.2 USBHS Host Interrupt Enable Register (R8_UH_INT_EN)

Offset address: 0x02 bits

	name	Access description: RW FIFO overflow	Reset value
7	RB_UHIE_FIFO_OVER	interrupt enabled.	0
6	RB_UHIE_TX_HALT	RW transmit pause interrupt enable.	0
5	RB_UHIE_SOF_ACT	RW SOF packet transmission interrupt enabled.	0
4	RB_UHIE_TRANSFER	RW USB End or Transfer Complete Interruption Enable.	0
3	RB_UHIE_RESUME_ACT	RW bus recovery interrupt enable.	0
2	RB_UHIE_WKUP_ACT	RW wake-up interrupt enable.	0
[1:0]	Reserved	RO is reserved.	0

25.2.2.3 USBHS Host Device Address Register (R8_UH_DEV_AD)

Offset address: 0x03 bits

	name	access	describe	Reset value
7	Reserved	RO is reserved.		0
[6:0]	RB_UH_DEV_ADDR[6:0]	RW is the address of the currently active USB device.		0

25.2.2.4 USBHS Host Control Register (R32_UH_CONTROL) Offset Address: 0x04

Bit	name	access	describe	Reset value
[31:24] Reserved		RO is reserved.		0
twenty three	RB_UH_RX_NO_RES	RW	IN-DATA No acknowledgment, used for synchronous transmission or high-speed transmission. SPLIT package.	X
twenty two	RB_UH_TX_NO_RES	RW	OUT/SETUP–DATA does not expect a response and is used for synchronization. Transmit or high-speed Split packets.	X
twenty one	RB_UH_RX_NO_DATA	RW	No data packets are expected after the IN token packet, used for high speed. SPLIT package.	X
20	RB_UH_TX_NO_DATA	RW	No data packet follows the OUT/SETUP token packet; this is used for high-speed applications. SPLIT package.	X
19	RB_UH_PRE_PID_EN	RW	The low-speed preamble packet PRE PID enable bit, when the port is in operation. When operating at full speed, sending low-speed packets (PRE) requires enabling [something]. Enter this position. 1: Enable, used for connecting to a low-speed external full-speed hub. USB device communication; 0: Disable low-speed preamble.	X

18	RB_UH_SPLIT_VALID	RW	RW sending of SPLIT packets is valid.	X
17	RB_UH_LPM_VALID	RW	RW sends LPM packets.	X
16	RB_UH_HOST_ACTION	RW	The host enables transactions; this bit is reset after the transaction is completed. Dynamically clear to zero.	0
[15:11] Reserved		RO	RO is reserved.	0
10	RB_UH_BUF_MODE	RW	Data buffer control bit. 1: Use R32_UH_RX_DMA for transmission, and use it for reception. Use R32_UH_TX_DMA; 0: Use R32_UH_TX_DMA for transmission, and use it for reception. Use R32_UH_RX_DMA.	X
[9:8]	RB_UH_T_TOG_MASK [1:0]	RW	Data PID sent: 00: PID_DATA0; 01: PID_DATA1; 10: PID_DATA2; 11: PID_MDATA.	X
[7:4]	RB_UH_T_ENDP_MASK [3:0]	RW	The endpoint number of the transaction token packet sent by the RW host.	X
[3:0]	RB_UH_T_TOKEN_MASK [3:0]	RW	The transaction token packet PID sent by the RW host.	X

25.2.2.5 USBHS Host Interrupt Flag Register (R8_UH_INT_FLAG)

Offset address: 0x08

bits	name	access	describe	Reset value
7	RB_UHIF_FIFO_OVER	RW1Z	FIFO overflow interrupt flag.	0
6	RB_UHIF_TX_HALT	RW1Z	Transmission abort interrupt flag, used in master mode for subsequent interrupts. Data overflow during transmission, or data sent at EOF2 time point. Not finished, stop sending, write 1 to clear to zero.	0
5	RB_UHIF_SOF_ACT	RW1Z	SOF packet transmission completion interrupt flag; write 1 to clear.	0
4	RB_UHIF_TRANSFER	RW1Z	USB transaction transfer completion interrupt flag; write 1 to clear.	0
3	RB_UHIF_RESUME_ACT_IF	RW1Z	bus recovery interrupt flag, write 1 to clear.	0
2	RB_UHIF_WKUP_ACT	RW1Z	wakes up the interrupt flag; writing 1 clears it.	0
[1:0]	Reserved	RO	RO is reserved.	0

25.2.2.6 USBHS Host Interrupt Status Register (R8_UH_INT_ST)

Offset address: 0x09

bits	name	access	describe	Reset value
[7:5]	Reserved	RO	RO is reserved.	X
4	RB_UHIS_PORT_RX_RESUME	RW	A value of 1 in the RW bit indicates that the port has received a wake-up signal.	X
[3:0]	RB_UH_R_TOKEN_MASK [3:0]	RO	The PID received by RO.	X

25.2.2.7 USBHS Host Miscellaneous Status Register (R8_UH_MIS_ST)

Offset address: 0x0A

bit	name	Access description: SEO on RO USB bus.		Reset value
7	RB_UHMS_BUS_SE0			X
6	RB_UHMS_BUS_J	J is the RO USB bus.		X
[5:4]	RB_UHMS_LINESTATE[1 :0]	The Linestate signal of the RO PHY.		X
3	RB_UHMS_USB_WAKEUP	RO	USB bus wake-up status: 1: Bus wake-up in progress; 0: Bus wake-up not in progress.	X
2	RB_UHMS_SOF_ACT	RO	USB bus SOF packet transmission status bits: 1: Sending SOF packets; 0: Sending completed or idle.	X
1	RB_UHMS_SOF_PRE	RO	The USB bus SOF packet indicates the following status: 1: An SOF packet is about to be sent; 0: No SOF packet sent.	X
0	RB_UHMS_SOF_FREE	RO	Port enable status: 1: Port enable; 0: Port is not enabled.	X

25.2.2.8 USBHS Host Power Management Data Register (R32_UH_LPM_DATA)

Offset Address:

Bit 0x0C	name	access	describe	Reset value
[31:11] Reserved		RO is reserved.		0
[10:0]	RB_UH_LPM_DATA[10:0]	RW link power management package data content.		X

25.2.2.9 USBHS Host Split Register (R32_UH_SPLIT_DATA) Offset

Address: 0x10

Bit	name	access	describe	Reset value
[31:19] Reserved		RO is reserved.		0
[18:0]	RB_UH_SPLIT_DATA[18 :0]	RW	The host endpoint sends the data content of the Split packet from... The digits from most significant to least significant are ET, E, S, Port, SC, Hub Addr.	X

25.2.2.10 USBHS Host Frame Register (R32_UH_FRAME)

Offset Address: 0x14

Bit	name	access	describe	Reset value
[31:26] Reserved		RO is reserved.		0
25	RB_UH_SOF_CNT_CLR	RW SOF count is reset to zero.		0
twenty four	RB_UH_SOF_CNT_EN	RW SOF counter enable.		0
[23:19] Reserved		RO is reserved.		0
[18:16]	RB_UH_MFRAME_NO[2:0]	RO is the microframe number of the SOF packet that will be sent soon.		0

[15:11] Reserved		RO is reserved.	0
[10:0]	RB_UH_FRAME_NO[10:0]	RW is the frame number of the SOF packet that will be sent soon.	0

25.2.2.11 USBHS Host Transmit Length Register (R32_UH_TX_LEN)

Offset address: 0x18

bit name		access	describe	Reset value
[31:11] Reserved		RO is reserved.		0
[10:0] RB_UH_TX_LEN[10:0]		RW is the length of data sent.		X

25.2.2.12 USBHS Host Receive Length Register (R32_UH_RX_LEN)

Offset address: 0x1C

Bit	name	access	describe	Reset value
[31:11] Reserved		RO is reserved.		0
[10:0] RB_UH_RX_LEN[10:0]		RW is the length of the received data.		X

25.2.2.13 USBHS Host Receive Maximum Length Register (R32_UH_RX_MAX_LEN) Offset Address:

0x20 Bit Name [31:11]

Reserved		access	describe	Reset value
		RO is reserved.		0
[10:0]	RB_UH_RX_MAX_LEN[10:0]	RW maximum receive length.		X

25.2.2.14 DMA Receive Address Register (R32_UH_RX_DMA) Offset

Address: 0x24 bits

	name	access	describe	Reset value
[31:24] Reserved		RO Reserved.		0
[23:0] R32_UH_RX_DMA[23:0]		RW	Receive address, the lower two bits are invalid, a 4-byte pair is required. together.	X

25.2.2.15 DMA Transmit Address Register (R32_UH_TX_DMA) Offset

Address: 0x28

Bit	name	access	describe	Reset value
[31:24] Reserved		RO Reserved.		0
[23:0] R32_UH_TX_DMA[23:0]		RW	Sending address, the lower two bits are invalid, a 4-byte pair is required. together.	X

25.2.2.16 USBHS Host Port Control Register (R32_UH_PORT_CTRL) Offset Address: 0x2C

Bit	name	access	describe	Reset value
[31:17] Reserved		RO is reserved.		0
16	RB_UH_BUS_RST_LONG	RW	Bus reset time selection: 1: Longer reset time;	0

			0: Normal reset.	
[15:12]	RB_UH_PORT_SLEEP_BESL[3:0]	RW	Wake-up time control: 0000: 125us; 1000: 3ms; 0001: 150us; 1001: 4ms; 0010: 200us; 1010: 5ms; 0011: 300us; 1011: 6ms; 0100: 400us; 1100: 7ms; 0101: 500us; 1101: 8ms; 0110: 1ms; 1110: 9ms; 0111: 2ms; RO 1111: 10ms.	X
[11:9] Reserved		reserved.		0
8	RB_UH_CLR_PORT_SLEEP	WO PORT exits SLEEP state (LPM).		0
[7:6]	Reserved	RO is reserved.		0
5	RB_UH_CLR_PORT_CONNECT	WO puts the port into port state.		0
4	RB_UH_CLR_PORT_EN	WO	PORT exits the enabled state and enters the DISABLED state. state.	0
3	RB_UH_SET_PORT_SLEEP	WO PORT enters SLEEP state (LPM).		0
2	RB_UH_CLR_PORT_SUSP	WO PORT Exit the suspended state.		0
1	RB_UH_SET_PORT_SUSP	WO PORT enters a suspended state.		0
0	RB_UH_SET_PORT_RESET	WO port sends a reset.		0

25.2.2.17 USBHS Host Port Configuration Register (R8_UH_PORT_CFG)

Offset address: 0x30

bits	name	Access description: Enabled by pulling down	Reset value
7	RB_UH_PD_EN	a 15K resistor in RW host mode.	0
[6:1]	Reserved	RO is reserved.	0
0	RB_UH_HOST_EN	RW	USB port mode selection: 1: The port is in host mode; 0: The port is in device mode.	0

25.2.2.18 USBHS Host Port Interrupt Enable Register (R8_UH_PORT_INT_EN)

Offset address: 0x32

	name	access	describe	Reset value
Position [7:6]	Reserved	RO is reserved.		0
5	RB_UHIE_PORT_SLP	RW port sleep state change interrupt enabled.		0
4	RB_UHIE_PORT_RESET	RW port reset state change interrupt enabled.		0
3	Reserved	RO is reserved.		0
2	RB_UHIE_PORT_SUSP	RW port pause state change interrupt enabled.		0
1	RB_UHIE_PORT_EN	RW port enable state change interrupt enable.		0
0	RB_UHIE_PORT_CONNEC	RW Enables interrupt for port connection status changes.		0

	T			

25.2.2.19 USBHS Host Port Test Mode Register (R8_UH_PORT_TEST_CT) Offset Address: Bit 0x33 [7:5]

	name	access	describe	Reset value
	Reserved	RO is reserved.		0
4	RB_UH_TEST_SE0_NAK	RW test TEST_SE0_NAK.		0
3	RB_UH_TEST_PACKET	RW tests TEST_PKT.		0
2	RB_UH_TEST_FORCE_EN	RW Enables test mode.		0
1	RB_UH_TEST_K	RW test output K.		0
0	RB_UH_TEST_J	RW Test output J.		0

25.2.2.20 USBHS Host Port Status Register (R16_UH_PORT_ST) Offset Address: 0x34 bits

	name	access	describe	Reset value
[15:12]	Reserved	RO is reserved.		0
11	RB_UHIS_PORT_TEST	Is the RO port in test mode?		0
10	RB_UHIS_PORT_HS	Is the RO port connection speed high-speed?		0
9	RB_UHIS_PORT_LS	Is the RO port connection speed low?		0
[8:6]	Reserved	RO is reserved.		0
5	RB_UHIS_PORT_SLP	The RO port is in sleep mode.		0
4	RB_UHIS_PORT_RST	RO port reset status.		1
3	Reserved	RO is reserved.		0
2	RB_UHIS_PORT_SUSP	The RO port is in a paused state.		0
1	RB_UHIS_PORT_EN	Enable the RO port.		0
0	RB_UHIS_PORT_C0NNECT	RO port connection status.		0

25.2.2.21 USBHS Host Port Status Change Register (R8_UH_PORT_CHG) Offset Address: 0x36 bits

	name	access	describe	Reset value
[7:6]	Reserved	RO is reserved.		0
5	RB_UHIF_PORT_SLP	When the RW1Z port changes its sleep state, writing 1 clears it to zero.		0
4	RB_UHIF_PORT_RESET	The RW1Z port reset state changes; writing 1 clears it.		0
3	Reserved	RO is reserved.		0
2	RB_UHIF_PORT_SUSP	The RW1Z port pause state changes; writing 1 clears it to zero.		0
1	RB_UHIF_PORT_EN	The RW1Z port enable state changes; writing 1 clears it.		0
0	RB_UHIF_PORT_CONNECT	When the RW1Z port connection status changes, writing 1 clears the connection.		0

25.2.2.22 USBHS Host BC Charging Control Register (R32_UH_BC_CTRL) Offset Address: Bit 0x3C

	name	access	describe	Reset value

[31:11] Reserved		RO is reserved.		0
10	RB_UDM_VSRC_ACT	RO	In automatic mode, this bit indicates the UDM output VBC_SRC, otherwise... It is then controlled by RB_UDM_BC_CMPE. Note: When UDP is above VBC_SRC but not high. UDP, this bit is 1; otherwise, it is 0.	0
9	RB_UDM_BC_VSRC	RW	UDM pin BC protocol source voltage enable: 1: The UDM pin outputs the BC protocol source voltage VBC_SRC; 0: Output is disabled.	0
8	RB_UDP_BC_VSRC	RW	UDP pin BC protocol source voltage enable: 1: The UDP pin outputs the BC protocol source voltage VBC_SRC; 0: Output is disabled.	0
7	Reserved	RO reserved.		0
6	RB_BC_AUTO_MODE	RW	Automatic mode enabled: 1: Enable; 0: Off.	0
5	RB_UDM_BC_CMPE	RW	UDM pin BC protocol comparator enable: 1: Enable; 0: Off.	0
4	RB_UDP_BC_CMPE	RW	UDP pin BC protocol comparator enable: 1: Enable; 0: Off.	0
[3:2]	Reserved	RO is reserved.		0
1	RB_UDM_BC_CMPO	RO	UDM pin BC protocol comparator status: 1: The UDM pin voltage is higher than the BC protocol reference value VBC_REF; 0: The UDM pin voltage is lower than the BC protocol reference value VBC_REF.	0
0	RB_UDP_BC_CMPO	RO	UDP pin BC protocol comparator status: 1: The UDP pin voltage is higher than the BC protocol reference value VBC_REF; 0: The UDP pin voltage is lower than the BC protocol reference value VBC_REF.	0

Chapter 26 USB Full-Speed Host/Device Controller (USBFS/OTG_FS)

26.1 Introduction to USB Controllers

USB 2.0 Fullspeed Host Controller and Device Controller (USBFS), conforming to the USB 2.0 Fullspeed standard. Provides 16 ports. Configured USB device endpoints and a set of host endpoints. Supports control/bulk/synchronous/interruptible transfers, double-buffered mechanism, and USB bus suspend. /Restore operation and provide standby/wake-up functions.

The OTG_FS is a dual-role USB controller that supports both host and device functionality, and is compatible with the On-The-Go Supplement to the USB 2.0 specification. It can also be configured to support only host functionality or only device functionality, while remaining compatible with the full-speed USB 2.0 specification.

The main features are as follows:

Support is provided in the USB On-The-Go Supplement (physical layer of the OTG_FS controller), which is defined as optional in the Revision 1.3 specification.

Project OTG Protocol

The software allows configuration of USB full-speed host, USB full-speed/low-speed devices, and USB dual-role devices.

Provides power-saving function

Supports control transfer, batch transfer, interrupted transfer, and real-time/synchronous transfer.

Provides bus reset, suspend, wake-up, and resume functions .

Supports data packets up to 64 bytes, has a built-in FIFO, and supports interrupts and DMA.

26.2 Register Description

The USB-related registers are divided into three parts, with some registers being reused in host and device modes.

USB Global Register

USB Device Control Register

USB host control register

26.2.1 Global Register Description

Table 26-1 List of USBFS Related Registers

name	Access address	describe	Reset value
R8_USB_CTRL	0x40023400	USB Control Register	0x06
R8_USB_INT_EN	0x40023402	USB Interrupt Enable Register	0x00
R8_USB_DEV_AD	0x40023403	USB Device Address Register	0x00
R8_USB_MIS_ST	0x40023405	USB Miscellaneous Status Register	0xXX
R8_USB_INT_FG	0x40023406	USB interrupt flag register	0x20
R8_USB_INT_ST	0x40023407	USB Interrupt Status Register	0xXX
R16_USB_RX_LEN	0x40023408	USB Receive Length Register	0x0XXX
R32_USB_OTG_CR	0x40023454	USB OTG Control Register	0x00000000
R32_USB_OTG_SR	0x40023458	USB OTG Status Register	0x0000000X

26.2.1.1 USB Control Register (R8_USB_CTRL)

Offset address: 0x00

Bit	name	access	describe	Reset value
7	RB_UC_HOST_MODE	RW	USB working mode selection bit: 1: Host mode;	0

			0: Device mode.	
6	RB_UC_LOW_SPEED	RW	USB bus signal transmission rate selection bit: 1: 1.5Mbps; 0: 12Mbps.	0
5	RB_UC_DEV_PU_EN	RW	In USB device mode, USB device enable and internal pull-up The resistor control bit, when set to 1, enables USB device transmission. It also enables the internal pull-up resistor.	0
[5:4] MASK_UC_SYS_CTRL		RW refers to the USB system configuration in the table below.		0
3	RB_UC_INT_BUSY	RW	Automatically pause before the USB transfer completion interruption flag is cleared. Enable bit: 1: Before the interrupt flag UIF_TRANSFER is cleared, the system will automatically... Automatic pause, automatic response busy NAK in device mode, main Automatically pause subsequent transmissions in machine mode; 0: Do not pause.	0
2	RB_UC_RST_SIE	RW	USB protocol processor software reset control bit: 1: Force reset the USB protocol processor (SIE). The software needs to be reset; 0: Do not reset.	1
1	RB_UC_CLR_ALL	RW	Clear the USB FIFO and interrupt flag: 1: Force clear and reset; 0: Unclear.	1
0	RB_UC_DMA_EN	RW	USB DMA and DMA interrupt control bits: 1: Enable DMA function and DMA interrupt; 0: DMA is off.	0

The USB system control combination consists of RB_UC_HOST_MODE and MASK_UC_SYS_CTRL.

Table 26-2 USB System Control Combination

RB_UC_HOST_MODE	MASK_UC_SYS_CTRL	USB System Control Description
0	00	Disable USB device functionality and turn off the internal pull-up resistor.
0	01	To enable USB device functionality, disable the internal pull-up resistor; an external pull-up resistor is required.
0	1x	Enables USB device functionality by activating the internal 1.5K pull-up resistor. Resistors take precedence over pull-down resistors and can also be used in GPIO mode.
1	00	USB host mode, in normal working condition.
1	01	USB host mode, forces DP/DM output to SE0 state.
1	10	USB host mode, force DP/DM output to J state.
1	11	USB host mode, force DP/DM output K state/wake-up.

26.2.1.2 USB Interrupt Enable Register (R8_USB_INT_EN) Offset Address: Bit

0x02

	name	access	describe	Reset value
7	Reserved	RO is reserved.		0
6	RB_UIE_DEV_NAK	RW	USB device mode, received NAK interrupt: 1: Enable interrupt; 0: Interruption is disabled.	0

5	RB_U_1WIRE_MODE	RW	USB single-wire mode enabled: 1: Enable USB single-wire mode; 0: Disable USB single-wire mode.	0
4	RB_UIE_FIFO_OV	RW	FIFO overflow interrupt: 1: Enable interrupt; 0: Interruption is disabled.	0
3	RB_UIE_HST_SOF	RW	USB host mode, SOF timer interrupt: 1: Enable interrupt; 0: Interruption is disabled.	0
2	RB_UIE_SUSPEND	RW	USB bus suspend or wake-up event interrupt: 1: Enable interrupt; 0: Interruption is disabled.	0
1	RB_UIE_TRANSFER	RW	USB transfer completed but interrupted: 1: Enable interrupt; 0: Interruption is disabled.	0
0	RB_UIE_DETECT	RW	USB host mode, USB device connection or disconnection events. Interruption: 1: Enable interrupt; 0: Interruption is disabled.	0
	RB_UIE_BUS_RST	RW	USB device mode, USB bus reset event interrupt: 1: Enable interrupt; 0: Interruption is disabled.	0

26.2.1.3 USB Device Address Register (R8_USB_DEV_AD) Offset Address: 0x03

Bit	name	access	describe	Reset value
7	RB_UDA_GP_BIT	RW	USB General Flags, user-defined.	0
[6:0]	MASK_USB_ADDR	RW	Host mode: The address of the currently operating USB device; Device mode: The USB's own address.	0

26.2.1.4 USB Miscellaneous Status Register (R8_USB_MIS_ST) Offset Address:

Bit 0x05

	name	access		Reset value
7	RB_UMS_SOF_PRES	RO	Description of SOF packet status bits in USB host mode: 1: An SOF packet is about to be sent. If there are other USB ports at this time... Data packets will be automatically delayed; 0: No SOF packet sent.	X
6	RB_UMS_SOF_ACT	RO	SOF packet transfer status bits in USB host mode: 1: Sending SOF packets; 0: Sending completed or idle.	X
5	RB_UMS_SIE_FREE	RO	USB protocol processor idle status bits: 1: The protocol unit is idle; 0: Busy, in the process of transferring data via USB.	1
4	RB_UMS_R_FIFO_RDY	RO	USB Receive FIFO Data Ready Status Bits:	0

			1: The receive FIFO is not empty; 0: The receive FIFO is empty.	
3	RB_UMS_BUS_RST	RO	USB bus reset status bit: 1: The USB bus is currently in a reset state; 0: The USB bus is currently in a non-reset state.	X
2	RB_UMS_SUSPEND	RO	USB suspend status bit: 1: The USB bus is in a suspended state; it has been idle for some time. There is USB activity; 0: The USB bus is in a non-suspended state.	0
1	RB_UMS_DM_LEVEL	RO	In USB host mode, the device is just connected to the USB port. The voltage level of the DM pin is used to determine the speed. 1: High level/Low speed; 0: Low level/Full speed.	0
0	RB_UMS_DEV_ATTACH	RO	USB device connection status of the port in USB host mode State: 1: The port is already connected to a USB device; 0: No USB device is connected to the port.	0

26.2.1.5 USB Interrupt Flag Register (R8_USB_INT_FG) Offset Address: 0x06

Bit	name	access	describe	Reset value
7	RB_U_IS_NAK	RO	In USB device mode, the NAK response status bits are: 1: Responding with NAK during the current USB transfer process; 0: No NAK response.	0
6	RB_U_TOG_OK	RO	The current USB transfer DATA0/1 synchronization flag is matched. State: 1: Synchronization; 0: out of sync.	0
5	RB_U_SIE_FREE	RO	USB protocol processor idle status bits: 1: USB is idle; 0: Busy, in the process of transferring data via USB.	1
4	RB_UIF_FIFO_OV	RW	USB FIFO overflow interrupt flag: Write 1 to clear. 1: FIFO overflow triggered; 0: No event.	0
3	RB_UIF_HST_SOF	RW	In USB host mode, the SOF timer interrupt flag is set. Write 1 to clear: 1: Triggered upon completion of SOF packet transmission; 0: No event.	0
2	RB_UIF_SUSPEND	RW	USB bus suspend or wake-up event interrupt flag, write 1. Reset: 1: Triggered by a USB suspend or wake-up event; 0: No event.	0
1	RB_UIF_TRANSFER	RW	USB transfer completion interrupt flag: Write 1 to clear. 1: Triggered upon completion of a USB transfer; 0: No event.	0

0	RB_UIF_DETECT	RW	USB device connection or disconnection in USB host mode Interruption flag bit: Write 1 to clear: 1: Triggered by detecting a USB device connection or disconnection; 0: No event.	0
	RB_UIF_BUS_RST	RW	USB bus reset event interrupt in USB device mode Flag bit, write 1 to clear: 1: USB bus reset event triggered; 0: No event.	0

26.2.1.6 USB Interrupt Status Register (R8_USB_INT_ST) Offset Address: 0x07

Bit	name	access		Reset value
7	RB_UIS_IS_NAK	RO	Describes the NAK response status bit in USB device mode, similar to... RB_U_IS_NAK: 1: Responding with NAK during the current USB transfer process; 0: No NAK response.	0
6	RB_UIS_TOG_OK	RO	The current USB transfer DATA0/1 synchronization flag is matched. Status, same as RB_U_TOG_OK: 1: Synchronization; 0: out of sync.	0
[5:4] MASK_UIS_TOKEN		RO	In device mode, the token for the current USB transfer transaction. PID identifier.	X
[3:0]	MASK_UIS_ENDP	RO	In device mode, the endpoint of the current USB transfer transaction Number.	X
	MASK_UIS_H_RES	RO	In host mode, the response to the current USB transfer transaction. PID identifier, 0000 indicates that the device did not respond or timed out; Other values represent the response PID.	X

MASK_UIS_TOKEN is a token PID used in USB device mode to identify the current USB transfer transaction: 00 indicates an OUT packet; 01 indicates a PIN. 10 indicates an IN packet; 11 indicates a SETUP packet.

MASK_UIS_H_RES is only valid in host mode. In host mode, this value is applied when the host sends an OUT/SETUP token packet. The PID represents the handshake packet's ACK/NAK/STALL signal, or a device that did not respond/timed out. If the host sends an IN token packet, then the PID represents the packet's... PID (DATA0/DATA1) or handshake packet PID.

26.2.1.7 USB Receive Length Register (R16_USB_RX_LEN) Offset Address: Bit 0x08

	name	access	describe	Reset value
[15:10] Reserved		RO is reserved.		0
[9:0] R16_USB_RX_LEN		RO: Number of bytes of data currently received by the USB endpoint.		X

26.2.1.8 USB OTG Control Register (R32_USB_OTG_CR) Offset Address: 0x54 Bit

Name [31:6] Reserved

		access	describe	Reset value
		RO is reserved.		0

5	RB_CR_SESS_VTH	RW	OTG Session Effective Threshold Voltage Setting: 1: The SESS_VLD level is 1.4V; 0: SESS_VLD level is 0.8V.	0
4	RB_CR_VBUS_VTH	RW	OTG VBUS threshold voltage setting: 1: The VBUS_VLD level is 4.4V; 0: VBUS_VLD level is 4.8V.	0
3	RB_CR_OTG_EN	RW	OTG Function Enabled: 1: Enable; 0: Prohibited.	0
2	RB_CR_IDPU	RW	USB_OTG_ID pin pull-up enable: 1: Enable; 0: Prohibited.	0
1	RB_CR_CHARGE_VBUS	RW	OTG VBUS charging enable: 1: Enable; 0: Prohibited.	0
0	RB_CR_DISCHAR_VBUS	RW	OTG VBUS discharge enable: 1: Enable; 0: Prohibited.	0

23.2.1.9 USB OTG Status Register (R32_USB_OTG_SR) Offset Address: 0x58 bits

	name	access	describe	Reset value
[31:4] Reserved		RO is reserved.		0
3	RB_SR_ID_DIG	RO	OTG ID logo: 1: Equipment B; 0: A device.	0
2	RB_SR_SESS_END	RO	OTG session termination flag: 1: Effective; 0: Invalid.	0
1	RB_SR_SESS_VLD	RO	OTG session validity flag: 1: Valid, the session valid level is greater than the threshold voltage; 0: Invalid, the session valid level is less than the threshold voltage.	0
0	RB_SR_VBUS_VLD	RO	OTG VBUS input level: 1: VBUS voltage is greater than the threshold voltage; 0: VBUS voltage is less than the threshold voltage.	X

26.2.2 Device Register Description In USB

device mode, the USB OTG module provides eight sets of bidirectional endpoint configuration registers (endpoint numbers 0-7), which can be mapped to endpoint numbers. In the configuration of 8-15, the maximum packet length for all endpoints except endpoint 3 is 64 bytes, and the maximum packet length for endpoint 3 is 1023 characters. Section (synchronous transmission).

Endpoint 0 is the default endpoint, supporting control transfers. Sending and receiving share a single 64-byte data buffer.

Endpoints 1-15 can be configured with independent 64-byte send and receive buffers or dual 64-byte data buffers, supporting bulk transfers. Interrupted transmission and real-time/synchronous transmission.

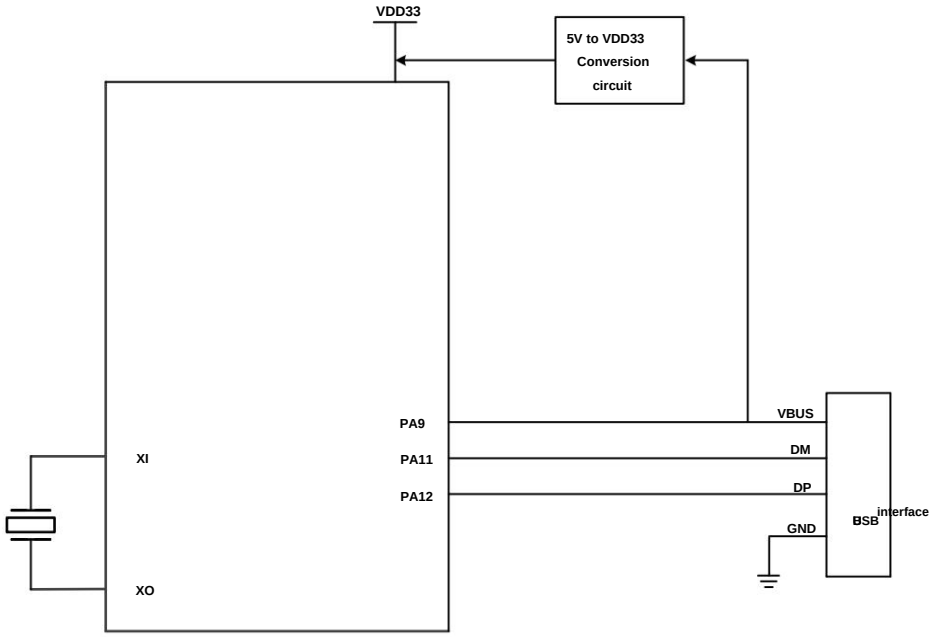
Each set of endpoints has a control register R8_UEPn_CTRL and a transmit length register R16_UEPn_T_LEN, used to set the...

The endpoint's synchronization trigger bit, responses to OUT and IN transactions, and the length of the data sent, etc.

The host and device roles of a USB OTG module are determined by the state of the OTG_FS_ID pin. When the OTG_FS_ID pin is floating, its built-in pull-up resistor will set the RB_SR_ID_DIG bit in the USB OTG status register R32_USB_OTG_SR to 1, at which point the controller should be initialized as device B. When the OTG_FS_ID pin is grounded, the RB_SR_ID_DIG bit in the USB OTG status register R32_USB_OTG_SR is 0, at which point the controller should be initialized as device A.

Device B, at the start of a session, ensures that the VBUS level is lower than VSESS_VLD,min by checking RB_SR_SESS_VLD in the OTG status register R32_USB_OTG_SR. If this bit is 0, a new session can begin. If this bit is 1, the RB_CR_DISCHAR_VBUS bit in the OTG control register R32_USB_OTG_CR can be set to 1 to discharge the VBUS level, making it lower than the session threshold level.

When used as a Class B device, it needs to draw power from VBUS, requiring an external conversion circuit, as shown in Figure 26-1. Figure 26-1 Connection diagram for OTG Class B device.



The USB bus pull-up resistors, essential for USB devices, can be enabled or disabled by software at any time. When RB_UC_DEV_PU_EN in the USB control register R8_USB_CTRL is set to 1, the controller internally connects pull-up resistors to the DP/DM pins of the USB bus based on the speed setting of RB_UD_LOW_SPEED, and enables the USB device function. When a USB bus reset, USB bus suspend or wake-up

event is detected, or when USB successfully completes data transmission or reception, the USB protocol processor sets the corresponding interrupt flag. If interrupt enable is enabled, a corresponding interrupt request will also be generated. The application can directly query or query and analyze the interrupt flag register R8_USB_INT_FG within the USB interrupt service routine, based on...

RB_UIF_BUS_RST and RB_UIF_SUSPEND are processed accordingly. Furthermore, if RB_UIF_TRANSFER is valid, the USB interrupt status register R8_USB_INT_ST needs to be analyzed further, and processing should be performed based on the current endpoint number MASK_UIS_ENDP and the current transaction token PID identifier MASK_UIS_TOKEN. If the synchronization trigger bit RB_UEP_R_TOG for each endpoint's OUT transaction is set beforehand, then RB_U_TOG_OK or RB_UIS_TOG_OK can be used to determine whether the synchronization trigger bit of the currently received data packet matches the synchronization trigger bit of that endpoint. If the data is synchronized, the data is valid; if the data is not synchronized, the data should be discarded. After each USB send or receive interrupt, the synchronization trigger bit of the corresponding endpoint should be correctly modified for synchronization detection of the next data packet sent or received. In addition, setting RB_UEP_T_AUTO_TOG or RB_UEP_R_AUTO_TOG can automatically toggle the corresponding synchronization trigger bit after successful transmission or reception.

Each endpoint prepares to send data in its own buffer, and the length of the data to be sent is set independently in R8_UEPn_T_LEN. Each endpoint receives data in its own buffer, but the length of the received data is in the USB receive length register R8_USB_RX_LEN, which can be distinguished based on the current endpoint number when USB receive is interrupted.

Table 26-3 List of Device-Related Registers

name	Access address description		Reset value
R8_UDEV_CTRL	0x40023401	USB Device Physical Port Control	0xX0
R8_UEP4_1_MOD	0x4002340C	Register Endpoint 1 (9) / 4 (8/12) Mode Control Register	0x00
R8_UEP2_3_MOD	0x4002340D	Register endpoint 2 (10) / 3 (11) mode control register instrument	0x00
R8_UEP5_6_MOD	0x4002340E	Endpoint 5 (13) / 6 (14) mode control register instrument	0x00
R8_UEP7_MOD	0x4002340F Endpoint 7 (15) Mode Control Register 0x40023410		0x00
R32_UEP0_DMA	Endpoint 0 Buffer Start Address 0x40023414 Endpoint		0xXXXXXXXX
R32_UEP1_DMA	1 (9) Buffer Start Address 0x40023418 Endpoint 2 (10) Buffer		0xXXXXXXXX
R32_UEP2_DMA	Start Address 0x4002341C Endpoint 3 (11) Buffer Start Address		0xXXXXXXXX
R32_UEP3_DMA	0x40023420 Endpoint 4 (8/12) Buffer Start Address 0x40023424		0xXXXXXXXX
R32_UEP4_DMA	Endpoint 5 (13) Buffer Start Address 0x40023428 Endpoint 6 (14)		0xXXXXXXXX
R32_UEP5_DMA	Buffer Start Address 0x4002342C Endpoint 7 (15) Buffer Start		0xXXXXXXXX
R32_UEP6_DMA	Address 0x40023430 Endpoint 0 Transmit Length Register		0xXXXXXXXX
R32_UEP7_DMA	0x40023432 Endpoint 0 Transmit Control Register 0x40023433		0xXXXXXXXX
R8_UEP0_T_LEN	Endpoint 0 Receive Control Register 0x40023434		0xXX
R8_UEP0_TX_CTRL	Endpoint 1 (9) Transmit Length Register 0x40023436		0x00
R8_UEP0_RX_CTRL	Endpoint 1 (9) Transmit Control Register 0x40023437		0x00
R8_UEP1_T_LEN	Endpoint 1 (9) Receive Control Register 0x40023438 Endpoint		0xXX
R8_UEP1_TX_CTRL	2 (10) Transmit Length Register 0x4002343A Endpoint 2 (10)		0x00
R8_UEP1_RX_CTRL	Transmit Control Register 0x4002343B Endpoint 2 (10)		0x00
R8_UEP2_T_LEN	Receive Control Register 0x4002343C Endpoint 3 (11) Transmit		0xXX
R8_UEP2_TX_CTRL	Length Register 0x4002343E Endpoint 3 (11) Transmit Control		0x00
R8_UEP2_RX_CTRL	Register 0x4002343F Endpoint 3 (11) Receive Control Register		0x00
R16_UEP3_T_LEN	0x40023440 Endpoint 4 (8/12) Transmit Length Register		0xXX
R8_UEP3_TX_CTRL	0x40023442 Endpoint 4 (8/12) Transmit Control Register		0x00
R8_UEP3_RX_CTRL	0x40023443 Endpoint 4 (8/12) Receive Control Register		0x00
R8_UEP4_T_LEN	0x40023444 Endpoint 5 (13) Transmit Length Register 0x40023446		0xXX
R8_UEP4_TX_CTRL	Endpoint 5 (13) Transmit Control Register 0x40023447 Endpoint 5		0x00
R8_UEP4_RX_CTRL	(13) Receive Control Register 0x40023448 Endpoint 6 (14) Transmit		0x00
R8_UEP5_T_LEN	Length Register 0x4002344A Endpoint 6 (14) Transmit Control		0xXX
R8_UEP5_TX_CTRL	Register 0x4002344B Endpoint 6 (14) Receive Control Register		0x00
R8_UEP5_RX_CTRL	0x4002344C Endpoint 7 (15) Transmit Length Register		0x00
R8_UEP6_T_LEN	0x4002344E Endpoint 7 (15) Transmit Control Register		0xXX
R8_UEP6_TX_CTRL	0x4002344F Endpoint 7 (15) Receive Control Register		0x00
R8_UEP6_RX_CTRL			0x00
R8_UEP7_T_LEN			0xXX
R8_UEP7_TX_CTRL			0x00
R8_UEP7_RX_CTRL			0x00

26.2.2.1 USB Device Physical Port Control Register (R8_UDEV_CTRL)

Offset address: 0x01

bit	name	access		Reset value
7	RB_UD_PD_DIS	RW	Description of the internal pull-down power of the UD+/UD- pins of the USB device port Restriction control bit: 1: Disable internal dropdown; 0: Enables the internal dropdown. It can be used to provide pull-down resistors in GPIO mode.	1
6	Reserved	RO is reserved.		0
5	RB_UD_DP_PIN	RO	Current UD+ pin status: 1: High level; 0: Low level.	X
4	RB_UD_DM_PIN	RO	Current UD pin status: 1: High level; 0: Low level.	X
3	Reserved	RO is reserved.		0
2	RB_UD_LOW_SPEED	RW	USB device physical port low-speed mode enable bit: 1: Select 1.5Mbps low-speed mode; 0: Select 12Mbps full speed mode.	0
1	RB_UD_GP_BIT	RW: Universal flag for USB device mode, user-defined. 0: USB device		
0	RB_UD_PORT_EN	RW	physical port enable bit: 1: Enable physical port; 0: Disable physical ports.	0

26.2.2.2 Endpoint 1 (9)/4 (8/12) Mode Control Register (R8_UEP4_1_MOD)

Offset address: 0x0C

Bit	name	access	describe	Reset value
7	RB_UEP1_RX_EN	RW	1: Enable endpoint 1 (9) to receive (OUT); 0: Disable endpoint 1 (9) from	0
6	RB_UEP1_TX_EN	RW	receiving. 1: Enable endpoint 1 (9) from sending (IN); 0: Disable endpoint 1 (9) from sending.	0
5	Reserved	RO is reserved.		0
4	RB_UEP1_BUF_MOD	RW endpoint 1 (9) data buffer mode control bit. 1: Enable		0
3	RB_UEP4_RX_EN	RW	endpoint 4 (8/12) receive (OUT); 0: Disable endpoint 4 (8/12) from receiving.	0
2	RB_UEP4_TX_EN	RW	1: Enable endpoint 4 (8/12) from transmitting (IN). 0: Disable endpoint 4 (8/12) from sending.	0
1	Reserved	RO is reserved.		0
0	RB_UEP4_BUF_MOD	RW endpoint 4 (8/12) data buffer mode control bit. 0		

Note: Endpoint configuration options map to endpoint 9, endpoint configuration options map to endpoints 8 and 12.

26.2.2.3 Offset address of endpoint 2 (10)/3 (11) mode control register (R8_UEP2_3_MOD) : 0x0D bit

	name	access	describe	Reset value
7	RB_UEP3_RX_EN	RW 1: Enable endpoint 3 (11) to receive (OUT);		0

			0: Endpoint 3 (11) is disabled from receiving.	
6	RB_UEP3_TX_EN	RW	1: Enable endpoint 3 (11) to send (IN); 0: Disable endpoint 3 (11) from sending.	0
5	Reserved	RO is reserved.		0
4	RB_UEP3_BUF_MOD	RW endpoint 3 (11) data buffer mode control bit. 1: Enable		0
3	RB_UEP2_RX_EN	RW	endpoint 2 (10) receive (OUT); 0: Disable endpoint 2 (10) from	0
2	RB_UEP2_TX_EN	RW	receiving. 1: Enable endpoint 2 (10) from sending (IN); 0: Disable endpoint 2 (10) from sending.	0
1	Reserved	RO is reserved.		0
0	RB_UEP2_BUF_MOD	RW endpoint 2 (10) data buffer mode control bit.		0

Note: Endpoint 2 Configuration options map endpoint 10, endpoint configuration 11 endpoint mapping

26.2.2.4 Offset address of endpoint 5 (13)/6 (14) mode control register (R8_UEP5_6_MOD) : 0x0E

bit

	name	access		Reset value
7	RB_UEP6_RX_EN	RW	Description 1: Enable endpoint 6 (14) to receive (OUT); 0: Disable endpoint 6 (14) from	0
6	RB_UEP6_TX_EN	RW	receiving. 1: Enable endpoint 6 (14) from sending (IN); 0: Disable endpoint 6 (14) from sending.	0
5	Reserved	RO is reserved.		0
4	RB_UEP6_BUF_MOD	RW endpoint 6 (14) data buffer mode control bit. 1: Enable		0
3	RB_UEP5_RX_EN	RW	endpoint 5 (13) receive (OUT); 0: Disable endpoint 5 (13) from	0
2	RB_UEP5_TX_EN	RW	receiving. 1: Enable endpoint 5 (13) from sending (IN); 0: Disable endpoint 5 (13) from sending.	0
1	Reserved	RO is reserved.		0
0	RB_UEP5_BUF_MOD	RW endpoint 5 (13) data buffer mode control bit.		0

Note: Endpoint 5 Configuration options map endpoint 13, endpoint configuration 14 endpoint mapping

26.2.2.5 Endpoint 7 (15) Mode Control Register (R8_UEP7_MOD)

Offset address: 0x0F

Bit	name	access	describe	Reset value
[7:4] Reserved		RO is reserved.		0
3	RB_UEP7_RX_EN	RW	1: Enable endpoint 7 (15) to receive (OUT); 0: Disable endpoint 7 (15) from	0
2	RB_UEP7_TX_EN	RW	receiving. 1: Enable endpoint 7 (15) from sending (IN); 0: Disable endpoint 7 (15) from sending.	0
1	Reserved	RO is reserved.		0
0	RB_UEP7_BUF_MOD	RW endpoint 7 (15) data buffer mode control bit.		0

Note: Endpoint 7 configuration options map to endpoint 15.

The data for USB endpoints 1-15 are configured using combinations of RB_UEPn_RX_EN, RB_UEPn_TX_EN, and RB_UEPn_BUF_MOD. For details on buffer modes, please refer to Table 26-4. In dual 64-byte buffer mode, USB data transfer will be based on...

RB_UEP_*_TOG=0 selects the first 64 bytes of the buffer, and RB_UEP_*_TOG=1 selects the last 64 bytes of the buffer.

Setting RB_UEP_*_AUTO_TOG=1 enables automatic switching. Note that during synchronous transmission, when (RB_UEPn_RX_EN, ,
RB_UEPn_TX_EN, RB_UEPn_BUF_MOD) are only (1,0,0) or (0,1,0), endpoint 3 supports a maximum of [missing value].
1023 bytes.

Table 26-4 Endpoint n Buffer Patterns (n=1-7)

RB_UEPn_ RX_EN	RB_UEPn_ TX_EN	RB_UEPn_ BUF_MOD	Description: Listed from low to high address, starting with R16_UEPn_DMA.
0	0		The X endpoint is disabled, and the R16_UEPn_DMA buffer is not used.
1	0		0 Single 64-byte receive buffer (OUT).
1	0		1. Dual 64-byte receive buffer (OUT), selected by RB_UEP_R_TOG.
0	1		0 Single 64-byte send buffer (IN).
0	1		1. Dual 64-byte transmit buffer (IN), selected by RB_UEP_T_TOG.
1	1		0 Single 64-byte receive buffer (OUT), single 64-byte transmit buffer (IN).
1	1	1	Dual 64-byte receive buffers (OUT), selected via RB_UEP_R_TOG. Dual 64-byte transmit buffers (IN), selected via RB_UEP_T_TOG. The complete 256 bytes are arranged as follows: UEPn_DMA+0 address: endpoint receive address when RB_UEP_R_TOG=0; UEPn_DMA+64 address: endpoint receive address when RB_UEP_R_TOG=1; UEPn_DMA+128 address: endpoint send address when RB_UEP_T_TOG=0; UEPn_DMA+192 Address: RB_UEP_T_TOG=1 is the endpoint sending address.

21-4 8-15 1-7

Note: The table configuration options support n=1-7, and the endpoint configuration is mapped to the endpoint configuration.

26.2.2.6 Endpoint n Buffer start address (R32_UEPn_DMA) (n=0-7) Offset address: 0x10 + (4*n)

Bit	name	access		Reset value
[31:0]	R32_UEPn_DMA	RW	Describes the starting address of the endpoint n buffer. Addresses must be 4-byte aligned.	X

Note: 1. The length of the receive data buffer is >= 2. min (maximum possible packet length endpoint, + 2 (64 bytes).

Endpoint configuration is supported configurable endpoint) 8-15 Endpoint.
DMA 0-7

26.2.2.7 Endpoint n Transmit Length Register (R8_UEPn_T_LEN) (n=0-2,4-7) Offset address: 0x30 + (4*n)

Bit	name	access		Reset value
[7:0]	R8_UEPn_T_LEN	RW	Description of setting up USB endpoint n: Number of bytes of data to be sent n=0, 1, 2, 4, 5, 6, 7.	X

Note 1. Endpoint send length configuration supports endpoints and can map the send length of endpoints.
0-7 8-15 2. Maximum number of bytes
supported by the host for synchronous endpoints.
1023

26.2.2.8 Endpoint n Transmit Length Register (R16_UEPn_T_LEN) (n=3) Offset Address: 0x3C

Bit name		access	describe	Reset value
[15:10]	Reserved	RO is reserved.		0
[9:0]	R16_UEP3_T_LEN	RW sets the number of bytes of data that USB endpoint 3 is prepared to send.		X

Note 1. Endpoint send length configuration supports endpoints and can map the send length of endpoints.
0-7 8-15 2. Maximum number of bytes
supported by the host for synchronous endpoints.
1023

26.2.2.9 Endpoint n Control Register (R8_UEPn_TX_CTRL) (n=0-7)

Offset address: 0x32 + (4*n)

Bit	name	access	describe	Reset value
[7:4] Reserved		RO is reserved.		0
3	RB_UEP_T_AUTO_TOG	RW	Synchronous trigger bit auto-flip enable control bit: 1: Automatically reverse the corresponding synchronization after successful data transmission. Trigger bit; 0: Does not automatically flip; can be switched manually. Note: This endpoint is reserved.	0
2	RB_UEP_T_TOG	RW	The transmitter (handling IN transactions) of USB endpoint n Backup synchronization trigger bit: 1: Send DATA1; 0: Send DATA0.	0
[1:0] MASK_UEP_T_RES		RW	The sender at endpoint n controls the response to IN transactions: 00: DATA0/DATA1 data is ready and expects ACK; 01: Response to DATA0/DATA1 and expect no response. Used for real-time/synchronous transmission of non-endpoint 0; 10: Respond with NAK or Busy; 11: Response STALL or error.	00b

Note: Endpoint configuration supports 0,7,8-15 endpoints and can map configured endpoints.

26.2.2.10 Endpoint n Control Register (R8_UEPn_RX_CTRL) (n=0-7) Offset address: 0x33 + (4*n)

Bit	name	access	describe	Reset value
[7:4] Reserved		RO reserved.		0
3	RB_UEP_R_AUTO_TOG	RW	Synchronous trigger bit auto-flip enable control bit: 1: Automatically flip the corresponding synchronization after successful data reception. Trigger bit; 0: Does not automatically flip; can be switched manually. Note: This endpoint is reserved.	0
2	MASK_UEP_R_TOG	RW	The receiver at USB endpoint n (handles OUT transactions) Expected synchronization trigger bit: 1: Expected DATA1; 0: Expected DATA0.	0
[1:0] MASK_UEP_R_RES		RW	The receiver at endpoint n controls the response to OUT transactions: 00: Acknowledgment (ACK); 01: Timeout/No Response, used for real-time responses to non-endpoint 0. Synchronous transmission; 10: Respond with NAK or Busy; 11: Response STALL or error.	00b

Note: Endpoint configuration supports 0,7,8-15 endpoints and can map configured endpoints.

26.2.3 USB Host Register

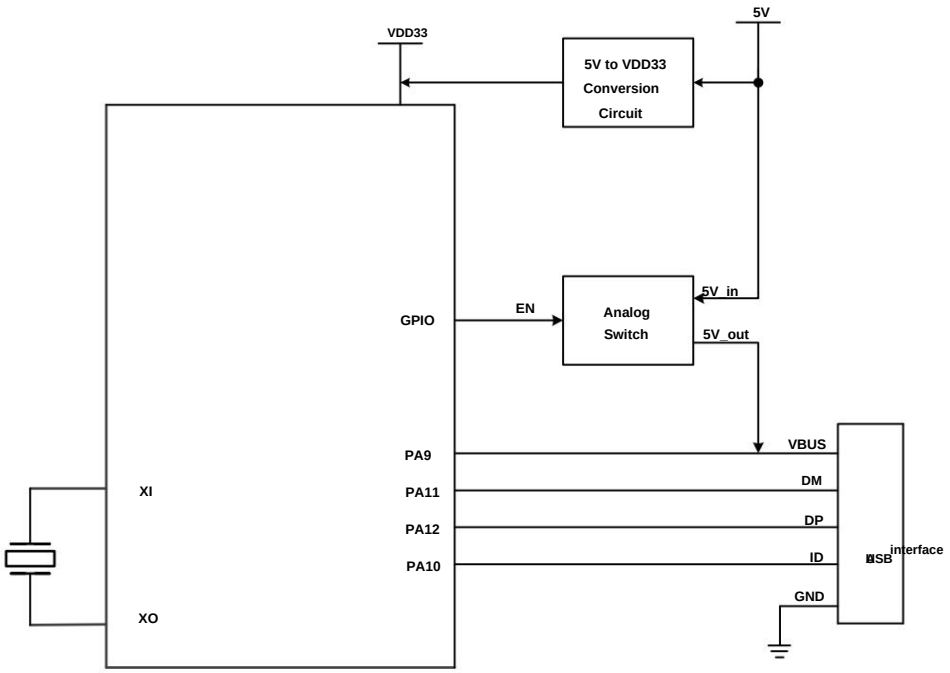
In USB OTG host mode, the chip provides a set of bidirectional host endpoints, including a transmit endpoint (OUT) and a receive endpoint.

IN, the maximum length of a data packet is 1023 bytes, and it supports control transmission, interrupt transmission, bulk transmission and real-time/synchronous transmission.

In USB OTG host mode, if the controller cannot provide 5V power to the VUBS, an external charge pump is required. If the application board can provide this power...

The 5V power supply can be controlled by an analog switch to turn VBUS on and off, as shown in Figure 26-2.

Figure 26-2 OTG Class A device connection



Each USB transaction initiated by the host endpoint automatically sets the RB_UIF_TRANSFER interrupt flag upon completion. Application

The program can directly query or query and analyze the interrupt flag register R8_USB_INT_FG in the USB interrupt service routine, based on each interrupt.

The flags are processed accordingly; furthermore, if RB_UIF_TRANSFER is valid, the USB interrupt status register needs to be analyzed.

The register R8_USB_INT_ST performs corresponding processing based on the PID identifier MASK_UIS_H_RES of the current USB transfer transaction response.

If the synchronization trigger bit (RB_UH_R_TOG) for the IN transaction on the host receiver endpoint is set in advance, then it can be achieved through...

RB_U_TOG_OK or RB_UIS_TOG_OK determines whether the synchronization trigger bit of the currently received data packet is synchronized with the host receiver endpoint.

A step triggers a bit match; if the data is synchronized, it is valid; if it is out of sync, it should be discarded. This process is repeated after each USB step.

After a transmission or reception interruption, the synchronization trigger bit of the corresponding host endpoint should be correctly modified to synchronize the next transmitted data packet and the received data.

Test whether the next received data packet is synchronized; additionally, this can be achieved by setting RB_UH_T_AUTO_TOG and RB_UH_R_AUTO_TOG.

Now, the corresponding synchronization trigger bit will automatically flip after a successful transmission or reception.

The USB host token setting register R8_UH_EP_PID is used to set the endpoint number of the target device being operated and the current USB transfer event.

The PID packet identifier is the token for the service. The data corresponding to the SETUP and OUT tokens is provided by the host sending endpoint and is the data prepared for transmission.

In the R32_UH_TX_DMA buffer, the length of the data to be sent is set in R16_UH_TX_LEN; the data corresponding to the IN token is...

The target device returns the received data to the host receiving endpoint. The received data is stored in the R32_UH_RX_DMA buffer, and the length of the received data is stored in [the buffer name is missing].

In R16_USB_RX_LEN, the maximum packet length that the host endpoint can receive needs to be written into the R16_UH_RX_MAX_LEN register in advance.

middle.

Table 26-5 List of Host-Related Registers

name	Access address description		Reset value
R8_UHOST_CTRL	0x40023401	USB host physical port control register	0xX0
R32_UH_EP_MOD	0x4002340D	USB Host Endpoint Mode Control Register	0x00000000
R32_UH_RX_DMA	0x40023418	USB host receive buffer start address	X
R32_UH_TX_DMA	0x4002341C	USB host transmit buffer start address	X
R16_UH_SETUP	0x40023436	USB Host Auxiliary Configuration Register	0x0000

R8_UH_EP_PID	0x40023438	USB Host Token Setting Register	0x00
R8_UH_RX_CTRL	0x4002343B	USB Host Receive Endpoint Control Register	0x00
R16_UH_TX_LEN	0x4002343C	USB host transmit length register	X
R8_UH_TX_CTRL	0x4002343E	USB Host Transmit Endpoint Control Register	0x00

23.2.3.1 USB Host Physical Port Control Register (R8_UHOST_CTRL) Offset Address: Bit

0x01

	name	access		Reset value
7	RB_UH_PD_DIS	RW	Description of the internal pull-down power supply of the USB host port UD+/UD- pins Restriction control bit: 1: Disable internal dropdown; 0: Enables the internal dropdown. It can be used to provide pull-down resistors in GPIO mode.	1
6	Reserved	RO is reserved.		0
5	RB_UH_DP_PIN	RO	Current UD+ pin status: 1: High level; 0: Low level.	x
4	RB_UH_DM_PIN	RO	Current UD pin status: 1: High level; 0: Low level.	x
3	Reserved	RO is reserved.		0
2	RB_UH_LOW_SPEED	RW	USB host port low-speed mode enable bit: 1: Select 1.5Mbps low-speed mode; 0: Select 12Mbps full speed mode.	0
1	RB_UH_BUS_RST	RW	USB host mode bus reset control bit: 1: Force USB bus reset; 0: End output.	0
0	RB_UH_PORT_EN	RW	USB host port enable bit: 1: Enable the host port; 0: Disable host port. This bit is automatically cleared to 0 when the USB device is disconnected.	0

23.2.3.2 USB Host Endpoint Mode Control Register (R32_UH_EP_MOD) Offset Address:

0x0D

Bit	name	access	describe	Reset value
7	Reserved	RO Reserved.		0
6	RB_UH_EP_TX_EN	RW	Host transmit endpoint transmit (SETUP/OUT) enable bit: 1: Enable endpoint sending; 0: Endpoint sending is disabled.	0
5	Reserved	RO is reserved.		0
4	RB_UH_EP_TBUF_MOD	RW	Host sending endpoint transmits data buffer mode control. Bit.	0
3	RB_UH_EP_RX_EN	RW	Host receive endpoint receive (IN) enable bit: 1: Enable endpoint reception;	0

			0: Endpoint reception is disabled.	
[2:1] Reserved		RO is reserved.		0
0	RB_UH_EP_RBUF_MOD	RW	USB host receiver endpoint receive data buffer mode Control bit.	0

The combination of RB_UH_EP_TX_EN and RB_UH_EP_TBUF_MOD controls the host's endpoint data buffer mode, as shown in the table below.

Table 26-6 Host Transmit Buffer Mode

RB_UH_EP_TX_EN	RB_UH_EP_TBUF_MOD	Description: Starting at address R32_UH_TX_DMA
0	X	The endpoint is disabled and the R32_UH_TX_DMA buffer is not being used.
1	0	Single 64-byte send buffer (SETUP/OUT).
1	1	Dual 64-byte send buffers, selected via RB_UH_T_TOG: When RB_UH_T_TOG=0, the first 64 bytes of buffer are selected; When RB_UH_T_TOG=1, the last 64 bytes of buffer are selected.

The host receive endpoint data buffer mode is controlled by the combination of RB_UH_EP_RX_EN and RB_UH_EP_RBUF_MOD, as shown in the table below.

Table 26-7 Host Receive Buffer Mode

RB_UH_EP_RX_EN	RB_UH_EP_RBUF_MOD Structure	Description: Starting at address R32_UH_TX_DMA
0	X	The endpoint is disabled and the R32_UH_RX_DMA buffer is not used.
1	0	Single 64-byte receive buffer (IN).
1	1	Dual 64-byte receive buffers, selected via RB_UH_R_TOG: When RB_UH_R_TOG=0, the first 64 bytes of buffer are selected; When RB_UH_R_TOG=1, the last 64 bytes of buffer are selected.

26.2.3.3 USB host receive buffer start address (R32_UH_RX_DMA) offset address: 0x18

Bit	name	access	describe	Reset value
[31:21] Reserved		RO is reserved.		0
[20:0] R32_UH_RX_DMA		RW	The starting address of the host endpoint data receive buffer. The lower 20 bits are valid, and the address must be 4-byte aligned.	X

26.2.3.4 USB Host Transmit Buffer Start Address (R32_UH_TX_DMA)

Offset address: 0x1C

Bit name		access	describe	Reset value
[31:21] Reserved		RO is reserved.		0
[20:0] R32_UH_TX_DMA		RW	The starting address of the host endpoint data transmission buffer. The lower 20 bits are valid, and the address must be 4-byte aligned.	X

26.2.3.5 USB Host Auxiliary Setup Register (R16_UH_SETUP) Offset Address: 0x36

bits

	name	access	describe	Reset value
[15:11] Reserved		RO reserved. Low-		0
10	RB_UH_PRE_PID_EN	RW	speed preamble PRE PID enable bit: 1: Enable for use with low-speed USB via external hub.	0

			Device communication; 0: Disable low-speed preamble.	
[9:3]	Reserved	RO reserved.		0
2	RB_UH_SOF_EN	RW	Automatically generate SOF packet enable bit: 1: The host automatically generates SOF packets; 0: Disable automatic SOF function.	0
[1:0]	Reserved	RO is reserved.		0

23.2.3.6 USB Host Token Setting Register (R8_UH_EP_PID) Offset Address: 0x38

bits

	name	Access description reset value	
[7:4] MASK_UH_TOKEN		RW sets the token PID identifier for this USB transfer transaction. 0 RW sets the	
[3:0] MASK_UH_ENDP		endpoint number of the target device being operated on. 0	

26.2.3.7 USB Host Receive Endpoint Control Register (R8_UH_RX_CTRL)

Offset address: 0x3B

Bit	name	access	describe	Reset value
[7:4] Reserved		RO reserved.		0
3	RB_UH_R_AUTO_TOG	RW	Synchronous trigger bit auto-flip enable control bit: 1: Automatically flip the corresponding expectation after successful data reception. Synchronization trigger bit (RB_UH_R_TOG); 0: Manually control the synchronization trigger bit (RB_UH_R_TOG).	0
2	RB_UH_R_TOG	RW	Synchronization prepared by the host receiver (handling IN transactions) Trigger bit: 1: No response, used for real-time/synchronization of non-zero endpoints. transmission; 0: ACK response.	0
1	Reserved	RO is reserved.		0
0	RB_UH_R_RES	RO	Host receiver response control bits for IN transactions: 1: No response, used for real-time/synchronization of non-zero endpoints. transmission; 0: ACK response.	0

26.2.3.8 USB Host Transmit Length Register (R16_UH_TX_LEN) Offset Address: 0x3C

Bit	name	access	describe	Reset value
[15:0] R16_UH_TX_LEN		RW	Configure the USB host transmit endpoint to prepare data for transmission. Number of bytes.	X

26.2.3.9 USB Host Transmit Endpoint Control Register (R8_UH_TX_CTRL)

Offset address: 0x3E

Bit	name	access	describe	Reset value
[7:4] Reserved		RO is reserved.		0
3	RB_UH_T_AUTO_TOG	RO Synchronous Trigger Bit Auto-Flip Enable Control Bit:		0

			1: Automatically reverse the corresponding synchronization after successful data transmission. Trigger bit (RB_UH_T_TOG); 0: Manually control the synchronization trigger bit (RB_UH_T_TOG).	
2	RB_UH_T_TOG	RW	USB host transmitter (handles SETUP/OUT transactions) Prepared synchronization trigger bit: 1: Indicates sending DATA1; 0: Indicates that DATA0 is being sent.	0
1	Reserved	RO is reserved.		0
0	RB_UH_T_RES	RW	USB host transmitter response to SETUP/OUT transactions Control bit: 1: No response expected, for real-time/non-zero endpoints Synchronous transmission; 0: Expected response ACK.	0

Chapter 27 USB 3.0 Ultra-High Speed Host/Device Controller (USBSS)

The module descriptions in this chapter apply only to CH32H417 and CH32H416 microcontroller products.

The USB 3.0 UltraSpeed Controller functions as both a host controller and a device controller, and features a built-in ultra-high-speed USB PHY physical layer. The transducer enables USB 3.0 interface products to function and supports 5Gbps USBSS ultra-high-speed signals.

The USBSS module provides a link-layer register access interface for application code, used to manage device connection and disconnection, bus status, Power mode. Provides host (HOST) and device (DEVICE) access interfaces for implementing the USB 3.0 protocol.

The protocol specifies various data transmissions and upper-layer protocols.

USBSS module peripheral base address: 0x40034000

27.1 Main Features

Supports USB 3.0 protocol specifications and USB 3.2 Gen1 .

Supports USB Host and USB Device functions.

ÿ Power management modes support low-power states for U1/U2/U3

Supports driving USB 3.0 hubs ; supports

control transfer, bulk transfer, interrupt transfer, and real-time/synchronous transfer.

Non -zero endpoints support packets up to 1024 bytes in size and burst mode.

Supports direct access to data in each endpoint buffer via DMA .

ÿ Self-developed controller and transceiver, high-speed integrated design, measured at 450 Mbytes per second.

27.2 Register Description

The USBSS related registers are divided into 4 parts:

USBSS Global Register

USBSS Device Register

USBSS Host Control Register

ÿ USBSS LINK Control Register

27.2.1 Global Register Description

Table 27-1 USBSS Global Register List

name	Access address	describe	Reset value
R32_USBSS_CTRL	0x40034070	USBSS Control Register	0x00000006
R32_USBSS_STATUS	0x40034074	USBSS Status Register	0x00000000
R32_UH_TX_DMA R32_UEP0_TX_DMA	0x4003408C	Transmit Buffer Address Register	0xXXXXXXXX
R32_UH_RX_DMA R32_UEP0_RX_DMA	0x40034090	Receive Buffer Address Register	0xXXXXXXXX

27.2.1.1 USBSS Control Register (R32_USBSS_CTRL) Offset Address: 0x70

Name

Bit		access	describe	Reset value
31	Reserved	RO	reserved.	0
[30:24]	RB_DEV_ADDR	RW	Host mode: The address of the currently operating USB device;	0

			Device mode: The address of the USB device.	
twenty three	RB_UIE_FIFO_RXOV	RW	Receive FIFO overflow interrupt enable: 1: Enable; 0: Off.	0
twenty two	RB_UIE_FIFO_TXOV	RW	Send FIFO overflow interrupt enable: 1: Enable; 0: Off.	0
twenty one	Reserved	RO reserved.		0
20	RB_UIE_ITP	RW	Send ITP to complete interrupt enable: 1: Enable; 0: Off.	0
19	RB_UIE_RX_PING	RW	Receive PING-TP to complete interrupt enable: 1: Enable; 0: Off.	0
18	RB_UDIE_STATUS	RW	Device mode: Receive STATUS Transaction completion interrupt enabled: 1: Enable; 0: Off.	0
18	RB_UHIE_NOTIF	RW	Master mode: Receive DEV_NOTIF-TP completion interrupt. able: 1: Enable; 0: Off.	0
17	RB_UDIE_SETUP	RW	Device mode: Receive SETUP transaction completion interrupt enable: 1: Enable; 0: Off.	0
17	RB_UHIE_ERDY	RW	Master mode: Receive SETUP transaction completion interrupt enable: 1: Enable; 0: Off.	0
16	RB_UIE_TRANSFER	RW	USB transaction completion interrupt enabled: 1: Enable; 0: Off.	0
15	Reserved	RO is reserved.		0
14	RB_TX_ERDY_MODE	RW	Used in device mode: 1: Do not insert ERDY-TP between two DPH packets; 0: Allows ERDY-TP to be inserted between two DPH packets.	0
[13:10] Reserved		RO Reserved.		0
[9:8]	RB_REG_HP_PEND[1 :0]	RW	Packet Pending control bits for sending TP/DP packets: In burst mode, the least significant bit represents the PP bit of the last packet. The status, with the high-order bits indicating the PP bit status of non-last packets. In non-emergency mode, only low-level effective	0
7	RB_HOST_MODE	RW	USB working mode selection bit: 1: Host mode; 0: Device mode.	0
6	RB_ITP_EN	RW	In host mode, ITP transmission enabled: 1: Enable;	0

			0: Off.	
5	RB_SETUP_FLOW	RW	SETUP Transaction Flow Control 1. Upon receiving the SETUP transaction, flow control is initiated, requiring configuration. Prepare; 0: No flow control is required after receiving a SETUP transaction.	0
4	Reserved	RO is reserved.		0
3	RB_DMA_MODE	RW	1. When performing burst transmission DPH, do not prefetch the next data packet. Enter the FIFO; 0: When performing a burst transfer DPH, prefetch the next data packet. in FIFO. Note: Enabling prefetching can handle sudden events. DPH Minimize packet spacing to increase transmission bandwidth; default prefetching.	0
2	RB_FORCE_RST	RW	Protocol layer and FIFO module reset requires software clearing: 1: Reset; 0: Do not reset.	1
1	RB_USB_CLR_ALL	RW	Reset all software configuration registers, active high, requires software... Clear all items; Clear all interrupt flags, clear device addresses, clear all interrupt flags. Configuration with endpoints. the remarks setting mode, it should be received Set the bit after HOT_RESET/WARM_RESET, then clear it. zero.	1
0	RB_DMA_EN	RW	Enable DMA: 1: Enabled, the controller can access data in SRAM; 0: Disabled. The controller cannot access data in SRAM.	0

27.2.1.2 USBSS Status Register (R32_USBSS_STATUS) Offset Address: 0x74

Bit	name	access	describe	Reset value
[31:30]	RB_HRX_RES [1:0]	RW	Received a response TP 00: ACK-TP from the device; 01: RX_FAILED; 10: NRDY-TP; 11: STALL-TP. Note: TX_FAILED/RX_FAILED means that when the host initiates... During a transaction, the HUB's downlink port is in a low-power state, and the HUB will... Mark the packet as deferred and return the host information; otherwise, it is not valid. The step transaction host should wait until it receives the device's ERDY-TP. The transaction will then be repeated.	0
[29:24] Reserved		RO reserved.		0
[23:22]	RB_HTX_RES [1:0]	RW	Received a response TP from the device: 00: ACK-TP; 01: TX_FAILED; 10: NRDY-TP; 11: STALL-TP.	0

			Note: TX_FAILED/RX_FAILED means that when the host initiates... During a transaction, the HUB's downlink port is in a low-power state, and the HUB will... Mark the packet as deferred and return the host information; otherwise, it is not valid. The step transaction host should wait until it receives the device's ERDY-TP. The transaction will then be repeated.	
twenty one	Reserved	RO is reserved. This bit		0
[20:16] RB_HOST_ACK_NUMP		RW	is invalid during synchronous transmissions, but not during control transmissions or block transmissions. The following definitions apply to input and interrupt transmission: After sending DPH+DPP (OUT/SETUP transaction), receive The NUMP field in ACK-TP indicates the device. The number of packets that this endpoint can receive next, if NUMP=0 indicates that the endpoint can no longer receive DPP and will wait until... This endpoint of the device can receive DPH+DPP, and the device will send... The ERDY-TP packet notifies the host. However, this does not affect the host's ability to send packets to the host at this time. Other endpoints send data. Specifically, if the ACK-TP response of the SETUP transaction... A NUMP value of 0 indicates that the system has entered flow control mode, and the host should... Pause the data phase or state phase until a data source is received. The device's ERDY-TP. Additionally, the host can send SETUP events. The task is to lift the flow control status. Of course, when in flow control status, the main... Machines can also perform data phases and state phases, and will not An error occurs simply because that endpoint of the device is not ready. It will then return NRDY-TP.	0
[15:13] Reserved		RO is reserved. The		0
12	RB_EP_DIR	RW	direction of the endpoint currently marked with a transmission completion interruption flag, such as... If an interrupt flag is present during both transmission and reception, then this bit will be 1, indicating priority. Process the transmission first. (SETUP/STATUS independent interrupt) 1: Send (IN); 0: Receive (OUT).	0
11	Reserved	RO is reserved.		0
[10:8] RB_EP_ID		RW	The endpoint number currently has a transmission completion interruption flag. If there are multiple endpoints... If multiple endpoints have interrupt flags simultaneously, priority is given to endpoint type. The order of transmission levels is as follows (high -> low): Synchronization transmission > Control transmission > For block transfers/interrupt transfers, the smaller the endpoint number for the same endpoint type... The higher the priority, the better.	0
7	RB_UIF_FIFO_RXOV RW1Z		Receive FIFO overflow interrupt flag: 1: Interrupts are enabled; 0: Interrupts are not enabled.	0
6	RB_UIF_FIFO_TXOV RW1Z		Send FIFO overflow interrupt flag: 1: Interrupts are enabled; 0: Interrupts are not enabled.	0
5	Reserved	RO reserved. ITP		0
4	RB_UIF_ITP	RW1Z	transmission complete interrupt flag: 1: Interrupts are enabled; 0: Interrupts are not enabled.	0

3	RB_UIF_RX_PING	RW1Z	Receive PING-TP completion interrupt flag: 1: Interrupts are enabled; 0: Interrupts are not enabled.	0
2	RB_UHIF_NOTIF	RW1Z	Master mode: Receive DEV_NOTIF-TP completion interrupt flag Bit: 1: Interrupts are enabled; 0: Interrupts are not enabled.	
2	RB_UDIF_STATUS	RW1Z	Device mode: Receive STATUS transaction completion interrupt flag Bit: 1: Interrupts are enabled; 0: Interrupts are not enabled.	0
1	RB_UHIF_ERDY	RW1Z	Master mode: Receive ERDY-TP completion interrupt flag: 1: Interrupts are enabled; 0: Interrupts are not enabled.	
1	RB_UDIF_SETUP	RW1Z	Device mode: Receive SETUP transaction completion interrupt flag: 1: Interrupts are enabled; 0: Interrupts are not enabled.	0
0	RB_UIF_TRANSFER	RW1Z	USB transaction completion interruption flag: 1: Interrupts are enabled; 0: Interrupts are not enabled.	0

27.2.1.3 Transmit Buffer Address Register (R32_UH_TX_DMA/U3EP0_TX_DMA) Offset Address: 0x8C Name

Bit		access	describe	Reset value
[31:0] U3EP0_TX_DMA		RW	Host mode: The starting address of the host's transmit buffer. Device mode: Endpoint 0: Start address of the send buffer.	X

27.2.1.4 Receive Buffer Address Register (R32_UH_RX_DMA/U3EP0_RX_DMA) Offset Address: 0x90 Name

Bit		access	describe	Reset value
[31:0] U3EP0_RX_DMA		RW	Host mode: The starting address of the host receive buffer. Device mode: Endpoint 0: Start address of the receive buffer.	X

27.2.2 Device Register Description

Table 27-2 USBSS Device Register List

name	Access address	describe	Reset value
R32_USBSS_ITP	0x40034078	Interval register 0x00000000 in USB ITP packets	
R32_USBSS_ITP_ADJ	0x4003407C	The ITP packet interval in USB is adaptive to 0x00000000	

		register	
R16_UEP_TX_EN	0x40034080	Endpoint transmit enable register;	0x00000000
R16_UEP_RX_EN	0x40034082	Endpoint receive enable register;	0x00000000
R32_UEP0_TX_CTRL	0x40034084	Endpoint 0 transmit control register;	0x00000000
R32_UEP0_RX_CTRL	0x40034088	Endpoint 0 receive control register	0x00000000

27.2.2.1 **Offset address of the Interval register (R32_USBSS_ITP) for the ITP packet in USB :**

0x78

Bit	name	access	describe	Reset value
[31:14] Reserved		RO is reserved.		0
[13:0] REG_ITP_INTERVAL		RO	The Bus interval Counter field in the received ITP. Note: The Bus interval Counter increments every 125µs.	0

27.2.2.2 Interval Adaptive Register (R32_USBSS_ITP_ADJ) for ITP Packets in USB

Offset address: 0x7C

Bit	name	access	describe	Reset value
[31:21] Reserved		RO is reserved.		0
[20:8] ITP_DELTA		RO	The high 13 bits of ITS in the ITP received in device mode. (Delta) represents the distance from the beginning of the current ITP packet to the previous one. The time difference between bus interval boundaries. If the host is precise The host should send an ITP every 125µs and set this bit. It is 0.	0
7	ITP_DELAYED	RO	In device mode, the Delayed bit of the Link Control Word in the received ITP is [value missing] when the ITP is transmitted with a delay. 1.	0
[6:0]	ITP_ADJ_CR	RO	In device mode, the Bus Interval in the received ITP The Adjustment Control field allows for power-on reset or setting. After the backup is disconnected, this field in the ITP sent by the host should be [value]. 0.	0

27.2.2.3 Endpoint transmit enable register (R16_UEP_TX_EN) offset

address: 0x80 Name

Bit		access		Reset value
[15:1] RB_EP_TX_EN		RW	Description of upload enable for endpoints 1-15: 1: Enable; 0: Prohibited.	0
0	Reserved	RO is reserved.		1

27.2.2.4 Endpoint Receive Enable Register (R16_UEP_RX_EN) Offset

Address: 0x82

Bit	name	access	describe	Reset value
[15:1] RB_EP_RX_EN		RW	Downlink enable for endpoints 1-15: 1: Enable; 0: Prohibited.	0

0	Reserved	RO is reserved.		1

27.2.2.5 Endpoint 0 Transmit Control Register (R32_UEP0_TX_CTRL)

Offset address: 0x84

Bit	Name	access	The	Reset value
31	RB_UIF_EP0_TX_AC T	RW	upload transaction completion interruption flag is cleared by writing 0 in the software. Hardware set to 1.	0
[20:26] Reserved		RO is reserved.		0
25	RB_EP0_TX_FLOW	RO is a flag indicating that NRDY-TP transmission (response) is complete.		0
twenty four	RB_EP0_TX_PP	The PP bit in the ACK-TP received by RO. Writing this bit		0
twenty three	RB_EP0_TX_ERDY	RW	to 1 triggers the transmission of ERDY; the bit is reset after ERDY transmission is complete. The hardware is automatically reset.	0
[22:21]	RB_EP0_TX_RES[1: 0]	RW	Response to ACK-TP (numpý0): 00: NRDY-TP; 01: DPH; 10: STALL-TP; 11: No response (invalid). IN is automatically cleared after the transaction is completed.	0
[20:16]	RB_EP0_TX_SEQ[4: 0]	RW	The current sequence number of the endpoint; after receiving the SETUP transaction, this... The digits are automatically cleared to zero.	0
[15:11] Reserved		RO is reserved.		0
[10:0]	RB_EP0_TX_LEN[10 :0]	The RW endpoint transmit length register has a maximum value of 512 bytes.		x

27.2.2.6 Endpoint 0 Receive Control Register (R32_UEP0_RX_CTRL)

Offset address: 0x88

Bit	name	access	describe	Reset value
31	RB_UIF_EP0_RX_AC T	RW	The transaction completion interruption flag is cleared by writing 0 in the software. Hardware setting 1.	0
[30:25] Reserved		RO is reserved.		0
twenty four	RB_EP0_RX_PP	The PP site in the DPH received by RO.		0
twenty three	RB_EP0_RX_ERDY	RW	Write 1 to this bit, send ERDY, and reset this bit after ERDY is sent. The hardware is automatically reset.	0
[22:21]	RB_EP0_RX_RES[1: 0]	RW	Response to DPH: 00: NRDY-TP; 01: ACK-TP; 10: STALL-TP; 11: No response (invalid). EPn_R_RES is automatically cleared after the SETUP/OUT transaction is completed; In addition, the SETUP packet directly responds with ACK-TP, which is unrelated to this.	0
[20:16]	RB_EP0_RX_SEQ[4: 0]	The RW endpoint expects to receive the sequence number.		0
[15:11] Reserved		RO is reserved.		0
[10:0] RB_EP0_RX_LEN[10		RO endpoint receive length register.		x

	:0]			

27.2.3 Device Registers -- Endpoints 1~7 Transmit Related Register Table

27-3 Endpoint 1 Receive Related Register List

name	Access Address	Description	Reset value
R8_UEP1_TX_CFG		Endpoint 1 Configuration Register	0x86
R8_UEP1_TX_CR		Endpoint 1 Control Register	0x10
R8_UEP1_TX_SEQ		Endpoint 1 Serial Number Register	0x00
R8_UEP1_TX_ST		Endpoint 1 Status Register	0x00
R8_UEP1_TX_CHAIN_CR		Endpoint 1 CHAIN Control Register	0x00
R8_UEP1_TX_CHAIN_ST		Endpoint 1 CHAIN Status Register	0x00
R16_UEP1_TX_CHAIN_LEN		Endpoint 1 CHAIN Length of Last Packet Sent	0x00
R8_UEP1_TX_CHAIN_EXP_NUMP	0x400340C8	Endpoint 1 Expected number of NUMPs to be sent	0x00
R8_UEP1_TX_CHAIN_NUMP	0x400340C9	Endpoint 1 Number of NUMPs already sent	0x00
R16_UEP1_TX_DMA_OFS	0x400340CA	DMA offset length of endpoint 1	0x00
R32_UEP1_TX_DMA	0x400340CC	DMA start address for endpoint 1	0x0400

Table 27-4 Endpoint 2 Receive Related Register List

name	Access Address	Description	Reset value
R8_UEP2_TX_CFG		Endpoint 2 Configuration Register	0x86
R8_UEP2_TX_CR		Endpoint 2 Control Register	0x10
R8_UEP2_TX_SEQ		Endpoint 2 Serial Number Register	0x00
R8_UEP2_TX_ST		Endpoint 2 Status Register	0x00
R8_UEP2_TX_CHAIN_CR		Endpoint 2 CHAIN Control Register	0x00
R8_UEP2_TX_CHAIN_ST		Endpoint 2 CHAIN Status Register	0x00
R16_UEP2_TX_CHAIN_LEN		Endpoint 2 CHAIN Length of Last Packet Sent	0x00
R8_UEP2_TX_CHAIN_EXP_NUMP	0x400340E8	Endpoint 2 Expected number of NUMPs to be sent	0x00
R8_UEP2_TX_CHAIN_NUMP	0x400340E9	Endpoint 2 Number of NUMPs already sent	0x00
R16_UEP2_TX_DMA_OFS	0x400340EA	DMA offset length of endpoint 2	0x00
R32_UEP2_TX_DMA	0x400340EC	DMA start address for endpoint 2	0x0400

Table 27-5 List of Receive-Related Registers for Endpoint 3

name	Access Address	Description	Reset value
R8_UEP3_TX_CFG		Endpoint 3 Configuration Register	0x86
R8_UEP3_TX_CR		Endpoint 3 Control Register	0x10
R8_UEP3_TX_SEQ		Endpoint 3 Serial Number Register	0x00
R8_UEP3_TX_ST		Endpoint 3 Status Register	0x00
R8_UEP3_TX_CHAIN_CR		Endpoint 3 CHAIN Control Register	0x00
R8_UEP3_TX_CHAIN_ST		Endpoint 3 CHAIN Status Register	0x00
R16_UEP3_TX_CHAIN_LEN		3 CHAIN Length of Last Packet Sent	0x00
R8_UEP3_TX_CHAIN_EXP_NUMP	0x40034108	Endpoint 3 Expected number of NUMPs to be sent	0x00
R8_UEP3_TX_CHAIN_NUMP	0x40034109	Endpoint 3 Number of NUMPs already sent	0x00
R16_UEP3_TX_DMA_OFS	0x4003410A	DMA offset length of endpoint 3	0x00

R32_UEP3_TX_DMA	DMA start address of endpoint 3 at 0x4003410C	0x0400

Table 27-6 Endpoint 4 Receive Related Register List

name	Access address	describe	Reset value
R8_UEP4_TX_CFG	0x40034120	Endpoint 4 Configuration	0x86
R8_UEP4_TX_CR	Register; 0x40034121	Endpoint 4 Control	0x10
R8_UEP4_TX_SEQ	Register; 0x40034122	Endpoint 4 Serial Number	0x00
R8_UEP4_TX_ST	Register; 0x40034123	Endpoint 4 Status	0x00
R8_UEP4_TX_CHAIN_CR	Register; 0x40034124	Endpoint 4 CHAIN Control	0x00
R8_UEP4_TX_CHAIN_ST	Register; 0x40034125	Endpoint 4 CHAIN Status	0x00
R16_UEP4_TX_CHAIN_LEN	Register; 0x40034126	Endpoint 4 CHAIN Length of Last Packet Sent; 0x00	
R8_UEP4_TX_CHAIN_EXP_NUMP	0x40034128	Endpoint 4 Expected number of NUMPs to be sent	0x00
R8_UEP4_TX_CHAIN_NUMP	0x40034129	Endpoint 4 Number of NUMPs already sent	0x00
R16_UEP4_TX_DMA_OFS	0x4003412A	DMA offset length of endpoint 4	0x00
R32_UEP4_TX_DMA	0x4003412C	DMA start address of endpoint 4	0x0400

Table 27-7 Endpoint 5 Receive-Related Register List

name	Access address	description	Reset value
R8_UEP5_TX_CFG	0x40034140	Endpoint 5 Configuration	0x86
R8_UEP5_TX_CR	Register; 0x40034141	Endpoint 5 Control	0x10
R8_UEP5_TX_SEQ	Register; 0x40034142	Endpoint 5 Serial Number	0x00
R8_UEP5_TX_ST	Register; 0x40034143	Endpoint 5 Status	0x00
R8_UEP5_TX_CHAIN_CR	Register; 0x40034144	Endpoint 5 CHAIN Control	0x00
R8_UEP5_TX_CHAIN_ST	Register; 0x40034145	Endpoint 5 CHAIN Status	0x00
R16_UEP5_TX_CHAIN_LEN	Register; 0x40034146	Endpoint 5 CHAIN Length of Last Packet Sent; 0x00	
R8_UEP5_TX_CHAIN_EXP_NUMP	0x40034148	Endpoint 5 Expected number of NUMPs to be sent	0x00
R8_UEP5_TX_CHAIN_NUMP	0x40034149	Endpoint 5 Number of NUMPs already sent	0x00
R16_UEP5_TX_DMA_OFS	0x4003414A	DMA offset length of endpoint 5	0x00
R32_UEP5_TX_DMA	0x4003414C	DMA start address of endpoint 5	0x0400

Table 27-8 List of Receive-Related Registers for Endpoint 6

name	Access Address	Description 0x40034160	Reset value
R8_UEP6_TX_CFG	Endpoint 6 Configuration Register 0x40034161		0x86
R8_UEP6_TX_CR	Endpoint 6 Control Register 0x40034162		0x10
R8_UEP6_TX_SEQ	Endpoint 6 Serial Number Register 0x40034163		0x00
R8_UEP6_TX_ST	Endpoint 6 Status Register 0x40034164		0x00
R8_UEP6_TX_CHAIN_CR	Endpoint 6 CHAIN Control Register 0x40034165		0x00
R8_UEP6_TX_CHAIN_ST	Endpoint 6 CHAIN Status Register 0x40034166 Endpoint		0x00
R16_UEP6_TX_CHAIN_LEN	6 CHAIN Length of Last Packet Sent 0x00		
R8_UEP6_TX_CHAIN_EXP_NUMP	0x40034168	Endpoint 6 Expected number of NUMPs to be sent	0x00
R8_UEP6_TX_CHAIN_NUMP	0x40034169	Endpoint 6 Number of NUMPs already sent	0x00
R16_UEP6_TX_DMA_OFS	0x4003416A	DMA offset length of endpoint 6	0x00
R32_UEP6_TX_DMA	0x4003416C	DMA start address of endpoint 6	0x0400

Table 27-9 Endpoint 7 Receive-Related Register List

name	Access Address Description 0x40034180	Reset value
R8_UEP7_TX_CFG	Endpoint 7 Configuration Register 0x40034181	0x86
R8_UEP7_TX_CR	Endpoint 7 Control Register 0x40034182 Endpoint	0x10
R8_UEP7_TX_SEQ	7 Serial Number Register 0x40034183 Endpoint 7	0x00
R8_UEP7_TX_ST	Status Register 0x40034184 Endpoint 7 CHAIN	0x00
R8_UEP7_TX_CHAIN_CR	Control Register 0x40034185 Endpoint 7 CHAIN Status	0x00
R8_UEP7_TX_CHAIN_ST	Register 0x40034186 Endpoint 7 CHAIN Last Packet Length	0x00
R16_UEP7_TX_CHAIN_LEN	0x00	
R8_UEP7_TX_CHAIN_EXP_NUMP 0x40034188	Endpoint 7 Expected number of NUMPs to be sent 0x40034189	0x00
R8_UEP7_TX_CHAIN_NUMP	Endpoint 7 Number of NUMPs already sent	0x00
R16_UEP7_TX_DMA_OFS	0x4003418A DMA offset length of endpoint 7	0x00
R32_UEP7_TX_DMA	0x4003418C DMA start address for endpoint 7	0x0400

27.2.3.1 Endpoint n Configuration Register (R8_UEPn_TX_CFG) (n=1-7)

Offset address: 0xC0 + (n-1)*0x20 Name access

Bit			describe	Reset value
7	RB_EP_TX_CHAIN_AUTO	RW	CHAIN automatic switching mode is recommended.	1
6	RB_EP_TX_FIFO_MODE	RW	This bit being 1 indicates that the current endpoint is using FIFO mode. Before setting the first bit to 1, the starting address and the last bit of the FIFO should be configured. Bundle address.	0
5	RB_EP_TX_FIFO_CFG	RW	1: Access offset addresses 0xC~0xF, the operation object is UEP_DMA[31:16]/UEP_DMA[15:0]; 0: Access offset address 0xC~0xF, the operation object is UEP_DMA.	0
4	Reserved	RO	is reserved.	0
3	RB_EP_TX_EOB_MODE	RW	1: If a short packet is sent, then EOB/LPF in ACK-TP = 0; 0: If a short packet is sent, then EOB/LPF in the ACK-TP = 1; Note: CHAIN_ST EOB_LPF and the mode setting EOB/LPF is an OR relationship; this bit only applies to short packets, while CHAIN_ST->EOB_LPF applies to the entire CHAIN. The end One package, regardless of its length.	0
2	RB_EP_TX_ERDY_AUTO	RW	In ERDY Auto Mode, the hardware sends ERDY signals; no software is required. Control; it is recommended to clear this bit to the	1
1	RB_EP_TX_SEQ_AUTO	RW	synchronization endpoint. 1: Disable software writing to R16_EPn_ST->EP_SEQ_NUM; 0: Allow software to write R16_EPn_ST->EP_SEQ_NUM.	1
0	RB_EP_TX_ISO_MODE		A value of 1 for RW indicates that the current endpoint is a synchronous endpoint.	0

27.2.3.2 Endpoint n Control Register (R8_UEPn_TX_CR) (n=1-7)

Offset address: 0xC1 + (n-1)*0x20 Name access

Bit			describe	Reset value

7	RB_EP_TX_HALT	RW	Endpoint stop, high active, endpoint stop response to DPH STALL.	0
6	RB_EP_TX_CLR	RW1	Write 1 to clear all configuration values and states of the endpoint, except... UEP_CFG.	0
5	RB_EP_TX_CHAIN_C LR	RW1 write 1 clears all CHAIN configuration values and statuses.		0
[4:0]	RB_EP_TX_ERDY_NU MP	RW	The hardware sends the ERDY-TP nump field, typically configured with the following values. To specify the number of supported bursts; for example, to support 16 burst levels. The field is configured with a value of 16.	0x10

27.2.3.3 Endpoint n Serial Number Register (R8_UEPn_TX_SEQ) (n=1-7)

Offset address: 0xC2 + (n-1)*0x20 Name access

Bit			describe	Reset value
[7:5] Reserved		RO is reserved.		0
[4:0] RB_TX_EP_SEQ_NUM		RW	The current sequence number of the endpoint, in non-SEQ_AUTO mode, is... Write, read-only in SEQ_AUTO mode.	0

27.2.3.4 Endpoint n Status Register (R8_UEPn_TX_ST) (n=1-7)

Offset address: 0xC3 + (n-1)*0x20

Bit	name	access	describe	Reset value
7	RB_TX_EP_INT_FLA G	RO	The endpoint current interrupt flag is read-only; all... If CHAIN_IF is 0, then this bit is 0; otherwise, it is 1.	0
6	RB_TX_EP_FC_ST	The RW1Z endpoint displays the current flow control status; writing 1		0
5	RB_TX_EP_ERDY_RE Q	RW1	clears it. This indicates that it is currently sending an ERDY. Writing 1 to this bit will send ERDY, in non-ERDY_AUTO mode. Used in this formula.	0
4	RB_TX_CHAIN_RES	RO	CHAIN response status, corresponding to 4 independent CHAINs: 1: Response ACK_TP; 0: Response NRDY-TP. Note: In endpoint stop mode (EP_HALT=1), this bit has no effect in the STALL response.	0
[3:0] RB_TX_CHAIN_EN		The RO CHAIN enable state corresponds to 4 independent CHAINs.		0

27.2.3.5 Endpoint n CHAIN Control Register (R8_UEPn_TX_CHAIN_CR) (n=1-7) Offset address: 0xC4 + (n-1)*0x20

Bit	name	access	describe	Reset value
[7:6] RB_TX_CUR_USE		RO is the currently used CHAIN serial number.		0
[5:4] RB_TX_CUR_CFG		RO is the currently configured CHAIN serial number.		0
3	Reserved	RO is reserved. A		0
2	RB_TX_FORCE_RET	RW	valid RO bit forces a return to the selected CHAIN state machine configuration. This	0
[1:0] RB_TX_RET_SEL		RW	bit is set. When FORCE_RET is valid, this bit indicates the returned CHAIN. Status and configuration.	0

27.2.3.6 Endpoint n CHAIN Status Register (R8_UEPn_TX_CHAIN_ST) (n=1-7) Offset

Address: 0xC5 + (n-1)*0x20 Name

Bit	Access			Reset value
7	RB_TX_CHAIN_EN	RO	Describe the currently used CHAIN enable,after configuration. Automatic hardware reset after UEP_CHAIN_NUMP register 1. The CHAIN will be automatically cleared after the transmission is completed.	0
6	RB_TX_CHAIN_IF	WO	This bit is only written to; writing 1 releases the current CHAIN_IF. This interrupt is generated when the current CHAIN's NUMP_EMPTY is 1.	0
5	RB_TX_EOB_LPF	RW	The EOB/LPF position in the last DPH of the current chain: For bulk transfers, after this bit is set to 1, the CHAIN transfer ends. After entering flow control (End Of Block), it will not be connected to the next... CHAIN links are sent together; For synchronous transmission, after this bit is set to 1, the CHAIN transmission ends. After that, the microframe stops sending DPH, indicating that the last packet was for this purpose. The last packet of the microframe.	0
4	Reserved	RO is reserved.		0
3	RB_TX_NUMP_EMPTY	If NUMP in the current CHAIN is 0, then this position is 1.		0
2	RB_TX_DPH_PP	RW represents the current PP bit status in the received DPH.		0
[1:0]	RB_TX_CHAIN_NO	RW is the sequence number of the CHAIN that caused the interrupt.		0

27.2.3.7 Endpoint n CHAIN Send last packet length (R16_UEPn_TX_CHAIN_LEN) (n=1-7) Offset

address: 0xC6 + (n-1)*0x20 Name

Bit	Access		describe	Reset value
[15:11]	Reserved	RO is reserved.		0
[10:0]	CHAIN_TX_LEN	RW indicates the length of the last packet sent by the CHAIN.		0

27.2.3.8 Endpoint n Expected number of NUMPs to be sent (R8_UEPn_TX_CHAIN_EXP_NUMP)

(n=1-7) Offset address: 0xC8 +

Bit	(n-1)*0x20 Name access		describe	Reset value
[7:0]	TX_CHAIN_EXP_NUM P	RW: The number of DPP packets that can be sent from the currently completed CHAIN.		0

27.2.3.9 Endpoint n has sent the number of NUMPs (R8_UEPn_TX_CHAIN_NUMP) (n=1-7)

Offset address: 0xC9 + (n-1)*0x20

Bit	Name Access		Describes	Reset value
[7:0]	TX_CHAIN_NUMP	RW	the number of DPP packets that have been transmitted; for packets where CHIN_EN is... When 0 is reached, writing to UEP_CHAIN_NUMP will cause this register to automatically... Dynamically clear to zero.	0

27.2.3.10 DMA offset length of endpoint n (R16_UEPn_TX_DMA_OFS) (n=1-7) Offset

address: 0xCA + (n-1)*0x20 Name

Bit	Access description [15:0] CHAIN_TX_DMA_OFS RW		Reset value
Offset address of DPP in this CHAIN.			0x0400

27.2.3.11 DMA start address of endpoint n (R32_UEPn_TX_DMA) (n=1-7)

Offset address: 0xCC + (n-1)*0x20 Name

Bit	access		describe	Reset value
[31:0] CHAIN_TX_DMA		RW	In normal mode: The CHAIN is the starting address of the DMA for sending data. In FIFO mode: The FIFO start address is mapped to addresses 16-23 in SRAM. Bit; The FIFO end address is mapped to addresses 16-23 in SRAM. Bit.	0

27.2.4 Device Registers – Endpoints 1~7 Receive-Related Registers Table

27-10 Endpoint 1 Receive-Related Register List

name	Access Address	Description 0x400340DO	Reset value
R8_UEP1_RX_CFG	Endpoint 1 Configuration Register 0x400340D1		0x86
R8_UEP1_RX_CR	Endpoint 1 Control Register 0x400340D2		0x10
R8_UEP1_RX_SEQ	Endpoint 1 Serial Number Register 0x400340D3		0x00
R8_UEP1_RX_ST	Endpoint 1 Status Register 0x400340D4		0x00
R8_UEP1_RX_CHAIN_CR	Endpoint 1 CHAIN Control Register 0x400340D5		0x00
R8_UEP1_RX_CHAIN_ST	Endpoint 1 CHAIN Status Register		0x00
R16_UEP1_RX_CHAIN_LEN	0x400340D6	The last packet received by endpoint 1 CHAIN length	0x00
R8_UEP1_RX_CHAIN_MAX_NUMP	0x400340D8 Endpoint 1 can receive the number of NUMPs.		0x00
R8_UEP1_RX_CHAIN_NUMP	0x400340D9 Number of NUMPs received by endpoint 1		0x00
R16_UEP1_RX_DMA_OFS	0x400340DA DMA offset length of endpoint 1		0x0400
R32_UEP1_RX_DMA	0x400340DC DMA start address of endpoint 1		0x00

Table 27-11 List of Receive-Related Registers for Endpoint 2

name	Access address	describe	Reset value
R8_UEP2_RX_CFG	0x400340FO Endpoint 2 Configuration		0x86
R8_UEP2_RX_CR	Register; 0x400340F1 Endpoint 2 Control		0x10
R8_UEP2_RX_SEQ	Register; 0x400340F2 Endpoint 2 Serial Number		0x00
R8_UEP2_RX_ST	Register; 0x400340F3 Endpoint 2 Status		0x00
R8_UEP2_RX_CHAIN_CR	Register; 0x400340F4 Endpoint 2 CHAIN Control		0x00
R8_UEP2_RX_CHAIN_ST	Register; 0x400340F5 Endpoint 2 CHAIN Status		0x00
R16_UEP2_RX_CHAIN_LEN	0x400340F6	Register; Endpoint 2 CHAIN Last Packet Received length	0x00
R8_UEP2_RX_CHAIN_MAX_NUMP	0x400340F8 Number of NUMPs that endpoint 2 can receive.		0x00
R8_UEP2_RX_CHAIN_NUMP	0x400340F9 Number of NUMPs received by endpoint 2;		0x00
R16_UEP2_RX_DMA_OFS	0x400340FA DMA offset length of endpoint 2;		0x0400
R32_UEP2_RX_DMA	0x400340FC DMA start address of endpoint 2.		0x00

Table 27-12 List of Receive-Related Registers for Endpoint 3

name	Access address	describe	Reset value

R8_UEP3_RX_CFG	0x40034110 Endpoint 3 Configuration		0x86
R8_UEP3_RX_CR	Register; 0x40034111 Endpoint 3 Control		0x10
R8_UEP3_RX_SEQ	Register; 0x40034112 Endpoint 3 Serial Number		0x00
R8_UEP3_RX_ST	Register; 0x40034113 Endpoint 3 Status		0x00
R8_UEP3_RX_CHAIN_CR	Register; 0x40034114 Endpoint 3 CHAIN Control		0x00
R8_UEP3_RX_CHAIN_ST	Register; 0x40034115 Endpoint 3 CHAIN Status		0x00
R16_UEP3_RX_CHAIN_LEN	0x40034116	Register; Endpoint 3 CHAIN Last Packet Received length	0x00
R8_UEP3_RX_CHAIN_MAX_NUMP	0x40034118 Endpoint 3 can receive the number of NUMPs.		0x00
R8_UEP3_RX_CHAIN_NUMP	0x40034119 Number of NUMPs received by endpoint 3;		0x00
R16_UEP3_RX_DMA_OFS	0x4003411A DMA offset length of endpoint 3;		0x0400
R32_UEP3_RX_DMA	0x4003411C DMA start address of endpoint 3.		0x00

Table 27-13 Endpoint 4 Receive Related Register List

name	Access Address	Description 0x40034130	Reset value
R8_UEP4_RX_CFG	Endpoint 4 Configuration Register 0x40034131		0x86
R8_UEP4_RX_CR	Endpoint 4 Control Register 0x40034132		0x10
R8_UEP4_RX_SEQ	Endpoint 4 Serial Number Register 0x40034133		0x00
R8_UEP4_RX_ST	Endpoint 4 Status Register 0x40034134		0x00
R8_UEP4_RX_CHAIN_CR	Endpoint 4 CHAIN Control Register 0x40034135		0x00
R8_UEP4_RX_CHAIN_ST	Endpoint 4 CHAIN Status Register Endpoint 4 CHAIN		0x00
R16_UEP4_RX_CHAIN_LEN	0x40034136	Last Packet Received length	0x00
R8_UEP4_RX_CHAIN_MAX_NUMP	0x40034138 Endpoint 4 can receive the number of NUMPs.		0x00
R8_UEP4_RX_CHAIN_NUMP	0x40034139 Number of NUMPs received by endpoint 4;		0x00
R16_UEP4_RX_DMA_OFS	0x4003413A DMA offset length of endpoint 4;		0x0400
R32_UEP4_RX_DMA	0x4003413C DMA start address of endpoint 4.		0x00

Table 27-14 Endpoint 5 Receive-Related Register List

name	Access address	describe	Reset value
R8_UEP5_RX_CFG	0x40034150 Endpoint 5 Configuration		0x86
R8_UEP5_RX_CR	Register; 0x40034151 Endpoint 5 Control		0x10
R8_UEP5_RX_SEQ	Register; 0x40034152 Endpoint 5 Serial Number		0x00
R8_UEP5_RX_ST	Register; 0x40034153 Endpoint 5 Status		0x00
R8_UEP5_RX_CHAIN_CR	Register; 0x40034154 Endpoint 5 CHAIN Control		0x00
R8_UEP5_RX_CHAIN_ST	Register; 0x40034155 Endpoint 5 CHAIN Status		0x00
R16_UEP5_RX_CHAIN_LEN	0x40034156	Register; Endpoint 5 CHAIN Last Packet Received length	0x00
R8_UEP5_RX_CHAIN_MAX_NUMP	0x40034158 Endpoint 5 can receive the number of NUMPs.		0x00
R8_UEP5_RX_CHAIN_NUMP	0x40034159 Number of NUMPs received at endpoint 5;		0x00
R16_UEP5_RX_DMA_OFS	0x4003415A DMA offset length at endpoint 5;		0x0400
R32_UEP5_RX_DMA	0x4003415C DMA start address at endpoint 5.		0x00

Table 27-15 List of Receive-Related Registers for Endpoint 6

name	Access Address	Description 0x40034170	Reset value
R8_UEP6_RX_CFG		Endpoint 6 Configuration Register 0x40034171	0x86
R8_UEP6_RX_CR		Endpoint 6 Control Register 0x40034172	0x10
R8_UEP6_RX_SEQ		Endpoint 6 Serial Number Register 0x40034173	0x00
R8_UEP6_RX_ST		Endpoint 6 Status Register 0x40034174	0x00
R8_UEP6_RX_CHAIN_CR		Endpoint 6 CHAIN Control Register 0x40034175 Endpoint	0x00
R8_UEP6_RX_CHAIN_ST		6 CHAIN Status Register Endpoint 6 CHAIN Last Packet	0x00
R16_UEP6_RX_CHAIN_LEN	0x40034176	Received length	0x00
R8_UEP6_RX_CHAIN_MAX_NUMP	0x40034178	Endpoint 6 can receive the number of NUMPs.	0x00
R8_UEP6_RX_CHAIN_NUMP		0x40034179 Number of NUMPs received by endpoint 6;	0x00
R16_UEP6_RX_DMA_OFS		0x4003417A DMA offset length of endpoint 6;	0x0400
R32_UEP6_RX_DMA		0x4003417C DMA start address of endpoint 6.	0x00

Table 27-16 Endpoint 7 Receive-Related Register List

name	Access Address	Description 0x40034190	Reset value
R8_UEP7_RX_CFG		Endpoint 7 Configuration Register 0x40034191	0x86
R8_UEP7_RX_CR		Endpoint 7 Control Register 0x40034192	0x10
R8_UEP7_RX_SEQ		Endpoint 7 Serial Number Register 0x40034193	0x00
R8_UEP7_RX_ST		Endpoint 7 Status Register 0x40034194	0x00
R8_UEP7_RX_CHAIN_CR		Endpoint 7 CHAIN Control Register 0x40034195 Endpoint	0x00
R8_UEP7_RX_CHAIN_ST		7 CHAIN Status Register Endpoint 7 CHAIN Last Packet	0x00
R16_UEP7_RX_CHAIN_LEN	0x40034196	Received length	0x00
R8_UEP7_RX_CHAIN_MAX_NUMP	0x40034198	Endpoint 7 can receive the number of NUMPs.	0x00
R8_UEP7_RX_CHAIN_NUMP		0x40034199 Number of NUMPs received by endpoint 7;	0x00
R16_UEP7_RX_DMA_OFS		0x4003419A DMA offset length of endpoint 7;	0x0400
R32_UEP7_RX_DMA		0x4003419C DMA start address of endpoint 7.	0x00

27.2.4.1 Endpoint n Configuration Register (R8_UEPn_RX_CFG) (n=1-7)

Offset address: 0xD0 + (n-1)*0x20 Name

Bit	access		describe	Reset value
7	RB_EP_RX_CHAIN_AUTO	RW	CHAIN automatic switching mode is recommended.	1
6	RB_EP_RX_FIFO_MODE	RW	This bit being 1 indicates that the current endpoint is using FIFO mode. Before setting the first bit to 1, the starting address and the last bit of the FIFO should be configured. Bundle address.	0
5	RB_EP_RX_FIFO_CFG	RW	1: Access offset addresses 0xC~0xF, the operation object is UEP_DMA; 0: Access offset address 0xC~0xF, the operation object is R16_EPn_FIFO_START/R16_EPn_FIFO_DEST.	0
4	RB_EP_RX_TOUT_MODE	RW	This bit is 1; after receiving consecutive burst packets, PP=1 and... It's not a short packet; the DPH wasn't received within the timeout period, resulting in a TOUT_IF error.	0

			Break.	
3	RB_EP_RX_NRDY_MO DE	RW	This bit is 1; it is cleared when a short packet or DPH (PP=0) is received. All CHAIN_EN responses will result in a response to the subsequent DPH. NRDY; requires software reconfiguration.	0
2	RB_EP_RX_ERDY_AU TO	RW	In ERDY Auto Mode, the hardware sends ERDY signals; no software is required. Control; it is recommended to clear this bit to the synchronization endpoint;	1
1	RB_EP_RX_SEQ_AUT O	RW	1: Disable software from writing to R16_EPn_ST->EP_SEQ_NUM; 0: Allow software to write R16_EPn_ST->EP_SEQ_NUM.	1
0	RB_EP_RX_ISO_MOD E	A value of 1 for RW indicates that the current endpoint is a synchronous endpoint.		0

27.2.4.2 Endpoint n Control Register (R8_UEPn_RX_CR) (n=1-7)

Offset address: 0xD1 + (n-1)*0x20 Name access

Bit			describe	Reset value
7	RB_EP_RX_HALT	RW	Endpoint stop, high active, endpoint stop response to DPH STALL.	0
6	RB_EP_RX_CLR	RW1Z	Write 1 to clear all configuration values and states of the endpoint, except... UEP_CFG.	0
5	RB_EP_RX_CHAIN_C LR	RW1Z Write 1: Clears all CHAIN configuration values and statuses.		0
[4:0]	RB_EP_RX_ERDY_NU MP	RW	The hardware sends the ERDY-TP nump field, typically configured with the following values. To specify the number of supported bursts; for example, to support 16 burst levels. The field is configured with a value of 16.	0x10

27.2.4.3 Endpoint n Serial Number Register (R8_UEPn_RX_SEQ) (n=1-7)

Offset address: 0xD2 + (n-1)*0x20 Name access

Bit			describe	Reset value
[7:5] Reserved		RO is reserved.		0
[4:0] RB_EP_RX_SEQ_NUM		RW	The current sequence number of the endpoint, available in non-SEQ_AUTO mode. Write, read-only in SEQ_AUTO mode. Note: Under normal circumstances, this field is automatically controlled by the hardware, and the software does not need to write EP_SEQ_NUM.	0

27.2.4.4 Endpoint n Status Register (R8_UEPn_RX_ST) (n=1-7)

Offset address: 0xD3 + (n-1)*0x20

Bit	name	access	describe	Reset value
7	RB_EP_RX_INT_FLA G	RO	The endpoint current interrupt flag is read-only; all... If CHAIN_IF is 0, then this bit is 0; otherwise, it is 1.	0
6	RB_EP_RX_FC_ST	The RW1Z endpoint displays the current flow control status; writing 1		0
5	RB_EP_RX_ERDY_RE Q	RW1	clears it. This indicates that it is currently sending an ERDY. Writing 1 to this bit will send ERDY, in non-ERDY_AUTO mode. Used in this formula.	0
4	RB_EP_RX_CHAIN_R ES	RO	CHAIN response status, corresponding to 4 independent CHAINs: 1: Response ACK_TP;	0

			0: Response NRDY-TP. Note: In endpoint stop mode (EP_HALT=1), this bit has no effect in the STALL response.	
[3:0]	RB_EP_RX_CHAIN_EN	RO	The CHAIN enable state corresponds to 4 independent CHAINs.	0

27.2.4.5 Endpoint n CHAIN Control Register (R8_UEPn_RX_CHAIN_CR) (n=1-7) Offset Address: 0xD4 + (n-1)*0x20

Name Access

Bit			describe	Reset value
[7:6]	RB_EP_RX_CUR_USE	RO	The current CHAIN serial number being used.	0
[5:4]	RB_EP_RX_CUR_CFG	RO	The currently configured CHAIN serial number.	0
3	Reserved	RO	is reserved. A	0
2	RB_EP_RX_FORCE_RET	RW	valid RO bit forces a return to the selected CHAIN state machine configuration. Place.	0
[1:0]	RB_EP_RX_RET_SEL	RW	When FORCE_RET is valid, this bit indicates the returned CHAIN. Status and configuration.	0

27.2.4.6 Endpoint n CHAIN Status Register (R8_UEPn_RX_CHAIN_ST) (n=1-7) Offset Address: 0xD5 + (n-1)*0x20

Name Access

Bit				Reset value
7	RB_EP_RX_CHAIN_EN	RO	Describe the currently used CHAIN enable after configuration. Automatic hardware reset after UEP_CHAIN_NUMP register 1. The CHAIN will be automatically cleared after the transmission is completed.	0
6	RB_EP_RX_CHAIN_IF	RW1	This bit is only written to; writing 1 releases the current CHAIN_IF. This interrupt is generated if any of the following conditions are met: 1. The length of the currently received packet is less than 1024; 2. In the DPH received by the asynchronous endpoint, PP is 0; 3. The LPF in the DPH received by the synchronization endpoint is 1; 4. The current CHAIN's NUMP_EMPTY is 1; 5. If no DPH is received within 4us after DPH is completed, an interrupt will occur. (Optional). Upon receiving a short packet, optional hardware can automatically perform NRDY or disable all... chain; When PP=0 is received, the optional hardware can automatically perform NRDY or disable the settings. There is a chain; If DPP is not received within the timeout period, you can choose to automatically perform hardware NRDY or disable it. All chains. If the expected number of packets are not received in synchronous mode (CRC error). error)	0
5	RB_EP_RX_LPF_FLAG	RO	Used only for synchronous downlink endpoints, the LPF in the currently received DPH state.	0
4	RB_EP_RX_ISO_PKT_ERR	RO	Used only for synchronous download endpoints, currently received DPP contains CRC32 error.	0

3	RB_EP_RX_NUMP_EMPTY		If NUMP in the current CHAIN is 0, then this position is 1.	0
2	RB_EP_RX_DPH_PP		The PP site status in the DPH currently received by RO.	0
[1:0]	RB_EP_RX_CHAIN_NO		RO is the sequence number of the CHAIN that caused the current interruption.	0

27.2.4.7 Length of the last packet received by endpoint n CHAIN (R16_UEPn_RX_CHAIN_LEN) (n=1-7)

Offset address: 0xD6 + (n-1)*0x20 Name access

Bit			describe	Reset value
[15:11] Reserved			RO is reserved.	0
[10:0]	RB_EP_RX_CHAIN_RX_LEN		RO: The length of the last packet received in the currently completed chain.	0

27.2.4.8 Number of NUMPs that endpoint n can receive (R8_UEPn_RX_CHAIN_MAX_NUMP) (n=1-7) Offset address:

0xD8 + (n-1)*0x20

Bit	name	access	describe	Reset value
[7:0]	RX_CHAIN_MAX_NUMP		RW: The number of DPP packets that the chain can receive.	0

27.2.4.9 Number of NUMPs received by endpoint n (R8_UEPn_RX_CHAIN_NUMP) (n=1-7) Offset address: 0xD9 + (n-1)*0x20

Name Access description [7:0] RX_CHAIN_NUMP

Bit	RO Number of DPP packets received by this CHAIN.			Reset value
				0

27.2.4.10 DMA Offset Length of Endpoint n (R16_UEPn_RX_DMA_OFS) (n=1-7) Offset Address: 0xDA +

(n-1)*0x20 Name Access Description RW Offset

Bit	address of DPP in this CHAIN.			Reset value
[15:0] CHAIN_DMA_OFS				0x0400

27.2.4.11 DMA start address of endpoint n (R32_UEPn_RX_DMA) (n=1-7)

Offset address: 0xDC + (n-1)*0x20 Name access

Bit			describe	Reset value
[31:0] CHAIN_DMA_BA		RW	In normal mode: The CHAIN is the starting address of the DMA for receiving data. In FIFO mode: The FIFO start address is mapped to addresses 16-23 in SRAM. Bit; The FIFO end address is mapped to addresses 16-23 in SRAM. Bit.	0

27.2.5 Host Register Description

Table 27-17 USBSS Host Register List

name	Access address	describe	Reset value

R32_UH_TX_CTRL	0x40034084 Host Transmit Control Register;		0x0000XXXX
R32_UH_RX_CTRL	0x40034088 Host Receive Control Register;		0xXX00XXXX
R32_UH_TX_DMA_OFS	0x40034094 Host Transmit Address Offset Register;		0x00000400
R32_UH_RX_DMA_OFS	0x40034098 Host Receive Address Offset Register;		0x00000400
R16_HOST_TX_NUMP	0x4003409C Host Transmit NUMP Register;		0x0000
R16_HOST_RX_NUMP	0x4003409E Host Receive NUMP Register;		0xXX00
R32_HOST_STATUS	0x400340A0 Host Status Register;		0x00000100
R16_HOST_TX_FC_STATUS	0x400340A4 Host Endpoint Transmit Flow Control		0x0000
R16_HOST_RX_FC_STATUS	Register; 0x400340A6 Host Endpoint Receive Flow Control Register		0x0000
R32_TP_RX_DATA0	0x400340A8	The DATA0 register of DEV_NOTIF	0xXXXXXXXX
R32_TP_RX_DATA1	0x400340AC	DATA1 register of DEV_NOTIF	0xXXXXXXXX
R32_TP_RX_DATA2	0x400340B0	The DATA2 register of DEV_NOTIF	0xXXXXXXXX

27.2.5.1 Host Transmit Control Register (R32_UH_TX_CTRL) Offset

Address: 0x84 Name

Bit		access	The IN	Reset value
31	RB_UH_TX_ACT	RW	statement describes a transaction completion interruption flag; it is cleared by writing 0 in software and by hardware. Item 1.	0
30	RB_UH_TX_ISO	The RW host is preparing to send an ISO data packet.		0
29	RB_UH_TX_SETUP	RW	This indicates that the data packet being sent by the host is a Setup data packet. Set the setup flag.	0
28	RB_UH_TX_STATUS	RW indicates that the packet sent by the host is a STATUS TP.		0
[27:24] Reserved		RO is reserved.		0
twenty three	RB_UH_TX_LPF	RW	For burst transmissions, this bit only indicates the last packet. LPF/EOB, the preceding LPF/EOB uses a fixed value of 0. For synchronization endpoints, the following definition applies: Last Packet Fla--IN returns the LPF in the DPH packet during the transaction; this bit is... 1 indicates that this packet is the last packet in this frame interval. For asynchronous endpoints, the following definition applies: End Of Block - -IN returns the EOB from the DPH packet in the transaction; when this bit is 1. The endpoint enters flow control mode, and the host waits for the device to enter this state. After receiving ERDY-TP, restart the IN transaction.	0
[22:21] RB_UH_TX_RES		RW	Response to DPH+DPP: 00: NRDY-TP; 01: ACK-TP; 10: STALL-TP; 11: No response (invalid). EPn_R_RES is automatically cleared after the SETUP/OUT transaction is completed. EPn_R_RES.	0
[20:16] RB_UH_TX_SEQ		RW	The hardware automatically increments SEQ_NUM when the endpoint receives a data packet. Except for endpoint 0: For synchronization endpoints, the hardware automatically clears upon receiving an ITP packet. Zero, because the protocol requires the first packet of the frame interval to have a SEQ. It is 0.	0

[15:12] RB_UH_TX_EP		RW	Indicates the destination (device) of packets sent in host mode. Target endpoint number)	x
11	Reserved	RO reserves the		0
[10:0] RB_UH_TX_LEN		RW	endpoint receive length register, which is used for burst transmissions. The degree represents the length of the last packet in a burst transmission. His package must be 1024B according to the agreement.	x

27.2.5.2 Host Receive Control Register (R32_UH_RX_CTRL) Offset Address: 0x88

Name

Bit		access	The OUT	Reset value
31	RB_UH_RX_ACT	RW	flag indicates a transaction completion interruption; it is cleared by writing 0 in software and by hardware. Item 1.	0
30	RB_UH_RX_ISO	The RO receives data packets (DPP) synchronously.		0
29	Reserved	RO Reserved. The		0
[28:24] RB_UH_RX_NUMP		RW	number of packets (DPPs) that the endpoint can receive (burst transmission). lose).	x
twenty three	Reserved	RO is retained.		0
[22:21] RB_UH_RX_RES		R0	Responses to DPH+DPP or STATUS-TP: 00: NRDY-TP; 01: ACK-TP; 10: STALL-TP; 11: No response (invalid). The SETUP/OUT/STATUS parameters are automatically cleared after a transaction is completed.	0
[20:16] RB_UH_RX_SEQ		RW	The endpoint expects to receive SEQ_NUM, which is automatically incremented by 1 by the hardware. Except for point 0. For synchronization endpoints, the hardware automatically clears upon receiving an ITP packet. Zero, because the protocol requires the first packet of the frame interval to have a SEQ. It is 0.	0
[15:12] RB_UH_RX_EP		RW	Indicates the source (device endpoint) of the packet received in host mode. Number).	x
11	Reserved	RO is reserved.		0
[10:0] RB_UH_RX_LEN		R0	The endpoint receive length register, for burst transmissions, is this length... The degree represents the length of the last packet in a burst transmission. His package must be 1024B according to the agreement.	x

27.2.5.3 Host Transmit Address Offset Register (R32_UH_TX_DMA_OFS) Offset Address: 0x94

Name

Bit		access	describe	Reset value
[31:0] RB_UH_TX_DMA_OFS		After the RW host completes the transmission, the address offset of the DMA is set.		0x00000 400

27.2.5.4 Host Receive Address Offset Register (R32_UH_RX_DMA_OFS) Offset Address: 0x98

Name

Bit		access	describe	Reset value

[31:0] RB_UH_RX_DMA_OFS			After the RW host finishes receiving the data, it sets the address offset of the DMA.	0x00000400

27.2.5.5 Host sends NUMP register (R16_HOST_TX_NUMP)

Offset address: 0x9C

Bit	name	access	describe	Reset value
[15:8] Reserved		RO Reserved. The		0
[7:0] RB_UH_TX_NUMP		RW	number of packets the host expects to send; for each packet sent, the hardware... Automatically decrement by 1.	0

27.2.5.6 Host Receive NUMP Register (R16_HOST_RX_NUMP)

Offset address: 0x9E

Bit	Name	Access description: The number of DPPs the RW host		Reset value
[15:8] RB_UH_RX_DPP_NUM		has already accepted. The number of packets the host		X
[7:0] RB_UH_RX_NUMP		RW	expects to receive, if it's a synchronous transmission. The hardware automatically decrements by 1 for each received message.	0

27.2.5.7 Host Status Register (R32_HOST_STATUS) Offset Address: 0xA0 Bit

Name [31:20] Reserved

		access	describe	Reset value
		RO (Reserved). In		0
[19:18]	RB_UH_ITP_PRESAGE	RO	host mode, this bit indicates the time when the ITP packet was sent. The host will not send ITP packets within 00:9us; 01: The host sends an ITP packet within 9us; 11: The host sends an ITP packet within 6us; Within 10:3us, the host sends an ITP packet.	0
17	RB_UH_RX_ISO_PKT_ERR	RO	During synchronous transmission, the CRC error of the received data packet (DPP) is... error.	0
16	RB_UH_RX_EOB_LPF	RO	This bit indicates the EOB/LPF status in the received packet. For burst transmissions, this bit only indicates the last packet. LPF/EOB, the preceding LPF/EOB uses a fixed value of 0. For synchronization endpoints, the following definition applies: Last Packet Flag -- IN This bit represents the LPF (Last Packet Flag) returned in the DPH packet during the transaction. A value of 1 indicates that the packet is the last packet in this frame interval. For asynchronous endpoints, the following definition applies: End Of Block - -IN returns the EOB from the DPH packet in the transaction; when this bit is 1. The endpoint enters flow control mode, and the host waits for the device to enter this state. Restart the IN transaction after sending ERDY-TP.	0
15	RB_UH_RX_ERDY_DIR	RO	Received ERDY from the device, this segment indicates the endpoint's direction. Towards.	0
[14:13] Reserved		RO is reserved.		0
[12:8]	RB_UH_RX_ERDY_NUMP	RO	Received ERDY from the device; this segment indicates the device's corresponding... The number of data packets an endpoint can send/receive.	0x1
[7:4] RB_UH_RX_ERDY_EP		The RO receives the ERDY from the device; this segment represents the device's ERDY.		0

			The endpoint number.	
[3:0] Reserved		RO is reserved.		0

27.2.5.8 Host endpoint transmit flow control register (R16_HOST_TX_FC_STATUS) offset

address: 0xA4 Name

Bit		Access description: The flow control status of the	Reset value
[15:1] EPx_TX_FC		RW1Z host for sending endpoints 1-15.	0
0	Reserved	RO is reserved.	0

27.2.5.9 Host endpoint receive flow control register (R16_HOST_RX_FC_STATUS) offset

address: 0xA6

Bit	name	access	describe	Reset value
[15:1] EPx_RX_FC		The RW1Z host controls the flow of data to receiver endpoints 1-15.		0
0	Reserved	RO is reserved.		0

27.2.5.10 DEV_NOTIF-TP Data 0 Register (R32_TP_RX_DATA0) Offset Address: 0xA8 Name

Bit		access	After	Reset value
[31:0] USB3_NOTIF_DATA0		RO	the DEV_NOTIF-TP reception is completed, HP stores the data in the register. Storage.	X

27.2.5.11 DEV_NOTIF-TP Data 1 Register (R32_TP_RX_DATA1) Offset Address: 0xAC

Bit	name	access	describe	Reset value
[31:0] USB3_NOTIF_DATA1		RO	After DEV_NOTIF-TP reception is complete, HP stores the data in the register. Storage.	X

27.2.5.12 DEV_NOTIF-TP Data 2 Register (R32_TP_RX_DATA2) Offset Address: 0xB0

Bit	name	access	describe	Reset value
[31:0] USB3_NOTIF_DATA2		RO	After DEV_NOTIF-TP reception is complete, HP stores the data in the register. Storage.	X

27.2.6 USBSS LINK Register Description

Table 27-18 USBSS LINK Register List

name	Access address	describe	Reset value
R32_LINK_CFG	0x40034000	LINK configuration register	0xC0000128
R32_LINK_CTRL	0x40034004	LINK control register	0xXX000013
R32_LINK_INT_CTRL	0x40034008	LINK interrupt enable register	0x00000000
R32_LINK_INT_FLAG	0x4003400C	LINK Interrupt Flag Register	0x00000000
R32_LINK_STATUS	0x40034010	LINK status register	0xX0000430
R8_LINK_ITP_PRE	0x40034017	LINK ITP Timeout Mode Register	0x00
R8_LINK_U2_INACT_TIME	0x4003401D	LINK U2 Inactivity Timeout Counter Threshold is set to 0x00	

		Register	
R8_LINK_U1_WKUP_FILTER 0x40034028		U1 Wake-up LFPS Duration Configuration Register	0x4A
R8_LINK_U2_WKUP_FILTER 0x4003402C		U2 Wake-up LFPS Duration Configuration Register	0x02
R8_LINK_U3_WKUP_FILTER 0x40034030		U3 Wake-up LFPS Duration Configuration Register	0x64
R16_LINK_ISO_DLY	0x40034040	LINK Synchronization Delay	0x0028
R16_LINK_LPM_CR	Register 0x40034050	Link Power Management Register	0x0180
R32_LINK_LMP_PORT_CAP	0x40034054	PORT_CAP register	0xXXXX0000
R32_LINK_LMP_RX_DATA0	0x40034058	LMP Receive Data 0 Register	0xXXXXXXXX
R32_LINK_LMP_RX_DATA1	0x4003405C	LMP Receive Data 1 Register	0xXXXXXXXX
R32_LINK_LMP_RX_DATA2	0x40034060	LMP Receive Data 2 Register	0xXXXXXXXX
R32_LINK_LMP_TX_DATA0	0x40034064	USB Custom HP Data 0 Register	0xXXXXXXXX
R32_LINK_LMP_TX_DATA1	0x40034068	USB Custom HP Data 1 Register	0xXXXXXXXX
R32_LINK_LMP_TX_DATA2	0x4003406C	USB Custom HP Data 2 Register	0xXXXXXXXX

27.2.6.1 LINK Configuration Register (R32_LINK_CFG) Offset

Address: 0x00 Name

Bit		access		Reset value
31	RB_LINK_RESET	RW	Describe LINK reset, including resetting the state machine and all interrupt flags. Highly effective.	1
[30:22] Reserved		RO is reserved.		0
twenty one	RB_LINK_TOUT_MOD E	RW	1:3.2 SPEC; 0:3.0 SPEC. Note: Different protocol versions may affect the calculation. flow control. Timer duration: PM_LC_TIMER: 3.2_SPEC: 4us / 3.0_SPEC: 3us; PM_ENTRY_TIMER: 3.2_SPEC: 8us / 3.0_SPEC: 6us; PEND_HP_TIMERÿ3.2_SPECÿ10us/3.0_SPECÿ 3us.	0
20	RB_LINK_U1_PING_ EN	RW	Enable PING_FPFS under U1: 1: Send; 0: Do not send.	0
[19:18] RB_LINK_SKP_MODE		RW	SKP rejection control used in descrambling: 00: The lower byte of the lower two bytes is SKP, and the lower two bytes are closed. Section descrambling enabled; the lower byte of the two high bytes is SKP. Disable descrambling enable for the high two bytes; 01/11: The lower two bytes are both SKP, so the lower two bytes are disabled. Descrambling enable: The high two bytes are both SKP, disable the high two bytes. Descrambling enabled; 10: Either of the lower two bytes is SKP, thus disabling the lower two bytes. Byte descrambling enabled; either of the two highest bytes is SKP disables descrambling of the high two bytes.	0
17	RB_LINK_U2_ALLOW	RW	High validity; upon receiving LGO_U2, respond with LAU to allow. Enter U2 state; otherwise, respond upon receiving LGO_U2.	0

			LXU refused to enter the U2 state. Note: For the downlink port, this enable is definitely related to the U2_TIMEOUT value. If U2_TIMEOUT > 0, configure U2_ALLOW. Valid; otherwise, invalid.	
16	RB_LINK_U1_ALLOW	RW	High validity; upon receiving LGO_U1, respond with LAU to allow. Enter U1 state; otherwise, respond after receiving LGO_U1. LXU refused to enter the U1 state. Note: For the downlink port, this enable is definitely related to the U1_TIMEOUT value. If U1_TIMEOUT > 0, configure U1_ALLOW. It works, or it doesn't.	0
15	RB_LINK_LTSSM_MO DE	RW	The link state machine enters DISABLE mode: 1: In accordance with the agreement, under circumstances such as timeout or number of tests, proceed... Disabled; 0: Failure to comply with the agreement regarding timeouts, number of tests, etc. Enter DISABLED;	0
14	RB_LINK_LOOKBACK _ACT	RW	Used in LOOPBACK mode for the LOOPBACK master controller Code type transmission (LOOPBACK slave keeps 0), ACT is A high-level signal that lasts for a period of time. 1. The LOOPBACK host needs to send when this bit goes high. BRST to LOOPBACK Slave Reset LOOPBACK The machine's error counter (slave loopback BRST). 2. When ACT is high, the host sends the LOOPBACK signal. The BDAT is sent to the slave device, which then compares the received BDAT with... Error counting is performed based on whether the scrambled coded zeros are equal (from...). (Machine loop BDAT). 3. When ACT is low, the LOOPBACK host sends BERC. To the slave (the slave returns the error count BCNT to the master).	0
13	RB_LINK_LOOPBACK _EN	RW	Allow entry into the LOOPBACK enable bit, active high, in conjunction with... The LOOKBACK option in TX/RX_TS_CFG[3] is enabled. Both are required for LINK to enter LOOKBACK mode. (Used to enable data loopback and error handling of the slave device) False count, LOOPBACK host keeps 0).	0
12	RB_LINK_U2_DET_E N	RW	Detection method for connected devices in U2 state: 1: Software actively initiates the detection of connected devices in U2 mode; 0: Hardware automatically times and detects whether the device is connected in the U2 state. Prepare.	0
[11:10]	RB_LINK_CP78_SEL	RW	In Compliance Pattern 7/8, send consecutive 0s. Or the length of consecutive 1s: 00: 190 bits; 01: 120 bits; 10:50 bits; 11:250 bits.	0
[9:8]	RB_LINK_TX_DEEMP H	RW	De-emphasis control of transmitter: 00: -6.0 dB de-emphasis; 01: -3.5 dB de-emphasis (default);	01b

			10: No de-emphasis; 11: Retained. Note: Normal working mode can only be 0/1; other values are used for... COMPLIANCE.	
7	RB_LINK_TX_SWING	RW	Transmitter signal swing control: low swing results in low power consumption, but... Affecting transmission distance 1: Low Swing (400mv-1200mv); 0: Full Swing (800mv-1200mv).	0
6	RB_LINK_RX_EQ_EN	RW	receiver equalization enable control is optional as specified in the protocol.	0
5	RB_LINK_LFPS_RX_ PD	RW	LFPS receive control; this bit being 1 indicates that LFPS reception is disabled. receive	1
4	RB_LINK_COMPLIAN CE_EN	RW	Does POLLING_LFPS timeout trigger the COMPLIANCE module? Mode: 1: Entry timed out; 0: Timeout will prevent entry.	0
3	RB_LINK_PHY_RESE T	RW	PIPE interface reset 1: Reset; 0: Clear reset.	1
2	RB_LINK_SS_PLR_S WAP	RW	Swap the polarities of SSTX and SSRX as follows: 1: SSTXA(SSTXM), SSTXB(SSTXP); SSRXA(SSRXM),SSRXB(SSRXP). 0: SSTXA(SSTXP), SSTXB(SSTXM), SSRXA(SSRXP),SSRXB(SSRXM). Note: This is in adaptive mode, rx The difference is that polarity rx polarity where only SSRXM and SSRXP are swapped.	0
1	RB_LINK_RX_TERM_ EN	RW	Receiver terminal resistor control: 1: Connect a resistor to the connection point; 0: Remove the terminating resistor.	0
0	RB_LINK_DOWN_MOD E	RW	Peripheral type: 1: downstream; 0: upstream. In host mode, it should be configured as downstream; in device mode... Configure it as upstream. Specifically: When OTG mode support is required, it should be configured as follows: downstream , Port_Capability(LMP) Tiebreaker should be set to 0, indicating that when linking to a host... When connected to a device, this port will become a device. The port is the host. The protocol stipulates that when both ports are... When downstream or upstream, the Tiebreaker is Port 1 will become the downstream (host), and then Both parties need to re-exchange Port_Capability(LMP).	0

27.2.6.2 LINK Control Register (R32_LINK_CTRL) Offset Address:

0x04 Name

Bit		access	describe	Reset value

[31:24]	LINK_RX_TS_CFG	RO	Received link control: BIT0: 1: Reset; BIT1: 0: Normal transmission. Reserved. BIT2: 1: Enable Loopback; 0: Disable Loopback. BIT3: 1: Disable scrambling; 0: Enable scrambling. BIT4~7: Reserved.	0
[23:16]	LINK_TX_TS_CFG	RW	Send the link for the TS1/TS2 training sequences configuration: BIT0: 1: HOT_RESET; 0: Normal training mode. BIT1: Reserved. BIT2: 1: Enable Loopback; 0: Disable Loopback. BIT3: 1: Disable scrambling; 0: Enable scrambling. BIT4~7: Reserved. Specifically, reserved bits must be set to 0; BIT0 and BIT2 are not. It can be 1 at the same time; In device mode, if a valid HOT_RESET is received, The HOT_RESET interrupt flag is generated, and BIT0 is simultaneously activated by hardware. Automatically set to 1 in response to the host's HOT_RESET.	0
15	LINK_TX_LGO_U3	RW	After this bit becomes valid, LGO_U3 is sent (active high), and the hardware automatically... Reset to zero; The agreement stipulates that the receiving party cannot refuse to enter the U3 state; The LINK status is U3, indicating that it has entered the U3 low-power state. At this point, the software should change the current power mode to P3.	0
14	LINK_TX_LGO_U2	RW	After this bit becomes valid, LGO_U2 is sent (active high), and the hardware automatically... Reset to zero; If the other party responds with LXU, it indicates a refusal to enter U2 status. State; if the LINK query state is U2, it indicates that the other party has accepted. Entering the U2 low-power state, the software should change the current power... The source mode is P2.	0
13	LINK_TX_LGO_U1	RW	After this bit becomes valid, LGO_U1 is sent (active high), and the hardware automatically... Reset to zero; If the other party responds with LXU, it indicates a refusal to enter the U1 mode. If the LINK status is U1, it indicates that the other party has accepted. Entering the U1 low-power state, the software should change the current power... The source mode is P1.	0
12	LINK_POLLING_EN	RW	Highly effective; if SS.RX_DETECT detects a TERM, it will... After the software sets this bit to be valid, a POLLING handshake is performed. Note: AUTO_MODE[9] is 1 When, if AUTO_MODE[8] It is 1 and the current power supply is P2, begin to hand POLLING_LFPS hand; When AUTO_MODE[9] is true, if AUTO_MODE[7] is true... 1 1. And the current power supply is P0, start POLLING_LFPS. hand.	0
11	LINK_REG_ROUT_EN	RW	Enable the routing function of the HUB, active high, with registers but no... interface:	0

			1. The uplink port can normally route the header packet to the downlink port; 0: The uplink port cannot route the header packet to the downlink port.	
10	LINK_LUP_LDN_EN	RW	In U0 state, if there is no data, whether to send data every 10μs. LUP, LDN packages: 1: Enable sending; 0: Sending is prohibited.	0
9	LINK_TX_UX_EXIT	RW1Z	High validity, automatic hardware reset: 1. If LINK is in the U3 state, it is equivalent to executing... After a U3_WAKEUP handshake fails, the software should... The host should re-initiate U3_WAKUP after at least 100ms. In this mode, if three consecutive handshakes fail, the software should be set to... Set LINK to INACTIVE, check TERM, and the device can... The connection has been broken. If U3_WAKEUP is received, then... If for some reason the handshake signal cannot be returned, when ready to proceed... After entering U0, U3_WAKEUP should be sent proactively. 2. If LINK is in the U2 state, it is equivalent to executing... U2_EXIT(LFPS). 3. If LINK is in state U1, it is equivalent to executing... U1_EXIT(LFPS). 4. If the U1_EXIT/U2_EXIT handshake fails, LINK enters INACTIVE.	0
8	LINK_TX_WARM_RST RW1Z		High-active bit, cleared by software. A valid high-active bit will be sent. warm-reset. In host mode: the power is raised by software; you can choose between software or hardware. Pull down 1. The duration of warm_reset is determined by the software. The duration for the machine to send a warm-reset should be within [time period]. Between 80ms and 120ms (typically 100ms), when the device After a warm-reset is detected, it should be in tResetDelay Send within the time range of (18ms~50ms) The warm_reset response will continue until a warm_reset is detected from the host. After the warm-reset is complete, the device sends a warm_reset message. Finish. 2. After the software pulls the price up, the hardware will automatically time it after 100ms. Pull low (requires configuring corresponding hardware timeout enable) reg_auto_mode[13]) In device mode: the device controller detects that the host is continuously After sending a warm reset 18ms later, a check will be automatically declared. A warm reset was detected and a reset was performed. Specifically, the protocol stipulates that only the host has the capability to actively send a warm reset; device mode can only passively respond to a warm reset. reset.	0
7	LINK_GO_RX_DET	RW1Z	Before setting this bit, PD_MODE should be set to P2 mode. LINK enters SS.RX_DETECT TERM_PRESENT Software query to find out if there is a connection or if the connection is broken.	0

			On. High validity, automatically reset to zero.	
6	LINK_GO_RECOVERY RW1Z		Configure LINK in SS.RECOVERY, active high, hardware. Automatically reset.	0
5	LINK_GO_INACTIVE RW1Z		Configure LINK to enter SS.INACTIVE, active high, hardware. Automatically reset.	0
4	LINK_GO_DISABLED	RW	Setting the LINK to SS.DISABLED (high-active) requires... Reset the software. If the hardware automatically enters SS.DIABLED, then this bit is... The hardware is automatically set to 1. In device mode, if a USB 2.0 bus response is received... For this part, the software should first set PD_MODE to P2 mode, then... Clearing this bit will automatically put LTSSM into SS.RX_DETECT. The equipment performs 8 checks (one check every 12ms), such as If no TERM is detected, the device enters the DISABLE state. It communicates in USB 2.0 mode. In host mode, data should be sent via the USB 2.0 controller. For the BUS_RESET signal, the software should first set PD_MODE to [mode value]. In P2 mode, then clear this bit. If TERM is detected, A POLLING_LFPS handshake will be performed. Otherwise, the software should be configured... The power mode is P3, and communication is via USB 2.0.	1
[3:2] Reserved		RO is reserved.		0
[1:0] LINK_PD_MODE		RW	Configure the current power mode of the PHY, corresponding to the PIPE. PO/P1/P2/P3: 00: PO: Normal working mode; 01: P1: Low power mode. In this mode, both ports are in low power mode. Regularly send PING-LFPS to maintain the connection; 10: P2: Low-power mode, this mode consumes less power than P1 mode. The lower the temperature, the longer it takes to exit the mode; 11: P3: Ultra-low power mode, in which the PHY's time... The clock (125MHz) will be turned off. Changing this configuration value will cause LINK_BUSY to go high, and the software... If the LINK_BUSY value is 0, the switch is complete. Querying PD_MODE_CURRENT in LINK_STATUS yields... The current power mode of the PHY (optional operation).	11b

27.2.6.3 LINK Interrupt Enable Register (R32_LINK_INT_CTRL)

Offset address: 0x08

Bit	Name	access	describe	Reset value
31	LINK_IE_STATE_CHG	RW	Link state machine change flag interrupt enabled: 1: Enable; 0: Prohibited.	0
30	LINK_IE_U1_TOUT	RW	U1 Timeout Interrupt Enable: 1: Enable; 0: Prohibited.	0
29	LINK_IE_U2_TOUT	RW	U2 timeout interrupt enabled:	0

			1: Enable; 0: Prohibited.	
28	LINK_IE_UX_FAIL	RW	UX conversion failure interrupt enable: 1: Enable; 0: Prohibited.	0
27	LINK_IE_TX_WARMR ST	RW	Sending `warm_reset` ends interrupt enable: 1: Enable; 0: Prohibited.	0
26	LINK_IE_UX_EXIT_ FAIL	RW	Enable interrupt for exiting Ux if it fails: 1: Enable; 0: Prohibited.	0
[25:24] Reserved		RO retains		0
twenty three	LINK_IE_RX_LMP_T OUT	RW	receive LMP timeout interrupt enable: 1: Enable; 0: Prohibited.	0
twenty two	LINK_IE_TX_LMP	RW	Upon successful entry into U0, an HP packet can be sent to enable interruption. 1: Enable; 0: Prohibited.	0
twenty one	LINK_IE_RX_LMP	RW	Received link command flag interrupt enabled: 1: Enable; 0: Prohibited.	0
20	LINK_IE_RX_DET	RW	Link enters Rx.Detect state interrupt enabled: 1: Enable; 0: Prohibited.	0
19	LINK_IE_LOOPBACK	RW	The link enters loopback mode for testing and error isolation. Interrupt enable: 1: Enable; 0: Prohibited.	0
18	LINK_IE_COMPLIAN CE	RW	The link enters compliance testing, using compatibility testing or physical testing. Layer consistency test interruption enabled: 1: Enable; 0: Prohibited.	0
17	LINK_IE_HPBUF_FU LL	RW	hp_buf transmits FIFO full interrupt enabled: 1: Enable; 0: Prohibited.	0
16	LINK_IE_HPBUF_EM PTY	RW	hp_buf sends FIFO null interrupt enable: 1: Enable; 0: Prohibited.	0
15	LINK_IE_HOT_RST	RW	Hot reset, using TS1/TS2 ordered sets. Reset interrupt enable: 1: Enable; 0: Prohibited.	0
14	LINK_IE_U3_WAKEU P	RW	In U3 state, receiving a low-frequency periodic signal (LFPS) to wake up Awake interruption enable: 1: Enable;	0

			0: Prohibited.	
13	LINK_IE_WARM_RST	RW	Enable warm reset (not connected) interrupt using LPFS: 1: Enable; 0: Prohibited.	0
12	LINK_IE_UX_EXIT	RW	The interrupt enable was interrupted upon receiving a request to exit the Ux: 1: Enable; 0: Prohibited.	0
11	LINK_IE_TXEQ	RW	LINK enters POLLING_RXEQ, used for receiver equalization training. Training interruption enable: 1: Enable; 0: Prohibited.	0
10	LINK_IE_TERM_PRES	RW	LINK enters RX_DETECT, used to detect the remote receiver. Tracking segment impedance detection interrupt enable: 1: Enable; 0: Prohibited.	0
9	LINK_IE_UX_REJ	RW	Sending LGO_Ux, receiving LXU indicating rejection of incoming Ux transmission. Interrupt enable: 1: Enable; 0: Prohibited.	0
8	LINK_IE_U3_WK_TOUT	RW	Enable interrupt after timeout for request to exit U3 command: 1: Enable; 0: Prohibited.	0
7	LINK_IE_GO_U0	RW	LINK is in U0 interrupt enabled state: 1: Enable; 0: Prohibited.	0
6	LINK_IE_GO_U1	RW	LINK is in U1 interrupt enabled state: 1: Enable; 0: Prohibited.	0
5	LINK_IE_GO_U2	RW	LINK is in U2 interrupt enabled state: 1: Enable; 0: Prohibited.	0
4	LINK_IE_GO_U3	RW	LINK is in U3 interrupt enabled state: 1: Enable; 0: Prohibited.	0
3	LINK_IE_DISABLE	RW	LINK is in DISABLED interrupt enabled: 1: Enable; 0: Prohibited.	0
2	LINK_IE_INACTIVE	RW	LINK is in INACTIVE interrupt enabled: 1: Enable; 0: Prohibited.	0
1	LINK_IE_RECOVERY	RW	Because the LINK is in the RECOVERY interrupt enable state: 1: Enable; 0: Prohibited.	0
0	LINK_IE_READY	RW	LINK completes initialization, including the two ports beforehand. (Header Sequence Number	0

			Interrupt enable (Advertisement), (RX Header Buffer Credit Advertisement): 1: Enable; 0: Prohibited.	

27.2.6.4 LINK Interrupt Flag Register (R32_LINK_INT_FLAG)

Offset address: 0x0C

Bit	Name	access	describe	Reset value
31	LINK_IF_STATE_CHG	RW1Z	link_reset resets the value to 0 (writing 1 clears it): 1: A flag indicating a change in the link state machine; 0: The link state machine has not changed.	0
30	LINK_IF_U1_TOUT	RW1Z	U1 timeout interrupt; (write 1 to clear): 1: Timeout; 0: Not timed out;	0
29	LINK_IF_U2_TOUT	RW1Z	U2 timeout interrupt; (write 1 to clear): 1: Timeout; 0: Not timed out;	0
28	LINK_IF_UX_FAIL	RW1Z	Entry to Ux failed and was interrupted; (Write 1 to clear): 1: Failed to enter Ux; 0: No action.	0
27	LINK_IF_TX_WARMRST	RW1Z	Send `warm_reset` to end the interrupt; (write 1 to clear): 1: Sending warm_reset completes the process; 0: No action.	0
26	LINK_IF_UX_EXIT_FAIL	RW1Z	Exiting Ux failed and was interrupted; (writing 1 clears the value): 1: Failed to exit Ux; 0: No action.	0
[25:24] Reserved		RO is reserved.		0
twenty three	LINK_IF_RX_LMP_TOUT	RW1Z	After LINK initialization, the exchange of Port Capabilities/ Configuration LMPs timed out. (20us) Interruption flag.	0
twenty two	LINK_IF_TX_LMP	RW1Z	LINK initialization complete interrupt flag; this signal, when pulled high, triggers a software interrupt. Configure and send Port Capabilities LMP to the end Oral capacity exchange.	0
twenty one	LINK_IF_RX_LMP	RW1Z	Receive LMP interrupt flag: Control the sending of relevant LMPs during the port capability exchange phase: Downlink port receives Port Capabilities LMP, software Port Configuration LMP (Port Configuration LMP) is used to send configuration information. Uplink port receives Port Capabilities LMP, software Port Configuration Response LMP.	0
20	LINK_IF_RX_DET	RW1Z	The interrupt flag for entering RX_DETECT state is set to 1. The software sets PWR_MODE to P2, preparing to proceed. RX_DETECT.	0
19	LINK_IF_LOOPBACK RW1Z LINK is the interrupt flag for entering the LOOPBACK state.			0
18	LINK_IF_COMPLIAN RW1Z LINK is the interrupt flag for entering COMPLIANCE state.			0

	CE			
17	LINK_IF_HPBUF_FU LL	RW1Z	Header Packet buffer full interrupt flag.	0
16	LINK_IF_HPBUF_EM PTY	RW1Z	Header Packet buffer empty interrupt flag.	0
15	LINK_IF_HOT_RST	RW1Z	The device received the HOT RESET interrupt flag: The software should clear after,the signal goes high. TX_TS_CFG[0] End (Complete) HOTRESET process.	0
14	LINK_IF_WAKEUP	RW1Z	The power supply is in P3 mode, and an LFPS signal interruption has been detected. Logo.	0
13	LINK_IF_WARM_RST	RW1Z	The device received a WARM RESET status change (valid -> (Invalid or invalid -> valid) interrupt flag.	0
12	LINK_IF_UX_EXIT	RW1Z	LINK has received a request from LFPS to exit U1/U2/U3. After setting the break flag to 1, the software should set PWR_MODE to 1. Set to P0. After setting PWR_MODE to P0, Ux exits. The timer starts counting down and continues until U1_EXIT_TIMER is reached. After U2_EXIT_TIMER and U3_WAKE_TIMER, LINK Exit Ux mode and enter Recovery mode.	0
11	LINK_IF_TXEQ	RW1Z	LINK enters TXEQ state interrupt flag: This indicates that the POLLING handshake is complete and the software is waiting for it to complete. PWR_MODE is set to P0 to enter the TXEQ phase.	0
10	LINK_IF_TERM_PRE S	RW1Z	detected a TERM disconnection or connection interruption flag.	0
9	LINK_IF_UX_REJ	The RW1Z LINK interrupt flag prevents entry into low-power mode (U1/U2). 0		
8	LINK_IF_U3_WK_TO UT	RW1Z	wake-up timeout interrupt flag from U3 (10ms).	0
7	LINK_IF_GO_U0	RW1Z	LINK enters U0 state interrupt flag.	0
6	LINK_IF_GO_U1	RW1Z	LINK enters U1 state interrupt flag.	0
5	LINK_IF_GO_U2	RW1Z	LINK enters U2 state interrupt flag.	0
4	LINK_IF_GO_U3	RW1Z	LINK enters U3 state interrupt flag.	0
3	LINK_IF_DISABLE	RW1Z	LINK enters SS.DISABLE state interrupt flag: Note: Direct software configuration GO_DISABLE Enable (link_ctrl) prevents LINK directly enters the DISABLE state. interrupts from going high.	0
2	LINK_IF_INACTIVE	RW1Z	LINK interrupts the entry into SS.INACTIVE state. Note: Direct software configuration enable GO_INACTIVE (to be (link_ctrl) directs INACTIVE, interrupts will LINK not be pulled high.	0
1	LINK_IF_RECOVERY	RW1Z	LINK enters the SS.RECOVERY state interrupt flag. After position 1, you should query LINK_ERR_STATUS to obtain the information. The reason why LINK enters SS.RECOVERY. Note: Enabled directly by software configuration GO_RECOVERY for direct import. , directs the link (link_ctrl) LINK RECOVERY, interruption will not cause a spike.	0
0	LINK_IF_READY	RW1Z	LINK enters U0 state and completes LINK initialization interrupt flag 0.	

			Will.	

27.2.6.5 LINK Status Register (R32_LINK_STATUS) Offset Address:

0x10 Name

Bit		access	describe	Reset value
31	LINK_HPBUF_EMPTY	The RO HP buffer is empty.		x
30	LINK_HPBUF_FULL	RO HP buffer is full. hp_buf		x
29	LINK_HPBUF_IDLE	RO	sends FIFO idle status: 1: hp_buf sends data via FIFO; 0: hp_buf sends an empty FIFO.	x
[28:23] RESERVE		RO is reserved.		0
twenty two	LINK_U3_SLEEP_AL LOW	The RO link is in U3 state, which allows it to sleep.		x
twenty one	LINK_U2_SLEEP_AL LOW	The RO link is in U2 state, which allows it to sleep.		x
20	LINK_RXDET_SLEEP _ALLOW	The RO link is in the RXDET state, which allows it to sleep.		x
19	LINK_WAKUP	RO received a link wake-up signal.		x
18	LINK_RX_LFPS	The RO link received the LFPS signal.		x
17	LINK_RX_DETECT	The RO link is at P2 and is performing rx_detect.		x
16	LINK_RX_UX_EXIT_ REQ	RO	Received a request to exit Ux: 1: LINK received a valid Ux exit LFPS signal. To exit U1/U2/U3, the software should configure the power mode. For P0 (Ux exits timing after switching from power mode to P0). The device starts timing and enters LINK mode after reaching the preset value. RECOVERY); 0: No change.	0
[15:12] Reserved		RO reserved.		0
[11:8] LINK_STATE		RO	Link status: 0000: STATE_U0; 0001: STATE_U1; 0010: STATE_U2; 0011: STATE_U3; 0100: STATE_DISABLED; 0101: STATE_RXDET; 0110: STATE_INACTIVE; 0111: STATE_POLLING; 1000: STATE_RECOVERY; 1001: STATE_HOTRST; 1010: STATE_COMPLIANCE; 1011: STATE_LOOPBACK; Other: Reserved.	4h
7	Reserved	RO is reserved.		0
6	LINK_TXEQ	The RO link is in POLLING_RXEQ mode.		0

[5:4]	LINK_PD_MODE_MAS K	RO	Link power status: 00: P0 state; 01: P1 status; 10: P2 state; 11: Default state P3 After	10b
3	LINK_READY	RO	LINK enters the U0 state, initialization (broadcast) is completed. After setting this bit to 1, exiting the U0 state will automatically clear this bit to zero.	0
2	LINK_BUSY	RO	LINK busy state; this bit is 1 when PD_MODE is switched. The system will be automatically reset by hardware after the switch is complete.	0
1	LINK_RX_WARM_RST	RO	Received a valid warm-reset signal from the host (received) The hardware automatically restarts 18ms after the host's warm_reset.	0
0	LINK_RX_TERM_PRE S	RO	After RX_DETECT, if a receiver termination resistor is present, this... The bit is 1.	0

27.2.6.6 LINK ITP Timeout Mode Register (R8_LINK_ITP_PRE)

Offset address: 0x17

Bit	Name	access	describe	Reset value
[7:0]	ITP_PRE	RO	ITP Timeout Mode: 1: Ignore uplink and downlink port ITP transmit/receive pairs for link idle status. The influence of state; 0: Downlink port transmits non-delayed ITP / Uplink port receives non-delayed ITP. When ITP is delayed, the link is not idle.	0x80

27.2.6.7 LINK U2 Inactivity Timeout Counter Threshold Register (R8_LINK_U2_INACT_TIMER) Offset Address: 0x1D Name

Bit		access	describe	Reset value
[7:0]	U2_INACTIVE_TIME R	RW	The value of the inactivity timeout counter threshold for U2: U2 Timeout value (this value indicates inactivity during U0/U1) The maximum allowed time for the state; if the timeout occurs, it will enter the U2 state. Value range: 0x00~0xff, 256us/LSB. Note: whenU1/U2 Timeout The value is 0xff When the timeout period for the inactive link is infinite, the timer... 0xff U1/U2 Request Please remember that the connection will not be initiated automatically.	0

27.2.6.8 U1 Wake-up LFPS Valid Duration Register (R8_LINK_U1_WKUP_FILTER) Offset Address: 0x28 Name

Bit		access		Reset value
[7:0] U1_WKUP_FILTER		RW	Describes the validity period of the LFPS received when U1 exits. Port Sending an LFPS signal to initiate a handshake, when the received LFPS signal reaches this point... During this period, the received U1 EXIT can be considered valid, and the transmission can be initiated by pulling it high. LFPS_last, handshake successful, defaults to 600ns. U1_WKUP_FILTER[7]: Control The time unit for U1_WKUP_FILTER[6:0];	0x4A

			1: U1_WKUP_FILTER[6:0] is in microseconds (μs) and has a duration of [duration]. (n)us; 0: U1_WKUP_FILTER[6:0] is in units of 8ns, and its duration is... (8n)ns, where n takes values from 0 to 124; Note that when the time is long, it must be used.1us. The configuration method of U1_WKUP_FILTER[7] is otherwise Unable to wake up.	

27.2.6.9 U2 Wake-up LFPS Valid Duration Register (R8_LINK_U2_WKUP_FILTER)

Offset address: 0x2C

Bit	name	access	describe	Reset value
[7:0]	U2_WKUP_FILTER	RW	The LFPS received during U2 wake-up and loopback exit. Effectively determine the duration, the port sends LFPS to initiate a handshake, when When the received LFPS reaches this time, send LFPS_last. High, handshake successful, the received U2 EXIT can be considered valid. The duration is (n)us, with a default of 2us.	0x02

27.2.6.10 U3 Wake-up LFPS Valid Duration Register (R8_LINK_U3_WKUP_FILTER)

Offset address: 0x30

Bit	name	access	describe	Reset value
[7:0]	U3_WKUP_FILTER	RW	The effective judgment duration of the LFPS received during U3 wake-up. The port sends LFPS to initiate a handshake; when the received LFPS reaches... At that time, the LFPS_last sent went high, indicating a successful handshake. The received U3 EXIT can be considered valid, with a duration of (n)us. Default is 100us.	0x64

27.2.6.11 LINK Synchronization Delay Register (R16_LINK_ISO_DLY)

Offset address: 0x40

Bit	name	access	describe	Reset value
[15:0]	LINK_ISOCH_DLY	RW	The delay time between parsing a serial bit stream into parallel data, default Recognize 40ns, during enumeration SET ISOCH DELAY the host will use this. Information is sent to the device, which then writes the delay information. This register, the lower 3 bits of this register are invalid (due to clock frequency). The frequency is 125MHz, and the delay time needs to be set to 8*n).	0x28

27.2.6.12 Link Power Management Register (R16_LINK_LPM_CR) Offset Address: 0x50

Name

Bit		access	describe	Reset value
[15:14]	Reserved	RO is reserved.		0
13	PHY_CHSEL_AUTO	RW	PHY layer transmit/receive channel type is automatically selected: 1: If the connection status flag remains unchanged after the device test is completed. To switch to another channel type, the default channel type is... Standard USB interface; 0: The PHY layer is prohibited from automatically switching transmit/receive channel types; it can only...	0

			Use the software to set the send/receive type; the default is a standard USB interface.	
12	LPM_RXDET_EN	RO	When the lpm count reaches the expected value RXDET_EXP, it is confirmed that there is... The device is connected to the PHY, and this bit is pulled high.	0
11	LPM_TERM_PRESENT	The R0 PHY layer detected a device connection.		0
10	LPM_TERM_CHG	The RO connection device has changed; the device has been inserted or removed.		
9	LPM_EN	RW	CLKLPM is fixed at 12MHz when LINK_PD_MODE=3 & LPM_EN=0 & OSCOUTEN=0 & PLLALIV=0. CLKOSCO will stop; the crystal oscillator will cease oscillation. 1: Enable CLKLPM in L1 mode; 0: Prohibited. Note: This bit is unrelated to the USBSS controller; the system enters low... Power consumption requires the crystal oscillator to be turned off; this bit should be cleared to zero.	0
8	LPM_RST	RW	Reset signal for LPM-related registers: 1: Reset LPM counter and LPM terminal detection; 0: Do not reset.	1
[7:0] RXDET_EXP		RW detects the external device counter register.		0x80

27.2.6.13 PORT_CAP register (R32_LINK_LMP_PORT_CAP)

Offset address: 0x54

Bit	Name	access	The	Reset value
31	LINK_LMP_RX_CAP_ VLD	RO	description indicates that a valid PORT_CAP-LMP has been received, and the protocol specifies that in the process of... Within 20us after entering U0 state, both parties exchange LINK. PORT_CAP-LMP. Note: This signal will appear in the following time: LINK The state changes and enters polling Reset to 0.	0
30	LINK_LMP_TX_CAP_ VLD	RO	PORT Capability configuration complete indicator (software operation) Write 1 (clear 0): 1: Indicates that PORT Capability configuration is incomplete; 0: Indicates that PORT Capability configuration is complete (host and...) The equipment falls into two categories. The host receives the port configuration response information from the device. Clear to 0 after (PORT Configuration Response); After receiving the host's port configuration, the device responds with a PORT Configuration Response and clears the string to 0.	0
[29:24] LINK_SPEED		RO	[24] Position 1 indicates that the device supports USB 3.2 Gen1. (5Gbps).	x
[23:0]	LINK_REG_PORT_CA P	RO	1. LINK_REG_PORT_CAP[23:20]: Tiebreaker, Used for two devices that simultaneously have both uplink and downlink capabilities. PORT(LINK_REG_PORT_CAP[17:16]=11) (Links to PORT) When arbitrating the PORT type, the port with the higher value becomes the downlink port. For ports, if the tiebreakers of two ports are equal, PORT_CAP will be swapped again. 2. LINK_REG_PORT_CAP[17:16]: Direction LINK_REG_PORT_CAP[17]: This bit is 1, indicating that the port is open.	0

			The port can be configured as an uplink port; LINK_REG_PORT_CAP[16]: This bit is 1, indicating that the port is open. The port can be configured as a downlink port; 3 LINK_REG_PORT_CAP[7:0] : Nub HP Buffers, indicating the number of header packet buffers supported by this device; 4. Other bits are reserved.	

27.2.6.14 LMP Receive Data 0 Register (R32_LINK_LMP_RX_DATA0) Offset Address: 0x58

Bit	name	access	describe	Reset value
[31:0]	LINK_LMP_RX_DATA 0	RO	After LMP reception is complete, HP data is stored in this register. [31:0].	X

27.2.6.15 LMP Receive Data 1 Register (R32_LINK_LMP_RX_DATA1) Offset Address: 0x5C

Name

Bit		access	After	Reset value
[31:0]	LINK_LMP_RX_DATA 1	RO	LMP reception is complete, HP data is stored in this register. [63:32].	X

27.2.6.16 LMP Receive Data 2 Register (R32_LINK_LMP_RX_DATA2) Offset Address: 0x60

Name

Bit		access	After	Reset value
[31:0]	LINK_LMP_RX_DATA 2	RO	LMP reception is complete, HP data is stored in this register. [95:64].	X

27.2.6.17 USB Custom HP Data 0 Register (R32_LINK_LMP_TX_DATA0) Offset Address: 0x64 Name

Bit		access	describe	Reset value
[31:0]	LINK_LMP_TX_DATA 0	RW	User-defined data to send HP [31:0].	X

27.2.6.18 USB Custom HP Data 1 Register (R32_LINK_LMP_TX_DATA1) Offset Address: 0x68 Name

Bit		access	describe	Reset value
[31:0]	LINK_LMP_TX_DATA 1	RW	user-defined data to send HP [63:32].	X

27.2.6.19 USB Custom HP Data 2 Register (R32_LINK_LMP_TX_DATA2) Offset Address: 0x6C

Name

Bit		access	describe	Reset value
[31:0]	LINK_LMP_TX_DATA 2	RW	user-defined data to send HP [95:64].	X

Note:

1. After writing LINK_LMP_TX_DATA2, the hardware will automatically generate a trigger signal to send the information of DATA0/DATA1/DATA2.
go out.

2. Custom sending is applied to packets whose LMP (TX_PORT_CAP/TX_PORT_CFG, etc.)/PING-TP registers are not defined.
gives away.

Chapter 28 Controller Area Network (CAN)

Controller Area Network (CAN) is a high-performance communication protocol used for serial data communication. A CAN controller provides a complete CAN protocol.
The proposed implementation scheme supports CAN protocol 2.0A and 2.0B (active). The CAN controller can be used to build a robust local area network for security.
Distributed real-time control, which processes a large number of data packets with a relatively small CPU load, has a wide range of applications in the industrial and automotive fields.

28.1 Key Features

• Provides 3 CAN controllers, compatible with CAN specifications 2.0A and 2.0B (active). • Programmable
transmission rate, up to 1 Mbit/s. • Supports time-triggered communication to
avoid blocking low-priority messages.
Each CAN group supports three transmit mailboxes. The priority of transmitted messages can be determined by the message identifier or the order of transmission requests, and can be recorded.
Timestamp of the SOF message sent
Each CAN group supports two receive FIFOs with a three-level mailbox depth. A total of 42 filter groups are available for configuration across the three CAN groups.
The unit can be configured in 32-bit or 16-bit mode, mask bit or identifier list mode, which can minimize software intervention in packet filtering.
The FIFO overflow handling method is flexible and can record the timestamp of the received message SOF.
It occupies 4 interrupt vectors, and each interrupt source can be configured independently.

28.2 CAN Controller Operating Modes

The CAN controller can manipulate the SLEEP or INRQ bits in the CANx_CTLR register to achieve initialization mode, sleep mode, and other functions.
It can switch between three working modes: Mode 3 and Normal Mode.

28.2.1 Initialization Mode: After reset,

CAN defaults to sleep mode to reduce power consumption. Message transmission and reception are disabled in this mode. The internal pull-up resistor on the TX pin...
Yes, the TX pin outputs a recessive bit. Setting the INRQ bit in the CANx_CTLR register to 1 requests the CAN controller to enter initialization mode.
When the INAK bit in the CANx_STATR register is automatically set to 1, the initialization state is successfully entered. Similarly, the INRQ bit in the CANx_CTLR register...
Clearing the INAK bit in the CANx_STATR register automatically resolves the request to the CAN controller to exit initialization mode. Successful exit from initialization occurs when the INAK bit is automatically cleared to 0.
Transformation state.

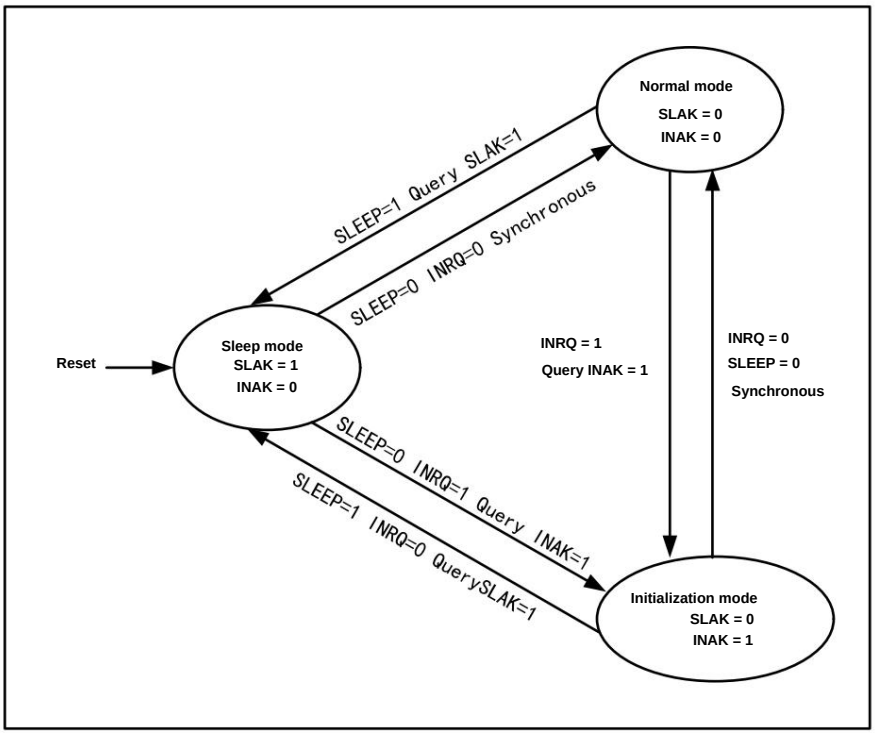
Filter bank initialization can be performed in non-initialization mode, but the FINIT key of register CANx_FCTLR must be checked.
Set the bit to 1, and message reception will be disabled.

28.2.2 Sleep Mode : Setting the

SLEEP bit in the CANx_CTLR register to 1 requests the CAN controller to enter sleep mode. When the SLEEP bit in the CANx_STATR register is set to 1...
If the SNAK bit is automatically set to 1, the CAN controller successfully enters sleep mode. At this time, the clock of the CAN controller stops, but the mailbox register is still accessible.
To enter initialization mode from sleep mode, the SLEEP bit of CANx_CTLR must be cleared to 0 and the INRQ bit set to 1. This is done when the CANx_STATR register...
If the INAK bit is automatically set to 1, the initialization state is complete.

To transition from sleep mode to normal mode, the SLEEP bit of CANx_CTLR must be cleared to 0, and the SNAK bit of the CANx_STATR register must also be cleared.
Automatically resetting to 0 will enter normal mode.

Figure 28-1 CAN Working Mode Switching



28.3 CAN Controller Test Mode

In initialization mode, operations on the SILM and LBKM bits of the CANx_BTIMR register can select a test mode.

Then, by clearing the INRQ bit of the CANx_CTLR register, the system exits initialization mode and enters test mode. Test mode is divided into silent mode and...

There are three modes: mode 1, loop mode 2, and silent loop mode 3.

28.3.1 Silent Mode: Setting the SILM

bit in the CANx_BTIMR register to 1 selects the option to enter silent mode. In this mode, the CAN controller can still receive signals.

It cannot send messages externally; it is always in a recessive position to avoid affecting the bus. However, messages can be controlled by the node it resides in.

The controller receives this information. Silent mode is typically used for CAN bus status analysis.

28.3.2 Loopback Mode: Setting the

LBKM bit in the CANx_BTIMR register to 1 allows selection to enter loopback mode. In this mode, the CAN controller can send external signals.

It can send messages but cannot receive external messages; however, sent messages can be received by the controller of the node it resides in, indicating that the receive filtering mechanism is effective.

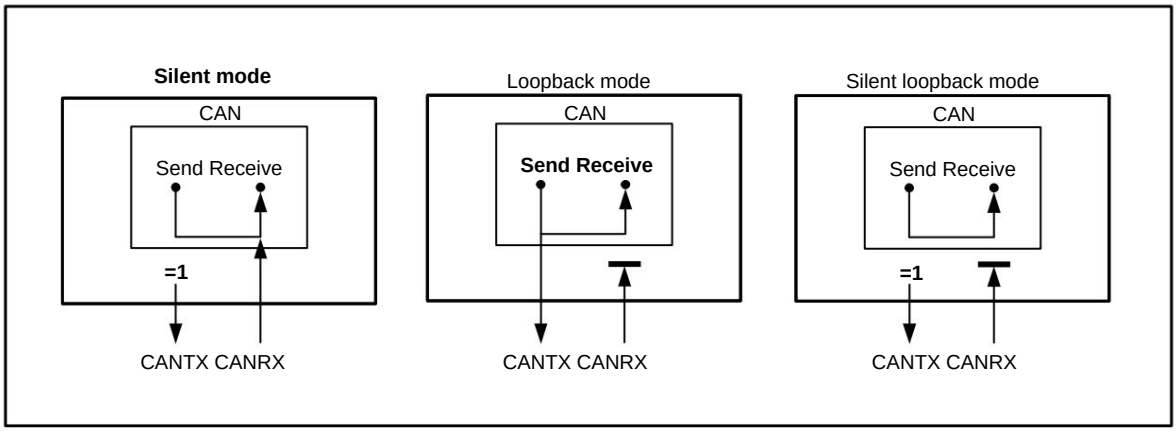
The constant loopback mode is used for transmit and receive testing of CAN controllers.

28.3.3 Silent Loopback Mode

Setting SILM and LBKM bits to 1 in the CANx_BTIMR register allows entry into silent loopback mode. This mode is typically used for CAN control.

The controller is closed for self-test. In this mode, there is no impact on the CAN bus. The RX pin is disconnected from the bus, and the TX pin is set to a recessive bit.

Figure 28-2 Three test modes for the CAN bus



28.4 CAN Controller Operating Status in MCU Debug Mode

When the MCU enters debug mode, the kernel is paused, but the CAN controller can be configured via bits in the debug module.

It is in normal operation or stopped state.

28.5 CAN Controller Functional Description

28.5.1 Sending Process

The sending process is as follows: If any of the three sending mailboxes is empty, the application layer software only has access to the registers of the empty mailboxes.

Write permissions are granted to operate on registers CANx_TXMIRx, CANx_TXMDTRx, CANx_TXMDLRx, and CANx_TXMDHRx.

Configure the message identifier, message length, timestamp, and message data. After the data is ready, configure the TXRQ register in CANx_TXMIRx.

Location 1: Request sent, the mailbox enters registered status and is prioritized; once it becomes the highest priority mailbox, it becomes a scheduled sender.

The system is in a send state, waiting for the CAN bus to become idle; when the CAN bus becomes idle, the message in the pre-determined send mailbox immediately enters the send state; message transmission.

After completion, the mailbox becomes an empty mailbox again, and the RQCP and TXOK bits of the CANx_TSTATR register are set to 1 to indicate that the transmission was successful;

If arbitration fails during transmission, the ALST bit in the CANx_TSTATR register is set to 1; if a transmission error occurs, the TERR bit is set to 1.

28.5.2 Transmission Priority

Transmission priority can be determined by an identifier or the order of transmission requests. Setting the TXFP bit in the CANx_CTLR register to 1 indicates the order of transmission requests.

Determining the order of transmission, sending according to the order of transmission requests is mainly used for segmented transmission; clearing the TXFP bit to 0 determines the transmission order based on identifier priority.

The order is as follows: the smaller the identifier, the higher the priority. In the case of the same identifier, the lower the email address number, the higher the priority.

28.5.3 Transmission Abort Handling A

transmission request can be aborted by setting the ABRQ bit in the CANx_TSTATR register to 1. This applies when the mailbox status is registered or scheduled for transmission.

When the mailbox is in the sending state, the sending request is directly aborted; when the mailbox is in the sending state, the abort request may succeed (stop sending), or it may...

It may fail (transmission complete), and the result can be checked by the TXOK bit of the CANx_TSTATR register.

28.5.4 When the traditional CAN communication bus

is busy, it is easy to cause low-priority messages to be blocked for a long time, or even fail to meet their time limit requirements.

To address this bottleneck, time-triggered protocols have been developed, which have seen significant industrial applications.

The time-triggered mode is designed to work in conjunction with this type of protocol.

In time-triggered mode, two modes are available. To use this mode, automatic retransmission must be disabled via CANx_TTCTLR configuration.

The MODE bit in the register selects between the default mode and the enhanced mode. Setting the TTCM and NART bits in the CANx_CTLR register to 1 enables this mode.

In intermittent trigger mode and with automatic retransmission disabled, the MODE bit of the CANx_TTCTLR register is set to 0 by default, in which case it operates in default mode, internally...

The timer is activated to generate timestamps for the transmit and receive mailboxes. The timer increments at the CAN bit time, and the internal timer is sampled at the sampling point of the start-of-frame bit for both receive and transmit, generating a timestamp. To use enhanced mode, the MODE bit of the CANx_TTCTLR register must be set to 1. In this mode, at least three nodes must exist in the entire CAN network. One node transmits the time base. After receiving the timestamp from this time base node, other nodes reset their internal counters by writing 1 to the TIMRST bit of the CANx_TTCTLR register, thus synchronizing the internal counters. This ensures time synchronization for all CAN nodes except the node transmitting the time base. The data to be transmitted is then written to the transmit mailbox, and the time trigger count values for each node are configured sequentially.

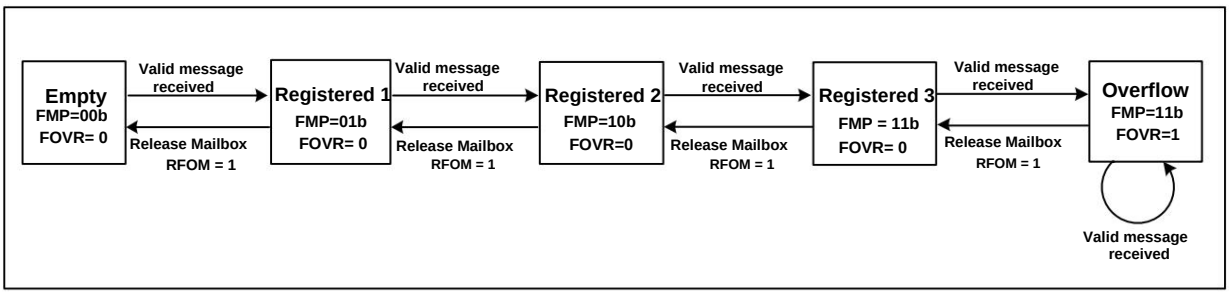
The time-triggered count (TIMCNT in the CANx_TTCNT register) and the final count value of the internal counter (TIMCMV in the CANx_TTCTLR register) are determined by the number of CAN nodes, the CAN communication rate, and the number of bits per frame. After configuration, each node waits for the internal counter to reach the time-triggered count value before triggering the transmission action.

28.5.5 Receiving and Processing Flow

The reception of CAN bus messages is handled by the controller hardware, eliminating the need for MCU intervention and reducing the MCU's processing load. Received messages are stored in two FIFOs with a depth of three mailboxes, according to the settings in the CANx_FAFIFOR register. The application layer can only retrieve valid received messages by reading them from the receive FIFO mailboxes.

Initially, the receive FIFO is empty, and the FMR[1:0] value of the receive FIFO register CANx_RFIFOx is binary 00b. After receiving a valid receive message, it changes to the registration 1 state, and the controller automatically sets the FMR[1:0] of the receive FIFO register CANx_RFIFOx to binary 01b. If the mailbox data registers CANx_RXMDLRx and CANx_RXMDHRx are read at this time, the mailbox is released by setting the RFOM bit of the receive FIFO register CANx_RFIFOx to 1, and the receive FIFO state becomes empty again. If the mailbox is not released in the registration 1 state, after the next valid receive message is received, the receive FIFO state switches to the registration 2 state, and the FMR[1:0] of the receive FIFO register CANx_RFIFOx is automatically set to binary 10b. If the mailbox data register is read and the mailbox is released, the state returns to registration 1. If the mailbox is not released in the registration 2 state, the receive FIFO enters registration 3. The status is as follows: if the message is read and the mailbox is released in the registration 3 status, the system will return to the registration 2 status. If the mailbox is not released in the registration 3 status, the message will inevitably be lost when the next valid message is received.

Figure 28-3 Receive FIFO State Switching Diagram



In the message loss scenario described above, where the receive FIFO is full and message overflow leads to message loss, the FOVR bit of the receive FIFO register CANx_RFIFOx will be automatically set to 1 by hardware for overflow lookup. If the RFLM bit of the CANx_CTLR register is set to 1, the receive FIFO locking function is enabled, and the discarded message is a newly received message; if the RFLM bit of the CANx_CTLR register is cleared to 0, the receive FIFO locking function is disabled, and the last received message among the three original messages in the receive FIFO will be overwritten by the new message. Setting the relevant bits in the

CANx_INTENR register can generate an interrupt when the receive FIFO state changes for more efficient processing.

For details on receiving messages, see Section 28.6 CAN Interrupts.

28.5.6 Receive Message Identifier Filtering

The module has up to 42 filter groups. By configuring these filter groups, each CAN node can receive signals that conform to the filtering rules.

Messages that do not conform to the filtering rules are discarded by the hardware without the need for software intervention.

Each filter group consists of two 32-bit registers, CANx_FxR0 and CANx_FxR1. The bit width of each filter group can be independently configured as either a single 32-bit filter or two 16-bit filters by setting individual bits in the CANx_FSCFGR register. Each filter group can be configured to mask or identifier list mode by setting individual bits in the CANx_FMCFGR register. Each filter group can be enabled or disabled by setting individual bits in the CANx_FWR register. Setting individual bits in the CANx_FAFIFOR register allows you to select which filters to pass.

Which receive FIFO is the message stored in?

As shown in Table 28-1 below, in masked mode, the two registers are the identifier register and the mask register, which need to be matched.

When used together, each bit of the identifier register indicates whether the corresponding bit is expected to be dominant or recessive, and each bit of the mask register indicates whether the corresponding bit is dominant or recessive.

The expected values of the corresponding identifier register bits must be consistent.

Table 28-1 32-bit Masking Mode

Identifier register	CANx_FxR1[31:24]	CANx_FxR1[23:16] device		CANx_FxR1[15:8]	CANx_FxR1[7:0]		
Shielding bit register	CANx_FxR2[31:24]	CANx_FxR2[23:16] device		CANx_FxR2[15:8]	CANx_FxR2[7:0]		
Map STID[10:3]		STID[2:0] EXID[17:13]		EXID[12:5]	EXID[4:0] IDE	RTR 0	

In identifier list mode, both registers are used as identifier registers, and the receive message identifier must be matched with one of these registers.

Only by maintaining consistency can you pass the screening.

Table 28-2 32-bit Identifier List Pattern

Identifier register	CANx_FxR1[31:24]	CANx_FxR1[23:16] device		CANx_FxR1[15:8]	CANx_FxR1[7:0]		
Shielding bit register	CANx_FxR2[31:24]	CANx_FxR2[23:16] device		CANx_FxR2[15:8]	CANx_FxR2[7:0]		
Mapping	STID[10:3]	STID[2:0] EXID[17:13]		EXID[12:5]	EXID[4:0] IDE	RTR 0	

In 16-bit mode, the register set is split into four registers, and the mask bit mode for each filter group can have 2 mask bit modes.

There are four filters, each containing a 16-bit identifier register and a 16-bit mask register; in identifier list mode, there are four...

The registers are all used as identifier registers.

Table 28-3 16-bit Masking Mode

Identifier register n,	CANx_FxR1[15:8]	CANx_FxR1[7:0]			
Mask register n,	CANx_FxR1[31:24]	CANx_FxR1[23:16]			
Identifier register n+1,	CANx_FxR2[15:8]	CANx_FxR2[7:0]			
Mask register n+1,	CANx_FxR2[31:24]	CANx_FxR2[23:16]			
Mapping	STID[10:3]	STID[2:0]	RTR	IDE	EXID[17:15]

Table 28-4 16-bit Identifier List Pattern

Identifier register n	CANx_FxR1[15:8]	CANx_FxR1[7:0]			
Mask register n	CANx_FxR1[31:24]	CANx_FxR1[23:16]			
Identifier register n+1	CANx_FxR2[15:8]	CANx_FxR2[7:0]			
Mask register n+1	CANx_FxR2[31:24]	CANx_FxR2[23:16]			
Mapping	STID[10:3]	STID[2:0]	RTR	IDE	EXID[17:15]

Once a message enters the FIFO mailbox, it is read and stored by the application. The application typically distinguishes message numbers based on message identifiers.

According to the data, the CAN controller provides filter numbers for messages filtered through different filters in the receive FIFO; these numbers are stored in a register.

In the FMI[7:0] of CANx_RXMDTRx, the numbering does not consider whether the filter group is enabled. See the example in Figure 28-4 for the numbering rules.

When a message can pass through multiple filters, the filter number stored in the receiving mailbox is determined according to the filter priority.

The filter priority rules are as follows: This determines which filter number to store.

All 32-bit filters have higher priority than 16-bit filters .

For filters of the same width, filters with identifier lists have higher priority than filters with mask bit patterns .

For filters with the same width and pattern, the filter with the smaller number has higher priority.

As shown in Figure 28-5: When receiving a message, the identifier is first matched against a 32-bit identifier list pattern filter. If no match is found...

The matched data is then compared with a 32-bit mask pattern filter. If no match is found, the data is further compared with a 16-bit identifier list pattern filter.

The process involves matching and filtering; if no match is found, the packet is then compared against a 16-bit mask pattern filter. If still no match is found, the packet is discarded.

If a match is found, the message is stored in the mailbox of the receive FIFO, and the identifier number is stored in the FIMI of the CANx_RXMDTRx register.

Figure 28-4 Example of filter numbering

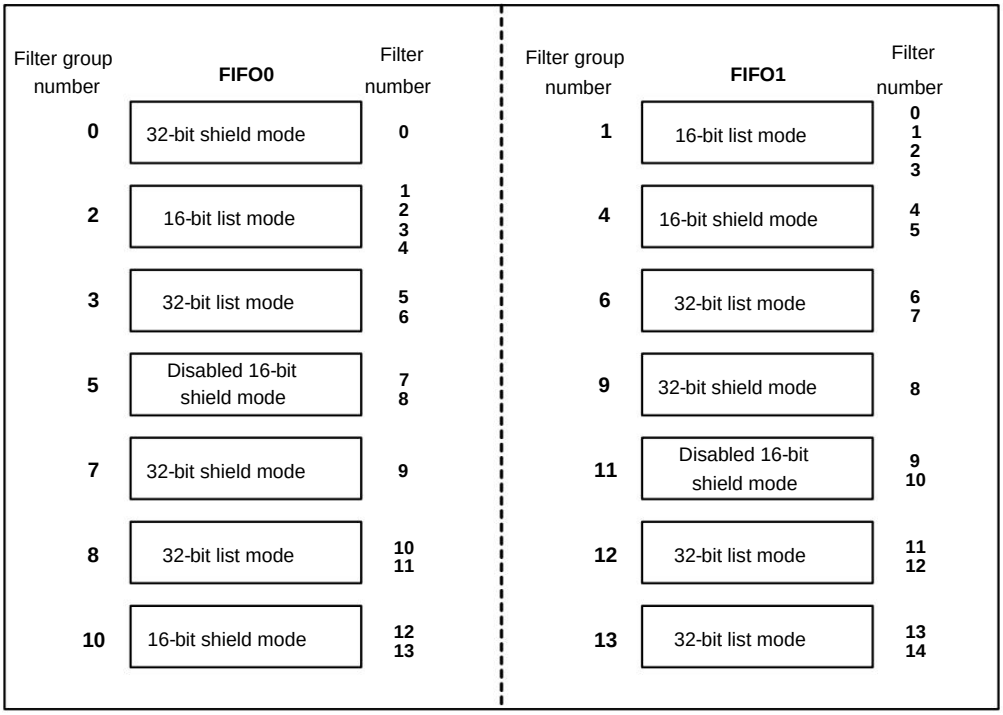
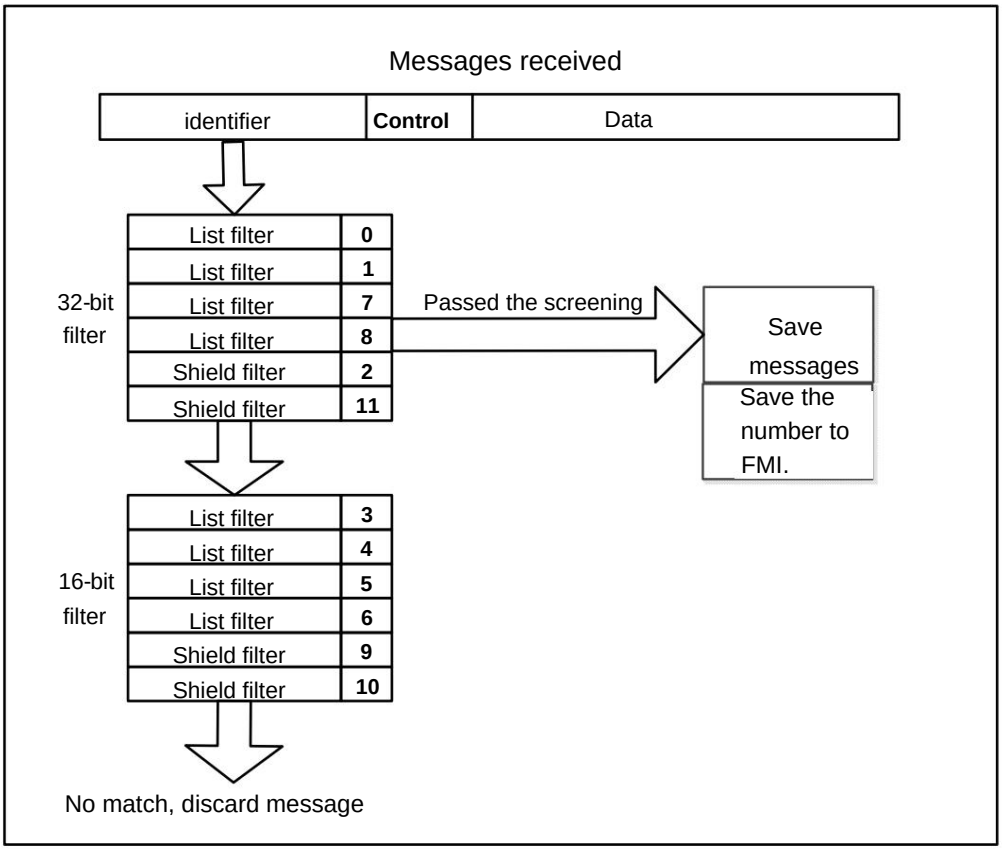


Figure 28-5 Filter example



28.5.7 Error Handling The CAN

controller relies on the status error register CANx_ERRSR for error management on the bus. Status Error Register

In CANx_ERRSR, TEC and REC represent the transmit and receive error counts, respectively, which increase as transmit and receive errors increase.

The value of the signal decreases upon successful transmission, and the stability of the CAN bus can be judged based on these values.

When the TEC and REC values in the CANx_ERRSR status error register are less than 128, the current CAN node is in an error-active state.

It can participate in bus communication normally and issue an active error flag when an error is detected.

When the TEC and REC values in the CANx_ERRSR status error register are greater than 127, the current CAN node is in an error passive state.

Furthermore, when an error is detected, active error flags are not allowed; only passive error flags can be issued.

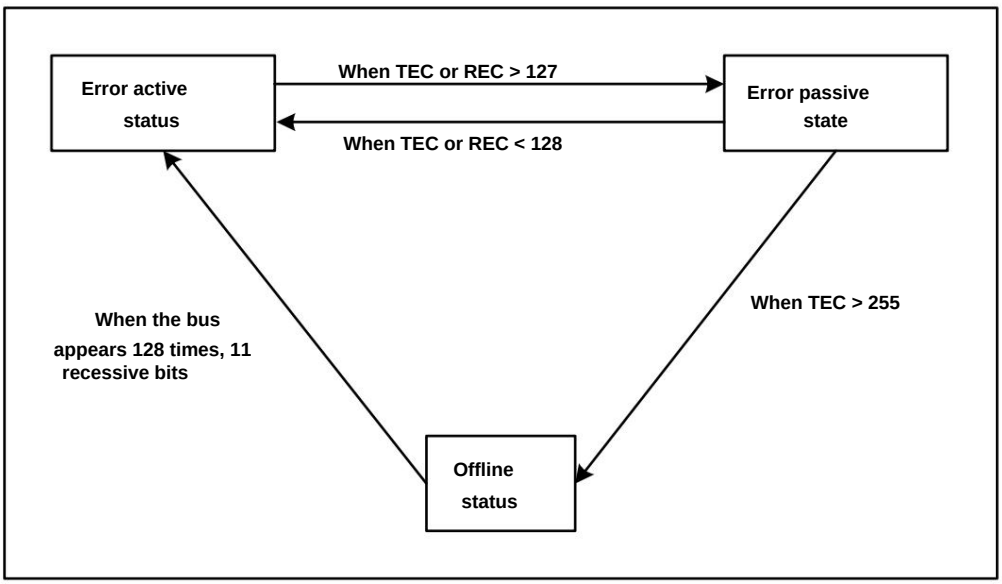
When the TEC value in the status error register CANx_ERRSR is greater than 255, the current CAN node enters offline status.

When the bus detects 128 occurrences of 11 consecutive recessive bits, it reverts to the error active state. This recovery method is controlled by the master control register.

The ABOM bit in the register CANx_CTLR has an impact. If ABOM is set to 1, the hardware automatically exits offline mode. If ABOM is 0, software intervention is required.

The device must enter the initialization mode by operating the INRQ bit, and then exit the initialization before it can exit the offline state.

Figure 28-6 CAN Error State Switching Diagram



28.5.8 The timing sequence

follows the CAN bus standard, dividing each bit's time into four segments: synchronization segment, propagation time segment, phase buffer segment 1, and phase...

Bit buffer segment 2. These segments consist of a minimum time unit Tq. The CAN controller monitors changes on the CAN bus by sampling, via frame start.

Synchronize the edges of the bits.

The CAN controller re-divides the above four segments into three segments, namely:

ÿSynchronization Segment (SS): This is the synchronization segment in the CAN standard, fixed at one minimum time unit. Under normal circumstances, it represents the expected bit...

The transition occurred within this time period.

ÿTime Segment 1 (BS1): Contains the propagation time segment and phase buffer segment 1 in the CAN standard, and can be set to contain 1 to 16 segments.

The small time unit can be automatically extended to compensate for the positive phase drift caused by frequency accuracy errors at different nodes on the CAN bus.

The end of the time period marks the location of the sampling point.

ÿSegment 2 (BS2): This is also known as Phase Buffer Segment 2 in the CAN standard. It can be set to 1 to 8 minimum time units.

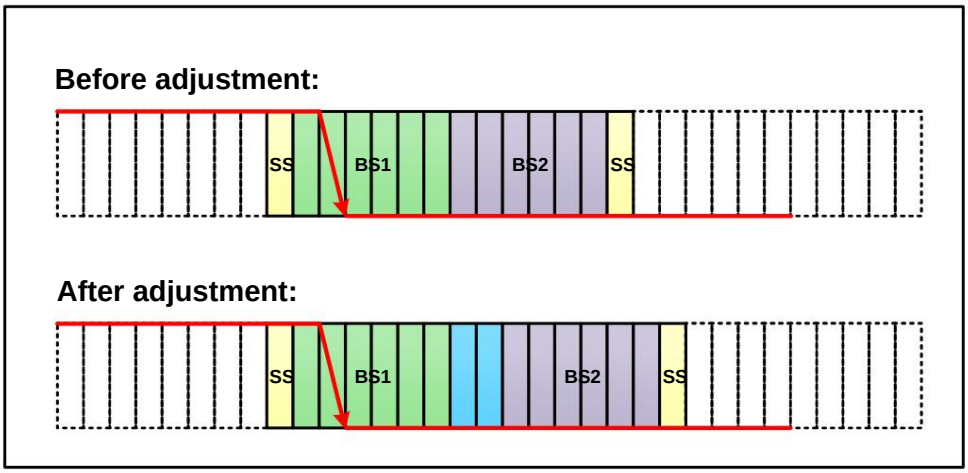
It is automatically shortened to compensate for the negative phase drift caused by frequency accuracy errors at different nodes on the CAN bus.

The resynchronization jump width (SJW) is the maximum number of time units that can be extended or reduced in each bit, and its range can be set to 1.

Up to 4 minimum time units.

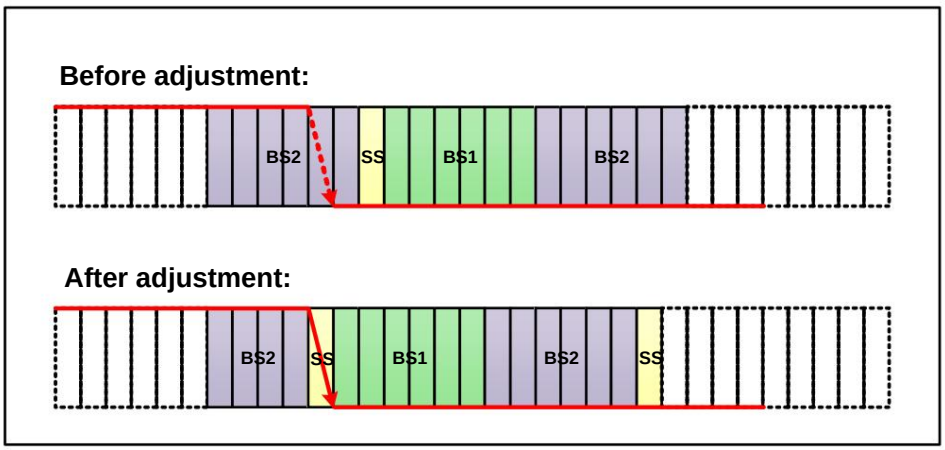
All of the above parameters can be configured in the CAN bus timing register CANx_BTIMR.

Figure 28-7 shows the transition occurring in BS1.



As shown in Figure 28-7, SJW is 2. If a bus level transition is detected during time period 1, then the length of time period 1 needs to be extended, with a maximum extension...
The long SJW delays the location of the sampling point.

Figure 28-8 shows the transition occurring in BS2.



As shown in Figure 28-8, SJW is 2. If a bus level transition is detected during time period 2, then the length of time period 2 needs to be reduced, with a maximum reduction...
Small SJW, thus allowing for earlier location of sampling points.

28.6 CAN Interrupt

The CAN controller has four interrupt vectors: transmit interrupt, FIFO_0 interrupt, FIFO_1 interrupt, error interrupt, and status change interrupt.
Break.

Setting the CAN interrupt enable register CANx_INTENR allows you to enable or disable individual interrupt sources.

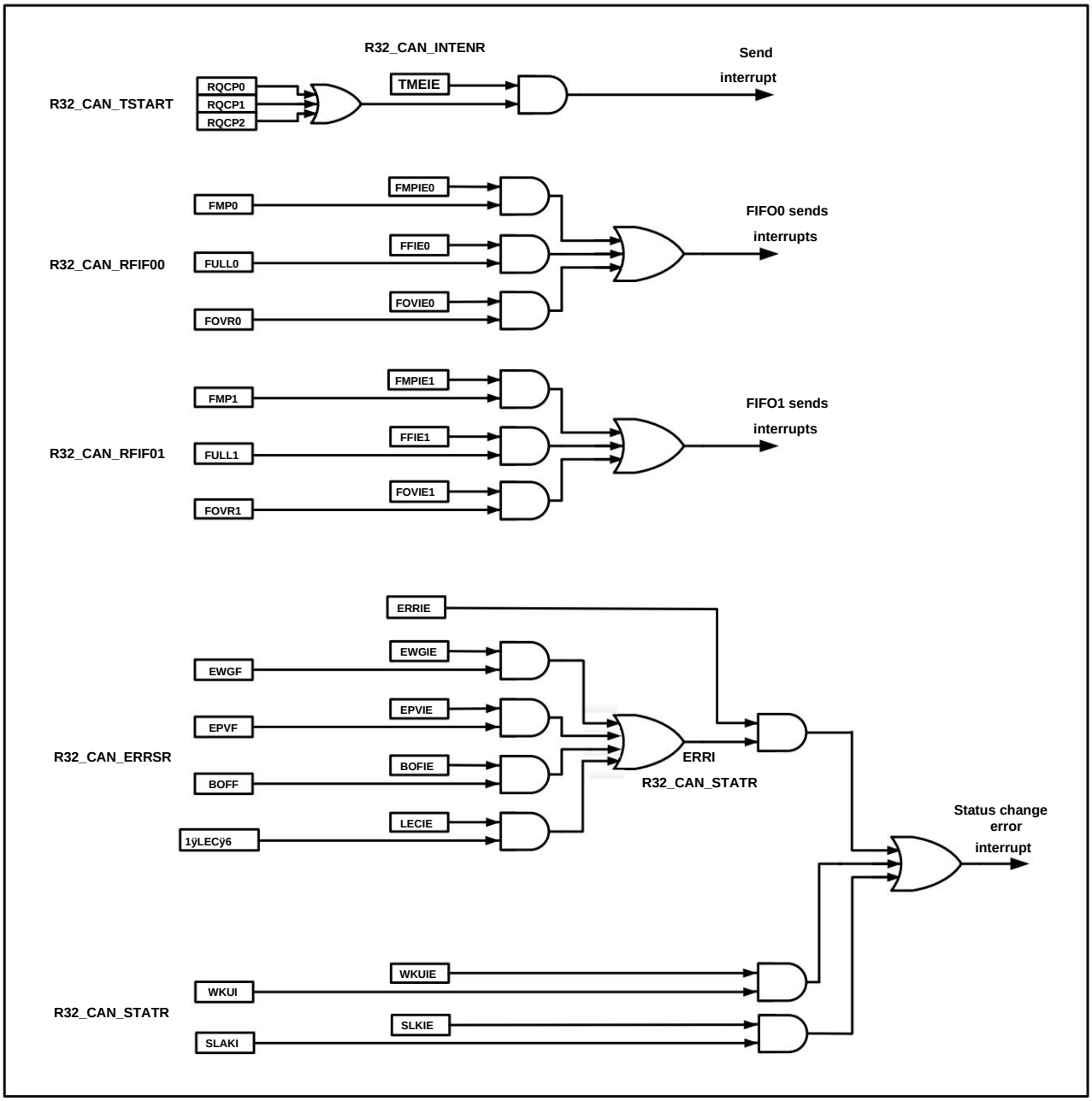
The transmit interrupt is triggered by the transmit mailbox becoming empty. After the interrupt occurs, the RQCP0, RQCP1, and RQCP2 registers of the CANx_TSTATR register are checked.
The RQCP2 bit is used to determine which mailbox became empty.

The FIFO0 interrupt is generated by receiving a new message, the receive mailbox becoming full, and an overflow event. After the interrupt occurs, the CANx_RFIFO0 register is checked.
The FMP0, FULL0, and FOVER0 bits are used to determine which mailbox became empty.

The FIFO1 interrupt is generated by receiving a new message, the receive mailbox becoming full, and an overflow event. After the interrupt occurs, the CANx_RFIFO1 register is checked.
The FMP1, FULL1, and FOVER1 bits are used to determine which mailbox became empty.

Errors and state change interruptions are caused by error, wake-up, and sleep events.

Figure 28-9 CAN Interrupt Logic Diagram



28.7 Register Description

The registers related to the CAN controller must be operated in 32-bit word mode, and the peripheral clock of CAN must be enabled during operation. To avoid the current node affecting the entire CAN bus, application software can only modify the bit timing register in initialization mode. CANx_BTIMR.

Table 28-6 List of CAN1 Related Registers

name	Access Address Description: 0x40006400 CAN1	Reset value
R32_CAN1_CTLR	Master Control Register; 0x40006404 CAN1 Master	0x00010002
R32_CAN1_STATR	Status Register; 0x40006408 CAN1 Transmit	0x0000C002
R32_CAN1_TSTATR	Status Register; 0x4000640C CAN1 Receive FIFO0	0x1C000000
R32_CAN1_RFIFO0	Control and Status Register; 0x00000000; 0x40006410 CAN1 Receive FIFO1 Control and Status	
R32_CAN1_RFIFO1	Register.	

R32_CAN1_INTENR	0x40006414 CAN1 Interrupt Enable Register;	0x00000000
R32_CAN1_ERRSR	0x40006418 CAN1 Error Status Register;	0x00000000
R32_CAN1_BTIMR	0x4000641C CAN1 Bit Timing Register;	0x01230000
R32_CAN1_TTCTLR	0x40006420 CAN1 Time Trigger Control Register;	0x0000FFFF
R32_CAN1_TTCNT	0x40006424 CAN1 Time Trigger Counter Register;	0x00000000
R32_CAN1_TERR_CNT	0x40006428 CAN1 Offline Recovery Error Counter	0x00000000

Table 28-7 List of CAN2 Related Registers

name	Access Address Description: 0x40006800 CAN2	Reset value
R32_CAN2_CTLR	Master Control Register; 0x40006804 CAN2	0x00010002
R32_CAN2_STATR	Master Status Register; 0x40006808 CAN2	0x0000C002
R32_CAN2_TSTATR	Transmit Status Register; 0x4000680C CAN2	0x1C000000
R32_CAN2_RFIFO0	Receive FIFO0 Control and Status Register; 0x00000000; 0x40006810 CAN2 Receive FIFO1	
R32_CAN2_RFIFO1	Control and Status Register; 0x00000000; 0x40006814 CAN2 Interrupt Enable Register;	
R32_CAN2_INTENR	0x40006818 CAN2 Error Status Register;	0x00000000
R32_CAN2_ERRSR	0x4000681C CAN2 Bit Timing Register;	0x00000000
R32_CAN2_BTIMR	0x40006820 CAN2 Time Trigger Control	0x01230000
R32_CAN2_TTCTLR	Register; 0x40006824 CAN Time Trigger Counter Value	0x0000FFFF
R32_CAN2_TTCNT	Register; 0x40006828 CAN2 Offline Recovery Error	0x00000000
R32_CAN2_TERR_CNT	Counter.	0x00000000

Table 28-8 List of CAN3 Related Registers

name	Access Address Description: 0x40007800 CAN3	Reset value
R32_CAN3_CTLR	Master Control Register; 0x40007804 CAN3	0x00010002
R32_CAN3_STATR	Master Status Register; 0x40007808 CAN3	0x0000C002
R32_CAN3_TSTATR	Transmit Status Register; 0x4000780C CAN3	0x1C000000
R32_CAN3_RFIFO0	Receive FIFO0 Control and Status Register; 0x00000000; 0x40007810 CAN3 Receive FIFO1	
R32_CAN3_RFIFO1	Control and Status Register; 0x00000000; 0x40007814 CAN3 Interrupt Enable Register;	
R32_CAN3_INTENR	0x40007818 CAN3 Error Status Register;	0x00000000
R32_CAN3_ERRSR	0x4000781C CAN3 Bit Timing Register;	0x00000000
R32_CAN3_BTIMR	0x40007820 CAN3 Time Trigger Control	0x01230000
R32_CAN3_TTCTLR	Register; 0x40007824 CAN3 Time Trigger Counter	0x0000FFFF
R32_CAN3_TTCNT	Value Register; 0x40007828 CAN3 Offline Recovery Error	0x00000000
R32_CAN3_TERR_CNT	Counter.	0x00000000

Table 28-9 List of CAN1 mailbox-related registers

name	Access address description	Reset value
R32_CAN1_TXMI0R	0x40006580 CAN1 Transmit Mailbox 0 Identifier Register;	X
R32_CAN1_TXMDT0R	0x40006584 CAN1 Transmit Mailbox 0 Data Length and Timestamp Register X; 0x40006588	
R32_CAN1_TXMDL0R	CAN1 Transmit Mailbox 0 Low Byte Data Register; 0x4000658C	X
R32_CAN1_TXMDH0R	CAN1 Transmit Mailbox 0 High Byte Data Register; 0x40006590	X
R32_CAN1_TXMI1R	CAN1 Transmit Mailbox 1 Identifier Register; 0x40006594	X
R32_CAN1_TXMDT1R	CAN1 Transmit Mailbox 1 Data Length and Timestamp Register X	

R32_CAN1_TXMDL1R	0x40006598 CAN1 Transmit Mailbox 1 Low Byte Data Register 0x4000659C		X
R32_CAN1_TXMDH1R	CAN1 Transmit Mailbox 1 High Byte Data Register 0x400065A0 CAN1		X
R32_CAN1_TXMI2R	Transmit Mailbox 2 Identifier Register 0x400065A4 CAN1 Transmit		X
R32_CAN1_TXMDTR2	Mailbox 2 Data Length and Timestamp Register X 0x400065A8 CAN1 Transmit Mailbox 2 Low Byte		
R32_CAN1_TXMDLR2	Data Register 0x400065AC CAN1 Transmit Mailbox 2 High Byte Data Register		X
R32_CAN1_TXMDH2R	0x400065B0 CAN1 Receive FIFO0 Mailbox Identifier Register		X
R32_CAN1_RXMI0R			X
R32_CAN1_RXMDT0R	0x400065B4	CAN1 receives data length and timestamp register from FIFO0 mailbox. instrument	X
R32_CAN1_RXMDL0R	0x400065B8 CAN1 Receive FIFO0 Mailbox Low Byte Data Register; 0x400065BC		X
R32_CAN1_RXMDH0R	CAN1 Receive FIFO0 Mailbox High Byte Data Register; 0x400065C0 CAN1 Receive		X
R32_CAN1_RXMI1R	FIFO1 Mailbox Identifier Register		X
R32_CAN1_RXMDT1R	0x400065C4	CAN1 receives data length and timestamp register from FIFO1 mailbox. instrument	X
R32_CAN1_RXMDL1R	0x400065C8 CAN1 Receive FIFO1 Mailbox Low Byte Data Register; 0x400065CC		X
R32_CAN1_RXMDH1R	CAN1 Receive FIFO1 Mailbox High Byte Data Register		X

Table 28-10 List of CAN2 mailbox-related registers

name	Access address description		Reset value
R32_CAN2_TXMI0R	0x40006980 CAN2 Transmit Mailbox 0 Identifier Register; 0x40006984		X
R32_CAN2_TXMDT0R	CAN2 Transmit Mailbox 0 Data Length and Timestamp Register X; 0x40006988 CAN2 Transmit		
R32_CAN2_TXMDL0R	Mailbox 0 Low Byte Data Register 0x4000698C; CAN2 Transmit Mailbox 0		X
R32_CAN2_TXMDH0R	High Byte Data Register 0x40006990 CAN2 Transmit Mailbox 1 Identifier		X
R32_CAN2_TXMI1R	Register 0x40006994; CAN2 Transmit Mailbox 1 Data Length and		X
R32_CAN2_TXMDT1R	Timestamp Register X; 0x40006998 CAN2 Transmit Mailbox 1 Low Byte Data Register 0x4000699C;		
R32_CAN2_TXMDLR1	CAN2 Transmit Mailbox 1 High Byte Data Register 0x400069A0; CAN2		X
R32_CAN2_TXMDH1R	Transmit Mailbox 2 Identifier Register 0x400069A4; CAN2 Transmit Mailbox		X
R32_CAN2_TXMI2R	2 Data Length and Timestamp Register X; 0x400069A8 CAN2 Transmit		X
R32_CAN2_TXMDTR2	Mailbox 2 Low byte data register 0x400069AC CAN2 Transmit mailbox 2; High byte data register		
R32_CAN2_TXMDLR2	0x400069B0 CAN2 Receive FIFO0 Mailbox identifier register		X
R32_CAN2_TXMDH2R			X
R32_CAN2_RXMI0R			X
R32_CAN2_RXMDT0R	0x400069B4	CAN2 receives data length and timestamp register from FIFO0 mailbox. instrument	X
R32_CAN2_RXMDL0R	0x400069B8 CAN2 Receive FIFO0 Mailbox Low Byte Data Register; 0x400069BC		X
R32_CAN2_RXMDH0R	CAN2 Receive FIFO0 Mailbox High Byte Data Register; 0x400069C0 CAN2 Receive		X
R32_CAN2_RXMI1R	FIFO1 Mailbox Identifier Register		X
R32_CAN2_RXMDT1R	0x400069C4	CAN2 receives data length and timestamp register from FIFO1 mailbox. instrument	X
R32_CAN2_RXMDL1R	0x400069C8 CAN2 Receive FIFO1 Mailbox Low Byte Data Register; 0x400069CC		X
R32_CAN2_RXMDH1R	CAN2 Receive FIFO1 Mailbox High Byte Data Register		X

Table 28-11 List of CAN3 mailbox-related registers

name	Access address description		Reset value

R32_CAN3_TXMI0R	0x40007980 CAN3 Transmit Mailbox 0 Identifier Register; 0x40007984		X
R32_CAN3_TXMDT0R	CAN3 Transmit Mailbox 0 Data Length and Timestamp Register X; 0x40007988 CAN3 Transmit		
R32_CAN3_TXMDL0R	Mailbox 0 Low Byte Data Register 0x4000798C; CAN3 Transmit Mailbox 0		X
R32_CAN3_TXMDH0R	High Byte Data Register 0x40007990 CAN3 Transmit Mailbox 1 Identifier		X
R32_CAN3_TXMI1R	Register 0x40007994 CAN3 Transmit Mailbox 1 Data Length and		X
R32_CAN3_TXMDT1R	Timestamp Register X; 0x40007998 CAN3 Transmit Mailbox 1 Low Byte Data Register 0x4000799C;		
R32_CAN3_TXMDLR1	CAN3 Transmit Mailbox 1 High Byte Data Register 0x400079A0; CAN3		X
R32_CAN3_TXMDH1R	Transmit Mailbox 2 Identifier Register 0x400079A4; CAN3 Transmit Mailbox		X
R32_CAN3_TXMI2R	2 Data Length and Timestamp Register X; 0x400079A8 CAN3 Transmit		X
R32_CAN3_TXMDT2R	Mailbox 2 Low byte data register 0x400079AC CAN3 Transmit mailbox 2; High byte data register		
R32_CAN3_TXMDL2R	0x400079B0 CAN3 Receive FIFO0 Mailbox identifier register		X
R32_CAN3_TXMDH2R			X
R32_CAN3_RXMI0R			X
R32_CAN3_RXMDT0R	0x400079B4	CAN3 receives data length and timestamp register from FIFO0 mailbox. instrument	X
R32_CAN3_RXMDL0R	0x400079B8 CAN3 Receive FIFO0 Mailbox Low Byte Data Register; 0x400079BC		X
R32_CAN3_RXMDH0R	CAN3 Receive FIFO0 Mailbox High Byte Data Register; 0x400079C0 CAN3 Receive		X
R32_CAN3_RXMI1R	FIFO1 Mailbox Identifier Register		X
R32_CAN3_RXMDT1R	0x400079C4	CAN3 receives data length and timestamp register from FIFO1 mailbox. instrument	X
R32_CAN3_RXMDL1R	0x400079C8 CAN3 Receive FIFO1 Mailbox Low Byte Data Register; 0x400079CC		X
R32_CAN3_RXMDH1R	CAN3 Receive FIFO1 Mailbox High Byte Data Register		X

Table 28-12 List of CAN Filter Related Registers

name	Access address	describe	Reset value
R32_CAN1_FCTLR	0x40006600 CAN1 Filter Master Control Register; 0x40006604		0x2A1C0E01
R32_CAN1_FMCFGR	CAN1 Filter Mode Register; 0x40006608 CAN1 Filter Mode		0x00000000
R32_CAN1_FMCFGR1	Register 1; 0x4000660C CAN1 Filter Bit Width Register;		0x00000000
R32_CAN1_FSCFGR	0x40006610 CAN1 Filter Bit Width Register 1; 0x40006614		0x00000000
R32_CAN1_FSCFGR1	CAN1 Filter FIFO Association Register; 0x40006618 CAN1		0x00000000
R32_CAN1_FAFIFOR	Filter FIFO Association Register 1; 0x4000661C CAN1 Filter		0x00000000
R32_CAN1_FAFIFOR1	Activation Register; 0x40006620 CAN1 Filter Activation Register 1;		0x00000000
R32_CAN1_FWR	0x40006640 CAN1 Filter Group 0 Register 1; 0x40006644		0x00000000
R32_CAN1_FWR1	CAN1 Filter Group 0 Register 2; 0x40006648 CAN1 Filter		0x00000000
R32_CAN1_F0R1	Group 1 Register 1; 0x4000664C CAN1 Filter Group 1 Register		X
R32_CAN1_F0R2	2 0x40006650 CAN1 Filter Group 2 Register 1 0x40006654		X
R32_CAN1_F1R1	CAN1 Filter Group 2 Register 2 0x40006658 CAN1 Filter		X
R32_CAN1_F1R2	Group 3 Register 1 0x4000665C CAN1 Filter Group 3 Register		X
R32_CAN1_F2R1	2 0x40006660 CAN1 Filter Group 4 Register 1 0x40006664		X
R32_CAN1_F2R2	CAN1 Filter Group 4 Register 2		X
R32_CAN1_F3R1			X
R32_CAN1_F3R2			X
R32_CAN1_F4R1			X
R32_CAN1_F4R2			X

R32_CAN1_F5R1	0x40006668	CAN1 Filter Group 5 Register 1	X
R32_CAN1_F5R2	0x4000666C	CAN1 Filter Group 5 Register 2	X
R32_CAN1_F6R1	0x40006670	CAN1 Filter Group 6 Register 1	X
R32_CAN1_F6R2	0x40006674	CAN1 Filter Group 6 Register 2	X
R32_CAN1_F7R1	0x40006678	CAN1 Filter Group 7 Register 1	X
R32_CAN1_F7R2	0x4000667C	CAN1 Filter Group 7 Register 2	X
R32_CAN1_F8R1	0x40006680	CAN1 Filter Group 8 Register 1	X
R32_CAN1_F8R2	0x40006684	CAN1 Filter Group 8 Register 2	X
R32_CAN1_F9R1	0x40006688	CAN1 Filter Group 9 Register 1	X
R32_CAN1_F9R2	0x4000668C	CAN1 Filter Group 9 Register 2	X
R32_CAN1_F10R1	0x40006690	CAN1 Filter Group 10 Register 1	X
R32_CAN1_F10R2	0x40006694	CAN1 Filter Group 10 Register 2	X
R32_CAN1_F11R1	0x40006698	CAN1 Filter Group 11 Register 1	X
R32_CAN1_F11R2	0x4000669C	CAN1 Filter Group 11 Register 2	X
R32_CAN1_F12R1	0x400066A0	CAN1 Filter Group 12 Register 1	X
R32_CAN1_F12R2	0x400066A4	CAN1 Filter Group 12 Register 2	X
R32_CAN1_F13R1	0x400066A8	CAN1 Filter Group 13 Register 1	X
R32_CAN1_F13R2	0x400066AC	CAN1 Filter Group 13 Register 2	X
R32_CAN1_F14R1	0x400066B0	CAN1 Filter Group 14 Register 1	X
R32_CAN1_F14R2	0x400066B4	CAN1 Filter Group 14 Register 2	X
R32_CAN1_F15R1	0x400066B8	CAN1 Filter Group 15 Register 1	X
R32_CAN1_F15R2	0x400066BC	CAN1 Filter Group 15 Register 2	X
R32_CAN1_F16R1	0x400066C0	CAN1 Filter Group 16 Register 1	X
R32_CAN1_F16R2	0x400066C4	CAN1 Filter Group 16 Register 2	X
R32_CAN1_F17R1	0x400066C8	CAN1 Filter Group 17 Register 1	X
R32_CAN1_F17R2	0x400066CC	CAN1 Filter Group 17 Register 2	X
R32_CAN1_F18R1	0x400066D0	CAN1 Filter Group 18 Register 1	X
R32_CAN1_F18R2	0x400066D4	CAN1 Filter Group 18 Register 2	X
R32_CAN1_F19R1	0x400066D8	CAN1 Filter Group 19 Register 1	X
R32_CAN1_F19R2	0x400066DC	CAN1 Filter Group 19 Register 2	X
R32_CAN1_F20R1	0x400066E0	CAN1 Filter Group 20 Register 1	X
R32_CAN1_F20R2	0x400066E4	CAN1 Filter Group 20 Register 2	X
R32_CAN1_F21R1	0x400066E8	CAN1 Filter Group 21 Register 1	X
R32_CAN1_F21R2	0x400066EC	CAN1 Filter Group 21 Register 2	X
R32_CAN1_F22R1	0x400066F0	CAN1 Filter Group 22 Register 1	X
R32_CAN1_F22R2	0x400066F4	CAN1 Filter Group 22 Register 2	X
R32_CAN1_F23R1	0x400066F8	CAN1 Filter Group 23 Register 1	X
R32_CAN1_F23R2	0x400066FC	CAN1 Filter Group 23 Register 2	X
R32_CAN1_F24R1	0x40006700	CAN1 Filter Group 24 Register 1	X
R32_CAN1_F24R2	0x40006704	CAN1 Filter Group 24 Register 2	X
R32_CAN1_F25R1	0x40006708	CAN1 Filter Group 25 Register 1	X
R32_CAN1_F25R2	0x4000670C	CAN1 Filter Group 25 Register 2	X
R32_CAN1_F26R1	0x40006710	CAN1 Filter Group 26 Register 1	X
R32_CAN1_F26R2	0x40006714	CAN1 Filter Group 26 Register 2	X

R32_CAN1_F27R1	0x40006718 CAN1 Filter Group 27 Register 1 0x4000671C	X
R32_CAN1_F27R2	CAN1 Filter Group 27 Register 2 0x40006720 CAN1 Filter	X
R32_CAN1_F28R1	Group 28 Register 1 0x40006724 CAN1 Filter Group 28	X
R32_CAN1_F28R2	Register 2 0x40006728 CAN1 Filter Group 29 Register 1	X
R32_CAN1_F29R1	0x4000672C CAN1 Filter Group 29 Register 2 0x40006730	X
R32_CAN1_F29R2	CAN1 Filter Group 30 Register 1 0x40006734 CAN1 Filter	X
R32_CAN1_F30R1	Group 30 Register 2 0x40006738 CAN1 Filter Group 31	X
R32_CAN1_F30R2	Register 1 0x4000673C CAN1 Filter Group 31 Register 2	X
R32_CAN1_F31R1	0x40006740 CAN1 Filter Group 32 Register 1 0x40006744	X
R32_CAN1_F31R2	CAN1 Filter Group 32 Register 2 0x40006748 CAN1 Filter	X
R32_CAN1_F32R1	Group 33 Register 1 0x4000674C CAN1 Filter Group 33	X
R32_CAN1_F32R2	Register 2 0x40006750 CAN1 Filter Group 34 Register 1	X
R32_CAN1_F33R1	0x40006754 CAN1 Filter Group 34 Register 2 0x40006758	X
R32_CAN1_F33R2	CAN1 Filter Group 35 Register 1 0x4000675C CAN1 Filter	X
R32_CAN1_F34R1	Group 35 Register 2 0x40006760 CAN1 Filter Group 36	X
R32_CAN1_F34R2	Register 1 0x40006764 CAN1 Filter Group 36 Register 2	X
R32_CAN1_F35R1	0x40006768 CAN1 Filter Group 37 Register 1 0x4000676C	X
R32_CAN1_F35R2	CAN1 Filter Group 37 Register 2 0x40006770 CAN1 Filter	X
R32_CAN1_F36R1	Group 38 Register 1 0x40006774 CAN1 Filter Group 38	X
R32_CAN1_F36R2	Register 2 0x40006778 CAN1 Filter Group 39 Register 1	X
R32_CAN1_F37R1	0x4000677C CAN1 Filter Group 39 Register 2 0x40006780	X
R32_CAN1_F37R2	CAN1 Filter Group 40 Register 1 0x40006784 CAN1 Filter	X
R32_CAN1_F38R1	Group 40 Register 2 0x40006788 CAN1 Filter Group 41	X
R32_CAN1_F38R2	Register 1 0x4000678C CAN1 Filter Group 41 Register 2	X
R32_CAN1_F39R1		X
R32_CAN1_F39R2		X
R32_CAN1_F40R1		X
R32_CAN1_F40R2		X
R32_CAN1_F41R1		X
R32_CAN1_F41R2		X

Note: The clock for the CAN filter table is provided by [Provider Name]. Therefore, to configure or use the filter table function, the clock must be enabled.CAN1 CAN1

28.7.1 CANx Master Control Register (CANx_CTLR) (x=1/2/3) Offset Address: 0x00

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16	
Reserved					CFGCANM	DBF

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RST	Reserved							TTCM	ABOM	AWUM	NART	RFLM	TXFP	SLEEP	INRQ

Bit	name	access	describe	Reset value
[31:18] Reserved		RO is reserved.		0
17	CFGCANM	RW Configuration for CAN Offline Recovery Time:		0

			1: Restore from offline to normal operation conforming to the CAN protocol; 0: It's faster to recover from offline to normal.	
16	DBF	RW	Debugging whether to disable CAN bus operation: 1. During debugging, CAN transmission and reception are disabled, but the receive FIFO is active. The control and read/write operations are all normal; 0: During debugging, the CAN controller works normally.	1
15	RST	RW1	CAN controller software reset request; writing 0 to this bit is invalid. 1: Reset the CAN controller. After the reset, the controller will proceed... Enter sleep mode, then the hardware will automatically reset to 0; 0: CAN controller is in normal state.	0
[14:8] Reserved		RO reserves the		0
7	TTCM	RW	option to allow time-triggered mode: 1: Enable time-triggered mode; 0: Disable time-triggered mode. The time-triggered mode is mainly used in conjunction with the TTCAN protocol.	0
6	ABOM	RW	Automatic offline exit control: 1: The hardware detected 128 consecutive recessive bits of 11 each, from... Automatically exit offline status; 0: Requires software manipulation of the INRQ bit in the CAN_CTLR register. Set to 1 and then clear to 0. When 128 consecutive instances of 11 hidden characters are detected... After completing the sexual position, exit offline mode.	0
5	AWUM	RW	CAN controller automatic wake-up enable: 1: When a message is detected, the hardware automatically wakes up, and the registers... The SLEEP and SLAK bits of CAN_STATR are automatically cleared to 0; 0: SLEEP requires software manipulation of the CAN_CTLR register. Clear the bit to 0 and wake up the CAN controller.	0
4	NART	RW	Automatic message retransmission is disabled: 1. Regardless of whether the message is sent successfully or not, it can only be sent once. 0: The CAN controller will retransmit until the transmission is successful.	0
3	RFLM	RW	Enable FIFO message locking mode: 1: When the receive FIFO overflows, unread messages in the received mailbox remain. If the mailbox is not released, newly received messages are discarded; 0: When the receive FIFO overflows, the received mailbox messages have not been received. When the mailbox is not released, newly received messages will be overwritten. Cover the original message. Note: This bit is only used for traditional purposes.CAN.	0
2	TXFP	RW	Email priority selection: 1. Priority is determined by the order in which requests are sent; 0: Priority is determined by the message identifier.	0
1	SLEEP	RW	Sleep mode request bit: 1: Setting this value to 1 requests the CAN controller to enter sleep mode. After the activity is completed, the controller enters sleep mode. If AWUM Position 1 means that when a message is received, the controller will press SLEEP. Clear bits to 0; 0: After the software is cleared to 0, the controller exits sleep mode.	1

0	INRQ	RW	Initialization mode request bits: 1: Setting to 1 requests the CAN controller to enter initialization mode. After the initial activity is completed, the controller enters the initialization mode, and the hardware... Set the INAK bit of the CAN_STATR register to 1; 0: Setting the value to 0 requests the CAN controller to exit initialization mode and proceed. Entering normal mode, the hardware controls the CAN_STATR register. Clear the INAK bit to 0.	0

28.7.2 CANx Master Status Register (CANx_STATR) (x=1/2/3) Offset Address: 0x04

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				RX	SAMP	RXM	TXM		Reserved		SLAKI	WKUI	ERRI	SLAK	INAK

Bit	name	access	describe	Reset value
[31:12] Reserved		RO is reserved.		0
11	RX	The current actual level of the RO CAN controller receive pin RX.		1
10	SAMP	RO	The voltage of the previous receive bit on the CAN controller receive pin RX flat.	1
9	RXM	RO	Receive mode query bit: 1: The current CAN controller is in receive mode; 0: The current CAN controller is in non-receive mode.	0
8	TXM	RO	Send mode query bit: 1: The current CAN controller is in transmit mode; 0: The current CAN controller is in non-transmit mode.	0
[7:5] Reserved		RO is reserved.		0
4	SLAKI	RW1Z	When sleep interrupt is enabled, i.e., when the CAN_INTENR register is... When SLKIE is set to 1, the interrupt generation flag is set; writing 1 clears it. Writing 0 is invalid. 1: When entering sleep mode, an interrupt is generated, and the hardware is set to 1; 0: When exiting sleep mode, the system can be cleared via hardware or software.	0
3	WKUI	RW1	Wake up the interrupt flag. This is done when the CAN_INTENR register is set to... When WKUI is in position 1, if the CAN controller is in sleep mode When the SOF bit is detected, the hardware sets it to 1. The software sets it to 1. 0, setting it to 0 is invalid.	0
2	ERRI	RW1	Error interrupt status flag bit. When the CAN_INTENR register... When the ERRIE is set to 1, an error occurs and a state change occurs. The program has failed. Setting this bit to 1 and clearing it to 0 is invalid; setting it to 0 is also invalid.	0
1	SLAK	RO	Sleep mode indicator. 1: The CAN controller is in sleep mode; 0: The CAN controller is not in sleep mode.	1
0	INAK	RO Initialization Mode Indicator Bit.		0

			1: The CAN controller is in initialization mode; 0: The CAN controller is operating in non-initialization mode.	

28.7.3 CANx Transmit Status Register (CANx_TSTATR) (x=1/2/3) Offset Address: 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOW2	LOW1	LOW0	TME2	TME1	TME0	CODE[1:0]		ABRQ2			Reserved		TERR2	ALST2	TXOK2	RQCP2	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABRQ1	Reserved			TERR1	ALST1	TXOK1	RQCP1	ABRQ0		Reserved		TERR0	ALST0	TXOK0	RQCP0	

Bit	name	access	describe	Reset value
31	LOW2	RO	Indicates the lowest priority flag for sending to mailbox 2: 1: This indicates that sending to email address 2 has the lowest priority; 0: Indicates that the priority of sending to email address 2 is not the lowest.	0
30	LOW1	RO	Send the lowest priority flag to mailbox 1: 1: This indicates that sending to email address 1 has the lowest priority; 0: Indicates that the priority of sending to email address 1 is not the lowest.	0
29	LOW0	RO	Send the lowest priority flag to mailbox 0: 1: Indicates that sending to email address 0 has the lowest priority; 0: Indicates that the priority of sending to email address 0 is not the lowest.	0
28	TME2	RO	Indicates the empty flag for sending to mailbox 2: 1: Indicates that there are no messages waiting to be sent to email address 2; 0: Indicates that email address 2 has messages waiting to be sent.	1
27	TME1	RO	The empty flag indicating that the email address is being sent to mailbox 1 is empty. 1: Indicates that there are no messages waiting to be sent to email address 1; 0: Indicates that there are messages waiting to be sent to email address 1.	1
26	TME0	RO	The null flag indicating that the email address is sent to mailbox 0 is empty. 1: Indicates that there are no messages waiting to be sent to mailbox 0; 0: Indicates that there are messages waiting to be sent to email address 0.	1
[25:24] CODE[1:0]		RO	Email ID: If more than one email address is empty, it means the next one will be empty. The email address; when the email address is completely empty, it indicates the lowest priority. The email address.	0
twenty three	ABRQ2	RW1	Send a request to email address 2 to abort the transmission. The software is set to 1, which allows... Abort the sending request from mailbox 2 when the sent message is cleared. If the hardware is cleared to 0, and mailbox 2 is emptied, setting the software to 1 will have no effect.	0
[22:20] Reserved		RO Reserved.		0
19	TERR2	RW1	Send email 2 failed to send flag; if email 2 fails to send... If transmission fails, this bit is automatically set to 1. Software sets it to 1; clearing it to 0... Write 0. Invalid.	0
18	ALST2	RW1	The arbitration failure flag is set to send to email address 2. If a low arbitration priority causes a transmission failure, this bit will be automatically set to 1. Setting the software to 1 clears it to 0; writing 0 to the software is invalid.	0

17	TXOK2	RW1	Email 2 successfully sent: 1: Last time it was sent successfully; 0: Last send failed. Setting the software to 1 clears it to 0; writing 0 to the software is invalid.	0
16	RQCP2	RW1	Send email 2 to request completion flag; when sending email 2 This bit is automatically set to 1 when the send or abort request is completed. (Software) Setting the file to 1 and clearing it to 0 is invalid; writing 0 to the software is also invalid.	0
15	ABRQ1	RW1	Send a request to email address 1 to abort the transmission. The software is set to 1, which allows... Abort the sending request from mailbox 1 when the sent message is cleared. Hardware clearing to 0. Software writing to 0 is ineffective.	0
[14:12] Reserved		RO Reserved.		0
11	TERR1	RW1	Sending email 1 fails flag; if sending email 1 fails... If transmission fails, this bit is automatically set to 1. Software sets it to 1; clearing it to 0... Write 0. Invalid.	0
10	ALST1	RW1	The arbitration failure flag is set for email address 1. If a low arbitration priority causes a transmission failure, this bit will be automatically set to 1.	0
9	TXOK1	RW1	Successful delivery flag for email address 1: 1: Last time it was sent successfully; 0: Last send failed. Setting the software to 1 clears it to 0; writing 0 to the software is invalid.	0
8	RQCP1	RW1	Send email 1 to request completion flag; when sending email 1 This bit is automatically set to 1 when the sending or aborting request is completed. Setting the software to 1 clears it to 0; writing 0 to the software is invalid.	0
7	ABRQ0	RW1	Send a request to email address 0 to abort the transmission. The software can be set to 1. Abort the sending request for mailbox 0 when the sent message is cleared. Hardware clearing to 0. Software writing to 0 is ineffective.	0
[6:4] Reserved		RO Reserved.		0
3	TERR0	RW1	Send email address 0: Sending failure flag. If transmission fails, this bit is automatically set to 1. Software sets it to 1; clearing it to 0... Write 0. Invalid.	0
2	ALST0	RW1	The arbitration failure flag is set to send to email address 0. If a low arbitration priority causes a transmission failure, this bit will be automatically set to 1. Setting the software to 1 clears it to 0; writing 0 to the software is invalid.	0
1	TXOK0	RW1	Sending email 0 Successful sending flag: 1: Last time it was sent successfully; 0: Last send failed. Setting the software to 1 clears it to 0; writing 0 to the software is invalid.	0
0	RQCP0	RW1	Send email 0 request completion flag; when sending email 0 This bit is automatically set to 1 when the send or abort request is completed. (Software) Setting the file to 1 and clearing it to 0 is invalid; writing 0 to the software is also invalid.	0

28.7.4 CANx Receive FIFO 0 Status Register (CANx_RFIFO0) (x=1/2/3)

Offset address: 0x0C

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16

Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										RFOM0	FOVR0	FULL0	Reserved	FMP0[1:0]	

Bit	name	access	describe	Reset value
[31:6]	Reserved	RO	is reserved. If	0
5	RFOM0	RW1	the software sets this position to 1, then the current receive FIFO_0 is released. Email messages are automatically cleared to 0 after release; software writes 0 (no response). effect.	0
4	FOVR0	RW1	Receive the overflow flag of FIFO_0. When there are three overflow flags in FIFO_0... When a new message is received, the hardware sets the bit to 1. This bit needs to be... To clear a program from 1 to 0, you must use the software to set it to 1. Writing 0 to the software is invalid.	0
3	FULL0	RW1	Receive the full flag of FIFO_0. When there are three [full flags] in FIFO_0... When sending a message, the hardware sets this bit to 1. This bit needs to be cleared from 1 to 0 via software. Writing 0 to the software is invalid.	0
2	Reserved	RO	is reserved.	0
[1:0]	FMP0[1:0]	RO	receives the number of FIFO_0 messages.	0

28.7.5 CANx Receive FIFO 1 Status Register (CANx_RFIFO1) (x=1/2/3)

Offset address: 0x10

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										RFOM1	FOVR1	FULL1	Reserved	FMP1[1:0]	

Bit name		access	describe	Reset value
[31:6]	Reserved	RO	is reserved. If	0
5	RFOM1	RW1	the software sets this position to 1, then the current receive FIFO_1 is released. Email messages are automatically cleared to 0 after release; software writes 0 (no response). effect.	0
4	FOVR1	RW1	Receive the overflow flag of FIFO_1. When there are three overflow flags in FIFO_1... When a new message is received, the hardware sets the bit to 1. This bit needs to be... To clear a program from 1 to 0, you must use the software to set it to 1. Writing 0 to the software is invalid.	0
3	FULL1	RW1	Receive the full flag of FIFO_1. When there are three [full flags] in FIFO_1... When sending a message, the hardware sets this bit to 1. This bit needs to be cleared from 1 to 0 via software. Writing 0 to the software is invalid.	0
2	Reserved	RF	reserved.	0
[1:0]	FMP1[1:0]	RO	receives the number of FIFO_1 messages.	0

28.7.6 CANx Interrupt Enable Register (CANx_INTENR) (x=1/2/3)

Offset address: 0x14

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
Reserved					SLKIE WKUIE

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRIE	Reserved			LECIE	BOFIE	EPVIE	EWGIE	Reserved	FOVIE1	FFIE1	FMPIE1	FOVIE0	FFIE0	FMPIE0	TMEIE

Bit	name	access	describe	Reset value
[31:18] Reserved		RO is reserved.		0
17	SLKIE	RW	Sleep interruption enable bit: 1: An interruption occurs when entering sleep mode; 0: No interruption occurs when entering sleep mode.	0
16	WKUIE	RW	Wake-up interrupt enable bit: 1: An interrupt is generated when the CAN controller is woken up; 0: No interrupt is generated when the CAN controller is woken up.	0
15	ERRIE	RW	Error interrupt enable bit, CAN error interrupt always enable bit. 1: An interrupt is generated when the CAN controller encounters an error; 0: No interrupt is generated when the CAN controller generates an error.	0
[14:12] Reserved		RF reserved.		0
11	LECIE	RW	Last error number interrupt enable bit: 1: When an error is detected, the hardware updates LEC[2:0], and updates... When the ERRI bit is 1, an error interrupt is triggered. 0: When an error is detected, the hardware updates LEC[2:0], otherwise it does not. The new ERRI bit does not trigger an error interrupt.	0
10	BOFIE	RW	Offline interrupt enable bit: 1: When entering offline mode, update the ERRI bit to 1, triggering... Error interruption; 0: When entering offline mode, the ERRI bit is not updated, and no action is taken. An error occurred and the connection was interrupted.	0
9	EPVIE	RW	Error passive interrupt enable bit: 1: When entering an error passive state, update the ERRI bit to 1. Trigger an error interrupt; 0: When entering an error passive state, the ERRI bit is not updated. No error interruption is triggered.	0
8	EWGIE	RW	Error warning interrupt enable bit: 1. When the number of errors reaches the warning threshold, update the ERRI bit. A value of 1 triggers an error interrupt; 0: When the number of errors reaches the warning threshold, ERRI is not updated. The bit does not trigger an error interrupt.	0
7	Reserved	RF reserved.		0
6	FOVIE1	RW	Receive FIFO_1 overflow interrupt enable bit: 1: When FIFO_1 overflows, a FIFO_1 interrupt is triggered; 0: When FIFO_1 overflows, the FIFO_1 interrupt is not triggered.	0

5	FFIE1	RW	Receive FIFO_1 full interrupt enable bit: 1: When FIFO_1 is full, trigger an interrupt in FIFO_1; 0: When FIFO_1 is full, the FIFO_1 interrupt is not triggered.	0
4	FMPIE1	RW	Receive FIFO_1 message registration interrupt enable bit: 1: Triggered when FIFO_1 updates the FMP bit and it is not 0. FIFO_1 interrupt; 0: When FIFO_1 updates the FMP bit and it is not 0, no trigger is given. FIFO_1 interrupted.	0
3	FOVIE0	RW	Receive FIFO_0 overflow interrupt enable bit: 1: When FIFO_0 overflows, a FIFO_0 interrupt is triggered; 0: When FIFO_0 overflows, the FIFO_0 interrupt is not triggered.	0
2	FFIE0	RW	Receive FIFO_0 full interrupt enable bit: 1: When FIFO_0 is full, trigger FIFO_0 interrupt; 0: When FIFO_0 is full, the FIFO_0 interrupt is not triggered.	0
1	FMPIE0	RW	Receive FIFO_0 message registration interrupt enable bit: 1: Triggered when FIFO_0 updates the FMP bit and it is not 0. FIFO_0 interrupt; 0: When FIFO_0 updates the FMP bit and it is not 0, no event is triggered. FIFO_0 interrupt.	0
0	TMEIE	RW	Sending email address was interrupted. 1: An interrupt occurs when the sending mailbox is empty; 0: No interruption occurs when the sending email address is empty.	0

28.7.7 CANx Error Status Register (CANx_ERRSR) (x=1/2/3) Offset Address: 0x18

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
REC[7:0]			TEC[7:0]		

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

Bit	name	access	describe	Reset value
[31:24]	REC[7:0]	RO	Receive error counter. When a CAN receive error occurs, this count is adjusted according to the error condition. The counter increments by 1 or 8; upon successful reception, the counter decrements by 1 or 8. Set to 120 (error count value is greater than 127). Counter value When the value exceeds 127, CAN enters an error passive state.	0
[23:16]	TEC[7:0]	RO	Send error counter. When a CAN transmission error occurs, this count is adjusted according to the error condition. The counter increments by 1 or 8; upon successful transmission, the counter decrements by 1 or 8. Set to 120 (error count value is greater than 127). Counter value When the value exceeds 127, CAN enters an error passive state.	0
[15:7]	Reserved	RO is reserved.		0

[6:4]	LEC{2:0}	RW	The previous error code. When a transmission error is detected on the CAN bus, the controller... Error handling settings: Set to 000b when messages are sent and received correctly. 000: No error; 001: Bit stuffing error; 010: FORM format error; 011: ACK confirmation error; 100: Recessive bit error; 101: Dominant bit error; 110: CRC error; 111: Software settings. When an application detects an error, it typically sets the code to... 111b indicates that a code name update can be detected.	0
3	Reserved	RO reserved.		0
2	BOFF	RO	Offline status flag: When the CAN controller enters offline mode, the hardware automatically resets. 1; When exiting offline mode, the hardware is automatically reset to 0.	0
1	EPVF	RO	Error passive flag: When the transmit/receive error counter reaches the passive error threshold, that is When the value is greater than 127, the hardware is set to 1.	0
0	EWGF	RO	Error warning flags: When the transmit/receive error counter reaches the warning threshold, i.e., greater than... When the value is 96, the hardware is set to 1.	0

28.7.8 CANx Bit Timing Register (CANx_BTIMR) (x=1/2/3) Offset Address: 0x1C

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
SILM	LBKM	Reserved	SJW[3:0]	TS2[3:0]	TS1[3:0]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BTR_TS1_T				Reserved		BRP[9:0]									

Bit	name	access	describe	Reset value
31	SILM	RW	Silent mode setting: 1: Enter silent mode; 0: Exit silent mode.	0
30	LBKM	RW	Loopback mode setting: 1: Enter loopback mode; 0: Exit loop mode.	0
[29:28] Reserved		RO is reserved.		0
[27:24] SJW[3:0]		RW	The resynchronization jump width setting value is defined. When resynchronizing, the bit that can be extended or reduced is... The actual maximum number of small time units is [value]. (SJW[3:0]+1), the range can be set to 1 to 16 most	0001b

		Small time unit.	
[23:20] TS2[3:0]	RW	Time period 2 setting value. The definition specifies how many minimum time units time period 2 occupies. The actual value is (TS2[3:0]+1).	0010b
[19:16] TS1[3:0]	RW	Time period 1 setting value. The definition specifies how many minimum time units time period 1 occupies. The actual value is (TS1[3:0]+1).	0011b
[15:12] BTR_TS1_T	RW	For traditional CAN, CLAS_LONG_TS1=0, then TS1 is... TS[3:0] (4bit); CLAS_LONG_TS1=1, then TS1 It is TS[1:0]+BTR_TS1_T[15:12](6bit).	0
[11:10] Reserved	RO Reserved.		0
[9:0] BRP[9:0]	RW	Minimum time unit length setting. Tq = (BRP[9:0]+1) × tHCLKS	0

28.7.9 CANx Time Trigger Control Register (CANx_TTCTLR) (x=1/2/3) Offset Address: 0x20

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16	
Reserved					MODE	TIMRST

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

Bit	name	access	describe	Reset value
[31:18] Reserved		RO reserved.		0
17	MODE	RW	Time-triggered mode selection bit: 1: Enhanced Mode; 0: Default mode.	0
16	TIMRST	WZ	Internal counter reset control bit: Write 1 resets the internal counter; the hardware automatically	0
[15:0] TIMCMV[15:0]		clears it to 0. RW returns the final value of the internal counter.		0xFFFF

28.7.10 CANx Time Trigger Count Register (CANx_TTCNT) (x=1/2/3)

Offset address: 0x24

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16	
Reserved						

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

Bit	name	access	describe	Reset value
[31:16] Reserved		RO is reserved.		0
[15:0] TIMCNT[15:0]		RW time trigger count value.		0

28.7.11 CANx Offline Recovery Error Counter (CANx_TERR_CNT) (x=1/2/3)

Offset address: 0x28

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

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

Bit name	access	describe	Reset value
[31:9] Reserved	RO	Reserved.	0
[8:0] TX_ERR_CNT	RW	Current offline recovery error count value; modifying this count value will... Resume immediately from offline.	0

28.7.12 CANx Transmit Mailbox Identifier Register (CANx_TXMIyR) (x=1/2/3, y=0/1/2)

Offset address: 0x180, 0x190, 0x1A0

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
STID[10:0]/EXID[28:18]											EXID[17:13]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXID[12:0]													IDE	RTR	TXRQ

Bit	name	access	describe	Reset value
[31:21]	STID[10:0]/ EXID[28:18]	RW	is the high 11 bits of the standard or extended identifier.	X
[20:3] EXID[17:0]		RW	The lower 18 bits of the extended	X
2	IDE	RW	identifier. Identifier selection flags: 1: Use extended identifiers; 0: Select the standard identifier.	X
1	RTR	RW	Remote frame (also known as remote control frame) selection flag: 1: The current frame is a remote frame; 0: The current frame is a data frame.	X
0	TXRQ	RW	Data transmission request flag: When the software is set to 1, it requests to send data from the email address. When the mailbox is empty, the hardware is reset to 0.	0

28.7.13 CANx Transmit Mailbox Data Length and Timestamp Register (CANx_TXMDTyR) (x=1/2/3, y=0/1/2)

Offset address: 0x184, 0x194, 0x1A4

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
TIME[15:0]														

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

Bit	name	access	describe	Reset value
[31:16] TIME[15:0]		RW	is a 16-bit timer value used to send the message at the SOF time.	X
[15:9] Reserved		RO	Reserved.	0
8	TGT	RW	Message timestamp transmission selection flag. This bit is set in TTCM. 1. Valid when the message length is 8. 1: Send timestamp, the value is the immediate value of TIME[15:0]. The last two bytes of the 8-byte message were replaced; 0: Do not send timestamps.	X
[7:4] Reserved		RO	is reserved.	0
[3:0] DLC[3:0]		RW	The data length of the data frame or the data length requested by the remote frame. The data length can be set from 0 to 64.	0

28.7.14 CANx Transmit Mailbox Low Byte Data Register (CANx_TXMDLyR) (x=1/2/3, y=0/1/2)

Offset address: 0x188, 0x198, 0x1A8

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
DATA3[7:0]											DATA2[7:0]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA1[7:0]								DATA0[7:0]							

Bit	name	access	describe	Reset value
[31:24] DATA3[7:0]		RW	sends the content of data byte 3.	X
[23:16] DATA2[7:0]		RW	sends the content of data byte 2.	X
[15:8] DATA1[7:0]		RW	sends the content of data byte 1.	X
[7:0] DATA0[7:0]		RW	sends the content of data byte 0.	X

28.7.15 CANx Transmit Mailbox High Byte Data Register (CANx_TXMDHyR) (x=1/2/3, y=0/1/2)

Offset address: 0x18C, 0x19C, 0x1AC

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
DATA7[7:0]											DATA6[7:0]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA5[7:0]								DATA4[7:0]							

Bit	name	access	describe	Reset value
[31:24] DATA7[7:0]		RW	sends the content of data byte 7.	X
[23:16] DATA6[7:0]		RW	sends the content of data byte 6.	X
[15:8] DATA5[7:0]		RW	sends the content of data byte 5.	X
[7:0] DATA4[7:0]		RW	sends the content of data byte 4.	X

28.7.16 CANx Receive Mailbox Identifier Register (CANx_RXMIyR) (x=1/2/3, y=0/1) Offset Address: 0x1B0, 0x1C0

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16

STID[10:0]/EXID[28:18]	EXID[17:13]

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

Bit	name	access	describe	Reset value
[31:21]	STID[10:0]/ EXIDH[28:18]	RO: The high 11 bits of the standard or extended identifier.		X
[20:3] EXIDL[17:0]		The lower 18 bits of the RO extended		X
2	IDE	RO	identifier. Identifier selection flags. 1: Use extended identifiers; 0: Select the standard identifier.	X
1	RTR	RO	Remote frame (also known as remote control frame) selection flag. 1: The current frame is a remote frame; 0: The current frame is a data frame.	X
0	Reserved	RO is reserved.		0

28.7.17 CANx Receive Mailbox Data Length and Timestamp Register (CANx_RXMDTyR) (x=1/2/3, y=0/1)

Offset address: 0x1B4, 0x1C4

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
TIME[15:0]														

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

Bit	name	access	describe	Reset value
[31:16] TIME[15:0]		RO is a 16-bit timer value used to receive the message SOF time.		0
[15:7] Reserved		RO is reserved.		X
6	RES	RO is the RES bit of the currently received frame.		0
5	ESI	RO: ESI bit of the currently received frame.		0
4	BRS	RO: BRS bit of the current received frame.		0
[3:0] DLC[3:0]		RO	Length of the received message data. DLC=0-8: Indicates a data frame length of 0 to 8, with remote frames being... 0.	X

28.7.18 CANx Receive Mailbox Low Byte Data Register (CANx_RXMDLyR) (x=1/2/3, y=0/1) Offset Address: 0x1B8

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
DATA3[7:0]											DATA2[7:0]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA1[7:0]								DATA0[7:0]							

Bit	name	Access description: RO receives 3 bytes of data	Reset value
[31:24]	DATA3[7:0]	from the message.	X
[23:16]	DATA2[7:0]	RO receives the second data byte of the message.	X
[15:8]	DATA1[7:0]	RO receives the first data byte of the message.	X
[7:0]	DATA0[7:0]	RO receives the first 0 data bytes of the message.	X

28.7.19 CANx Receive Mailbox High Byte Data Register (CANx_RXMDHyR) (x=1/2/3, y=0/1) Offset Address: 0x1BC

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
DATA7[7:0]											DATA6[7:0]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA5[7:0]								DATA4[7:0]							

Bit	name	Access description: RO receives 7 bytes of data	Reset value
[31:24]	DATA7[7:0]	from the message.	X
[23:16]	DATA6[7:0]	RO receives 6 bytes of data from the message.	X
[15:8]	DATA5[7:0]	RO receives 5 bytes of data from the message.	X
[7:0]	DATA4[7:0]	RO receives 4 data bytes of the message.	X

28.7.20 CANx Filter Master Control Register (CANx_FCTLR) (x=1) Offset Address: 0x200

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
Reserved											CAN3SB[5:0]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CAN2SB[4:0]					Reserved							FINIT

Bit	name	access	describe	Reset value
[31:22]	Reserved	RO	is reserved.	0x2A
[21:16]	CAN3SB[5:0]	RW	CAN3 filter start group (CAN3SB[5:0]+1). 000000: Reserved; 000001: CAN3 filter group 2~41; 000010: CAN3 filter group 3~41; ... 100111: CAN3 filter group 40~41; 101000: CAN3 filter group 41; 101001: Reserved; 101010: CAN3 does not participate in filter group allocation; Other: Reserved.	0x1C
[15:13]	Reserved	RO	is reserved.	0
[12:8]	CAN2SB[4:0]	RW	CAN2 filter start group (CAN2SB[4:0]).	0xE

			00000: CAN1 does not participate in filter group allocation; CAN2 passes through. Filter group 0~CAN3SB; 00001: CAN1 filter group 0; CAN2 filter group 1~CAN2EB; 00010: CAN1 filter group 0~1; CAN2 filter group 2~CAN2EB; 00011: CAN1 filter group 0~2; CAN2 filter group 3~CAN2EB; ... 11011: CAN1 filter group 0~26; CAN2 filter group 27~CAN2EB; 11100: CAN1 filter group 0~27; CAN2 not involved. Filter group allocation; 11101: CAN1 filter group 0~27; CAN2 filter group 29~CAN2EB; ... 11111: CAN1 filter group 0~27; CAN2 filter group 31~CAN2EB. Note: When CAN3SB[5:0]=42, CAN2EB=41; When CAN3SB[5:0]=not in 42, CAN2EB=CAN3SB[5:0].	
[7:1] Reserved		RO	Reserved.	0
0	FINIT	RW	Filter initialization mode enable flag. 1: The filter group is in initialization mode; 0: The filter group is in normal mode.	1

Note: 1. When CAN2SB[4:0] is equal to 28, the value range of CAN3SB[5:0] is 27~40,

2. When CAN2SB[4:0] is not equal to 28, the value of CAN2SB[4:0] must be less than or equal to the value of CAN3SB[5:0].

28.7.21 CANx Filter Mode Register (CANx_FMCFGR) (x=1) Offset Address: 0x204

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				FBM27	FBM26	FBM25	FBM24	FBM23	FBM22	FBM21	FBM20	FBM19	FBM18	FBM17	FBM16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FBM15	FBM14	FBM13	FBM12	FBM11	FBM10	FBM9	FBM8	FBM7	FBM6	FBM5	FBM4	FBM3	FBM2	FBM1	FBM0		

Bit	name	access	describe	Reset value
[31:28] Reserved		RO	is reserved. The	0
[27:0] FBMx		RW	operating mode control bit for filter bank x; FINIT must be 1 for operation to be enabled. It can be written to. 1: The registers of filter group x are in identifier list mode; 0: The register of filter group x is in mask mode.	0

28.7.22 CANx Filter Mode Register (CANx_FMCFGR1) (x=1)

Offset address: 0x208

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	FBM41	FBM40	FBM39	FBM38	FBM37	FBM36	FBM35	FBM34	FBM33	FBM32	FBM31	FBM30	FBM29	FBM28	

Bit name	access	describe	Reset value
[31:14] Reserved	RO is reserved. The		0
[13:0] FBMx	RW	operating mode control bit for filter bank x; FINIT must be 1 for operation to be enabled. It can be written to. 1: The registers of filter group x are in identifier list mode; 0: The register of filter group x is in mask mode.	0

28.7.23 CANx Filter Bit Width Register (CANx_FSCFGR) (x=1) Offset Address:

0x20C

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved				FSC27	FSC26	FSC25	FSC24	FSC23	FSC22	FSC21	FSC20	FSC19	FSC18	FSC17	FSC16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FSC15	FSC14	FSC13	FSC12	FSC11	FSC10	FSC9	FSC8	FSC7	FSC6	FSC5	FSC4	FSC3	FSC2	FSC1	FSC0

Bit	name	access	describe	Reset value
[31:28] Reserved		RO is reserved.		0
[27:0] FSCx		RW	The bit width control bit for filter group x; FINIT must be 1 for writing to be possible. enter. 1: The register for filter group x is a single 32-bit register; 0: The registers of filter group x are two 16-bit registers.	0

28.7.24 CANx Filter Bit Width Register (CANx_FSCFGR1) (x=1) Offset Address:

0x210

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	FSC41	FSC40	FSC39	FSC38	FSC37	FSC36	FSC35	FSC34	FSC33	FSC32	FSC31	FSC30	FSC29	FSC28	

Bit name	access	describe	Reset value
[31:14] Reserved	RO is reserved.		0
[13:0] FSCx	RW	The bit width control bit for filter group x; FINIT must be 1 for writing to be possible. enter. 1: The register for filter group x is a single 32-bit register; 0: The registers of filter group x are two 16-bit registers.	0

28.7.25 CANx Filter FIFO Associated Register (CANx_FAFIFOR) (x=1)

Offset Address: 0x214

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved							FFA27	FFA26	FFA25	FFA24	FFA23 FFA22 FFA21 FFA20	FFA19	FFA18	FFA17 FFA16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FFA15	FFA14	FFA13	FFA12	FFA11	FFA10	FFA9	FFA8	FFA7	FFA6	FFA5	FFA4	FFA3	FFA2	FFA1	FFA0

Bit	name	access	describe	Reset value
[31:28]	Reserved	RO	is reserved.	0
[27:0]	FFAx	RW	The associated FIFO control bit of filter group x, FINIT is 1. Only then can it be written. 1: Filter group x is associated with FIFO_1; 0: Filter group x is associated with FIFO_0.	0

28.7.26 CANx Filter FIFO Associated Register (CANx_FAFIFOR1) (x=1)

Offset Address: 0x218

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	FFA41	FFA40	FFA39	FFA38	FFA37	FFA36	FFA35	FFA34	FFA33	FFA32	FFA31	FFA30	FFA29	FFA28	

Bit	name	access	describe	Reset value
[31:14]	Reserved	RO	reserved.	0
[13:0]	FFAx	RW	Associated FIFO control bit for filter group x, FINIT is 1. Only then can it be written. 1: Filter group x is associated with FIFO_1; 0: Filter group x is associated with FIFO_0.	0

28.7.27 CANx Filter Activation Register (CANx_FWR) (x=1)

Offset address: 0x21C

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved							FACT 27	FACT 26	FACT 25	FACT twenty four	FACT twenty three	FACT twenty two	FACT twenty one	FACT 20	FACT 19	FACT 18	FACT 17	FACT 16

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

Bit	name	access	describe	Reset value
[31:28]	Reserved	RO	is reserved.	0

[27:0] FACTx		RW	The activation control bit for filter group x. 1: Filter group x is activated; 0: Filter group x is disabled.	0

28.7.28 CANx Filter Activation Register (CANx_FWR1) (x=1)

Offset address: 0x220

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		FACT 41	FACT 40	FACT 39	FACT 38	FACT 37	FACT 36	FACT 35	FACT 34	FACT 33	FACT 32	FACT 31	FACT 30	FACT 29	FACT 28

Bit name		access	describe	Reset value
[31:14] Reserved		RO	Reserved.	0
[13:0] FACTx		RW	Activation control bit for filter group x. 1: Filter group x is activated; 0: Filter group x is disabled.	0

28.7.29 CANx filter group filter register (CANx_FiRy) (x=1, i=0-41, y=1/2) offset address: 0x240-0x38C

31	30	29	28	27	26	25	24	23	twenty two	twenty one	2019	18	17	16
FB31	FB30	FB29	FB28	FB27	FB26	FB25	FB24	FB23	FB22	FB21	FB20 FB19	FB18	FB17	FB16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FB15	FB14	FB13	FB12	FB11	FB10	FB9	FB8	FB7	FB6	FB5	FB4	FB3	FB2	FB1	FB0

Bit	name	access	describe	Reset value
[31:0] FB		RW	The flag bit in the register of the filter bank must be 1 for FINIT to be active. It can be written to. **Identifier patterns:** 1: The expected level of the corresponding bit is a recessive bit; 0: The corresponding bit is expected to be a dominant bit. **Masking mode:** 1. It must be consistent with the corresponding identifier register bit; 0: Does not need to be consistent with the corresponding identifier register bit.	0

Chapter 29 Digital Image Interface (DVP)

The Digital Video Port (DVP) is used to connect to a camera module to acquire image data streams. It provides 8/10/12...

It supports parallel interface communication, with a maximum pixel clock input frequency of 150MHz. It also supports image data organized in its original line and frame formats.

It supports formats such as YUV and RGB, as well as compressed image data in formats like JPEG, and can receive external 8-bit, 10-bit, and 12-bit video feeds.

The head module outputs a high-speed parallel data stream. During reception, synchronization primarily relies on VSYNC and HSYNC signals. Image cropping functionality is supported.

29.1 Key Features

ÿConfigurable 8/10/12-bit data width modes

Supports YUV and RGB data formats .

Supports JPEG compressed data .

Built -in FIFO, supports DMA transfer

Supports double -buffered reception

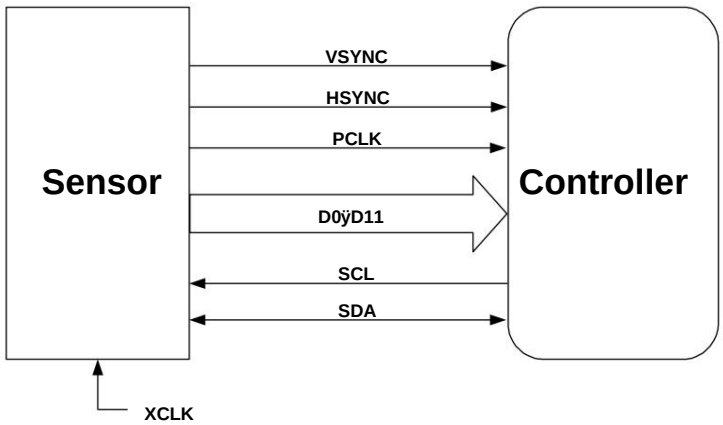
Supports cropping function

Supports continuous mode and snapshot mode .

29.2 Functional Description

29.2.1 Connecting to the sensor

Figure 29-1 DVP Interface Connection



ÿ PLCK (Pixel clock): Pixel clock, each clock cycle corresponds to one pixel of data (uncompressed data). External DVP interface transmission.

The sensor output HCLK clock supports a maximum of 150MHz.

HSYNC (horizontal synchronization): Horizontal synchronization signal.

ÿ VSYNC (vertical synchronization): Frame synchronization signal.

ÿ DATA: Pixel data or compressed data, with a bit width of 8/10/12 bits.

XCLK : The reference clock for the sensor, which can be provided by the microcontroller or externally, and is generally provided by a crystal oscillator.

• I2C Interface: Used to configure sensor registers. I2C timing communication can be simulated via a controlled general-purpose GPIO port, or it can be used...

Hardware I2C interface operation.

Table 29-1 DVP Pins

name	Signal type description
PLCK	Pixel clock input
DATA[11:0]	Pixel data input
VSYNC	Frame synchronization signal input

HSYNC	Line synchronization signal input

29.2.2 Working Timing The

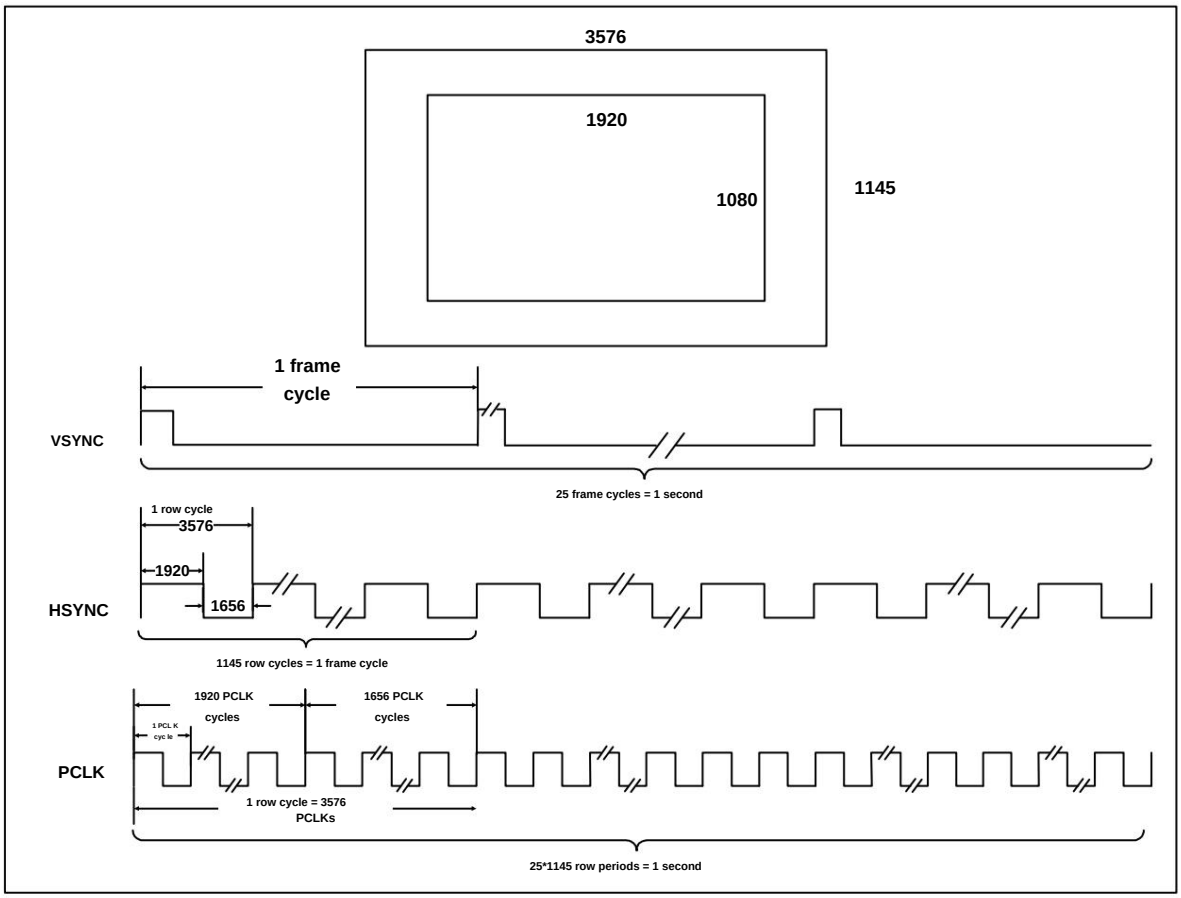
digital signal data stream output by the DVP interface sensor module generally has a certain relationship with the image size. The following example illustrates this.

Example of image data.

As shown in Figure 29-2, the sensor internally displays a complete image size of 3576*1145. After internal scaling, the final image size is...

The image data output from the interface is 1920*1080 in size, and the image refresh rate is 25.

Figure 29-2 Timing Description



HCLK is the time it takes for one pixel to travel, so HSYNC is 3576 times longer than HCLK. Out of 3576 pixels, only 1920 are transmitted.

The pixels are valid; the sensor does not transmit data for the remaining 1656 pixels. VSYNC is the frame synchronization signal, so VSYNC...

The time is 3576*1145 times that of HCLK; similarly, the sensor only transmits data within the effective pixel time of 1920*1080 pixels. For example...

If the sensor transmits JPEG compressed data, there may be no need to use the HSYNC signal.

The relationship between DVP interface signals and image data is primarily explained in the datasheet of the selected sensor.

29.2.3 Description of RGB/YUV/JPEG compressed data formats

RGB

The three primary colors are red, green, and blue.

YUV

The luminance signal is Y, and the chromaticity signals are U and V, Pb and Pr, Cb and Cr.

ÿ JPEG (Joint Photographic Experts Group)

There is lossy compression, but the lost portion is not easily perceptible to the human eye. It utilizes the human eye's ability to detect high-frequency information in computer color processing.

The characteristic of insensitivity. Remove visually redundant information (spatial redundancy) and remove redundant information in the data itself (structural redundancy).

29.3 Applications of Digital Image Interfaces

29.3.1 Digital Image Interface Configuration

Description 1) In DVP data reception, each frame of data is stored alternately by BUF0 and BUF1, starting from BUF0. For RGB and YUV data...

The hardware resets and selects BUF0 each time the frame signal changes from an invalid level to an valid level. Once a line of data is filled, the stream will switch.

The BUF1 is replaced to achieve alternating storage; for JPEG compressed data, the hardware will set BUF0 and BUF1 according to the configured DMA receive length.

The switching threshold of BUF1.

2) When the data bus width is 10 bits or 12 bits, the system will automatically perform unsigned expansion on each received data.

The 16 bits are then stored.

3) The R16_DVP_ROW_NUM and R16_DVP_COL_NUM registers must match the actual image size output by the sensor.

4) In RGB video stream mode, R16_DVP_COL_NUM represents the number of valid HCLK cycles for one row of data, and R16_DVP_ROW_NUM represents...

This indicates the number of rows contained in a frame of image data; in JPEG image mode, R16_DVP_COL_NUM is used to configure the DMA length.

In this case, the R16_DVP_ROW_NUM register has no effect.

29.3.2 Application Description of Digital Image Interface

When using a digital image interface to receive image data, the relevant control registers for the DVP must be correctly configured to match the image sensor's...

The pattern matches; the specific steps are as follows:

1) Clear the RB_DVP_ALL_CLR and RB_DVP_RCV_CLR fields using the R8_DVP_CR1 register.

2) Configure the image mode, data bit width, HCLK polarity, HSYNC polarity, and VSYNC polarity through the R8_DVP_CR0 register to make it...

Matches the output of SENSOR.

3) Configure registers R16_DVP_ROW_NUM and R16_DVP_COL_NUM based on the effective image pixels output by the configured image sensor.

To match the output of the sensor, in JPEG image mode, simply configure the R16_DVP_COL_NUM register.

4) Configure the DMA receive address through the R32_DVP_DMA_BUF0/1 register.

5) If using snapshot mode, the RB_DVP_CM field needs to be configured through the R8_DVP_CR1 register to enable snapshot mode.

6) If using the clipping mode, the RB_DVP_CROP and RB_DVP_FCRC fields need to be configured via the R8_DVP_CR1 register to enable clipping.

The clipping function and frame capture rate are controlled by configuring R16_DVP_HOFFCNT, R16_DVP_VST, R16_DVP_CAPCNT, and

R16_DVP_VLINE sets the size of the cropped image.

7) As needed, enable the corresponding interrupt via the R8_DVP_IER register, configure the interrupt priority via the interrupt controller PFIC, and enable...

DVP interrupt is possible.

8) Enable DMA through the R8_DVP_CR1 register and enable the DVP interface through the R8_DVP_CR0 register.

9) Wait for the relevant receive interrupt to occur and process the received data in a timely manner.

29.3.3 Capture Mode Description

The DVP interface supports two capture modes: snapshot (single frame) mode and continuous mode.

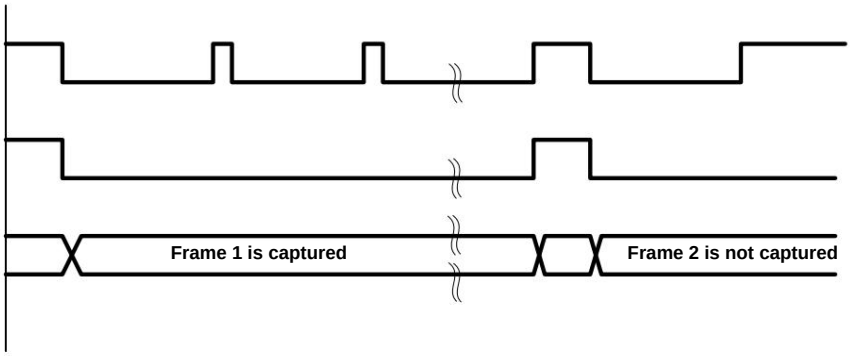
29.3.3.1 Snapshot Mode In this

mode, only a single frame is captured (RB_DVP_CM in the R8_DVP_CR1 register is set to 1). After enabling the DVP interface, ...

Once the system detects the start of a frame, it will begin sampling image data. After receiving a complete frame of data, the DVP interface will be closed.

(Clear the RB_DVP_ENABLE field of the R8_DVP_CR0 register to 0). The single-frame capture waveform in snapshot mode is shown in Figure 29-3.

Figure 29-3 Single-frame capture waveform in snapshot mode

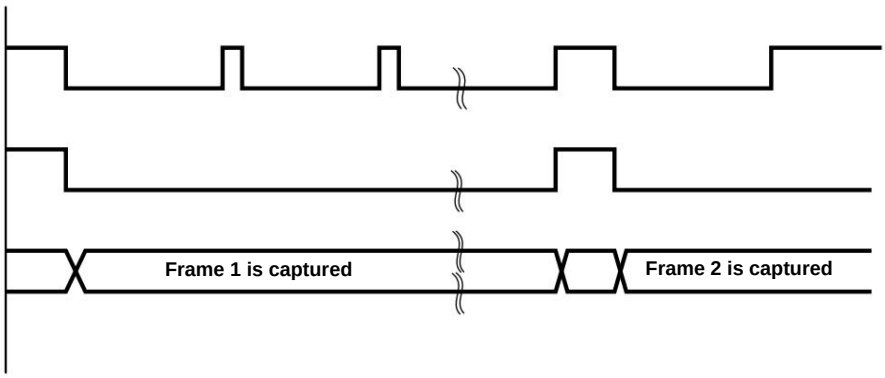


29.3.3.2 Continuous Mode In this

mode (RB_DVP_CM in register R8_DVP_CR1 is cleared to 0), when the DVP interface is enabled, register R8_DVP_CR0...

Before the RB_DVP_ENABLE field of the device is cleared to 0, it will continuously sample data for each frame. The frame capture waveform in continuous mode is shown in Figure 29-4.

Figure 29-4 Frame capture waveform in continuous mode



29.3.4 Description of the cropping function

DVP can use the cropping function to extract a rectangular window from the received image, with the starting coordinates of this rectangular window being the top left corner of the rectangle.

X coordinate R16_DVP_HOFFCNT, Y coordinate R16_DVP_VST) and window size (R16_DVP_CAPCNT represents the horizontal dimension, ...

R16_DVP_VLINE (representing the vertical dimension) is configurable. The coordinates and size of the cropping window are shown in Figure 29-5. The cropping window data capture waveform...

The shape is shown in Figure 29-6.

Figure 29-5 Coordinates and size of the cropping window

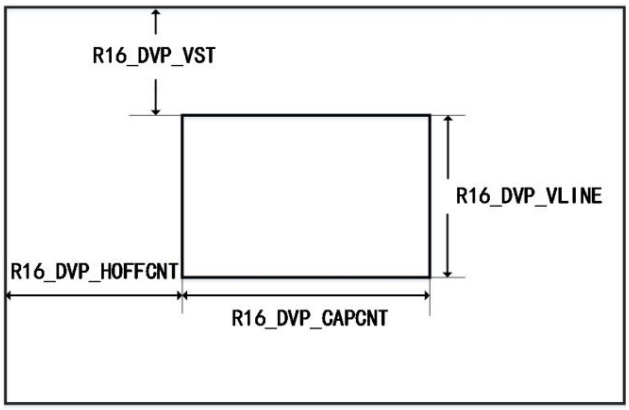
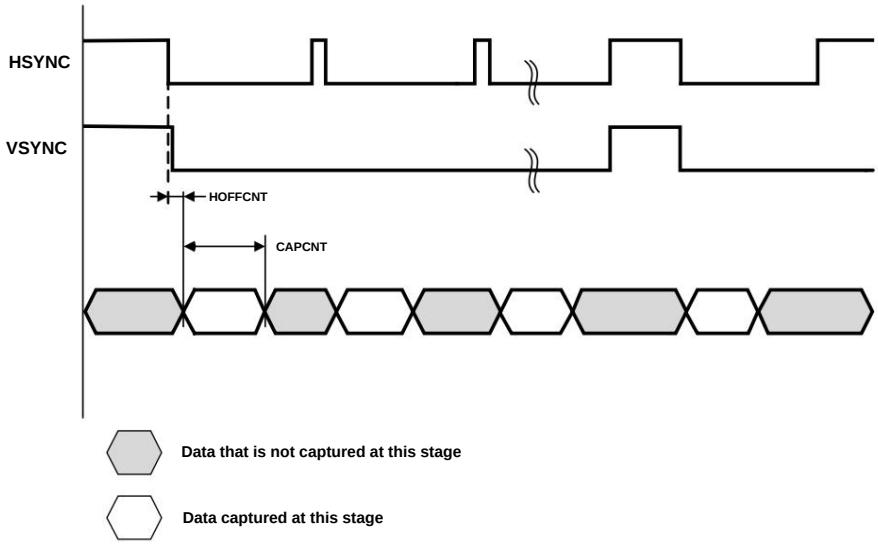


Figure 29-6 Cropping window data capture waveform



29.4 Register Description

Table 29-2 List of DVP-related registers

name	Access address	describe	Reset value
R8_DVP_CR0	0x40025800	DVP Control Register 0	0x00
R8_DVP_CR1	0x40025801	DVP Control Register 1	0x06
R8_DVP_IER	0x40025802	DVP Interrupt Enable Register	0x00
R16_DVP_ROW_NUM	0x40025804	Image Row Count Configuration	0x0000
R16_DVP_COL_NUM	Register 0x40025806	Image Column Count	0x0000
R32_DVP_DMA_BUF0	Configuration Register 0x40025808	DVP DMA	0x00000000
R32_DVP_DMA_BUF1	Address 0 Register 0x4002580C	DVP DMA	0x00000000
R8_DVP_IFR	Address 1 Register 0x40025810	DVP Interrupt	0x00
R8_DVP_STATUS	Flag Register 0x40025811	DVP Receive FIFO Status	0x00
R16_DVP_ROW_CNT	Register 0x40025814	DVP Row Counter	0x0000
R16_DVP_HOFFCNT	Register 0x40025818	Horizontal Displacement Register at	0x0000
R16_DVP_VST	Window Start 0x4002581A	Row Count Register at	0x0000
R16_DVP_CAPCNT	Window Start 0x4002581C	Capture Count	0x0000
R16_DVP_VLINE	Register 0x4002581E	Vertical Row Count	0x0000
R32_DVP_DR	Register 0x40025820	Data register	0x00000000

29.4.1 DVP Configuration Register (R8_DVP_CR0) Offset

Address: 0x00

Bit	name	access	describe	Reset value
6	RB_DVP_JPEG	RW	JPEG mode enabled: 1: JPEG compression format; 0: Original data format.	0
[5:4]	RB_DVP_MSK_DAT_MOD[1:0]	RW	DVP data bit width configuration: 00: 8-bit mode; 01: 10-bit mode;	00b

			1x: 12-bit mode.	
3	RB_DVP_P_POLAR	RW	HCLK polarity configuration: 1: Data is sampled at the falling edge of HCLK; 0: Data is sampled at the rising edge of HCLK.	0
2	RB_DVP_H_POLAR	RW	HSYNC polarity configuration: 1: HSYNC low level indicates data is valid; 0: HSYNC High level data is valid.	0
1	RB_DVP_V_POLAR	RW	VSYNC polarity configuration: 1: VSYNC high level indicates valid data; 0: VSYNC Low level indicates data is valid.	0
0	RB_DVP_ENABLE	RW	DVP function enabled: 1: Enable DVP; 0: Disable DVP.	0

29.4.2 DVP Configuration Register (R8_DVP_CR1) Offset Address: Bit

0x01

	name	access	describe	Reset value
[7:6]	RB_DVP_FCRC[1:0]	RW	DVP frame capture rate control: 00: Capture all frames; 01: Capture once every other frame; 10: Capture once every three frames. 11: Retained.	00b
5	RB_DVP_CROP	RW	Crop function control: 1: Only capture data in the window specified by the clipping register; 0: Capture the complete image.	0
4	RB_DVP_CM	RW	Capture Mode: 1: Snapshot mode; 0: Continuous mode.	0
3	RB_DVP_BUF_TOG	RWT	Buffer address flag. Hardware controls the toggle, software sets it to 1. Transform this bit; writing 0 is invalid. 1: Data is stored in receiving address 1; 0: Data is stored in receive address 0.	0
2	RB_DVP_RCV_CLR	RW	Receive logic reset control: 1: Reset the receiver logic circuit; 0: Cancel the reset operation.	1
1	RB_DVP_ALL_CLR	RW	Flags and FIFO clearing control, written to 1 or 0 by software: 1: Reset flag and FIFO; 0: Cancel the reset operation.	1
0	RB_DVP_DMA_ENABLE	RW	DMA enable control bit: 1: Enable DMA; 0: DMA is disabled.	0

29.4.3 DVP Interrupt Enable Register (R8_DVP_IER) Offset

Address: Bit 0x02

	name	access	describe	Reset value
[7:5] Reserved		RO reserved. End-		0
4	RB_DVP_IE_STP_FRM	RW	of-frame interrupt enabled. (When VSYNC changes from active level to active level) An interrupt is generated when the level is invalid. 1: Enable frame end interrupt; 0: Disable frame end interrupt.	0
3	RB_DVP_IE_FIFO_OV	RW	Receive FIFO overflow interrupt enable: 1: Enable FIFO overflow interrupt; 0: Disable FIFO overflow interrupt.	0
2	RB_DVP_IE_FRM_DONE	RW	Frame reception complete interrupt enabled. (Counter reached) An interrupt occurs when configuring the ROW/COL_NUM value, indicating the last... (Data has been written to RAM) 1: Enable frame reception completion interrupt; 0: Disable frame reception completion interruption.	0
1	RB_DVP_IE_ROW_DONE	RW	End-of-line interrupt enabled. (Counter reaches COL_NUM configuration) (Interruption generated when value is displayed) 1: Enable line termination interruption; 0: Prohibits line termination.	0
0	RB_DVP_IE_STR_FRM	RW	A new frame begins with interrupt enable. (VSYNC starts from an invalid level) An interrupt is generated when the signal level changes to an active level, indicating the start of a new frame. (Start, data coming soon) 1: Enable interrupt at the start of a new frame; 0: Disable interruption at the start of a new frame.	0

29.4.4 DVP Image Valid Row Count Configuration Register (R16_DVP_ROW_NUM)

Offset Address:

Bit	0x04 Name	access	describe	Reset value
[15:0] RB_DVP_ROW_NUM[15:0] RW			In RGB mode, it represents the number of rows contained within a frame of image data. This register has no practical meaning in JPEG mode.	0

29.4.5 DVP Image Valid Column Configuration Register (16_DVP_COL_NUM)

Offset Address: 0x06

Bit	name	access	describe	Reset value
[15:0] RB_DVP_COL_NUM[15:0] RW			In RGB mode, this represents the HCLK cycles contained within a single row of data. Period. In JPEG mode, used to configure the DMA receive length. Spend.	0

29.4.6 DVP DMA Receive Address 0 Register (R32_DVP_DMA_BUF0) Offset

Address: 0x08 bits

	Name access		describe	Reset value
[31:0] RB_DVP_DMA_BUF0 [31:0] RW DMA receive address 0 (32-byte aligned).				0

29.4.7 DVP DMA Receive Address 1 Register (R32_DVP_DMA_BUF1) Offset

Address: 0x0C bit

	Name access		describe	Reset value
[31:0] RB_DVP_DMA_BUF1 [31:0] RW DMA receive address 1 (32-byte aligned).				0

29.4.8 DVP Interrupt Flag Register (R8_DVP_IFR) Offset

Address: 0x10 Bit

Name		access	describe	Reset value
[7:5] Reserved		RO is reserved.		0
4	RB_DVP_IF_STP_FRM	RW1Z frame end interrupt flag, active high, cleared by writing 1.		0
3	RB_DVP_IF_FIFO_OV	RW1Z Receive FIFO Overflow Interrupt Flag, active high, cleared by writing 1. RW1Z		
2	RB_DVP_IF_FRM_DONE	Frame Receive Complete Interrupt Flag, active high, cleared by writing 1.		0
1	RB_DVP_IF_ROW_DONE	RW1Z is the end-of-line interrupt flag, active high, cleared by writing 1.		0
0	RB_DVP_IF_STR_FRM	RW1Z Frame Start Interrupt Flag, active high, cleared by writing 1.		0

29.4.9 DVP Receive FIFO Status Register (R8_DVP_STATUS) Offset

Address: 0x11

Bit	name	access	describe	Reset value
7	Reserved	RO is reserved.		0
[6:4] RB_DVP_FIFO_CNT[2:0] RO FIFO counter.				0
3	Reserved	RO is reserved.		0
2	RB_DVP_FIFO_OV	RO	FIFO overflow status: 1: FIFO overflow; 0: FIFO has not overflowed.	0
1	RB_DVP_FIFO_FULL	RO	FIFO at full capacity: 1: Cache full; 0: FIFO not full.	0
0	RB_DVP_FIFO_RDY	RO	FIFO ready status: 1: There is data in the FIFO; 0: No data in FIFO.	0

29.4.10 DVP Received Image Line Count Register (R16_DVP_ROW_CNT)

Offset Address:

0x14 Bit Name		access	This	Reset value
[15:0] RB_DVP_ROW_CNT[15:0] RO			describes the number of rows contained in a single frame of image data during actual reception. The register is updated at the end of the frame. In JPEG format, the value of this register is meaningless.	0

29.4.11 Horizontal displacement register at the start of the DVP window (R16_DVP_HOFFCNT)

Offset address: 0x18

Bit	name	access	describe	Reset value
[15:0] RB_DVP_HOFFCNT[15:0] RW			Within the window row, each row needs to be left blank with an HCLK cycle before capturing data. Period number. The lower 14 bits are valid.	0

29.4.12 DVP Window Start Row Number Register (R16_DVP_VST)

Offset address: 0x1A Bit

name		access	Describes	Reset value
[15:0] RB_DVP_VST[15:0]		RW	the row number from which the image begins to be captured; data before this row is not captured. Received. The lower 13 bits are valid.	0

29.4.13 DVP Capture Count Register (R16_DVP_CAPCNT)

Offset Address:

0x1C Bit Name		access		Reset value
[15:0] RB_DVP_CAPCNT[15:0]		RW	Describes the number of HCLK cycles to be captured within the clipping window. (Lower 14 bits) efficient.	0

29.4.14 DVP Vertical Row Count Register (R16_DVP_VLINE)

Offset address: 0x1E Bit

name		Access description reset value	
[15:0] RB_DVP_VLINE[15:0]		RW: The number of rows to capture within the cropping window. The lower 14 bits are valid. 0	

29.4.15 DVP Data Register (R32_DVP_DR) Offset

Address: 0x20

Bit	name	access	describe	Reset value
[31:0] RB_DVP_DR[31:0]		RO	The DVP interface triggers once every time it receives 4 bytes of data. DMA request, a 4-byte deep FIFO can reserve space for DMA transfers. There is ample time. This effectively prevents DMA overflow.	0

Chapter 30 Random Number Generator (RNG)

The chip incorporates a hardware random number generator based on continuous analog noise, which provides a 32-bit random number generator through internal analog circuitry.
A random number of bits.

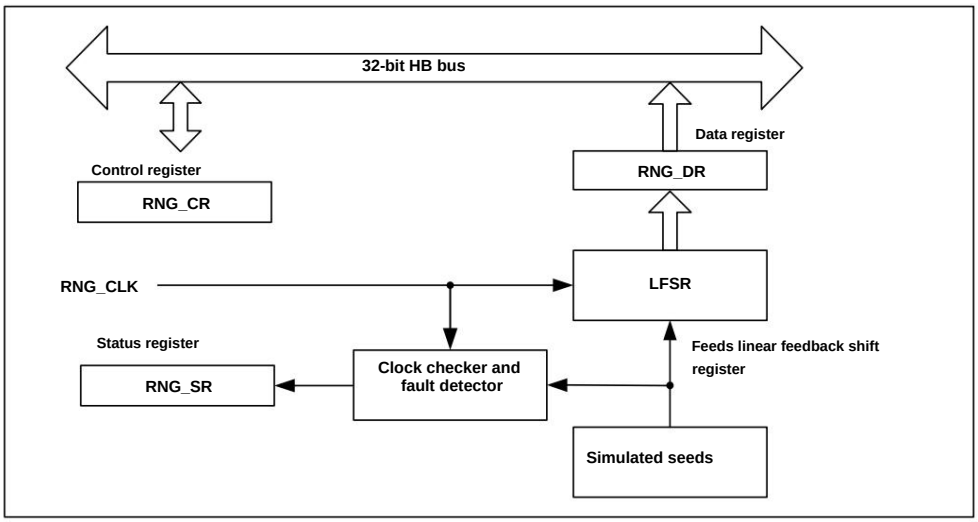
30.1 Key Features

It can generate 32-bit random numbers .

Error management is possible .

ÿ Can be disabled individually to reduce power consumption

Figure 30-1 RNG Module Block Diagram



30.2 Functional Description

The random transmitter is implemented using analog circuitry, which generates a seed for the linear feedback shift register (RNG_LFSR) used to generate...
32-bit random numbers. RNG_LFSR is clocked at a constant frequency by a dedicated clock (PLL48CLK), therefore the quality of the random numbers is similar to that of HCLK.
Clock-related. When a large number of seeds are introduced into RNG_LFSR, the contents of RNG_LFSR will be transferred to the data register (R32_RNG_DR).

30.2.1 RNG Operations : The specific

operation steps are as follows:

1) To enable interrupts, the IE bit in the R32_RNG_CR register must be set to 1 (this is generated when random numbers are ready or an error occurs).
Interruption).

2) Enable random number generation by configuring the RNGEN bit of the R32_RNG_CR register, while also activating the analog section, RNG_LFSR, and error handling.
Detector.

3) If interrupts are enabled, each time an interrupt occurs, the SEIS and CEIS bits in the R32_RNG_SR register are checked to confirm that no interrupt has occurred.
An error has occurred and DRDY is 1, confirming that random numbers are ready. The contents of the R32_RNG_DR register can then be read.

30.2.2 Error Management RNG errors

include clock errors and seed errors. When a clock error occurs, the RNG can no longer generate random numbers; in this case, the clock signal needs to be checked.
Check if the clock controller is configured correctly and can provide the RNG clock, then clear the CEIS bit. The RNG should function normally when the CEIS bit is 0.
Working. When a clock error occurs, it has no effect on the previous random number and can be used normally. When a seed error occurs, R32_RNG_DR...
The value in the register cannot be used with this random number. If you want to reuse RNG, you need to first clear the SEIS bit, then clear the RNGEN bit and...

Set to 1, reinitialize and restart RNG.

30.3 Register Description

Table 30-1 List of RNG-related registers

name	Access	describe	Reset value
R32_RNG_CR	address 0x40023C00	RNG control	0x00000000
R32_RNG_SR	register 0x40023C04	RNG status register	0x00000000
R32_RNG_DR	0x40023C08	RNG data register	0x00000000

30.3.1 RNG Control Register (R32_RNG_CR)

Offset address: 0x00

31	30	29	28	27	26	25	24	23	twenty two	twenty one	20	19	18	17	16
Reserved															

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

Bit	Name access		describe	Reset value
[31:4]	Reserved	RO	is reserved.	0
3	IE	RW	Interrupt enable control: 1: Enable RNG interrupt; 0: Disable RNG interruption.	0
2	RNGEN	RW	Random number generator enabled: 1: Enable the random number generator; 0: Disable random number generator.	0
[1:0]	Reserved	RW	reserved.	0

30.3.2 RNG Status Register (R32_RNG_SR)

Offset address: 0x04

31	30	29	28	27	26	25	24	23	twenty two	twenty one	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									SEIS	CEIS	Reserved	SECS	CECS	DRDY	

Bit	Name access		describe	Reset value
[31:7]	Reserved	RO	Reserved.	0
6	SEIS	RW	Seed error interruption status (this bit is set simultaneously with SECS): 1: One of the following error sequences was detected: - More than 64 identical consecutive bits; - More than 32 consecutive alternating 0s and 1s. 0: No inverted error sequence detected.	0
5	CEIS	RW	Clock error interrupt status (this bit is set simultaneously with CECS): 1: PLL48CLK clock not detected correctly;	0

			0: PLL48CLK clock was detected correctly.	
[4:3] Reserved		RO reserved.		0
2	SECS	RO	Seed error current status: 1: One of the following error sequences was detected: - More than 64 identical consecutive bits; - More than 32 consecutive alternating 0s and 1s. 0: No erroneous sequence detected.	0
1	CECS	RO	Clock error current status: 1: PLL48CLK clock not detected correctly; 0: PLL48CLK clock was detected correctly.	0
0	DRDY	RO	Data ready (this bit is cleared to 0 after reading the R32_RNG_DR register): 1: The R32_RNG_DR register is valid, this random number is available; 0: The R32_RNG_DR register is invalid; this random number is unavailable.	0

30.3.3 RNG Data Register (R32_RNG_DR) Offset Address:

0x08

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

Bit	Name access		describe	Reset value
[31:0] RNDATA		RO is a 32-bit random number.		0

Chapter 31 Ethernet Transceivers (ETH)

The module descriptions in this chapter apply only to the CH32H417 microcontroller product.

The term "Ethernet transceiver" mentioned in this chapter is a technical term, referring to the Ethernet data link inside a microcontroller.

A path layer transceiver, with a maximum communication rate of gigabits per second (1Gbps), is a communication peripheral. The "MAC" mentioned in this chapter refers to...

The role of an Ethernet transceiver at the data link layer is that it is a component of an Ethernet transceiver.

An Ethernet transceiver (MAC) is an important high-speed communication component for microcontrollers, enabling microcontrollers to...

The controller connects to Ethernet at gigabit speeds, enabling extremely fast data communication.

31.1 Main Features

The Ethernet transceiver of a microcontroller is an important high-speed communication peripheral, integrating a gigabit MAC (Media Access Control) port.

The device includes a 256-bit wide DMA controller, a management counter (MMC), a precision time protocol controller (PTP), and a 10 Mbps/100 Mbps speed controller.

It uses a 10M/100M Ethernet physical layer (PHY). With an external Gigabit Ethernet physical layer (PHY), it can achieve gigabit speeds (1Gbps).

It connects to Ethernet at high speed for data transmission and reception. Ethernet transceivers operate at the data link layer and require software implementation of the TCP/IP protocol.

Stack and interface. The Ethernet transceiver supports a standard RGMII interface for PHY connection; however, to achieve gigabit access speeds, users must use...

RGMII interface; the Ethernet transceiver controls the PHY via the SMI interface, and the timing of the interface is automatically implemented by the MAC, without requiring user intervention in software.

The RGMII interface supports transmit clock phase flipping and relative data delay, with a maximum delay of 4 nanoseconds.

The Ethernet transceiver's MAC supports Ethernet using the standard IEEE 802.3 protocol, and supports magic frames and specific wake-up frames. Ethernet transceiver.

The DMA controller paired with the device manages data transmission and reception and memory movement in the form of descriptors. The number of descriptors is determined by the user based on communication...

The density is determined automatically; the DMA controller can write received data to or from the memory space specified by the descriptor at a speed of 256 bits.

Retrieve the data to be sent. Furthermore, the Ethernet transceiver's MAC also supports the IEEE 1588 Precision Time Protocol.

31.1.1 MAC Features ÿ Supports

RGMII interface

RGMII supports transmit clock delay and toggle.

Supports full-duplex and half-duplex modes, and data transfer rates of 10Mbps/100Mbps/1000Mbps.

• Supports automatic insertion of frame header sequences

and SFD . • Supports automatic insertion and CRC

check. • Supports automatic calculation and verification of IP/ICMP/TCP/UDP protocol packets.

Supports frame length control .

Supports transmission interval adjustment

Supports VLAN frames

Supports perfect address filtering and hash filtering for frame receive addresses .

Supports frame transmission address filtering

Supports multicast frame reception control

Supports promiscuous mode

Supports SMI management interface

Supports magic frames and custom wake-up frames to wake up the microcontroller.

Supports dedicated Ethernet wake-up interrupt entry.

Supports data loopback at the link layer .

Built -in pre-assigned globally unique MAC address

Application options include an Ethernet controller MAC with an external 1Gbps PHY, or an Ethernet controller MAC with a built-in 10Mbps/100Mbps PHY.

31.1.2 DMA Characteristics

ÿ 256-bit MAC-dedicated DMA ÿ Minimizes

CPU operations

• Supports byte-aligned RAM access. • Manages

transmit and receive buffers using descriptors.

ÿ Part of the status feedback for receiving and transmitting is in the descriptor

ÿ It can dynamically modify descriptors and buffers that are not currently in use.

Supports manual stop or start .

Supports chained or ring-shaped connection descriptors.

Supports multiple interrupt sources, including interrupts upon completion of send/receive operations .

31.1.3 MMC Module Features ÿ

Supports manual reset, stop, or freeze

Multiple counters supporting various counting modes for both sending and receiving.

Supports multiple interrupt methods

31.1.4 PTP Module Features ÿ

Supports IEEE 1588 protocol

Supports automatically saving the current time during transmission and reception.

It comes with a 32-bit second timer and a 32-bit signed subsecond timer.

Supports both fine-tuning and coarse-tuning of the time.

Supports one interrupt source

Supports PPS output

31.1.5 Internal 10M/100M Physical Layer Features : ÿ Supports

10BASE-T and 100BASE-TX and auto-negotiation.

Supports Auto-MDIX switching of TX/RX, automatically identifying positive and negative signal lines.

Supports Wake-on-LAN (WOL)

Supports interrupt functionality

Supports half-duplex and full-duplex .

Five network status LEDs

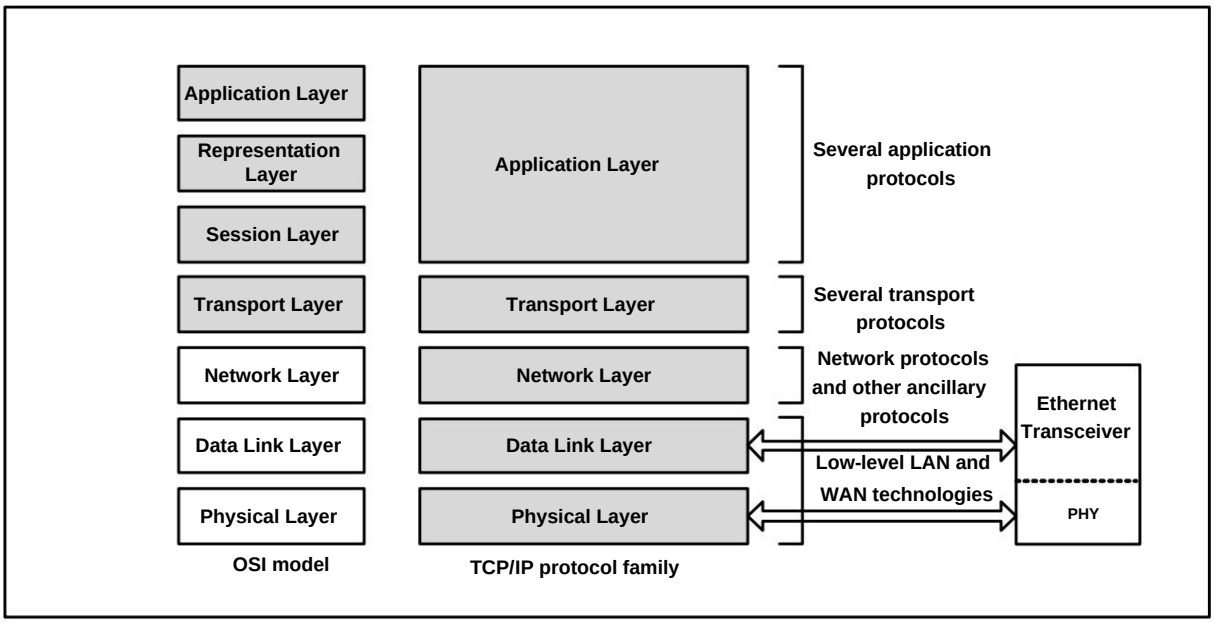
31.2 Overview

Ethernet transceivers operate at the data link layer and physical layer of the OSI seven-layer model. This is necessary for establishing connections within the widely used Ethernet network.

For communication using protocols such as IP, TCP, and UDP, users also need to implement the TCP/IP protocol stack in software. Ethernet transceivers are controlled by media access control.

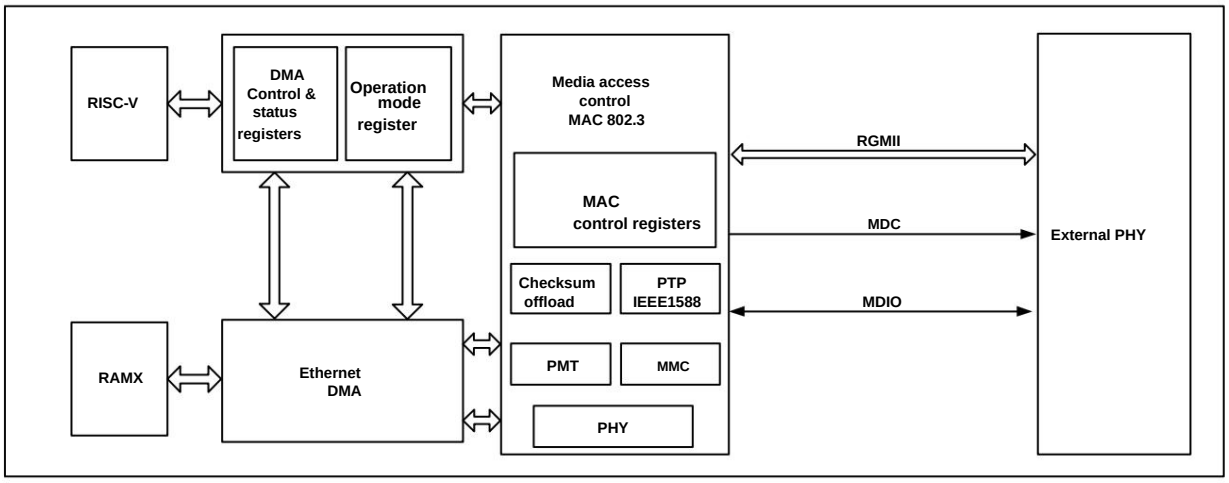
It consists of a MAC layer, a paired DMA layer, and their control registers and internal physical layer.

Figure 31-1 Position of Ethernet transceivers in the OSI and TCP/IP models



The Ethernet transceiver's MAC is designed according to the IEEE 802.3 protocol specification, and is equipped with a 256-bit wide DMA to ensure fast data forwarding from the network cable to the microcontroller's memory. The Ethernet transceiver has a robust and complete set of DMA control registers, MAC control registers, and mode control registers. The microcontroller's CPU operates the Ethernet transceiver's registers via the HB bus. The Ethernet transceiver connects to the Gigabit Ethernet physical layer via the RGMII interface. The pins of the Ethernet transceiver's RGMII interface are multiplexed; see Section 31.3 for details. The Ethernet transceiver manages the Ethernet physical layer through the SMI interface. When using the Ethernet transceiver, the HB bus clock must not be lower than 50MHz.

Figure 31-2 Block diagram of an Ethernet transceiver



In addition, the Ethernet transceiver also supports the IEEE 1588 Precision Time Protocol (PTP), providing precise time data for microcontroller systems or external devices.

31.3 Ethernet Transceiver Pin Distribution and Configuration

The microcontroller supports the RGMII interface. The RGMII interface can be used for 10 Mbps, 100 Mbps, and Gigabit Ethernet. The table below shows the standard RGMII interface. The distribution of the interface and internal physical layer on the package pins.

Table 31-1 Media Independent Interfaces and Media Dependent Interfaces of the Built-in Physical Layer of the Microcontroller

Other related pin layouts and required configurations

pin	RGMII	RGMII pin configuration
PD15	GTXC	Push-pull multiplexed output
PA14	RXDV	Floating input
PA15	GRXC	Floating input
PC6	RXD3	Floating input
PD13	TXD0	Push-pull multiplexed output
PD12	TXD1	Push-pull multiplexed output
PD11	TXD2	Push-pull multiplexed output
PD10	TXD3	Push-pull multiplexed output
PC8	RXD1	Floating input
PC0	MDC	Push-pull multiplexed output
PC1	MDIO	Push-pull multiplexed output
PC9	RXD0	Floating input
PC7	RXD2	Floating input
PD14	TXEN	Push-pull multiplexed output
PC2	PPS	Push-pull multiplexed output
-	MDITP	No IO configuration required
-	MDITN	No IO configuration required
-	MDIRP	No IO configuration required
-	MDIRN	No IO configuration required

31.4 Physical Layer (PHY) Management and Data Interaction

The Ethernet transceiver's MAC address manages the PHY via the Site Management Interface (SMI interface). The microcontroller operates in RGMII mode. Use different pins to bring out the SMI interface.

31.4.1 SMI Interface The

SMI interface is a serial communication interface that uses two lines, MDC (clock line) and MDIO (data line), to access the PHY registers. This system manages the PHY chips, supporting up to 32 PHY lines. MDC is the clock line, which remains low when idle, and MDIO is the data line. The SMI read/write operations and frame composition are all controlled by the MAC; the user only needs to write the address and data. The relevant registers are... For the MII address register (R32_ETH_MACMIIAR) and the MII data register (R32_ETH_MACMIIDR).

31.4.1.1 Frame Format The

format of the SMI interface management frame is shown in Table 31-2 below.

Table 31-2 SMI Frame Format

	Preamble STR		OP PHY ADR REG ADR			T	DATA	P
The definitions of each field		01	10	PPPPP	RRRRR	Z0	DDDDDDDDDDDDDDDD	Z
in the management frame		01	01	PPPPP	RRRRR	10	DDDDDDDDDDDDDDDD	Z

for reading 32 ones and writing 32 ones are as follows:

1) Preamble: A leading character consisting of 32 ones, used for MAC and PHY synchronization;

2) STR: Start character, always "01";

3) OP: Operator, read as "10", written as "01";

4) PHY ADR: Physical layer address, 5 bits;

5) REG ADR: Register address, 5 bits;

6) T: Switching character, two bits, used to toggle control of the MDIO line between the MAC and PHY. During read operations, the MAC maintains control of the MDIO line.

High impedance: The PHY maintains high impedance for the first bit and pulls down for the second bit, gaining control of MDIO; during write operations, the MAC switches to MDIO.

The line is first set high and then pulled down. The PHY maintains a high impedance state to MDIO, and the MAC maintains control over MDIO.

7) DATA: Data field, 16 bits, data read and written by the MAC to the PHY, MSB first;

8) P: Both MAC and PHY maintain a high impedance state to MDIO, but the pull-up resistor of PHY will pull MDIO high.

31.4.1.2 The read/write timing

operation for writing to the PHY register is as follows:

When the user sets the MII write bit (R32_ETH_MACMIIAR: MW) and busy bit (R32_ETH_MACMIIAR: MB), the SMI interface...

It will send the PHY address and register address to the PHY, and then send data (R32_ETH_MACMIIDR). Data is sent via the SMI interface.

During this process, the values of the MII address register and the MII data register cannot be modified; during this process, the busy bit remains high, and the MII address...

Write operations to registers or MII data registers will be ignored and will not affect the overall transfer. When the write operation is complete, the SMI connection...

The busy flag will be cleared, allowing users to determine the end of the write operation based on the busy flag. See the write section in Figure 31-3.

The operation of reading the PHY register is as follows:

When the user sets the MII busy bit in the MII address register (R32_ETH_MACMIIAR) of the Ethernet MAC, while keeping the MII write bit...

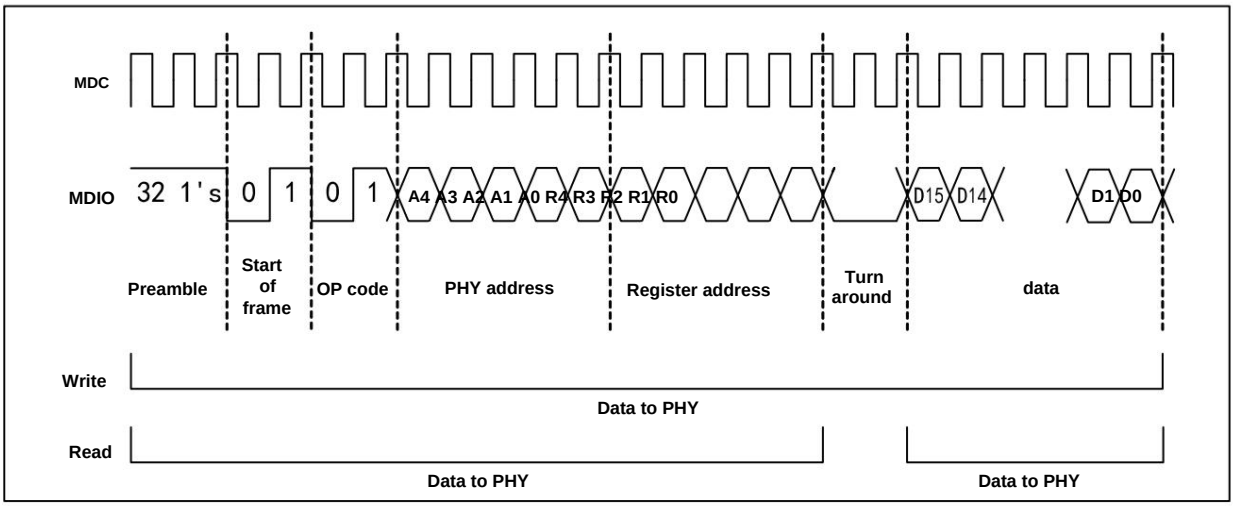
Upon reset, the SMI interface sends the PHY address and register address to perform a read operation on the PHY register. Throughout the transmission process, [the following is used:]

Users cannot modify the contents of the MII address register and MII data register. During transmission, the busy bit remains high, controlling the MII address register.

Write operations to the register or MII data register will be ignored and will not affect the correct completion of the entire transfer. After the read operation is complete, SMI...

The interface will clear the busy bit and write the data read from the PHY into the MII data register, as shown in the read section of Figure 31-3.

Figure 31-3 SMI Interface Read/Write Speed Chart



31.4.1.3 SMI Clock Generally, the

SMI clock needs to be maintained within a fixed range to ensure that the SMI clock is actually received by the PHY. Specifically...

Refer to the manual of the PHY chip selected by the user.

The SMI clock frequency is obtained by dividing the HB clock through the CR field of the MII address register (R32_ETH_MACMIIAR). The table below shows...

The frequency division relationship between the SMI clock and the HB clock. The default selection is 42 division.

Table 31-3 Frequency division relationship between SMI clock and HB clock

	Suitable range of HB clocks	SMI Clock
0000b	60MHz or above	HB Clock / 42

0010b	20-35MHz	HB Clock / 16
0011b	35-60MHz	HB Clock / 26
Other values	Meaningless	

31.4.2 RGMII Interface 31.4.2.1

Pinout The pinouts and

functions of RGMII are shown in the table below.

Table 31-4 RGMII Pinout and Functions

pin	Function
GTXC	Send clock
TXCTL	Sending data control
TXD0	Send data cable [0:3]
TXD1	
TXD2	
TXD3	
GRXC	Receive clock
RXCTL	Receive data control
RXD0	Receive data lines [0:3]
RXD1	
RXD2	
RXD3	

Note: (1) TXCTL/RXCTL are used as valid signals for transmitting and receiving data in the Ethernet transceiver of this microcontroller; (2) RGMII has its own... SMI interface.

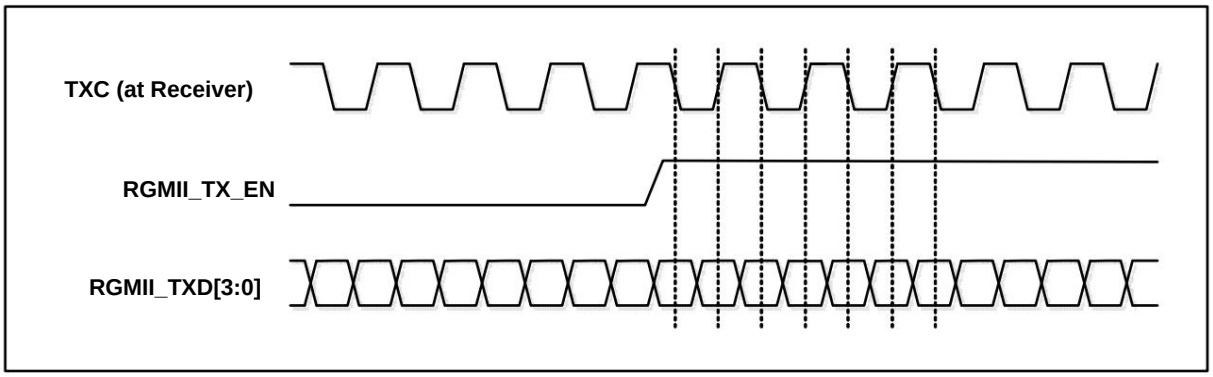
31.4.2.2 Timing: When

RGMII operates at 10 Mbps and 100 Mbps speeds, its timing is similar to that of MII; when operating at 1 Gbps speeds, the clock speed of RGMII is 125 MHz. It uses a double-edge sampling method. RGMII does not support half-duplex.

Because RGMII uses double-edge sampling and operates at a clock frequency of 125MHz, circuit designers should pay attention to signal integrity. Sexual issues.

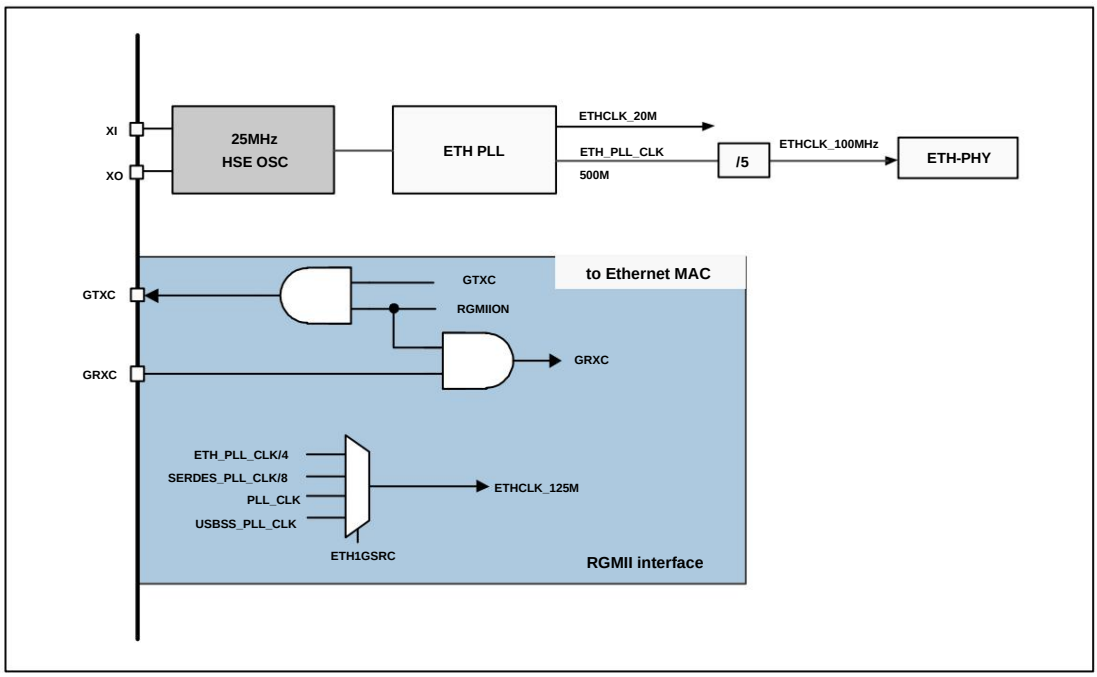
The receiver of an RGMII transceiver should receive a clock that lags behind the data by 90° to ensure correct sampling. Clock delay output and phase flipping functions.

Figure 31-4 Timing diagram of RGMII



31.4.3 Peripheral Clock Source Configuration

Figure 31-5 Ethernet peripheral clock tree



When using RGMII, RCC_CFGR0[21] needs to be set to 1 to enable RGMII mode, and RCC_CFG2[31:30] needs to be set to select.

A suitable 125MHz clock source is provided to the MAC. The RGMII transmit clock GTXC is generated by the MAC, and the receive clock GRXC is generated by the PHY.

See the EVT examples on the official website for details.

31.5 IEEE 802.3 and IEEE 1588

The IEEE 802.3 protocol and its supplementary protocols constitute the current official standard for Ethernet, which defines in detail the methods currently used in Ethernet.

In terms of aspects, we can consider Ethernet as part of the physical layer and data link layer in the OSI model. This section focuses on its application.

This discussion covers the user-required aspects of the IEEE 802.3 protocol, specifically frame format and MAC address related details. Additionally, it addresses the microcontroller's...

Ethernet transceivers also support the IEEE 1588 Precision Time Protocol, and this article will also discuss some aspects of IEEE 1588.

In the Ethernet model built upon the IEEE 802.3 protocol, the data transmission unit is the Ethernet frame. Ethernet frames transmit data across different media using different...

Different encoding methods are used for transmission at different rates. After reception, the data is decoded by the physical layer and sent to the MAC interface via the MII interface. The MAC checksum is then processed.

After filtering, the application information is extracted by the TCP/IP protocol stack and sent to different application processes.

31.5.1 Frame Format The

frame format of Ethernet frames is shown in Table 31-7.

Table 31-7 Common Ethernet Frame Formats

Preamble (SFD), Destination Address, Source Address, Length or Type, Data Field, CRC						
7 Bytes	1 Byte	6 Bytes	6 Bytes	2 Bytes	46-1500 Bytes	4 Bytes
Total length: 64 to 1518 bytes plus 8 bytes of physical layer header (preamble and SFD)						

Preamble: A 56-bit (7-byte) alternating low and high level jump, with a fixed value, represented in hexadecimal as 0xAA-0xAA-0xAA-0xAA-0xAA-0xAA.

This field is used for clock synchronization. This field is automatically added/removed by hardware and does not require user intervention.

SFD: Start of Frame Delimiter, 8 bits (1 byte), value 10101011b. The SFD is used to inform the receiver that this is the last time the frame will be executed.

The clock synchronization opportunity is followed by the target address. This byte is automatically added/removed by the hardware and does not need to be controlled by the user.

Destination address: The address of the device that the device sending this frame expects to receive it; this refers to the hardware address, also known as the MAC address.

Assigned globally by IEEE, each MAC address is 6 bytes (48 bits). The MAC address of our (WCH) microcontrollers is programmed at the factory.

Inside the chip, the address is sent following the principle of least significant bit first.

Source address: The hardware address of the device that sent this frame. The source address must be a unicast address.

Length or type: 2 bytes. Ethernet generally uses it as the type, and it is also used as the length in the IEEE 802.3 standard, with a value of 1536 (i.e., 0x0600).

The boundary, 1536 and above, indicates the protocol, specifying which upper-layer protocol the data is organized according to. For example, 0x0806 indicates ARP.

0x0800 represents IPv4, 0x86dd represents IPv6; numbers below 1536 represent data length.

Data field: Minimum 46 bytes, maximum 1500 bytes. If the data field is less than 46 bytes, padding is added to increase it to 46 bytes; if it exceeds 1500 bytes...

Please assemble the 1500-byte data into a separate frame. The data field contains the data that needs to be actually sent.

CRC: Cyclic Redundancy Check, here we are using CRC32 check.

31.5.2 Frame Transmission

When an Ethernet transceiver transmits data frames, its dedicated DMA controller retrieves data from the transmit descriptor (as described in 31.5.1).

The data to be sent is retrieved from the specified RAM and pushed into the MAC via a dedicated data bus. The FIFO's fill level is then returned to DMA control.

In the DMA controller, once all the data to be sent has been sent, the DMA controller will send a data start signal to the MAC, and the MAC will then start sending.

Data is read from the FIFO, RGMII sends a preamble and SFD, and then sends the actual data; the DMA controller sends a data end signal to the MAC.

Then, the MAC is added with an automatically calculated CRC.

The MAC will automatically send frames according to the configured device type, calculate a checksum for the checksum field, and replace the original checksum field.

The value. This feature can be turned off.

The MAC will automatically add a CRC32 checksum to the padding, but it can also be set to not add a CRC checksum. This is in accordance with the IEEE 802.3 protocol.

According to the regulations, the data field length must be no less than 46 bytes. When the MAC detects that the data field to be sent is shorter than 46 bytes, it will automatically adjust the digital...

Padding is added after the field. If automatic padding is selected, a CRC32 checksum will always be added, regardless of register configuration. The MAC uses CRC32.

The test formula is as follows:

$$G(") = " \quad ^{\#\$} + " \ ^{\$\%} + " \ ^{\$\#} + " \ ^{\$\$} + " \ ^{\&\%} + " \ ^{\&\$} + " \ ^{\&\&} + " \ ^{\&'} + " \ ^{(} + " \ ^{)} \ ^{*} + " \quad + " \ ^{+} + " \ ^{\$} + " + 1$$

When IEEE 1588 (PTP) functionality is enabled, an enhanced descriptor is required for storing timestamps.

31.5.3 Frame Reception When

receiving Ethernet frames, the Ethernet transceiver moves the Ethernet frame from RGMII to the MAC, where it first enters the FIFO.

Then it is forwarded to a buffer in RAM by DMA. The MAC filters and inspects the Ethernet frames, including perfect address filtering and...

HASH filtering and checks typically involve performing frame length checks, IP/ICMP/TCP/UDP checks, and CRC checks on frames. Frames that fail filtering or checks will be flagged.

Frames that exceed the specified length will be marked by the descriptor or discarded, and packets that exceed the specified length may be truncated.

When IEEE 1588 (PTP) functionality is enabled, an enhanced descriptor is required for storing timestamps.

31.5.4 Frame Filtering : Using

MAC frame filters, perfect or hash filtering can be performed on the destination MAC address and source MAC address of received frames.

This feature allows you to configure frames that don't pass the filter to be dropped, received, or marked in the receive descriptor. Please refer to the MAC documentation for details on this feature.

Description of the frame filtering register (R32_ETH_MACFFR). The MAC has four built-in MAC address registers, of which MAC address register 0...

The default MAC address register is used to store its own MAC address; the other three MAC address registers can be used for perfect filtering, corresponding to the source address of the received frame.

Compare the address or destination address. After setting R32_ETH_MACFFR: RA, all received frames will be received, but the corresponding status will be different.

It will be marked in the state field of the first word of the subsequent descriptor. Setting R32_ETH_MACFFR:PM will have a similar effect.

However, the state will not be marked. Setting R32_ETH_MACFFR: DAIF/SAIF will invert the result, changing the original result after filtering.

Items that pass the filter will be discarded or marked, while those that do not pass the filter will be transferred to the buffer in RAM.

31.5.4.1 Unicast Filtering

MAC uses the HPF bit and HU bit to determine whether to perform HASH filtering or perfect address filtering on unicast frames.

31.5.4.2 Multicast Filtering

MAC uses the HPF bit and HM bit to determine whether multicast frames undergo hash filtering or perfect address filtering. When the PAM bit is set, all...

All multicast packets can pass through the filter.

31.5.4.3 Broadcast

filtering: By setting the BFD bit, MAC can block all broadcast packets.

31.5.4.4 Source Address Filtering

Selection: Setting the AE bit in the MAC address register enables the MAC address register, while setting the SA bit determines whether to use that MAC address.

The MAC address register is used to compare whether it is a source address sample or a destination address sample.

31.5.4.5 Summary

The filtering settings for the target address and source address are shown in Table 31-8 and Table 31-9, respectively.

Table 31-8 shows the degree to which the settings of each bit in R32_ETH_MACFFR affect the acceptance of the destination MAC address of the received frame.

Frame type PM	HPF	HU	DAIF	HM	PAM	Effect
Broadcast Frame 1	-	-	-	-	-	pass
0	-	-	-	-	-	Failed (BFD set)
unicast frames 1	-	-	-	-	-	All frames passed
0	-	0	0	-	-	Perfect filter matching
0	-	0	1	-	-	Perfect filter matching fails
0	0	1	0	-	-	HASH filtering matching is performed using...
0	0	1	1	-	-	HASH filter matching fails.
0	1	1	0	-	-	- Perfect filtering or hash filtering matching is performed via
0	1	1	1	-	-	Perfect filtering or hash filtering does not work during matching. Pass
Multicast Frames 1	-	-	-	-	-	pass
-	-	-	-	-	1	pass
0	-	-	0	0	0	Perfect filter matching
0	0	-	0	1	0	HASH filtering matching is performed using...
0	1	-	0	1	0	0. Perfect filtering or hash filtering during matching
0	-	-	1	0	0	Perfect filter matching fails
0	0	-	1	1	0	HASH filter matching fails.
0	1	-	1	1	0	Perfect filtering or hash filtering does not work during matching. Pass

Table 31-9 shows the degree to which the settings of each bit in R32_ETH_MACFFR affect the acceptance of the source MAC address of the received frame.

Frame type RA SAIF SAF effect				
unicast frames 1	-	-	All frames passed	
0	0	0	A perfect filter match passes, but frames that fail are marked without being discarded.	
0	1	0	A perfect filter match fails; the failed frame is marked but not discarded.	
0	0	1	Frames that pass the perfect filter match are allowed; frames that fail are discarded.	
0	1	1	Frames that fail to match the perfect filter are discarded.	

Note: "-" indicates that the setting of this bit is not important.

31.5.5 MMC

The Management Counter (MMC) primarily counts various indicator frame reception and transmission states and MAC operation states. It can...

This generates a setup and an interrupt. Normally, we can obtain the currently received frame count using the "good" frame receive counter (MMCRGUFCR).

The number of intact frames is obtained by sending a "good" frame counter (MMCTGFCR) to determine the number of successfully transmitted frames.

Use the CRC checksum error counter (MMCRFCECR) to check for any received frames with CRC errors. Generally, if the data is normal...

However, the CRC error indicates a problem with the RGMII line layout.

Users need to understand one concept: what kind of frame is a "good" frame? Normally, once a frame is initiated for transmission, its frame...

If a frame meets the length requirements (autofill is enabled) and automatic CRC calculation is enabled, then it is considered a "good" frame. And a frame...

If received, as long as its CRC is correct, the frame length is within the Ethernet frame length range, and the frame length and length field values are consistent (if the length type field...), it is considered acceptable.

If the length is not specified, or if there is no misalignment, then it is considered a "good" received frame.

31.5.6 PMT 31.5.6.1

Overview The main

function of the PMT (Power Management Platform) is to wake the microcontroller from low-power mode via Ethernet. Gigabit Ethernet and above

The Ethernet controller supports two types of frames for system wake-up: the magic frame and the (remote) wake-up frame. When the Ethernet receives these two types of frames, due to the micro...

The controller is in low-power mode, so it may not necessarily generate a receive interrupt, even if it is valid, but if it passes through the MAC magic frame.

Upon recognition of the wake-up frame, a PMT interrupt is generated. The PMT interrupt is independent of the Ethernet interrupt and is detected by querying the PMT control and status.

The register can identify which Ethernet frame caused the interrupt.

31.5.6.2 Magic Package AMD's

magic package (also known as a phantom packet) is a commonly used Ethernet method for waking up microcontrollers.

A frame has a fixed and special frame format, meaning that it can be received by the target network card through frame filtering. It can be encapsulated into a broadcast frame, and the frame contains continuous...

A 6-byte full height (0xFF) followed by the target network interface's MAC address repeated 16 times. This combination can exist in any frame.

Because of its location, magic frames can be encapsulated as MAC frames, IP packets, or even UDP packets. The following is the format of a magic frame.

xx xx xx xx xx xx xx xx (No restrictions on previous data, or even no previous data is required) —ff ff ff ff ff ff 84 c2 e4 01 02 02 84 c2 e4 01 02 02 84 c2 e4 01 02 02 84 c2 e4 01 02 02 84 c2 e4 01

02 02 84 c2 e4 01 02 02 84 c2 e4 01 02 02 84 c2 e4 01 02 02 84 c2 e4 01 02 02 84 c2 e4

01 02 02 84 c2 e4 01 02 02 84 c2 e4 01 02 02 84 c2 e4 01 02 02 84 c2 e4 01 02 02 84 c2

e4 01 02 02 84 c2 e4 01 02 02 xx xx xx xx xx xx (Some network cards require a password to be appended to the end of the magic frame).

31.5.6.3 Wake-up Frame Due

to limitations in the magic frame format, the format of the (remote) wake-up frame can be defined by the user. This applies as long as it's a remote wake-up call.

Ethernet frames in the frame filter register (R32_ETH_MACRWUFFR) group are considered wake-up frames. Users can configure remote wake-up frames accordingly.

The remote wake-up frame filter register group is used to define the wake-up frame that meets your needs. It's important to note that the remote wake-up frame filter register group has only one address input.

The interface allows users to write data to it through eight consecutive write operations, and then read the data back through eight consecutive read operations.

Wake up the data in the frame filter register group.

The structure of the remote wake-up frame filtering register group is shown in the table below:

Table 31-10 Structure of the Remote Wake-up Frame Filtering Register Group

Filter Register 0 Byte Mask of Filter Register 0								
Filter Register 1 Byte Mask of Filter Register 1								
Filter Register 2 Byte Mask of Filter Register 2								
Filter Register 3 Byte Mask of Filter Register 3								
Filter register 4 Retain filter	Retain filter	Retain filter	Retain filter					

	3 Commands		2 Commands		1 command 0 command		
Filter register 5, Filter 3, Offset filter 2, Offset filter 1, Offset filter 0, Offset							
Filter register 6 Filter 1 CRC-16 Filter register 7 Filter 3 CRC-16				Filter 0 CRC-16 Filter 2 CRC-16			

As can be seen from the table above, the four fields of the filter register—byte mask, command, offset, and filter—work together to determine a frame.

Whether it is a remote wake-up frame or not, there are actually four different frames that can be set to meet the requirements.

The highest bit of the 32-bit byte mask must be 0. Bit[30:0] corresponds to the first 31 bytes of data defined by the offset field.

If a bit is set to 1, it indicates that the corresponding byte participates in the CRC-16 check, and a maximum of 31 bytes participate in the calculation;

The highest bit of the 4-bit command field indicates which frame it applies to: 1 means it is only valid for multicast addresses, and 0 means it is only valid for unicast addresses.

Bits 2 and 1 of the command field are reserved, and bit 0 is the enable bit. Setting it high indicates that this set of filters is enabled.

The offset field indicates how many bytes to offset from the frame header to start calculating the CRC16 value. The minimum value is 12. If the offset field is 12, then...

The CRC16 value is calculated starting from the 13th byte (the parameter model uses CRC-16-IBM, and the parameters are default).

The filter stores the CRC result value that the user expects to calculate. The MAC will then compare its calculated CRC16 value with this field.

The values are compared; if they match, it is considered a (remote) wake-up frame. If wake-up frame interrupt and PMT interrupt are enabled, then...

A PMT interrupt is generated.

Additionally, according to the bit definition of the PMT control status register, if the GU bit is set, then unicast frames that have passed frame filtering will also be...

It is considered to be a wake-up frame.

31.5.7 IEEE1588 PTP 31.5.7.1

PTP Principles and Implementation

The IEEE 1588 standard defines a precise protocol for acquiring time, with the goal of achieving time synchronization within a 10-microsecond error range.

The NTP protocol was originally capable of time synchronization at the 200-microsecond level, but in reality, this level is still insufficient for industrial automation.

To address the need for time synchronization across domains, the Network Precise Clock Synchronization Committee drafted the PTP protocol, which was adopted by the IEEE Standards Committee at the end of 2002.

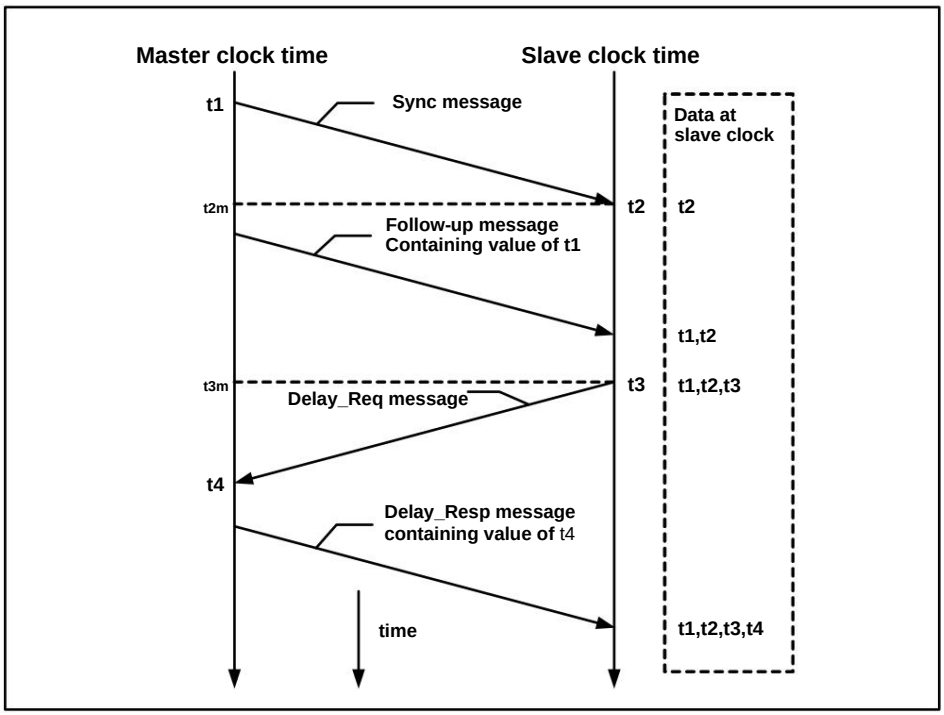
It will be adopted as the IEEE 1588 standard.

The implementation of the PTP protocol requires both the host and slave devices to accurately record the time of MAC reception and transmission of Ethernet frames. This requires both the host and slave devices to...

Each device has its own high-precision time counter. Subsequently, the PTP-enabled master and slave devices synchronize time through a set procedure, allowing the slave device to...

This obtains the time difference between itself and the master and makes corrections. The diagram below shows the master-slave time synchronization process.

Figure 31-6 Timing diagram of IEEE 1588 PTP protocol synchronization message



Step-by-step description:

The master sends a SYN message to the slave. Upon receiving this message, the slave records its local time t2 when it receives the SYN message. The master then sends a follow-up message to the slave, containing the master's time t1 when it sent the SYN message. The slave sends a delay-req message to the master, recording the sending time t3. The master then sends a delay-resq message to the slave, containing the receiving time t4. In practice, the master sends a SYN message every two seconds, and each subsequent SYN message can be considered a follow-up message of the previous SYN message, both including the sending time of the previous SYN message. Through this process, the slave can determine the network delay Tdelay between the master and slave networks and calculate the time offset between the master and slave times: (t2-t1) + (t4-t3).

$$Tdelay = \frac{}{2}$$

The time of any host is the time minus the offset.

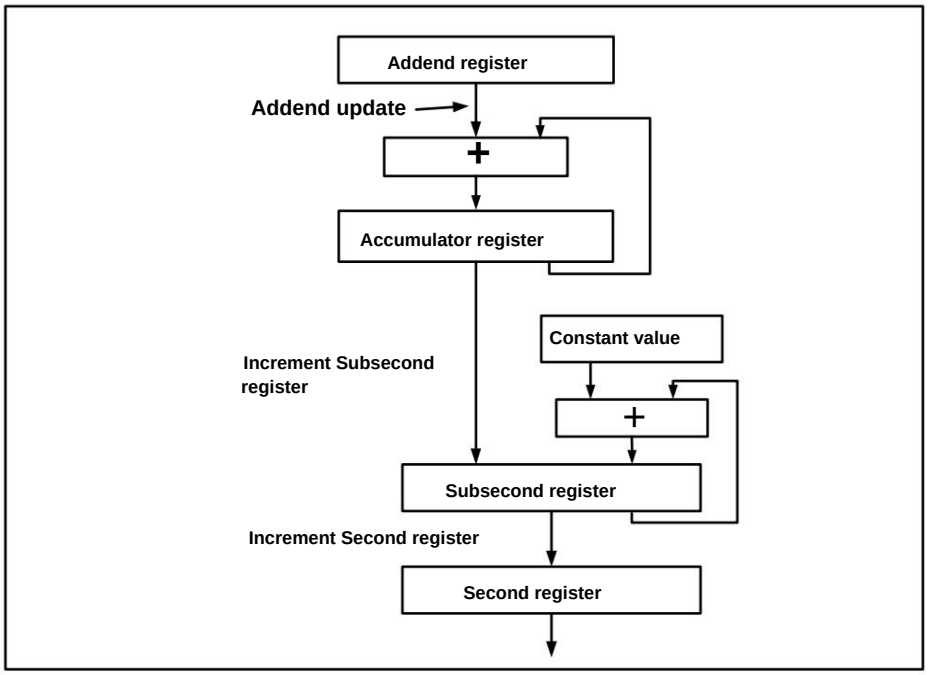
PTP's synchronization process is generally implemented using the UDP protocol, although users can also implement their own custom protocols. PTP is highly dependent on the latency stability of the internal network.

31.5.7.2 Local Time Update and Correction

To enable PTP (Perimeter to Teleport) functionality, the device needs a high-precision local time counter, accurate to at least nanoseconds. The microcontroller's Gigabit Ethernet counter PTP module has a 32-bit second counter and a 31-bit sub-second counter. When the sub-second counter overflows, the second counter increments automatically, thus achieving a local time resolution of approximately 0.46 nanoseconds.

There are two methods for updating local time: coarse adjustment and fine adjustment. The timing of the coarse adjustment update is externally determined. When a time update is needed, the time update bits (PTPTSCR:TSSTU) in the timestamp control register are set, and the microcontroller's PTP module subtracts or adds the value of the timestamp update register (TSHUR, TSLUR) to the second counter and sub-second counter. Coarse adjustment is a relatively simple and convenient time update mechanism, but its accuracy is poor. Fine adjustment is the more commonly used time update method. Its update process is as follows:

Figure 31-7 Flowchart for updating time using fine-tuning method



Using the fine-tuning mode requires a clear understanding of the system clock frequency and the fine-tuning process. Unlike the coarse-tuning mode, the fine-tuning mode updates the accumulator (32-bit, not listed in the register list, but the Accumulator register in the diagram) when it overflows. The accumulator increments the value of the addend register (PTPTSAR, the Addend register in the diagram) every system clock cycle. Once it overflows, a time update event is generated, that is, the value of the subsecond increment register (PTPSSIR, the Constant Value in the diagram) is added to the subsecond counter, completing the time update. When the subsecond counter overflows, the second counter increments. In fact, the time for the subsecond counter to increment by one bit is 1/(2^31) = 0.46566128730… nanoseconds. The user needs to ensure that the time taken for the accumulator to overflow is exactly equal to the value of the subsecond increment register multiplied by the time for the subsecond counter to increment by one bit. Local time correction is relatively simple. Set the time

correction bit (PTPTSCR:TSSTI) of the timestamp control register, and the second count will be updated.

The values of the timer and subsecond counter will be replaced by the values of the timestamp update counters (TSHUR, TSLUR).

31.6 DMA Operations

31.6.1 Overview In an

Ethernet transceiver, data enters the FIFO from the RGMII interface and is then transferred to RAM by the DMA. Even the largest ordinary Ethernet frame, with a data portion of up to 1500 bytes, can be received and transmitted in just a few tens of microseconds. Even considering MAC reception detection, DMA transfer time, and frame interval time, the CPU still needs to process a frame within one hundred microseconds. The advantages of Ethernet transceivers are high speed and high throughput. To maintain this advantage, the CPU intervention required during Ethernet frame reception and transmission must be minimized. This is where a dedicated DMA for Ethernet transceivers comes in. The DMA used in Ethernet is 256-bit and is managed by

the CPU through two data structures: traditional control and status registers, and receive and transmit descriptors. Because the DMA is 256-bit wide, the base address of the receive and transmit descriptor queue in RAM must be 32-byte aligned.

31.6.2 DMA Descriptor

Users typically access traditional peripherals by writing to control bits in registers and by reading status registers.

The status and return information are stored in these registers. These registers are independent of the kernel, SRAM, and non-volatile memory; they are physical entities and can be called "hardware registers." Traditional communication peripherals, such as USART or SPI, have a data register to temporarily store transmit and receive data.

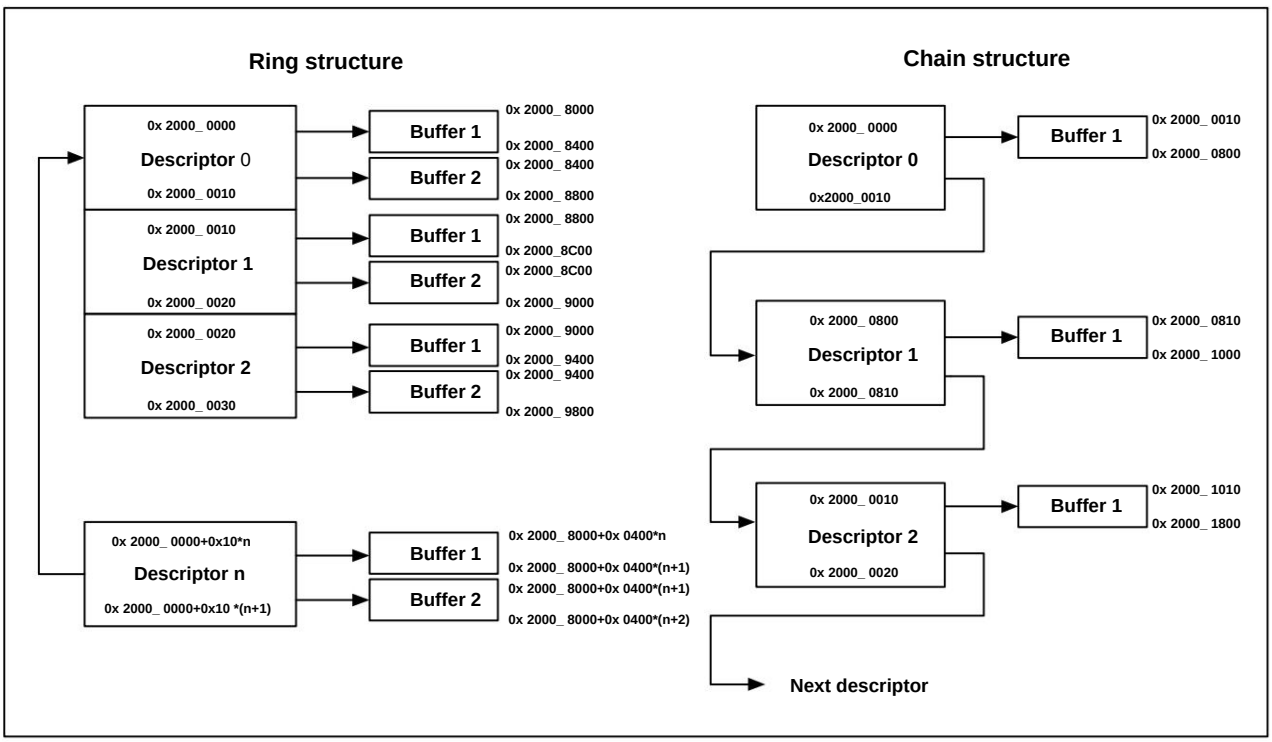
The data is stored in a specific address space by a DMA that stores all received data in a unified manner.

Ethernet transceivers, with their extremely high data transmission speed, massive data throughput, and unique data frame organization, differ from traditional communication transceivers in their operation. Ethernet transceivers receive large amounts of data as densely as possible, requiring them to store the received data stream in frames within a separate memory space. They then handle the receiving and sending operations themselves, allowing the CPU to read and process this data with minimal operations, freeing up the corresponding memory space and ensuring that the DMA controller and CPU do not conflict over memory usage permissions. The size of the buffer allocated in memory to temporarily store Ethernet frames is generally set to the maximum packet length of an Ethernet frame, which IEEE 802.3 specifies as 1518 bytes (including the source and destination hardware addresses, length or type fields, and CRC32 checksum field, totaling 18 bytes). The number of Ethernet frame transmit/receive buffers is determined by the user based on the actual frequency of interaction and the microcontroller's memory resources.

Ethernet frame buffers are organized in a queue-like manner. In the receiving direction, Ethernet frames are enqueued by being written to memory via DMA through the MAC and dequeued by being read by the CPU. In the transmitting direction, Ethernet frames are enqueued by being written to memory by the CPU and pushed onto the transmission buffer by DMA FIFO (First-In, First-Out) dequeues the data. The depth of the queue is the number of buffers. Unlike ordinary communication peripherals, the starting address and number of buffers in an Ethernet frame are not fixed in a register, but managed by a special data structure. This data structure is stored in memory, and each unit of this data structure manages one buffer, storing the buffer's starting address, length, settings required by the DMA controller when calling this buffer, the write-back status after DMA transmission completion, and the address of the next unit of this data structure. This data structure functions as a control register or status register in traditional communication peripherals, but its actual location is in memory, so it can be called a "software register," formally known as a "DMA descriptor."

DMA descriptors are divided into transmit and receive types, each with a fixed format. The storage structure of DMA descriptors is of two types: a linked list, where the fourth word of each descriptor is the address of the next descriptor, and the DMA controller directly reads the next descriptor from TDes3/RDes3; and a ring structure, where all descriptors must be closely arranged, and the DMA controller fetches the next descriptor from the end of the current descriptor. When the DMA controller detects the TER/RER bit in TDes4/RDes4, it fetches the next descriptor from the beginning of the descriptor list. The address of the descriptor array must be 32-byte aligned. The size of a single regular descriptor is 16 bytes, and the size of a single enhanced descriptor is 32 bytes. Figure 31-8 illustrates a buffer allocation scheme for the ring and linked structures, respectively, for user

reference when allocating space. Figure 31-8 Buffer Allocation Schemes for Ring and Linked Structures



31.6.2.1 Transmit DMA Descriptor Tables 31-11

and 31-12 show the structure of the transmit descriptor.

Table 31-11 Structure of a Regular Send Descriptor

	31							0
TDes0	OWM (31)	CTRL (30:26)	TTSE (25)	Save Keep	control (23:20)	reserve (19:18)	TTSS (17)	state (16:0)
TDes1	reserve (31:29)	Buffer 2 byte count (28:16)			reserve (15:13)	Buffer 1 byte count (12:0)		
TDes2	Buffer 1 address							
TDes3	Buffer 2 address / address of the next descriptor							

Table 31-12 Structure of the Enhanced Send Descriptor

	31							0
TDes0	OWM (31)	CTRL (30:26)	TTSE (25)	Save Keep	control (23:20)	reserve (19:18)	TTSS (17)	state (16:0)
TDes1	reserve (31:29)	Buffer 2 byte count (28:16)			reserve (15:13)	Buffer 1 byte count (12:0)		
TDes2	Buffer 1 address							
TDes3	Buffer 2 address / address of the next descriptor							
TDes4	reserve							
TDes5	reserve							
TDes6	Low bit of timestamp							
TDes7	High bit of timestamp							

As shown in the table above, a typical send descriptor consists of four 32-bit words: TDes0, TDes1, TDes2, and TDes3.

TDes0 is used for control and returning the transmission status, TDes1 is used to indicate the transmission length, and TDes2 is used to indicate the position of the transmission buffer.

TDes3 is used to represent the address used for the second send buffer (when TCH is not set).

Compared to the regular send descriptor, the enhanced send descriptor adds four 32-bit words: TDes4, TDes5, TDes6, and TDes7.

In the IEEE 1588 standard, TDes6 is used to represent the low-order bits of the timestamp, and TDes7 is used to represent the high-order bits.

The descriptions of each 32-bit word are as follows.

Table 31-13 Definitions of each bit of TDes0

31	30	29	28	27	26	25	24					23:22	twenty one	19	19:18	17	16	
OWN	IC	LS	FS	DC	DP	TTE	Res					CIC		TER	TCH	Res		TSS	IHE	

15	14	13	12	11			10	9	8	7	6	5	4	3	2	1	0
ES	JT	FF	IPE	LCA	NC	LCO	EC	VF				CC			ED	UF	DB

Bit	name	describe

31	OWN	Descriptor ownership bit. This bit indicates who owns this descriptor. 1: This descriptor belongs to the DMA; the CPU does not have permission to modify this descriptor. 0: This descriptor belongs to the CPU, and the CPU can modify the value of this descriptor. When this bit is 0, after the CPU completes its operations on the descriptor and buffer, it is... The CPU sets this bit to 1; when this bit is 1, the DMA completes the process of accessing the descriptor and buffer. After the operation, the DMA automatically writes 0. This completes the mapping between the user software and hardware. The handover of operations between the descriptor and the send/receive buffer.
30	IC	Transmission completion interrupt enable bit. When this bit is set, after transmitting the current frame, The transmit interrupt flag (ETH_DMASR: TS) will be set.
29	LS	End-of-segment indicator bit. Setting this bit indicates that the buffer indicated by this descriptor contains... The end of the frame.
28	FS	First segment indicator bit. Setting this bit indicates that the buffer indicated by this descriptor contains... The beginning of the frame.
27	DC	Disable automatic CRC calculation. Setting this bit prevents the DMA controller from calculating the CRC. The CRC32 checksum of the Ethernet frame will not have a value appended to the end of the frame. This bit This setting is only effective when the FS bit is set. Additionally, the DP bit has higher priority when set. In this position.
26	DP	Disable autofill bit. Setting this bit prevents the DMA controller from filling strings shorter than 64 words. The DMA adds auto-padding to the Ethernet frame. When this bit is 0, the DMA will automatically... Add padding and CRC checksum for Ethernet frames shorter than 64 bytes, ignoring... Whether DC is set.
25	TTE	Timestamp transmission enable bit. Provided ETH_PTPTSCR: TSE is set. Setting this bit will cause the DMA controller to print the Ethernet frame indicated by the face descriptor. Enable IEEE 1588 timestamp functionality. This bit is only valid when FS is set.
twenty four	Reserved	Not used.
[23:22] CIC		Checksum and insertion control field. 00: Disable checksum insertion; 01: Enable only the calculation and insertion of the IP header checksum; 10: Retained; 11: Enable the calculation and insertion of IP header checksum and payload checksum. Enable the calculation of the false header checksum.
twenty one	TER	End-of-transmission descriptor flag setting bit. The user sets this bit to instruct DMA control... The controller indicates that the current send descriptor is already the last one in the send descriptor array. Descriptors. The DMA controller will next read the first element of the transmit descriptor array A descriptor.
20	TCH	Next descriptor address valid indicator bit. This bit indicates that the second address is valid. The address of the next descriptor, not the address of the next buffer. This location. When the bit is set, the value of the TBS2 field has no effect. This bit is only valid when the FS bit is set. The TER bit has higher priority than this bit.
[19:18] Reserved		Not used.
17	TSS	Send timestamp capture status bit. The DMA controller sets this bit to indicate that transmission... The interstipation has been captured and stored in TDes6 and TDes7.
16	IHE	IP header error status bits. The DMA controller checks the received data. IP header: For IPv4, the DMA controller checks if the header length field is correct. Yes; for IPv6, the DMA controller checks if the header is 40 bytes.

		In addition, the IP protocol type must be consistent with the type/length field in the Ethernet frame.
15	ES	Error summary bit. The DMA controller sets this bit if an error is encountered while sending a frame. The ES bit is set when one of the following bits is set: UF[TDES0:1] Data underflow error bits; IPE[TDES0:12] IP data error bits; Clear the empty bits in frame FF[TDES0:13]; JT[TDES0:14] Jabber timeout; IHE[TDES0:16] IP header error bit.
14	JT	Jabber timeout bit. When this bit is set, it indicates MAC address spaceout. A verbose timeout error occurred at the sending end. This bit is only present in the JD bit. (ETH_MACCR:22) will only be set if it has not been set before.
13	FF	Frame clear bit. This bit is set when the DMA control is activated by a CPU command. The device clears the frame from the FIFO.
12	IPE	IP header error indicator bits. MAC will flag received TCP/UDP/ICMP packets as errors. The total packet length in the IPv4 or IPv6 header is compared with the actual packet length. If there is a discrepancy, this position will be used.
11	LCA	Carrier loss indication bit. This bit is set to indicate that a carrier loss occurred during frame transmission. The CSR signal is invalid, meaning it is in a non-functional state. This bit is only active in half-work mode. use.
10	NC	No carrier indication bit. This bit indicates the carrier sense signal of the physical layer when the frame is transmitted. This bit is invalid. This bit only works in half-work mode.
9	LCO	Late Collision Indicator Bit. This bit indicates that a collision occurred after the preamble of the frame has been sent. This bit only works in half-working mode.
8	EC	Too many collisions indicator bit. This bit indicates that more than 16 bits of a frame were transmitted. Conflict. If the RD bit of the MACCR is set, this bit indicates that it was sent only once. Conflict. This bit only works in half-working mode.
7	VF	VLAN bit. This bit is set when a VLAN frame is sent.
[6:3]	CC	Collision counter field. This field indicates how many collisions occurred during frame transmission. EC is invalid when set. This field only works in half-duration mode.
2	EC	Extend too many indicator bits. This bit setting indicates that when MACCR's DC is set, The transmission of the frame failed because the delay exceeded 24288 bits. This bit only exists in... It works in half-work mode.
1	UF	UF data underflow error bit. This bit is fetched from the specified RAM when the DMA controller sends data. When the data detection buffer is empty, transmission enters a pause state, and the corresponding bit is set. The relevant bits of the DMASR register.
0	DB	Delay indicator bit. This bit is set to indicate that transmission was lost due to carrier occupancy. Failure. This bit only works in half-working mode.

Table 31-14 Definitions of each bit of TDes1

31	30 29 28 27 26 25 24 23 22 21 20 19														18 17 16		
Reserved															TBS2		

15 14 13 12 11					10 9		8	7	6	5	4	3	2	1	0
Reserved					TBS1										

Bit	name	describe
[31:29] Reserved		Not used.
[28:16] TBS2		Size of send buffer 2. (Length must be a multiple of 32)
[15:13] Reserved [12:0]		Not used.
	TBS1	Size of send buffer 1. (Length must be a multiple of 32)

Table 31-15 Definitions of each bit in TDes2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16									
TBAD1																								

15	14	13	12	11			10	9	8	7	6	5	4	3	2	1	0
TBAD1																	

Bit	name	describe
[31:0] TBAD1		TDes2 uses all 32 bits as a single unit to store transmit buffer 1. The address of the buffer. (The buffer's starting address must be 32-byte aligned)

Table 31-16 Definitions of each bit in TDes3

31	30	29	28	27	26	25	24	23	22	21	20	19	20	17	16									
TDAD2																								

15	14	13	12	11			10	9		8	7	6	5	4	3	2	1	0
TDAD2																		

Bit	name	describe
[31:0] TDAD2		This field is used to store the address of the next file descriptor. When TCH is not set, this... This field stores the address of send buffer 2. (The starting address of the buffer requires 32 words.) Section alignment)

31.6.2.2 Receive DMA Descriptor

Table 31-17 Structure of a Standard Receiver Descriptor

	31																0
RDes0	OWM (31)	state (30:0)															
RDes1	control (31)	reserve (30:29)	Buffer 2 byte count (28:16)										RER (15:14)	reserve 13	Buffer 1 byte count number (12:0)		
RDes2	Buffer 1 address																
RDes3	Address of buffer 2, address of the next descriptor																

Table 31-18 Structure of the Enhanced Receiver Descriptor

	31					0
RDes0	OWM (31)	state (30:0)				
RDes1	control (31)	reserve (30:29)	Buffer 2 byte count (28:16)	RER (15:14)	reserve 13	Buffer 1 byte count number (12:0)
RDes2	Buffer 1 address					
RDes3	Address of buffer 2, address of the next descriptor					
RDes4	reserve					
RDes5	reserve					
RDes6	Low bit of timestamp					
RDes7	High bit of timestamp					

As can be seen from the table above, a standard receive descriptor is also composed of four 32-bit words: RDes0, RDes1, RDes2, and RDes3. RDes0 primarily returns the status during reception, RDes1 contains the length of the received data, and RDes2 defines the address of the receive buffer. RDes3 is the address of the second buffer (RCH not set), and the address of the next buffer (RCH set).

Compared to the regular receive descriptor, the enhanced receive descriptor adds four 32-bit words: RDes4, RDes5, RDes6, and RDes7. In the IEEE 1588 standard, RDes6 is used to represent the low-order bits of the timestamp, and RDes7 is used to represent the high-order bits.

The meanings of each bit in the receive descriptor are as follows:

Table 31-19 Definitions of each bit of RDes0

31	30	29:16
OWN	AFM	FL

15	14	13	12	11			10	9	8	7	6	5	4	3	2	1	0
ES	DE	SAF	LE	OE	VLAN	FS	LS	IPHCE	LCO	PT	RWT	RE	DE	CE	PCE								

Bit	name	describe
31	OWN	Descriptor ownership bit. This bit indicates who owns this descriptor. 1: This descriptor belongs to the DMA; the CPU does not have permission to modify this descriptor. 0: This descriptor belongs to the CPU, and the CPU can modify the value of this descriptor. When this bit is 0, after the CPU completes its operations on the descriptor and buffer, it is... CPU set to 1; When this bit is 1, after the DMA completes its operations on the descriptor and buffer, it is... DMA automatically writes to 0. This completes the user software and hardware's control over the descriptor and transmit/receive operations. Handover of operations in the buffer.
30	AFM	The destination address did not pass the flag. This occurs if the received frame does not pass the MAC address. If the address filter is applied, this flag will be set.

[29:16] FL		Frame length field. This field is valid when the ES bits [RDes0:15] are 0. In the LS bits... When [RDes0:8] is 1, this field indicates the frame received by the DMA controller. Length, including CRC; when the LS bit is 0, this field indicates the current DMA level. The cumulative length sent from the controller to memory is in bytes.
15	ES	Error summary bit. This bit is set when the MAC detects any of the following errors: CE[RDes0:1]: CRC error; RE[RDes0:3]: Receive error; RWT[RDes0:4]: Watchdog timeout; IPHCE[RDes0:7]: Giant frame (Note that when reporting IPHCE, it is necessary to distinguish between the two frames). (Is it a giant frame or an IP header error?) OE[RDes0:11]: Overflow error; DE[RDes0:14]: Descriptor error.
14	DE	Descriptor error bit. This bit being set indicates an error due to a buffer specified by the descriptor. The current frame cannot be loaded and the connection is interrupted, and the DMA does not occupy the next descriptor. This bit... This is only effective when the LS bits [RDes0:8] are set.
13	SAF	Source address filtering failed flag. This bit being set indicates that the frame failed to pass. MAC source address filter.
12	LE	Length error bit. Setting this bit indicates the actual received frame length and Ethernet length. The length indicated in the type/length field does not match. This bit is only present in FT[RDes0:5]. It is effective when set.
11	OE	Overflow error bit. This bit is set to indicate that the receiver has overflowed due to a FIFO overflow. The received frame was corrupted.
10	VLAN	VLAN tag bit. Setting this bit indicates that a VLAN frame has been received.
9	FS	First descriptor indicator bit. This bit indicates that this descriptor contains the first frame. head.
8	LS	Tail descriptor indicator bit. This bit indicates that this descriptor contains the end of a frame. tail.
7	IPHCE	IP header checksum error flags. This setting indicates whether it is IPv4 or IPv6. The header contains an error; the specific reasons may be: 1. The protocol indicated by the Ethernet frame type/length field does not match the actual IP version. Consistent; 2. The IP header checksum is incorrect; 3. The IP header length is incorrect.
6	LCO	Late Conflict Indicator Bit. This bit indicates that a late conflict has occurred. This bit only... It works in half-duplex mode.
5	FT	Frame type indicator bit. A value of 1 indicates that the received frame is of Ethernet type. Encapsulated frame (RFC 894). A value of 0 in this bit indicates that the received frame is... Frames encapsulated using the IEEE 802.3 type (RFC1042). When the frame length is less than 14... This bit is invalid when dealing with bytes. Note: For special meanings of FT, please refer to Table 31-18.
4	RWT	Receive watchdog timeout flag. This bit indicates that when receiving the current frame, The watchdog timed out, and the current frame was truncated.
3	RE	Receive error flag. This bit is set to indicate the error during frame reception (RX_DV). When valid, the RX_ERR signal is active.
2	DE	Dribble bit error bit. This bit indicates the length of the frame received by the MAC.

		If the degree is not an integer multiple of 8 bits, there may be a phenomenon of missing cycles.
1	CE	CRC error. This bit indicates that a CRC check error exists in the received frame. This bit is only valid when the LS bit [RDes0:8] is set.
0	PCE	Payload checksum error. This bit indicates the MAC received... The TCP/UDP/ICMP packet does not match the value of its checksum field identifier.

It can be seen that a verification mechanism is implemented in the MAC reception process. In fact, this is done at the Ethernet frame layer (data link layer) and network layer. Both IPv4/IPv6 and the transport layer (TCP/UDP/SCTP) specify the data length for their respective layers and take certain measures to ensure the correctness of the data content.

Verification of methods. Bits 0, 5, and 7 of RDes0 all mention data verification, which are summarized in the table below.

Table 31-20 Relationship between the values of RDes0:7/5/0 and the received frame states

RDes0:5 RDes0:7 RDes0:0			
Frame type Indicator bit (FT)	IP masthead proofreading Check and error Flag (IPHCE)	Load calibration Verification and Error error (PCE)	Frame state
1	0	0	IP type frame, no IP header and payload checksum errors detected.
1	0	1	IP type frame, payload checksum error
1	1	0	IP type frame, IP header checksum error.
1	1	1	IP type frame, IP header checksum error, payload checksum error
0	0	0	IEEE 802.3 type encapsulated frame (RFC1042)
0	0	1	IP type frame, no IP header checksum error detected, because load check is not affected. Supported but not detected
0	1	1	Frames that are not IP type (e.g., ARP frames)
0	1	0	reserve

Table 31-21 Definitions of each bit of RDes1

31	30 29 28 27 26 25 24 23 22 21	2019	18 17 16
DIC Reserved		RBS2	

15 14 13 12 11	10 9	8	7	6	5	4	3	2	1	0
RER RCH Res	RBS1									

Bit	name	describe
31	DIC	Disable the receive completion interrupt setting bit.
[30:29] Reserved		Not used.
[28:16] RBS2		Receive buffer size 2. (Length must be a multiple of 32)
15	RER	End-receive descriptor flag. Indicates that the current descriptor is the last one symbol. The DMA controller will return to the descriptor pair column base address register. (R32_ETH_DMARDLAR) fetch the next descriptor.
14	RCH	Next receive descriptor address valid bits. This bit indicates the last 32-bit address. The bit word contains the address of the next receive descriptor; otherwise, it contains the address of the second buffer. The address of the district.
13	Reserved	Not used.
[12:0] RBS1		Size of receive buffer 1. (Length must be a multiple of 32)

Table 31-22 Definitions of each bit of RDes2

31	30 29 28 27 26 25 24 23 22 21 20 19								18 17 16	
RBAD1										

15 14 13 12 11			10	9	8	7	6	5	4	3	2	1	0
RBAD1													

Bit	name	describe
[31:0] RBAD		RDes2 uses all 32 bits as a single unit to store the receive buffer. Address. (The starting address of the buffer must be 32-byte aligned)

Table 31-23 Definitions of each bit of RDes3

31 30 29 28 27 26 25 24 23 22 21 20 19									18 17 16	
RDAD2										

15 14 13 12 11			10 9		8	7	6	5	4	3	2	1	0
RDAD2													

Bit	name	describe
[31:0] RDAD		When RCR is set, it is used to store the address of the next buffer; otherwise, it is used to store... The address of the second buffer. (The starting address of the buffer must be 32-byte aligned)

31.6.3 Data Buffer Alignment Since

both the transmit/receive buffer and the transmit/receive descriptor are invoked by the DMA controller and actually reside in RAM space, and DMA is 256...
Since they are bit-based, setting up send/receive descriptor queues and buffers requires ensuring that their starting addresses are aligned to 32 bytes.

To improve efficiency, it is most appropriate to receive an Ethernet frame entirely within a single buffer, such as setting the buffer to 1518 bytes.
It can accommodate the largest standard Ethernet frame, including source and destination address fields, length type fields, data padding fields, and CRC check fields. (Required)
Note that tagged VLAN frames have a maximum frame length of 4 bytes (1522 bytes) longer than a regular Ethernet frame because of buffering.
The length of the buffer must be a multiple of 32, so a buffer can be set to 1536 bytes.

31.6.4 DMA Transmit/Receive

Configuration The DMA is responsible for automatically transferring and pushing the received and transmitted data. However, if a fatal error occurs, the DMA will stop operating and...
New DMA status register. Users need to initialize the DMA and manually start it for it to continue operating.

31.6.4.1 Send DMA Configuration

The steps for establishing the DMA controller's operational mechanism are as follows:

1) Set up a send buffer queue and fill the send buffer with the content to be sent. Set up a send descriptor queue and fill in the send descriptors.

Set each bit in each field, set OWN, and hand over management of the send descriptor to the DMA controller; register the initial address of the descriptor.

Enter into the DMATDLAR register;

2) Set the ST bit (DMAOMR:13) to enable DMA;

3) In running mode, the DMA descriptor automatically reads the contents of the send descriptor and pushes the data according to the address and length indicated by it.

The process continues until the FIFO is reached. After completion, the DMA will read the next transmit descriptor in a chained structure for the next transmission.

If the DMA controller detects that the OWM bit is not set, resulting in unauthorized access to the send descriptor, or if there is another normal error, it will terminate the transmission.

Set the TBUS bit (DMASR:2) or other bits (for errors not caused by OWN being 0) and the NIS bit (DMASR:16).

4) A single frame is not allowed to span multiple descriptors; a frame must be clearly associated with one descriptor and one buffer.

5) If the MAC has IEEE1588 PTP mode enabled and enhanced descriptors are enabled, then the DMA controller will push data to the FIFO.

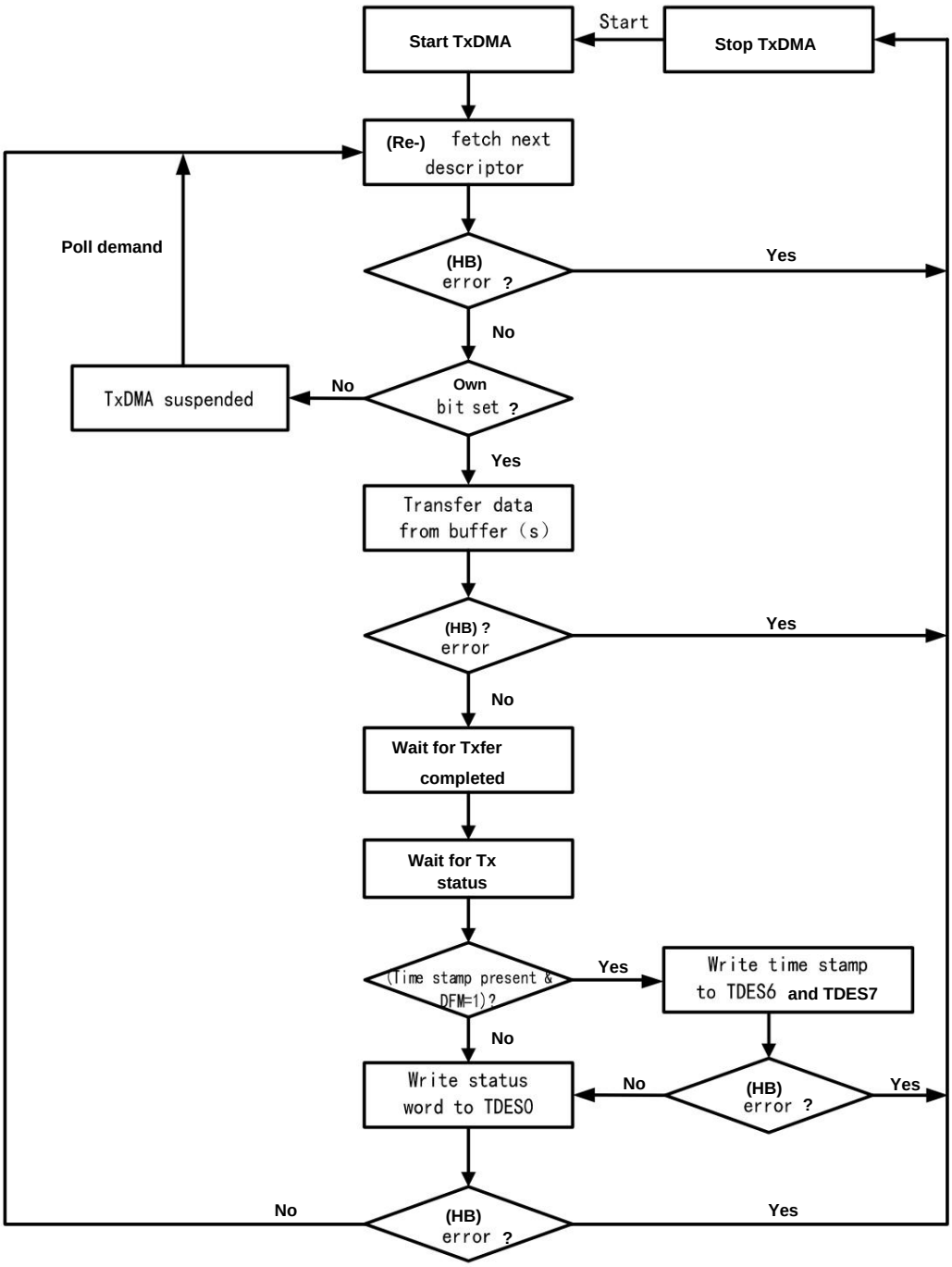
Afterwards, the MAC will write the transmission timestamp to TDes6 and TDes7 and reset the OWN bit.

6) After sending a frame, if the transmit descriptor enables the transmit completion interrupt (TDes1:31 is set), the DMA controller will...

Set the transmit completion interrupt flag (DMASR:0), and then continue to fetch the next transmit descriptor.

The following diagram illustrates the default sending process.

Figure 31-9 Sending process



31.6.4.2 The steps for configuring and

establishing a DMA receive transfer mechanism are as follows:

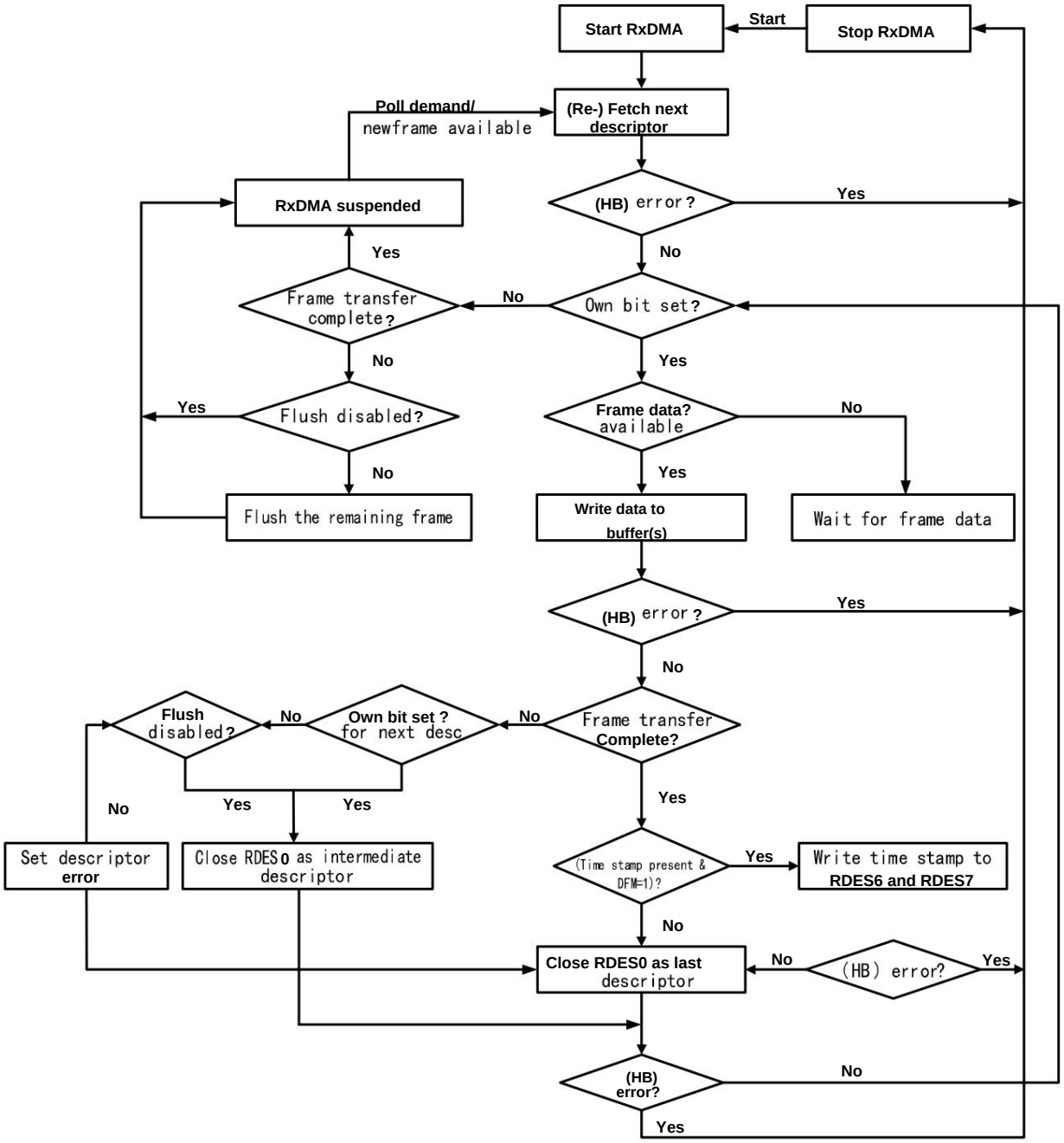
1) Set up the transmit buffer queue and descriptor queue, and configure each field and bit of the receive descriptor; register the initial address of the descriptor.

1) Set the OWN bit in the DMARDLAR register to grant the DMA controller access to the descriptor; 2) Set the SR bit to start the receive process; 3) During the receive flow mechanism, the DMA controller acquires the next descriptor, checks the receive descriptor configuration, performs filtering and identification checks on the frame content when the FIFO receives the next frame, forwards the frame data to the buffer specified by the descriptor, writes the descriptor status field, acquires the next receive descriptor, and reports a receive completion interrupt. If the frame fails the filter, it will be marked or discarded. If the frame has a checksum error, CRC error, or is too short, it will be marked. If the frame is too long, it may be interrupted or a receive watchdog timeout error will be reported. The DMA controller will stop the receive process if it encounters a fatal error such as an unavailable receive descriptor, which the user needs to pay special attention to; 4) The user needs to enable at least one receive completion interrupt and restore the used receive descriptors to the standby state in the Ethernet interrupt function to ensure that the receive process can continue uninterrupted. Users can pass the address of the data buffer to be processed in the interrupt function, or handle some abnormal events that interrupt the receiving process.

5) If the MAC has IEEE1588 PTP mode enabled and enhanced descriptors are enabled, data is transferred in the DMA controller, and the descriptor status is written. When the domain is accessed, the current timestamp is also written to the last two words of the descriptor.

The following diagram illustrates the default receive stream mechanism:

Figure 31-10 Receiving Process



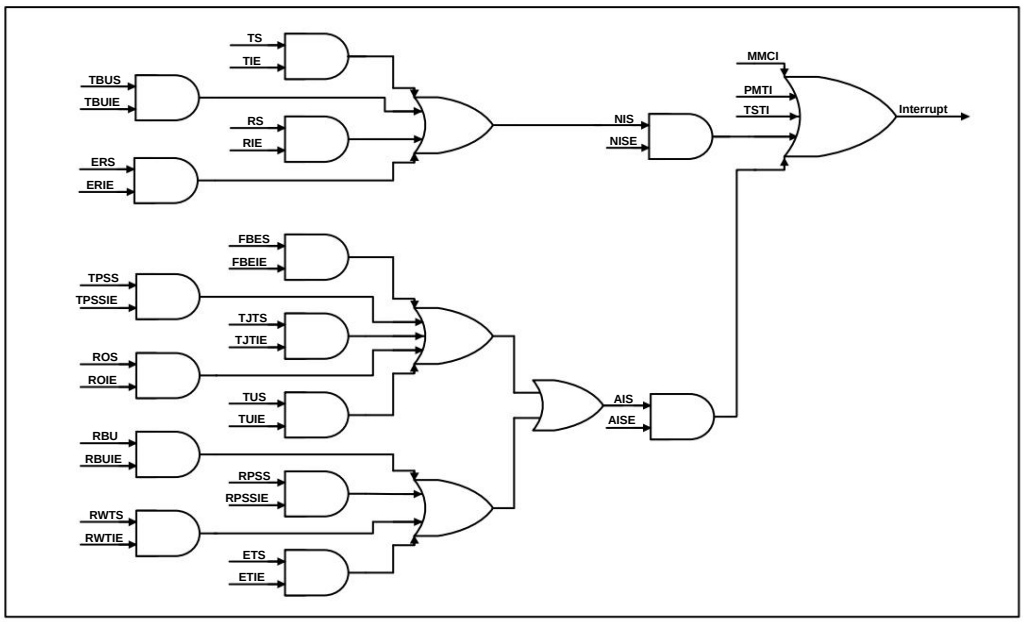
31.7 Interruption

Ethernet transceivers have two interrupt vectors: one for the Ethernet Wake-up event and the other for normal transmit/receive events. When a Wake-up event is detected...
A wake-up frame or magic frame triggers an Ethernet wake-up event. Common Ethernet transmit/receive interrupt events include DMA interrupts and ETH interrupts.
Break.

31.7.1 DMA Interrupts DMA interrupts

can be broadly divided into two groups: normal interrupts (NIS) and abnormal interrupts (AIS). Abnormal interrupts generally mean...
Any anomalies in data transmission or reception require special attention. When handling interrupts, users need to retrieve all interrupt flags and update the flags that have already been generated.
All interrupt sources are handled. The following diagram illustrates the interrupts of an Ethernet transceiver.

Figure 31-11 Interruption Diagram



DMA interrupt is the most important interrupt in an Ethernet transceiver. Typical user logic relies on interrupts to receive frames and confirm frame transmission. Send it out, or promptly handle any interrupted sending and receiving logic.

31.7.2 ETH Interruptions

ETH interrupts mainly include PTP time alarm triggers, transmit/receive counters reaching a certain set value in the MMC register, and PMT. Its primary use is functional, and it is not as complex or important as DMA interrupts. Users can use PTP interrupts to implement alarms. Speed testing can be implemented using MMC interrupts, or remote wake-up can be implemented using PMT interrupts. Additionally, ETH supports remote wake-up when the built-in physical layer connection status changes. An interruption occurs when changes occur.

31.7.3 PMT Interrupt The

reason for reiterating the PMT interrupt separately is that this interrupt has its own interrupt vector, and care should be taken when using it.

31.8 Register Description

Table 31-24 List of Ethernet-related registers

MAC control related register address mapping

name	Access address	describe	Reset value
R32_ETH_MACCR	0x40028000	MAC control register	0x00000000
R32_ETH_MACFFR	0x40028004 Frame Filtering Register;		0x00000000
R32_ETH_MACHTHR	0x40028008 Hash List Register High Byte; 0x4002800C		0x00000000
R32_ETH_MACHTLR	Hash List Register Low Byte		0x00000000
R32_ETH_MACMIIAR	0x40028010	MII Address Register	0x00000000
R32_ETH_MACMIIDR	0x40028014	MII Data Register	0x00000000
R32_ETH_MACFCR	0x40028018	MAC flow control register	0x00000000
R32_ETH_MACVLAN	0x4002801C	VLAN tag register	0x00000000
R32_ETH_MACRWUFFR	0x40028028 Wake-up Frame Filter Register		0x00000000
R32_ETH_MACPMTCSR	0x4002802C	PMT Control and Status Register	0x00000000
R32_ETH_MACSR	0x40028038	MAC interrupt status register	0x00000000

R32_ETH_MACIMR	0x4002803C	MAC interrupt mask register	0x00000000
R32_ETH_MACA0HR	0x40028040	MAC Address Register 0 High 32 Bits	0x8000FFFF
R32_ETH_MACA0LR	0x40028044	MAC Address Register 0, lower 32 bits	0xFFFFFFFF
R32_ETH_MACA1HR	0x40028048	MAC Address Register 1 High 32 Bits	0x0000FFFF
R32_ETH_MACA1LR	0x4002804C	MAC Address Register 1, lower 32 bits	0xFFFFFFFF
R32_ETH_MACA2HR	0x40028050	MAC Address Register 2 High 32 Bits	0x0000FFFF
R32_ETH_MACA2LR	0x40028054	MAC Address Register 2, lower 32 bits	0xFFFFFFFF
R32_ETH_MACA3HR	0x40028058	MAC Address Register 3 High 32 Bits	0x0000FFFF
R32_ETH_MACA3LR	0x4002805C	MAC Address Register 3, lower 32 bits	0xFFFFFFFF
R32_ETH_PHY_CFGR	0x40028080	MAC_PHY configuration register	0x40000001

MMC control-related register address mapping. Note: addresses are not contiguous.

name	Access address	describe	Reset value
R32_ETH_MMCCR	0x40028100	MMC Control Register	0x00000000
R32_ETH_MMCRIR	0x40028104	MMC Receive Register	0x00000000
R32_ETH_MMCTIR	0x40028108	MMC Transmit Interrupt Register	0x00000000
R32_ETH_MMCRIMR	0x4002810C	MMC Receive Interrupt Mask Register	0x00000000
R32_ETH_MMCTIMR	0x40028110	MMC Transmit Interrupt Mask Register	0x00000000
R32_ETH_MMCTGFSCCR	0x4002814C	MMC good frame send counter 0x00000000 after a collision	
R32_ETH_MMCTGFMSCCR	0x40028150	MMC good frame counter 0x00000000 after multiple collisions	
R32_ETH_MMCTGFCR	0x40028168	MMC Good Frame Count Register	0x00000000
R32_ETH_MMCRFCECR	0x40028194	The MMC receives a CRC error count register with error count 0x00000000.	
R32_ETH_MMCRFAECR	0x40028198	MMC Receive Alignment Error Frame Count Register 0x00000000	
R32_ETH_MMCRAFCR	0x4002819C	MMC Receive All Frame Count Registers	0x00000000
R32_ETH_MMCRGUFCR	0x400281C4	MMC Received Unicast Frame Count Register	0x00000000

IEEE 1588 (PTP) related register address mapping

Name Access Address		describe	Reset value
R32_ETH_PTPTSCR	0x40028700	PTP Timestamp Control Register	0x00000000
R32_ETH_PTPSSIR	0x40028704	PTP subsecond increment register	0x00000000
R32_ETH_PTPTSHR	0x40028708	PTP timestamp register high bit	0x00000000
R32_ETH_PTPTSLR	0x4002870C	PTP timestamp register low bit	0x00000000
R32_ETH_PTPTSHUR	0x40028710	PTP timestamp update register high bit	0x00000000
R32_ETH_PTPTSLUR	0x40028714	PTP timestamp update register low bit	0x00000000
R32_ETH_PTPTSAR	0x40028718	PTP timestamp increment register	0x00000000
R32_ETH_PTPTTHR	0x4002871C	PTP target register high bit	0x00000000
R32_ETH_PTPTTLR	0x40028720	PTP Destination Register Low Bit	0x00000000

Note regarding DMA-related register address mapping: addresses are not contiguous.

name	Access address	describe	Reset value
R32_ETH_DMABMR	0x40029000	DMA bus mode register	0x00000001
R32_ETH_DMATPDR	0x40029004	DMA transmit query register	0x00000000
R32_ETH_DMARPDR	0x40029008	DMA Receive Queries Register	0x00000000

R32_ETH_DMARDLAR	0x4002900C	DMA Receive Descriptor Address Register	0x00000000
R32_ETH_DMATDLAR	0x40029010	DMA transmit descriptor address register	0x00000000
R32_ETH_DMASR	0x40029014	DMA Status Register	0x00000000
R32_ETH_DMAOMR	0x40029018	DMA Operation Mode Register	0x00000000
R32_ETH_DMAIER	0x4002901C	DMA interrupt enable register	0x00000000
R32_ETH_DMAMFBOCR	0x40029020	DMA lost frame register	0x00000000
R32_ETH_DMACHTDR	0x40029048	DMA Current Transmit Descriptor Register	0x00000000
R32_ETH_DMACHRDR	0x4002904C	DMA Current Receive Descriptor Register	0x00000000
R32_ETH_DMACHTBAR	0x40029050	DMA Current Transmit Buffer Register	0x00000000
R32_ETH_DMACHRBAR	0x40029054	DMA Current Receive Buffer Register	0x00000000

Internal 10M/100M physical layer related register addresses

Name Offset Address Description 0x00 Basic Control Register 0x01		Reset value
R16_BMCR	Basic Status Register 0x04 Auto-	0x3100
R16_BMSR	Negotiation Message Register 0x05	0x7849
R16_ANAR	Auto-Negotiation Link Capability Register	0x01E1
R16_ANALPAR	0x1F Page Select Register	0x0001
R16_PAGE_SEL		0x0000

Note: The offset address of the internal physical layer register is used in the interface. SMI

PAGE0 Related Register Address

name	Offset address	describe	Reset value
R16_PHY_STATUS	0x1A	PHY Status Register	0x0020
R16_INTERRUPT_IND	0x1E Interrupt	Indicator Register	0x0000

PAGE7 Relevant Register Address

name	Offset address		Reset value
R16_INTERRUPT_MASK	0x13	Description of the interrupt/LED function register	0x0030

31.8.1 Bits of MAC Control Related Registers

31.8.1.1 MAC Control Register (R32_ETH_MACCR)

Offset address: 0x00

31	30	29 28 27 26 25 24 23 22 21	2019	18	17	16
TCD[2:0]		Reserved	WD JD Reserved	IFG[2:0]		Reserved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FES[1:0]		Reserved	LM	DM	IPCO	Reserved	APCS		Reserved			TE	RE	TCF	Reserved

Bit	name	access		Reset value
[31:29]	TCD[2:0]	RW	Describes the transmit clock delay field. This field is used to delay the transmit clock. MAC control When the controller selects the transmission through the TCF bit (ETH_MACCR[1]),	0

			The clock outputs a time delay. The formula for calculating the delay time is: Tdalay=TCD(ETH_MACCR[31:29])*0.5ns;	
[28:24] Reserved		RO reserved.		0
23	WD	RW	Watchdog setting bit: 1: With the MAC address's watchdog timer disabled, it can receive a maximum of 16384 bytes of data. Too many frames; 0: Mac watchdog timer enabled, only accepting data up to 2048 bytes. Too many frames, the excessively long parts will be cut off.	0
22	JD	RW	Jabber setting bits: 1: With the Jabber timer disabled on the MAC, a maximum of 16384 characters can be sent. Ethernet frames; 0: If a user attempts to send a message exceeding 2048 bytes in length... If the frame is too large, the MAC will shut down the transmitter.	0
[21:20] Reserved		RO Reserved.		0
[19:17] IFG[2:0]		RW	Interframe gap setting field. This sets the minimum interval between sending two frames. Time gap. 000: 96-bit time; 001: 88-bit time; 010: 80-bit time; 011: 72-bit time; 100: 64-bit time; 101:56 bits of time; 110:48 bits of time; 111:40 timestamp.	0
16 Reserved		RO reserved.		0
[15:14] FES[1:0]		RW	Ethernet speed setting field: 00: 10 Mbit/s; 01: 100 Mbit/s; 10: 1 Gbit/s; 11: Reserved, not used.	0
13 Reserved		RO is reserved.		0
12	LM	RW Self-looping mode enable bit. Set this bit to enable self-looping mode.		0
11	DM	RW is the full-duplex mode enable bit. Setting this bit enables full-duplex mode.		0
10	IPCO	RW	IPv4 checksum enable bit: 1: Enable IPv4 verification; the MAC controller will check TCP... UDP and ICMP headers are checked and verified; 0: Disables the IPv4 checksum verification function on the receiving end, and the corresponding PCE, The PHCE flag is always 0. See the definitions of each bit in the receive descriptor.	0
[9:8] Reserved		RO reserved.		0
7	APCS	RW	Fill & CRC auto-strip enable bit: 1: MAC only applies when the length/type field value is less than or equal to 1500 bytes. At that time, remove the Pad/FCS field from the frame. If the length/type field is not present... A value greater than or equal to 1501 bytes will not change the frame content; 0: MAC does not modify the frame content.	0
[6:4] Reserved		RO is reserved.		0

3	TE	RW	Send enable bit: 1: Enable the MAC transmitter; 0: After sending the current frame, the MAC closes the transmitter and stops sending. Send any frame.	0
2	RE	RW	Receive enable bit: 1: Enable MAC receiver; 0: After receiving the current frame, the MAC closes the receiver and stops receiving. Receive any frame.	0
1	TCF	RW	Transmit clock toggle setting bit: 1: Invert the internal transmit clock of the chip as the chip's output. GTXC; 0: The chip's internal transmit clock is directly used as the chip's output. GTXC.	0
0	Reserved	RO is reserved.		0

31.8.1.2 MAC Frame Filtering Register (R32_ETH_MACFFR) Offset

Address: 0x04

31	30	29 28 27 26 25 24 23 22	2019	18	17 16
RA	Reserved				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					HPF	SAF	SAIF	PCF[1:0]		BFD	PAM	DAIF	HM	HU	PM

Bit	name	access	describe	Reset value
31 RA		RW	Receive all bits: 1: The MAC forwards all received frames to the receive queue, regardless of their origin. Did it pass through the filter? 0: MAC only forwards frames that have passed the filter to the receive queue.	0
[30:11] Reserved		RO is reserved.		0
10 HPF		RW	HASH filtering or perfect filtering options: 1. With HM or HU set, it conforms to the HASH filter. Even a perfect filter can filter by address; 0: Provided that HM or HU is set, as long as the hash condition is met... A filter can filter by address.	0
9	SAF	RW	Source address filter selection bits: 1: MAC will directly discard frames that have not passed the source MAC address filtering; 0: MAC marks frames that have not passed the source MAC address filtering.	0
8	SAIF	RW	Source address filtering result inverted bits: 1: If the source address of the received frame matches the address in the MAC address register... If the source address used is the same, it is considered that the source address filter has not been passed. 0: If the source address of the received frame matches the address in the MAC address register... If the source address used is inconsistent, it is considered that the source address filter has not been passed.	0
[7:6] PCF[1:0]		RW	Flow control frames pass through the control domain: 00/01: The MAC does not forward any flow control frames to the application;	0

			10: MAC forwards all flow control frames to the application, including those not yet connected. Flow control frames that pass through address filters; 11: MAC only forwards flow control frames that pass through the address filter.	
5	BFD	RW	Broadcast frame receive control bits: 1: Discard all broadcast frames; 0: Receive all broadcast frames.	0
4 PAM		RW	Control all multicast frame bits: 1: All multicast frames can be filtered by address; 0: Whether a multicast frame can pass the filter depends on the value of HM.	0
3	DAIF	RW	Reverse the control bits of the result of the target MAC address filter: 1: For multicast and unicast frames, determine whether the filter passes the result. Reversing the order will then take effect; 0: The filter results are working correctly.	0
2	HM	RW	Multicast frame filtering mode selection bit: 1. Perform HASH address filtering; 0: Perform perfect address filtering.	0
1	HU	RW	Unicast frame filtering mode selection bit: 1. Perform HASH address filtering; 0: Perform perfect address filtering.	0
0	PM	RW	Promiscuous mode enable bit. All frames can pass through the address filter, and Do not label the filtered results.	0

31.8.1.3 Hash List Registers (R32_ETH_MACHTHR, R32_ETH_MACHTLR)

Offset address: 0x08

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
HTH[31:16]														

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

Offset address: 0x0C

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
HTL[31:16]														

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

Bit name	access	describe	Reset value
[31:0] HTH[31:0]		The high 32 bits of the RW hash list.	0
[31:0] HTL[31:0]		The lower 32 bits of the RW hash list.	0

31.8.1.4 MII Address Register (R32_ETH_MACMIIAR) Offset Address:

0x10

31	30	29 28 27 26 25 24 23 22	twenty one	2019	18	17	16
Reserved							

15	14	13	12	11	10	9	8	7	6	5	4 3	2	1	0
PA[4:0]					MR[4:0]					Reserved	CR[2:0]		MW	MB

Bit	name	access	describe	Reset value
[31:16]	Reserved	RO (Reserved).		0
[15:11]	PA[4:0]	RW	Physical layer address field. The user writes the physical layer address to be operated into... This domain.	0
[10:6]	MR[4:0]	RW	Physical layer register address field. The user will specify the address of the registers they need to operate on. Write the address into this domain.	0
5	Reserved	RO is reserved.		0
[4:2]	CR[2:0]	RW	Clock range setting field. 000: HCLK/42; 001: HCLK/62; 010: HCLK/16; 011: HCLK/26; 100: HCLK/102; 101: HCLK/74; Other: Reserved.	0
1	MW	RW	Read/write settings: 1: Perform a write operation on the physical layer; 0: Perform a read operation on the physical layer.	0
0	MB	RW1Z	MII Busy Flag: This bit is set by the user to indicate that the hardware is commanded to begin reading or writing. During operations, the physical address, register address, and... The data field remains unchanged. Once the hardware clears this bit, the operation is complete.	0

31.8.1.5 MII Data Register (R32_ETH_MACMIIDR) Offset Address:

0x14

31	30	29 28 27 26 25 24 23 22	twenty one	2019	18	17 16
Reserved						

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

Bit	name	access	describe	Reset value
[31:16]	Reserved	RO is reserved.		0
[15:0]	MD[15:0]	RW	MII Operation Data Field. This field is used to store data that will be transmitted via the MII interface. Data read by the physical layer, or data written to the physical layer.	0

31.8.1.6 MAC Flow Control Register (R32_ETH_MACFCR)

Offset address: 0x18

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
PT[15:0]														

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

Bit	name	access		Reset value
[31:16]	PT[15:0]	RW	Describes the Pause interval field. This field is used to control the Pause time domain. The value is the time taken to send 64 bytes via the current MII interface.	0
[15:4]	Reserved	RO	reserved.	0
3	UPFD	RW	Unicast Pause frame detection bits: 1: MAC simultaneously checks if the Pause frame is in MAC address register 0. Defined unicast address; 0: MAC only accepts pauses with a unique address defined by the protocol specification. frame.	0
2	RFCE	RW	Receive flow control enable bit: 1: The MAC panel parses the Pause frame and shuts down the transmitter for a period of time; 0: MAC does not parse Pause frames.	0
1	TFCE	RW	Send flow control enable bit: 1: MAC enables flow control, allowing the transmission of Pause frames; 0: MAC disables flow control and does not send Pause frames.	0
0	FCB	RW1Z	Flow control busy flag. Setting this flag allows you to send a Pause frame. The data is cleared by hardware after transmission is complete. This is followed by a check of the entire MACFCR register. During row operations, the FCB bit must be set to 0.	0

31.8.1.7 VLAN Tag Register (R32_ETH_MACVLAN) Offset Address:

0x1C

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
Reserved														VLANT

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

Bit	name	access	describe	Reset value
[31:17]	Reserved	RO	reserved.	0
16	VLANT	RW	Tag comparison control bits: 1: Only use bits [11:0] of bytes 15 and 16 of the VLAN frame. All are compared with the corresponding bits in the VLANTI field; 0: All 16 bits of data in bytes 15 and 16 of the VLAN frame are... Compare with the VLANTI field.	0
[15:0]	VLANTI[15:0]	RW	Tag comparison sample field. According to the IEEE 802.1 protocol, VLAN frames The [15:13]th position is the user priority, and [12] is the specification format indicator.	0

			Symbol, bit [11:0] VLAN identifier field. If VLANTI is all 0, then the MAC will no longer care about VLAN frames. The 15th and 16th bytes, while the 13th and 14th bytes are 0x8100. (Note the endianness) When it is a VLAN frame, it is determined to be a VLAN frame.	

31.8.1.8 Wake-up Frame Filter Register (R32_ETH_MACRWUFFR) Offset

Address: 0x28

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
RWUFFR[31:16]														

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

Bit	name	access	The	Reset value
[31:0]	RWUFFR[31:0]	RW	wake-up frame filter register actually consists of eight different registers. It can read all registers by performing eight consecutive read operations, and... Performing eight consecutive write operations on it can write all eight registers. Registers. For a detailed description of each bit in these eight registers, see Section 31.5.6.3. The description.	0

31.8.1.9 PMT Control and Status Register (R32_ETH_MACPMTCSR)

Offset address: 0x2C

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
WFFRP	Reserved													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						GU	Reserved	PHY_PMEBR	WFR	MPR	Reserved	PHY_PMEB_EN	WFE	MPE	PD

Bit	name	access	This	Reset value
31	WFFRP	RW1Z	command resets the wake-up frame filter register pointer. Setting this bit will... The MACRWUFFR register is cleared to zero. This bit will be cleared within one clock cycle. The period will be automatically reset to zero.	0
[30:10]	Reserved	RO	Reserved.	0
9	GU	RW	Global unicast bit. Setting this bit will cause the MAC to send all packets passing through the filter. Unicast frames are all considered wake-up frames.	0
8	Reserved	RO	is reserved.	0
7	PHY_PMEBR	RZ	This bit is set when a wake-up frame from the PHY is received. It is automatically cleared upon reading. zero.	0
6	WFR	RZ	Wake-up frame receive flag. This flag is set when a wake-up frame is received. Bits. Reading automatically clears the bits.	0
5	MPR	RZ	Magic frame receive flag. This bit is set to 0 when a magic frame is received.	

			Bits. Reading automatically clears the bits.	
4	Reserved	RO	is reserved.	0
3	PHY_PMEB_EN	RW	PHYPMEB Enable bit. Setting this bit allows the PHY to generate a wake-up frame when it receives a wake-up frame. PMT incident	0
2	WFE	RW	Wake-up frame enable bit. Setting this bit allows a wake-up frame to be generated upon receipt. PMT event.	0
1	MPE	RW	Magic frame enable bit. Setting this bit allows magic frames to be generated upon receipt. PMT event.	0
0	PD	RW1Z	Power-down control bit. Setting this bit will put the MAC into power-down mode: Discard. All other frames until it receives a wake-up frame or magic frame, wake up. The MAC address is automatically cleared afterward. Before setting this bit, you need to disable WFE or MPE. Set.	0

31.8.1.10 MAC Interrupt Status Register (R32_ETH_MACSR) Offset

Address: 0x38

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
Reserved														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						TSTS	Reserved		MMCTS	MMCRS	MMCS	PMTS		Reserved	

Bit	name	access	describe	Reset value
[31:10]	Reserved	RO	reserved.	0
9	TSTS	RZ	Timestamp-triggered interrupt flag. This flag is triggered when the PTP system clock setting is reached. This flag will be set when the specified time is reached.	0
[8:7]	Reserved	RO	is reserved.	0
6	MMCTS	RO	MMC transmits the interrupt flag. When MMCTIR in the MMC register set... This bit is set when any interrupt occurs in the register. When the MMC register... When all MMCTIR registers in the group are cleared, this bit is also cleared.	0
5	MMCRS	RO	MMC receive interrupt flag. When MMCRIR in the MMC register set... This bit is set when any interrupt occurs in the register. When the MMC register... When all MMCRIR registers in the group are cleared, this bit is also cleared.	0
4	MMCS	RO	MMC status flag. Triggering an event occurs when MMCTS or MMCRS is set. This bit is set; it is cleared when both MMCTS and MMCRS are reset to zero. zero.	0
3	PMTS	RO	PMT status flag. In power-down mode, if the MAC receives a magic signal... If a MAC is woken up by a legal frame or a wake-up frame, this bit will be set. This bit is cleared to zero after clearing the WFR and MPR bits.	0
[2:0]	Reserved	RO	is reserved.	0

31.8.1.11 MAC Interrupt Mask Register (R32_ETH_MACIMR) Offset

Address: 0x3C

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16

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

Bit	name	access	describe	Reset value
[31:10]	Reserved	RO is reserved.		0
9	TSTIM	RW: Timestamp interrupt mask bit. Setting this bit will disable timestamp		
[8:4]	Reserved	interrupts. 0 RO: Reserved.		0
3	PMTIM	RW PMT interrupt mask bit. Setting this bit will mask the PMT interrupt.		0
[2:0]	Reserved	RO is reserved.		0

31.8.1.12 MAC Address Register 0 High 32 Bits (R32_ETH_MACA0HR)

Offset Address: 0x40

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
MO	Reserved													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACA0H[15:0]															

Bit	name	access	describe	Reset value
31	MO	RO is always 1.		1
[30:16]	Reserved	RO is reserved.		0
[15:0]	MACA0H[15:0]	The high 16 bits of the RW MAC address, i.e., bits 47 to 32.		0xFFFF

31.8.1.13 MAC Address Register 0 Lower 32 bits (R32_ETH_MACA0LR)

Offset Address: 0x44

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
MACA0L[31:16]														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACA0L[15:0]															

Bit	name	access	The	Reset value
[31:0]	MACA0L[31:0]	RW	lower 32 bits, or bits 31 to 0, describe the MAC address. Generally, a MAC address... The address of address register 0 is the address of the MAC itself.	0xFFFF FFFF

31.8.1.14 MAC Address Register 1, High 32 Bits (R32_ETH_MACA1HR),

Offset Address: 0x48

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
AE	SA	MBC[5:0]								Reserved				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACA1H[15:0]															

Bit	name	access	describe	Reset value
31 AE		RW	Address filtering enable bit: 1: The address filter uses MAC address 1 for perfect filtering; 0: Address filter ignores MAC address 1.	0
30 SA		RW	Address role selection slot: 1: MAC address 1 is used to compare with the source address of the received frame; 0: MAC address 1 is used to match the destination address of the received frame. Compare.	0
[29:24] MBC[5:0]		RW	Mask control field. Each bit of the MBC corresponds to the mask MAC address. A specific byte at address 1 is used to prevent a specific word at MAC address 1 from being accessed. The section participates in the comparison of the source address or destination address of the received frame. Setting R32_ETH_MACA1HR[29] will disable MACA1H[15:8]; Setting R32_ETH_MACA1HR[28] will disable MACA1H[7:0]; Setting R32_ETH_MACA1HR[27] will disable MACA1L[31:24]; Setting R32_ETH_MACA1HR[26] will disable MACA1L[23:16]; Setting R32_ETH_MACA1HR[25] will disable MACA1L[15:8]; Setting R32_ETH_MACA1HR[24] will disable MACA1L[7:0].	0
[23:16] Reserved		RO is reserved.		0
[15:0] MACA1H[15:0]		The high 16 bits of the RW MAC address, i.e., bits 47 to 32.		0xFFFF

31.8.1.15 MAC Address Register 1, Lower 32 bits (R32_ETH_MACA1LR), Offset

Address: 0x4C

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
MACA1L[31:16]														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACA1L[15:0]															

Bit	name	access	describe	Reset value
[31:0] MACA1L[31:0]		The lower 32 bits of the RW MAC address, specifically bits 31 to 0.		0xFFFF FFFF

31.8.1.16 MAC Address Register 2 High 32 Bits (R32_ETH_MACA2HR)

Offset address: 0x50

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
AE	SA		MBC[5:0]								Reserved			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACA2H[15:0]															

Bit	name	access	describe	Reset value
31 AE		RW	Address filtering enable bit: 1: The address filter uses MAC address 2 for perfect filtering; 0: Address filter ignores MAC address 2.	0
30 SA		RW	Address role selection slot: 1: MAC address 2 is used to compare with the source address of the received frame; 0: MAC address 2 is used to match the destination address of the received frame. Compare.	0
[29:24] MBC[5:0]		RW	Mask control field. Each bit of the MBC corresponds to the mask MAC address. A specific byte in address 2 is used to prevent a specific word in MAC address 2 from being accessed. The section participates in the comparison of the source address or destination address of the received frame. Setting R32_ETH_MACA2HR[29] will disable MACA2H[15:8]; Setting R32_ETH_MACA2HR[28] will disable MACA2H[7:0]; Setting R32_ETH_MACA2HR[27] will disable MACA2L[31:24]; Setting R32_ETH_MACA2HR[26] will disable MACA2L[23:16]; Setting R32_ETH_MACA2HR[25] will disable MACA2L[15:8]; Setting R32_ETH_MACA2HR[24] will disable MACA2L[7:0].	0
[23:16] Reserved		RO is reserved.		0
[15:0] MACA2H[15:0]		The high 16 bits of the RW MAC address, i.e., bits 47 to 32.		0xFFFF

31.8.1.17 MAC Address Register 2, Lower 32 bits (R32_ETH_MACA2LR), Offset

Address: 0x54

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
MACA2L[31:16]														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACA2L[15:0]															

Bit	name	access	describe	Reset value
[31:0] MACA2L[31:0]		The lower 32 bits of the RW MAC address, specifically bits 31 to 0.		0xFFFF FFFF

31.8.1.18 MAC Address Register 3, High 32 Bits (R32_ETH_MACA3HR), Offset

Address: 0x58

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
AE	SA	MBC[5:0]									Reserved			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACA3H[15:0]															

Bit	name	access	describe	Reset value
31 AE		RW address filter enable bit:		0

			1: The address filter uses MAC address 3 for perfect filtering; 0: Address filter ignores MAC address 3.	
30 SA		RW	Address role selection slot: 1: MAC address 3 is used to compare with the source address of the received frame; 0: MAC address 3 is used to match the destination address of the received frame. Compare.	0
[29:24] MBC[5:0]		RW	Mask control field. Each bit of the MBC corresponds to the mask MAC address. A specific byte at address 3 is used to prevent a specific word at MAC address 3 from being accessed. The section participates in the comparison of the source address or destination address of the received frame. Setting R32_ETH_MACA3HR[29] will disable MACA3H[15:8]; Setting R32_ETH_MACA3HR[28] will disable MACA3H[7:0]; Setting R32_ETH_MACA3HR[27] will disable MACA2L[31:24]; Setting R32_ETH_MACA3HR[26] will disable MACA2L[23:16]; Setting R32_ETH_MACA3HR[25] will disable MACA3L[15:8]; Setting R32_ETH_MACA3HR[24] will disable MACA3L[7:0].	0
[23:16] Reserved		RO is reserved.		0
[15:0] MACA3H[15:0]		The high 16 bits of the RW MAC address, i.e., bits 47 to 32.		0xFFFF

31.8.1.19 MAC Address Register 3, Lower 32 Bits (R32_ETH_MACA3LR)

Offset address: 0x5C

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
MACA3L[31:16]														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACA3L[15:0]															

Bit	name	access	describe	Reset value
[31:0]	MACA3L[31:0]		The lower 32 bits of the RW MAC address, specifically bits 31 to 0.	0xFFFF FFFF

31.8.1.20 ETH_PHY Configuration Register (R32_ETH_PHY_CFGR)

Offset address: 0x80

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
PHY_RSTN	PHY_PD	Reserved												

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					DUPIEX	SPEED	LINK_STATUS	PHYADDR_EN	Reserved		REPHYADDR[4:0]				

Bit	name	Access Description	Reset value
31	PHY_RSTN	RW Ethernet PHY Global Reset Signal:	0

			1: Normal working mode; 0: Reset Ethernet PHY.	
30	PHY_PD	RW	In power-down mode, the MAC can access the PHY registers through the MDIO interface. Storage: 1: Power-off mode; 0: Normal working mode.	1
[29:11]	Reserved		RO is reserved.	0
10	DUPLEX		RO 1: Full-duplex connection; 0: Half-duplex connection.	0
9	SPEED		RO 1: Connection speed is 100 0: Connection speed is 10	0
8	LINK_STATUS		Mbps; RO 1: PHY connection Mbps. 0: Not connected.	0
7	PHYADDR_EN	RW	successful; Reconfigure PHY address: 1: bit 4-0 is used as the address of the PHY. 0: The default PHY address is 1.	0
[6:5]	Reserved		RO is reserved.	0
[4:0]	REPHYADDR[4:0]		When bit7 is 1 in RW, bits[4:0] are used as the address of the PHY.	0x1

31.8.2 Bit Fields of MMC Related Registers

31.8.2.1 MMC Control Register (R32_ETH_MMCCR) Offset

Address: 0x0100

31	30	29	28	27	26	25	24	23	22	2019	18	17	16
Reserved													

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

Bit	name	access	describe	Reset value
[31:4]	Reserved	RO	Reserved.	0
3	MCF	RW	Counter freeze control bit. Setting this bit will freeze all MMC counters. All values are frozen. Resetting this bit will resume the counting of each counter. (The last part, "Frozen," appears to be an error and is left untranslated.) Setting the ROR bit and then reading any counter will clear that counter. zero.	0
2	ROR	RW	Read-time reset control bit. Setting this bit will cause the system to reset after reading any counter. Reset the counter value.	0
1	CSR	RW	Counter stop bit. Setting this bit will cause the counter to increment to its maximum value. It stops after the value is set and does not automatically return to zero.	0
0	CR	RW1Z	Counter reset control bit. Setting this bit will reset all counters. The value will be automatically reset after one system cycle.	0

31.8.2.2 MMC Receive Interrupt Register (R32_ETH_MMCRIR)

Offset address: 0x0104

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
Reserved													RGUFS	Reserved

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

Bit name	access	describe	Reset value
[31:18] Reserved	RO is reserved.		0
17 RGUFS	RZ will be set when more than half of the frames have been received.		0
[16:6] Reserved	RO is reserved.		0
5 RFCES	RZ sets this bit when more than half of the frames received have CRC check errors.		0
[4:0] Reserved	RO is reserved.		0

31.8.2.3 MMC Transmit Interrupt Register (R32_ETH_MMCTIR)

Offset Address: 0x0108

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
Reserved												TGFS	Reserved	

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

Bit name	access	describe	Reset value
[31:22] Reserved	RO is reserved.		0
21 TGFS	This bit will be set when more than half of the frames have been sent by RZ.		0
[20:0] Reserved	RO is reserved.		0

31.8.2.4 MMC Receive Interrupt Mask Register (R32_ETH_MMCRIMR)

Offset address: 0x010C

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
Reserved													RGUFM	Reserved

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

Bit	name	access	describe	Reset value
[31:18] Reserved		RO Reserved.		0
17 RGUFM		RW	Interruption mask bit when more than half of the frames have been received. Setting this bit will mask the received frames. An interrupt is generated when the counter value reaches half.	0
[16:6] Reserved		RO is reserved.		0

5	RFCEM	RW	Receive CRC error frames more than half of the time, interrupt mask bit. Setting this bit will mask the connection. The middle frame is generated when the frame counter value for receiving CRC errors reaches half. Break.	0
[4:0] Reserved		RO is reserved.		0

31.8.2.5 MMC Transmit Interrupt Mask Register (R32_ETH_MMCTIMR) Offset

Address: 0x0110

31	30	29	28	27	26	25	24	23	22	2019	18	17	16
Reserved										TGFM	Reserved		

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

Bit	name	access	describe	Reset value
[31:22] Reserved		RO Reserved. Interruption		0
21	TGFM	RW	mask bit when more than half of good frames have been transmitted. Setting this bit will block the transmission of good frames. An interrupt is generated when the counter value reaches half.	0
[20:0] Reserved		RO is reserved.		0

31.8.2.6 MMC Good Frame Sending Counter after a Collision (R32_ETH_MMCTGFSCCR)

Offset address: 0x014C

31	30	29	28	27	26	25	24	23	22	2019	18	17	16
TGFSCCR[31:16]													

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

Bit	name	access	This	Reset value
[31:0]	TGFSCCR[31:0]	RO	field describes a domain that is successfully transmitted in half-duplex mode after a single collision. The number of frames, i.e., the counter for sending good frames after a single collision.	0

31.8.2.7 Good Frame Sending Counter (R32_ETH_MMCTGFMSCCR) **Offset Address after**

Multiple MMC Collisions : 0x0150

31	30	29	28	27	26	25	24	23	22	2019	18	17	16
TGFMSCCR[31:16]													

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

Bit	name	access	This	Reset value
[31:0]	TGFMSCCR[31:0]	RO	domain is described as being contained in half-duplex mode, and succeeds after one or more collisions. The number of frames sent, i.e., a counter for sending good frames after multiple collisions.	0

31.8.2.8 MMC Good Frame Count Register (R32_ETH_MMCTGFCR)

Offset Address: 0x0168

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
TGFC[31:16]														

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

Bit	name	access	describe	Reset value
[31:0]	TGFC[31:0]	RO	MAC count of the number of correct frames sent.	0

31.8.2.9 MMC Receive CRC Error Count Register (R32_ETH_MMCRFCECR)

Offset Address: 0x0194

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
RFCECR[31:16]														

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

Bit	name	Access description reset value	
[31:0]	RFCECR[31:0]	RO is a counter for frames received by the MAC that contain CRC check errors.	0

31.8.2.10 MMC Receive Alignment Error Count Register (R32_ETH_MMCRFAECR)

Offset address: 0x0198

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
RFAECR[31:16]														

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

Bit	name	access	describe	Reset value
[31:0]	RFAECR[31:0]	RO	MAC is a counter for frames with alignment errors received.	0

31.8.2.11 MMC Receive All Frame Count Register (R32_ETH_MMCRAFCR)

Offset address: 0x019C

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
CRAFCR[31:16]														

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

Bit	name	access	describe	Reset value
[31:0]	RAFCR[31:0]		The number of all frames received by the RO MAC, regardless of whether there are errors.	0

31.8.2.12 MMC Received Good Frame Count Register (R32_ETH_MMCRGUFCR)

Offset address: 0x01C4

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
RGUFCR[31:16]														

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

Bit	name	access	describe	Reset value
[31:0]	RGUFCR[31:0]		The number of normal frames received by RO MAC.	0

31.8.3 Bit Fields of IEEE 1588 (PTP) Related Registers

31.8.3.1 PTP Timestamp Control Register (R32_ETH_PTPTSCR) Offset

Address: 0x0700

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
Reserved														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										TSARU	TSITE	TSSTU	TSSTI	TSFCU	TSE

Bit	name	access	describe	Reset value
[31:6]	Reserved	RO	Reserved.	0
5	TSARU	RW	Addend register update control bit. Setting this bit will update the addend register. The value will be added to the accumulator. Use this bit in fine-tuning mode. Accumulation This bit is automatically cleared to zero after the incrementer completes. This bit can only be set when it is 0. Bit.	0
4	TSITE	RW	Timestamp interrupt trigger enable bit. When this bit is set, the PTP system will trigger an interrupt. An interrupt will be generated when the set target time register value is reached.	0
3	TSSTU	RW	System time update control bit. Setting this bit will update the PTP system time. Add the value to the update register. This bit will be automatically cleared after the update is complete.	0
2	TSSTI	RW	Timestamp initialization control bit. Setting this bit sets the PTP system time. This will be replaced with the value in the update register. After the update is complete, this bit.. Automatically reset.	0
1	TSFCU	RW	Update mode selection control bit: 1: Indicates the use of fine-tuning mode; 0: Indicates the use of coarse adjustment mode.	0
0	TSE	RW	Additional timestamp enable bit: 1. After sending or receiving, add a timestamp to the descriptor; 0: Do not add a timestamp to the descriptor.	0

31.8.3.2 PTP subsecond increment register (R32_ETH_PTPSSIR)

offset address: 0x0704

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
Reserved														

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

Bit name	access	describe	Reset value
[31:8] Reserved	RO is reserved.		0
[7:0] STSSI[7:0]	RW	Sub-second step value. In coarse tuning mode, PTP system per clock cycle. The timeout will automatically increment this value; in fine-tuning mode, it will increment when the accumulator overflows. When the time is displayed, the PTP system time will increment by this value.	0

31.8.3.3 PTP timestamp register high-order bit (R32_ETH_PTPTSHR)

offset address: 0x0708

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
STS[31:16]														

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

Bit name	Access description: RO PTP system time	Reset value
[31:0] TSHR[31:0]	value, real-time value, in seconds.	0

31.8.3.4 Offset address of the low byte of the PTP timestamp register

(R32_ETH_PTPTSLR) : 0x070C

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
STPNS	STSS[30:16]													

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

Bit	name	access	describe	Reset value
31	STPNS	RO	System time sign flag. 1 indicates the current time is negative, 0 indicates... The current time is positive.	0
[30:0]	STSS[30:0]	RO	PTP system time value, real-time value, in sub-seconds, approximately 0.46 seconds. Nanoseconds. When STSS overflows, STS increments by 1 second.	0

31.8.3.5 High 32 bits of the PTP timestamp update register (R32_ETH_PTPTSHUR)

Offset address: 0x0710

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16

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

Bit	name	access	describe	Reset value
[31:0]	TSUS[31:0]	RW	The timestamp updates the second value, which is used to replace the high digits of the PTP system time. This indicates the number of seconds added or subtracted from the system time.	0

31.8.3.6 Lower 32 bits of the PTP timestamp update register (R32_ETH_PTPTSLUR)

Offset address: 0x0714

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
TSUPNS	TSUSS[30:16]													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSUSS[30:0]															

Bit	name	access	describe	Reset value
31	TSUPNS	RW	Update the positive and negative flags of the timestamp. When the time update method is... When TSUS and TSUSS directly replace the PTP system time, TSUPNS should The value is 0; when the update time method is TSUS and TSUSS, it is performed on the original... When adding or subtracting time values in the PTP system, TSUPNS being 1 indicates that in PTP... Subtracting TSUS and TSUSS from the system time results in 0, indicating that... The PTP system time is based on the addition of TSUS and TSUSS.	0
[30:0]	TSUSS[30:0]	RW	The timestamp is updated to a sub-second value to replace the high-order bits of the PTP system time. Or it can be represented by the subsecond value added or subtracted from the system time.	0

31.8.3.7 PTP Timestamp Adder Register (R32_ETH_PTPTSAR) Offset

Address: 0x0718

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
TSA[31:16]														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSA[15:0]															

Bit	name	access		Reset value
[31:0]	TSA[31:0]	RW	Describes a timestamp plus a numerical value. This register is only accessed in fine-tuning mode. The value of TSA is added to the accumulator in each system cycle. If the accumulator overflows, the system time will be triggered in fine-tuning mode. renew.	0

31.8.3.8 High 32 bits of PTP Target Time Register (R32_ETH_PTPTTHR)

Offset address: 0x071C

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
TTSH[31:16]														

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

Bit	name	access		Reset value
[31:0]	TTSH[31:0]	RW	Describe the target time in seconds. If the PTP system time reaches or exceeds this value.. If a value is specified and the relevant interrupt is enabled, then an interrupt will be generated.	0

31.8.3.9 Lower 32 bits of the PTP target time register (R32_ETH_PTPTTLR)

Offset address: 0x0720

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
TTSL[31:16]														

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

Bit	name	access		Reset value
[31:0]	TTSL[31:0]	RW	Describes the target sub-second time value. If the PTP system time reaches or exceeds this.. If a value is specified and the relevant interrupt is enabled, then an interrupt will be generated.	0

31.8.4 Bits of DMA Control Related Registers

31.8.4.1 DMA Bus Mode Register (R32_ETH_DMABMR) Offset

Address: 0x1000

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
ERX_RST	ETX_RST	DMARX_RST	DMATX_RST	Reserved										

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

Bit	name	access	describe	Reset value
31	ERX_RST	Write 1 to RW to reset the MAC receiver module, and write 0 to undo the reset.		0
30	ETX_RST	Write 1 to RW to reset the MAC sending module, and write 0 to undo the reset.		0
29	DMARX_RST	Write 1 with RW to reset the DMA receiver module, and write 0 to cancel the reset.		0
28	DMATX_RST	Write 1 to reset the DMA transmit module, write 0 to undo the reset.		0
[27:8]	Reserved	RO is reserved.		0

7	DFM	RW	Descriptor pattern: 1: Enhanced descriptor mode; 0: Normal descriptor mode.	0
[6:2]	DSL[4:0]	RW	Descriptor jump length: These bits define the relationship between two descriptors that are not connected in a chain. The jump distance is measured in words (32 bits). An address jump refers to the jump distance from... The address difference between the end of the current descriptor and the beginning of the next descriptor Value. When the DSL domain is 0, in a ring structure, the DMA considers the description... The predicates are arranged consecutively.	0
1	Reserved	RO is reserved.		0
0	SR	RW	Software reset: When set to 1, the MAC's DMA controller resets all MAC subsystems. Internal registers and logic circuits. Different clock domains within the MAC. This bit is automatically cleared after the block completes the reset operation. It is then cleared during MAC rewriting. Before the register, you should ensure that the bit is 0.	1

31.8.4.2 DMA transmit query register (R32_ETH_DMATPDR) offset address:

0x1004

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
TPDR[31:16]														

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

Bit	name	access	describe	Reset value
[31:0]	TPDR[31:0]	RW	Send a query command. The user can write any value to this register to... Restart the suspended sending process. After restarting the sending process, this message... The register will be automatically cleared.	0

31.8.4.3 DMA Receive Queries Register (R32_ETH_DMARPDR) Offset

Address: 0x1008

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
RPDR[31:16]														

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

Bit	name	access	This	Reset value
[31:0]	RPDR[31:0]	RW	describes how to receive query commands. The receiving process may be interrupted by various unexpected events. The user needs to write any value to this register to restart the receiving process. After the receiving process is restarted, this register will be automatically cleared.	0

31.8.4.4 DMA Receive Descriptor Address Register (R32_ETH_DMARDLAR)

Offset address: 0x100C

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
RDLAR[31:16]														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDLAR[15:0]															

Bit	name	access	describe	Reset value
[31:0]	RDLAR[31:16]	RW	This register is used to store the location of the first receive DMA descriptor. Address. Note that the descriptor needs to be 16-byte aligned, so this register... The last 4 digits should be 0.	0

31.8.4.5 DMA Transmit Descriptor Address Register (R32_ETH_DMATDLAR)

Offset address: 0x1010

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
TDLAR[31:16]														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDLAR[15:0]															

Bit	name	access	This	Reset value
[31:0]	TDLAR[31:0]	RW	register is used to store the location of the first transmit DMA descriptor. Address. Note that the descriptor needs to be 16-byte aligned, so this register... The last 4 digits should be 0.	0

31.8.4.6 DMA Status Register (R32_ETH_DMASR)

Offset Address: 0x1014

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved	TSTS	PMTS	MMCS		Reserved	EBS[2:0]			TPS[2:0]		RPS[2:0]			NIS
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AIS	ERS	Reserved	PLS	ETS	RWTS	RPSS	RBUS	RS	TUS	ROS	TJTS	TBUS	TPSS	TS						

Bit	name	access	describe	Reset value
[31:30]	Reserved	RO	Reserved.	0
29	TSTS	RO	Timestamp trigger status flag. An interrupt event is generated in the PTP section. This bit will be set when PTP interrupt is enabled, and will generate an error if PTP interrupt is enabled. Interruption. This bit will automatically reappear after all flags in the PTP section are cleared. Clear.	0

28 PMTS	RO	PMT trigger status flag. When an interrupt event occurs in the PMT section, This bit will be set, and an interrupt will be generated if the PMT interrupt is enabled. This bit will be automatically cleared after all flags in the PMT section are cleared.	0
27 MMCS	RO	MMC trigger status flag. This flag is used when an interrupt event occurs in the MMC section. This bit will be set, and an interrupt will be generated if MMC interrupts are enabled. This bit will be automatically cleared after all flags in the MMC section are cleared.	0
26 Reserved	RO	Reserved.	0
[25:23] EBS[2:0]	RO	Error status field. This field indicates the type of bus error that caused the error. It is only effective when the DMASR[13] bit is set. DMASR[23]: 0: Error occurred while sending DMA forwarding data; 1: An error occurred while receiving DMA forwarded data. DMASR[24]: 0: Error occurred while reading and forwarding data; 1: An error occurred while writing data for forwarding. DMASR[25]: 0: Error accessing descriptor; 1: An error occurred while accessing the data cache.	0
[22:20] TPS[2:0]	RO	Sending process status field. This field is used to indicate the current sending DMA status. state. 000: Stop, received a reset or stop sending command; 001: Running, fetching send descriptor; 010: Running, waiting for status information; 011: Running, reading data from the send buffer and pushing it into the FIFO; 100, 101: Retained; 110: Pause, send descriptor unavailable or send buffered data. overflow; 111: Running, closing the send descriptor.	0
[19:17] RPS[2:0]	RO	Receive process status field. This field is used to indicate the current receive DMA status. state. 000: Stop, received a reset or stop sending command; 001: Running, retrieving receive descriptor; 010: Reserved; 011: Running, waiting to receive data packets; 100: Paused, receive descriptor unavailable; 101: Running, closing the receive descriptor; 110: Reserved; 111: Running, pushing received data from FIFO into memory.	0
16 NIS	RW1Z	Normal interrupt summary bits. Interrupts enabled in the DMAIER register. If any of the following bits are set, the NIS bit will also be set. -DMASR[0]: Transmit interrupt; -DMASR[2]: Send buffer unavailable; -DMASR[6]: Receive interrupt; -DMASR[14]: Early Receive Interrupt; -DMASR[11]: Internal physical layer connection state change. The NIS bit is a sticky bit; when it causes the corresponding interrupt state to be set to 1, it will trigger an interrupt. When the status bit is cleared to zero, the NIS bit must also be cleared by writing a 1. zero.	0

15 AIS		RW1Z	Interrupt summary bits. Interrupts enabled in the DMAIER register. If any of the following bits are set, the AIS bit will also be set. -DMASR[1]: Sending process stopped; -DMASR[3]: Jabber timeout; -DMASR[4]: Receive FIFO overflow; -DMASR[5]: Data underflow during transmission; -DMASR[7]: Receive buffer unavailable; -DMASR[8]: The receiving process is stopped; -DMASR[9]: Watchdog timeout; -DMASR[10]: Early transmission; -DMASR[13]: Bus error. The AIS bit is a sticky bit; when it causes the corresponding interrupt state to be set to 1, it will trigger an interrupt. When the status bit is cleared to zero, the AIS bit must also be cleared by writing a 1. zero.	0
14 ERS		RW1Z	Early Receive Status Bit. This bit is set to indicate that a data frame has been received. The DMA has filled the first buffer, but the complete frame has not yet been completed. Received. After RS is set, the ERS bit is automatically cleared.	0
[13:12] Reserved		RO	Reserved.	0
11 PLS		RW1Z	Internal physical layer connection status change flag. This bit indicates that the object... The connection to the layer is either connected or disconnected. Alternatively, write 1 to clear the connection.	0
10 ETS		RW1Z	Early transmission status bit. Setting this bit indicates that the entire transmission frame has been pushed onto the disk. FIFO.	0
9	RWTS	RW1Z	Watchdog timeout flag. Setting this flag indicates that the frame length has exceeded the limit. 2048 bytes.	0
8	RPSS	RW1Z	Receive process stopped status bit. This bit indicates that the receive process has stopped. stop.	0
7	RBUS	RW1Z	Receive buffer unavailable status bit. This bit indicates the receive descriptor. The CPU has the necessary permissions, DMA cannot acquire them, and the receiving process has been paused. The user needs to free the descriptor and fill the RPDR register with a value. The value is used to resume the receiving process. DMA will retry acquiring the value when receiving the next frame. Retrieve the descriptor.	0
6 RS		RW1Z	Receive completion status bit. This bit indicates that the reception of one frame has been completed. Upon receiving the frame, the frame information is also updated in the descriptor.	0
5	TUS	RW1Z	Data overflow bit. This bit is set to indicate that the data is buffered in the transmission frame. A data underflow occurred during transmission. At this point, the transmission process paused and the data was... According to the underflow error location.	0
4 ROS		RW1Z	Receive status overflow bit. Setting this bit indicates that a receive buffer overflow has occurred. Data overflow.	0
3	TJTS	RW1Z	Send Jabber timer timeout bit. This bit being set indicates... The sender is too busy and transmission has stopped; the descriptor is verbose. The (Jabber) timeout bit has been set.	0
2	TBUS	RW1Z	Send buffer unavailable status bit. Setting this bit indicates a send description. The symbol is occupied by the CPU, DMA cannot acquire it, and the sending process has been suspended. Bits [22:20] show the current sending status. The application needs... Release the send descriptor.	0

1	TPSS	RW1Z	Sending process stopped status bit. Setting this bit indicates that the sending process has stopped. It has been stopped.	0
0	TS	RW1Z	Transmission completion flag. This bit being set indicates that a frame has been completed. Send; ownership of the descriptor has been returned to the CPU.	0

31.8.4.7 DMA Operation Mode Register (R32_ETH_DMAOMR) Offset

Address: 0x1018

31	30	29 28 27	26 25 24 23 22	twenty one	2019	18	17 16
Reserved			DTCEFD	Reserved		TSF FTF	Reserved

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

Bit name	access	describe	Reset value
[31:27] Reserved	RO is reserved.		0
26 DTCEFD	RW	Control bits for TCP/IP checksum error frames are not discarded. 1: Detection of errors in the verification and existence of protocols such as TCP/IP/ICMP/UDP. If the timing is incorrect, do not discard the frame; 0: If the FEF bit is 0, the MAC will discard all frames with errors.	0
[25:22] Reserved	RO reserved.		0
21 TSF	RW	Send store-and-forward control bits: 1. The transmission process will only begin after the entire frame is written into the FIFO. send; 0: Once the data written to the FIFO during the sending process reaches a certain value, it will... Start sending.	0
20 FTF	RW clears the transmit FIFO control bit. Setting this bit will reset the transmit FIFO.		0
[19:14] Reserved	RO reserved.		0
13 ST	RW	Start or stop transmission control bits: 1: Set the sending process to running state; 0: After sending the current frame, the sending process enters a stopped state.	0
[12:8] Reserved	RO reserved.		0
7 FEF	RW	Forwarding error frame control bits: 1: Except for frames that are too short, all frames will be forwarded to the DMA; 0: The receive FIFO will discard frames with errors.	0
6 FUGF	RW	Forwarding frame control bits that are too short: 1: Receive FIFO forwarding frames that are too short; 0: Receive FIFO discards all frames shorter than 64 bytes.	0
[5:2] Reserved	RO reserved.		0
1 SR	RW	Start or stop receive control bit: 1: Start the receive process; DMA retrieves the receive descriptor from the current position or The receiver descriptor is retrieved from the identifier header position; 0: After forwarding the currently received frame, the receive DMA enters a stop state. The mode is such that the next transmission will begin from the current receive descriptor position.	0

0	Reserved	RO is reserved.	0

31.8.4.8 DMA Interrupt Enable Register (R32_ETH_DMAIER) Offset

Address: 0x101C

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
Reserved					NISE

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AISE	ERS	Reserved	PLE	ETIE	RWTIE	RPSIE	RBUIE	RIE	TUIE	ROIE	TJTIE	TBUIE	TPSIE	TIE					

Bit	name	access	describe	Reset value
[31:17]	Reserved	RO	Reserved.	0
16	NISE	RW	Normal interrupt enable bit. Enabling this bit will enable the following interrupts. -DMASR[0]: Transmit interrupt; -DMASR[2]: Send buffer unavailable; -DMASR[6]: Receive interrupt; -DMASR[14]: Early receive interrupt.	0
15	AISE	RW	Abnormal interrupt. Enabling this bit will enable the following interrupts. -DMASR[1]: Sending process stopped; -DMASR[3]: Jabber timeout; -DMASR[4]: Receive FIFO overflow; -DMASR[5]: Data underflow during transmission; -DMASR[7]: Receive buffer unavailable; -DMASR[8]: The receiving process is stopped; -DMASR[9]: Watchdog timeout; -DMASR[10]: Early transmission; -DMASR[13]: Bus error.	0
14	ERS	RW	Early receive interrupt enable bit. Enabling this bit will generate an early receive interrupt. Break. NISE must be set. AISE must be set.	0
[13:12]	Reserved	RO	is reserved.	0
11	PLE	RW	internal 100M physical layer connection status change interrupt enable bit.	0
10	ETIE	RW	Early transmit interrupt enable bit. Enabling this bit will generate an early transmit interrupt. Disconnect. AISE must be	0
9	RWTIE	RW	set. Receive watchdog interrupt enable bit. Enabling this bit enables receive interrupt. Watchdog timeout interrupt. AISE must be set.	0
8	RPSIE	RW	Receive process stop interrupt enable bit. Enabling this bit will generate a receive interrupt. The process was interrupted due to a halt.	0
7	RBUIE	RW	Receive buffer unavailable interrupt enable bit. Enabling this bit will generate an interrupt. Receive buffer unavailable interrupt. AISE must be set.	0
6	RIE	RW	Receive complete interrupt enable bit. Enabling this bit will generate a receive complete interrupt. An interrupt will occur. The NISE bit must	0
5	TUIE	RW	be set. This enables the transmit underflow interrupt. Enabling this bit will generate a transmit underflow. Interruption. AISE must be set.	0
4	ROIE	RW	receive overflow interrupt. Enabling this bit will generate a receive overflow interrupt.	0

			AISE must be set.	
3	TJTIE	RW	Enable the Jabber timeout interrupt. Simply enable this bit. To generate a transmit jabber timeout interrupt. AISE must be set. Transmit	0
2	TBUIE	RW	buffer unavailability interrupt enabled. Enabling this bit allows a transmit timeout to be generated. The buffer cannot be interrupted. The NISE bit must be	0
1	TPSIE	RW	set to enable the transmit process stop interrupt. Enabling this bit allows transmission to begin. The process is stopped and interrupted. AISE must be	0
0	TIE	RW	set. Transmission completion interrupt is enabled. Enabling this bit will generate a transmission completion signal. Interruption. NISE must be set.	0

31.8.4.9 DMA Lost Frame Register (R32_ETH_DMAMFBOCR) Offset

Address: 0x1020

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved												OFOC	MFA [10:0]	OMFC

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

Bit name	access	describe	Reset value
[31:29] Reserved	RO is reserved.		0
28 OFOC	RZ FIFO overflow bit of the overflow counter.		0
[27:17] MFA[10:0]	Frames lost by the RZ application.		0
16 OMFC	RZ indicates a lost frame counter overflow bit.		0
[15:0] MFC[15:0]	RZ	Lost frame counter. This field indicates a frame loss due to the receive buffer becoming unavailable. The number of frames lost as a result.	0

31.8.4.10 DMA Current Transmit Descriptor Register (R32_ETH_DMACHTDR)

Offset address: 0x1048

31	30	29	28	27	26	25	24	23	22	2019	18	17	16
HTDAR[31:16]													

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

Bit	name	access	describe	Reset value
[31:0]	HTDAR[31:0]	RO	This register value points to the currently used send descriptor. The registers are updated in real time by the DMA.	0

31.8.4.11 DMA Current Receive Descriptor Register (R32_ETH_DMACHRDR)

Offset address: 0x104C

31	30	29	28	27	26	25	24	23	22	2019	18	17	16
HRDAR[31:16]													

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

Bit	name	access	This	Reset value
[31:0]	HRDAR[31:0]	RO	register value points to the currently used receive descriptor. The registers are updated in real time by the DMA.	0

31.8.4.12 DMA Current Transmit Buffer Register (R32_ETH_DMACHTBAR)

Offset address: 0x1050

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
HTBAR[31:16]														

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

Bit	name	access	This	Reset value
[31:0]	HTBAR[31:0]	RO	register value points to the currently used transmit buffer. The registers are updated in real time by the DMA.	0

31.8.4.13 DMA Current Receive Buffer Register (R32_ETH_DMACHRBAR)

Offset address: 0x1054

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
HRBAR[31:16]														

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

Bit	name	access	This	Reset value
[31:0]	HRBAR[31:0]	RO	register value points to the currently used receive buffer. The registers are updated in real time by the DMA.	0

31.8.5 Internal 10M/100M Physical Layer Related Register

Addresses 31.8.5.1 Basic Control Register

(R16_BMCR) Offset

Address: 0x00	Bit	Name	access	describe	Reset value
	15	RST	RW/SC	1: PHY reset; 0: Normal operation.	0
	14	LOOPBACK	RW	1: Enable loopback; 0: Normal operation.	0
	13	SPEED_SELECTION	RW	1:100Mb/s; 0: 10Mb/s. After automatic negotiation, the negotiation speed status will be displayed.	1

12	AUTO_NEGOTIATION	RW	1: Enable automatic negotiation; 0: Automatic negotiation is prohibited.	1
11	POWER_DOWN	RW	1: Turn off the power; 0: Normal operation.	0
10	ISOLATE	RW	1. The MII/RMII interface is isolated from the PHY, but the PHY can still respond. MDC/MDIO; 0: Normal operation.	0
9	RESTART_AUTO_NEGOTIATION	RW/SC	1: Automatic renegotiation; 0: Normal operation.	0
8	DUPLEX_MODE	RW	1: Full-duplex; 0: Half-duplex.	1
7	COLLISION_TEST	RW	Crash test. 1: Conflict detection enabled; 0: Normal operation.	0
[6:0] Reserved		RO is reserved.		0

31.8.5.2 Basic Status Register (R16_BMSR) Address Offset:

0x01 Bit Name

		access		Reset value
15	100BASE_T4	RO	Description 1: Supports enabling 100BASE-T4; 0: 100BASE-T4 is not supported.	0
14	100BASE_TX_FULL_DUPLEX	RO	1. Supports enabling 100BASE-TX full-duplex; 0: 100BASE-TX full-duplex is not supported.	1
13	100BASE_TX_HALF_DUPLEX	RO	1. Supports enabling 100BASE-TX half-duplex; 0: 100BASE-TX half-duplex is not supported.	1
12	10BASE_T_FULL_DUPLEX	RO	1. Supports enabling 10BASE-T full-duplex; 0: 10BASE-T full-duplex is not supported.	1
11	10BASE_T_HALF_DUPLEX	RO	1. Supports enabling 10BASE-T half-duplex; 0: 10BASE-T half-duplex is not supported.	1
[10:7] Reserved		RO is reserved.		0
6	MF_PREAMBLE_SUPPRESSION	RO allows the reception of management frames with preamble suppression.		1
5	AUTO_NEGOTIATION_COMPLETE	RO	1: Automatic negotiation completed; 0: Autonegotiation not	0
4	REMOTE_FAULT	RO	completed. 1: Remote error detected (read clear). 0: No remote error condition detected.	0
3	AUTO_NEGOTIATION_ABILITY	RO	1. The PHY can perform self-negotiation; 0: PHY cannot perform auto-negotiation.	1
2	LINK_STATUS	RO	1: A valid link has been established; 0: No valid link established. 1:	0
1	JABBER_DETECT	RO	Jabber condition detected; 0: No Jabber condition detected.	0
0	EXTENDED_CAPABILITY	RO	1: Extended register functionality; 0: No extended register functionality.	1

31.8.5.3 Auto-negotiation Message Register (R16_ANAR)

Offset Address: 0x04

Bit	name	access	describe	Reset value
15	NP	RW	Next page: 1: Send protocol rule data page; 0: Send primary capability data page.	0
14	ACK	RO	1: Confirm receipt of link partner capability data words; 0: No acknowledgment signal received.	0
13	RF	RW	1: Notification of remote error detection capability; 0: Do not notify remote error detection capability.	0
12	Reserved	RO is reserved.		0
11	ASYPAUSE	RW	1. Supports asymmetric flow control; 0: Asymmetric flow control is not supported.	0
10	PAUSE	RW	1. Supports flow control; 0: Flow control is not supported.	0
9	100BASE_T4	RO	1. Local nodes support 100Base-T4; 0: Local nodes do not support 100Base-T4.	0
8	100BASE_TX_FD	RW	1. Local nodes support 100Base-TX full-duplex; 0: Local nodes do not support 100Base-TX full-duplex.	1
7	100BASE_TX	RW	1. Local nodes support 100Base-TX; 0: Local nodes do not support 100Base-TX.	1
6	10BASE_T_FD	RW	1. Local nodes support 10Base-TX full-duplex; 0: Local nodes do not support 10Base-TX full-duplex.	1
5	10BASE_T	RW	1. Local nodes support 10Base-T half-duplex; 0: Local nodes do not support 10Base-T half-duplex.	1
[4:0] SELECT[4:0]		RO	This node currently only supports the CSMA/CD 00001 binary code selector; no other protocols are supported.	1

31.8.5.4 Auto-negotiation Link Capability Register (R16_ANLPAR)

Offset address: 0x05

Bit	name	access	describe	Reset value
15 NP		RW	Next page. 1: Send the protocol details data page; 0: Send basic capability data page.	0
14 ACK		RO	1: Confirm receipt of the peer's auto-negotiation function register; 0: No acknowledgment signal received.	0
13 RF		RW 1: Notify remote error detection function;		0

			0: Do not notify the remote error detection function.	
12	Reserved	RO	is reserved.	0
11	ASYPAUSE	RO	1. Supports asymmetric pause via notification; 0: Asymmetric pause is not supported. When auto-negotiation is enabled, this bit reflects the capabilities of the linking party.	0
10	PAUSE	RO	1. Supports symmetrical pause upon notification; 0: Symmetrical pause is not supported.	0
9	100BASE_T4	RO	1. PHY supports 100BAES-T4; 0: PHY does not support 100BASE-T4.	0
8	100BASE_TX_FD	RO	1. The PHY supports 100BASE-TX full-duplex. 0: PHY does not support 100BASE-TX full-duplex.	0
7	100BASE_TX	RO	1. PHY supports 100BASE-TX; 0: PHY does not support 100BASE-TX.	0
6	10BASE_T_FD	RO	1. The PHY supports 10BASE-T full-duplex. 0: PHY does not support 10BASE-T full-duplex.	0
5	10BASE_T	RO	1. The PHY supports 10BASE-T; 0: PHY does not support 10BASE-T.	0
[4:0]	SELECT[4:0]	RO	PHY supports binary encoding selectors. Currently, only CSMA/CD 00001 is special; no other protocols support it.	0x01

31.8.5.5 Page Select Register (R16_PAGE_SEL) Offset

Address: 0x1F

Bit	name	access	describe	Reset value
[15:8]	Reserved	RO	is reserved.	0
[7:0]	SEL_PAGE_ADDR[7:0]	RW	Select page address: 00000000: 11111111.	0

31.8.6 PAGE0 Related Registers 31.8.6.1

PHY Status Register (R16_PHY_STATUS) Offset Address: 0x1A

Bit	name	access	describe	Reset value
15	INCOMPATIBLE_LINK	RO	After automatic negotiation and priority resolution, the capabilities of the local device and the peer device are compatible. Capacitive state: 1: Incompatible; 0: Compatible.	0
[14:7]	Reserved	RO	is reserved.	0
6	SIGNAL_STATUS	RO	100M signal status: 1: Signal status is good; 0: No signal.	0

5	MDI_MDIX_STATUS	RO	MDI/MDIX Status: 1: MDI; 0: MDIX.	1
4	POLARITY_STATUS	RO	Polarity state: 1: Polarity reversal; 0: Polarity is normal.	0
[3:0] PHY_SPEED_DUP[3:0]		RO	PHY operating speed and full/half duplex working mode.	0

PHY_SPEED_DUP Details Table			
3	PHY100_INDICATE	RO	RO operating speed is 100M.	0
2	PHY10_INDICATE	RO	runs at a speed of 10M.	0
1	FULL_INDICATE		Full-duplex.	0
0	HALFFULL_INDICATE	RO	Half-duplex.	0

31.8.6.2 Interrupt Indicator Register (R16_INTERRUPT_IND) Offset

Address: 0x1E

Bit	name	access	describe	Reset value
[15:12]	Reserved		RO is reserved.	0
11	LINKSTATUS_CHG		RZ link status change notification.	0
[10:0]	Reserved		RO is reserved.	0

31.8.7 PAGE 7 Related

Registers 31.8.7.1 Interrupt/LED Flag Register (R16_INTERRUPT_MASK)

PAGE7 Offset address: 0x13

Bit	name	access	describe	Reset value
[15:14]	Reserved		RO is reserved.	0
13	INT_LINKCHG		RW link change interruption MASK.	0
[12:10]	Reserved		RO is reserved.	0
9	LED_EN		RW LED function enabled.	0
[8:0]	Reserved		RO is reserved.	0x30

Chapter 32 SDIO Interface (SDIO)

The module descriptions in this chapter apply only to CH32H417 and CH32H415 microcontroller products.

The term "SDIO" as used in this chapter refers to a communication interface on a microcontroller designed for operating external storage cards such as SD cards or other devices.
The SDIO interface is a peripheral device of the microcontroller. The microcontroller's SDIO is directly connected to the HB bus, and its clock is directly provided by HCLK.
It can achieve high communication speeds. The SDIO of a microcontroller is used as an SDIO master, and the controlled devices are collectively referred to as SDIO devices. Applications
In general, SDIO is used to read and write SD cards, TF cards, or eMMC cards, or to control other devices that use SDIO as a communication interface, such as...
WiFi/4G module.

32.1 Main Features

32.1.1 Features ÿ Supports

SD cards, SDIO cards and MMC cards
Supports 1-bit, 4-bit, and 8-bit buses .
ÿ Maximum communication clock speed up to 100MHz
ÿ Compatible with MMC specification 4.5 (backward compatible)
Compatible with SD Card Specification 2.0 and SDIO Card Specification
2.0 . Not compatible with SPI or QSPI.

32.1.2 Overview

Microcontroller SDIO supports communication with memory cards such as SD cards or MMC cards. It's important to clarify that SDIO merely provides a set of physical...
The current SD card and MMC card specifications require specific clock, data, and command control timings for a single command transfer, as well as the sequential combination of commands.
The user determines this through the program. Furthermore, for various memory cards, SDIO only enables read and write functions; the file system provides...
The functionality for files needs to be implemented by the user through a program that builds the file system.

Unlike the QSPI interface, SDIO does not have a chip select pin and has an additional CMD pin, which can be considered a special data pin.
The data bus is specifically designed for transmitting commands and responses; SDIO offers three data bus widths: 1 bit, 4 bits, and 8 bits; the SDIO clock is configured during setup.
It typically operates at frequencies below 400kHz. When performing actual data transmission, it can be configured to the maximum clock frequency supported by the SDIO device.
The maximum SDIO clock output supported by the microcontroller is half of HCLK. According to the protocol, when the clock received by the SDIO device exceeds a certain threshold...
When the value is low, the peak values of the clock line, data line, and command line waveforms need to be reduced to save the time consumed by the waveform rise and fall.

Unlike SD cards, SDIO cards often refer to non-storage devices such as Wi-Fi/Bluetooth modules and 4G modules that use the SDIO interface. If
Unless otherwise specified, the content described in this chapter applies to SD cards. Content applicable only to SDIO cards or MMC cards will be specifically indicated.
In this chapter, SD cards are considered as potential targets first, followed by SDIO cards, and finally MMC cards.

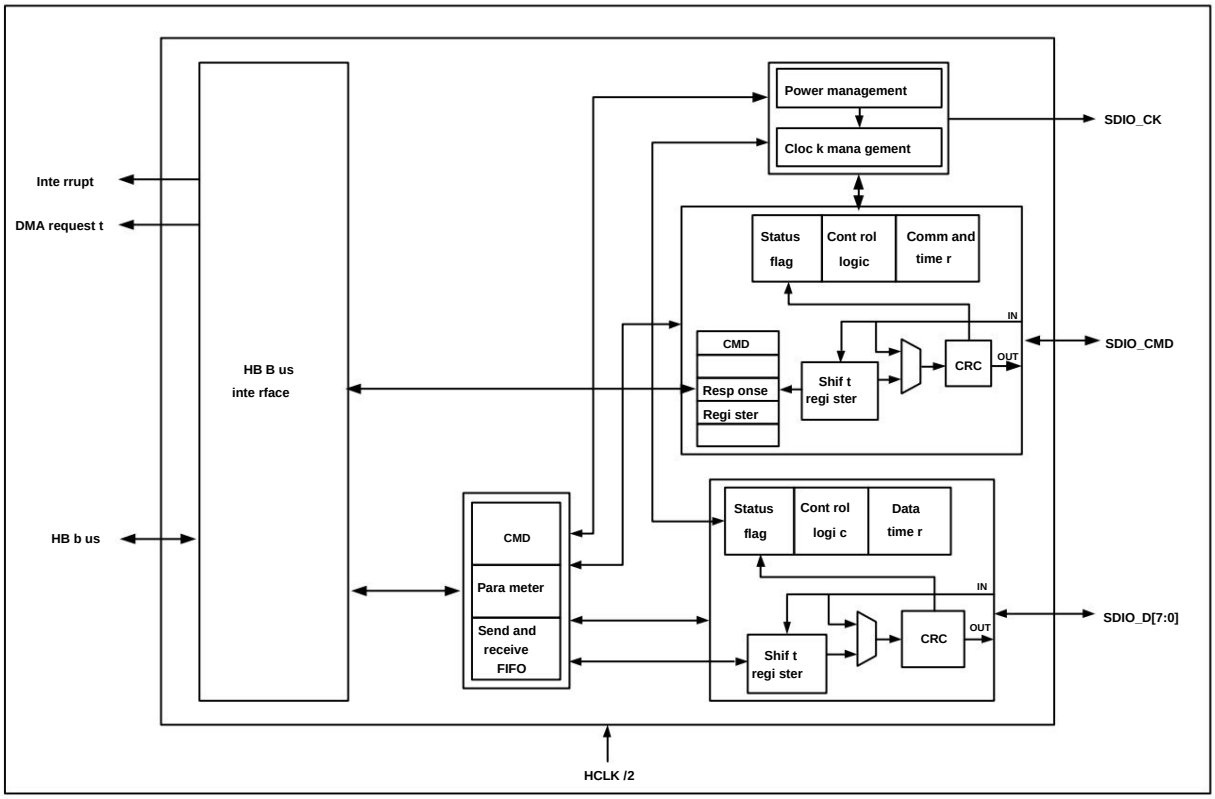
32.2 Interface and Clock

SDIO receives control from the CPU via the HB bus. SDIO's registers have a FIFO interface, through which the CPU or DMA reads data.
Write to the FIFO to acquire or send data. SDIO is directly clocked by HCLK and has an interrupt entry point, supporting multiple interrupt sources.
The SDIO pins that are directly controlled are SDIO_CK, SDIO_CMD, and SDIO_D[7:0], which are connected to the SDIO device through several pins.
SDIO is a host-driven communication interface, where all transmissions must be initiated by the microcontroller.

32.2.1 Peripheral Structure The

structure of SDIO is shown in Figure 32-1.

Figure 32-1 Structure diagram of SDIO



SDIO is a peripheral that allows a microcontroller to directly operate SDIO devices as a host. It generally consists of an HB interface, a clock control section, and a CMD line.

The SDIO consists of five modules: the control section, the data line control section, and the control register section. SDIO is a half-duplex peripheral; the CPU sends data to the control register.

The register is written with the commands and data to be sent. The command and data lines control the module to push the commands or data onto the I/O pins and add...

CRC is applied. The data flow of SDIO is handled by a general-purpose DMA, which processes the data transfer from RAM to the SDIO FIFO. The SDIO FIFO has 32 32-bit FIFOs.

The size of the bits.

32.2.2 Pins and their configuration

The pins that SDIO needs to configure and their modes are shown in Table 32-1.

Table 32-1 SDIO Pin Configuration

SDIO multiplexing function	Pin mode to be configured
SDIO_D[0:3]	Push-pull multiplexed output
SDIO_CK	Push-pull multiplexed output
SDIO_CMD	Push-pull multiplexed output
SDIO_D4	Push-pull multiplexed output
SDIO_D5	Push-pull multiplexed output
SDIO_D6	Push-pull multiplexed output
SDIO_D7	Push-pull multiplexed output

Note: In slave mode, the SDIO_CK pin should be configured as a floating input.

32.2.3 Clock

The clock pin for SDIO is SDIO_CK, and its output clock is obtained by dividing HCLK. The division factor can be configured to any value between 2 and 2^61.

Integer value. SDIO devices typically only support a single-bus mode with a maximum clock speed of 400kHz during initialization. After initialization, the host...

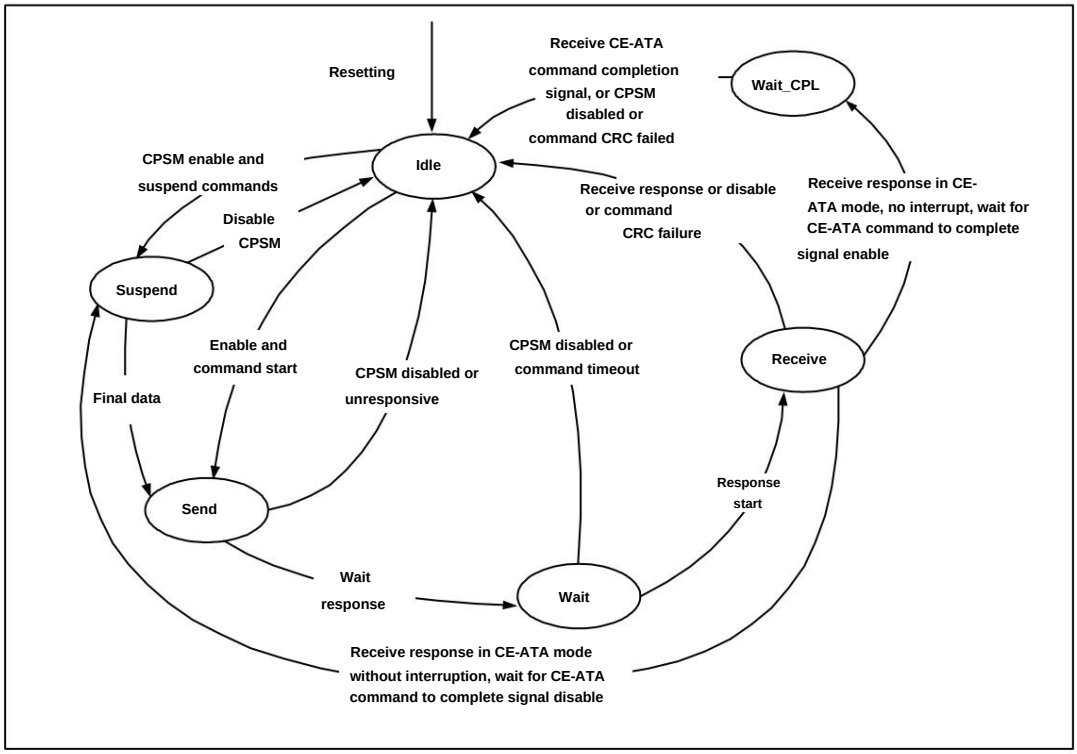
Typically, a switch to low voltage is initiated, while the clock is boosted to the maximum clock speed acceptable to both the microcontroller and the SDIO device.

Different versions and speed classes of SD cards support different clock speeds and switching processes, which users need to understand themselves.

32.2.4 Command State Machine

The command workflow of SDIO follows the state machine shown in the diagram below.

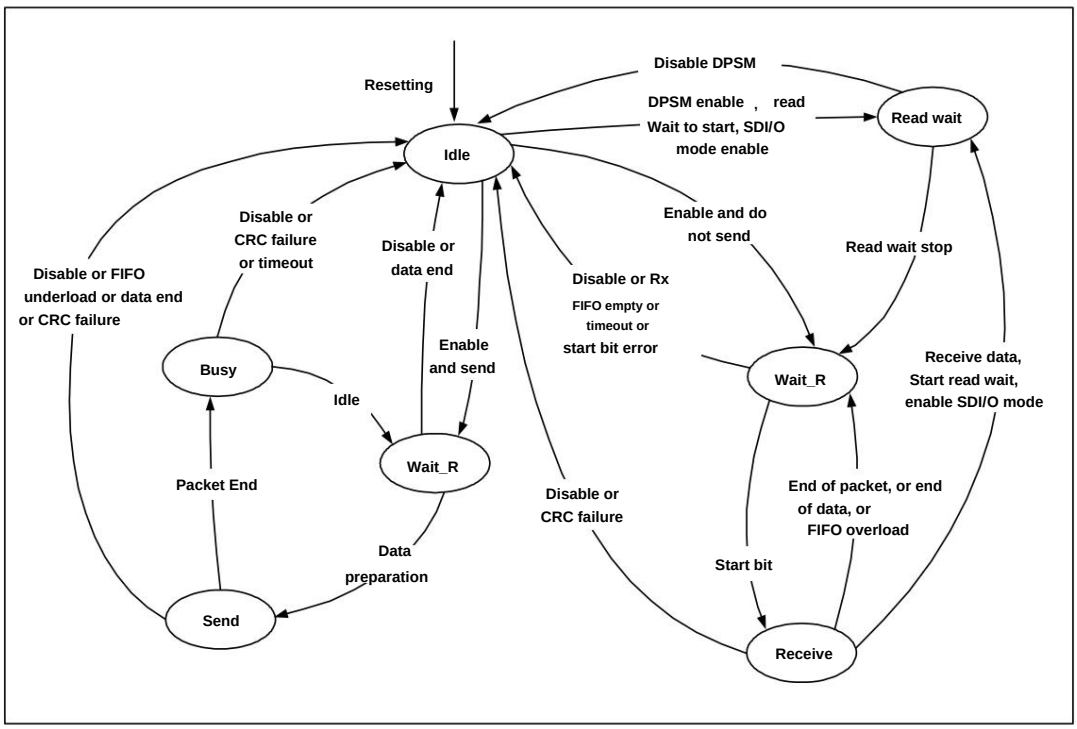
Figure 32-2 Command State Machine



32.2.5 Data State Machine

SDIO's data workflow follows the state machine shown in the diagram below.

Figure 32-3 Data State Machine



32.3 SDIO Protocol Overview

Communication on SDIO is based on a transmission as the smallest unit. Each transmission always begins with the host sending a command on the CMD line. Some commands... After sending, the SDIO device will also send a data message on the CMD line to reply to the host, called a "response," sometimes accompanied by additional data. Data transmission occurs on line D. The format of commands and responses is fixed; the definition of each bit in each field varies depending on the command or response. This should be determined accordingly. Responses and data transmissions must be issued or stopped within a specified time after the command or response ends; otherwise, a timeout error will occur.

The purpose of this section is to provide users with a basic understanding of the specifications necessary for using SDIO in a concise manner, and not to... Ensure comprehensiveness and timely updates. The microcontroller's SDIO only guarantees that it implements the SD 2.0, SDIO 2.0, and MMC 4.5 specifications. Basic component operations are supported, but features defined in higher-version specifications, such as dual-edge sampling, may not be supported. Users will need to consider these factors when developing specific applications. SDIO interactive programming should be based on the SD specification, SDIO specification and MMC specification.

32.3.1 Bus Timing: SD card transfers

are initiated by the host via CMD. The SD card may not respond, or it may respond with a short or long response. The response will be followed by data transmission. In SDIO card or MMC card communication, the SDIO device may also actively report an interrupt. The situation is illustrated in the following figure set.

Figure 32-4 No-response timing and no-data timing of SDIO (taking SD card as an example)

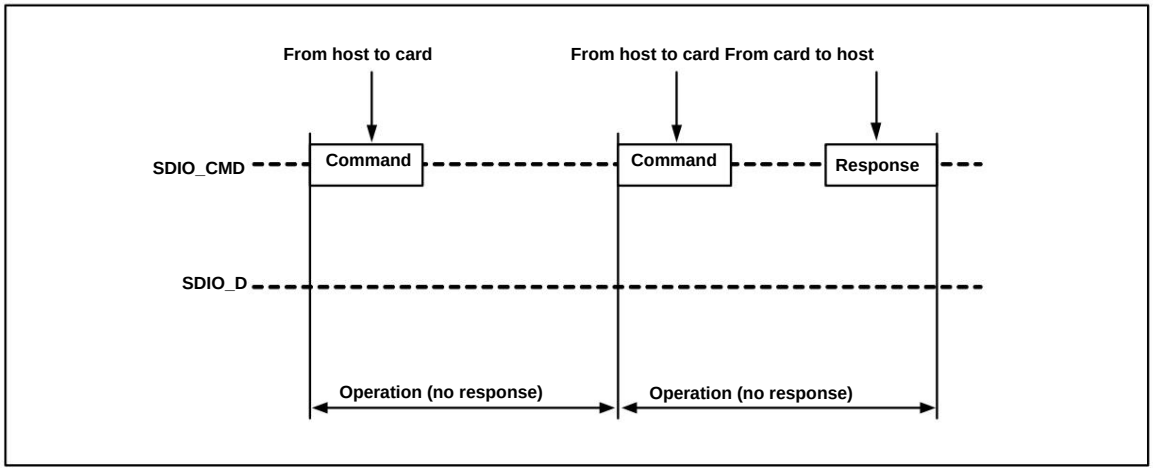


Figure 32-5 SDIO multi-block read timing (taking SD card as an example)

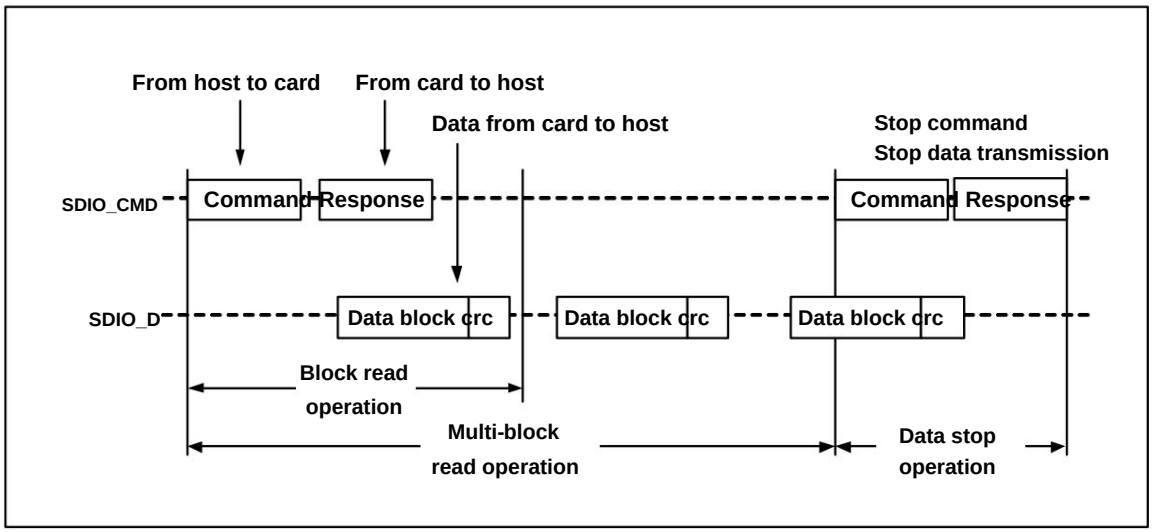


Figure 32-6 SDIO multi-block write timing (taking SD card as an example)

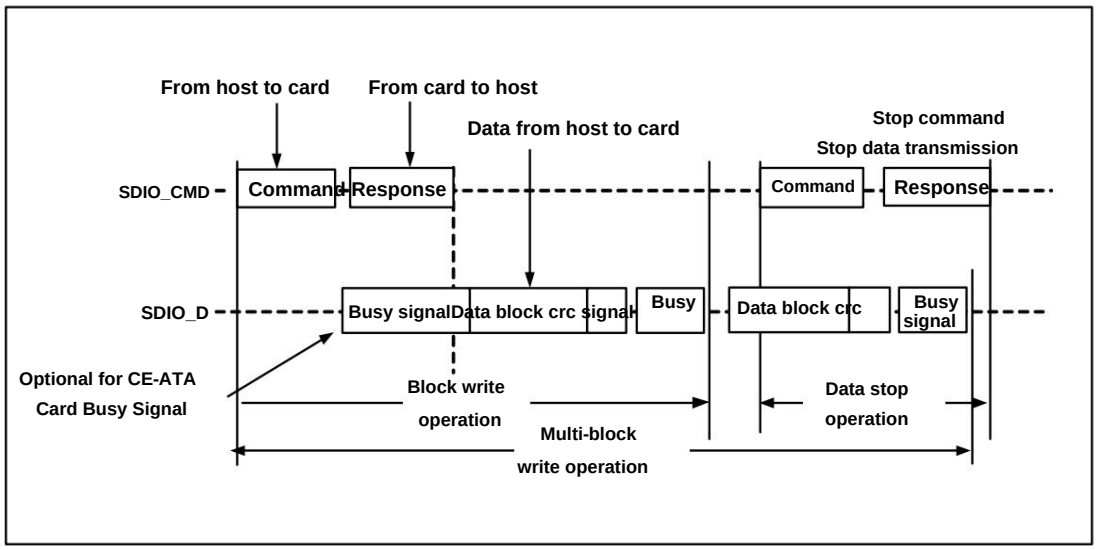


Figure 32-7 Data flow read timing of SDIO (taking SDIO card as an example)

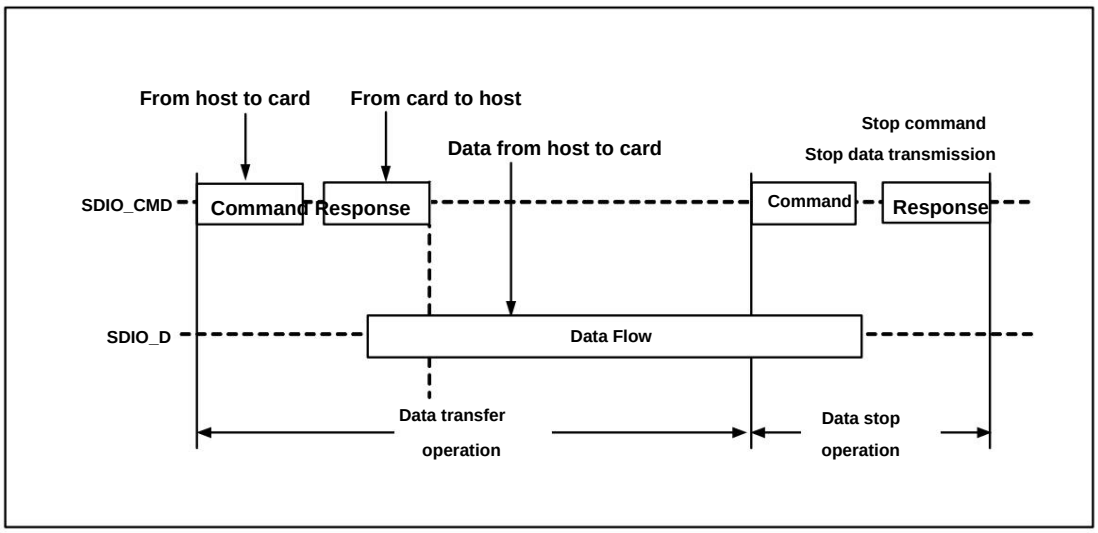
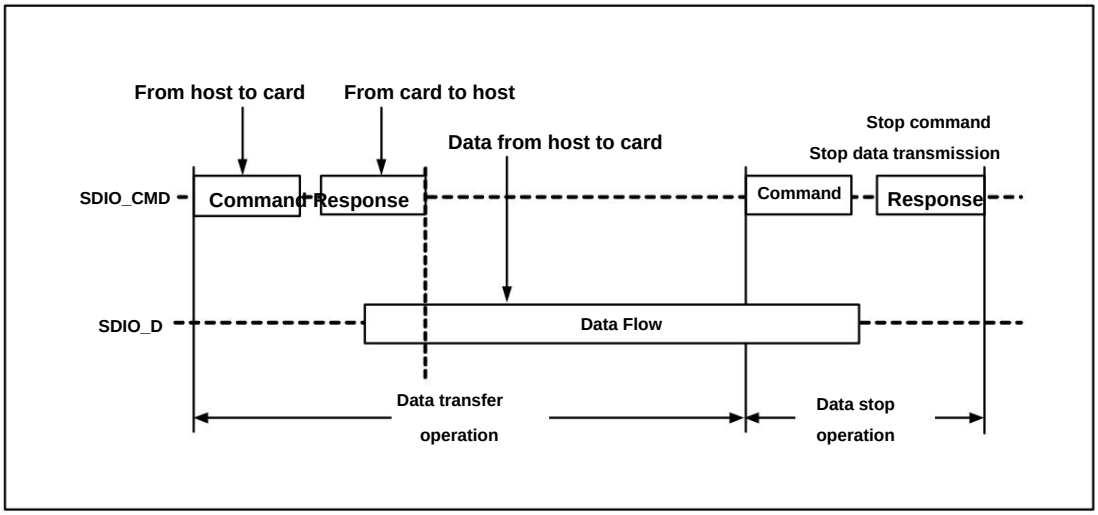


Figure 32-8 Data flow write timing of SDIO (taking SDIO card as an example)



32.3.2 Commands

Most SDIO transmissions begin with a command (CMD). The SDIO master uses commands to inform the device of its intent. The command format is as follows:

The table below.

Table 32-2 Command Format

Bit/Field Name Start Bit Transmission Bit Command Index				Command	CRC7 end bit	
Position/Width 47 values		46	[45:40]/6bits	parameters [39:8]/32bits	[7:1]/7bits	0
	0b	1b	X	X	X	1b

There are roughly four types of commands:

1) Broadcast command (bc): Sent to all cards on the bus; returns if no response is received.

2) Broadcast command with response (bcr): sent to all cards on the bus, and a response is returned;

3) Point-to-point command (ac): sent to a specific card, but no data is transmitted;

4) Point-to-point command with data transmission (ADTC): sent to a specific card, along with data transmission;

Here are some commonly used commands.

32.3.2.1 Basic Commands

Basic commands are some of the more basic functions supported by the SD card.

Table 32-3 Basic Commands for SD Cards

Command Index	Type	parameter	Reply type	abbreviation	describe
CMD0	The padding bits in bc [31:0] are meaningless; there is no			GO_IDLE_STATE.	Reset all cards to IDLE state.
CMD2 bcr	[31:0] padding bits are meaningless; R2			ALL_SEND_CID	All cards return CID.
CMD3 bcr	[31:0] has no meaningless padding; R6			SEND_RELATIVE_ADDR	Respond to the new RCA.
CMD7	ac	[31:16]RCA; The [15:0] padding is meaningless;	R1b Select card	SELECT/DESELECT_CARD	Select or deselect Select a card.
CMD8 bcr		[31:12] Retained; [11:8] Power supply; [7:0] Check mode;	R7	SEND_IF_COND	Send SD card interface bar Item.
CMD9	ac	[31:16]RCA; The [15:0] padding is meaningless;	R2	SEND_CSD	CSD required.
CMD10 ac		[31:16]RCA; The [15:0] padding is meaningless;	R2	SEND_CID	CID required.
CMD11	The fill bits in ac [31:0] are meaningless;		R1	VOLATGE_SWITCH	Switch to 1.8V power flat.
CMD12 ac	[31:0] padding bits are meaningless;		R1b	STOP_TRANSMISSION	Force card to stop transmission lose.
CMD13 ac		[31:16]RCA; [15] Send task status message Storage; [14:0] Pad bits;	R1	SEND_STATUS/SEND_TASK_STATUS	Send status or task Status register.
CMD15 ac		[31:16]RCA; The [15:0] padding is meaningless;	No	GO_INACTIVE_STATE	Require card to enter INACTIVE mode.

32.3.2.2 Erase Command

SD cards also use a FLASH memory structure, and the FLASH memory needs to be erased before writing. However, SD cards integrate the erasure logic internally, which is executed before writing.

If no erase operation is found before the command is executed, it will be automatically added. In many cases, especially before large-scale writes, if an erase operation is performed manually...

In addition to helping improve efficiency.

Table 32-4 SD Card Erase Commands [1]

Command index type parameter ac [31:0] Start address		Reply type	abbreviated description	
CMD32	for erasing [2] ac [31:0] End address for erasing	R1	ERASE_WR_BLK_START sets the starting address for erasure.	
CMD33	[2] ac [31:0] Erasing mode	R1	ERASE_WR_BLK_END sets the end address of the erase operation.	
CMD38		R1b	The ERASE parameter is 0, which means normal erasure.	

Note 1: The erase command here is the Specific defined in the protocol The erase command for the card differs from the erase command for the SDIO card. Card and MMC

2: The erase address for commonly used SDHC/SDXC (2GB to 2TB) level cards must be a block address, i.e., byte aligned. 512

32.3.2.3 Read Command for Block Transfer

Table 32-5 Block read commands for SD cards [1]

Command index type ac	parameter	Reply type	abbreviation	describe
CMD16	[31:0] block length	R1	SET_BLOCKLEN is the write block length, 512.	
CMD17 adtc [31:0] Address of a single block [1]		R1	READ_SINGLE_BLOCK sets the starting address for a single read.	
CMD18 adtc [31:0] Addresses of multiple blocks [1]		R1	READ_MULTIPLE_BLOCK sets the starting address for a single read.	
CMD19	adtc [31:0] Reserved bits	R1	SEND_TUNING_BLOCK	Sending a message indicating a change in mode 64-byte sequence
CMD20	ac [31:28] Reserved bits [27:0] Speed control bit ac	R1b	SPEED_CLASS_CONTROL speed control	
CMD22	[31:6] Reserved bit [5:0]Extended address	R1	ADDRESS_EXTENSION SDUC will be used.	
CMD23	ac [31:0]Block counter	R1	SET_BLOCK_COUNT block counter	

Note 1: The erase address for commonly used SDHC/SDXC (2GB to 2TB) level cards must be a block address, i.e., byte aligned. 512

32.3.2.4 Write command for block transfer

Table 32-6 Block Read Commands for SD Cards [1]

Command index type parameter ac [31:0] block		Reply type	abbreviation	describe
CMD16	length	R1	SET_BLOCKLEN	Write block length, 512
The address of block CMD24 adtc [31:0][1]		R1	WRITE_BLOCK	Configure single block write Starting address
CMD25 adtc [31:0] Addresses of multiple blocks [1]		R1	WRITE_MULTIPLE_BLOCK	Configure multiple block writes Starting address
CMD27 adtc [31:0] padding bits		R1	PROGRAM_CSD	Programmable for CSD Word programming

Note 1: Currently commonly used SDHC/SDXC (2GB) i.e., up to For cards with a capacity of 2TB, the erase address must be a block address, 512 Byte alignment.

32.3.3 Response The

response, as a necessary reply from the SDIO device to the host, is also transmitted over the CMD line and must be returned within the specified time.

The response is transmitted with the most significant bit first and the least significant bit last. The response length and the definition of each bit and field are specifically determined by the response type, but all...

Each response begins with a start bit that is fixed at 0, followed by a transmission direction bit that is fixed at 0 [1]. All responses end with a...

There are 7 stop bits, fixed at 1 [1]. SD cards support R1/R1b/R2/R3/R6/R7, and SDIO cards also support R4/R5.

The formula is shown below.

32.3.3.1 R1 Response: Normal

response, 48 bits in total length, with CRC7 checksum, and a 32-bit card status field. The format is shown in the table below.

Table 32-7 R1 Format

Bit/Field Name Start Bit Transmit Bit Response Index				Card	CRC7 end bit	
Position/Width 47 values		46	[45:40]/6bits	status [39:8]/32bits	[7:1]/7bits	0
	0b	0b Follows CMD index		X	X	1b

32.3.3.2 R1b Response The

format of R1b is the same as R1, but a busy signal can be added after the response, i.e., clamping the data line D2. The host receives the busy signal.

After detecting that SDIO_D2 is low, appropriate processing is required.

32.3.3.3 R2 Response: A

response to several specific commands, totaling 136 bits in length. CRC7 checksum is included in the card status field, which is 128 bits. Card Status

The state field stores the value of the CID register or the CSD register. The CID register is generally used as the response to CMD2/CMD10, while the CSD register is generally used as...

This is a response to CMD9. For the specific meanings of the CID/CSD registers, please refer to section 28.4.2, Device Registers. Note that R2 will only respond to...

In the [127:1] segment of the CID/CSD register, the reserved bit [2] of CID/CSD[0] is fixed at 1, which is occupied by the end bit, which is also fixed at 1.

The format is shown in the table below.

Table 32-8 R2 Format

Bit/Field Name Start Bit Transmission Bit		Command Index	Card status	End position
Number/Width 135 Value	134	[133:128]/6 bits	[127:1]/127 bits	0
0b	0b	111111b	CID/CSD	1b

32.3.3.4 R3 Response : A

dedicated response to the OCR register, 48 bits in total, without CRC7 checksum. The OCR register is generally used as a response to ACMD41.

The format is shown in the table below.

Table 32-9 Format of R3

Bit/Field Name Start Bit Transmission Bit Command Index			Card status	Preserve the end bit	
Position/Width 47 values	46	[45:40]/6 bits	[39:8]/32 bits [7:1]/7 bits		0
0b	0b	111111b	OCR	1111111b	1b

32.3.3.5 R4 Response : A

dedicated response to CMD5, including OCR and related registers. It is 48 bits long and has a CRC7 checksum. R4 is used in SDIO.

The format of the card is as shown in the table below.

Table 32-10 R4 Format

Bit/field name bit width (unit: bit)			value
Start bit,	47	1	0b
transfer	46	1	0b
bit,	[45:40]	6	111111b
reserved card ready	39	1	X
Number of I/O functions [38:36]		3	X
Current register	[35]	1	X
fill bits	[34:33]	2	00b
S18A	32	1	X

IO OCR	[31:8]	twenty four	OCR
End bit of CRC	[7:1]	7	1111111b
check field	0	1	1b

Note: MMC cards have a different format than regular cards.R4 SDIO

32.3.3.6 R5 Response : A

dedicated response to CMD5, 48 bits in length, with CRC7 checksum. R5 is used in SDIO cards, and its format is shown in the table below.

Table 32-11 R5 Format

Bit/Field Name Start Bit Transmit Bit Command Index Padding Bit Response Format Read/Write Data CRC7 End Bit								
Position 47 Width		46	[45:40]	[39:24]	[23:16]	[15:8]	[7:1]	0
	1	1	6	16	8	8	7	1
value	0b	0b	110100b	0000h	X	X	X	1b

Note: MMC cards have a different format than regular cards.R5 SDIO

32.3.3.7 R6 Response : A

dedicated response to RCA, 48 bits in length, with CRC7 checksum, formatted as shown in the table below.

Table 32-12 R6 Format

Bit/domain name start bit transfer bit command index card RCA [45:40] [39:24]					Card status bit CRC7 end bit		
Rank	47	46			[23:8]	[7:1]	0
Width	1	1	6	16	16	7	1
value	0b	0b	000011b	X	X	X	1b

32.3.3.8 R7 Response is a

dedicated response to CMD8, indicating the supported voltage information. It is 48 bits long and has CRC7 checksum. The format is shown in the table below.

Table 32-13 R7 Format

bit/field name	Start Bit	transmission Bit	Command Line lead	Reserved bits	PCIe 1V2 support	PCIe response answer	Received electricity Pressure	Check back Feed	CRC7	Finish Bit
Position 47		46 [45:40] [39:22]			twenty one	20	[19:16]	[15:8] [7:1]		0
Width 1		1	6	18	1	1	4	8	7	1
Value 0b		0b 001000b 00000h			X	X	X	X	X	1b

32.3.4 Data Transmission

Data transmission occurs on the SDIO_D data line, which has three widths: 1, 4, and 8 bits. Data transmission typically begins with the start bit of one clock cycle.

At the very beginning, the most significant byte of each byte is listed first, followed by the least significant byte. For the block transfer mode supported only by SD cards, each block of data transfer...

A CRC check is performed after the process ends.

Enable the RANDOM_LEN_EN bit, configure the byte length represented by DBLOCKSIZE2, and configure the number of transfers represented by R32_SDIO_DLEN.

Based on the total length, enable the DTEN bit and fill the data into R32_SDIO_FIFO to complete the block transfer of arbitrary byte length with CRC checksum.

lose.

MMC cards also support a data stream transmission mode, in which CRC is not included. The following diagram shows the data transmission format.

Figure 32-9 Format of data transmission bytes on an SD card single bus

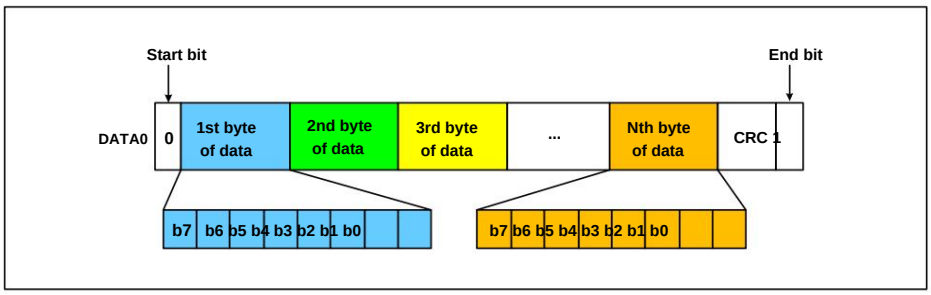


Figure 32-10 Format of data transfer bytes on SD card 4-bus

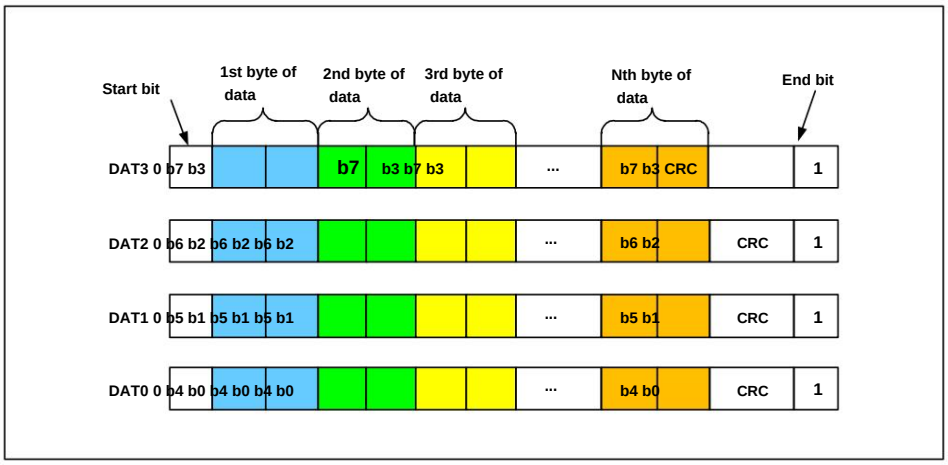


Figure 32-11 Format of SD card single-bus data transmission of a 512-bit word

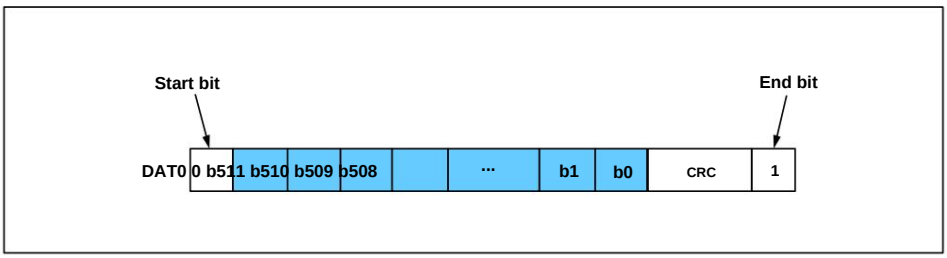
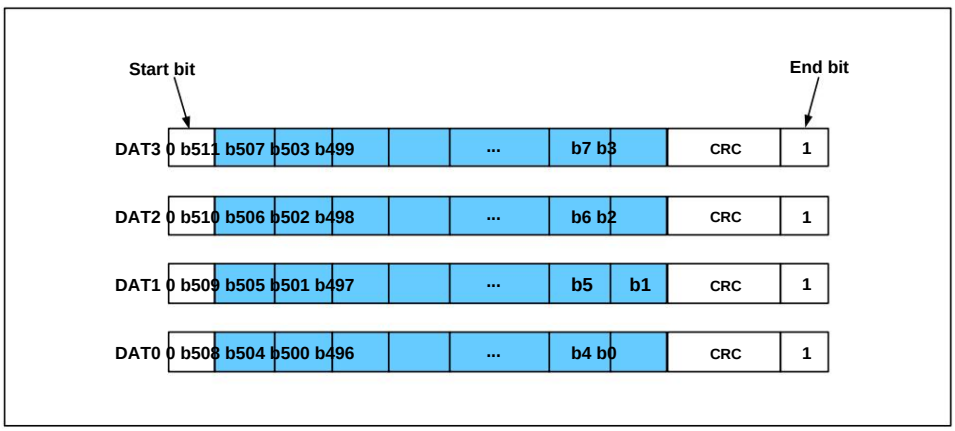


Figure 32-12 SD Card 4-bus data transmission format of a 512-bit word



32.3.5 Enabling the

SLV_MODE bit in Slave mode means the system is in a state of waiting for host commands. Received command parameters will be placed in RESP1.

The index will be placed in RESP2, and an R1 type response will be automatically sent to the host, with the response parameter being the value of CMDARG.

Data transmission and reception in slave mode reference master mode, but the slave can use the SLV_FORCE_ERR bit to force a CRC error in the data block, thus indicating...

This indicates unexpected data read/write operations.

Note: The slave device only supports type-based responses. The command index and parameter meanings used in master-slave communication are defined by the software.R1

32.4 Application

32.4.1 Device Initialization and Device Registers

32.4.1.1 OCR Register The

Operation Conditions Register stores information about the SD card's received power supply voltage.

Information and related status bits. The definitions of the related bits are shown in the table below.

Table 32-14 OCR Register Bit Definitions

Rank	Bit definition	describe
[0:3]	reserved	Supported VDD33 voltage range, in volts
4	reserve	
5	reserve	
6	reserve	
7 is reserved for low voltage range		
8	reserve	
9	reserve	
10	reserve	
11	reserve	
12	reserve	
13	reserve	
14	reserve	
15	2.7-2.8	
16	2.8-2.9	
17	2.9-3.0	
18	3.0-3.1	
19	3.1-3.2	
20	3.2-3.3	
twenty one	3.3-3.4	
twenty two	3.4-3.5	
twenty three	3.5-3.6	
24 Accept switch to 1.8V Reserved		
[25:26]		
27. Supports status bits for capacities exceeding 2TB.		
28	reserve	
29	UHS-II Card Status	This bit being set indicates that this card supports the UHS-II interface.
30	Bits: Card Capacity	This bit being set indicates that the card capacity is greater than 2GB.
Status (CCS) 31. Card Power-On Status Bit (busy): This bit is set after the card is powered on. Other bits only become meaningful after power-on.		

32.4.1.2 CID Register The CID

register stores some identification information.

Table 32-15 Definitions of each bit and field in the CID register

Bit/Field Name Abbreviation Width in bits			
	MID	8	[127:120]
	OID 16 [119:104]		
Manufacturer ID, Application ID.			[103:64]
Product Name (PNM 40),		8	[63:56]
Product Version (PRV), Product Serial			[55:24]
Number (PSN 32), Reserved			[23:20]
(None), Production Date (MDT 12)			[19:8]
CRC7	CRC	7	[7:1] [0]
Fixed bit, fixed at 1. No 1.			

32.4.1.3 CSD Register The CSD

register stores the characteristic data of the SD card. Taking the most commonly used SDHC and SDXC cards, specifically the second version of the CSD, as an example...

For example, the definitions of each field are shown in the table below.

Table 32-16 Meaning of each field in the CSD register

name	abbreviation	width	value	Reading and writing	Rank
CSD version	CSD_STRUCTURE	2	01b	RO	[127:126]
retains	None	6	00_0000b	RO	[125120
read access between	TAAC	8	0Eh	RO	[119:112]
Using clock cycles The readings are represented Access time	NSAC	8	00h	RO	[111:104]
Maximum data release Send speed	TRAN_SPEED	8	32h5Ah0Bh2Bh	RO	[103:96]
Card command	CCC	12	X1X1101101X1b	RO	[95:84]
class read data block most Large length	READ_BL_LEN	4	9	RO	[83:80]
Allowed block portion read	READ_BL_PARTIAL	1	0	RO	[79]
Block write misalignment	WRITE_BLK_MISALIGN	1	0	RO	[78]
Block read misalignment	READ_BLK_MISALIGN	1	0	RO	[77]
Executed DSR	DSR_IMP	1	X	RO	[76]
reserves no device size		6	00_0000b	RO	[75:70]
	C_SIZE	twenty two	XXXXXXh	RO	[69:48]
enables single-block erase;		1	0	RO	[47]
ERASE_BLK_EN enables erase sector size;		1	1	RO	[46]
SECTOR_SIZE enables write protection group.		7	7Fh	RO	[45:39]
Small	WP_GRP_SIZE	7	0000000b	RO	[38:32]
Write-protected group enables	WP_GRP_ENABLE	1	0	RO	[31]

able					
Retain	none	2	00b	RO	[30:29]
write speed factor R2W_FACTOR Maximum write		3	010b	RO	[28:26]
data Block length	WRITE_BL_LEN	4	9	RO	[25:22]
Allowed block portion Write	WRITE_BL_PARTIAL	1	0	RO	[twenty one]
Preserving file format group		5	00000b	RO	[20:16] [15]
FILE_FORMAT_GRP, copy flag TMP_WRITE_PROTECT,		1	0	RO	[14]
permanent write protection PERM_WRITE_PROTECT,		1	X	RW OTP	[13]
temporary write protection TMP_WRITE_PROTECT, file		1	X	RW OTP	[12]
format preserved.		1	X	RW	[11:10]
	FILE_FOMAT	2	00b	RO	[9:8] [7:1]
	None	2	00b	RO	
CRC	CRC	7	0000000b	RW	
Unused, must 1 must be used	none	1	1b	RO	[0]

32.4.1.4 The RCA register relative

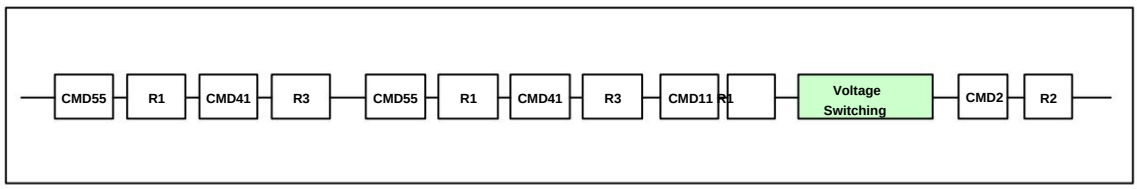
to the card address register stores the card address, which is 16 bits long and has a default value of 0.

32.4.2 Voltage Switching

During the later stages of SD card initialization, interface level switching is required to switch the I/O levels of the SD card's clock, data, and command lines.
Up to 1.8V. For devices with insufficient slew rate, using a lower voltage level can help increase the frequency. However, the SD supply...
The electrical voltage does not necessarily change; low-voltage SD cards only appeared in newer versions of the protocol.

The steps for switching voltage are shown in the following diagram.

Figure 32-13 Voltage switching sequence



32.4.3 During clock switching

initialization, the SD card clock is only 400kHz. After the voltage switching is complete, the clock can be boosted to a higher level, for example...
The SDHC card in UHS-I mode, at its lowest speed, can reach a bus clock speed of 80MHz. However, given the microcontroller's I/O output capabilities, the clock speed should be adjusted accordingly.
Limited to 50MHz.

32.5 Interruption

32.5.1 SDIO Interrupts SDIO

supports multiple interrupt sources. As shown in the interrupt enable register (R32_SDIO_IER), there are 24 situations that can trigger an interrupt.

Users may enable this feature at their own discretion.

32.5.2 SDIO Device Interrupt

It should be noted that not only SDIO peripherals can report interrupts to the CPU, but external SDIO cards and MMC cards can also report interrupts to SDIO peripherals.

In 4-bit bus mode, the interrupt line is D1; in 8-bit bus mode, the interrupt line is D7, active low. If in idle state...

If SDIO detects that D1 or D7 is low, it should read the status register or interrupt flag register of the SDIO device and respond to the interrupt in a timely manner.

The CPU can determine whether the SDIO master has received an interrupt by checking the SDIOIT bit in the R32_SDIO_STA register.

32.6 Register Description

Table 32-17 List of SDIO Related Registers

name	Access address	describe	Reset value
R32_SDIO_POWER	0x40018000	Power Register;	0x00000000
R32_SDIO_CLKCR	0x40018004	Clock Register;	0x00000000
R32_SDIO_ARG	0x40018008	Command Parameter Register;	0x00000000
R32_SDIO_CMD	0x4001800C	Command Register;	0x00000000
R32_SDIO_RESPCMD	0x40018010	Response Register;	0x00000000
R128_SDIO_RESPX	0x40018014	Response Parameter Register;	0x00000000
R32_SDIO_DTIMER	0x40018024	Data Timing Register;	0x00000000
R32_SDIO_DLEN	0x40018028	Transfer Length Register;	0x00000000
R32_SDIO_DCTLR	0x4001802C	Data Control Register;	0x00000000
R32_SDIO_DCOUNT	0x40018030	Transfer Count Register;	0x00000000
R32_SDIO_STA	0x40018034	Status Register;	0x00000000
R32_SDIO_ICR	0x40018038	Interrupt Clear Register;	0x00000000
R32_SDIO_MASK	0x4001803C	Interrupt Enable Register	0x00000000
R32_SDIO_FIFOCNT	0x40018048	FIFO counter	0x00000000
R32_SDIO_DCTRL2	0x40018060	Data control register 2	0x00000000
R32_SDIO_FIFO	0x40018080	FIFO register	0x00000000

32.6.1 Power Register (R32_SDIO_POWER)

Offset address: 0x00

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

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

Bitname	access	describe	Reset value
[31:2] Reserved	RO	reserved.	0
[1:0] PWRCTRL[1:0]	RW	Power detection bit: 00: Power off, clock stops; 01: Reserved; 10: Retained power-on state; 11: Power-on state, card clock is on.	0

32.6.2 Clock Register (R32_SDIO_CLKCR)

Offset address: 0x04

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

15 14 13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	HWFC_EN	NEGEDGE	WIDBUS[1:0]		BYPASS	PWRSAV	CLKEN	CLKDIV[7:0]					

Bit	name	access	describe	Reset value
[31:15]	Reserved	RO	Reserved.	0
14	HWFC_EN	RW	Hardware flow control enable bit; when this bit is set, TXFIFOE and RXFIFOE are enabled. The signal only works then. 1: Enable hardware flow control; 0: Disable hardware flow control.	0
13	NEGEDGE	RW	SDIO_CK phase selection bit: 1: SDIO_CK is generated on the falling edge of HLCK; 0: SDIO_CK is generated on the rising edge of HCLK.	0
[12:11]	WIDBUS[1:0]	RW	Bus width configuration field: 00: 1-bit bus mode, using SDIO_D0; 01: 4-bit bus mode, using SDIO_D[3:0]; 10: 8-bit bus mode, using SDIO_D[7:0]; 11: Not used.	0
10	BYPASS	RW	Clock bypass enable bit: 1: SDIO_CK is directly connected to HCLK; 0: SDIO_CK is obtained by frequency division using a frequency divider. Note: The HWFC_EN bit must be enabled when using this bit.	0
9	PWRSAV	RW	Idle clock state configuration bit. When this bit is set, the bus idles... Disable the SDIO_CK output to save power. 1: SDIO_CK is only output when needed; 0: SDIO_CK is always output.	0
8	CLKEN	RW	Clock enable bit: 1: SDIO_CK is allowed to output; 0: SDIO_CK is disabled for output.	0
[7:0]	CLKDIV[7:0]	RW	Clock divider field; this field represents the relationship between SDIO_CK and HCLK. Tie. SDIO_CK = HCLK/(CLKDIV+2). Note that during the initialization phase, SDIO_CK should be below 400kHz.	0

Note: (1) The clock configuration register is used to control the parameters related to SDIO_CK. It should be noted that this register is used during data reading and writing.

7 HCLK

(1) Cannot be changed within the cycle; (2) No clock register needs to be configured when the slave is in use.

32.6.3 Command Parameter Register (R32_SDIO_ARG)

Offset Address: 0x08

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

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

Bit	name	access	describe	Reset value
[31:0]	CMDARG	RW	Host mode: The command's parameter field. This field stores the parameters in the command. The parameters will be sent to the CMD line as part of the command. Slave mode: 32-bit response parameter register for slave device reply.	0

32.6.4 Command Register (R32_SDIO_CMD) Offset

Address: 0x0C

31	30	29	28	27	26	25	24	23	22	twenty one	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	ATACMD	NIEN	ENCMDCompl	SDIOSuspend	CPSMEN	WAITPEND	WAITINT	WAITRESP[1:0]		CMDINDEX[5:0]					

Bit	name	access	describe	Reset value
[31:15]	Reserved	RO	is reserved.	0
14	ATACMD	RW	executes CE-ATA commands. If this bit is set, CPSM will jump to CMD61.	0
13	NIEN	RW	disables the CE-ATA interrupt setting bit. If this bit is set, CE-ATA... No interruption will occur.	0
12	ENCMDCompl	RW	Enables the CMD completion signal. If this bit is set, the command completes. A signal will be generated.	0
11	SDIOSuspend	RW	Pause command sent bit. If this bit is set, a pause command will be sent. Number. Applicable only to SDIO cards.	0
10	CPSMEN	RW	CPSM (Command Channel State Machine) enable bit. Setting this bit will... Enable CPSM.	0
9	WAITPEND	RW	Command wait control bit. If this bit is set, CPSM will wait before sending the command. It will wait for data transmission to	0
8	WAITINT	RW	complete. The command waits for the interrupt control bit. If this bit is set, CPSM will disable overcurrent protection. It controls the timing and waits for an interrupt to occur.	0
[7:6]	WAITRESP[1:0]	RW	Response type field. This field indicates the response that CPSM expects to receive. type. 00: No response, waiting for the CMDSENT flag; 01: Short response, waiting for the CMDREND or CCRCFAIL flag; 10: No response, waiting for the CMDSENT flag; 11: Long response, waiting for the CMDREND or CCRCFAIL flag.	0
[5:0]	CMDINDEX[5:0]	RW	command index field. This field indicates the specific command value.	0

32.6.5 Response Register (R32_SDIO_RESPCMD)

Offset address: 0x10

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

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

Bit	name	access	describe	Reset value
[31:6]	Reserved	RO	is reserved.	0
[5:0]	RESPCMD[5:0]		The RO field records the index value of the received response.	0

32.6.6 Access to the Response Parameter Register

Bit	name	(R128_SDIO_RESPX)	describe	Reset value
[127:0]	CARDSTATUSx	RO	Host mode: When the response is a long response, all 128 bits represent the card. Status; when the response is a short response, the lower 32 bits indicate the card status. The SDIO peripheral first receives the most significant bit of the card status and then... R128_SDIO_RESX starts storing from the least significant bit.	0

32.6.6.1 Offset address of the high 32 bits of the response parameter register

(R128_SDIO_RESP1[127:96]) : 0x14

31	30	29	28	27	26	25	24	23	22	...	20	19	18	17	16
CARDSTATUS1[127:112]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARDSTATUS1[111:96]															

Bit	name	access	describe	Reset value
[31:0]	CARDSTATUS1	RO	Host mode: Long response: Card status [127:96]; Short response: Card status [31:0]. Slave mode: Receives command parameters.	0

32.6.6.2 Offset address of the second highest 32 bits of the response parameter register

(R128_SDIO_RESP2[95:64]) : 0x18

31	30	29	28	27	26	25	24	23	22	...	20	19	18	17	16
CARDSTATUS2[95:80]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARDSTATUS2[79:64]															

Bit	name	Access Description	Reset value
[31:0]	CARDSTATUS2	RO Host Mode: Card Status [95:64];	0

			Slave mode: Receives the index of the commands received.	

32.6.6.3 Offset address of the second lowest 32 bits of the response parameter register

(R128_SDIO_RESP3[63:32]) : 0x1C

31	30	29	28	27	26	25	24	23	22	...	20	19	18	17	16
CARDSTATUS3[63:48]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARDSTATUS3[47:32]															

Bit	name	Access	description RO host mode: card	Reset value
[31:0]	CARDSTATUS3	status	[63:32].	0

32.6.6.4 Offset address of the lower 32 bits of the response parameter register

(R128_SDIO_RESP4[31:0]) : 0x20

31	30	29	28	27	26	25	24	23	22	...	20	19	18	17	16
CARDSTATUS4[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARDSTATUS4[15:0]															

Bit	name	Access	description RO host mode: card	Reset value
[31:0]	CARDSTATUS4	status	[31:0].	0

32.6.7 Data Timer Register (R32_SDIO_DTIMER) Offset Address:

0x24

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

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

Bit	name	access	describe	Reset value
[31:0]	DATATIME	RW	data timeout duration. Measured in SDIO_CK cycles.	0

32.6.8 Transfer Length Register (R32_SDIO_DLEN) Offset

Address: 0x28

31	30	29	28	27	26	25	24	23	22	...	20	19	18	17	16
Reserved											DATALENGTH[24:16]				

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

Bit	name	access	describe	Reset value
[31:25] Reserved		RO	Reserved.	0
[24:0]	DATALENGTH[24:0]	RW	Data length field. The value of this field is loaded when transmission is initiated. To the transfer counter. For block transfers, the value of this field must be the block size. Smaller integer multiples, block size is defined by the SDIO device and stored in In R32_SDIO_DCTLR[7:4], common values are 512B, etc.	0

32.6.9 Data Control Register (R32_SDIO_DCTRL) Offset Address: 0x2C

31	30 29 28	27 26 25 24 23 22	2019	18	17	16
Reserved						

15	14 13 12	11	10	9	8	7 6 5 4 3 2	1	0
Reserved		SDIOEN	RWMOD	RWSTOP	RWSTART	DBLOCKSIZE[3:0]	DMAEN	DTMODE	DTDIR	DTEN

Bit	name	access	describe	Reset value
[31:12] Reserved		RO	is reserved.	0
11	SDIOEN	RW	SDIO Enable bit: When this bit is set, DPSM can perform SDIO. Some specific operations of the card.	0
10	RWMOD	RW	Read wait mode: 1. Use SDIO_D2 to control read wait time; 0: Stop SDIO_CK control read wait.	0
9	RWSTOP	RW	Read the stop bit; if the RWSTART bit is set, then read the stop bit. It will be stopped soon.	0
8	RWSTART	RW	is the start of read wait bit. Setting this bit will execute a read wait operation.	0
[7:4]	DBLOCKSIZE[3:0]	RW	Data block length field. This field stores the length of the data block, using the block size... The block transfer length must be defined before transmission. The value that can be written to this field is... The range from 0 to 1110b indicates a block transfer length of 2 BLKLEN, i.e., 0. Between 16384 bytes.	0
3	DMAEN	RW	DMA enable bit: 1: Enable DMA; 0: DMA is off.	0
2	DTMODE	RW	Transmission mode setting bit. Setting this bit sets the transmission mode. 1: Streaming; 0: Block transfer.	0
1	DTDIR	RW	Transmission direction setting bit. Set this bit to set the transmission direction. 1: The controller is stuck; 0: Controller to card;	0
0	DTEN	RW	Transmission enable bit. Setting this bit will initiate data transmission. The specific process is as follows: after setting this bit, DPSM enters either Wait_S or Wait_R. The process (depending on the direction of transmission).	0

32.6.10 Transfer Count Register (R32_SDIO_DCOUNT)

Offset address: 0x30

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

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

Bit	name	access	describe	Reset value
[31:25] Reserved		RO Reserved.		0
[24:0]	DATACOUNT[24:0]	RO	Transmission data counter field. The transmit length register is used when initiating transmission. The value of the counter will be loaded into this counter and will continue to be used during the transmission process. Decreasing.	0

32.6.11 Status Register (R32_SDIO_STA) Offset Address:

0x34

31 30 29 28	27 26	25 24 23 22 21	2019	18	17	16
Reserved			CEATAEND	SDIOIT	RXDAVL	TXDAVL	RXFIFOE	TXFIFOE	RXFIFOF	TXFIFOF

15	14 13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXFIFOHF	TXFIFOHE	RXACT	TXACT	CMDACT	DBCKEND	STBITERR	DATAEND	CMDSENT	CMDREND	RXOVERR	TXUNDERR	DTIMEOUT	CTIMEOUT	DCRCFAIL	CCRCFAIL

Bitname	access	describe	Reset value
[31:24] Reserved	RO is reserved.		0
23 CEATAEND	When RO is set, CMD61 receives the CE-ATA completion signal. When RO is		
22 SDIOIT	set, SDIO receives a device interrupt.		0
21 RXDAVL	When RO is set, FIFO data is available for receiving.		0
20 TXDAVL	When the RO bit is set, data in the FIFO is available for transmission.		0
19 RXFIFOE	When RO is set, the receive FIFO is empty.		0
18 TXFIFOE	When RO is set, the FIFO is sent empty.		0
17 RXFIFOF	When RO is set, the receive FIFO is full.		0
16 TXFIFOF	When RO is set, the FIFO is full.		0
15 RXFIFOHF	When RO is set, the receive FIFO is half full.		0
14 TXFIFOHE	When RO is set, the FIFO is half empty.		0
13 RXACT	When RO is set, data is being received.		0
12 TXACT	When RO is set, data is being sent.		0
11 CMDACT	When RO is set, a command is being transmitted. When		0
10 DBCKEND	RO	this bit is set, a data block has been sent or received and the CRC checksum is correct. Verification passed.	0
9 STBITERR	RO	When this bit is set, in wide bus mode, all data lines are not active. A start signal has been detected.	0
8 DATAEND	When RO is set, data transmission has ended (transmission counter is zero).		
7 CMDSENT	When RO is set, the command has been sent.		0

6	CMDREND	When RO is set, a response has been received and the CRC check was successful.		
5	RXOVERR	When RO is set, the receive FIFO overflows.		0
4	TXUNDERR	When RO is set, a FIFO underflow is sent.		0
3	DTIMEOUT	When RO is set, the data times out.		0
2	CTIMEOUT	RO	When this bit is set, the command timeout exceeds 64 SDIO_CK cycles. Expect.	0
1	DCRCFAIL	RO	When this bit is set, a data block has been sent or received, but the CRC... Verification failed.	0
0	CCRCFAIL	When the RO bit is set, a response has been received, but the CRC check failed.		0

32.6.12 Interrupt Clear Register (R32_SDIO_ICR)

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

15 14 13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			DBCKENDC	STBITERRC	DATAENDC	CMDSENTC	CMDRENDC	RXOVERRC	TXUNDERRC	DTIMEOUTC	CTIMEOUTC	DCRCFAILC	CCRCFAILC

Bitname		access	describe	Reset value
[31:24] Reserved		RO is reserved.		0
23	CEATAENDC	RW sets this bit to clear the CEATAEND bit in the status register.		0
22	SDIOITC	RW sets this bit to clear the SDIOIT bit in the status register.		0
[21:11] Reserved		RW reserved.		0
10	DBCKENDC	RW sets this bit to clear the DBCKEND bit in the status register.		0
9	STBITERRC	RW sets this bit to clear the STBITERR bit in the status register.		0
8	DATAENDC	RW sets this bit to clear the DATAEND bit in the status register.		0
7	CMDSENTC	RW sets this bit to clear the CMDSENT bit in the status register.		0
6	CMDRENDC	RW sets this bit to clear the CMDREND bit in the status register.		0
5	RXOVERRC	RW sets this bit to clear the RXOVERR bit in the status register.		0
4	TXUNDERRC	RW sets this bit to clear the TXUNDERR bit in the status register.		0
3	DTIMEOUTC	RW sets this bit to clear the DTIMEOUT bit in the status register.		0
2	CTIMEOUTC	RW sets this bit to clear the CTIMEOUT bit in the status register.		0
1	DCRCFAILC	RW sets this bit to clear the DCRCFAIL bit in the status register.		0
0	CCRCFAILC	RW sets this bit to clear the CCRCFAIL bit in the status register.		0

32.6.13 Interrupt Enable Register (R32_SDIO_MASK) Offset

Address: 0x3C

31 30 29 28 27 26 25 24 23 22 21	20	19	18	17 16
Reserved	CEATAENDIE	SDIOITIE	RXDAVLIE	TXDAVLIE	RXFIFOEIE	TXFIFOEIE	RXFIFOFIE	TXFIFOFIE

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

RXFIFOHFIE	TXFIFOHEIE	RXACTIE	TXACTIE	CMDACTIE	DBCKENDIE	STBITERRIE	DATAENDIE	CMDSENTIE	CMDRENDIE	RXOVERRIE	TXUNDERRIE	DTIMEOUTIE	CTIMEOUTIE	DCRCFAILIE	CCRCFAILIE

Bit	name	access	describe	Reset value
[31:24] Reserved		RO is reserved.		0
23	CEATAENDIE	RW	Setting this bit will cause the status register to generate a value when the CEATAEND bit is set. Break.	0
22	SDIOITIE	Setting RW to this bit will cause the status register to generate an interrupt when the SDIOIT bit is		
21	RXDAVLIE	set. Setting RW to this bit will cause the status register to generate an interrupt when the RXDAVL		
20	TXDAVLIE	bit is set. Setting RW to this bit will cause the status register to generate an interrupt when the		
19	RXFIFOEIE	RW	TXDAVL bit is set. Setting RW to this bit will cause the status register to generate an interrupt when the RXFIFOE bit is set. Break.	0
18	TXFIFOEIE	RW	Setting this bit will cause the status register to generate a value when the TXFIFOE bit is set. Break.	0
17	RXFIFOFIE	RW	Setting this bit will cause the status register to generate a value when the RXFIFOF bit is set. Break.	0
16	TXFIFOFIE	RW	Setting this bit will cause the status register to generate a value when the TXFIFOF bit is set. Break.	0
15	RXFIFOHFIE	RW	Setting this bit will cause the status register to generate a value when the RXFIFOHF bit is set. Break.	0
14	TXFIFOHEIE	RW	Setting this bit will cause the status register to generate a value when the TXFIFOHE bit is set. Break.	0
13	RXACTIE	Setting RW to this bit will cause the status register to generate an interrupt when the RXACT bit is		
12	TXACTIE	set. Setting RW to this bit will cause the status register to generate an interrupt when the TXACT bit		
11	CMDACTIE	is set. Setting RW to this bit will cause the status register to generate an interrupt when the CMDACT		
10	DBCKENDIE	RW	bit is set. Setting RW to this bit will cause the status register to generate an interrupt when the DBCKEND bit is set. Break.	0
9	STBITERRIE	RW	Setting this bit will cause the status register to generate a value when the STBITERR bit is set. Break.	0
8	DATAENDIE	RW	Setting this bit will cause the status register to generate an error when the DATAEND bit is set. Break.	0
7	CMDSENTIE	RW	Setting this bit will cause the status register to generate a status signal when the CMDSENT bit is set. Break.	0
6	CMDRENDIE	RW	Setting this bit will cause the status register to generate a status code when the CMDREND bit is set. Break.	0
5	RXOVERRIE	RW	Setting this bit will cause the status register to generate an intermediate value when the RXOVERR bit is set. Break.	0
4	TXUNDERRIE	RW	Setting this bit will cause the status register to generate a value when the TXUNDERR bit is set. Break.	0
3	DTIMEOUTIE	RW	Setting this bit will cause the status register to generate a timeout value when the DTIMEOUT bit is set. Break.	0
2	CTIMEOUTIE	RW	Setting this bit will cause the status register to generate a value when the CTIMEOUT bit is set. Break.	0
1	DCRCFAILIE	When RW is set to this bit, the status register will generate a 0 when the DCRCFAIL bit is set.		

			Break.	
0	CCRCFAILIE	RW	Setting this bit will cause the status register to generate an error when the CCRCFAIL bit is set. Break.	0

32.6.14 FIFO counter register (R32_SDIO_FIFOCNT) offset address: 0x48

31	30	29	28	27	26	25	24	23	22	twenty one	20	19	18	17	16
FIFOCOUNT[31:16]															

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

Bit	name	access	The	Reset value
[31:0]	FIFOCOUNT[31:0]	RO	description of the FIFO includes items that have not yet been written to or read from the FIFO. Number of words (32 bits). In setting R32_SDIO_DCTLR: When R32_SDIO_DCTLR is active, if DPSM is idle, the FIFO counter... The transfer length value will be loaded from R32_SDIO_TLEN, if this value If it is not divisible by 4, then the last 1 to 3 bytes will be treated as one. Individual character processing.	0

32.6.15 Data Control Register 2 (R32_SDIO_DCTRL2) Offset Address: 0x60

31	30	29	28	27	26	25	24	23	22	twenty one	2019	18	17	16
Reserved				SLV_CK_PHASE		SLV_FORCE_ERR		SLV_MODE		Reserved				RANDOM_LEN_EN

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				DBLOCKSIZE2											

Bit	name	access	describe	Reset value
[31:27]	Reserved	R0 is reserved.		0
26	SLV_CK_PHASE	RW	When SLV_MODE is 1, the phase selection during DATA output from the mode is... Bit: 1: Output DATA on the falling edge of the internal SDCK; 0: Output DATA on the rising edge of the internal SDCK. When SLV_MODE is 0: 1: Writing 0 to the DTEN bit does not clear the FIFO; 0: Write 0 to the DTEN bit to clear the FIFO.	0
25	SLV_FORCE_ERR	RW	Software-forced CRC error in slave mode 1: The software forces the CRC of the data block to be an error value; 0: Use the correct CRC value from the hardware.	0
24	SLV_MODE	RW	From the mode enable bit: 1: The controllers CPSM and DPSM operate in slave mode;	0

			0: The controller is operating in host mode.	
[23:17] Reserved		R0 is reserved.		0
16	RANDOM_LEN_EN	RW	Enable bit for arbitrary byte length of data block: 1: Data blocks can be of any length from 0 to 2048, with a length of... DBLOCKSIZE2; 0: The data block can only be 2 DBLOCKSIZE bytes.	0
[15:12] Reserved		R0 is reserved.		0
[11:0] DBLOCKSIZE2		RW (Random Byte Length) field for data block length in arbitrary byte length mode.		0

32.6.16 FIFO register (R32_SDIO_FIFO) offset address:

0x80

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

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

Bit	name	access	describe	Reset value
[31:0]	FIFODATA[31:0]	RW	FIFO Data Field. This field contains the data for the FIFO. Reading or writing to this field will... Read received data or send data to be sent. SDIO FIFO consists of 32 words (one word is 32 bits).	0

Chapter 33 SD/EMMC Controller (SDMMC)

The module descriptions in this chapter apply only to CH32H417 and CH32H416 microcontroller products.

The system provides one SDMMC controller master/slave interface, with a transmission clock up to 200MHz, supporting 1/4/8-wire communication modes.

Supports dual-edge sampling and can be connected to external SD/TF cards, eMMC cards, and other devices. The application code allows for flexible configuration of various commands for data transmission and reception.

Parameters such as the mode and length of the response packet and valid data packet, and the double buffer length switching limit.

Note: When using SDMMC in single-wire or four-wire mode, the pins corresponding to unused data lines cannot be used for multiplexed output. GPIO is a multiplexed input, and can also be used for general purposes. GPIO Output.

33.1 Main Features

Supports SD physical layer 1.0 and 2.0 specifications, and supports UHS-I SDR50, DDR50 and SDR104 modes of SD 3.0 specification.

Compliant with eMMC card specifications 4.4 and 4.5.1; supports eMMC card 5.0 specifications for HS200 and HS400. Communication modes

support single-wire, four-wire, and eight-wire modes.

• Maximum communication clock speed up to 200MHz for single edge and 180MHz for dual edge •

Supports dual edge sampling

ÿ Flexible and configurable data packet length, command format, and response status

Provides hardware functionality to automatically stop the clock during data block intervals.

Supports devices compliant with the SD interface protocol, such as SD cards, SDIO cards, and eMMC cards.

Supports SDIO slave interface, enabling data exchange with chips that support SDIO master interface.

ÿ DMA double buffering function

33.2 Functional Description

33.2.1 Communication Clock ÿ

Frequency Configuration

The controller interface provides two clock modes: low-speed mode and high-speed mode. Generally, the SD protocol specifies the clocking mode for SD interface devices.

Initially, a communication clock of approximately 400kHz is used to ensure better communication compatibility. After obtaining the parameter information of the external device,

It can switch to a higher clock frequency for communication, depending on the configuration supported internally.

Set bit 9 of the R16_EMMC_CLK_DIV register to select the clock mode, and set the divider values of bits [4:0] to obtain the final SDIO.

Interface output clock. Bit 8 controls whether the EMMC peripheral outputs a clock signal to the MSDCK pin. It is necessary to ensure that the MSDCK pin's I/O mode...

The configuration is push-pull output.

Phase configuration

When communicating with external SDIO interface devices, factors such as high clock frequency, hardware routing, and device characteristics can cause signal interference.

The sampling number is incorrect.

The system SDIO controller host interface provides an output clock phase flip (180° offset) function, which can be achieved by setting a register.

Writing 1 to bit 10 of R16_EMMC_CLK_DIV will physically toggle the output clock signal, but the controller's internal clock will remain unchanged.

This remains unchanged. This method allows for adjustment of communication timing.

Note: At higher SDCLK frequencies, if the recommended timing adjustment register values cannot ensure stable communication, the user needs to initiate bus sampling tuning.

The sequence is used to find better sampling points.

33.2.2 Command Sending and Response SDIO master

interfaces send fixed-length 48-bit data packets of commands to slave devices, which are transmitted serially on the MCMD line. Some commands are not...

A response from the slave device is required; some commands require a response packet or data packet of varying length from the slave device.

ÿ Typical command format: Start bit (0) + Transmission bit (1) + Command index + Parameters + CRC7 + End bit (1)

The start bit, transmission bit, CRC7, and end bit are automatically filled by the hardware. The command index and parameters vary depending on the command and need to be passed to the hardware controller by the application code. The parameter fields in the SD command are configured by writing to the R32_EMMC_ARGUMENT register; bits [5:0] of the R16_EMMC_CMD_SET register are filled to complete the command index, and bits [11:8] of the register are configured to detect the expected response packet for this command. The write operation to the R16_EMMC_CMD_SET register will trigger the hardware to complete the

transmission of the command packet sequence on the MCMD signal line. ÿ Short response format: start bit (0) + transmission

bit (0) + command index + parameter + CRC7 + end bit (1) ÿ Long response format: start bit (0) + transmission bit (0) + reserved field (111111b) + data + end bit (1)

The slave response packets are divided into 48-bit and 136-bit lengths, with 32 bits and 128 bits respectively being valid data bits for the user. After the master sends a command, the valid data in the slave interface response packet can be obtained by querying the R16_EMMC_INT_FG register. If the flag bit RB_EMMC_IF_CMDDONE is set to 1, the R32_EMMC_RESPONSE register can be read. If the response packet is abnormal, bits [2:0] of register R16_EMMC_INT_FG will have the corresponding flags set.

33.2.3 Continuous Reading of Multiple Data Blocks

For the data reading operation (transferring via data line) in the SD protocol command, the following configuration is required:

1) Initialization: Configure the R16_EMMC_CONTROL register to set the sampling edges of the command and data, the data line width for transmission and reception, and enable DMA function. Configure the R8_EMMC_TIMEOUT register RB_EMMC_TOCNT_MASK bit to set the command timeout and data timeout. Configure the R32_EMMC_DMA_BEG1 register to set the starting address of the DMA for storing read data. Configure the R32_EMMC_BLOCK_CFG register to set the length of a single data block and the total number of data blocks to be read during transmission. Configure the R32_EMMC_TRAN_MODE register to set the DMA transfer direction to "SD to controller". Optional settings include whether to automatically stop the clock during the block transfer interval to give the application code sufficient time to process the data before the software restarts the transfer.

2) Send a command to the SDIO slave interface device. This command instructs the slave device to transfer data from the data line to the master interface. For example, CMD17 and CMD18 in the SD protocol.

3) Controller Interface Reads Data from Data Line: When the slave interface device responds to the host command, if the device is ready for data transmission, it will output data from the data line based on the host clock signal, with the start signal being low. The host controller samples the signal and transfers it to the user-configured SRAM area via internal DMA. Each transfer of data of the size set by RB_EMMC_BKSIZE_MASK indicates the completion of a data block transfer, and the total number of internally transferred data blocks is decremented by 1 until all data blocks have been transferred, at which point the internal DMA will no longer perform data transfer. During this process, the hardware will set the RB_EMMC_IF_BKGAP flag after each data block is read, and set the RB_EMMC_IF_TRANDONE flag when all data blocks have been read. If a data timeout or a CRC error occurs while the controller is waiting to receive data, the RB_EMMC_IF_DATTMO and RB_EMMC_IF_TRANERR flags will be set respectively. Application code can configure the corresponding interrupt control bits in the interrupt enable register R16_EMMC_INT_EN. When the corresponding interrupt flag is set, the EMMC interrupt service can be triggered.

4) Status Detection: The R32_EMMC_STATUS register can query the level status of the data line MD0 and the command line MCMD in real time, and make judgments on the timing specified by some protocols. The MASK_BLOCK_NUM field records the number of data blocks that have been successfully transferred by DMA in the current command transmission.

33.2.4 Writing multiple data blocks consecutively

1) Initialization: Configure the R16_EMMC_CONTROL register to set the sampling edges for commands and data, the width of the transmit and receive data lines, and enable DMA. Configure the R8_EMMC_TIMEOUT register RB_EMMC_TOCNT_MASK bit to set the command timeout and data timeout. After the command is successfully sent, configure the R32_EMMC_BLOCK_CFG register to set the length of a single data block and the total number of data blocks to be read during the transmission, i.e., start data transmission. Configure the R32_EMMC_TRAN_MODE register to set the DMA transmission direction to "controller to SD". 2) Send commands to the SDIO slave

interface device. This command requires the master interface to transmit data to the slave device through the data line, such as CMD24 and CMD25 in the SD protocol. 3) The controller interface writes data to the

data line: When the slave interface device responds to the master command, it generates a response packet and releases the data.

When the line is in a non-low state, the host can drive the data line to output a data block. This can be achieved by writing to the DMA register R32_EMMC_DMA_BEG1.
Alternatively, performing a write operation to the R32_EMMC_WRITE_CONT register will also start the host interface driver data line. The difference lies in the operation register...
The hardware will load the current value of register R32_EMMC_DMA_BEG1 into its internal memory and retrieve it from this address.
Data is sent to the controller for transmission; however, when a write operation is performed on the R32_EMMC_WRITE_CONT register, the hardware will use the current internal DMA.
The system continues fetching SRAM data from the moved position and sending it to the controller. Therefore, if the DMA address of the data needs to be changed (not compared to the previous address),...
(Continuous) The R32_EMMC_DMA_BEG1 register needs to be rewritten to continue data output on the data line. If it is connected to the address where data was previously written...
Continuing, the hardware will automatically fetch the data address and move it internally. You can directly perform a write operation on the R32_EMMC_WRITE_CONT register to allow...
The controller continues data output via the data line. Each output of data of the size set by RB_EMMC_BKSIZE_MASK indicates the completion of a data block transmission.
The internal DMA will output data, and at the same time, the total number of internal data blocks will be reduced by 1 until all data blocks have been transmitted. The internal DMA will then stop outputting data.
During this process, the hardware will set the RB_EMMC_IF_BKGAP flag after each data block is read. Once all data blocks have been read...
The hardware will set the RB_EMMC_IF_TRANDONE flag if a data timeout occurs while the controller is waiting to receive data or if data is not received.
A CRC error will trigger the RB_EMMC_IF_DATTMO and RB_EMMC_IF_TRANERR flags respectively. Application code can configure this.
The corresponding interrupt control bit in the interrupt enable register R16_EMMC_INT_EN can trigger an EMMC interrupt when the corresponding interrupt flag is set.
Serve.

setting. 1. When the controller interface writes data blocks to an external device, it needs to wait for the command to be sent and the response to be completed before DMA Address, because
is an address setting operation that notifies the hardware driver data line. If the command has not been sent completely at this time, the device will not accept data.
Note 2: If an error (RB_EMMC_IF_TRANERR) occurs during the transfer process of a command execution (including data reading and data writing). The
CRC hardware will not count the erroneous data block in the success count, nor will it decrement the total block count; it will continue transmitting the next block. DMA Still
However, it will place the erroneous data block into the appropriate address range. Application code should pay attention to how it is handled at this point. CRC SRAM

33.2.5 DMA Double Buffering Function

The system provides DMA double buffering functionality, with two DMA address registers, R32_EMMC_DMA_BEG1 and R32_EMMC_DMA_BEG2.
And the data block length threshold RB_EMMC_DMATN_CNT for DMA switching. Data starting at the address of register R32_EMMC_DMA_BEG1.
Once the block length threshold transfer is complete, the hardware will switch to the address region set by R32_EMMC_DMA_BEG2 for data transfer, and will proceed with the transfer once the threshold is reached.
After reaching the threshold, switch back to the R32_EMMC_DMA_BEG1 region, and repeat this cycle until the total number of data blocks configured by RB_EMMC_BKNUM_MASK is reached.
All data has been transmitted.

Each time a non-zero value is written to the RB_EMMC_BKNUM_MASK register of the R32_EMMC_BLOCK_CFG register, data transfer will be initiated.
Simultaneously, the hardware will also point the DMA to the R32_EMMC_DMA_BEG1 address region. The flag setting conditions during the transfer process are as described above.
The application code can optionally enable the corresponding interrupt bit to trigger the EMMC interrupt service.

33.2.6 Slave Mode

Operation in Slave Mode

The hardware in slave mode only supports R1 type acknowledgements, therefore the R16_EMMC_CMD_SET register has no function in slave mode.

1) The master machine queries the slave machine's data status:

The slave device pre-fills its expected data reception/transmission status into the ARGUMENT register, and the master sends CMD13 to query the slave device status.
The host determines whether to read from or write to the slave based on the response.

Note: Whether to use CMD13 and the bit definitions in the slave response are both software-defined.

2) The host sends CMD18 to read multiple blocks:

The CMD18 command parameters on the master device no longer specify the sector address, but rather the number of blocks to be read. After the slave device receives the receive command interrupt, it configures...
Set the DMA address and, read Configure R32_EMMC_TRAN_MODE Receive command parameter register to fill it in.
The R32_EMMC_BLOCK_CFG register.

3) The host sends CMD25 to write multiple blocks:

The CMD25 command parameters on the master device no longer specify the sector address, but rather the number of blocks to be written. After the slave device receives the receive command interrupt, it configures...
Set DMA address , Configure R32_EMMC_TRAN_MODE , Read the number of blocks from the receive command parameter register and fill it in.
The R32_EMMC_BLOCK_CFG register.

Slave error handling

The slave responds to all commands from the master with an R1 type response, the response parameter being the value of the slave's ARGUMENT register. The slave may...
If a CRC error occurs when receiving a command and the host does not respond, the host should retry the current command.

If an internal FIFO overflow occurs during transmission or reception, the hardware will automatically force the data block to have a CRC error.
This notification indicates a problem with the host transmission; the software can also set RB_SLV_FORCE_ERR to force the transmission of erroneous CRC blocks.

The slave hardware parses a special command, CMD63, to simplify the slave software's handling of transmission errors. If the master has sent/received...
If a CRC error occurs during the process, the host immediately sends CMD63. Upon receiving CMD63, the slave device's software needs to reset the data state machine.
The hardware will automatically point the DMA address to the starting address of the previous erroneous packet. This way, once the master and slave agree that transmission can continue, the read operation will be able to proceed smoothly.
For write operations, the slave software only needs to set the number of blocks in R32_EMMC_BLOCK_CFG. After setting R32_EMMC_BLOCK_CFG, the software will automatically write the number of blocks.
By counting the number of blocks and writing to the RESPONSE3 register, retransmission can be completed and transmission can continue.

Double buffering mode

If the slave device uses double buffering mode, and an error occurs when the master device reads the last block of data from buf0, the master device will buffer the last block of data from buf1.
During a data block, CMD63 is sent to indicate a CRC error in the previous packet; therefore, the slave should only send a new data block after the first block of buf1 has been successfully transmitted.
The data in buf0 is updated, and the newly added interrupt bit RB_SIF_SLV_BUF_RELEASE is generated when the first sector of the new buf is successfully transmitted.
born.

1 block | Update buf1 buffer

buf0 ----------

1 block | Update buf0 buffer

buf1 ----------

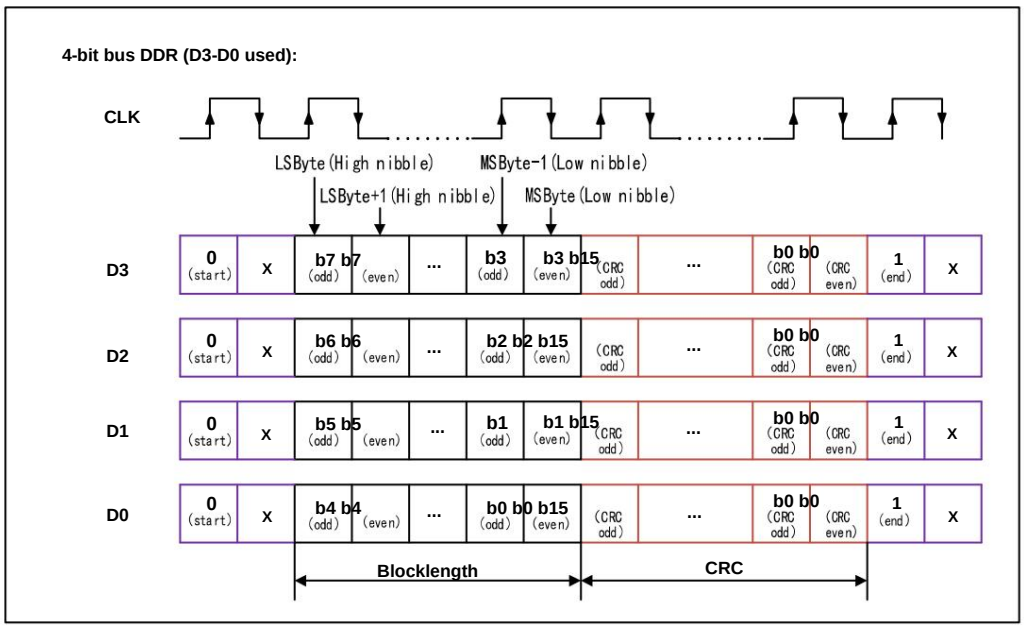
DDR mode

supports both 4-wire and 8-wire DDR modes.

1) 4-wire DDR mode:

This means that data is output and sampled using the rising and falling edges of the clock to improve the data transmission rate. In application, 4-wire mode should be enabled.
Registers and DDR mode registers;

Figure 33-1 4-line DDR mode

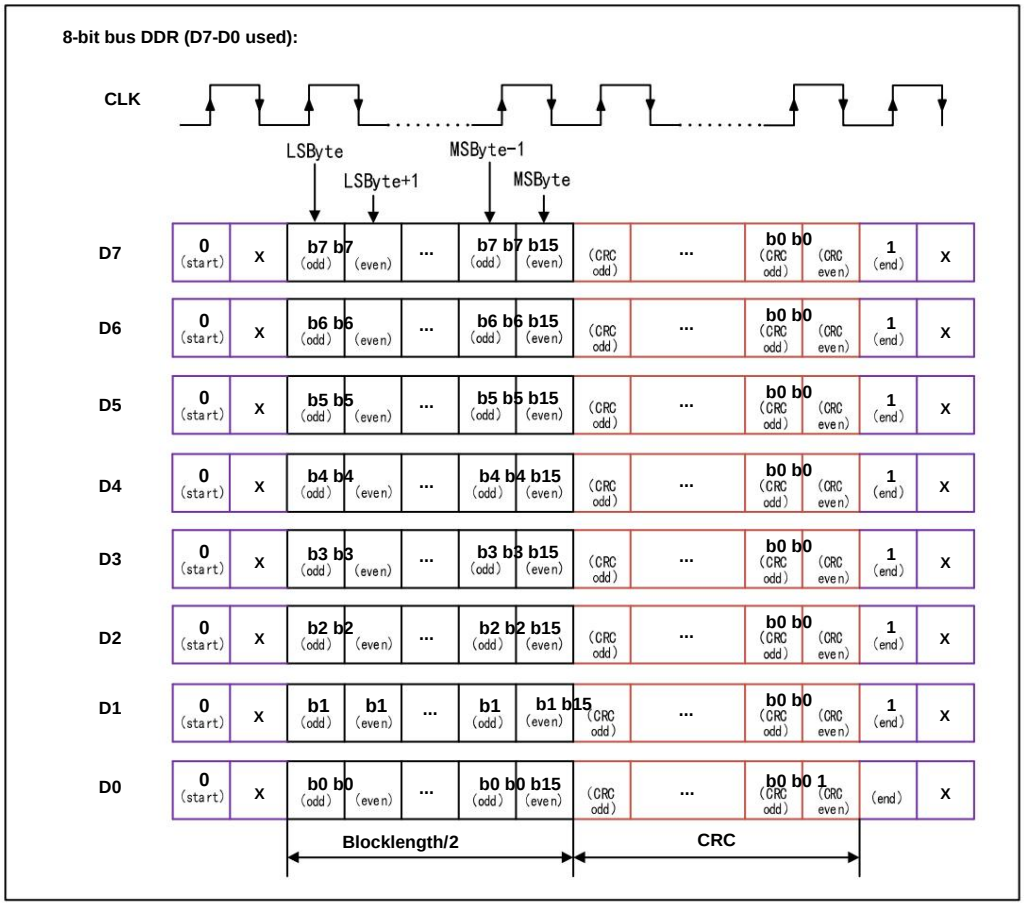


2) 8-wire DDR mode:

This means that data is output and sampled using the rising and falling edges of the clock to improve the data transmission rate. In application, 8-wire mode should be enabled.

Registers and DDR mode registers;

Figure 33-2 8-line DDR mode



33.3 Register Description

Table 33-1 List of EMMC-related registers

name	Access Address Description 0x40024000		Reset value
R32_EMMC_ARGUMENT	Command Parameter Register 0x40024004		0x00000000
R16_EMMC_CMD_SET	Command Set Register 0x40024008		0x0000
R32_EMMC_RESPONSE0	Acknowledgment Parameter Register 0		0x00000000
R32_EMMC_RESPONSE1	0x4002400C Acknowledgment Parameter		0x00000000
R32_EMMC_RESPONSE2	Register 1 0x40024010 Acknowledgment		0x00000000
R32_EMMC_RESPONSE3	Parameter Register 2 0x40024014		0x00000000
R32_EMMC_WRITE_CONT	Acknowledgment Parameter Register 3		0x00000000
R16_EMMC_CONTROL	0x40024014 Continue Write Start		0x0015
R8_EMMC_TIMEOUT	Register 0x40024018 Control Register		0x0C
R32_EMMC_STATUS	0x4002401C Timeout Counter		0x00000000
R16_EMMC_INT_FG	Register 0x40024020 Status Register		0x0000
R16_EMMC_INT_EN	0x40024024 Interrupt Flag Register 0x40024028 Interrupt Enable Register		0x0000
R32_EMMC_DMA_BEG1	0x4002402C	DMA start address 1 register	0x0000XXXX

R32_EMMC_BLOCK_CFG	0x40024030 Transport Block Configuration	0x00000000
R32_EMMC_TRAN_MODE	Register; 0x40024034 Transport Mode	0x00001040
R16_EMMC_CLK_DIV	Register; 0x40024038 Clock Configuration Register	0x0213
R32_EMMC_DMA_BEG2	0x4002403C DMA Start Address 2 Register	0x00000000
R32_EMMC_TUNE_DATO	0x40024040 Data Output Delay Register	0x00000000
R32_EMMC_TUNE_DATI	0x40024044 Data Input Delay Register	0x00000000
R32_EMMC_TUNE_CLK_CMD	0x40024048 Clock and Command Delay Register	0x00000000

33.3.1 Command Parameter Register (R32_EMMC_ARGUMENT) Offset

Address: 0x00

31	30 29 28	27 26 25 24 23 22 21	2019	18	17 16
EMMC_ARGUMENT[31:16]					

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

Bit	name	access	describe	Reset value
[31:0]	EMMC_ARGUMENT	RW	Host mode: 32-bit command parameter register of SD/eMMC. Slave mode: Register the 32-bit response parameters of the slave device's reply. Utensils.	0

33.3.2 Command Set Register (R16_EMMC_CMD_SET) Offset

Address: 0x04 Bit

Name [15:12] Reserved		access	describe	Reset value
		RO is reserved.		0
11	RB_EMMC_CKIDX	RW	Command index for verifying the response: 1. Required; 0: Not required.	0
10	RB_EMMC_CKCRC	RW	Verify the CRC of the response: 1: Required; 0: Not required.	0
[9:8]	RB_EMMC_RPTY_MASK	RW	Expected response type: 00: No response; 01: The response length is 136 bits; 10: The response length is 48 bits; 11: The response length is 48 bits and it is an R1b type response.	0
[7:6] Reserved [5:0]		RO is reserved.		0
RB_EMMC_CMDIDX_MASK		RW	Index number of the currently sent command.	0

33.3.3 Reply Parameter Register (R32_EMMC_RESPONSE) Offset

Address: 0x08, 0x0C, 0x10, 0x14, 0x14 bit access

	name		describe	Reset value
[31:0]	R32_EMMC_RESPONSE0	RO	Response parameter register 0.	0
[63:32]	R32_EMMC_RESPONSE1	RO	Response parameter register 1.	0
[95:64]	R32_EMMC_RESPONSE2	RO	Host Mode: Response Parameter Register 2;	0

			Slave mode: Receives the index of the commands received.	
[127:96]	R32_EMMC_RESPONSE3	RO	Host mode: Response parameter register 3; Slave mode: Receive command parameter register.	0
[127:96]	R32_EMMC_WRITE_CONT	WO	Multiplexes the EMMC_RESPONSE3 register for use in multi-block write operations. During the process, the write operation is initiated.	0

Note: When the response length is 1 bit, the valid data length is 1 bit; when the response length is 1 bit, the valid data length is 1 bit. The R32_EMMC_RESPONSE3 register stores the valid data parameters of the response.

The R32_EMMC_RESPONSE3 register is being reused. This reuse occurs when multiple data blocks are written to the card consecutively using the CMD25 command. When a block transfer is complete, writing to this register will cause the controller to automatically offset the address by the block length and continue writing data. If necessary... To change the DMA address, write to the DMA address register, which will initiate the write operation, eliminating the need to start by writing to the register.

33.3.4 Control Register (R16_EMMC_CONTROL) Offset Address: 0x18

Bit Name [15:10]

Reserved		access	describe	Reset value
		RO reserved.		0
9	RB_SLV_FORCE_ERR	RW	Software-forced data block CRC error in slave mode: 1: The software forces the CRC of the data block to be an error value; 0: Use the correct CRC value from the hardware.	0
8	RB_SLV_MODE	RW	Slave mode enable bit: 1: Slave mode, SDCK input; 0: Host mode, SDCK output.	0
[7:6] Reserved		RO is reserved.		0
5	RB_EMMC_NEGSMP	RW	CMD and Data signal line sampling mode selection bits: 1: Sampling on the falling edge; 0: Sampling on the rising edge.	0
4	RB_EMMC_RST_LGC	RW	Internal logic reset of the module: 1: Perform a reset; 0: Normal operation;	1
3	RB_EMMC_DMAEN	RW	Controller DMA Enable: 1: Enable DMA; 0: DMA is off.	0
2	RB_EMMC_ALL_CLR	RW	SD controller logic reset: 1. Resetting the SD controller logic requires software clearing; 0: Normal operation.	1
[1:0]	RB_EMMC_LW_MASK	RW	Valid data width for sending and receiving data: 00: The transceiver uses only D[0], a single data line; 01: The transceiver uses D[3:0], 4 data cables; 10: The transceiver uses D[7:0], 8 data lines.	01b

33.3.5 Timeout Control Register (R8_EMMC_TIMEOUT) Offset Address: 0x1C

Bit	name	access	describe	Reset value
[7:4]	Reserved	RO.		0
[3:0]	RB_EMMC_TOCNT_MASK	RW	Response/Data Timeout Configuration.	0xC

			Non-zero: Sets the timeout period, valid values are 1-15; Calculation: Module clock cycle * 4194304 * RB_EMMC_TOCNT_MASK. For example: if the SDCLK cycle is 10ns and 12 is written, then the timeout period is... The result is 10ns * (4194304) * (12) = 503ms. 0: Not available.	

Note 1: The above data timeout includes the following four situations:

1) R1b Timeout after response D[0] busy 2)

When writing data blocks, the CRC status is followed by D[0] Timeout;

While writing data blocks, wait CRC status time out;

4) When reading a data block, wait for the start bit to time out. Note 2: Command acknowledgments also support a timeout mechanism. If the acknowledgment times out, an interrupt will occur via R16_EMMC_INT_FG in the interrupt register. Interruption to

The command timeout uses the maximum timeout value given by the protocol: 64Tsdclk.

33.3.6 Status Indicator Register (R32_EMMC_STATUS) Offset Address:

0x20

31	30	29 28 27 26 25 24 23 22 21	2019	18	17	16
Reserved					RB_EMMC_DAT0STA	RB_EMMC_CMDSTA

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

Bit	name	access	describe	Reset value
[32:18] Reserved		RO is reserved.		0
17	RB_EMMC_DAT0STA	RO	Current level status of data line D0: 1: High level; 0: Low level.	0
16	RB_EMMC_CMDSTA	RO	CMD signal line current level status: 1: High level; 0: Low level.	0
[15:0] MASK_BLOCK_NUM		RO: The number of blocks successfully transferred in the current multi-block transfer operation.		0

33.3.7 Interrupt Flag Register (R16_EMMC_INT_FG) Offset Address:

0x24 Bit Name [15:11]

Reserved		access	describe	Reset value
		RO is reserved.		0
10	RB_SIF_SLV_BUF_RELEASE	RW1Z	In slave double-buffered mode, the BUF release flag is cleared by writing 1. 1. The software can update (write/read) the corresponding data in the buffer. according to; 0: The corresponding buf data is still in use.	0
9	RB_EMMC_IF_SDIOINT	RW1Z	SDIO card interrupt flag: Write 1 to clear. 1: The SDIO card generates a card interrupt; 0: No event.	0
8	RB_EMMC_IF_FIFO_OV	RW1Z	FIFO overflow flag; write 1 to clear.	0

			1: FIFO overflow triggered; 0: No event.	
7	RB_EMMC_IF_BKGAP	RW1Z	Single-block transfer complete flag: Write 1, clear: 1: Triggered upon completion of single-block transmit/receive; 0: No event.	0
6	RB_EMMC_IF_TRANDONE RW1Z		The flag indicating all blocks have been transferred is written to 1 and then cleared. 1: Triggered when all requested blocks have been transmitted; 0: No event.	0
5	RB_EMMC_IF_TRANERR RW1Z		Data transmission CRC error flag, write 1 to clear: 1: CRC error triggered; 0: No event.	0
4	RB_EMMC_IF_DATTMO RW1Z		Data timeout flag: Write 1 to clear. 1: Data timeout triggered; 0: No event.	0
3	RB_EMMC_IF_CMDDONE RW1Z		Command completion flag / Slave command completion flag, write 1 to clear zero: 1: The master sends a command and receives a response; the slave receives... Command complete; 0: No event.	0
2	RB_EMMC_IF_REIDX_ER RW1Z		Response index number verification error flag: Write 1 to clear: 1: Triggered by a response index number verification error; 0: No event.	0
1	RB_EMMC_IF_RECRC_WR RW1Z		Respond to the CRC check error flag, write 1 to clear it: 1: Triggered by CRC check error response; 0: No event.	0
0	RB_EMMC_IF_RE_TMOUT RW1Z		Receive response timeout flag: Write 1 to clear. 1: Response timeout triggered; 0: No event.	0

33.3.8 Interrupt Enable Register (R16_EMMC_INT_EN) Offset Address: 0x28

Bit	name	access	describe	Reset value
[15:11] Reserved		RO is reserved.		0
10	RB_SIE_SLV_BUF_REL EASE	RW	In slave double-buffered mode, BUF releases interrupt enable: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
9	RB_EMMC_IE_SDIOINT RW		SDIO card interrupt enable: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
8	RB_EMMC_IE_FIFO_OV RW		FIFO overflow interrupt enable: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
7	RB_EMMC_IE_BKGAP	RW	Single-block transfer completion interrupt enable: 1: Enable the corresponding interrupt;	0

			0: Disable the corresponding interrupt.	
6	RB_EMMC_IE_TRANDONE RW		All block transfer complete interrupt enabled: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
5	RB_EMMC_IE_TRANERR RW		Block transfer CRC error interruption: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
4	RB_EMMC_IE_DATTMO	RW	Data timeout interruption: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
3	RB_EMMC_IE_CMDDONE RW		Command completion interrupted: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
2	RB_EMMC_IE_REIDX_ER RW		Response index verification error interrupted: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
1	RB_EMMC_IE_RECRC_WR RW		CRC check error response interruption: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
0	RB_EMMC_IE_RE_TMOUT RW		Command response timeout interrupted: 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0

33.3.9 Data Block DMA Start Address Register (R32_EMMC_DMA_BEG1) Offset Address: 0x2C

31	30	29 28 27 26 25 24 23 22 21	2019	18 17 16
RB_EMMC_DMAAD1_MASK[31:16]				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RB_EMMC_DMAAD1_MASK[15:0]															

Bit	name	access		Reset value
[31:0]	RB_EMMC_DMAAD1_MASK [31:0]	RW	Describes the starting address of the read/write data buffer, with the lower 4 bits fixed at 0 (16 (Byte alignment). Once this register is set, the system will begin booting. Receive or send.	0

Note: This register stores the starting address of the read data when reading from it. When writing data to the card, SD (SRAM) stores the starting address of the data to be written. Addressing is performed in the region.

SRAM

If multiple consecutive read/write SD operations are performed, then after a single block transfer is completed (RB_EMMC_IF_BKGAP), the user can proceed as needed. The DMA address should be writable through the DMA_BEG1 register. However, the DMA address must not be changed during the transfer process, otherwise it may cause data loss. Error in data storage. When performing multiple consecutive block writes, after each block transfer is completed, the R32_EMMC_WRITE_CONT or DMA_BEG1 register needs to be written. This method initiates continued write operations. Multi-block reads do not require this.

33.3.10 Transport Block Configuration Register (R32_EMMC_BLOCK_CFG)

Offset address: 0x30

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
Reserved			RB_EMMC_BKSIZE_MASK		

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

Bit	name	access	describe	Reset value
[31:28] Reserved [27:16]		RO is reserved.		0
RB_EMMC_BKSIZE_MASK	RW	Single block transfer size (1-2048 bytes).		0
[15:0]	RB_EMMC_BKNUM_MASK	RW	The number of blocks to be transferred in this DMA operation (1 to 65535 blocks), internally Automatically clears to zero; if the number of blocks is not zero, a receive or transmit signal is generated.	0

33.3.11 Transfer Mode Register (R32_EMMC_TRAN_MODE)

Offset address: 0x34

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

15	14	13 12	11	10 9 8	7	6	5	4	3	2	1	0
Reserved	RB_EMMC_DMATN_CNT				Reserved			RB_EMMC_AUTOGAPSTOP	Reserved	RB_EMMC_MODE_BOOT	RB_EMMC_GAP_STOP	RB_EMMC_DMA_DIR

Bit	name	access	describe	Reset value
[31:21] Reserved		RO is reserved.		0
[20:19]	RB_SW	RW	Clock switching in DDR mode: 00: Clock does not switch; 01: Automatic switching; 10: Manual switching; 11: Retained.	0
18	RB_CARE_NEG	RW	In DDR mode, data is checked on the falling edge of the clock: 1: Inspection; 0: No check.	0
17	RB_DDR_MODE	RW	Enable DDR mode: 1: Enable; 0: Off.	0
16	RB_EMMC_DULEDMA_EN	RW	Enable DMA double buffering: 1: Enable; 0: Off.	0
15	Reserved	RO is reserved.		0
[14:8]	RB_EMMC_DMATN_CNT	RW	In double-buffered mode, sets the block count value for buffer switching.	0x10
[7:5] Reserved		is reserved; bit 5 must be written with		010b
4	RB_EMMC_AUTOGAPSTOP	RW	0. Hardware automatic data single-block clock stop enable: 1: Enable;	0

			0: Off.	
3	Reserved	RW must be written		0
2	RB_EMMC_MODE_BOOT	RW	as 0. Set eMMC card transfer mode: 1: Boot mode; 0: Normal mode. Note: This is only for eMMC cards.	0
1	RB_EMMC_GAP_STOP	RW	The data block completes clock-stop mode. When reading block data, this bit... Setting it to 1 will cause the hardware to automatically shut down after counting the samples of one data block. Clock output, initiating a new block transfer, requires software clearing to 0, delay. After 1 SCLK time, reset to 1 and start counting again. If the RB_EMMC_AUTOGAPSTOP function is enabled, this bit will... Set the component to 1, and the software only needs to be cleared to 0.	0
0	RB_EMMC_DMA_DIR	RW	DMA transfer direction: 1: Controller to SD; 0: SD to controller.	0

33.3.12 Clock Configuration Register (R16_EMMC_CLK_DIV) Offset

Address: 0x38- bit

Name		access	describe	Reset value
[15:11] Reserved		RO is reserved.		0
10	RB_EMMC_PHASEINV	RW SDCK clock output phase inversion.		0
9	RB_EMMC_CLKMode	RW	Clock frequency mode selection bit: 1: High-speed mode, 25M-200MHz; 0: Low speed mode, below 5MHz.	1
8	RB_EMMC_CLKOE	RW	SD physical clock signal line output control bit: 1: Turn on and output the communication clock; 0: Off.	0
[7:5] Reserved		RO is reserved.		0
[4:0] RB_EMMC_DIV_MASK		RW	SD controller clock (SDCLK) division factor: When RB_EMMC_CLKMode = 1, then SDCLK = SYSPLL_SEL/RB_EMMC_DIV_MASK; When RB_EMMC_CLKMode = 0, then SDCLK = SYSPLL_SEL/RB_EMMC_DIV_MASK/64. The minimum value is 2; writing 1 is equivalent to turning off the SDC analog clock.	10011b

Note: In slave mode, RB_EMMC_DIV_MASK needs to be configured to the minimum value of 2. This value does not affect the input clock value.

33.3.13 Data Block DMA Start Address Register (R32_EMMC_DMA_BEG2) Offset Address:

0x3C

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
RB_EMMC_DMAAD2_MASK[31:16]														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RB_EMMC_DMAAD2_MASK[15:0]															

Bit	name	access	describe	Reset value
[31:0]	RB_EMMC_DMAAD2_MASK [31:0]	RW	The starting address of the read/write data buffer, with the lower 4 bits fixed at 0 (16 Byte alignment).	0

Note: This register is in use from SD When reading data, the system stores the starting address SRAM SD The starting address in the memory. When writing data to the card. of the data being read and the starting address of the data to be written. Addressing occurs within a region. SRAM

33.3.14 Data Output Delay Register (R32_EMMC_TUNE_DATO) Offset

Address: 0x40

31 30 29 28 27 26 25 24 23 22 21 2019 18 17 16			
RB_TUNNE_DAT7_O[3:0]	RB_TUNNE_DAT6_O[3:0]	RB_TUNNE_DAT5_O[3:0]	RB_TUNNE_DAT4_O[3:0]

15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
RB_TUNNE_DAT3_O[3:0]	RB_TUNNE_DAT2_O[3:0]	RB_TUNNE_DAT1_O[3:0]	RB_TUNNE_DAT0_O[3:0]

Bit	name	access	describe	Reset value
[31:28]	RB_TUNNE_DAT7_O [3:0]	RW	The delay of the RW D7 output.	0
[27:24]	RB_TUNNE_DAT6_O [3:0]	RW	The delay of the RW D6 output.	0
[23:20]	RB_TUNNE_DAT5_O [3:0]	RW	The delay of the RW D5 output.	0
[19:16]	RB_TUNNE_DAT4_O [3:0]	RW	The delay of the RW D4 output.	0
[15:12]	RB_TUNNE_DAT3_O [3:0]	RW	The delay of the RW D3 output.	0
[11:8]	RB_TUNNE_DAT2_O [3:0]	RW	The delay of the RW D2 output.	0
[7:4]	RB_TUNNE_DAT1_O [3:0]	RW	The delay of the RW D1 output.	0
[3:0]	RB_TUNNE_DAT0_O [3:0]	RW	The delay of the RW D0 output.	0

Note: Output delay = 0.1ns * RB_TUNNE_DATx_O.

33.3.15 Data Input Delay Register (R32_EMMC_TUNE_DATI) Offset Address:

0x44

31 30 29 28 27 26 25 24 23 22 21 2019 18 17 16			
RB_TUNNE_DAT7_I[3:0]	RB_TUNNE_DAT6_I[3:0]	RB_TUNNE_DAT5_I[3:0]	RB_TUNNE_DAT4_I[3:0]

15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
RB_TUNNE_DAT3_I[3:0]	RB_TUNNE_DAT2_I[3:0]	RB_TUNNE_DAT1_I[3:0]	RB_TUNNE_DAT0_I[3:0]

Bit	name	access	describe	Reset value
[31:28]	RB_TUNNE_DAT7_I	RW	The delay of the RW D7 input.	0

	[3:0]			
[27:24]	RB_TUNNE_DAT6_I [3:0]	The delay of the RW D6 input.		0
[23:20]	RB_TUNNE_DAT5_I [3:0]	The delay of the RW D5 input.		0
[19:16]	RB_TUNNE_DAT4_I [3:0]	The delay of the RW D4 input.		0
[15:12]	RB_TUNNE_DAT3_I [3:0]	The delay of the RW D3 input.		0
[11:8]	RB_TUNNE_DAT2_I [3:0]	The delay of the RW D2 input.		0
[7:4]	RB_TUNNE_DAT1_I [3:0]	The delay of the RW D1 input.		0
[3:0]	RB_TUNNE_DAT0_I [3:0]	The delay of the RW D0 input.		0

Note: Input delay = 0.1ns * RB_TUNNE_DATx_I.

33.3.16 Clock and Command Delay Register (R32_EMMC_TUNE_CLK_CMD)

Offset Address: 0x48

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
Reserved			RB_TUNNE_CMD_I[3:0]	RB_TUNNE_CMD_O[3:0]	

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

Bit	name	access	describe	Reset value
[31:24] Reserved		RO is reserved.		0
[23:20]	RB_TUNNE_CMD_I [3:0]	RW	CMD input delay: Delay value = 0.2ns * RB_TUNNE_CMD_I.	0
[19:16]	RB_TUNNE_CMD_O [3:0]	RW	CMD output delay: Delay value = 0.2ns * RB_TUNNE_CMD_O.	0
[15:8] Reserved		RO is reserved.		0
[7:4]	RB_TUNNE_CLK_I [3:0]	RW	Delay of CLK input: Delay value = 0.2ns * RB_TUNNE_CLK_I. Note: The delay of this clock input only delays eMMC Card The strobe signal field is invalid when using a card that does not have this signal.	0
[3:0]	RB_TUNNE_CLK_O [3:0]	RW	CLK output delay: Delay = 0.2ns * RB_TUNNE_CLK_O.	0

Chapter 34 Programmable Protocol I/O Microcontrollers (PIOC)

The chip embeds a programmable protocol I/O microcontroller based on a single-clock-cycle dedicated reduced instruction set RISC. The kernel, running at the system clock frequency, has a 2K instruction program ROM, 49 SFR registers, and a PWM timer/counter, supporting 2 Protocol control of I/O pins.

34.1 Main Features

ÿ RISC core, optimized single-cycle bit manipulation instruction set, fully static instruction set

It reuses 4KB of system SRAM as a 2KB program ROM, supporting program pause and dynamic loading.

It provides one bidirectional and one unidirectional register of 33 bytes each, and offers 6 levels of independent stack.

Supports two general-purpose bidirectional I/O protocol controls and supports input level change detection.

By dynamically loading different protocol programs, it can support single-wire and two-wire interfaces with various protocol specifications.

Supports 1024-level serial connection of single-line ARGB chips .

34.2 Instruction and Program Space and Stack

34.2.1 The instruction set

eMCU adopts a reduced instruction set RISC8B core with a data width of 8 bits and an instruction width of 16 bits.

There are a total of 66 instructions. Except for jump instructions (which are two clock cycles) and program space read/write instructions (which are two clock cycles), all other instructions are single clock cycles. Clock cycle. For more information, please refer to the RISC8B core instruction set and assembler tool documentation CHRISC8B.PDF.

34.2.2 Program ROM The eMCU's

program ROM is multiplexed from 4K bytes of system SRAM. When SB_MST_CLK_GATE is 0, it is dedicated to the host-side system SRAM. The standard SRAM is used as the program ROM for the eMCU when SB_MST_CLK_GATE is 1.

The eMCU's program space is 2048 words, with addresses ranging from 0x0000 to 0x07FF. Instructions at address 0 in the program ROM have... For special purposes, this instruction will not be executed.

The eMCU chip supports read operations on the program space. The steps are: first, write the lower 8 bits of the target address into SFR_INDIR_ADDR. The high 3 bits of the target address are written into register A, and then the RDCODE instruction is executed, which can read 16 bits of data from the program space at once. It consists of two bytes, a low byte and a high byte. The low byte is returned in register A, and the high byte is returned in SFR_INDIR_ADDR. This operation can be used... Look up the table using double bytes.

eMCU chips do not support write operations to the program space.

34.2.3 The eMCU has a 6-

level deep stack memory with an 11-bit data width. This is used to save the program return address during subroutine calls. It can also be used with the PUSHAS instruction to save variables and state data during program execution to the stack.

34.2.4 Sleep and Wake-up When

SB_MST_CLK_GATE is 0, the eMCU program will pause execution. After executing the SLEEP or SLEEPX instruction, the eMCU will enter sleep/wake-up mode. Enter sleep mode. The eMCU is a fully static design, so pausing the program has the same effect as sleeping.

The eMCU supports two wake-up methods after sleep: wake-up via level change of the pin that enables level change detection, and wake-up via SB_MST_CLK_GATE. The change from 0 to 1 awakens the user.

34.2.5 The WAITB instruction supports

eight types of event waits. When an event is invalid, the WAITB wait continues indefinitely, and the program counter (PC) remains constant.

Change; upon detecting the target event, exit the wait and execute the next instruction. Optional timer timeout reset can be enabled during the wait process.

The instruction parameter WB_DATA_SW_MR_0 indicates that the program will wait until SB_DATA_SW_MR is 0 before exiting; if it is already 0, it will exit directly.

The instruction parameter WB_BIT_CYC_TAIL_1 indicates that the program will wait until SB_BIT_CYC_TAIL is 1 before exiting; if it is already 1, it will exit directly. The

instruction parameter WB_PORT_I0_FALL indicates that the program will wait until a falling edge is detected in SB_PORT_IN0 before exiting; if a falling edge has been detected previously...

It will also exit directly upon arrival, clearing the falling edge detection record upon exit;

The instruction parameter WB_PORT_I0_RISE indicates that the program will wait until a rising edge is detected in SB_PORT_IN0 before exiting, as previously recorded.

It will exit directly upon arrival, clearing the rising edge detection record upon exit;

The instruction parameter WB_DATA_MW_SR_1 indicates that the program will wait until SB_DATA_MW_SR is 1 before exiting; if it is already 1, it will exit directly.

The instruction parameter WB_PORT_XOR1_1 indicates that the program will wait until SB_PORT_XOR1 is 1 before exiting; if it is already 1, it will exit directly.

The instruction parameter WB_PORT_XOR0_0 indicates that the program will wait until SB_PORT_XOR0 is 0 before exiting; if it is already 0, it will exit directly.

The instruction parameter WB_PORT_XOR0_1 indicates that the program will wait until SB_PORT_XOR0 is 1 before exiting; if it is already 1, it will exit directly.

34.2.6 Bit Transfer Instructions Bit

transfer instructions support two bit manipulation registers: BP1F and BG1F correspond to register SFR_INDIR_ADDR, and BP2F and BG2F correspond to...

The register should be SFR_DATA_EXCH.

The bit transfer instructions BP1F and BP2F support four independent bit outputs: parameter 0# BIO_FLAG_C corresponds to the SB_FLAG_C bit, parameter 1#

BO_BIT_TX_O0 corresponds to SB_BIT_TX_O0, parameter 2# BO_PORT_OUT0 corresponds to SB_PORT_OUT0, and parameter 3# BO_PORT_OUT1 corresponds to SB_PORT_OUT1.

The bit transfer instructions BG1F

and BG2F support four independent bit inputs: parameter 0# BIO_FLAG_C corresponds to the SB_FLAG_C bit, parameter 1# BI_BIT_RX_I0 corresponds to SB_BIT_RX_I0,

parameter 2# BI_PORT_IN0 corresponds to SB_PORT_IN0, and parameter 3# BI_PORT_IN1 corresponds to SB_PORT_IN1. The BCTC bit transfer instruction supports four independent

bit inputs: 0# parameter

BI_C_XOR_IN0 updates bit SB_FLAG_C to the XOR result of SB_FLAG_C and SB_PORT_IN0; 1# parameter BI_BIT_RX_I0 corresponds to SB_BIT_RX_I0; 2# parameter

BI_PORT_IN0 corresponds to SB_PORT_IN0; and 3# parameter BI_PORT_IN1 corresponds to SB_PORT_IN1.

34.3 Data Space, Registers, and Addressing

The eMCU's data space includes 49 Special Function Registers (SFRs), addressable via 8-bit addresses ranging from 0x00 to 0x3F. All registers have a data width of 1 byte

(8 bits). Some SFRs are read-write or read-only on the host side, supporting 8-bit, 16-bit, or 32-bit

read/write widths.

It has write priority; when the eMCU and the host write to the same register at the same time, the eMCU write operation is automatically discarded.

The addressing modes of eMCU include: immediate addressing, immediate fast addressing, normal direct addressing, extended direct addressing, and indirect addressing.

Bit addressing, the last four are used to address registers.

Indirect addressing has an addressing range of 0x000-0x0FF, covering all registers of the eMCU. The eMCU provides two sets of indirect addressing registers, each consisting

of an address register and a data read/write port. First, the address of the destination register is written to the address register, and then the destination register can be read or

written through the data read/write port.

Extended direct addressing has an addressing range of 0x000-0x1FF, where the instruction directly provides a 9-bit register address. It applies only to the following two conditions.

Instructions for directly reading and writing registers: MOV register, A or F instruction, MOVA register instruction.

The addressing range of normal direct addressing is 0x000-0x0FF. The instruction directly provides the 8-bit register address and is applicable to all direct addressing

instructions except for the extended direct addressing instructions mentioned above, such as CLR register, ADD register, BS register, bit, etc. Indirect addressing should be used

when a larger addressing range is required.

Immediate addressing is used to quickly write operands from the instruction code to the destination register without using A as an intermediary. MOVA1F is used to quickly

set register #1 (SFR_PORT_DIR), and MOVA2F is used to quickly set register #2 (SFR_PORT_IO). MOVA1P and MOVA2P, through indirect addressing, are applicable to the fast setting

of all registers, equivalent to immediate addressing followed by indirect addressing within a single cycle.

Bit addressing provides a 3-bit address directly from the instruction, while register addressing can be achieved using either direct addressing or indirect addressing.

They respectively implement the addressing range of 0x000-0x0FF, thus allowing any bit of any register to be addressed.

34.4 Special Function Registers (SFRs)

Some SFRs or some bits are not actually implemented; they are reserved bits. When read, they are 0, and when written, they must retain the original value or be written as 0.

name	Access address	describe	Host side reading Write	Reset value
R8_SFR_INDIR_PORT	0x40025C00 Indirect Addressing Data Read/Write Port UUUUUUUU 0x40025C01			XXXXXXXX
R8_SFR_INDIR_PORT2	Indirect Addressing 2 Data Read/Write Port UUUUUUUU 0x40025C02 Program			XXXXXXXX
R8_SFR_PRG_COUNT	Counter (PC) Low Byte UUUUUUUU 0x40025C03 Status Register 0x40025C04			00000000
R8_SFR_STATUS_REG	Indirect Addressing Address Register		UUUUSUSU	0000-0-0
R8_SFR_INDIR_ADDR	0x40025C05 Timer 0 Count Register 0x40025C06 Timer Control Register 0x40025C07			XXXXXXXX
R8_SFR_TMR0_COUNT	Timer 0 Initial Value Register 0x40025C08 Code Bit Period Register 0x40025C09			00000000
R8_SFR_TIMER_CTRL	Indirect Addressing 2 Address Register 0x40025C0A		RRRRRRRR	00000000
R8_SFR_TMR0_INIT	Port Direction Setting Register 0x40025C0B Port input/output register 0x40025C0C,			00000000
R8_SFR_BIT_CYCLE	encoding bit configuration register 0x40025C1C,		WWWWWWWW	00000000
R8_SFR_INDIR_ADDR2	system configuration register (eMCU) read/write and host read-only register.			00000000
R8_SFR_PORT_DIR			RRRRRRRR	00000000
R8_SFR_PORT_IO			RRRRRRRR	XXXXXX00
R8_SFR_BIT_CONFIG			WWWRRRRR	00010000
R8_SFR_SYS_CFG			RRRWWWWW	00000000
R8_SFR_CTRL_RD	0x40025C1D	Instrument	RRRRRRRR	00000000
R8_SFR_CTRL_WR	0x40025C1E	Host read/write and eMCU read-only register Instrument	WWWWWWWW	00000000
R8_SFR_DATA_EXCH	0x40025C1F Data Exchange Register;		WWWWWWWW	00000000
R8_SFR_DATA_REG0	0x40025C20 Data Register 0; 0x40025C21		WWWWWWWW	00000000
R8_SFR_DATA_REG1	Data Register 1; 0x40025C22 Data Register		WWWWWWWW	00000000
R8_SFR_DATA_REG2	2; 0x40025C23 Data Register 3; 0x40025C24		WWWWWWWW	00000000
R8_SFR_DATA_REG3	Data Register 4; 0x40025C25 Data Register		WWWWWWWW	00000000
R8_SFR_DATA_REG4	5; 0x40025C26 Data Register 6; 0x40025C27		WWWWWWWW	00000000
R8_SFR_DATA_REG5	Data Register 7; 0x40025C28 Data Register		WWWWWWWW	00000000
R8_SFR_DATA_REG6	8; 0x40025C29 Data Register 9; 0x40025C2A		WWWWWWWW	00000000
R8_SFR_DATA_REG7	Data Register 10; 0x40025C2B Data		WWWWWWWW	00000000
R8_SFR_DATA_REG8	Register 11; 0x40025C2C Data Register		WWWWWWWW	00000000
R8_SFR_DATA_REG9	12; 0x40025C2D Data Register 13;		WWWWWWWW	00000000
R8_SFR_DATA_REG10	0x40025C2E Data Register 14 0x40025C2F		WWWWWWWW	00000000
R8_SFR_DATA_REG11	Data Register 15 0x40025C30 Data Register		WWWWWWWW	00000000
R8_SFR_DATA_REG12	16 0x40025C31 Data Register 17		WWWWWWWW	00000000
R8_SFR_DATA_REG13			WWWWWWWW	00000000
R8_SFR_DATA_REG14			WWWWWWWW	00000000
R8_SFR_DATA_REG15			WWWWWWWW	00000000
R8_SFR_DATA_REG16			WWWWWWWW	00000000
R8_SFR_DATA_REG17			WWWWWWWW	00000000

R8_SFR_DATA_REG18	Data registers 0x40025C32 (Data Register 18),		WWWWWWWW	00000000
R8_SFR_DATA_REG19	0x40025C33 (Data Register 19), 0x40025C34		WWWWWWWW	00000000
R8_SFR_DATA_REG20	(Data Register 20), 0x40025C35 (Data Register		WWWWWWWW	00000000
R8_SFR_DATA_REG21	21), 0x40025C36 (Data Register 22), 0x40025C37		WWWWWWWW	00000000
R8_SFR_DATA_REG22	(Data Register 23), 0x40025C38 (Data Register		WWWWWWWW	00000000
R8_SFR_DATA_REG23	24), 0x40025C39 (Data Register 25), 0x40025C3A		WWWWWWWW	00000000
R8_SFR_DATA_REG24	(Data Register 26), 0x40025C3B (Data Register		WWWWWWWW	00000000
R8_SFR_DATA_REG25	27), 0x40025C3C (Data Register 28), 0x40025C3D		WWWWWWWW	00000000
R8_SFR_DATA_REG26	(Data Register 29), 0x40025C3E (Data Register		WWWWWWWW	00000000
R8_SFR_DATA_REG27	30), and 0x40025C3F (Data Register 31) are all		WWWWWWWW	00000000
R8_SFR_DATA_REG28	represented in binary. The bit values are		WWWWWWWW	00000000
R8_SFR_DATA_REG29	explained below:		WWWWWWWW	00000000
R8_SFR_DATA_REG30			WWWWWWWW	00000000
R8_SFR_DATA_REG31			WWWWWWWW	00000000

0: Always 0 after reset;

1: Always 1 after reset;

X: Reset does not affect the data, and the initial data value is uncertain;

-: The data is always cleared to 0 after power-on reset. System reset or host forced reset does not affect the data.

The instructions for host-side read/write operations are as follows:

W: The bit that can be read or written;

R: Read-only bit;

U: Invisible bit, cannot be read or written;

S: Read-only swap bit. The SFR_TIMER_CTRL visible on the host side differs from the eMCU side portion, where SB_TMR0_ENABLE and

The two bits SB_TMR0_OUT_EN are replaced by SB_GP_BIT_Y and SB_GP_BIT_X respectively.

34.4.1 Indirect Addressing Data Read/Write Port (SFR_INDIR_PORT)

34.4.2 Address Register for Indirect Addressing (SFR_INDIR_ADDR)

34.4.3 Data read/write port for indirect addressing 2 (SFR_INDIR_PORT2)

34.4.4 Address register for indirect addressing 2 (SFR_INDIR_ADDR2)

These are two sets of indirect addressing registers, each consisting of an address register and a data read/write port. First...

Write the address of the destination register to the address register, and then read or write the destination register through the read/write data port.

Read/write indirect addressing data read/write port SFR_INDIR_PORT , It means reading and writing address registers that are addressed indirectly.

SFR_INDIR_ADDR specifies the destination register at the specified address. Reading and writing data via indirect addressing port SFR_INDIR_PORT2 involves reading and writing...

The destination register whose address is specified by the indirect addressing address register SFR_INDIR_ADDR2.

The dedicated instruction MOVIP is used to load an 8-bit destination address into SFR_INDIR_ADDR, and SFR_INDIR_ADDR can also be loaded by other instructions.

Non-dedicated instructions can be modified independently.

The dedicated instruction MOVIA is used to load an 8-bit destination address into SFR_INDIR_ADDR2, and SFR_INDIR_ADDR2 can also be loaded by it.

It is not modified independently by dedicated instructions.

The address register SFR_INDIR_ADDR2 for indirect addressing 2 has an auto-increment feature. This is achieved when executing "MOV SFR_INDIR_PORT2,A".

When a read instruction is executed or a write instruction "MOVA SFR_INDIR_PORT2" is executed on RAM, SFR_INDIR_ADDR2 will be executed after the instruction is completed.

Automatically increment by 1.

The special instructions MOVA1P and MOVA2P are used to directly and quickly write the operands from the instruction code into the destination register, equivalent to writing them within a single cycle.

First use immediate addressing, then use indirect addressing.

34.4.5 Status Register (SFR_STATUS_REG)

The status register contains the ALU's result status, general bit variables, and stack usage flags. After performing arithmetic or logical operations,

The eMCU will set corresponding status bits based on the result, so that the program can make a judgment before taking further action. General-purpose bit variables can be set by the application program.

Define its purpose.

The timer timeout reset is used to monitor the timeout of the WAITB wait instruction, disabling SB_TMR0_ENABLE and enabling SB_EN_TOUT_RST.

Subsequently, during the execution of the WAITB instruction, the timer counts, and when it overflows, SB_TMR0_CYCLE becomes 1, causing a timeout reset. Each time...

The WAITB instruction exits the WAITB wait if the condition is met, and the timer initial value is automatically reloaded during the wait. This applies when SB_TMR0_ENABLE is 0.

The execution period of non-WAITB instructions is not counted, meaning that timer timeout reset only applies to WAITB instructions.

Bit	name	describe	Reset value
7	Reserved		0
6	Reserved		0
5	SB_STACK_USED	Reserved. Reserved. Current stack uses flags: 1: Already pushed onto the stack; 0: Not yet used or has been completely popped from the stack.	0
4	SB_EN_TOUT_RST	Timer timeout reset enable: 1: Allows resetting the eMCU when the timer overflows; 0: Disabled.	0
3	SB_GP_BIT_Y	The generic bit variable Y is defined and used by the application. The value is 0 after power-on reset and is not affected by system reset or host forced reset.	-
2	SB_FLAG_Z	ALU zero flag, check if the result is 00H: 1: The result is 0; 0: The result is not 0.	0
1	SB_GP_BIT_X	The generic bit variable X is defined and used by the application. The value is 0 after power-on reset and is not affected by system reset or host forced reset.	-
0	SB_FLAG_C	ALU carry flag, indicating whether the result is a carry or generated by a shift: 1: The result generates a carry; 0: The result did not carry over.	0

34.4.6 Low byte of the program counter PC (SFR_PRG_COUNT)

The eMCU's program counter (PC) is 11 bits wide. The lower 8 bits of the PC can be modified by reading and writing SFR_PRG_COUNT, but...

This affects the high 3 bits of the PC, allowing for short jumps and table lookups within a 256-byte range.

34.4.7 Port Direction Setting Register (SFR_PORT_DIR) The following table

describes the port direction setting register SFR_PORT_DIR.

Bit	name	describe	Reset value
7	SB_PORT_MOD3	Pin-mode control, with the host side defining the application.	0
6	SB_PORT_MOD2		0
5	SB_PORT_MOD1		0
4	SB_PORT_MOD0		0

3	SB_PORT_PU1	IO1 pin pull-up enable: 1: Permission; 0: Disabled.	0
2	SB_PORT_PU0	IO0 pin pull-up enable: 1: Permission; 0: Disabled.	0
1	SB_PORT_DIR1	IO1 pin direction control: 1: Output; 0: Input.	0
0	SB_PORT_DIR0	IO0 pin direction control: 1: Output; 0: Input.	0

34.4.8 Port Input/Output Register (SFR_PORT_IO) The port contains two

bidirectional input/output pins, each of which can be independently set to either input direction or output direction by the respective direction setting bit SB_PORT_DIR.

Output direction: If it's the output direction, the data in SB_PORT_OUT will be output to the pin. The default direction after reset is input.

SB_PORT_IN is used to obtain the current pin input level (when SB_PORT_IN_EDGE=0) or the pin input level of the previous cycle.

(When SB_PORT_IN_EDGE=1), the latter is about half a clock cycle ahead of the former.

Both pins have level change detection functionality. When the input level is inconsistent with the data in the output data register SB_PORT_OUT, the system will detect the level change.

When this happens, the corresponding SB_PORT_XOR becomes 1.

The IO0 pin has more available input commands and methods, making it more suitable for input or bidirectional data compared to IO1.

The following table describes the port input/output register SFR_PORT_IO.

Bit	name	describe	Reset value
7	SB_PORT_IN_XOR	The result of XORing the input states of IO1 and IO0 pins: 1: IO1 and IO0 have different voltage levels; 0: IO1 and IO0 have the same voltage level.	X
6	SB_BIT_RX_I0	The bit reception status of the IO0 pin is decoded data.	X
5	SB_PORT_IN1	IO1 pin input status (refer to SB_PORT_IN_EDGE).	X
4	SB_PORT_IN0	IO0 pin input status (refer to SB_PORT_IN_EDGE).	X
3	SB_PORT_XOR1	The XOR result of the IO1 pin input state and SB_PORT_OUT1: 1: The IO1 level is different from SB_PORT_OUT1 and is used to detect level changes; 0: IO1 level SB_PORT_IN1 is the same as SB_PORT_OUT1.	X
2	SB_PORT_XOR0	The result of XORing the IO0 pin input state with SB_PORT_OUT0: 1: The IO0 level is different from SB_PORT_OUT0 and is used to detect level changes; 0: The IO0 level is the same for SB_PORT_IN0 and SB_PORT_OUT0.	X
1	SB_PORT_OUT1	IO1 pin output data: 1: High output; 0: Low output.	0
0	SB_PORT_OUT0	IO0 pin output data: 1: High output; 0: Low output. The timer output signal takes priority; when SB_TMR0_OUT_EN is 1, the output... SB_TMR0_CYCLE or its divided frequency signal.	0

34.4.9 Timer 0 Count Register (SFR_TMR0_COUNT)

34.4.10 Initial value register for Timer 0 (SFR_TMR0_INIT)

34.4.11 Timer Control Register (SFR_TIMER_CTRL)

Timer 0 is an 8-bit counter. The count value is stored in the SFR_TMR0_COUNT register, which supports direct...
Write data to load the new value. The clock frequency of Timer 0 is selected by SB_TMR0_FREQ. Timer 0 can operate as a timer.
Mode or pulse width modulator mode.

In timer mode, the initial count value of Timer 0 is pre-stored in the initial value register SFR_TMR0_INIT.
The value always counts from the initial value to 0FFH. When the count overflows from 0FFH, Timer 0 automatically resets from the initial value register SFR_TMR0_INIT.
The initial value is reloaded, and then the count starts from the initial value and moves towards 0FFH. When the count value reaches 0FFH from 0FEH, Timer 0...
The counting cycle signal SB_TMR0_CYCLE is automatically set to 1; when the count value overflows from 0FFH and returns to the initial count value, the timer is set to 1.
Timer 0 automatically clears the counting cycle signal SB_TMR0_CYCLE to 0. The frequency of SB_TMR0_CYCLE is (SB_TMR0_FREQ clock).
The frequency is ()/(256-SFR_TMR0_INIT), and its duty cycle is 1/(256-SFR_TMR0_INIT), meaning there is one clock cycle in each period.
The value of SB_TMR0_CYCLE is 1.

In pulse width modulator mode, Timer 0 always counts from 00H to 0FFH, then overflows from 0FFH and starts counting again.
A new count begins at 00H, with a period of always 256 clock cycles. The count continues until the initial value in the Timer 0 initial value register SFR_TMR0_INIT is reached.
When the count value is within the specified range, Timer 0 automatically sets the counting cycle signal SB_TMR0_CYCLE to 1; when the count value overflows from 0FFH...
At 00H, Timer 0 automatically clears the counting cycle signal SB_TMR0_CYCLE to 0. The duty cycle of SB_TMR0_CYCLE is (255-
SFR_TMR0_INIT)/256.

SB_TMR0_OUT_EN is used to set whether the internal signal SB_TMR0_CYCLE or its divided-by-two signal is output to the IO0 pin.
When the signal is 1, the pin outputs a high level; otherwise, it outputs a low level. If the count cycle signal output is enabled, then in timer mode,
The divided-by-two signal of SB_TMR0_CYCLE (frequency reduced to half and duty cycle to 50%) will be output through the IO0 pin; in pulse width modulation
In controller mode, the SB_TMR0_CYCLE signal will be output through the IO0 pin.
The following table describes the timer control register SFR_TIMER_CTRL.

Bit	name	describe	Reset value
7	SB_EN_LEVEL1	A change in the IO1 pin level activates the interrupt flag and wake-up enable: 1: Enables activation of interrupt flags and wake-up; 0: Not activated, not woken up.	0
6	SB_EN_LEVEL0	A change in the IO0 pin level activates the interrupt flag and wake-up enable: 1: Enables activation of interrupt flags and wake-up; 0: Not activated, not woken up.	0
5	SB_TMR0_ENABLE	Timer 0 count enable: 1: Counting is allowed; 0: Counting is prohibited.	0
4	SB_TMR0_OUT_EN	Timer 0's count cycle signal output is enabled: 1: Output to IO0 pin is allowed; 0: Output is disabled.	0
3	SB_TMR0_MODE	Operating mode of Timer 0: 1: Pulse Width Modulator Mode; 0: Timer mode.	0

[2:0] SB_TMR0_FREQ		Select the clock frequency for Timer 0 (based on a division of the system clock frequency): 000: 1024X; 001:256X; 010:64X; 011:16X; 100:8X; 101:4X; 110:2X; 111:1X.	000

34.4.12 System Configuration Register (SFR_SYS_CFG)

34.4.13 Host read/write to eMCU read-only register (SFR_CTRL_WR)

34.4.14 eMCU Read/Write Host Read-Only Register (SFR_CTRL_RD)

The system configuration register provides some control bits for the host and status bits for bidirectional data exchange.

SB_INT_REQ is fully controlled by the eMCU and is used to request an interrupt or cancel a request from the host, but the host can access it through a read-only register. The spoofing operation of SFR_CTRL_RD clears SB_INT_REQ to 0, and writing to SFR_CTRL_RD by the host has no actual effect on the data therein.

After the eMCU writes SFR_CTRL_RD, SB_DATA_SW_MR is automatically set to 1; after the host reads SFR_CTRL_RD, SB_DATA_SW_MR is automatically cleared. 0.

After the host writes SFR_CTRL_WR, SB_DATA_MW_SR is automatically set to 1. After the eMCU reads SFR_CTRL_WR, SB_DATA_MW_SR is automatically set to 1. ÿÿ0.

SB_MST_CLK_GATE also controls the multiplexing of the eMCU program ROM; when SB_MST_CLK_GATE is 0, it is dedicated to the host side. The system SRAM, totaling 4KB, can dynamically load new programs for the eMCU; when SB_MST_CLK_GATE is 1, it is dedicated to the eMCU. The program ROM contains 2K instructions.

Bits 0 through 4 of the system configuration register are fully controlled by the host and are read-only on the eMCU side.

The following table describes the system configuration register SFR_SYS_CFG.

Bit	name	describe	Reset value
7	SB_INT_REQ	Interrupt request activation bit: 1: Request an interrupt from the host; 0: Cancel the interruption request.	0
6	SB_DATA_SW_MR	SFR_CTRL_RD Waiting to read status bit: 1: The eMCU has been written but not yet read by the host side; 0: The eMCU has not been written to or the host side has already read it.	0
5	SB_DATA_MW_SR	SFR_CTRL_WR read wait status bit: 1: The host side has written the data, but the eMCU side has not yet read it; 0: The host side has not yet written to it or the eMCU side has already read it.	0
4	SB_MST_CFG_B4 Configuration information bit, software-defined purpose		0
3	SB_MST_IO_EN1	IO1 pin mode and output control switch: 1: Controlled by eMCU; 0: Controlled by the host.	0
2	SB_MST_IO_EN0	IO0 pin mode and output control switch: 1: Controlled by eMCU; 0: Controlled by the host.	0
1	SB_MST_RESET	Force eMCU reset: 1: The host computer additionally forces a separate reset of the eMCU; 0: No forced reset; the eMCU resets along with the host.	0

0	SB_MST_CLK_GATE	eMCU global clock control: 1: Enable the eMCU clock; 0: Turn off the eMCU clock, suspend its program, equivalent to sleep.	0

34.4.15 Code Bit Cycle Register (SFR_BIT_CYCLE)

34.4.16 Encoded Bit Configuration Register (SFR_BIT_CONFIG) The eMCU

supports two bit modulation methods: PWM duty cycle modulation and Manchester modulation.

Setting SB_BIT_CYCLE to a value greater than 3 enables encoding and decoding. When SB_BIT_TX_EN=0, reception is initiated from IO0 according to the preset width.

Receive input status; decoded data is stored in SB_BIT_RX_I0. Transmit when SB_BIT_TX_EN=1, and store in SB_BIT_TX_O0.

The raw data to be sent is encoded and automatically sent to SB_PORT_OUT0 for pin output.

SB_BIT_CYC_CNT and SB_BIT_CYC_TAIL identify the periodic state of the encoded bits. When SB_BIT_CYC_TAIL=1, it indicates a positive cycle.

If the bit is in the last 25% of the bit cycle, it means that SB_BIT_RX_I0 has completed the receive decoding and the transmit encoding has been completed and reached SB_PORT_OUT0.

Setting SB_BIT_CYCLE=0 forces SB_BIT_CYC_TAIL to 1, making it easier for Manchester to enable first-bit encoding and decoding.

Setting SFR_BIT_CYCLE will clear the rising and falling edge detection records of IO0, making it easier for the WAITB instruction to wait again.

The following table describes the code bit cycle register SFR_BIT_CYCLE.

Bit	name	This	Reset value
7	SB_BIT_TX_O0	describes the raw bit data of the encoded bits to be sent; this bit is a double-buffered structure. Once a new bit cycle begins, this bit can be preloaded with the data for the next bit.	0
[6:0]	SB_BIT_CYCLE	The width of the encoded bit is set in clock cycles, which is the actual clock speed of the bit. Decrease the number by 1. If the high 5 bits are 0, encoding/decoding is disabled; otherwise, it is enabled.	0000000

The following table describes the encoding bit configuration register SFR_BIT_CONFIG.

Bit	name	describe	Reset value
7	SB_BIT_TX_EN	Enable transmission of encoded bits: 1: Start sending, encode SB_BIT_TX_O0 and send it to SB_PORT_OUT0; 0: Sending is prohibited; receiving is allowed if SB_BIT_CYCLE is valid.	0
6	SB_BIT_CODE_MOD	Modulation method of coded bits: 1: Manchester modulation, 0 does not flip within the period, 1 flips within the period; 0: PWM duty cycle modulation, 0 has a duty cycle of 25%, and 1 has a duty cycle of 75%.	0
5	SB_PORT_IN_EDGE	Pin input level sampling point selection: 1: Sampling at the edge of the previous cycle is equivalent to advancing by about half a cycle; 0: Sampling at the midpoint of the current clock cycle, which is relatively more real-time.	0
4	SB_BIT_CYC_TAIL	Periodic state of the encoded bits: 1: If the bit is in the last 25% of the bit cycle, it means that bit encoding and decoding are complete; 0: Within the first 75% of the position cycle.	1
3	SB_BIT_CYC_CNT6	The periodic timing status of the encoded bits, with bit periodic timing measured in clock cycles. It is a 7-bit counter, SB_BIT_CYC_CNT6ÿ3 are the bits of the counter [6:3].	0
2	SB_BIT_CYC_CNT5		0
1	SB_BIT_CYC_CNT4		0
0	SB_BIT_CYC_CNT3		0

34.4.17 Data Register (SFR_DATA_x) (x=0-31)

SFR_DATA_x are both read-and-write data registers, with host-side write operations taking precedence; they are software-defined for specific purposes.

SFR_DATA_EXCH also supports single-cycle bit transfer instructions.

34.5 Application

The eMCU operates at the same clock frequency as the host computer, facilitating dynamic data exchange. It has a limited number of bit manipulation instructions, single-cycle I/O setup, and single-cycle data acquisition.

I/O status or copied bit data, capable of hardware encoding and decoding PWM or Manchester modulation bit data, suitable for various medium and low speed communications.

The protocol's I/O implementation, and I/O control requiring precise timing.

Chapter 35 Operational Amplifiers (OPA) and Comparators (CMP)

This module contains three independently configurable operational amplifiers (OPA1/OPA2/OPA3) and one independently configurable voltage comparator. (CMP), where the operational amplifier supports low offset input voltage, supports gain selection, and can also be used as a voltage comparator.

The operational amplifier's inputs and outputs can be connected to I/O ports, and the input channels (pins) and gains are selectable. The output channels can be configured to general-purpose I/O ports. An additional internal output channel is directly connected to the comparator input, supporting a high-speed mode, which

can be used to improve the slew rate. Each voltage comparator's inputs and outputs are connected to I/O ports, and the input pins are selectable. The output pins can be configured to general-purpose I/O ports or multiplexed as TIM internal sampling channels (without occupying I/O pins).

35.1 Main Features

• Selectable OPA input/output channels • OPA1
output channel directly connected to comparator input • OPA
supports voltage buffers and programmable gain amplifiers (PGAs) • Selectable CMP input pins
• CMP positive input supports optional
DAC output, negative input supports optional internal self-biasing voltage • CMP supports optional digital filtering and
hysteresis • CMP output pins can be selected as
general-purpose I/O ports or TIM internal sampling channels

35.2 Functional Description

35.2.1 Operational Amplifiers 1/2/3 (OPA1/OPA2/OPA3) OPA1, OPA2, and OPA3

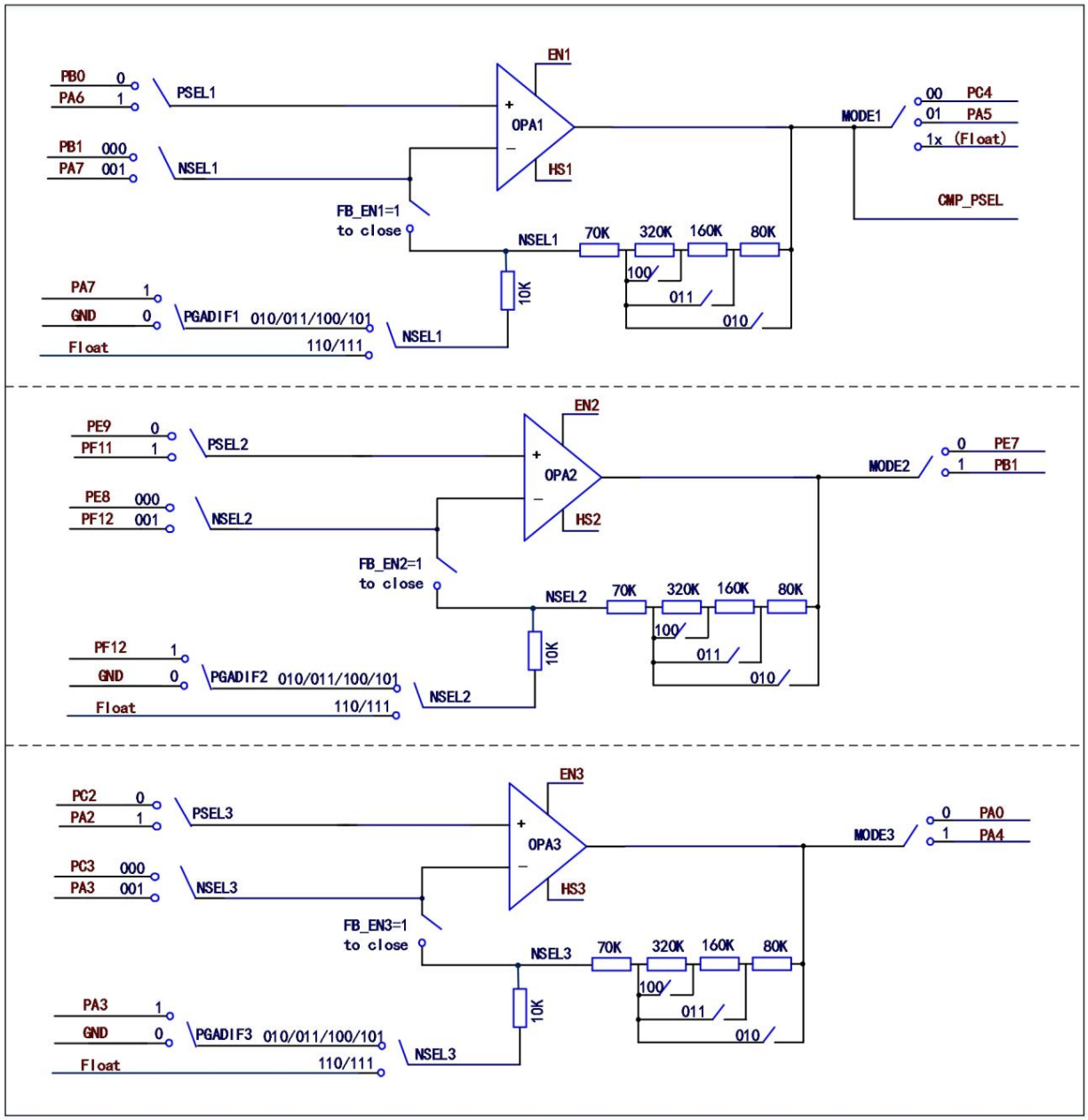
support voltage buffers and programmable gain amplifiers (PGAs). As shown in Figure 35-1, the functions of these three OPAs are basically the same. OPA1, compared to OPA2/OPA3, has an additional output channel connected to the input of the voltage comparator CMP_P2. In the R32_OPA_CTLR1 register: configuring the

EN1 bit enables OPA1; configuring the PSEL1 bit selects the positive input channel of OPA1; configuring the NSEL1 bit selects the negative input channel of OPA1 or selects PGA mode, and sets the internal gain of the OPA; configuring...
The FB_EN1 bit enables PGA mode feedback for OPA1; the PGADIF1 bit enables differential input PGA mode, allowing OPA1 to be used as a PGA in conjunction with NSEL1, with the N terminal connected to PA7; the HS1 bit enables high-speed mode, which can improve the slew rate; the MODE1[1:0] bit can be configured to select the output channel as GPIO or the positive input of the comparator.

In the R32_OPA_CTLR2 register: Configure the EN2 bit to enable OPA2; configure the PSEL2 bit to select the positive input channel of OPA2; configure the NSEL2 bit to select the negative input channel of OPA2 or select PGA mode, and set the internal gain of OPA; configure...
The FB_EN2 bit enables PGA mode feedback for OPA2; the PGADIF2 bit enables differential input PGA mode, allowing OPA2 to be used as a PGA in conjunction with NSEL2, with the N terminal connected to PF12; the HS2 bit enables high-speed mode, which improves the slew rate; and the MODE2 bit selects the output channel.

In the R32_OPA_CTLR3 register: Configure the EN3 bit to enable OPA3; configure the PSEL3 bit to select the positive input channel of OPA3; configure the NSEL3 bit to select the negative input channel of OPA3 or select PGA mode, and set the internal gain of OPA; configure...
The FB_EN3 bit enables PGA mode feedback for OPA3; the PGADIF3 bit enables differential input PGA mode, allowing OPA3 to be used as a PGA in conjunction with NSEL3, with the N terminal connected to PA3; the HS3 bit enables high-speed mode, which improves the slew rate; and the MODE3 bit selects the output channel.

Figure 35-1 Structural diagram of OPA1/OPA2/OPA3



35.2.2 Comparator (CMP) The CMP supports

optional hysteresis characteristics and digital filtering function at the output end. The output filtering is optional. The structure diagram is shown in Figure 35-2.

In the R32_CMP_CTLR register: the EN bit is configured to enable the CMP module; the PSEL[1:0] bits are configured to select the P-side channel.

The input can be via GPIO or connected internally to the OPA operational amplifier; configuring the NSEL[1:0] bits allows selection of the N-terminal channel via GPIO input or...

The chip can be internally connected to the DAC output or optionally has an internal self-biasing voltage; the HYPSEL bit can be configured to select the comparator hysteresis voltage.

Configure the VREF bit to select the internal bias voltage of the CMP; configure the FILT_EN bit to enable the digital filtering function of the CMP output; configure...

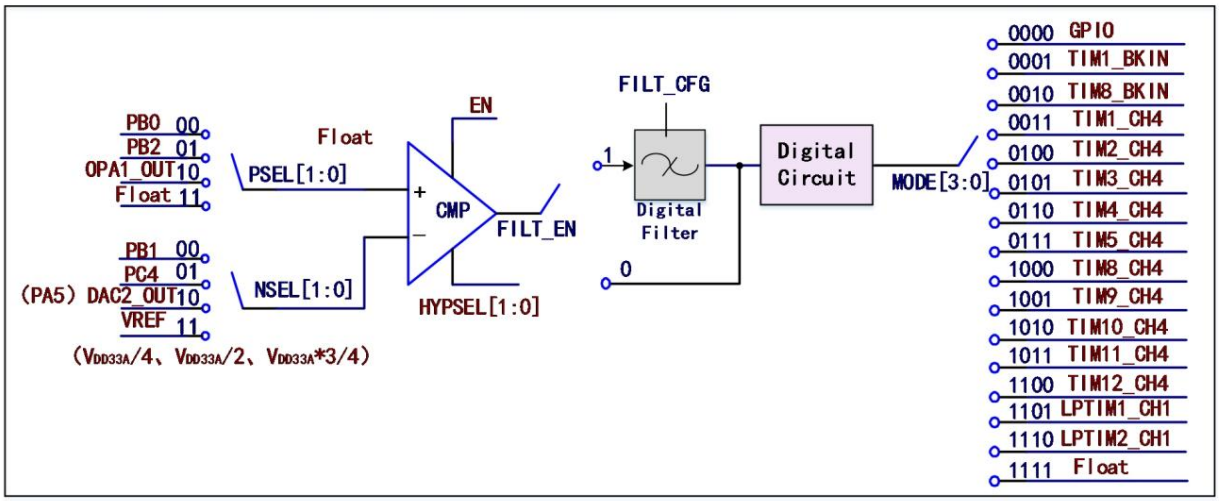
The FILT_CFG[8:0] bits can be used to select the CMP output digital filter sampling interval. The sampling interval setting should be greater than the noise width and less than twice the noise width.

Sound width; configure FILT_BASE[2:0] bits to configure the time base for CMP output digital filtering sampling; configure MODE[3:0] bits to...

Select the CMP2 output to connect internally to IO or connect the output to the timer brake input/timer internal sampling channel.

In the R32_CMP_STATR register: the output result of CMP filtering function being turned on/off can be read through the OUT_FILT bit.

Figure 35-2 CMP Structure Diagram



35.3 Register Description

Table 35-4 List of OPA-related registers

name	Access address	describe	Reset value
R32_OPA_CTLR1	0x40017800	OPA Control Register 1	0x00000076
R32_OPA_CTLR2	0x40017804	OPA Control Register 2	0x00000070
R32_OPA_CTLR3	0x40017808	OPA Control Register 3	0x00000070
R32_CMP_CTLR	0x4001780C	CMP Control Register	0x000000FF
R32_CMP_STATR	0x40017810	CMP Status Register	0x00000000

35.3.1 OPA Control Register 1 (R32_OPA_CTLR1) Offset Address:

0x00

31	30 29 28 27 26 25 24 23 22 21	2019	18	17 16
Reserved				

15	14 13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				HS1	PGADIF1	FB_EN1	Reserved	NSEL1[2:0]			PSEL1	MODE1[1:0]		EN1

Bit	name	access	describe	Reset value
[31:11]	Reserved	RO	is reserved.	0
10	HS1	RW	OPA1 High-Speed Mode Enabled: 1: Enable; 0: Prohibited.	0
9	PGADIF1	RW	OPA1 is used in conjunction with NSEL1 as a PGA, and the N terminal is connected to OPA1_CHN1. (PA7): 1: Enable Differential Input PGA mode; 0: Disable differential input PGA mode.	0
8	FB_EN1	RW	OPA1's PGA mode feedback enable:	0

			1: Enable; 0: Prohibited.	
7	Reserved	RO	is reserved.	0
[6:4]	NSEL1[2:0]	RW	OPA1 Negative Terminal Channel Selection and PGA Gain Selection: 000: OPA1_CHN0 (PB1); 001: OPA1_CHN1 (PA7); 010: PGA mode, no negative input channel, internal gain of 8; 011: PGA mode, no negative input channel, internal gain of 16; 100: PGA mode, no negative input channel, internal gain of 32; 101: PGA mode, no negative input channel, internal gain of 64; Other: Shut down.	111b
3	PSEL1	RW	OPA1 Positive End Channel Selection: 1: OPA1_CHP1 (PA6); 0: OPA1_CHP0 (PB0).	0
[2:1]	MODE1[1:0]	RW	OPA1 Output Channel Selection: 00: The output signal is output through PC4; 01: The output signal is output through PA5; 1X: The output signal is not connected to the IO, but is still connected to the positive input of the comparator. end.	11b
0	EN1	RW	OPA1 Enable: 1: Enable; 0: Prohibited.	0

35.3.2 OPA Control Register 2 (R32_OPA_CTLR2) Offset Address:

0x04

31	30 29 28 27 26 25 24 23 22 21	2019	18	17 16
Reserved				

15	14 13 12	11	10	9	8	7	6 5	4	3	2	1	0
Reserved			HS2	PGADIF2	FB_EN2	Reserved	NSEL2[2:0]		PSEL2	Reserved	MODE2	EN2

Bit	name	access	describe	Reset value
[31:11]	Reserved	RO	is reserved.	0
10	HS2	RW	OPA2 High-Speed Mode Enabled: 1: Enable; 0: Prohibited.	0
9	PGADIF2	RW	OPA2 is used in conjunction with NSEL2 as a PGA, and the N-terminal is connected to OPA2_CHN1. (PF14): 1: Enable Differential Input PGA mode; 0: Disable differential input PGA mode.	0
8	FB_EN2	RW	OPA2's PGA mode feedback enable: 1: Enable; 0: Prohibited.	0

7	Reserved	RO	is reserved.	0
[6:4]	NSEL2[2:0]	RW	OPA2 Negative Terminal Channel Selection and PGA Gain Selection: 000: OPA2_CHN0 (PE8); 001: OPA2_CHN1 (PF12); 010: PGA mode, no negative input channel, internal gain of 8; 011: PGA mode, no negative input channel, internal gain of 16; 100: PGA mode, no negative input channel, internal gain of 32; 101: PGA mode, no negative input channel, internal gain of 64; Other: Shut down.	111b
3	PSEL2	RW	OPA2 Positive Channel Selection: 1: OPA2_CHP1 (PF11); 0: OPA2_CHP0 (PE9).	0
2	Reserved	RO	is reserved.	0
1	MODE2	RW	OPA2 Output Channel Selection: 1: The output signal is output through PB1; 0: The output signal is output through PE7.	0
0	EN2	RW	OPA2 Enable: 1: Enable; 0: Prohibited.	0

35.3.3 OPA Control Register 3 (R32_OPA_CTLR3) Offset Address:

0x08

31	30 29 28 27 26 25 24 23 22 21	2019	18	17 16
Reserved				

15	14 13 12 11	10	9	8	7	6 5 4	3	2	1	0
Reserved		HS3	PGADIF3	FB_EN3	Reserved	NSEL3[2:0]	PSEL3	Reserved	MODE3	EN3

Bit	name	access	describe	Reset value
[31:11]	Reserved	RO	is reserved.	0
10	HS3	RW	OPA3 High-Speed Mode Enabled: 1: Enable; 0: Prohibited.	0
9	PGADIF3	RW	OPA3 is used in conjunction with NSEL3 as a PGA, and the N terminal is connected to OPA3_CHN1. (PA3): 1: Enable Differential Input PGA mode; 0: Disable differential input PGA mode.	0
8	FB_EN3	RW	OPA3's PGA mode feedback enable: 1: Enable; 0: Prohibited.	0
7	Reserved	RO	is reserved.	0
[6:4]	NSEL3[2:0]	RW	OPA3 Negative Terminal Channel Selection and PGA Gain Selection: 000: OPA3_CHN0 (PC3);	111b

			001: OPA3_CHN1 (PA3); 010: PGA mode, no negative input channel, internal gain of 8; 011: PGA mode, no negative input channel, internal gain of 16; 100: PGA mode, no negative input channel, internal gain of 32; 101: PGA mode, no negative input channel, internal gain of 64; Other: Shut down.	
3	PSEL3	RW	OPA3 Positive Channel Selection: 1: OPA3_CHP1 (PA2); 0: OPA3_CHP0 (PC2).	0
2	Reserved	RO is reserved.		0
1	MODE3	RW	OPA3 Output Channel Selection: 1: The output signal is output through PA4; 0: The output signal is output through PA0.	0
0	EN3	RW	OPA3 Enable: 1: Enable; 0: Prohibited.	0

35.3.4 CMP Control Register (R32_CMP_CTLR) Offset

Address: 0x0C

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
Reserved	FILT_BASE[2:0]	Reserved	FILT_CFG[8:0]		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		FILT_EN	VREF[1:0]		HYPSEL[1:0]		EN	MODE[3:0]				NSEL[1:0]		PSEL[1:0]	

Bit	Name access		describe	Reset value
31	Reserved	RO is reserved.		0
[30:28]	FILT_BASE[2:0]	RW	CMP filter sampling time base configuration.	0
[27:25]	Reserved	RW	RO is reserved.	0
[24:16]	FILT_CFG[8:0]		CMP filter sampling interval configuration.	0
[15:14]	Reserved	RO.		0
13	FILT_EN	RW	CMP digital filtering enable: 1: Turn on; 0: Off.	0
[12:11]	VREF[1:0]	RW	CMP internal bias voltage selection bit: 00: Invalid; 01: 1/4VDD33A; 10: 1/2 VDD33A; 11:3/4VDD33A.	0
[10:9]	HYPSEL[1:0]	RW	CMP Hysteresis Voltage Selection Bit: 00:0mV; 01: 10mV;	0

			10:20mV; 11:30mV.	
8	EN	RW	CMP Enable: 1: Enable; 0: Off.	0
[7:4]	MODE[3:0]	RW	CMP output channel selection: 0000: Connected to GPIO (PA6: AF10; PA8: AF12; PB12: AF13; PC5: AF13; PE6: AF11; PE12: AF13; PE15: AF13); 0001: TIM1_BKIN; 0010: TIM8_BKIN; 0011: TIM1_CH4; 0100: TIM2_CH4; 0101: TIM3_CH4; 0110: TIM4_CH4; 0111: TIM5_CH4; 1000: TIM8_CH4; 1001: TIM9_CH4; 1010: TIM10_CH4; 1011: TIM11_CH4; 1100: TIM12_CH4; 1101: LPTIM1_CH1; 1110: LPTIM2_CH1; 1111: Disable output function.	1111b
[3:2]	NSEL[1:0]	RW	CMP Negative Channel Selection Bit: 00: CHN0 (PB1); 01: CHN1 (PC4); 10: DAC2 output (PA5); 11: CMP internal bias voltage.	11b
[1:0]	PSEL[1:0]	RW	CMP Positive Channel Selection Bit: 00: CHP0 (PB0); 01: CHP1 (PB2); 10: OPA1 output; 11: Retained.	11b

35.3.5 CMP Status Register (R32_CMP_STATR) 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															OUT_FILT

Bit	Name access		describe	Reset value
[31:1]	Reserved	RO	is reserved.	0

0	OUT_FILT	RO	CMP output: When CMP digital filtering is enabled (FILT_EN = 1), the output is the filtered... Wave value; when CMP digital filtering is off (FILT_EN = 0), the output is... These are normal values.	0

Chapter 36 Serial Audio Interface (SAI)

The Serial Audio Interface (SAI) supports various common audio protocols, such as I2S, PCM/DSP, AC'97, LSB or MSB alignment, and...

TDM is suitable for both mono and stereo applications. To achieve flexibility and configurability of the SAI interface, it employs two completely independent...

The audio submodules contain up to 4 IO pins (SD, SCK, FS and MCLK).

SAI can be configured as any combination of master/slave and send/receive, and its operation can be set according to the synchronous/asynchronous configuration of the audio submodule.

The mode is full-duplex/simplex.

36.1 Main Features

It provides two independent control modules, each audio submodule can be configured as any combination of master/slave and send/receive, and each has...

An 8-shaped FIFO ; the two

audio submodules can operate in synchronous or asynchronous mode, with master/slave configurations independent of each other.

ÿ Flexible slot length configuration, and the ability to configure slots as valid or invalid.

ÿ Frame synchronization configuration (valid level, valid length, and offset)

Each audio frame contains up to 16 configurable slots . Up to 16 slots,

each supporting data of 8, 10, 16, 20, 24, or 32 bits in size.

Supports LSB or MSB data transmission

Supports multiple audio protocols including I2S standard, LSB or MSB alignment, PCM/DSP, TDM, and AC'97.

It features mono/stereo audio capabilities and supports mute settings.

ÿ Serial Clock Strobe Edge Selection (SCK)

Each audio submodule has two independent DMA interfaces, supporting slave mode with frequencies up to 4MHz.

Error flags and interrupt sources:

- FIFO overflow and underflow;

- Frame synchronization is detected in advance when switching modes;

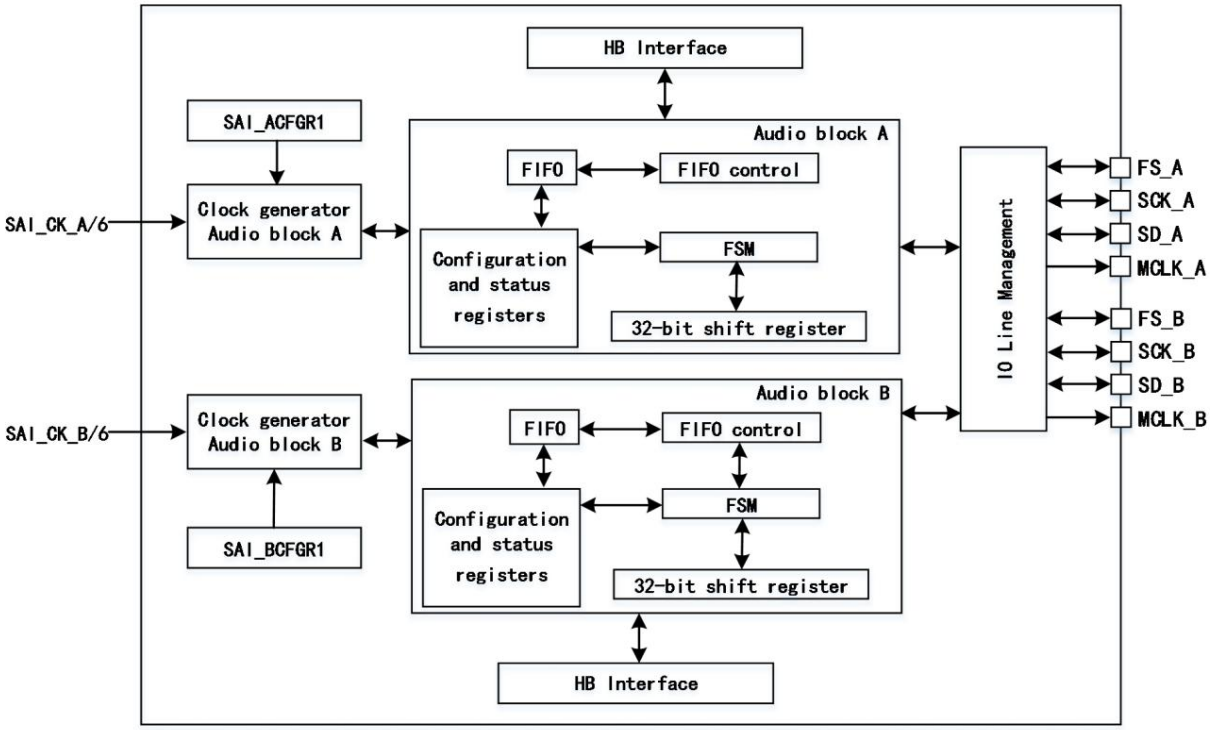
- Frame synchronization lag detection when in mode;

- AC'97 codec is not ready;

- Clock configuration error.

36.2 Overview

Figure 36-1 Structural diagram of SAI



SAI mainly consists of two independent audio sub-modules, which are connected to the output through an I/O management module. The I/O line controller manages SAI.

The four dedicated pins (SD, SCK, FS, MCLK) of the specified audio module are defined. If both submodules are declared as synchronization modules, then...

Some pins are shared, thus reserving some pins for general-purpose I/O. Whether the MCLK pin can be used as an output pin depends on the specific application.

And decoding requirements and whether the audio module is configured in primary mode.

Each audio module integrates a 32-bit shift register, which is controlled by the module's own functional state machine. Data storage and reading...

All access is accomplished through a dedicated FIFO. The FIFO can be accessed by the CPU or by DMA to reduce the communication burden on the CPU.

The audio submodule can be used as a transmitter or receiver in either master or slave mode. Master mode means generating the SCK bit clock from SAI and...

The frame synchronization signal, while slave mode means that the SCK bit clock and frame synchronization signal come from another external or internal master device. In special cases,

The direction of the FS signal is not directly related to the master or slave mode definition. In the AC'97 protocol, even if the SAI (link controller) is set to cancel...

The SCK clock is consumed, and the FS signal will also be output as SAI (this is also true in slave mode).

36.3 Functional Description

36.3.1 Clock Generator SAI's two

independent audio submodules each have their own clock generator, but these two clock generators are functionally identical.

different.

When the audio module is configured in master mode, the clock generator will be responsible for providing the bit clock (SCK) and the master clock for the external decoder.

Clock (MCLK). The clock generator will be turned off when the audio module is defined as slave mode.

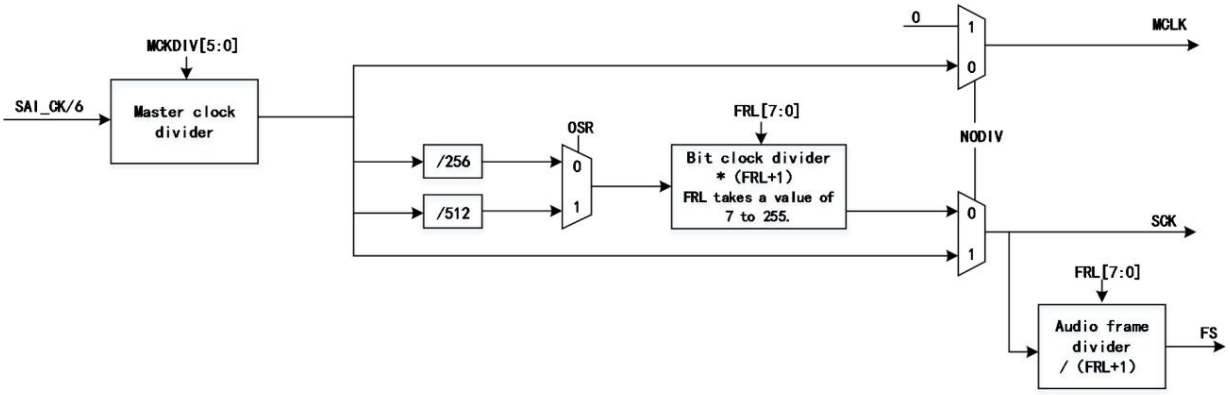
The bit clock strobe edge on SCK can be configured via the CKSTR bit in the R32_SAI_xCFGR1 register. This bit controls master and slave modes.

Both are applicable.

It is recommended to select 295MHz PLL output as the input clock for SAI.

The architecture of the audio module clock generator is shown in Figure 36-2.

Figure 36-2 Architecture of the audio module clock generator



Note: When set to 1, if the pin is configured as a peripheral pin, its signal level will be set to 1.NODIV MCLK GPIO SAI0

The clock source for the clock generator comes from the product clock controller. The SAI_CK_x clock is equivalent to the master clock and can be accessed via the MCKDIV[5:0] bits.

Frequency division for external decoder:

If MCKDIV[5:0] is not equal to 0000b, then MCLK_x = SAI_CK_x / (MCKDIV[5:0] * 2)

If MCKDIV[5:0] equals 0000b, then MCLK_x = SAI_CK_x

The MCLK signal is used only in TDM.

The frequency division must be uniform to maintain a 50% duty cycle on both the MCLK output and the SCK clock. If the MCKDIV[5:0] bits are set...

0000b, using a frequency divider, can make MCLK equal to SAI_CK_x.

In SAI, a single ratio MCLK/FS = 256 is used. In most cases, three frequency ranges will be encountered, as shown in Table 36-1.

Show.

Table 36-1 Examples of possible audio sampling ranges

Common audio sampling frequencies can be obtained by inputting the SAI_CK_x clock frequency.		MCKDIV[5:0]
192kHz * 256	192kHz	000000
	96kHz	000001
	48kHz	000010
	16kHz	000110
	8kHz	001100
44.1kHz * 256	44.1kHz	000000
	22.05kHz	000001
	11.025kHz	000010
SAI_CK_x = MCLK	MCLK	000000

Note: This occurs when the product clock controller selects an external clock source instead of the clock itself.PLL

If the corresponding audio module is used as the main module and the NODIV bit of the R32_SAI_xCFGR1 register is set to 0, it can be used via I/O port for...

An external decoder generates the master clock. The clock generator will be turned off when the audio module is defined as slave mode. The values of the NODIV and MCKDIV bits will be...

They are ignored. Furthermore, the MCLK I/O pin is also released and can be used as a general-purpose I/O pin.

The bit clock is derived from the master clock. The bit clock divider sets the division between the bit clock (SCK) and the master clock (SCK) according to the following formula.

Frequency coefficient:

SCK = MCLK*(FRL[7:0]+1)/256 Where:

256 is a fixed ratio between MCLK and the audio sampling frequency;

FRL[7:0] is the bit clock cycle in the audio frame - 1, configured in the R32_SAI_xFRCR register.

In main mode, (FRL[7:0]+1) must be equal to a power of 2 to obtain an even number of MCLK pulses over the bit clock cycle.

A 50% duty cycle will be guaranteed on the clock (SCK).

The SAI_CK_x clock can also be equal to the bit clock frequency. In this case, the NODIV bit in the R32_SAI_xCFGR1 register should be set.

1. The values within the MCKDIV divider and the bit clock divider will be ignored. At this point, the number of bits per frame is fully configurable, without...

It must be equal to a power of 2.

36.3.2 Operating Mode The SAI

audio submodule can be configured to master/slave or transmit/receive in any combination via the MODE[1:0] bits of the R32_SAI_xCFGR1 register.

The operating mode.

Each audio module in SAI is enabled via the SAIEN bit in the R32_SAI_xCFGR1 register. In slave mode, this bit is enabled once activated.

If the transmitter or receiver is active, it will be sensitive to activity on the clock line, data line, and synchronization line.

In master TX mode, even if there is no data in the FIFO, the enabled audio module will immediately generate a bit clock for the external slave module, but the FS...

Signal generation is controlled by the presence of data in the FIFO. After the FIFO receives the first data to be sent, this data will be output to the outside.

The FIFO is a slave module. If there is no data to send in the FIFO, a value of 0 will be sent in the subsequent audio frame, and an underflow flag will be generated.

In slave mode, audio frames begin when the audio module is enabled and when the SOF bit is detected.

In TX mode, underflow events are impossible on the first frame after enabling the audio module because of the forced operation sequence at this time.

as follows:

1) Write to R32_SAI_xDATAR via software or DMA;

2) The "Wait until FIFO threshold (FLTH)" flag is different from 000b (FIFO is empty);

3) Enable the audio module in send mode.

In master

mode, SAI provides clock signals to connected external devices:

1) The bit clock and frame synchronization are output on pins SCK and FS, respectively;

2) If needed, SAI can also generate the master clock on the MCLK pin.

FS is generated when the FIFO is not empty and the frame begins, while CK and MCLK are generated regardless of whether the FIFO is empty, as long as the audio...

Submodules will be generated when they are enabled.

In slave

mode, SAI receives clock signals from external devices.

1) When the SAI submodule is configured to asynchronous mode, the SCK pin and FS pin will be configured as inputs.

2) If the SAI submodule is configured to run synchronously with the second audio submodule, the corresponding SCK and FS pins will be released.

It can be used as general-purpose I/O.

In slave mode, the MCLK pin is not used and can be assigned to other functions. It is recommended to enable the slave device before enabling the master device.

36.3.3 Internal Synchronization Mode:

The audio submodule can run synchronously with another audio submodule within the same SAI, also known as internal synchronization mode. In this case,

Both will share the bit clock and frame synchronization signal to reduce the number of external pins required for communication. The audio module configured in synchronous mode will release...

The SCK, FS, and MCLK pins are used as GPIOs, while audio modules configured in asynchronous mode will use their SCK, FS, and MCLK I/O pins.

Pin (if the audio module is configured as the master module).

Typically, the audio module in synchronous mode can be used to configure SAI in full-duplex mode. One of the two audio modules can be configured as...

The master module and the slave module can be configured as one; alternatively, both can be configured as slave modules; one module can be an asynchronous module (R32_SAI_xCFGR1 register).

The SYNCEN[1:0] bits of the register are set to 00b), and the other is the synchronization module (R32_SAI_xCFGR1 register SYNCEN[1:0] bits are set to 00b).

01b).

Note: Due to the internal resynchronization phase, the frequency must be greater than twice the bit rate clock frequency. HCLK

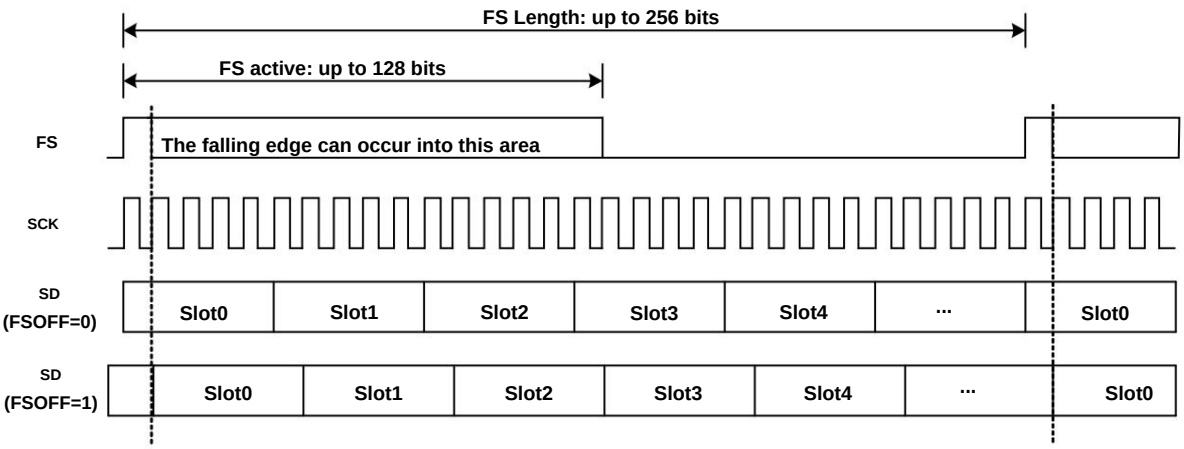
36.3.4 Frame Configuration

36.3.4.1 The Frame

Synchronization (FS) signal is used as the frame synchronization signal (SOF) in audio. The waveform of this signal can be configured by adjusting the R32_SAI_xFRCR register.

The frame is fully configurable and supports various audio protocols with special specifications during frame synchronization. The following diagram illustrates the configurability of the audio frame.

Figure 36-3 Audio Frame



In AC'97 mode or SPDIF mode (R32_SAI_xCFGR1 register PRTCFG[1:0] bits are 10b or 01b), the frame is the same

The waveform of the step signal is forced to support the AC'97 protocol. The value of the R32_SAI_xFRCR register is ignored.

Each audio module is independent of the others, and therefore requires specific configuration.

36.3.4.2 Frame Length Master

Mode :

Setting the FRL[7:0] field of the R32_SAI_xFRCR register to 1 can configure the length of the audio frame to a maximum of 256 bit clock cycles.

If the frame length is greater than the number of slots declared for that frame, the remaining bits to be sent will be padded with 0s, or the SD line will be released as a high group.

The state depends specifically on the state of the TRIS bit in the R32_SAI_xCFGR2 register. In receive mode, the remaining bits are ignored.

If the NODIV bit is cleared, (FRL+1) must be equal to a power of 2 (from 8 to 256) to ensure that each bit of the audio frame is clocked.

Each period contains an integer number of MCLK pulses. If NODIV is set to 1, the (FRL+1) field can take any value from 8 to 256.

The length

of the slave audio frame is primarily used to specify the number of bit clock cycles for each audio frame sent from the external master module to the slave module. It is mainly used...

The system detects premature or delayed frame synchronization signals during audio frame transmission from the main module. An error will occur in this case.

In slave mode, the configuration of the FRL[7:0] bits of the R32_SAI_xFRCR register is unrestricted.

36.3.4.3 Frame Synchronization

Polarity: The valid polarity of the FS pin is set by the FSPOL bit in the R32_SAI_xFRCR register, and this polarity is used to initiate the frame. SOF Signal

Sensitive to edges.

In slave mode, the audio module remains in a waiting state until it receives a valid frame signal to perform a send or receive operation.

The signal is synchronized with the frame signal. The frame is synchronized only if no SOF signal is detected during communication and the SOF signal matches the expected SOF signal.

Synchronized polarity is required for it to be effective.

In main mode, a frame synchronization signal is sent each time an audio frame is completed, until the SAIEN bit of the R32_SAI_xCTLR1 register is reached.

The FIFO is cleared. If there is no data in the FIFO at the end of the previous audio frame, the underflow will be handled as described in section 36.3.12.1.

However, the audio communication stream will not be interrupted.

36.3.4.4 Frame Synchronization Effective Level Length

The R32_SAI_xFRCR register's FSALL[6:0] bits are used to configure the length of the effective level of the frame synchronization signal. This length can be set to 1.

Up to 128 bit clock cycles.

In I2S, LSB, or MSB alignment modes, the effective length is set to half the frame length; while in PCM/DSP or TDM modes, there is

The effective length is set to 1 bit.

36.3.4.5 Frame Synchronization Offset:

Based on the audio protocols supported in the application (e.g., I2S standard protocol and MSB alignment protocol), the audio frame synchronization offset is set.

The frame synchronization signal is enabled at the last or first bit of transmission. This can be selected via the FSOFF bit of the R32_SAI_xFRCR register.

One of the two configurations.

36.3.4.6 The function of the FS signal

The meaning of the FS signal varies depending on the function of the FS. The meaning of the FS signal can be configured in the FSDEF bit of the R32_SAI_xFRCR register.

righteous:

- When the FSDEF bit is 1, the FS signal simultaneously carries the SOF signal and the channel identification signal within the frame, suitable for I2S, MSB, or LSB.

Audio frame structures such as alignment protocols;

- When the FSDEF bit is 0, the FS signal behaves as an SOF signal, which is suitable for PCM/DSP, TDM, AC'97 and audio protocols.

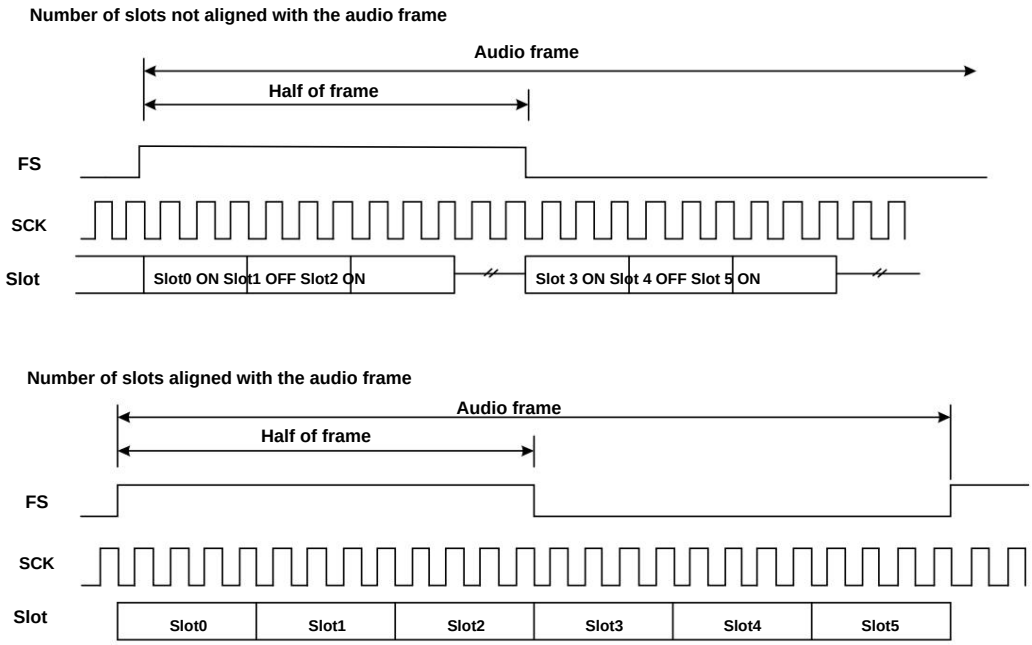
When interpreting the FS signal as an intra-frame SOF signal and channel identification signal, the declared number of slots should be evenly distributed among the left channel and...

Right channel. If the number of bit clock cycles in half an audio frame is greater than the number of dedicated slots for a certain channel, and the TRIS bit is set to 0, then...

The remaining bit clock cycles in the R32_SAI_xCFGR2 register will output 0. Conversely, if the TRIS bit is set to 1, the SD line will go high.

Impedance state. During reception, the remaining bit clock cycles are only considered when the channel changes.

Figure 36-4 shows that the function of FS is to combine the SOF signal and the channel identification signal.



Note: Frame length should be an even number.

If the FSDEF bit of the R32_SAI_xFRCR register remains cleared, the FS signal is equivalent to the SOF signal.

The slot number defined by bits NBSLOT[3:0] of the R32_SAI_xSLOTR register is multiplied by bits SLOTSZ[1:0] of the R32_SAI_xSLOTR register.

If the result of setting the Slot bit length is less than the frame size (FRL[7:0] bits of the R32_SAI_xFRCR register), then:

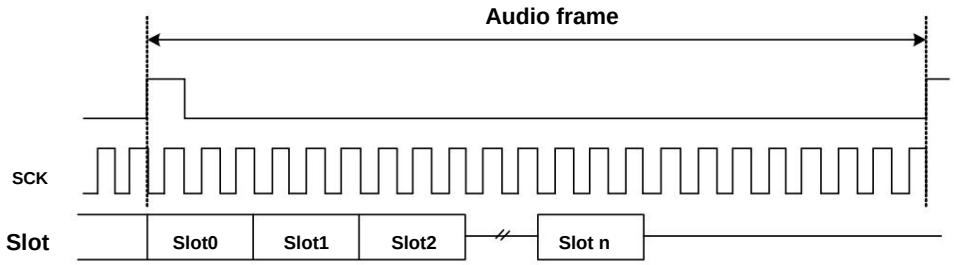
If the TRIS bit of the R32_SAI_xCFGR2 register is set to 0, the remaining bits after the last slot will be forced to 0 until the last slot is reached.

The sender's frame ends.

If the TRIS bit of the R32_SAI_xCFGR2 register is set to 1, the data lines will be released to a high-impedance state when these remaining bits are transmitted.

In receive mode, these bits are discarded.

Figure 36-5 shows that the function of FS is to provide the SOF signal.



Data =0 after slot n if TRIS = 0
SD output released (HI-Z) after slot n if TRIS =1

In transmit mode, when the audio module is configured to capture the SPDIF output on the SD line, the FS signal will not be used. Correspondingly...
The FS I/O will be freed up for other uses.

36.3.5 Slot Configuration The

basic unit of an audio frame is the slot, and each audio frame can be divided into a maximum of 16 slots. The number of slots in an audio frame is determined by...
NBSLOT[3:0]+1 is used to determine this.

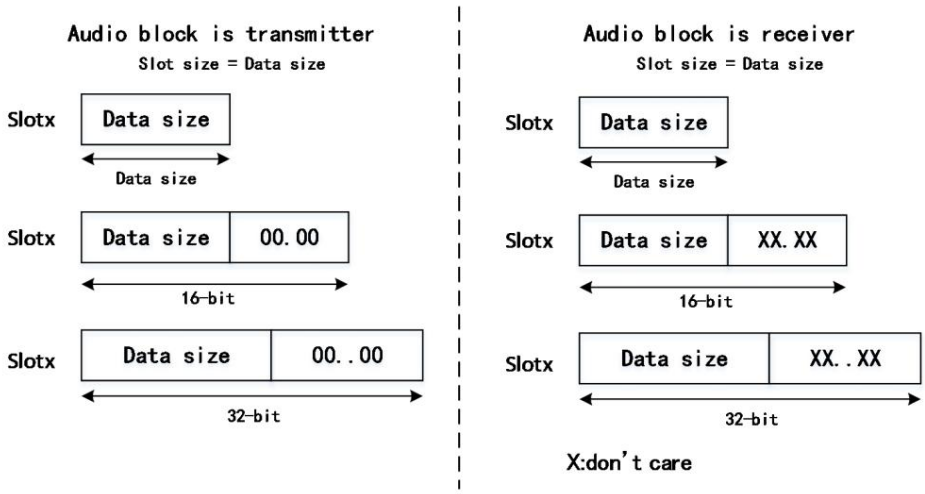
For AC'97 or SPDIF protocols (when PRTCFG[1:0] bits are set to 10b or 01b), the number of slots will be automatically...
Configure according to the corresponding protocol specifications, and the value of NBSLOT[3:0] will be ignored.

By setting the SLOTEN[15:0] bits of the R32_SAI_xSLOTR register, each slot can be set to an active or inactive state.

When an invalid slot is transmitted, the SD data line will be forced to either be set to 0 or placed in a high-impedance state, depending on the transmit mode.
The TRIS bit configuration (see the Invalid Slot Output Data Line Management section for details). In receive mode, data received after the invalid slot ends...
The data will be ignored. Therefore, there will be no FIFO access, nor will there be any FIFO read/write requests associated with this invalid slot state.

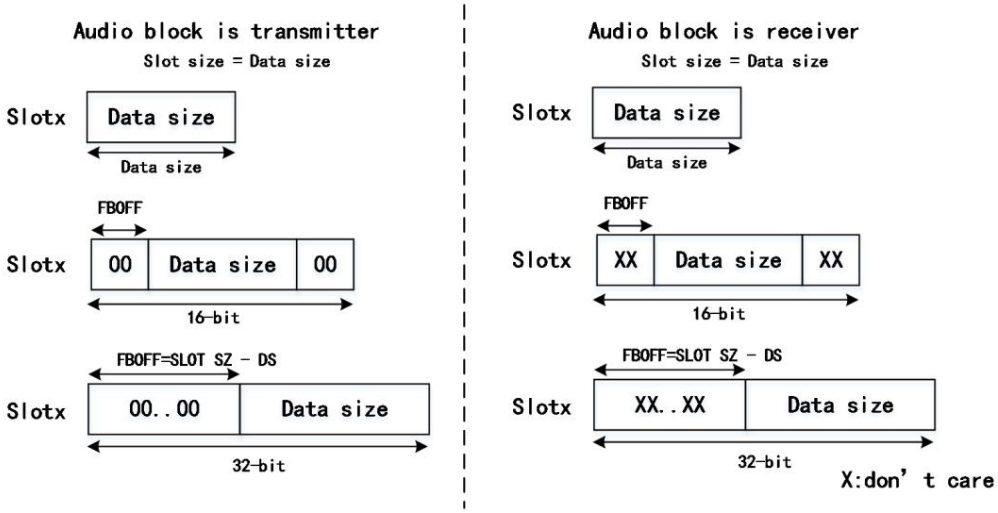
Furthermore, the size of the slot can be selected by setting the SLOTSZ[1:0] bits of the R32_SAI_xSLOTR register to 1, as shown in Figure 36-6.
Show.

Figure 36-6 Slot size configuration when FBOFF bit is 0



The position of the first data bit to be transmitted within a slot can be configured using the FBOFF[4:0] bits of the R32_SAI_xSLOTR register.
An offset value of one bit. As shown in Figure 36-7, in transmit mode, a value of 0 will be injected from the beginning of the slot until the preset offset is reached.
In receive mode, bits are shifted; however, in receive mode, bits in the offset phase are ignored. This feature applies to LSB alignment protocols (if the offset is equal to...).
(Slot size minus data size).

Figure 36-7 First offset



To avoid malfunctioning SAI behavior, the following conditions must be met:

FBOFF ÿ (SLOTSZ - DS),

DS ÿ SLOTSZ

NBSLOT x SLOTSZ ÿ FRL (frame length)

When the FSDEF bit in the R32_SAI_xFRCR register is set to 1, the number of slots must be even.

In AC'97 and SPDIF protocols (PRTCFG[1:0] bits set to 10b or 01b), the slot size is as per Section 36.3.7:

The definitions in the AC'97 link controller are set automatically.

36.3.6 Internal FIFO Each SAI

audio submodule has an independent 8-character-wide synchronous FIFO to improve transmission efficiency. These FIFOs can...

Accessed by the CPU or DMA, the FIFO request interrupt mechanism is used to request CPU and DMA access.

Depending on whether the audio module is defined as a transmitter or a receiver, reads or writes to the FIFO are performed accordingly. This applies regardless of the access size.

Each time a FIFO is read from or written to, the operation is performed on a single FIFO word. Each FIFO word contains one audio slot.

After each access to the R32_SAI_xDATAR register, the FIFO pointer is incremented by one word. Therefore, there is only one FIFO request and one R32_SAI_xSR register.

The register FREQ bit (FIFO request flag) is associated.

An interrupt will be generated when the FREQIE bit of the R32_SAI_xINTENR register is enabled. The triggering condition for this interrupt depends on the operating mode.

FIFO threshold, FIFO state, and DMA burst transfer size.

The specific mechanisms for triggering interrupts will be detailed in Table 36-2 and the subsequent paragraphs "Interrupts Generated in Transmit Mode" and "Interrupts Generated in Receive Mode".

The section on "Interruption" elaborates on this point.

Table 36-2 Conditions for FIFO Request Interruption

Send: MODE[1:0]=x0 Receive: MODE[0]=x1							
FIFO threshold FTH[2:0]	FIFO state FLTH[2:0]		FIFO threshold FTH[2:0]	FIFO state FLTH[2:0]			
	=000		=000	Empty = 000	Not empty ÿ 001	1/4 Full = 001 ÿ 1/4 Full ÿ 010	
Empty	1/4 Full = 001	1/2	Empty < 1/4 Full < 010 < 1/2	1/2 Full = 010	ÿ 1/2 Full ÿ 011	3/4 Full = 011 ÿ 3/4 Full ÿ 100	
Full = 010	3/4 Full = 011		Full < 011 < 3/4 Full < 100				
			Fully = 100 Not Full < 101				
Fully = 100	Fully = 101						

Note: (1) If the fundamental condition is not met, the interrupt request will be cleared by the hardware.

(2) Disable register, the data R32_SAI_xFIFOR will be cleared automatically. After that, it can be reinitialized. FIFO Pointer. If FFLUSH

read/write pointer will be reset to SAI FIFO 0.

Interrupts are generated in transmit

mode based on the FIFO configuration in transmit mode: When

the FIFO threshold bits in the R32_SAI_xCFGR2 register are configured to be empty (FTH[2:0] set to 000b), if there is no data in the register (FLTH[2:0] bits in the R32_SAI_xSR register are less than 001b), an interrupt will be generated (the FREQ bits in the R32_SAI_xSR register are set to 1 by hardware). When the FIFO is no longer empty (FLTH[2:0] bits in the R32_SAI_xSR are not 000b), that is, when one or more data are stored in the FIFO, this interrupt (the FREQ bits in the R32_SAI_xSR register) is cleared by hardware.

When the FIFO threshold bits in the R32_SAI_xCFGR2 register are configured to be one-quarter full (FTH[2:0] set to 001b), an interrupt will be generated if less than one-quarter of the FIFO contains data (FLTH[2:0] bits in R32_SAI_xSR are less than 010b) (the FREQ bits in the R32_SAI_xSR register are set to 1 by hardware). When at least one-quarter of the FIFO contains data (FLTH[2:0] bits in the R32_SAI_xSR register are greater than or equal to 010b), this interrupt (the FREQ bits in the R32_SAI_xSR register) is cleared by hardware.

When the FIFO threshold bits in the R32_SAI_xCFGR2 register are configured to half-full (FTH[2:0] set to 010b), an interrupt will be generated if less than half of the FIFO contains data (FLTH[2:0] bits in the R32_SAI_xSR register are less than 011b). When at least half of the FIFO contains data (FLTH[2:0] bits in the R32_SAI_xSR register are greater than or equal to 011b), this interrupt (FREQ bits in the R32_SAI_xSR register) will be cleared by hardware.

When the FIFO threshold bits in the R32_SAI_xCFGR2 register are configured to be three-quarters full (FTH[2:0] set to 011b), if less than three-quarters of the FIFO contains data (FLTH[2:0] bits in the R32_SAI_xSR register are less than 100b), an interrupt will be generated (the FREQ bits in the R32_SAI_xSR register are set to 1 by hardware). An interrupt will occur when at least three-quarters of the FIFO contains data (R32_SAI_xSR register...). If the FLTH[2:0] bits are greater than or equal to 100b, this interrupt (FREQ bit of R32_SAI_xSR register) is cleared by hardware.

When the FIFO threshold bits in the R32_SAI_xCFGR2 register are configured so that the FIFO is full (FTH[2:0] is set to 100b), if the FIFO is not full (FLTH[2:0] bits in the R32_SAI_xSR register are less than 101b), an interrupt will be generated (the FREQ bits in the R32_SAI_xSR register are set to 1 by hardware). When the FIFO is full (FLTH[2:0] bits in the R32_SAI_xSR register are equal to 101b), this interrupt (FREQ bits in the R32_SAI_xSR register) is cleared by hardware.

An interrupt is generated in receive mode.

An interrupt is generated based on the FIFO configuration

in receive mode: When the FIFO threshold bits in the R32_SAI_xCFGR2 register are configured to be empty (FTH[2:0] set to 000b), an interrupt will be generated if there is at least one data in the R32_SAI_xDATAR register (the FLTH[2:0] bits in the R32_SAI_xSR register are greater than or equal to 001b). When the FIFO becomes empty (the FLTH[2:0] bits in the R32_SAI_xSR register are equal to 000b), that is, when no data is stored in the FIFO, this interrupt (the FREQ bit in the R32_SAI_xSR register) is cleared by hardware.

When the FIFO threshold bits in the R32_SAI_xCFGR2 register are configured to be one-quarter full (FTH[2:0] set to 001b), an interrupt will be generated (the FREQ bit in the R32_SAI_xSR register is set to 1 by hardware) if at least one-quarter of the FIFO data units are available (FLTH[2:0] bits in the R32_SAI_xSR register are greater than or equal to 010b). When less than one-quarter of the FIFO data units are available (FLTH[2:0] bits in the R32_SAI_xSR register are less than 010b), this interrupt (FREQ bit in the R32_SAI_xSR register) is cleared by hardware. When the FIFO threshold

bits in the R32_SAI_xCFGR2 register are configured to half-full (FTH[2:0] set to 010b), an interrupt will be generated if at least half of the FIFO data units are available (FLTH[2:0] bits in the R32_SAI_xSR register are greater than or equal to 011b). The FREQ bit in the R32_SAI_xSR register is set to 1 by hardware. When less than half of the FIFO data units are available (FLTH[2:0] bits in the R32_SAI_xSR register are less than 011b), this interrupt (FREQ bit in the R32_SAI_xSR register) is cleared by hardware.

When the FIFO threshold bits in the R32_SAI_xCFGR2 register are configured to be three-quarters full (FTH[2:0] set to 011b), an interrupt will be generated (the FREQ bit in the R32_SAI_xSR register is set to 1 by hardware) if at least three-quarters of the FIFO data units are available (FLTH[2:0] bits in the SAI_xSR register are greater than or equal to 100b). When less than three-quarters of the FIFO data units are available (FLTH[2:0] bits in the R32_SAI_xSR register are less than 100b), this interrupt (the FREQ bit in the R32_SAI_xSR register) is cleared by hardware.

When the FIFO threshold bits in the R32_SAI_xCFGR2 register are configured to indicate that the FIFO is full (FTH[2:0] is set to 100b), if the FIFO is already full (FLTH[2:0] bits in the R32_SAI_xSR register are equal to 101b), an interrupt will be generated (the FREQ bits in the R32_SAI_xSR register are set to 1 by hardware). When the FIFO is not full (FLTH[2:0] bits in the R32_SAI_xSR register are less than 101b), this interrupt (R32_SAI_xSR register...) will be ge

The FREQ register is cleared by hardware.

Note: Similar to the generation of interrupts, if the DMAEN bit of the R32_SAI_xCFGR1 register is set to 1, then SAI can use DMA.

36.3.7 AC'97 Link Controller

SAI can be used as an AC'97 link controller, and this mode is configured via the PRTCFG[1:0] bits of the R32_SAI_xCFGR1 register.

When this protocol is selected, many configuration fields are ignored, so the number of slots and the slot size are fixed, and the frame synchronization signal is defined.

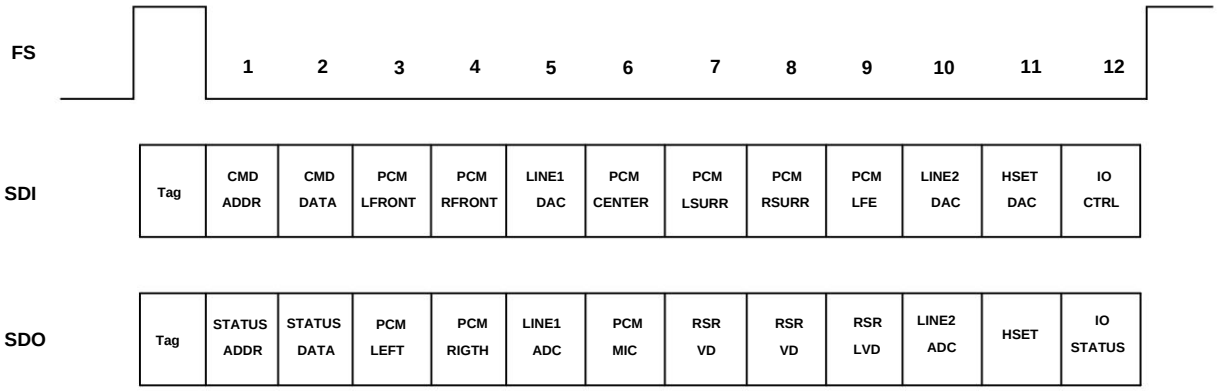
The waveform must be fixed and accurate; otherwise, SAI cannot be guaranteed to function properly.

- The NBSLOT[3:0] bits and the SLOTSZ[1:0] bits will be ignored;
- The number of slots is fixed at 13. The first slot is 16 bits wide, and all other slots are 20 bits wide (data slots).
- The FBOFF[4:0] bits of the SAI_xSLOTR register are ignored;
- The SAI_xFRCR register is ignored;
- MCLK is not used.

Regardless of whether a master or slave configuration is used, because the AC'97 link controller drives the FS signal, the FS signal emitted by the asynchronous module will...

Automatically configured as output.

Figure 36-8 AC'97 audio frame



Note: In the AC'97 protocol, the TAG bit will be reserved (always 0), so regardless of what value is written into it, the TAG bit 2 will remain

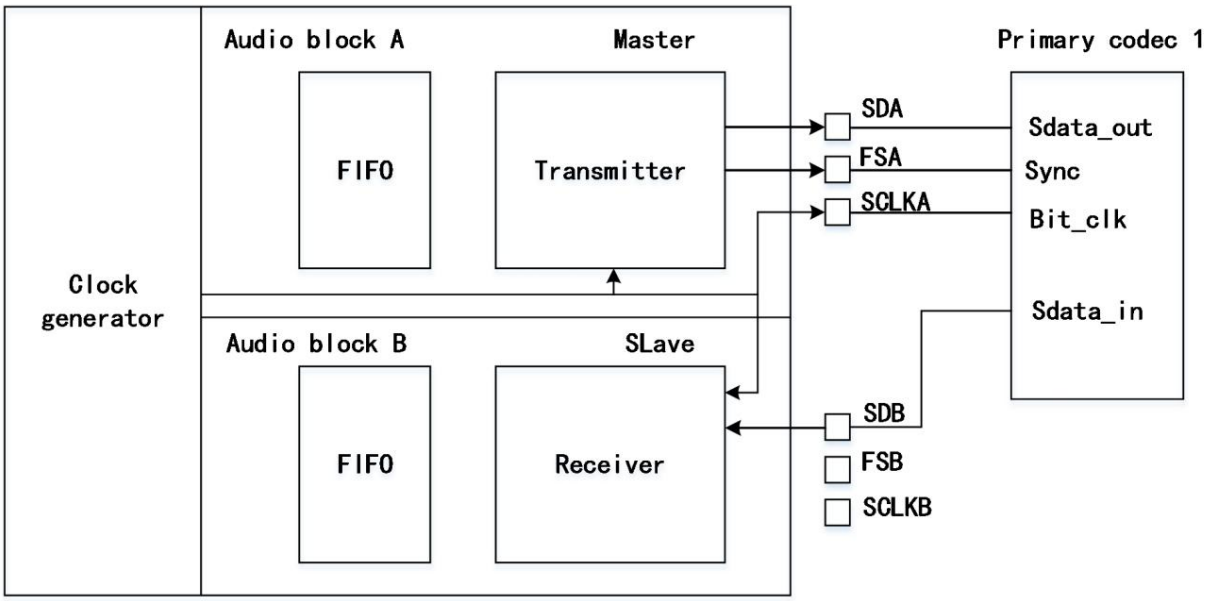
All values are forced to be 0. For details on representation, please refer to the AC'97 protocol standard.

TAG

A SAI can be used for AC'97 point-to-point communication, as shown in Figure 36-9. In the SAI, audio module A must be declared as a heterogeneous module.

The master transmitter is set up, while audio module B is defined as a slave receiver and internally synchronized with audio module A.

Figure 36-9 SAI used for AC'97 point-to-point communication



In receiver mode, the SAI used as the AC'97 link controller does not require a FIFO request; therefore, no data is stored in the FIFO when the codec ready bit in Slot0 decodes to low. If the CNRDYIE bit in the R32_SAI_xINTENR register is enabled, the CNRDY flag in the R32_SAI_xSR register will be set to 1 and an interrupt will be generated. This flag is dedicated to the AC'97 protocol.

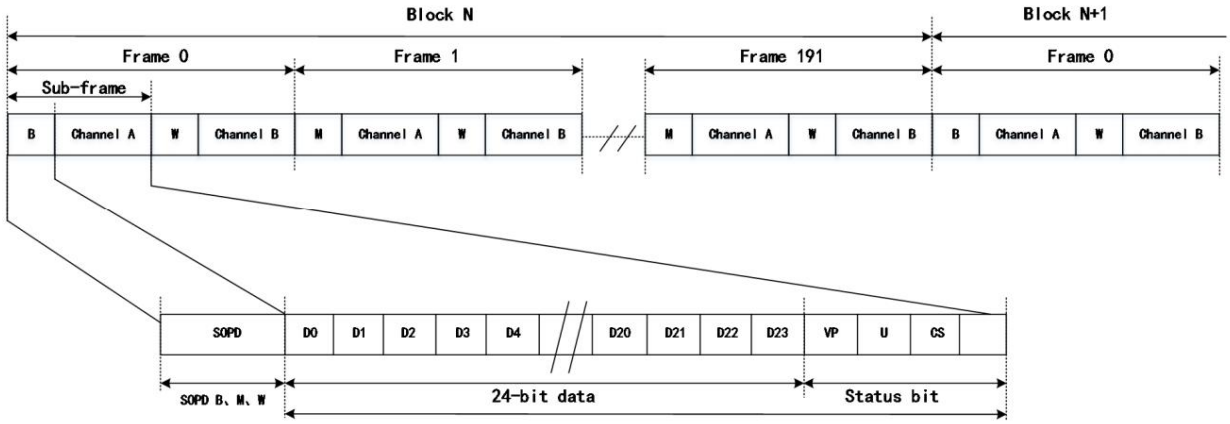
Clock generator programming in AC'97 mode

In AC'97 mode, the frame length is fixed at 256 bits, and its frequency should be set to 48kHz. Section 36.3.1: SAI Clock Generator The given rules should be used when FRL=255 to generate an appropriate frame rate (FFS).

36.3.8 SPDIF Output

SPDIF (Sony/Philips Digital Interface) is a digital audio interconnect for consumer audio devices, used to output audio over reasonable short distances. The SPDIF interface is only available in transmit mode and supports the IEC 60958 standard. SPDIF mode can be selected by setting the PRTCFG[1:0] bits of the R32_SAI_xCFGR1 register to 01b. For the SPDIF protocol: 1) Enable only the SD data lines; 2) Release the FS, SCK, and MCLK I/O pins; 3) To enable the SAI clock generator and manage the data rate on the SD lines, force the MODE[1] bit to 0 to select master mode; 4) Force the data size to 24 bits. Values set in the DS[2:0] bits of the R32_SAI_xCFGR1 register are ignored; 5) The clock generator must be configured to define the symbol rate; the known bit clock should be twice the symbol rate. The data is encoded using the Manchester protocol; 6) The SAI_xFRCR and SAI_xSLOTR registers are ignored. SAI is internally configured to conform to the SPDIF protocol requirements, as shown in Figure 36-0.

Figure 36-10 SPDIF format



An SPDIF block contains 192 frames. Each frame consists of two 32-bit subframes, typically one for the left channel and one for the right. Channel. Each subframe consists of an SOPD pattern (4 bits) used to specify whether the subframe is the start of a block (and to identify the channel). A), or whether it identifies channel A at some point in the block, or whether it refers to channel B (see Table 36-3). The next 28 bits are the channel... The information consists of 24 data bits and 4 status bits.

Table 36-3 SOPD Mode

SOPD	Header code		describe
	The last digit is 0.	The last digit is 1.	
B	11101000	00010111	Channel A data, and is the starting subframe of a block.
w	11100100	00011011	Channel B data
M	11100010	00011101	Channel A data

The data filling in the R32_SAI_xDATAR register for SPDIF data transfer should follow the following rules:

- R32_SAI_xDATAR[26:24] contains channel status bits, user bits, and validity bits;
- R32_SAI_xDATAR[23:0] contains 24 bits of data for the channel under consideration.

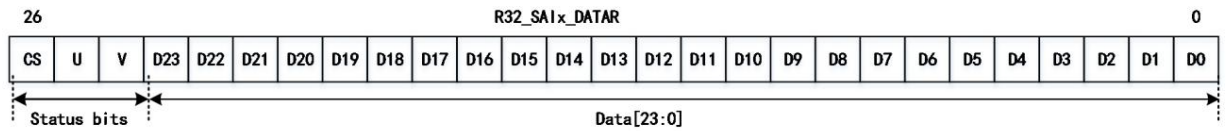
Note: (1) If the data size is bits, the data should be mapped to R32_SAI_xDATAR[23:4].

20

16

(2) If the data size is bits, the data should be mapped to R32_SAI_xDATAR[23:8].

Figure 36-11 Schematic diagram of R32_SAI_xDATAR register data



Note: R32_SAI_xDATAR[23] always represents the MSB. When performing a transfer, the LSB always takes precedence.

SAI first sends the appropriate header for each subframe in the block, followed by R32_SAI_xDATAR (in Manchester protocol) on the SD line. (The code is encoded). SAI terminates a subframe by transmitting the parity bits calculated as described in Table 36-4.

Table 36-4 Odd Number of Check Digits

R32_SAI_xDATAR[26:0] Odd	The value of the transmission parity bit P
number of	0
0s Odd number of 1s	1

In SPDIF mode, underflow is the only error flag used in the R32_SAI_xSR register. This is because SAI can only be used in transmit mode. Therefore, to recover from an underflow error detected by an underflow interrupt or underflow status bit, the following steps should be performed in sequence:

1) If DMA is already in use, then DMA flow needs to be disabled via the DMA peripheral.

2) Disable SAI operation and confirm that the peripheral is physically disabled by polling the SAIEN bit in the R32_SAI_xCFGR1 register.

3) Clear the flag by writing 1 to the OVRUDR bit of the R32_SAI_xSR register.

4) Flush the FIFO by setting the FFLUSH bit in the R32_SAI_xCFGR2 register to 1. The software needs to point to the beginning of the new data block (report).
The address of the subsequent data (the data in header B). If DMA is used, the DMA source start address pointer should be updated accordingly.

5) If the DMA controller restarts data transfer based on the new source start address pointer, the DMA stream is re-enabled.

6) Re-enable SAI by setting the SAIEN bit in the R32_SAI_xCFGR1 register to 1.

Clock generator programming in SPDIF generator mode:

For an SPDIF generator, SAI should provide a bit clock with an equal symbol rate:

Table 36-5 Audio Sampling Frequency and Symbol Rate (SHARK)

Audio sampling frequency (FS)	Symbol rate
44.1kHz	2.8224MHz
48kHz	3.072MHz
96kHz	6.144MHz
192kHz	12.288MHz

Typically, the relationship between audio sampling rate (FS) and bit clock rate (FSCK) is given by the following formula:

$$,- = \frac{F-./\_01\_2}{64}$$

36.3.9 Stereo/Mono

In transmit mode, if the number of slots is equal to 2 (bits NBSLOT[3:0] of the R32_SAI_xSLOTR register are set to 0001b), then
Addressable mono mode without any data preprocessing in memory. In this case, due to the number of Slot0 during transmission...
The data is copied to Slot1, and the FIFO access time will be reduced by half.

To enable mono mode, the following is required:

1) Set the MONO bit of the R32_SAI_xCFGR1 register to 1;

2) Set the NBSLOT bit of the R32_SAI_xSLOTR register to 1 and set the SLOTEN bit to 3.

In receive mode, setting the MONO bit to 1 is only meaningful when the number of slots is equal to 2 (the same as in transmit mode).
When set to 1, only the data in Slot0 will be stored in the FIFO; the data in Slot1 will be stored because it is considered identical to the data in the previous Slot.
Discarded. If the data stream in receive mode is a true stereo audio stream in which the left and right channel data are significantly different, then MONO
The bit is meaningless. The conversion from the output stereo file to the equivalent mono file is done by the software.

36.3.10 Mute Mode Mute mode

can be used when the audio submodule is used as a transmitter or receiver.

Audio submodule in send mode

In send mode, mute mode can be selected at any time. Mute mode is effective for all audio frames. During frame transmission...
Setting the MUTE bit in the R32_SAI_xCFGR2 register to 1 enables mute mode. This mute mode bit is only activated at the end of a frame. If
Setting MUTE to 1 at the end of a frame activates the mute mode at the start of a new audio frame and continues for the entire frame length until the next frame ends.
The frame is then strobed to determine whether the next frame will still be a silent frame.

If the number of slots set by the NBSLOT[3:0] bits of the R32_SAI_xSLOTR register is less than or equal to 2, then the mute mode can be specified.
The value sent is either 0 or the last value for each slot. This is determined by the MUTEVAL bit in the R32_SAI_xCFGR2 register.
choose.

If the number of slots set in bits NBSLOT[3:0] of the R32_SAI_xSLOTR register is greater than 2, then because each bit of each slot...
The value sent is 0, so the MUTEVAL bit in the R32_SAI_xCFGR2 register is meaningless.

In silent mode, the FIFO pointer continues to increment, which means that data in the FIFO that was requested to be transmitted in silent mode will be discarded.

Audio submodule in receive mode

In receive mode, for a given number of consecutive audio frames (bits MUTECNT[5:0] of the R32_SAI_xCFGR2 register), a silence mode from an external transmitter can be detected if 0 is received in all declared valid slots of the audio frames. When the corresponding number of silence frames are detected, the MUTEDET flag of the R32_SAI_xSR register is set to 1, and an interrupt is generated when the MUTEDETIE bit of the R32_SAI_xCFGR2 register is set to 1.

The silence frame counter is reset to zero when the audio submodule is disabled or when at least one audio frame is received within a valid slot. An interrupt is generated only once when the counter reaches the value specified in the MUTECNT[5:0] bits. Subsequent interrupt events are reinitialized when the counter is reset.

Note: Mute mode is not available for the SPDIF audio module.

36.3.11 Compression Expansion Mode

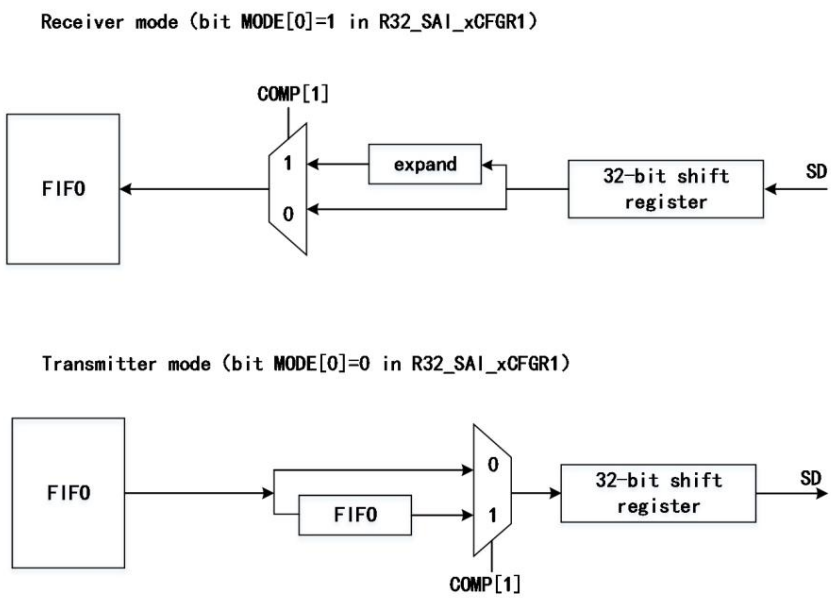
In the field of mobile communications, data processing before transmission often involves the application of data compression and expansion algorithms. Applications can select SD serial communication mode based on the COMP[1:0] bits of the R32_SAI_xCFGR2 register (used only when TDM mode is selected). Whether data is processed before being transmitted on the output line (compressed), and whether data is extended after being received on the SD serial input line (extended), are shown in Figure 36-11. The two supported companding modes are ÿ-Law and A-Law, both of which are recommended by CCITT G.711.

The companding standard used in the United States and Japan is ÿ-Law, which allows a 14-bit dynamic range (10 bits of COMP[1:0] in the R32_SAI_xCFGR2 register). The European companding standard is A-Law, which supports a 13-bit dynamic range (11 bits of COMP[1:0] in the R32_SAI_xCFGR2 register). The calculation method for ÿ-Law or A-Law companding standards depends on the setting of the CPL bits in the R32_SAI_xCFGR2 register.

The calculation is performed using the two's complement of 1 or the two's complement of 2.

In the ÿ-Law and A-Law standards, data is encoded in an 8-bit format aligned to MSB, with a compressed data width. The bit depth is always 8 bits. Therefore, when the SAI audio module is enabled (R32_SAI_xCFGR1 register SAIEN bit is set to 1) and one of the two companding modes is selected via the COMP[1:0] bits, the DS[2:0] bits of the R32_SAI_xCFGR1 register will be forced to 010b. If companding is not required, the COMP[1:0] bits should remain cleared.

If the SAI audio module is configured as a transmitter and the COMP[1] bit of the R32_SAI_xCFGR2 register is set to 1, compression mode will be used; if the SAI audio module is declared as a receiver, expansion algorithm will be used. Figure 36-12 Data compression and expansion hardware in the SAI audio module



Note: This does not apply when AC'97 is selected.

Management of output data lines on invalid slots

In transmit mode, the behavior of the SD line output can be controlled via the TRIS bit when an invalid slot is transmitted on the data line.

- When an invalid slot is sent, SAI forces the SD output line to 0.

- After the last data bit has been transmitted, the SD output line will be placed in a high-impedance state to release the data line for other connections.

The node's sender is used.

Two transmitters must not be driven simultaneously on the same SD output pin to prevent short circuits. This is to ensure proper transmission.

If a gap exists and the data is less than 32 bits, the SLOTSZ[1:0] bits in the R32_SAI_xSLOTR register can be set to 10 bits to move the data.

The data is extended to 32 bits. Subsequently, if the next slot is declared invalid, the SD output pin will be extended at the end of the LSD of the valid slot.

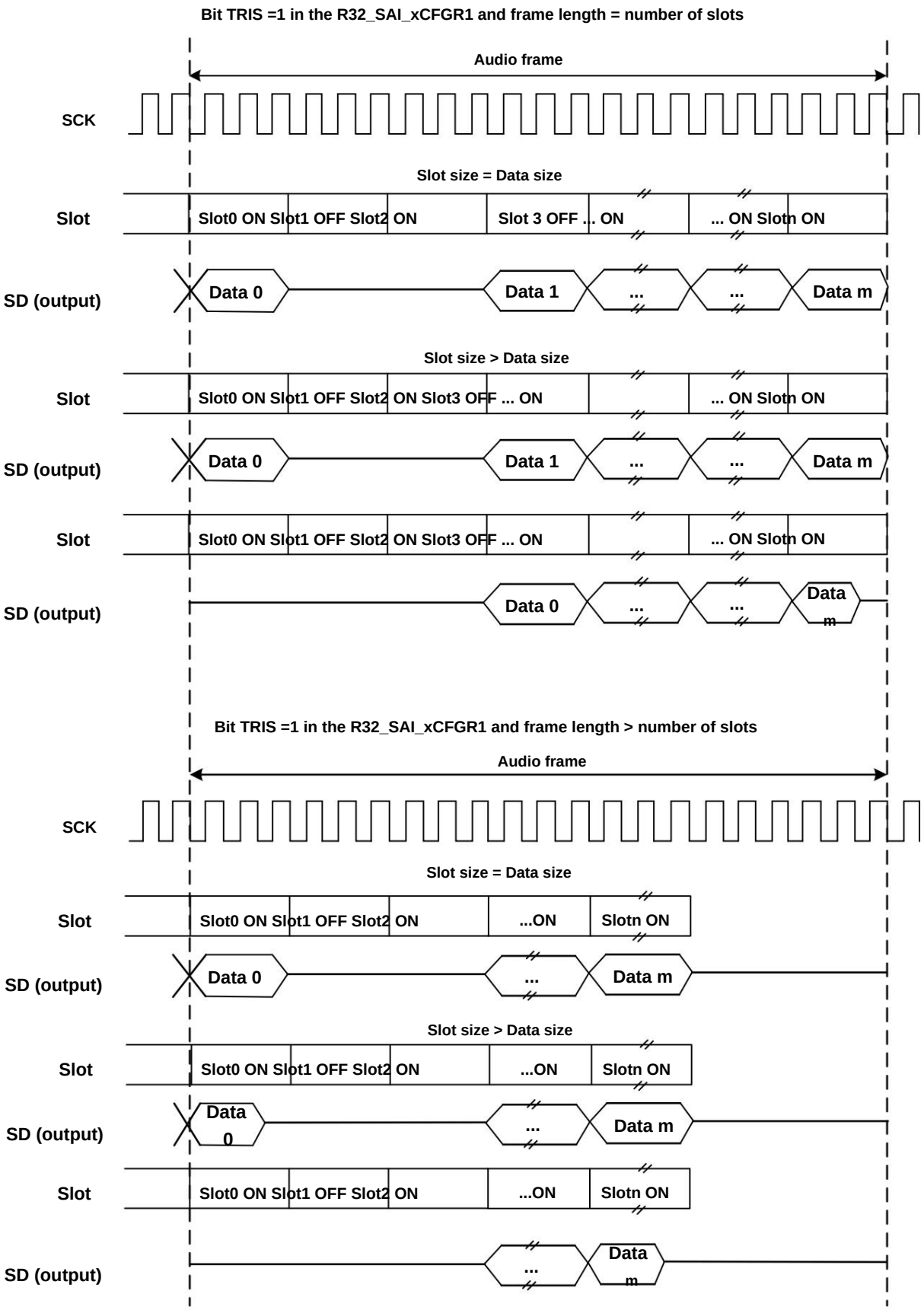
(The zero-filling phase for extending the data to 32 bits) is set to a tri-state.

Furthermore, if the product of the number of slots and the slot size is less than the frame length, then the SD output will be padded with zeros at the end of the audio frame.

The line will be set to a tri-state.

Figure 36-13 illustrates these behaviors.

Figure 36-13 Three-state strategy on the SD output line when sending an invalid slot.

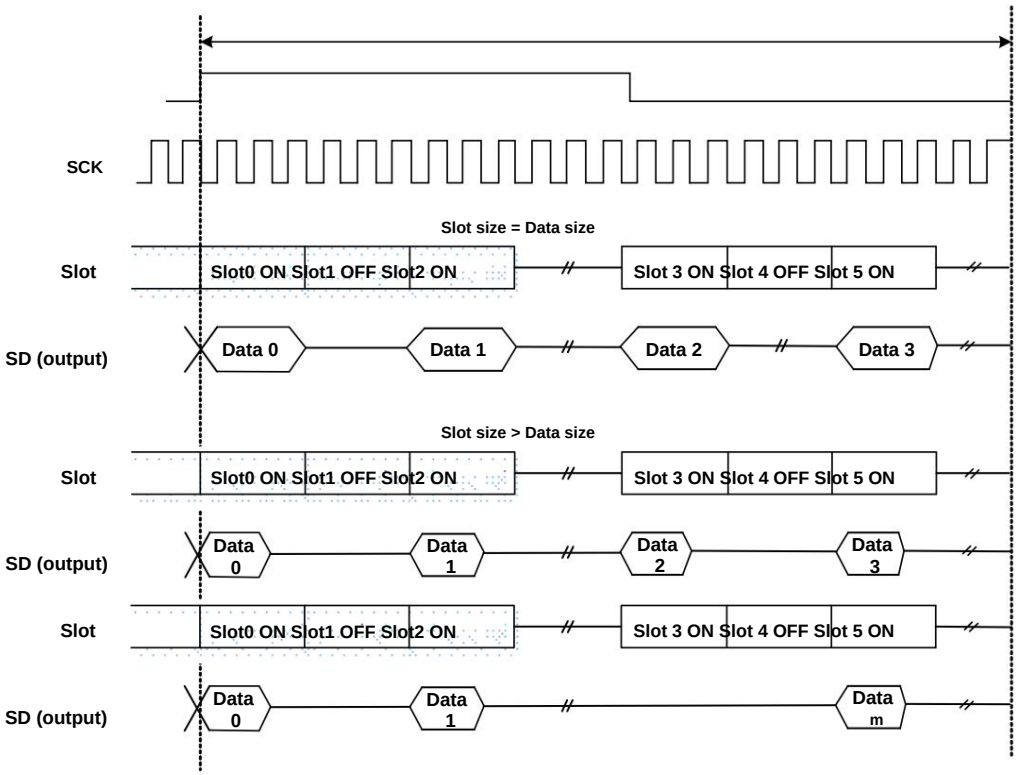


When the selected audio protocol uses the FS signal as the SOF signal or channel identification signal (R32_SAI_xFRCR register FSDEF location)

1) When the tri-state mode is managed according to Figure 36-14 (where the R32_SAI_xCFGR1 register TRIS bit is set to 1, FSDEF bit is set to 1),

The half-frame length is greater than the number of slots/2 and NBSLOT=6.

Figure 36-14 shows the three-state strategy on the output data line when using protocols such as I2S.



If the TRIS bit of the R32_SAI_xCFGR2 register is cleared, all high-impedance states on the SD output lines in Figures 36-13 and 36-14 will be disabled.

This will be replaced with the driver using value 0.

36.3.12 Error Flags SAI Use the

following error flags:

- FIFO overflow/underflow;
- Frame synchronization early detection;
- Frame synchronization lag detection;
- Codec not ready (AC'97 only);
- The main mode clock configuration is incorrect.

36.3.12.1 FIFO Overflow/Underflow (OVRUDR)

The FIFO overflow/underflow bits are the OVRUDR bits of the R32_SAI_xSR register.

Because the audio module can function as both a receiver and a transmitter, and each audio module in SAI is specified to have its own...

The R32_SAI_xSR register is used for overflow and underflow errors, so the same bit is used for both.

If the

audio module is configured as a receiver, then when an audio frame is received even though the FIFO is full and can no longer store received data, overflow will occur .

An overflow will occur. In this case, received data will be lost, and the OVRUDR flag in the R32_SAI_xSR register will be set to 1; if

Setting the OVRUDRIE bit in the R32_SAI_xINTENR register to 1 will also generate an interrupt. Internally, it will record the slot number at the time of the overflow. FIFO

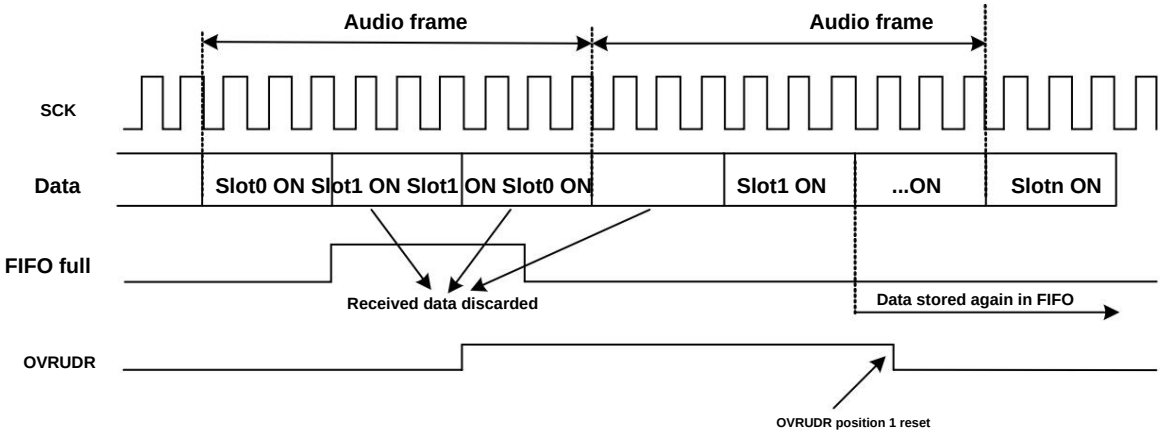
No more data can be stored until space is freed up to store new data. SAI will terminate when the FIFO has freed up space for at least one data item.

The audio module receiver will begin receiving new data from the new audio frame from the slot number recorded internally after overflow is detected, thus avoiding...

To prevent data slot misalignment in the target memory, please refer to Figure 36-15.

To clear the OVRUDR flag, simply write 1 to the OVRUDR bit of the R32_SAI_xSR register.

Figure 36-15 Overflow error detection



Underflow can occur when the audio module in SAI is used as a transmitter and the FIFO is empty when data needs to be sent. If the detection...

If underflow is detected, the slot number of the event is stored and a MUTE value (00b) is sent until the FIFO is ready to send and detect underflow.

The overflowing slot corresponds to the data, please refer to Figure 36-16 for details. This avoids the memory pointer from interfering with the slot in the audio frame.

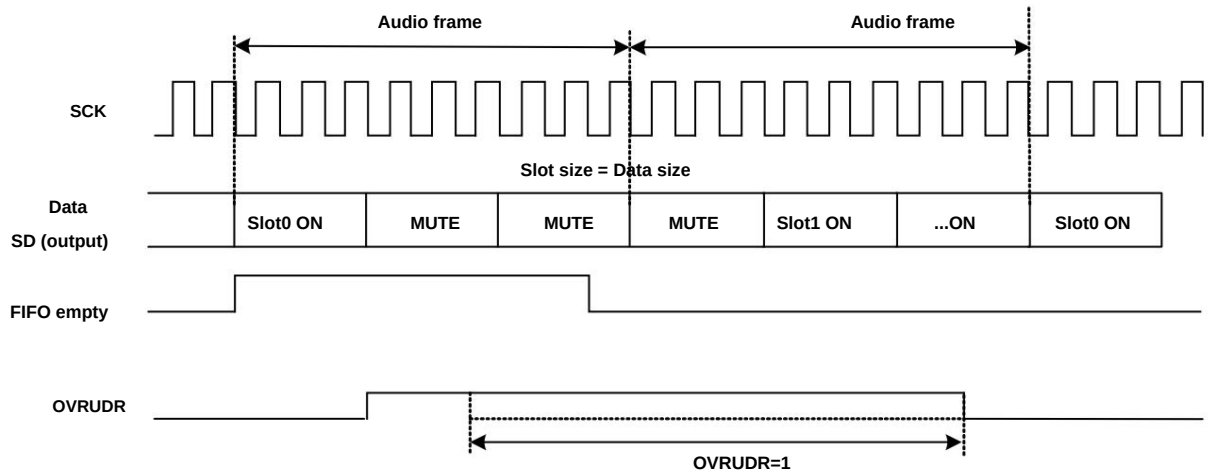
Step failed.

An underflow event will set the OVRUDR flag in the R32_SAI_xSR register to 1 if the OVRUDRIE bit in the R32_SAI_xINTENR register is set to 1.

Setting this flag to 1 will also generate an interrupt. To clear this flag, simply write 1 to the OVRUDR bit of the R32_SAI_xSR register.

An underflow event occurs when the audio submodule is configured as master or slave mode.

Figure 36-16 FIFO underflow event



36.3.12.2 Frame Synchronization Early Detection (AFSDET) The

AFSDET flag is only used when the SAI audio module is used in slave mode; it is disabled in master mode.

In the R32_SAI_xFRCR register, the frame length, frame polarity, and frame offset configuration are all known. This flag is used to indicate whether the detection occurred earlier than expected.

Frame synchronization (FS) signal detected. When frame detection advance occurs, the AFSDET flag in the R32_SAI_xSR register will be set to 1.

For current audio frames that are insensitive to the FS (First-Side Frame) advance, this detection will have no effect. In other words, "parasitic" events in the FS signal will be flagged.

It will be recorded, but it will not interfere with the processing of the current audio frame.

An interrupt will be generated if the AFSDETIE flag in the R32_SAI_xINTENR register is set to 1. Clearing the AFSDET flag requires...

Write 1 in the first position.

To resynchronize with the main module after a frame detection early error occurs, the following steps must be followed:

1) Disable the SAI module by resetting the SAIEN bit in the R32_SAI_xCFGR1 register. To ensure SAI is disabled, it is necessary to read back...

The SAIEN bit was checked and its value was confirmed to be 0.

2) Refresh the FIFO by setting the FFLUSH bit of the R32_SAI_xCFGR2 register to 1.

3) Re-enable the SAI peripheral by setting the SAIEN bit of the R32_SAI_xCFGR1 register to 1.

4) The SAI module will wait for FS to be enabled in order to restart synchronization with the main module.

Note: The SAIEN flag is not enabled in AC'97 mode because a signal is also SAI The audio module acts as a link controller, even when declared as a slave module.

generated during block creation. FS Since the AC'97 protocol does not use signals, this function is meaningless in SPDIF mode.

36.3.12.3 Frame Synchronization Lag Detection (LFSDET) The

LFSDET flag is only used when the SAI audio module is used as a slave module; it is disabled in master mode.

The frame length, frame polarity, and frame offset configuration are all known in the R32_SAI_xFRCR register. If the external master module fails to send the data at the scheduled time...

If the FS signal is generated too late, the LFSDET flag will be set to 1. If the LFSDETIE bit in the R32_SAI_xINTENR register is set to 1, it will also...

An interrupt will be generated. The LFSDET flag will be cleared by writing 1 to the LFSDET bit in the R32_SAI_xSR register.

When a corresponding error is detected, the frame synchronization lag detection flag is set to 1, and SAI needs to resynchronize with the main module. See section [link to relevant documentation] for details.

Section 36.3.12.2: The order described in Frame Synchronization Early Detection (AFSDET).

In noisy environments, the audio module's state machine may incorrectly detect interference with the SCK clock and shift the SAI data to the wrong position.

Incorrect frame position. SAI will detect this event and report it as a frame synchronization lag detection error.

If the external master module is not managing audio data frame transmission in continuous mode, it will not corrupt the frames; most applications do not.

This situation may occur. In this case, the LFSDET flag will be set to 1.

Note: The LFSDET flag is not enabled in AC'97 mode because the audio module acts as a link controller, even when declared as a slave.

using the SAI module, definitions will also not Signals. When Because the AC'97 protocol does The signal is invalid, therefore this function has no practical meaning in SPDIF mode.

be generated.

36.3.12.4 Codec not ready (CNRDY AC'97)

When the SAI audio module is configured to operate in AC'97 mode (i.e., the R32_SAI_xCFGR1 register PRTCFG[1:0] bits are set...

The CNRDY flag in the R32_SAI_xSR register only becomes meaningful when set to 10b. The CNRDYIE flag in the R32_SAI_xINTENR register is also meaningful if the value is set to 10b.

If the CNRDY flag is set to 1, an interrupt will be generated.

During the reception of AC'97 audio frames in TAG0 (slot0), if the codec is not yet ready for communication, the CNRDY flag is displayed.

The bit will be set to 1. In this case, data will not be automatically stored in the FIFO until TAG0 indicates that the codec is ready, because...

The codec is not yet ready. Once the codec is ready, it will capture all valid slots defined in the R32_SAI_xSLOTR register.

To clear the CNRDY flag, the CNRDY bit in the R32_SAI_xSR register must be written to 1.

Master mode clock configuration error (NODIV=0) When

the audio module is working in master mode (MODE[1:0]=0x) and NODIV is 0, if the following conditions are met, then as long as

When SAI is enabled, the WCKCFG flag will be set to 1.

- (FRL+1) is not a power of 2;

- (FRL+1) is not between 8 and 256.

The MODE, NODIV, and SAIEN bits belong to the R32_SAI_xCFGR1 register, and the FRL bit belongs to the R32_SAI_xFRCR register.

If WCKCFGIE is set to 1, an interrupt will be generated when the WCKCFG flag in the R32_SAI_xSR register is set to 1. To clear this flag...

To resolve this, simply write 1 to the WCKCFG bit of the R32_SAI_xSR register.

When WCKCFG is set to 1, the audio module is automatically disabled, so the SAIEN bit will be hardware cleared.

36.3.13 DMA Interface

To reduce CPU load and optimize bus bandwidth, each SAI audio module has an independent DMA interface for...

The R32_SAI_xDATAR register is used for read/write operations (accessing the internal FIFO). Each audio submodule is equipped with support for basic DMA requests.

/ DMA channel of the response protocol.

When configuring DMA transfers to serve the audio submodule, the DMAEN bit in the R32_SAI_xCFGR1 register must be set to 1. DMA requests are made by...

The FIFO controller manages it directly, and its generation depends on the FIFO threshold. When the audio submodule is configured for transmit mode, the FIFO threshold must be...

Set it to a specific value to ensure that there is enough remaining space to implement a complete DMA burst write operation even in the worst case.

Otherwise, a FIFO overflow error may occur. When the audio submodule is configured in receive mode, the FIFO threshold must be set to a specific value.

This ensures that there is enough data in the FIFO to complete a full DMA burst read operation, thereby avoiding FIFO underflow errors.

The DMA transfer direction is related to the SAI audio submodule configuration:

- If the audio module is used as a transmitter, the audio module's FIFO controller will output a DMA request to load SAI_xDATAR into the FIFO.

Data written to the register;

- If the audio module is used as a receiver, the DMA request is related to a read operation from the R32_SAI_xDATAR register.

Configure the SAI interface to DMA mode in the following order:

1) Configure SAI and FIFO thresholds to specify when to initiate DMA requests.

2) Configure the SAIDMA channel.

3) Enable DMA.

4) Enable the SAI interface.

Note: Before configuring the module, it must be disabled. SAI DMA SAI aisle.

36.3.14 SAI Interruption

Table 36-6 SAI Interruption Requests

Interrupt source	interrupt division	Interruption conditions	Interrupt enable control	Interrupt clear control
FREQ	ask	MODE[1:0] represents any mode.	FREQIE	depending on: –FIFO threshold setting – Communication transmission direction See section 36.3.6 for details.
MUTEDET (Silent)		MODE[1:0] is the receiver	MUTEDETIE	CMUTEDET = 1
OVRUDR error		MODE[1:0] represents any mode.	OVRUDRIE	COVRUDR = 1
AFSDET error		MODE[1:0] is slave mode (not applicable) AC'97 mode and SPDIF mode)	AFSDETIE	CAFSDET = 1
LFSDET error		MODE[1:0] is slave mode (not applicable) AC'97 mode and SPDIF mode)	LFSDETIE	CLFSDET = 1
CNRDY	mistake	MODE[1:0] is the slave mode (AC'97 mode only)	CNRDYIE	CCNRDY = 1
WCKCFG Error	MODE[1:0] is the primary mode and NODIV = 0		WCKCFGIE	CWCKCFG = 1

Enable interrupts in the following order:

1) Disable SAI interruption;

2) Configure the SAI function register;

3) Enable SAI interrupt sources;

4) Enable SAI.

36.4 Register Description

Table 36-7 List of SAI-related registers

name	Access address	describe	Reset value
R32_SAI_ACFGR1	0x40015804	SAI Module A Configuration Register 1	0x00000040
R32_SAI_ACFGR2	0x40015808	SAI Module A Configuration Register 2	0x00000000
R32_SAI_AFRCR	0x4001580C	SAI Module A Frame Configuration	0x00000007
R32_SAI_ASLOTR	Register 0x40015810	SAI Module A Slot Register	0x00000000
R32_SAI_AINTENR	0x40015814	SAI Module A Interrupt Enable Register	0x00000000
R32_SAI_ASR	0x40015818	SAI Module A Status Register	0x00000008
R32_SAI_ADATAR	0x40015820	SAI Module A Data Register	0x00000000
R32_SAI_BCFGR1	0x40015824	SAI Module B Configuration Register 1	0x00000040
R32_SAI_BCFGR2	0x40015828	SAI Module B Configuration Register 2	0x00000000
R32_SAI_BFRCR	0x4001582C	SAI Module B Frame Configuration	0x00000007
R32_SAI_BSLOTR	Register 0x40015830	SAI Module B Slot Register	0x00000000
R32_SAI_BINTENR	0x40015834	SAI Module B Interrupt Enable Register	0x00000000
R32_SAI_BSR	0x40015838	SAI Module B Status Register	0x00000008
R32_SAI_BDATAR	0x40015840	SAI Module B Data Register	0x00000000

36.4.1 SAI Module x Configuration Register 1 (R32_SAI_xCFGR1) (x=A/B)

Offset address: 0x04, 0x24

31 30 29	28	27 26 25	24 23 22 21 20	19	18	17 16	
Reserved		OSR	MCKDIV[5:0]	NODIV	Reserved	DMAEN	SAIEN

15 14 13	12	11 10	9	8	7 6 5	4	3 2	1 0
Reserved	MONO	SYNCEN[1:0]	CKSTR	LSBFIRST	DS[2:0]	Reserved	PRTCFG[1:0]	MODE[1:0]

Bit	name	access	describe	Reset value
[31:27]	Reserved	RO	is reserved.	0
26	OSR	RW	Oversampling rate of the master clock: 1: FFS×512; 0: FFS×256.	0
[25:20]	MCKDIV[5:0]	RW	Master clock divider: 000000: Master clock input divided by 1; Other: The master clock frequency will be calculated according to the following formula: FSCK_x = 3456_78_9 / :01;/< Note: Configure this bit only if the audio module is disabled.	0
19	NODIV	RW	No frequency divider: 1: Do not use a frequency divider in the clock generator (in this case, the master clock is divided). (Frequency bit not working) 0: Enable the master clock generator.	0
18	Reserved	RO	is reserved.	0
17	DMAEN	RW	DMA Enable: 1: Enable;	0

		RW	0: Off. Note: In receive mode, the DMAEN position must be configured. 1 MODE[1:0] bits, to avoid DMA The request is due to the fact that the audio module will default to sending mode after a reset.	
16	SAIEN	RW	Audio module enabled: 1: Enable the SAIx audio module; 0: Disable SAIx audio module. Note: When A module (A or B) When configuring the primary mode, SAI clock must be present in the SAI module input before the SAIEN position can be set. 1.	0
[15:13]	Reserved	RO	is reserved.	0
12	MONO	RW	Mono mode selection: 1: Mono mode; 0: Stereo mode. Note: This bit is only meaningful when the number is zero. Slot 2	0
[11:10]	SYNCEN[1:0]	RW	Synchronous enable: 00: The audio submodule is in asynchronous mode; 01: The audio submodule is synchronized with another internal audio submodule. This In this case, the audio submodule must be configured as slave mode; Other: Reserved. Note: (1) After enabling SPDIF mode, the audio submodule should be configured. Set to asynchronous; (2) Configure this bit only if the audio module is disabled.	0
9	CKSTR	RW	Clock strobe edge: 1: Modify the signal generated by SAI on the falling edge of SCK, and sample the signal received by SAI on the rising edge of SCK; 0: The signal generated by SAI is modified on the rising edge of SCK, while the signal received by SAI is sampled on the falling edge of SCK. Note: Configure this bit only if the audio module is disabled.	0
8	LSBFIRST	RW	Least significant bit priority: 1: The LSB that prioritizes data transmission; 0: MSB with priority for data transmission. Note: (1) Configure this bit only when the audio module is disabled. (2) LSB When using the mode, the received data is in the high bits, such as the received data size bits. 16 The bit, with data content of 0x1234, is actually received. The received data is 0x12340000.	0
[7:5]	DS[2:0]	RW	Data size selection: 000/001: Reserved; 010: 8 bits; 011: 10 digits; 100: 16 bits; 101:20 bits; 110: 24 bits; 111: 32 bits. Note: Configure this bit only if the audio module is disabled.	010b

4	Reserved	RO reserved.		0
[3:2] PRTCFG[1:0]		RW	Protocol configuration: 00: Free Protocol. This is achieved by setting most of the configuration register bits and... Frame configuration register, the free protocol allows users to use audio modules This powerful configuration feature allows for handling specific audio protocols (such as...) I2S, LSB/MSB alignment, TDM, PCM/DSP...); 01: SPDIF protocol; 10: AC'97 Protocol; 11: Retained. Note: Configure this bit only if the audio module is disabled.	0
[1:0] MODE[1:0]		RW	SAIx audio module mode: 00: Master transmitter; 01: Main receiver; 10: From the transmitter; 11: From the receiver. Note: (1) If the audio module is configured in SPDIF mode, the master send mode (MODE[1:0] 00) will be forced. In master send mode, the audio module will immediately begin generating audio. FS and clock. (2) This bit is configured only if the audio module is disabled.	0

36.4.2 SAI Module x Configuration Register 2 (R32_SAI_xCFGR2) (x=A/B)

Offset address: 0x08, 0x28

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
Reserved					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMP[1:0] CPL			MUTECNT[5:0]						MUTEVAL	MUTE	TRIS	FFLUSH	FTH[2:0]		

Bit	name	access	describe	Reset value
[31:16] Reserved		RO is retained.		0
[15:14] COMP[1:0]		RW	Compandulation mode: 00: Companding algorithm is not supported; 01: Reserved; 10: ÿ-Law algorithm; 11: A-Law Algorithm. Note: Compandulation mode can only be used when a protocol is selected. TDM	0
13 CPL		RW	Two's complement: 1:2 complement representation; 0:1 two's complement representation. Note: Only when the companding mode is ÿ-Law or A-Law algorithm. This bit is valid.	0
[12:7] MUTECNT[5:0]		RW	Silent Counter: The values set in these bits will be related to the connection detected in the receive mode.	0

			The number of silent frames is compared. When the number of silent frames is similar to this value... At the same time, the MUTEDET flag is set to 1, and the MUTEDETIE position is 1. In such cases, an interrupt will also be generated. Note: This field is only used in receive mode.	
6	MUTEVAL	RW	Mute setting: 1: Send the previous value during silent mode; 0: The bit value 0 is sent during silent mode. Note: This bit is meaningless for the SPDIF audio module, and therefore does not enable... use.	0
5	MUTE	RW	Silent mode enabled: 1: Enable silent mode; 0: Disable silent mode. Note: This bit is meaningless for the SPDIF audio module, and therefore does not enable... use.	0
4	TRIS	RW	Three-state management of data cables: 1: The SD output line will be released when the last data bit of the previous valid slot (the next slot is invalid) is transmitted (high impedance). state); 0: When Slot is invalid, the SD output line is still driven by SAI.	0
3	FFLUSH	RW	FIFO refresh: 1: FIFO refresh. Setting this bit to 1 will trigger a FIFO refresh. All internal FIFO pointers (read and write) will be cleared to zero. This situation... In this case, the data still remaining in the FIFO is lost (either sent or received). (Data will not be lost further). DMA must be disabled before refreshing SAI. Data flow/interruption; 0: Disable FIFO refresh.	0
[2:0] FTH[2:0]		RW	FIFO threshold setting: 000: FIFO is empty; 001: &—+ FIFO; 010: &—$ FIFO; 011: #—+ FIFO; 100: FIFO is full. Other: Reserved.	0

36.4.3 SAI Module x Frame Configuration Register (R32_SAI_xFRCR) (x=A/B)

Offset address: 0x0C, 0x2C

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
Reserved				FSOFF	FSPOL	FSDEF

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

ved		

Bit	name	access	describe	Reset value
[31:19] Reserved		RO reserved.		0
18	FSOFF	RW	Frame synchronization offset: 1: Enable FS in the bit before the first bit of Slot 0; 0: Enable FS in the first bit of Slot 0.	0
17	FSPOL	RW	Frame synchronization polarity: 1: FS is active high (rising edge); 0: FS is active low (falling edge). Note: An external pull-up resistor is required when FSPOL=0.	0
16	FSDEF	RO	Frame synchronization definition: 1: The FS signal consists of the SOF signal and the channel identification signal; 0: The FS signal is the start frame signal. When this position is 1, the R32_SAI_xSLOTR register is defined as follows: The number of slots must be even. This means that half of the slots will be used for... Left channel, other slots are used for right channel (e.g., for I2S) For protocols such as MSB/LSB alignment, this bit must be set to 1.	0
15	Reserved	RO reserved.		0
[14:8]	FSALL[6:0]	RW	Frame synchronization active level length: Specify the effective level length of the FS signal in the audio frame, in bit clock mode. Number (SCK) + 1 (FSALL[6:0] + 1).	0
[7:0]	FRL[7:0]	RW	Define the audio frame length in terms of SCK clock cycles: [Frame length is missing from the original text] The number of bits is equal to FRL[7:0] + 1. The number of bits sent in an audio frame must be greater than or equal to 8; otherwise, the audio... The module will encounter an operational error. The data size is 8 bits and is... The R32_SAI_xSLOTR register's NBSLOT[4:0] only defines When a slot (NBSLOT[3:0] = 0000) is selected, it belongs to this... This is one of the situations. In master mode, if the master clock (MCLK_x pin is raised) is used If the frame length is between 8 and 256, then it should be equal to 2. A number raised to a power of several. When not using the master clock (NODIV = 1), the building... The proposal suggests programming the frame length to a value between 8 and 256.	0x07

Note: The R32_SAI_AFRCR and R32_SAI_BFRCR registers must be configured with the audio module disabled, and for AC'97 and...

The SPDIF audio protocol is meaningless.

36.4.4 SAI Module x Slot Register (R32_SAI_xSLOTR) (x=A/B)

Offset address: 0x10, 0x30

31	30	29 28 27 26 25 24 23 22	twenty one	2019	18	17 16
SLOTEN[15:0]						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				NBSLOT[3:0]				SLOTSZ [1:0]		Rese rved	FBOFF[4:0]				

Bit	name	access	describe	Reset value
[31:16]	SLOTEN[15:0]	RW	Slot Enable: 1: Effective Slot; 0: Invalid Slot. Each SLOTEN bit corresponds to a Slot bit from 0 to 15. Set (up to 16 slots).	0
[15:12]	Reserved	RO	reserved.	0
[11:8]	NBSLOT[3:0]	RW	Number of slots in the audio frame: The set value represents the number of slots in the audio frame + 1 (including invalid slots). The maximum number of slots is 16. When the FSDEF bit of the R32_SAI_xFRCR register is set to 1, the number of slots should be even.	0
[7:6]	SLOTSZ[1:0]	RW	Slot size: 00: Slot size and data size (in R32_SAI_xCFGR1) The register DS[3:0] is equivalent to the one specified in the register. 01: 16 bits; 10: 32 bits; 11: Retained. The slot size must be greater than or equal to the data size. If this condition is not met... Under this condition, SAI's behavior will be uncertain.	0
5	Reserved	RO	reserved.	0
[4:0]	FBOFF[4:0]	RW	First bit offset: The set value defines the position of the first data transmission bit in the Slot. It represents an offset value. In send mode, this data field uses... Extra bits will be forcibly cleared to zero. In receive mode, extra bits will be discarded. The position to be received.	0

Note: The R32_SAI_ASLOTR and R32_SAI_BSLOTR registers must be configured with the audio module disabled, and for AC'97 and The SPDIF audio protocol is meaningless.

36.4.5 SAI Module x Interrupt Enable Register (R32_SAI_xINTENR) (x=A/B)

Offset address: 0x14, 0x34

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
Reserved					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									LFSDETIE	AFSDETIE	CNRDYIE	FREQIE	WCKCFGIE	MUTEDETIE	OVRUDRIE

Bit	name	access	describe	Reset value
[31:7]	Reserved	RO	reserved. Frame	0
6	LFSDETIE	RW	synchronization hysteresis detection interrupt enabled: 1: Enable interrupt; 0: Interruption is disabled.	0
5	AFSDETIE	RW	frame synchronization early detection interrupt enabled:	0

			1: Enable interrupt; 0: Interruption is disabled.	
4	CNRDYIE	RW	Codec not ready interrupt enabled: 1: Enable interrupt; 0: Interruption is disabled.	0
3	FREQIE	RW	FIFO request interrupt enable: 1: Enable interrupt; 0: Interruption is disabled.	0
2	WCKCFGIE	RW	Clock configuration error interrupt enabled: 1: Enable interrupt; 0: Interruption is disabled.	0
1	MUTEDETIE	RW	Mute detection interrupt enable: 1: Enable interrupt; 0: Interruption is disabled.	0
0	OVRUDRIE	RW	Overflow/underflow interrupt enable: 1: Enable interrupt; 0: Interruption is disabled.	0

36.4.6 SAI Module x Status Register (R32_SAI_xSR) (x=A/B) Offset Address: 0x18, 0x38

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									LFSDET	AFSDET	CNRDY	FREQ	WCKCFG	MUTEDET	OVRUDR

Bit	name	access	describe	Reset value
[31:19] Reserved		RO	is reserved.	0
[18:16]	FLTH[2:0]	RO	FIFO threshold flag: If the SAI module is configured as a transmitter: 000: FIFO is empty; 001: FIFO <= 1/4, but not empty; 010: 1/4 < FIFO <= 1/2; 011: 1/2 < FIFO <= 3/4; 100: 3/4 < FIFO, but not full; 101: FIFO is full; Other: Reserved. If the SAI module is configured as a receiver: 000: FIFO is empty; 001: FIFO < 1/4, but not empty; 010: 1/4 <= FIFO < 1/2; 011: 1/2 == FIFO < 3/4;	0

			100: 3/4 == FIFO, but not full; 101: FIFO is full; Other: Reserved.	
[15:7] Reserved		RO reserved.		0
6	LFSDET	RW1Z	Frame synchronization lag detection: 1: The frame synchronization signal did not appear at the correct time; 0: No error. Note: Writing "1" is invalid. 0, write 0	0
5	AFSDET	RW1Z	Frame synchronization early detection: 1: Frame synchronization signal was detected in advance; 0: No error. Note: Writing "1" is invalid. 0, write 0	0
4	CNRDY	RW1Z	Codec not ready: 1: External AC'97 codec is not ready; 0: External AC'97 codec is ready. Note: (1) Only if the R32_SAI_xCFGR1 register is selected This bit is used only when the AC'97 audio module is configured as a receiver. (2) Write 1 to clear 0, write 0 invalid.	0
3	FREQ	RW1Z	FIFO request flag: 1: FIFO request to read or write R32_SAI_xDATAR; 0: No FIFO request. The requested content depends on the audio module's configuration: – If the module is configured for send mode, then FIFO requests and sends Write the relevant information in R32_SAI_xDATAR. – If the audio module is configured for receive mode, then the FIFO request and Read the relevant information from R32_SAI_xDATAR. Note: Writing this bit is invalid. 1: Clear to 0, write 0	1
2	WCKCFG	RW1Z	Clock configuration error flags: 1: The clock configuration does not conform to the frame length specification defined in frame synchronization. (Configuration of FRL[7:0] bits in the R32_SAI_xFRCR register); 0: Clock configuration is correct. Note: Writing "1" is invalid. 0, write 0	0
1	MUTEDET	RW1Z	Quietness indicator: 1: A specified number of consecutive audio frames were detected on the SD input line. The MUTE value (0 value) in the data; 0: No MUTE value was detected on the SD input line. Note: Writing this bit is invalid. 1: Clear to 0, write 0	0
0	OVRUDR	RW1Z	Overflow/underflow error flags: 1: An overflow/underflow error was detected; 0: No overflow/underflow error. Note: Writing "1" is invalid. 0, write 0	0

36.4.7 SAI Module x Data Register (R32_SAI_xDATAR) (x=A/B) Offset Address: 0x20, 0x40

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	describe	Reset value
[31:0]	DATA[31:0]	RW	data: If the FIFO is not full, writing to this register can load data into the FIFO. If the FIFO is not empty, reading this register will clear the FIFO.	0

Chapter 37 QSPI Interface (QuadSPI)

The module descriptions in this chapter apply only to CH32H417 and CH32H416 microcontroller products.

The chip has two built-in dedicated QSPI communication interfaces (QuadSPI) that can connect to single, dual, or quad (data lines) SPI FLASH memory.
Storage medium. The main operating modes of this interface include the following:

Indirect Mode: In this mode, the QSPI interface performs all operations using the QSPI registers. This mode is suitable for applications requiring high frequency...
Application scenarios involving complex read and write operations.

• Status Polling Mode: In this mode, the system periodically reads the external FLASH status register. When the flag is set to 1...
(If erasure or programming is complete), the system will generate an interrupt. This mode is suitable for applications that require real-time monitoring of the external Flash status.
scene.

Memory-mapped mode: In this mode, external Flash memory is mapped into the microcontroller's address space, and the system treats it as internal.
The memory is accessed. The address of this memory-mapped memory can only be used for data access.

When using dual-flash mode, the system can access two Quad-SPI FLASH memories simultaneously, doubling both throughput and capacity.
This pattern is suitable for application scenarios that require large amounts of data processing.

37.1 Main Features

Three functional modes: indirect mode, status polling mode, and memory-mapped mode.

• Dual flash memory mode, accessing two flash memories in parallel. •

Integrated FIFO for sending and receiving.

Supports SDR mode

ÿ Fully programmable frame format for both indirect mode and memory-mapped mode

ÿ Fully programmable opcodes for indirect mode and memory-mapped mode

A DMA trigger signal is generated when the FIFO threshold is reached and the transfer is complete .

An interrupt will be generated when the FIFO threshold is reached, a timeout occurs, the operation completes, or an access error occurs.

37.2 Overview

Figure 37-1 QSPI Functional Block Diagram (Dual Flash Mode Disabled)

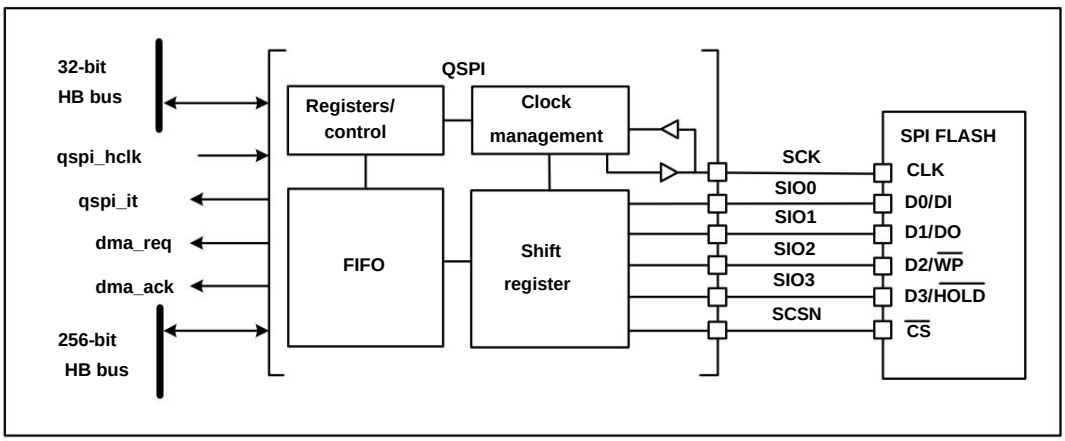
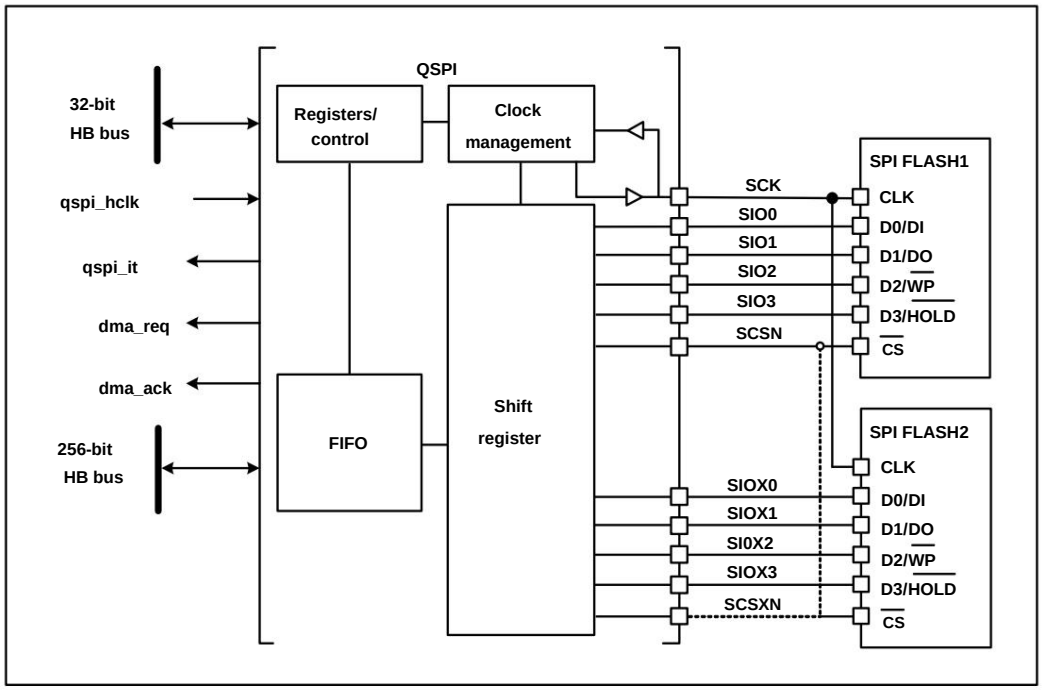


Figure 37-2 QSPI Functional Block Diagram (Dual Flash Mode Enabled)



37.3 Functional Description

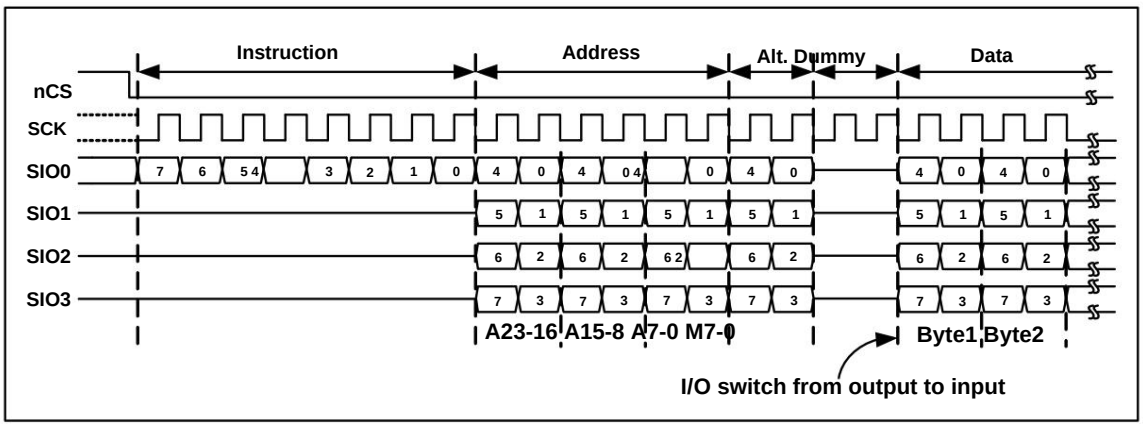
37.3.1 QSPI Command Sequence

QSPI communicates with FLASH via commands. Each command consists of five stages: instruction, address, alternating bytes, null instruction, and data.

According to the data. Any stage can be skipped, but at least one of the instruction, address, alternating byte, or data stages must be retained. nCS in each instruction

The command descends before the instruction begins and rises again after each instruction is completed.

Figure 37-3 Example of a read command in four-line mode



In the instruction

phase , an 8-bit instruction configured in the INSTRUCTION[7:0] field of the R32_QSPIx_CCR register is sent.

The data is sent to the FLASH memory to specify the type of operation to be performed. Most FLASH memory can only communicate via the SIO0 signal (single-wire SPI mode).

Instructions are received one bit at a time. However, by configuring the IMODE[1:0] field in the R32_QSPIx_CCR register, instructions can be received in a manner that is 1 bit at a time.

Selective transmission during the command phase allows for the transmission of either 2 bits at a time (via SIO0/SIO1 in two-wire SPI mode) or 4 bits at a time (via four-wire SPI mode).

In wired SPI mode, this is achieved via SIO0/SIO1/SIO2/SIO3. If IMODE=00b, the instruction stage is skipped, and the command sequence starts from the address stage.

The segment (if it exists) begins.

In the

address phase, 1-4 bytes can be sent to the FLASH memory to indicate the address to be operated on. The number of address bytes to be sent is configured in the ADSIZE[1:0] field of the R32_QSPIx_CCR register. In indirect mode and automatic polling mode, the address bytes to be sent are specified in the ADDRESS[31:0] field of the R32_QSPIx_AR register. In memory-mapped mode, the address is provided directly b

By configuring the ADMODE[1:0] field in the R32_QSPIx_CCR register, the address phase can transmit 1 bit (via SIO0 in single-wire SPI mode), 2 bits (via SIO0/SIO1 in two-wire SPI mode), or 4 bits (via SIO0/SIO1/SIO2/SIO3 in four-wire SPI mode) at a time. If ADMODE=00b, the address phase is skipped, and the command sequence proceeds directly to the next phase (if it exists).

In the alternating

byte phase , bytes 1-4 are sent to the FLASH memory, typically used to control the operating mode. The number of alternating bytes to be sent is configured in the ABSIZE[1:0] fields of the R32_QSPIx_CCR register. The specified bytes are configured in the

R32_QSPIx_ABR register. The alternating byte phase can send 1 bit (via SIO0 in single-wire SPI mode), 2 bits (via SIO0/SIO1 in dual-wire SPI mode), or 4 bits (via SIO0/SIO1/SIO2/SIO3 in quad-wire SPI mode) at a time, configured via the ABMODE[1:0] fields in the R32_QSPIx_CCR register. If ABMODE=00b, the alternating byte phase is skipped, and the command sequence proceeds directly to the next phase. (If present).

During the alternating byte phase, sometimes only a single half-byte needs to be sent instead of a full byte. For example: in two-wire mode, when alternating bytes are sent using only two cycles. In this case, the firmware can use four-wire mode (ABMODE=11b) and send one byte by setting bits [7] and [3] of ALTERNATE to 1 (SIO3 is held high) and bits [6] and [2] to 0 (SIO2 is held low). In this case, the high 2 bits of the half-byte are stored in bits [4:3] of ALTERNATE, and the low 2 bits are stored in bits [1] and [0]. For example, if half-byte 2 (0010) is sent via SIO0/SIO1, then ALTERNATE should be set to 0x8A

(1000_1010). Empty

instruction cycle phase During the empty instruction cycle phase, when using a higher clock frequency, in order to ensure that the FLASH has enough time to prepare data, no data should be sent or received within the given 1-31 cycles. The number of cycles in this phase is specified by

the DCYC[4:0] fields of the R32_QSPIx_CCR[22:18] register. In SDR mode, the duration is determined by a certain number of full clock cycles. If DCYC

The command sequence proceeds directly to the data phase (if it exists).

The no-instruction cycle phase operates in a fixed single-

threaded

mode. During the data phase , any number of bytes can be received from or sent to the FLASH memory. In indirect mode and auto-polling mode, the number of bytes to be received/sent is specified in the R32_QSPIx_DLR register. In indirect write mode, the data to be sent to the FLASH memory must be written to the R32_QSPIx_DR register. In indirect read mode, data received from the FLASH memory is obtained by reading the R32_QSPIx_DR register.

In memory-mapped mode, the read data is directly sent back to

the system. By configuring the ABMODE[1:0] field in the R32_QSPIx_CCR register, the data phase can send/receive 1 bit (via SIO0 in single-wire SPI mode), 2 bits (via SIO0/SIO1 in two-wire SPI mode), or 4 bits (via SIO0/SIO1/SIO2/SIO3 in four-wire SPI mode) at a time. If DMODE=00b, the data phase is skipped, and the command sequence is completed immediately upon pulling nCS high. This configuration only applies to indirect write mode.

37.3.2 QSPI Signal Interface Protocol Modes

Single-Wire SPI

Mode The traditional SPI mode allows serial transmission/reception of a single bit. In this mode, data is sent to the FLASH via SIO0. From Data received by the FLASH is delivered via SIO1.

To use this single bit mode, you can configure different command stages separately by setting the IMODE/ADMODE/ABMODE/DMODE fields in R32_QSPIx_CCR to 01b.

In each phase configured for single-line mode:

• SIO0 is in output mode . •

SIO1 is in input mode (high impedance). •

SIO2 is in output mode and forced to 0 (to disable "write protection"). • SIO3

is in output mode and forced to 1 (to disable "hold").

If DMODE=01b, the same applies to the no-instruction phase. In two-

wire SPI mode , two

bits are sent/received simultaneously via the SIO0/SIO1 signals. This is achieved by adjusting the R32_QSPIx_CCR register.

The IMODE/ADMODE/ABMODE/DMODE fields, configured to 10b, allow for corresponding settings for different command phases.

In each phase configured for dual-line mode: ÿ SIO0/

SIO1 are in a high-impedance state (input) during data read operations, and output under other conditions. ÿ SIO2 is in output

mode and forced to 0. ÿ SIO3 is in output mode and

forced to 1.

During the no-instruction phase, if DMODE=01b, SIO0/SIO1 remain in a high-impedance state.

In **four-wire SPI**

mode , four bits of data are simultaneously sent/received via the SIO0/SIO1/SIO2/SIO3 signals. This is achieved by using R32_QSPIx_CCR.

The IMODE/ADMODE/ABMODE/DMODE fields of the register are configured as 11b, which allows for corresponding settings for different command stages.

In each stage configured for four-line mode:

ÿ SIO0/SIO1/SIO2/SIO3 are all in a high-impedance state (input) during the data read phase, and are outputs under other conditions. ÿ SIO2 and SIO3 are only used in QSPI mode. If no phase is configured to use four-wire SPI mode, even if QSPI is active, the corresponding pins of SIO2 and SIO3 can be used for other functions.

During the no-instruction phase, if DMODE=11, then SIO0/SIO1/SIO2/SIO3 are all in a high-impedance state.

SDR Mode:

QSPI operates in Single Data Rate (SDR) mode. In SDR mode,

when QSPI drives the SIO0, SIO1, SIO2, and SIO3 signals, these signals transition only on the falling edge of SCK. In SDR mode, when receiving data,

QSPI assumes that the FLASH transmits data via the falling edge of SCK, using subsequent edges of SCK.

The signal is sampled along the rising edge.

In dual flash

mode , QSPI enters dual flash mode when the DFM bit (R32_QSPIx_CR[6]) is set to 1. QSPI uses two external four-wire SPI FLASH (FLASH1 and FLASH2), sending/receiving 8 bits of data per cycle, effectively doubling throughput and capacity. Each FLASH uses the same SCK and can optionally use the same nCS signal, but its SIO0, SIO1, SIO2, and SIO3 signals are independent. Dual flash mode can be used in conjunction with single-bit mode, double-

bit mode, four-bit mode, and SDR mode. The size of the FLASH is specified in FSIZE[4:0] (R32_QSPIx_DCR[20:16]),

and the specified parameters should reflect the total capacity of the FLASH, i.e., twice the capacity of a single component. When address X is even, the byte assigned to address X by QSPI is stored in address X/2 of FLASH1, and the byte assigned to address X+1 by QSPI is stored in address X/2 of FLASH2. That is, bytes at even addresses are stored in FLASH1, and bytes at odd addresses are stored in FLASH2.

In dual-flash mode, the number of bytes required to read the FLASH status register is twice that in single-flash mode. That is, if each FLASH provides 8 valid bits after executing the status register fetch instruction, QSPI must be configured to read 2 bytes (16 bits) of data, and QSPI must receive 1 byte from each FLASH. If each FLASH provides 16 bits of status information, QSPI must be configured to read 4 bytes to obtain all status bits from both FLASH in dual-flash mode. The least significant byte in the resulting data register is the least significant byte of the FLASH1 status register, and the next byte is the least significant byte of the FLASH2 status register. The third byte of the data register is the second byte of FLASH1, and the fourth byte is the second byte of FLASH2 (if the FLASH has a 16-bit status register).

In dual-flash mode, an even number of bytes must always be accessed. Therefore, when DRM=1, the data length field (R32_QSPIx_DLR[0]) is used.

Bit 0 is always kept at 1. In dual-

flash mode, the behavior of the FLASH1 interface signals is basically the same as in normal mode. During the instruction, address, alternating byte, and empty instruction cycles, the waveforms of the FLASH2 interface signals are exactly the same as those of the FLASH1 interface signals. That is, each FLASH always receives the same instructions and addresses. Then, during the data phase, the SIOx and SIOxx buses transmit data in parallel, but the data sent to (or received from) FLASH1 is different from that in FLASH2.

37.3.3 QSPI Indirect Mode

In indirect mode, commands are triggered by writing to the QSPI register; data is transmitted by reading and writing to the data register, similar to other communication peripherals.

When FMODE=00b (R32_QSPIx_CCR[27:26]), QSPI is in indirect write mode, sending bytes to FLASH during the data phase. Data is provided by writing to the data register (R32_QSPIx_DR); when FMODE=01b, QSPI is in indirect read mode, receiving bytes from FLASH during the data phase. Data is obtained by reading from the data register (R32_QSPIx_DR).

The number of bytes to read/write is specified in the data length register (R32_QSPIx_DLR). If R32_QSPIx_DLR = 0xFFFF_FFFF (all bits set to 1), the data length is considered undefined, and QSPI will continue transferring data until the end of the FLASH (defined by FSIZE) is reached. If no bytes are transferred, the DMODE bit of the R32_QSPIx_CCR register is set to 00b. If R32_QSPIx_DLR = 0xFFFF_FFFF and FSIZE = 0x1F (data length is 4GB), when accessing flash in close-mode, the configuration should ensure that flash address + DLR <= FSIZE. In this case, the transfer will continue indefinitely, stopping only after QSPI is disabled or a termination request is received. After reading the last memory address (address 0xFFFF_FFFF), reading will continue from address 0x0000_0000. When the programmed number of bytes to be sent or received is reached, if

TCIE=1, TCF is set to 1 and an interrupt is generated. When the amount of data is uncertain, the system will set TCF to 1 based on the FLASH size defined in the R32_QSPIx_CR register when the external SPI limit is reached.

QSPI supports write flow control, which means that when the length filled in the DLR is not fully transmitted and the FIFO is empty, the transmission shift register is paused. In this mode, for efficiency reasons, the software should first write data to the DR register to fill the FIFO, or set DMAEN to 1 to automatically request some data from the DMA before writing the START bit to start transmission.

In essence, command-

triggered startup occurs once the firmware provides the final information required by the command. Depending on the QSPI configuration, in indirect mode... The three ways to trigger command startup are as

follows: 1) Write to the R32_QSPIx_CCR register INSTRUCTION[7:0] if no address is required (when ADMODE=00b) and no data is required from the firmware (when FMODE=01b or DMODE=00b); 2) Write to the R32_QSPIx_AR register

ADDRESS[31:0] if the address is required (when ADMODE=00b) and no data is required from the firmware (when FMODE=01b or DMODE=00b); 3) Write to the R32_QSPIx_DR register DATA[31:0] if the address is required (when

ADMODE!=00b) and data is required from the firmware (when FMODE=00b and DMODE!=00b). Writing to the alternating byte register (R32_QSPIx_ABR) will never trigger command startup. If alternating bytes are required, they must be

written beforehand.

Programming begins. Once the command is initiated, the BUSY bit in the R32_QSPIx_SR register will be automatically set to 1.

In indirect mode, data is

transferred via a 32-byte FIFO within QSPI. The R32_QSPIx_SR register FLEVEL[5:0] handles FIFO and data management.

Indicates how many bytes the FIFO currently stores.

In indirect write mode (FMODE=00b), when the firmware writes data to the R32_QSPIx_DR register, the data is added to the FIFO. A word write adds 4 bytes to the FIFO, a half-word write adds 2 bytes, and a byte write adds only 1 byte. If the data added to the FIFO by the firmware exceeds the value indicated by DL[31:0], the excess bytes are cleared from the FIFO when the write operation is complete (TCF is set to 1). Byte/half-word accesses to the R32_QSPIx_DR register must be strictly limited to the least significant byte/half-word of that 32-bit register.

FTHRES[4:0] is used to define the threshold of the FIFO. When the threshold is reached, FTF (FIFO threshold flag) is set to 1. In the indirect reading mode...

Under this formula, FTF is set to 1 when the number of valid bytes read from the FIFO exceeds the threshold. After reading the last byte from the FLASH, if there is still data in the FIFO, FTF will be set to 1 regardless of the setting of FTHRES. In indirect write mode, FTF is set to 1 when the number of empty bytes in the FIFO exceeds the threshold. If FTIE=1, an interrupt is generated when FTF is set to 1. Once the threshold condition is no longer met, i.e., after the CPU or DMA has transferred enough data, HW will clear FTF to zero.

In indirect mode, QSPI pauses reading bytes from Flash when the FIFO is full to avoid overflow. Reading from Flash resumes only when the FIFO is empty (FLEVEL ÿ 11). Therefore, if FTHRES ÿ 13, the application must read a sufficient number of bytes to ensure QSPI can continue retrieving data from Flash. Conversely, as long as 11 < FLEVEL < FTHRES, the FTF flag will always be 0.

37.3.4 QSPI Status Flag Polling Mode

In auto-polling mode, QSPI periodically initiates commands to read a certain number of status bytes (up to 4). Some status bits can be isolated by masking the received bytes. An interrupt can be generated when the selected bit has a preset value. Access to the FLASH begins in

the same way as in indirect read mode: if no address is needed (AMODE=00b), access begins when writing to R32_QSPIx_CCR. Conversely, if an address is needed, the first access begins when writing to R32_QSPIx_AR. At this time, BUSY goes high and remains high even during periodic accesses. In auto-polling mode, the contents of the R32_QSPIx_PSMAR register MASK[31:0] are used to mask data from

the FLASH. If MASK[n]=0, bit n in the result is masked and ignored. If MASK[n]=1, and the contents of bit [n] are the same as those in the R32_QSPIx_PSMAR register MATCH[n], a match exists for bit n.

If the polling match mode bit (PMM, bit 23 of R32_QSPIx_CR) is 0, the "AND" match mode is activated, meaning the Status Match Flag (SMF) is set to 1 only when all unmasked bits match. If PMM=1, the "OR" match mode is activated, meaning the SMF is set to 1 if a match is found on any unmasked bit. If SMIE=1, an interrupt is invoked when the SMF is set to 1. If the automatic polling mode stops (APMS bit is 1), the operation stops, and the BUSY bit is cleared when a match is detected. Otherwise, the BUSY bit remains 1, and periodic access continues until an abort or QSPI is disabled (EN=0).

The data register (R32_QSPIx_DR) contains the most recently received status byte (FIFO disabled). The contents of the data register are not subject to matching. The impact of the masking method used in the configuration logic. The FTF status bit is set to 1 after a new status read is completed, and cleared to zero once data is read.

37.3.5 QSPI Memory Mapping Mode

When configured in memory-mapped mode, external SPI devices are treated as internal memory. If the QSPI peripheral is not properly configured and enabled, the QSPI Flash memory area will be inaccessible. Even with a larger FLASH capacity, its address space is limited to a maximum of 256MB. Note: Memory-mapped

addresses can only be used for data access, not instruction access. Byte, half-word, and word access

types are supported. QSPI accepts the next microcontroller access and preloads the bytes at the following address.

If subsequent accesses are to consecutive addresses, the access process can be accelerated because the data has already been prefetched.

By default, QSPI never interrupts its prefetch operation, maintaining previous read operations even if the FLASH has not been accessed for a long time, and nCS will remain low. However, FLASH power consumption increases when nCS is low. Therefore, the application may activate the timeout counter (TCEN=1, bit 3 of R32_QSPIx_CR). This releases nCS after the FIFO is filled with prefetched data and there has been no access within the duration of TIMEOUT[15:0] (R32_QSPIx_LPTR) cycles. The BUSY signal goes high once the first memory-mapped access occurs. Due to the prefetch operation, the BUSY signal only drops after a timeout, abort, or peripheral disable.

37.3.6 The QSPI FLASH configuration device

configuration register (R32_QSPIx_DCR) is used to define the characteristic parameters of the external SPI FLASH. Among them, FSIZE[4:0]

The capacity of the external storage is determined according to the following formula:

Number of bytes in FLASH = 2[FSIZE+1]

FSIZE+1 is the number of address bits required for FLASH addressing. In indirect mode, the maximum FLASH capacity is 4GB (using 32-bit).

(Addressing is performed), but the addressable space is limited to 256MB in memory-mapped mode. If DFM=1, FSIZE represents the sum of the capacities of the two FLASH chips.

When QSPI executes two commands consecutively, it sets the chip select signal (nCS) high between the two commands, by default for only one SCK cycle. If the FLASH requires a longer time between commands, the chip select high time (CSHT) field can be used to specify the minimum number of SCK cycles (maximum 8) that nCS must remain high. The clock mode (CKMODE) bit indicates the logic level of the SCK signal between commands (when nCS=1).

37.3.7 QSPI Configuration

QSPI configuration involves two phases: QSPI IP configuration and QSPI FLASH configuration. Once QSPI is configured and enabled, its three functions can be used. One of the operating modes: indirect mode, status polling mode, and memory mapping mode.

QSPI IP Configuration: Configured via R32_QSPIx_CR. The user must configure the division factor of the clock prescaler for incoming data and the sampling shift settings. The FIFO level for generating the DMA trigger signal or interrupt is programmed in the FTHRES bit. If a timeout counter is required, the TCEN bit can be set to 1 and the timeout value programmed in the R32_QSPIx_LPTR register. Dual flash mode can be activated by setting DFM to 1.

QSPI FLASH Configuration: Parameters related to the external target FLASH are configured via the R32_QSPIx_DCR register. The user must program the FLASH size in the FSIZE bit, the minimum time the chip select is held high in the CSHT bit, and set the function mode (mode 0 or mode 3) in the MODE bit.

37.3.8 QSPI Usage: The operating mode

is selected via the R32_QSPIx_CCR register FMODE[1:0]. Indirect Mode: When FMODE=00b,

indirect write

mode is selected, allowing data to be sent to the FLASH. When FMODE=01b, indirect read mode is selected.

Get the mode and read the data from the FLASH.

When QSPI is used in indirect mode, frames are constructed in the following manner:

1) Specify the target address in the R32_QSPIx_AR register; 2) Specify the

operating mode in the R32_QSPIx_CR register; 3) Specify the number of bytes

to be read or written in the R32_QSPIx_DLR register; 4) Specify the frame format, mode, and instruction

code in the R32_QSPIx_CCR register; 5) Read data from/write data to the FIFO via the R32_QSPIx_DR

register; 6) Specify the optional alternating bytes to be sent immediately after the address phase in the

R32_QSPIx_ABR register. When writing to the control register (R32_QSPIx_CR), the user can specify the following settings: ÿ Enable bit

(EN) set to 1 ÿ Timeout counter enable bit (TCEN) ÿ FIFO threshold (FTRHES) to indicate when the FTF flag is

set to 1 ÿ Interrupt enable ÿ Auto-polling mode

parameters: matched mode and stop mode (valid

when FMODE=11b) ÿ Clock prescaler: When writing to the communication configuration register

(R32_QSPIx_CCR), the

user specifies the following parameters:

• Specifies the number of null instruction bytes via the DCYC

bit . • Specifies the presence of null instruction bytes via the DBMODE bit . •

Specifies the instruction bytes via the INSTRUCTION bit . • Specifies

the number of alternating bytes via the ABSIZE bit (1/2/3/4) . • Specifies the

instruction transmission mode via the IMODE bit (1/2/4 lines) . • Specifies the address

length via the ADSIZE bit (8/16/24/32 bits) . • Specifies the address transmission mode via

the ADMODE bit (none/1/2/4 lines).

ÿSpecify the alternating byte transmission mode (no /1/2/4 lines) via the ABMODE bit .

ÿ Specify the data transmission/reception mode (none/1/2/4 lines) via the DMODE bit .

If the data register (R32_QSPIx_DR) and address register (R32_QSPIx_AR) do not need to be updated for a certain command, then in

This command sequence is initiated immediately upon writing to the R32_QSPIx_CCR register. This occurs when both ADMODE and DMODE are 00b, or indirectly...

This situation occurs when only ADMODE=00b is set in read mode (FMODE=01b). It also occurs when an address is required (ADMODE!=00) and data is registered.

When the device does not require writing (FMODE=01b or DMODE=00b), once the address is updated by writing to the R32_QSPIx_AR register, the command sequence...

The column then starts immediately. In the case of data transmission (FMODE=00b and DMODE!=00b), data is transmitted via the R32_QSPIx_DR register to...

Writing to a FIFO will trigger the start of communication.

Enable Status Flag Polling

Mode : Set the FMODE field of the R32_QSPIx_CCR register to 10b. In this mode, status flag polling mode will be sent...

The program frames and periodically retrieves data. The maximum amount of data read in each frame is 4 bytes. If the R32_QSPIx_DLR register requests data...

Data exceeding 4 bytes will be ignored, and only 4 bytes will be read. The periodicity is specified by the R32_QSPIx_PISR register.

After retrieving the status data, it can be processed internally to achieve the following objectives:

Automatically stop periodic retrieval of status bytes

Set the state matching flag to 1 and generate an interrupt when enabled.

The received value can be masked and compared with the value stored in the R32_QSPIx_PSMKR register. ,

The values in the R32_QSPIx_PSMAR register are ORed and ANDed. If a match is found, the status match flag is set to 1, and the value is then used to perform an OR operation.

An interrupt will also be generated if an interrupt is possible; if AMPS is set to 1, QSPI will automatically stop. In all cases, the latest retrieval...

The values are available in the R32_QSPIx_DR register.

In memory - mapped

mode, despite access latency, external FLASH is still treated as internal storage. This mode only supports...

Perform a read operation on the external FLASH. Memory mapping can be activated by setting FMODE in the R32_QSPIx_CCR register to 11b.

Shooting mode. When the master device accesses the memory-mapped space, it will send configured frames and instructions.

The FIFO is used as a prefetch buffer to support sequential linear read operations. In this mode, any data stored in the R32_QSPIx_DR register...

Access to the register always returns zero. In memory-mapped mode, the data length register (R32_QSPIx_DLR) is invalid.

37.3.9 Instructions are sent only once. Some

FLASH (e.g., Winbound) can provide a mode in which instructions must be sent only with the first command sequence.

The first command is executed based on the address, and subsequent commands are executed directly based on the address. Users can implement this feature by using the SIOO bit of the R32_QSPIx_CCR register.

The SIOO bit applies to all functional modes: indirect mode, status polling mode, and memory-mapped mode. When SIOO is set to 1, only...

The first command is sent, and a write operation is performed on the R32_QSPIx_CCR register. Subsequent command sequences will skip the instruction phase and proceed directly to the next step.

The R32_QSPIx_CCR register is written again. However, the SIOO bit is invalid when IMODE=00b (no instruction).

37.3.10 QSPI Error Management

Possible causes of errors include:

In indirect mode or status flag polling mode, **if an incorrect address is set in the R32_QSPIx_AR register, i.e., according to...**

The FLASH size defined in the R32_QSPIx_DCR register FSIZE[4:0] will be set to 1, and if enabled, an interrupt will also be generated.

In indirect mode, **if the length of the address plus the data exceeds the size of the FLASH, TEF will be set to 1 when the access is triggered.**

37.3.11 QSPI Busy Bit and Abort Function

When QSPI begins operating on the FLASH, the BUSY bit in the R32_QSPIx_SR register is automatically set to 1. In indirect mode,

QSPI completes the requested command sequence and resets the BUSY bit when the FIFO is empty; in automatic polling mode, the BUSY bit is reset only on the last polling command.

After a periodic access is completed, the BUSY bit goes low only when a match occurs (APMS=1) or due to an aborted operation. In memory mapping...

In shoot mode, after the first access, the BUSY bit will only go low if a timeout event or abort occurs.

Any operation can be aborted by setting the ABORT bit to 1 in the R32_QSPIx_CR register. Once the abort is complete, the BUSY bit and The ABORT bit is automatically reset, and the FIFO is cleared.

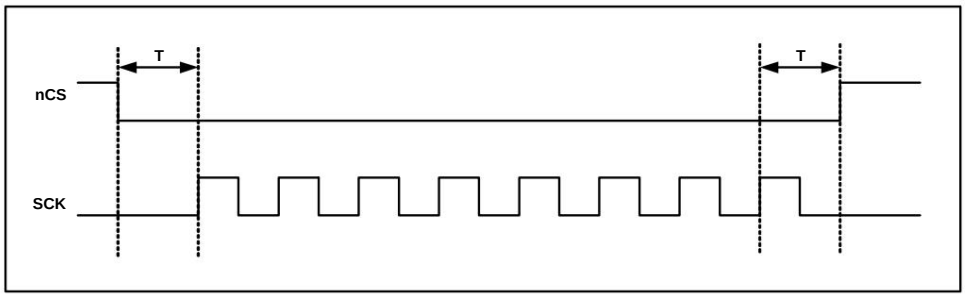
Note: Some FLASH devices may exhibit abnormal behavior if write operations to the status register are aborted.

37.3.12 nCS Behavior By

default, nCS is high, deselecting external FLASH. nCS drops before the start of an operation and rises immediately after the operation is complete. When

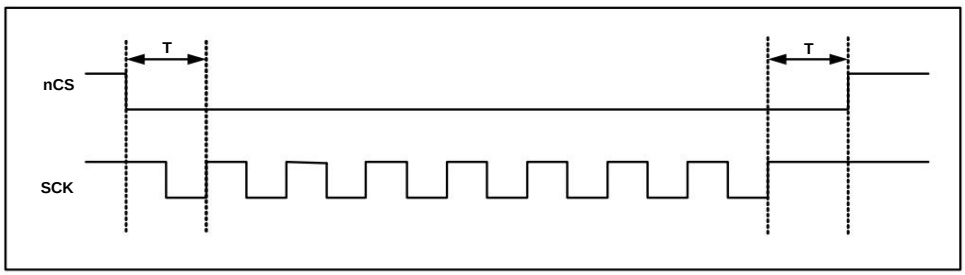
CKMODE=0 ("Mode 0", i.e., SCK remains low when no operation is performed), nCS drops low one SCK cycle before the first rising SCK edge of the operation, and rises high one SCK cycle after the last rising SCK edge of the operation, as shown in the figure below.

Figure 37-4 nCS (T=SCK period) when CKMODE=0



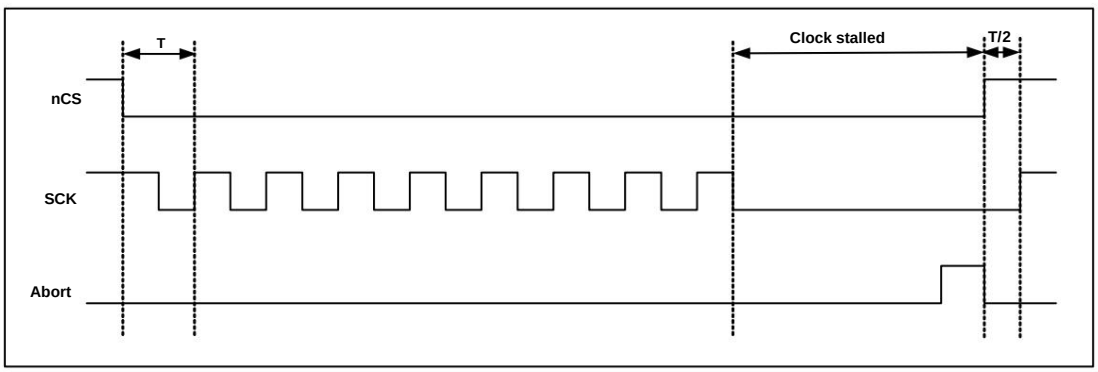
When CKMODE=1 ("Mode 3", SCK rises high when no operation is performed), nCS still drops low one SCK cycle before the first rising SCK edge, and rises high one SCK cycle after the last rising SCK edge, as shown in the figure below.

Figure 37-5 nCS (T=SCK period) when CKMODE=1



If the FIFO remains full during a read operation or empty during a write operation, the operation stops and SCK remains low until firmware intervention occurs. If an abort occurs when the operation stops, nCS rises high immediately after the abort request, and SCK rises high after half a cycle, as shown in Figure 37-6. Figure 37-6 nCS when

CKMODE=1 and an abort occurs (T=SCK cycle)



When not in dual-flash mode (DFM=0) and FSEL=0 (default), only FLASH1 is accessed. Therefore, if FSEL=1, only FLASH1 is accessed. When asked, FLASH1 maintains a high nCS level. In dual-flash mode, the nCS behavior of FLASH1 and FLASH2 is identical. Therefore, if If FLASH2 exists and the application is always in dual-flash mode, then the nCS signal of FLASH1 can also be used for FLASH2. The nCS pin output of FLASH2 can be used for other functions.

37.3.13 Interruption

Table 37-1 Interrupt Requests

Interruption event	Event marker	Enable control bit
Timeout	TOF	TOIE
status matching	SMF	SMIE
FIFO threshold	FTF	FTIE
transmission	TCF	TCIE
completed transmission error	TEF	TEIE

37.4 Register Description

Table 37-2 List of QSPI1 Relevant Registers

name	Access address	describe	Reset value
R32_QSPI1_CR	0x40024C00 QSPI1 Control Register;		0x00000000
R32_QSPI1_DCR	0x40024C04 QSPI1 Device Configuration Register;		0x00000000
R32_QSPI1_SR	0x40024C08 QSPI1 Status Register; 0x40024C0C		0x00000000
R32_QSPI1_FCR	QSPI1 Flag Clear Register; 0x40024C10 QSPI1 Data		0x00000000
R32_QSPI1_DLR	Length Register; 0x40024C14 QSPI1 Communication		0x00000000
R32_QSPI1_CCR	Configuration Register; 0x40024C18 QSPI1 Address		0x00000000
R32_QSPI1_AR	Register; 0x40024C1C QSPI1 Alternate Byte		0x00000000
R32_QSPI1_ABR	Register; 0x40024C20 QSPI1 Data Register;		0x00000000
R32_QSPI1_DR	0x40024C24 QSPI1 Polling Status Mask		0x00000000
R32_QSPI1_PSMKR	Register; 0x40024C28 QSPI1 Polling Status Match Register;		0x00000000
R32_QSPI1_PSMAR	0x40024C2C QSPI1 Polling Interval Register; 0x40024C30		0x00000000
R32_QSPI1_PIR	QSPI1 Low Power Timeout Register		0x00000000
R32_QSPI1_LPTR			0x00000000

Table 37-3 List of QSPI2 Related Registers

name	Access address	describe	Reset value
R32_QSPI2_CR	0x40025000 QSPI2 Control Register; 0x40025004		0x00000000
R32_QSPI2_DCR	QSPI2 Device Configuration Register; 0x40025008		0x00000000
R32_QSPI2_SR	QSPI2 Status Register; 0x4002500C QSPI2 Flag		0x00000000
R32_QSPI2_FCR	Clear Register; 0x40025010 QSPI2 Data Length		0x00000000
R32_QSPI2_DLR	Register; 0x40025014 QSPI2 Communication		0x00000000
R32_QSPI2_CCR	Configuration Register; 0x40025018 QSPI2 Address		0x00000000
R32_QSPI2_AR	Register; 0x4002501C QSPI2 Alternate Byte		0x00000000
R32_QSPI2_ABR	Register; 0x40025020 QSPI2 Data Register; 0x40025024		0x00000000
R32_QSPI2_DR	QSPI2 Polling Status Mask Register		0x00000000
R32_QSPI2_PSMKR			0x00000000

R32_QSPI2_PSMAR	0x40025028 QSPI2 Polling Status Matching Register;	0x00000000
R32_QSPI2_PIR	0x4002502C QSPI2 Polling Interval Register;	0x00000000
R32_QSPI2_LPTR	0x40025030 QSPI2 Low Power Timeout Register	0x00000000

37.4.1 QSPI Control Register (R32_QSPIx_CR) (x=1/2)

Offset address: 0x00

31 30 29 28 27 26 25 24 23	twenty two	twenty one	2019	18	17 16
PRESCALER[7:0]	PMM	APMS	Reserved	TOIE SMIE FTIE	TCIE TEIE

15 14	13	12 11 10 9 8	7	6	5	4	3	2	1	0
Reserved	SIOXEN	FTHRES[4:0]	FSEL	DFM	START	Reserved	TCEN	DMAEN	ABORT	EN

Bit	name	access	describe	Reset value
[31:24]	PRESCALER[7:0]	RW	Clock prescaler: This field defines the frequency division used to generate SCK based on the HB bus clock. Coefficient (value + 1). 0x0: FSCK=FHCLK, HCLK clock is directly used as QSPI SCK (pre-processing clock). The frequency divider is bypassed. 0x1: FSCK = FHCLK / 2; 0x2: FSCK=FHCLK/3; ... 0xFF: FSCK=FHCLK/256. For odd-numbered clock division factors, the duty cycle of SCK is not 50%. The low level of the clock signal lasts longer than the high level. One cycle. This field can only be modified when BUSY=0.	0
23	PMM	RW	Polling matching mode: 1: OR Matching mode. If any unmatched data is received from FLASH... If the mask bit matches the corresponding bit in the match register, then the SMF is set. 1; 0: AND matching mode. If all unscreened data received from FLASH... If all mask bits match the corresponding bits in the match register, then the SMF is set. 1. This bit can only be modified when BUSY=0.	0
22	APMS	RW	Automatic polling mode stopped: 1: When a match occurs, the automatic polling mode stops; 0: Stop automatic polling mode only by aborting or disabling QSPI. This bit can only be modified when BUSY=0.	0
twenty one	Reserved	RO	is reserved.	0
20	TOIE	RW	TimeOut interrupt enable: 1: Enable interrupt; 0: Interruption is disabled.	0
19	SMIE	RW	status matching interrupt enabled:	0

			1: Enable interrupt; 0: Interruption is disabled.	
18 FTIE		RW	FIFO threshold interrupt enable: 1: Enable interrupt; 0: Interruption is disabled.	0
17 TCIE		RW	Transmission completion interrupt enable: 1: Enable interrupt; 0: Interruption is disabled.	0
16 TEIE		RW	Transmission error interrupt enable: 1: Enable interrupt; 0: Interruption is disabled.	0
[15:14] Reserved		RO is reserved.		0
13 SIOXEN		RW	SIOX Enable: 1: 4-wire I/O output enable; 0: 4-line I/O output is off.	0
[12:8] FTHRES[4:0]		RW	FIFO threshold levels: Defining the FIFO threshold flag in indirect mode will result in a FIFO threshold flag. (FTF, R32_QSPIx_SR[2]) Byte count threshold set to 1. In indirect write mode (FMODE=00): 00000: If there are one or more free bytes in the FIFO, it is possible to... If it is ready to be written, then FTF is set to 1; 00001: If there are 2 or more free bytes in the FIFO, it is possible to... If it is ready to be written, then FTF is set to 1; ... 11111: If there are 32 free bytes available for writing in the FIFO. If input is received, the FTF is set to 1. In indirect read mode (FMODE=01): 00000: If there is one or more valid bytes in the FIFO. If it is available for reading, then FTF is set to 1; 00001: If there are 2 or more valid bytes in the FIFO... If it is available for reading, then FTF is set to 1; ... 11111: If there are 32 valid bytes available for reading in the FIFO. If the condition is met, the FTF is set to 1.	0
7	FSEL	RW	FLASH selection: 1: Select FLASH2; 0: Select FLASH1. This bit can only be modified when BUSY=0. It is ignored when DFM=1. Bit.	0
6	DFM	RW	Dual flash memory mode: 1: Enable dual flash memory mode; 0: Disable dual flash memory mode. This bit can only be modified when BUSY=0.	0
5	START	In both indirect mode and status polling mode, the required registers for WZ are already configured to 0.		

			After setting it, write 1 to the START bit to start the flash command sequence. The column will automatically clear this bit to zero.	
4	Reserved	RO	reserved.	0
3	TCEN	RW	Timeout counter enabled: This bit is only valid in memory-mapped mode (FMODE=11). Activate After this position, if there is a period of time (via TIMEOUT[15:0]), (R32_QSPIx_LPTR) definition) has not been accessed. This will release chip select (nCS), thereby reducing power consumption. Enable the timeout counter. By default, even if the FLASH is not accessed for a long time, QSPI will not stop prefetching operations; previous read operations will be preserved. The device remains active and nCS remains low. Because nCS remains... When the level is low, the power consumption of the FLASH increases, and the application may become active. Timeout counter (TCEN=1, bit 3 of R32_QSPIx_CR). From After the FIFO is filled with prefetched data, if... , The duration of TIMEOUT[15:0] (R32_QSPIx_LPTR) cycles. If there is no access to the nCS, then release the nCS. 1: Enable the timeout counter. In memory-mapped mode, FLASH holds... Release chip selection after 15:00 cycles of inactivity TIMEOUT. (nCS); 0: Disable timeout counter, access in memory-mapped mode. Afterwards, chip select (nCS) remains active. This bit can only be modified when BUSY=0.	0
2	DMAEN	RW	DMA mode enabled: 1: Enable DMA; 0: Disable DMA.	0
1	ABORT	RW	Abort request/clear FIFO: 1: Request to abort/clear FIFO; 0: Do not request abort/clear FIFO. Abort Request: This bit halts the execution of the sequence of commands. After abort... Automatically reset upon completion. This bit can stop the current transmission. In polling or memory-mapped mode, this bit is also used for reset. APMS bit or DMAEN bit. Clear FIFO: Setting this bit to 1 will clear the FIFO. All The internal FIFO pointer (for reading and writing) will be cleared to zero. In this case, Data still remaining in the FIFO is lost (data sent or received is not lost). (It will continue to be lost). When the QSPI read/write direction is switched, the FIFO needs to be cleared once. When the IDLEF bit of the R32_QSPIx_SR register is set to 1, ABORT Enabling this bit will generate a transmission completion flag. , The TCF bit in the R32_QSPIx_SR register is set.	0
0	EN	RW	Enable: 1: Enable QSPI;	0

			0: Disable QSPI.	

37.4.2 QSPI Device Configuration Register (R32_QSPIx_DCR) (x=1/2) Offset Address: 0x04

31 30 29 28 27 26 25	twenty four	23 22	21 20 19	18	17	16
Reserved			FSIZE[4:0]			

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

Bit	name	access	describe	Reset value
[31:21] Reserved		RO is reserved.		0
[20:16] FSIZE[4:0]		RW	Flash size: The number of bytes in FLASH = 2[FSIZE+1], FSIZE+1 is the number of address bits required for FLASH addressing. Indirectly... In this mode, the FLASH capacity can reach up to 4GB (using 32-bit encoding). Row addressing), but addressable space limitations in memory-mapped mode. It is 256MB. If DFM=1, FSIZE represents the sum of the capacities of the two FLASH chips. This field can only be modified when BUSY=0.	0
[15:11] Reserved		RO reserved.		0
[10:8] CSHT[2:0]		RW	Chip select high level duration: CSHT+1 defines the chip select (nCS) in the commands sent to FLASH. The minimum number of SCK cycles that must be kept high. When the PRESCALER[7:0] bits are configured to be non-0x2: 000: nCS is kept high for at least one cycle between FLASH commands. Expect; 001: nCS remains high for at least two cycles between FLASH commands. Expect; ... 111: nCS is kept high for at least 8 cycles between FLASH commands. Expect. When the PRESCALER[7:0] bits are configured to 0x2: 000: nCS is kept high for at least one cycle between FLASH commands. Expect; 001: nCS remains high for at least two cycles between FLASH commands. Expect; ... 110: nCS remains high for at least 7 cycles between FLASH commands. Expect; 111: Reserved. This field can only be modified when BUSY=0.	0
[7:1] Reserved		RO is reserved.		0

0	CKMODE	RW	Mode 0/Mode 3: 1: When nCS is high (chip select released), SCK must remain high. Level. This is called mode 3; 0: When nCS is high (chip select released), SCK must remain low. Level. This is called mode 0. This field can only be modified when BUSY=0.	0

37.4.3 QSPI Status Register (R32_QSPIx_SR) (x=1/2)

Offset address: 0x08

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		FLEVEL[5:0]						Reserved	BUSY	TOF	SMF	FTF	TCF	TEF	

Bit	name	access	describe	Reset value
[31:17] Reserved		RO	reserved.	0
16	IDLEF	RO	Free flag: When this position is 1, the QSPI bus is in an idle state.	0
[15:14] Reserved		RO	is reserved.	0
[13:8] FLEVEL[5:0]		RO	FIFO Level: This field indicates the number of valid bytes in the FIFO. The FIFO is empty when... FLEVEL=0, FLEVEL=32 when full. In memory-mapped mode and automatic status polling mode, FLEVEL is zero.	0
[7:6] Reserved		RO	reserved.	0
5	BUSY	RO	Busy: During the operation, this position is 1. After the operation on the FLASH is completed... Furthermore, when the FIFO is empty, this bit is automatically cleared to zero.	0
4	TOF	RO	Timeout indicator: This bit is set to 1 when a timeout occurs. Writing 1 to CTOF will clear this bit. zero.	0
3	SMF	RO	Status matching flag: In automatic polling mode, if the unmasked received data matches... The corresponding bits in the register (R32_QSPIx_PSMAR) match. The bit is set to 1. Writing 1 to CSMF will clear the bit to zero.	0
2	FTF	RO	FIFO threshold flag: In indirect mode, if the FIFO threshold is reached, or if data is read from FLASH... After the data retrieval is complete, if there is still data in the FIFO, this bit will be set to 1. This applies as long as the threshold... If the value condition is no longer "true", the bit will be automatically cleared to zero. In automatic polling mode, this bit is updated each time the status register is read. This bit is set to 1; when reading the data register, this bit is cleared to 0.	0
1	TCF	RO	transmission completion flag:	0

			In indirect mode, when the amount of data transmitted reaches the programmed setting... The value, or the position 1 if transmission is aborted in any mode. To When a 1 is written to CTCF, this bit is cleared to zero.	
0	TEF	RO	Transmission error flags: This bit is set to 1 when accessing an invalid address in indirect mode. Writing 1 to CTEF will clear this bit.	0

37.4.4 QSPI Flag Clear Register (R32_QSPIx_FCR) (x=1/2) Offset Address: 0x0C

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

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

Bit	name	access	describe	Reset value
[31:5]	Reserved	RO	reserved.	0
4	CTOF	RW1Z	Clear timeout flag: Writing 1 will clear the TOF flag in the R32_QSPIx_SR register. zero.	0
3	CSMF	RW1Z	Clear the status match flag: Writing 1 will clear the SMF flag in the R32_QSPIx_SR register. zero.	0
2	Reserved	RO	reserved.	0
1	CTCF	RW1Z	Clear transfer complete flag: Writing 1 will clear the TCF flag in the R32_QSPIx_SR register. zero.	0
0	CTEF	RW1Z	Clear transmission error flags: Writing 1 will clear the TEF flag in the R32_QSPIx_SR register. zero.	0

37.4.5 QSPI Data Length Register (R32_QSPIx_DLR) (x=1/2) Offset Address: 0x10

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
DL[31:16]														

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

Bit	name	access	describe	Reset value
[31:0]	DL[31:0]	RW	Data length: The amount of data to be retrieved in indirect mode and state polling mode (Value +1). Values no greater than 3 should be used for the state polling pattern.	0

			(Indicates 4 bytes). In indirect mode, all positions 1 indicate an undefined length, QSPI Data will continue to be transmitted until it reaches the memory defined by FSIZE. end. 0x0000_0000: 1 byte is transmitted; 0x0000_0001: Transmit 2 bytes; 0x0000_0002: 3 bytes are transmitted; 0x0000_0003: Transmit 4 bytes; ... 0xFFFF_FFFD: Transfer 4,294,967,294 (4G-2) bytes. byte; 0xFFFF_FFFE: Transfer 4,294,967,295 (4G-1) bytes. byte; 0xFFFF_FFFF: Undefined length -- transmit all bytes up to the specified length. Reach the end of the FLASH defined by FSIZE. If If FSIZE=0x1F, then reading will continue indefinitely. In dual flash memory mode (DFM=1), even if this bit is written to 0, DL[0] It also always remains at 1, thus ensuring that an even number of records are transmitted on each access. 1 byte. This field does not work in memory-mapped mode (FMODE=10); This field can only be written when BUSY=0.

37.4.6 QSPI Communication Configuration Register (R32_QSPIx_CCR) (x=1/2) Offset

Address: 0x14

31 30 29 28			27 26 25 24				23 22 21	2019					18	17 16
Reserved			SIOO	FMODE [1:0]		DMODE [1:0]	Rese rved	DCYC[4:0]					ABSIZE [1:0]	

15 14	13 12	11 10	9 8	7 6 5 4 3 2 1 0
ABMODE [1:0]	ADSIZE [1:0]	ADMODE [1:0]	IMODE [1:0]	INSTRUCTION[7:0]

Bit	name	access	describe	Reset value
[31:29]	Reserved	RO	reserved.	0
28	SIOO	RW	One-time command sending mode: When IMODE=00, this bit has no effect. 1: Send instructions only for the first command; 0: Send instructions in each transaction. This field can only be written when BUSY=0.	0
[27:26]	FMODE[1:0]	RW	Functional Modes: 00: Indirect write mode; 01: Indirect read mode; 10: Automatic polling mode; 11: Memory mapping mode. This field can only be written when BUSY=0.	0

[25:24] DMODE[1:0]	RW	Data pattern: 00: No data; 01: Single-line data transmission; 10: Dual-line data transmission; 11: Four-wire data transmission. This field also defines the empty instruction phase. This field can only be written when BUSY=0.	0
23 Reserved	RO is reserved.		0
[22:18] DCYC[4:0]	RW	Null instruction cycle count: This field defines the duration of the empty instruction phase. In SDR mode... Below, it specifies the number of SCK cycles (0-31). This field can only be written when BUSY=0.	0
[17:16] ABSIZE[1:0]	RW	Alternating byte length: 00: 8-bit alternating byte; 01: 16-bit alternating byte; 10: 24 alternating bytes; 11: 32 alternating bytes. This field can only be written when BUSY=0.	0
[15:14] ABMODE[1:0]	RW	Alternating byte pattern: 00: No alternating bytes; 01: Single-wire transmission of alternating bytes; 10: Two-wire transmission of alternating bytes; 11: Four-wire transmission of alternating bytes. This field can only be written when BUSY=0.	0
[13:12] ADSIZE[1:0]	RW	Address length: 00: 8-bit address; 01: 16-bit address; 10: 24-bit address; 11: 32-bit address. This field can only be written when BUSY=0.	0
[11:10] ADMODE[1:0]	RW	Address mode: 00: No address; 01: Single-line transmission address; 10: Dual-line transmission address; 11: Four-wire transmission address. This field can only be written when BUSY=0.	0
[9:8] IMODE[1:0]	RW	Command mode: 00: No instruction; 01: Single-line transmission command; 10: Two-wire transmission command; 11: Four-wire transmission command; This field can only be written when BUSY=0.	0
[7:0] INSTRUCTION[7:0]	RW	instruction: Specifies the instructions to be sent to an external SPI device. This field can only be written when BUSY=0.	0

37.4.7 QSPI Address Register (R32_QSPIx_AR) (x=1/2)

Offset address: 0x18

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
ADDRESS[31:16]														

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

Bit	name	access	describe	Reset value
[31:0]	ADDRESS[31:0]	RW	address: Specify the address to send to the external flash memory. When BUSY=0 or FMODE=11 (memory-mapped mode), it will be ignored. Write it to this field. In dual-flash mode, since the address is always an even address, ADDRESS[0] is automatically kept at 0.	0

37.4.8 QSPI Alternating Byte Register (R32_QSPIx_ABR) (x=1/2)

Offset Address: 0x1C

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
ALTERNATE[31:16]														

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

Bit	name	access	describe	Reset value
[31:0]	ALTERNATE[31:0]	RW	Alternating bytes: Optional specification to send to external SPI device immediately after address data. This field can only be written when BUSY=0.	0

37.4.9 QSPI Data Register (R32_QSPIx_DR) (x=1/2)

Offset address: 0x20

31	30	29	28	27	26	25	24	23	22	21	2019	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	describe	Reset value
[31:0]	DATA[31:0]	RW	data: Specifies the data to be exchanged with an external SPI device. In indirect write mode, the data written to this register is in the data...	0

			The data is sent to the FLASH memory at each stage, and stored in the FIFO beforehand. For example... If the FIFO is too full, writing will be paused until the FIFO has enough space. The space accepts the data to be written before continuing. In indirect mode, this register can be read (via FIFO). Data already received from FLASH. If the number of bytes contained in the FIFO is greater than... If the number of bytes required for a read operation is small and BUSY=1, the read will be paused. Continue taking data until enough data appears or the transmission is complete (in no particular order). Only then did it continue. In automatic polling mode, this register contains the last data read from FLASH (without masking). This register supports word, half-word, and byte access. In write mode, writing a byte will increment the FIFO by one word. For half-word writing, the number of characters increases by 2, while for full-word writing, the number of characters increases by 4. Similarly, in indirect read mode, byte reads will erase... In a FIFO, one byte is read in half-word and erased twice, while a word is read... If you choose one, erase 4. Access in indirect mode must be made with the least significant bit of this register. Qi: Byte reading must read DATA[7:0] while half-word reading must Read DATA[15:0].

37.4.10 QSPI Polling Status Mask Register (R32_QSPIx_PSMKR) (x=1/2)

Offset address: 0x24

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
MASK[31:16]														

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

Bit	name	access	describe	Reset value
[31:0]	MASK[31:0]	RW	Status masking: The status bytes received in polling mode are masked. For bit n: 1: Do not mask bit n of the data received in automatic polling mode. Its value is considered in the matching logic; 0: Masks bit n of the data received in automatic polling mode. The value is not considered in the matching logic. This field can only be written when BUSY=0.	0

37.4.11 QSPI Polling Status Matching Register (R32_QSPIx_PSMAR) (x=1/2)

Offset address: 0x28

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
MATCH[31:16]														

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

MATCH[15:0]

Bit	name	access	describe	Reset value
[31:0]	MATCH[31:0]	RW	State matching: This value will be compared with the mask status register for matching. This field can only be written when BUSY=0.	0

37.4.12 QSPI Polling Interval Register (R32_QSPIx_PIR) (x=1/2) Offset Address: 0x2C

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

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

Bit	name	access	describe	Reset value
[31:16]	Reserved	RO	is reserved.	0
[15:0]	INTERVAL[15:0]	RW	Polling interval: The number of SCK cycles between read operations during the automatic polling phase. This field can only be written when BUSY=0.	0

37.4.13 QSPI Low Power Timeout Register (R32_QSPIx_LPTR) (x=1/2)

Offset address: 0x30

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

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

Bit	name	access	describe	Reset value
[31:16]	Reserved	RO	reserved.	0
[15:0]	TIMEOUT[15:0]	RW	Timeout duration: In memory-mapped mode, QSPI will pre-process after each access. Retrieve the subsequent bytes and store them in the FIFO. This field indicates the... After the FIFO is full, how many SCK clock cycles does QSPI wait before allowing it to run? When nCS is raised to a high level, the FLASH is set to a low-power state. This field can only be written when BUSY=0.	0

Chapter 38 Single-Wire Protocol Main Interface (SWPMI)

Single-Wire Protocol Master Interface (SWPMI) is a full-duplex single-wire communication technology based on the ETSI TS 102 613 standard.

Fan implements Single-Wire Protocol (SWP) communication.

In SWPMI, data can be transmitted in two physical ways: the first is through the voltage domain (S1 signal) from the master device.

The first method involves data transmission from the slave device to the master device; the second method uses the current domain (S2 signal) to achieve data transmission from the slave device to the master device. (S1 signal)

The S1 signal uses pulse width modulation (PWM) for transmission, while the S2 signal transmits data from changes in current.

38.1 Main Features

Supports full - duplex communication mode

Supports automatic handling of fill positions .

Supports automatic SWP bus status management .

Provides a loopback mode for testing .

Supports automatic start-of-frame (SOF) and automatic end-of-frame (EOF) processing .

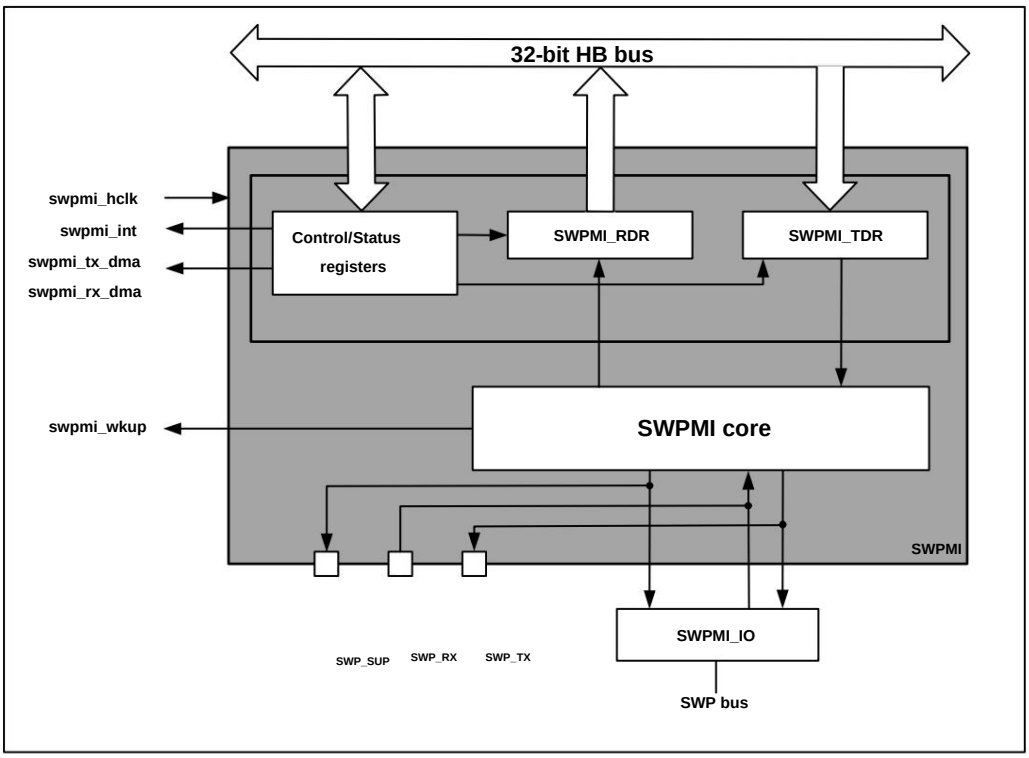
Supports configurable bit rate up to 2 Mbit/s and configurable interrupts.

Provides CRC error, underflow, overflow, and device recovery monitoring flags.

Supports CRC-16 calculation, generation, and verification.

38.2 Overview

Figure 38-1 Block diagram of the main interface of the single-line protocol



38.3 Functional Description

38.3.1 SWP Initialization and Activation

SWP initialization and activation transition the SWPMI_IO state from low to high. When using the internal transceiver, software operation is performed by pressing...

The following process will proceed:

1) Enable the SWPMI_IO transceiver by setting the SWPTEN bit to 1 in the R32_SWPMI_CR register, and set SWPMI_IO to low power.

Flat (SWP bus "disabled");

2) Wait for the R32_SWPMI_SR register RDYF bit to be set to 1 (polling flag or using the R32_SWPMI_IER register RDYIE bit to enable it).

(can be interrupted);

3) By setting the SWPACT bit to 1 in the R32_SWPMI_CR register, the SWP bus changes from "disabled" to "suspended," i.e.

Activate SWP.

38.3.2 SWP Bus Status

The SWP bus has three states: "disabled", "suspended", and "activated".

State transitions include:

ÿActivation : Changes from "Disabled" to "Pending" state;

ÿSuspended : Changes from "Activated" state to "Suspended" state;

ÿDisabled : Changes from "Pending" to "Disabled" state;

Master device recovery: The master device initiates the transition from the "suspended" state to the "activated" state;

ÿ Resume from device: The transition from "suspended" state to "activated" state initiated by the slave device.

During

the reset phase and immediately after reset, SWPMI_IO is configured in analog mode. For the SWP bus activation procedure, please refer to 38.3.1.

Subsection.

If the

SWP bus is communicating with the slave device (whether in transmit or receive mode), the SWP bus will always remain in the "active" state.

If no further send or receive requests are received, the SWP bus will switch back to the "suspended" state after 7 idle bits.

Disable

1) Disable request

When the SWP is in a "suspended" state, if communication is no longer needed, the user can request to switch the SWP by disabling the SWPMI peripheral.

To switch to the "Disabled" state, the DEACT bit of the R32_SWPMI_CR register must be set to 1. If SWPMI is not...

If the "Recover from Device" state is detected, the DEACTF bit of the R32_SWPMI_ISR register will be set to 1, and the R32_SWPMI_ICR register will also be reset.

The SWPACT bit will be cleared. If the DEACT bit is set to 1 and SWPMI detects a "recover from device" state, then the R32_SWPMI_ISR register will be cleared.

The SRF flag in register R32 will be set to 1, DEACTF will remain cleared, SWPACT will remain set to 1, and DEACT will be cleared.

Clear the DEACT bit to zero, then set the SWPACT bit to 1 to reactivate SWP.

2) Disabled mode

The user must clear the SWPACT bit of the R32_SWPMI_CR register for SWP to immediately switch to "disabled" mode and ignore...

The "recover from device" state may be passed in.

Note: To further reduce current consumption, the SWPMI_IO pin in the controller can be configured to push-pull output mode, and...

The

GPIO is set to a low level, and finally the SWPTEN bit of the R32_SWPMI_CR register is cleared.

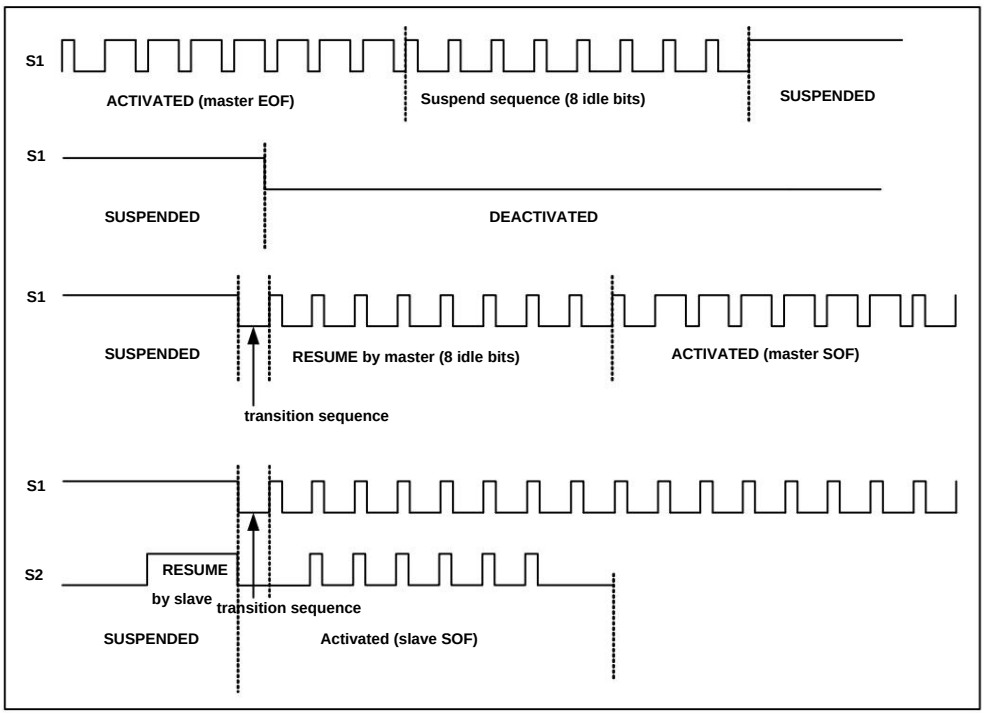
After the master

device restores SWPMI enable, the user can request SWPMI frame transmission. First, the SWPMI transmission sends a transition sequence and 8 idle bits, i.e., the master device...

Restore and then begin frame transmission. After "Master Device Restored," the SWP transitions from the "Suspended" state to the "Activated" state. Please refer to [link/reference].

Figure 38-2: SWP bus status.

Figure 38-2 SWP Bus Status Diagram



Recovery from device

When SWPMI is enabled, if SWPMI receives a recovery status signal from the slave device, SWP will transition from the "suspended" state to the "activated" state. "Live" state. "Recover from device" will trigger the SRF flag in the R32_SWPMI_ISR register to be set to 1.

38.3.3 SWPMI_IO (Internal Transceiver) Bypass

The microcontroller integrates an SWPMI_IO (internal transceiver) conforming to the ETSI TS 102 613 technical specification. To implement this transceiver... Bypassing the device can be achieved by setting the SWP_TBYP bit in the R32_SWPMI_OR register to 1. In bypass mode, SWPMI_IO is disabled, and SWP_RX is disabled. The SWP_TX and SWPMI_SUP signals are available as multiplexed functions on these three GPIOs. This configuration can be used to connect an external transceiver.

Note: In SWPMI_IO bypass mode, the SWPTEN bit of the R32_SWPMI_CR register must be cleared.

38.3.4 The SWPMI bit rate is set in the

R32_SWPMI_BRR register according to the following formula:

FSWP = FHCLK/((BR[7:0]+1)x4)

Note: The maximum bit rate is 2 Mbit/s.

38.3.5 SWPMI Frame Processing

An SWP frame consists of the following parts: Start of Frame (SOF), a payload of 1 to 30 bytes, and a 16-bit CRC (Cyclic Redundancy Check). (Verification) and EOF (End of Frame). See Figure 38-3.

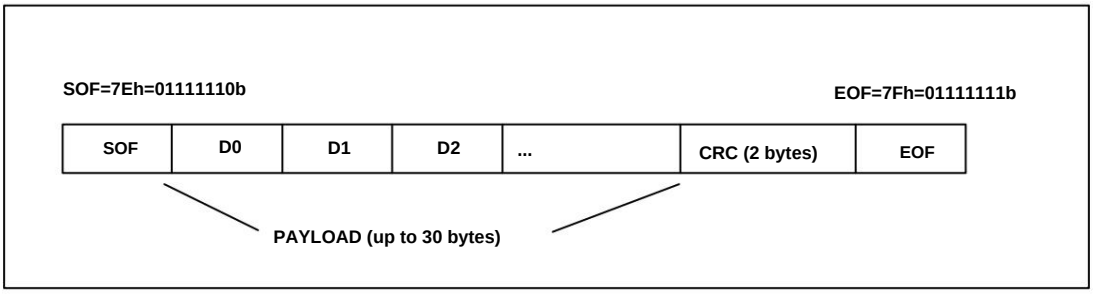
SWPMI integrates a 32-bit data transmit register (R32_SWPMI_TDR) and a 32-bit data receive register. (R32_SWPMI_RDR).

In data transmission mode, SOF insertion, EOF insertion, CRC calculation, and insertion are all automatically managed by SWPMI. Users only need to provide... The size and content of the payload. After writing data to the R32_SWPMI_TDR register, a frame will be sent immediately. It will be transmitted via a specific flag. The flag is used to indicate when the transmit data register is empty or when the data frame transmission is complete.

In data reception mode, SOF deletion, EOF deletion, CRC calculation, and checking are all automatically managed by SWPMI. Users only need to read the data. The size and content of the payload. Specific flags will indicate whether data reception is complete, the register is full, or a possible CRC error.

Error.

Figure 38-3 SWP Frame Structure



Stuff insertion (in transmit mode) and deletion (in receive mode) are both automatically managed by the SWPMI kernel. These operations are unknown to the user. Transparent.

38.3.6 Transmission Process

Before transmitting a frame, the user must activate SWP; please refer to 38.3.1. There are several software implementation methods for frame transmission, including... Software bufferless mode, single software buffer mode, and multiple software buffer mode. When using a software buffer, data flow... Data is transferred from the software buffer of RAM memory to the SWPMI peripheral transmit register via the DMA channel.

In **software-buffer-free mode ,**

DMA is not required. Polling the status flags within the main loop or SWPMI interrupt routine handles SWP frame transmission. Send. SWPMI contains a 32-bit transmit data register (R32_SWPMI_TDR). A write operation to this register will trigger up to [number missing] times. A 4-byte send operation.

The no-software buffer mode can be selected by clearing the TXDMA bit of the R32_SWPMI_CR register.

A frame transmission is triggered the first time data is written to the R32_SWPMI_TDR register. In a word, bits [7:0] represent the number of data bytes in the payload, and bits [15:8], [23:16], and [31:24] respectively contain the effective payload bytes. The first 3 bytes of the payload, where bits [15:8] contain the first byte of the payload, bits [23:16] contain the second byte, and bits [31:24] contain the third byte. Includes the third byte. Subsequent write operations to the R32_SWPMI_TDR register will only contain the subsequent payload data bytes (per...). Each write operation can handle a maximum of 4 bytes.

Note: The low-significant byte of the first bit word written to the R32_SWPMI_TDR register is used for the data bytes in the payload. The number is encoded. This value should be between 1 and values must be between 32 and 32. Values outside this range in the least significant byte will be disregarded and will not be considered. 30 to trigger data transmission.

Writing to the R32_SWPMI_TDR register will trigger the following series of operations:

1) When the SWP bus is in a "suspended" state, the system will send a transition sequence along with 8 idle bits (master device resumes). Conversely, This operation will not be triggered if the SWP bus is in an "activated" state;

2) Send frame start;

3) Send the payload according to the contents of the R32_SWPMI_TRD register. If the number of bytes in the payload exceeds 3, as long as... The TXBEF flag in the R32_SWPMI_ISR register is not set to 1. Whenever the TXE flag in the R32_SWPMI_ISR register is set to 1, R32_SWPMI_TDR needs to be refilled by the software;

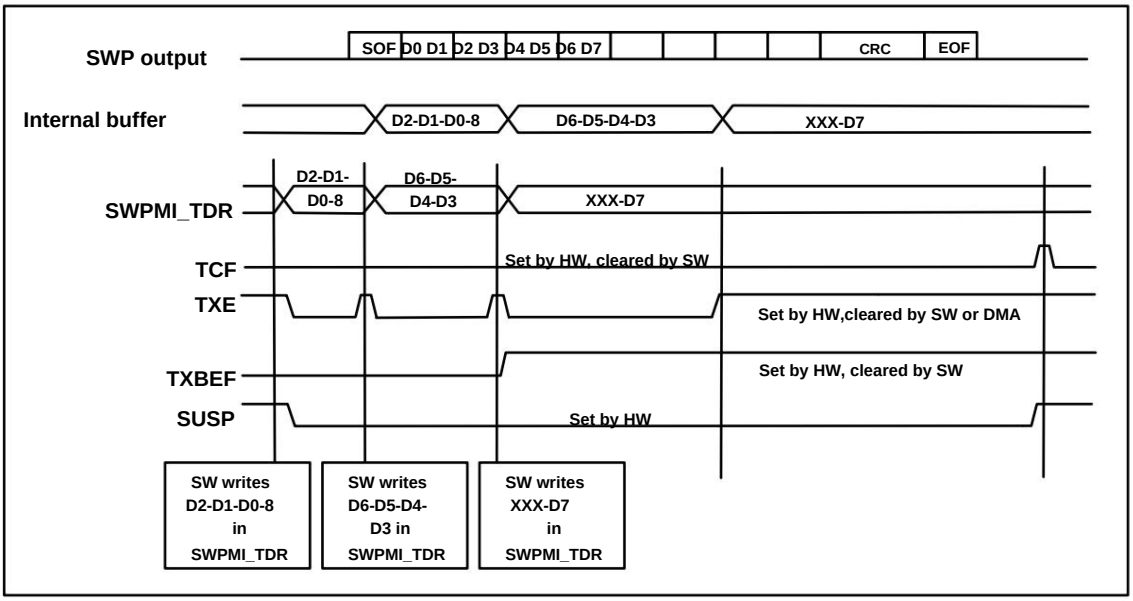
4) Send a 16-bit CRC (automatically calculated by the SWPMI kernel);

5) End of Frame (EOF).

When the software performs a write operation on the R32_SWPMI_TDR register, it will automatically clear the TXE flag.

After a complete frame is successfully transmitted, if no other frames have been requested (i.e., no further requests have been made to transmit R32_SWPMI_TDR after the TXBEF flag was set to 1), (If a write operation is performed), then after EOF is sent, and after 7 idle bits, the TCF and SUSP in the R32_SWFMI_ISR register... The flag will be set to 1. Additionally, setting the TCIE bit to 1 in the R32_SWPMI_IER register will trigger an interrupt, as shown in Figure 38-4.

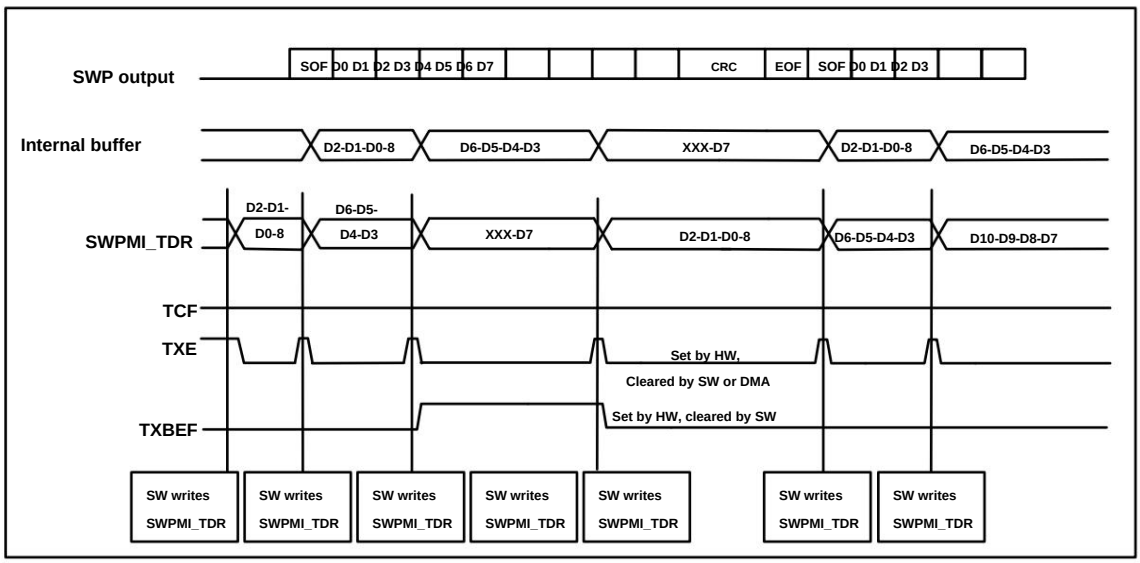
Figure 38-4 SWPMI Sending in Software Bufferless Mode



If a frame transmission request is initiated again before the EOF transmission ends, the TCF flag will not be set to 1, and this frame will be treated as part of the previous frame.

Connect (there is only one empty space between the two, please refer to Figure 38-5).

Figure 38-5 SWPMI transmission in software-buffer-free mode, continuous frames.



In **single-software buffer mode,**

complete SWP frames can be sent via DMA without CPU intervention. DMA will refill the 32 bits.

The R32_SWPMI_TDR register allows software to use the SWPMI_TXBEF flag to poll whether frame transmission has ended. Select single software buffer mode:

Set the TXDMA bit to 1 and clear the TXMODE bit to 0 in the R32_SWPMI_CR register.

The DMA channel or data stream must be configured in the following mode (see the DMA section):

The memory size is set to 32 bits;

Peripheral size is set to 32 bits;

Enable memory increment mode;

Loop mode is prohibited ;

• Disallow peripheral incremental mode;

• Memory-to-memory mode is disabled; •
The number of words to be transferred must be set according to
the SWP frame length; • The destination address is
the R32_SWPMI_TDR register; • The data transfer
direction is set to read from memory; • The source
address is the SWP frame buffer in
RAM. The user then performs the following operations:
1) Set the TXDMA bit in the R32_SWPMI_CR register to 1; 2)
Set the TXBEIE bit in the R32_SWPMI_IER register to 1; 3) Fill the buffer in RAM (the number of data bytes in the payload is

represented by the least significant byte of the first word); 4) Trigger DMA transfer and frame
transmission by enabling the data stream or channel in the DMA module. SWPMI will initiate a DMA request when the TXE flag in R32_SWPMI_ISR is set
When the device performs a write operation, the TXE flag will be automatically cleared to zero.

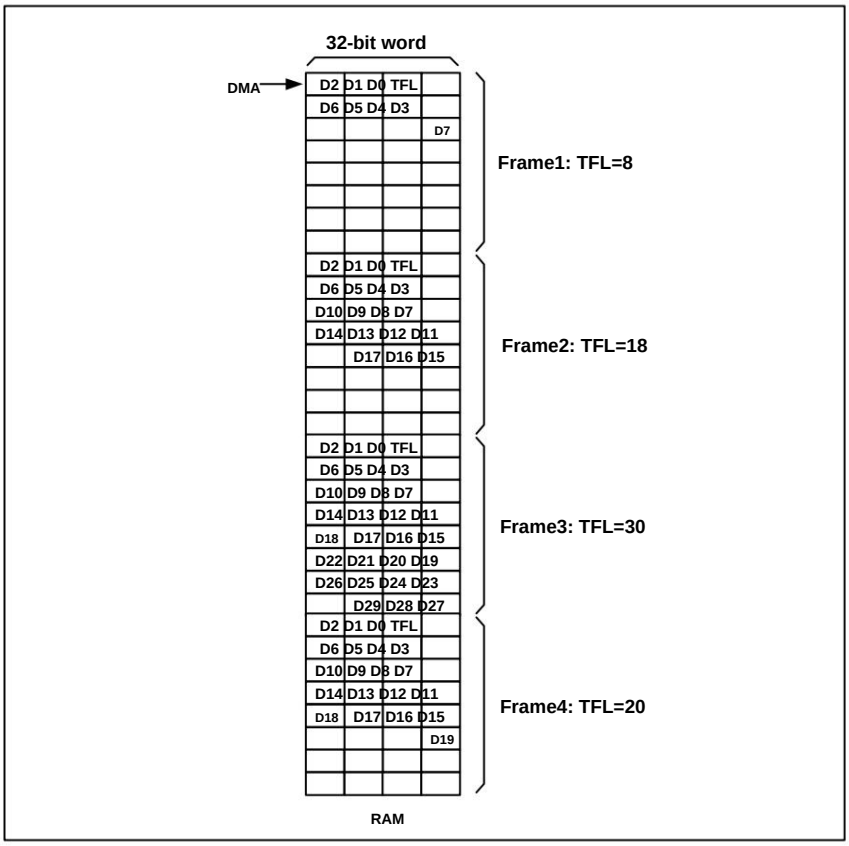
In the SWPMI interrupt routine, the user needs to check the TXBEF bit of the R32_SWPMI_ISR register. If this bit is set to 1, and there are other bits...
If another frame needs to be sent, the user must perform the
following operations: 1) Disable the data stream or
channel in the DMA module; 2) Update the buffer in the RAM memory with the content of the
next frame to be sent; 3) Set the total number of words to be
transferred in the DMA module; 4) Trigger the next frame transmission by enabling the data stream
or channel in the DMA module; 5) Clear the TXBEF flag by setting the CTXBEF bit of the R32_SWPMI_ICR register to 1.

Multi-software buffer mode:

Multi-software buffer mode supports configuring multiple frame buffers in RAM memory to ensure continuous data transmission and maintain extremely low latency.
CPU load is reduced, and more latency is provided to the software to perform buffer updates (using DMA). The software can monitor the DMA counter in
real time and update the SWP frame in RAM memory accordingly.

For optimal performance, the multi-software buffer mode should be used in conjunction with DMA in circular
mode. In the SWP frame payload, regardless of the number of bytes, each transmit buffer in the RMA memory should have a fixed length, i.e., eight
32-bit words. These transmit buffers need to be filled by software, maintaining an 8-byte offset between two consecutive buffers. The first data byte of the
buffer indicates the number of bytes in the frame payload, as shown in the figure below.

Figure 38-6 SWPMI Multi-Software Buffer Mode Transmission



The multiple software buffer mode is selected by setting the TXDMA and TXMODE bits in the R32_SWPMI_CR register to 1. To use four transmit buffers, the user must configure DMA as follows: The DMA

channel or data stream must be configured in the following modes (see the DMA section):

ÿ Memory size is set to 32 bits; ÿ

Peripheral size is set to 32 bits; ÿ The

number of words to be transferred must be set to 32 (8 words per buffer); ÿ Data

transfer direction is set to read from memory; ÿ Memory-

to-memory mode is disabled; ÿ Peripheral

increment mode is disabled; ÿ

Memory increment mode is enabled; ÿ

Cyclic mode is enabled; ÿ

The source address is buffer 1 in RAM; ÿ The

destination address is the R32_SWPMI_TDR register. The

user then performs the following

operations: 1) Set the TXDMA bit in the R32_SWPMI_CR register

to 1; 2) Set the TXBEIE bit in the R32_SWPMI_IER register to 1; 3)

Fill buffers 1, 2, 3, and 4 in RAM (the number of data bytes in the payload is represented by the least significant byte of the first word); 4) Trigger DMA transfer and frame transmission by enabling the data

stream or channel in the DMA module. In the SWPMI interrupt routine, the user must check the TXBEF

bit in the R32_SWPMI_ISR register. If this bit is 1, the user should set the CTXBEF bit in the R32_SWPMI_ICR register to 1 to clear the TXBEF flag and update buffer 1 in RAM. The next time an SWPMI interrupt occurs, the user will update buffer 2, and so on.

The software can also read the DMA counter (the amount of data to be transferred) in the DMA register to retrieve frames that have been transferred and sent from RAM. For example, if the software uses four send buffers and the DMA counter value is 17, it means that two buffers in the RAM area are ready to be updated. This feature effectively solves the situation where the software needs to send multiple frames before processing an SWPMI interrupt. If this happens, the software must update multiple buffers.

When no frames need to be sent, the user must disable the DMA module's circular mode. Transmission will stop when buffer 4 finishes transmitting. If transmission is stopped early, such as when buffer 2 ends, the user must set the least significant byte of the first word in buffers 3 and

4 to 0. When the number of payload bytes read in the least significant byte of the first word is 0, the TXDMA bit in the R32_SWPMI_CR register will be cleared by hardware.

38.3.7 Reception Process

Before transmitting frames, the user must activate SWP. Please refer to 38.3.1. After setting the SWPACT bit in the R32_SWPMI_CR register to 1, the "Recover from Device" state sets the SRF flag in the R32_SWPMI_ISR register to 1 and automatically enables SWPMI for frame reception. If the SWP bus is in the "Active" state, and frame transmission is in progress, the SWPMI kernel does not need to be in the "Recover from

Device" state and can immediately begin receiving. There are several software implementation methods for frame reception, including no software buffer mode, single software buffer mode, and multiple software buffer mode. The use of the software buffer requires a DMA channel to transfer data from the receive data register in the SWPMI peripheral to the software buffer in RAM memory.

In software-buffered mode,

DMA is not required. SWP frame transmission is handled by polling the status flags within the main loop or the SWPMI interrupt routine. SWPMI includes a 32-bit receive data register (R32_SWPMI_RDR), allowing up to four bytes to be received before reading this register. Software-buffered

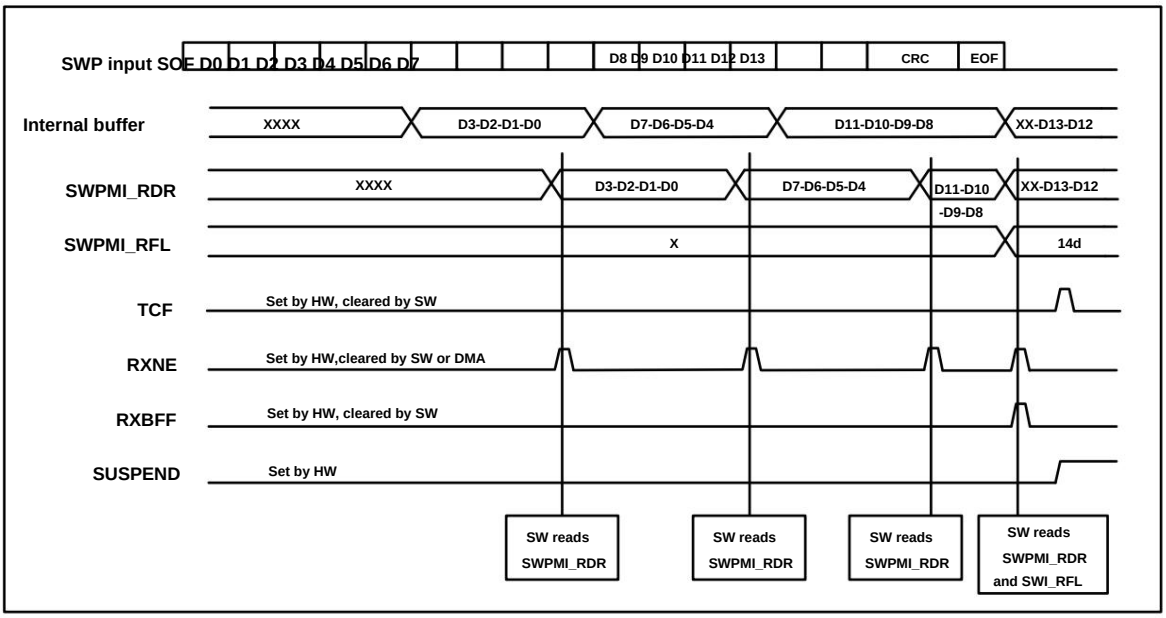
mode can be selected by resetting the RXDMA bit in the R32_SWPMI_CR register. After receiving the

start of a frame (SOF), the subsequent payload bytes are stored in the R32_SWPMI_RDR register. When R32_SWPMI_RDR is full, the RXNE flag in R32_SWPMI_ISR is set to 1. Additionally, an interrupt is generated if the RIE bit in the R32_SWPMI_IER register is set to 1. Users can read the R32_SWPMI_RDR register to obtain data; the RXNE flag is automatically cleared when software reads the R32_SWPMI_RDR register.

Once a complete frame (CRC, EOF) is received, the RXNE and RXBFF flags in the R32_SWPMI_ISR register are set to 1. The user must read the last one or a few bytes of the payload in the R32_SWPMI_RDR register and clear the RXBFF flag by setting the CRXBFF flag to 1 in the R32_SWPMI_ICR register. The number of data bytes in the payload can be obtained from the R32_SWPMI_RFL register. The RXNE flag will be automatically reset again when the software reads the R32_SWPMI_RDR register, as shown in the figure below.

When RXNE is cleared, reading the R32_SWPMI_RDR register will return 0.

Figure 38-7 SWPMI No Software Buffer Mode Receiver



In **single-software buffer mode,**

complete SWP frames can be sent via DMA without CPU intervention. DMA transfers received data from 32...

The R32_SWPMI_RDR register is transferred to RAM memory, and the software can use the SWPMI_RBFF flag to poll whether frame reception has ended.

Software buffer mode: Set the RXDMA bit to 1 and the RXMODE bit to 0 in the R32_SWPMI_CR register.

The DMA channel or data stream must be configured in the following mode (see the DMA section):

The memory size is set to 32 bits;

Peripheral size is set to 32 bits;

Enable memory increment mode;

Loop mode is prohibited ;

• Disallow incremental peripheral mode;

ÿDisable memory-to-memory mode;

The number of words to be transmitted is set to 8;

The target address is the **SWP frame buffer in RAM;**

ÿ Set the data transfer direction to read from memory;

The **source address is the R32_SWPMI_RDR register.**

The user then performed the following actions:

1) Set the RXDMA bit in the R32_SWPMI_CR register to 1;

2) Set the RXBEIE bit in the R32_SWPMI_IER register to 1;

3) By enabling the data stream or channel in the DMA module.

When the RXNE flag in R32_SWPMI_ISR is set to 1, SWPMI will initiate a DMA request. The DMA will then register with R32_SWPMI_RDR.

When a register is written, the RXNE flag will be automatically cleared.

In the SWPMI interrupt routine, the user needs to check the RXBEF bit of the R32_SWPMI_ISR register. If this bit is set to 1, and there are other bits...

If another frame needs to be sent, the user must perform the following operations:

1) Disable data flow or channels in the DMA module;

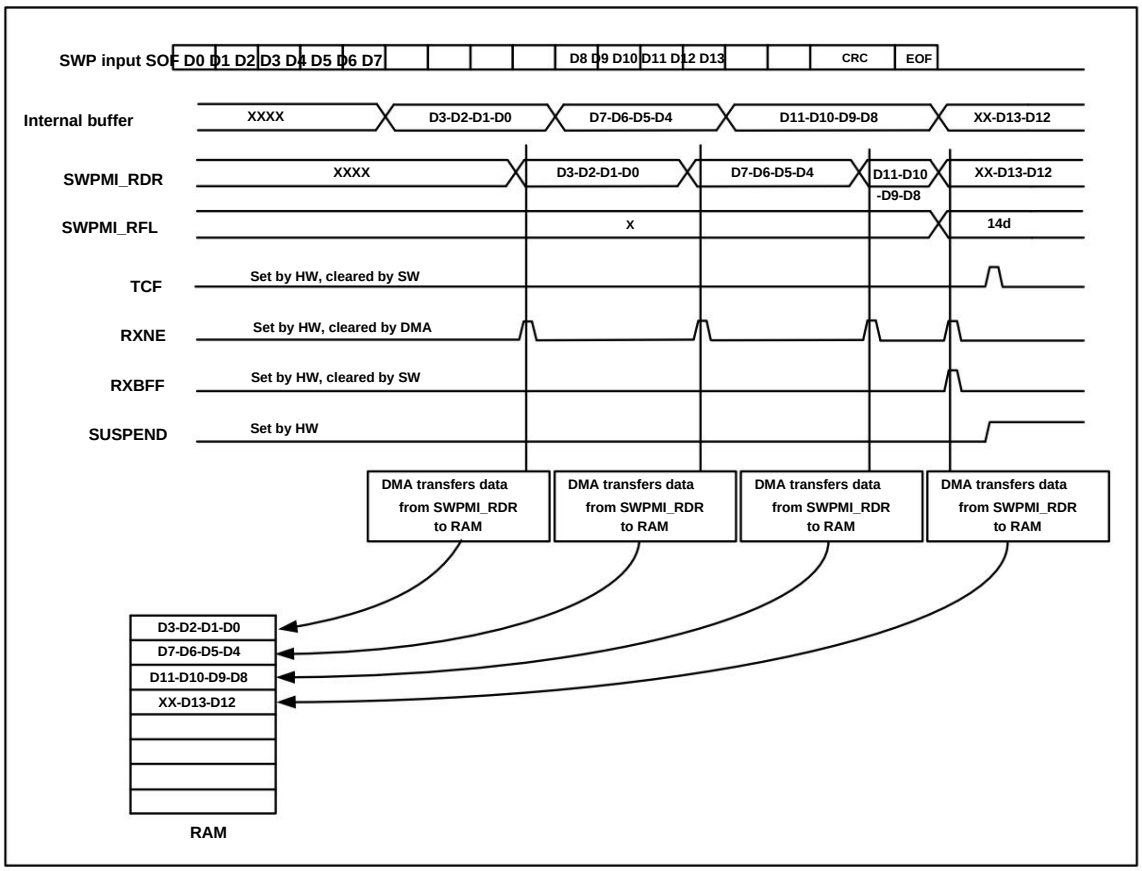
2) Read the number of bytes of the received frame payload from the R32_SWPMI_RFL register;

3) Read the frame payload from the RAM buffer;

4) Enable the data stream or channel in the DMA module;

5) Clear the RXBEF flag by setting the CTXBEF bit to 1 in the R32_SWPMI_ICR register. See Figure 38-8 for details.

Figure 38-8 SWPMI Single Software Buffer Mode Receiver



Multi-software buffer mode:

Multi-software buffer mode supports configuring multiple frame buffers in RAM memory to ensure continuous data transmission and maintain extremely low latency. CPU load. The frame payload and frame status flags are stored in RAM. Software can check the DMA counters and status flags at any time. The system is used to process SWP frames received in RAM memory.

For best results, the multi-software buffer mode should be used in conjunction with DMA in circular mode. Select the multi-software buffer mode: Set the RXDMA and RXMODE bits in the R32_SWPMI_CR register to 1.

To enable the use of n receive buffers in RAM, the DMA channel or data stream must be configured in the following mode (see DMA section). point):

The memory size is set to 32 bits;

Peripheral size is set to 32 bits;

The number of words to be transmitted **must be set to 8*n (8 words per buffer);**

ÿ Set the data transmission direction to read from the peripheral device;

ÿDisable memory-to-memory mode;

• Disallow peripheral incremental mode;

Enable memory increment mode;

Enable cycle mode;

The **source address is the R32_SWPMI_TDR register;**

The target address is buffer 1 in RAM.

The user then performed the following actions:

1) Set the RXDMA bit in the R32_SWPMI_CR register to 1;

2) Set the RXBEIE bit in the R32_SWPMI_IER register to 1;

3) Enable the data stream or channel in the DMA module.

In the SWPMI interrupt routine, the user must check the RXBEF bit of the R32_SWPMI_ISR register. If this bit is set to 1, the user should...

The R32_SWPMI_ICR register CTXBEF is set to 1 to clear the RXBEF flag, and the data can be read from the first buffer (in DMA2_CMAR1).

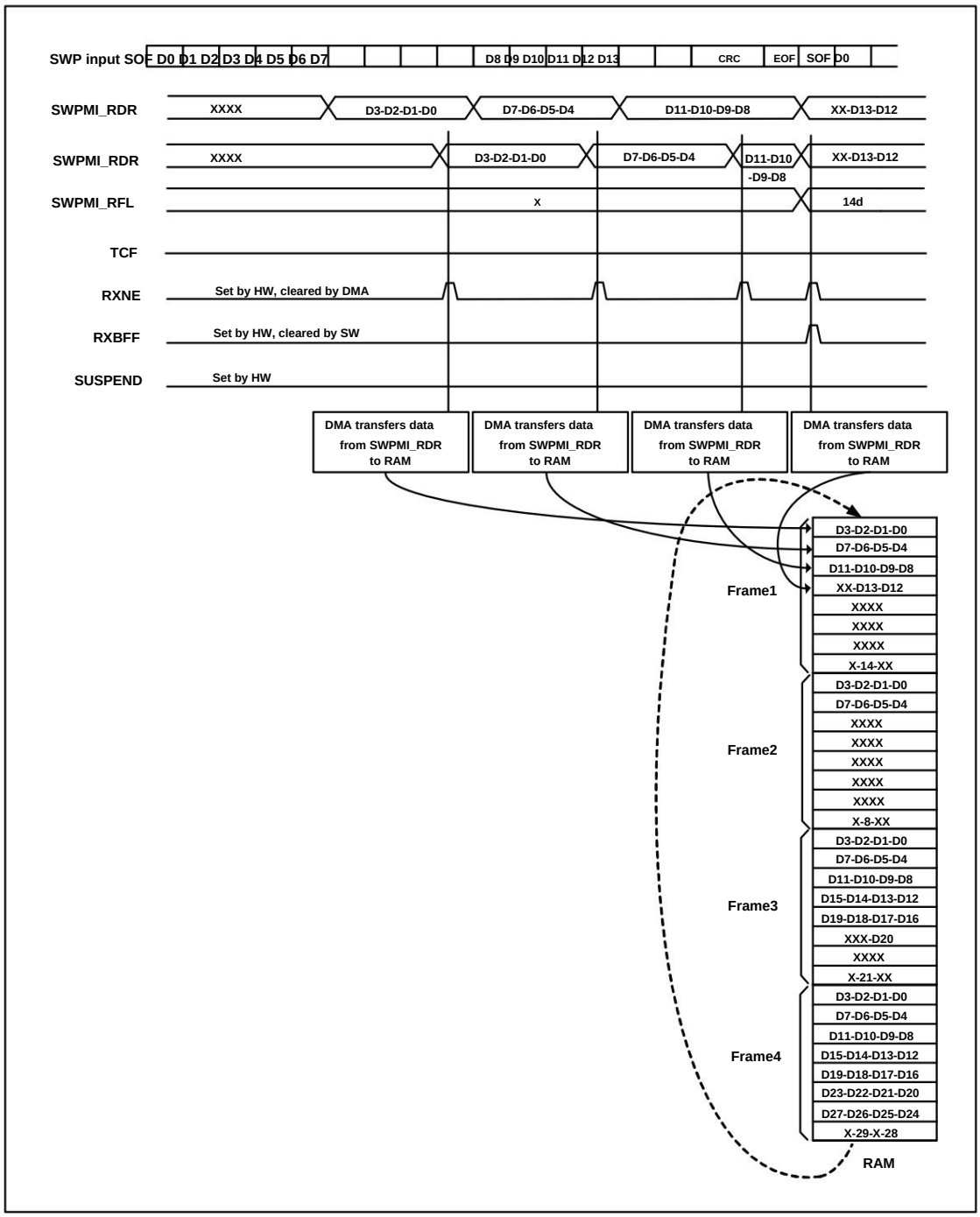
The first frame payload received at the RAM address set in the memory. The number of data bytes in the payload can be found in the 8th word from the end.

[23:16] is obtained from.

When the next SWPMI interrupt routine occurs, the user will read from the second buffer (address set in DMA2_CMAR1 + 8).

The second frame received in the process is repeated in a loop. See Figure 38-9.

Figure 38-9 SWPMI Multi-Software Buffer Mode Receiver



If the application software cannot guarantee that it will process the SMPMI interrupt before receiving the next frame, then the state of each buffer will be changed in the 8th buffer.

Presented in the most significant byte of the conflict zone word:

• Receive Overflow Flag: The RXOVRF flag in the R32_SWPMI_ISR register, represented by bit 25 of the 8th word. (For more information...)

For an explanation of overflow errors, please refer to

Section 38.3.8. ÿ Receive buffer full flag: The RXBFF flag in the R32_SWPMI_ISR register, represented by bit 26 of the 8th word. ÿ CRC error flag: The RXBERF flag in the R32_SWPMI_ISR register, represented by bit 24 of the 8th word. (For more information on CRC...)

For an incorrect description, please refer to Section 38.3.8.

When a CRC error occurs, both the RXBFF and RXBERF flags are set to 1, meaning bits 24 and 26 will both be set to 1. When an overflow occurs, the overflow flag is set to 1, i.e., bit 25 will be set to 1. The receive buffer full flag is only set to 1 if an overflow occurs when the last byte is received; subsequently, bits 25 and 26 will be set to 1 for the current and next frame reception.

The software can read the DMA counter in the DMA register to retrieve frames transferred via DMA and stored in RAM. For example, when
When the software uses four receive buffers and the DMA counter value is 17, it indicates that two buffers in the RAM area are ready for the software to read. In multi-software buffer receive mode, if the software is reading bits 24, 25, and 26 of the 8th word, then it is not necessary to clear the RXBERF, RXOVRF, and RXBFF flags after each received frame.

38.3.8 Error Management

Underflow During Payload Transmission If

a transmit underflow occurs during the transmission of the frame payload, the TXUNRF flag in the R32_SWPMI_ISR register will be triggered as a warning. If the TXBUNREIE bit in the R32_SWPMI_IER register is set to 1, an interrupt will be generated. When a transmit underflow occurs, SWPMI will stop transmitting the payload and transmit a corrupted CRC (the first bit of the first CRC byte transmitted is inverted), followed by the EOF flag. If DMA is used, the TXDMA bit in the R32_SWPMI_CR register will be automatically cleared.

When TXUNRF is set to 1, any write operations to the R32_SWPMI_TDR register will be ignored. The user must then...
Set the CTXUNRF bit in the R32_SWPMI_ICR register to 1 to clear the TXUNRF flag.

Overflow during payload reception : If a

receive overflow occurs during the receive frame payload process, the RXOVRF flag in the R32_SWPMI_ISR register will be triggered as a warning. When a receive overflow occurs, incoming data will not be used by SWPMI to update R32_SWPMI_RDR, and therefore the incoming data will be lost. Data reception continues until

EOF. Once the overflow condition disappears, the RXBFF flag will be set to 1. When the RXBFF flag is set to 1, the user can check the RXOVRF flag. To clear the RXOVRF flag, the user needs to set the CRXOVRF bit in the R32_SWPMI_ICR register to 1. If the user needs to detect overflow immediately, the RXBOVREIE bit in the R32_SWPMI_IER register can be set to 1, thus ensuring that an interrupt is triggered immediately upon the occurrence of an overflow. When both the RXOVRF and RXNE flags are

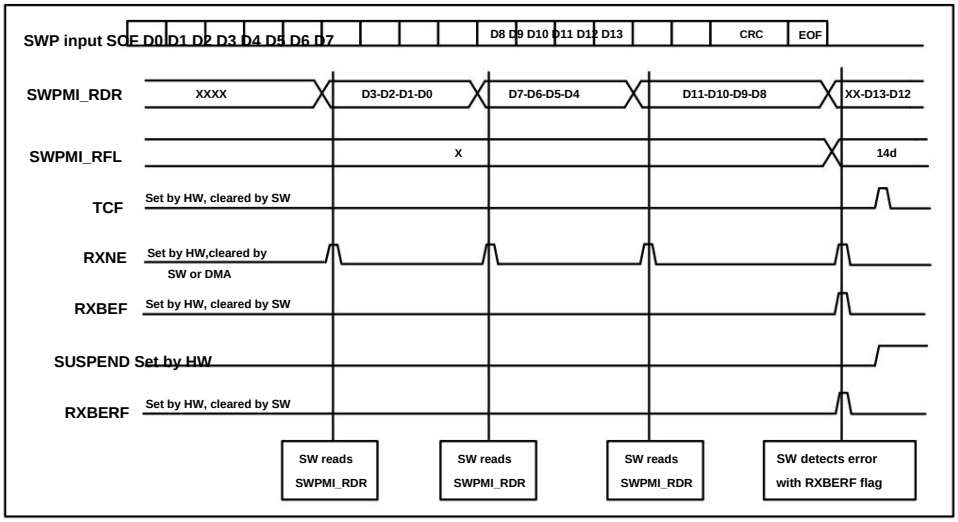
set to 1, two R32_SWPMI_RDR read operations will be performed after an overflow event. This indicates that at least one received byte has been lost, and the bytes loaded in R32_SWPMI_RDR include the byte just received before the overflow occurred. In multi-software buffer mode, if the last word of a received frame sets the

RXOVRF flag to 1, then the current frame and the next frame...
Set the overflow bit (bit 25 of the 8th word) to 1.

If a CRC error occurs during payload reception ,

after receiving two CRC bytes, if a CRC error occurs, the RXBERF flag in the R32_SWPMI_ISR register will be set to 1 after EOF reception is complete. At this time, if the RXBEIE bit in the R32_SWPMI_IER register is set to 1, an interrupt will be triggered (see Figure 38-10). To clear the RXBERF flag, the user must set the CRXBERF bit in the R32_SWPMI_ICR register to 1.

Figure 38-10 SWPMI Single Buffer Mode Receive (with CRC Error)



If **padding bits are lost or corrupted during payload**

reception, the RXBERF value in R32_SWPMI_ISR will be updated after EOF reception.

The RXBFF flag will be set to 1.

Upon **receiving a**

corrupted EOF , SWPMI will begin accumulating received bytes after receiving a SOF (ignoring any possible SOFs).

Upon receiving the EOF, SWPMI will be ready to begin a new round of frame reception and wait for the next SOF. If the EOF is corrupted, SWPMI_ISR will be activated.

The RXBERF and RXBFF flags in the register will be set to 1 after the next EOF (End of Receive) is completed.

Note: Even if a corrupted payload is received, the system will continue to receive payloads. Therefore, if the number of received bytes exceeds 30, the payload will be invalidated.

The number of bytes in the payload at EOF can be 31. The number of bytes in the payload is read from the R32_SWPMI_RFL register, or depending on the operating mode.

Same, in RAM, read from bits [23:16] of the word in the memory buffer.

38.3.9 Loopback Mode

Loopback mode can be used for testing. The user must set the LPBK bit in the R32_SWPMI_CR register to 1 to enable loopback mode.

When loopback mode is enabled, the SWP_TX and SWP_RX signals will be connected. Therefore, all frames sent via SWPMI will be retrieved.

38.3.10 SWPMI Interrupts All

SWPMI interrupts are connected to the PFIC. To enable SWPMI interrupts, follow these steps:

1) Configure and enable the SWPMI interrupt channel in the PFIC;

2) Configure SWPMI to generate an SWPMI interrupt (refer to the R32_SWPMI_IER register).

Table 38-2 Interrupt Requests

Interrupt	Interrupt event flag	interrupt enable control bit
event: Receive buffer	RXBFF	RXBFIE
full, transmit buffer	TXBEF	TXBEIE
empty, receive buffer error (CRC error)	RXBERF	RXBEIE
Receive buffer	RXOVRF	RXBOVEREIE
overflow, transmit	TXUNRF	TXBUNREIE
buffer underflow, receive	RXNE	RIE
data register not empty, transmit data register full.	TXE	TIE

Transmission	TCF	TCIE
completion flag, device	SRF	SRIE
recovery flag, transceiver ready flag.	RDYF	RDYIE

38.4 Register Description

Table 38-3 List of SWPMI Related Registers

name	Access Address Description: 0x40008400	Reset value
R32_SWPMI_CR	SWPMI Configuration/Control Register; 0x40008404	0x00000000
R32_SWPMI_BRR	SWPMI Bitrate Register; 0x4000840C SWPMI	0x00000001
R32_SWPMI_ISR	Interrupt and Status Register; 0x40008410 SWPMI	0x000002C2
R32_SWPMI_ICR	Interrupt Flag Clear Register; 0x40008414 SWPMI Interrupt	0x00000000
R32_SMPMI_IER	Enable Register; 0x40008418 SWPMI Receive	0x00000000
R32_SWPMI_RFL	Frame Length Register; 0x4000841C SWPMI Transmit	0x00000000
R32_SWPMI_TDR	Data Register; 0x40008420 SWPMI Receive Data	0x00000000
R32_SWPMI_RDR	Register; 0x40008424 SWPMI Options Register	0x00000000
R32_SWPMI_OR		0x00000000

38.4.1 SWPMI Configuration/Control Register (R32_SWPMI_CR) Offset

Address: 0x00

31	30	29 28 27	26 25 24 23 22 21	2019	18	17 16
Reserved						

15	14	13 12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			SWPTEN	DEACT	Reserved				SWPACT	LPBK	TXMODE	RXMODE	TXDMA	RXDMA

Bit	name	access	describe	Reset value
[31:12] Reserved		RO reserved.		0
11	SWPTEN	RW	Single-wire protocol master transceiver enabled: 1: The SWPMI_IO transceiver is controlled by SWPMI; 0: The SPWMI_IO pin is controlled by the GPIO controller.	0
10 DEACT		The RW single-line protocol main interface is disabled, used to request SWP "disabled" status.	State. Except for the possibility of an incoming "recover from device" state causing SWP to remain in the "activated" state, setting this bit to 1 and clearing it to zero .. SWPACT has the same effect.	0
[9:6] Reserved		RO reserved.		0
5	SWPACT	RW	Single-line protocol main interface activated: 1: SWPMI_IO is released, and the SWP bus switches to "Pending". state; 0: SWPMI_IO is pulled down to ground, and the SWP bus is switched to "disabled". Use "state". To set SWPACT to 1, DEACT must be cleared first.	0
4	LPBK	RW loopback mode enabled:		0

			1: Enable feedback mode; 0: Disable loopback mode.	
3	TXMODE	RW	Send buffer mode: 1: Configure SWPMI to multi-software buffer mode for sending; 0: Configure SWPMI to single software buffer mode for sending. Note: SWPACT Position 1 At that time, this bit cannot be written.	0
2	RXMODE	RW	Receive buffer mode: 1: Configure SWPMI to multi-software buffer mode for receiving; 0: Configure SWPMI for single software buffer mode for receiving. Note: This bit cannot be written when the SWPACT position is selected. 1	0
1	TXDMA	RW	Send DMA Enable: 1: Enable DMA for transmission; 0: DMA is disabled for sending. Note: If the payload size of the sent frame is specified as 0x00 (least significant byte), TXDMA will be automatically zeroed. In the event of an overflow event... TDR (The first word in the frame) When the TXUNRF flag in the R32_SWP_ISRregister is set, also... TXDMA It will be automatically reset to zero.	0
0	RXDMA	RW	Receive DMA Enable: 1: Regarding DMA enable for receiving; 0: DMA is disabled for receiving.	0

Table 38-4 Buffer Mode Selection for Transmit/Receive

Buffer mode	No software buffer	Single software buffer	Multiple software buffers
RXMODE/TXMODE	X	0	1
RXDMA/TXDMA	0	1	1

38.4.2 SWPMI Bit Rate Register (R32_SWPMI_BRR) Offset Address: 0x04

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

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

Bit	name	access	describe	Reset value
[31:8] Reserved		RO is reserved.		0
[7:0]	BR[7:0]	RW	The bit rate prescaler must be configured according to the following formula, taking into account... In the case of programming FHCLK in RCC (Reset and Clock Control) , This bit field is used to set the SWP bus bit rate: FSWP = FHCLK / ((BR[7:0]+1)x4) Note: The programming bit rate must be within the following range: 100 kbit/s to 2 Mbit/s. At SWPACT register in R32_SWPMI_CR... Position1	0x01

			Write to BR[7:0].	

38.4.3 SWPMI Interrupts and Status Register (R32_SWPMI_ISR)

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				RDYF	DEACTF	SUSP	SRF	TCF	TXE	RXNE			TXUNRF	RXOVRF	RXBERF	TXBEF	RXBFF

Bit	name	access	describe	Reset value
[31:12] Reserved		RO is reserved.		0
11	RDYF	RO	Transceiver ready flag: This bit is immediately reset by the hard drive when the transceiver is ready. Set the component to 1. This is achieved by setting SWPTEN in the R32_SWPMI_CR register. After enabling the SWPMI_IO transceiver at position 1, the software must wait... Set this flag to 1, then set SWPACT to 1 to activate SWP. Wire. 1: The transceiver is ready; 0: Transceiver not ready.	0
10 DEACTF		RO	Disabled flag: 1: The SWP bus is in a "disabled" state; 0: The SWP bus is in an "activated" or "suspended" state. state.	0
9	SUSP	RO	Hang up the sign: 1: The SWP bus is in a "suspended" or "disabled" state; 0: The SWP bus is in an "activated" state.	1
8	SRF	RO	Device recovery flag: This bit is set to 1 by hardware to indicate detection. Return to "Recover from Device" state. This bit is cleared by software using the following method: Write 1 to the CSRF bit in the R32_SWPMI_ICR register. 1: Detecting "from" while the SWP bus is in a "suspended" state. Device restored to "state"; 0: No "Recover from Device" status detected.	0
7	TCF	RO	Transmission Complete Flag: This flag is immediately generated by the hardware when both sending and receiving are complete and the SWP switches to the "Pending" state. Set to 1. This bit is cleared by software by setting it to R32_SWPMI_ICR. Write 1 to the CTCF bit in the register. 1: Sending and receiving have both been completed and SWP has been switched to "Hanged". "Start" state; 0: Sending or receiving was not completed.	1
6	TXE	RO	The transmit data register is empty: 1: Data has been sent and written to the transmit data register R32_SWPMI_TDR The data in R32_SWPMI_TDR can be written again; 0: Not yet sent to the transmit data register	1

			Data in R32_SWPMI_TDR.	
5	RXNE	RO	Receive data register is not empty: 1: The data received in the R32_SWPMI_RDR register is ready. Ready to read; 0: No data was received in the R32_SWPMI_RDR register.	0
4	TXUNRF	RO	Send underflow error flag: This flag is set to 1 by hardware to indicate An underflow occurs during payload transmission, i.e., the software or DMA fails to... Write operations to R32_SWPMI_TDR promptly. This flag is controlled by software. The method to clear the file is to add a value to the R32_SWPMI_ICR register. Write 1 to the CTXUNRF bit. 1: An underflow error was detected during transmission; 0: No underflow error during transmission.	0
3	RXOVRF	RO	Receive overflow error flag: This flag is set to 1 by hardware to indicate... An overflow occurs during payload reception, i.e., the software or DMA fails to... Perform timely read operations on R32_SWPMI_RDR. This flag is controlled by software. The method to clear the file is to add a value to the R32_SWPMI_ICR register. CRXOVRF bit write 1. 1: An overflow error was detected during reception; 0: No overflow error during reception.	0
2	RXBERF	RO	Receive CRC error flag: This flag is set to 1 by hardware to indicate... A CRC error exists in the received frame. It is the same as the RXBFF flag. Set to 1. This bit is cleared by software by setting it to R32_SWPMI_ICR. Write 1 to the CRXBERF bit in the register. 1: A CRC error was detected during reception; 0: No CRC error during reception.	0
1	TXBEF	RO	Send buffer empty flag: This flag is set to 1 by hardware to indicate that the send buffer is empty. This indicates that no further R32_SWPMI_TDR update is required to complete the current process. Previous frame sent. This bit is cleared by software by sending... Write 1 to the CTXBEF bit in the R32_SWPMI_ICR register. 1: The frame transmission buffer has been cleared; 0: The frame send buffer has not been cleared.	1
0	RXBFF	RO	Receive buffer full flag: When R32_SWPMI_RDR contains This flag is set to 1 by hardware when the last word of the received frame is received. The flag is cleared by software by setting a value in the R32_SWPMI_ICR register. Write 1 to the CRXBFF bit in the memory. 1: The last word of the received frame has been transmitted to R32_SWPMI_RDR; 0: The last word of the received frame has not yet been transmitted to R32_SWPMI_RDR.	0

38.4.4 SWPMI Interrupt Flag Clear Register (R32_SWPMI_ICR) Offset Address: 0x10

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

15	14	13	12 11	10 9	8	7	6	5	4	3	2	1	0
Reserved			CRDYF	Reserved	CSRF	CTCF	Reserved	CTXU	CRXO	CRXBE	CTX	CRX			

						NRF VRF RF BEF BFF				

Bit	name	access	describe	Reset value
[31:12] Reserved		RO	Reserved.	0
11	CRDYF	RW1Z	Clear transceiver ready flag: Writing 1 to this bit will clear the , RDYF flag in the R32_SWPMI_ISR register. Writing 0 to a bit has no effect.	0
[10:9] Reserved		RO	Reserved.	0
8	CSRF	RW1Z	Clear Slave Recovery Flag: Writing 1 to this bit will clear the , SRF flag in the R32_SWPMI_ISR register. Writing 0 to a bit has no effect.	0
7	CTCF	RW1Z	Clear Transmission Complete Flag: Writing a 1 to this bit will , clear the TCF flag in the R32_SWPMI_ISR register. Writing 0 to a bit has no effect.	0
[6:5] Reserved		RO	is	0
4	CTXUNRF	RW1Z	reserved. Clearing the underflow error flag: Writing 1 to this bit... The TXUNRF flag in the R32_SWPMI_ISR register will be cleared. Writing 0 to this bit has no effect.	0
3	CRXOVRF	RW1Z	Clear the receive overflow error flag: When a 1 is written to this bit, The RXBOCREF flag in the R32_SWPMI_ISR register will be cleared. Writing 0 to this bit has no effect.	0
2	CRXBERF	RW1Z	Clear the receive CRC error flag: When a 1 is written to this bit, The RXBERF flag in the R32_SWPMI_ISR register will be cleared. Writing 0 to this bit has no effect.	0
1	CTXBEF	RW1Z	Clear the send buffer empty flag: When a 1 is written to this bit, The TXBEF flag in the R32_SWPMI_ISR register will be cleared. Writing 0 to this bit has no effect.	0
0	CRXBFF	RW1Z	Clear the receive buffer full flag: When a 1 is written to this bit, The RXBFF flag in the R32_SWPMI_ISR register will be cleared. Writing 0 to this bit has no effect.	0

38.4.5 SWPMI Interrupt Enable Register (R32_SMPMI_IER) Offset Address:

0x14

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

15	14	13 12 11	10 9	8 7	6	5	4	3	2	1	0
Reserved			RDYIE	Reserved	SRIE	TCIE	TIE	RIE			TXUNRIE	RXOVRIE	RXBERIE	TXBEIE	RXBFIE

Bit	name	access	describe	Reset value
[31:12] Reserved		RO	reserved.	0
11	RDYIE	RW	Transceiver ready interrupt enabled: 1: When the RDYF flag in the R32_SWPMI_ISR register is set to 1, it generates...	0

			SWPMI interrupted; 0: Interruption is disabled.	
[10:9] Reserved		RO is reserved.		0
8	SRIE	RW	Restore interrupt enable from device: 1: When the SRF flag in the R32_SWPMI_ISR register is set to 1, it generates... SWPMI interrupted; 0: Interruption is disabled.	0
7	TCIE	RW	Send completion interrupt enabled: 1: When the TCF flag in the R32_SWPMI_ISR register is set to 1, it generates... SWPM interrupted; 0: Interruption is disabled.	0
6	TIE	RW	Send interrupt enable: 1: When the TXE flag in the R32_SWPMI_ISR register is set to 1, it generates... SWPMI interrupted; 0: Interruption is disabled.	0
5	RIE	RW	Receive interrupt enable: 1: When the RXNE flag in the R32_SWPMI_ISR register is set to 1, it generates... SWPMI interrupted; 0: Interruption is disabled.	0
4	TXUNRIE	RW	Send underflow error interrupt enable: 1: Set the TXBUNRF flag in the R32_SWPMI_ISR register to 1. An SWPMI interrupt is generated at that time; 0: Interruption is disabled.	0
3	RXOVRIE	RW	Receive overflow error interrupt enable: 1: Set the RXBOVRF flag in the R32_SWPMI_ISR register to 1. An SWPMI interrupt is generated at that time; 0: Interruption is disabled.	0
2	RXBERIE	RW	Enable CRC error reception interrupt: 1: When the RXBERF flag in the R32_SWPMI_ISR register is set to 1 Generate an SWPMI interrupt; 0: Interruption is disabled.	0
1	TXBEIE	RW	Interrupt enabled when send buffer is empty: 1: When the TXBEF flag in the R32_SWPMI_ISR register is set to 1 Generate an SWPMI interrupt; 0: Interruption is disabled.	0
0	RXBFIE	RW	Receive buffer full interrupt enabled: 1: When the RXBFF flag in the R32_SWPMI_ISR register is set to 1 Generate an SWPMI interrupt; 0: Interruption is disabled.	0

38.4.6 SWPMI Receive Frame Length Register (R32_SWPMI_RFL)

Offset address: 0x18

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
Reserved					

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

Bit	name	access	describe	Reset value
[31:5]	Reserved	RO	Reserved.	0
[4:0]	RFL[4:0]	RO	Received frame length: the number of data bytes in the payload of the received frame. The number. The two least significant bits RFL[1:0] indicate the last time. The number of bytes related to the R32_SWPMI_RDR register read operation.	0

38.4.7 SWPMI Transmit Data Register (R32_SWPMI_TDR) Offset

Address: 0x1C

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
TD[31:16]														

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

Bit	name	access		Reset value
[31:0]	TD[31:0]	WO	Description of data transmission: Writing to this register will trigger SOF transmission. The TXE flag will be cleared after the next payload data is sent. Will.	0

38.4.8 SWPMI Receive Data Register (R32_SWPMI_RDR) Offset

Address: 0x20

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
RD[31:16]														

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

Bit	name	access		Reset value
[31:0]	RD[31:0]	RO	Description of received data: Contains the received data; reading this register will.. Clear the RXNE flag.	0

38.4.9 SWPMI Option Register (R32_SWPMI_OR) 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														Reserved	SWP_TBYP

Bit	name	access	describe	Reset value
[31:2] Reserved		RO is reserved.		0
1	Reserved	RO reserved.		0
0	SWP_TBYP	RW	Internal transceiver disabled: 1: Disable internal transceivers. SWP_RX, SWP_TX, and SWPMI_SUP Signals are available as multiplexed functions on GPIO. By selecting... This configuration allows connection to an external transceiver; 0: Enables the internal transceiver. The external interface of SWPMI is SWPMI_IO (SWP_RX, SWP_TX, and SWP_SUP signals in GPIO) (Not available above).	0

Chapter 39 Electronic Signatures (ESIG)

The electronic signature contains chip identification information: flash memory capacity and a unique identifier. It is programmed into the memory module by the manufacturer at the time of manufacture.

The system storage area of the block can be read via SWD (SDI) or application code.

39.1 Functional Description

Flash memory capacity: Indicates the size that the current chip user application can use.

Unique Identifier: A 96-bit binary code, unique to each microcontroller, accessible only to the user, not modified.

Unique identification information can be used as a security password, encryption/decryption key, product serial number, etc., for microcontrollers (products) to improve system security.

Mechanisms or identification information.

Users can access the above content in 8/16/32 bit format.

39.2 Register Description

Table 39-1 List of ESIG Related Registers

name	Access address description 0x1FFFF7E0 Flash		Reset value
R16_ESIG_FLACAP	memory 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

39.2.1 Accessing the Flash Memory Capacity Register

Bit	(R16_ESIG_FLACAP) Name	describe	Reset value
[15:0] F_SIZE[15:0]	RO	Flash memory capacity in kilobytes. Example: 0x0080 = 128KB	X

39.2.2 UID Register (R32_ESIG_UNIID1)

31 30 29 28	27 26 25 24 23 22 21	2019	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	describe	Reset value
[31:0] U_ID[31:0]		Bits 0-31 of RO UID.		X

39.2.3 UID Register (R32_ESIG_UNIID2)

31 30 29 28	27 26 25 24 23 22 21	2019	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	describe	Reset value
[31:0]	U_ID[63:32]		The 32nd to 63rd bits of RO UID.	X

39.2.4 UID Register (R32_ESIG_UNIID3)

31 30 29 28	27 26 25 24 23 22 21	2019	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	describe	Reset value
[31:0]	U_ID[95:64]		The 64th to 95th bits of RO UID.	X

Chapter 40 Flexible Storage Controller (FMC)

The module descriptions in this chapter apply only to the CH32H417 microcontroller product.

A Flexible Memory Controller (FMC) is a controller used to manage external memory, including: NAND memory controllers, NOR/PSRAM controllers.
Storage controller, synchronous DRAM (SDRAM/Mobile LPSDR SDRAM) controller.

40.1 Main Features

FMC is used to manage expansion and connection of different types of memory, including: SDRAM, NAND Flash, and synchronous/asynchronous static memory.
Its main purpose is to: meet the access time requirements of external memory devices; and convert HB data communication transactions into appropriate external device protocols.
Discussion.

All external memories share address, data, and control signals, but each has its own independent chip select signal. The FMC can only access one memory at a time.
An external device.

The main features of the FMC controller are as follows:

Devices connected to statically mapped memory :

- Static Random Access Memory (SRAM)
- PSRAM (4 storage areas)
- NOR Flash/OneNAND Flash
- NAND Flash memory with hardware ECC, capable of checking up to 8KB of data.

Connect to synchronous DRAM (SDRAM/Mobile LPSDR SDRAM) memory

External asynchronous wait control

ÿ 16*32-bit deep write FIFO ÿ Each

memory area can be configured independently

Each storage area has independent chip select control.

• Memory data bus with 8-bit, 16-bit, or 32-bit width

Programmable continuous clock output to support asynchronous and synchronous access

The SDRAM controller features a cacheable 6*32-bit deep read FIFO (6*14-bit address tag).

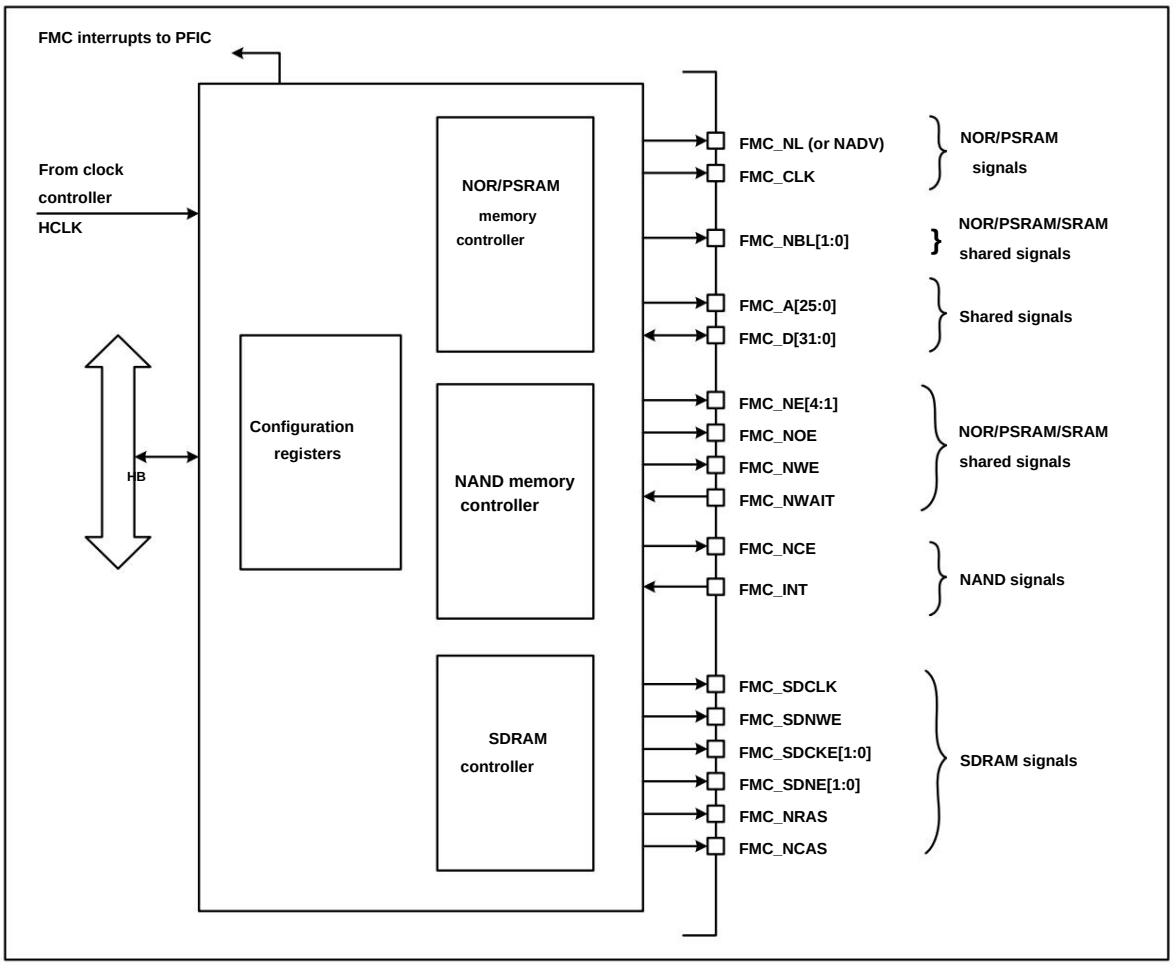
Write enable and byte channel select outputs, compatible with PSRAM, SRAM, and SDRAM devices.

Supports burst mode, enabling faster access to synchronous devices (such as NOR Flash, PSRAM, and SDRAM).

40.2 Overview

As shown in the figure below, the FMC consists of four main modules: HB interface, SDRAM controller, NAND controller, and NOR Flash/PSRAM/SRAM.
Controller.

Figure 40-1 Structural block diagram of FMC



40.3 HB Interface

The HB slave interface allows the CPU and other HB bus master devices to access external memory. HB transactions are translated into external device protocols. In particular, when the selected external memory is 16-bit or 8-bit wide, then a 32-bit wide transaction in HB will be split into multiple chains. Successive 16-bit or 8-bit accesses. The FMC chip select (FMC_NEx) does not toggle between consecutive accesses unless in extended mode access mode. In the case of formula D.

FMC will generate an HB error in the following situations:

Read or write to an disabled FMC storage area.

When the SDRAM address range is violated (accessing a reserved address range).

When writing to a write-protected SDRAM memory area (WP bit set to 1 in the R32_FMC_SDCRx register).

ÿ Read or write to the NOR Flash storage area when the FACCEN bit in the R32_FMC_BCRx register is reset.

The specific impact of this HB error depends on the HB master device that is attempting to perform read or write access. If it is a DMA controller, a DMA transfer error will be generated and the corresponding DMA channel will be automatically disabled.

The HB clock (HCLK) is the reference clock for the FMC.

40.3.1 Supported memory and transaction general

transaction rules

require HB transaction data widths of 8, 16, 32, or 256 bits; however, the data width of accessed external devices is fixed. This difference can cause data width mismatch issues. To avoid these problems, the following transaction rules must be followed:

• HB transaction data width equals memory data width: Normal operation. •

HB transaction data width greater than memory width: FMC will split the HB transaction into multiple smaller, contiguous memory accesses to match.

External data width. The FMC chip select (FMC_NEx) does not flip between consecutive accesses.

ÿ HB Transaction data width is smaller than memory width: Transmission consistency may be affected, depending on the type of external device.

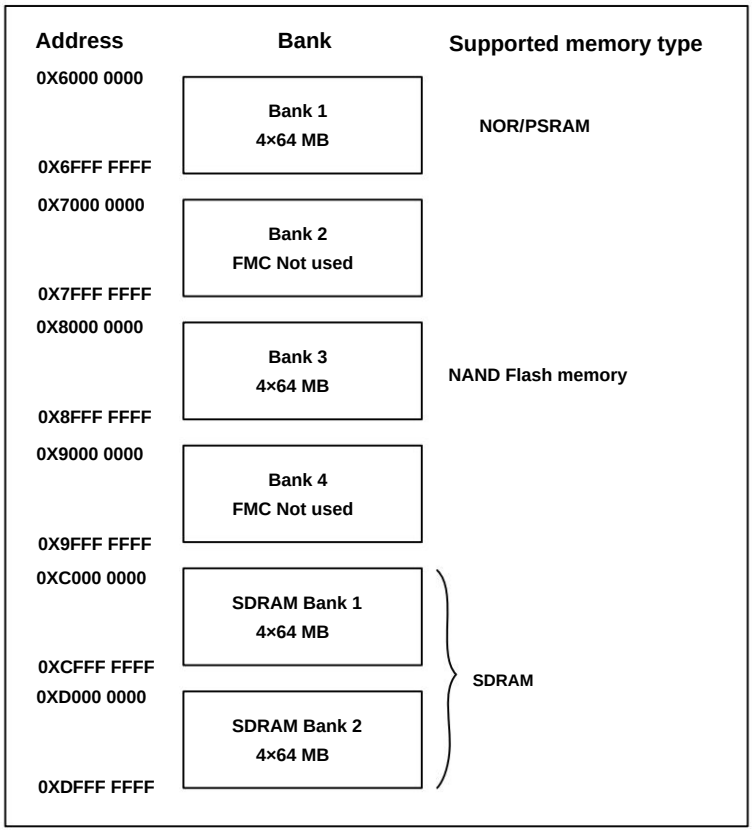
- Accessing devices with byte selection capabilities (ROM, SRAM, SDRAM, PSRAM): The FMC allows read and write transactions and accesses the correct data through its byte selection channels NBL[1:0]; addresses the byte to be written via NBL[1:0]; reads all memory bytes (NBL[1:0] remains low during the read transaction), and then discards invalid bytes. - Accessing devices without byte selection capabilities (NOR, NAND Flash): This occurs when byte access to 16-bit wide FLASH memory is required. Because these devices do not support byte-mode access (only 16-bit words can be read or written to Flash), read transactions are allowed but write transactions are not (the controller reads the entire 16-bit memory word but only uses the required byte).

For NOR Flash/PSRAM and SDRAM rollback support, since not all master devices can issue rollback transactions, synchronous memory must operate in a linear burst mode of undefined length. Configuration. If the master device generates an HB rollback transaction: ÿ Read operations will be split into two linear burst transactions. ÿ If write FIFO is enabled, write operations will be split into two linear burst transactions; if write FIFO is disabled, write operations will be split into multiple linear burst transactions.

40.4 External Device Address Mapping

From the FMC's perspective, external memory is divided into fixed-size storage regions of 256MB each, as shown in the diagram below:

Figure 40-2 FMC Storage Area



Storage area 1 can connect up to four NOR Flash or PSRAM devices. This storage area is divided into four NOR/PSRAM/SR sub-areas, each with four dedicated chip select signals: Storage Area 1 NOR/PSRAM1, Storage Area 1 NOR/PSRAM2, and Storage Area 1...

NOR/PSRAM3, Memory Area 1, NOR/PSRAM4. Memory Area 3 is used to connect NAND Flash devices. This space contains the MPU memory.

Features must be reconfigured into the device via software. Memory regions 4 and 5 are used to connect SDRAM devices (one device per memory region).

For each storage region, the type of memory to be used can be configured by the user application through configuration registers.

The FMC memory region mapping can be modified via the BMP[1:0] bits in the R32_FMC_BCR1 register. The following table shows the data used for swapping.

The configuration of NOR/PSRAM memory area with SDRAM memory area or remapped SDRAM memory area 2 allows for the storage of two different memory areas.

Access SDRAM memory regions in address mapping.

Table 40-1 FMC Storage Area Mapping Options

start address - end address	BMP[1:0]=00 (Default mapping)	BMP[1:0]=01 Swap NOR/PSRAM and SDRAM memory areas
0x60000000-0x6FFFFFFF	NOR/PSRAM storage area	SDRAM memory area 1
0x70000000-0x7FFFFFFF	FMC not used	
0x80000000-0x8FFFFFFF NAND storage area		
0x90000000-0x9FFFFFFF	FMC not used	
0xC0000000-0xCFFFFFFF SDRAM memory area 1		NOR/PSRAM storage area
0xD0000000-0xDFFFFFFF SDRAM memory area 2 SDRAM memory area 2		

40.4.1 NOR/PSRAM Address Mapping

The HADDR[27:26] bits are used to select one of the four memory regions shown in Table 40-1.

Table 40-2 NOR/PSRAM Memory Region Selection

HADDR[27:26]ÿ1ÿ	Selected storage area
00	Storage area 1 NOR/PSRAM1
01	Storage area 1 NOR/PSRAM2
10	Storage area 1 NOR/PSRAM3
11	Storage area 1 NOR/PSRAM4

Note 1: HADDR HB. Note that these are internal address lines, but they also participate in addressing external memory.

The HADDR[25:0] bits contain the address of external memory. Since HADDR is a byte address, and memory is addressed in words, therefore...

The actual address sent to memory will vary depending on the memory data width, as shown in the table below.

Table 40-3 NOR/PSRAM Memory Area Addresses

memory width (1)	Data address sent to memory	Maximum memory capacity (bits)
8-bit	HADDR[25:0]	64MB × 8 = 512Mb
16-bit	HADDR[25:1]>>1	64MB / 2 × 16 = 512Mb
32-bit	HADDR[25:2]>>2	64MB / 4 × 32 = 512Mb

Note 1: When the width of the external memory is 16 bits, the FMC will use the internal HADDR[25:1] address as the address for the external memory. Addresses FMC_A[24:0]. When the memory width is 32...

For this bit, FMC will use the internal HADDR[25:2] address for external addressing. Regardless of the width of the external memory, FMC_A[0] should be connected to the external memory address.

40.4.2 NAND Flash Address Mapping The NAND

storage area is divided into several memory areas, as shown in the table below.

Table 40-4 NAND Memory Mapping and Timing Registers

start address end address	FMC memory area memory space timing register	

0x88000000 0x8BFFFFFF		Storage Area 3 - NAND Flash	Feature area	R32_FMC_PATT (0x8C)
0x80000000 0x83FFFFFF			General area	R32_FMC_PMEM (0x88)

For NAND Flash memory, the general area and feature area storage space are divided into three parts, all located in the lower 256KB, as follows:

As shown in the table:

Table 40-5 NAND Storage Area Selection

Partial Name	HADDR[17:16]	Address range
Address Area	1X	0x020000-0x03FFFF
Command	01	0x010000-0x01FFFF
Area Data Area	00	0x000000-0x0FFFF

Data area: the first 64KB in the general/feature storage space; Command area: the second 64KB in the general/feature storage space;
Address area: the next 128KB in the general/features storage space.

Application software uses these three areas to access NAND Flash memory:

When sending commands to NAND Flash, the software must write the command value to any memory cell in the command area.

Specify the NAND Flash address to read or write; the software must write the address value to any memory cell within the address region. Because...

The address length can be 4 or 5 bytes (depending on the actual memory size). To specify the complete address, you need to specify the address...

Multiple consecutive write operations are performed on the address region.

Reading or writing data: The software will read data from any storage unit in the data area or write data to it.

Because NAND Flash memory automatically increments its address, it is not necessary to increment the address of the data region to access contiguous memory bits.
Place.

40.4.3 SDRAM Address Mapping

The HADDR[28] bit (internal HB address line 28) is used to select one of two memory regions, as shown in Table 40-6.

Table 40-6 SDRAM Memory Region Selection

HADDR[28]	Selected memory region control register		Timing register
0	SDRAM Bank1	R32_FMC_SDCR1	R32_FMC_SDTR1
1	SDRAM Bank2	R32_FMC_SDCR2	R32_FMC_SDTR2

The table below shows the SDRAM mapping with 13 rows and 11 columns.

Table 40-7 SDRAM Address Mapping

memory width (1)	internal storage area row address		Column address (2)	Maximum memory capacity (MB)
8-bit	HADDR[25:24] 16-	HADDR[23:11]	HADDR[10:0]	64MB: 4x8Kx2K
bit HADDR[26:25] 32-bit HADDR[27:26]		HADDR[24:12]	HADDR[11:1]	128MB: 4x8Kx2Kx2
		HADDR[25:13]	HADDR[12:2]	256MB: 4x8Kx2Kx4

Note 1: When linking bit memory, the FMC internally uses the HADDR[11:1] internal address lines for external addressing. (Linking bit memory)
16 HB 32

When the device is in use, the FMC internally uses the HADDR[12:2] address lines for external addressing. Regardless of the width of the external memory, FMC_A[0] must be connected to the external memory address.
A[0]

Note 2: AutoPrecharge is not supported. FMC_A[10] must be connected to external memory address A[10], but always remain "low".

The HADDR[27:0] bits will be translated to an external SDRAM address according to the SDRAM controller configuration. Data size: 8, 16, or 32 bits;
Column size: 8, 9, 10, or 11 bits; Row size: 11, 12, or 13 bits; Number of internal storage areas: 2 or 4 internal storage areas

Domain. The table below shows the SDRAM address mappings for different SDRAM controller configurations.

Table 40-8 SDRAM Address Mapping with 8-bit Data Bus Width[1][2]

row size Configuration	HADDR (HB internal address line) 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			
11 row size Configuration	Res.	Bank [1:0]	Row[10:0]	Column[7:0]
	Res.	Bank [1:0]	Row[10:0]	Column[8:0]
	Res.	Bank [1:0]	Row[10:0]	Column[9:0]
	Res.	Bank [1:0]	Row[10:0]	Column[10:0]
12-bit row size Configuration	Res.	Bank [1:0]	Row[11:0]	Column[7:0]
	Res.	Bank [1:0]	Row[11:0]	Column[8:0]
	Res.	Bank [1:0]	Row[11:0]	Column[9:0]
	Res.	Bank [1:0]	Row[11:0]	Column[10:0]
13 people row size Configuration	Res.	Bank [1:0]	Row[12:0]	Column[7:0]
	Res.	Bank [1:0]	Row[12:0]	Column[8:0]
	Res.	Bank [1:0]	Row[12:0]	Column[9:0]
	Res.	Bank [1:0]	Row[12:0]	Column[10:0]

Note 1: BANK[1:0] is the storage area address BA[1:0]. Note 2: When using an internal storage area, BA1 must always be set to 0.

Accessing the reserved (Res.) address range will generate an error.

HB

Table 40-9 SDRAM Address Mapping with 16-bit Data Bus Width[1][2]

row size Configuration	HADDR (HB internal address line) 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				
11 row size Configuration	Res.	Bank [1:0]	Row[10:0]	Column[7:0]	BM0(3)
	Res.	Bank [1:0]	Row[10:0]	Column[8:0]	BM0
	Res.	Bank [1:0]	Row[10:0]	Column[9:0]	BM0
	Res.	Bank [1:0]	Row[10:0]	Column[10:0]	BM0
12-bit row size	Res.	Bank [1:0]	Row[11:0]	Column[7:0]	BM0

Configuration	Res.	Bank [1:0]	Row[11:0]	Column[8:0]	BM0
	Res.	Bank [1:0]	Row[11:0]	Column[9:0]	BM0
	Res.	Bank [1:0]	Row[11:0]	Column[10:0]	BM0
13 people row size Configuration	Res.	Bank [1:0]	Row[12:0]	Column[7:0]	BM0
	Res.	Bank [1:0]	Row[12:0]	Column[8:0]	BM0
	Res.	Bank [1:0]	Row[12:0]	Column[9:0]	BM0
	Res.	Bank [1:0]	Row[12:0]	Column[10:0]	BM0

Note 1: BANK[1:0] is the storage area address BA[1:0]. When only one internal storage area is used, BA1 must always be set to 0.

Note 2: Accessing reserved space (Res.) will generate an error.

3. BM0-16 is a byte mask for bit access.

HB

Table 40-10 SDRAM Address Mapping with 32-bit Data Bus Width (1) (2)

row size Configuration	HADDR (HB internal address line) 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				
11 row size Configuration	Res.	Bank [1:0]	Row[10:0]	Column[7:0]	BM[1:0]ÿ3ÿ
	Res.	Bank [1:0]	Row[10:0]	Column[8:0]	BM[1:0]
	Res.	Bank [1:0]	Row[10:0]	Column[9:0]	BM[1:0]
	Res.	Bank [1:0]	Row[10:0]	Column[10:0]	BM[1:0]
12-bit row size Configuration	Res.	Bank [1:0]	Row[11:0]	Column[7:0]	BM[1:0]
	Res.	Bank [1:0]	Row[11:0]	Column[8:0]	BM[1:0]
	Res.	Bank [1:0]	Row[11:0]	Column[9:0]	BM[1:0]
	Res.	Bank [1:0]	Row[11:0]	Column[10:0]	BM[1:0]
13 people row size Configuration	Res.	Bank [1:0]	Row[12:0]	Column[7:0]	BM[1:0]
	Res.	Bank [1:0]	Row[12:0]	Column[8:0]	BM[1:0]
	Res.	Bank [1:0]	Row[12:0]	Column[9:0]	BM[1:0]
	Bank	Row[12:0]		Column[10:0]	BM[1:

	[1:0]			0]

Note 1: BANK[1:0] is the storage area address BA[1:0]. When only one internal storage area is used, BA1 must always be set to 0.

Note 2: Accessing reserved space (Res.) will generate an error.HB

Note 3: BM[1:0] is the byte mask for bit access.

40.5 NOR Flash/PSRAM Controller

FMC generates appropriate signal timings to drive the following types of memory:

Asynchronous SRAM and ROM: 8-bit, 16-bit, 32-bit;

PSRAM (Cellular RAM): Asynchronous mode, burst mode with synchronous access, multiplexed or non-multiplexed;

ÿ NOR Flash: Asynchronous mode, burst mode with synchronous access, multiplexed or non-multiplexed.

The FMC outputs a unique chip select signal NE[4:1] for each memory region. All other signals (address, data, and control) are...

Shared signals. The FMC supports various devices through programmable timing, including:

Programmable continuous clock (FMC_CLK) output

ÿ The wait period is programmable (up to 15 clock cycles).

ÿ Bus cycle time is programmable (up to 15 clock cycles)

Independent read and write timings and protocols to support various memory types and timings.

Output enable and write enable delays are programmable (up to 15 clock cycles).

To generate a continuous clock for FMC_CLK, memory region 1 must also be configured to support synchronous mode (see Section 40.5.6).

(NOR/PSRAM control register). Since all synchronous memories use the same clock, this is crucial for generating continuous output clocks and executing...

During synchronous access, the HB data size must be the same as the storage data width (MWID); otherwise, the FMC_CLK frequency will be adjusted according to the HB data transaction.

Changes may occur (for information on the FMC_CLK division ratio formula, please refer to Section 40.5.5: Synchronization Transactions).

Each memory region has a fixed size of 64MB. Each memory region is configured via a dedicated register (see section 40.5.6).

Section: NOR/PSRAM Control Registers). Programmable parameters for the memory include access time (as shown in the table below) and wait management.

Support (for accessing NOR Flash and PSRAM in burst mode).

When reading data using a NOR Flash/PSRAM controller, only DMA 256-bit reads are possible.

Table 40-11 Programmable Access Parameters for NOR/PSRAM

parameter	Function access mode maximum minimum value unit				
Address establishment duration		asynchronous	0	15	HB clock cycle (HCLK)
The address hold phase lasts for an asynchronous duration, allowing for multiplexing of I/O.			1	15	HB clock cycle (HCLK)
Data creation; duration of the data creation phase		asynchronous	1	256	HB clock cycle (HCLK)
Bus turnaround; Bus turnaround phase duration; Asynchronous and synchronous read/write;			0	15	HB clock cycle (HCLK)
Clock division ratio	Construct a memory clock cycle (CLK) Required HB clock cycles Number of periods (HCLK)	synchronous	2	16	HB clock cycle (HCLK)
Data delay	The first data release was made in a sudden outbreak. Clock cycles issued by forward memory Number of periods	synchronous	2	17	Memory clock cycle Period (HCLK)

40.5.1 External Memory Interface Signals

For signals typically used to connect NOR Flash, SRAM, and PSRAM, please refer to Tables 40-12, 40-13, 40-14, and 40-15.

15. The prefix "N" indicates a signal that is active low.

Table 40-12 Non-multiplexed I/O NOR Flash

FMC signal name	I/O	Functions
CLK	O	Clock (used for synchronized access)
A[25:0]	O	Address bus
D[31:0]	I/O	bidirectional data bus
NE[x]	O	Chip select, x=1ÿ4
NOE	O	Output enable
NWE	O	Write Enable
NL (=NADV)	O	Latch Enable (For some NOR Flash devices, this signal is also called Address Valid (NADV))
NWAIT	I	FMC's NOR Flash awaits input signal

The maximum capacity is 512Mb (26 address lines).

Table 40-13 16-bit Multiplexed I/O NOR Flash

FMC signal name	I/O	Function
CLK	O	Clock (used for synchronized access)
A[25:16]	O	Address bus
AD[15:0]	I/O	16-bit multiplexed, bidirectional address/data bus (16-bit address A[15:0] and data bus are multiplexed)
NE[x]	O	Chip select, x=1ÿ4
NOE	O	Output enable
NWE	O	Write Enable
NL (=NADV)	O	Latch Enable (For some NOR Flash devices, this signal is also called Address Valid (NADV))
NWAIT	I	FMC's NOR Flash awaits input signal

The maximum capacity is 512Mb.

Table 40-14 Non-multiplexed I/O PSRAM/SRAM

FMC signal name	I/O	Functions
CLK	O	Clock (for PSRAM synchronous access only)
A[25:0]	O	Address bus
D[31:0]	I/O	bidirectional data bus
NE[x]	O	Chip select, x=1ÿ4 (in PSRAM applications, this is called NCE (Cellular RAM, i.e., CRAM))
NOE	O	Output enable
NWE	O	Write Enable
NL (=NADV)	O	Address valid signal for PSRAM input only (Memory signal name: NADV)
NWAIT	I	PSRAM sends data to FMC to wait for input signals.
NBL [1:0]	O	Byte channel output. Bytes 0 to 3 control. (High byte and low byte enable)

The maximum capacity is 512Mb.

Table 40-15 16-bit Multiplexed I/O PSRAM

FMC signal name	I/O	Function
CLK	O	Clock (used for synchronized access)
A[25:16]	O	Address bus
AD[15:0]	I/O	16-bit multiplexed, bidirectional address/data bus (The 16-bit address A[15:0] and data D[15:0] are multiplexed on the data bus)
NE[x]	O	Chip select, x=1 ÿ4 (in PSRAM applications, this is called NCE (Cellular RAM, i.e., CRAM))
NOE	O	Output enable
NWE	O	Write Enable
NL (=NADV)	O	Address valid signal for PSRAM input (Memory signal name: NADV)
NWAIT	I	PSRAM sends data to FMC to wait for input signals.
NBL [1:0]	O	Byte channel output. Byte 0 to Byte 1 control. (High byte and low byte enable)

The maximum capacity is 512Mb (26 address lines).

40.5.2 Supported Memory and Transactions

Supported devices, access modes, and transactions when the memory data bus width of NOR Flash, PSRAM, and SRAM is 16 bits.

Examples are shown in the table below. In this example, transactions that FMC does not allow (or does not support) are shown in gray.

Table 40-16 NOR Flash/PSRAM: Examples of Supported Memory and Transactions

equipment	Pattern R/W	HB Data size	memory Data size	Is it allowed?	Notes
NOR Flash (Multiplexed I/O and non-multiplexed I/O) using I/O)	Asynchronous	8	16		-
	R Asynchronous	8	16		-
	W Asynchronous	16	16		-
	R Asynchronous	16	16		-
	W Asynchronous	32	16		Two FMC visits
	R Asynchronous	32	16		Two FMC visits
	W Asynchronous	-	16		This mode is not supported.
	Page R	8	16		-
	Synchronous	16	16		-
	R Synchronous	32	16		-
PSRAM (Multiplexed I/O and) Non-multiplexed I/O	R Synchronous	8	16		-
	R Asynchronous	8	16	Yes/No Yes/No Yes/No Yes/No Yes Yes Yes	Use byte channe
	R Asynchronous	16	16		-
	W Asynchronous	16	16		-
	R Asynchronous	32	16		Two FMC visits
	W Asynchronous	32	16		Two FMC visits
	R Asynchronous	-	16		This mode is not supported.
	W Asynchronous	8	16		-
	Page R	16	16		-
	Synchronous	32	16		-
	R Synchronous R Synchronous W	8	16	Yes Yes Yes No No Yes Yes Yes	Use byte channel NBL[1:0

	Synchronous	16/32	16		-
SRAM and ROM	W Asynchronous	8/16	16		-
	R Asynchronous	8/16	16	Yes, yes, we are using the byte channel NBL[1:0].	
	W Asynchronous R	32	16	It is divided into 2 FMC visits	
	Asynchronous W	32	16	yes	Divided into 2 FMC visits, Use byte channel NBL[1:0]

40.5.3 General Timing Rules

Signal synchronization

ÿ All controller output signals change on the rising edge of the internal clock (HCLK);

In synchronous mode (read/write), all output signals change on the rising edge of HCLK. Regardless of the CLKDIV value, all outputs ...

The signal changes are as follows:

- The NOEL/NWEL/NEL/NADVL/NADVH/NBLL address validity output changes on the falling edge of the FMC_CLK clock.
- The NOEH/NWEH/NEH/NOEH/NBLH address validity output changes on the rising edge of the FMC_CLK clock.

40.5.4 Asynchronous Transactions for NOR Flash/PSRAM Controllers

Asynchronous static memory (NOR Flash, PSRAM, SRAM) –

Signals are synchronized via an internal clock HCLK. This clock is not sent to the memory.

The FMC always samples the data first and then disables the chip select signal NE. This ensures compliance with memory data retention timing requirements.

(The shortest time from chip select enable to data conversion is typically 0ns).

If extended mode is enabled (EXTMOD bit set to 1 in the R32_FMC_BCRx register), a maximum of four extended modes are available (A, B, C, D, E, F ... D, E, F, C, D, E, F, D, E, F, D, E, F, D, E, F, D, E, F, D, E, F,

(B, C, D). Read and write operations can be performed using a combination of modes A, B, C, and D. For example, a read operation can be performed in mode A.

In mode B, a fetch operation is performed, while in mode B, a write operation is performed.

If extended mode is disabled (the EXTEMOD bit in the R32_FMC_BCRx register is reset), the FMC can operate in either mode 1 or mode 2.

The process is as follows:

- When SRAM/PSRAM memory type is selected, Mode 1 is the default mode (MTYP=0x0 in R32_FMC_BCRx register or...).

  0x01);

- When NOR memory type is selected, mode 2 is the default mode (MTYP=0x10 in R32_FMC_BCRx register).

Mode 1 - SRAM/PSRAM (CRAM) is

shown in the figure below. Supported modes for read and write transactions, and the required R32_FMC_BCRx and R32_FMC_BTRx/R32_FMC_BWTRx register configuration.

Figure 40-3 Mode 1 Read Access Waveform

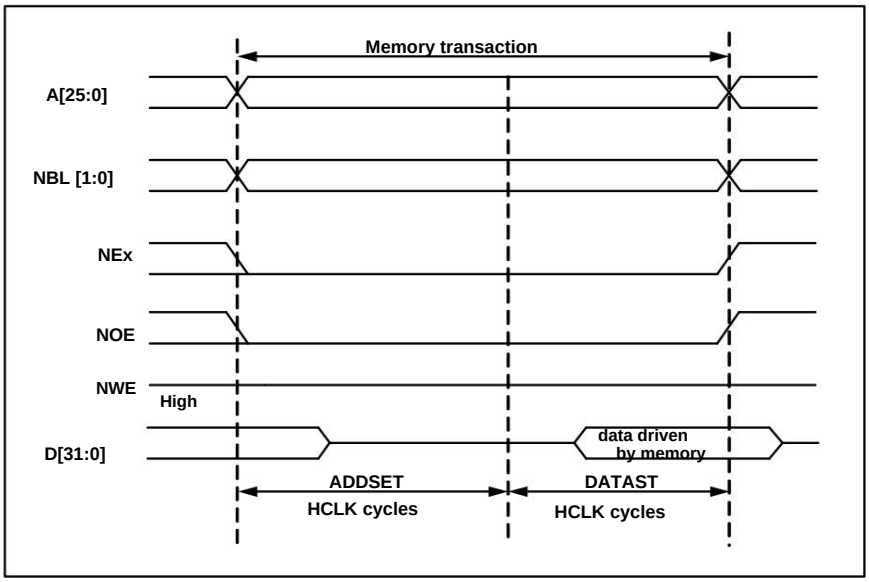
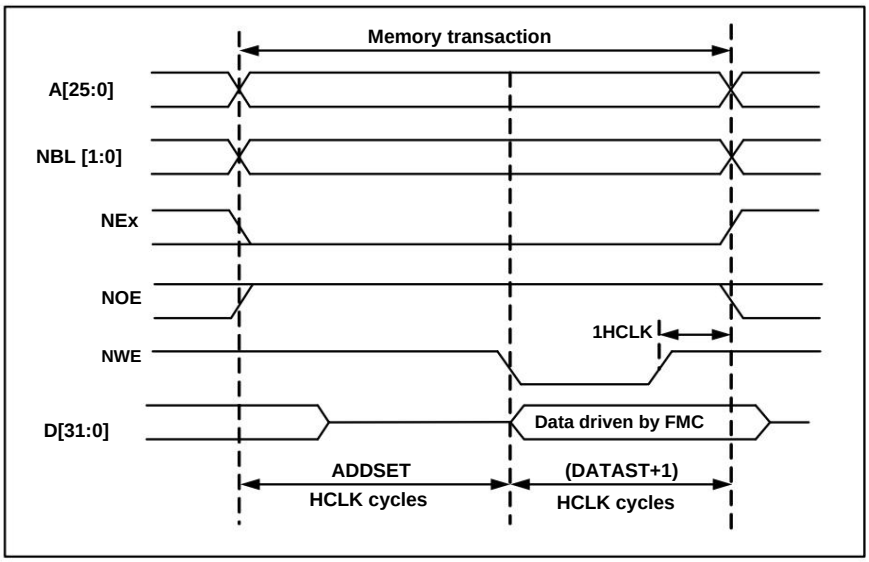


Figure 40-4 Mode 1 Write Access Waveform



An HCLK cycle at the end of a write transaction helps ensure the retention time of addresses and data after the rising edge of NEW. Because of this...

During the HCLK period, the DATAST value must be greater than 0 (DATAST>0).

Table 40-17 R32_FMC_BCRx bit fields

Position	Name	Value to be set
31	FMC_EN	0x1
[30:26]	reserved	0x00
[25:24]	BMP	0x0 or 0x1
[23:20]	reserve	0x0
19	CBURSTRW	0x0 (has no effect on asynchronous mode)
[18:16]	CPSIZE	0x0 (Has no effect on asynchronous mode)
15	ASYNCWAIT	is set to 1 if the memory supports this feature. Otherwise, it remains at 0.
14	EXTMOD	0x0
13	WAITEN	0x0 (Has no effect on asynchronous mode)
12	WREN	Configure as needed

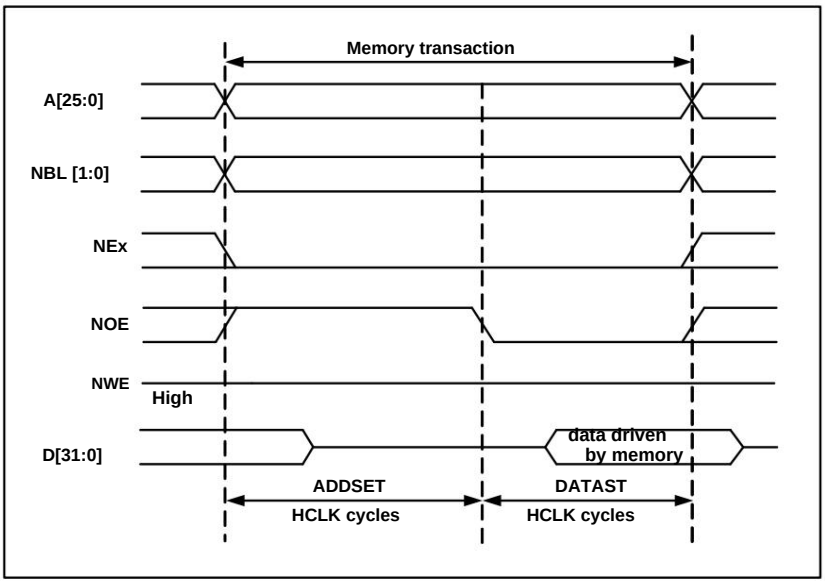
11	reserve	0x0
10	WRAPMOD	0x0
9	WAITPOL is only meaningful when bit 15 is 1.	
8	BURSTEN	0x0
7	reserve	0x1
6	FACCEN	irrelevant
[5:4]	MWID	Configure as needed
[3:2]	MTYP	Configure as needed, except for 0x2 (NOR Flash).
1	MUXE	0x0
0	MBKEN	0x1

Table 40-18 FMC_BTRx Bit Fields

Position	Name	Value to be set
[31:30]	reserve	0x00
[29:28]	ACCMOD	irrelevant
[27:24]	DATLAT	irrelevant
[23:20]	CLKDIV	irrelevant
[19:16]	BUSTURN	The time between NEx going high and NEx going low (BUSTURN HCLK)
[15:8]	DATAST	The duration of the second access phase (for write access, it is DATAST+1). HCLK cycles, read access is DATAST HCLK cycles.
[7:4]	ADDHLD	irrelevant
[3:0]	ADDSET	The duration of the first access phase (ADDSET HCLK cycles). The minimum value of ADDSET is 0.

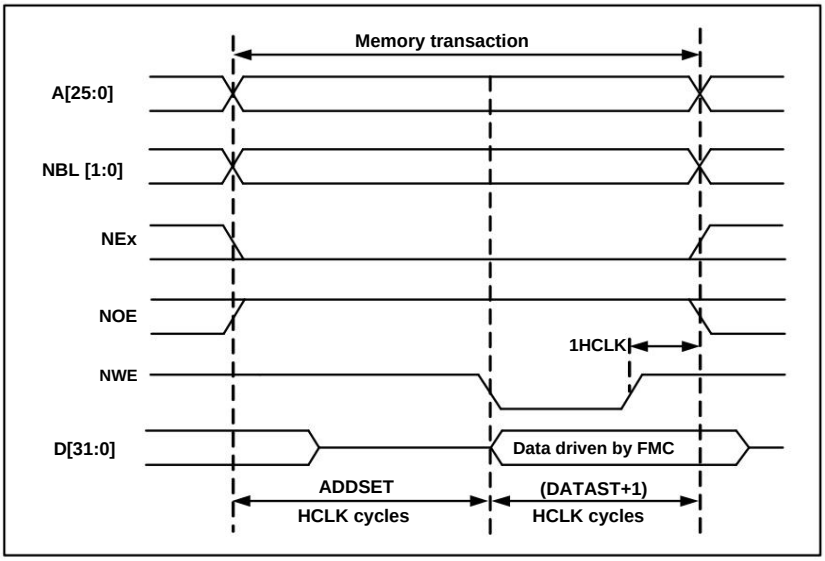
Mode A-SRAM/PSRAM (CRAM) OE switching

Figure 40-5 Mode A Read Access Waveform



Note: NBL[1:0] is low when read access is performed.

Figure 40-6 Mode A Write Access Waveform



The difference from Mode 1 lies in the switching of NOE and the independent read and write timing.

Table 40-19 R32_FMC_BCRx bit fields

Position	Name	Value to be set
31	FMC_EN	0x1
[30:26]	reserved	0x00
[25:24]	BMP	0x0 or 0x1
[23:20]	reserve	0x0
19	CBURSTRW	0x0 (has no effect on asynchronous mode)
[18:16]	CPSIZE	0x0 (Has no effect on asynchronous mode)
15	ASYNCWAIT	is set to 1 if the memory supports this feature. Otherwise, it remains at 0.
14	EXTMOD	0x1
13	WAITEN	0x0 (Has no effect on asynchronous mode)
12	WREN	Configure as needed
11	WAITCFG	is irrelevant.
10	reserve	0x0
9	WAITPOL	is only meaningful when bit 15 is 1.
8	BURSTEN	0x0
7	reserve	0x1
6	FACCEN	irrelevant
[5:4]	MWID	Configure as needed
[3:2]	MTYP	Configure as needed, except for 0x2 (NOR Flash).
1	MUXEN	0x0
0	MBKEN	0x1

Table 40-20 R32_FMC_BTRx Bit Fields

Position	Reserved	Value to be set
[31:30]	bit name	0x00
[29:28]	ACCMOD	irrelevant

[27:24]	DATLAT	irrelevant
[23:20]	CLKDIV	irrelevant
[19:16]	BUSTURN	The time between NEx going high and NEx going low (BUSTURN HCLK)
[15:8]	DATAST	Duration of the second phase of write access (DATAST + 1 HCLK cycles) Expect).
[7:4]	ADDHLD	irrelevant
[3:0]	ADDSET	Duration of the first phase of write access (ADDSET + 1 HCLK cycle) (Period). The minimum value of ADDSET is 0.

Table 40-21 R32_FMC_BWTRx bit fields

	Reserved	Value to be set
Position	bit name	0x0
numbers	ACCMOD	0x0
[31:30]	DATLAT	irrelevant
[29:28] [27:24] [23:20]	CLKDIV	irrelevant
[19:16]	BUSTURN	The time between NEx going high and NEx going low (BUSTURN HCLK)
[15:8]	DATAST	Duration of the second phase of write access (DATAST HCLK weeks) Expect).
[7:4]	ADDHLD	irrelevant
[3:0]	ADDSET	Duration of the first phase of write access (ADDSET HCLK weeks) (Period). The minimum value of ADDSET is 0.

Mode 2/B-NOR Flash

Figure 40-7 Read access waveforms for Mode 2 and Mode B

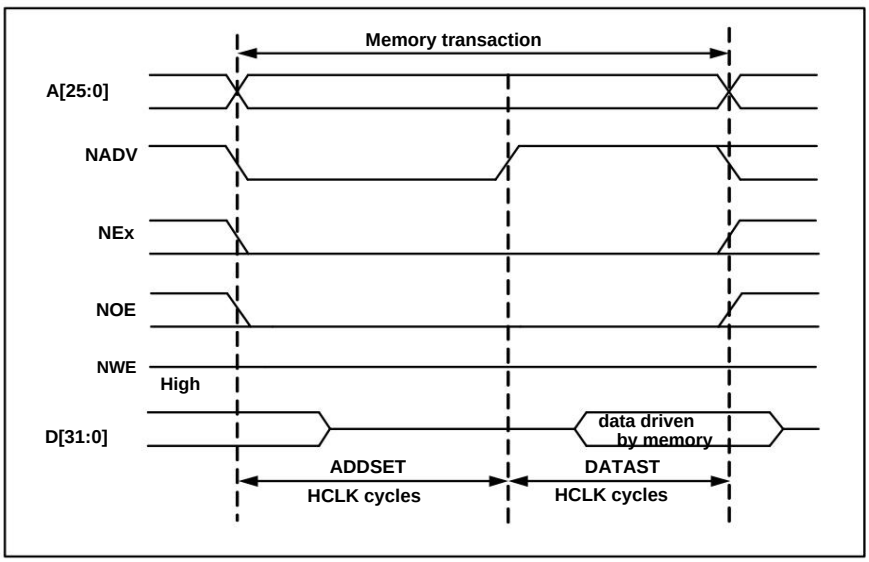


Figure 40-8 Mode 2 Write Access Waveform

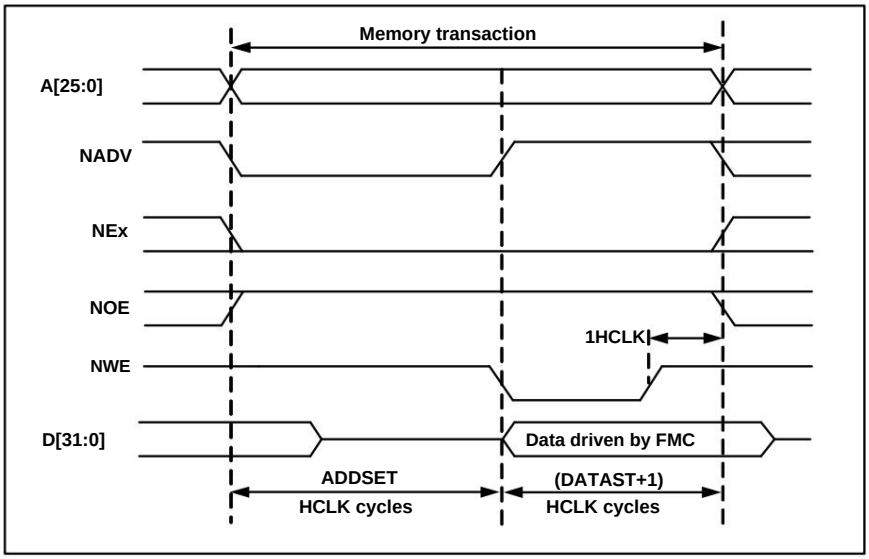
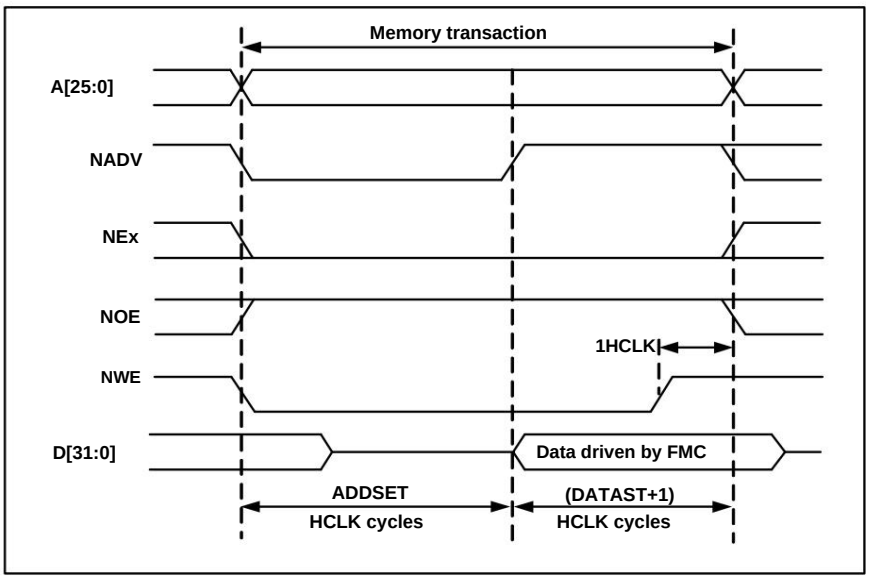


Figure 40-9 Mode B Write Access Waveform



The difference from Mode 1 is the NWE switching and independent read and write timing when setting extended mode (Mode B).

Table 40-22 R32_FMC_BCRx bit fields

Position	Name	Value to be set
31	FMC_EN	0x1
[30:26]	reserved	0x00
[25:24]	BMP	0x0 or 0x1
[23:20]	reserve	0x0
19	CBURSTRW	0x0 (has no effect on asynchronous mode)
[18:16]	CPSIZE	0x0 (Has no effect on asynchronous mode)
15	ASYNCWAIT	is set to 1 if the memory supports this feature. Otherwise, it remains at 0.
14	EXTMOD	mode B is 0x1, mode 2 is 0x0 0x0 (has no effect on
13	WAITEN	asynchronous mode)
12	WREN	Configure as needed
11	WAITCFG	is irrelevant.

10	reserve	0x0
9	WAITPOL is only meaningful when bit 15 is 1.	
8	BURSTEN	0x0
7	reserve	0x1
6	FACCEN	0x1
[5:4]	MWID	Configure as needed
[3:2]	MTYP	0x2 (NOR Flash)
1	MUXEN	0x0
0	MBKEN	0x1

Table 40-23 R32_FMC_BTRx bit fields

Position	Reserved	Value to be set
[31:30]	bit name	0x00
[29:28]	If ACCMOD is set to extended mode, it is 0x1.	
[27:24]	DATLAT	irrelevant
[23:20]	CLKDIV	irrelevant
[19:16]	BUSTURN	The time between NEx going high and NEx going low (BUSTURN HCLK)
[15:8]	DATAST	Duration of the second phase of read access (DATAST HCLK cycles) Expect).
[7:4]	ADDHLD	irrelevant
[3:0]	ADDSET	Duration of the first phase of read access (ADDSET HCLK cycles) (Period). The minimum value of ADDSET is 0.

Table 40-24 R32_FMC_BWTRx bit fields

Position	Name	Value to be set
[31:30]	reserve	0x0
[29:28]	If ACCMOD is set to extended mode, it is 0x1.	
[27:24]	DATLAT	irrelevant
[23:20]	CLKDIV	irrelevant
[19:16]	BUSTURN	The time between NEx going high and NEx going low (BUSTURN HCLK)
[15:8]	DATAST	Duration of the second phase of read access (DATAST HCLK cycles) Expect).
[7:4]	ADDHLD	irrelevant
[3:0]	ADDSET	Duration of the first phase of read access (ADDSET HCLK cycles) (Period). The minimum value of ADDSET is 0.

Note: The R32_FMC_BWTRx register is only valid when extended mode (mode B) is set; otherwise, its contents are irrelevant.

Switching between C-NOR and Flash-OE modes

Figure 40-10 Mode C Read Access Waveform

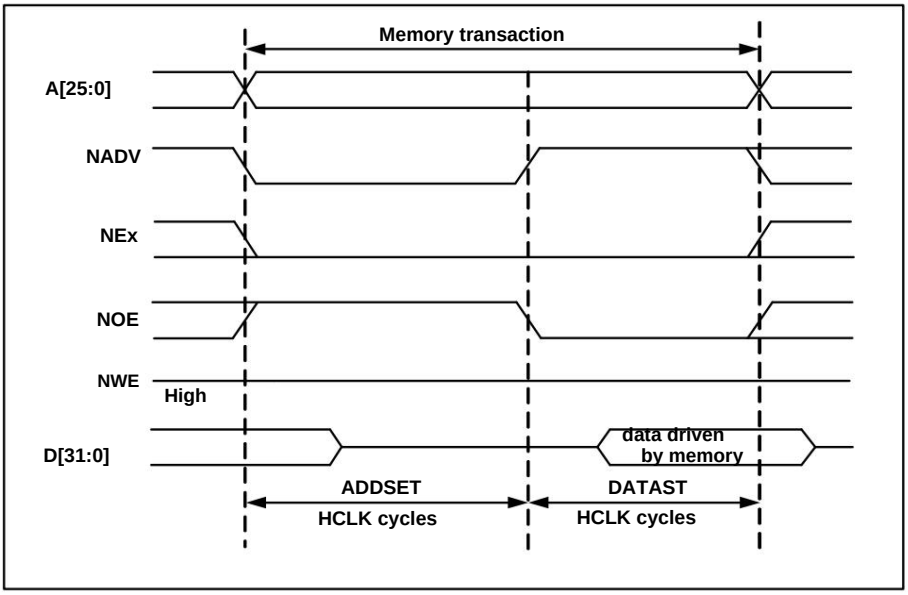
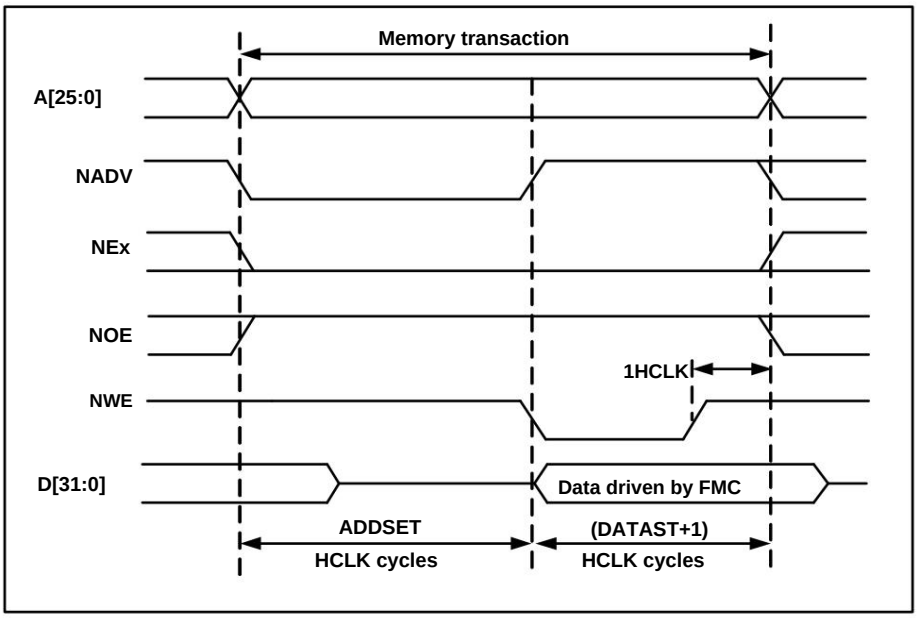


Figure 40-11 Mode C Write Access Waveform



The difference from Mode 1 lies in the switching of NOE and the independent read and write timing.

Table 40-25 R32_FMC_BCRx bit fields

Position	Name	Value to be set
31	FMC_EN	0x1
[30:26]	reserved	0x00
[25:24]	BMP	0x0 or 0x1
[23:20]	reserve	0x0
19	CBURSTRW 0x0 (has no effect on asynchronous mode)	
[18:16]	CPSIZE	0x0 (Has no effect on asynchronous mode)
15	ASYNCWAIT is set to 1 if the memory supports this feature. Otherwise, it remains at 0.	
14	EXTMOD	0x1

13	WAITEN	0x0 (Has no effect on asynchronous mode)
12	WREN	Configure as needed
11	WAITCFG is irrelevant.	
10	reserve	0x0
9	WAITPOL is only meaningful when bit 15 is 1.	
8	BURSTEN	0x0
7	reserve	0x1
6	FACCEN	0x1
[5:4]	MWID	Configure as needed
[3:2]	MTYP	0x2 (NOR Flash)
1	MUXEN	0x0
0	MBKEN	0x1

Table 40-26 R32_FMC_BTRx Bit Fields

Position	Name	Value to be set
[31:30]	reserve	0x0
[29:28]	ACCMOD	0x2
[27:24]	DATLAT	0x0
[23:20]	CLKDIV	0x0
[19:16]	BUSTURN	The time between NEx going high and NEx going low (BUSTURN HCLK)
[15:8]	DATAST	Duration of the second phase of write access (DATAST + 1 HCLK cycles) Expect)
[7:4]	ADDHLD	irrelevant
[3:0]	ADDSET	Duration of the first phase of the write access (ADDSET + 1 HCLK) (Period). The minimum value of ADDSET is 0.

Table 40-27 R32_FMC_BWTRx Bit Fields

	Reserved	Value to be set
Position	bit name	0x0
numbers	ACCMOD	0x2
[31:30]	DATLAT	irrelevant
[29:28] [27:24] [23:20]	CLKDIV	irrelevant
[19:16]	BUSTURN	The time between NEx going high and NEx going low (BUSTURN HCLK)
[15:8]	DATAST	Duration of the second phase of write access (DATAST HCLK weeks) Expect).
[7:4]	ADDHLD	irrelevant
[3:0]	ADDSET	Duration of the first phase of write access (ADDSET HCLK weeks) (Period). The minimum value of ADDSET is 0.

Mode D - Asynchronous Access to Extended Addresses

Figure 40-12 Mode D Read Access Waveform

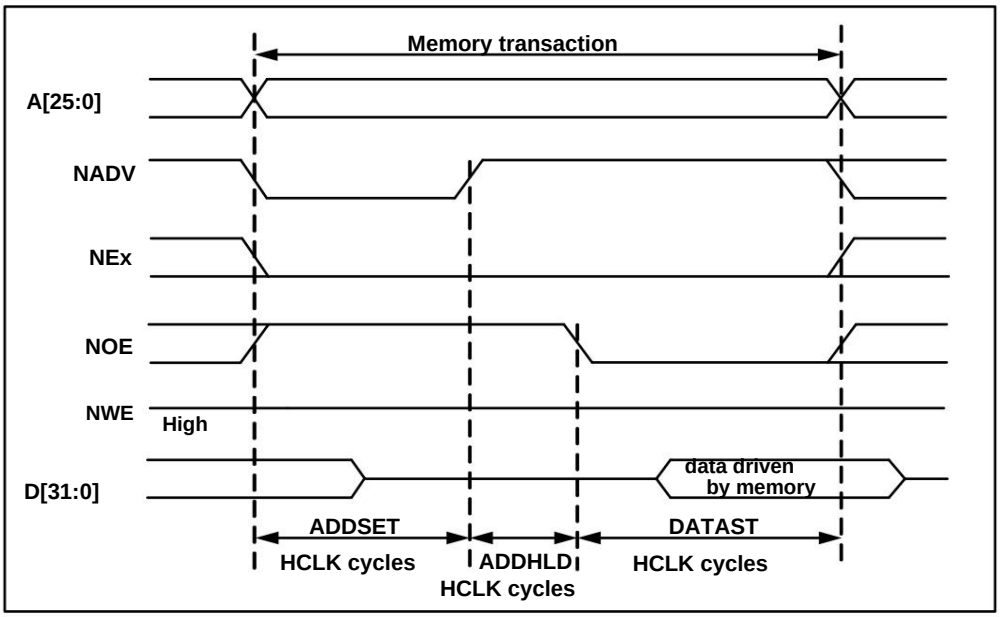
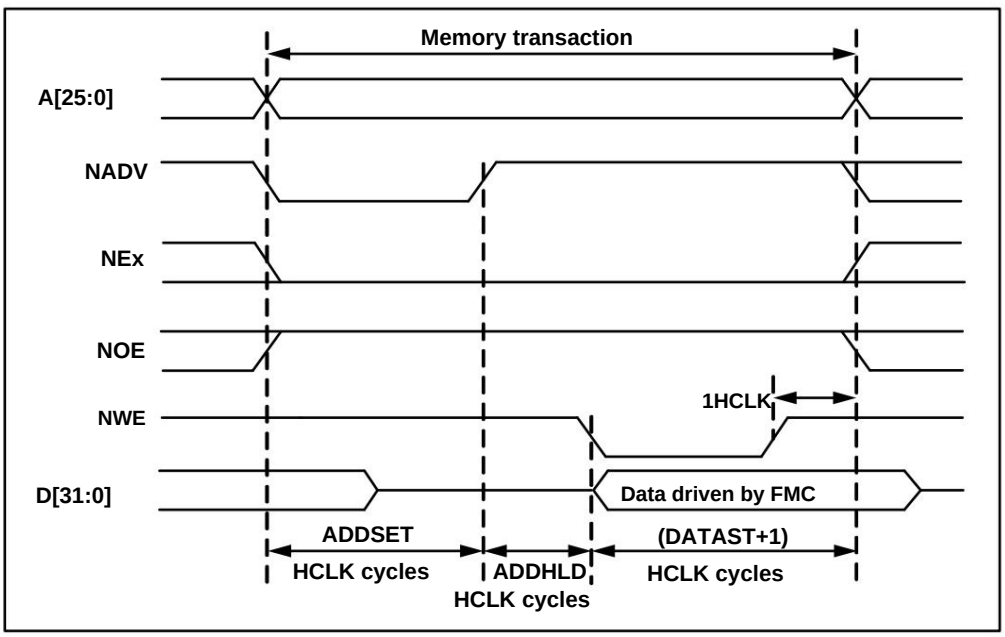


Figure 40-13 Mode D Write Access Waveform



The difference from Mode 1 lies in the switching of NOE after NADV changes and the independent read and write timings.

Table 40-28 R32_FMC_BCRx bit fields

Position	Name	Value to be set
31	FMC_EN	0x1
[30:26]	reserved	0x00
[25:24]	BMP	0x0 or 0x1
[23:20]	reserve	0x0
19	CBURSTRW 0x0 (has no effect on asynchronous mode)	
[18:16]	CPSIZE	0x0 (Has no effect on asynchronous mode)
15	ASYNCWAIT is set to 1 if the memory supports this feature. Otherwise, it remains at 0.	

14	EXTMOD	0x1
13	WAITEN	0x0 (Has no effect on asynchronous mode)
12	WREN	Configure as needed
11	WAITCFG is irrelevant.	
10	reserve	0x0
9	WAITPOL is only meaningful when bit 15 is 1.	
8	BURSTEN	0x0
7	reserve	0x1
6	FACCEN is set according to memory support.	
[5:4]	MWID	Configure as needed
[3:2]	MTYP	Configure as needed
1	MUXEN	0x0
0	MBKEN	0x1

Table 40-29 R32_FMC_BTRx Bit Fields

Position	Name	Value to be set
[31:30]	reserve	0x0
[29:28]	ACCMOD	0x3
[27:24]	DATLAT	irrelevant
[23:20]	CLKDIV	irrelevant
[19:16]	BUSTURN	The time between NEx going high and NEx going low (BUSTURN HCLK).
[15:8]	DATAST	Duration of the second phase of write access (DATAST + 1 HCLK cycles) Expect).
[7:4]	ADDHLD	The duration of the intermediate stage of the read access (ADDHLD HCLK cycles).
[3:0]	ADDSET	Duration of the first phase of write access (ADDSET + 1 HCLK cycle) (Period). The minimum value of ADDSET is 1.

Table 40-30 R32_FMC_BWTRx bit fields

Position	Reserved	Value to be set
[31:30]	bit name	0x0
[29:28]	ACCMOD	0x3
[27:24]	DATLAT	irrelevant
[23:20]	CLKDIV	irrelevant
[19:16]	BUSTURN	The time between NEx going high and NEx going low (BUSTURN HCLK).
[15:8]	DATAST	The duration of the second access phase (for write access, it is DATAST+1). HCLK cycle).
[7:4]	ADDHLD	The duration of the write access intermediate phase (ADDHLD HCLK cycles).
[3:0]	ADDSET	Duration of the first phase of write access (ADDSET HCLK weeks) (Period). The minimum value of ADDSET is 0.

Multiplexing mode - Multiplexing asynchronous access to NOR Flash

Figure 40-14 Multiplexed read access waveform

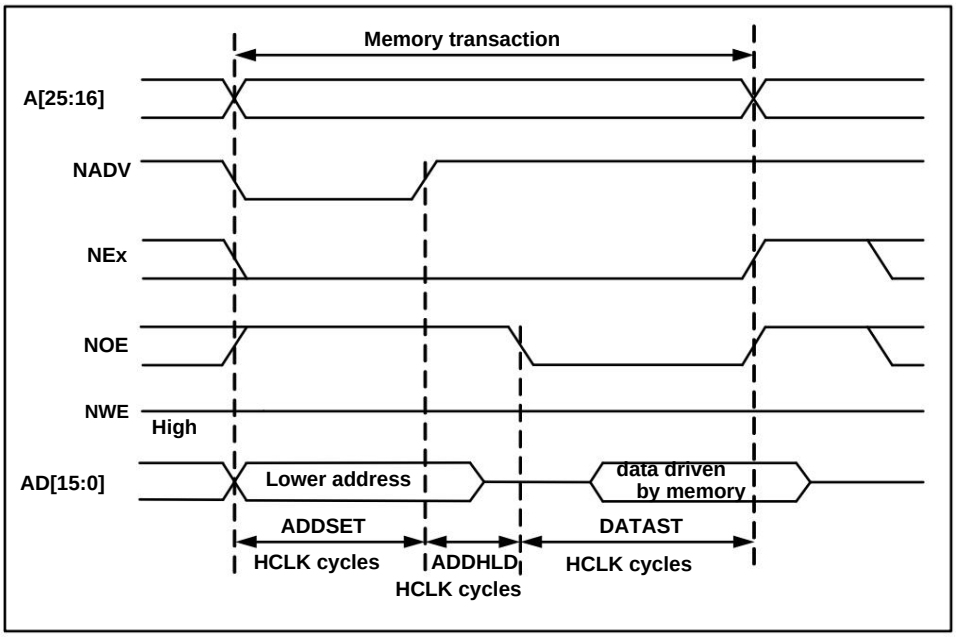
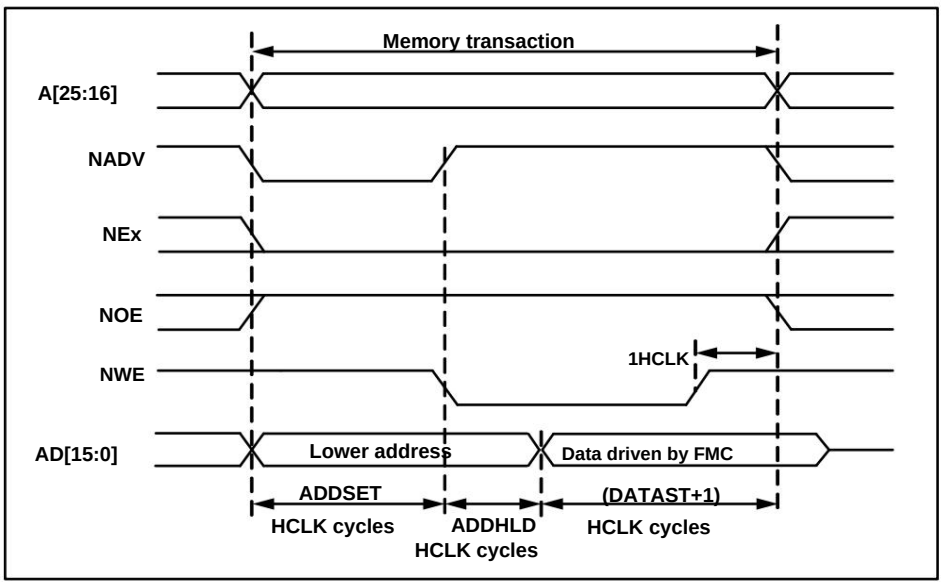


Figure 40-15 Multiplexed write access waveform



The difference from mode D is that the low address byte is driven on the data bus.

Table 40-31 R32_FMC_BCRx bit fields

Position	Name	Value to be set
31	FMC_EN	0x1
[30:26]	reserve	0x00
[25:24]	BMP	0x0 or 0x1
[23:20]	reserve	0x0
19	CBURSTRW	0x0 (has no effect on asynchronous mode)
[18:16]	CPSIZE	0x0 (Has no effect on asynchronous mode)
15	ASYNCWAIT	is set to 1 if the memory supports this feature. Otherwise, it remains at 0.
14	EXTMOD	0x0

13	WAITEN	0x0 (Has no effect on asynchronous mode)
12	WREN	Configure as needed
11	WAITCFG is irrelevant.	
10	reserve	0x0
9	WAITPOL is only meaningful when bit 15 is 1.	
8	BURSTEN	0x0
7	reserve	0x1
6	FACCEN	0x1
[5:4]	MWID	Configure as needed
[3:2]	MTYP	0x2 (NOR Flash)
1	MUXEN	0x1
0	MBKEN	0x1

Table 40-32 R32_FMC_BTRx bit fields

Position	Name	Value to be set
[31:30]	reserve	0x0
[29:28]	ACCMOD	0x0
[27:24]	DATLAT	irrelevant
[23:20]	CLKDIV	irrelevant
[19:16]	BUSTURN	The time between NEx going high and NEx going low (BUSTURN HCLK).
[15:8]	DATAST	The duration of the second access phase (read access is DATAST times) HCLK cycles, write access is DATAST+1 HCLK cycles.
[7:4]	ADDHLD	Duration of accessing the intermediate stage (ADDHLD HCLK cycles).
[3:0]	ADDSET	The duration of the first access phase (ADDSET HCLK cycles). The minimum value of ADDSET is 1.

WAIT Management in Asynchronous

Access: If the asynchronous memory issues a WAIT signal, indicating that it is not yet ready to accept or provide data, the R32_FMC_BCRx register... The ASYNCWAIT bit must be set to 1.

If the WAIT signal is active (its level depends on the WAITPOL bit), then the second access is controlled by the DATAST bit. The data setup phase will be extended until WAIT becomes invalid. Unlike the data setup phase, this is handled by ADDSET and ADDHLD. The first access phase of bit control (address setup and address holding phases) is not sensitive to WAIT, therefore the first access phase will not be delayed. long.

The data setup phase must be configured so that WAIT can be detected 4 HCLK cycles before the end of the memory transaction. The following must be considered: condition:

1) Alignment of the WAIT and NOE/NEW signals issued by the memory:

DATAST ÿ (4*HCLK) + max_wait_assertion_time 2) The WAIT signal emitted by the memory is aligned with NEX (or the NOE/NWE signal does not toggle):

if

max_wait_assertion_time>address_phase+hold_phase

So:

DATASTÿ(4*HCLK)+(max_wait_assertion_time−address_phase−hold_phase)

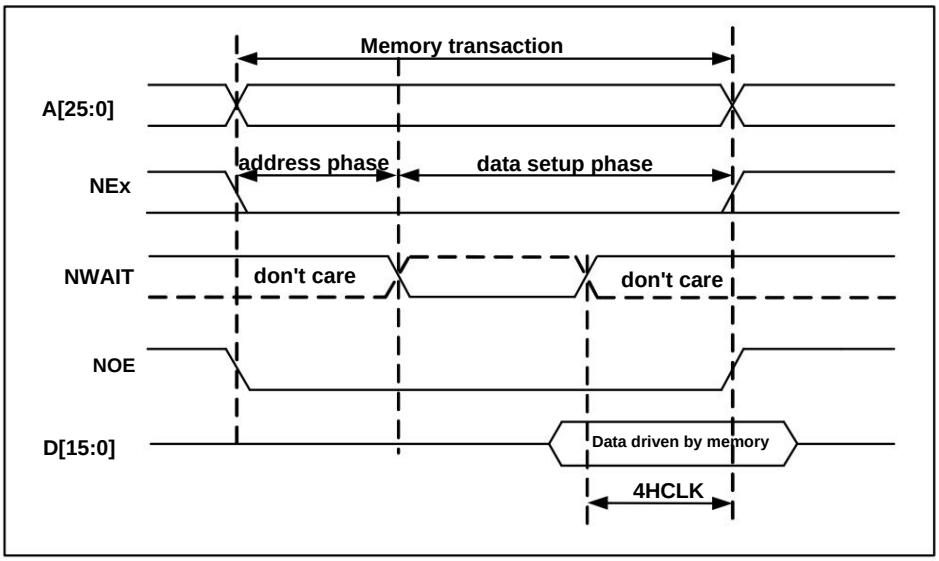
otherwise

DATASTÿ4*HCLK

Among them, max_wait_assertion_time is the time spent by the memory to enable the WAIT signal after NEx/NOE/NWE goes low.
The longest time.

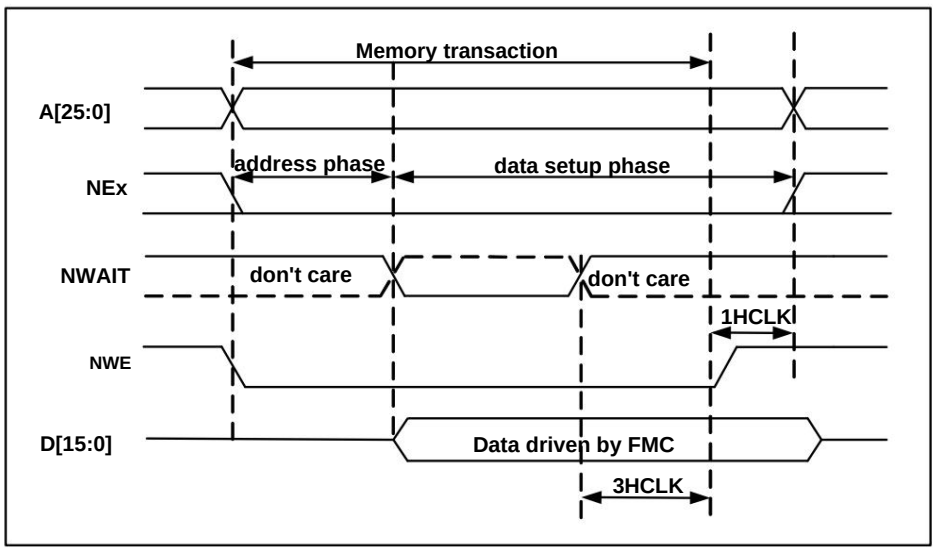
Figures 40-15 and 40-16 show the number of HCLK clock cycles added during the memory access phase after asynchronous memory release
WAIT (unrelated to the above case).

Figure 40-16 Asynchronous wait during read access waveform



Note: The polarity of NWAIT depends on the WAITPOL bit setting in the R32_FMC_BCRx register.

Figure 40-17 Asynchronous wait during write access waveform



Note: The polarity of NWAIT depends on the WAITPOL bit setting in the R32_FMC_BCRx register.

40.5.5 The synchronous

transaction memory clock FMC_CLK is a divisor of HCLK. It depends on the value of CLKDIV and the MWID/HB data size, as shown in the following formula.
Show:

FMC_CLK division ratio = max(CLKDIV+1, MWID(HB data size)). When MWID is 16-bit
or 8-bit, the FMC_CLK division ratio is always defined by the setting value of CLKDIV.

When MWID is 32 bits, the FMC_CLK division ratio also depends on the HB data size.

Example:

When CLKDIV=1, MWID=32 bits, and HB data size=8 bits, FMC_CLK=HCLK/4. When CLKDIV=1, MWID=16 bits, and HB data size=8 bits, FMC_CLK=HCLK/2.

The NOR Flash specifies the minimum time from NADV enable to CLK high. To comply with this constraint, the FMC will not send the clock to the memory during the first internal clock cycle of synchronous access (before NADV enable). This ensures that the rising edge of the memory clock occurs in the middle of the NADV low pulse. Data latency is the number of cycles

required before data is sampled.

The value of DATLAT must be consistent with the NOR FLASH configuration register.

The delay value is consistent with the value specified in the code. When NADV is low in the data delay count, FMC will not be counted in the clock cycle.

Some NOR Flash memory incorporates the NADV low-level period into the data delay count, thus affecting the NOR Flash delay and FMC DATLAT parameters. The exact relationship can be any of the following: ÿ NOR Flash latency = (DATLAT + 2) CLK clock cycles ÿ NOR Flash latency = (DATLAT + 3) CLK clock cycles

Recently, some memories have enabled NWAIT during the latency phase. In this case, DATLAT can be set to a minimum value. Then, the FMC samples the data and waits long enough to evaluate its validity. This allows the FMC to detect the latency in the memory and process the actual data.

Other memory will not enable NWAIT during the latency period. In this case, the latency of FMC and memory must be set correctly, otherwise... This could lead to invalid data being misused as valid data, or valid data being lost during the initial stage of memory access.

When the selected

memory region is configured for synchronous burst mode, for example, if a single burst transaction (HB) is requested from a 16-bit memory, the FMC will execute a burst transaction of length 1 (if the HB transfer is 16 bits) or a burst transaction of length 2 (if the HB transfer is 32 bits), and then disable the chip select signal on the last data strobe. Compared to asynchronous read operations, this is not the most efficient transfer

method in terms of cycle time. However, random asynchronous reads require a re-run... Programming memory access modes will result in a longer overall

time. Pages exceeding the boundaries of Cellular RAM

1.5 are not allowed to be accessed by bursts. When the memory page size is determined by configuring the CPSIZE bit in the R32_FMC_BCR1 register, the FMC controller automatically separates burst accesses.

For synchronous

NOR Flash, wait management evaluates NWAIT after a configured delay period (equivalent to (DATLAT+2) CLK clock cycles). If NWAIT is active (low when WAITPOL=0,

high when WAITPOL=1), a wait state is inserted until NWAIT becomes inactive (high when WAITPOL=0, low when WAITPOL=1). When NWAIT becomes inactive, data is considered active immediately (bit WAITCFG=1) or on the next clock edge (bit WAITCFG=0).

effect.

During the wait cycle, the controller continues to send clock pulses to the memory via the NWAIT signal, maintaining chip select and output enable. The signal is valid, but the data is not considered valid. In

burst mode, the NOR Flash NWAIT signal has two timing configurations: ÿ The Flash emits the NWAIT signal one data cycle before the wait cycle (default after reset). ÿ The Flash emits the NWAIT signal during the wait cycle.

FMC supports both NOR Flash wait cycle configurations, which are configured for each chip select via the WAITCFG bit (x=0..3) in the R32_FMC_BCRx register.

Figure 40-18 Waiting for configuration waveform

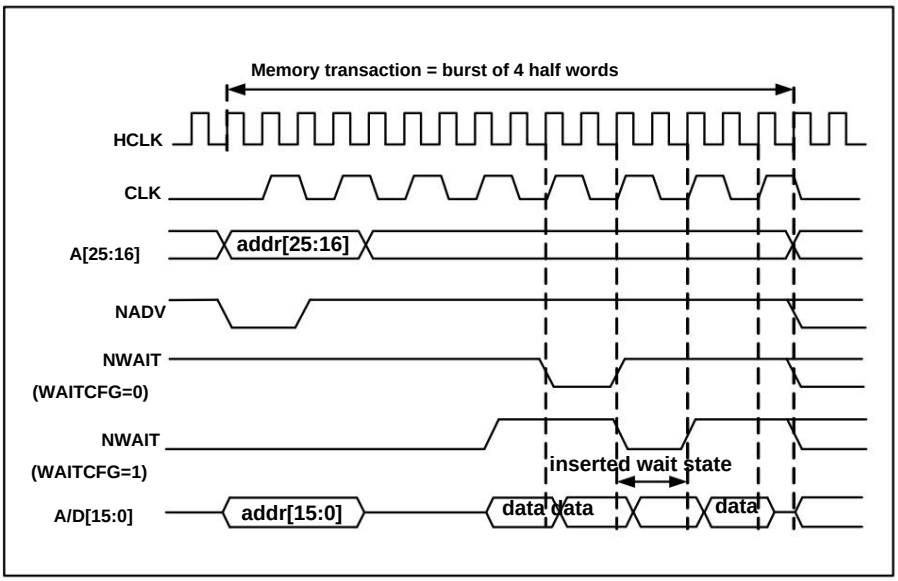
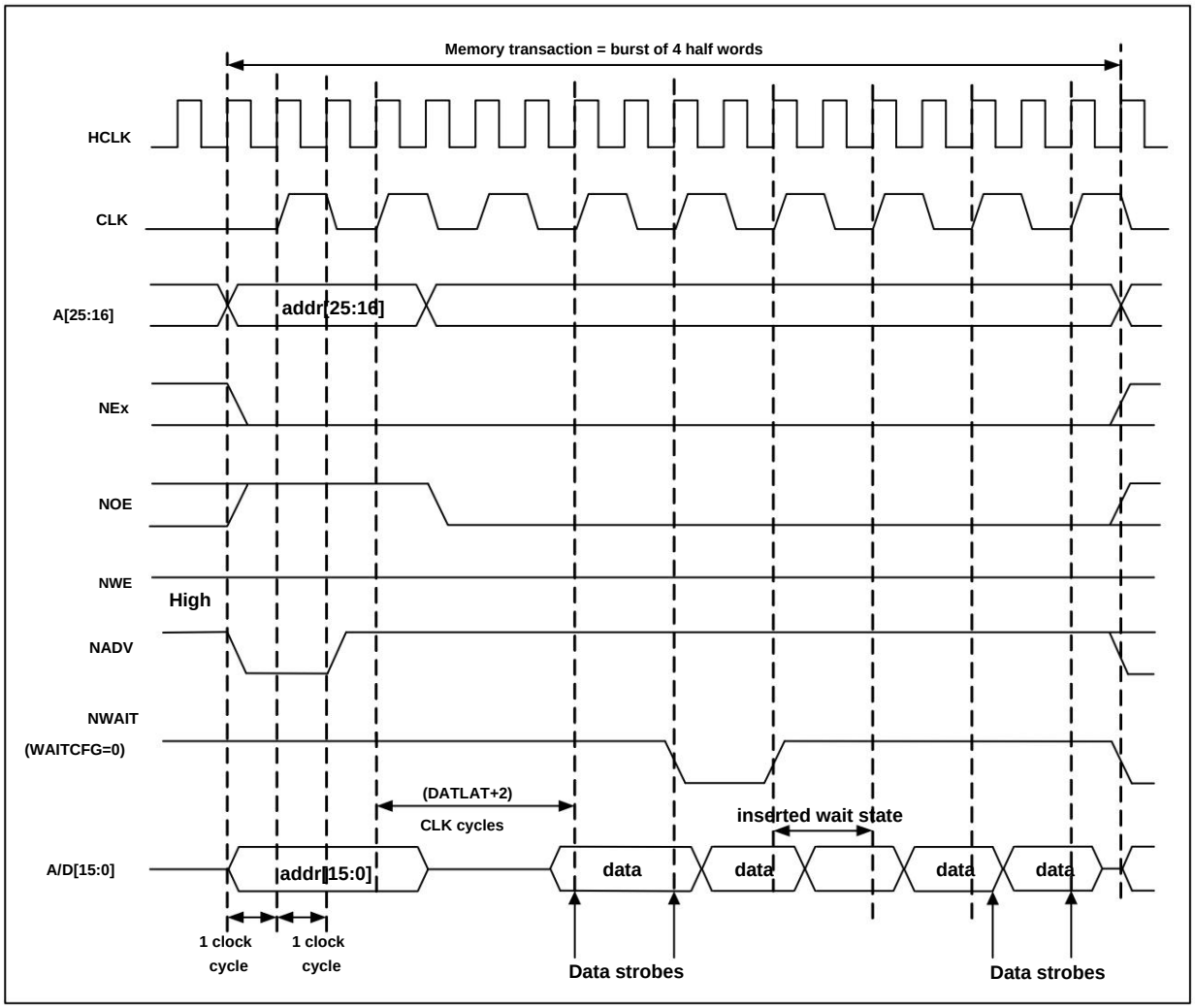


Figure 40-19 Waveforms of Synchronous Multiplexing Read Mode - NOR, PSRAM (CRAM)



Note: Byte channel output NBL Not shown, they are for NOR Access is kept high for PSRAM (CRAM) access and low for PSRAM (CRAM) access. tie.

Table 40-33 R32_FMC_BCRx bit fields

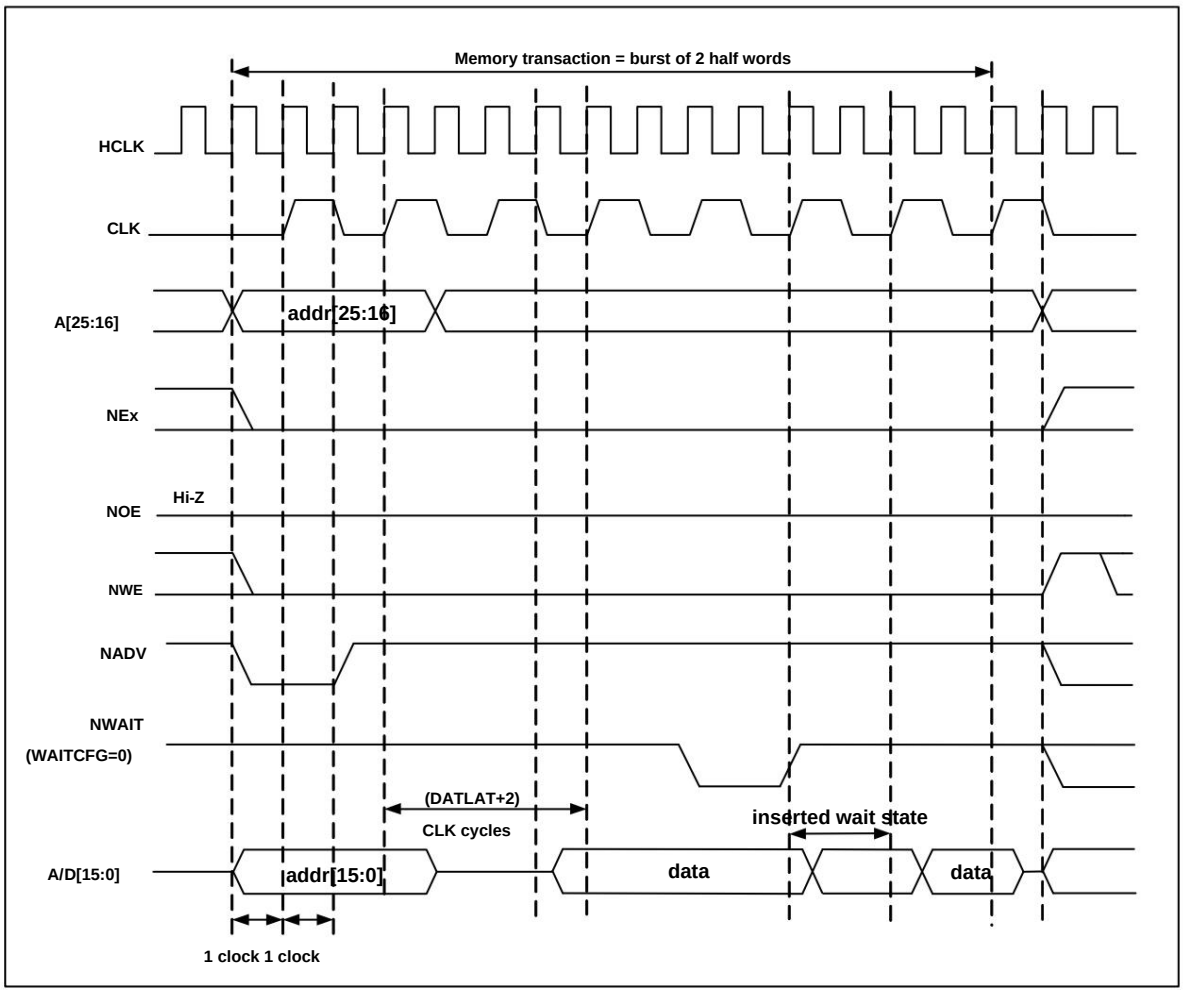
Position	Name	Value to be set

31	FMC_EN	0x1
[30:26]	reserved	0x00
[25:24]	BMP	0x0 or 0x1
[23:20]	reserve	0x0
19	CBURSTRW has no effect on synchronous reading.	
[18:16]	CPSIZE	0x0 (Has no effect on asynchronous mode)
15	ASYNCWAIT	0x0
14	EXTMOD	0x0
13	WAITEN is set to 1 if the memory supports this feature, otherwise it remains at 0.	
12	WREN	No impact on synchronous reading
11	Whether WAITCFG is set depends on the memory configuration.	
10	reserve	0x0
9	Whether WAITPOL is set depends on the memory configuration.	
8	BURSTEN	0x1
7	reserve	0x1
6	FACCEN	Set if memory supports it (NOR Flash)
[5:4]	MWID	Configure as needed
[3:2]	MTYP	0x1 or 0x2
1	MUXEN can be configured as needed.	
0	MBKEN	0x1

Table 40-34 R32_FMC_BTRx bit fields

Position	Name	Value to be set
[31:30]	reserve	0x0
[29:28]	ACCMOD	0x0
[27:24]	DATLAT data latency	
[23:20]	CLKDIV	0x0, sets CLK=HCLK (not supported) 0x1, making CLK = 2 * HCLK …
[19:16]	BUSTURN	The time between NEx going high and NEx going low (BUSTURN) HCLK)
[15:8]	DATAST	irrelevant
[7:4]	ADDHLD	irrelevant
[3:0]	ADDSET	irrelevant

Figure 40-20 Waveform of Synchronous Multiplexed Write Mode - PSRAM (CRAM)



1. The memory must issue the NWAIT signal one cycle in advance, and WAITCFG must be 0.

programmed accordingly. Note 2. Byte channel (NBL) outputs are not displayed; they remain low when active. NEx

Table 40-35 R32_FMC_BCRx bit fields

Position	Name	Value to be set
31	FMC_EN	0x1
[30:26]	reserved	0x00
[25:24]	BMP	0x0 or 0x1
[23:20]	reserve	0x0
19	CBURSTRW	0x1
[18:16]	CPSIZE	Configure as needed (CRAM1.5 is 0x1).
15	ASYNCWAIT	0x0
14	EXTMOD	0x0
13	WAITEN	is set to 1 if the memory supports this feature, otherwise it remains at 0.
12	WREN	0x1
11	WAITCFG	0x0
10	reserve	0x0
9	Whether WAITPOL is set depends on the memory configuration.	
8	BURSTEN has no effect on synchronous writes.	
7	reserve	0x1
6	FACCEN is set according to memory support.	

[5:4]	MWID	Configure as needed
[3:2]	MTYP	0x1
1	MUXEN can be configured as needed.	
0	MBKEN	0x1

Table 40-36 R32_FMC_BTRx Bit Fields

	Reserved	Value to be set
Position	bit name	0x0
numbers	ACCMOD	0x0
[31:30] [29:28] [27:24]	DATLAT data latency	
[23:20]	CLKDIV	0x0, sets CLK=HCLK (not supported) 0x1, making CLK = 2 * HCLK
[19:16]	BUSTURN	The time between NEx going high and NEx going low (BUSTURN) HCLK)
[15:8]	DATAST	irrelevant
[7:4]	ADDHLD	irrelevant
[3:0]	ADDSET	irrelevant

40.5.6 NOR/PSRAM Control Register Table

40-37 List of NOR/PSRAM Related Registers

name	Access address		Reset value
R32_FMC_BCR1	Description of SRAM/NOR-Flash Chip Select Control Register 1 (0x40025400):		
R32_FMC_BTR1	0x00003052; SRAM/NOR-Flash Chip Select Timing Register 1 (0x0FFFFFFF); SRAM/		
R32_FMC_BCR2	NOR-Flash Chip Select Control Register 2 (0x000030D2); SRAM/NOR-Flash Chip Select		
R32_FMC_BTR2	Timing Register 2 (0x0FFFFFFF); SRAM/NOR-Flash Chip Select Control Register 3		
R32_FMC_BCR3	(0x000030D2); SRAM/NOR-Flash Chip Select Timing Register 3 (0x0FFFFFFF); SRAM/		
R32_FMC_BTR3	NOR-Flash Chip Select Control Register 4 (0x000030D2). 0x4002541C SRAM/NOR-Flash		
R32_FMC_BCR4	Chip Select Timing Register 4 0x0FFFFFFF 0x40025504 SRAM/NOR-Flash Write Timing		
R32_FMC_BTR4	Register 1 0x0FFFFFFF 0x4002550C SRAM/NOR-Flash Write Timing Register 2		
R32_FMC_BWTR1	0x0FFFFFFF 0x40025514 SRAM/NOR-Flash Write Timing Register 3 0x0FFFFFFF		
R32_FMC_BWTR2	0x4002551C SRAM/NOR-Flash Write Timing Register 4 0x0FFFFFFF		
R32_FMC_BWTR3			
R32_FMC_BWTR4			

40.5.6.1 SRAM/NOR-Flash Chip Select Control Register 1 (R32_FMC_BCR1) Offset

Address: 0x00

31	30	29 28 27 26 25 24 23 22	20	19	18	17 16
FMC_EN	Reserved	BMP	Reserved	CBURSTRW	CPSIZE[2:0]	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ASYNCWAIT	EXTMOD	WAITEN	WREN	WAITCFG	Reserved	WAITPOL	BURSTEN	Reserved	FACCEN	MWID[1:0]		MTYP[1:0]		MUXEN	MBKEN

Bit	name	access	describe	Reset value
31	FMC_EN	RW	NAND/NOR/PSRAM enable bit: 1: Turn on; 0: Off,	0
[30:26] Reserved		RO is reserved.		0
[25:24] BMP		RW address remapping enables the address switching function in Table 40-1.		0
[23:20] Reserved		RO reserved.		0
19 CBURSTRW		RW	Synchronous write enabled: 1: Write in synchronous mode; 0: Always write in asynchronous mode.	0
[18:16] CPSIZE[2:0]		RW	CRAM page size: 000: No burst separation occurs after crossing the page boundary; 001: 128 bytes; 010: 256 bytes; 011: 512 bytes; 100: 1024 bytes; Other: Reserved.	0
15 ASYNCWAIT		RW	Waiting signals during asynchronous transmission: This bit enables/disables the FMC from using a wait signal, even in asynchronous mode. The agreement is also valid during its term. 1. When running asynchronous protocols, consider the NWAIT signal; 0: The NWAIT signal is not considered when running asynchronous protocols.	0
14 EXTMOD		RW	Extended mode enabled: The FMC allows non-multiplexed asynchronous access to the R32_FMC_BWTR register. The write timing is configured, and this configuration is enabled by the EXTEMOD bit. This leads to different timing sequences for read and write operations. 1: Consider the value in the R32_FMC_BWTR register; 0: The value in the R32_FMC_BWTR register is not considered. Note: If extended mode is disabled, FMC can be in either mode or mode. 1 2 The following applies when running: – When selecting the SRAM/PSRAM memory type, the mode. 1. For silence Recognize the mode (MTYP=0x0 or 0x01); – When selecting the memory type, the mode is the default mode. NOR 2 (MTYP=0x10)	0
13 WAITEN		RW	Waiting for enable bit: 1: Enable the NWAIT signal (considering its level; if enabled, in...) (Insert wait period after configured delay period); 0: Disable NWAIT signal (ignoring its level and not configured further). (The wait period is inserted after the Flash delay period).	1
12 WREN		RW	Write enable bit: 1: FMC enables writing within the storage area; 0: FMC prohibits writing to the storage area; if a write operation is performed... An HB error will be reported.	1
11	WAITCFG	RW	Waiting timing configuration: 1: The NWAIT signal is valid during the wait period (not applicable to...).	0

			PSRAM); 0: The NWAIT signal has one data cycle preceding the wait cycle. effect.	
10	Reserved	RO	reserved.	0
9	WAITPOL	RW	Waiting signal polarity bit: 1: NWAIT is active high; 0: NWAIT is active low.	0
8	BURSTEN	RW	Synchronous read enable bit: 1: Enable synchronous read mode. Perform read access in synchronous mode; 0: Synchronous read mode is disabled. Read access is performed in asynchronous mode.	0
7	Reserved	RO	is reserved.	0
6	FACCEN	RW	Flash access enabled: 1: Enable access to the corresponding NOR Flash; 0: Access to the corresponding NOR Flash is prohibited.	1
[5:4]	MWID[1:0]	RW	Memory data bus width: 00: 8 bits; 01: 16 bits; 10: 32 bits; 11: Retained.	01b
[3:2]	MTYP[1:0]	RW	Memory type: 00: SRAM (default value after reset for memory areas 2-4); 01: PSRAM (CRAM); 10: NOR Flash/OneNAND Flash (for storage area 1, (Default value after reset) 11: Retained.	00b
1	MUXEN	RW	Address/data multiplexing enable bit: When this position is 1, the address and data value are multiplexed on the data bus. Only valid for NOR and PSRAM memories. 1: Address/data is multiplexed on the data bus; 0: Address/data is not reused.	1
0	MBKEN	RW	Memory region enable bit: Enable memory region. After reset, memory region 1 is enabled, and other memory regions are disabled. All areas are prohibited. Accessing prohibited storage areas will trigger the HB bus. Error on. 1: Enable the corresponding storage area; 0: The corresponding storage area is disabled.	0

40.5.6.2 SRAM/NOR-Flash Chip Select Control Register x (R32_FMC_BCRx) (x=2/3/4)

Offset address: 0x8*(x–1)

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ASYNC	EXT WAI	WR WAIT	Rese W	AIT BURS	Reserved	MWID[1:0]		MTYP[1:0]		MUX	MBK				

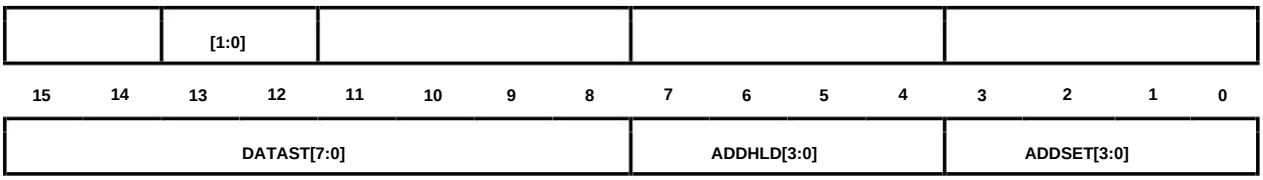
WAIT MOD TEN EN CFG rved POL TEN											EN	EN

Bit	name	access	describe	Reset value
31	Reserved	RW	reserved.	0
[30:26]	Reserved	RO	is reserved.	0
[25:24]	Reserved	RW	reserved.	0
[23:20]	Reserved	RO	reserved.	0
19	CBURSTRW	RW	Synchronous write enabled: 1: Write in synchronous mode; 0: Always write in asynchronous mode.	0
[18:16]	CPSIZE[2:0]	RW	CRAM page size: These bits are used for Cellular RAM 1.5. Sudden access that crosses the address boundaries between pages is not allowed. If By configuring these bits, the FMC controller will... Automatically separate burst access. 000: No burst separation occurs after crossing the page boundary; 001: 128 bytes; 010: 256 bytes; 011: 512 bytes; 100: 1024 bytes; Other: Reserved.	0
15	ASYNCWAIT	RW	Waiting signals during asynchronous transmission: This bit enables/disables the FMC from using a wait signal, even in asynchronous mode. The agreement is also valid during its term. 1. When running asynchronous protocols, consider the NWAIT signal; 0: The NWAIT signal is not considered when running asynchronous protocols.	0
14	EXTMOD	RW	Extended mode enabled: The FMC allows non-multiplexed asynchronous access to the R32_FMC_BWTR register. The write timing is configured, and this configuration is enabled by the EXTEMOD bit. This leads to different timing sequences for read and write operations. 1: Consider the value in the R32_FMC_BWTR register; 0: The value in the R32_FMC_BWTR register is not considered. Note: If extended mode is disabled, FMC can be in either mode or mode. 1 2 The following applies when running: – When selecting the SRAM/PSRAM memory type, the mode1 or silence Recognize the mode (MTYP=0x0 or 0x01); – When selecting the memory type, the mode is the default mode. NOR 2 (MTYP=0x10)	0
13	WAITEN	RW	Waiting for enable bit: 1: Enable the NWAIT signal (considering its level; if enabled, in...) (Insert wait period after configured delay period); 0: Disable NWAIT signal (ignoring its level and not configured further). (The wait period is inserted after the Flash delay period).	1
12	WREN	RW	Write Enable Bit:	1

			1: FMC enables writing within the storage area; 0: FMC prohibits writing to the storage area; if a write operation is performed... An HB error will be reported.	
11	WAITCFG	RW	Waiting timing configuration: 1: The NWAIT signal is valid during the wait period (not applicable to...). PSRAM); 0: The NWAIT signal has one data cycle preceding the wait cycle. effect.	0
10	Reserved	RO	reserved.	0
9	WAITPOL	RW	Waiting signal polarity bit: 1: NWAIT is active high; 0: NWAIT is active low.	0
8	BURSTEN	RW	Synchronous read enable bit: 1: Enable synchronous read mode. Perform read access in synchronous mode; 0: Synchronous read mode is disabled. Read access is performed in asynchronous mode.	0
7	Reserved	RO	is reserved.	1
6	Reserved	RW	reserved.	1
[5:4]	MWID[1:0]	RW	Memory data bus width: 00: 8 bits; 01: 16 bits; 10: 32 bits; 11: Retained.	01b
[3:2]	MTYP[1:0]	RW	Memory type: 00: SRAM (default value after reset for memory regions 2...4); 01: PSRAM (CRAM); 10: NOR Flash/OneNAND Flash (for storage area 1, (Default value after reset) 11: Retained.	0
1	MUXEN	RW	Address/data multiplexing enable bit: When this position is 1, the address and data value are multiplexed on the data bus. Only valid for NOR and PSRAM memories. 1: Address/data is multiplexed on the data bus; 0: Address/data is not reused.	1
0	MBKEN	RW	Memory region enable bit: Enable memory region. After reset, memory region 1 is enabled, and other memory regions are disabled. All areas are prohibited. Accessing prohibited storage areas will trigger the HB bus. Error on. 1: Enable the corresponding storage area; 0: The corresponding storage area is disabled.	0

40.5.6.3 SRAM/NOR-Flash Chip Select Timing Register x (R32_FMC_BTRx) (x=1/2/3/4) Offset Address: 0x04 + 0x8*(x–1)

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
Reserved	ACCMOD	DATLAT[3:0]	CLKDIV[3:0]	BUSTURN [3:0]	



Bit	name	access	describe	Reset value
[31:30]	Reserved	RO	reserved.	0
[29:28]	ACCMOD[1:0]	RW	Access mode: Specify the asynchronous access mode,as shown in the sequence diagram. Only when When the EXTEMOD bit in the R32_FMC_BCRx register is set to 1, these Only when the position is effective. 00: Access mode A; 01: Access Mode B; 10: Access Mode C; 11: Access mode D.	0
[27:24]	DATLAT[3:0]	RW	Data latency of synchronous memory: For synchronous access with read/write burst mode enabled (BURSTEN/ CBURSTRW position 1), this field defines the period before the first data is read or written. Number of memory clock cycles to send to memory (+2): This timing parameter is expressed in FMC_CLK period rather than HCLK period. This value is unrelated to asynchronous access mode. 0000: Data over 2 CLK clock cycles during the first burst access. Delay; 1111: During the first burst access, the number of 17 CLK clock cycles... According to reports, there has been a delay.	1111b
[23:20]	CLKDIV[3:0]	RW	The clock division ratio of the FMC_CLK signal: 0000: Reserved; 0001: FMC_CLK period = 2 * HCLK period; 0010: FMC_CLK period = 3 * HCLK period; 1111: FMC_CLK period = 16 * HCLK period. In asynchronous NOR Flash, SRAM, or PSRAM access modes, This value is irrelevant. Note: For information on the FMC_CLK division ratio formula, please refer to section [section number missing]. Section 40.5.5: Synchronizing Transactions.	1111b
[19:16]	BUSTURN[3:0]	RW	Duration of the bus turnaround phase: Writing these bits via software can be used to write to the end of a write-read (and read-write) transaction. Add a delay when the bundle is finished. Bus turnaround delays will also be inserted into two processes performed on different memory areas. Between two consecutive read/write transactions. When moving to the same FMC storage region, regardless of the configured BUSTURN Regardless of the timing, only one HCLK clock cycle will be inserted. When two When consecutive write transactions point to the same FMC storage area, no This will introduce bus turnaround delay. Bus turnaround latency can match the shortest time between consecutive transactions. (tEHEL changes from NEx high level to NEx low level) and memory	1111b

			The longest time required to release the data bus after a read access. (tEHQZ): (BUSTRUN+1)HCLK period ÿ tEHELmin, (BUSTRUN+2)HCLK period ÿ tEHQZmax (if EXTEMOD=0) (BUSTRUN+2)HCLK period ÿ max(tEHELmin, tEHQZmax) (if EXTMOD=1). 0000: Duration of the BUSTURN phase = 0 HCLK clock cycles increment; ... 1111: Duration of the BUSTURN phase = 15 * HCLK clock cycles.	
[15:8] DATAST[7:0]		RW	Duration of the data phase: The duration of the data phase can be defined by writing these bits in software (see Figures 40-3 to 40-15), applicable to asynchronous access mode: 0000 0000: Reserved; 0000 0001: Duration of the DATAST phase = 1 * HCLK clock cycle; 0000 0010: Duration of the DATAST phase = 2 * HCLK clock cycles; ... 1111 1111: The duration of the DATAST phase = 255 * HCLK clock cycles. For information on the duration of the data phase for each memory type and access mode, please refer to Figures 40-3 to 40-15. For example: In Mode 1, write access, and with DATAST=1, the duration of the data phase = DATAST+1 = 2 HCLK clock cycles. Note: This value is irrelevant in synchronous access mode.	1111 1111b
[7:4] ADDHLD[3:0]		RW	Duration of the address holding phase: The duration of the address holding phase can be defined by writing these bits in software (see Figures 40-3 to 40-15), applicable to mode D or multiplexed access: 0000: Reserved; 0001: Duration of the ADDHLD phase = 1 * HCLK clock cycle; 0010: Duration of the ADDHLD phase = 2 * HCLK clock cycles; ... 1111: Duration of the ADDHLD phase = 15 * HCLK clock cycles. For information on the duration of the address hold phase for each access mode, please refer to the corresponding figures (Figures 40-3 to 40-15). Note: This value is not used in synchronous access mode because the address hold always lasts for one memory clock cycle [unspecified].	1111b
[3:0] ADDSET[3:0]		RW	Duration of the address setup phase: The duration of the address setup phase can be defined by writing these bits in software (see Figures 40-3 to 40-15), applicable to SRAM,	1111b

			ROM, asynchronous NOR Flash, and PSRAM: 0000: Duration of the ADDSET phase = 0 * HCLK clock cycles; ... 1111: Duration of the ADDSET phase = 15 * HCLK clock cycles; Information regarding the duration of the address setup phase for each access mode. For more information, please refer to the corresponding images (Figures 40-3 to 40-15). Note: This value is irrelevant in synchronous access mode. The minimum value of ADDSET is D 1. Below, the minimum value of ADDSET is	

Note: PSRAM (CRAM) has an uncertain data latency due to internal refresh. Therefore, these memories will have a latency across the entire latency period.

Segment transmissionThe NWAIT signal is used to extend the delay as needed. For PSRAM (CRAM), the DATLAT field must be set to 0.

FMC It will immediately exit the delay phase, begin sampling NWAIT in memory, and then begin reading or writing once the memory is ready.

This method also applies to the latest generation of synchronous Flash, which differs from earlier Flash in that it sends a NWAIT signal (detection)

Check the specific Flash data table used.

40.5.6.4 SRAM/NOR-Flash Write Timing Register x (R32_FMC_BWTRx) (x=1/2/3/4) Offset Address: 0x104 + 0x8*(x–1)

31 30	29 28 27 26 25 24 23 22 21	2019	18 17 16
Reserved	ACCMOD [1:0]	Reserved	BUSTURN [3:0]

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

Bit	name	access	describe	Reset value
[31:30]	Reserved	RO	reserved.	0
[29:28]	ACCMOD[1:0]	RW	Access mode: Specify the asynchronous access mode, as shown in the following sequence diagram. Only when When the EXTEMOD bit in the R32_FMC_BCRx register is set to 1, these Only when the position is effective. 00: Access mode A; 01: Access Mode B; 10: Access Mode C; 11: Access mode D.	0
[27:20]	Reserved	RO	is reserved.	0
[19:16]	BUSTURN[3:0]	RW	Duration of the bus turnaround phase: These bits can be added at the end of a write-read transaction by writing them in the software. A delay, thus matching the minimum time between two consecutive transactions. (between ENx high level and ENx low level): (BUSTRUN+1)HCLK period ÿ tEHELmin. 0000: Duration of the BUSTURN phase = when increasing by 0 HCLKs Clock cycle; ... 1111: Duration of the BUSTURN phase = Increase by 15 HCLK Clock cycle.	1111b

[15:8] DATAST[7:0]		RW	Duration of the data phase: The duration of these bit-defined data phases can be determined by writing them into the software. (Please refer to Figures 40-3 to 40-15), applicable to asynchronous SRAM, PSRAM and NOR Flash access modes: 0000 0000: Duration of the DATAST phase = 1 * HCLK clock. cycle; 0000 0001: Duration of the DATAST phase = 2 * HCLK clock. cycle; 0000 0010: Duration of the DATAST phase = 3 * HCLK clock. cycle; ... 1111 1111: Duration of the DATAST phase = 256 * HCLK hours Clock cycle.	1111 1111b
[7:4] ADDHLD[3:0]		Duration of the RW address hold phase: The duration of the address holding phase can be defined by writing these bits in the software. Time (please refer to Figures 40-3 to 40-15), applicable to asynchronous complex Access: 0000: Reserved; 0001: Duration of the ADDHLD phase = 1 * HCLK clock cycle; 0010: Duration of the ADDHLD phase = 2 * HCLK clock cycles; ... 1111: Duration of the ADDHLD phase = 15 * HCLK clock cycles. Note: During synchronization NOR This value is not used in Flash access mode. Because the duration of the address holding phase is always 1 Flash memory. Clock cycle.		1111b
[3:0] ADDSET[3:0]		RW	Duration of the address establishment phase: These bits can be written by software to define the ground plane in HCLK periodicity. The duration of the site establishment phase (please refer to Figures 40-3 to 40-15). Suitable for asynchronous access mode: 0000: Duration of the ADDSET phase = 0 * HCLK clock cycles; ... 1111: Duration of the ADDSET phase = 15 * HCLK clock cycles. Note: This value is not used in synchronous access mode because the duration of the address establishment phase is always [unspecified]. 1 One Flash clock cycle. In reuse mode, the minimum value of ADDSETis 1.	1111b

40.6 NAND Flash Controller

FMC generates the corresponding signal timings to drive the following types of devices: 8-bit and 16-bit NAND Flash.

NAND storage regions are configured via dedicated registers (Section 40.6.7). Programmable memory parameters include access timing (such as...).

(See the table below) and ECC configuration.

Table 40-36 Programmable NAND Flash Access Parameters

parameter		Access mode	unit	minimum value	maximum value
Memory setup time	Function command enable address setup Number of clock cycles required (HCLK)	Reading/Writing	HB clock cycle (HCLK)	1	255

Memory wait	Minimum duration for command enable (Based on HCLK clock cycles)	Reading/Writing	HB clock cycle (HCLK)	2	255
Memory retention	After the command is disabled, the address must be retained. (If write access was performed, the number must still be maintained) The number of clock cycles (HCLK) of the data	Reading/Writing	HB clock cycle (HCLK)	1	254
Memory data bus High resistance state	After write access begins, the data bus Clock during high impedance state Number of cycles (HCLK)	Write	HB clock cycle (HCLK)	1	255

40.6.1 External Memory Interface

Signals 8-bit NAND Flash

Table 40-37 8-bit NAND Flash

FMC signal name I/O		Function
A[17]	O	NAND Flash Address Latch Enable (ALE) signal
A[16]	O	NAND Flash Command Latch Enable (CLE) signal
D[7:0]	I/O	8-bit multiplexed bidirectional address/data bus
NCE	O	Film Selection
NOE (=NRE)	O	Output Enable (Memory Signal Name: Read Enable, NRE)
NWE		Write Enable
NWAIT/INT	I	input FMC NAND Flash ready/busy signal

Since FMC can manage a sufficient number of address cycles, there is theoretically no capacity limit.

Note: The prefix "N" indicates a signal that is active low.

16-bit NAND Flash

Table 40-38 16-bit NAND Flash

FMC signal name I/O		Function
A[17]	O	NAND Flash Address Latch Enable (ALE) signal
A[16]	O	NAND Flash Command Latch Enable (CLE) signal
D[15:0]	I/O	16-bit multiplexed bidirectional address/data bus
NCE	O	Film Selection
NOE (=NRE)	O	Output Enable (Memory Signal Name: Read Enable, NRE)
NWE		Write Enable
NWAIT/INT	I	input FMC NAND Flash ready/busy signal

Since FMC can manage a sufficient number of address cycles, there is theoretically no capacity limit.

Note: The prefix "N" indicates a signal that is active low.

40.6.2 NAND Flash Supported Memory and Transactions

The table below describes the supported devices, access modes, and transactions. Transactions that the NAND Flash controller does not allow (or does not support) are listed in gray.

Color display.

Table 40-39 Supported Memory and Transactions

Device Mode	R/W		HB Data size	memory Data size	Allowed/Not Allowed	Notes
8-bit NAND asynchronous	R		8	8	yes	-

	Asynchronous W	8	8	-
	Asynchronous R	16	8	Yes, it is divided into two FMC visits.
	Asynchronous W	16	8	It is divided into 2 FMC visits
	Asynchronous R	32	8	It is divided into 4 FMC visits
	Asynchronous W	32	8	It is divided into 4 FMC visits
16-bit NAND	Asynchronous R	8	16	-
	Asynchronous W	8	16	Yes/No - Yes/
	Asynchronous R	16	16	-
	Asynchronous W	16	16	-
	Asynchronous R	32	16	No (FMC access divided into 2 steps)
	Asynchronous W Asynchronous R Asynchronous W	32	16	It is divided into 2 FMC visits

40.6.3 NAND Flash Timing Diagram The

NAND Flash storage area is managed through the following set of registers:

• Control Register: FMC_PCR •

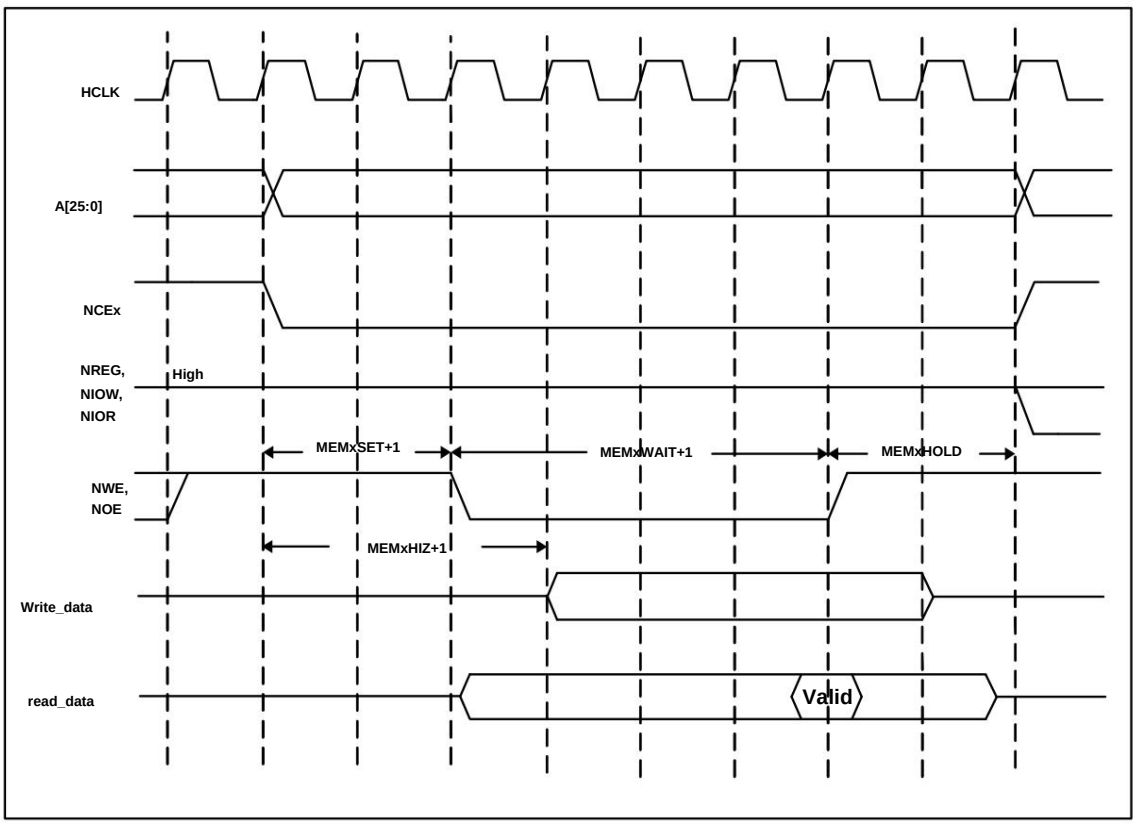
Interrupt Status Register: FMC_SR

• ECC Register: FMC_ECCR •

Timing Register for General Memory Space: R32_FMC_PMEM

• Timing Register for Specialized Memory Space:

R32_FMC_PATT Figure 40-21 General Memory Access Waveform of NAND Flash Controller



Note: Keep high during write access (invalid). Keep high during read access (invalid).NOE,NWE

Each timing configuration register contains four parameters, three of which define the HCLK for the three access phases of the NAND Flash.

The number of cycles, another parameter, defines the timing at which the data bus is started to be driven when a write access occurs. The diagram above illustrates general-purpose memory access.

The timing parameter definition and the access timing of the characteristic memory space are similar.

40.6.4 NAND Flash Operation

The Command Latch Enable (CLE) and Address Latch Enable (ALE) signals of a NAND Flash device are driven by the address signals of the FMC controller. This means that to send a command or address to the NAND Flash, the CPU must perform a write operation at a specific address in its

memory space. A typical page read operation from a NAND Flash device requires the

following steps: 1) Configure the FMC_PCR and FMC_PMEM registers (for some devices, configure R32_FMC_PATT; see Section 40.6.5: NAND Flash Pre-Wait Functions) according to the characteristics of the NAND Flash (PWID bit indicates the NAND Flash data bus width, PTYP=1, PWAITEN=0 or 1 as needed; for timing configuration, see Section 40.4.2: NAND Flash Address Mapping), thereby configuring and enabling the corresponding memory regions.

2) The CPU performs a byte write operation to the general-purpose memory space. At this time, the data byte is equal to a Flash command byte (e.g., byte 0x00 for Samsung NAND Flash devices). During the write strobe (a low-level pulse on NWE), the LE input of the NAND Flash is valid, so the written byte is regarded as a command of the NAND Flash. After the command is latched by the memory device, it does not need to be written again when a page read operation is performed

subsequently. 3) By writing STARTAD[7:0], STARTAD[16:9], STARTAD[24:17] and ... to the general-purpose memory space or the feature space... The CPU can send the start address (STARTAD) of a read operation using the 4 bytes STARTAD[25] (3 bytes for smaller capacity devices). During the write strobe (a low pulse on NWE), the ALE input of the NAND Flash device is active, so the written byte is treated as the start address of the read operation. With the help of the special memory space, some of the pre-wait functions required by NAND Flash can be implemented using another different timing configuration of the FMC (see Section 40.6.5: NAND Flash Pre-Wait Functions for details).

4) The controller waits for the NAND Flash to be ready (R/NB signal) before making a new access to the same or another storage area. (At high level). During the wait period, the controller keeps the NCE signal active (low level).

5) Then the CPU can perform byte read operations from general-purpose memory space, thereby reading NAND Flash pages (data fields) byte by byte. +Spare field). 6) The

next NAND Flash page can be read without any CPU command or address write operation. This can be achieved in three different ways: – Perform

the operation described in step 5 –

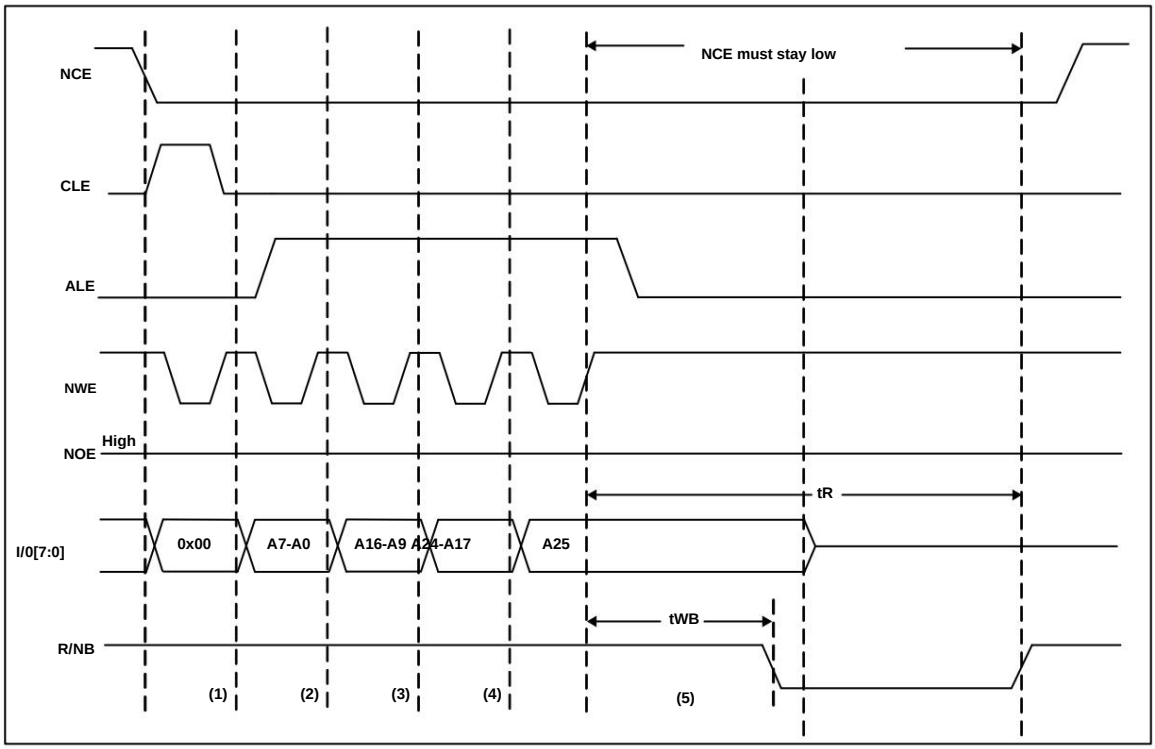
Randomly access a new address by restarting the operation in step 3 – Send

a new command to the NAND Flash device by restarting step 2

40.6.5 NAND Flash Wait-Ahead Function

Some NAND Flash devices require the controller to wait for the R/NB signal to go low after writing the last part of the address, as shown in the figure below.

Figure 40-22 Accessing non-"CE-independent" NAND-Flash



1) The CPU writes byte 0x00 at address 0x70010000.

2) The CPU writes bytes A7 to A0 at address 0x70020000.

3) The CPU writes bytes A16 to A9 at address 0x70020000.

4) The CPU writes bytes A24 to A17 at address 0x70020000.

5) The CPU writes byte A25 at address 0x78020000: The FMC performs a write access via the R32_FMC_PATT2 timing definition, where...

ATTHOLD ÿ 7 (assuming (7+1)*HCLK = 112ns > tWB maximum value). This ensures that NCE remains low until R/NB changes again.

It changes from a low level to a high level (this function is only required for NAND Flash where the NCE signal has an effect).

When this function is required, the MEMHOLD value can be programmed to ensure that the tWB timing is met. However, the CPU performs all operations on the NAND Flash...

Both read and write accesses will have a hold delay of (MEMHOLD+1) HCLK cycles (this delay is inserted at the rising and falling edges of the NWE signal).

(between visits).

To overcome this timing limitation, a special memory space can be used, and its timing registers can be programmed with ATTHOLD values that satisfy the tWB timing.

And keep MEMHOLD at its minimum value. Then, the CPU must use general-purpose memory space for all NAND Flash reads and writes.

Except when accessing the NAND Flash device and writing the last address byte, in which case the CPU must perform a write operation on the special memory space.

do.

40.6.6 Error Correction Code (ECC) Calculation (NAND Flash)

The FMC NAND card controller includes two error correction code calculation hardware modules, one for each storage region. These modules can be configured in software.

Reduce host CPU workload when managing ECC.

These two ECC modules are identical, associated with storage area 2 and storage area 3 respectively. Therefore, hardware ECC computation is not applicable to either.

The memory connected to storage area 4.

When performing read or write operations on NAND Flash, the corresponding values are 256, 512, 1024, 2048, 4096, or 8192 bytes.

The ECC algorithm used in FMC can correct 1-bit errors and detect 2-bit errors. This operation is based on the Hamming coding algorithm and includes...

Includes calculating row and column parity checks.

Whenever the NAND Flash storage area is active, the ECC module monitors the NAND Flash data bus and read/write signals.

Numbers (NCE and NWE).

Operate ECC as follows:

When accessing NAND Flash memory area 2 or memory area 3, the data appearing on the D[15:0] bus will be latched and used for...

ECC calculation.

When any other address in the NAND Flash is accessed, the ECC logic enters an idle state and performs no operation. Therefore, ...

Write operations used to define NAND Flash commands or addresses are invalid during ECC calculations.

After the host CPU completes the required number of bytes read/write operations on the NAND Flash, it must read the R32_FMC_ECCR register.

Only then can the calculated value be retrieved. After reading, these registers should be cleared by resetting the ECCEN bit to zero. To calculate a new data block,

The ECCEN bit in the FMC_PCR register must be set to 1.

To perform ECC calculations:

1) Enable the ECCEN bit in the R32_FMC_PCR register.

2) Write data to the NAND Flash page. During the NAND page writing process, the ECC module will calculate the ECC value.

3) Read the ECC value provided in the R32_FMC_ECCR register and store it in a variable.

4) Clear the ECCEN bit in the R32_FMC_PCR register and then enable it, then read back the data written from the NAND page. During the read operation...

During NAND page operations, the ECC module will calculate the ECC value.

5) Read the new ECC value provided in the R32_FMC_ECCR register.

6) If the ECC values read twice are the same, no correction is needed; otherwise, an ECC error exists, and the software correction routine will return.

Return information regarding whether the error can be corrected.

40.6.7 NAND Flash Control Register

Table 40-40 NAND Flash Related Register List

name	Access address	describe	Reset value
R32_FMC_PCR	0x40025480	NAND Flash Control Register;	0x00000018
R32_FMC_SR	0x40025484	FIFO Status and Interrupt Register;	0x00000040
R32_FMC_PMEM	0x40025488	General Purpose Memory Space Timing Register;	0xFCFCFCFC;
R32_FMC_PATT	0x4002548C	Special Purpose Memory Space Timing Register;	0xFCFCFCFC;
R32_FMC_ECCR	0x40025494	ECC Result Register	0x00000000

40.6.7.1 NAND Flash Control Register (R32_FMC_PCR) Offset

Address: 0x80

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
Reserved				ECCPS[2:0]	TAR[3]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAR[2:0]			TCLR[3:0]				Reserved		ECCEN	PWID[1:0]		PTYP	PBKEN	PWAITEN	Reserved

Bit	name	access	describe	Reset value
[31:20]	Reserved	RO	is reserved.	0
[19:17]	ECCPS[2:0]	RW	ECC page size: 000: 256 bytes; 001: 512 bytes; 010: 1024 bytes; 011: 2048 bytes; 100: 4096 bytes; 101:8192 bytes.	0

[16:13]	TAR[3:0]	RW	Delay from ALE to RE: Set the time from ALE low to RE low using HB clock cycles (HCLK): t_ar = (TAR + SET + 2) * THCLK, where THCLK is the HCLK clock cycle. 0000: 1 HCLK cycle (default); 1111: 16 HCLK cycles. Note: Depending on the address space, SET can be MEMSET or ATTSET.	0
[12:9]	TCLR[3:0]	RW	Delay from CLE to RE: The time from CLE low to RE low is set using HB clock cycles (HCLK): t_clr = (TCLR + SET + 2) * THCLK. THCLK is the HCLK clock cycle. 0000: 1 HCLK cycle (default); 1111: 16 HCLK cycles. Note: Depending on the address space, SET can be MEMSET or ATTSET.	0
[8:7]	Reserved	RO	is reserved.	0
6	ECCEN	RW	ECC calculation logic enable bit: 1: Enable ECC logic; 0: Disables and resets ECC logic (default value after reset).	0
[5:4]	PWID[1:0]	RW	Data bus width: 00: 8 bits; 01: 16 bits; 10: Retained; 11: Retained.	01b
3	PTYP	RW	Memory type: Define the type of device attached to the corresponding storage area: 1: NAND Flash (default value after reset); 0: Reserved; the reset value must be retained.	1
2	PBKEN	RW	NAND Flash storage area enable bit: Enable storage regions. Accessing prohibited storage regions will trigger an HB total. An online error. 1: Enable the corresponding storage area; 0: Disable the corresponding storage area (default value after reset).	0
1	PWAITEN	RW	Waiting feature enable bit: 1: Enable; 0: Disabled.	0
0	Reserved	RO	is reserved.	0

40.6.7.2 FIFO Status and Interrupt Register (R32_FMC_SR) Offset

Address: 0x84

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									FEMPT	FEN	ILEN	IREN	FS	ILS	IRS

Bit	name	access	describe	Reset value
[31:7] Reserved		RO is reserved.		0
6	FEMPT	RO	FIFO is empty: 1: FIFO is empty; 0: FIFO is not empty.	1
5	IFEN	RW	Interrupt falling edge detection enable bit: 1: Enable interrupt falling edge detection request; 0: Interrupt falling edge detection request disabled.	0
4	ILEN	RW	Interrupt high-level detection enable bit: 1: Enable interrupt high-level detection request; 0: Interrupt high-level detection request disabled.	0
3	IREN	RW	Interrupt rising edge detection enable bit: 1: Enable interrupt rising edge detection request; 0: Disallow rising edge detection requests.	0
2	IFS	RW0	Interrupt falling edge status: This flag is set to 1 by hardware, reset by software, and cleared by writing 0. 1: A falling edge interruption occurs; 0: No interrupted falling edge occurred.	0
1	ILS	RW0	Interrupt high level state: This flag is set to 1 by hardware, reset by software, and cleared by writing 0. 1: An interrupt high level occurs; 0: No interrupt high level occurred.	0
0	IRS	RW0	Interrupt rising edge status: This flag is set to 1 by hardware, reset by software, and cleared by writing 0. 1: An interrupted rising edge occurs; 0: No interrupted rising edge occurred.	0

40.6.7.3 General Purpose Memory Space Timing Register (R32_FMC_PMEM)

Offset address: 0x88

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
MEMHIZ[7:0]			MEMHOLD[7:0]		

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

Bit	name	access		Reset value
[31:24] MEMHIZ[7:0]		RW	Description of the high-impedance state time of the general-purpose memory data bus: 00000000: 1 HCLK cycle; 11111110: 255 HCLK cycles; 11111111: Reserved.	1111 1100b
[23:16] MEMHOLD[7:0]		RW	General memory retention time: 00000000: Reserved; 00000001: 1 HCLK cycle; 11111110: 254 HCLK cycles;	1111 1100b

			11111111: Reserved.	
[15:8]	MEMWAIT[7:0]	RW	General-purpose memory latency: 00000000: Reserved; 00000001: 2 HCLK cycles (+ introduced when NWAIT is disabled) Waiting period); 11111110: 255 HCLK cycles (introduced when NWAIT is disabled) (waiting period); 11111111: Reserved.	1111 1100b
[7:0]	MEMSET[7:0]	RW	General purpose memory setup time: 00000000: 1 HCLK cycle; 11111110: 255 HCLK cycles; 11111111: Reserved.	1111 1100b

40.6.7.4 Feature Memory Space Timing Register (R32_FMC_PATT)

Offset address: 0x8C

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

Bit	name	access		Reset value
[31:24]	ATTHIZ[7:0]	RW	Description of the high-impedance state time of the memory data bus: 00000000: 0 HCLK cycles; 11111110: 255 HCLK cycles; 11111111: Reserved.	1111 1100b
[23:16]	ATTHOLD[7:0]	RW	Feature memory retention time: 00000000: Reserved; 00000001: 1 HCLK cycle; 11111110: 254 HCLK cycles; 11111111: Reserved.	1111 1100b
[15:8]	ATTWAIT[7:0]	RW	Feature memory latency: 00000000: Reserved; 00000001: 2 HCLK cycles (+ introduced when NWAIT is disabled) Waiting period); 11111110: 255 HCLK cycles (introduced when NWAIT is disabled) (waiting period); 11111111: Reserved.	1111 1100b
[7:0]	ATTSET[7:0]	RW	Feature memory setup time: 00000000: 1 HCLK cycle; 11111110: 255 HCLK cycles; 11111111: Reserved.	1111 1100b

40.6.7.5 ECC Result Register (R32_FMC_ECCR)

Offset address: 0x94

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
ECC														

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

Bit	name	access	describe	Reset value
[31:0] ECC		RO	ECC Results: This field contains the value calculated by the ECC calculation logic, as follows: As shown in the table.	0

ECCPS[2:0]	Page size in bytes	(ECC bits)
000	256	ECC[21:0]
001	512	ECC[23:0]
010	1024	ECC[25:0]
011	2048	ECC[27:0]
100	4096	ECC[29:0]
101	8192	ECC[31:0]

40.7 SDRAM Controller

40.7.1 Key Characteristics of SDRAM Controller The key

characteristics of the SDRAM controller are as follows:

Supports word, half-word, and byte access.

Power -down mode

ÿ CAS latency 1, 2, 3 ÿ Self-

refresh mode

Programmable timing parameters

SDRAM power-on initialization via software

Two SDRAM memory regions, which can be configured independently.

Multi -storage area ping-pong access

The SDRAM clock can be HCLK, HCLK/2, or HCLK/3 . It supports

automatic refresh and a programmable refresh rate.

ÿ Data bus width for 8-bit, 16-bit, and 32-bit memory

ÿAutomatic management of storage area boundaries

The FIFO can be cached and supports 6 rows * 32 bits depth (6 * 14 bits address tag).

ÿ 13-bit address row, 11-bit address column, 4 internal storage areas: 4*16M*32bit (256MB), 4*16M*16bit

(128MB), 4*16M*8bit (64MB)

40.7.2 SDRAM External Memory Interface Signals

During startup, the SDRAM I/O used to connect the FMC SDRAM controller to the external SDRAM device must be accessed via the user application.

The pins are configured. SDRAM controller I/O pins not used by the application can be used for other purposes.

Table 40-41 SDRAM Signals

SDRAM signals	I/O Type	illustrate	Reuse function

SDCLK	O	SDRAM clock	-
SDCKE[1:0]	O	SDCKE0: SDRAM memory area 1 clock enable SDCKE1: SDRAM memory area 2 clock enable	-
SDNE[1:0]	O	SDNE0: SDRAM memory area 1 chip enable SDNE1: SDRAM Memory Area 2 Chip Enable	-
A[12:0]	O Address		FMC_A[12:0]
D[31:0]	I/O bidirectional data bus		FMC_D[31:0]
BA[1:0]	O Storage area address		FMC_A[15:14]
NRAS	O line address strobe		-
NCAS	O Column Address Selection		-
SDNWE	O Write enable write		-
DQM[3:0]	O	access output byte mask (Memory signal name: DQM[3:0])	-

40.7.3 SDRAM Controller Function Description

All SDRAM controller outputs (signals, address, and data) can be configured to change on the falling edge of the memory clock (FMC_SDCLK). change.

SDRAM

initialization sequence is managed by software. If two memory regions are used, the R32_FMC_SDCMR register must be initialized. Set the target storage region bits CTB1 and CTB2 to 1, and simultaneously generate initialization sequences for storage region 1 and storage region 2:

1) Program the characteristics of the memory device into the R32_FMC_SDCRx register. This includes SDRAM clock frequency, RBURST, and RPIPE characteristics. The property must be programmed into the R32_FMC_SDCR1 register.

2) Program the memory device timings into the R32_FMC_SDTRx register. The TRP and TRC timings must be programmed into the register. In the R32_FMC_SDTR1 register.

3) Set the MODE[2:0] bits in the R32_FMC_SDCMR register to “001” and configure the destination in the R32_FMC_SDCMR register. Mark the memory region bits (CTB1 and/or CTB2) to begin providing clock signals to the memory (SDCKE drive is high).

4) Wait for the specified delay period. The typical delay is 100us.

5) Set the MODE[2:0] bits in the R32_FMC_SDCMR register to “010” and configure the destination in the R32_FMC_SDCMR register. Mark the memory region bits (CTB1 and/or CTB2) to send a "full precharge" command.

6) Set the MODE[2:0] bits in the R32_FMC_SDCMR register to “011” and configure the destination in the R32_FMC_SDCMR register. The number of storage locations (CTB1 and/or CTB2) and continuous automatic refresh commands (NRFS).

7) Based on the SDRAM device configuration MRD field, set the MODE[2:0] position in the R32_FMC_SDCMR register to "100" and configure... Set the target memory region bits (CTB1 and/or CTB2) in the R32_FMC_SDCMR register to send the "Load Mode Register" command and Programming SDRAM devices. Specifically:

a) The CAS delay must be selected based on the value in R32_FMC_SDCR1/2;

b) The burst length (BL) must be set to 1 by setting the MRD[2:0] bits in the R32_FMC_SDCMR register to 000.

If the mode registers of the two SDRAM memory regions are different, this step must be repeated twice, once for each memory region and to the end. The corresponding storage area bit is set to 1.

8) Program the refresh rate in the R32_FMC_SDRTR register. The refresh rate corresponds to the delay between refresh cycles. Its value must be... Compatible with SDRAM devices.

At this stage, the SDRAM device is ready to accept commands. If a system reset occurs during SDRAM access, then... The data bus may still be driven by the SDRAM device. Therefore, the SDRAM device, NOR Flash/PSRAM/SRAM must be reinitialized after a reset. Only the NAND Flash controller can send a new access command.

Note: If two SDRAM devices are connected to the FMC, the command mode register can be accessed simultaneously for both devices (load mode register). In the case of a device command, the SDRAM storage area will be determined according to the R32_FMC_SDTR1 register configured timing parameters (TMRD TRAS

(Timing) Issue access commands.

SDRAM controller write cycle

The SDRAM controller can accept both single and burst write requests and treat them as single memory accesses. In both cases, SDRAM...
The controller tracks valid rows in each storage region to enable continuous write access to different storage regions (multi-storage-region ping-pong).
Before performing any write access, the WP bit in the R32_FMC_SDCRx register must be cleared to disable the SDRAM memory area.

Write protection.

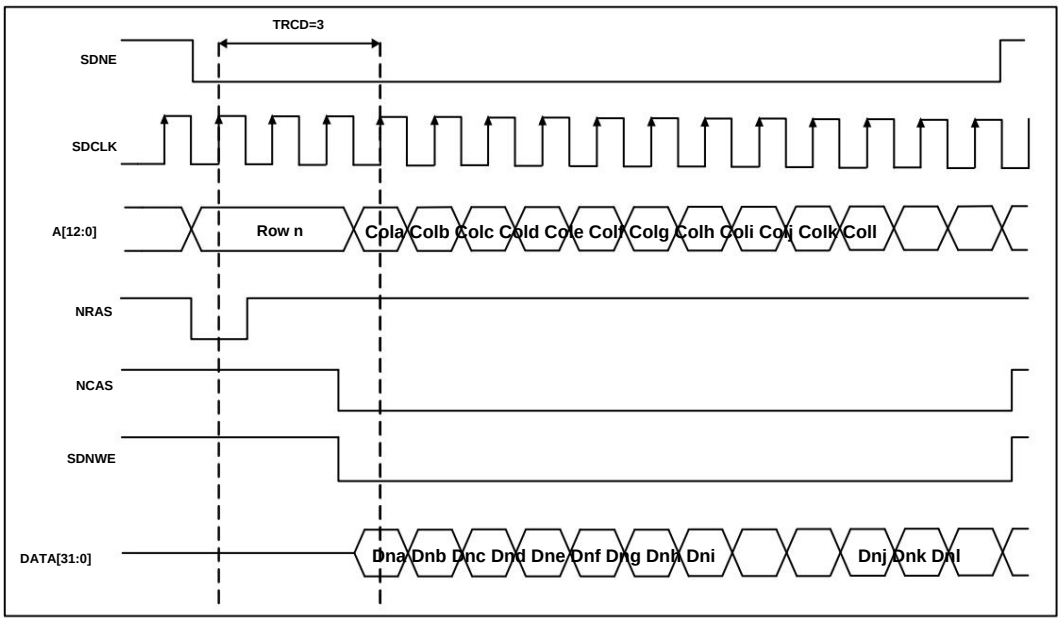
The SDRAM controller always checks for the next access:

1) If the next access occurs on the same line or on another valid line, the write operation is performed directly.

2) If the next access points to an invalid row, the SDRAM controller will generate a precharge command, activate the new row, and initialize it.

Write the command.

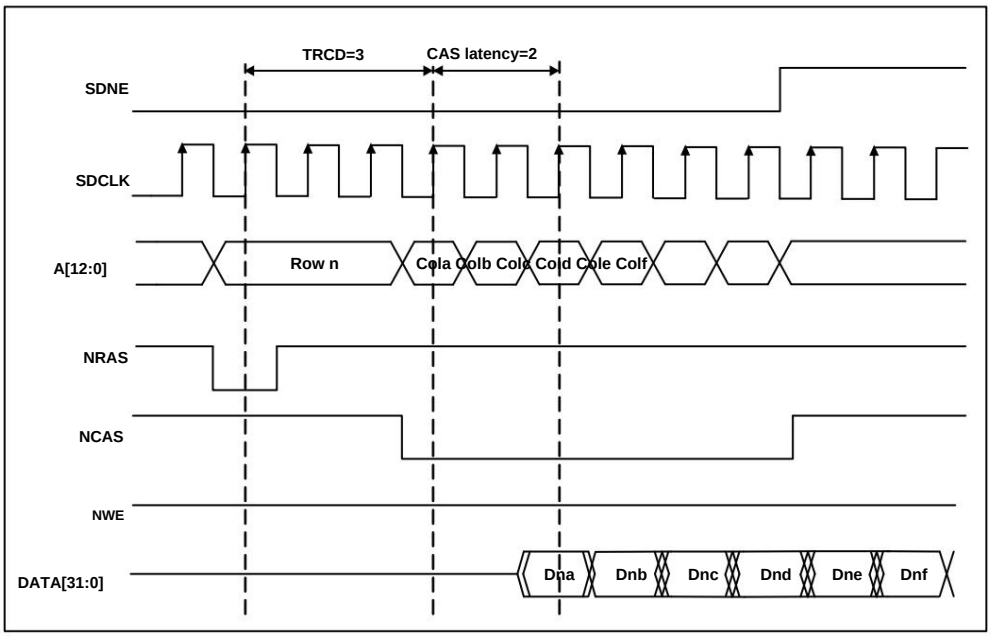
Figure 40-23 Burst Write to SDRAM Access Waveform



SDRAM controller read cycle

The SDRAM controller can accept both single and burst read requests and treat them as single memory accesses. In both cases, SDRAM...
The controller tracks valid rows in each storage region to enable continuous read access to different storage regions (multi-storage-region ping-pong).
access).

Figure 40-24 Burst SDRAM Read Access



The FMC SDRAM controller features a cacheable read FIFO (6 rows * 32 bits), which is used to calculate the CAS latency period according to the following formula and Data to be read is stored in advance during the RPIPE delay. The RBURST bit in the R32_FMC_SDCR1 register must be set to 1 for the data to be accepted.

One read access. Expected data volume = CAS latency + 1 + (RPIPE latency) / 2, example:

ÿ CAS latency = 3, RPIPE latency = 0: Four data (uncommitted) are stored in FIFO.

ÿ CAS latency = 3, RPIPE latency = 2: Five data (uncommitted) are stored in FIFO.

Each line read from the FIFO has a 14-bit address marker to identify its own content: 11 bits for the column address and 2 bits for the select. Select the internal storage region and valid row. One bit is used to select the SDRAM device. During an HB burst read, if the end of the row is reached prematurely, then... Data read prematurely (uncommitted) is not stored in the read FIFO. For a single read access, the data will be correctly stored in the read FIFO.

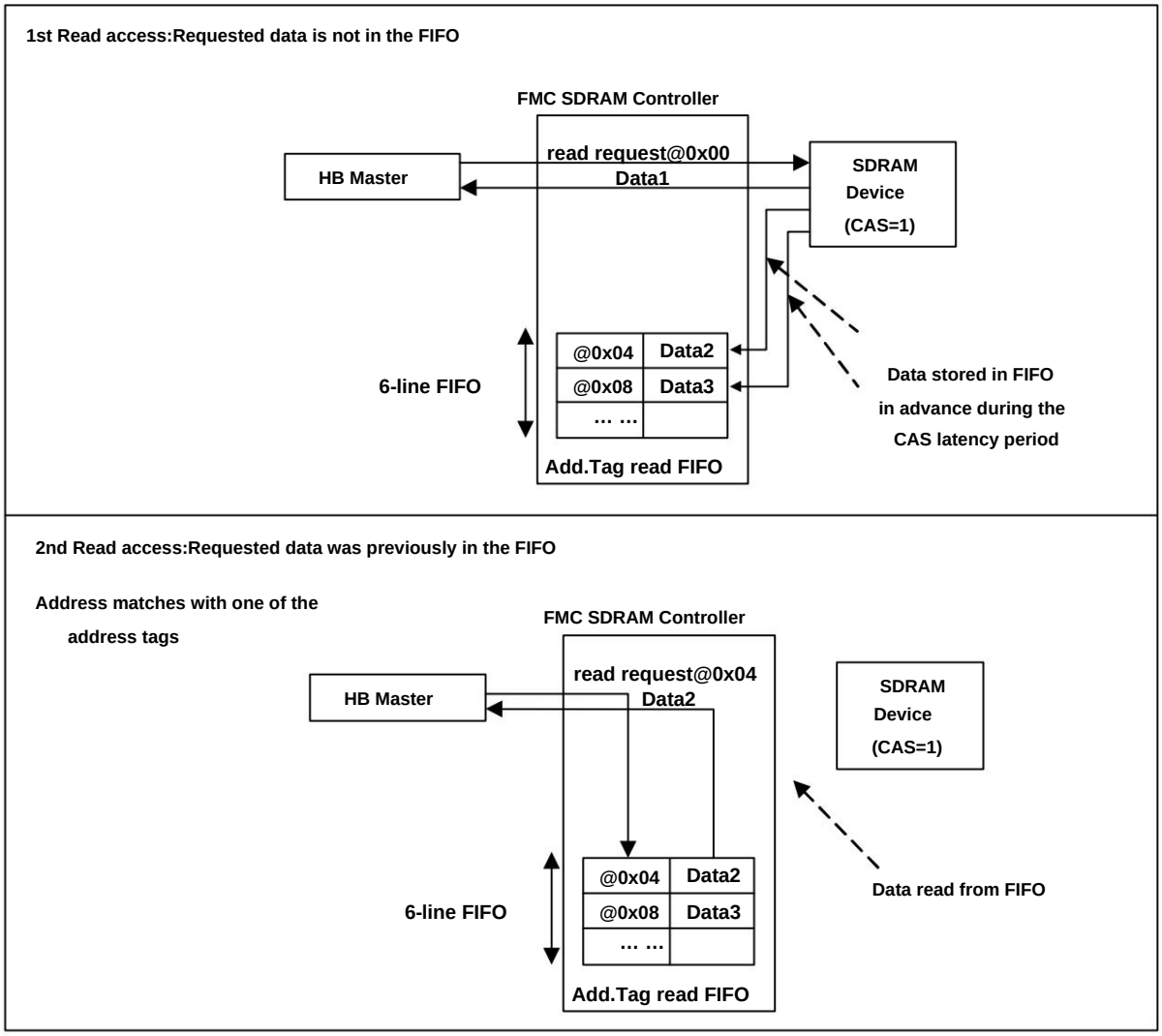
Whenever a read request occurs, the SDRAM controller will check:

• Check if the address matches one of the address tags. If a match is found, read the data directly from the FIFO and clear the corresponding address tag/row.

The content is compressed, and the remaining data in the FIFO is compressed to avoid blank lines.

Otherwise , send a new read command to the memory and update the FIFO with the new data. If the FIFO is full, older data will be lost.

Figure 40-25 Read access logic diagram when RBURST is at position 1 (CAS=1, RPIPE=0)



During a read access or precharge command, the read FIFO will be refreshed and ready to be filled with new data.

Upon receiving the first read request, if the current access has not yet reached the row boundary, the SDRAM controller will perform a CAS latency period and...

The next read access is accepted during the RPIPE delay (if configured). This is achieved by incrementing the memory address. The following must be met:

condition:

The RBURST control bit in the R32_FMC_SDCR1 register must be set to 1. Address management

depends on the next HB request .

The next HB request is consecutive (HB burst).

In this case, the SDRAM controller will increment the address.

The next HB request is not consecutive.

- If the new read request points to the same row as the previous request or another valid row, the new address will be transferred to memory, and the master will...

The device stops working during the CAS delay period, waiting to retrieve new data from the memory.

If a new read request points to an invalid row, the SDRAM controller generates a precharge command, activates the new row, and initializes the read command.

If the RURST position is 0, the read FIFO is not used.

Row and storage region boundary

management: When a read/write access crosses a row boundary, if the next read/write access is consecutive and the current access has already reached the row boundary, then...

The SDRAM controller will perform the following operations:

1) Precharge the valid rows;

2) Activate the new line;

3) Initiate read/write commands.

For various column and data bus width configurations, automatic activation of the next row at row boundaries is supported.

The SDRAM controller can insert additional clock cycles between the following commands as needed:

Insert a command between the precharge and activation commands to match the TRP parameters (only when the next access points to another row in the same storage region) .

hour)

Insert a parameter between the activation and read commands to match the TRCD parameters.

These parameters are defined in the FMC_SDTRx register.

For information on cross-row boundary reads and burst write access, please refer to the following figure.

Figure 40-26 Read access across row boundaries

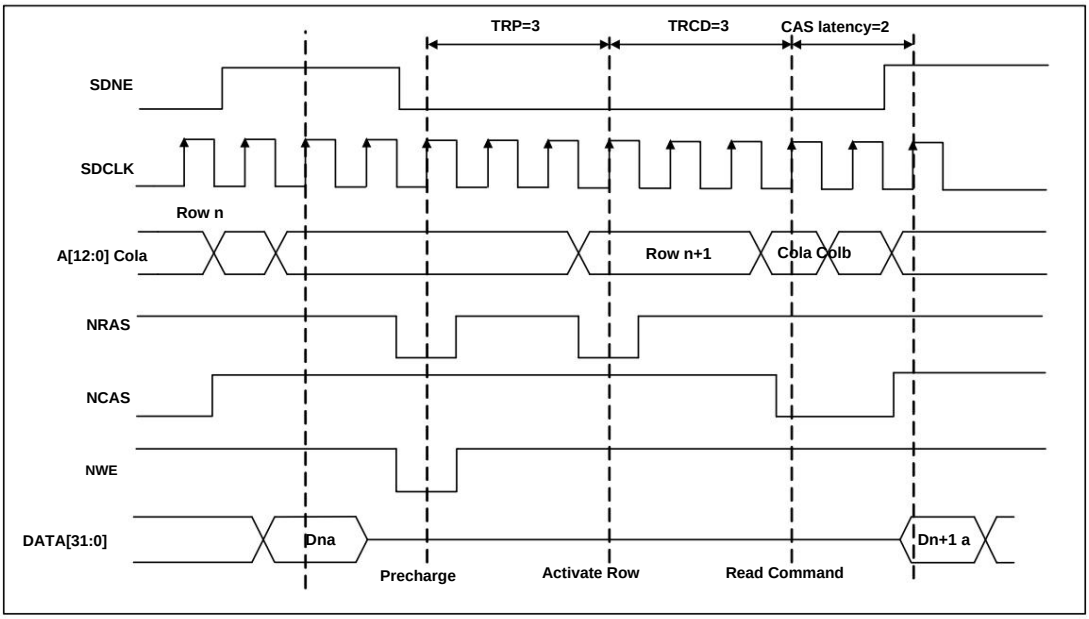
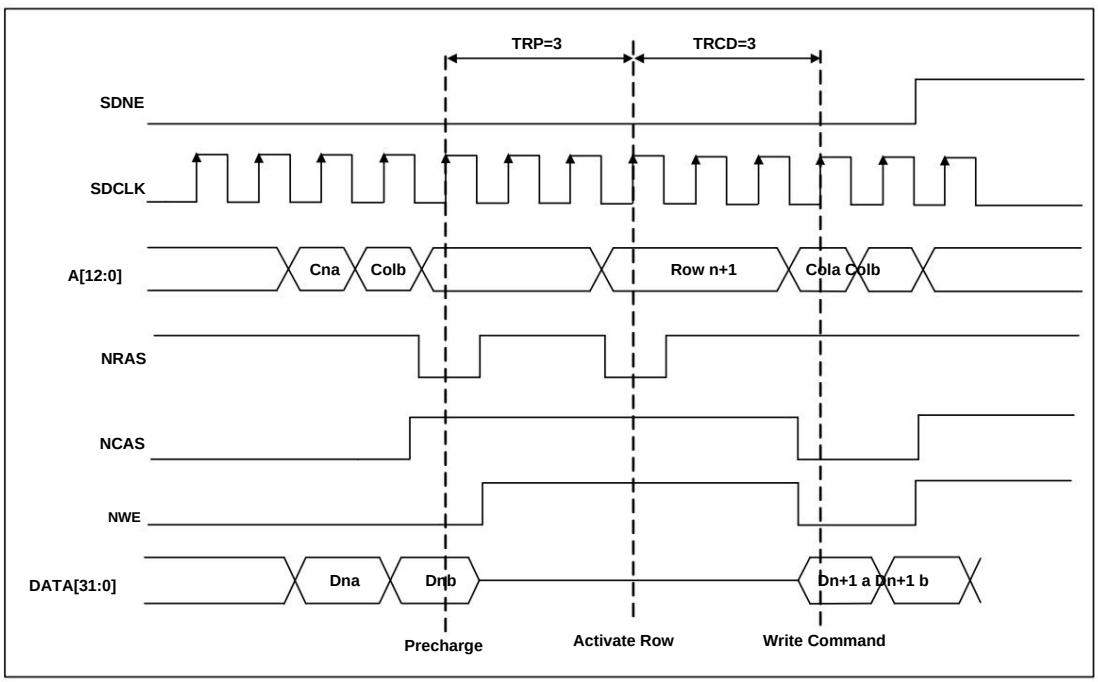


Figure 40-27 Write access across row boundaries



If the next access is consecutive and the current access crosses a memory region boundary, the SDRAM controller will activate the next memory access. The system first rows the region and issues a new read/write command. Two scenarios are possible: ÿ If the current memory region is not the last, the active row in the new memory region must be precharged. Automatic activation of the next row at memory region boundaries is supported for various row/column and data bus width configurations. ÿ If the current memory region is the last, automatic activation of the next row is supported only when addressing SDRAM devices with 13-bit rows, 11-bit columns, 4 internal memory regions, and a 32-bit data bus. Otherwise, the SDRAM address range will be violated, and an HB error will be generated.

For an SDRAM memory with a 13-bit row address, 11-bit column address, 4 internal memory regions, and a 32-bit bus width, the SDRAM controller will continue read/write operations through a second SDRAM device (assuming the second SDRAM device has been initialized): - The SDRAM controller will activate the first row (assuming a valid row exists in the first internal memory region after precharging the valid row) and issue a new read/write command; - If the first row is already activated, the SDRAM controller will only issue read/write commands.

The SDRAM controller's automatic refresh command is used to refresh the contents of the SDRAM device. The SDRAM controller sends this command periodically. It uses an internal counter to load the COUNT value in the R32_FMC_SDRTR register. This value defines the number of memory clock cycles (refresh rate) between refresh cycles. An internal pulse is generated when the counter reaches zero.

If an ongoing memory access occurs, the auto-refresh request will be delayed. However, if both a memory access and an auto-refresh request occur simultaneously, the auto-refresh request will be processed first.

If memory is accessed during an auto-refresh, the access request is cached and processed after the auto-refresh is complete. If a new auto-refresh request occurs before the previous auto-refresh request has completed, the RE in the status register... The (Refresh Error) bit will be set to 1. If this bit is already enabled (REIE=1), an interrupt will be generated.

If the SDRAM rows are not idle (not all rows are closed), the SDRAM controller will generate a PALL (full precharge). (Electric) command, and then perform an automatic refresh.

If an auto-refresh command is generated from the R32_FMC_SDCMR command mode register (mode bit="011"), a PALL command must be issued first. Command (mode bit="010").

40.7.4 Low Power Mode Two low power modes are available: ÿ Self-refresh mode: The SDRAM device itself performs an automatic refresh cycle to retain data, without the need for an external clock. In power-down mode, the SDRAM controller performs an automatic refresh cycle.

The self-refresh mode is selected by setting the MODE position to "101" and configuring the target memory region bits (CTB1 and/or CTB2) in the R32_FMC_SDCMR register.

The SDRAM clock stops running after the TRAS delay, while the internal refresh timer only stops counting if one of the following conditions is met: ÿ A self-refresh command has been issued to both devices ÿ One of the devices is not active (the SDRAM memory area is not initialized).

Before entering self-refresh mode, the SDRAM controller automatically sends a PALL command.

If the write data FIFO is not empty, all data will be sent to memory before self-refresh mode is activated, and the BUSY status flag will remain set to 1. In self-refresh mode, all inputs to the SDRAM device are invalid except for SDCKE, which remains low.

SDRAM devices must be in self-refresh mode for a minimum of TRAS time and be able to remain in self-refresh mode for an even longer period of time.

To ensure this minimum duration, the BUSY status flag will remain high during the TRAS delay after self-refresh activation.

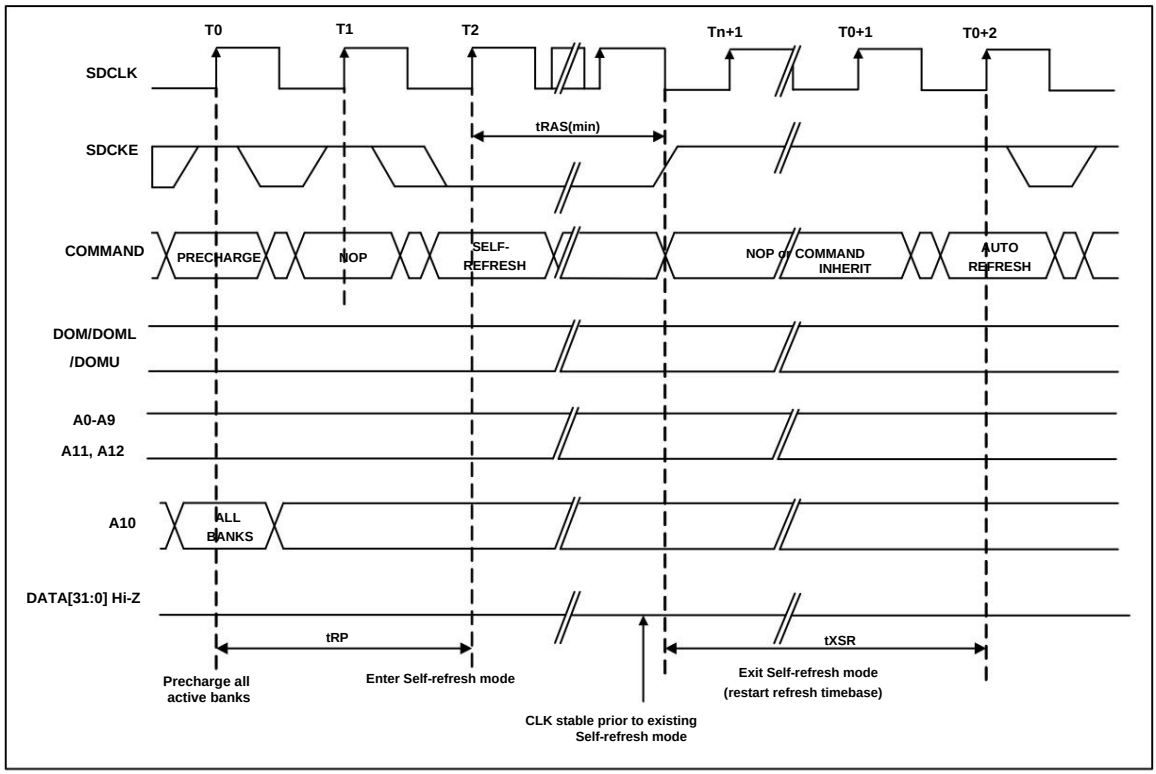
The SDRAM controller generates a command sequence immediately after an SDRAM device is selected to exit self-refresh mode. Memory access complete.

Afterwards, the selected device will remain in normal mode.

To exit self-refresh mode, the MODE setting must be set to "000" (normal mode) and the R32_FMC_SDCMR register must be configured.

Target storage region bits (CTB1 and/or CTB2).

Figure 40-28 Self-refresh mode

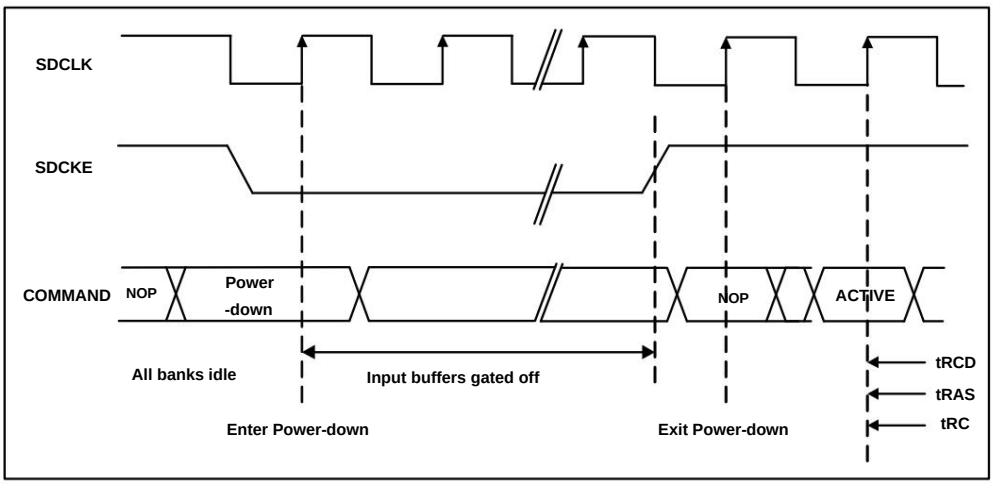


Power-down

mode is achieved by setting the MODE bit to "110" and configuring the target memory region bits (CTB1 and/or CTB2) in the R32_FMC_SDCMR register.

Select the mode.

Figure 40-29 Power-down mode



If the write data FIFO is not empty, all data will be sent to memory before power-down mode is activated.

The SDRAM controller exits power-down mode immediately after an SDRAM device is selected. After memory access is complete, the selected SDRAM... The device will remain in normal mode.

During power-down mode, all input/output buffers of the SDRAM device will be disabled, except for SDCKE which remains low.

SDRAM devices cannot remain in power-down mode for longer than the refresh cycle and cannot perform a self-refresh cycle themselves. Therefore, SDRAM... The controller performs a refresh by doing the following:

1) Exit power-down mode and drive SDCKE high.

2) Generate the PALL command (provided that an active line exists in power-down mode).

3) Generate automatic refresh command

4) Drive SDCKE low again to return to power-down mode.

To exit power-down mode, the MODE setting must be set to "000" (normal mode) and the destination value in the R32_FMC_SDCMR register must be configured.

Mark the storage area bits (CTB1 and/or CTB2).

40.7.5 SDRAM Control Register

Table 40-42 SDRAM Related Register List

name	Access address	describe	Reset value
R32_FMC_SDCR1	0x40025540	SDRAM Control Register 1	0x000002D0
R32_FMC_SDCR2	0x40025544	SDRAM Control Register 2	0x000002D0
R32_FMC_SDTR1	0x40025548	SDRAM Timing Register 1	0x0FFFFFFF
R32_FMC_SDTR2	0x4002554C	SDRAM Timing Register 2	0x0FFFFFFF
R32_FMC_SDCMR	0x40025550	SDRAM Command Mode Register	0x00000000
R32_FMC_SDRTR	0x40025554	SDRAM Refresh Timer Register	0x00000000
R32_FMC_SDSR	0x40025558	SDRAM Status Register	0x00000000
R32_SDRAM_MISC	0x40025580	SDRAM User-defined Register	0x00000000

40.7.5.1 SDRAM Control Register x (R32_FMC_SDCRx) (x=1/2)

Offset address: 0x140 + 4 * (x – 1)

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
Reserved					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	RPIPE[1:0]		RBURST	SDCLK[1:0]		WP	CAS[1:0]		NB	MWID[1:0]	NR[1:0]			NC[1:0]	

Bit	name	access	describe	Reset value
[31:15]	Reserved	RO	is reserved.	0
[14:13]	RPIPE[1:0]	RW	Read the pipe: 00: 0 HCLK clock cycles delay; 01: 1 HCLK clock cycle delay; 10: 2 HCLK clock cycles delay; 11: Retained. Note: The corresponding bits in the R32_FMC_SDCR2 register are irrelevant bits.	0
12	RBURST	RW	Breaking news: 0: Do not manage single read requests as burst requests; 1. Always manage single read requests as bursty requests.	0

			Note: The corresponding bits in the R32_FMC_SDCR2 register are irrelevant bits.	
[11:10]	SDCLK[1:0]	RW	SDRAM clock configuration: 00: Disable SDCLK clock; 01: SDCLK cycle = HCLK cycle; 10: SDCLK cycle = 2 * HCLK cycle; 11: SDCLK cycle = 3 * HCLK cycle; Note: The corresponding bits in the R32_FMC_SDCR2 register are irrelevant bits.	11b
9	WP	RW	Write protection: 1: Ignore write access; 0: Write access is allowed.	0
[8:7]	CAS[1:0]	RW	CAS delay: 00: Reserved; 01: 1 cycle; 10: 2 cycles; 11: 3 cycles.	10b
6	NB	RW	Number of internal storage areas: 1: Four internal storage areas; 0: 2 internal storage areas.	0
[5:4]	MWID[1:0]	RW	Memory data bus width: 00: 8 bits; 01: 16 bits; 10: 32 bits; 11: Retained.	10b
[3:2]	NR[1:0]	RW	Row address bits: 00: 11 bits; 01: 12 bits; 10:13 bits; 11: Retained.	0
[1:0]	NC[1:0]	RW	Column address bits: 00: 8 bits; 01: 9 digits; 10: 10 digits; 11: 11th position.	0

Note: Before modifying the RBURST RPIPE setting or disabling the SDCLK clock, the user must first PALL Command to ensure that the positive result is completed first send or perform the operation.

40.7.5.2 SDRAM Timing Register x (R32_FMC_SDTRx) (x=1/2)

Offset address: 0x148 + 4 * (x – 1)

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
Reserved		TRCD[3:0]	TRP[3:0]	TWR[3:0]	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRC[3:0]				TRAS[3:0]				TXSR[3:0]				TMRD[3:0]			

Bit	name	access	describe	Reset value
[31:28] Reserved		RO is reserved.		0
[27:24] TRCD[3:0]		RW	Row-to-column delay: 0000: 1 cycle; 0001: 2 cycles; ... 1111: 16 cycles.	1111b
[23:20] TRP[3:0]		RW	Line precharge delay: Configure the TRP timing only in the R32_FMC_SDTR1 register. For example... If two SDRAM devices are used, the slowest device must be used. Timing configuration (TRP). 0000: 1 cycle; 0001: 2 cycles; ... 1111: 16 cycles. Note: The corresponding bits in the R32_FMC_SDTR2 register are irrelevant bits.	1111b
[19:16] TWR[3:0]		RW	Recovery delay: 0000: 1 cycle; 0001: 2 cycles; ... 1111: 16 cycles. Note: TWR must be set to match the write recovery time (tWR) defined in the SDRAM datasheet, ensuring that: TWR ≥ TRAS - TRCD, and TWR ≥ TRC - TRCD - TRP. For example: TRAS = 4 cycles, TRCD = 2 cycles. Therefore, TWR ≥ 2 cycles. TWR must be set to 0x1. If you use a device, and two SDRAM Then it must Must R32_FMC_SDTR1 and R32_FMC_SDTR2 are configured with the same TWR timing (corresponding to the slower SDRAM device).	1111b
[15:12] TRC[3:0]		RW	Row loop delay: Configure TRC timing only in the R32_FMC_SDTR1 register. For example... If two SDRAM devices are used, the slowest device must be used. Timing configuration TRC. 0000: 1 cycle; 0001: 2 cycles; ... 1111: 16 cycles. Note: TRC must be used with Defined in the SDRAM device datasheet TRC and The TRFC (Automatic Refresh Cycle) timings are matched. Note: The corresponding bits in the R32_FMC_SDTR2 register are irrelevant bits.	1111b
[11:8] TRAS[3:0]		RW	Refresh time: 0000: 1 cycle; 0001: 2 cycles; ... 1111: 16 cycles.	1111b

[7:4] TXSR[3:0]		RW	Exit self-refresh delay: 0000: 1 cycle; 0001: 2 cycles; ... 1111: 16 cycles. Note: If two SDRAM devices are used, then it must be... R32_FMC_SDTR1 R32_FMC_SDTR2 configuration is the same TXSR And timing (corresponding to slower SDRAM devices).	1111b
[3:0] TMRD[3:0]		RW	Load mode register to activation: 0000: 1 cycle; 0001: 2 cycles; ... 1111: 16 cycles.	1111b

Note: If two SDRAM devices are connected, the command mode register can be accessed simultaneously for both devices (load mode register command).
In this case, access will be issued according to the timing parameters (TMRD timing) configured for the memory region in the R32_FMC_SDTR1 register.
1. The TRAS command Configuration and timing
are only performed in the R32_FMC_SDTR1 register. If two SDRAM devices are used, then the following must be used:
TRP and TRC are
the timing configurations for the slowest device.
and TRP TRC

40.7.5.3 SDRAM Command Mode Register (R32_FMC_SDCMR) Offset

Address: 0x150

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											MRD[12:7]				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MRD[6:0]							NRFS[3:0]				CTB1	CTB2	MODE[2:0]		

Bit	name	access	describe	Reset value
[31:22] Reserved		RO is		0
[21:9] MRD[12:0]		RW	reserved. Mode register definition: This 13-bit field defines the contents of the SDRAM mode register. It can be accessed via... The Load Mode Register command programs the mode register.	0
[8:5] NRFS[3:0]		RW	Automatic refresh count: These are the continuous self-refreshes issued when MODE="011". Number of commands. 0000: 1 self-refresh cycle; 0001: 2 self-refresh cycles; ... 1110: 15 self-refresh cycles; 1111: 16 self-refresh cycles.	0
4	CTB1	RW	Command target storage area 1: 1: Command has been sent to SDRAM storage area 1; 0: The command was not sent to SDRAM storage area 1.	0
3	CTB2	RW	Command target storage area 2: 1: The command has been sent to SDRAM storage area 2.	0

			0: The command was not sent to SDRAM memory area 2;	
[2:0] MODE	[2:0]	RW	Command mode: 000: Normal mode; 001: Clock configuration enabled; 010: PALL ("Precharge all memory regions") command; 011: Automatic refresh command; 100: Load Mode Register; 101: Self-refresh command; 110: Power off command; 111: Reserved. Note: After the command is issued, at least one command target storage area bit... (CTB1 or) CTB2) must be set to 1; otherwise, the command will be ignored. Note: If two SDRAM memory areas are used, it must be done via... The location and direction will be simultaneously sent from the two devices. CTB1 CTB2 1 Refresh and PALL The command must be executed; otherwise, it will be ignored. Note: If only one SDRAM storage area is used, and by... Its related CTB 1 Location To issue a command, another unused storage area CTB 0	0

40.7.5.4 SDRAM Refresh Timer Register (R32_FMC_SDRTR) Offset Address:

0x154

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

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

Bit	name	access	describe	Reset value
[31:15] Reserved		RO	is reserved.	0
14	REIE	RW	RES Interrupt Enable: 1: An interrupt is generated when RE=1; 0: Interruption is disabled.	0
[13:1]	COUNT[12:0]	RW	Refresh the timer count: This 13-bit field defines the refresh rate of the SDRAM device for storing... This field represents the number of clock cycles. It must be set to at least 41. SDRAM clock cycle. Refresh rate = (COUNT + 1) * SDRAM frequency clock; COUNT = (SDRAM refresh cycle / number of rows) - 20.	0
0	CRE	RO	Clear refresh error flags: This bit is used to clear the refresh error flag in the status register. (RE). 1: Clear refresh error flags; 0: No operation.	0

Note: The programmed COUNT value cannot be equal to the sum of the following timings: TWR + TRP + TRC + TRCD + 4 memory clock cycles.

40.7.5.5 SDRAM Status Register (R32_FMC_SDSR)

Offset address: 0x158

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										BUSY	MODES2[1:0]		MODES1[1:0]		RE

Bit	name	access	describe	Reset value
[31:6]	Reserved	RO	reserved.	0
5	BUSY	RO	Busy status: 1: The SDRAM controller is not ready to accept new requests; 0: The SDRAM controller is ready to accept new requests.	0
[4:3]	MODES2[1:0]	RO	State mode of storage area 2: 00: Normal mode; 01: Self-refresh mode; 10: Power-off mode.	0
[2:1]	MODES1[1:0]	RO	State mode of storage area 1: 00: Normal mode; 01: Self-refresh mode; 10: Power-off mode.	0
0	RE	RO	Refresh error flags: 1: A refresh error was detected; 0: No refresh error detected. An interrupt will be generated when REIE=1 and RE=1.	0

40.7.5.6 SDRAM User-Defined Register (R32_SDRAM_MISC)

Offset Address: 0x180

31	30	29	28	27	26	25	24	23	twenty two	twenty one	20	19	18	17	16
Reserved														En_Bank2	En_Bank1

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

Bit	name	access	describe	Reset value
[31:18]	Reserved	RO	is reserved.	0
17	En_Bank2	RW	enables SDRAM2.	0
16	En_Bank1	RW	enables SDRAM1.	0
15	Enhance_read_mode	RW	1: Read-enhanced/prefetch mode; 0: Normal read mode.	0

			The controller will pre-read data from the following 8, 16, and 32 addresses (root). Depending on the bit width of the SDRAM, 256 bits of data need to be retrieved. Reading speed can be improved when addresses are contiguous. Note: The RBURST bit must be disabled when this bit is enabled.	
[14:8] Reserved		RW is		0
[7:4] Phase_Sel[3:0]		RW	reserved. Output the phase offset of SDR_CLK: Bit[7]: Output phase inversion; Bit[6:4]: The phase increases by approximately 0.4ns each time.	0
[3:0] NRFS_CNT[3:0]		RW	Define the number of times the controller refreshes each time: 0000: 1 cycle; 0001: 2 cycles; ... 1110: 15 cycles; 1111: 16 cycles.	0

Chapter 41 Encryption Module (ECDC)

The system has a built-in block cipher algorithm module, supporting both AES and SM4 block cipher algorithms, as well as electronic codebooks (ECB) and other data structures. CTR (Cipher Transmitter) mode. The module completes an encryption/decryption process in 128-bit data units, providing access to data in memory. DMA encryption/decryption and SFR register single-pass encryption/decryption mode.

41.1 Main Features

ÿ ECB and CTR modes for 128-bit keys in the SM4 algorithm

ÿ AES algorithm 128/192/256-bit key ECB mode and CTR mode

Supports direct encryption of single 128-bit data via software-written SFRs .

Supports DMA (memory-to-memory) encryption/decryption software to handle data blocks of specified lengths.

41.2 AES/SM4 Algorithm

AES (Advanced Encryption Standard) is a block cipher algorithm that uses a symmetric block cipher system. It is considered one of the most popular algorithms in key encryption. The SM4 block cipher algorithm is generally used for dedicated block ciphers in wireless LANs and trusted computers. Cryptographic algorithms can also be used for data encryption and protection in other environments.

During data encryption and decryption, a key needs to be loaded. For the AES algorithm, the key length is set to 128/192/256 bits. When using this method, the user key needs to be expanded to 11×128/13×128/15×128 bits respectively. The SM4 algorithm, however, expands the 128-bit key... The user key is expanded into a 32×32-bit extended key. These extended keys are stored in internal registers for easy access during encryption and decryption. use.

41.3 ECB and CTR Modes

AES/SM4 supports two modes: Electronic Codebook (ECB) mode and Counter (CTR) mode. The CTR mode offers higher security. The capability should be higher than ECB mode. See the working block diagrams in Figures 41-1 and 41-2.

Figure 41-1 ECB mode encryption and decryption process

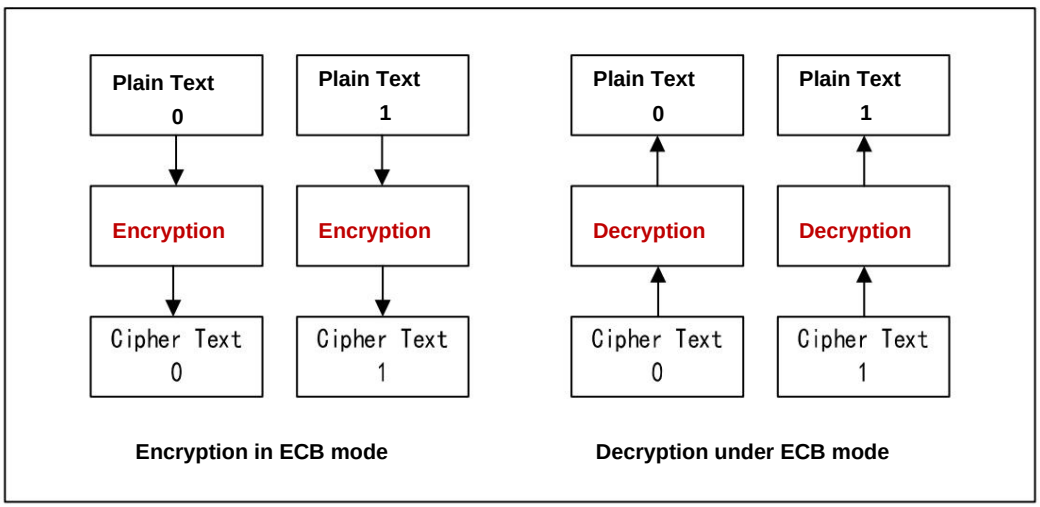
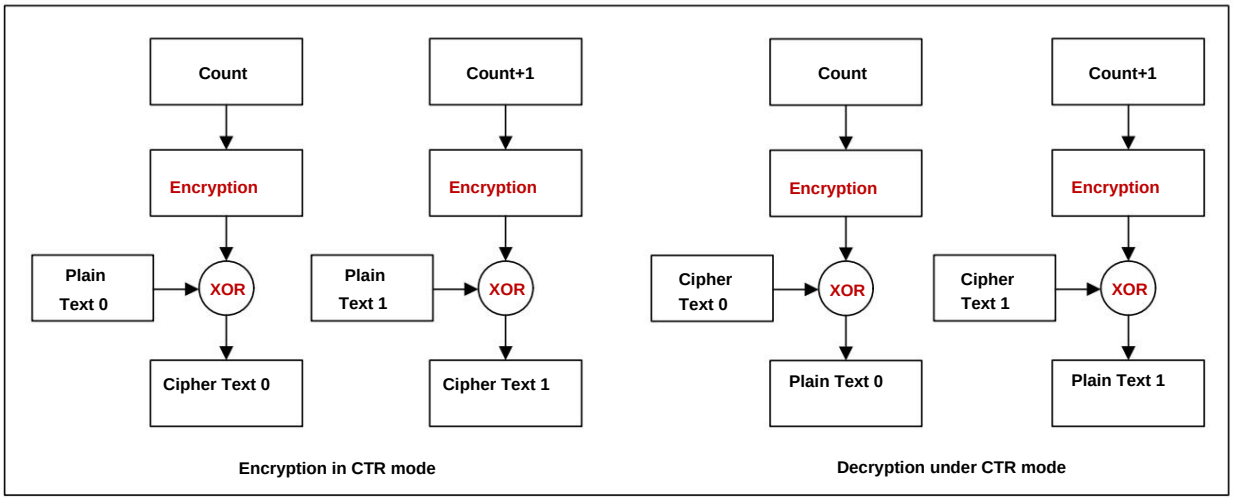


Figure 41-2 CTR mode encryption and decryption process



In ECB mode, there is a one-to-one correspondence between plaintext and ciphertext; the encrypted plaintext is directly...

In CTR mode, a 128-bit count value needs to be preloaded, encrypted, and then the encrypted count value is used as the ciphertext.

The plaintext is XORed to obtain the ciphertext. It's worth noting that in CTR decryption mode, only the count value is encrypted, not decrypted.

41.4 Module Application

41.4.1 Encryption/Decryption Module Initialization

ÿ Clock Configuration: Set the RB_ECDC_CLKDIV_MASK bit in the R32_ECEC_CTRL register to configure the module's operating clock. If

If you don't want to use this module, you can set it to 0 to disable the module and reduce power consumption.

Endianness : Setting the RB_ECDC_DAT_MOD bit in the R32_ECEC_CTRL register allows you to select the endianness of encrypted and decrypted data.

Storage method. Stored in 128-bit (16-byte) packets, both big-endian and small-endian.

Key expansion: An expanded key is used instead of the user key during encryption and decryption. After setting the user key length and content,

The hardware needs to be notified to perform a key expansion operation. Once the key is expanded, it will be saved to the module's internal registers. Only then can normal operation proceed.

Data encryption and decryption are performed. The R32_ECEC_CTRL register is configured with RB_ECDC_WRSRAM_EN and RB_ECDC_ALGRM_MOD.

RB_ECDC_CIPHER_MOD and RB_ECDC_KLEN_MASK enable SRAM data encryption/decryption, select the encryption/decryption algorithm, and choose the block cipher mode.

Key length; then configure the user key register group R32_ECDC_KEY, filling in the user key content. Set the R32_ECEC_CTRL register...

After setting RB_ECDC_KEYEX_EN to 1 and then clearing it to 0, the hardware begins to execute the key expansion function.

Initial counter value: If CTR mode is used for encryption and decryption, an initial counter value is required. Set the CTR mode counter value...

The register group R32_ECDC_IV is filled with a 128-bit counter value, and each successful data encryption/decryption (128 bits of data) is performed internally by the module.

The counter value will be incremented by 1. This counter value is not required in ECB mode.

41.4.2 The encryption/decryption

configuration module's encryption/decryption data stream includes two methods: memory-to-memory and single 128-bit data encryption/decryption. Please refer to the table for details.

Configuration 41-1.

Table 41-1 Configuration of Different Data Stream Modes

Encryption/decryption data streams	RB_ECDC_WRSRAM_EN	RB_ECDC_NORMAL_EN	RB_ECDC_MODE_SEL	other
SRAM Data Encryption	1	1	0	Registers need to be configured R32_ECDC_SRC_ADDR, R32_ECDC_DST_ADDR and R32_ECDC_SRAM_LEN
SRAM Data Decryption	1	1	1	

128 bits of data per time encryption	-	1	0	Write to R32_ECDC_SGSD register Enter the raw data (plaintext or plaintext) into the device. (ciphertext)
128 bits of data per time Decryption	-	1	1	Read R32_ECDC_SGRT register The device reads the encrypted and decrypted results. data.

The data transfer between memory and memory is initiated by writing non-zero data to the R32_ECDC_SRAM_LEN length register, and then...
Check the RB_ECDC_IF_WRSRAM bit in the R32_ECDC_INT_FG register. If it is set to 1, it indicates that the data conversion from memory to memory is complete.
Successfully clear the RB_ECDC_IF_WRSRAM bit status. If the RB_ECDC_IE_WRSRAM function in the R32_ECDC_CTRL register is enabled...
Yes, an interrupt service will be triggered when the RB_ECDC_IF_WRSRAM flag is set.

Single-pass encryption/decryption must be configured for big-endian (RB_ECDC_DAT_MOD=1) for execution. 128-bit data single-pass encryption/decryption conversion.
The write operation to the highest 32 bits of the R32_ECDC_SGSD register is initiated by polling the R32_ECDC_INT_FG register.
The RB_ECDC_IF_SINGLE bit, if set to 1, indicates that a single 128-bit data conversion is complete, and clears the RB_ECDC_IF_SINGLE bit status.
If the RB_ECDC_IE_SINGLE function of the R32_ECEC_CTRL register is enabled, when the RB_ECDC_IF_SINGLE flag is set...
This will trigger a service interruption.

41.5 Register Description

Table 41-2 ECDC Module Register List

name	Access address	describe	Reset value
R32_ECDC_CTRL	0x40016C00 ECDC Control Register;		0x00000020
R32_ECDC_INT_FG	0x40016C04 ECDC Interrupt Flag Register;		0x00000000
R32_ECDC_KEY_255T224	0x40016C08 User Key Register 7; 0x40016C0C		0xXXXXXXXX
R32_ECDC_KEY_223T192	User Key Register 6; 0x40016C10 User Key		0xXXXXXXXX
R32_ECDC_KEY_191T160	Register 5; 0x40016C14 User Key Register 4;		0xXXXXXXXX
R32_ECDC_KEY_159T128	0x40016C18 User Key Register 3; 0x40016C1C		0xXXXXXXXX
R32_ECDC_KEY_127T96	User Key Register 2; 0x40016C20 User Key		0xXXXXXXXX
R32_ECDC_KEY_95T64	Register 1; 0x40016C24 User Key Register 0;		0xXXXXXXXX
R32_ECDC_KEY_63T32	0x40016C28 CTR Mode Count Register 3;		0xXXXXXXXX
R32_ECDC_KEY_31T0	0x40016C2C CTR Mode Count Register 2;		0xXXXXXXXX
R32_ECDC_IV_127T96	0x40016C30 CTR Mode Count Register 1; 0x40016C34		0xXXXXXXXX
R32_ECDC_IV_95T64	CTR Mode Count Register 0; 0x40016C40 Single-time		0xXXXXXXXX
R32_ECDC_IV_63T32	encryption/decryption raw data 3 0x40016C44 Single-		0xXXXXXXXX
R32_ECDC_IV_31T0	time encryption/decryption raw data 2 0x40016C48		0xXXXXXXXX
R32_ECDC_SGSD_127T96	Single-time encryption/decryption raw data 1		0x00000000
R32_ECDC_SGSD_95T64	0x40016C4C Single-time encryption/decryption raw		0x00000000
R32_ECDC_SGSD_63T32	data 0 0x40016C50 Single-time encryption/decryption		0x00000000
R32_ECDC_SGSD_31T0	result 3 0x40016C54 Single-time encryption/decryption		0x00000000
R32_ECDC_SGRT_127T96	result 2 0x40016C58 Single-time encryption/		0x00000000
R32_ECDC_SGRT_95T64	decryption result 1 0x40016C5C Single-time		0x00000000
R32_ECDC_SGRT_63T32	encryption/decryption result 0 0x40016C60		0x00000000
R32_ECDC_SGRT_31T0	Source address of the encryption/decryption		0x00000000
R32_ECDC_SRC_ADDR	SRAM region		0x00000000

R32_ECDC_DST_ADDR	0x40016C64 Encrypt/decrypt SRAM destination address;	0x00000000
R32_ECDC_SRAM_LEN	0x40016C68 Encrypt/decrypt SRAM size	0x00000000

41.5.1 ECDC Control Register (R32_ECDC_CTRL) Offset Address: 0x00

Bit Name

		access	describe	Reset value
[31:26] Reserved		RO reserved. 1:		0
25	RB_ECDC_AES/SM4_CL OCK_EN	RW	Enable encryption/decryption hardware clock: 0: Disable encryption/decryption hardware clock; Note: This bit is set to 1 when encryption/decryption is used; otherwise, this bit is set to 1. Set to 0 It can also save power consumption.	0
twenty four	RB_ECDC_CLOCK_SELE CT	RW	1: Encryption and decryption work under pllclk (maximum supported 400M); 0: The operating frequency is the frequency division clock indicated by [6:4]. Note: The ECDC clock must be greater than or equal to the HCLK clock.	0
[23:19] Reserved		RO is reserved.		0
18	RB_ECDC_IE_WRSRAM	RW	Memory-to-memory encryption/decryption completion interrupt enabled. 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
17	RB_ECDC_IE_SINGLE	RW	Single encryption/decryption completion interrupt enabled. For 128-bit registers. Conversion complete. 1: Enable the corresponding interrupt; 0: Disable the corresponding interrupt.	0
[16:14] Reserved		RO is reserved. This		0
13	RB_ECDC_DAT_MOD	WO	selects whether to use endianness for encryption/decryption. 1: Big-endian method; 0: Little-endian mode. Note: Single 128-bit encryption/decryption must be configured for big-endian. Mode.	0
12	Reserved	RO Reserved.		0
[11:10] RB_ECDC_KLEN_MASK		RW	Key length setting. 00: 128 bits; 01: 192 bits; 10:256 bits; 11: Retained.	0
9	RB_ECDC_CIPHER_MOD RW		Block cipher mode selection. 1: CTR mode; 0: ECB mode.	0
8	RB_ECDC_ALGRM_MOD	RW	Encryption/decryption algorithm mode selection. 1: AES; 0:SM4.	0
7	RB_ECDC_WRSRAM_EN	RW	Enable SRAM data encryption/decryption. Control bits required. Use RB_ECDC_KEYEX_EN together. 1: Turn on; 0: Off.	0

[6:4]	RB_ECDC_CLKDIV_MASK	RW	Encryption/decryption clock division coefficients. Calculate: EDclk = PLLCLK / ED_CLK_PRE. A minimum value of 2, writing 1, is equivalent to disabling the ECDC module's operating clock. Write 0 to disable the ECDC module's operating clock. Note: The ECDC clock must be greater than or equal to the HCLK clock.	010b
3	RB_ECDC_MODE_SEL	RW	Encryption/decryption mode selection. 1: Decryption mode; 0: Encryption mode.	0
2	Reserved	RW	Reserved	0
1	RB_ECDC_NORMAL_EN	RW	Enabling encryption and decryption is enabled in normal encryption and decryption mode; the software is detecting... After the key expansion is complete, set this bit to 1 before starting encryption and decryption. 1: Encryption/decryption enabled; 0: Not working.	0
0	RB_ECDC_KEYEX_EN	RW	The key expansion function enable control bit is activated by a high-level pulse. After configuring the key length, this bit needs to be set to 1, and the key will be awaited. Encryption and decryption can only proceed after the expansion is complete. Note: Application code-driven implementations need to be set to high and then low.	0

41.5.2 ECDC Interrupt Flag Register (R32_ECDC_INT_FG)

Offset address: 0x04

Bit	name	access	describe	Reset value
[31:19]	Reserved	RO	is reserved. The	0
18	RB_ECDC_IF_WRSRAM	RW1Z	interrupt flag for memory-to-memory encryption/decryption completion is written to 1 to clear. 0. 1: Encryption/decryption complete event; 0: No event.	0
17	RB_ECDC_IF_SINGLE	RW1Z	A single encryption/decryption completion interruption flag is set; writing 1 clears it to 0. 1: Single encryption/decryption completion event; 0: No event.	0
[16:0]	Reserved	RW	Reserved	0

41.5.3 User Key Register Group (R32_ECDC_KEY) Offset Addresses:

0x08, 0x0C, 0x10, 0x14, 0x18, 0x1C, 0x20, 0x24 Name Access

Bit			describe	Reset value
[31:0]	R32_ECDC_KEY_255T224	RW	User key 224-256 bits.	X
[31:0]	R32_ECDC_KEY_223T192	RW	User key 192-223 bits.	X
[31:0]	R32_ECDC_KEY_191T160	RW	User key, bits 160-191.	X
[31:0]	R32_ECDC_KEY_159T128	RW	User key, bits 128-159.	X
[31:0]	R32_ECDC_KEY_127T96	RW	User key 96-127 bits.	X
[31:0]	R32_ECDC_KEY_95T64	RW	User key, 64-95 bits.	X
[31:0]	R32_ECDC_KEY_63T32	RW	User RW user key is 32-63 bits.	X
[31:0]	R32_ECDC_KEY_31T0		key bits 0-31.	X

41.5.4 CTR Mode Count Value Register Group (R32_ECDC_IV)

Offset address: 0x28, 0x2C, 0x30, 0x34 (bit name

access)		describe	Reset value
[31:0] R32_ECDC_IV_127T96	RW count value is 96-127 bits.		X
[31:0] R32_ECDC_IV_95T64	RW count value is 64-95 bits.		X
[31:0] R32_ECDC_IV_63T32	RW count value is 32-63 bits.		X
[31:0] R32_ECDC_IV_31T0	RW count value 0-31 bits.		X

41.5.5 Single Encryption/Decryption Raw Data Register Group (R32_ECDC_SGSD) Offset

Address: 0x40, 0x44, 0x48, 0x4C Bit Name Access

			Reset value
Description [31:0] R32_ECDC_SGSD_127T96 RW Raw data 96-127 bits.			0
[31:0] R32_ECDC_SGSD_95T64 [31:0]	RW raw data is 64-95 bits.		0
R32_ECDC_SGSD_63T32 [31:0]	RW raw data is 32-63 bits.		0
R32_ECDC_SGSD_31T0	RW represents raw data bits 0-31.		0

Note: Internal encryption and decryption are performed at a fixed size of 128 bits. The hardware automatically enables encryption/decryption after the write operation to the R32_ECDC_SGSD_31T0 register is completed.

A single encryption/decryption conversion is initiated. The `RB_ECDC_IF_SINGLE` method is used to determine whether the conversion is complete.

41.5.6 Single Encryption/Decryption Result Register Group (R32_ECDC_SGRT)

Offset Address: 0x50, 0x54, 0x58, 0x5C Bit Name

Access		describe	Reset value
[31:0] R32_ECDC_SGRT_127T96 RW Data bits 96-127.			0
[31:0] R32_ECDC_SGRT_95T64 [31:0]	RW data is 64-95 bits.		0
R32_ECDC_SGRT_63T32 [31:0]	RW data is 32-63 bits.		0
R32_ECDC_SGRT_31T0	RW data bits 0-31.		0

Note: The encryption and decryption results are stored in fixed big-endian format, with a 128-bit structure.

41.5.7 Encrypting/Decrypting SRAM Region Source Address (R32_ECDC_SRC_ADDR)

Offset Address: 0x60

Bit	Name Access	describe	Reset value
[31:0] ECDC_SRAM_SRC_ADDR RW: The source starting address for encrypting/decrypting SRAM data.			0

Note: This addressing is in the region. DMA SRAM

41.5.8 SRAM Encryption/Decryption Destination Address (R32_ECDC_DST_ADDR)

Offset Address: 0x64

Bit	name	access	describe	Reset value
[31:0] ECDC_SRAM_DST_ADDR RW			The starting address of the destination for encrypting/decrypting SRAM data, in units of 128-bit. Writing a non-zero value will initiate SRAM data processing. Encryption and decryption process. (32-byte aligned)	0

41.5.9 Encrypting/Decrypting SRAM Size (R32_ECDC_SRAM_LEN) Offset

Address: 0x68 -bit

Name		access	describe	Reset value
[31:13] Reserved		RO is reserved.		0

[12:0] ECDC_SRAM_LEN		RW	Encryption/decryption SRAM data length, in 128 bits. When writing... When the value is non-zero, the encryption/decryption process of the SRAM data will be initiated.	0

Chapter 42 Digital Filters for ÿÿ Modulators (DFSDM)

The DFSDM is a high-performance module specifically designed for connecting external ÿÿ modulators to an MCU. It includes two external digital serial interfaces. It features an input port and two digital filters, offering flexible ÿÿ digital processing options to provide up to 24-bit ADC final resolution. (DFSDM) It also features the option to select parallel data stream input from the internal ADC peripheral or device memory.

42.1 Main Features

Provides 2 multiplexed input digital serial channels

Two internal digital parallel channels support optional inputs.

Supports adjustable digital signal processing

Supports output data resolution up to 24 bits .

Supports signed data formats .

Supports automatic data offset correction (offset values are stored in registers by the user).

Includes two conversion modes: single conversion mode and continuous conversion mode.

Conversion can be initiated synchronously via software triggering, internal timers, external events, or using the first DFSDM filter .

Supports simulated watchdog timer

Built- in short-circuit detector to detect saturated analog input values (lower and upper limits).

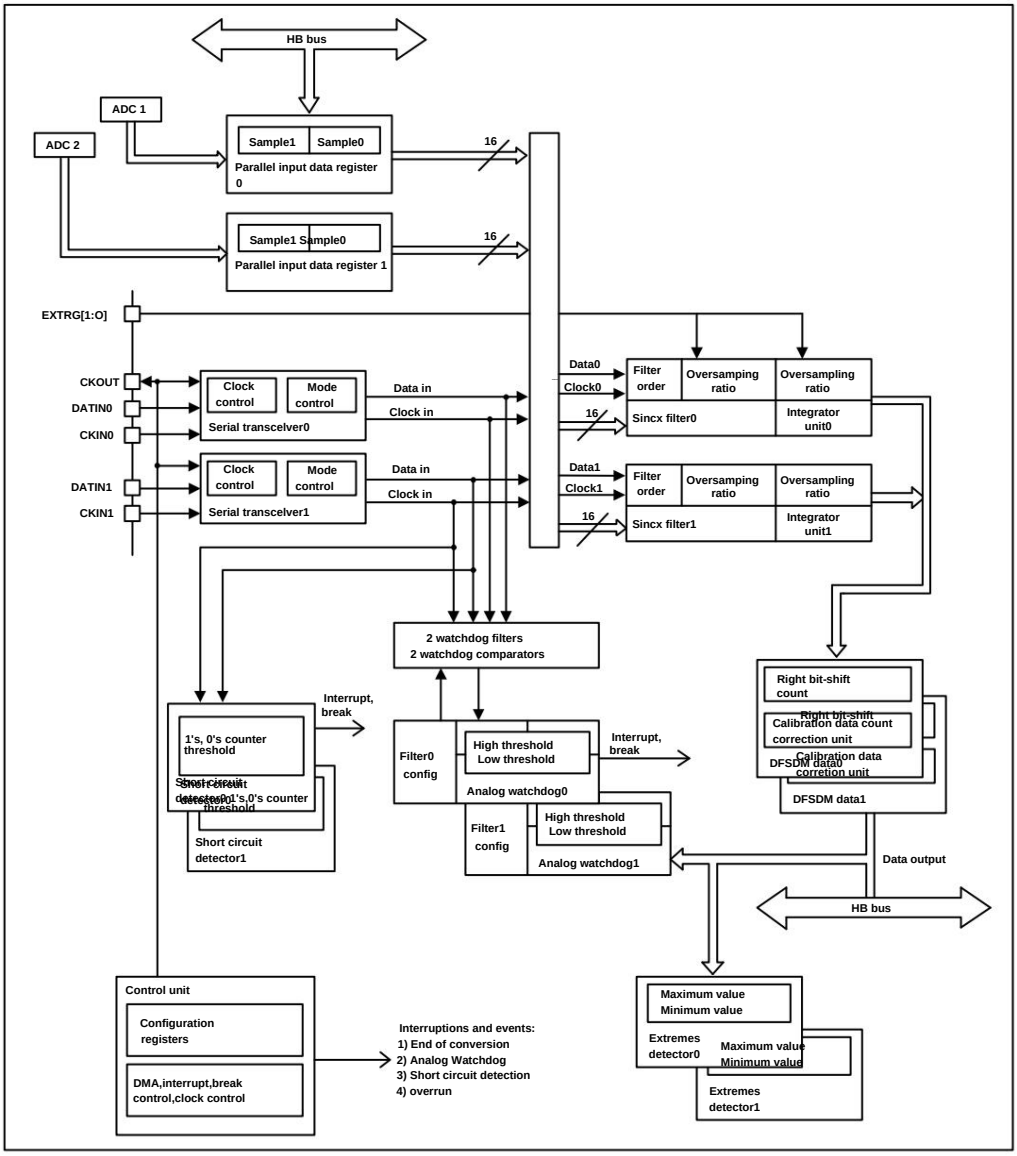
Supports circuit breaking when simulated watchdog events and short-circuit detection events occur .

Built -in extreme value detector

Supports interrupts and DMA .

42.2 Overview

Figure 42-1 Block diagram of DFSDM



42.3 Functional Description

42.3.1 DFSDM Reset and Clock

The DFSDM switch is

controlled by setting the DFSDMEN bit in the R32_DFSDM_CH0CFGR1 register to 1, thus globally enabling the DFSDM interface.

After the interface is globally enabled, all configured input channels y (y=0/1) and digital filters DFSDM_FLTx (x=0/1) will be in their respective...

The enable bits (Channel enable bit CHEN in R32_DFSDM_CHyCFGR1 and DFSDM_FLTx enable bit in R32_DFSDM_FLTxCR1)

It starts working when DFEN is set to 1.

The digital filter DFSDM_FLTx (x=0/1) is enabled by setting DFEN to 1 in the R32_DFSDM_FLTxCR1 register.

After DFSDM_FLTx (DFEN=1), the Sincx digital filter unit and integrator unit will be reinitialized.

By clearing DFEN, all ongoing transformations will immediately stop, and DFSDM_FLTx will be placed in stop mode. Except...

Except for R32_DFSDM_FLTxAWSR and R32_DFSDM_FLTxISR (both of which were reset), all register settings remain unchanged.

Channel y is enabled by setting CHEN to 1 in the R32_DFSDM_CHyCFGR1 register (y=0/1). After the channel is enabled, it will be accessed from the outside.

The ÿÿ modulator or parallel internal data source (CPU/DMA write operation of ADC or memory) receives serial data.

Before stopping the system clock and putting the device into stop mode, DFSDM must be globally disabled (by setting DFSDMEN=0 in R32_DFSDM_CH0CFGR1).

The internal

DFSDM clock , fDFSDMCLK, drives the channel transceivers, digital processing modules (including digital filters and integrators), and subsequent add-on modules (such as analog watchdogs, short-circuit detectors, extreme value detectors, and control modules). This clock is generated by the RCC module and originates from the peripheral clock HCLK or the audio clock. The DFSDM clock automatically stops in stop mode (for all DFSDM_FLTx, x=0/1, DFEN=0).

The DFSDM serial channel transceiver can receive an external serial clock to sample an external serial data stream. If standard SPI encoding is used, The internal DFSDM clock must be at least 4 times faster than the external serial clock;

The DFSDM supports providing an external output clock signal to drive the clock input of an external ÿÿ modulator. This output clock signal is output via the CKOUT pin. The frequency of the output clock signal must conform to the range specified in the device datasheet and can be obtained from the DFSDM clock or audio clock (see the CKOUTSRC bit in the R32_DFSDM_CH0CFGR1 register) using a frequency divider. The programmable range of the frequency divider is 2 to 256 (the CKOUTDIV bit in the R32_DFSDM_CH0CFGR1 register). The audio clock is obtained by configuring PLLCLK by selecting PLL_CLK through the SYSPLL_SEL[2:0] field in the RCC configuration.

42.3.2 Serial Channel Transceivers This

peripheral has two multiplexed serial data channels, which can be selected for conversion using various filters, analog watchdogs, or short-circuit detectors. These serial transceivers receive data streams from an external ÿÿ modulator. The data streams can be transmitted in SPI format (see the SITP bit of the R32_DFSDM_CHyCFGR1 register).

The channel can be enabled by setting CHEN to 1 in the R32_DFSDM_CHyCFGR1 register.

The serial inputs (data

and clock signals) of the DATINy and CKINy pins can be redirected from subsequent channel pins. The serial input channel redirection function is configured via the CHINSEL bit in the R32_DFSDM_CHyCFGR1 register. Channel redirection can be used to acquire audio data from a

PDM (Pulse Density Modulation) stereo microphone. A PDM stereo microphone contains one data signal and one clock signal, where the data signal provides information for the left and right audio channels (sampling on the rising edge of the clock for the left channel and sampling on the falling edge of the clock for the right channel).

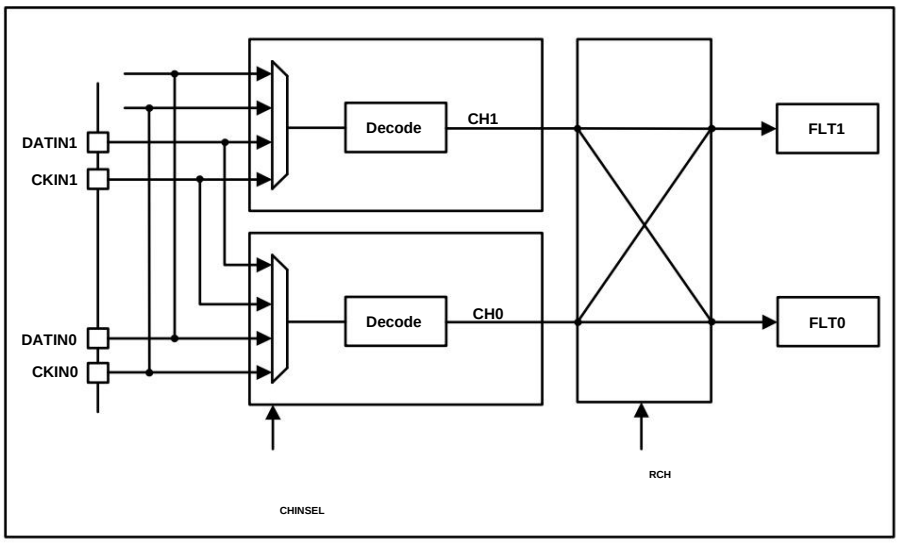
Configuration of PDM microphone input serial channel:

The PDM microphone signal (data, clock) will be connected to the DFSDM input serial channel y (DATINy, CKOUT) pins . Channel y will be configured as follows: CHINSEL=0 (input from given channel pins: DATINy and CKINy) . Channel (y-1) (modulo 2) will be configured as follows:

- CHINSEL=1 (Input from subsequent channel ((y-1)+1) pins: DATINy and CKINy)

ÿ Channel y: SITP=0 (rising edge strobe data), i.e., the left audio channel of channel y. ÿ Channel (y-1): SITP=1 (falling edge strobe data), i.e., the right audio channel of channel y-1. ÿ Two DFSDM filters will be assigned to channel y and channel (y-1) (used to filter the left and right channels of the PDM microphone).

Figure 42-2 Input Channel Pin Redirection



A **clock signal**

output is provided on the CKOUT pin to drive the external ÿÿ modulator clock input. The frequency of the CKOUT signal is obtained from the DFSDM clock or the audio clock (refer to the CKOUTSRC bit in the R32_DFSDM_CH0CFGR1 register) via a prescaler (refer to the CKOUTDIV bit in the R32_DFSDM_CH0CFGR1 register). If the output clock stops, the CKOUT signal is set low (the output clock can be stopped by setting CKOUTDIV=0 in the R32_DFSDM_CH0CFGR1 register or setting DFSDMEN=0 in the R32_DFSDM_CH0CFGR1 register). The output clock stops at the following times: ÿ When CKOUTSRC=0, after 4 system clock cycles after DFSDMEN is cleared ÿ When CKOUTSRC=1, after 1 system clock cycle and 3 audio clock cycles after DFSDMEN is cleared .

Before changing CKOUTSRC, the software must wait for CKOUT to stop to avoid generating glitches on the CKOUT pin. The output clock signal frequency must be in the range of 0 to 20 MHz.

In **SPI data input format**

operation , the data stream is transmitted serially via data and clock signals. The data signal is always provided by the DATINy pin. The clock signal can be provided externally via the CKINy pin or internally via a signal taken from the CKOUT signal source. If an external clock source is selected (SPICKSEL[1:0]=0), the SITP bit in the R32_DFSDM_CHyCFGR1 register is used. The data signal is sampled on the rising or falling edge of the clock on the (CKINy) pin.

Internal clock source—see SPICKSEL[1:0] in the R32_DFSDM_CHyCFGR1 register: ÿ CKOUT signal: - Used to

connect an external ÿÿ modulator, which directly uses its clock input (from CKOUT) to generate the output serial communication. clock.

- Sampling point: rising edge/falling edge, depending on SITP settings. ÿ CKOUT/2 signal (generated on the rising edge of CKOUT):

- Used to connect an external ÿÿ modulator, which divides the clock input (from CKOUT) by 2 to generate the output serial communication. Clock (this output clock change is valid on the rising edge of each clock input). - Sampling point: every second falling edge of CKOUT. ÿ CKOUT/2 signal (generated on the falling edge of CKOUT):

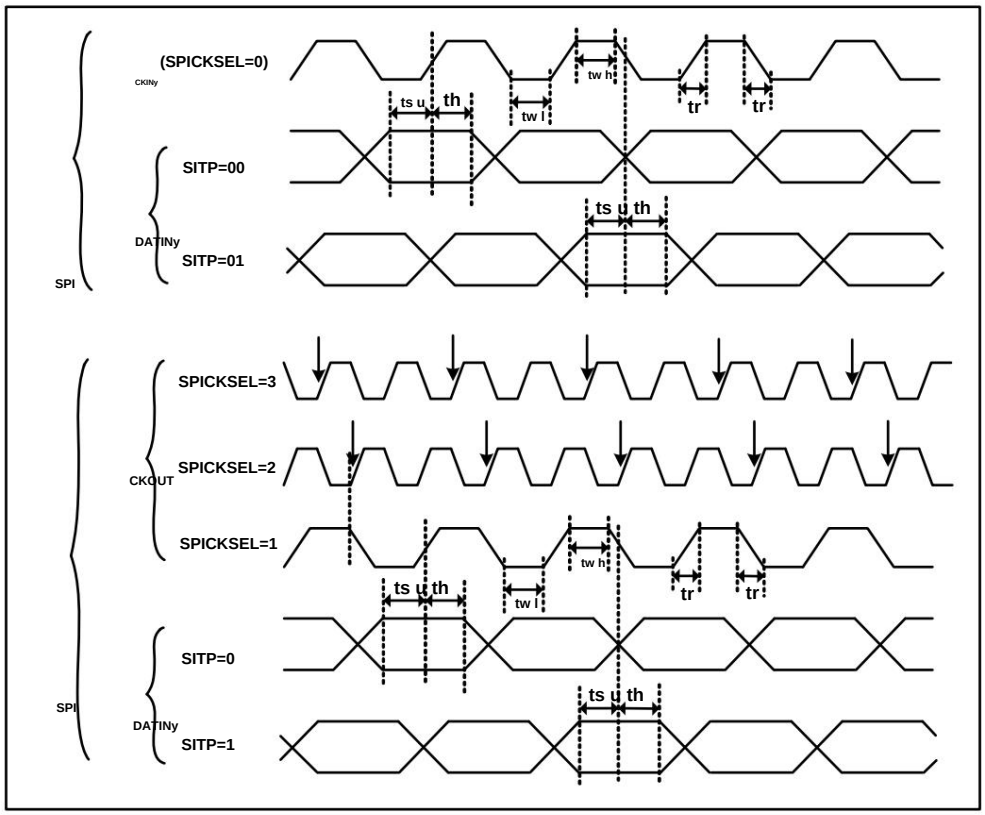
- Used to connect an external ÿÿ modulator, which divides the clock input (from CKOUT) by 2 to generate the output serial communication. Clock (This output clock change is valid on the falling edge of each clock input). - Sampling point: Every second rising edge of CKOUT.

Note: Only external ÿÿ The modulator only activates when it uses the CKOUT signal as a clock input (to achieve synchronized clock and data operations).

It can use an internal clock source. Using an internal clock source eliminates the CKINy For pin connections (the CKINy pin can be used for other purposes). If...

SPI encoding, the clock source signal frequency must be in the range of 0 to 20 MHz and less than fDFSDMCLK/4.

Figure 42-3 Timing diagram of the channel transceiver



The **clock missing**

detection function can detect the absence or presence of clock at the channel serial clock input, ensuring the proper functioning of the conversion process and error reporting.

Clock miss detection is enabled or disabled on each input channel y via the CKABEN bit of the R32_DFSDM_CHyCFGR1 register. If enabled...

If a clock loss detection is performed on a given channel, clock loss detection will be performed continuously on that channel. If an input clock error occurs, a clock loss will occur.

The flag will be set to 1 (CKABF[y]=1), and an interrupt will be invoked (when CKABIE=1). The clock missing flag will be cleared (R32_DFSDM_FLT0ICR).

After the register CLRCKABF is updated, the clock missing flag will be refreshed. When the corresponding channel y is disabled, the clock missing status bit CKABF[y] will still be active.

It will be set to 1 by the hardware (if CHEN[y]=0, then CKABF[y] will remain in the set state).

In the event of a clock failure, data conversion (and/or analog watchdog and short-circuit detectors) will provide erroneous data. (In the report...)

When the clock is missing, the user should handle the event and discard the relevant data.

The clock missing function only applies when the system clock is used for the CKOUT signal (CKOUTSRC=0 in the R32_DFSDM_CH0CFGR1 register).

Available at any time.

When the transceiver is not synchronized, the clock missing flag will be set, and the clock will not be able to be accessed via the R32_DFSDM_FLT0ICR register.

The CLRCKABF[y] bit is cleared. The software operation sequence related to the clock loss detection function should be as follows:

Enable a specified channel by setting CHEN=1 .

ÿ Attempt to clear the clock missing flag (by setting CLRCKABF=1) until the clock missing flag is actually cleared (CKABF=0).

At that time, the transceiver is synchronized (signal clock is valid) and can receive data.

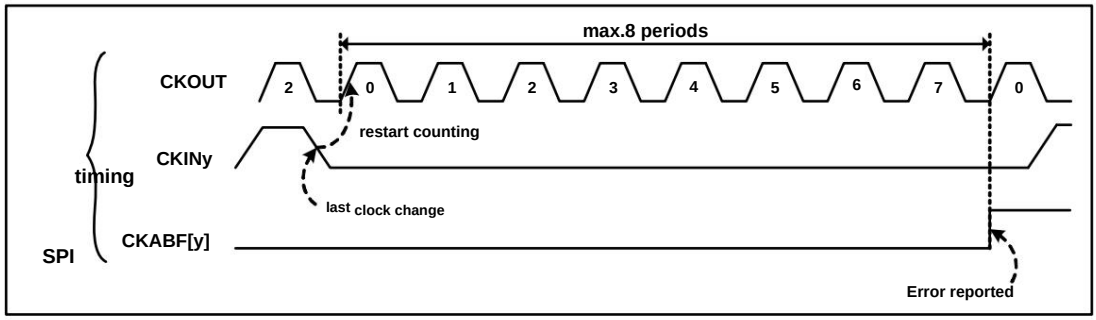
Enable the clock loss function CKABEN=1 and the associated interrupt CKABIE=1 to detect if the SPI clock is lost.

If the SPI data format is used, clock loss detection will be based on comparing the CKOUT signal generated by the external input clock and the output clock.

The external input clock signal in the input channel must pass through the CKOUT signal (by register CKOUTDIV of R32_DFSDM_CH0CFGR1).

The field control changes once every 8 signal cycles.

Figure 42-4 SPI Clock Missing Timing Diagram



If an input clock recovery error occurs, the clock missing flag will be set to 1 (CKABF[y]=1), and an interrupt will be invoked (CKABIE=1).

The clock missing flag is refreshed after it is cleared (via CLRCKABF in the R32_DFSDM_FLT0ICR register).

External serial clock frequency

measurement allows for real-time acquisition of the data rate from an external ÿÿ modulator by measuring the serial clock input frequency of the measurement channel. This is crucial for applications...

Purpose is of paramount importance.

The external serial clock input frequency can be timed by counting the DFSDM clock (fDFSDMCLK) over a transition duration.

The counter is used for measurement. The counting begins at the first input data clock after the conversion trigger (normal or injected) and ends at the end of the conversion (conversion termination).

The conversion ends at the last input data clock before the bundle flag is set to 1. When the conversion is complete (JEOCF=1 or REOCF=1), it will be recorded in the register.

The counter CNVCNT[27:0] in R32_DFSDM_FLTxCNVTIMR updates the time of each conversion (first serial sample and last serial sample).

(Time between serial samples). The user can then calculate the data rate based on the digital filter settings (FORD, FOSR, IOSR, FAST).

Rate. Only when the filter is bypassed (FOSR=0, only the integrator is active, R32_DFSDM_FLTxCNVTIMR register CNVCNT[27:0]=0)

The external serial frequency measurement will only stop when the time is right.

For parallel data input (see Section 42.3.4: Parallel Data Input), the frequency of the measurement represents the average frequency during one conversion.

Average input data rate.

Note: When a conversion is interrupted (e.g., by disabling/enabling the selected channel), the interruption time is also counted in CNVCNT[27:0]. Therefore, it is recommended not to interrupt the conversion to obtain accurate conversion time results. It is recommended to measure the clock in continuous fast mode.

Conversion time:

Injection conversion or regular conversion when FAST=0 (or initial conversion when FAST=1):

For Sincx filters (x=1..5):

$$t=CNVCNT/fDFSDMCLK=[FOSR*ÿ2\ IOSR+FORD+1ÿ+2]/fCKIN$$

For FastSinc filters:

$$t=CNVCNT/fDFSDMCLK=[FOSR*ÿFOSR*ÿ2\ IOSR+5ÿ+2ÿ]/fCKIN$$

Regular conversions when FAST=1 (except for the first conversion):

For Sincx and FastSinc filters:

$$t=CNVCNT/fDFSDMCLK=[FOSR*2IOSR]/fCKIN$$

If FOSR = FOSR[9:0] + 1 = 1 (filter bypass, only integrator is effective), then:

$$t=2IOSR/fCKIN\ (...but\ CNVCNT=0)$$

in:

1) fCKIN is the channel input clock frequency (on the given channel CKINy pin) or the input data rate (when parallel data is input);

2) FOSR is the filter oversampling rate: FOSR = FOSR[9:0] + 1 (refer to the R32_DFSDM_FLTxFCR register);

3) 2 IOSR is the integrator oversampling rate (refer to the R32_DFSDM_FLTxFCR register);

4) FORD is the filter order: FORD=FORD[2:0] (refer to the R32_DFSDM_FLTxFCR register).

Channel offset settings

Each channel has its own offset setting (in a register), and this offset value will ultimately be derived from the various conversion results of a given channel (Note:
Subtract from (either input or normal). Perform offset correction after right shifting the data. The offset value is stored as a 24-bit signed value.
In the OFFSET[23:0] field of the R32_DFSDM_CHyCFGR2 register.

To align the data

right shift with a 24-bit value, a series of right shifts are defined for each channel. These right shifts will be applied to each bit of a given channel.
The conversion result (injected or normal). The right-shifted bits are stored in bits DTRBS[4:0] of the R32_DFSDM_CHyCFGR2 register.

A right shift rounds the result to the nearest integer value. The sign of the shifted result is preserved to obtain a valid 24-bit signed byte.
The result data of the formula.

42.3.3 Configuring the Input Serial Interface The

following parameters must be configured for the input serial interface:
Output clock prescaler. An output clock (range 2 to 256) can be generated from a DFSDM clock using a programmable prescaler.
This setting is determined by the CKOUTDIV[7:0] bits in the R32_DFSDM_CH0CFGR1 register.
• Serial interface type and input clock phase. Select the SPI interface and the sampling edge of the input clock. This option is determined by...
The SITP bit in the R32_DFSDM_CHyCFGR1 register specifies this.
ÿ Input clock source selection. You can select either an external source from the CKINy pin or an internal source from the CKOUT pin. This selection is determined by...
The SPICKSEL[1:0] field in the R32_DFSDM_CHyCFGR1 register determines this.
ÿ Final data right shift setting. Defines the number of bits to shift the final data right to ensure the result is aligned with a 24-bit value. This configuration is set by...
The DTRBS[4:0] bits are defined in the R32_DFSDM_CHyCFGR2 register.
ÿChannel Offset Configuration. Sets the analog offset (i.e., the offset of the connected external ÿÿ modulator) for a specific serial channel. This value is determined by...
The OFFSET[23:0] bits in the R32_DFSDM_CHyCFGR2 register are set.
• Enable channel short-circuit detector and clock loss function. Enable via the SCDEN bit in the R32_DFSDM_CHyCFGR1 register.
Alternatively, disable the short-circuit detector for a given serial channel by enabling or disabling clock loss detection via the CKABEN bit.
ÿSimulated watchdog filter and short-circuit detector threshold configuration. Used to set the channel simulated watchdog filter parameters and channel short-circuit detection.
Device parameters. These configurations are defined in the R32_DFSDM_CHyAWSCDR register.

42.3.4 Parallel Data Input

Each input channel provides a register for 16-bit parallel data input (except for serial data input).
Input can only come from internal data sources:
Internal ADC results (ADC1 and ADC2 only)
Direct CPU/DMA write operations

The DATMPX[1:0] fields of the R32_DFSDM_CHyCFGR1 register are used to select whether to use serial or parallel data for a given channel.
Input. DATMPX[1:0] also defines parallel data sources: internal ADC or write operations performed directly by CPU/DMA.

Each channel includes a 32-bit data input register (R32_DFSDM_CHyDATINR), into which 16 bits can be written.
The data is in a 16-bit signed format. This data can also be used as input to digital filters that accept 16-bit parallel data.

If serial data input is selected (DATMPX[1:0]=0), the R32_DFSDM_CHyDATINR register is write-protected.

Input from internal ADCs (ADC1 and ADC2 only)

For parallel ADC data input (DATMPX[1:0]=1), the ADC[y+1] result is assigned to the channel y input (ADC1 fill).
The R32_DFSDM_CH0DATINR register is filled by ADC2, and the R32_DFSDM_CH1DATINR register is filled by ADC2. The conversion from ADC[y+1] ends.
This will cause the data in channel y to be updated (parallel data from ADC[y+1] is used as the next sample from the digital filter). The conversion ends.
When the event occurs, data from ADC[y+1] is written to the R32_DFSDM_CHyDATINR register (INDAT0[15:0] fields).

The data encapsulation mode setting (DATPACK[1:0] in the R32_DFSDM_CHyCFGR1 register) has no effect on the ADC data input.

Note: ADC extended specification: internal ADC When configured in interleaved mode (e.g., ADC1 with interleaved mode - see reference) ADC2 ADC (Standards)

Each result of the auto-OR operation is transmitted to the same bus (i.e., the bus connected to the DFSDM channel ADC1 ADC2 16 ADC1 0 (Fixed connection). Therefore, there will be double the input data rate in the DFSDM channel (even samples from ADC1, odd samples from...). ADC2). The associated channel will be idle.

Input from memory (written directly by CPU/DMA)

When the CPU or DMA (DATMPX[1:0]=2) directly writes data to the R32_DFSDM_CHyDATINR register, this data can...
As input, it is used to process digital data streams from memory or peripherals.
Data can be written to the R32_DFSDM_CHyDATINR register by the CPU or DMA:

1) CPU data writing:

Input data is written directly by the CPU into the R32_DFSDM_CHyDATINR register.

2) DMA data writing:

DMA should be configured for memory-to-memory transfer mode. , To transfer data from the memory buffer to
The target memory address is the address of the R32_DFSDM_CHyDATINR register. Data is transmitted via DMA.
The transfer speed is from the memory to the DFSDM parallel input.

This DMA is different from the DMA used to read the DFSDM conversion results. These two DMAs can be used simultaneously; the first DMA (configured as...)
The first DMA (Memory-to-Memory Transfer) is used for input data writing, and the second DMA (Configured for Peripheral-to-Memory Transfer) is used for data result reading.
Pick.

Access to R32_DFSDM_CHyDATINR can be 16-bit or 32-bit wide, allowing one or two to be loaded in a single write operation.
Sampling. The 32-bit input data register (R32_DFSDM_CHyDATINR) can be filled with one or two 16-bit data samples, specifically...
The data encapsulation operation mode depends on the DATPACK[1:0] field defined in the R32_DFSDM_CHyCFGR1 register:

1) Standard Mode (DATPACK[1:0]=00b):

Only one sample is stored in the INDAT0[15:0] field of the R32_DFSDM_CHyDATINR register, which is used as the input of channel y.
Input data. The high 16 bits (INDAT1[15:0]) are ignored and write-protected. The digital filter must perform one input sampling (via...)
INDAT0[15:0]), in order to clear the data registers filled by CPU/DMA. This mode is related to the R32_DFSDM_CHyDATINR register.
The device's 16-bit CPU/DMA access is used in conjunction to load a sample on each write operation.

2) Interleaved mode (DATPACK[1:0]=01b):

The R32_DFSDM_CHyDATINR register is used as two sampling buffers. The first sample is stored in INDAT0[15:0], and the second...
Each sample is stored in INDAT1[15:0]. The digital filter must perform two input samples through channel y to clear the data. ,
The R32_DFSDM_CHyDATINR register. This mode is combined with 32-bit CPU/DMA access to the R32_DFSDM_CHyDATINR register.
Used to load two samples on each write operation.

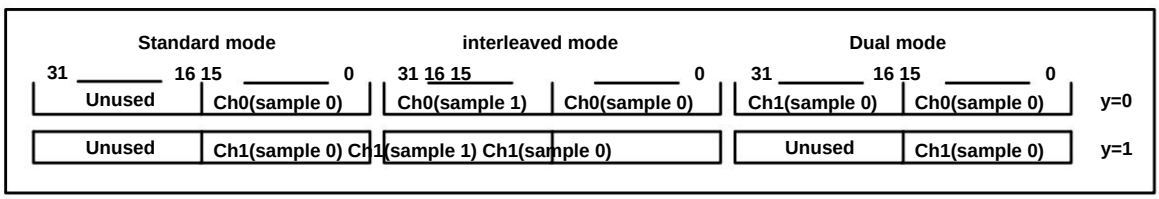
3) Dual-channel mode (DATPACK[1:0]=10b):

Two samples are written to the R32_DFSDM_CHyDATINR register. The data in INDAT0[15:0] is used for channel y.
The data in INDAT1[15:0] is used for channel y+1. The data in INDAT1[15:0] will be automatically copied to the next (y+1) channel data register.
The data is stored in INDAT0[15:0] of the register DFSDM_CH[y+1]DATINR. The digital filter must perform two samplings (once from channel y, and once from channel y).
Another time from channel (y+1) to clear the R32_DFSDM_CHyDATINR register.

Dual-channel mode can only be set for even-numbered channels (y=0) (DATPACK[1:0]=10b). If odd-numbered channels (y=1) are used...
If set to dual-channel mode, then for that channel, the INDAT0[15:0] and INDAT1[15:0] sections are write-protected. If an even-numbered channel is set to...
In dual-channel mode, the next odd-numbered channel must be set to standard mode (DATPACK[1:0]=00b) in order to properly match the even-numbered channel.

For information on the R32_DFSDM_CHyDATINR register data mode and channel data sample allocation, please refer to the following figure.

Figure 42-5 R32_DFSDM_CHyDATINR Register Operation Mode and Allocation



First, enable the selected input channel (channel y) to perform data acquisition (start channel y conversion), then you must...
The R32_DFSDM_CHyDATINR register is written to load one or two samples. Otherwise, the write operation will be lost during the next processing.
The data.

For example: For single-transformation and interleaved modes, do not start writing to R32_DFSDM_CHyDATINR before initiating a single-transformation.
All data in the data sample (that exists in R32_DFSDM_CHyDATINR before the conversion begins) will be discarded.

42.3.5 Channel Selection: There

are two multiplexed channels that can be selected for injection channel group conversion and/or regular channel conversion.

The injection channel group can select any one of the two channels or all of them. The R32_DFSDM_FLTxJCHGR register contains...
JCHG[1:0] selects the injection group channel, where JCHG[y]=1 indicates that channel y has been selected.

Injection transformation can operate in scan mode (JSCAN=1) or single mode (JSCAN=0). In scan mode, each selected channel...
The channels will be switched sequentially. The lowest channel (channel 0, if selected) will be switched first, followed by the next higher channel, and so on.
All channels selected by JCHG[1:0] are converted. In single mode (JSCAN=0), only the selected channel is converted.
The selection will move to the next channel. When JSCAN=0, writing to JCHG[1:0] will reset the channel selection to the lowest selected channel.

Injected transformations can be initiated by software or triggers. Such transformations are never interrupted by regular transformations.

The normal channel selects only one of the two channels. The RCH bit in the R32_DFSDM_FLTxCR1 register indicates the selected channel.

Regular transformations can only be initiated by software (not by triggers). Requesting an injection transformation will temporarily interrupt the continuous sequence of regular transformations.
List.

Performing a conversion on a disabled channel (CHEN=0 in the R32_DFSDM_CHyCFGR1 register) will cause the conversion to never finish.
This is because no input data is provided (no clock signal). In this case, the given channel nęeds to be enabled.
(CHEN=1 in the R32_DFSDM_CHyCFGR1 register), or stop the conversion (DFEN=0 in the R32_DFSDM_FLTxCR1 register).

42.3.6 Digital Filter Configuration

DFSDM includes a Sincx-type digital filter implementation. This Sincx filter performs filtering on the input digital data stream, which results in a reduction of...
Increase the output data rate (decimation) and enhance the output data resolution. Sincx digital filters can be configured to achieve the desired output.
Data rate and required output data resolution. Configurable parameters include:

ÿ Filter order/type: (Refer to the FORD[2:0] bits in the R32_DFSDM_FLTxFCR register):

- FastSinc
- Sinc1
- Sinc2
- Sinc3
- Sinc4
- Sinc5

ÿ Filter oversampling/decimation rate: (Refer to bits FOSR[9:0] in the R32_DFSDM_FLTxFCR register):

- FOSR=1-1024 — Applicable to FastSinc and Sincx filters x=FORD=1..3 -

FOSR=1-215 — Applicable to Sincx filters x=FORD=4 -

FOSR=1-73 — Applicable to Sincx filters x=FORD=5

Filters support the following transfer functions (impulse response in the H domain):
Sincx filter types:

$$H(z) == \frac{\&>?@ABCD^{2}}{\&>E@F\ G}$$

ÿ FastSinc filter type:

$$H(z) == \frac{\&>?@ABCD^{\$}}{\&>?@F\ G} H\ (1 + z> (\$HIJKL))$$

Figure 42-6 Example: Sinc3 filter response

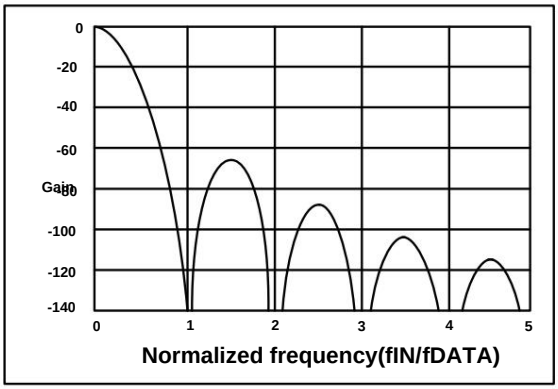


Table 42-1 Maximum output resolution of the filter (peak data values from the filter output), based on certain FOSR values

FOSR	Sinc1	Sinc2	FastSinc	Sinc3	Sinc4	Sinc5
x	+/-x	+/-x2	+/-2x2	+/-x3	+/-x4	+/-x5
4	+/-4	+/-16	+/-32	+/-64	+/-256	+/-1024
8	+/-8	+/-64	+/-128	+/-512	+/-4096	-
32	+/-32	+/-1024	+/-2048	+/-32768	+/-1048576	+/-33554432
64	+/-64	+/-4096	+/-8192	+/-262144	+/-16777216	+/-1073741824
128	+/-128	+/-16384	+/-32768	+/-2097152	+/268435456	
256	+/-256	+/-65536	+/-131072	+/16777216 Under full-scale input conditions, the result will be...		
1024	+/-1024	+/-1048576	+/-2097152	+/1073741824 Overflow (>32-bit signed integer value)		

42.3.7 Integrator Unit The

integrator further improves the decimation rate and resolution of data from the digital filter. For a given number of data from the filter...

Based on the sampling, the integrator performs a simple summation.

The integrator oversampling rate parameter defines how many data points' sums are used as a single data output from the integrator. IOSR can be set to a value between [value missing].

Within the range of 1 to 256 (refer to the description of the IOSR[3:0] bits in the R32_DFSDM_FLTxFCR register).

Table 42-2 Maximum output resolution of the integrator (peak data value from the integrator output).

Based on certain IOSR values, FOSR=256, and the Sinc3 filter type (maximum data).

IOSR	Sinc1	Sinc2	FastSinc	Sinc3	Sinc4	Sinc5
x	+/-FOSR.x	+/-FOSR2 .x	+/-2FOSR2 .x	+/-FOSR3 .x	+/-FOSR4 .x	+/-FOSR5 .x
4	-	-	-	+/-67108864	-	-
32	-	-	-	+/-536870912	-	-
128	-	-	-	+/-2147483648	-	-
256	-	-	-	+/-232	-	-

42.3.8 Simulated Watchdog Timer A

simulated watchdog timer is used to trigger an external signal (disconnect) when a given upper threshold is reached or exceeded, or a given lower threshold is reached or fallen below.

(Circuit or interrupt). An interrupt/event/circuit break can then be invoked to generate the circuit.

Each analog watchdog timer monitors the serial data receiver output based on the AWFSEL bit in the R32_DFSDM_FLTxCR1 register.

Output (after simulating a watchdog filter on each channel) or data output register (current injection or normal conversion result). Is it necessary?

The input channel to be monitored via the simulated watchdog x can be selected via AWDCH[1:0] in the R32_DFSDM_FLTxCR2 register.

The analog watchdog transition on the input channels is independent of standard transition. In standard transition mode, the analog watchdog transition occurs on each input channel.

It uses its own filters and signal processing, independent of injection or conventional master conversion. In continuous conversion mode, it performs processing on the selected input channel.

Perform a simulated watchdog transition so that the channel can still be monitored even when injection or regular master transition is paused (RCIP=0, JCIP=0).

The upper and lower threshold registers are compared with a given data value (set via bits AWHT[23:0] of the R32_DFSDM_FLTxAWHTR register and bits AWLT[23:0] of the R32_DFSDM_FLTxAWLTR register). Two options are available for comparing the

threshold registers with the data value:

Option 1: In this case, the input data is taken from the final output data register (AWFSEL=0). This option has the following characteristics:

- High input data resolution (up to 24 bits) - Slow response

time – not suitable for fast-response applications such as overcurrent detection - The

final data requires shift and offset data correction for comparison - The final data is only

available after performing a regular or injected master conversion - Available

with parallel input data source (DATMPX[1:0] ÿ 0 in the R32_DFSDM_CHyCFGR1 register) ÿ Option 2: In this case, the input data is taken from the output of either serial data receiver (AWFSEL=1). This option has the following characteristics: - The input serial data is processed through a dedicated analog

watchdog Sincx channel filter, and the filter oversampling rate (1..32) and filter order (1..3) can be configured (see the AWFOSR[4:0] and AWFORD[1:0] bit settings in the R32_DFSDM_CHyAWSCDR register)

- Lower resolution (up to 16 bits) - Fast response

time - Suitable for applications requiring fast response, such as overcurrent/overvoltage detection - Data

is provided in continuous mode, independent of regular or injected master conversion

activities. When monitoring the input channel (AWFSEL=1), the data used for comparison with the threshold is taken from the AWDCH[1:0] field (R32_DFSDM_FLTxCR2 register) of the selected channel. The filter result for each selected channel is compared with a threshold pair (AWHT[23:0]/AWLT[23:0]). At this time, only the high 16 bits (AWHT[23:8]/AWLT[23:8]) are defined with the analog...

The watchdog filter output compares a 16-bit threshold because the data from the analog watchdog filter has a maximum resolution of 16 bits. In this case (AWFSEL=1), the AWHT[7:0]/AWLT[7:0] bits are not compared. The parameters (filter order) of the analog

watchdog filter configuration for each input channel are set in the R32_DFSDM_CHyAWSCDR register.

The number of AWFORD[1:0] and the filter oversampling rate AWFOSR[4:0]).

The analog watchdog filter data for a given channel y can be read from the WDATA[15:0] fields of the DFSDM_CHyWDATR register via firmware. If CHEN=1 in the R32_DFSDM_CHyCFGR1 register, the analog watchdog filter data is continuously converted, with the data rate given by the analog watchdog filter settings and the channel input clock frequency. The analog watchdog filter conversion

operates similarly to a regular fast continuous conversion (without an integrator). Analog watchdog filter

Output (channel input clock frequency is fCKIN) the number of serial samples required to produce one result:

First conversion:

For Sincx filters (x=1..5): Number of samples = [FOSR*(FORD+1)+2] For FastSinc filters:

Number of samples = [FOSR*4+2+1] Subsequent conversion: For

Sincx and

FastSinc filters: Number of samples = [FOSR*FORD] Where: FOSR is the filter

oversampling rate: FOSR=AWFOSR[4:0]+1 (refer to the R32_DFSDM_CHyAWSCDR register) FORD is the filter order:

FORD=AWFORD[1:0] (refer to the R32_DFSDM_CHyAWSCDR register) For the output data register monitor

(AWFSEL=0), the comparison is performed after the final right shift and offset correction of the data are completed (refer to the OFFSET[23:0] and DTRBS[4:0] fields of the R32_DFSDM_CHyCFGR2 register). The comparison is performed after each injection or regular conversion of the channel selected by the AWDCH[1:0] field (in the R32_DFSDM_FLTxCR2 register). The state of the simulated watchdog event is reflected in the R32_DFSDM_FLTxAWSR register, which locks the given event.

The AWHTF[y]=1 flag indicates whether the AWHT[23:0] value has been exceeded on channel y. The AWLTF[y]=1 flag indicates whether the AWLT[23:0] value has been exceeded on channel y. For events latched in the R32_DFSDM_FLTxAWSR register, they are cleared by writing 1 to the corresponding clear bit CLRAWHTF[y] or CLRAWLTF[y] in the R32_DFSDM_FLTxAWCFR register.

The global state of the analog watchdog is reflected by the AWDF flag in the R32_DFSDM_FLTxISR register (used for fast interrupt source detection). AWDF=1 indicates that at least one watchdog event has occurred (AWHTF[y]=1 or AAWTF[y]=1 for at least one channel). The AWDF bit is cleared only when all AWHTF[1:0] and AAWTF[1:0] are cleared. Analog watchdog

events can be assigned to disconnected output signals. Two disconnected outputs can be assigned to events exceeding the upper threshold or falling below the lower threshold (DFSDM_BK[1:0]). Disconnected signals are assigned to a given analog watchdog event via the BKAWH[1:0] and BKAWL[1:0] fields in the R32_DFSDM_FLTxAWHTR and R32_DFSDM_FLTxAWLTR registers.

42.3.9 Short Circuit Detector The

purpose of a short circuit detector is to emit a signal within a very fast response time when an analog signal reaches its saturation value (outside full scale) and remains there for a given time. This feature can detect short circuits or open circuit faults (such as overcurrent or overvoltage). Interrupt/event/open circuit generation can be invoked.

The input data for the short-circuit detector is taken from the

channel transceiver output. Each input channel has an incrementing counter that counts consecutive 0s or 1s on the serial data receiver output. The counter is restarted if the received data stream changes (data signal from 1 to 0 or from 0 to 1). A short-circuit event is triggered if the counter reaches the short-circuit threshold register value (SCDT[7:0] bits in the R32_DFSDM_CHyAWSCDR register). Each input channel is equipped with a short-circuit detector. Continuous monitoring of any channel can be selected by setting the SCDEN bit (in the R32_DFSDM_CHyCFGR1 register) to 1. Each channel has its own short-circuit detector settings (threshold bits SCDT[7:0], status bits SCDF[1:0], and status clear bits CLRSCDF[1:0]). The status flag SCDF[y] can also be cleared by hardware when the corresponding channel y is disabled (CHEN[y]=0). For each channel, short-

circuit detector events can be assigned to open-circuit output signals DFSDM_BK[1:0]. Two open-circuit outputs can be assigned to short-circuit detector events. Open-circuit signals are assigned to short-circuit detector events for a given channel via the BKSCD[1:0] field in the R32_DFSDM_CHyAWSCDR register.

If the parallel input data channel is selected (DATMPX[1:0] in the R32_DFSDM_CHyCFGR1 register ÿ 0), then it is impossible to make Use a short-circuit detector.

There are a total of two disconnect outputs (shared with the simulated watchdog function).

42.3.10 Extreme Value Detector

The extreme value detector is used to acquire the minimum and maximum values (peak-to-peak

values) of the final output data word. If the output data word is higher than the value stored in the extreme value detector's maximum value register (the EXMAX[23:0] bits in the R32_DFSDM_FLTxEXMAX register), the register will be updated with the current output data word value, and the channel storing its data is located in the EXMAXCH bit (of the R32_DFSDM_FLTxEXMAX register).

If the output data word is lower than the value stored in the minimum value register of the extreme value detector (the EXMIN[23:0] bits in the R32_DFSDM_FLTxEXMIN register), the register will be updated with the current output data word value, and the channel storing its data is located in the EXMINCH bit (of the R32_DFSDM_FLTxEXMIN register).

The minimum and maximum value register values can be refreshed by software (by reading the R32_DFSDM_FLTxEXMAX or R32_DFSDM_FLTxEXMIN registers). After refreshing, the minimum value data register (R32_DFSDM_FLTxEXMIN) of the extreme value detector is filled with 0x7FFFFF (maximum positive value), and the maximum value register (R32_DFSDM_FLTxEXMAX) of the extreme value detector is filled with

0x800000 (minimum negative value). The extreme value detector performs the comparison operation only after completing the right shift and offset data correction. For each extreme value detector, the input channels involved in calculating the extreme value are selected by the EXCH[1:0] bits (in the R32_DFSDM_

42.3.11 Data Unit Module

The data unit module is the last module in the entire processing path: external ÿÿ modulator — serial transceiver — Sinc filter Integrator—Data Unit Block.

The output data rate depends on the serial data stream rate and the filter and integrator settings. The maximum output data rate is:

Data rate [samples/second] = MNOPQ / IBCDR($ 6S4TUIBDVU&)U$ÿFAST=0ÿSincx filter

Data rate [samples/second] = MNOPQ / IBCDR($ 6S4TU*)U$ÿFAST=0ÿFastSinc filter

or

Data rate [samples/second] = MNOPQ / IBCDR$6S4TÿFAST=1

Maximum output data rate with parallel data input:

Data rate [samples/second] = MVWXWPQ_DWXY / IBCDR($ 6S4TUIBDVU&)U$ÿFAST=0ÿSincx filter

or

Data rate [samples/second] = MVWXWPQ_DWXY / IBCDR($ 6S4TU*)U$ÿFAST=0ÿFastSinc filter

or

Data rate [samples/second] = MVWXWPQ_DWXY / IBCDR$6S4T FAST=1 or any bypass case (FOSR=1)

Where fDATAIN_RATE is the input data rate of the ADC or CPU/

DMA. A right shift of the final data is performed in this module because the final data width is 24 bits, while the data from the processing path can be up to 32 bits. This right shift can be configured in the range of 0-31 bits for each selected input channel (refer to bits DTRBS[4:0] of the R32_DFSDM_CHyCFGR2 register). The right shift rounds the result to the nearest integer value. The sign of the shift result is preserved to obtain a valid 24-

bit signed result data. Next, offset correction is performed on the result. The offset correction value (stored in OFFSET[23:0] in the R32_DFSDM_CHyCFGR2 register) is subtracted from the output data of the given channel. The data in the OFFSET[23:0]

field is set by the software through the appropriate calibration procedure. Since all operations in the digital processing are performed on a 32-

bit signed register, the following conditions must be met to ensure

that the result does not overflow: 1) FOSRFORD.2IOSRÿ2 31, applicable to Sincx filters (x=1..5); 2) FOSR2 .2IOSRÿ2 31, applicable to FastSi

Note: When the filter and integrator are bypassed (IOSR[3:0]=0, FOSR[9:0]=0), the input data rate (fDATAIN_RATE) must be limited to be able to read all output data: fDATAIN_RATEÿfHCLK; where fHCLK is the bus frequency of the DFSDM peripheral connection.

42.3.12 Signed Data Format Each

channel of the DFSDM input serial channel can be connected to an external ÿÿ modulator. An external ÿÿ modulator is configured with two differential inputs (positive and negative), which can be used for both differential and single-

ended signal measurements. It is always assumed that the output of the ÿÿ modulator is in signed format (i.e., the 0/1 data stream from

the ÿÿ modulator represents the values -1 and +1). Signed data format in registers: Data stored in the final output data, analog watchdog, extreme value detector, and offset correction registers are all in signed format. The most significant bit (msb) of the output data word is used to represent the sign of the value (using two's complement format).

42.3.13 The method for

initiating the conversion

injection is as follows: ÿ Software: Write 1 to JSWSTART in the R32_DFSDM_FLTxCR1 register. ÿ Trigger: With JEXTEN active, select the trigger signal through JEXTSEL[3:0] and determine the valid edge (refer to the R32_DFSDM_FLTxCR1 register).

If JSYNC is set to 1, DFSDM_FLT0 will synchronously initiate injection conversions: For DFSDM_FLTx (x>0), injection conversions will automatically begin in DFSDM_FLT0; injection conversions are software-triggered (by setting the JSWSTART bit to 1 in the R32_DFSDM_FLT0CR2 register). Each injection conversion in DFSDM_FLTx (x>0) will be executed according to its local settings (such as JSCAN and J

Only one injection conversion is allowed within a specific time period. Therefore, if an injection conversion request has already been issued but has not yet been completed, any subsequent injection conversions will be blocked. All subsequent startup requests will be ignored.

The following methods are used to initiate a regular conversion: ÿ Software: Write 1 to RSWSTART in the R32_DFSDM_FLTxCR1 register. ÿ If RSYNC=1, then use DFSDM_FLT0 synchronous startup: For DFSDM_FLTx (x>0), regular conversion will automatically start in DFSDM_FLT0; regular conversion is triggered by software (by setting the RSWSTARTT bit to 1 in the R32_DFSDM_FLT0CR2 register). Each regular transformation in DFSDM_FLTx (x>0) will be performed according to its local configuration settings (RCONT and RCH, etc.).

42.3.14 Continuous and Fast Continuous Modes

Setting RCONT to 1 in the R32_DFSDM_FLTxCR1 register will perform regular conversions in continuous mode. RCONT=1 indicates that after writing 1 to RSWSTART, the channel selected by the RCH bit will be repeatedly converted. Regular conversions in continuous mode can be stopped by writing 0 to RCONT. Clearing RCONT will immediately stop the ongoing conversion. In continuous mode, the data rate can be increased by setting the FAST bit in the R32_DFSDM_FLTxCR1 register to 1. In this case, if a channel is continuously converted, the filter does not need to be refilled with new refresh data because the internal data of the filter is valid, which is the data obtained from previous sampling of continuous data. The speed increase is related to the selected filter order. After starting continuous conversion with RSWSTART=1, the first conversion in fast mode (FAST=1) will be performed fully (the same as when FAST=0), and subsequent conversions will be completed in short intervals. Conversion time in continuous mode: when FAST=0 (or fast conversion when FAST=1): For Sincx filters:

t=CNVCNT/fDFSDMCLK=[FOSR*ÿ2 IOSR+FORD+1ÿ+2]/fCKIN

For FastSinc filters:

t=CNVCNT/fDFSDMCLK=[FOSR*ÿ2 IOSR+5ÿ+2]/fCKIN

When FAST=1 (except for the first conversion): For Sincx and FastSinc filters:

t=CNVCNT/fDFSDMCLK=[FOSR*2IOSR]/fCKIN When FOSR=FOSR[9:0]+1=1 (filter bypass, only integrator is effective): t=2IOSR/fCKIN (...but CNVCNT=0) Continuous mode is not applicable to injected conversions. Injected conversions can be initiated by a timer trigger to use precise timing simulation of continuous mode. If a regular continuous conversion is in progress (RCONT=1), and if a write access is made to the R32_DFSDM_FLTxCR1 register requesting a regular continuous conversion (RCONT=1), the regular continuous conversion will be restarted from the next conversion cycle (similar to a new regular continuous conversion applied to a new channel selection, even if there are no changes in the R32_DFSDM_FLTxCR1 register).

42.3.15 Request priority injection

transformations have higher priority than regular transformations. An ongoing regular transformation will be immediately interrupted by the injected transformation request; during an injected transformation This regular transformation will restart upon completion. If an injection transformation is pending or in progress... , Then other injection conversions cannot be initiated: as long as If the JCIP bit is set to 1 in the R32_DFSDM_FLTxISR register, requests to initiate an injection conversion (via JSWSTART or trigger) will be ignored. Similarly, if a regular conversion is pending or in progress, other regular conversions cannot be initiated: as long as...

If the RCIP bit is set to 1 in the R32_DFSDM_FLTxISR register, the request to start a regular conversion (using RSWSTART) will be ignored.

However, if an injection transformation is requested during a regular transformation, the regular transformation will stop immediately and the injection transformation will be started. A restart will occur after a normal conversion, and this delayed restart will be indicated by the RPEND bit.

Injection transformations have higher priority than regular transformations because they can temporarily interrupt a continuous sequence of regular transformations. Complete the injection transformation. During the sequence, if RCONT remains set to 1, continuous normal conversions will restart (the RPEND bit will issue a delay based on the result of the first normal conversion). (Late start signal).

When initiating related operations by performing the same write operation on DFSDM, or if multiple operations are in progress after other operations have finished. In a pending state, priority is also crucial. For example, assuming injection conversion is in progress (JCIP=1), then for R32_DFSDM_FLTxCR1... A single write operation will write 1 to RSWSTART to request a regular conversion. Upon completion of the injection sequence, the priority indicates which regular conversion will proceed next. The rule is converted, and the corresponding delayed start is indicated by the RPEND bit.

42.3.16 Power optimization in operating mode : To reduce

power consumption, the DFSDM filter and integrator will automatically enter an idle state (RCIP=0, JCIP=0) when not used for conversion.

42.3.17 Trigger Signals and Trigger Sources

Table 42-3 Connection of Trigger Signal and Trigger Source

Trigger Name	Trigger source trigger name		Trigger source
DFSDM_JTRG0	TIM1_TRGO	DFSDM_JTRG8	TIM9_TRGO
DFSDM_JTRG1	TIM2_TRGO	DFSDM_JTRG9	TIM10_TRGO
DFSDM_JTRG2	TIM3_TRGO	DFSDM_JTRG10	TIM11_TRGO
DFSDM_JTRG3	TIM4_TRGO	DFSDM_JTRG11	TIM12_TRGO
DFSDM_JTRG4	TIM5_TRGO	DFSDM_JTRG12	EXTI11
DFSDM_JTRG5	TIM6_TRGO	DFSDM_JTRG13	EXTI15
DFSDM_JTRG6	TIM7_TRGO	DFSDM_JTRG14	LPTIM1
DFSDM_JTRG7	TIM8_TRGO	DFSDM_JTRG15	LPTIM2

42.3.18 DFSDM Disconnection

Table 42-4 DFSDM Disconnection Connection

circuit breaker name	Target of circuit breaker
DFSDM_BK[0]	TIM1 Disconnection 1
DFSDM_BK[1]	TIM1 Disconnection 2

42.4 Interruption

To enhance the performance of the central processing unit (CPU), a set of interrupts related to CPU events has been implemented:

ÿInjection conversion completed and interrupted:

- Enable via the JEOCIE bit in register R32_DFSDM_FLTxCR2
- Indicated via the JEOCF bit of the R32_DFSDM_FLTxISR register
- Clear by reading the R32_DFSDM_FLTxJDATAR register (injected data)
- Indicates which channel the conversion ended, and reports it in the JDATACH bit of the R32_DFSDM_FLTxJDATAR register.

tell

ÿ Normal conversion completed and interrupted:

- Enable via the REOCIE bit in register R32_DFSDM_FLTxCR2
- Indicated via the REOCF bit of the R32_DFSDM_FLTxISR register

- Clear by reading the R32_DFSDM_FLTxRDATAR register (regular data)

- Indicates which channel the conversion ended, and reports it in the RDATACH bit of the R32_DFSDM_FLTxRDATAR register.

tell

ÿInjection conversion data overflow interruption:

- When the injected conversion data is not read from the R32_DFSDM_FLTxJDATAR register (via CPU or DMA) and is newly injected

This interrupt occurs during an inbound conversion overwrite.

- Enable via the JOVRIE bit in the R32_DFSDM_FLTxCR2 register

- Indicated via the JOVRF bit of the R32_DFSDM_FLTxISR register

- Clear by writing 1 to the CLRJOVRF bit of the R32_DFSDM_FLTxICR register.

ÿ Normal conversion data overflow interruption:

- When regular conversion data is not read from the R32_DFSDM_FLTxRDATAR register (via CPU or DMA) and is replaced by a new regular conversion...

This interruption occurs during the conversion overwrite.

- Enable via the ROVRIE bit in register R32_DFSDM_FLTxCR2

- Indicated via the ROVRF bit of the R32_DFSDM_FLTxISR register.

- Clear by writing 1 to the CLRROVRF bit of the R32_DFSDM_FLTxICR register.

Simulated watchdog interrupt:

- When converting data (output data or data in the analog watchdog filter—depending specifically on R32_DFSDM_FLTxCR1)

The AWFSEL bit setting in the register exceeds the threshold in the R32_DFSDM_FLTxAWHTR/R32_DFSDM_FLTxAWLTR register.

This interruption occurs when the value reaches the upper limit or falls below the corresponding lower threshold.

- Enable via the AWDIE bit of the R32_DFSDM_FLTxCR2 register (of the selected channel AWDCH[1:0]).

- Indicated by the AWDF bit of the R32_DFSDM_FLTxISR register

- The simulated watchdog threshold is indicated separately via the AWHTF[1:0] and AULTF[1:0] fields of the R32_DFSDM_FLTxAWSR register.

Limit or lower limit error

- Clear the register by writing 1 to the corresponding CLRAWHTF[1:0] or CLRAWLTF[1:0] bits in the R32_DFSDM_FLTxAWCFR register.

remove

Short -circuit detector interruption:

This interrupt occurs when the number of stable data points exceeds the threshold in the R32_DFSDM_CHyAWSCDR register.

- The R32_DFSDM_FLTxCR2 register (selected via the SCDEN bit of the R32_DFSDM_CHyCFGR1 register)

SCDIE bit enable

- The SCDF[1:0] bits of the R32_DFSDM_FLTxISR register indicate (and also report channels where short-circuit detector events have occurred)

- Clear by writing 1 to the corresponding CLRSCDF[1:0] bits in the R32_DFSDM_FLTxICR register.

Channel clock missing interrupt:

This interrupt occurs when a clock is missing on the CKINy pin (see Section 42.3.2: Clock Missing Serial Channel Transceiver).

(Detection)

- via the R32_DFSDM_FLTxCR2 register (selected via the CKABEN bit of the R32_DFSDM_CHyCFGR1 register).

CKABIE bit enable of the device

- Indicated via the CKABF[y] bit of the R32_DFSDM_FLTxISR register.

- Clear by writing 1 to the CLRCKABF[y] bit of the R32_DFSDM_FLTxICR register.

Table 42-5 DFSDM Interruption Requests

Interrupt event	Event marker	Event/Interrupt Clearing Method	Interrupt Enable Control Bit
injection, conversion	JEOCF	Read R32_DFSDM_FLTxJDATAR Read	JEOCIE
end, regular conversion	REOCF	R32_DFSDM_FLTxRDATAR Write	REOCIE
end, injection data	JOVRF	CLRJOVRF=1 Write	JOVRIE
overflow, regular data	ROVRF	CLRROVRF=1 Write	ROVRIE
overflow, simulated watchdog.	AWDF,	CLRAWHTF[1:0]=1	AWDIE,

	AWHTF[1:0]ÿ AWLTF[1:0]	Write CLRAWLTF[1:0]=1	(AWDCH[1:0])
Short circuit detector	SCDF[1:0]	Write CLRSCDF[1:0]=1	SCDIE, (SCDEN)
Channel clock missing	CKABF[1:0]	Write CLRCKABF[1:0]=1	CKABIE (CKABEN)

42.5 DFSDM DMA Transfer

To reduce CPU intervention, DMA transfers can be used to move the conversion to memory. DMA transfers injecting conversions are achieved by...

Enable DMA transfer by setting the JDMAEN bit to 1 in the R32_DFSDM_FLTxCR1 register. Normal DMA transfers are performed by using the R32_DFSDM_FLTxCR1 register.

Enable the RDMAEN bit of the register to 1.

Note: via DMA The transfer interrupt flag is automatically cleared (in the R32_DFSDM_FLTxISR register) at the end of the injection or normal conversion.

JEOCF DMA R32_DFSDM_FLTxRDATAR (or REOCF bit), because it is reading the R32_DFSDM_FLTxJDATAR or register.

42.6 Register Description

Table 42-6 List of DFSDM Channel-Related Registers

name	Access address	describe	Reset value
R32_DFSDM_CH0CFGR1	0x40017000	DFSDM Channel 0 Configuration Register 1	0x00000000
R32_DFSDM_CH0CFGR2	0x40017008	DFSDM Channel 0 Configuration Register 2	0x00000000
R32_DFSDM_CH0AWSCDR	0x40017010	DFSDM Channel 0 Simulates Watchdog and Short Circuit Detector register	0x00000000
R32_DFSDM_CH0WDATR	0x40017018	DFSDM Channel 0 Watchdog Filter Data Register	0x00000000
R32_DFSDM_CH0DATINR	0x40017020: DFSDM Channel 0 Data Input Register 0x00000000 0x40017004: DFSDM		
R32_DFSDM_CH1CFGR1	Channel 1 Configuration Register 1 0x4001700C: DFSDM		0x00000000
R32_DFSDM_CH1CFGR2	Channel 1 Configuration Register 2		0x00000000
R32_DFSDM_CH1WSCDR	0x4001714	DFSDM Channel 1 Simulates Watchdog and Short Circuit Detector register	0x00000000
R32_DFSDM_CH1WDATR	0x400171C	DFSDM Channel 1 Watchdog Filter Data Register	0x00000000
R32_DFSDM_CH1DATINR	0x40017024 DFSDM Channel 1 Data Input Register 0x00000000		

Table 42-7 List of DFSDM Filter-Related Registers

name	Access Address Description: 0x40017028		Reset value
R32_DFSDM_FLT0CR1	DFSDM Filter 0 Control Register 1; 0x40017030 DFSDM Filter		0x00000000
R32_DFSDM_FLT0CR2	0 Control Register 2		0x00000000
R32_DFSDM_FLT0ISR	0x40017038	DFSDM Filter 0 Interrupts and Status Registers instrument	0x00000000
R32_DFSDM_FLT0ICR	0x40017040	DFSDM filter 0 interrupt flag cleared Register	0x00000000
R32_DFSDM_FLT0JCHGR	0x40017048	DFSDM Filter 0 Injection Channel Group Selection register	0x00000001

R32_DFSDM_FLT0FCR3	0x40017050	DFSDM Filter 0 Control Register 3	0x00000000
R32_DFSDM_FLT0JDATAR	0x40017058	DFSDM filter 0 injection group data register	0x00000000
R32_DFSDM_FLT0RDATAR	0x40017060	DFSDM filter 0 Normal channel data transmission Register	0x00000000
R32_DFSDM_FLT0AWHTR	0x40017068	DFSDM filter 0 Simulated watchdog threshold Upper limit register	0x00000000
R32_DFSDM_FLT0AWLTR	0x40017070	DFSDM filter 0 Simulated watchdog threshold Lower limit register	0x00000000
R32_DFSDM_FLT0AWSR	0x40017078	DFSDM filter 0 simulates watchdog state register	0x00000000
R32_DFSDM_FLT0AWCFR	0x40017080	DFSDM filter 0 analog watchdog reset Flag register	0x00000000
R32_DFSDM_FLT0EXMAX	0x40017088	DFSDM filter 0 extreme value detector maximum Value Register	0x80000000
R32_DFSDM_FLT0EXMIN	0x40017090	DFSDM filter 0 extreme value detector minimum Value Register	0x00000000
R32_DFSDM_FLT0CNVTIMR	0x40017098	DFSDM Filter 0 Conversion Timer Register	0x00000000
R32_DFSDM_FLT1CR1	0x4001702C	DFSDM Filter 1 Control Register 1 0x40017034	0x00000000
R32_DFSDM_FLT1CR2	DFSDM Filter 1 Control Register 2		0x00000000
R32_DFSDM_FLT1ISR	0x4001703C	DFSDM Filter 1 Interrupts and Status Registers	0x00000000
R32_DFSDM_FLT1ICR	0x40017044	DFSDM Filter 1 Interrupt Flag Cleared Register	0x00000000
R32_DFSDM_FLT1JCHGR	0x4001704C	DFSDM Filter 1 Injection Channel Group Selection Register	0x00000001
R32_DFSDM_FLT1FCR3	0x40017054	DFSDM Filter 1 Control Register 3	0x00000000
R32_DFSDM_FLT1JDATAR	0x4001705C	DFSDM Filter 1 Injection Group Data Register	0x00000000
R32_DFSDM_FLT1RDATAR	0x40017064	DFSDM Filter 1 Normal Channel Data Transfer Register	0x00000000
R32_DFSDM_FLT1AWHTR	0x4001706C	DFSDM Filter 1 Simulated Watchdog Threshold Upper limit register	0x00000000
R32_DFSDM_FLT1AWLTR	0x40017074	DFSDM Filter 1 Simulated Watchdog Threshold Lower limit register	0x00000000
R32_DFSDM_FLT1AWSR	0x4001707C	DFSDM filter 1 simulates watchdog state register	0x00000000
R32_DFSDM_FLT1AWCFR	0x40017084	DFSDM filter 1 analog watchdog reset Flag register	0x00000000
R32_DFSDM_FLT1EXMAX	0x4001708C	DFSDM Filter 1 Extreme Value Detector Maximum Value Register	0x80000000
R32_DFSDM_FLT1EXMIN	0x40017094	DFSDM filter 1 extreme value detector minimum Value Register	0x00000000

R32_DFSDM_FLT1CNVTIM R	0x4001709C	DFSDM Filter 1 Conversion Timer Register	0x00000000

42.6.1 DFSDM Channel y Configuration Register 1 (R32_DFSDM_CHyCFGR1) (y=0/1)

Offset address: 0x00 + 0x04 * y

31	30	29 28 27 26 25 24	23 22 21	2019	18	17 16
DFSDMEN	CKOUTSRC	Reserved	CKOUTDIV[7:0]			

15	14	13 12	11 10	9	8	7	6	5	4	3 2	1	0
DATPACK [1:0]		DATMPX [1:0]	Reserved		CHINSEL	CHEN	CKABEN	SCDEN	Reserved	SPICKSEL [1:0]	Reserved	SITP

Bit	name	access	describe	Reset value
31	DFSDMEN	RW	Global enable of DFSDM interface: 1: Enable the DFSDM interface; 0: Disable DFSDM interface. If the DFSDM interface is enabled, the interface will be configured according to the enabled settings. Y-channel and enabled X-filter settings (The CHEN bit of R32_DFSDM_CHyCFGR1 and) The DFEN bit of R32_DFSDM_FLTxCR1 is used for operation. Set DFSDMEN=0 to clear data: – All registers R32_DFSDM_FLTxISR are set to reset. State (x=0, 1); – All registers R32_DFSDM_FLTxAWSR are set to reset. State (x=0, 1). Note: DFSDMEN exists only in register R32_DFSDM_CH0CFGR1 In the device (channel y=0).	0
30	CKOUTSRC	RW	Output serial clock source selection: 1: The output clock source is the audio clock; 0: The output clock source is from the peripheral clock. Only in (R32_DFSDM_CH0CFGR1 register) This value can only be modified when DFSDMEN=0. Note: CKOUTSRC exists only in register R32_DFSDM_CH0CFGR1. In the device (channel y=0).	0
[29:24]	Reserved	RO is		0
[23:16]	CKOUTDIV[7:0]	RW	reserved. Output serial clock divider: 0: Disable output clock generation (CKOUT signal is set low) state); 1ÿ255: Systems that define the serial clock output for the CKOUT signal Clock frequency division, ranging from 2 to 256 (divider value = CKOUTDIV + 1). CKOUTDIV also defines the threshold for clock missing detection. Only in (R32_DFSDM_CH0CFGR1 register) This value can only be modified when DFSDMEN=0. If DFSDMEN=0 (in the R32_DFSDM_CH0CFGR1 register),	0

			The CKOUT signal is then set to a low level (the setting is performed within one DFSDM clock cycle after DFSDMEN=0). Note: CKOUTDIV exists only in the R32_DFSDM_CH0CFGR1 register (channel y=0).	
[15:14] DATPACK[1:0]		RW	R32_DFSDM_CHyDATINR register data encapsulation mode: 00: Standard: Input data in the R32_DFSDM_CHyDATINR register is stored only in INDAT0[15:0]. To clear... The R32_DFSDM_CHyDATINR register must be used by the DFSDM filter to read one sample from channel y. 01: Interleaved: The input data in the R32_DFSDM_CHyDATINR register is stored as two samples: – The first sample is located in INDAT0[15:0] (allocated to channel y); – The second sample is located in INDAT1[15:0] (allocated to channel y). To clear the R32_DFSDM_CHyDATINR register, two samples must be read from channel y by the digital filter (INDAT0[15:0] is read as the first sample, and INDAT1[15:0] is read as the next sample). 10: Dual-channel: The input data in the R32_DFSDM_CHyDATINR register is stored as two samples: – The first sample is located in INDAT0[15:0] (allocated to channel y); – The second sample is located in INDAT1[15:0] (allocated to channel y+1). To clear the R32_DFSDM_CHyDATINR register, the first sample must be read from channel y by a digital filter, and the second sample must be read from channel y+1 by another digital filter. Dual-channel mode is only supported when , the number of channels is even (y=0). For odd-numbered channels (y=1), R32_DFSDM_CHyDATINR is write-protected. If an even-numbered channel is set to dual-channel mode, the next odd-numbered channel must enter standard mode (DATPACK[1:0]=0) to properly match	0
[13:12] DATMPX[1:0]		RW	the even-numbered channel. 11: Reserved. This value can only be modified when CHEN=0 (in the R32_DFSDM_CHyCFGR1 register). Channel y input data multiplexer: 00: Channel y input data is taken from an external serial input and is a 1-bit value. The R32_DFSDM_CHyDATINR register is write-pro The ADC output data is written to the INDAT0[15:0] section of the R32_DFSDM_CHyDATINR register; 10: Channel y input data is taken from the internal R32_DFSDM_CHyDATINR register (directly via CPU/DMA).	

			Write operation implementation). One or two 16-bit data samples can be written. Specifically, it depends on the DATPACK[1:0] bit field settings; 11: Retained. Only when CHEN=0 in the R32_DFSDM_CHyCFGR1 register This value can only be modified at this time. Note: DATMPX[1:0]=1 is only applicable to ADC2.	
[11:9] Reserved		RO reserved.		0
8	CHINSEL	RW	Channel input selection: 1: Channel input is taken from the next channel (channel (y+1) takes 2). Pins of the module; 0: Channel input is taken from the pin of the same channel y. Only when CHEN=0 in the R32_DFSDM_CHyCFGR1 register This value can only be modified at this time.	0
7	CHEN	RW	Channel y enabled: 1: Enable channel y; 0: Channel y is prohibited. If channel y is enabled, the connection will begin according to the given channel settings. Receive serial data.	0
6	CKABEN	RW	Channel y clock missing detector enabled: 1: Enable channel y clock missing detector; 0: Disable channel y clock missing detector.	0
5	SCDEN	RW	Channel y short-circuit detector enabled: 1: Input channel y will be continuously protected by the short-circuit detector; 0: Input channel y will not be protected by the short-circuit detector.	0
4	Reserved	RO reserved.		0
[3:2] SPICKSEL[1:0]		RW	Channel y SPI clock selection: 00: Clock comes from external CKINy input – SITP-based sampling Sample points; 01: Clock sourced from internal CKOUT output – based on SITP sampling Sample points; 10: The clock comes from the internal CKOUT—in CKOUT every second Sampling point at the falling edge. Used to connect an external ÿÿ modulator, which receives the clock input (from...). Divide CKOUT by 2 to generate the output serial communication clock (this... The output clock change is valid on the rising edge of each clock input. 11: Clock comes from internal CKOUT output—in each CKOUT The sampling point at the second rising edge. Used to connect an external ÿÿ modulator, which receives the clock input (from...). Divide CKOUT by 2 to generate the output serial communication clock (this... The output clock change is valid on the falling edge of each clock input. Only when CHEN=0 in the R32_DFSDM_CHyCFGR1 register This value can only be modified at this time.	0
1	Reserved	RO reserved.		0
0	SITP	RW	Channel y serial interface type: 1: Falling edge strobe data SPI;	0

			0: Rising edge strobe data SPI. Only when CHEN=0 in the R32_DFSDM_CHyCFGR1 register This value can only be modified at this time.	

42.6.2 DFSDM Channel y Configuration Register 2 (R32_DFSDM_CHyCFGR2) (y=0/1)

Offset address: 0x08 + 0x04 * y

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
OFFSET[23:8]														

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

Bit	name	access		Reset value
[31:8]	OFFSET[23:0]	RW	Description of channel y: 24-bit calibration offset: For each transformation result of channel y, OFFSET is executed. This value is set by the software.	0
[7:3]	DTRBS[4:0]	RW	Right shift of channel y data: 00000-01000 is valid, defining the shift of the integrator output data. Bit – How many bits to shift to the right to obtain the final result. In the process of... Shifting is performed before offset correction. Data shifting rounds the result. Input is the nearest integer value. The sign of the shifted result is preserved (so that...). Obtain valid 24-bit signed format result data. Only when CHEN=0 in the R32_DFSDM_CHyCFGR1 register This value can only be modified at this time. Other: Invalid.	0
[2:0]	Reserved	RO	is reserved.	0

42.6.3 DFSDM Channel y Analog Watchdog and Short-Circuit Detector Register (R32_DFSDM_CHyAWSCDR)

(y=0/1)

Offset address: 0x10 + 0x04 * x

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved											AWFORD[1:0]	Reserved	AWFOSR[4:0]	

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

Bit	name	access	describe	Reset value
[31:24]	Reserved	RO	reserved.	0
[23:22]	AWFORD[1:0]	RW	Channel y analog watchdog Sinc filter order: 00: FastSinc filter type; 01: Sinc1 filter type; 10: Sinc2 filter type;	0

			11: Sinc3 filter type. Only when CHEN=0 in the R32_DFSDM_CHyCFGR1 register Only then can that bit be modified.	
twenty one	Reserved	RO reserved.		0
[20:16] AWFOSR[4:0]		RW	Channel y analog watchdog filter oversampling rate: 0ÿ31: Defines the length of the Sinc type filter, the length range The range is 1 to 32 (AWFOSR+1). This number also represents the analog data rate. The extraction rate. Only when CHEN=0 in the R32_DFSDM_CHyCFGR1 register Only then can that bit be modified. Note: If AWFOSR=0, the filter has no effect (beside the filter). road).	0
[15:14] Reserved		RO reserved.		0
[13:12] BKSCD[1:0]		RW	Channel y short-circuit detector open-circuit signal assignment: 00: The open circuit signal was not assigned to the channel short circuit detector; 01: The open circuit 0 signal is assigned to the short circuit detector in channel 0; 10: The open circuit 1 signal is distributed to the short circuit detector of channel 1; 11: The open circuit signal is distributed to the corresponding channel short circuit detector.	0
[11:8] Reserved		RO is reserved.		0
[7:0] SCDT[7:0]		RW	Short-circuit detector threshold for channel y: These bits can be written by software to define the short-circuit detector's calculation. Counter threshold. If this value is reached, a short circuit occurs on a given channel. Road detector event.	0

42.6.4 DFSDM Channel y Watchdog Filter Data Register (R32_DFSDM_CHyWDATR) (y=0/1) Offset Address: 0x18 + 0x04 * y

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

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

Bit	name	access	describe	Reset value
[31:16] Reserved		RO reserved.		0
[15:0] WDATA[15:0]		RO	Input channel y watchdog data: The data is converted by the analog watchdog filter from the input channel y. The channel performs continuous data conversion (without triggering), and the conversion resolution is adjusted accordingly. It is subject to certain restrictions (OSR = 1...32/sinc, order = 1...3).	0

42.6.5 DFSDM Channel y Data Input Register (R32_DFSDM_CHyDATINR) (y=0/1)

Offset address: 0x20 + 0x04 * y

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
INDAT1[15:0]														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INDAT0[15:0]															

Bit	name	access		Reset value
[31:16]	INDAT1[15:0]	RW	Describe the input data for channel y or channel y+1: When DATMPX[1:0]=1 or DATMPX[1:0]=2, input and The row channel data will be processed through a digital filter. The data can... Writes can also be made via CPU/DMA (when DATMPX[1:0]=2). It can be written directly by the internal ADC (when DATMPX[1:0]=1). When DATPACK[1:0]=0 (standard mode), INDAT0[15:0] is write- protected (not used for input sampling). When DATPACK[1:0]=1 (interleaved mode), The second channel y data sample is stored in INDAT1[15:0]. The first channel y data sample is stored in INDAT0[15:0]. The DFSDM_FLTx filter reads these two samples sequentially as two... Data sampling for each channel y. When DATPACK[1:0]=2 (dual mode), For even-numbered y channels: the sampling in INDAT1[15:0] is automatically repeated. It is controlled into INDAT0[15:0] of channel (y+1). For odd-numbered y channels: INDAT1[15:0] is write-protected. INDAT0[15:1] uses a 16-bit signed format.	0
[15:0]	INDAT0[15:0]	RW	Channel y input data: When DATMPX[1:0]=1 or DATMPX[1:0]=2, input and The row channel data will be processed through a digital filter. Data can be written by CPU/DMA (when DATMPX[1:0]=2). It can also be written directly by the internal ADC (when DATMPX[1:0]=1). When DATPACK[1:0]=0 (standard mode), channel y data The samples are stored in INDAT0[15:0]. When DATPACK[1:0]=1 (interleaved mode), The first channel y data sample is stored in INDAT0[15:0]; The second channel y data sample is stored in INDAT1[15:0]. The DFSDM_FLTx filter reads these two samples sequentially as two... Data sampling for each channel y. When DATPACK[1:0]=2 (dual mode), For even-numbered y channels: channel y data samples are stored to... In INDAT0[15:0]. For odd-numbered y channels: INDAT0[15:0] is write-protected. INDAT0[15:0] uses a 16-bit signed format.	0

42.6.6 DFSDM Filter x Control Register 1 (R32_DFSDM_FLTxCR1) (x=0/1) Offset Address: 0x28 + 0x04 * x

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16

Reserved	AWFSEL	FAST	Reserved	RCH	Reserved	RDMAEN	Reserved	RSYNC	RCONT	RSWSTART	Reserved
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Reserved	JEXTEN[1:0]	Reserved	JEXTSEL[3:0]	Reserved	JDMAEN	JSCAN	JSYNC	Reserved	JSWSTART	DFEN

Bit	name	access	describe	Reset value
31	Reserved	RO is reserved.		0
30	AWFSEL	RW	Watchdog simulation quick mode selection: 1: Analog watchdog timer value for channel transceiver (in watchdog filter) after); 0: Analog watchdog timer for data output values (in the digital filter) (After). Comparison is performed after offset correction and shifting.	0
29	FAST	RW	Selecting the fast conversion mode for regular conversion: 1: Enables quick mode switching; 0: Disable fast mode switching. When performing a regular conversion in continuous mode, if fast conversion is enabled... The pattern, then each conversion (except the first conversion) is better than the standard. Transitions in this mode are executed quickly. This bit is useful for discontinuous transitions. No impact. Modification is only possible when DFEN=0 (R32_DFSDM_FLTxCR1). That position.	0
[28:25]	Reserved	RO reserved.		0
24	RCH	RW	Standard channel selection: 1: Select channel 1 as the normal channel; 0: Select channel 0 as the normal channel. Write these bits when RCIP=1, and then in the next regular conversion... Effective from start to finish. This is especially true in continuous mode (when RCONT=1). It's useful. It will also affect the regular conversions that are pending (because they are in progress). (Injection conversion paused).	0
[23:22]	Reserved	RO reserved.		0
twenty one	RDMAEN	RW	Enable DMA channel to read regular conversion data: 1: DMA channels are enabled for reading regular data; 0: DMA channels are not enabled for reading regular data. Modification is only possible when DFEN=0 (R32_DFSDM_FLTxCR1). That position.	0
20	Reserved	RO reserved.		0
19	RSYNC	RW	Start regular conversion synchronously using DFSDM_FLT0: 1: While initiating regular conversion in DFSDM_FLT0, at the same time... Initiate a regular conversion in DFSDM_FLTx; 0: Do not use DFSDM_FLT0 to synchronously start regular conversions. Modification is only possible when DFEN=0 (R32_DFSDM_FLTxCR1). That position.	0

18 RCONT	RW	Continuous mode selection for regular conversion: 1: Repeat the conversion of the regular channel after each conversion request is issued; 0: For each conversion request, the regular channel is converted only once. During continuous routine conversions, writing 0 to this bit will immediately... This means stopping continuous mode.	0
17 RSWSTART	RW1T	Software startup standard channel conversion: 1: Writing 1 will send a request to initiate a regular channel conversion, and will... Set RCIP to 1. If RCIP is already set to 1, then... Write operations initiated by RSWSTART are invalid. If RSYNC=1, then... Writing 1 is invalid; 0: Writing 0 has no effect. This bit is always read as 0.	0
[16:15] Reserved	RO reserved.		0
[14:13] JEXTEN[1:0]	RW	Injection conversion trigger enable and trigger edge selection: 00: Prohibit detection; 01: Each rising edge of the selected trigger will emit a startup injection signal. A request for a replacement; 10: Each falling edge of the selected trigger will emit a startup injection signal. A request for a replacement; 11: Both the rising and falling edges of the selected trigger will emit a start signal. The request to convert the input. Modification is only possible when DFEN=0 (R32_DFSDM_FLTxCR1). That position.	0
12 Reserved	RO is reserved.		0

<table>
<tr><td>[11:8] JEXTSEL[3:0]</td><td>RW</td><td>Trigger signal selection for initiating injection conversion:
0x0ÿ0xF: Trigger input (internal or external) selected according to the table below.
trigger).
Modification is only possible when DFEN=0 (R32_DFSDM_FLTxCR1).
That position.
Note: The maximum delay for synchronous triggering is one fDFSDMCLK.
The clock cycle (with definite jitter) has an asynchronous trigger latency of [time period].
fDFSDMCLK clock cycle (jitter lasts up to one cycle).
<table><tr><th></th><th>DFSDM_FLT0 DFSDM_FLT1</th><th></th></tr><tr><td>0x00</td><td>dfsdm_jtrg0</td><td>dfsdm_jtrg0</td></tr><tr><td>0x01</td><td>dfsdm_jtrg1</td><td>dfsdm_jtrg1</td></tr><tr><td>…</td><td></td><td></td></tr><tr><td>0xE</td><td>dfsdm_
jtrg14</td><td>dfsdm_
jtrg14</td></tr><tr><td>0xF</td><td>dfsdm_
jtrg15</td><td>dfsdm_
jtrg15</td></tr></table></td><td>0</td></tr>
<tr><td>[7:6] Reserved</td><td>RO reserved.</td><td></td><td>0</td></tr>
<tr><td>5</td><td>JDMAEN</td><td>RW</td><td>Enable DMA channel to read data injected into the channel group:
1: DMA channel is enabled to read injected data;
0: DMA channel is not enabled to read injected data.
Modification is only possible when DFEN=0 (R32_DFSDM_FLTxCR1).
That position.</td><td>0</td></tr>
<tr><td>4</td><td>JSCAN</td><td>RW</td><td>Injection conversion scan conversion mode:
1: Starting with the selected lowest channel, perform continuous injection on the channel group.
Continued conversion;
0: Perform a channel conversion on the injection channel group, then select the...
The next channel in the channel group.
Modification is only possible when DFEN=0 (R32_DFSDM_FLTxCR1).
That position.
If JSCAN=0, then writing to JCHG will open the channel.
Select reset as the minimum selected channel.</td><td>0</td></tr>
<tr><td>3</td><td>JSYNC</td><td>RW</td><td>Use DFSDM_FLT0 JSWSTART to trigger synchronous startup injection.
Change:
1: In DFSDM_FLT0, trigger startup injection via JSWSTART.
Simultaneously with the conversion, the injection conversion is initiated in the DFSDM_FLTx;
0: Do not use DFSDM_FLT0 synchronous startup injection conversion.
Modification is only possible when DFEN=0 (R32_DFSDM_FLTxCR1).
That position.</td><td>0</td></tr>
<tr><td>2</td><td>Reserved</td><td>RO is reserved.</td><td></td><td>0</td></tr>
</table>

1	JSWSTART	RW1T	Initiate injection channel group conversion: 1: Writing 1 will issue a conversion injection channel group request, and at the same time... JCIP becomes 1. If JCIP is already set to 1, then JSWSTART will be executed. The write operation performed is invalid. If JSYNC=1, then 1 is written. effect; 0: Writing 0 has no effect. This bit is always read	0
0	DFEN	RW	as 0. DFSDM_FLTx Enable: 1: Enable DFSDM_FLTx. If DFSDM_FLTx is enabled, then DFSDM_FLTx will start working according to its settings; 0: Disable DFSDM_FLTx. All transformations given DFSDM_FLTx are disabled. All changes will stop immediately, and all functions of DFSDM_FLTx will cease. Data cleared by setting DFEN=0: – Register R32_DFSDM_FLTxISR is set to reset state; – Register R32_DFSDM_FLTxAWSR is set to reset state.	0

42.6.7 DFSDM Filter x Control Register 2 (R32_DFSDM_FLTxCR2) (x=0/1) Offset Address: 0x30 + 0x04 * x

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
Reserved					AWDCH[1:0]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						EXCH[1:0]		Reserved	CKABIE	SCDIE	AWDIE	ROVRIE	JOVRIE	REOCIE	JEOCIE

Bit	name	access	describe	Reset value
[31:18] Reserved		RO reserved.		0
[17:16] AWDCH[1:0]		RW	Simulated watchdog channel selection: These bits are used to select the input path continuously monitored by the analog watchdog. road: 00: Disable watchdog simulation on channels 0 and 1; 01: Enable the simulated watchdog timer on channel 0; 10: Enable the simulated watchdog on channel 1; 11: Enable simulated watchdog on Channel 0 and Channel 1.	0
[15:10] Reserved		RO reserved.		0
[9:8] EXCH[1:0]		RW	Extreme value detector channel selection: These are the input channels that the bit selection extremum detectors will use. 00: The extreme value detector does not accept data from channel 0 and channel 1; 01: The extreme value detector receives data from channel 0; 10: The extreme value detector receives data from channel 1; 11: The extreme value detector accepts data from channel 0 and channel 1.	0
7	Reserved	RO reserved.		0
6	CKABIE	RW	Clock miss interrupt enabled: 1: Enable detection of channel input clock missing interrupt;	0

			0: Disables interrupt for missing input clock in the detection channel. Please refer to CKABF[1:0] in R32_DFSDM_FLTxISR. bright. Note: CKABIE exists only in the R32_DFSDM_FLT0CR2 register. In the middle (filter x=0).	
5	SCDIE	RW	Short circuit detector interrupt enable: 1: Enable short-circuit detector interrupt; 0: Disable short-circuit detector interruption. Please refer to the SCDF[1:0] description in R32_DFSDM_FLTxISR. Note: SCDIE exists only in the R32_DFSDM_FLT0CR2 register. (Filter x=0).	0
4	AWDIE	RW	Simulated watchdog interrupt enable: 1: Enable simulated watchdog interrupt; 0: Disable simulated watchdog interruption. Please refer to the AWDF description in R32_DFSDM_FLTxISR.	0
3	ROVRIE	RW	Enable regular data overflow interrupt: 1: Enable regular data overflow interrupt; 0: Disables regular data overflow interruptions. Please refer to the ROVRF description in R32_DFSDM_FLTxISR.	0
2	JOVRIE	RW	Injection data overflow interrupt enable: 1: Enable the data overflow interrupt; 0: Disallows data injection overflow interruption. Please refer to the JOVRF description in R32_DFSDM_FLTxISR.	0
1	REOCIE	RW	Normal conversion completion interrupt enable: 1: Enable interrupt for normal transition completion; 0: Disallow interruption at the end of normal conversion. Please refer to the REOCF description in R32_DFSDM_FLTxISR.	0
0	JEOCIE	RW	Injection conversion completion interrupt enable: 1: Enable the injection conversion completion interrupt; 0: Disallow injection conversion from interrupting. Please refer to the JEOCF description in R32_DFSDM_FLTxISR.	0

42.6.8 DFSDM Filter x Interrupt and Status Register (R32_DFSDM_FLTxISR) (x=0/1)

Offset address: 0x38 + 0x04 * x

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
Reserved			SCDF[1:0]	Reserved	CKABF[1:0]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	RCIP	JCIP	Reserved								AWDF	ROVRF	JOVRF	REOCF	JEOCF

Bit	name	access	describe	Reset value
[31:26] Reserved		RO is reserved.		0
[25:24] SCDF[1:0]		RO short circuit detector marking:		0

			00: No short-circuit detectors were detected on either channel 0 or channel 1. event; 01: The short-circuit detector counter on channel 0 has reached the limit. The value programmed into the R32_DFSDM_CHyAWSCDR register; 10: The short-circuit detector counter on channel 1 has reached the limit. The value programmed into the R32_DFSDM_CHyAWSCDR register; 11: The short-circuit detector counters on both Channel 0 and Channel 1 have reached [a certain value]. The value programmed into the R32_DFSDM_CHyAWSCDR register; This bit will be set to 1 by hardware, This bit can be used by software. Clear the CLRSCDF bit in the R32_DFSDM_FLTxICR register. SCDF can still function when CHEN[y]=0 (the given channel is disabled). Reset by hardware. Note: SCDF[1:0] exists only in the R32_DFSDM_FLT0ISR register. In the device (filter x=0).	
[23:18] Reserved		RO	reserved.	0
[17:16] CKABF[1:0]		RW	Clock missing flag: 00: Clock signals exist on both channel 0 and channel 1; 01: There is no clock signal on channel 0; 10: There is no clock signal on channel 1; 11: There is no clock signal on either channel 0 or channel 1. When the absence of a clock signal on the channel is detected, this bit is determined by hardware. Set to 1. When CHEN=0 (see R32_DFSDM_CHyCFGR1) (Register), which is maintained by hardware in a clock-miss state. During transmission and reception... When the device has not been synchronized, it is kept in a clock-missing state by the hardware. It can be accessed by software from the R32_DFSDM_FLTxICR register. The corresponding CLRCKABF bit is cleared to zero. Note: CKABF[1:0] only exists in R32_DFSDM_FLT0ISR. RO is reserved in the register .	0
15 Reserved		(filter	x=0).	0
14 RCIP		RW	Normal conversion is in progress: 1: Regular channel conversion is in progress, or a regular conversion request is being made. Awaiting processing; 0: No request was issued to switch to the normal channel. When RCIP=1, requests to initiate a regular conversion are ignored.	0
13 JCIP		RW	Injection conversion in progress: 1: The injection channel group conversion is in progress (towards...) JSWSTART has written 1) or an injection conversion request is in progress. Pending processing (due to triggered detection); 0: No request was issued for the conversion injection channel group (software and triggers) No requests were made. When JCIP=1, requests to initiate injection conversion are ignored.	0
[12:5] Reserved		RO	reserved.	0
4	AWDF	RW	Simulated watchdog: 1 The analog watchdog module detected a voltage exceeding the limit. R32_DFSDM_FLTxAWLTR or R32_DFSDM_FLTxAWHTR	0

			The value programmed into the register; 0: No simulated watchdog event occurred. This bit will be set to 1 by hardware, and it will be cleared to zero by software in the following way: Set all source flag bits in the R32_DFSDM_FLTxAWSR register Clear AWHTF[1:0] and AAWTF[1:0] to zero (by sending...) Write 1 to the clear bit in register R32_DFSDM_FLTxAWCFR accomplish).	
3	ROVRF	RW	Standard conversion overflow flag: 1: A normal conversion overflow occurred, meaning the normal conversion has... The process has been completed, but REOCF remains at 1. RDATAR is unaffected by overflow. ring; 0: No normal conversion overflow occurred. This bit will be set to 1 by hardware, This bit can be used by software. Clear the CLRROVRF bit in the R32_DFSDM_FLTxICR register zero.	0
2	JOVRF	RW	Injection conversion overflow flag: 1: An injection conversion overflow occurred, meaning the injection conversion has... Completed, but JEOCF remains 1. JDATAR is unaffected by overflow. ring; 0: No injection conversion overflow occurred. This bit will be set to 1 by hardware, This bit can be used by software. Clear the CLRJOVRF bit in the R32_DFSDM_FLTxICR register zero.	0
1	REOCF	RW	End of normal conversion: 1: The standard conversion has been completed and the data can be read; 0: Normal conversion not completed. This bit will be set to 1 by hardware when read by software or DMA. When R32_DFSDM_FLTxRDATAR is used, this bit is cleared to zero.	0
0	JEOCF	RW	Injection conversion end marker: 1: The injection and conversion have been completed and the data can be read; 0: Injection conversion not completed. This bit will be set to 1 by hardware when read by software or DMA. When R32_DFSDM_FLTxJDATAR is used, this bit is cleared to zero.	0

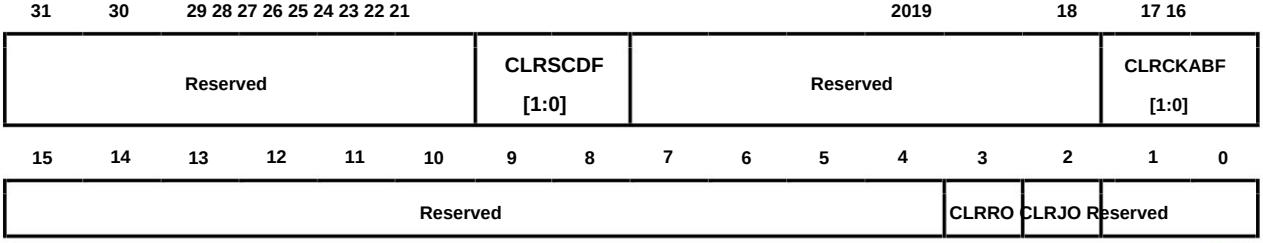
Note: For each flag bit, interrupts can be enabled by setting the corresponding bit in R32_DFSDM_FLTxCR2. If an interrupt is invoked,

1. This flag must be cleared before exiting the interrupt service routine. When DFEN=0, all bits of R32_DFSDM_FLTxISR will be automatically cleared.

Reset.

42.6.9 Clear the interrupt flag register for the DFSDM filter (R32_DFSDM_FLTxICR) (x=0/1)

Offset address: 0x40 + 0x04 * x



	VRF VRF		

Bit	name	access	describe	Reset value
[31:26] Reserved		RO reserved.		0
[25:24] CLRSCDF[1:0]		RW	Clear short-circuit detector flag: 00: Writing 0 has no effect; 01: The short-circuit detector flag on channel 0 is cleared; 10: Clear the short-circuit detector flag on channel 1; 11: Clear the short-circuit detector flags on Channel 0 and Channel 1. Note: CLRSCDF[1:0] only exists in R32_DFSDM_FLT0ICR. In the register (filter x=0).	0
[23:18] Reserved		RO reserved.		0
[17:16] CLRCKABF[1:0]		RW	Zero clock missing flag: 00: Writing 0 has no effect; 01: The clock missing flag on channel 0 is cleared; 10: The clock missing flag on channel 1 is cleared; 11: The clock missing flag on Channel 0 and Channel 1 is cleared. When the transceiver has not been synchronized, the clock missing flag will be set to 1, and Unable to be cleared by CLRCKABF. Note: CLRCKABF[1:0] only exists in R32_DFSDM_FLT0ICR In the register (filter x=0).	0
[15:4] Reserved		RO is reserved.		0
3	CLRROVRF	RW	Clear the normal conversion overflow flag: 1: Writing 1 will change the value in the R32_DFSDM_FLTxISR register. Clear the ROVRF bit; 0: Writing 0 has no effect.	0
2	CLRJOVRF	RW	Clear the injection conversion overflow flag to zero: 1: Writing 1 will change the value in the R32_DFSDM_FLTxISR register. Clear the JOVRF bit; 0: Writing 0 has no effect.	0
[1:0] Reserved		RO is reserved.		0

Note: The bit of R32_DFSDM_FLTxICR is always read as 0.

42.6.10 DFSDM Filter x Injection Channel Group Selection Register (R32_DFSDM_FLTxJCHGR) (x=0/1)

Offset address: 0x48 + 0x04 * x

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														JCHG[1:0]	

Bit	name	access	describe	Reset value
[31:2] Reserved		RO is reserved.		0
[1:0] JCHG[1:0]		RW Injection Channel Group Selection:		01b

			00: Channel 0 and Channel 1 do not belong to the injection group; 01: Channel 0 belongs to the injection group; 10: Channel 1 belongs to the injection group; 11: Both Channel 0 and Channel 1 belong to the injection group. If JSCAN=1, then convert each selected channel in turn. First Convert the smallest channel (channel 0, if selected), then convert the last channel. Maximum selected channel. If JSCAN=0, then only one of the selected channels is converted, and then The channel selection moves to the next channel. If JSCAN=0, then... JCHG will reset the channel selection to the minimum selected value when performing a write operation. aisle. At least one channel must always be selected for the injection group. If writing operations... If the write operation causes all JCHG bits to become zero, then the write operation is ignored.	

42.6.11 DFSDM Filter x Control Register 3 (R32_DFSDM_FLTxFCR3) (x=0/1)

Offset address: 0x50 + 0x04 * x

31 30 29 28 27 26 25 24 23 22 21	2019	18 17 16
FORD[2:0]	FOSR[9:0]	Reserved

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

Bit	name	access	describe	Reset value
[31:29]	FORD[2:0]	RO	Sinc filter order: 000: FastSinc filter type; 001: Sinc1 filter type; 010: Sinc2 filter type; 011: Sinc3 filter type; 100: Sinc4 filter type; 101: Sinc5 filter type; 110, 111: Retained. Modification is only possible when DFEN=0 (R32_DFSDM_FLTxCR1). That position.	0
[28:26]	Reserved	RO	is reserved.	0
[25:16]	FOSR[9:0]	RW	Sinc filter oversampling rate: 0ÿ1023: Defines the length of a Sinc type filter, the length range The range is 1 to 1024 (FOSR = FOSR[9:0] + 1). This number also represents... The decimation rate of the filter output data rate. Modification is only possible when DFEN=0 (R32_DFSDM_FLTxCR1). This position Note: If FOSR=0, the filter has no effect (filter bypass).	0
[15:4]	Reserved	RO	is reserved.	0
[3:0]	IOSR[3:0]	RW	The integrator oversampling rate is 2 IOSR. Define an integrator output data. The sample is synthesized from samples taken by a number of Sinc filters. (Integral)	0

			The output data rate of the device will be reduced by this value (extra data extraction). (Rate of use). Modification is only possible when DFEN=0 (R32_DFSDM_FLTxCR1). That position. Note: (1) If IOSR=0, the integrator has no effect (integrator) Bypass). (2) In continuous non-fast mode, IOSR must be greater than 0.	

42.6.12 DFSDM Filter x Injection Group Data Register (R32_DFSDM_FLTxJDATAR) (x=0/1)

Offset address: 0x58 + 0x04 * x

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
JDATA[23:9]														

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

Bit	name	access	describe	Reset value
[31:8]	JDATA[23:0]	RO	Inject group transformation data: The data obtained after each conversion of one channel in the injection group All data will be stored in this field. Data is valid when JEOCF=1. (Read) This register will clear the corresponding JEOCF.	0
[7:1]	Reserved		RO is reserved.	0
0	JDATACH	RO	Most recently converted injection channel: Each time a channel in the injection group is converted, it is updated. JDATACH indicates which channel was switched. Therefore, , the data held by JDATA[23:0] corresponds to the data indicated by JDATACH. aisle.	0

Note: Can be used DMA

Read data from this register. Half-word access can be used to read only the most significant 16 bits. The JEOCF register is cleared. Therefore, if activated to read data from this register. value.

If DMA is enabled, the firmware cannot read the register.

42.6.13 DFSDM Filter x General Channel Data Register (R32_DFSDM_FLTxRDATAR) (x=0/1)

Offset address: 0x60 + 0x04 * x

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
RDATA[23:8]														

15	14	13	12	11	10	9	8	7	6	5	4	3	twenty one	0
RDATA[7:0]								Reserved			RPEND	Reserved		RDATACH

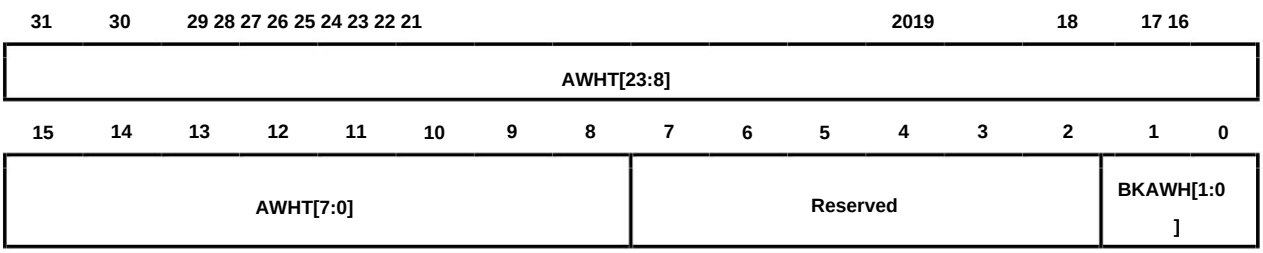
Bit	name	access	describe	Reset value

[31:8]	RDATA[23:0]	RO	Standard channel conversion data: Each time a regular transformation is completed, the corresponding data is stored in this... In the register. Data is valid when REOCF=1. Read this register. The device will clear the corresponding REOCF.	0
[7:5]	Reserved	RO	reserved. Data	0
4	RPEND	RO	pending processing in the regular channel: Caused by injection channel triggering during conversion Regular data in RDATA[23:0] is processed with a delay.	0
[3:1]	Reserved	RO	reserved.	0
0	RDATACH	RO	Recently converted regular channel: Each time a regular transformation is completed, RDATACH is updated to indicate this. Which channel was converted (because it can be updated during regular conversion) General channel selection in the R32_DFSDM_FLTxCR1 register RCH). Therefore, the data held by RDATA[23:0] corresponds to The channel indicated by RDATACH.	0

Note: Half-word access can be used to read only the most significant bit (MSB) of the converted data. Reading this register will also change the R32_DFSDM_FLTxISR value.

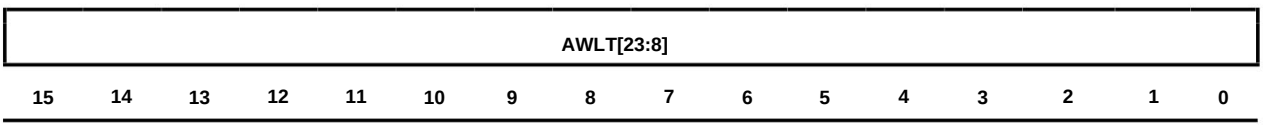
The zeroing process. REOCF

42.6.14 DFSDM Filter x Analog Watchdog Threshold Upper Limit Register (R32_DFSDM_FLTxAWHTR) (x=0/1) Offset Address: 0x68 + 0x04 * x



Bit	name	access	describe	Reset value
[31:8]	AWHT[23:0]	RW	Simulated watchdog threshold upper limit: These bits are written by software and are used to define the threshold of the simulated watchdog. Upper limit.	0
[7:2]	Reserved	RO	reserved.	0
[1:0]	BKAWH[1:0]	RW	Disconnect signal assignment for simulated watchdog threshold upper limit event: 00: The circuit breaker signal is not assigned to the upper limit of the analog watchdog threshold; 01: The open circuit 0 signal is assigned to the analog watchdog threshold upper limit event; 10: The circuit breaker 1 signal is assigned to the analog watchdog threshold upper limit event; 11: The circuit breaker signal is evenly distributed to the simulated watchdog threshold upper limit event.	0

42.6.15 DFSDM Filter x Analog Watchdog Threshold Lower Limit Register (R32_DFSDM_FLTxAWLTR) (x=0/1) Offset Address: 0x70 + 0x04 * x



AWLT[7:0]	Reserved	BKAWL[1:0]

Bit	name	access	describe	Reset value
[31:8]	AWLT[23:0]	RW	Simulated watchdog threshold lower limit: These bits are written by software and are used to define the threshold of the simulated watchdog. Lower limit. Note: For channel transceiver monitoring (AWFSEL=1), only the high 16 is supported. Bit (AWLT[23:8]) Definition 16 A bit threshold is used to compare the data from the analog watchdog filter output (because the data from the analog watchdog filter has a maximum resolution of [illegible]000 kbps). 16 (Position). In this way In this case, AWLT[7:0] is not compared.	0
[7:2]	Reserved	RO	reserved.	0
[1:0]	BKAWL[1:0]	RW	Disconnection signal assignment for simulated watchdog threshold lower limit event: 00: The circuit breaker signal is not assigned to the lower threshold event of the analog watchdog. 01: The open circuit 0 signal is assigned to the lower threshold event of the analog watchdog. 10: The circuit breaker 1 signal is assigned to the lower threshold event of the analog watchdog. 11: The circuit breaker signal is evenly distributed to the lower threshold event of the analog watchdog.	0

42.6.16 DFSDM Filter x Analog Watchdog Status Register (R32_DFSDM_FLTxAWSR) (x=0/1)

Offset address: 0x78 + 0x04 * x

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
Reserved					

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

Bit	name	access	describe	Reset value
[31:10]	Reserved	RO	reserved.	0
[9:8]	AWHTF[1:0]	RO	Simulated watchdog threshold upper limit flag: 00: There is no threshold upper limit error on either channel 0 or channel 1; 01: There is a threshold upper limit error on channel 0; 10: There is a threshold upper limit error on channel 1; 11: There is a threshold upper limit error in both Channel 0 and Channel 1. This bit is set to 1 by hardware. This bit can also be used by software. The corresponding CLRAWHTF in the R32_DFSDM_FLTxAWCFR register Clear the bits to zero.	0
[7:2]	Reserved	RO	is reserved.	0
[1:0]	AWLTF[1:0]	RO	Simulated watchdog threshold lower limit flag: 00: There is no lower threshold error in either channel 0 or channel 1; 01: There is a threshold lower limit error in channel 0; 10: There is a threshold lower limit error in channel 1; 11: There are threshold lower limit errors in both Channel 0 and Channel 1. This bit is set to 1 by hardware. This bit can also be used by software.	0

			The corresponding CLRAWLTF in the R32_DFSDM_FLTxAWCFR register Clear the bits to zero.	

Note: When DFEN=0, all bits of R32_DFSDM_FLTxAWSR will be automatically reset.

42.6.17 DFSDM Filter x Analog Watchdog Clear Flag Register (R32_DFSDM_FLTxAWCFR) (x=0/1) Offset Address: 0x80 + 0x04 * x

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

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

Bit	name	access	describe	Reset value
[31:10]	Reserved	RO	is reserved.	0
[9:8]	CLRAWHTF[1:0]	RW1Z	Clear the simulated watchdog threshold upper limit flag: 00: Writing 0 has no effect; 01: The threshold upper limit error flag for channel 0 is cleared to zero; 10: Clear the threshold upper limit error flag for Channel 1; 11: Clear the upper limit error flag for Channel 0 and Channel 1.	0
[7:2]	Reserved	RO	is reserved.	0
[1:0]	CLRAWLTF[1:0]	RW1Z	Clear the simulated watchdog threshold lower limit flag: 00: Writing 0 has no effect; 01: The lower threshold error flag for channel 0 is cleared; 10: Clear the lower threshold error flag for Channel 1; 11: Clear the lower threshold error flags for Channel 0 and Channel 1.	0

42.6.18 DFSDM filter x Extreme value detector maximum value register (R32_DFSDM_FLTxEXMAX) (x=0/1) offset address: 0x88 + 0x04 * x

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
EXMAX[23:8]														

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

Bit	name	access	describe	Reset value
[31:8]	EXMAX[23:0]	RZ	Maximum value of extreme value detector: These bits are set to 1 by hardware to indicate the conversion of DFSDM_FLTx. The maximum value. The EXMAX[23:0] bits are reset by reading this register. The value is (0x800000).	0
[7:1]	Reserved	RO	is reserved.	0
0	EXMAXCH	RZ	Extreme Value Detector Maximum Value Data Channel:	0

			These bits contain information about which channel's data is being stored. Up to EXMAX[23:0]. These bits can be cleared by reading this register. zero.	

42.6.19 DFSDM filter x Extreme value detector minimum value register (R32_DFSDM_FLTxEXMIN) (x=0/1) offset address: 0x90 + 0x04 * x

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
EXMIN[23:8]														

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

Bit	name	access	describe	Reset value
[31:8]	EXMIN[23:0]	RO	Minimum value of extreme value detector: These bits are set to 1 by hardware to indicate the conversion of DFSDM_FLTx. The minimum value. The EXMIN[23:0] bits are reset by reading this register. The value is (0x7FFFFF).	0
[7:1]	Reserved	RO	is reserved.	0
0	EXMINCH	RO	Minimum data channel for extreme value detector: These bits contain information about which channel's data is being stored. Up to EXMIN[23:0]. These bits can be cleared by reading this register. zero.	0

42.6.20 DFSDM Filter x Conversion Timer Register (R32_DFSDM_FLTxCNVTIMR) (x=0/1)

Offset address: 0x98 + 0x04 * x

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved											CNVCNT[27:16]			

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

Bit	name	access	describe	Reset value
[31:28]	Reserved	RO is		0
[27:0]	CNVCNT[27:0]	RO	reserved. 28-bit timer count transition time: CNVCNT = (CNVCNT[27:0] + 1) t=CNVCNT/fDfsdmclk; The timer's input clock comes from the DFSDM clock (system clock). fDFSDMCLK). Conversion time measurement begins at the start of each conversion and ends at... The process ends after every 36 conversions (i.e., the first and last serial sampling). (The interval between samples). Only when the filter is bypassed. The conversion time measurement will only stop when FOSR[9:0]=0. Stop, and CNVCNT[27:0]=0. The timekeeping is as follows:	0

			If FAST=0 (or FAST=1 for the first conversion in continuous mode), then: t=[FOSR*(2 IOSR+FORD+1)]/fCKIN (for Sincx filters); t=[FOSR*(2 IOSR+5)]/fCKIN (for FastSinc filters); If FAST=1 in continuous mode (except for the first conversion), then: t=[FOSR*2IOSR]/fCKIN; If FOSR=FOSR[9:0]+1=1 (filter bypass, only integrator active), then: CNVCNT=0 (timer stopped, conversion time: t=2IOSR/fCKIN); where fCKIN is the channel input clock frequency (on the given channel CKINy pin); in the case of parallel data input (from internal ADC or from CPU/DMA write operation), it represents the input Note: When the conversion is interrupted (e.g., by disabling/enabling the selected channel), the timer will also take the interruption time into account.	

Chapter 43 LCD-TFT Display Controller (LTDC)

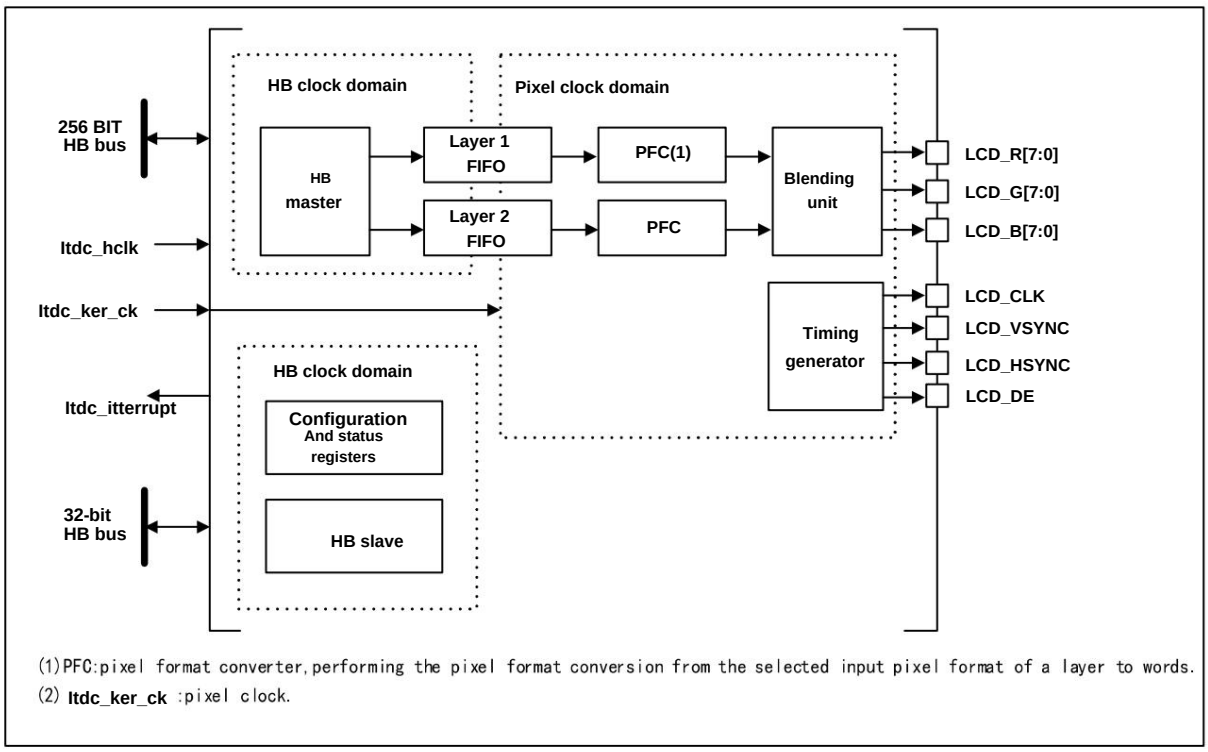
LCD-TFT (Thin Film Transistor-Liquid Crystal Display) display controllers (LTDC) primarily provide parallel digital RGB and horizontal synchronization. The vertical synchronization, pixel clock, and data enable signals can be used as output signals to various LCD and TFT panel interfaces.

43.1 Main Features

• Provides two display layers, each containing a proprietary 8*256-bit FIFO. • Supports color lookup tables (CLUTs), with up to 256 colors per layer . • Supports programmable timing for different display panels . • Supports programmable background colors . • Supports programmable HSYNC, VSYNC, and data enable signal polarity . • Each display layer can select up to 8 input color formats . • Supports flexible blending between two layers using alpha values (per pixel or constant) . • Supports chroma keying (transparent colors) . • Supports programmable window position and size . • Supports thin-film transistor (TFT) color displays . • Supports up to 3 programmable interrupt times.

43.2 Overview

Figure 43-1 Block diagram of LTDC



(1) PFC: pixel format converter, performing the pixel format conversion from the selected input pixel format of a layer to words.
(2) ltdc_ker_ck :pixel clock.

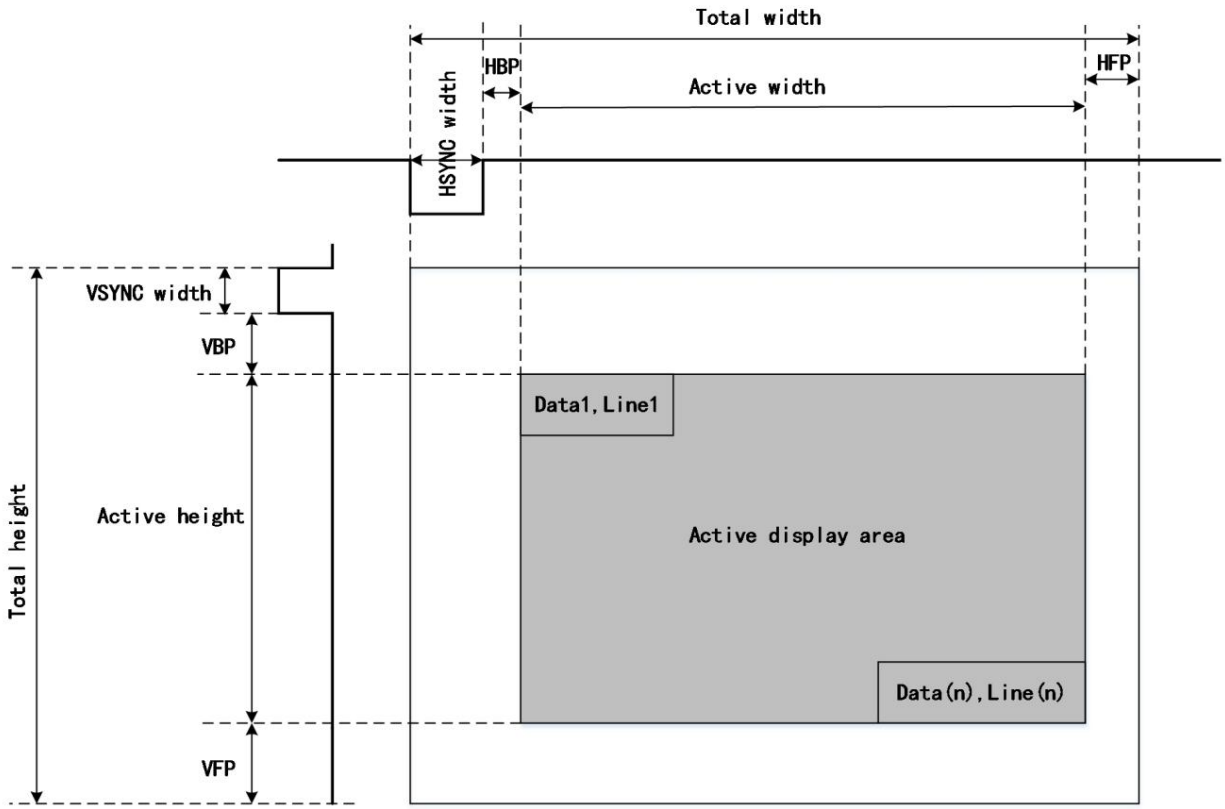
43.3 Functional Description

43.3.1 LTDC Synchronization Timing

The LTDC interface is a synchronous data interface that includes pixel clock, pixel data, and horizontal and vertical synchronization signals. (Figure 43-2)

This demonstrates the configurable timing parameters generated by the LTDC synchronous timing generator module.

Figure 43-2 LTDC Synchronization Timing



Note: (1) HSYNC and VSYNC widths are the horizontal and vertical synchronization widths, respectively; HBP and HFP are the horizontal trailing edge period and horizontal... VFP and VBP are the leading edge period and vertical trailing edge period, respectively.

(2) Enable LTDC When the timing sequence is generated, the timing generator module is reset to X = total width - 1 and Y = total height - 1, and during the vertical synchronization phase and... LTDC At that time, the timing generator module is reset to X = total width - 1 and Y = total height - 1, and during the vertical synchronization phase and... FIFO and trailing edge. (3) Prohibit The previous pixel is retained before refreshing. Therefore, only the blanking data is continuously output.

The following is an example of a synchronization timing configuration:

1) LCD-TFT timing (should be extracted from the panel datasheet);

2) Horizontal and vertical synchronization width: 0xA pixels, 0x2-1 lines;

3) Horizontal and vertical trailing edges: 0x14 pixels, 0x2 rows;

4) Effective width and effective height: 0x140 pixels, 0xF0 rows (320x240);

5) Horizontal leading edge: 0xA pixel;

6) Vertical leading edge: 0x4 rows;

The value programmed into the LTDC timing register will be:

1) R32_LTDC_SSCR register: will be programmed to 0x00090001. (HSW[11:0] is 0xA-1 and VSH[10:0] is 0x2-1);

2) R32_LTDC_BPCR register: will be programmed to 0x001D0003. (AHBP[11:0] is 0x1D (0xA + 0x14-1), AVBP[10:0]

It is 0x3 (0x2 + 0x2-1));

3) R32_LTDC_AWCR register: will be programmed to 0x015D00F3; (AAW[11:0] is 0x15D (0xA + 0x14 + 0x140-1).

AAH[10:0] is 0xF3 (0x2 + 0x2 + 0xF0 - 1));

4) R32_LTDC_TWCR register: will be programmed to 0x00000167. (TOTALW[11:0] is 0x167 (0xA + 0x14 + 0x140 + 0xA-1));

5) R32_LTDC_TWCR register: will be programmed to 0x000000F7. (TOTALH[10:0] is 0xF7 (0x2 + 0x2 + 0xF0 + 0x4-
1).

43.3.2 Pixel Format The

programmable pixel format is used for data stored in the layer's frame buffer. It can be configured for each layer via the R32_LTDC_LxPFCR register.
Up to 8 input pixel formats.

The ARGB8888 format requires 8 bits of data per channel (Alpha, Red, Green, and Blue). However, ARGB1555 and ARGB4444...
Some channels of the format are less than 8 bits. The pixel processing unit converts it to ARGB8888 by copying the high bits and padding the low bits.
When processing RGB888 and RGB565 formats, the pixel processing unit assumes Alpha=255, and if the number of bits per channel is less than 8, it will also...
Copy the high-order bits and fill them into the low-order bits.

43.3.3 Color Lookup Table (CLUT)

CLUT can be enabled for each layer at runtime via the CLUTEN bit of the R32_LTDC_LxCR register. CLUT only applies when using L8 and AL44.
When using the AL88 input pixel format, it is suitable for indexed colors.

First, CLUT must load the original R, G, and B values used to replace the corresponding pixels (indexed colors). Each color...
Each RGB value has its own corresponding address within the CLUT. The R, G, and B values and their respective addresses are all assigned via R32_LTDC_LxCLUTWR.
Register programming.

When using L8 and AL88 input pixel formats , CLUT must load 256 colors. The address of each color is in
Configured in the CLUTADD bit of the R32_LTDC_LxCLUTWR register.

When using the AL44 input pixel format, CLUT must load only 16 colors. The address of each color must be passed through a 4-bit L-pass filter.
The repeating bits are padded with 8 bits, as follows:

–L0 (index color 0), at address 0x00;

–L1, at address 0x11;

–L2, at address 0x22;

–.....ÿ

–L15, at address 0xFF.

43.3.4 Color Frame Buffer 43.3.4.1

Color Frame Buffer Address Each layer's

color frame buffer has a starting address, which is configured through the R32_LTDC_LxCFBAR register. When
When a layer is enabled, the data will be retrieved from the color frame buffer.

43.3.4.2 Color Frame Buffer Length Each

layer sets the total line length (in bytes) and number of lines of the color frame buffer, which can be set via R32_LTDC_LxCFBLR and...
Configure the R32_LTDC_LxCFBLNR register.

The settings for line length and number of lines are used to prevent data that exceeds the end of the frame buffer from being prefetched into the FIFO corresponding to the layer.

If set to less than the required byte, a FIFO underflow interrupt will be generated (if it was previously enabled).

If set to a value higher than the actual required number of bytes, useless data read from the FIFO will be discarded and will not be displayed.

43.3.4.3 Color Frame Buffer Spacing Each

layer of color frame buffers has a configurable spacing, which is the distance between the beginning of one line and the beginning of the next line (in bytes).
It is configured via the R32_LTDC_LxCFBLR register.

43.3.5 Layered Windows and Mixed Functionality

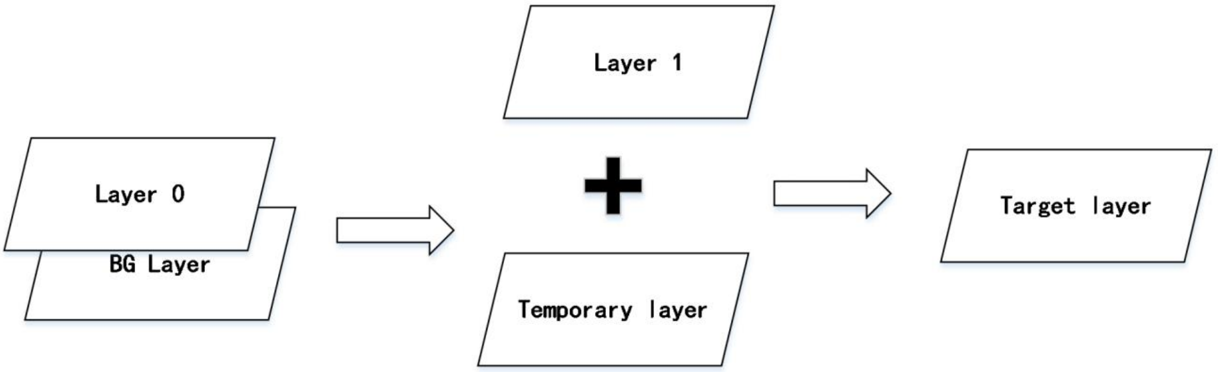
Each layer supports windowing functionality as well as the ability to blend two layers. LTDC first performs windowing operations for each layer, and then blends the two layers into one.

frame.

The window function defines a display window, with each layer having its own independent window position configuration register R32_LTDC_LxWHPCR and R32_LTDC_LxWVPCR. The window configuration defines the display window within a layer. Pixels within the window will retain their original values, but pixels outside the window... The pixel value will be replaced by the pixel value defined in the R32_LTDC_LxDCCR register.

The blending order is fixed, from bottom to top. The blending unit first blends layer 0 and the background layer to create a temporary layer, then blends layer 1 and... Temporary layer, to obtain the target layer. The ARGB values of the background layer are defined in the R32_LTDC_BCCR register. If a layer is disabled, blending functionality... Use the default color for this layer. To prevent the default color from being displayed when the layer is disabled, the blending mode for this layer in the R32_LTDC_LxBFCR register must be set. The number is set to its reset value. The mixing process is shown in Figure 43-3.

Figure 43-3 Flowchart of the mixing process



General mixing formula: BC = BF1 * C + BF2 * CS;

BC represents the blended color; BF1 represents the blend factor 1; C represents the current layer color; BF2 represents the blend factor 2; CS represents the color of the next layer. color.

The blending factor for the current pixel has two possible values, configured by a register. One is the pixel alpha multiplied by a constant alpha, and the other... One is the constant Alpha.

Note: Constant values are values programmed in registers and implemented in hardware. For example, Alpha 255: only enable the first layer, BF1 is equal parts.

configured as constant Alpha, and BF2 is configured as 1-constant Alpha. Constant Alpha: The constant

programmed in the LxCACR register is 240 (0xF0).

Therefore, the constant Alpha Value is

240/255=0.94 C: The red channel of the current layer is 0x80

Cs: Background red channel is 0x30

The first layer blends with the background color. Final result for the red channel = The constant Alpha * C + (1-constant Alpha) * Cs

= 0.94 * 0x80 + (1 - 0.94) * 0x30 = 0x7B.

is used for the green and blue channels.

43.3.6 Color Keys Color

keys (RGB) can be configured to represent transparent pixels. At runtime, color key values can be configured and used to replace the pixel RGB values.

When chroma keying is enabled, the current pixel (the pixel after format conversion and before CLUT mixing) will be compared with the chroma key. If the current pixel... If the pixel matches the programmed RGB value, then all channels (ARGB) of that pixel are set to 0.

The color key is configured via the LTDC_LxCKCR register. The programmed value depends on the pixel format, as it will be converted to ARGB888 upon pixel format conversion. Then compare it with the current pixel.

For example, if medium yellow (50% red + 50% green) is used as the transparency color bond, then:

- In RGB565, the value for medium yellow is 0x8400. Set LTDC_LxCKCR to 0x848200.

- In ARGB8888, the mid-yellow color is 0x808000, so set LTDC_LxCKCR to 0x808000.

- In all CLUT-based color modes (L8, AL88, AL44), set one of the palette entries to medium yellow 0x808000.

Set LTDC_LxCKCR to 0x808000.

43.3.7 Overloading Shadow Registers Some

configuration registers perform shadow operations. This occurs when performing a write operation on the active register, or when configuring the R32_LTDC_SRCR register.

At the start of the vertical blanking cycle following the segment, the shadow register value can be immediately reloaded into the active register. If immediate reload is selected...

If set, the overload should only be activated after all new registers have completed their write operations.

The shadow register should not be modified again before the reload is complete. Reading the shadow register will return the actual valid value. Newly written values can only be accessed within the shadow register.

Read after reload occurs.

If the corresponding enable is made in the R32_LTDC_IER register, a register reload interrupt can be generated.

The shadow registers are all Level 1 and Level 2 registers, except for the R32_LTDC_LxCLUTWR register.

43.3.8 Interrupt LTDC

provides three maskable interrupts, which are logically ORed into one interrupt vector.

Interrupt sources can be individually enabled or disabled via the R32_LTDC_IER register; the corresponding interrupt can be read via the R32_LTDC_ISR register.

Flags; the corresponding interrupt flag can be cleared by setting the R32_LTDC_ICR register to 1.

An interrupt will occur when the following events happen:

Row interrupt: This occurs when the programmed row is reached. The location of the row interrupt is programmed in the R32_LTDC_LIPCR register.

ÿ Register reload interrupt: occurs when shadow register reloading is performed during the vertical blanking cycle;

ÿ FIFO underflow interruption: Occurs when requesting a pixel from an empty layer FIFO.

43.4 Register Description

Table 38-1 List of LTDC-related registers

name	Access Address Description: 0x40014800 LTDC		Reset value
R32_LTDC_SSCR	Synchronization Size Configuration Register; 0x40014804 LTDC		0x00000000
R32_LTDC_BPCR	Trailing Edge Configuration Register; 0x40014808 LTDC		0x00000000
R32_LTDC_AWCR	Valid Width Configuration Register; 0x4001480C LTDC Total		0x00000000
R32_LTDC_TWCR	Width Configuration Register; 0x40014810 LTDC Global		0x00000000
R32_LTDC_GCR	Control Register; 0x40014814 LTDC Shadow Reload		0x00002220
R32_LTDC_SRCR	Configuration Register; 0x40014818 LTDC Background Color		0x00000000
R32_LTDC_BCCR	Configuration Register; 0x4001481C LTDC Interrupt Enable		0x00000000
R32_LTDC_IER	Register; 0x40014820 LTDC Interrupt Status Register;		0x00000000
R32_LTDC_ISR	0x40014824 LTDC Interrupt Clear Register; 0x40014828		0x00000000
R32_LTDC_ICR	LTDC Line Interrupt Position Configuration Register;		0x00000000
R32_LTDC_LIPCR	0x00000000		
R32_LTDC_CPSR	0x4001482C LTDC Current Position Status Register; 0x40014830		0x00000000
R32_LTDC_CDSR	LTDC Current Display Status Register; 0x40014834 LTDC Level 1		0x0000000F
R32_LTDC_L1CR	Control Register		0x00000000
R32_LTDC_L1WHPCR	0x40014838	LTDC Level 1 Window Horizontal Position Configuration Register	0x00000000
R32_LTDC_L1WVPCR	0x4001483C	LTDC Level 1 Window Vertical Position Configuration	0x00000000
R32_LTDC_L1CKCR	Register 0x40014840 LTDC Level 1 Color Key Configuration Register 0x00000000		
R32_LTDC_L1PFCR	0x40014844 LTDC Layer 1 Pixel Format Configuration Register 0x00000000		

R32_LTDC_L1CACR	0x40014848	LTDC Level 1 Constant Alpha Configuration Register	0x00000000
R32_LTDC_L1DCCR	0x4001484C	LTDC Level 1 Default Color Configuration Register	0x00000000
R32_LTDC_L1BFCR	0x40014850	LTDC Level 1 Mixing Coefficient Configuration Register	0x00000600
R32_LTDC_L1CFBAR	0x40014854	LTDC Level 1 Color Frame Buffer Address Register	0x00000000
R32_LTDC_L1CFBLR	0x40014858	LTDC Level 1 Color Frame Buffer Length Register	0x00000000
R32_LTDC_L1CFBLNR	0x4001485C	LTDC Level 1 Color Frame Buffer Line Number Register	0x00000000
R32_LTDC_L1CLUTWR	0x40014860	LTDC Level 1 CLUT Write Register;	0x00000000
R32_LTDC_L2CR	0x40014864	LTDC Level 2 Control Register	0x00000000
R32_LTDC_L2WHPCR	0x40014868	LTDC Level 2 Window Horizontal Position Configuration Register	0x00000000
R32_LTDC_L2WVPCR	0x4001486C	LTDC Level 2 Window Vertical Position Configuration	0x00000000
R32_LTDC_L2CKCR	Register 0x40014870	LTDC Layer 2 Color Key Configuration Register	0x00000000
R32_LTDC_L2PFCR	0x40014874	LTDC Layer 2 Pixel Format Configuration Register	0x00000000
R32_LTDC_L2CACR	0x40014878	LTDC Level 2 Constant Alpha Configuration Register	0x00000000
R32_LTDC_L2DCCR	0x4001487C	LTDC Level 2 Default Color Configuration Register	0x00000000
R32_LTDC_L2BFCR	0x40014880	LTDC Level 2 Blending Coefficient Configuration Register	0x00000600
R32_LTDC_L2CFBAR	0x40014884	LTDC Layer 2 Color Frame Buffer Address Register	0x00000000
R32_LTDC_L2CFBLR	0x40014888	LTDC Layer 2 Color Frame Buffer Length Register	0x00000000
R32_LTDC_L2CFBLNR	0x4001488C	LTDC Layer 2 Color Frame Buffer Line Number Register	0x00000000
R32_LTDC_L2CLUTWR	0x40014890	LTDC Level 2 CLUT Write Register	0x00000000

43.4.1 LTDC Synchronization Size Configuration Register (R32_LTDC_SSCR)

Offset address: 0x00

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
Reserved			HSW[11:0]		

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

Bit	name	access	describe	Reset value
[31:28]	Reserved	RO	reserved.	0
[27:16]	HSW[11:0]	RW	Horizontal sync width (in pixel clock cycles): This field defines the number of horizontally synchronized pixels minus 1.	0
[15:11]	Reserved	RO	is reserved.	0
[10:0]	VSH[10:0]	RW	Vertical Synchronization Height (in units of horizontal scan lines):	0

			This field defines the vertical synchronization height minus 1. It represents the horizontal synchronization line. number.	

43.4.2 **Offset address of LTDC trailing edge configuration register**

(R32_LTDC_BPCR) : 0x04

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved											AHBP[11:0]			

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

Bit	name	access	describe	Reset value
[31:28]	Reserved	RO is		0
[27:16]	AHBP[11:0]	RW	reserved. Accumulate horizontal trailing edges (in pixel clock cycles): This field defines the accumulated horizontal trailing edge width (horizontal synchronous pixel water addition). (Subtract 1 from the flat back edge pixel). The horizontal trailing edge is when the horizontal sync signal becomes invalid before the next scan line. The effective display interval between the start and end.	0
[15:11]	Reserved	RO preserved.		0
[10:0]	AVBP[10:0]	RW	Accumulate vertical trailing edges (in units of horizontal scan lines): This field defines the cumulative vertical trailing edge width (vertical synchronization line plus vertical). Subtract 1 from the next row. The vertical trailing edge is the first valid scan line opening from the start of one frame to the next. The number of horizontal scan lines initially included.	0

43.4.3 LTDC Effective Width Configuration Register (R32_LTDC_AWCR)

Offset address: 0x08

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved											AAW[11:0]			

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

Bit	name	access	describe	Reset value
[31:28]	Reserved	RO is		0
[27:16]	AAW[11:0]	RW	reserved. Accumulate the effective width (in pixel clock cycles): This field defines the cumulative effective width (horizontal synchronization pixels plus horizontal width). Add 1 to the effective pixel along the pixel (subtract 1 from the effective pixel). The effective width is the number of pixels in the effective display area of the panel scan line. number.	0
[15:11]	Reserved	RO reserved.		0
[10:0]	AAH[10:0]	RW	Accumulate effective height (in horizontal scan lines): This field defines the cumulative height (vertical synchronous line plus vertical subsequent line plus). (Effective height rows minus 1). Effective height is the number of effective rows in the panel.	0

43.4.4 LTDC Total Width Configuration Register (R32_LTDC_TWCR)

Offset address: 0x0C

31 30 29 28 27 26 25 24 23 22 21 2019 18 17 16	
Reserved	TOTALW [11:0]

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

Bit	name	access	describe	Reset value
[31:28]	Reserved	RO	reserved.	0
[27:16]	TOTALW[11:0]	RW	Total width (in pixel clock cycles): This field defines the cumulative total width (horizontal synchronization pixels plus horizontal trailing edge). (Pixels plus effective width plus horizontal leading edge pixels minus 1).	0
[15:11]	Reserved	RO	reserved.	0
[10:0]	TOTALH[10:0]	RW	Total height (in horizontal scan lines): This field defines the cumulative height (vertical synchronous line plus vertical subsequent line plus). Effective height plus vertical leading edge line minus 1).	0

43.4.5 LTDC Global Control Register (R32_LTDC_GCR)

Offset Address: 0x10

31 30 29 28 27 26 25 24 23 22 21 2019 18 17 16			
HSPOL	VSPOL	DEPOL	PCPOL	Reserved

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

Bit	name	access	describe	Reset value
31	HSPOL	RW	Horizontal synchronization polarity: 1: Horizontal synchronization polarity is active high; 0: Horizontal synchronization polarity, active low.	0
30	VSPOL	RW	Vertical synchronization polarity: 1: Vertical synchronization is active high; 0: Vertical synchronization is active low.	0
29	DEPOL	RW	Non-data-enabled polarity: 1: Non-data enable polarity, active high; 0: Non-data enable polarity, active low.	0
28	PCPOL	RW	Pixel clock polarity: 1: Pixel clock is active high; 0: Pixel clock polarity, active low.	0
[27:1]	Reserved	RO	is reserved.	0x444
0	LTDCEN	RW	LCD-TFT controller enable bit: 1: Enable;	0

			0: Prohibited.	

43.4.6 LTDC Shadow Overload Configuration Register (R32_LTDC_SRCR)

Offset address: 0x14

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

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

Bit	name	access	describe	Reset value
[31:2]	Reserved	RO	is reserved.	0
1	VBR	RW	Vertical blanking reload: 1: The shadow register during the vertical blanking cycle (after the effective display area) (starting from the first line) Internal overload; 0: No effect. Note: This bit is set to 1 by software and is only cleared by hardware after a reload.	0
0	IMR	RW	Reload immediately: 1: Shadow register is immediately reloaded; 0: No effect. Note: This bit is set to 1 by software and is only cleared by hardware after a reload.	0

Note: When reading back a valid value from the shadow register, the old value will not be read until the reload is complete.

43.4.7 LTDC Background Color Configuration Register (R32_LTDC_BCCR)

Offset address: 0x18

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
											BCRED[7:0]			

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

Bit	name	access	describe	Reset value
[31:24]	Reserved	RO	is reserved.	0
[23:16]	BCRED[7:0]	RW	background red value.	0
[15:8]	BCGREEN[7:0]	RW	background green value.	0
[7:0]	BCBLUE[7:0]	RW	background blue value.	0

43.4.8 LTDC Interrupt Enable Register (R32_LTDC_IER) Offset

Address: 0x1C

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

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

Reserved	RRIE	Reserved	FUIE	LIE

Bit	name	access	describe	Reset value
[31:4]	Reserved	RO	is reserved.	0
3	RRIE	RW	Register reload interrupt enabled: 1: Enable; 0: Prohibited.	0
2	Reserved	RO	is reserved.	0
1	FUIE	RW	FIFO underflow interrupt enable: 1: Enable; 0: Prohibited.	0
0	LIE	RW	Line interrupt enable: 1: Enable; 0: Prohibited.	0

43.4.9 LTDC Interrupt Status Register (R32_LTDC_ISR) Offset

Address: 0x20

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
Reserved					

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

Bit	name	access	describe	Reset value
[31:4]	Reserved	RO	is	0
3	RRIF	RO	reserved. Register reload interrupt flags: 1: When a vertical blanking overload occurs (and after reaching the effective region) A register reload interrupt is generated during the first line. 0: No register reload interrupt was generated.	0
2	Reserved	RO	is reserved.	0
1	FUIF	RO	FIFO underflow interruption flag: 1: When one of the layer FIFOs is empty and the number of pixels is read from the FIFO. When this happens, a FIFO underflow interrupt will occur; 0: No FIFO underflow interrupt was generated.	0
0	LIF	RO	Line interruption flag: 1: A line interrupt is generated when the programmed line is reached; 0: No line interruption occurred.	0

43.4.10 LTDC Interrupt Clear Register (R32_LTDC_ICR) Offset

Address: 0x24

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16

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

Bit	name	access	describe	Reset value
[31:4]	Reserved	RO	is reserved.	0
3	CRRIF		The RW1Z register is a reload interrupt clear flag. Writing 1 clears it to 0, writing 0	
2	Reserved		invalidates it. 0 RO is reserved.	0
1	CFUIF	RW1Z	FIFO underflow interrupt clear flag: write 1 to clear, write 0 to invalidate.	
0	CLIF	RW1Z	row interrupt clear flag: write 1 to clear, write 0 to invalidate.	0

43.4.11 LTDC Row Interrupt Location Configuration Register (R32_LTDC_LIPCR)

Offset Address: 0x28

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											LIPOS[10:0]				

Bit	name	access	describe	Reset value
[31:4]	Reserved	RO	is reserved.	0
[10:0]	LIPOS[10:0]	RW	line interrupt location.	0

43.4.12 LTDC Current Position Status Register (R32_LTDC_CPSR) Offset

Address: 0x2C

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
CXPOS[15:0]														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CYPOS[15:0]															

Bit	name	access	describe	Reset value
[31:16]	CXPOS[15:0]	RW	Current X position.	0
[15:0]	CYPOS[15:0]	RW	Current Y position.	0

43.4.13 LTDC Current Display Status Register (R32_LTDC_CDSR) Offset

Address: 0x30

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

Reserved	HSYNCS	VSYNCS	HDES	VDES

Bit	name	access	describe	Reset value
[31:4] Reserved		RO reserved.		0
3	HSYNCS	R0	Horizontal synchronization display status: 1: High level is active; 0: Active low.	1
2	VSYNCS	RO	Vertical synchronization display status: 1: High level is active; 0: Active low.	1
1	HDES	R0	Horizontal data display enabled status: 1: High level is active; 0: Active low.	1
0	VDES	R0	Vertical data display enabled status: 1: High level is active; 0: Active low.	1

43.4.14 LTDC Level x Control Register (R32_LTDC_LxCR) (x=1/2) Offset Address: 0x34 + 0x30*(x-1)

31	30 29 28 27 26 25 24 23 22 21	2019	18	17 16
Reserved				

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

Bit	name	access	describe	Reset value
[31:5] Reserved		RO reserved.		0
4	CLUTEN	RW	Color table lookup enabled: 1: Enable; 0: Prohibited. Note: CLUT is only available for L8, AL44, AL88 pixel format.	0
[3:2] Reserved		RO reserved.		0
1	COLKEN	RW	Chroma key enabled: 1: Enable; 0: Prohibited.	0
0	LEN	RW	Layer enable: 1: Enable; 0: Prohibited.	0

43.4.15 LTDC Level x Window Horizontal Position Configuration Register (R32_LTDC_LxWHPCR) (x=1/2)

Offset address: 0x38 + 0x30*(x-1)

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

Reserved	WHSPPOS[11:0]

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

Bit	name	access	describe	Reset value
[31:28]	Reserved	RO	reserved.	0
[27:16]	WHSPPOS[11:0]	RW	Window horizontal end position: WHSPPOS[11:0] >= AHBP[11:0] + 1.	0
[15:12]	Reserved	RO	reserved.	0
[11:0]	WHSTPOS[11:0]	RW	Window horizontal starting position: WHSTPOS[11:0] <= AAW[11:0].	0

For example, the R32_LTDC_BPCR register is configured as 0x000E0005 (AHBP[11:0] is 0xE), and the R32_LTDC_AWCR register is configured as follows:

The value is 0x028E01E5 (AAW[11:0] is 0x28E). To configure the horizontal position of a window with a size of 630x460 (within the valid data area),...

The horizontal starting offset is 5 pixels.

1) The first pixel of the layer window: WHSTPOS[11:0] should be programmed as 0x14 (0xE+1+0x5);

2) The last pixel of the layer window: WHSPPOS[11:0] should be programmed as 0x28A.

43.4.16 LTDC Window Vertical Position Configuration Register for Layer x (R32_LTDC_LxWVPCR) (x=1/2)

Offset address: 0x3C + 0x30*(x-1)

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved											WVSPPOS[10:0]			

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

Bit	name	access	describe	Reset value
[31:27]	Reserved	RO	is reserved.	0
[26:16]	WVSPPOS[10:0]	RW	Window vertical end position: WVSPPOS[10:0] >= AVBP[10:0] + 1.	0
[15:11]	Reserved	RO	is reserved.	0
[10:0]	WVSTPOS[10:0]	RW	Window vertical start position: WVSTPOS[10:0] <= AAH[10:0].	0

For example, the R32_LTDC_BPCR register is configured as 0x000E0005 (AVBP[10:0] is 0x5), and the R32_LTDC_AWCR register is configured as follows:

The value is 0x028E01E5 (AAH[10:0] is 0x1E5). To configure the vertical position of a window with a size of 630x460 (within the valid data area)...

(The vertical starting offset is 8 rows).

1) The first line of the layer window: WVSTPOS[10:0] should be programmed as 0xE (0x5 + 1 + 0x8);

2) The last line of the layer window: WVSPPOS[10:0] should be programmed as 0x1DA.

43.4.17 LTDC Layer x Color Key Configuration Register (R32_LTDC_LxCKCR) (x=1/2) Offset Address:

0x40 + 0x30*(x-1)

31	30	29	28	27	26	25	24	23	twenty two	twenty one	20	19	18	17	16
Reserved											CKRED[7:0]				

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

Bit	name	access	describe	Reset value
[31:24]	Reserved	RO	is reserved.	0
[23:16]	CKRED[7:0]	RW	is the red value of the chromaticity key.	0
[15:8]	CKGREEN[7:0]	RO	is the green value of the chromaticity bond.	0
[7:0]	CKBLUE[7:0]	RW	is the blue value of the chromaticity key.	0

43.4.18 LTDC Layer x Pixel Format Configuration Register (R32_LTDC_LxPFCR) (x=1/2) Offset Address:

0x44 + 0x30*(x-1)

31	30	29	28	27	26	25	24	23	twenty two	twenty one	20	19	18	17	16
Reserved															

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

Bit	name	access	describe	Reset value
[31:3]	Reserved	RO	reserved.	0
[2:0]	PF[2:0]	RW	Pixel format: 000: ARGB8888; 001: RGB888; 010: RGB565; 011: ARGB1555; 100: ARGB4444; 101: L8 (8-bit Luminance); 110: AL44 (4-bit Alpha, 4-bit Luminance); 111: AL88 (8-bit Alpha, 8-bit Luminance).	0

43.4.19 LTDC Level x Constant Alpha Configuration Register (R32_LTDC_LxCACR) (x=1/2)

Offset address: 0x48 + 0x30*(x-1)

31	30	29	28	27	26	25	24	23	twenty two	twenty one	20	19	18	17	16
Reserved															

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

Bit	name	access	describe	Reset value
[31:8]	Reserved	RO	is retained.	0
[7:0]	CONSTA[7:0]	RW	Constant Alpha: This field configures the constant Alpha used during mixing. The constant Alpha is determined by... The hardware implements 255 equal divisions. Example: If the programming constant Alpha is 0xFF, then the constant...	0xFF

			The alpha value is 255/255=1.	

43.4.20 LTDC Layer x Default Color Configuration Register (R32_LTDC_LxDCCR) (x=1/2) Offset address: 0x4C + 0x30*(x-1)

31	30	29	28	27	26	25	24	23	twenty two	twenty one	20	19	18	17	16
DCALPHA[7:0]											DCRED[7:0]				

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

Bit	name	access	describe	Reset value
[31:24]	DCALPHA[7:0]	RW	is the default Alpha value.	0
[23:16]	DCRED[7:0]	RW	's default color is red.	0
[15:8]	DCGREEN[7:0]	RW	's default color is green.	0
[7:0]	DCBLUE[7:0]	RW	's default color is blue.	0

43.4.21 LTDC Layer x Hybrid Coefficient Configuration Register (R32_LTDC_LxBFCR) (x=1/2) Offset Address: 0x50 + 0x30*(x-1)

31	30	29	28	27	26	25	24	23	twenty two	twenty one	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					BF1[2:0]			Reserved					BF2[2:0]		

Bit	name	access	describe	Reset value
[31:11]	Reserved	RO	retained.	0
[10:8]	BF1[2:0]	RW	Mixing factor 1: 100: constant Alpha; 110: Pixel Alpha * Constant Alpha; Other: Reserved.	110b
[7:3]	Reserved	RO	retained.	0
[2:0]	BF2[2:0]	RW	Mixing factor 2: 101: 1 - constant Alpha; 111: 1-pixel Alpha * constant Alpha; Other: Reserved.	0

43.4.22 LTDC Layer x Color Frame Buffer Address Register (R32_LTDC_LxCFBAR) (x=1/2)

Offset address: 0x54 + 0x30*(x-1)

31	30	29	28	27	26	25	24	23	twenty two	twenty one	20	19	18	17	16
CFBADD[31:16]															

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

Bit	name	Access	description: RW color frame buffer	Reset value
[31:0]	CFBADD[31:0]		start address.	0

43.4.23 LTDC Layer x Color Frame Buffer Length Register (R32_LTDC_LxCFBLR) (x=1/2)

Offset address: 0x58 + 0x30*(x-1)

31	30	29	28	27	26	25	24	23	twenty two	twenty one	20	19	18	17	16
Reserved									CFBP[12:0]						

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

Bit	name	access	describe	Reset value
[31:29]	Reserved	RO	reserved.	0
[28:16]	CFBP[12:0]	RW	Color frame buffer spacing (in bytes): This field is defined from the beginning of a row of pixels to the beginning of the next row. The increment.	0
[15:13]	Reserved	RO	reserved.	0
[12:0]	CFBLL[12:0]	RW	Color frame buffer line length: Define the length of a row of pixels (in bytes) + 31. The line length is calculated as: effective width x bytes per pixel. + 31.	0

43.4.24 LTDC Layer x Color Frame Buffer Line Number Register (R32_LTDC_LxCFBLNR) (x=1/2)

Offset address: 0x5C + 0x30*(x-1)

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

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

Bit	name	access	describe	Reset value
[31:12]	Reserved	RO	reserved.	0
[10:0]	CFBLNBR[10:0]	RW	Number of frames in the buffer: This field defines the number of rows in the buffer, which corresponds to the effective height.	0

Note: The line count and line length settings define the amount of data retrieved from each frame for each layer. If configured below the required number of bytes, it will result in... FIFO

An underflow interrupt (if enabled). On the other hand, the start address and spacing settings define the correct starting position for each row in memory.

43.4.25 LTDC Level x CLUT Write Register (R32_LTDC_LxCLUTWR) (x=1/2) Offset Address: 0x60 + 0x30*(x-1)

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
CLUTADD[7:0]							RED[7:0]							

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

GREEN[7:0]	BLUE[7:0]

Bit	name	access	describe	Reset value
[31:24]	CLUTADD[7:0]	WO	configures the CLUT address for each RGB value.	0
[23:16]	RED[7:0]	WO	(Red Value)	0
[15:8]	GREEN[7:0]	WO	Green value.	0
[7:0]	BLUE[7:0]	WO	(Blue Value)	0

Note: (1) The CLUT write register can be enabled and disabled via the R32_LTDC_LxCR register. (2) CLUT

It can be enabled and disabled via the R32_LTDC_LxCR register.

(3) CLUT and L8, AL44 AL88 It only makes sense for pixel format.

Chapter 44 Graphics Processing Hardware Accelerator (GPHA)

The Image Processing Hardware Accelerator (GPHA) is a DMA specifically designed for image processing, supporting either indexed color mode or direct color mode. It supports all classic color coding schemes. In addition, the GPHA module also comes with its own proprietary CLUT (Color Lookup Table).

44.1 Main Features

Supports single-HB master bus architecture

The HB slave programming interface supports 8/16/32-bit access (except for 32-bit CLUT access).

Supports programmable source and target region sizes and offsets .

Supports programmable source and destination addresses for the entire memory space .

Supports adjustable alpha values (source value, fixed value, modulation value) .

Supports hybrid operation from up to two sources .

Supports programmable source and destination color formats, supporting up to 11 color formats with up to 32 bits per pixel.

In indirect color mode, it supports two internal memories for storing CLUTs . It also supports programmable CLUT sizes, allowing CLUTs to be automatically loaded by the CPU or programmed. Furthermore, it supports internal timers to control the bandwidth of the color bus (HB).

Supports 3 operating modes: register-to-memory, memory-to-memory with pixel format conversion, and memory-to-memory with...

Supports pixel format conversion and mixing

Supports filling a specified portion or all of a target image with a specific color.

Supports copying part or all of the content of the source image to the corresponding part or all of the target image.

Supports pixel-format conversion and copying of part or all of the source image content to part or all of the target image content.

Supports mixing portions and/or all of two source images with different pixel formats, and then copying the result to portions with different color formats. In the entire target image

Supports pausing and suspending GPHA operations .

Supports generating interrupts when bus errors or access conflicts occur .

Supports generating interrupts when the process is completed .

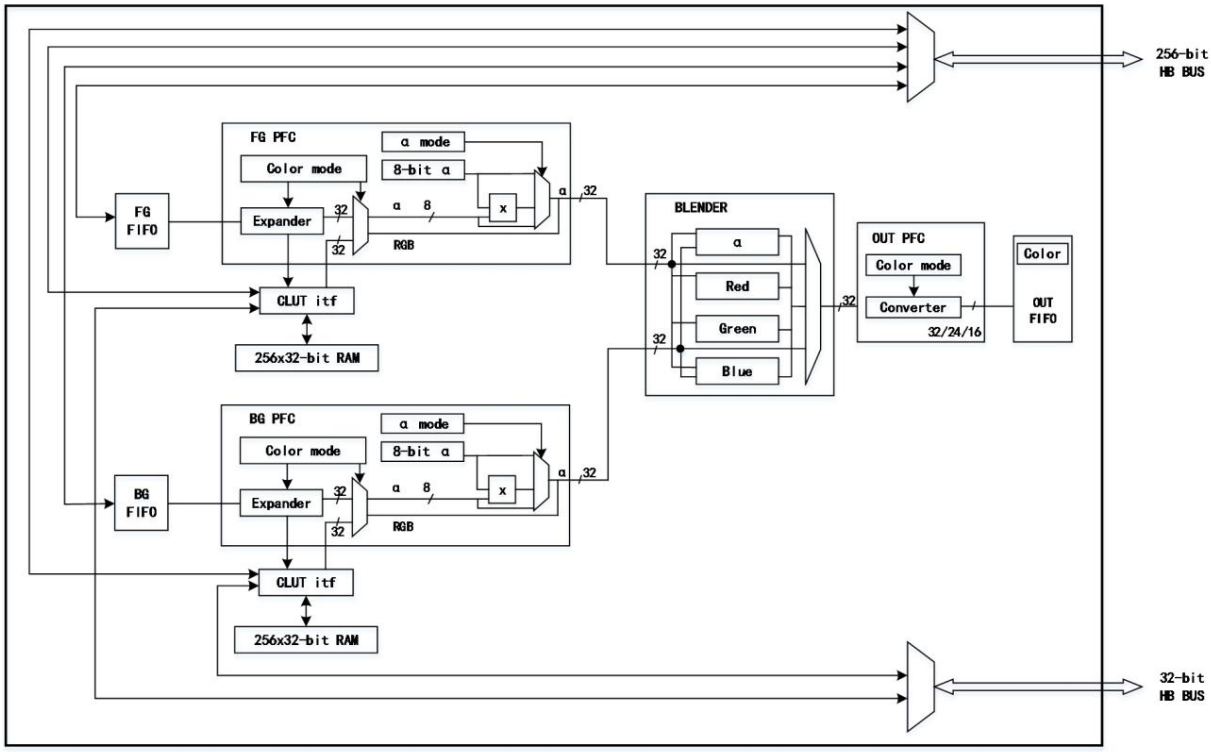
44.2 Functional Description

44.2.1 Overview GPHA

uses Foreground (FG) FIFO and Background (BG) FIFO to acquire the input data to be copied and processed. These FIFOs are based on... The corresponding pixel format converter (PFC) acquires pixels in a preset color format and performs color format conversion, where the foreground layer and background layer are involved. The data can be processed individually or in combination, and the final output is the converted color data. Figure 44-1 illustrates this in detail.

The structural block diagram.

Figure 44-1 GPHA Structure Diagram



The GPHA controller performs direct memory transfers and, as an HB master, can control the HB bus matrix to initiate HB transactions. The HB slave port is used to program the GPHA controller.

GPHA supports operation in the following three

modes: - Register to memory;

- Memory to memory with pixel format conversion; - Memory

to memory with pixel format conversion and mixing.

GPHA features include the following

operations: - Support for filling a specified portion or all of a target image with a specific

color; - Support for copying a portion or all of the content of a source image to the corresponding portion or

all of the target image; - Support for pixel format conversion and copying of a portion or all of the content of a source image to a

portion or all of the content of a target image; - Support for mixing portions and/or all of two source images with different pixel formats, and then copying the result to a portion or the entire target image with different color formats.

44.2.2 GPHA Control The GPHA

controller can be configured through the R32_GPHA_CTLR register to perform the following operations: 1)

Select the GPHA operating mode; 2)

Enable/disable GPHA interrupts; 3)

Start/suspend/abort ongoing data transmission.

44.2.3 GPHA Foreground FIFO and Background FIFO

The coordinated operation of the GPHA foreground (FG) FIFO and background (BG) FIFO is crucial for GPHA's efficient image processing. They acquire the input data to be copied and processed, and these FIFOs retrieve pixels according to the preset color format in the corresponding pixel format converter (PFC). They can be programmed via the following

register: - GPHA Foreground Memory Address Register (R32_GPHA_FGMAR)

- GPHA Foreground Offset Register (R32_GPHA_FGOR)
- GPHA Background Layer Memory Address Register (R32_GPHA_BGMAR)
- GPHA Background Layer Offset Register (R32_GPHA_BGBOR)
- GPHA Row Count Register (Number of Rows and Pixels per Row) (R32_GPHA_NLR)

GPHA operates in register-to-memory mode and does not activate any FIFO. GPHA operates in memory-to-memory mode and supports... BGFIFO will not be activated during pixel format conversion (without blending operations).

44.2.4 GPHA Foreground and Background Layer Pixel Format Converter (PFC)

The GPHA Foreground and Background Pixel Format Converter (PFC) is responsible for performing pixel format conversion to generate 32-bit values per pixel. In addition, PFC also has the ability to modify the alpha channel.

The converter converts the color format in the first stage. This is achieved by adjusting the CM[3:0] bits of the R32_GPHA_FGPFCCR register. Bits CM[3:0] of the R32_GPHA_BGPFCCR register configure the original color format of the foreground and background layer pixels, respectively. The color modes supported for input are shown in Table 44-1.

Table 44-1 Supported Color Modes for Input

CM[3:0]	Color mode
0000	ARGB8888
0001	RGB888
0010	RGB565
0011	ARGB1555
0100	ARGB4444
0101	L8
0110	AL44
0111	AL88
1000	L4
1001	A8
1010	A4

The alpha channel is an additional channel in an image used to represent the transparency of each pixel. It is often used in conjunction with red, green, and blue (RGB) channels. The channels are used together to form an RGBA model.

The following is an explanation of the encoding methods for the color formats appearing in Table 44-1:

Alpha value field: transparent, 0xFF corresponds to opaque pixels, 0x00 corresponds to transparent pixels;

The R field represents red.

G field: Represents green;

Field B: Represents blue;

The L field represents luminance. This field is an index of the CLUT used to retrieve three or four RGB/ARGB components.

If the original format is Direct Color mode, it is expanded to 8 bits per channel by copying the MSB to the LSB. This ensures that the conversion has... Good linearity.

If the original format does not include an alpha channel, the alpha value will be automatically set to 0xFF (opaque).

If the original format is indirect color mode, then CLUT is required, and each pixel format converter is paired with a 256-bit 32-bit bar.

The purpose is associated with CLUT.

For specific alpha modes A4 and A8, neither color information is stored nor indexed. The colors used to generate the image are fixed. The color of the foreground layer pixels is defined in the R32_GPHA_FGCOLR register and in the R32_GPHA_BGCOLR register.

The color of the background layer pixels.

The order of fields in the system memory is shown in Table 44-2.

Table 44-2 Data order in memory

Color mode	@+3	@+2	@+1	@+0
ARGB8888	A0[7:0]	R0[7:0]	G0[7:0]	B0[7:0]
RGB888	B1[7:0]	R0[7:0]	G0[7:0]	B0[7:0]
	G2[7:0]	B2[7:0]	R1[7:0]	G1[7:0]
	R3[7:0]	G3[7:0]	B3[7:0]	R2[7:0]
RGB565	R1[4:0]G1[5:3]	G1[2:0]B1[4:0]	R0[4:0]G0[5:3]	G0[2:0]B0[4:0]
ARGB1555	A1[0]R1[4:0]G1[4:3]	G1[2:0]B1[4:0]	A0[0]R0[4:0]G0[4:3]	G0[2:0]B0[4:0]
ARGB4444	A1[3:0]R1[3:0]	G1[3:0]B1[3:0]	A0[3:0]R0[3:0]	G0[3:0]B0[3:0]
L8	L3[7:0]	L2[7:0]	L1[7:0]	L0[7:0]
AL44	A3[3:0]L3[3:0]	A2[3:0]L2[3:0]	A1[3:0]L1[3:0]	A0[3:0]L0[3:0]
AL88	A1[7:0]	L1[7:0]	A0[7:0]	L0[7:0]
L4	L7[3:0]L6[3:0]	L5[3:0]L4[3:0]	L3[3:0]L2[3:0]	L1[3:0]L0[3:0]
A8	A3[7:0]	A2[7:0]	A1[7:0]	A0[7:0]
A4	A7[3:0]A6[3:0]	A5[3:0]A4[3:0]	A3[3:0]A2[3:0]	A1[3:0]A0[3:0]

It adopts ARGB8888 mode and supports 32-bit aligned 24-bit RGB888. , After generating a 32-bit value

Bits AM[1:0] of the R32_GPHA_FGPFCCR/R32_GPHA_BGPFCCR registers adjust the alpha channel. See the table for detailed configuration.

As shown in 44-3.

Table 44-3 Alpha Mode Configuration

AM[1:0]	Alpha mode
00	No adjustments will be made.
01	Replace with R32_GPHA_FGPFCCR/R32_GPHA_BGPFCCR registers ALPHA[7:0] value
10	Replace with the original alpha value and R32_GPHA_FGPFCCR/R32_GPHA_BGPFCCR registers ALPHA[7:0] The product of the quotients obtained by dividing the value by 255.
11	reserve

Note: To enable support for the alternative format, the R32_GPHA_FGPFCCR/R32_GPHA_BGPFCCR register bits must be set to 1.

inverts the value The be input. This operation also applies to the R32_GPHA_FGPFCCR/R32_GPHA_BGPFCCR registers.

value stored in alpha. Additionally, the R32_GPHA_FGPFCCR/R32_GPHA_BGPFCCR register bits can be used.

1. The order in which the R32_GPHA_FGCOLR/ R B CLUT

R32_GPHA_BGCOLR registers are used. RGB

44.2.5 GPHA Foreground Layer FIFO and Background Layer CLUT

Interface A CLUT is a table used to store color maps, mapping the index value of each pixel to a specific color value. The CLUT interface can...

Manages access to the CLUT memory and the automatic loading of the CLUT. The CLUT interface supports the following access methods:

- PFC reads the CLUT during pixel format conversion;
- The CPU accesses the CLUT from the device port via HB to read or write data;
- When performing automatic CLUT loading, CLUT is written via the HB master device port.

CLUT memory loading can be achieved through automatic loading and manual loading:

Auto-loading

1) Program the CLUT address to the R32_GPHA_FGCMAR register (foreground CLUT) or the R32_GPHA_BGCMAR register (background CLUT).
(CLUT).

2) Program the CLUT size to the R32_GPHA_FGPFCCR register (foreground CLUT) or the R32_GPHA_BGPFCCR register.
CS[7:0] bits of (background layer CLUT).

3) Set the R32_GPHA_FGPFCCR register (foreground CLUT) or the R32_GPHA_BGPFCCR register (background CLUT) to...
The START bit is set to 1 to initiate the transfer. During autoload, the CPU must not access the CLUT. If a conflict occurs, and R32_GPHA_CTLR...
If the CAEIE register bit is set to 1, a CLUT access error interrupt will occur.

Manually

loading the application requires manual programming via GPHA HB from the device port to map the local CLUT memory to the CLUT.

1) Program the CLUT pixel address to the FG_CLUT_INDEX bit of the R32_GPHA_FGCRWS register (foreground layer CLUT) or
The BG_CLUT_INDEX bit of the R32_GPHA_FGCRWS register (background layer CLUT).

2) Program the CLUT pixel values to the R32_GPHA_FGCDAT register (foreground layer CLUT) or the R32_GPHA_BGCDAT register.
(Background layer CLUT), the CPU automatically loads the pixel value into the corresponding CLUT memory.

The CLUT format supports 24-bit or 32-bit. It is accessed via the R32_GPHA_FGPFCCR register (foreground CLUT) or R32_GPHA_BGPFCCR.
The CCM bit of the register (background layer CLUT) can be configured with the CLUT color mode. For details on the configuration method, please refer to Table 44-4.

Table 44-4 Supported CLUT Color Modes

CCM	CLUT color mode
0	32-bit ARGB8888 24-
1	bit RGB888

Table 44-5 shows how CLUT data is organized in the system memory.

Table 44-5 CLUT Data Order in Memory

CLUT color mode	@+3	@+2	@+1	@+0
ARGB8888	A0[7:0]	R0[7:0]	G0[7:0]	B0[7:0]
RGB888	B1[7:0]	R0[7:0]	G0[7:0]	B0[7:0]
	G2[7:0]	B2[7:0]	R1[7:0]	G1[7:0]
	R3[7:0]	G3[7:0]	B3[7:0]	R2[7:0]

44.2.6 GPHA Mixer The GPHA

mixer mixes source pixels in pairs to calculate the result pixel.

The mixing will be performed according to the following formula:

$$CZ[\ = \frac{0]\hat{}\_]\hat{}U0`\hat{}\_`\hat{}>0`\hat{}\_abcd}{\_Sef}$$ (The quotient will be rounded down);

Where C = R, G, or B,

$$g{:}hij = \frac{g3kglk}{255}$$

$$gZ[\ = g3k + glk\ m\ g{:}hij$$

The mixer does not require any configuration registers. Whether the mixer is used depends on the MODE[1:0] bits of the R32_GPHA_CTLR register.
Defined GPHA operating mode.

44.2.7 GPHA Output PFC

The output PFC converts the pixel format from 32-bit to a specified output format, which can be determined via the R32_GPHA_OPFCCR register.
Configure the CM[2:0] bits.

Table 44-6 Supported Color Modes for Output

CM[2:0]	Color mode
000	ARGB8888
001	RGB888
010	RGB565
011	ARGB1555
100	ARGB4444

Note: To enable support for the alternative format, the R32_GPHA_OPFCCR register bit needs to be set to 1 to enable the calculation of the upcoming. The A alpha value is inverted. This operation also applies to values used in the R32_GPHA_OPFCCR register. Additionally, it can be done by... Setting

The order in which the R32_GPHA_OPFCCR, R32_GPHA_FGPFCCR, and R32_GPHA_BGPFCCR registers are used. The operation also applies to...

RGB

44.2.8 GPHA Output FIFO

The output FIFO programs the pixels according to the color format defined in the output PFC. The target region can be defined through the following registers:

- GPHA Output Memory Address Register (R32_GPHA_OMAR)

- GPHA Output Offset Register (R32_GPHA_OOR)

- GPHA Row Count Register (Number of Rows and Pixels per Row) (R32_GPHA_NLR)

If GPHA is operating in register-to-memory mode, the configured output rectangle will be set to the GPHA output color register.

The color fill is specified in (R32_GPHA_OCOLR), which contains a fixed 32-bit, 24-bit, or 16-bit value. Color format

The selection can be configured via the CM[2:0] bits of the R32_GPHA_OPFCCR register.

The order of fields in the system memory is shown in Table 44-7.

Table 44-7 Data order in memory

Color mode	@+3	@+2	@+1	@+0
ARGB8888	A0[7:0]	R0[7:0]	G0[7:0]	B0[7:0]
RGB888	B1[7:0]	R0[7:0]	G0[7:0]	B0[7:0]
	G2[7:0]	B2[7:0]	R1[7:0]	G1[7:0]
	R3[7:0]	G3[7:0]	B3[7:0]	R2[7:0]
RGB565	R1[4:0]G1[5:3]	G1[2:0]B1[4:0]	R0[4:0]G0[5:3]	G0[2:0]B0[4:0]
ARGB1555	A1[0]R1[4:0]G1[4:3]	G1[2:0]B1[4:0]	A0[0]R0[4:0]G0[4:3]	G0[2:0]B0[4:0]
ARGB4444	A1[3:0]R1[3:0]	G1[3:0]B1[3:0]	A0[3:0]R0[3:0]	G0[3:0]B0[3:0]

Note: Bit-aligned 32-bit RGB888 can be supported in ARGB8888 mode.

44.2.9 GPHA HB Master Port Timer

The HB master port embeds an 8-bit timer designed to selectively limit the bandwidth of the crossbar switch matrix. This timer is controlled by the HB timer.

The clock driver is responsible for counting the dead time between two consecutive accesses, thereby limiting bandwidth usage.

The timer's enable and dead-time values can be configured via the EN bit and DT[7:0] bits of the R32_GPHA_AMTCR register. HB dead-time

For information on setting up zones, please refer to section 44.3.16.

44.2.10 GPHA Transactions

A GPHA transaction consists of a series of preset data transmission sequences. The amount and width of the data can be programmed in software.

Each GPHA data transfer project requires a maximum of 4 steps:

1) Load data from the memory location addressed by the R32_GPHA_FGMAR register, and process it according to the definition in R32_GPHA_FGPFCCR.

Row pixel format conversion.

2) Load data from the memory location addressed by the R32_GPHA_BGMAR register, and process it according to the definition in R32_GPHA_BGPFCCR.

3) Process the alpha value

using PFC operations to obtain alpha channel information and blend all retrieved pixels. 4) Perform pixel format conversion on the synthesized

pixels according to the R32_GPHA_OPFCCR register, and then program the converted data into the channel.

The memory location addressed by the R32_GPHA_OMAR register.

44.2.11 GPHA configuration allows

source and destination data transfer to be addressed across the entire 4GB region (address range between 0x00000000 and 0xFFFFFFFF).

Set and memory.

GPHA can operate in the following three modes, selected via the MODE[1:0] bits of the R32_GPHA_CTLR register: - Register to memory -

Memory to memory with PFC

- Memory to memory with PFC and hybrid

44.2.11.1 Register-to-Memory Register-to-

memory mode is used to fill user-defined areas with predefined colors. The color format is set in

the R32_GPHA_OPFCCR register.

GPHA does not obtain data from any source. It only writes the colors defined in the R32_GPHA_OCOLR register to the R32_GPHA_OMAR register.

Addressing and the regions defined by R32_GPHA_NLR and R32_GPHA_OOR.

44.2.11.2 Memory-to-Memory and PFC Execution

In this mode, GPHA performs pixel format conversion on the source data and stores the result in the destination memory

location. The source region size is defined by the R32_GPHA_NLR and R32_GPHA_FGOR registers, while the destination region size is defined by the

R32_GPHA_NLR and R32_GPHA_OOR registers.

Data is retrieved from the location defined in the R32_GPHA_FGMAR register and processed by the foreground layer PFC. The raw pixel format is obtained through...

GPHA_FGPFCCR register configuration.

If the original pixel format is a direct color mode, all color channels are extended to 8 bits. If the pixel format

is an indirect color mode, the relevant CLUTs must be loaded into the CLUT memory.

CLUT loading can be completed automatically in the

following order: 1) Set the CLUT address in the R32_GPHA_FGCMAR register.

2) Set the CLUT size in the CS[7:0] bits of the R32_GPHA_FGPFCCR register. 3) Set the CLUT

format (24 or 32 bits) in the CCM bits of the R32_GPHA_FGPFCCR register. 4) Start CLUT loading by setting the

START bit of the R32_GPHA_FGPFCCR register to 1.

When CLUT loading is complete, the R32_GPHA_ISR register will set the CTCIF flag; if the R32_GPHA_CTLR register contains...

CTCIE position 1 will also cause an interrupt. The CLUT autoload process cannot be performed simultaneously with traditional GPHA transport.

CLUT can also be filled by the CPU or by any other master device via the HB port. CLUT is inaccessible during GPHA transports and when using CLUT

(Indirect Color Format).

44.2.11.3 Memory-to-memory and performing PFC and hybrid

In this mode, the foreground FIFO and background FIFO (located in the R32_GPHA_FGMAR register and the R32_GPHA_BGMAR register respectively) will be used.

(As defined in the register) Get two source images.

Two pixel format converters must be configured as described in the memory-to-memory mode. Because these two pixel format converters are each independent...

It is independent and has its own CLUT memory, so its configuration can be different.

After each pixel is converted to 32 bits by the corresponding PFC, it will be mixed according to the formula in 44.3.6. The output PFC will encode the

obtained 32-bit pixel values according to the specified output format, and the encoded data will be written to the target memory unit addressed by GPHA_OMAR.

44.2.11.4 Configuration Error Detection

GPHA checks the configuration for correctness before each transfer. When starting a new transfer/autoload, if a configuration error is detected, The hardware will set the configuration error interrupt flag. An interrupt will also be generated if the CEIE bit in the R32_GPHA_CTLR register is set to 1. The detectable configuration errors are as follows:

- Memory transfer (excluding register-to-memory mode): The MA bit of the R32_GPHA_FGMAR register does not match the CM bit of the R32_GPHA_FGPFCCR register; - Memory transfer

(excluding register-to-memory mode): The CM bit in the R32_GPHA_FGPFCCR register is invalid; - Memory transfer (excluding register-to-

memory mode): The PL bit of the R32_GPHA_NLR register is odd, while the CM bit of the R32_GPHA_FGPFCCR register is A4 or L4; - Memory transfer (excluding register-to-memory mode):

The LO bit of the R32_GPHA_FGOR register is odd, while... The CM bit of the R32_GPHA_FGPFCCR register is either A4 or L4;

- Memory transfer (mixed mode only): The MA bit of the R32_GPHA_BGMAR register is inconsistent with the CM bit of the R32_GPHA_BGPFCCR register; - Memory transfer:

The CM bit of the R32_GPHA_BGPFCCR register is invalid (mixed mode only); - Memory transfer (mixed mode only):

The PL bit of the R32_GPHA_NLR register is odd, while the R32_GPHA_BGPFCCR register... - Memory transfer (mixed

mode only): The LO bit of the R32_GPHA_BGOR register is odd, while the CM bit of the R32_GPHA_BGPFCCR register is A4 or L4; - Memory transfer: The MA bit of the R32_GPHA_OMAR

register does not match the CM bit of the R32_GPHA_OPFCCR register; - Memory transfer: The CM bit of the R32_GPHA_OPFCCR register

is invalid; - Memory transfer: The NL bit in the R32_GPHA_NLR register is 0; - Memory

transfer: The PL bit in the R32_GPHA_NLR register is 0;

44.2.12 GPHA Transfer Control (Start, Suspend, Abort, and Complete) After configuring GPHA, a

transfer can be started by setting the START bit in the R32_GPHA_CTLR register to 1. Upon completion of the transfer, the START bit is automatically reset, and the TCIF flag in the R32_GPHA_ISR register is set. An interrupt will also be generated if the TCIE bit in the R32_GPHA_CTLR register is set to 1.

User applications can suspend GPHA at any time by setting the SUSP bit of the R32_GPHA_CTLR register to 1. The transaction can then be aborted by setting the ABORT bit of the GPHA_CTLR register to 1, or restarted by resetting the SUSP bit of the R32_GPHA_CTLR register. User applications can also abort an ongoing

transaction at any time by setting the ABORT bit of the R32_GPHA_CTLR register to 1. In this case, the TCIF flag is not set. The CLUT autotransfer process can also be aborted or suspended via the START bit of

the R32_GPHA_FGPFCCR register and the R32_GPHA_BGPFCCR register itself.

44.2.13 The watermark

can be programmed to generate an interrupt when the last pixel of a specified row is written to the target storage area. The

row number is defined in bits LW[15:0] of the R32_GPHA_LWR register. The

TWIF flag of R32_GPHA_ISR is set when the last pixel of the row has been transferred; an interrupt is also generated if the TWIE bit of the R32_GPHA_CTLR register is set to 1.

44.2.14 Error management can

trigger errors: CLUT access conflicts (CPU attempts to access the CLUT during CLUT loading or GPHA transport execution), through The CAEIF flag is indicated by the R32_GPHA_ISR register.

This flag, along with the interrupt enable flag in the R32_GPHA_CTLR register (which controls the TEIE that generates an interrupt when needed), Related to CAEIE.

44.2.15 HB Dead Zone To

limit the use of HB bandwidth, a dead zone can be programmed between two consecutive HB accesses.

The dead-time function can be enabled by setting the EN bit of the R32_GPHA_AMTCR register to 1.

The dead-time value is stored in bits DT[7:0] of the R32_GPHA_AMTCR register. This value indicates the dead-time interval allowed between two consecutive transactions on the HB bus. The minimum number of cycles required.

Updates to the dead zone value made during GPHA operation will take effect during the next HB transfer.

44.2.16 Interruption

Table 44-8 Interrupt Requests

Interrupt event	Event marker	Enable control bit
configuration error	CEIF	CEIE
CLUT transfer complete	CTCIF	CTCIE
CLUT access error	CAEIF	CAEIE
transmission	TWF	TWIE
watermark transmission complete	TCIF	TCIE

44.3 Register Description

Table 44-9 List of GPHA-related registers

name	Access address	describe	Reset value
R32_GPHA_CTLR	0x40016800 GPHA Control Register;		0x00000000
R32_GPHA_ISR	0x40016804 GPHA Interrupt Status Register;		0x00000000
R32_GPHA_IFCR	0x40016808 GPHA Interrupt Flag Clear Register; 0x4001680C		0x00000000
R32_GPHA_FGMAR	GPHA Foreground Layer Memory Address Register; 0x00000000; 0x40016810 GPHA Foreground		
R32_GPHA_FGOR	Layer Offset Register; 0x40016814 GPHA Background		0x00000000
R32_GPHA_BGMAR	Layer Memory Address Register; 0x00000000; 0x40016818 GPHA Background Layer Offset		
R32_GPHA_BGOR	Register; 0x4001681C GPHA Foreground Layer PFC		0x00000000
R32_GPHA_FGPFCCR	Control Register; 0x40016820 GPHA Foreground Layer Color		0x00000000
R32_GPHA_FGCOLR	Register; 0x40016824 GPHA Background Layer PFC		0x00000000
R32_GPHA_BGPFCCR	Control Register; 0x40016828 GPHA Background Layer Color		0x00000000
R32_GPHA_BGCOLR	Register; 0x4001682C GPHA Foreground CLUT Memory		0x00000000
R32_GPHA_FGCMAR	Address Register 0x00000000 0x40016830 GPHA Background CLUT Memory Address Register		
R32_GPHA_BGCMAR	0x00000000 0x40016834 GPHA Output PFC Control Register 0x40016838 GPHA Output Color		
R32_GPHA_OPFCCR	Register 0x4001683C GPHA Output Memory Address		0x00000000
R32_GPHA_OCOLR	Register 0x40016840 GPHA Output Offset Register		0x00000000
R32_GPHA_OMAR	0x40016844 GPHA Line Count Register 0x40016848 GPHA Line		0x00000000
R32_GPHA_OOR	Watermark Register 0x4001684C GPHA HB Master		0x00000000
R32_GPHA_NLR	Device Timer Configuration Register		0x00000000
R32_GPHA_LWR	0x00000000 0x40016850 GPHA Foreground CLUT		0x00000000
R32_GPHA_AMTCR	Read/Write Settings Register 0x00000000		
R32_GPHA_FGCWRS			

R32_GPHA_FGCDAT	0x40016854 GPHA Foreground Layer CLUT Read/Write Data Register 0x00000000 0x40016858
R32_GPHA_BGCWRS	GPHA Background Layer CLUT Read/Write Settings Register 0x00000000 0x4001685C GPHA
R32_GPHA_BGCDAT	Background Layer CLUT Read/Write Data Register 0x00000000

44.3.1 GPHA Control Register (R32_GPHA_CTLR) Offset Address:

0x00

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
Reserved				MODE[2:0]	

15	14	13	12	11	10	9	8 7 6 5 4 3	2	1	0
Reserved	CEIE	CTCIE	CAEIE	TWIE	TCIE		Reserved	ABORT	SUSP	START

Bit	name	access	describe	Reset value
[31:19] Reserved		RO is reserved.		0
[18:16] MODE[2:0]		RW	GPHA operating mode: 001: Memory-to-memory and perform PFC (FG PFC activation only) (FG acquisition during live time); 010: Memory-to-memory and perform mixing (perform PFC and mixing) (Acquire FG and BG at the appropriate time). 011: Register to Memory (No FG and BG, only output stage) activation); Other: Reserved.	0
[15:14] Reserved		RO reserved.		0
13	CEIE	RW	Configuration error interrupt enable: 1: CE interrupt enabled; 0: CE interrupt disabled.	0
12	CTCIE	RW	CLUT Transmission Completion Interrupt Enable: 1: CTC interrupt enabled; 0: CTC interrupt disabled.	0
11	CAEIE	RW	CLUT access error interrupt enable: 1: CAE interrupt enable; 0: CAE interrupts are disabled.	0
10	TWIE	RW	Watermark transmission interruption enabled: 1: TW interrupt enable; 0: TW interrupt disabled.	0
9	TCIE	RW	Transmission completion interrupt enable: 1: TC interrupt enabled; 0: TC interrupt disabled.	0
[8:3] Reserved		RO reserved.		0
2	ABORT	RW	Abort: 1: Request to abort transmission; 0: Do not request transmission abort.	0
1	SUSP	RW	Hang up: 1: Transmission suspended;	0

			0: Transmission is not suspended.	
0	START	RW	start up: This bit can be used to start based on parameters loaded in various configuration registers. GPHA is activated. This bit will be automatically reset in the following situations: – Transmission ends; – The GPHA_CTLR register ABORT is accessed via the user application. Position 1: When data transmission is aborted; -When data transmission fails; – Data transfer fails to start due to configuration errors or because other transfer operations are already in progress (CLUT autoload).	0

44.3.2 GPHA Interrupt Status Register (R32_GPHA_ISR) Offset

Address: 0x04

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										CEIF	CTCIF	CAEIF	TWIF	TCIF	Reserved

Bit	name	access	describe	Reset value
[31:6] Reserved		RO	reserved.	0
5	CEIF	RO	Configuration error interrupt flag: R32_GPHA_CTLR R32_GPHA_FGPFCCR or The START bit of the R32_GPHA_BGPFCCR register is set to 1, and the encoding... This position 1 is when the configuration is incorrect.	0
4	CTCIF	RO	CLUT Transmission Complete Interruption Flag: Complete the copying of CLUT from the system storage to the GPHA internal storage. When using the device, this position is 1.	0
3	CAEIF	RO	CLUT access error interruption flag: Automatically copy the CLUT from system memory to GPHA internal storage. During the process, if the CPU accesses the CLUT, this bit will be set to 1.	0
2	TWIF	RO	Watermark transmission interruption flag: This position is set to 1 when the last pixel of the watermarked row is transmitted.	0
1	TCIF	RO	Transmission completion interruption flag: This position is 1 when a GPHA transport operation is completed (data transfer only).	0
0	Reserved	RO	is reserved.	0

44.3.3 Clearing the GPHA interrupt flag register (R32_GPHA_IFCR)

Offset address: 0x08

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

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

Reserved	CCEIF	CCTCIF	CAECIF	CTWIF	CTCIF	Reserved

Bit	name	access	describe	Reset value
[31:6]	Reserved	RO	is reserved.	0
5	CCEIF	RW1Z	clears the configuration error interrupt flag; writing 1 clears 0, writing 0 has no effect. 0	
4	CCTCIF	RW1Z	clears the CLUT transfer completion interrupt flag; writing 1 clears 0, writing 0 has no effect. effect.	0
3	CAECIF	RW1Z	Clear the CLUT access error interrupt flag; writing 1 clears 0, writing 0 results in nothing. effect.	0
2	CTWIF	RW1Z	Clears the transmission watermark interruption flag; writing 1 clears 0, writing 0 is	
1	CTCIF		invalid. 0 RW1Z Clears the transmission completion interruption flag; writing 1 clears 0,	
0	Reserved		writing 0 is invalid. 0 RO Reserved.	0

44.3.4 GPHA Foreground Layer Memory Address Register (R32_GPHA_FGMAR) Offset

Address: 0x0C

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

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

Bit	name	access		Reset value
[31:0]	MA[31:0]	RW	Description of the storage address of the foreground layer image: This register allows write operations only when data transfer is disabled. Once data transfer begins, the register will switch to read-only mode. Address alignment must match the image format, for example, using per pixel. In 32-bit format, address alignment must be 32 bits; per pixel In 16-bit format, address alignment must be 16 bits; while using per-image... When using a 4-bit format, the address alignment must be 8 bits.	0

44.3.5 GPHA Foreground Offset Register (R32_GPHA_FGOR)

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		LO[13:0]													

Bit	name	access	describe	Reset value
[31:14]	Reserved	RO	preserved.	0
[13:0]	LO[13:0]	RW	Row offset of the foreground layer image: This value is used to generate the address. Line offsets will be added to the end of each line, using...	0

			This determines the starting address of the next line. This register allows write operations only when data transfer is disabled. Once data transfer is initiated, the domain will become read-only. If the image format is 4 bits per pixel, then the line offset value must be . even.	

44.3.6 GPHA Background Layer Memory Address Register (R32_GPHA_BGMAR)

Offset Address: 0x14

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
MA[31:16]														

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

Bit	name	access	Description	Reset value
[31:19]	MA[31:0]	RW	of the storage address of the background layer image: This register allows write operations only when data transfer is disabled. Once the data transfer process is activated, the register will become read-only. state. Address alignment must match the image format, for example, using per pixel. In 32-bit format, address alignment must be 32 bits; per pixel In 16-bit format, address alignment must be 16 bits; while using per-image... When using a 4-bit format, the address alignment must be 8 bits.	0

44.3.7 GPHA Background Layer Offset Register (R32_GPHA_BGOR)

Offset address: 0x18

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

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

Bit	name	access	describe	Reset value
[31:14]	Reserved	RO	preserved.	0
[13:0]	LO[13:0]	RW	Background layer image row offset: This value is used to generate the address. Line offsets will be added to the end of each line, using... This determines the starting address of the next line. This register allows write operations only when data transfer is disabled. Once data transfer is initiated, the domain will become read-only. If the image format is 4 bits per pixel, then the line offset value must be . even.	0

44.3.8 GPHA Foreground Layer PFC Control Register (R32_GPHA_FGPFCCR)

Offset Address: 0x1C

31 30 29 28 27 26 25 24 23 22 21	2019	18 17 16
ALPHA[7:0]	Reserved RBS AI	Reserved AM[1:0]

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

Bit	name	access	describe	Reset value
[31:24]	ALPHA[7:0]	RW	Alpha value: This field defines the alpha value, which can be used as the original alpha value. The substitution or multiplication depends on the choice via the AM[1:0] bits. The alpha value. Note: This field allows write operations only when data transfer is disabled. Once data transfer is initiated, this field becomes read-only.	0
[23:22]	Reserved	RO is retained. Red		0
twenty one	RBS	RW	and blue are swapped: 1: Switching mode (BGR or ABGR); 0: Normal mode (RGB or ARGB).	0
20	AI	RW	Invert Alpha: 1: Invert Alpha; 0: Regular Alpha. Note: Once the transfer is started, this bit will become read-only.	0
[19:18]	Reserved	RO is reserved.		0
[17:16]	AM[1:0]	RW	Alpha Mode: 00: Do not modify the alpha channel value of the foreground layer image; 01: Replace the alpha channel value of the original foreground layer image with... ALPHA[7:0]; 10: Replace the alpha channel value of the original foreground layer image with... The product of ALPHA[7:0] and the original alpha channel value; Other: Reserved. Note: This field is write-only enabled when data transfer is disabled. Once data transfer is initiated, this field becomes read-only.	0
[15:8]	CS[7:0]	RW	CLUT size: The number of CLUT entries used in the foreground layer image is equal to CS[7:0] + 1. Note: Once CLUT transfer is initiated, this field will become read-only.	0
[7:6]	Reserved	RO is reserved.		0
5	START	RW	Initiate the automatic loading process for CLUT: 1: Startup; 0: Off. This bit is automatically reset in the following situations: – Transmission ends; – The R32_GPHA_CTLR register is accessed via a user application. ABORT position 1 when transmission is aborted; - In case of transmission error; -Due to configuration errors or other data transfer operations already in progress (data)	0

			The transmission or automatic background layer CLUT transmission caused the transmission to not start.	
4	CCM	RW	CLUT color mode: 1: RGB888; 0: ARGB8888. Note: This field is write-only enabled when data transfer is disabled. Once transfer begins, this bit becomes read-only.	0
[3:0]	CM[3:0]	RW	Color mode: 0000: ARGB8888; 0001: RGB888; 0010: RGB565; 0011: ARGB1555; 0100: ARGB4444; 0101: L8; 0110: AL44; 0111: AL88; 1000: L4; 1001: A8; 1010: A4; Other: Reserved. Note: This field is write-only enabled when data transfer is disabled. Once transfer begins, this bit becomes read-only.	0

44.3.9 GPHA Foreground Color Register (R32_GPHA_FGCOLR)

Offset address: 0x20

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved											RED[7:0]			

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

Bit	name	access	describe	Reset value
[31:24]	Reserved	RO	is reserved.	0
[23:16]	RED[7:0]	RW	The red value of the foreground layer image in A4 or A8 mode.	0
[15:8]	GREEN[7:0]	RW	The green value of the foreground layer image in A4 or A8 mode.	0
[7:0]	BLUE[7:0]	RW	The blue value of the foreground layer image in A4 or A8 mode.	0

Note: The R32_GPHA_FGCOLR register allows write operations only when data transfer is disabled. Once data transfer is initiated, this bit becomes read-only.

44.3.10 GPHA Background Layer PFC Control Register (R32_GPHA_BGPFCCR)

Offset Address: 0x24

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
ALPHA[7:0]											Reserved RBS AI		Reserved AM[1:0]	

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

		RT		

Bit	name	access	describe	Reset value
[31:24]	ALPHA[7:0]	RW	Alpha value: This field defines the alpha value, which can be used as the original alpha value. The substitution or multiplication depends on the bit selection via AM[1:0]. The chosen alpha value. Note: This field allows write operations only when data transfer is disabled. Once data transfer is initiated, this field becomes read-only.	0
[23:22]	Reserved		RO is retained. Red	0
twenty one	RBS	RW	and blue are swapped: 1: Switching mode (BGR or ABGR); 0: Normal mode (RGB or ARGB). Note: Once the transfer is started, this bit will become read-only.	0
20	AI	RW	Invert Alpha: 1: Invert Alpha; 0: Regular Alpha. Note: Once the transfer is started, this bit will become read-only.	0
[19:18]	Reserved		RO is reserved.	0
[17:16]	AM[1:0]	RW	Alpha Mode: 00: Do not modify the alpha channel value of the background layer image; 01: Replace the alpha channel value of the original background layer image with... ALPHA[7:0]; 10: Replace the alpha channel value of the original background layer image with... The product of ALPHA[7:0] and the original alpha channel value; Other: Reserved. Note: This field is write-only enabled when data transfer is disabled. Once data transfer is initiated, this field becomes read-only.	0
[15:8]	CS[7:0]	RW	CLUT size: The number of CLUT entries used for the background layer image is equal to CS[7:0] + 1. Note: Once CLUT transfer is initiated, this field will become read-only.	0
[7:6]	Reserved		RO is reserved.	0
5	START	RW	Initiate the automatic loading process for CLUT: 1: Startup; 0: Off. This bit is automatically reset in the following situations: – Transmission ends; – The R32_GPHA_CTLR register is accessed via a user application. ABORT position 1 when transmission is aborted; - In case of transmission error; -Due to configuration errors or other data transfer operations already in progress (data) The transmission or automatic background layer CLUT transmission caused the transmission to not start.	0
4	CCM	RW	CLUT color mode: 1: RGB888;	0

			0: ARGB8888. Note: This field is write-only enabled when data transfer is disabled. Once transfer begins, this bit becomes read-only.	
[3:0] CM[3:0]		RW	Color mode: 0000: ARGB8888; 0001: RGB888; 0010: RGB565; 0011: ARGB1555; 0100: ARGB4444; 0101: L8; 0110: AL44; 0111: AL88; 1000: L4; 1001: A8; 1010: A4; Other: Reserved. Note: This field is write-only enabled when data transfer is disabled. Once transfer begins, this bit becomes read-only.	0

44.3.11 GPHA Background Layer Color Register (R32_GPHA_BGCOLR)

Offset Address: 0x28

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved											RED[7:0]			

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

Bit	name	access	describe	Reset value
[31:24]	Reserved	RO	is reserved.	0
[23:16]	RED[7:0]		The red value in A4 or A8 mode of the RW background layer image.	0
[15:8]	GREEN[7:0]		The green value of the RW background layer image in A4 or A8 mode.	0
[7:0]	BLUE[7:0]		The blue value of the RW background layer image in A4 or A8 mode.	0

Note: The R32_GPHA_BGCOLR register allows write operations only when data transfer is disabled. Once data transfer is initiated, this bit becomes read-only.

44.3.12 GPHA Foreground Layer CLUT Memory Address Register (R32_GPHA_FGCMAR)

Offset address: 0x2C

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
MA[31:16]														

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

Bit	name	access	describe	Reset value
[31:0]	MA[31:0]		The CLUT storage address of the RW foreground layer image:	0

			This register allows write operations only when data transfer is disabled. Once a CLUT transfer is initiated, the register will become read-only. If the foreground layer CLUT format is 32-bit, then the address must be 32-bit. Alignment.	

44.3.13 GPHA Background Layer CLUT Memory Address Register (R32_GPHA_BGCMAR)

Offset address: 0x30

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
MA[31:16]														

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

Bit	name	access	describe	Reset value
[31:19]	MA[31:0]	RW	CLUT storage address of background layer image: This register allows write operations only when data transfer is disabled. Once a CLUT transfer is initiated, the register will become read-only. If the foreground layer CLUT format is 32-bit, then the address must be 32-bit. Alignment.	0

44.3.14 GPHA Output PFC Control Register (R32_GPHA_OPFCCR) Offset

Address: 0x34

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved											RBS	AI	Reserved	

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

Bit	name	access	describe	Reset value
[31:22]	Reserved	RO is retained. Red		0
twenty one	RBS	RW	and blue are swapped: 1: Switching mode (BGR or ABGR); 0: Normal mode (RGB or ARGB). Note: Once the transfer is started, this bit will become read-only.	0
20	AI	RW	Invert Alpha: 1: Invert Alpha; 0: Regular Alpha. Note: Once the transfer is started, this bit will become read-only.	0
[19:3]	Reserved	RO reserved.		0
[2:0]	CM[2:0]	RW	Color mode: 000: ARGB8888; 001: RGB888; 010: RGB565; 011: ARGB1555;	0

			100: ARGB4444; Other: Reserved. Note: This register allows write operations only when data transfer is disabled. Once a transfer is initiated, the register will become read-only.	

44.3.15 GPHA Output Color Register (R32_GPHA_OCOLR)

Offset address: 0x38

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
ALPHA[7:0]											RED[7:0]			

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

Bit	name	access	describe	Reset value
[31:24]	ALPHA[7:0]	RW	outputs the alpha channel of the color.	0
[23:16]	RED[7:0]	RW	outputs the red value of the image.	0
[15:8]	GREEN[7:0]	RW	outputs the green value of the image.	0
[7:0]	BLUE[7:0]	RW	outputs the blue value of the image.	0

44.3.16 GPHA Output Memory Address Register (R32_GPHA_OMAR) Offset

Address: 0x3C

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
MA[31:16]														

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

Bit	name	access	Describe	Reset value
[31:19]	MA[31:0]	RW	the storage address of the output FIFO: This register allows write operations only when data transfer is disabled. Once the data transfer process is activated, the register will become read-only. state. Address alignment must match the image format, for example, using per pixel. In 32-bit format, address alignment must be 32 bits; per pixel In 16-bit format, address alignment must be 16 bits. Note: If the output pixel format is RGB888, then the output pixels will be... The number must be a multiple of 4.	0

44.3.17 GPHA Output Offset Register (R32_GPHA_OOR) Offset

Address: 0x40

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

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

Reserved	LO[13:0]

Bit	name	access	describe	Reset value
[31:14]	Reserved	RO is		0
[13:0]	LO[13:0]	RW	reserved. Output line offset: This value is used to generate the address. Line offsets will be added to the end of each line, using... This determines the starting address of the next line. This register allows write operations only when data transfer is disabled. Once data transfer is initiated, the domain will become read-only.	0

44.3.18 GPHA Row Number Register (R32_GPHA_NLR) Offset

Address: 0x44

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved											PL[13:0]			

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

Bit	name	access	describe	Reset value
[31:30]	Reserved	RO is reserved.		0
[29:16]	PL[13:0]	RW	Number of pixels per row in the area to be transmitted: This register allows write operations only when data transfer is disabled. Once data transfer is initiated, the domain will become read-only. If the image format is 4 bits per pixel, then the line offset value must be even. Note: When R32_GPHA_CTLR The MODE[2:0]=011b value of the PL[13:0] bits should be greater than 0.	0
[15:0]	NL[15:0]	RW	Number of rows in the area to be transmitted: This register allows write operations only when data transfer is disabled. Once data transfer is initiated, the domain will become read-only.	0

44.3.19 GPHA Row Watermark Register (R32_GPHA_LWR) Offset

Address: 0x48

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

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

Bit	name	access	describe	Reset value
[31:16]	Reserved	RO reserved.		0
[15:0]	LW[15:0]	RW	Line watermark: The watermarked row is generated when the last pixel of the row has been transmitted.	0

			Disconnect. Write operations are only allowed when data transfer is prohibited. Data transfer Once started, the field will become read-only.	

44.3.20 GPHA HB Master Timer Configuration Register (R32_GPHA_AMTCR) 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
DT[7:0]								Reserved							EN

Bit	name	access	describe	Reset value
[31:16]	Reserved	RO	Reserved.	0
[15:8]	DT[7:0]	RW	A dead link inserted between two consecutive accesses on the HB master port. The zone value is represented by HB clock cycles. These bits represent two consecutive... The minimum number of cycles allowed between HB visits.	0
[7:1]	Reserved	RO	is reserved.	0
0	EN	RW	Enable dead-time functionality: 1: Enable; 0: Prohibited.	0

44.3.21 GPHA Foreground Layer CLUT Read/Write Settings Register (R32_GPHA_FGCWRS)

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							FG_CLUT_EN	FG_CLUT_INDEX[7:0]							

Bit	name	access	describe	Reset value
[31:9]	Reserved	RO	is reserved.	0
8	FG_CLUT_EN	RW	CPU read/write lookup table enable: 1: Enable; 0: Off.	0
[7:0]	FG_CLUT_INDEX[7:0]	RW	sets the lookup table address.	0

44.3.22 GPHA Foreground Layer CLUT Read/Write Data Register (R32_GPHA_FGCDAT)

Offset address: 0x54

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

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

Bit	name	access	describe	Reset value
[31:0]	FG_CLUT_DATA[31:0]	RW	reads data or writes lookup table SRAM data.	0

44.3.23 GPHA Background Layer CLUT Read/Write Settings Register (R32_GPHA_BGCWRS)

Offset address: 0x58

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved														

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

Bit	name	access	describe	Reset value
[31:9]	Reserved	RO	is reserved.	0
8	BG_CLUT_EN	RW	CPU read/write lookup table enable: 1: Enable; 0: Off.	0
[7:0]	BG_CLUT_INDEX[7:0]	RW	sets the lookup table address.	0

44.3.24 GPHA Background Layer CLUT Read/Write Data Register (R32_GPHA_BGCDAT)

Offset address: 0x5C

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
BG_CLUT_DATA[31:16]														

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

Bit	name	access	describe	Reset value
[31:0]	BG_CLUT_DATA[31:0]	RW	reads data or writes lookup table SRAM data.	0

Chapter 45 Debugging Support (DBG)

45.1 Main Features

This register allows configuration of the MCU in debug mode. It includes:

ÿ Counters that support Independent Watchdog Timer (IWDG)

Supports window watchdog (WWDG) counters .

Counters **that support timers**

Supports timeout control for I2CSMBus

Supports CAN communication

45.2 Register Description

45.2.1 RISC-V Debug MCU Configuration Register (R32_DBGMCU_CR)

Address: 0x7C0 (CSR)

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
Reserved	CAN3_STOP	CAN2_STOP	CAN1_STOP	I2C4_SMBUS_TIMEOUT				I2C3_SMBUS_TIMEOUT	LPTIM2_STOP	LPTIM1_STOP	TIM12_STOP	TIM11_STOP	TIM10_STOP	TIM9_STOP	TIM8_STOP	TIM7_STOP	TIM6_STOP	TIM5_STOP

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIM4_STOP	TIM3_STOP	TIM2_STOP	TIM1_STOP	I2C2_SMBUS_TIMEOUT	I2C1_SMBUS_TIMEOUT	WWDG_STOP	IWDG_STOP	Reserved							

Bit	name	access	describe	Reset value
31	Reserved	RW reserved.		0
30	CAN3_STOP	RW	CAN3 is the debug stop bit. CAN2 stops when the kernel enters debug mode. Stop running. 1: The CAN3 receive register will stop receiving data; 0: CAN3 is still functioning normally.	0
29	CAN2_STOP	RW	CAN2 debug stop bit. CAN2 stops when the kernel enters debug mode. Stop running. 1: The CAN2 receive register will stop receiving data; 0: CAN2 is still running normally.	0
28	CAN1_STOP	RW	CAN1 is the debug stop bit. CAN2 stops when the kernel enters debug mode. Stop running. 1: The CAN1 receive register will stop receiving data; 0: CAN1 is still functioning normally.	0
27	I2C4_SMBUS_TIMEOUT	RW	The I2C4_SMBUS timeout mode debug stop bit. This is used when the kernel enters debug mode. Stop SMBUS timeout mode when in the current state. 1: Freeze the timeout control of SMBUS; 0: Same as normal mode operation.	0

26	I2C3_SMBUS_TIMEOUT	RW	The I2C3_SMBUS timeout mode debug stop bit. This is used when the kernel enters debug mode. Stop SMBUS timeout mode when in the current state. 1: Freeze the timeout control of SMBUS; 0: Same as normal mode operation.	0
25	LPTIM2_STOP	RW	Low-power Timer 2 debug stop bit. This is used when the kernel enters debug mode. The time counter stops working. 1: The counter of low-power timer 2 stops working; 0: The counter of low-power timer 2 is still working normally.	0
twenty four	LPTIM1_STOP	RW	Low-power Timer 1 debug stop bit. This is used when the kernel enters debug mode. The time counter stops working. 1: The counter of low-power timer 10 stops working; 0: The counter of low-power timer 1 is still working normally.	0
twenty three	TIM12_STOP	RW	Timer 12 is the debug stop bit. It is used when the kernel enters debug mode. The counter has stopped working. 1: Timer 12's counter stops working; 0: Timer 12's counter is still working normally.	0
twenty two	TIM11_STOP	RW	Timer 11 is the debug stop bit. It is used when the kernel enters debug mode. The counter has stopped working. 1: The counter of Timer 11 stops working; 0: Timer 11's counter is still working normally.	0
twenty one	TIM10_STOP	RW	Timer 10 is the debug stop bit. It is used when the kernel enters debug mode. The counter has stopped working. 1: Timer 10's counter stops working; 0: Timer 10's counter is still working normally.	0
20	TIM9_STOP	RW	Timer 9 Debug Stop Bit. Counts when the kernel enters debug mode. The device stopped working. 1: Timer 9's counter has stopped working; 0: Timer 9's counter is still working normally.	0
19	TIM8_STOP	RW	Timer 8 Debug Stop Bit. Counts when the kernel enters debug mode. The device stopped working. 1: Timer 8's counter has stopped working; 0: Timer 8's counter is still working normally.	0
18	TIM7_STOP	RW	Timer 7 debug stop bit. Counts when the kernel enters debug mode. The device stopped working. 1: Timer 7's counter stops working; 0: Timer 7's counter is still working normally.	0
17	TIM6_STOP	RW	Timer 6 Debug Stop Bit. Counts when the kernel enters debug mode. The device stopped working. 1: Timer 6's counter has stopped working; 0: Timer 6's counter is still working normally.	0
16	TIM5_STOP	RW	Timer 5 Debug Stop Bit. Counts when the kernel enters debug mode. The device stopped working. 1: Timer 5's counter stops working; 0: Timer 5's counter is still working normally.	0
15	TIM4_STOP	RW	Timer 4 Debug Stop Bit. Counts to 0 when the kernel enters debug mode.	

			The device stopped working. 1: Timer 4's counter has stopped working; 0: Timer 4's counter is still working normally.	
14	TIM3_STOP	RW	Timer 3 Debug Stop Bit. Counts when the kernel enters debug mode. The device stopped working. 1: Timer 3's counter stops working; 0: Timer 3's counter is still working normally.	0
13	TIM2_STOP	RW	Timer 2 Debug Stop Bit. Counts when the kernel enters debug mode. The device stopped working. 1: Timer 2's counter stops working; 0: Timer 2's counter is still working normally.	0
12	TIM1_STOP	RW	Timer 1 Debug Stop Bit. Counts when the kernel enters debug mode. The device stopped working. 1: Timer 1's counter stops working; 0: Timer 1's counter is still working normally.	0
11	I2C2_SMBUS_TIMEOUT	RW	The I2C2_SMBUS timeout mode debug stop bit. This is used when the kernel enters debug mode. Stop SMBUS timeout mode when in the current state. 1: Freeze the timeout control of SMBUS; 0: Same as normal mode operation.	0
10	I2C1_SMBUS_TIMEOUT	RW	The I2C1_SMBUS timeout mode debug stop bit. This is used when the kernel enters debug mode. Stop SMBUS timeout mode when in the current state. 1: Freeze the timeout control of SMBUS; 0: Same as normal mode operation.	0
9	WWDG_STOP	RW	Window watchdog debug stop bit. This bit is used when the kernel enters debug mode. The watchdog timer at the test window has stopped working. 1: The window watchdog counter has stopped working; 0: The window watchdog counter is still working normally.	0
8	IWDG_STOP	RW	Independent watchdog debugging stop bit. Watches when the kernel enters debug mode. The watchdog has stopped working. 1: The watchdog counter has stopped working; 0: The watchdog timer is still working properly.	0
[7:0] Reserved		RW reserved.		0

Chapter 46 Flash Memory and User-Selected Words (FLASH)

46.1 Flash Memory Organization

The internal flash memory organization structure of the chip is as follows:

Table 46-1 Flash Memory Organization 1 (User Area 960K + BOOT Area 56K)

piece	name	Address range	Size (bytes)
main memory	Page 0	0x08000000 – 0x080000FF	256
	Page	0x08000100 – 0x080001FF	256
	1 Page	0x08000200 – 0x080002FF	256
	2 Page	0x08000300 – 0x080003FF	256
	3 Page	0x08000400 – 0x080004FF	256
	4 Page	0x08000500 – 0x080005FF	256
	5 Page	0x08000600 – 0x080006FF	256
	6 Page 7	0x08000700 – 0x080007FF	256
	…	…	…
	Page	0x080EFF00 – 0x080EFFFF	256
Information block	3839 System boot code storage 0x1FFF0000 – 0x1FFFDFFF		56K
	User selection word 0x1FFFF800 – 0x1FFFF8FF		256

Note: The main memory area described above is used for user application storage and is write-protected in units of bytes (32 pages); except for "factory"
The
"8K Commercial Configuration Word" area is factory locked and cannot be accessed by users, while other areas can be operated by users under certain conditions.

Table 46-2 Flash Memory Organization Structure 2 (User Area 480K + BOOT Area 28K)

piece	name	Address range	Size (bytes)
main memory	Page 0	0x08000000 – 0x080000FF	256
	Page	0x08000100 – 0x080001FF	256
	1 Page	0x08000200 – 0x080002FF	256
	2 Page	0x08000300 – 0x080003FF	256
	3 Page	0x08000400 – 0x080004FF	256
	4 Page	0x08000500 – 0x080005FF	256
	5 Page	0x08000600 – 0x080006FF	256
	6 Page 7	0x08000700 – 0x080007FF	256
	…	…	…
	Page	0x08077F00 – 0x08077FFF	256
Information block	1919 System boot code storage 0x1FFF0000 – 0x1FFF6FFF		28K
	User selection word 0x1FFFF800 – 0x1FFFF8FF		256

Note: The main memory area described above is used for user application storage and is write-protected in units of bytes (16 pages); except for "factory"
The
"4K Commercial Configuration Word" area is factory locked and cannot be accessed by the user, while other areas can be operated by the user under certain conditions.

46.2 Flash Programming and Security

46.2.1 Two Programming/Erasing Methods ÿ

Standard Programming: This is the default programming method (compatibility method). In this mode, the CPU performs programming in 2-byte increments.

When DBMODE=1, the chip performs an erase operation of 8KB at a time; when DBMODE=0, the chip performs an erase operation of 4KB at a time.

ÿ Fast Programming: This method uses page operations (recommended). After unlocking according to a specific sequence, it performs a single 256-byte programming operation.

When DBMODE=1, the chip performs an erase operation of 64KB at a time; when DBMODE=0, the chip performs an erase operation of 32KB at a time.

46.2.2 Security - Preventing unauthorized access (read, write, erase) ÿ Page write protection

Read protection

When the chip is in read-protected state:

1) When DBMODE=1, pages 0-31 (8KB) of the chip's main memory are in automatic write-protected state and are not controlled by the R32_FLASH_WPR register.

Remove read protection; all main memory pages are now controlled by the R32_FLASH_WPR register.

2) When DBMODE=0, pages 0-15 (4KB) of the chip's main memory are in automatic write-protected state and are not controlled by the R32_FLASH_WPR register.

Remove read protection; all main memory pages are now controlled by the R32_FLASH_WPR register.

3) The system boot code area, SWD or SDI mode, and RAM area cannot be erased or programmed into the main memory. Erasable or programmable areas...

The user selects a word area. If an attempt is made to remove read protection (program the user word), the chip will automatically erase the entire user area.

46.3 Enhanced Flash Read Mode

Enhanced Flash Read Mode is applicable when user programs are running in Flash. Enabling this mode improves Flash access efficiency.

To enter enhanced read mode, the EHMOD bit in the R32_FLASH_ACTLR register must be set to 1. At this point, the EHMODS bit is read; having EHMODS set to 1 indicates enhanced read mode.

This mode is now active. To disable it, you must first clear the EHMOD bit to 0 and then set RSENACT to 1.

Note: When using FLASH enhanced read mode, please pay attention to the following points:

1) Before performing any erase or program operation on the FLASH (including user word programming such as deread protection), you must first exit the program.

Enhance the read mode; otherwise, erase and programming operations will fail.

twenty Before entering stop mode, you must exit enhanced read mode first, otherwise stop mode may malfunction.

three) After the power reset and system reset are completed, the chip automatically exits the enhanced read mode under hardware control.

46.4 Register Description

Table 46-3 List of FLASH Related Registers

name	Access address	describe	Reset value
R32_FLASH_ACTLR	0x40022000	Access Control Register	0x00000000
R32_FLASH_KEYR	0x40022004	FPEC key register	0xXXXXXXXX
R32_FLASH_OBKEYR	0x40022008	OBKEY register	0xXXXXXXXX
R32_FLASH_STATR	0x4002200C	Status Register;	0x00008000
R32_FLASH_CTLR	0x40022010	Control Register;	0x00008080
R32_FLASH_ADDR	0x40022014	Address Register;	0x00000000
R32_FLASH_OBR	0x4002201C	Select Word Register;	0xXXXXXXFE
R32_FLASH_WPR	0x40022020	Write Protect Register;	0xXXXXXXXX
R32_FLASH_MODEKEYR	0x40022024	Extended Key Register	0xXXXXXXXX
R32_BOOT_MODEKEYR	0x40022028	BOOT key register	0xXXXXXXXX
R32_FLASH_CFGR0	0x4002202C	Configuration Register 1	0xXXXXXXXX

46.4.1 Access Control Register (R32_FLASH_ACTLR) Offset Address:

0x00

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16

Reserved															
15	14 13	12	11	10	9	8	7	6	5	4	3	2	1	0	
FLASH_ST	FLASH_READY	Reserved	FLASH_RD_MD	Reserved		FLASH_LP	EHMOD	ENHANCE_STATUS	Reserved				SCK_CFG		

Bit	name	access	describe	Reset value
[31:16] Reserved		RO is reserved.		0
15	FLASH_ST	RO	FLASH low power status indicator bit: 1: The FLASH memory is in low-power mode; 0: FLASH is in normal working mode.	0
14	FLASH_READY	RO	FLASH Ready Indicator Bit: 1: Flash is in place; 0: Flash is not ready.	0
[13:12] Reserved		R0 is reserved.		0
11	FLASH_RD_MD	RW	FLASH read mode control bits: 1: Continuous read mode; 0: Single read mode. This is a normal unlock.	0
[10:9] Reserved		R0 is reserved.		0
8	FLASH_LP	RW	FLASH low power mode control bits: 1: FLASH Low Power Mode; 0: FLASH exits low-power mode.	0
7	EHMOD	RW	FLASH Enhanced Read Mode Control Bits: This mode can improve access speed when the program runs in FLASH. Ask about efficiency. 1: Enable FLASH enhanced read mode; 0: Disable Flash Enhanced Read Mode (requires RSENACT) When operating on all bits together, the exit step first clears the EHMOD bit to 0, then... Set RSENACT to 1. This is a normal unlock.	0
6	ENHANCE_STATUS	RO	Enhanced state: 1: The command read has been sent successfully; 0: Waiting.	0
[5:2] Reserved		RO is reserved.		0
[1:0] SCK_CFG		RW	Number of FLASH wait states: 00: HCLK; FLASH read operation -- HCLK ÿ 75MHz recommended. FLASH erase/write operation -- HCLK ÿ 60MHz recommended; 01: HCLK/2; FLASH read operation -- HCLK ÿ 150MHz recommended. FLASH erase/write operation -- HCLK ÿ 120MHz is recommended; 10: HCLK/4; FLASH read operations -- HCLK maximum 150MHz, not recommended.	0

			Use this gear. FLASH erase/write operation -- HCLK ÿ 150MHz recommended; 11: Retained.	

46.4.2 FPEC Key Register (R32_FLASH_KEYR) Offset Address:

0x04

31	30	29	28	27	26	25	24	23	22	21	2019	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	The	Reset value
[31:0]	KEYR[31:0]	WO	FPEC key, used to input the FPEC key, includes the following unlock key: RDPRT key = 0x000000A5; KEY1 = 0x45670123; KEY2 = 0xCDEF89AB.	X

46.4.3 OBKEY Register (R32_FLASH_OBKEYR) Offset Address:

0x08

31	30	29	28	27	26	25	24	23	22	21	2019	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	The	Reset value
[31:0]	OBKEYR[31:0]	WO	description selects the key used to deselect OPTWRE. KEY1 = 0x45670123; KEY2 = 0xCDEF89AB.	X

46.4.4 Status Register (R32_FLASH_STATR) Offset Address:

0x0C

31	30	29	28	27	26	25	24	23	twenty two	twenty one	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOOT_LOCK	BOOT_MODE	BOOT_STATUS	BOOT_AVA	Reserved						EOP	WRPRTERR	Reserved		WRBSY	BSY

Bit	name	access	describe	Reset value
[31:16]	Reserved	RO	is reserved.	0
15	BOOT_LOCK	RW1	BOOT area locked: 1: Lock;	1

			0: Unlock.	
14	BOOT_MODE	RW	Combined with BOOT_AVA, it can control the user space and BOOT space. Switching between them: 1. After a software reset, you can switch to the BOOT area or use... Household area; 0: After a software reset, you can switch to the user area.	0
13	BOOT_STATUS	RW0	The source of the currently executing program, write 0 to clear: 1: Indicates a program loaded from the BOOT area; 0: Indicates a program loaded from the user area.	0
12	BOOT_AVA	RO	Program initialization status: 1: Indicates booting from the BOOT area; 0: Indicates starting from the user area.	0
[11:6] Reserved		RO is reserved.		0
5	EOP	RW1Z	Indicates the end of the operation; writing 1 clears the value. The hardware sets this bit each time an erase or program is successfully	0
4	WRPRTERR	RW1Z	executed. A write-protection error is indicated; writing a 1 clears the bit. The hardware will set the bit when programming a write-protected address.	0
[3:2] Reserved		RO is reserved. This		0
1	WRBSY	RO	bit is used during fast page programming to indicate that programming data is being processed. During writing. In page programming, this bit is set when data is written. If the bit is set to '1', the hardware will automatically clear it to '0'; if the bit is '0', This indicates that the next piece of data can be written.	0
0	BSY	RO	Indicates busy status: 1: Indicates that flash memory operation is in progress; 0: Operation complete.	0

46.4.5 Control Register (R32_FLASH_CTLR)

Offset address: 0x10

31	30	29 28 27 26 25 24	23 22	twenty one	2019	18	17 16	
Reserved			RSENACT	PGSTRT	Reserved BER		Reserved	FTPG

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLOCK	Reserved	EOPIE		Reserved	ERRIE	OBWRE	Reserved	LOCK	STRT	OBER	OBPG	Reserved	PER	PG	

Bit	name	access	describe	Reset value
[31:23] Reserved		RO is reserved. Exiting		0
twenty two	RSENACT	WO	enhanced read mode will automatically clear the hardware; this requires further assistance. When operating on the EHMOD bit together, the exit procedure should first remove the EHMOD bit. Clear to 0, then set RSENACT to 1.	0
twenty one	PGSTRT	WO	Start. Set to 1 to initiate page programming once; hardware will automatically clear. This is a normal unlock.	0
[20:19] Reserved		RO is reserved.		0
18	BER	RW	When DBMODE=1, a 64KB erase is performed. When DBMODE=0, a 32KB erase is performed. This option is for both normal unlock and quick unlock.	0

17	Reserved	RO is reserved.		0
16	FTPG	RW	Perform a fast page programming operation. This option is for both normal unlock and quick unlock.	0
15	FLOCK	RW1	A fast programmable lock. Only '1' can be written to. When this bit is '1'... This indicates that the fast programmable/erase mode is unavailable. (This is because) a positive error was detected. After confirming the unlock sequence, the hardware clears this bit to '0'. The software is set to 1, and the lock is re-engaged. This is a normal unlock.	1
[14:13] Reserved		RO is reserved.		0
12	EOPIE	RW	Operation completion interrupt control (R32_FLASH_STATR register) (EOP set in the middle) 1: Allow interrupts; 0: Disallow interrupts. This is a normal unlock.	0
11	Reserved	RO is reserved.		0
10	ERRIE	RW	Error status interrupt control (R32_FLASH_STATR register) (Set PGERR/WRPRTERR bit) 1: Allow interrupts; 0: Disallow interrupts. This is a normal unlock.	0
9	OBWRE	RW0	The user selects a word lock, and the software clears it to 0. 1: Indicates that programmatic operations can be performed on the user-selected text. (Required) To write the correct value into the R32_FLASH_OBKEYR register The sequence is then set by hardware; 0: After the software is reset, the user-selected word is relocked. This is a normal unlock.	0
8	Reserved	RO Reserved.		0
7	LOCK	RW1	Locked. Can only be written to as '1'. When this bit is '1', it indicates FPEC. R32_FLASH_CTLR is locked and cannot be written. This was detected... After the correct unlock sequence is obtained, the hardware clears this bit to '0'. After an unsuccessful unlock attempt, until the next system... This bit will not change before the reset. This is a normal unlock.	1
6	STRT	RW1	Start. Setting to 1 initiates an erase operation; the hardware will automatically clear. 0 (BSY becomes '0'). This is a normal unlock.	0
5	OBER	RW	Perform user-selected word erasure. This is a normal unlock.	0
4	OBPG	RW	Perform user-selected word programming. This is a normal unlock.	0
[3:2] Reserved		RO is reserved.		0
1	PER	RW	When DBMODE=1, a sector (8KB) erase operation is performed. When DBMODE=0, a sector (4KB) erase operation is performed. This is a normal unlock.	0
0	PG	RW performs standard programming operations.		0

			This is a normal unlock.	

46.4.6 Address Register (R32_FLASH_ADDR)

Offset address: 0x14

31	30	29	28	27	26	25	24	23	22	21	2019	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	describe	Reset value
[31:0]	FAR[31:0]	WO	The flash memory address is the address used for programming and erasing. The value is the starting address for erasure. When the BSY bit in the R32_FLASH_STATR register is '1' This register cannot be written to at this time.	0

46.4.8 Select word register (R32_FLASH_OBR) offset address:

0x1C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											DATA1[7:0]				DATA0[7:6]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA0[5:0]						USARTDLEN	USBFSDLEN	Reserved					IWDG_SW	RDPRT	OBERR

	name		access	describe	Reset value
[31:26]	Reserved		RO	is reserved.	0
[25:18]	DATA1[7:0]		RO	data byte 1.	X
[17:10]	DATA0[7:0]		RO	data bytes 0.	X
9	USER [2:0]	USARTDLEN	RO	BOOT enables USART buttonless download functionality: 1: Turn on; 0: Off.	1
8		USBFSDLEN	RO	BOOT enables USBFS download functionality: 1: Turn on; 0: Off.	1
[7:3]		Reserved	RO	is reserved.	11111b
2		IWDG_SW	RO	Independent Watchdog Timer (IWDG) hardware enable bit; low level enables it. effect.	1
1	RDPRT		RO	Read protected status. 1: Indicates that read protection is currently active for the flash memory.	1
0	OBERR		RO	Incorrect word selection. 1: Indicates that the selected word and its complement do not match.	0

46.4.9 Write Protect Register (R32_FLASH_WPR)

Offset Address: 0x20

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
WRP[31:16]														

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

Bit	name	access	describe	Reset value
[31:0]	WRP[31:0]	RO	Flash memory is write-protected. 1: Write protection failed; 0: Write protection is enabled. When DBMODE=1, each bit of the chip represents 8K words. Section (page 32) stores write-protected state, in read-protected state The first 8KB are write-protected. When DBMODE=0, each bit of the chip represents 4K words. Section (page 16) stores write-protected state, in read-protected state The first 4KB are write-protected.	X

46.4.10 Extended Key Register (R32_FLASH_MODEKEYR)

Offset Address: 0x24

31	30	29	28	27	26	25	24	23	22	21	2019	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	describe	Reset value
[31:0]	MODEKEYR[31:0]	WO	Enter the following sequence to unlock quick programming/erase mode: KEY1 = 0x45670123; KEY2 = 0xCDEF89AB.	X

46.4.11 BOOT key register (R32_BOOT_MODEKEYR) offset address: 0x28

31	30	29	28	27	26	25	24	23	22	21	2019	18	17	16
BOOTKEYR[31:16]														

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

Bit	name	access	describe	Reset value
[31:0]	BOOTKEYR[31:0]		To unlock the BOOT lock, enter the following sequence:	X

			KEY1 = 0x45670123; KEY2 = 0xCDEF89AB.	

46.4.12 Configuration Register 1 (R32_FLASH_CFGR0) Offset

Address: 0x2C

31	30	29 28 27 26 25 24 23 22 21	2019	18	17 16
Reserved		DBMODE	Reserved		

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

Bit name		access	describe	Reset value
[31:29] Reserved		RO is reserved.		x
28	DBMODE	RO	FLASH mode status bits: 1: Double Flash Mode; 0: Single Flash mode.	x
[27:0] Reserved		RO is reserved.		x

46.5 Flash Memory Operation Procedure

46.5.1 Read operations

are performed directly in the general address space; any 8/16/32-bit data read operation can access the contents of the flash memory module and obtain the data.
The corresponding data.

46.5.2 Unlocking the Flash Memory

After a system reset, the Flash Memory Controller (FPEC) and the R32_FLASH_CTLR register are locked and inaccessible. This can be achieved by writing to...
Sequence to the R32_FLASH_KEYR register unlocks the flash controller module.

Unlock sequence:

1) Write KEY1 = 0x45670123 to the R32_FLASH_KEYR register (step 1 must be KEY1);

2) Write KEY2 = 0xCDEF89AB to the R32_FLASH_KEYR register (step 2 must be KEY2).

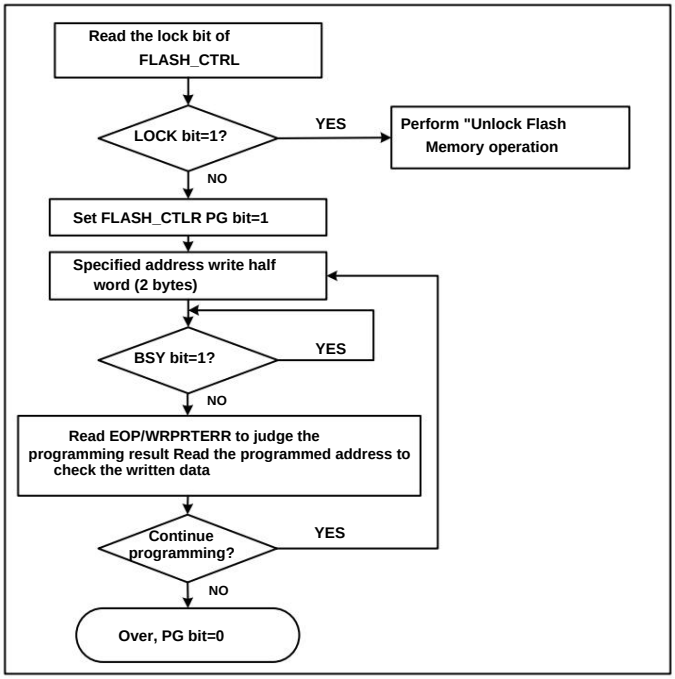
The above operations must be performed sequentially and consecutively; otherwise, it is considered an erroneous operation and will lock the FPEC module and the R32_FLASH_CTLR register.
This will generate a bus error until the next system reset.

The Flash Controller (FPEC) and the R32_FLASH_CTLR register can be accessed by changing the "LOCK" bit of the R32_FLASH_CTLR register.
Set to 1 to lock again.

46.5.3 Standard Main Memory Programming

Standard programming allows writing 2 bytes at a time. When the PG bit of the R32_FLASH_CTLR register is '1', each write to the flash memory address...
Writing a half-word (2 bytes) will initiate a programming session; writing any non-half-word data will generate a bus error in FPEC. During programming,
When the BSY bit is '1', programming ends; when the BSY bit is '0', the EOP bit is '1'.
Note: When the BSY bit is '1', write operations to any register will be prohibited.

Figure 46-1 FLASH Programming



1) Check the R32_FLASH_CTLR register LOCK. If it is 1, you need to perform the "unlock flash memory" operation.

2) Set the PG bit of the R32_FLASH_CTLR register to '1' to enable standard programming mode.

3) Write the half-word to be programmed to the specified flash memory address (even address).

4) Wait for the BSY bit to become '0' or the EOP bit in the R32_FLASH_STATR register to become '1' to indicate the end of programming, then clear the EOP bit to 0.

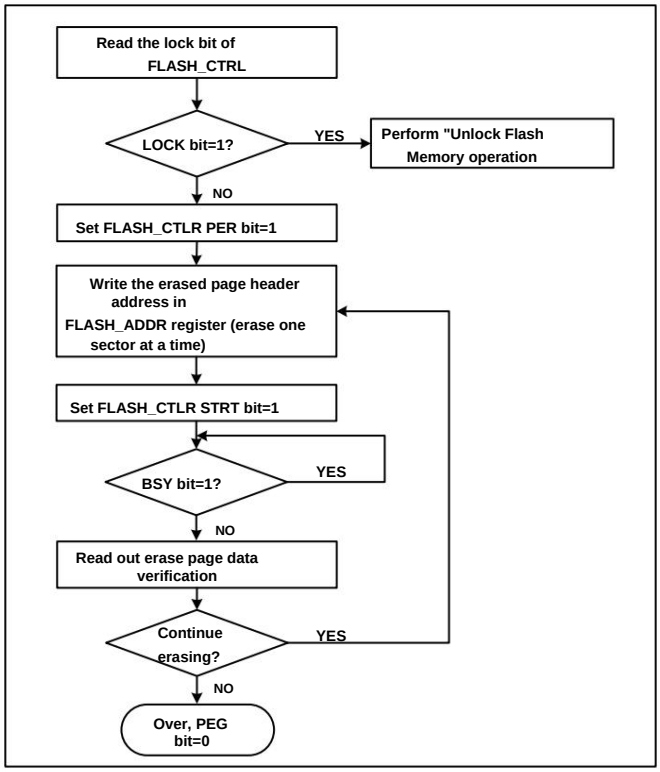
5) Check the R32_FLASH_STATR register for errors, or read the programming address data for verification.

6) To continue programming, repeat steps 3-5. To finish programming, clear the PG bit to 0.

46.5.4 Main memory standard erase flash memory

erased by standard page.

Figure 46-2 FLASH Page Erase



1) Check the LOCK bit of the R32_FLASH_CTLR register. If it is 1, you need to perform the "unlock flash memory" operation.

2) Set the PER bit of the R32_FLASH_CTLR register to '1' to enable standard page erase mode.

3) Write the page start address to be erased to the R32_FLASH_ADDR register.

4) Set the STRT bit of the R32_FLASH_CTLR register to '1' to initiate an erase operation.

5) Wait for the BSY bit to become '0' or the EOP bit in the R32_FLASH_STATR register to become '1' to indicate that the erasure is complete, then clear the EOP bit to 0.

6) Read and verify the data from the erased page.

7) To continue standard page erasure, repeat steps 3-5. To finish erasing, clear the PEG bit to 0.

Note: After successful erasure, read the word as -0xe339e339, read the half-word as -0xe339, read the even address byte as -0x39, and read the odd address as 0xe3.

46.5.5 Quick Programming Mode Unlocked

Fast programming mode operation can be unlocked by writing a sequence to the R32_FLASH_MODEKEYR register. After unlocking, R32_FLASH_CTLR

The FLOCK bit in the register will be cleared to 0, indicating that fast erase and programming operations can be performed. This is achieved by clearing the FLOCK bit in the R32_FLASH_CTLR register.

The software sets the "FLOCK" bit to 1 to lock the device again.

Unlock sequence:

1) Write KEY1 = 0x45670123 to the R32_FLASH_MODEKEYR register;

2) Write KEY2 = 0xCDEF89AB to the R32_FLASH_MODEKEYR register.

The above operations must be performed in sequence and consecutively; otherwise, they will be considered incorrect and the system will be locked until the next system reset.

Note: Quick programming operations require unlocking both "LOCK" and "FLOCK" locks.

46.5.6 Main Memory Quick Programming Quick

programming is performed page-by-page (256 bytes).

1) Check the LOCK bit of the R32_FLASH_CTLR register. If it is '1', you need to perform the "unlock flash memory" operation.

2) Check the FLOCK bit in the R32_FLASH_CTLR register. If it is '1', you need to perform the "Quick Programming Mode Unlock" operation.

3) Check the BSY bit of the R32_FLASH_STATR register to confirm that no other programming operations are in progress.

4) Set the FTPG bit of the R32_FLASH_CTLR register to '1' to enable fast page programming mode.

5) Write data to the FLASH address using 32-bit method, for example:

(uint32_t)0x8000000 = 0x12345678;

6) Wait for the WRBSY value in the R32_FLASH_STATR register to be '0' before writing the next data.

7) Repeat steps 5-6 a total of 64 times.

8) Set the FTPG bit of the R32_FLASH_CTLR register to '1' to enable fast page programming.

9) Wait for the BSY bit to become '0' or the EOP bit in the R32_FLASH_STATR register to become '1', indicating that a fast page programming operation is complete.

Clear the EOP bit to 0.

10) Check the R32_FLASH_STATR register for errors, or read the programming address data for verification.

11) To continue with quick page programming, repeat steps 5-10. To finish programming, clear the FTPG bit to 0.

46.5.7 Main Memory Fast Erase Fast erase

erases blocks by block.

1) Check the LOCK bit of the R32_FLASH_CTLR register. If it is 1, you need to perform the "unlock flash memory" operation.

2) Check the FLOCK bit in the R32_FLASH_CTLR register. If it is 1, you need to perform the "Quick Programming Mode Unlock" operation.

3) Check the BSY bit of the R32_FLASH_STATR register to confirm that no other programming operations are in progress.

4) Set the BER bit of the R32_FLASH_CTLR register to '1' to enable the fast block erase mode.

5) Write the starting address of the fast erase block to the R32_FLASH_ADDR register.

6) Set the STRT bit of the R32_FLASH_CTLR register to '1' to initiate a fast block erase operation.

7) Wait for the BSY bit to become '0' or the EOP bit in the R32_FLASH_STATR register to become '1' to indicate that the erasure is complete, then clear the EOP bit to 0.

8) Check the R32_FLASH_STATR register for errors, or read the erase page address data for verification.

9) To continue with quick page erasure, repeat steps 5-8. To finish erasing, clear the BER bit to 0.

Note: After successful erasure, read the word as -0xe339e339, read the half-word as -0xe339, read the even address byte as -0x39, and read the odd address as 0xe3.

46.6 User-selected words

The user-selected word is stored in the FLASH memory and will be reloaded into the corresponding register after a system reset. The user can erase it at will.

Besides programming, the user selection word information block consists of 8 bytes in total (4 bytes for write protection, 1 byte for read protection, and 1 byte for configuration).

The option (2 bytes for storing user data) has a one's complement bit for verification during loading. The following describes the option word.

Information structure and meaning.

Table 46-4 32-bit select word format division

[31:24]	[23:16] [15:8] Select byte 1 Select byte 0 Inverse	[7:0]
Select word byte 1 inverse code	code	Select word byte 0

Table 46-5 Structure of User Selected Word Information

address / Bit	[31:24]	[23:16]	[15:8]	[7:0]
0x1FFFF800	nUSER	USER	nRDPR	RDPR
0x1FFFF804	nData1	Data1	nData0	Data0
0x1FFFF808	nWRPR1	WRPR1	nWRPR0	WRPR0
0x1FFFF80C	nWRPR3	WRPR3	nWRPR2	WRPR2

Name/Bytes	describe	Reset value
RDPR	The read protection control bit configures whether the code in the flash memory can be read. 0xA5: If this byte is 0xA5 (nRDP must be 0x5A), it indicates the current... The code is not read-protected and can be read.	0xA5

			Other values: Indicate code read protection status, not readable, when DBMODE=1. Pages 0-31 (8K) of the chip will be automatically write-protected and will not be controlled by WRPR0; when When DBMODE=0, pages 0-15 (4K) of the chip will be automatically write-protected and will not be affected by WRPR0. control.	
USER	7	USARTDLEN	BOOT enables USART keyless download function [1]: 1: Turn on; 0: Off.	1
	6	USBFSDLEN	BOOT enables USBFS download functionality[1]: 1: Turn on; 0: Off.	1
	[5:1] Reserved.			0x1F
	0	IWDGSW	Independent Watchdog Timer (IWDG) hardware enable bit: 0: The IWDG function is enabled by hardware (determined by the LSI clock); 1: The IWDG function is enabled by software and cannot be enabled by hardware.	1
Data0–Data1			Stores 2 bytes of user data. Write	X
WRPR0-WRPR3			protection control bits. Each bit controls one sector in main memory. Sector (8KB/sector when DBMODE=1; 4KB/sector when DBMODE=0) Write-protected status (byte/sector): 0: Enable write protection; 1: Disable write protection. Four bytes are used to protect the main memory. WRPRO: Write protection control for sectors 0-7; WRPR1: Write protection control for sectors 8-15; WRPR2: Write protection control for sectors 16-23; WRPR3: Bits 0-6 provide write protection for sectors 24-30; Bit 7 provides write protection for sectors 24-30. Write protection for sectors 31-119.	0xFFFF FFFF

Note [1]: For specific operation methods and precautions of this function, please refer to the CH32H417 evaluation board instruction manual.

46.6.1 User Select Word Unlocking: User

select word operations can be unlocked by writing a sequence to the R32_FLASH_OBKEYR register. After unlocking, the R32_FLASH_CTLR register...

The OBWRE bit of the register will be set to 1, indicating that user-selectable word erasure and programming are possible. This is achieved by adjusting the R32_FLASH_CTLR register.

The "OBWRE" bit is cleared to 0 by the software to lock it again.

Unlock sequence:

1) Write KEY1 = 0x45670123 to the R32_FLASH_OBKEYR register;

2) Write KEY2 = 0xCDEF89AB to the R32_FLASH_OBKEYR register.

Note: Users need to unlock both "LOCK" and "OBWRE" layers to select text operations.

46.6.2 User-selectable word programming

only supports standard programming methods, writing half a word (2 bytes) at a time. In practice, when programming user-selectable words, FPEC...

Only the low byte of the half-word is used, and the high byte is automatically calculated (the high byte is the inverse of the low byte), and then programming operations begin. This will

Ensure that the bytes selected by the user and their complements are always correct.

1) Check the LOCK bit of the R32_FLASH_CTLR register. If it is 1, you need to perform the "unlock flash memory" operation.

2) Check the BSY bit of the R32_FLASH_STATR register to confirm that no other programming operations are in progress.

3) Check the OBWRE bit of the R32_FLASH_CTLR register. If it is 0, you need to perform the "user select word unlock" operation.

4) Set the OBPG bit of the R32_FLASH_CTLR register to '1'. 5) Write the half-word
(2 bytes) to be programmed to the specified address. 6) Wait for the BSY
bit to become '0' or the EOP bit of the R32_FLASH_STATR register to become '1', indicating the end of programming, then clear the EOP bit to 0. 7) Read the
programming address data for
verification. 8) To continue programming, repeat steps 5-7. To end programming, clear the OBPG bit to 0.

Note: When the "Read Protection" setting in the selection word is changed to "Unprotected", a full erase operation of the main memory area will be automatically performed. If any selection other than "Read Protection" is changed, a full erase operation will not occur.

46.6.3 User-selected word erasure directly

erases the entire 128-byte user-selected word region.

1) Check the LOCK bit of the R32_FLASH_CTLR register. If it is 1, a "unlock flash memory" operation is required. 2) Check the BSY bit
of the R32_FLASH_STATR register to confirm that no programming operation is in progress. 3) Check the OBWRE bit of the
R32_FLASH_CTLR register. If it is 0, a "user select word unlock" operation is required. 4) Set the OBER bit of the R32_FLASH_CTLR register to
'1', then set the STRT bit of the R32_FLASH_CTLR register to '1'.

'1' enables user-selected word erasure.

5) Wait for the BSY bit to become '0' or the EOP bit in the R32_FLASH_STATR register to become '1' to indicate the end of erasure, then clear the EOP bit to 0. 6)
Read the erased address data for
verification. 7) Clear the OBER bit to 0 to finish.

Note: After successful erasure, the word read is -0xe339e339, the half-word read is -0xe339, and the byte read is -0x39.

46.6.4 Removing Read Protection:

Whether the flash memory is read protected is determined by the user-selected word. Read the R32_FLASH_OBR register; when the RDPRT bit is '1', it indicates that...
The front flash memory is in read-protected mode, and flash memory operations are subject to a series of safety protections under read-protection. The
read-protection process is as follows: 1) Erase the entire user select word area. At this time, the read protection field RDPR
is still active; 2) Program the user select word, writing the correct RDPR code 0xA5 to remove the read protection of the flash memory; (this step will first cause
the system to automatically perform a full erase
operation on the flash memory); 3) Perform a power-on reset to reload the select byte (including the new RDPR code), at which point the read protection is removed.

46.6.5 De-write Protection: Whether

the flash memory is write-protected is determined by the user-selected word. Read the R32_FLASH_WPR register; each bit represents a sector of the flash memory.
The storage space is configured such that a bit '1' indicates a non-write-protected state, and a bit '0' indicates a write-protected state. The write-
protection process is as follows: 1) Erase the
entire user selection word area; 2) Write the correct RDPR code 0xA5
to allow read access; 3) Perform a system reset and reload the selection bytes (including the new WRPR[3:0] bytes), thus removing the write protection.

Chapter 47 Universal High-Speed Interface (UHSIF)

The module descriptions in this chapter apply only to the CH32H417 microcontroller product.

The chip integrates a set of universal high-speed interfaces, UHSIF, with a transmission clock speed of up to 125MHz, supporting 8-bit, 16-bit, or 32-bit data widths.

The theoretical maximum speed is 500 Mbytes/s.

For specific applications, please refer to and call the provided subroutine library.