TX8C101x User manual

泰芯半导体
TaiXin Semiconductor

珠海泰芯半导体有限公司
Zhuhai Taixin Semiconductor Co., Limited

珠海市高新区港湾一号科创园港 11 栋 3 楼

Revision records

Date	Version	Description	Redactor
2023-08-31	V2.8	1, update the English name description of the company in the footer;	FWF
2023-08-30	V2.7	1, update the logo pattern;	FWF
2023-08-21	V2.6	1, modify the description of the ADC reference voltage;	FWF
2023-05-31	V2.5	1, modify the description of the ADC reference voltage;	FWF
2023-05-17	V2.4	1, modify the description of GPIO low power wake-up, change to level wake-up; 2. Move the detailed description of the wake-up register to low power management; 3. Modify the description of the ADC's internal reference voltage;	FWF
2023-02-03	V2.3	1. Solve the problem of garbled code when some devices are opened to read;	HLW
2022-10-20	V2.2	1, modify the comparator part of the functional description; 2, modify the LP_CON description;	ZJ, HLW
The 2022-8-10	V2.1	Modify the frequency division ratio description of the ADC module.	HLW
The 2022-7-28	V2.0	Change the model to TX8C101x, unifying the TX8C1010 and TX8C1011 series.	HLW
The 2022-7-21	V1.13	Amend Section 1 to reduce package model number.	ZJ
The 2022-4-17	V1.12	Fix SPI_CON register bit2 misdescription of sampling pattern.	YLC
The 2022-3-18	V1.11	Fix the error description for bits 1 and 0 of the AIP_CON1 register.	ZJ
The 2022-3-10	V1.10	Modify the description of the n-tube p-tube switch function of the analog comparator CON4.	YLC
The	V1.9	Modified description of analog comparator P03 PAD short circuit protection function.	ZJ

2022-2-25			
The 2022-2-23	V1.8	Fixed TX8C101xSO14 PIN8 pin P14 error description	ZJ
The 2022-2-11	V1.7	Flash control module Add address instructions	HLW
2021-12-28	V1.6	Change encapsulation information for SOP8	ZJ, HLW
2021-12-17	V1.5	Fix SYS_CON2 bit 6 and 7 description error, modify pin diagram description	ZJ
2021-12-07	V1.4	Add the pin diagram for the TX8C101xQF16.	HLW
2021-08-23	V1.0	Initial version, subject to future updates without notice, please contact our staff for the latest version.	ZJ, HLW

Contents

1. Product Overview

1.1. Instructions

TX8C101x is a high performance and low power 8051 core MCU, operating at a maximum frequency of 32MHz, with built-in 4K+256 bytes flash memory (supporting EEPROM-like) and 512 bytes SRAM.

Analog resources: 1 12-bit 200Ksps ADC, 2 multi-function comparators.

Timers, PWM resources (both are mutually exclusive functions and cannot be used simultaneously with the same Timer) :

- 2 16-bit advanced timers, capable of supporting 2 pairs of complementary outputs or 4 independent PWM outputs (same cycle, independently configured duty cycle)
- 1 16-bit general purpose timer (both support Capture, Count, PWM functions)
- 2 8-bit general purpose timers (can be combined into 1 16-bit general purpose timer, both support Capture, Count, PWM functions)

Standard communication interface: 1 SPI interface, 2 UART interface, 1 IIC interface (only TX8C1011 series support).

It supports a wide range of voltage supply, the working voltage is 2.4V ~ 5.5V (can support battery application scenarios), and the working temperature range is -40℃ ~ 85℃. A variety of power-saving operation modes ensure the requirements of low-power applications, and the lowest power consumption mode is 3uA.

TX8C1010 provides SOP8, MSOP10, SOP14, SOP16, QFN16 a total of 5 kinds of packaging forms, according to the different packaging forms, the configuration of peripheral resources in the device is not the same.

Application occasions:

- Small appliances
- E-cigarettes
- Bluetooth charging bin, wireless charging
- Toys

1.2. Features

- **Kernel**
  - Ultra Fast 8051 core (1T)
  - Instructions are fully compatible with traditional 8051
  - Maximum working frequency: 32MHz
  - 14 interrupt sources, support hardware two-level priority
  - Support for online downloads
  - Support for code encryption
  - Supports live burning
  - TX8C1011 series supports online debugging function
- **Operating voltage**
  - 2.4V~5.5V wide voltage range power supply
- **Memory**
  - 4K+256 bytes Flash for storing user code and EEPROM-like support (typical value of 100,000 erases and writes)
  - 512 bytes of RAM
- **Clock**
  - Internal 1-32mhz high precision HIRC with support for calibration (error ±1%)
  - Internal 64KHz low-speed LIRC, calibration supported (error ±1%)

- External 32.768 KHz low speed crystal, need external capacitor

- **Reset**
  - Power on reset
  - Undervoltage reset
  - Reduction foot reduction
  - Watchdog overflow reset
- **GPIO**
  - Up to 14 GPIos
  - All ports can input and output 5V signals
  - All support rising edge/falling edge/double edge interrupts
  - All support wakeup function
  - There are full drive and small drive two gears.
  - Support OD output low mode.
  - Supports independent control of pull-down resistor, 30KΩ resistance
- **LVD low voltage detection reset**
  - Provides 4 levels of low voltage detection voltage (1.8/2.0V, 2.1/2.3V, 2.4/2.6V, 3.4/3.6V)
- **Digital peripherals**
  - 1 SPI high-speed serial interface, supporting master-slave mode
  - 1 I2C interface, support multiple master and slave mode (only TX8C1011 series support)
  - 2 UART ports with up to 4Mbps support
- **Timer resources**
  - 2 16-bit advanced timers, can support 2 pairs of complementary outputs or 4 independent PWM outputs (same cycle, independent duty cycle

configuration), support dead zone insertion and event braking function, support single pulse mode

- 1 16-bit general timer, both support Capture, Count, PWM function
- 2 8-bit universal timers (can be combined into 1 16-bit universal Timer, both support Capture, Count, PWM functions), can support infrared send and receive functions (requires two timers)
- 1 watchdog timer

- **High security**
  - 16 bit CRC validation is supported to ensure data accuracy
- **Low power consumption**
  - Support Idle, Stop, Sleep low power mode
  - Static power consumption 3uA @25 ° C
  - Low power wake-up time is less than 100us
- **1 high precision 12-bit analog-to-digital converter (ADC)**
  - The fastest conversion clock supports 4MHz and the fastest speed is 200Ksps
  - Offset correction step 2mV, DNL +-2 INL +-4
  - There are 13 external input channels and 2 analog channels
  - The effective bit of ADC is about 10bit (5V voltage regulator is supplied, and the ADC is connected to the VCC of the chip through the internal switch. This voltage is used as the reference voltage of the ADC, and the full scale of the ADC is equal to VCC).
- **2 analog comparators (ACmps)**
  - 2 low misaligned comparators with correction step 1mV
  - The comparator supports 120 divider gears for negative input of

  precision BG or VDDADC
  - Both comparators support rail-to-rail input mode, and the positive and negative sides support 2 GPIOs each
  - Dry suction protection is supported
  - Supports short circuit protection
- **High reliability**
  - ESD HBM 8KV
  - Latch-up ±200mA @25℃
- **96-bit chip Unique ID (UID)**
- **Packaging**
  - Die Form
  - SOP8/MSOP10/SOP14/SOP16/QFN16
- **Operating temperature range**
  - -40℃ ~ 85℃

2. Central processing unit

TX8C101x is fully compatible with the traditional 8051 microcontroller. All mnemonic and binary codes of instructions are compatible with the 8051. The TX8C101x processor uses some architecture optimizations to expand the SP, DPTR and other commonly used registers. Compared with the traditional 8051, the performance has been greatly improved.

The ALU in TX8C101x can realize various 8-bit operations with ACC (0xE0), B

(0xF0), PSW (0xD0) registers.

The ALU can perform typical operations as follows:

- Basic arithmetic operations: addition, subtraction, multiplication, division
- Other arithmetic operations: self-addition, self-subtraction, BCD adjustment, comparison
- Logical operations: AND, OR, XOR, negation, shift
- Boolean bit operations: set, zero, negate, jump by bit judgment, carry operation

2.1. Accumulator (ACC)

The ALU is an 8Bit wide arithmetic logic unit. All the mathematical and logical operations of the MCU are completed through it. It can add, subtract, shift data and logic operations; The ALU also controls the status bit (in PSW status register), which is used to represent the status of the operation result.

The ACC register is an 8Bit register where the ALU's result can be stored.

Addr = 0xE0 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	ACC	Accumulator register	RW	0x0

2.2. Register (B)

The B register is used when using multiplication and division instructions, the result of multiplication is 8bit higher, and the result of division is 8bit lower.

It can also be used as a general purpose register if multiplication and division instructions are not used.

Addr = 0xF0 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	B	B register	RW	0x0

2.3. Stack pointer register (SP)

The SP register points to the lower 8bit address of the stack, defaults to 0x07 after reset, and the value of this SP can be modified. The operations that affect the SP are: instruction PUSH, LCALL, ACALL, POP, RET, RETI, as well as the entry interrupt.

Addr = 0x81 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	SP	Stack pointer register	RW	0x7

2.4. Stack Pointer register (SPH)

The SPH register points to the stack high 8bit address, the significant bit 1bit, and the default bit 0x0 after reset, which in combination with the SP means that the area of the stack starts at 0x07 of the RAM address. This value can be modified, if the stack area is set to start at 0x0B, the values of SPH and SP will be set to 0x0 and 0x0A respectively after reset.

The operations that affect SPH are: instructions PUSH, LCALL, ACALL, POP, RET, RETI as well as the entry interrupt.

Addr = 0x9B (SFR)

Bit(s)	Name	Description	R/W	Reset

7:1	-	-	-	-
0	SPH	Stack pointer register high bit	RW	0x0

2.5. Data pointer register (DPTR0/DPTR1)

Data pointer is mainly used in MOVX, MOVC instructions, its role is to locate the address of RAM and ROM. There are two data pointer registers DPTR0 and DPTR1 inside the chip, which are selected through the DPSEL register.

Each set of Pointers includes two 8-bit registers: DPTR0={DPH0,DPL0} and DPTR1={DPH1,DPL1}.

Addr = 0x82 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	DPL0	DPTR0 data pointer register eight bits lower	RW	0x0

Addr = 0x83 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	DPH0	DPTR0 data pointer register eight bits high	RW	0x0

Addr = 0x84 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	DPL1	DPTR1 data pointer register eight bits lower	RW	0x0

Addr = 0x85 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	DPL1	The DPTR1 data pointer register is eight bits high	RW	0x0

2.6. Data Pointer Control Register (DPCFG)

Addr = 0x86 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:6	IA	**Interrupt start address and add way select bits** IA[0] 0x0: Interrupt address = (interrupt vector *8) +3 0x1: Interrupt address = (interrupt vector +1) *3 IA[1] 0x0: Interrupt start address is 0x300 0x1: Interrupt start address is 0x8000	RW	0x2
5	DPID0	**DPTR0 plus 1/ minus 1** 0x0: Add 1 to DPTR0 0x1: DPTR0 minus 1	RW	0x0
4	DPID1	**DPTR1 add 1/ subtract 1** 0x0: DPTR1 plus 1 0x1: DPTR1 minus 1	RW	0x0
3	DPAID	**DPTR0/DPTR1 self-add and self-subtract enable bits** 0x0: Off 0x1: Open	RW	0x0
2	DPTSL	**DPSEL automatically flips the enable bit** 0x0: Off 0x1: Open	RW	0x0
1	-	-	-	-
0	DPSEL	**Select the DPTR0 /DPTR1 bits** 0x0: DPTR0 is valid 0x1: DPTR1 is valid	RW	0x0

2.7. Program Status register (PSW)

Addr = 0xD0 (SFR)

Bit(s)	Name	Description	R/W	Reset
7	CY	**Carry flag bit** 0x0: No carry bit 0x1: Carry free	RW	0x0
6	AC	**Auxiliary carry flag bit** 0x0: No carry bit 0x1: Carry free	RW	0x0
5	F0	**Universal flag bit 0**	RW	0x0
4:3	RS1,RS0	**Register set select bit** 0x0: Register group 0 0x1: Register group 1 0x2: Register group 2 0x3: Register group 3	RW	0x0
2	OV	**Overflow flag bit** 0x0: No overflow for arithmetic or logical operations 0x1: An arithmetic or logical operation has an overflow	RW	0x0
1	F1	**Generic flag bit 1**	RW	0x0
0	P	**The parity flag bit** 0x0:1 is even in ACC 0x1:1 is odd in ACC	RW	0x0

2.8. Program counter (PC)

The program counter (PC) controls the instruction execution order in the program memory FLASH. It can address the whole range of FLASH. After getting the instruction code, the program counter (PC) will automatically add one to point to the address of the next instruction code. However, when executing jump, conditional jump, assignment to PCL, subprogram call, initialized reset, interrupt, interrupt return, subprogram return and other operations, the PC will load the address related to the instruction instead of the address of the next instruction.

When a conditional jump instruction is encountered and the jump condition is met, the next instruction read during the execution of the current instruction will be discarded, and a null instruction operation cycle will be inserted, and then the correct instruction can be obtained. Otherwise, the next instruction will be executed sequentially.

3. Memory

TX8C101x has three kinds of internal memory: IDATA, XDATA, program memory.

Program memory can only read not write, the size of the program memory is 4K bytes. XDATA is 1K bytes in size (with 512 bytes for XSFR) and 256 bytes for IDATA.

3.1. Program memory

The program pointer of TX8C101x is 16 bits, and the maximum addressable space of TX8C101X is 64K bytes. In fact, only 4K bytes of program storage space are realized.

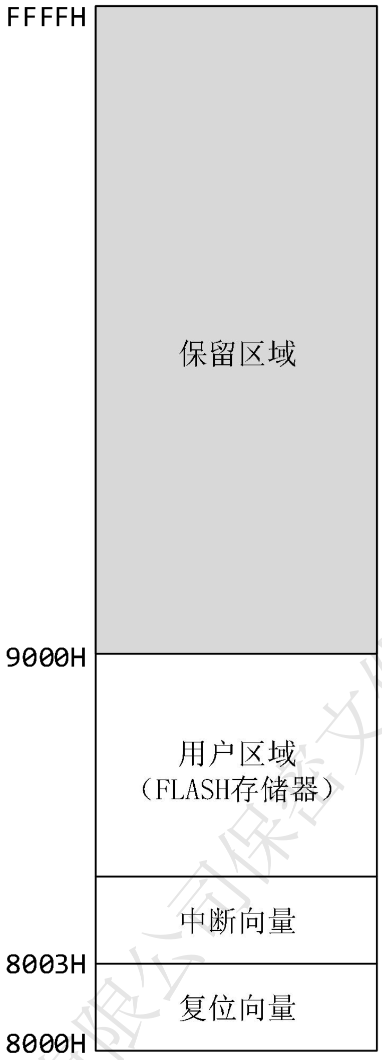


Figure 3-1 program storage space

After the reset, the MCU executes from 0x8000. Starting from 0x8003 is the interrupt vector table. When an interrupt occurs and the interrupt is enabled, the PC will jump to the corresponding interrupt vector position to execute.

3.2. XDATA

XDATA has 256 bytes and addresses 0x300 to 0x3FF and can be used for data storage.

3.3. IDATA

The size of the internal data memory space is 256 bytes.

The lower 128 bytes of the address space of the internal data memory can be accessed by byte, and the upper 128 bytes and the SFR share the same address space. Direct access to the upper 128 bytes will access the SFR space, and the upper 128 bytes can only be accessed by indirect addressing.

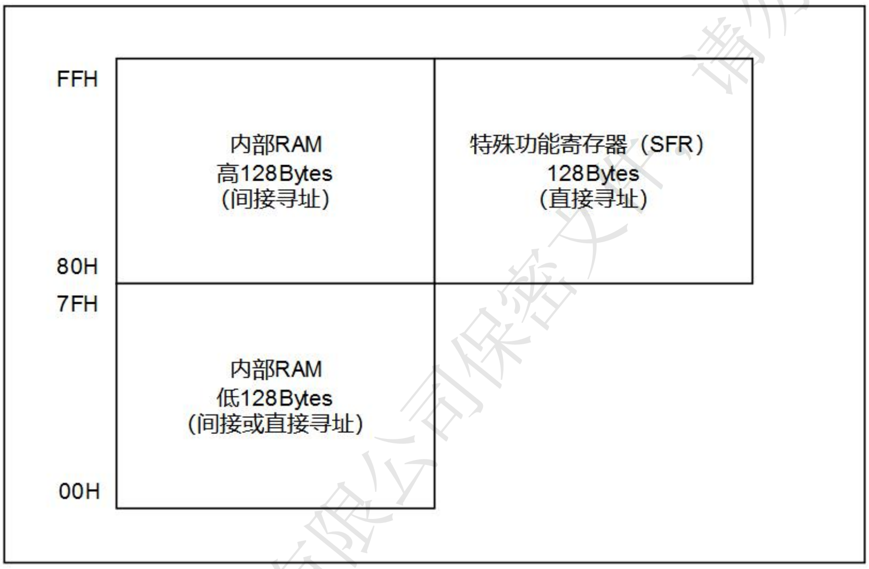
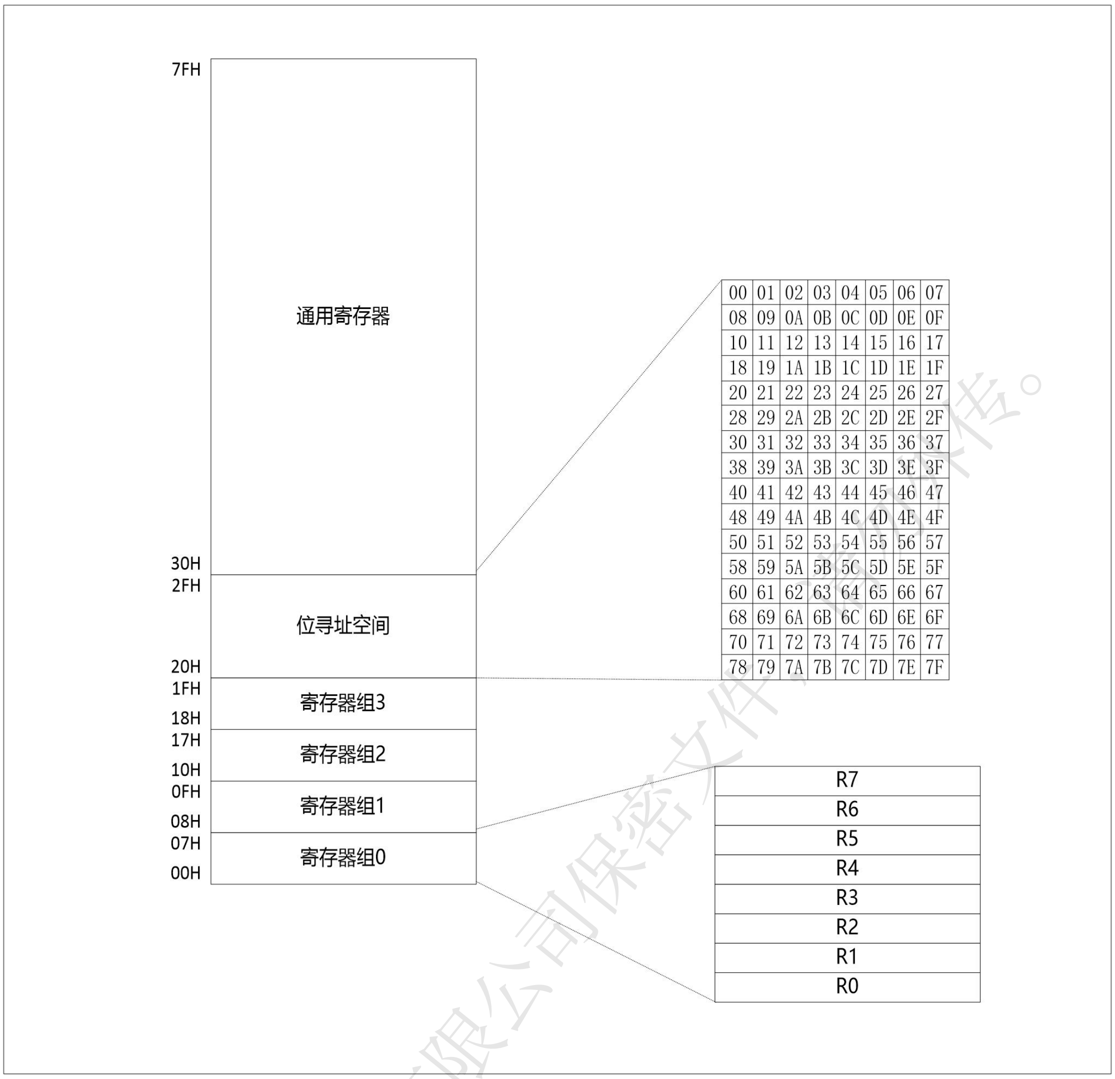


Figure 3.2 The data store



00	01	02	03	04	05	06	07
08	09	0A	0B	0C	0D	0E	0F
10	11	12	13	14	15	16	17
18	19	1A	1B	1C	1D	1E	1F
20	21	22	23	24	25	26	27
28	29	2A	2B	2C	2D	2E	2F
30	31	32	33	34	35	36	37
38	39	3A	3B	3C	3D	3E	3F
40	41	42	43	44	45	46	47
48	49	4A	4B	4C	4D	4E	4F
50	51	52	53	54	55	56	57
58	59	5A	5B	5C	5D	5E	5F
60	61	62	63	64	65	66	67
68	69	6A	6B	6C	6D	6E	6F
70	71	72	73	74	75	76	77
78	79	7A	7B	7C	7D	7E	7F

Figure 3.3 Internal low 128 bytes data space allocation

3.4. SFR space

	0H/8H	1H/9H	2H/AH	3H/BH	4H/CH	5H/DH	6H/EH	7H/FH
F8H	UART0_BAUD0	UART0_BAUD1	UART0_DATA	UART1_CON	UART1_STA	UART1_BAUD0	UART1_BAUD1	UART1_DATA
F0H	B	SPI0_CON	SPI0_BAUD	SPI0_DAT	SPI0_STA	STMR_ALLCON	UART0_CON	UART0_STA
E8H	STMR2_BRAKE	STMR2_DTR	STMR2_PCONRA	STMR2_PCONRB	STMR2_IE	STMR2_SR	ADK_CFG3	-
E0H	ACC	STMR2_CMPAH	STMR2_CMPBL	STMR2_CMPBH	STMR2_CR	STMR2_FCONR	STMR2_VPERR	STMR2_DTUA
D8H	STMR1_PCONRB	STMR1_IE	STMR1_SR	STMR2_CNTL	STMR2_CNTH	STMR2_PRL	STMR2_PRH	STMR2_CMPAL
D0H	PSW	STMR1_CR	STMR1_FCONR	STMR1_VPERR	STMR1_DTUA	STMR1_BRAKE	STMR1_DTR	STMR1_PCONRA
C8H	STMR1_CNTL	STMR1_CNTH	STMR1_PRL	STMR1_PRH	STMR1_CMPAL	STMR1_CMPAH	STMR1_CMPBL	STMR1_CMPBH
C0H	TMR2_CONL	TMR2_CONH	TMR2_CNTL	TMR2_CNTH	TMR2_PRL	TMR2_PRH	TMR2_PWML	TMR2_PWMH
B8H	IP0	IP1	LVD_CON0	LVD_CON1	LVD_CON2	LVD_CON3	LP_CON	SYS_PND
B0H	TMR1_CONL	TMR1_CONH	TMR1_CNTL	TMR1_CNTH	TMR1_PRL	TMR1_PRH	TMR1_PWML	TMR1_PWMH
A8H	IE0	IE1	FLASH_TRIM	WKUP_CON0	WKUP_PND	WDT_CON	WDT_KEY	-
A0H	FLASH_CON	FLASH_STA	FLASH_DATA	FLASH_TIM0	FLASH_TIM1	FLASH_CRCLEN	FLASH_PASSWORD	FLASH_ADDR
98H	ADK_CHS0	ADK_CHS1	ADK_CFG2	SPH	PCON1	ADK_CON	CRC_REG	CRC_FIFO
90H	P1	ADC_CFG0	ADC_CFG1	ADC_STA	ADC_DATAH0	ADC_DATAL0	ADC_DATAH1	ADC_DATAL1
88H	TMR0_CONL	TMR0_CONH	TMR0_CNTL	TMR0_CNTH	TMR0_PRL	TMR0_PRH	TMR0_PWML	TMR0_PWMH
80H	P0	SP	DPL0	DPH0	DPL1	DPH1	DPS	PCON0

4. System clock

4.1. Clock System Overview

The system chip has a 32MHz high speed and high precision RC oscillator, and an external 32.768KHz low speed crystal oscillator, as well as a 64KHz low speed RC oscillator integrated in the internal PMU.

4.2. Clock system main functions

The clock source of TX8C101x chip comes from three different clocks, which are the off-chip low speed crystal of 32.768KHz, the on-chip low speed RC of 64K and the on-chip high speed RC of 32M. As shown in Figure 4-1, the system clock can be selected from the above three clock sources by CLK_CON0[1:0], and the selected clock is the fastest clock (hereinafter referred to as sys_clk_pre). As shown in Figure 4-2, sys_clk_pre is divided by CLK_CON2[3:0], and the clock after frequency division is the system clock (hereinafter referred to as sys_clk). sys_clk is used by most peripherals and analog-to-digital mixing modules in the system. For example, UART, SPI, CRC16 and other peripherals all use sys_clk. As shown in Figure 4-2, the filter clock of GPIO port, the clock of Timer2 module and the low power detection are special modules that use sys_clk, the off-chip low speed crystal of 32.768KHz, the on-chip low speed RC of 64K, and the on-chip high speed RC of 32M after frequency division clock to select.

4.3. Block diagram of the clock system

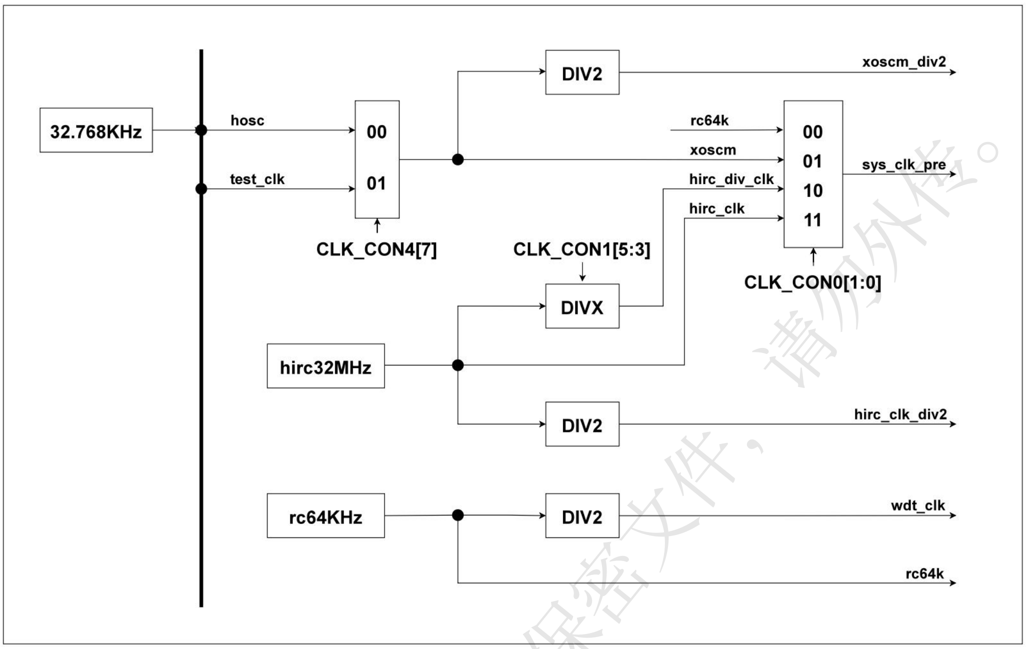


Figure 4-1 block diagram of the system clock

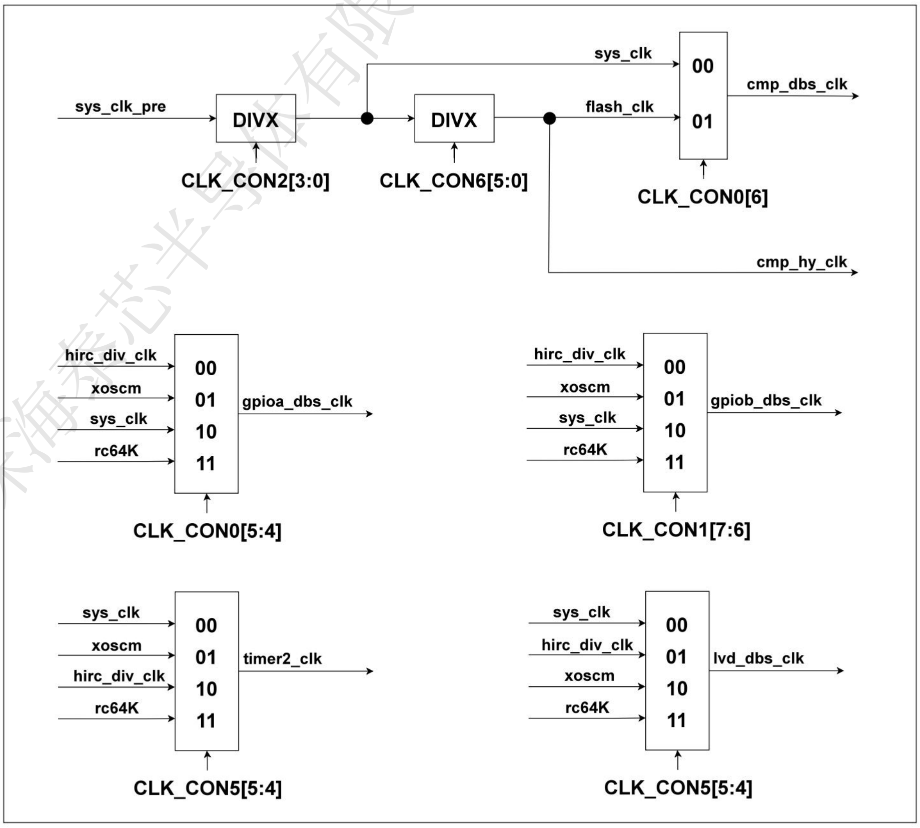


Figure 4.2 Block diagram of the module clock

4.4. System oscillator

4.4.1. Internal low speed RC oscillator

PMU is integrated with a 64KHz normally open low speed RC oscillator, when the chip is powered on, the system works at the 64KHz clock, complete the power-on reset process, until the system reset to release the program.

4.4.2. Internal high-speed RC oscillator

An on-chip high-speed RC oscillator is integrated inside the chip, which supports the output clock of up to 32MHz for the system. It is turned off by default. Users need to configure the system register AIP_CON0[7]=1 to turn on the high-speed RC oscillator in the program, and can also turn off the clock source by configuing AIP_CON0[7]=0. Note that the system needs to switch to the low speed RC oscillator before turning off the clock. The high speed RC oscillator can be calibrated by a calibration procedure during mass production to ensure its accuracy meets the requirements of the application scenario.

4.4.3. External low-speed crystal oscillator

A crystal oscillator starter circuit is integrated inside the chip, which can be connected to a low speed 32.768KHz passive crystal oscillator as the working clock source of the system. If it is off by default, the clock source can be turned on by configuing the system register AIP_CON1[3]=1; It can also be turned off by configuring AIP_CON1[3]=0.

5. Reset the system

5.1. Power-on reset

Power on chip POR reset.

5.2. Power down reset power down

BOR reset for chip power failure.

5.3. Watchdog reset

The chip has a watchdog module that runs independently of the system and is used to protect the system from resetting and restarting after an exception has occurred. The working clock of the watchdog module is a 64KHz low-speed RC 2 division clock, which is independent of the system clock and operates at 32KHz. The default configuration is to reset the system every 2 seconds. So the user program needs to feed the dog before the watchdog resets, so that it can be retimed. The user can configure the watchdog reset time interval from 8ms to 262s, and can choose the watchdog to generate an interrupt, no longer. You can choose between interrupt and reset.

5.3.1. Control the register list

Table 5-1 List of WDT registers

Address	Register Name	Description
0xAD (SFR)	WDT_CON	WDT_CON register
0xAE (SFR)	WDT_KEY	WDT_KEY register

5.3.2. Register details

5.3.2.1. WDT_CON

Addr = 0xAD (SFR)

Bit(s)	Name	Description	R/W	Reset
7	WAKEEN	**WDT wakeup feature enable bit** Write WDT_KEY=0xEE, set Write WDT_KEY=0x22, reset 0x0: Close 0x1: Open	RO	0x0
6	WDTPND	**The WDT counter is full of marker bits** Write WDT_KEY=0xAA to clear the marker bit 0x0: The counter is not full 0x1: The counter is full	RO	0x0
5	INTEN	**WDT interrupt function enable bit** Write WDT_KEY=0x5A, set Write WDT_KEY=0xA5, reset 0x1: Turn on interrupt function 0x0: Enable reset functionality	RO	0x0
4	WDTE	**WDT enable bit**	RW	0x1

		Write WDT_KEY=0xCC, set Write WDT_KEY=0xDD, reset 0x0: Disable watchdog function 0x1: Turn on the watchdog function		
3:0	PSR	**Pre-division coefficient** You must write WDT_KEY=0x55 before configuring this bit field each time 0x0: Regardless of frequency 0x1:2 frequency division 0x2:4 split frequency 0x3:8 frequency division 0x4:16 split frequency 0x5:32 split frequency 0x6:64 split frequency 0x7:128 split frequency 0x8:256 split frequency 0x9:512 split frequency 0xA: 1024 frequency division 0xB: 2048 split frequency 0xC: 4096 split frequency 0xD: 8192 split frequency 0xE: 16384 divider 0xF: 32768 split frequency Watchdog reset time =1/32K*256* division factor	RW	0x8

5.3.2.2. WDT_KEY

Addr = 0xAE (SFR)

Bit(s)	Name	Description	R/W	Reset

7:0	WDT_KEY	**Feed the dog data register** The software must write 0xAA at regular intervals to complete the dog-feeding operation, otherwise, the watchdog will generate a reset when the counter is zero When pending is 1, write 0xAA to clear pending 0x55: Indicates that wdt_psr is allowed to be accessed and set 0xDD: Turn off the watchdog 0xCC: Start watchdog work 0xAA: Feed the dog and clear wdt_pending 0xA5: Interrupt is closed 0x5A: Enable interrupt 0x22: Turn off wake up 0xEE: Turn wake up on	WO	0x0

5.4. Low power detection reset

PMU internal integrated low voltage detection and over current detection function circuit, used to detect PMU power supply part of the abnormal situation, and can detect the low voltage and over current and other abnormal situation through the interrupt report to the CPU for system exception handling program. In addition, the low voltage abnormal signal can generate a reset signal to reset the system, so as to avoid the circuit working abnormally under low voltage conditions and cause the user program to run. The threshold for low voltage detection can be set through the LVD control register. The LVD control register can be set to filter and dejitter the abnormal signal, to avoid the normal power supply voltage drop caused by the transient change of the system and the occurrence of accidental low power reset system.

5.4.1. Control register list

Table 5-2 List of LVD registers

Address	Register Name	Description
0xBA (SFR)	LVD_CON0	LVD_CON0 register
0xBB (SFR)	LVD_CON1	LVD_CON1 register
0xBC (SFR)	LVD_CON2	LVD_CON2 register
0xBD (SFR)	LVD_CON3	LVD_CON3 register

5.4.2. Register details

5.4.2.1. LVD_CON0

Addr = 0xBA (SFR)

Bit(s)	Name	Description	R/W	Reset
7	-	-	-	-
6	LVDOE	**LVD interrupt and reset functions output to the system enable bit** To use all LVD related functions, LVDOE must be set to bit 1 0x0: Off 0x1: Open	RW	0x1
5	LVDVDDRSTEN	**LVD VDD low voltage reset function enable bit** 0x0: Off 0x1: Open	RW	0x1
4	LVDVCCRSTEN	**LVD VCC low voltage reset function enable bit** 0x0: Off 0x1: Open	RW	0x1

3:2	PMULVD5SET	**VCC power supply voltage low voltage detection threshold set** 0x0:1.8/2.0V (power on threshold/power off threshold) 0x1:2.1/2.3V (power-up threshold/powerdown threshold) 0x2:2.4/2.6V (power-up threshold/powerdown threshold) 0x3:3.4/3.6V (power-up threshold/powerdown threshold)	RW	0x0
1	PMULVD15EN	**1.5V digital logic system operating voltage VDD low current detection function enable bit** 0x0: Off 0x1: Open	RW	0x1
0	PMULVD5EN	**VCC power supply VCC voltage low voltage detection function enable bit** 0x0: Off 0x1: Open	RW	0x1

5.4.2.2. LVD_CON1

Addr = 0xBB (SFR)

Bit(s)	Name	Description	R/W	Reset
7	-	-	-	-
6	VDDOCPND	**VDD overcurrent detection marker bit** Write 1 Clear the marker bit 0x0: VDD is not overflowing 0x1: VDD is overflowing	RW	0x0
5	LVDVDDPND	**VDD low current detection marker bit** Write 1 Clear the marker bit 0x0: VDD has no low power 0x1: VDD is low	RW	0x0

4	LVDVCCPND	**VCC low current detection marker bit** Write 1 Clear the marker bit 0x0: VCC has no low power 0x1: VCC low power	RW	0x0
3	LVDVCCSYNDIS	**LVD VCC low current detects synchronizer off bit** 0x0: Turn on the synchronizer 0x1: Turn off the synchronizer	RW	0x1
2	VDDOCBPSEN	**VDD overcurrent filter dejitter function off bit** 0x0: Turn on filter 0x1: Turn off the filter	RW	0x1
1	LVDVDDBPSEN	**VDD low electrical filter dejitter function off bit** 0x0: Turn on filter 0x1: Turn off the filter	RW	0x1
0	LVDVCCBPSEN	**VCC Low electrical filter dejitter function off bit** 0x0: Turn on filter 0x1: Turn off the filter	RW	0x1

5.4.2.3. LVD_CON2

Addr = 0xBC (SFR)

Bit(s)	Name	Description	R/W	Reset
7	-	-	-	-
6:0	DBSHLMT	**LVD low current and overcurrent anomaly detection filter high level filter clock cycle set number** **Note: The** LVD **filter clock can be selected through the system configuration register CLKCON5[3:2].** The user can choose the filter	RW	0x2

		function according to the use scenario. The delay time will be determined by the set filter clock period and the number of configured filter high level and low level filter cycles. The longer the filter clock period is set, the more the number of filter cycles will lead to a longer delay. The user can configure the delay reasonably according to the tolerance of the delay. In some delay sensitive application scenarios, the filter function can be turned off.		

5.4.2.4. LVD_CON3

Addr = 0xBD (SFR)

Bit(s)	Name	Description	R/W	Reset
7	-	-	-	-
6:0	DBSLLMT	**LVD low current and overcurrent anomaly detection filter low level filter clock cycle set number** **Note: The** LVD filter clock can be selected through the system configuration register CLKCON5[3:2]. The user can choose the filter function according to the use scenario. The delay time will be determined by the set filter clock period and the number of configured filter high level and low level filter cycles. The longer the filter clock period is set, the more the number of filter cycles will lead to a longer delay. The user can configure the delay reasonably according to the tolerance of the delay. In some delay sensitive application scenarios, the filter	RW	0x2

		function can be turned off.		

6. Low power management

The TX8C101x chip system supports three low power modes with different power levels, from high to low: Idle Mode, Stop Mode and Sleep Mode. The lowest power consumption is Sleep low power mode, in which the leakage of the whole chip can be about 3uA at normal temperature.

6.1. Idle Mode and wakeup

Enter Idle Mode by configuring the system register LP_CON[7]=1. In Idle mode only the CPU work clock is turned off and the CPU stops working. After waking up the Idle Mode, the interrupt service subroutine of the current wakeup Idle Mode will be entered and executed.

6.2. Stop Mode and wake up

Enter Stop Mode by configuring the system register LP_CON[1]= 1. In Stop mode, the system clock is turned off, and the CPU and most peripherals in the system clock domain stop working. Stop Mode is woken by selecting a variety of wake-up sources, including: all GPIOs, comparators, basic Timer2, watchdog, LVDVCC(Low voltage detection signal for the power supply). After Stop Mode wakes up, it will continue to run the following user program.

6.3. Sleep Mode and wake up

Enter the lowest power consumption Sleep Mode by configuring the system register LP_CON[0]=1. In Sleep mode, the system clock is turned off, the CPU and most of the system clock domain peripherals stop working, in addition to the PMU other analog modules should be turned off, but XOSC can choose to turn on/off. By selecting a variety of awaken to awaken the Sleep Mode source, wake up the source including: all GPIO level transformation sensei, basic Timer2 timer interrupt sensei, watchdog reset. After Sleep Mode is awakened, the following user program will continue to run through the LP_CON[6]=1 configured before entering Sleep mode. If LP_CON[6]=0, the system will be reset to run the user program again.

6.4. Low power consumption awakens the unit structure

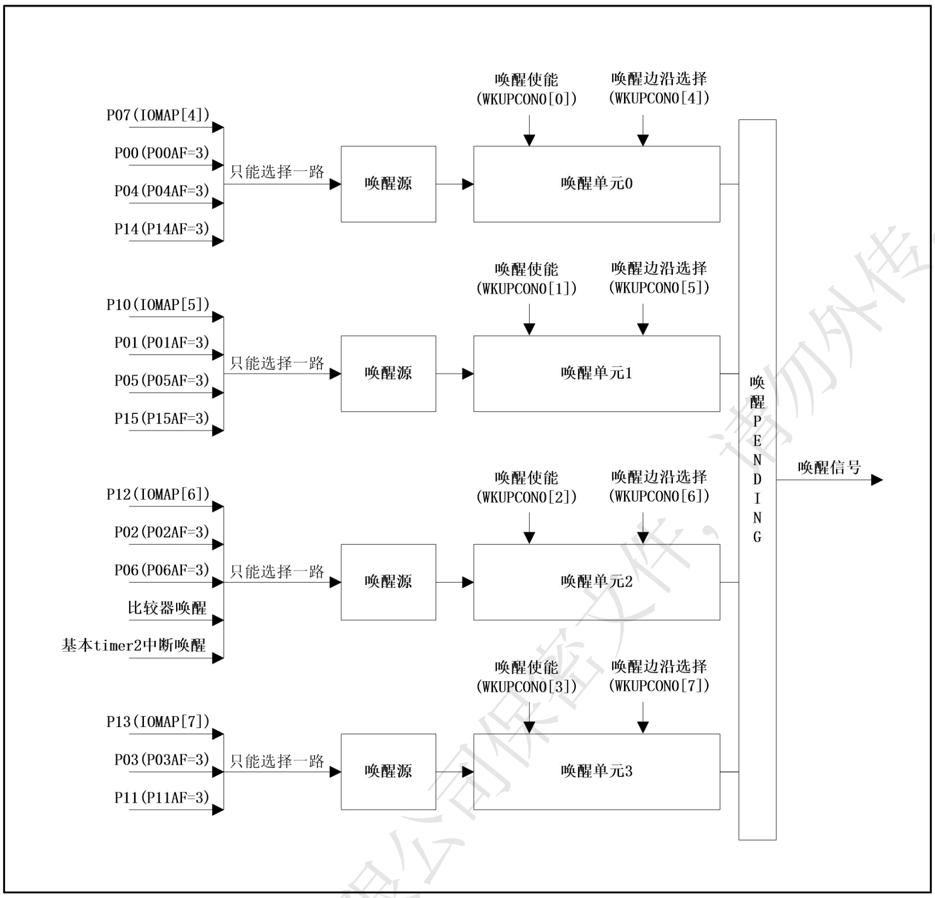


Figure 6-1 wake structure with low consumption

6.5. Register details

6.5.1. WAUP_CON0

Addr = 0 xab (SFR)

Bit(s)	Name	Description	R/W	Reset
7	WKUP3EDG	Low power mode wakeup channel 3 input trigger level setting 0x0: High level triggers wakeup 0x1: Low level triggers wakeup	RW	0x0
6	WKUP2EDG	Low power mode wakeup channel 2 input trigger level setting	RW	0x0

		0x0: High level triggers wakeup 0x1: Low level triggers wakeup		
5	WKUP1EDG	Low power mode wakeup channel 1 Input trigger level setting 0x0: High level triggers wakeup 0x1: Low level triggers wakeup	RW	0x0
4	WKUP0EDG	Low power mode wakeup channel 0 input trigger level setting 0x0: High level triggers wakeup 0x1: Low level triggers wakeup	RW	0x0
3	WKUP3EN	Low-power mode wakeup channel 3 function enable bit 0x0: Off 0x1: Open	RW	0x0
2	WKUP2EN	Low power mode wakeup channel 2 function enable bit 0x0: Off 0x1: Open	RW	0x0
1	WKUP1EN	Low power mode wakeup channel 1 function enable bit 0x0: Off 0x1: Open	RW	0x0
0	WKUP0EN	Low power mode wakeup channel 0 function enable bit 0x0: Off 0x1: Open	RW	0x0

6.5.2. WAUP_PND

Addr = 0xAC (SFR)

Bit(s)	Name	Description	R/W	Reset
7	WKUP3PCLR	Low power mode wakeup channel 3 clear pending	RW	0x0

		bits 0x0: No operation 0x1: Clear pending		
6	WKUP2PCLR	Low power mode wakeup channel 2 clear pending bits 0x0: No operation 0x1: Clear pending	RW	0x0
5	WKUP1PCLR	Low power mode wakeup channel 1 clear pending bit 0x0: No operation 0x1: Clear pending	RW	0x0
4	WKUP0PCLR	Low power mode wakeup channel 0 clear pending bits 0x0: No operation 0x1: Clear pending	RW	0x0
3	WKUP3PND	Low power mode wakeup channel 3 wakes up pending bit 0x0: No pending 0x1: There is pending	RW	0x0
2	WKUP2PND	Low power mode wakeup channel 2 wakes up pending bit 0x0: No pending 0x1: There is pending	RW	0x0
1	WKUP1PND	Low power mode wakeup channel 1 wakes up pending bit 0x0: No pending 0x1: There is pending	RW	0x0
0	WKUP0PND	Low power mode wakeup channel 0 wakes up pending bit 0x0: No pending 0x1: There is pending	RW	0x0

6.5.3. LP_CON

Addr = 0xBE (SFR)

Bit(s)	Name	Description	R/W	Reset
7	IDLE	**Idle Low power mode enabled** 0x0: On to enter Idle Low power mode 0x1: Off	RW	0x1
6	SLEEPGOEN	**Sleep low power mode after waking up to continue to run the subsequent program enable bit** 0x0: Sleep mode after waking up reset to run the program again 0x1: Continue to run subsequent programs after waking up in Sleep mode	RW	0x1
5	CPDIS	**User program protection function off bit** Neither CPU nor ISD is able to write registers 0x0: Open (default state) 0x1: Closed	RO	0x0
4	LPGLIRCEN	**Low power into low speed RC selection** In the low-power Sleep mode, the register can be configured to 1, and the RC64K low-speed clock can be automatically switched on after the system enters the Sleep mode. The purpose is to reduce the leakage power of Sleep. You can also choose not to turn off RC64K. **Note:** This feature can only be used when the system is woken up by GPIO in Sleep low power mode. If the timer is TIMER2, the clock can not be turned off, resulting in TIMER2 has no working clock and cannot wake up the system.	RW	0x0
3	CMCEDIS	**Program memory empty chip hardware automatic check function off bit** By default, when the chip is powered on, it	RW	0x0

		will automatically check whether there is a burning program in the program memory. If there is no burning program, it will hold the cpu and not run any programs. 0x0: Enable empty chip automatic check function 0x1: Disable the automatic check function for empty slices		
2	TMHCPU	**hold the CPU enable in test mode** **Note:** User programs do not write this register casually, it will cause the risk of abnormal system function!! 0x0: Do not hold CPU 0x1: hold CPU	RW	0x0
1	STOP	**Stop Low power mode enabled** 0x0: Closed 0x1: On, enter Stop low power mode	RW	0x0
0	SLEEP	**Sleep Low power mode enabled** 0x0: Off 0x1: On, enter Sleep low power mode	RW	0x0

7. System Control module

7.1. Feature Overview

The system control module is mainly used to manage and configure the system functions, including the system analog module, clock source, power supply system, clock management system, low power consumption and wake-up system and other system function configuration.

7.2. Register list

Table 7.1 List of system registers

Address	Register Name	Description
0x08 (XSFR)	SYS_CON0	SYS_CON0 register
0x09 (XSFR)	SYS_CON1	SYS_CON1 register
0x0A (XSFR)	SYS_CON2	SYS_CON2 register
0x0B (XSFR)	SYS_CON3	SYS_CON3 register
0x0C (XSFR)	SYS_CON4	SYS_CON4 register
0x0D (XSFR)	SYS_CON5	SYS_CON5 register
0x10 (XSFR)	CLK_CON0	CLK_CON0 register
0x11 (XSFR)	CLK_CON1	CLK_CON1 register
0x12 (XSFR)	CLK_CON2	CLK_CON2 register
0x13 (XSFR)	CLK_CON3	CLK_CON3 register
0x14 (XSFR)	CLK_CON4	CLK_CON4 register
0x15 (XSFR)	CLK_CON5	CLK_CON5 register
0x16 (XSFR)	CLK_CON6	CLK_CON6 register
0x28 (XSFR)	IO_MAP	IO_MAP register
0x2A (XSFR)	IO_MAP1	IO_MAP1 register
0x30 (XSFR)	AIP_CON0	AIP_CON0 register
0x31 (XSFR)	AIP_CON1	AIP_CON1 register
0x32 (XSFR)	AIP_CON2	AIP_CON2 register
0x33 (XSFR)	AIP_CON3	AIP_CON3 register
0x34 (XSFR)	AIP_CON4	AIP_CON4 register
0xAB (SFR)	WKUP_CON0	WKUP_CON0 register

0xAC (SFR)	WKUP_PND	WKUP_PND register
0xBE (SFR)	LP_CON	LP_CON register
0xBF (SFR)	SYS_PND	SYS_PND register

7.3. Register details

7.3.1. SYS_CON0

Addr = 0x08 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7	STMR1SOFTRST	**Advanced Timer1 Soft reset** 0x0: Soft reset 0x1: Soft reset release	RW	0x1
6	TMR2SOFTRST	**Basic Timer2 Soft reset** 0x0: Soft reset 0x1: Soft reset release	RW	0x1
5	TMR1SOFTRST	**Basic Timer1 soft reset** 0x0: Soft reset 0x1: Soft reset release	RW	0x1
4	TMR0SOFTRST	**Basic Timer0 soft reset** 0x0: Soft reset 0x1: Soft reset release	RW	0x1
3	-	-	-	-
2	SPI0SOFTRST	**SPI0 soft reset** 0x0: Soft reset 0x1: Soft reset release	RW	0x1
1	UART1SOFTRST	**UART1 Soft reset** 0x0: Soft reset	RW	0x1

		0x1: Soft reset release		
0	UART0SOFTRST	**UART0 Soft reset** 0x0: Soft reset 0x1: Soft reset release	RW	0x1

7.3.2. SYS_CON1

Addr = 0x09 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7:6	MEMDVS[1:0]	**On-chip SRAM dynamic voltage scaling value** **Note: The** user should never configure this register, otherwise it will cause the chip to behave indeterminately **！！！！**	RW	0x0
5	FASTRSTEN	**Quick reset wakeup Sleep Mode enabled** This feature is mainly configured to save reset time when set to wake sleep by reset in sleep low power mode 0x0: Off 0x1: Open	RW	0x0
4	GPIOSOFTRST	**GPIO module soft reset** 0x0: Soft reset 0x1: Soft reset release	RW	0x1
3	ADCSOFTRST	**ADC Soft Reset** 0x0: Soft reset 0x1: Soft reset release	RW	0x1
2	WDTSOFTRST	**Watchdog soft reset** 0x0: Soft reset 0x1: Soft reset release	RW	0x1
1	CRCSOFTRST	**CRC Soft reset** 0x0: Soft reset 0x1: Soft reset release	RW	0x1
0	STMR2SOFTRST	**Advanced Timer2 Soft reset**	RW	0x1

		0x0: Soft reset 0x1: Soft reset release		

7.3.3. SYS_CON2

Addr = 0x0A (XSFR)

Bit(s)	Name	Description	R/W	Reset
7	-	-	-	-
6	TMR2SWSYNCBPS	**Base Timer2 software sync bypass enable bit** 0x0: Off 0x1: Open	RW	0x1
5	IODBSSOFTRST	**IO debouce module soft reset** 0x0: Soft reset 0x1: Soft reset release	RW	0x1
4	-	-	-	-
3	TMR22IREN	**The basic Timer2 is combined with the advanced Timer2 to complete the infrared send function enable bit** The basic Timer2 is used as the carrier PWM, and the advanced Timer2 channel A is used as the modulated wave PWM 0x0: Off 0x1: Open	RW	0x0
2:1	-	-	-	-
0	LVDVCCWKEN	**LVDVCC wakes the enable bit** 0x0: Off 0x1: Open	RW	0x0

Note: Bits in the SYS_CON2 register are reserved for special test functions, user programs cannot write operations at will, which may bring system risks!

7.3.4. SYS_CON3

Addr = 0x0B (XSFR)

Bit(s)	Name	Description	R/W	Reset
7:6	EXTSLPCNT	**The time configuration for the low-power Sleep Mode process to exit the low-power LDO delay** Low power mode into the system clock switch to low speed must be set before the RC 64 KHZ, so the delay time = n * T64k 0 x0:1 cycle system 0x1:2 system cycles 0x2:3 system cycles 0x3:4 system cycles (recommended configuration)	RW	0x0
5:4	CMPUPCNT	**The low-power Sleep Mode process quits the delay time configuration that turns on the program memory power in the low-power process** Low power mode into the system clock switch to low speed must be set before the RC 64 KHZ, so the delay time = n * T64k 0x0:1 system cycles 0x1:2 system cycles 0x2:3 system cycles 0 x3: four system cycle (recommended)	RW	0x0
3:2	OPMLDOCNT	**The low-power Sleep Mode process turns on the main LDO delay time configuration** The system clock must be switched to a low speed RC of 64KHz before entering the low-power sleep mode, so the delay time =n*T64k 0x0:1 system cycles 0x1:2 system cycles 0x2:3 system cycles	RW	0x0

		0x3:4 system cycles (recommended configuration)		
1:0	CLSMLDOCNT	**The low-power Sleep Mode process turns off the main LDO delay time configuration** The system clock must be switched to a low speed RC of 64KHz before entering the low-power sleep mode, so the delay time =n*T64k 0x0:1 system cycles 0x1:2 system cycles 0x2:3 system cycles 0x3:4 system cycles (recommended configuration)	RW	0x0

7.3.5. SYS_CON4

Addr = 0x0C (XSFR)

Bit(s)	Name	Description	R/W	Reset
7	P07DBSEN	**P07 Input filter function enable bit** 0x0: Closed 0x1: Open	RW	0x0
6	P06DBSEN	**P06 Input filter function enable bit** 0x0: Off 0x1: Open	RW	0x0
5	P05DBSEN	**P05 Input filter function enable bit** 0x0: Off 0x1: Open	RW	0x0
4	P04DBSEN	**P04 Input filter function enable bit** 0x0: Off 0x1: Open	RW	0x0
3	P03DBSEN	**P03 Input filter function enable bit** 0 x0: closed 0x1: Open	RW	0x0

2	P02DBSEN	P02 Input filter function enable bit 0x0: Off 0x1: Open	RW	0x0
1	P01DBSEN	P01 Input filter function enable bit 0x0: Off 0x1: Open	RW	0x0
0	P00DBSEN	P00 Input filter function enable bit 0x0: Off 0x1: Open	RW	0x0

7.3.6. SYS_CON5

Addr = 0x0D (XSFR)

Bit(s)	Name	Description	R/W	Reset
7:6	-	-	-	-
5	P15DBSEN	P15 Input filter function enable bit 0x0: Off 0x1: Open	RW	0x0
4	P14DBSEN	P14 Input filter function enable bit 0x0: Off 0x1: Open	RW	0x0
3	P13DBSEN	P13 Input filter function enable bit 0x0: Off 0x1: Open	RW	0x0
2	P12DBSEN	P12 Input filter function enable bit 0x0: Off 0x1: Open	RW	0x0
1	P11DBSEN	P11 Input filter function enable bit 0x0: Closed 0x1: Open	RW	0x0
0	P10DBSEN	P10 Input filter function enable bit	RW	0x0

		0x0: Off 0x1: Open		

7.3.7. CLK_CON0

Addr = 0x10 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7	-	-	-	-
6	CMPDBSSEL	Comparator filters clock selection bits 0x0: Select eflash_clk 0 x1: select sys_clk	RW	0x0
5:4	PODBSCLKSEL	P0 filters the clock selection bits 0x0: Select hirc_div_clk 0x1: Select xoscm 0x2: Select sys_clk 0x3: Select rc64k	RW	0x0
3:2	CLKTOIOSEL	IO output clock source selection bit 0 x0: select sys_clk 0x1: Select hirc_div_clk 0x2: Select lirc 0x3: Select xoscm	RW	0x0
1:0	SYSCLKSEL	System clock selection bit 0x0: Select rc64k 0x1: Select xoscm 0x2: Select hirc_div_clk 0x3: Select hirc_clk	RW	0x0

7.3.8. CLK_CON1

Addr = 0x11 (XSFR)

Bit(s)	Name	Description	R/W	Reset

7:6	P1DBSCLKSEL	**P1 Filter clock selection bit** 0x0: Select hirc_div_clk 0x1: Select xoscm 0 x2: select sys_clk 0x3: Select rc64k	RW	0x0
5:3	HIRCCLKDIV	**High speed HRCOSC clock source division setting** Configuration ratio is n+1 clock 0x0: Regardless of frequency 0x1:2 frequency division 0x2:3 split frequency ... 0 x6:7 points frequency 0x7: Off	RW	0x6
2:0	CLKTOIODIV	**IO output clock source divider setting** Configure the ratio for n+1 clocks 0x0: Regardless of frequency 0x1:2 frequency division 0x2:3 split frequency ... 0 x6:7 points frequency 0x7: Off	RW	0x0

7.3.9. CLK_CON2

Addr = 0x12 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7:4	–	–	–	–
3:0	SYSCLKDIV	**System clock division setting** The configuration ratio is n+1 clocks 0x0: Regardless of frequency 0x1:2 frequency division 0x2:3 split frequency	RW	0x0

		... 0 xe: 15 points and frequency 0xF: Off		

7.3.10. CLK_CON3

Addr = 0x13 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7	TMR2CLKEN	**Basic Timer2 module clock enable bit** 0x0: Turn off the clock 0x1: Turn the clock on	RW	0x1
6	TMR1CLKEN	**Basic Timer1 module clock enable bit** 0x0: Turn off the clock 0x1: Turn the clock on	RW	0x1
5	TMR0CLKEN	**The basic Timer0 module clock enable bit** 0x0: Turn off the clock 0x1: Turn the clock on	RW	0x1
4	CRCCLKEN	**CRC module clock enable bit** 0x0: Turn off the clock 0x1: Turn the clock on	RW	0x1
3	-	-	-	-
2	SPI0CLKEN	**SPI0 module clock enable bit** 0x0: Turn off the clock 0x1: Turn the clock on	RW	0x1
1	UART1CLKEN	**UART1 module clock enable bit** 0x0: Turn off the clock 0x1: Turn the clock on	RW	0x1
0	UART0CLKEN	**UART0 module clock enable bit** 0x0: Turn off the clock 0x1: Turn the clock on	RW	0x1

7.3.11. CLK_CON4

Addr = 0x14 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7	TESTCLKEN	**Test the clock enable bit** 0x0: Turn off the clock 0x1: Turn the clock on	RW	0x0
6	-	-	-	-
5	RAMCLKEN	**On-chip SRAM clock enable bit** 0x0: Turn off the clock 0x1: Turn the clock on	RW	0x1
4	CMPDBSCLKEN	**Comparator debouce clock enable bit** 0x0: Turn off the clock 0x1: Turn the clock on	RW	0x1
3:2	-	-	-	-
1	STMR2CLKEN	**Advanced Timer2 module clock enable bit** 0x0: Turn off the clock 0x1: Turns on the clock	RW	0x1
0	STMR1CLKEN	**Advanced Timer1 module clock enable bit** 0x0: Turn off the clock 0x1: Turn the clock on	RW	0x1

7.3.12. CLK_CON5

Addr = 0x15 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7:6	-	-	-	-
5:4	TMR2CLKSEL	**Basic Timer2 module clock selection bit** 0x0: Select sys_clk 0x1: Select xoscm 0x2: Select hirc_div_clk 0x3: Select rc64k	RW	0x0

3:2	LVDDBSCLKSEL	**LVD module filters the clock source selection bit** 0x0: Select sys_clk 0x1: Select hirc_div_clk 0x2: Select xoscm 0x3: Select rc64k	RW	0x0
1	-	-	-	-
0	TCLKEN	**Test clock 1 enable bit** 0x0: Turn off the clock 0x1: Turn the clock on	RW	0x1

7.3.13. CLK_CON6

Addr = 0x16 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7:6	-	-	-	-
5:0	MCLKDIV	**Memory burn clock divider setting** Configuration ratio is n+1 clock 0x00: Regardless of frequency 0x01:2 frequency division 0x02:3 split frequency ... Frequency 0 x3e: 63 points 0x3F: Off	RW	0x0

7.3.14. AIP_CON0

Addr = 0x30 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7	HRCEN	**HRC clock enable signal** 0x0: Off 0x1: Open	RW	0x0

6:0	HRCSC	HRC clock frequency fine tuning (step=1%) 0x00: low ... 0 x7f: high	RW	0x48

7.3.15. AIP_CON1

Addr = 0x31 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7	-	-	-	-
6:5	HRCSCADD_B2T01	HRC clock frequency fine-tuned 2 bits higher (step=0.5%) 0x0: low ... 0 x3: high	RW	0x1
4	HRCTESTEN	HRC internal analog voltage test enable signal 0x0: Off 0x1: Open	RW	0x0
3	XOSCEN	Crystal's enable signal 0x0: Closed 0x1: Open	RW	0x0
2	XOSCHY	Output clock lag window selection 0x0: No hysteresis 0x1: There is +-10% hysteresis	RW	0x1
1	HRCSCADD_B0	1 bit lower for HRC clock frequency fine tuning (step=0.5%)	RW	0x0
0	HRCSR	HRC clock frequency coarsely tuned 0x0:16MHz 0x1:32MHz	RW	0x1

7.3.16. AIP_CON2

Addr = 0x32 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7	ADCCMPEN	**The CMP enable signal in the ADC** 0x0: Off 0x1: Open	RW	0x0
6	ADCCMPTRIMEN	**Comparator calibration function enable signal in ADC** 0x0: Off 0x1: Open	RW	0x0
5:0	ADCCMPTRIM	**Comparator calibration value in ADC** MSB: Symbol is, low 5 is numerical value	RW	0x0

7.3.17. AIP_CON3

Addr = 0x33 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7:6	ADCDUMMY	**Keep registers** <bit 6>VDDACMP test is able to switch <bit 7> reserved	RW	0x0
5:4	ADCTENSEL	**ADC test signal selection** 0 x0: VREFP 0 x1: reservations 0 x2: reservations 0x3: Turn off test signal	RW	0x3
3:2	ADCSELEXREF	**The ADC selection signal external reference** 0 x0: not to choose external reference 0 x1: select EXREF0 for reference voltage 0 x2: select EXREF1 for reference voltage 0 x3: illegal state **Note:** when selecting an external reference,	RW	0x0

		must choose to shut down the internal reference!		
1:0	ADCBIASSEL	**ADC biassel current selection** 0x0:0.75X 0x1:1X 0x2:1X 0x3:1.25X	RW	0x1

7.3.18. AIP_CON4

Addr = 0x34 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7	-	-	-	-
6	ADCPADFORCE	**Force the ATO to the ADC input port** For testing	RW	0x0
5	ADCSELTRIMIB	**ADC calibration current selection** 0x0:1X 0x1:2X	RW	0x0
4	ADCSELINREF	**The internal reference in the ADC enables the signal** 0x0: Off 0x1: Open To select the internal reference, you must first disconnect the external reference	RW	0x1
3	ADCCMPBSSEL	**Comparator bias current selection in ADC** 0x0:1X 0x1:1.25X	RW	0x0
2	ADCVCMSEL	**ADC common mode voltage selection** 0x0:0.375 fullscale 0x1:0.5 fullscale	RW	0x1
1	ADCVREFSEL	**Internal reference voltage selection signal in the ADC** 0x0:1.2V	RW	0x1

		0x1:2.4V		
0	ADCBIASEN	The ADC biasen current can make the signal 0x0: Off 0x1: Open	RW	0x0

7.3.19. IO_MAP

Addr = 0x28 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7	WKUPIN3	Low-power mode wakeup channel 3 input pin selection 0x0: P13 is not selected 0 x1: select P13	RW	0x0
6	WKUPIN2	Low power mode wakeup channel 2 input pin selection 0x0: P12 is not selected 0x1: Select P12	RW	0x0
5	WKUPIN1	Low power mode wakeup channel 1 input pin selection 0x0: P10 is not selected 0x1: Select P10	RW	0x0
4	WKUPIN0	Low power mode wakeup channel 0 input pin selection 0x0: P07 is not selected 0x1: Select P07	RW	0x0
3:2	CLKTOIOMAP	IO pin output clock source function to choose a foot 0x0: Turn off IO output clock function 0x1: Select the P04 output clock 0x2: Select P05 output clock 0x3: Select P12 output clock	RW	0x0
1:0	ISPMAP	Choose a burning/debugging pin feet 0 x0: close the burning/debugging	RW	0x1

		capabilities 0x1: Select P10 [HCK],P07 [HDA] 0x2: Select P00 [HCK],P01 [HDA] 0x3: Turn off the burn/debug function		

7.3.20. IO_MAP1

Addr = 0x2A (XSFR)

Bit(s)	Name	Description	R/W	Reset
7	MEMDVSE	**On-chip SRAM dynamic voltage scaling value enable bit** **Note: The** user should never configure this register, otherwise it will cause the chip to behave indeterminately **！！！！**	RW	0x0
6:5	MEMDVS[3:2]	**On-chip SRAM dynamic voltage scaling values** **Note: The** user should never configure this register, otherwise it will cause the chip to behave indeterminately **！！！！**	RW	0x0
4	RSTBEN	**RSTB reset function enable bit** 0x0: Disable RSTB reset function 0x1: Turn on RSTB reset function	RW	0x0
3:2	MPDNCNT	**The low-power Sleep Mode flow enters the delay time configuration to turn off the program memory power** The system clock must be switched to a low speed RC of 64KHz before entering the low-power sleep mode, so the delay time =n*T64k. 0x0:1 system cycles 0x1:2 system cycles 0x2:3 system cycles 0x3:4 system cycles (recommended configuration)	RW	0x0

1:0	RSTBSEL	**RSTB reset pin selection** 0x0: Do not select any pin foot as the reset function foot 0x1: Select P00 as the reset pin 0x2: Select P05 as the reset pin 0x3: Select P00, P05 as the reset pin, the actual P00 is effectively reset	RW	0x0

7.3.21. SYS_PND

Addr = 0xBF (SFR)

Bit(s)	Name	Description	R/W	Reset
7:6	-	-	-	-
5	SLPSTACLR	**Write 1 Clear the system sleep flag bit**	RW	0x0
4	SFTRSTCLR	**Write 1 Clear the system soft reset flag bit**	RW	0x0
3:2	-	-	-	-
1	SLPPND	**System sleep flag bit**	RW	0x0
0	SFTRSTPND	**System soft reset flag bit** Write 1 System soft reset.	RW	0x0

8. Interrupt the system

8.1. Interrupt Overview

The TX8C101x supports up to 14 interrupt sources. Each interrupt source has an independent interrupt enable signal, and its enable switch can be controlled by software. The interrupt controller has the following features:

- Receive interrupts from 14 interrupt sources
- Each interrupt has a fixed interrupt number, and the smaller the interrupt number, the higher the priority

- Interrupt latency: 5~8 machine cycles

8.2. Structure block diagram

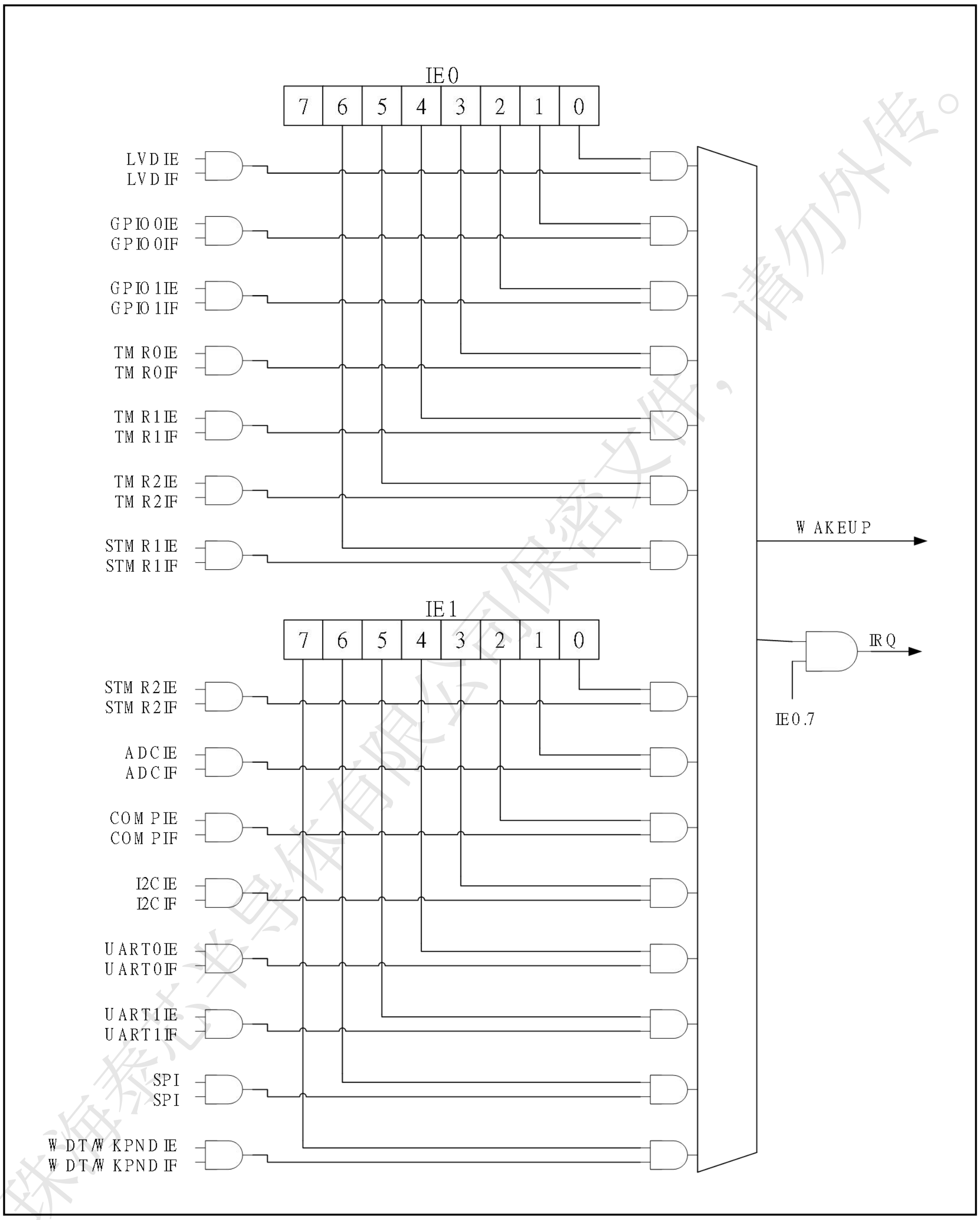


Figure 8-1 Interrupt structure diagram

8.3. Interrupt vector table

Interrupt controller supports 14 interrupt sources. When an interrupt occurs and the interrupt is enabled, jump to the corresponding vector address to execute the LCALL instruction to enter the interrupt service routine.

Table 8.1 Interrupt vector table

Interrupt sources	Interrupt level	Interrupt number	Interrupt address	Instructions
LVD	low	0	0003H	Low pressure test
P0	low	1	0006H	P0 port interrupt
P1	low	2	0009H	P1 port interruption
TMR0	low	3	000CH	TMR0 interrupted
TMR1	low	4	000FH	TMR1 interrupted
TMR2	low	5	0012H	TMR2 interrupted
STMR1	low	6	0015H	STMR1 interrupted
STMR2	low	7	0018H	STMR2 interrupted
ADC	low	8	001BH	ADC conversion completion interrupted
COMP	low	9	001EH	Simulate comparator interrupts
-	-	-	0021H	-
UART0	low	11	0024H	UART0 state interrupted
UART1	low	12	0027H	UART1 status interrupted
SPI0	low	13	002AH	SPI0 interrupt
WDT/WKUP_PND	low	14	002DH	Watchdog interrupts

8.4. Register list

Table 8-2.List of interrupt registers

Address	Register Name	Description
0xA8 (SFR)	IE0	Interrupt Enable 0 Register
0xA9 (SFR)	IE1	Interrupt Enable 1 Register
0xB8 (SFR)	IP0	Interrupt Priority 0 Register
0xB9 (SFR)	IP1	Interrupt Priority 1 Register

8.5. Register details

8.5.1. IE0

Addr = 0xA8 (SFR)

Bit(s)	Name	Description	R/W	Reset
7	EA	**Global Interrupt enabled** 0 x0: ban all interrupts 0x1: Allow all interrupts that are not blocked	RW	0x0
6	STMR1	**STMR1 Interrupt enabled** 0x0: Disallow STMR1 interrupt 0 x1: allow STMR1 not blocked interrupts	RW	0x0
5	TMR2	**TMR2 Interrupt enabled** 0x0: TMR2 interrupt is forbidden 0x1: TMR2 interrupts that are not blocked are allowed	RW	0x0
4	TMR1	**TMR1 Interrupt enabled** 0x0: TMR1 interrupt is forbidden	RW	0x0

		0x1: TMR1 interrupts that are not blocked are allowed		
3	TMR0	**TMR0 Interrupt enabled** 0x0: TMR0 interrupt is forbidden 0x1: TMR0 unblocked interrupts are allowed	RW	0x0
2	GPIO1	**GPIO1 interrupt enabled** 0x0: Disable GPIO1 interrupt 0x1: GPIO1 interrupts that are not blocked are allowed	RW	0x0
1	GPIO0	**GPIO0 Interrupt enabled** 0 x0: GPIO0 interrupt is prohibited 0x1: GPIO0 unblocked interrupts are allowed	RW	0x0
0	LVD	**LVD interrupt enabled** 0x0: LVD interrupt is forbidden 0x1: Interrupts where the LVD is not blocked are allowed	RW	0x0

8.5.2. IE1

Addr = 0xA9 (SFR)

Bit(s)	Name	Description	R/W	Reset
7	WDT/WKUP	**WDT/WKUP interrupt enabled** 0x0: Disable WDT/WKUP interrupt 0x1: Interrupts where WDT/WKUP is not blocked are allowed	RW	0x0
6	SPI	**SPI interrupt enabled** 0x0: Disable SPI interrupt 0x1: Allow interrupts where SPI is not blocked	RW	0x0
5	UART1	**UART1 Interrupt enabled** 0x0: Disable UART1 interrupt 0 x1: allow UART1 not blocked interrupts	RW	0x0
4	UART0	**UART0 interrupts enabled**	RW	0x0

		0 x0: UART0 interrupt is prohibited 0x1: Interrupts where UART0 is not blocked are allowed		
3	-	-	-	-
2	COMP	**Comparator interrupt enabled** 0x0: Comparator interrupt is forbidden 0x1: Interrupts where the comparator is not masked are allowed	RW	0x0
1	ADC	**ADC interrupt enabled** 0x0: ADC interrupt is forbidden 0x1: Interrupts where the ADC is not blocked are allowed	RW	0x0
0	STMR2	**STMR2 Interrupt enabled** 0x0: Disallow STMR2 interrupt 0x1: Interrupts that are not blocked by STMR2 are allowed	RW	0x0

8.5.3. IP0

Addr = 0xB8 (SFR)

Bit(s)	Name	Description	R/W	Reset
7	-	-	-	-
6	STMR1	**STMR1 Interrupt priority** 0x0: The priority level is 0 0x1: Priority level is 1	RW	0x0
5	TMR2	**TMR2 Interrupt priority** 0x0: The priority level is 0 0x1: Priority level is 1	RW	0x0
4	TMR1	**TMR1 Interrupt priority** 0x0: The priority level is 0 0x1: Priority level is 1	RW	0x0
3	TMR0	**TMR0 Interrupt priority**	RW	0x0

		0x0: The priority level is 0 0x1: Priority level is 1		
2	GPIO1	**GPIO1 Interrupt priority** 0x0: The priority level is 0 0x1: Priority level is 1	RW	0x0
1	GPIO0	**GPIO0 Interrupt priority** 0x0: The priority level is 0 0x1: Priority level is 1	RW	0x0
0	LVD	**LVD interrupt priority** 0x0: The priority level is 0 0x1: Priority level is 1	RW	0x0

8.5.4. IP1

Addr = 0xB9 (SFR)

Bit(s)	Name	Description	R/W	Reset
7	WDT/WKUP	**WDT/WKUP Interrupt priority** 0x0: The priority level is 0 0x1: The priority level is 1	RW	0x0
6	SPI	**SPI Interrupt priority** 0x0: The priority level is 0 0x1: Priority level is 1	RW	0x0
5	UART1	**UART1 Interrupt priority** 0x0: The priority level is 0 0x1: Priority level is 1	RW	0x0
4	UART0	**UART0 Interrupt priority** 0x0: The priority level is 0 0x1: Priority level is 1	RW	0x0
3	-	-	-	-
2	COMP	**Comparator interrupt priority** 0x0: The priority level is 0 0x1: Priority level is 1	RW	0x0

1	ADC	**ADC interrupt priority** 0x0: The priority level is 0 0x1: Priority level is 1	RW	0x0
0	STMR2	**STMR2 Interrupt priority** 0x0: The priority level is 0 0x1: Priority level is 1	RW	0x0

8.6. The interrupt priority and interrupt nesting

Chip regulations two interrupt priority level, which can realize interrupt nesting level 2. When an interrupt has responded, if there is a request from a high-level interrupt, the latter can interrupt the former to achieve interrupt nesting. Each interrupt can be configured with a priority level of 0-1. The interrupt priority level, the greater the interrupt priority is higher. The same level is not nested, using the principle of time first, first come first execution.

9. I/O port

9.1. Structure diagram

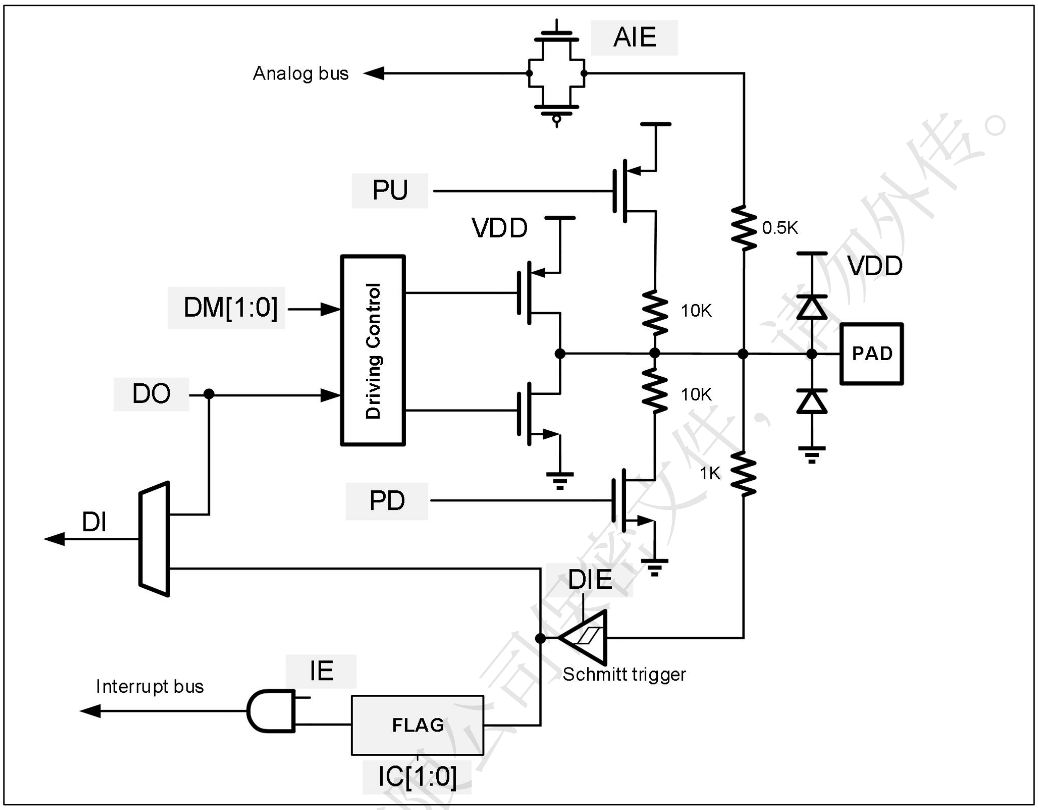


Figure 9-1 IO structure

9.2. The register list

Table 9-1 GPIO registers list

Address	Register Name	Description
0 x80 (SFR)	P0	P0 data register
0 x50 (XSFR)	P0_PU	P0 pull-up enable register
0 x51 (XSFR)	P0_PD	P0 pull-down enable register
0 x52 (XSFR)	P0_MD0	P0 mode register 0
0 x53 (XSFR)	P0_MD1	P0 mode register 1

0x54（XSFR）	P0_AF0	P0 alternal function config register 0
0 x55（XSFR）	P0_AF1	P0 alternal function config register 1
0x56（XSFR）	P0_TRG0	P0 interrupt trigger config register 0
0x57（XSFR）	P0_TRG1	P0 interrupt trigger config register 1
0 x58（XSFR）	P0_PND	P0 interrupt pending register
0x59（XSFR）	P0_IMK	P0 interrupt mask register
0x5A（XSFR）	P0_AIOEN	P0 analog function enable register
0x5B（XSFR）	P0_DRV	P0 driving current config register
0x5C（XSFR）	P0_OD	P0 open-drain enable register
0x90（SFR）	P1	P1 data register
0x60（XSFR）	P1_PU	P1 pull-up enable register
0x61（XSFR）	P1_PD	P1 pull-down enable register
0x62（XSFR）	P1_MD0	P1 mode register 0
0x63（XSFR）	P1_MD1	P1 mode register 1
0x64（XSFR）	P1_AF0	P1 alternal function config register 0
0x65（XSFR）	P1_AF1	P1 alternal function config register 1
0x66（XSFR）	P1_TRG0	P1 interrupt trigger config register 0
0x67（XSFR）	P1_TRG1	P1 interrupt trigger config register 1
0 x68（XSFR）	P1_PND	P1 interrupt pending register
0x69（XSFR）	P1_IMK	P1 interrupt mask register
0x6A（XSFR）	P1_AIOEN	P1 analog function enable register
0 x6b（XSFR）	P1_DRV	P1 driving current config register
0x6C（XSFR）	P1_OD	P1 open-drain enable register

9.3. Register details

9.3.1. P0

Addr = 0x80 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	P0	P0 data register	RW	0x00

9.3.2. P0_PU

Addr = 0x50 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7	P07PU	P07 pull-up resistor (30K) enable control bit 0x0: Off 0x1: Open	RW	0x0
6	P06PU	P06 Pull-up resistor (30K) enable control bit 0x0: Off 0x1: Open	RW	0x0
5	P05PU	P05 pull-up resistor (30K) enable control bit 0x0: Off 0x1: Open	RW	0x0
4	P04PU	P04 Pull-up resistor (30K) enables the control bit 0x0: Off 0x1: Open	RW	0x0
3	P03PU	P03 Pull-up resistor (30K) enable control bit 0x0: Off 0x1: Open	RW	0x0
2	P02PU	P02 Pull-up resistor (30K) enables the control bit 0x0: Off 0x1: Open	RW	0x0

1	P01PU	P01 Pull-up resistor (30K) enables the control bit 0x0: Off 0x1: Open	RW	0x0
0	P00PU	P00 pull-up resistor (30K) enables the control bit 0x0: Off 0x1: Open	RW	0x0

9.3.3. P0_PD

Addr = 0x51 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7	P07PD	P07 pull-down resistor (30K) enables control bit 0x0: Off 0x1: Open	RW	0x0
6	P06PD	P06 pull-down resistor (30K) enables control bit 0x0: Off 0x1: Open	RW	0x0
5	P05PD	P05 pull-down resistor (30K) enables control bit 0x0: Closed 0x1: Open	RW	0x0
4	P04PD	P04 Pull-down resistor (30K) enables the control bit 0x0: Off 0x1: Open	RW	0x0
3	P03PD	P03 Pull-down resistor (30K) enables control bit 0x0: Off 0x1: Open	RW	0x0

2	P02PD	P02 pull-down resistor (30K) enables the control bit 0x0: Off 0x1: Open	RW	0x0
1	P01PD	P01 pull-down resistor (30K) enables the control bit 0x0: Off 0x1: Open	RW	0x0
0	P00PD	P00 pull-down resistor (30K) enables control bit 0x0: Off 0x1: Open	RW	0x0

9.3.4. P0_MD0

Addr = 0x52 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7:6	P03MD	P03 mode configuration bit 0 x0: input mode 0 x1: output mode 0 x2: multi-function IO model 0x3: Analog IO mode	RW	0x0
5:4	P02MD	P02 pattern configuration 0 x0: input mode 0x1: Output mode 0 x2: multi-function IO model 0x3: Analog IO mode	RW	0x0
3:2	P01MD	P01 pattern configuration 0 x0: input mode 0 x1: output mode 0 x2: multi-function IO model 0 x3: simulate the IO model Note: if you burn the mouth as P01 requires	RW	0x0

		the corresponding will IO_MAP (1-0), namely ISPMAP is set to 0 x03, to switch to a GPIO function		
1:0	P00MD	**P00 pattern configuration** 0 x0: input mode 0 x1: output mode 0 x2: multi-function IO model 0x3: Simulated IO mode **Note:** If the burn port is P00, you need to correspondingly set IO_MAP[0:1], i.e., ISPMAP to 0x03 to switch to GPIO function	RW	0x0

9.3.5. P0_MD1

Addr = 0x53 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7:6	P07MD	**P07 Mode configuration bit** 0x0: Input mode 0x1: Output mode 0x2: Multi-function IO mode 0 x3: simulate the IO model **Note:** if you burn the mouth as P07 requires the corresponding will IO_MAP (1-0), namely ISPMAP is set to 0 x03, to switch to a GPIO function	RW	0x0
5:4	P06MD	**P06 pattern configuration** 0x0: Input mode 0x1: Output mode 0x2: Multi-function IO mode 0x3: Analog IO mode	RW	0x0
3:2	P05MD	**P05 mode configuration bit** 0x0: Input mode 0x1: Output mode	RW	0x0

		0x2: Multi-function IO mode 0 x3: simulate the IO model		
1:0	P04MD	**P04 pattern configuration** 0 x0: input mode 0 x1: output mode 0x2: Multi-function IO mode 0 x3: simulate the IO model	RW	0x0

9.3.6. P0_AF0

Addr = 0x54 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7:6	P03AF	**P03 multi-function mode, function configuration** 0x0: STMR1_CHA function 0 x1: SPI0_DIO function 0x2: STMR2_CHB function 0x3: PORT_WKUP_IN3 function	RW	0x0
5:4	P02AF	**P02 multifunctional mode, function configuration** 0 x0: STMR1_CHB function 0x1: SPI0_CLK function 0 x2: reservations 0 x3: PORT_WKUP_IN2 function	RW	0x0
3:2	P01AF	**P01 multifunctional mode, function configuration** 0 x0: TMR2_PWM function 0 x1: UART1_TX function 0 x2: reservations 0 x3: PORT_WKUP_IN1 function	RW	0x0
1:0	P00AF	**P00 In Multi-function mode, function configuration bit**	RW	0x0

		0x0: TMR2_PWM function 0x1: UART1_RX function 0x2: Keep 0x3: PORT_WKUP_IN0 function		

9.3.7. P0_AF1

Addr = 0x55 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7:6	P07AF	**P07 multifunctional mode, function configuration** 0x0: CMP0_DIG_OUT function 0x1: reserved 0x2: TMR2_PWM function 0x3: UART0_RX function	RW	0x0
5:4	P06AF	**P06 multifunctional mode, function configuration** 0x0: STMR1_CHB function 0x1: TMR0_PWM function 0x2: STMR2_CHA function 0x3: PORT_WKUP_IN2 function	RW	0x0
3:2	P05AF	**P05 In multi-function mode, function configuration bit** 0x0: STMR2_CHA function 0x1: TMR0_CAP function 0x2: Keep 0x3: PORT_WKUP_IN1 function	RW	0x0
1:0	P04AF	**P04 In Multi-function mode, function configuration bit** 0x0: STMR2_CHB function 0 x1: CMP0_DIG_OUT function 0x2: Keep 0 x3: PORT_WKUP_IN0 function	RW	0x0

9.3.8. P0_TRG0

Addr = 0 x56 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7:6	P03TRG	**P03 interrupt trigger source configuration** 0x0: Rising and falling edges are triggered 0x1: Falling edge triggered 0x2: Rising edge trigger 0 x3: along the trigger rising or falling	RW	0x0
5:4	P02TRG	**P02 Interrupt trigger source configuration bit** 0x0: Rising and falling edge trigger 0 x1: falling edge trigger 0 x2: rising along the trigger 0x3: Rising and falling edge trigger	RW	0x0
3:2	P01TRG	**P01 interrupt trigger source configuration** 0 x0: along the trigger rising or falling 0 x1: falling edge trigger 0 x2: rising along the trigger 0 x3: along the trigger rising or falling	RW	0x0
1:0	P00TRG	**P00 interrupt trigger source configuration** 0 x0: along the trigger rising or falling 0 x1: falling edge trigger 0 x2: rising along the trigger 0 x3: along the trigger rising or falling	RW	0x0

9.3.9. P0_TRG1

Addr = 0x57 (XSFR)

Bit(s)	Name	Description	R/W	Reset

7:6	P07TRG	**P07 Interrupt trigger source configuration bit** 0x0: Rising and falling edge trigger 0x1: Falling edge triggered 0x2: Rising edge trigger 0x3: Rising and falling edge trigger	RW	0x0
5:4	P06TRG	**P06 interrupt trigger source configuration** 0 x0: along the trigger rising or falling 0x1: Falling edge triggered 0x2: Rising edge trigger 0x3: Rising and falling edge trigger	RW	0x0
3:2	P05TRG	**P05 interrupt trigger source configuration** 0x0: Rising and falling edge triggered 0 x1: falling edge trigger 0x2: Rising edge trigger 0x3: Rising and falling edges are triggered	RW	0x0
1:0	P04TRG	**P04 Interrupt trigger source configuration bit** 0x0: Rising and falling edge trigger 0x1: Falling edge triggered 0x2: Rising edge trigger 0x3: Rising and falling edge trigger	RW	0x0

9.3.10. P0_PND

Addr = 0x58 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7	P07PND	**P07 Interrupt flag bit** 0x0: No interrupt was triggered 0x1: Triggered interrupt	RW	0x0
6	P06PND	**P06 Interrupt flag bit** 0x0: No interrupt was triggered	RW	0x0

		0x1: Triggered interrupt		
5	P05PND	**P05 Interrupt flag bit** 0x0: No interrupt was triggered 0 x1: raise the interrupt	RW	0x0
4	P04PND	**P04 Interrupt flag bit** 0x0: No triggered interrupt 0x1: Triggered interrupt	RW	0x0
3	P03PND	**P03 interrupt flag** 0x0: No triggered interrupt 0x1: Triggered interrupt	RW	0x0
2	P02PND	**P02 interrupt flag** 0 x0: not raise the interrupt 0x1: Triggered interrupt	RW	0x0
1	P01PND	**P01 interrupt flag** 0x0: No triggered interrupt 0x1: Triggered interrupt	RW	0x0
0	P00PND	**P00 interrupt flag** 0x0: No interrupt was triggered 0x1: Triggered interrupt	RW	0x0

Note: write CPU P0_PND operation, and the clear all the pending!!!!!!

9.3.11. P0_IMK

Addr = 0 x59 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7	P07IMK	**P07 Interrupt masking bit** 0x0: Turn off interrupt 0x1: Open interrupt	RW	0x0
6	P06IMK	**P06 Interrupt masking bit** 0x0: Turn off interrupt 0x1: Open interrupt	RW	0x0
5	P05IMK	**P05 Interrupt masking bit**	RW	0x0

		0x0: Turn off interrupt 0 x1: open the interrupt		
4	P04IMK	**P04 Interrupt masking bit** 0 x0: close the interrupt 0x1: Open interrupt	RW	0x0
3	P03IMK	**P03 Interrupt masking bit** 0x0: Turn off interrupt 0x1: Open interrupt	RW	0x0
2	P02IMK	**P02 Interrupt shielded bit** 0x0: Turn off interrupt 0 x1: open the interrupt	RW	0x0
1	P01IMK	**P01 Interrupt masking bit** 0x0: Interrupt is closed 0 x1: open the interrupt	RW	0x0
0	P00IMK	**P00 Interrupt masking bit** 0x0: Turn off interrupt 0 x1: open the interrupt	RW	0x0

9.3.12. P0_AIOEN

Addr = 0 x5a (XSFR)

Bit(s)	Name	Description	R/W	Reset
7	P07AIOEN	**P07 Emulated enable bit** 0x0: Off 0x1: Open	RW	0x0
6	P06AIOEN	**P06 Emulated enable bit** 0x0: Off 0x1: Open	RW	0x0
5	P05AIOEN	**P05 Emulated enable bit** 0x0: Off 0x1: Open	RW	0x0
4	P04AIOEN	**P04 Simulate the enable bit** 0x0: Off	RW	0x0

		0x1: Open		
3	P03AIOEN	**P03 Emulated enable bit** 0x0: Off 0x1: Open	RW	0x0
2	P02AIOEN	**P02 Simulate the enable bit** 0x0: Off 0x1: Open	RW	0x0
1	P01AIOEN	**P01 Simulates the enable bit** 0x0: Off 0x1: Open	RW	0x0
0	P00AIOEN	**P00 Emulated enable bit** 0x0: Off 0x1: Open	RW	0x0

9.3.13. P0_DRV

Addr = 0x5B (XSFR)

Bit(s)	Name	Description	R/W	Reset
7	P07DRV	**P07 Current drive capability configuration** 0x0:10mA 0x1:50mA	RW	0x0
6	P06DRV	**P06 Current drive capability configuration** 0x0:10mA 0x1:50mA	RW	0x0
5	P05DRV	**P05 Current drive capability configuration** 0x0:10mA 0x1:50mA	RW	0x0
4	P04DRV	**P04 Current drive capability configuration** 0 x0:10 ma 0 x1:50 ma	RW	0x0
3	P03DRV	**P03 Current drive capability configuration** 0x0:10mA 0x1:50mA	RW	0x0

2	P02DRV	P02 Current drive capability configuration 0x0:10mA 0x1:50mA	RW	0x0
1	P01DRV	P01 Current drive capability configuration 0x0:10mA 0x1:50mA	RW	0x0
0	P00DRV	P00 Current drive capability configuration 0x0:10mA 0x1:50mA	RW	0x0

9.3.14. P0_OD

Addr = 0x5C (XSFR)

Bit(s)	Name	Description	R/W	Reset
7	P07OD	P07 open-drain enable position 0x0: Off 0x1: Open	RW	0x0
6	P06OD	P06 open-drain enable position 0x0: Off 0x1: Open	RW	0x0
5	P05OD	P05 open-drain enable position 0x0: Off 0x1: Open	RW	0x0
4	P04OD	P04 open-drain enable bit 0x0: Off 0x1: Open	RW	0x0
3	P03OD	P03 open-drain enable position 0x0: Off 0x1: Open	RW	0x0
2	P02OD	P02 open-drain enable position 0x0: Off 0x1: Open	RW	0x0
1	P01OD	P01 open-drain enable position	RW	0x0

		0 x0: closed 0x1: Open		
0	P00OD	P00 open - drain can make a tapping 0x0: Off 0x1: Open	RW	0x0

9.3.15. P1

Addr = 0 x90 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	P1	P0 data register	RW	0x1

9.3.16. P1_PU

Addr = 0x60 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7:6	-	-	-	-
5	P15PU	P15 Pull-up resistor (30K) enable bit 0x0: Off 0x1: Open	RW	0x0
4	P14PU	P14 Pull-up resistor (30K) enable bit 0x0: Off 0x1: Open	RW	0x0
3	P13PU	P13 Pull-up resistor (30K) enable bit 0x0: Off 0x1: Open	RW	0x0
2	P12PU	P12 Pull-up resistor (30K) enable bit 0x0: Off 0x1: Open	RW	0x0
1	P11PU	P11 Pull-up resistor (30K) enable bit 0x0: Off 0x1: Open	RW	0x0

0	P10PU	P10 Pull-up resistor (30K) enable bit 0x0: Off 0x1: Open	RW	0x0

9.3.17.　P1_PD

Addr = 0x61 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7:6	-	-	-	-
5	P15PD	P15 pull-down resistor (30K) enable bit 0x0: Off 0x1: Open	RW	0x0
4	P14PD	P14 Pull-down resistor (30K) enable bit 0x0: Off 0x1: Open	RW	0x0
3	P13PD	P13 Pull-down resistor (30K) enable bit 0x0: Off 0x1: Open	RW	0x0
2	P12PD	P12 pull-down resistor (30K) enable bit 0x0: Off 0x1: Open	RW	0x0
1	P11PD	P11 pull-down resistor (30K) enable bit 0x0: Off 0x1: Open	RW	0x0
0	P10PD	P10 pull-down resistor (30K) enable bit 0x0: Off 0x1: Open	RW	0x0

9.3.18.　P1_MD0

Addr = 0x62 (XSFR)

Bit(s)	Name	Description	R/W	Reset

7:6	P13MD	**P13 mode configuration bit** 0x0: Input mode 0x1: Output mode 0x2: Multi-function IO mode 0x3: Analog IO mode	RW	0x0
5:4	P12MD	**P12 Mode configuration bit** 0x0: Input mode 0x1: Output mode 0x2: Multi-function IO mode 0x3: Analog IO mode	RW	0x0
3:2	P11_MD	**P11 mode configuration bits** 0x0: Input mode 0x1: Output mode 0x2: Multi-function IO mode 0x3: Analog IO mode	RW	0x0
1:0	P10_MD	**P10 mode configuration bit** 0x0: Input mode 0x1: Output mode 0x2: Multi-function IO mode 0x3: Analog IO mode **Note:** If the burn port is P10, you need to set IO_MAP[0:1], that is, ISPMAP to 0x03 to switch to GPIO function	RW	0x0

9.3.19. P1_MD1

Addr = 0x63 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7:4	-	-	-	-
3:2	P15MD	**P15 mode configuration bit** 0x0: Input mode 0x1: Output mode 0x2: Multi-function IO mode	RW	0x0

		0x3: Analog IO mode		
1:0	P14MD	**P14 Mode configuration bit** 0x0: Input mode 0x1: Output mode 0x2: Multi-function IO mode 0x3: Analog IO mode	RW	0x0

9.3.20. P1_AF0

Addr = 0x64 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7:6	P13AF	**P13 In Multi-function mode, function configuration bit** 0x0: STMR2_CHA function 0x1: STMR1_CHA function 0x2: TMR2_PWM function 0x3: UART0_TX function	RW	0x0
5:4	P12AF	**P12 In Multi-function mode, function configuration bit** 0x0: STMR2_CHB function 0x1: STMR1_CHB function 0x2: TMR1_PWM 0x3: UART0_RX function	RW	0x0
3:2	P11AF	**P11 In multi-function mode, function configuration bit** 0x0: CMP1_DIG_OUT function 0x1: TMR2_CAP function 0x2: Keep 0x3: PORT_WKUP_IN3 function	RW	0x0
1:0	P10AF	**P10 In Multi-function mode, function configuration bit** 0x0: reserved	RW	0x0

		0x1: reserved 0 x2: reserved 0x3: reserved		

9.3.21. P1_AF1

Addr = 0x65 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7:4	-	-	-	-
3:2	P15AF	**P15 in multi-function mode, function configuration bit** 0x0: CMP0_DIG_OUT function 0x1: TMR1_CAP function 0x2: Keep 0x3: PORT_WKUP_IN1 function	RW	0x0
1:0	P14AF	**P14 multifunctional mode, function configuration** 0 x0: CMP1_DIG_OUT function 0 x1: TMR0_CAP function 0x2: Keep 0x3: PORT_WKUP_IN0 function	RW	0x0

9.3.22. P1_TRG0

Addr = 0 x66 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7:6	P13TRG	**P13 interrupt trigger source configuration** 0 x0: along the trigger rising or falling 0x1: Falling edge triggered 0x2: Rising edge trigger 0x3: Rising and falling edge trigger	RW	0x0
5:4	P12TRG	**P12 interrupt trigger source configuration**	RW	0x0

		0 x0: along the trigger rising or falling 0x1: Falling edge triggered 0x2: Rising edge trigger 0x3: Rising and falling edge trigger		
3:2	P11TRG	**P11 interrupt trigger source configuration** 0 x0: along the trigger rising or falling 0x1: Falling edge triggered 0x2: Rising edge trigger 0x3: Rising and falling edge trigger	RW	0x0
1:0	P10TRG	**P10 interrupt trigger source configuration** 0 x0: along the trigger rising or falling 0x1: Falling edge triggered 0x2: Rising edge trigger 0x3: Rising and falling edges are triggered	RW	0x0

9.3.23. P1_TRG1

Addr = 0x67 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7:4	-	-	-	-
3:2	P15TRG	**P15 Interrupt trigger source configuration bit** 0x0: Rising and falling edge trigger 0x1: Falling edge triggered 0x2: Rising edge trigger 0x3: Rising and falling edge trigger	RW	0x0
1:0	P14TRG	**P14 Interrupt trigger source configuration bit** 0x0: Rising and falling edge trigger 0x1: Falling edge triggered 0x2: Rising edge trigger 0x3: Rising and falling edge trigger	RW	0x0

9.3.24. P1_PND

Addr = 0x68 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7:6	-	-	-	-
5	P15PND	**P15 Interrupt flag bit** 0x0: No triggered interrupt 0x1: Triggered interrupt	RW	0x0
4	P14PND	**P14 interrupt flag** 0 x0: not raise the interrupt 0x1: Triggered interrupt	RW	0x0
3	P13PND	**P13 interrupt flag** 0 x0: not raise the interrupt 0x1: Triggered interrupt	RW	0x0
2	P12PND	**P12 Interrupt flag bit** 0x0: No interrupt was triggered 0x1: Triggered interrupt	RW	0x0
1	P11PND	**P11 Interrupt flag bit** 0x0: No interrupt was triggered 0x1: Triggered interrupt	RW	0x0
0	P10PND	**P10 Interrupt flag bit** 0x0: No interrupt was triggered 0x1: Triggered interrupt	RW	0x0

Note: CPU write P1_PND operation, and clear all pending!!

9.3.25. P1_IMK

Addr = 0x69 (XSFR)

Bit(s)	Name	Description	R/W	Reset

7:6	-	-	-	-
5	P15IMK	**P15 Interrupt masking bit** 0x0: Turn off interrupt 0x1: Open interrupt	RW	0x0
4	P14IMK	**P14 Interrupt masking bit** 0x0: Turn off interrupt 0x1: Open interrupt	RW	0x0
3	P13IMK	**P13 Interrupt masking bit** 0x0: Turn off interrupt 0x1: Open interrupt	RW	0x0
2	P12IMK	**P12 Interrupt masking bit** 0x0: Turn off interrupt 0x1: Open interrupt	RW	0x0
1	P11IMK	**P11 interrupt mask bit** 0 x0: close the interrupt 0x1: Open interrupt	RW	0x0
0	P10IMK	**P10 interrupt mask bit** 0 x0: close the interrupt 0x1: Open interrupt	RW	0x0

9.3.26. P1_AIOEN

Addr = 0 x6a (XSFR)

Bit(s)	Name	Description	R/W	Reset
7:6	-	-	-	-
5	P15AIOEN	**P15 Simulates the enable bit** 0x0: Off 0x1: Open	RW	0x0
4	P14AIOEN	**P14 Simulate the enable bit** 0x0: Off 0x1: Open	RW	0x0
3	P13AIOEN	**P13 Simulate the enable bit** 0x0: Off	RW	0x0

		0x1: Open		
2	P12AIOEN	P12 Simulate the enable bit 0x0: Off 0x1: Open	RW	0x0
1	P11AIOEN	P11 Simulate the enable bit 0x0: Off 0x1: Open	RW	0x0
0	P10AIOEN	P10 Simulate the enable bit 0x0: Closed 0x1: Open	RW	0x0

9.3.27. P1_DRV

Addr = 0x6B (XSFR)

Bit(s)	Name	Description	R/W	Reset
7:6	-	-	-	-
5	P15DRV	P15 Current drive capability configuration 0x0:10mA 0x1:50mA	RW	0x0
4	P14DRV	P14 current drive capacity configuration 0x0:10mA 0 x1:50 ma	RW	0x0
3	P13DRV	P13 Current drive capability configuration 0x0:10mA 0x1:50mA	RW	0x0
2	P12DRV	P12 Current drive capability configuration 0x0:10mA 0x1:50mA	RW	0x0
1	P11DRV	P11 Current drive capability configuration 0x0:10mA 0x1:50mA	RW	0x0
0	P10DRV	P10 Current drive capability configuration 0x0:10mA	RW	0x0

		0x1:50mA		

9.3.28. P1_OD

Addr = 0x6C (XSFR)

Bit(s)	Name	Description	R/W	Reset
7:6	-	-	-	-
5	P15OD	**P15 open-drain enable bit** 0x0: Off 0x1: Open	RW	0x0
4	P14OD	**P14 open-drain enable position** 0x0: Closed 0x1: Open	RW	0x0
3	P13OD	**P13 open-drain enable bit** 0 x0: closed 0x1: Open	RW	0x0
2	P12OD	**P12 open - drain can make a tapping** 0 x0: closed 0x1: Open	RW	0x0
1	P11OD	**P11 open-drain enable bit** 0 x0: closed 0x1: Open	RW	0x0
0	P10OD	**P10 open-drain enable bit** 0x0: Off 0x1: Open	RW	0x0

10. SPI module

10.1. Feature Overview

The functional features of the SPI module are as follows:

- It supports two-wire mode
- Support master-slave half duplex transceiver
- Polar-phase programmable serial clock
- With MCU interrupt transmission end mark
- Main mode supports communication rates up to 8Mbps (F_{osc}=32MHz)

10.2. The module block diagram

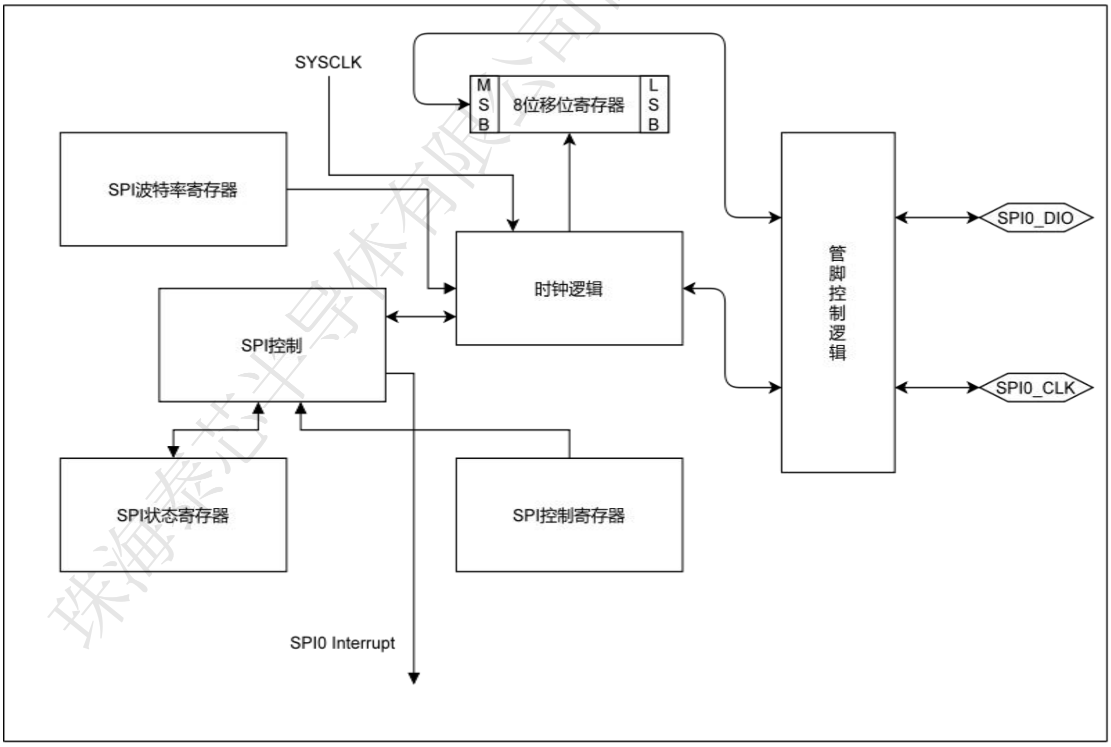


Figure 10.2.1 SPI module block diagram

10.3. List of registers

Table 10-1 List of SPI registers

Address	Register Name	Description
0xF1 (SFR)	SPI0_CON	SPI0 control register
0xF2 (SFR)	SPI0_BAUD	SPI0 baud rate register
0xF3 (SFR)	SPI0_DATA	SPI0 data register
0xF4 (SFR)	SPI0_STA	SPI0 status register

10.4. Register details

10.4.1. SPI0_CON

Addr = 0xF1 (SFR)

Bit(s)	Name	Description	R/W	Reset
7	-	-	-	-
6	SPISM	**Master and slave control bits** 0x0: Host 0x1: Slave machine	RW	0x0
5	SPIRXTX	**Spirxtx 4 Send receive control bits** 0x0: Send data 0x1: Data received	RW	0x0
4	-	-	-	-
3	SPIINTEN	**SPI interrupt enable bit** 0x0: Off 0x1: Open	RW	0x0
2	SPISMPSEl	**Sampling mode selection bit** 0x0: Start sampling from the second CLK clock edge 0x1: Start sampling from the first CLK clock	RW	0x0

		edge		
1	SPIIDST	**Clock line idle state select bit** 0x0: CLK idle is low 0x1: CLK idle is high	RW	0x0
0	SPIEN	**SPI enable bit** 0x0: Off 0x1: Open	RW	0x0

10.4.2. SPI0_BAUD

Addr = 0xF2 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	BAUD	**Baud rate control register** Calculate: BAUD rate = clk/(2*(BAUD+1))	WO	0x0

10.4.3. SPI0_DATA

Addr = 0xF3 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	DATA	**Data register** When enabled, write DATA to DATA to trigger send, read DATA to read the received data	RW	-

10.4.4. SPI0_STA

Addr = 0xF4 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:2	-	-	-	-
1	SPIINT	**SPI interrupt flag**	RO	0x0

		Write 1 to clear the zero		
0	SPIPEND	**SPI status flag bit** 0x0: Sending or receiving 0x1: Idle	RO	0x1

10.5. Instructions for Using the flow

1. Host TX: Configure SPI_ENABLE bit, SPI_RX_TX with 0 means send, write the DATA to be sent to data trigger send.
2. Host RX: Configure SPI_ENABLE bit, SPI_RX_TX with 1 means receive. Write any DATA to DATA trigger receive, receive complete (SPI_PENGING=1) read data read data.
3. Slave TX: Configure SPI_ENABLE bit, SPI_SM with 1 means slave mode, SPI_RX_TX with 0 means send. The DATA to be sent is written to DATA to trigger SPI to wait for the host clock.
4. Slave RX: Configure SPI_ENABLE bit, SPI_SM with 1 means slave mode, SPI_RX_TX with 1 means send. Writing any DATA to DATA triggers SPI to wait for the host clock.

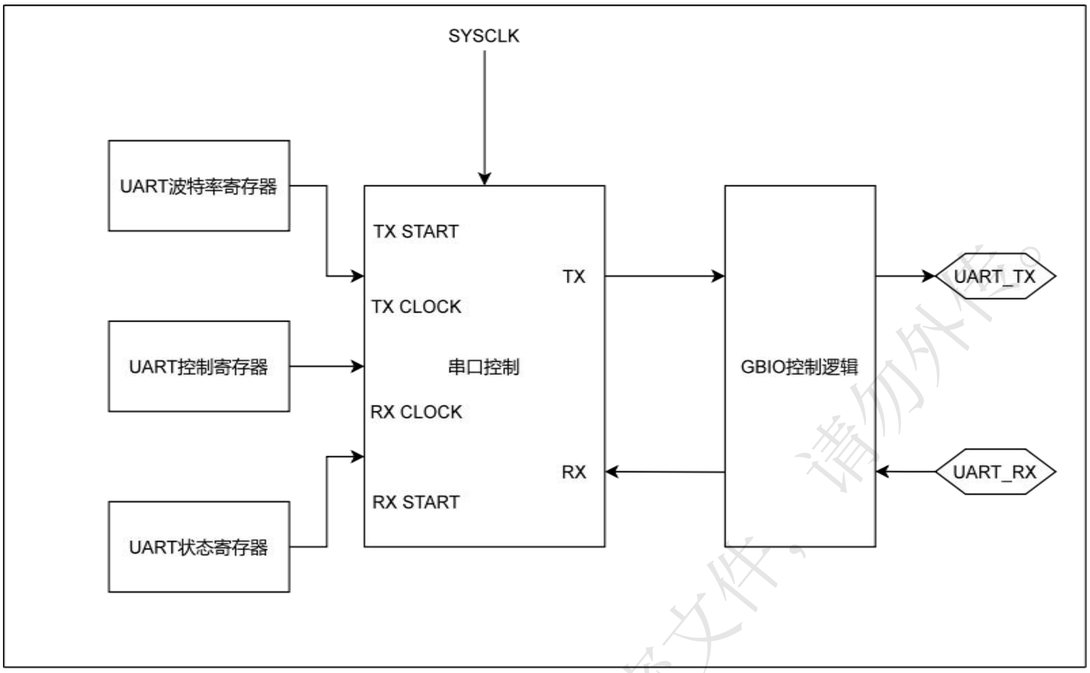
11. UART0/1 module

11.1. Feature Overview

UART module Features:

- Support half duplex
- Support for sending 9bit data
- Software parity is supported

11.2. Block diagram of modules



F

igure 11.2.1 Block diagram of UART module

11.3. List of registers

Table 11-1 List of UART registers

Address	Register Name	Description
0xF6 (SFR)	UART0_CON	UART0 control register
0xF7 (SFR)	UART0_STA	UART0 status register
0xF8 (SFR)	UART0_BAUD0	The low eight bits of the UART0 baud rate register
0xF9 (SFR)	UART0_BAUD1	The high eight bits of the UART0 baud rate register
0xFA (SFR)	UART0_DATA	UART0 data register
0xFB (SFR)	UART1_CON	UART1 control register
0xFC (SFR)	UART1_STA	UART1 status register

0xFD (SFR)	UART1_BAUD0	The low eight bits of the UART1 baud rate register
0xFE (SFR)	UART1_BAUD1	The high eight bits of the UART1 baud rate register
0xFF (SFR)	UART1_DATA	UART1 data register

11.4. Register details

11.4.1. UART0_CON

Addr = 0xF6 (SFR)

Bit(s)	Name	Description	R/W	Reset
7	STOPBIT	**The stopbit controls the bit** 0x0: Stop bits are not sent 0x1:1bit stop bit is sent	RW	0x0
6	NINTHBIT	**Ninthbit 2 Write the 9th bit of data that needs to be sent to this register**	RW	0x0
5	BIT9EN	**9bit data enable bits are sent** 0x0: Send 8bit data at a time 0x1: Send 9bit data at a time	RW	0x0
4	UARTEN	**UART enable bit** 0x0: Off 0x1: Open	RW	0x0
3	TXINV	**TX level takes the reverse control bit** 0x0: No negation 0x1: Take the inverse	RW	0x0
2	RXINV	**RX level takes the reverse control bit** 0x0: No negation 0x1: Take the negation	RW	0x0
1	TXRXSEL	**TX/RX select bit** 0x0: TX mode 0x1: RX mode	RW	0x0

0	UARTIE	**RX interrupt enable bit** 0x0: Off 0x1: Open	RW	0x0

11.4.2. UART0_STA

Addr = 0xF7 (SFR)

Bit(s)	Name	Description	R/W	Reset
7	RXBIT9	**9th bit received, read the bit to read out the 9th bit data**	RO	0x0
6	FERR	**The received parity bit data** The parity is carried out by the software (if BIT9_EN is not enabled when the received 8bit data is added to the parity bit, the parity bit is read at this bit)	RO	0x0
5	RXDONE	**RX status flag bit** This bit is 1 to indicate that the buff receives full data, write 1 to clear or read before starting the reception of the next frame of data	RO	0x0
4	TXDONE	**TX status flag bit** 0x0: Data is being sent 0x1: Idle	RO	0x0
3	UARTINT	**Interrupt flag bit** A high level indicates an interrupt is requested, write 1 to clear the interrupt	RO	0x0
2:0	-	-	-	-

11.4.3. UART0_BAUD0

Addr = 0xF8 (SFR)

Bit(s)	Name	Description	R/W	Reset

7:0	UARTBAUDL	**Baud rate register low 8Bytes** UART baud rate register, calculated by sysclk/(baud+1)	WO	0x0

11.4.4. UART0_BAUD1

Addr = 0xF9 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	UARTBAUDH	**Baud rate register high 8Bytes** UART baud rate register, calculated by sysclk/(baud+1)	WO	0x0

11.4.5. UART0_DATA

Addr = 0xFA (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	DATA	**Data register** Writing data to this register after enabling triggers the sending of the data and reading the register to get the received data	RW	0x0

11.4.6. UART1_CON

Addr = 0xFB (SFR)

Bit(s)	Name	Description	R/W	Reset
7	STOPBIT	**The stopbit controls the bit** 0x0: Stop bits are not sent 0x1:1bit stop bit is sent	RW	0x0
6	NINTHBIT	**Ninthbit 2 Write the 9th bit of data that needs to be sent to this register**	RW	0x0
5	BIT9EN	**Send 9bit data enable bit**	RW	0x0

		0x0: Send 8bit data at a time 0x1:9bit data sent at a time		
4	UARTEN	**UART enable bit** 0x0: Off 0x1: Open	RW	0x0
3	TXINV	**TX level takes the reverse control bit** 0x0: No negation 0x1: Take the negation	RW	0x0
2	RXINV	**RX level takes the reverse control bit** 0x0: No negation 0x1: Take the negation	RW	0x0
1	TXRXSEL	**TX/RX select bit** 0x0: Off 0x1: Open	RW	0x0
0	UARTIE	**RX interrupt enable bit** 0x0: Off 0x1: Open	RW	0x0

11.4.7. UART1_STA

Addr = 0xFC (SFR)

Bit(s)	Name	Description	R/W	Reset
7	RXBIT9	**9th bit received, read the bit to read out the 9th bit data**	RO	0x0
6	FERR	**The received parity bit data** The parity is carried out by the software (if BIT9_EN is not enabled when the received 8bit data is added to the parity bit, the parity bit is read at this bit)	RO	0x0
5	RXDONE	**RX status flag bit** This bit is 1 to indicate that the buff receives full data, write 1 to clear or read before starting the reception of the next	RO	0x0

		frame of data		
4	TXDONE	**TX status flag bit** 0x0: Data is being sent 0x1: Idle	RO	0x0
3	UARTINT	**Interrupt flag bit** A high level indicates an interrupt is requested, write 1 to clear the interrupt	RO	0x0
2:0	-	-	-	-

11.4.8. UART1_BAUD0

Addr = 0xFD (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	UARTBAUDL	**Baud rate register low 8Bytes** UART baud rate register, calculated by sysclk/(baud+1)	WO	0x0

11.4.9. UART1_BAUD1

Addr = 0xFE (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	UARTBAUDH	**Baud rate register high 8Bytes** UART baud rate register, calculated by sysclk/(baud+1)	WO	0x0

11.4.10. UART1_DATA

Addr = 0xFF (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	DATA	**Data register** Writing data to this register after enabling	RW	0x0

		triggers the sending of the data and reading the register to get the received data		

11.5. Instructions for using the process

Send data:

Can make module (UART0 - > CON | = BIT (4)), will need to send the DATA to the DATA which start sending (UART0 - > DATA = x). If you need to send 9 bit DATA can make BIT9_EN and 9 first bit DATA to NINTH_BIT will begin before 8 bits of DATA in the DATA sent again.

Receive data:

Just can make the module began testing start BIT, when receiving a full frame data RX_DONE buff full, can buy 1 said the received data can be read at this time, it will be necessary RX_DONE write 1 reset will receive next frame data (UART0 - > = BIT STA (5)).

12. Basic Timer 0/1 module

12.1. Functions overview

Basic Timer0/1 is the basis of the two 8 bit timer function, support for multiple counting clock source selection, support the timer mode, counter mode, capture mode, working mode and PWM mode, etc. Support Timer0 and Timer1 cascade of 16 bit timer working mode.

12.2. The module block diagram

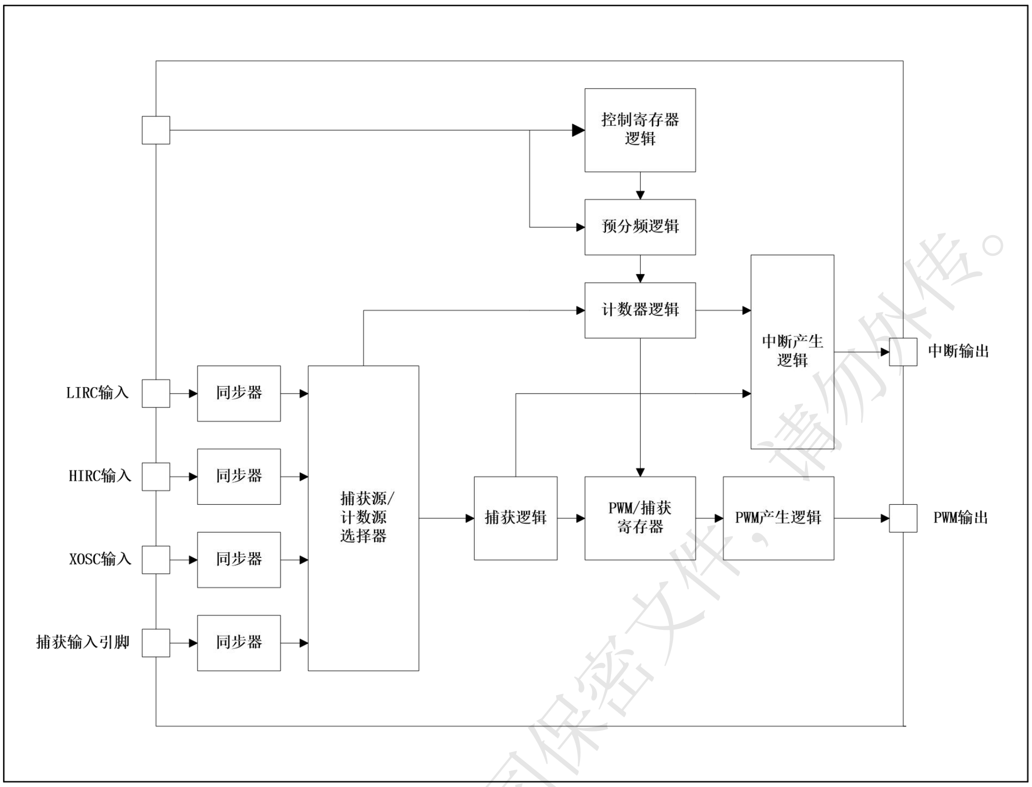


Figure 12.1 Block diagram of the Timer0/1 module

12.3. Register list

Table 12-1 List of Timer0/1 registers

Address	Register Name	Description
0x88 (SFR)	TMR0_CONL	TIMER0 control low 8bit register
0x89 (SFR)	TMR0_CONH	TIMER0 control high 8bit register
0x8A (SFR)	TMR0_CNTL	TIMER0 counter low 8bit register
0x8C (SFR)	TMR0_PRL	TIMER0 period low 8bit register
0x8E (SFR)	TMR0_PWML	TIMER0 PWM low 8bit register
0xB0 (SFR)	TMR1_CONL	TIMER1 control low 8bit register

0xB1 (SFR)	TMR1_CONH	TIMER1 control high 8bit register
0xB2 (SFR)	TMR1_CNTL	TIMER1 counter low 8bit register
0xB4 (SFR)	TMR1_PRL	TIMER1 period low 8bit register
0xB6 (SFR)	TMR1_PWML	TIMER1 PWM low 8bit register

12.4. Register details

12.4.1. TMR0_CONL

Addr = 0x88 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:5	PSC	**TIMER0 pre-frequency configuration** 0x0: No frequency division 0x1:2 frequency division 0x2:4 split frequency 0x3:8 split frequency 0x4:16 split frequency 0x5:32 split frequency 0x6:64 split frequency 0x7:128 split frequency	RW	0x0
4:2	INCSRC	**TIMER0 Count source Select configuration** 0x0: T0 rising edge 0x1: T0 falling edge 0x2: hirc_clk_div2 edge(rising & falling) 0x3: rc64k_div2 edge(rising & falling) 0x4: xoscm_div2 edge(rising & falling) 0x5: Timer1 ov 0x6: sys_clk 0x7: sys_clk	RW	0x0
1:0	MODE	**TIMER0 working mode configuration** 0x0: Off	RW	0x0

		0x1: COUNTER MODE 0x2: PWM MODE 0x3: CAPTURE MODE		

12.4.2. TMR0_CONH

Addr = 0x89 (SFR)

Bit(s)	Name	Description	R/W	Reset
7	TMRPND	**TIMER0 counts pending bits (write 1 to clear pending)** 0x0: No pending bits are counted 0x1: There is a count pending	RW	0x0
6	CAPPND	**TIMER0 captures pending bits (write 1 clear pending)** 0x0: No pending is captured 0x1: There is a captured pending	RW	0x0
5	TMRIE	**TIMER0 counts the interrupt enable bit** 0x0: Off 0x1: Open	RW	0x0
4	CAPIE	**TIMER0 captures the interrupt enable bit** 0x0: Off 0x1: Open	RW	0x0
3:2	INCSRC	**TIMER0 Capture source selection configuration** Pin 0x0: T0 serves as the capture source 0 x1: T0 pin as a source of capture 0x2: The digital output of comparator 0 serves as the capture source 0x3: The digital output of comparator 1 serves as the capture source	RW	0x0
1:0	CAPEDG	**TIMER0 captures the edge trigger setting of the T0 pin** 0x0: T0 rising edge triggers capture	RW	0x0

		0x1: T0 falling edge triggers capture 0x2: T0 double edge trigger capture 0x3: T0 double edge trigger capture		

12.4.3. TMR0_CNTL

Addr = 0x8A (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	CNT	**TIMER0 counter initial value**	RW	0x0

12.4.4. TMR0_PRL

Addr = 0x8C (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	PRD	**TIMER0 Count cycle set values**	RW	0xFF

12.4.5. TMR0_PWML

Addr = 0x8E (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	PWM	**TIMER0 duty cycle set value** In PWM working mode is, this value is the duty cycle setting value of the set bit PWM; In the capture mode, the value lock of the counter grasped after the capture source is captured is stored in the PWM register.	RW	0x0

12.4.6. TMR1_CONL

Addr = 0xB0 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:5	PSC	**TIMER1 pre-split configuration** 0x0: No frequency division 0x1:2 frequency division 0x2:4 split frequency 0x3:8 split frequency 0x4:16 split frequency 0x5:32 frequency division 0x6:64 split frequency 0x7:128 split frequency	RW	0x0
4:2	INCSRC	**TIMER1 Count source Select configuration** 0x0: T0 rising edge 0x1: T0 falling edge 0x2: hirc_clk_div2 edge(rising & falling) 0x3: rc64k_div2 edge(rising & falling) 0x4: xoscm_div2 edge(rising & falling) 0x5: Timer1 over 0x6: sys_clk 0x7: sys_clk	RW	0x0
1:0	MODE	**TIMER1 work mode configuration** 0x0: Off 0x1: COUNTER MODE 0x2: PWM MODE 0x3: CAPTURE MODE	RW	0x0

12.4.7. TMR1_CONH

Addr = 0xB1 (SFR)

Bit(s)	Name	Description	R/W	Reset

7	TMRPND	**TIMER1 counts pending bits (write 1 clear pending)** 0x0: No pending bits are counted 0x1: There is a count pending	RW	0x0
6	CAPPND	**TIMER1 captures pending bits (write 1 clear pending)** 0x0: No pending is captured 0x1: There is a captured pending	RW	0x0
5	TMRIE	**TIMER1 counts the interrupt enable bit** 0x0: Off 0x1: Open	RW	0x0
4	CAPIE	**TIMER1 captures the interrupt enable bit** 0x0: Off 0x1: Open	RW	0x0
3:2	INCSRC	**TIMER1 captures the source selection configuration** 0x0: T1 pin as the capture source 0x1: T1 pin serves as the capture source 0x2: The digital output of comparator 0 serves as the capture source 0x3: Comparator 1's digital output serves as the capture source	RW	0x0
1:0	CAPEDG	**TIMER1 captures the edge trigger setting of pin T0** 0x0: T1 rising edge triggers capture 0x1: T1 falling edge triggers capture 0x2: T1 double edge trigger capture 0x3: T1 bilateral edge trigger capture	RW	0x0

12.4.8. TMR1_CNTL

Addr = 0xB2 (SFR)

Bit(s)	Name	Description	R/W	Reset

7:0	CNT	**TIMER1 counter initial value**	RW	0x0

12.4.9. TMR1_PRL

Addr = 0xB4 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	PRD	**TIMER1 Count cycle setting values**	RW	0xFF

12.4.10. TMR1_PWML

Addr = 0xB6 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	PWM	**TIMER1 duty cycle setting value** Works in PWM mode is that the value is set a PWM duty ratio setting values; Work in capture mode, when capture to capture source after the value of the counter lock exist PWM register.	RW	0x0

12.5. Instructions for using the process

Take Timer0 as an example, Timer1 is the same as Timer0.

12.5.1. Counter/timer operation mode

(1) Write TMR0_CONH bit7, TMR0_CONH. Bit6 1 pending,

(2) Config counter initial value, write register TMR0_CNTL;

(3) Configuration values count cycle, write registers TMR0_PRL;

(4) Choose to counter the count of the source, write registers TMR0_CONL [6];

(5) Configuration count source preassigned frequency, write registers TMR0_CONL [then];

(6) If you choose to use count interrupt, write registers TMR0_CONH [5] configuration TMR_IE = 1;

(7) Configure TMR0_CONL[1:0]=2’b01 to work in Timer mode;

(8) Such as TMR0_CONH [7] = 1, i.e., the TIMER PENDING; If the interrupt is enabled, the interrupt will be entered and the corresponding interrupt service subroutine will be executed.

12.5.2. Capture working mode

(1) Write TMR0_CONH.bit7, TMR0_CONH.bit6 1 clear pending;

(2) Config counter initial value, write register TMR0_CNTL;

(3) Configure the count cycle value, write register TMR0_PRL;

(4) Configure TMR0_PWML=0;

(5) Select the counter’s count source and write register TMR0_CONL[4:2];

(6) Configure the pre-frequency division of the counter source, write register TMR0_CONL[7:5];

(7) Configure TMR0_CONH[3:2] to select the capture source;

(8) If the capture source selects pin T0, you need to configure TMR0_CONH[1:0] to select the capture edge;

(9) If you choose to use capture interrupts, write register TMR0_CONH[4] to configure CAP_IE=1;

(10) Configure TMR0_CONL[1:0]=2’b11 to work in capture mode;

(11) When TMR0_CONH[6]=1, CAP PENDING is generated; If the interrupt is enabled,

it will enter the interrupt and execute the corresponding interrupt service subroutine;

(12) Read the counter value when the capture event occurs, by reading the register TMR0_PWML.

12.5.3. PWM operation mode

(1) Write TMR0_CONH.bit7, TMR0_CONH.bit6 1 clear pending;

(2) Config counter initial value, write register TMR0_CNTL;

(3) Configure the count period value, write register TMR0_PRL;

(4) Configure the duty cycle of PWM, write register TMR0_PWML;

(5) Select the counter count source, write register TMR0_CONL[4:2];

(6) Configure the pre-frequency division of the counter source, write register TMR0_CONL[7:5];

(7) Configure TMR0_CONL[1:0]=2’b10 to work in PWM mode.

13. Basic Timer2 module

13.1. Feature Overview

The basic Timer2 is a 16bit basic function timer that supports a variety of counting clock source choices and supports a variety of operating modes such as timer mode, counter mode, capture mode and PWM mode.

13.2. Block diagram of the module

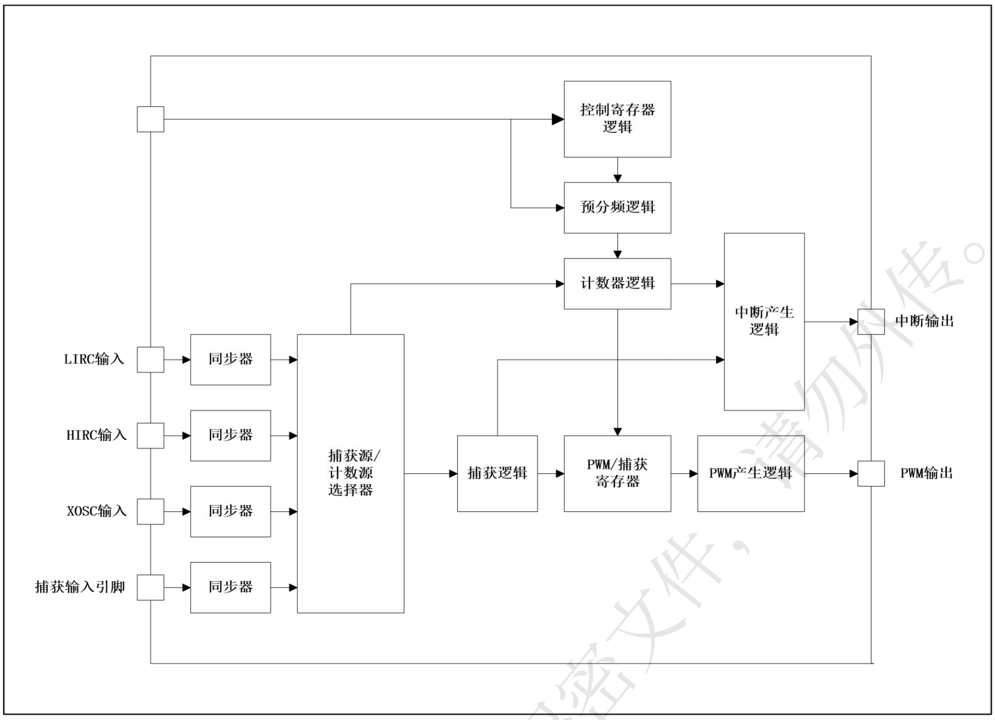


Figure 13.1 Block diagram of a Timer2

13.3. List of registers

Table 13-1 List of Timer2 registers

Address	Register Name	Description
0xC0 (SFR)	TMR2_CONL	TIMER2 control low 8bit register
0xC1 (SFR)	TMR2_CONH	TIMER2 control high 8bit register
0xC2 (SFR)	TMR2_CNTL	TIMER2 counter low 8bit register
0xC3 (SFR)	TMR2_CNTH	TIMER2 counter high 8bit register
0xC4 (SFR)	TMR2_PRL	TIMER2 period low 8bit register
0xC5 (SFR)	TMR2_PRH	TIMER2 period high 8bit register
0xC6 (SFR)	TMR2_PWML	TIMER2 PWM low 8bit register

0xC7 (SFR)	TMR2_PWMH	TIMER2 PWM high 8bit register

13.4. Register details

13.4.1. TMR2_CONL

Addr = 0xC0 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:5	PSC	**TIMER2 pre-split configuration** 0x0: No frequency division 0x1:2 frequency division 0x2:4 split frequency 0x3:8 split frequency 0x4:16 split frequency 0x5:32 split frequency 0x6:64 split frequency 0x7:128 split frequency	RW	0x0
4:2	INCSRC	**TIMER2 Count Source Select configuration** 0x0: T2 rising edge 0x1: T2 falling edge 0x2: hirc_clk_div2 edge(rising & falling) 0x3: rc64k_div2 edge(rising & falling) 0x4: xoscm_div2 edge(rising & falling) 0x5: sys_clk 0x6: sys_clk 0x7: sys_clk	RW	0x0
1:0	MODE	**TIMER2 working mode configuration** 0x0: Off 0x1: COUNTER MODE 0x2: PWM MODE 0x3: CAPTURE MODE	RW	0x0

13.4.2. TMR2_CONH

Addr = 0xC1 (SFR)

Bit(s)	Name	Description	RW	Reset
7	TMRPND	TIMER2 counts pending bits (write 1 clear pending) 0x0: No pending bits are counted 0x1: There is a count pending	RW	0x0
6	CAPPND	TIMER2 captures pending bits (write 1 clear pending) 0x0: No pending was captured 0x1: There is a captured pending	RW	0x0
5	TMRIE	TIMER2 counts the interrupt enable bit 0x0: Off 0x1: Open	RW	0x0
4	CAPIE	TIMER2 captures the interrupt enable bit 0x0: Off 0x1: Open	RW	0x0
3:2	INCSRC	TIMER2 Capture source selection configuration 0x0: T2 pin as the capture source 0x1: Pin T2 serves as the capture source 0x2: The digital output of comparator 0 serves as the capture source 0x3: Comparator 1's digital output serves as the capture source	RW	0x0
1:0	CAPEDG	TIMER2 captures the edge-triggered setting of pin T2 0x0: T2 rising edge triggers capture 0x1: T2 falling edge triggers capture 0x2: T2 double edge trigger capture 0x3: T2 bilateral edge trigger capture	RW	0x0

13.4.3. TMR2_CNTL

Addr = 0xC2 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	CNTL	TIMER2 counter low 8bit initial value	RW	0x0

13.4.4. TMR2_CNTH

Addr = 0xC3 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	CNTH	TIMER2 counter high 8bit initial value	RW	0x0

13.4.5. TMR2_PRL

Addr = 0xC4 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	PRDL	TIMER2 count cycle low 8bit setting value	RW	0xFF

13.4.6. TMR2_PRH

Addr = 0xC5 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	PRDH	TIMER2 count cycle high 8bit setting value	RW	0xFF

13.4.7. TMR2_PWML

Addr = 0xC6 (SFR)

Bit(s)	Name	Description	R/W	Reset

7:0	PWM	**TIMER2 duty cycle low 8bit setting value** In PWM working mode is, this value is the duty cycle setting value of the set bit PWM; In the capture mode, the value lock of the counter grasped after the capture source is captured is stored in the PWM register.	RW	0x0

13.4.8. TMR2_PWMH

Addr = 0xC7 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	PWM	**TIMER2 duty cycle high 8bit setting value** In PWM working mode is, this value is the duty cycle setting value of the set bit PWM; In the capture mode, the value lock of the counter grasped after the capture source is captured is stored in the PWM register.	RW	0x0

13.5. Instructions for using the process

13.5.1. Counter/timer mode of operation

(1) Write TMR2_CONH.bit7, TMR2_CONH.bit6 1 clear pending;

(2) Configure counter initial value, write register TMR2_CNTL, TMR2_CNTH;

(3) Configure the count cycle value, write registers TMR2_PRL, TMR2_PRH;

(4) Select the counter count source, write register TMR2_CONL[4:2];

(5) Configure the pre-frequency division of the counter source, write register TMR2_CONL[7:5];

(6) If you choose to use counting interrupts, write register TMR2_CONH[5] with

TMR_IE=1;

(7) Configure TMR2_CONL[1:0]=2’b01 to work in Timer mode;

(8) When TMR2_CONH[7]=1, TIMER PENDING is generated; If the interrupt is enabled, the interrupt will be entered and the corresponding interrupt service subroutine will be executed.

13.5.2. Capture working mode

(1) Write TMR2_CONH.bit7, TMR2_CONH.bit6 1 clear pending;

(2) Configure counter initial value, write register TMR2_CNTL, TMR2_CNTH;

(3) Configure the count cycle value, write registers TMR2_PRL, TMR2_PRH;

(4) Configure TMR2_PWML=0, TMR2_PWMH=0;

(5) Select the counter’s count source and write register TMR2_CONL[4:2];

(6) Configure the pre-frequency division of the counter source, write register TMR2_CONL[7:5];

(7) Configure TMR2_CONH[3:2] to select the capture source;

(8) If the capture source selects the T2 pin, you need to configure TMR2_CONH[1:0] to select the capture edge;

(9) If you choose to use capture interrupts, write register TMR2_CONH[4] to configure CAP_IE=1;

(10) Configure TMR2_CONL[1:0]=2’b11 to work in capture mode;

(11) When TMR2_CONH[6]=1, CAP PENDING is generated; If the interrupt is enabled, it will enter the interrupt and execute the corresponding interrupt service subroutine;

(12) Read the counter value when the capture event occurs by reading the

registers TMR2_PWML, TMR2_PWMH.

13.5.3. PWM working mode

(1) Write TMR2_CONH.bit7, TMR2_CONH.bit6 1 clear pending;

(2) Configure counter initial value, write register TMR2_CNTL, TMR2_CNTH;

(3) Configure the count cycle value, write registers TMR2_PRL, TMR2_PRH;

(4) Configure the duty cycle of PWM, write registers TMR2_PWML, TMR2_PWMH;

(5) Select the counter count source, write register TMR2_CONL[4:2];

(6) Configure the pre-frequency division of the counter source, write register TMR2_CONL[7:5];

(7) Configure TMR2_CONL[1:0]=2’b10 to work in PWM mode.

14. Advanced Timer 1/2 module

14.1. Feature Overview

Advanced Timer is one that contains two timers STMR1/2. STMR1/2 is an advanced counter with the same function and can be used to generate different forms of clock waveforms, one timer can generate a set of complementary PWMS with the same frequency, or 2-way PWMS with the same period and different duty cycles. External input can be captured for pulse width or period measurement.

The main features are as follows:

- Built-in 16-bit counter, counting up or down, automatic reloading
- Clock source can be selected from the system system clock or internal crystal

or external RTC clock

- Presplit frequency 1-16
- Post division 1, 2, 4, 8, 16, 32, 64, 128(cycle interval response)
- Sync with external signal (external clock, reset)
- Input capture (rising edge, falling edge and double edge) and compare functions
- Count input edges, optional rising edge, falling edge and double edge
- Brake input, can set the output of TMR1/2 to a specific state
- Support PWM output function
- It can output 2-way PWM with the same cycle and different duty cycle, or 1-way complementary PWM with programmable dead-time output
- Support brake function, brake input is comparator output
- Shadow register, trigger update event to update
- Interrupt, which generates an interrupt at the following event:
  - Counter overflow
  - The counter underflows
  - CHA/CHB input capture
  - CHA/CHB output comparison
  - Brake generation (software or comparator)

14.1.1. Basic action

1. Basic Waveform patterns

TIMER1/2 has 2 basic counting waveform modes, sawtooth wave mode and triangle wave mode. The waveform mode is due to no

The same internal counting action is subdivided, the triangle wave mode is divided into triangle wave A mode and triangle wave B mode. The basic waveform of sawtooth wave and triangle wave is as shown. The difference between delta A mode and delta B mode is that there is A difference in buffer transfer. Delta A mode only takes one buffer transfer per cycle (valley point), while delta B mode takes two buffer transfers per cycle (peak point and valley point).

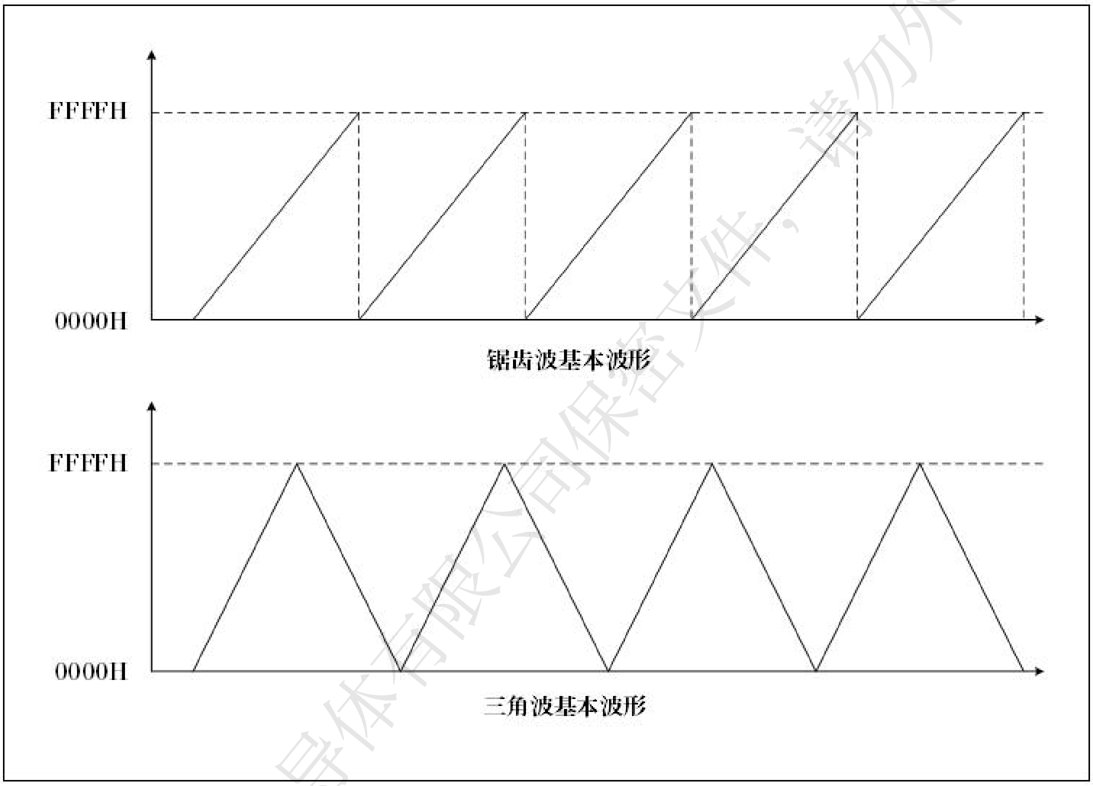


Figure 14.1 Diagram of the underlying waveform

Compare the output

STMR1/2 A timer has 2 comparison output ports (CHA, CHB), which can output the specified level when the count value is compared and matched with the count reference value. The CMPA_S ({CMPAH_S,CMPAL_S}) and CMPB_S ({CMPBH_S,CMPBL_S}) registers correspond to the count comparison reference values of CHA and CHB, respectively. When the counter value is equal to CMPA_S, the CHA port outputs the specified level;

When the counter and CMPB_S are equal, the CHB port outputs the specified level.

CHA, the count start level at the CHB port, and the level at which the count comparison matches are defined by PAINITVAL/PBINITVAL and CAPAVAL/CAPBVAL. Figure shows an example of action for comparing the output.

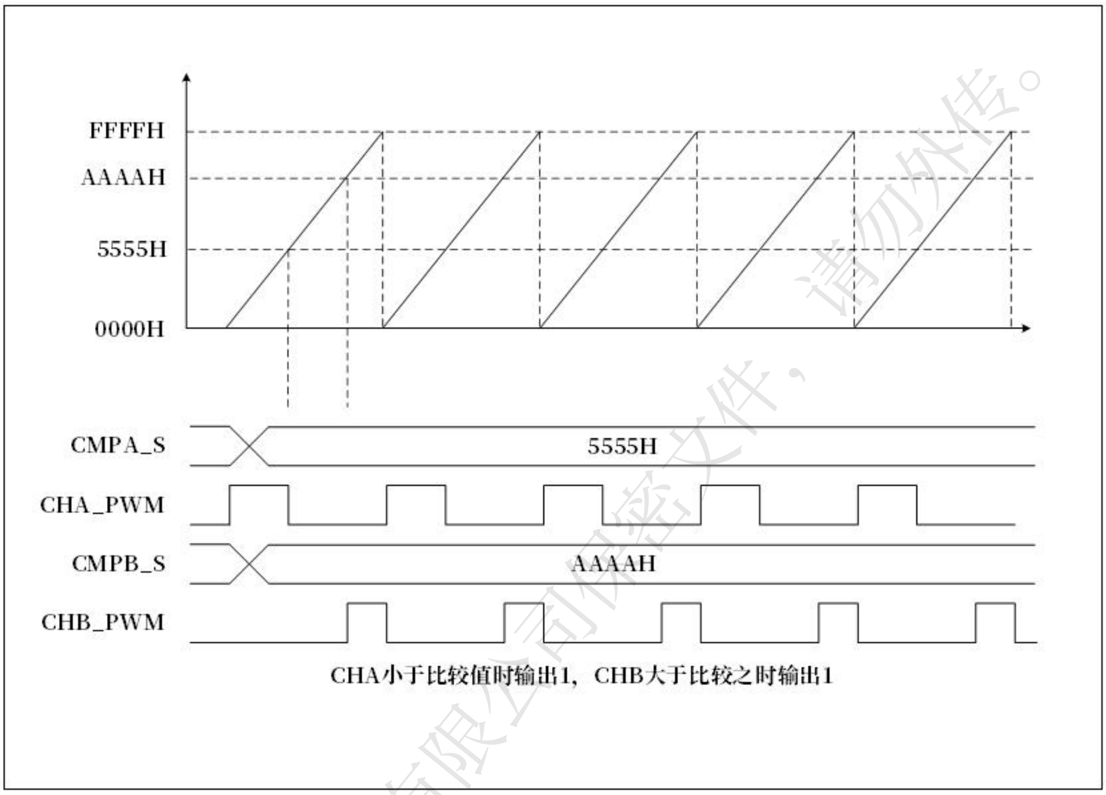


Figure 14.2.Comparison output waveform diagram

3 Capture the input

STMR1/2 has capture input function, with two groups of capture input registers (CMPA_S, CMPB_S), used to save the captured count value. The CAPA_EN/CAPB_EN bit of the port control register (PCONRA/PCONRB) is set to 1, and the capture input function of the corresponding port is effective. When the capture input condition is set and valid, the current count value is saved to the corresponding registers (CMPA_S, CMPB_S). The condition of each set of capture input can choose the rising edge, falling edge or rising and falling edge of CHA or CHB. The capture condition of the corresponding port is set by CAPA_MODE/CAPB_MODE.

The two registers STMRxPRL_S and STMRxPRH_S determine the overflow time of the counter inside the timer. In the capture mode, it is recommended to set both registers to 0xFF, and the saw-wave mode is recommended to use in the capture mode. The capture diagram refers to the figure below.

When reading the base registers STMRxPRL_S, STMRxPRH_S, STMRxCMPAL_S, STMRxCMPAH_S, STMRxCMPBL_S, STMRxCMPBH_S (periodic value and comparison value) in capture mode, The SEL_SREG setting of STMRxCR determines whether the value of the shadow register or the base register should be read. SEL_SREG is set to 0 to read the value of the base register (i.e., the capture value), otherwise the value of the shadow register is read. There is no point in writing to the comparison value shadow register in capture mode.

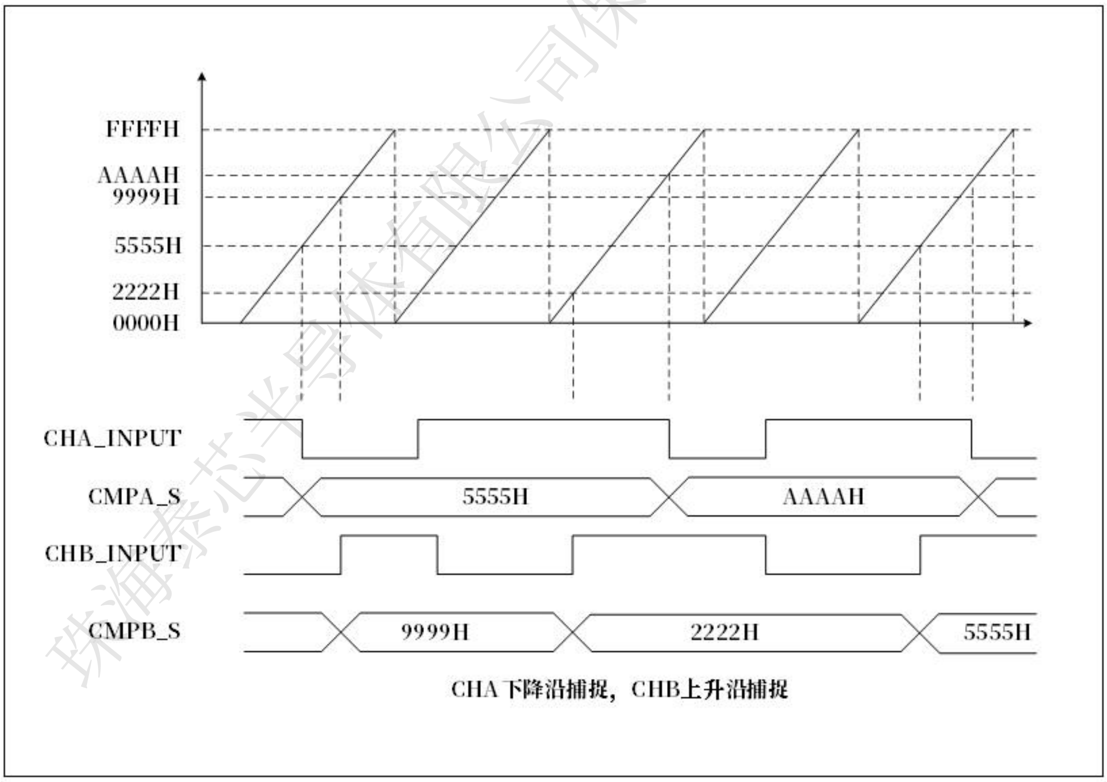


Figure 14.3 Capture mode waveform

14.1.2. Clock Source Selection

The STMR1/2's counting clock can have the following options:

- System clock (SYSCLK)
- Internal low speed RC oscillator 32kHz clock
- External crystal oscillator (counting clock is external crystal /2)
- CHA/ CHB

Clock frequency division 1-16 optional, can be internal low speed RC, external low speed RC, CHA, CHB rising edge, falling edge, rising and falling edge optional count.

14.1.3. Direction of count

The counter count direction of STMR1/2 can be changed by software. For different waveform modes, the method of changing the counting direction is slightly different.

14.1.3.1. Sawtooth wave counts direction

In sawtooth wave mode, the counting direction can be set in the counter count or when stopping.

In the incremental count, set STMRxCR.DIR=0 (decreasing count), then the counter count to overflow into decreasing count mode; When in decrement count, set STMRxCR.DIR=1 (increment count), the counter will change to increment count mode after underflow.

When counting stops, set the STMRxCR.DIR bit. The value of STMRxCR.DIR is updated as soon as the count starts.

Note: Change the value of STMRxCR.DIR when the counting mode is enabled. If there are other bits to be configured in the CR register, please configure the DIR bit last after configuring the other bits. Or do not use (or about and about), configure the CR register.

If the period and comparison values are modified at the same time when the DIR value is modified, it may occur that it needs to wait for the original period count to overflow to the top or bottom (according to the DIR direction) before it is modified according to the new period and comparison value.

14.1.3.2. Triangle wave count direction

In triangle wave mode, the counting direction can only be set when the counter stops. Setting the counting direction in the count is not valid. When the count stops, set the STMRxCR.DIR bit for immediate effect.

14.1.4. Digital filtering

CHA and CHB port inputs of STMR1/2 have digital filtering function. The filtering function of the corresponding port can be enabled by setting STMRxPCONRA.PA_FILTER_EN/STMRxPCONRB.PB_FILTER_EN. The filter clock is the current working clock of the counter.

When the filter sampling reference clock is sampled to a consistent level on the port 3 times, the level is transmitted to the module as a valid level; The level less than 3 times consistent is filtered out as external interference and is not transmitted to the module. The action is as shown.

Digital filtering is also used to filter the signal transmitted from the voltage comparator. It is enabled by STMR1_CR.THA_FILTER_EN/ STMR1_CR.THB_FILTER_EN, and the filtering clock is the system clock.

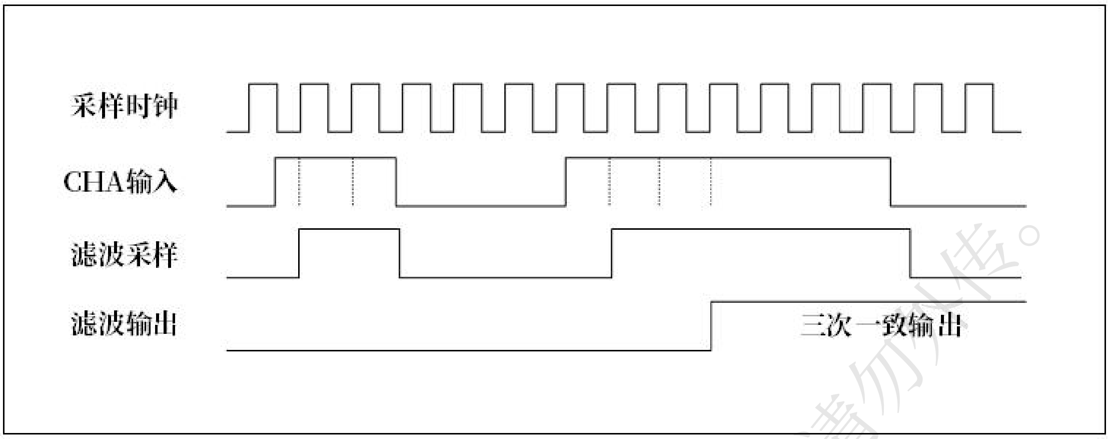


Figure 14.4 Digital filtering waveform

14.1.5. Software synchronization

STMR1/2 can realize the synchronous start of the target STMR1/2 by setting the software synchronous start register SSTAR (STMR_ALLCON[3]).

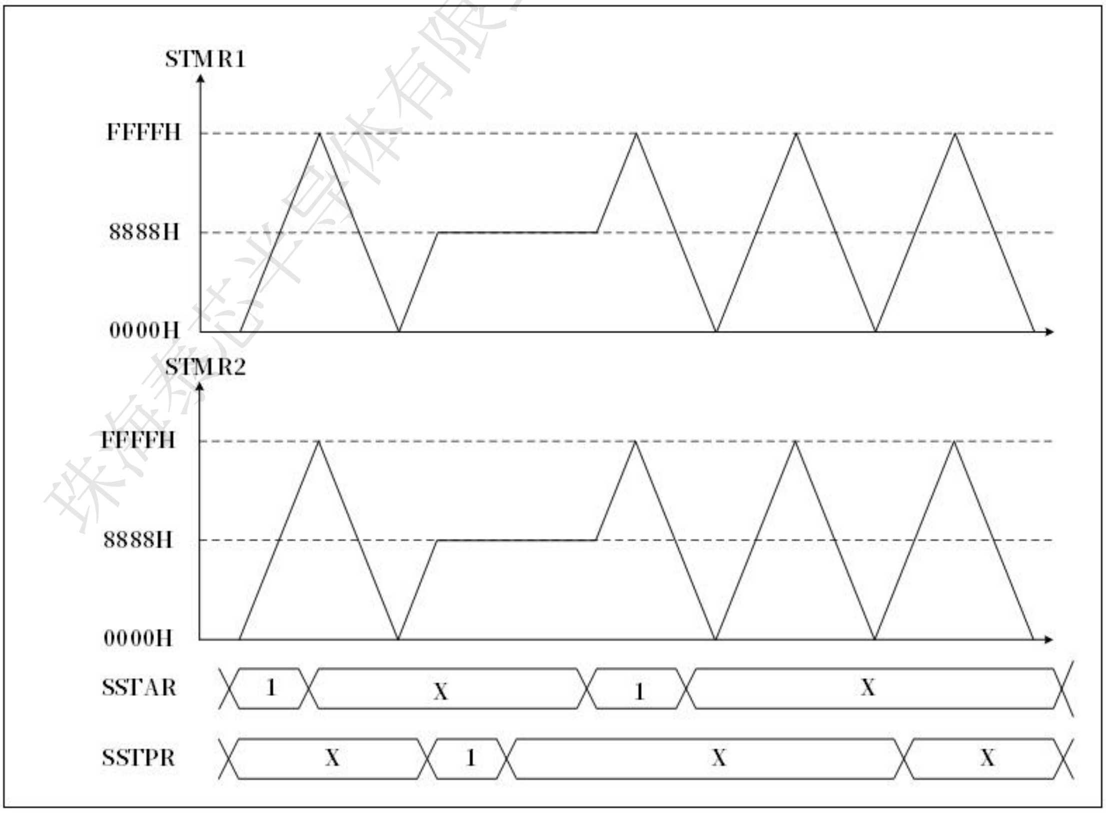


Figure 14.5 Waveform of synchronous start and stop

14.1.5.1. Software synchronization stop

STMR1/2 can achieve the synchronization stop of the target STMR1/2 by setting the software synchronization stop register SSTPR (STMR_ALLCON[4]). At this time, the counter is in the suspended state, and the synchronization start register SSTAR (STMR_ALLCON[3]) can continue counting by writing 1.

14.1.5.2. Software synchronously reset

STMR1/2 can realize the synchronous reset of target STMR1/2 by setting the software synchronous reset register (STMR_ALLCON[5]), and the counter will be reset to the initial state.

If STMR_ALLCON[3] is set, the software synchronous start of STMR1/2 can be realized.

The software synchronization action related register is a set of registers that are independent of STMR1/2 and shared among STMRS. Each bit of this set of registers is only valid when writing 1, and is invalid when writing 0.

14.1.6. Caching functions

Cache action means that when an update event occurs, the following events occur:

PR={PRH,PRL}; CMPA={CMPAH,CMPAL}; CMPB={CMPBH,CMPBL};

a. The value of the generic periodic reference shadow register PR is automatically transferred to the generic periodic reference register PR_S ({PRH_S,PRL_S});

b. The values of universal comparison reference shadow registers (CMPA, CMPB) are automatically transferred to universal comparison reference registers (CMPA_S, CMPB_S) (when comparing the output);

c. The values of the Universal Comparison Base value registers (CMPA_S, CMPB_S) are automatically transferred to the Universal Comparison Base value shadow registers (CMPA, CMPB) (when capturing input);

As shown in the figure, it is the timing diagram of the single buffer mode of the universal comparison reference value register when comparing the output action. It can be seen that the output duty cycle can be adjusted by changing the value of the Common Comparison Pivot Register (CMPA_S) during the count, and the output cycle can be adjusted by changing the value of the common Cycle Pivot Register (PR).

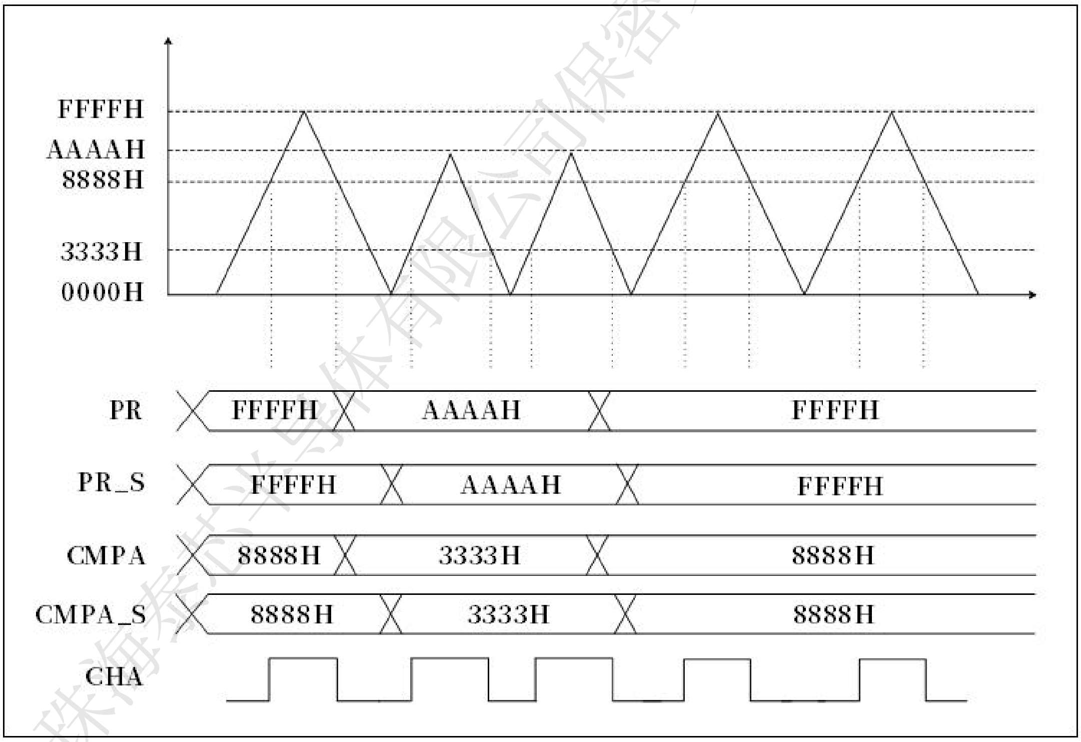


Figure 14.6 Automatic loading waveform diagram

14.1.6.1. Cache delivery time point

The period reference cache transfer time point is the sawtooth wave with an incremental count of overflow point or a decreasing count of underflow point.

In triangle wave mode, cache transfer occurs at (CNT=PR_S) overflow point or (CNT=0) underflow point.

In triangle wave mode A, cache transfer occurs at the count underflow point.

In triangular wave B mode, cache transfer occurs at count underflow points and count overflow points.

The capture input action cache transfer time point is when the capture input action.

When the counting mode is started (STMRxCR.TMREN) and there is a reset action during the normal comparison period, the universal cycle reference value, the universal comparison reference value, and so on will occur a cache transfer (update event) according to the corresponding cache action setting status.

14.1.7. Universal PWM output

14.1.7.1. Independent PWM output

The 2 ports of each timer CHA, CHB can output PWM wave independently. As shown in the figure, the timer CHA port outputs PWM wave.

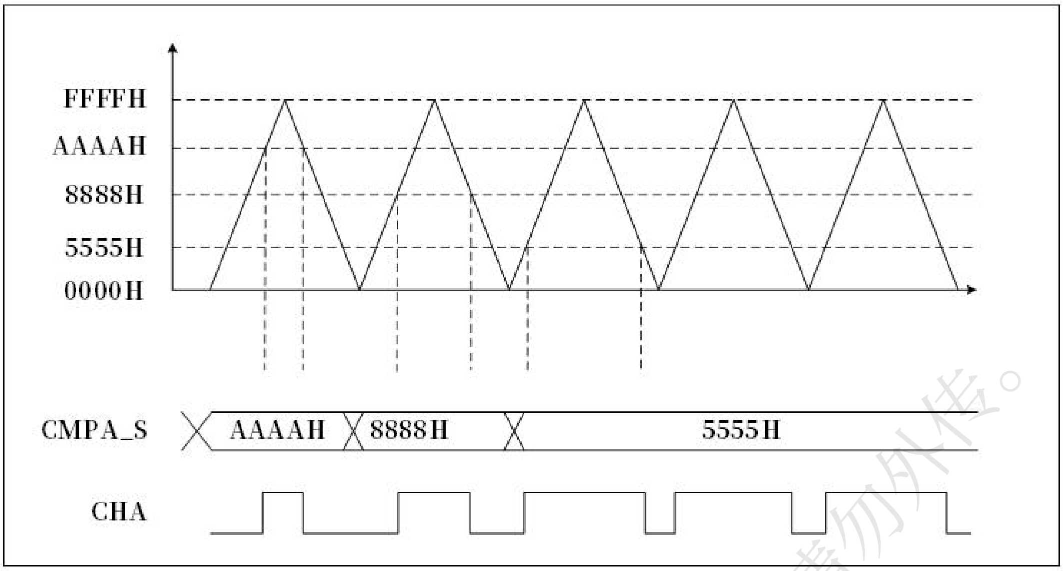


Figure 14.7 Timer CHA port outputs PWM

14.1.7.2. Complementary PWM output

CHA port and CHB port, in different modes by setting the polarity of the port can be combined to output complementary PWM waveform.

Software set CMPB_S complementary PWM output.

Software setting CMPB_S complementary PWM output means that in sawtooth wave mode and triangle wave A mode and triangle wave B mode, the complementary PWM output is formed by setting opposite output polarities. The value of the Universal Reference Value Register (CMPB_S) used for the CHB port waveform output is set directly by the register (CMPB) and has no direct relationship with the value of the Universal Reference Value Register (CMPA_S). The following figure shows an example of the software setting CMPB_S complementary PWM wave.

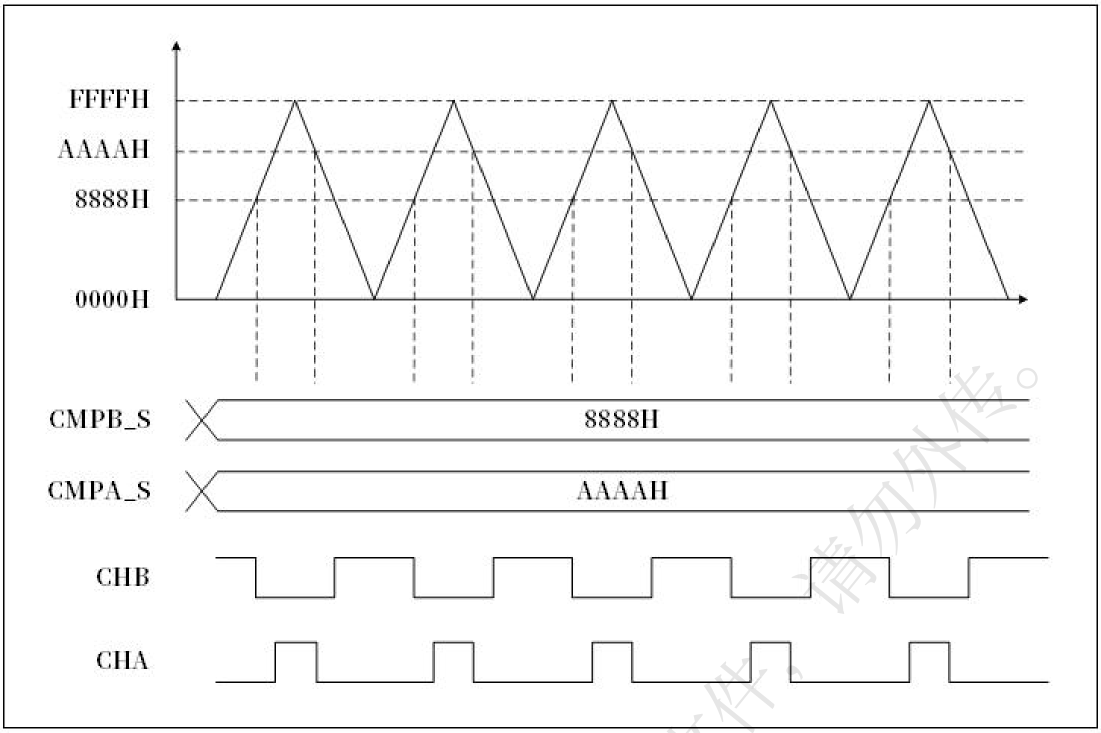


Figure 14.8 Example of software setting CMPB_S complementary PWM wave

Hardware setting CMPB_S complementary PWM output

Hardware setting CMPB_S complementary PWM output means that the value of Universal Comparison Reference Value Register (CMPB_S) used for CHB port waveform output is determined by the value operation of Universal Comparison Reference Value Register (CMPA_S) minus dead Time Reference Value Register (DTUA) in triangle wave A mode and triangle wave B mode. Figure shows an example of CMPB_S complementary PWM wave output set by hardware. The dead Time Reference Value Register (DTUA) is 8bit, and the adjustment range is 0~255.

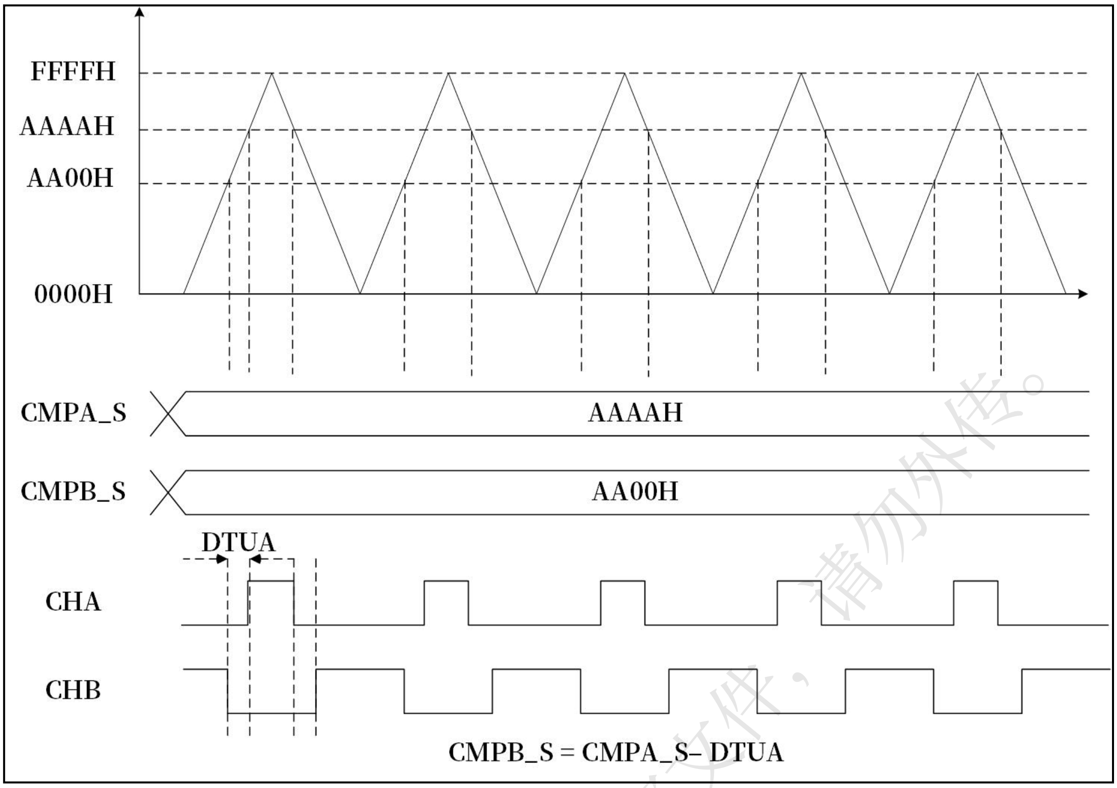


Figure 14.9 Complementary waveform with dead time

14.1.8. Period-interval response

STMR1/2's general comparison datum registers (CMPA_S, CMPB_S) can generate special valid request signals (CMPA_ACK/CMPB_ACK) when the counts are matched.

The request signal can generate a valid request signal after every few cycles. The stMRXvperr.pcnts bit of the valid cycle register (STMRxVPERR) is set to specify how many cycles the request signal is valid. In other cycles, even if the count value is equal to the value of the reference value register CMPA_S or CMPB_S, no valid request signal will be output. The figure shows an action example of a valid request signal at a cycle interval.

The request signals matched by the comparison values of CHA/ CHB are output separately.

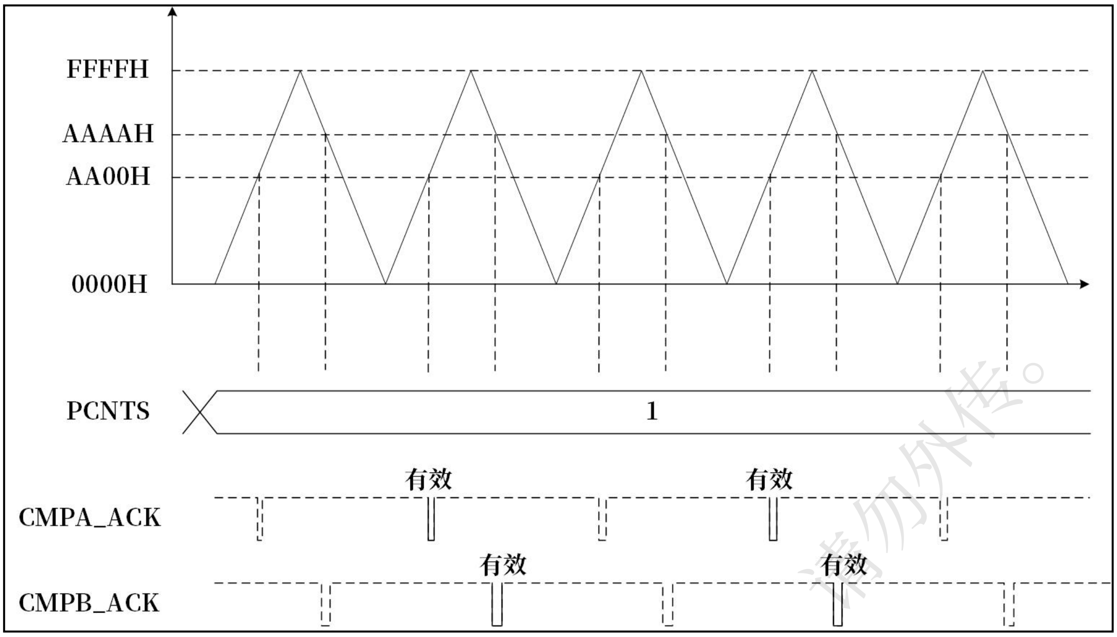


Figure 14.10 Action waveform of valid request signals for periodic intervals

14.1.9. Protection mechanism

STMRx can protect and control the output state of the port.

STMRx has two common port input invalidations (from analog comparators or software config), and the abnormal condition events on each port can be set from the brake control. When abnormal conditions are detected on these ports, the universal PWM output can be controlled.

During the normal output period of the port, if the brake event from the brake control is detected, the output state of the port can be changed to the preset state. The state of the universal PWM output port can be changed to output high resistance state, output low level or output high level (STMRxVPERR.BRAKEA_VAL/ STMRxDTR.BRAKEB_VAL setting decision) when the brake control abnormal event occurs.

For example, if the BRAKEA_VAL=2’b10 is set, during the normal output of the CHA port, if the brake event occurs, the output on the CHA port will become high resistance state.

Note: For software to brake CHA alone, please set STMRxVPERR.BRAKEA_SF; For individual brake CHB, please set STMRxDTR.BRAKEB_SF; Set these two registers to 1, with T1A_EN and T1B_EN enabled, the software brake signal is effective. STMRx_BRAKE.T1A_AOE/TIB_AOE sets whether software reset or hardware reset, and BRAKEA_VAL/BRAKEB_VAL bit sets the output state of the output end.

14.1.10. Interrupt description

STMR1/2 contains 6 interrupts for each of 4 categories. There are two general counting comparison matching interrupts (including two capture input interrupts), two counting cycle matching interrupts, and two brake protection interrupts.

14.1.11. Brake protection

When a braking event occurs (from the voltage comparator), the hardware automatically changes the port state to the preset state (high, low, high resistance state).

14.1.12. Internal interconnect

- Voltage comparators can trigger the brake function.
- The STMR1/2 interrupt can trigger the ADC sampling function.

14.1.13. Infrared function

The PWM of the advanced Timer2 channel A combined with the PWM of the basic Timer2 can be used as an infrared generator. Enable infrared function by setting SYS_CON2[3] to 1. At this time, the basic Timer2 is used as the carrier PWM, and the advanced Timer2 channel A is used as the modulation PWM. The infrared code is judged to be 1 or 0 by the software, and the period and comparison value shadow registers of the advanced Timer2 channel A are changed to perform infrared modulation. The result is output to GPIO through the PWM of Timer2.

14.2. Block diagram of the module

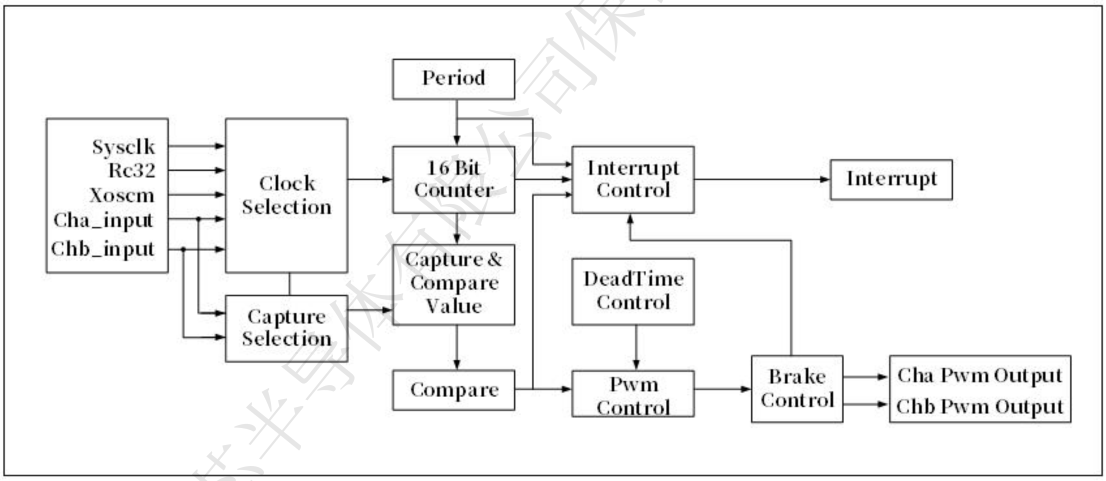


Figure 14.11 High-level Timer module architecture

14.3. List of registers

Table 14-1 List of STimer registers

Address	Register Name	Description
0xC8 (SFR)	STMR1_CNTL	STMR1 Count Low Register

0xC9 (SFR)	STMR1_CNTH	STMR1 Count High Register
0xCA (SFR)	STMR1_PRL	STMR1 Period Low Register
0xCB (SFR)	STMR1_PRH	STMR1 Period High Register
0xCC (SFR)	STMR1_CMPAL	STMR1 Channel A Comparison Value Low Register
0xCD (SFR)	STMR1_CMPAH	STMR1 Channel A Comparison Value High Register
0xCE (SFR)	STMR1_CMPBL	STMR1 Channel B Comparison Value Low Register
0xCF (SFR)	STMR1_CMPBH	STMR1 Channel B Comparison Value High Register
0xD1 (SFR)	STMR1_CR	STMR1 Control Register
0xD2 (SFR)	STMR1_FCONR	STMR1 Time Control Register
0xD3 (SFR)	STMR1_VPERR	STMR1 Count Period Register
0xD4 (SFR)	STMR1_DTUA	STMR1 DeadTime Register
0xD5 (SFR)	STMR1_BRAKE	STMR1 Brake Control Register
0xD6 (SFR)	STMR1_DTR	STMR1 DeadTime Control Register
0xD7 (SFR)	STMR1_PCONRA	STMR1 Channel A Control Register
0xD8 (SFR)	STMR1_PCONRB	STMR1 Channel B Control Register
0xD9 (SFR)	STMR1_IE	STMR1 Interrupt Enable Register
0xDA (SFR)	STMR1_SR	STMR1 Interrupt Flag Register
0xDB (SFR)	STMR2_CNTL	STMR2 Count Low Register
0xDC (SFR)	STMR2_CNTH	STMR2 Count High Register
0xDD (SFR)	STMR2_PRL	STMR2 Period Low Register
0xDE (SFR)	STMR2_PRH	STMR2 Period High Register
0xDF (SFR)	STMR2_CMPAL	STMR2 Channel A Comparison Value Low Register
0xE1 (SFR)	STMR2_CMPAH	STMR2 Channel A Comparison Value High Register
0xE2 (SFR)	STMR2_CMPBL	STMR2 Channel B Comparison Value Low Register

0xE3 (SFR)	STMR2_CMPBH	STMR2 Channel B Comparison Value High Register
0xE4 (SFR)	STMR2_CR	STMR2 Control Register
0xE5 (SFR)	STMR2_FCONR	STMR2 Time Control Register
0xE6 (SFR)	STMR2_VPERR	STMR2 Count Period Register
0xE7 (SFR)	STMR2_DTUA	STMR2 DeadTime Register
0xE8 (SFR)	STMR2_BRAKE	STMR2 Brake Control Register
0xE9 (SFR)	STMR2_DTR	STMR2 DeadTime Control Register
0xEA (SFR)	STMR2_PCONRA	STMR2 Channel A Control Register
0xEB (SFR)	STMR2_PCONRB	STMR2 Channel B Control Register
0xEC (SFR)	STMR2_IE	STMR2 Interrupt Enable Register
0xED (SFR)	STMR2_SR	STMR2 Interrupt Flag Register
0xF5 (SFR)	STMR_ALLCON	STMR ALL Control Register

14.4. Register details

14.4.1. STMR1_CR

Addr = 0xD1 (SFR)

Bit(s)	Name	Description	R/W	Reset
7	THBFILTEREN	**STMR1 brake input CHB filter control** 0x0: Brake input CHB off digital filtering 0x1: Brake input CHB on digital filtering **Note:** STMR1/2 shared, set in STMR1 only, STMR2 shared STMR1 set.	RW	0x0
6	THAFILTEREN	**STMR1 brake input CHA filter control** 0x0: Brake input CHA off digital filter 0x1: Brake input CHA on digital filtering	RW	0x0

		Note: STMR1/2 shared, set in STMR1 only, STMR2 shared STMR1 set.		
5	-	-	-	-
4	SELSREG	**STMR1 Read shadow register control** 0x0: Read register PR/CMPA/CMPB to get the value of register PR_S/CMPA_S/CMPB_S 0x1: Read register PR/CMPA/CMPB to get the value of register PR/CMPA/CMPB	RW	0x0
3	DIR	**STMR1 Count direction control** 0x0: Count down 0x1: Count up	RW	0x1
2:1	MODE	**STMR1 Counter count mode** 0x0: Sawtooth Wave count mode 0x1: Triangle Wave A count mode 0x2: Triangle Wave B count mode 0x3: Hold	RW	0x0
0	TMREN	**STMR1 Count enable control** 0x0: Turn off STMR1 counting 0x1: Turn STMR1 count on	RW	0x0

14.4.2. STMR1_FCONR

Addr = 0xD2 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:6	INCSEL	**STMR1 count source** 0x0: SYS_CLK 0x1: PIN_SEL rising edge 0x2: the falling edge of PIN_SEL 0x3: PIN_SEL edge (rising and falling)	RW	0x0
5:4	PINSEL	**STMR1 Count clock source** 0x0: XOSCM/2 0x1:32KHz RC	RW	0x0

		0x2: CHA input 0x3: CHB input		
3:0	PREDIV	**STMR1 counts predividers** 0~15 corresponds to 1~16 frequency division	RW	0x0

14.4.3. STMR1_CNTL

Addr = 0xC8 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	STMR1CNTL	**STMR1 counts 8 bits lower in the register**	RW	0x0

14.4.4. STMR1_CNTH

Addr = 0xC9 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	STMR1CNTH	**STMR1 count register 8 bits high**	RW	0x0

14.4.5. STMR1_PRL

Addr = 0xCA (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	STMR1PRL	**STMR1 cycle register 8 bits lower**	RW	0xFF

14.4.6. STMR1_PRH

Addr = 0xCB (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	STMR1PRH	**STMR1 cycle register is 8 bits high**	RW	0xFF

14.4.7.　STMR1_CMPAL

Addr = 0xCC (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	STMR1CMPAL	STMR1 CHA compare value register 8 bits lower	RW	0x0

14.4.8.　STMR1_CMPAH

Addr = 0xCD (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	STMR1CMPAH	STMR1 CHA is 8 bits higher than the value register	RW	0x0

14.4.9.　STMR1_CMPBL

Addr = 0xCE (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	STMR1CMPBL	STMR1 CHB comparison value register 8 bits lower	RW	0x0

14.4.10.　STMR1_CMPBH

Addr = 0xCF (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	STMR1CMPBH	STMR1 CHB is 8 bits higher than the value register	RW	0x0

14.4.11. STMR1_VPERR

Addr = 0xD3 (SFR)

Bit(s)	Name	Description	R/W	Reset
7	BRAKEASF	**STMR1 CHA software brake signal** 0x0: CHA software brake signal is invalid 0x1: CHA software brake signal is valid	RW	0x0
6:5	BRAKEAVAL	**STMR1 CHA brake output value** 0 x0: brake event is valid, CHA PWM output is 0 0x1: CHA PWM outputs 1 when the brake event is valid 0x2: CHA PWM output is off when the brake event is active CHA 0 x3: brake event is valid), PWM output shut down	RW	0x0
4:3	PCNTE	**STMR1 cycle interval response count condition** 0x0: The cycle interval function is invalid 0x1: Sawtooth wave up or down overflow point or triangle wave valley 0x2: sawtooth wave up or underflow point or triangle wave peak 0x3: Sawtooth wave upper or underflow point or triangle wave peak and trough	RW	0x0
2:0	PCNTS	**STMR1 cycle interval response time** 0x0:1 response per cycle 0x1: one response in 2 cycles 0x2: one response in 4 cycles 0x3: eight cycles respond once 0 x4:16 cycle response time	RW	0x0

		0x5: one response in 32 cycles 0x6: one response in 64 cycles 0x7: one response in 128 cycles		

14.4.12. STMR1_DTUA

Addr = 0xD4 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	STMR1DTUA	**STMR1 dead-time setting register**	RW	0x0

14.4.13. STMR1_BRAKE

Addr = 0xD5 (SFR)

Bit(s)	Name	Description	R/W	Reset
7	T1BMOE	**Normal PWM output sign after STMR1 CHB brake event** 0x1: CHB PWM returns to normal output 0x0: CHB PWM output according to brake configuration **Note: When the brake event** is valid, it will be immediately reset synchronously; When the effective brake signal disappears, according to the choice of T1BAOE, it will be set by software or automatically set by hardware.	RW	0x1
6	T1BAOE	**STMR1 CHB PWM normal output flag control bit** 0x0: T1BMOE can only be software set to 1 after a valid brake event 0x1: T1BMOE can be set 1 by software and overflow after a valid brake event	RW	0x0
5	T1BSEL	**STMR1 CHB brake source**	RW	0x0

		0x0: CHB selects analog comparator 0 output as brake source 0x1: CHB selects analog comparator 1 output as brake source		
4	T1BEN	**STMR1 CHB brake enabled** 0x0: CHB brake is not enabled 0x1: CHB brake enabled	RW	0x0
3	T1AMOE	**After STMR1 CHA braking event, PWM output signs normally** 0x0: CHA PWM output according to brake configuration 0x1: CHA PWM returns to normal output **Note: When the brake event** is valid, it will be immediately reset synchronously; When the effective brake signal disappears, according to the choice of T1AAOE, the software set 1 or the hardware set 1 automatically.	RW	0x1
2	T1AAOE	**STMR1 CHA PWM normal output flag control bit** 0x0: T1AMOE can only be software set to 1 after a valid brake event 0x1: T1AMOE can be set to 1 by software and overflow after a valid braking event	RW	0x0
1	T1ASEL	**STMR1 CHA Brake source** 0x0: CHA selects the analog comparator 0 output as the brake source 0x1: CHA selects analog Comparator 1 output as brake source	RW	0x0
0	T1AEN	**STMR1 CHA brake enabled** 0x0: CHA brake is not enabled 0x1: CHA brake enabled	RW	0x0

14.4.14. STMR1_DTR

Addr = 0xD6 (SFR)

Bit(s)	Name	Description	R/W	Reset
7	BRAKEBSF	**STMR1 CHB software brake signal** 0x0: CHB software brake signal is invalid 0x1: CHB software brake signal is valid	RW	0x0
6:5	BRAKEBVAL	**STMR1 CHB brake output value** 0x0: CHB PWM output 0 when the brake event is valid 0x1: CHB PWM output 1 when brake event is valid 0x2: CHB PWM output is off when brake event is active 0x3: CHB PWM output is off when brake event is active	RW	0x0
4	HWCPWM	**Hardware set CMPB_S register enabled in STMR1 complementary mode** 0x0: CMPB_S register is not enabled by hardware setting 0x1: CMPB_S register is enabled by hardware setting	RW	0x0
3	DTBHO	**CHB dead-zone output in STMR1 complementary mode** 0 x1: CHB PWM dead zone output shut down 0 x0: CHB normal output PWM dead zone	RW	0x0
2	DTBEN	**STMR1 CHB dead zone enabled** 0x0: Dead zone setting is invalid 0 x1: dead zone set up effectively	RW	0x0
1	DTAHO	**CHA dead-zone output in STMR1 complementary mode** 0x0: CHA PWM dead-time output is normal 0 x1: CHA PWM dead zone output shut down	RW	0x0

0	DTAEN	**STMR1 CHA dead zone can make** 0 x0: dead zone setting 0 x1: dead zone set up effectively	RW	0x0

14.4.15. STMR1_PCONRA

Addr = 0xD7 (SFR)

Bit(s)	Name	Description	R/W	Reset
7	PAINITVAL	**STMR1 CHA PWM initial output values** 0x0: The initial PWM state is 0 after CHA count is enabled 0x1: Initial PWM state is 1 after CHA count is enabled	RW	0x0
6:5	CMPAVAL	**STMR1 CHA PWM output value** 0x0: The count value is less than the CHA comparison value output 1 and greater than the output 0 0x1: The count value is greater than the CHA comparison output 1 and less than the output 0 0x2: The count value is equal to the CHA comparison value and the output is flipped 0 x3: output unchanged	RW	0x0
4	PAENO	**Can make STMR1 CHA PWM output** 0x0: CHA PWM output is not enabled 0 x1: CHA PWM output can make	RW	0x0
3	PAFILTEREN	**STMR1 CHA input filter enabled** 0 x0: CHA input signal filtering 0x1: CHA input signal filtering	RW	0x0
2:1	CAPAMODE	**STMR1 CHA capture point selection** 0x0: No capture	RW	0x0

		0x1: Capturing the rising edge 0x2: Capture the falling edge 0x3: Capture edges (rising and falling edges)		
0	CAPAEN	**STMR1 CHA capture mode enabled** 0x0: Capture mode is not enabled 0x1: Capture mode enabled	RW	0x0

14.4.16. STMR1_PCONRB

Addr = 0xD8 (SFR)

Bit(s)	Name	Description	R/W	Reset
7	PBINITVAL	**STMR1 CHB PWM initial output value** 0x0: The initial PWM state is 0 after CHB count is enabled 0x1: PWM initial state is 1 after CHB count is enabled	RW	0x0
6:5	CMPBVAL	**STMR1 CHB PWM output values** 0x0: The count value is less than the CHB comparison value output 1, greater than the output 0 0x1: The count value is greater than the CHB comparison value output 1, less than the output 0 0x2: The count value is equal to the CHB comparison value, the output is flipped 0x3: Output remains the same	RW	0x0
4	PBENO	**STMR1 CHB PWM output enabled** 0 x0: CHB PWM output is not enabled 0x1: CHB PWM output is enabled	RW	0x0
3	PBFILTEREN	**STMR1 CHB input filter enabled** 0x0: CHB input signal is not filtered 0x1: CHB input signal filtered	RW	0x0

2:1	CAPBMODE	**STMR1 CHB capture point selection** 0x0: No capture 0x1: Capturing the rising edge 0x2: Capture the falling edge 0x3: Capture edges (rising and falling edges)	RW	0x0
0	CAPBEN	**STMR1 CHB capture mode enabled** 0x0: Capture mode is not enabled 0 x1: capture mode enabled	RW	0x0

14.4.17. STMR1_IE

Addr = 0 xd9 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:6	-	-	-	-
5	BRAKEBIE	**STMR1 CHB brake interrupt enabled** 0x0: CHB brake interrupt is not enabled 0x1: CHB brake interrupt enabled	RW	0x0
4	BRAKEAIE	**STMR1 CHA brake interrupt enabled** 0x0: CHA brake interrupt is not enabled 0x1: CHA brake interrupt enabled	RW	0x0
3	CMPBIE	**STMR1 count value is equal to CHB occur more value/capture interrupt enabled** 0 x0: count value is equal to CHB comparison values or capture, interrupt is not enabled 0x1: Count value equal to CHB comparison value or capture occurred, interrupt enabled	RW	0x0
2	CMPAIE	**STMR1 count value equal to CHA comparison value/occurrence capture interrupt enable** 0x0: Count value equal to CHA comparison value or capture occurred, interrupt is not enabled 0 x1: count value is equal to the CHA to compare value or capture, interrupts enabled	RW	0x0

1	UDIE	**STMR1 count value equal to 0 Interrupt enable** 0x0: Interrupts with a count equal to 0 are not enabled 0 x1: count value is equal to zero interrupt enabled **Note:** Sawtooth wave mode counts up without 0 interrupt	RW	0x0
0	OVIE	**STMR1 count value equal to cycle interrupt enable** 0 x0: count value is equal to the periodic interrupt is not enabled 0 x1: count value is equal to the periodic interrupts enabled **Note:** Sawtooth wave mode counts down without cycle interruption	RW	0x0

14.4.18. STMR1_SR

Addr = 0xDA (SFR)

Bit(s)	Name	Description	R/W	Reset
7:6	-	-	-	-
5	BRAKEBIF	**STMR1 CHB brake break sign** 0x0: CHB valid braking did not occur 0x1: CHB effective braking has occurred **Note:** Write 1 to clear, write 0 to invalidate. Read as flag status.	RW	0x0
4	BRAKEAIF	**STMR1 CHA brake break sign** 0x0: CHA valid braking did not occur 0x1: CHA valid braking has occurred **Note:** Write 1 to clear, write 0 to invalidate. Read as flag status.	RW	0x0

3	CMPBIF	**STMR1 count value equal to CHB comparison value/occurrence of capture interrupt flag** 0x0: The count value is equal to the CHB comparison value or the capture did not occur 0x1: Count value equal to CHB comparison value or capture has occurred **Note:** write 1, 0 is invalid. Read as flag status.	RW	0x0
2	CMPAIF	**STMR1 count value equal to CHA comparison value/occurrence capture interrupt flag** 0x0: The count value is equal to the CHA comparison value or the capture did not occur 0 x1: count value is equal to the CHA to compare value or capture has occurred **Note:** Write 1 to clear, write 0 to invalidate. Read as flag status.	RW	0x0
1	UDIF	**STMR1 count value is equal to zero interrupt flag** 0x0: A count equal to 0 did not occur 0x1: Count value equal to 0 has occurred **Note:** Write 1 to clear, write 0 to invalidate. Read as flag status.	RW	0x0
0	OVIF	**The STMR1 count value is equal to the cycle interruption flag** 0x0: The count value is equal to the cycle did not occur 0x1: The count is equal to the cycle has occurred **Note:** Write 1 to clear, write 0 to invalidate. Read as flag status.	RW	0x0

14.4.19. STMR2_CR

Addr = 0xE4 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:6	-	-	-	-
5	CAPTMR1	**STMR2 Capture STMR1 brake filter** 0x0: STMR2 does not capture STMR1 brake filter control 0x1: STMR2 captures STMR1 brake filter control **Note:** After capture, STMR2 brake filter and STMR1 brake filter Settings are synchronized.	RW	0x0
4	SELSREG	**STMR2 Read shadow register control** 0x0: Read register PR/CMPA/CMPB to get the value of register PR_S/CMPA_S/CMPB_S 0 x1: read the register PR/CMPA CMPB get PR/CMPA CMPB register values	RW	0x0
3	DIR	**STMR2 Count directions** 0x0: Count down 0x1: Count up	RW	0x1
2:1	MODE	**STMR2 counter count mode** 0x0: Sawtooth Wave count mode 0x1: Triangle Wave A count mode 0x2: Triangle Wave B count mode 0x3: Hold	RW	0x0
0	TMREN	**STMR2 count enabled** 0x0: STMR2 count is not enabled 0x1: STMR2 count is enabled	RW	0x0

14.4.20. STMR2_FCONR

Addr = 0xE5 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:6	INCSEL	STMR2 count source 0x0: SYS_CLK 0x1: rising edge of PIN_SEL 0x2: PIN_SEL falling edge 0x3: PIN_SEL edge (rising and falling)	RW	0x0
5:4	PINSEL	STMR2 Count clock source 0x0: XOSCM/2 0x1:32KHz RC 0x2: CHA input 0x3: CHB input	RW	0x0
3:0	PREDIV	STMR2 counts the predivision frequency 0 to 15 corresponds to 1 to 16 dividers	RW	0x0

14.4.21. STMR2_PRL

Addr = 0xDD (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	STMR2PRL	STMR2 cycle register 8 bits lower	RW	0xFF

14.4.22. STMR2_PRH

Addr = 0xDE (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	STMR2PRH	STMR2 cycle register is 8 bits high	RW	0xFF

14.4.23. STMR2_CMPAL

Addr = 0xDF (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	STMR2CMPAL	STMR2 CHA compare value register 8 bits lower	RW	0x0

14.4.24. STMR2_CMPAH

Addr = 0xE1 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	STMR2CMPAH	STMR2 CHA is 8 bits higher than the value register	RW	0x0

14.4.25. STMR2_CMPBL

Addr = 0xE2 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	STMR2CMPBL	STMR2 CHB comparison value register 8 bits lower	RW	0x0

14.4.26. STMR2_CMPBH

Addr = 0xE3 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	STMR2_CMPBH	STMR2 CHB is 8 bits higher than the value register	RW	0x0

14.4.27. STMR2_VPERR

Addr = 0xE6 (SFR)

Bit(s)	Name	Description	R/W	Reset
7	BRAKEASF	**STMR2 CHA software brake signal** 0x0: CHA software brake signal is invalid 0x1: CHA software brake signal is valid	RW	0x0
6:5	BRAKEAVAL	**STMR2 CHA brake output values** 0x0: CHB PWM output 0 when the brake event is valid 0x1: CHB PWM output 1 when brake event is valid 0x2: CHB PWM output is off when brake event is active 0x3: When the brake event is effective, the CHB PWM output is turned off	RW	0x0
4:3	PCNTE	**STMR2 cycle interval response counting conditions** 0x0: The cycle interval function is invalid 0x1: Sawtooth wave up or down overflow point or triangle wave valley 0x2: sawtooth wave up or underflow point or triangle wave peak 0x3: Sawtooth wave upper or underflow point or triangle wave peak and trough	RW	0x0
2:0	PCNTS	**STMR2 cycle interval response time** 0x0:1 response per cycle 0x1:2 cycles respond once 0x2: one response in 4 cycles 0x3: one response in eight cycles 0x4: one response in 16 cycles 0x5: one response in 32 cycles 0x6: one response in 64 cycles	RW	0x0

		0x7: one response in 128 cycles		

14.4.28. STMR2_DTUA

Addr = 0xE7 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	STMR2_DTUA	**STMR2 dead time setting register**	RW	0x0

14.4.29. STMR2_BRAKE

Addr = 0xE8 (SFR)

Bit(s)	Name	Description	R/W	Reset
7	T1BMOE	**Normal PWM output signs after STMR2 CHB brake event** 0x0: CHB PWM output according to brake configuration 0x1: CHB PWM returns to normal output **Note: When the brake event** is effective, it will be immediately reset synchronously; When the effective brake signal disappears, according to the choice of T1BAOE, it will be set by software or automatically set by hardware.	RW	0x1
6	T1BAOE	**STMR2 CHB PWM normal output flag control bit** 0x0: T1BMOE can only be software set to 1 after a valid brake event 0x1: T1BMOE can be set 1 by software and overflow after a valid brake event	RW	0x0
5	T1BSEL	**STMR2 CHB brake source** 0x0: CHB selects analog comparator 0 output	RW	0x0

		as brake source 0x1: CHB selects analog comparator 1 output as brake source		
4	T1BEN	**STMR2 CHB brake enabled** 0x0: CHB brake is not enabled 0x1: CHB brake enabled	RW	0x0
3	T1AMOE	**After STMR2 CHA braking event, PWM output signs normally** 0x1: CHA PWM returns to normal output 0x0: CHA PWM output according to brake configuration **Note: When the brake event** is valid, it will be immediately reset synchronously; When the effective brake signal disappears, according to the choice of T1AAOE, the software set 1 or the hardware set 1 automatically.	RW	0x1
2	T1AAOE	**STMR2 CHA PWM normal output flag control bit** 0x0: T1AMOE can only be software set to 1 after a valid brake event 0x1: T1AMOE can be set to 1 by software and overflow after a valid braking event	RW	0x0
1	T1ASEL	**STMR2 CHA Brake source** 0x0: CHA selects the analog comparator 0 output as the brake source 0x1: CHA selects analog Comparator 1 output as brake source	RW	0x0
0	T1AEN	**STMR2 CHA brake enabled** 0x0: CHA brake is not enabled 0x1: CHA brake enabled	RW	0x0

14.4.30. STMR2_DTR

Addr = 0xE9 (SFR)

Bit(s)	Name	Description	R/W	Reset
7	BRAKEBSF	**STMR2 CHB software brake signal** 0x0: CHB software brake signal is invalid 0x1: CHB software brake signal is valid	RW	0x0
6:5	BRAKEBVAL	**STMR2 CHB brake output value** 0x0: CHB PWM output 0 when the brake event is valid 0x1: CHB PWM output 1 when brake event is valid 0x2: CHB PWM output is off when brake event is active 0x3: CHB PWM output is off when brake event is active	RW	0x0
4	HWCPWM	**Hardware set CMPB_S register enabled in STMR2 complementary mode** 0x0: The CMPB_S register is not enabled by the hardware setting 0x1: CMPB_S register is enabled by hardware setting	RW	0x0
3	DTBHO	**CHB dead-zone output in STMR2 complementary mode** 0x0: CHB PWM dead-zone normal output 0x1: CHB PWM dead-time output is off	RW	0x0
2	DTBEN	**STMR2 CHB dead zone enabled** 0x0: Dead zone setting is invalid 0x1: The dead zone setting is valid	RW	0x0
1	DTAHO	**CHA dead-zone output in STMR2 complementary mode** 0x1: CHA PWM dead-time output is off 0x0: CHA PWM dead-time output is normal	RW	0x0

0	DTAEN	**STMR2 CHA dead zone enabled** 0x0: Dead zone setting is invalid 0x1: The dead zone setting is valid	RW	0x0

14.4.31. STMR2_PCONRA

Addr = 0xEA (SFR)

Bit(s)	Name	Description	R/W	Reset
7	PAINITVAL	**STMR2 CHA PWM initial output values** 0x0: The initial PWM state is 0 after CHA count is enabled 0x1: Initial PWM state is 1 after CHA count is enabled	RW	0x0
6:5	CMPAVAL	**STMR2 CHA PWM output values** 0x0: The count value is less than the CHA comparison value output 1 and greater than the output 0 0x1: The count is greater than the CHA comparison output 1, less than the output 0 0x2: The count value is equal to the CHA comparison value and the output is flipped 0x3: Output remains the same	RW	0x0
4	PAENO	**STMR2 CHA PWM output enabled** 0x0: CHA PWM output is not enabled 0x1: CHA PWM output is enabled	RW	0x0
3	PAFILTEREN	**STMR2 CHA input filtering enabled** 0x0: CHA input signal is not filtered 0x1: CHA input signal filtered	RW	0x0
2:1	CAPAMODE	**STMR2 CHA capture point selection** 0x0: No capture 0x1: Capturing the rising edge 0x2: Capture the falling edge	RW	0x0

		0x3: Capture edges (rising and falling edges)		
0	CAPAEN	**STMR2 CHA capture mode enabled** 0x0: Capture mode is not enabled 0x1: Capture mode is enabled	RW	0x0

14.4.32. STMR2_PCONRB

Addr = 0xEB (SFR)

Bit(s)	Name	Description	R/W	Reset
7	PBINITVAL	**STMR2 CHB PWM initial output value** 0x0: The initial PWM state is 0 after CHB count is enabled 0x1: PWM initial state is 1 after CHB count is enabled	RW	0x0
6:5	CMPBVAL	**STMR2 CHB PWM output value** 0x0: The count value is less than the CHB comparison value output 1, greater than the output 0 0x1: The count value is greater than the CHB comparison value output 1, less than the output 0 0x2: The count value is equal to the CHB comparison value, the output is flipped 0x3: Output remains the same	RW	0x0
4	PBENO	**STMR2 CHB PWM output enabled** 0x0: CHB PWM output is not enabled 0x1: CHB PWM output enabled	RW	0x0
3	PBFILTEREN	**STMR2 CHB input filter enabled** 0x0: CHB input signal is not filtered 0x1: CHB input signal filtered	RW	0x0
2:1	CAPBMODE	**STMR2 CHB capture point selection**	RW	0x0

		0x0: No capture 0x1: Capturing the rising edge 0x2: Capture the falling edge 0x3: Capture edges (rising and falling edges)		
0	CAPBEN	**STMR2 CHB capture mode enabled** 0x0: Capture mode is not enabled 0x1: Capture mode is enabled	RW	0x0

14.4.33. STMR2_IE

Addr = 0xEC (SFR)

Bit(s)	Name	Description	R/W	Reset
7:6	-	-	-	-
5	BRAKEBIE	**STMR2 CHB brake interrupt enabled** 0x0: CHB brake interrupt is not enabled 0x1: CHB brake interrupt enabled	RW	0x0
4	BRAKEAIE	**STMR2 CHA brake interrupt enabled** 0x0: CHA brake interruption is not enabled 0x1: CHA brake interrupt enabled	RW	0x0
3	CMPBIE	**STMR2 CHB count value equal to the comparison value/capture interrupt enable occurs** 0x0: The count value is equal to the CHB comparison value or capture occurs, interrupt is not enabled 0x1: Count value equal to CHB comparison value or capture occurred, interrupt enabled	RW	0x0
2	CMPAIE	**STMR2 CHA count value equal to the comparison value/capture interrupt enable occurs** 0x0: Count value equal to CHA comparison	RW	0x0

		value or capture occurs, interrupt is not enabled 0x1: Count value equal to CHA comparison value or capture occurred, interrupt enabled		
1	UDIE	**STMR2 count value equal to 0 Interrupt enable** 0x0: Interrupts with a count equal to 0 are not enabled 0x1: Count value equal to 0 interrupt enable **Note:** Sawtooth wave mode counts up without 0 interrupt	RW	0x0
0	OVIE	**STMR2 count value equal to cycle interrupt enable** 0x0: Count value equal to cycle interrupt not enabled 0x1: Count value equal to cycle interrupt enabled **Note:** Sawtooth wave mode counts down without cycle interruption	RW	0x0

14.4.34. STMR2_SR

Addr = 0xED (SFR)

Bit(s)	Name	Description	R/W	Reset
7:6	-	-	-	-
5	BRAKEBIF	**STMR2 CHB brake interrupt flag** 0x0: CHB valid braking did not occur 0x1: CHB valid braking has occurred **Note:** Write 1 to clear, write 0 to invalidate. Read as flag status.	RW	0x0
4	BRAKEAIF	**STMR2 CHA brake break sign**	RW	0x0

		0x0: CHA valid braking did not occur 0x1: CHA valid braking has occurred **Note:** Write 1 to clear, write 0 to invalidate. Read as flag status.		
3	CMPBIF	**The STMR2 CHB count value is equal to the comparison value/capture interrupt flag occurs** 0x0: CHB count equal to comparison value or capture did not occur 0x1: CHB count equal to comparison value or capture has occurred **Note:** Write 1 to clear, write 0 to invalidate. Read as flag status.	RW	0x0
2	CMPAIF	**STMR2 CHA count value equal to comparison value/occurrence capture interrupt flag** 0x0: CHA count value equal to comparison value or capture did not occur 0x1: CHA count equal to comparison value or capture has occurred **Note:** Write 1 to clear, write 0 to invalidate. Read as flag status.	RW	0x0
1	UDIF	**The STMR2 count value is equal to 0 interrupt flag** 0x0: Count equal to 0 did not occur 0x1: Count equal to 0 has occurred **Note:** Write 1 to clear, write 0 to invalidate. Read as flag status.	RW	0x0
0	OVIF	**The STMR2 count value is equal to the cycle interruption flag** 0 x0: count value is equal to the cycle does not take place 0x1: The count is equal to the cycle has occurred **Note:** Write 1 to clear, write 0 to	RW	0x0

		invalidate. Read as flag status.		

14.4.35. STMR_ALLCON

Addr = 0xF5 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:6	-	-	-	-
5	SCLRR	**STMR1 and STMR2 counts are zeroed synchronously** Write 1 to clear, write 0 to invalidate	WO	0x0
4	SSTPR	**STMR1 and STMR2 stop counting in sync** Write 1 to be valid, write 0 to be invalid	WO	0x0
3	SSTAR	**STMR1 and STMR2 start counting synchronously** Write 1 is valid, write 0 is invalid	WO	0x0
2	TMR2CLRCNT	**TMR2 count clear** Write 1 to clear, write 0 to invalidate	WO	0x0
1	TMR1CLRCNT	**TMR1 count clear to zero** Write 1 to clear, write 0 to invalidate	WO	0x0
0	TMR0CLRCNT	**TMR0 count clear to zero** Write 1 to clear, write 0 to invalidate	WO	0x0

14.5. Instructions for using the process

- Select counting mode, counting direction;
- Select counting clock source (PIN_SEL), select counting mode (rising edge, falling edge) INC_SEL;
- Configure frequency division register, cycle register;
- Choose whether to open capture mode or PWM output mode;
- If you choose capture mode, configure the capture mode related registers;
- If PWM output mode is selected, configure the comparator register and other

PWM output related registers, including dead-time and brake related registers.

- Count enable position 1;

15. CRC16 module

15.1. Feature Overview

CRC16 functional features are as follows:

- Support CRC16 - CCITT - FALSE/CRC16 - XMODEM is calculated
- 1byte data is calculated per system clock cycle
- It supports changing the initial value to realize the support of different CRC protocols

Note: When FLASH uses the CRC check code, the CRC module is temporarily unavailable

15.2. The basic function

The CRC16-CCITT-FALSE protocol is x^16+x^12+x^5+1, the initial value is 0xffff, the input data is not negated, the output data is not negated, and the output data XOR 0x0.

CRC16-XMODEM protocol is x^16+x^12+x^5+1, the initial value is 0x0, the input data is not negated, the output data is not negated, the output data XOR 0x0.

CRC16 module does parallel CRC calculation of 1byte data without input once, which greatly improves the operation speed compared with serial calculation of CRC. Note that when using CRC16 module, you need to use CRC_REG register to configure the initial value of CRC16 calculation each time you use the module, and the initial value is 16 bits. So it needs to be configured twice, the initial value can only be 0xffff, 0x0, at the same time, the result of CRC16 operation is 16 bits, so the CRC_FIFO register needs to be read twice when reading the result, and the two read values are concatenated into a 16 bit result is the real result of CRC16.

15.3. Block diagram of modules

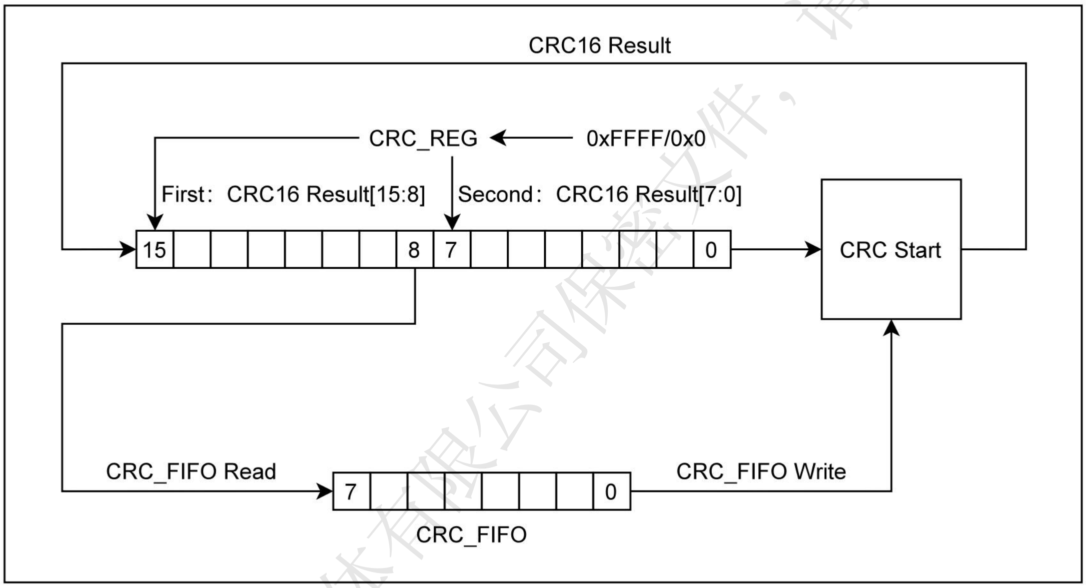


Figure 15.1 Block diagram of the CRC16 structure

15.4. The register list

Table 15-1 CRC register list

Address	Register Name	Description
0x9E (SFR)	CRC_REG	CRC initial register
0x9F (SFR)	CRC_FIFO	CRC data fifo register

15.5. Register details

15.5.1. CRC_REG

Addr = 0x9E (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	INITSET	**CRC initial value configuration** It is necessary to set the initial value of CRC check before each use. The default is 0xffff when power on, and the initial value needs to be written twice	W	0xFF

15.5.2. CRC_FIFO

Addr = 0x9F (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	DATA	**CRC Data configuration** Write: 1byte data for each CRC check Read: Read the register once to read the 1byteCRC result, if you need to read the CRC16 send over need to read twice to spell 16bit	RW	-

15.6. Using the process that

- Configure CRC_REG, generally 0xffff/0x0000 (you need to write this register twice)
- The need to check 1 byte data through CRC_FIFO this register to write in
- When you need to read the result, read the CRC_FIFO twice to form a CRC16

result

16. FLASH Controller Module

16.1. Feature Overview

Logic Flash (hereinafter referred to as "Flash") functional features are as follows:

- Flash controller with operation protection function (need to configure password register to enable before Flash operation)
- Can be achieved through configuration register to burn the Flash memory (128 byte) erase/full/sectors

Slice (4K) erase function, while supporting online burning

- By configuring the register, it can realize the CRC check of the Flash memory data
- It supports the use of EEPROM-like

16.2. The module block diagram

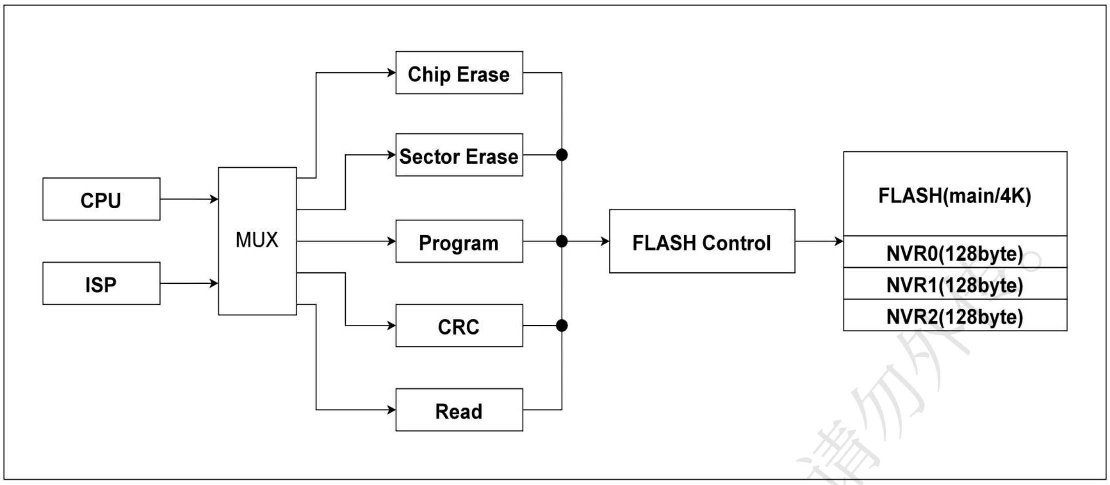


Figure 16-1 FLASH structure block diagram

16.3. List of registers

Table 16-1 FLASH register list

Address	Register Name	Description
0xA0 (SFR)	FLASH_CON	FLASH control register
0xA1 (SFR)	FLASH_STA	FLASH state register
0xA2 (SFR)	FLASH_DATA	FLASH program data register
0xA3 (SFR)	FLASH_TIM0	FLASH timing control register 0
0xA4 (SFR)	FLASH_TIM1	FLASH timing control register 1
0xA5 (SFR)	FLASH_CRCLEN	FLASH CRC data length register
0xA6 (SFR)	FLASH_PASSWORD	FLASH operation to protect register
0xA7 (SFR)	FLASH_ADDR	FLASH program/erase address register
0xAA (SFR)	FLASH_TRIM	FLASH test mode register

16.4. Register details

16.4.1. FLASH_CON

Addr = 0xA0 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:4	-	-	-	-
3	CRCST	**FLASH CRC function triggered** Write 1 to trigger CRC check, you need to configure the size of FLASH_CRCLEN to trigger this operation, and you can't run CRC code in the DATA area	WO	0x0
2	CERST	**FLASH full erase function triggered** Write 1 to trigger full sheet erase	WO	0x0
1	SERST	**FLASH sector erase function triggered** Write 1 to trigger sector erase, you need to configure 128byte aligned address to trigger	WO	0x0
0	PROGST	**FLASH burn function triggered** Write 1 Trigger the burn operation	WO	0x0

Note: The FLASH controller has the operation protection function, and the FLASH_PASSWORD register needs to be configured correctly before the above operation is triggered correctly.

16.4.2. FLASH_STA

Addr = 0xA1 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:5	-	-	-	-
4	NVRLOCK	**FLASH's information area sign signal** 0x0: The information area failed the	RO	0x1

		inspection (does not work properly) 0x1: information area passed validation (can be used)		
3	CRCPEND	**CRC mode work flag bit** 0x0: CRC checking is in progress 0x1: Idle state	RO	0x1
2	CERPEND	**Full film erase mode work logo** 0x0: Full chip erase is in progress 0x1: Idle state	RO	0x1
1	SERPEND	**Sector erase mode work flag** 0x0: Sector erasing is in progress 0x1: Idle state	RO	0x1
0	PROGPEND	**Burn mode work flag** 0x0: Burning is in progress 0x1: Idle state	RO	0x1

16.4.3. FLASH_DATA

Addr = 0xA2 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	DATA	**FLASH burns the data** The data of FLASH memory is 16 bits, and the register needs to be written twice when it needs to be operated. The first time is to write the lower 8 bits, and the second time is to write the higher 8 bits	RW	0x0

16.4.4. FLASH_TIMO

Addr = 0xA3 (SFR)

Bit(s)	Name	Description	R/W	Reset

7:6	TERH	Sector erase timing control (default time is 50ms) 0x0: Default value 0x1:51ms 0x2:52ms 0x3:53ms	RW	0x0
5:4	TPOAM	Post-sync signal Timing control signal (default time is 2us) 0x0: Default value 0x1:3us 0x2:4us 0x3:5us	RW	0x0
3:2	TPRAM	Presync signal timing control signal (default time is 2us) 0x0: Default value 0x1:3us 0x2:4us 0x3:5us	RW	0x0
1:0	TPGHF	Burn timing control signal (default time is 20us) 0x0: Default value 0x1:21us 0x2:22us 0x3:23us	RW	0x0

16.4.5. FLASH_TIM1

Addr = 0xA4 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:6	TPEL	Continuous operation timing control (default time is 2*Tsys) 0x0: Default value	RW	0x0

		0x1:3 *Tsys 0x2:4*Tsys 0x3:5*Tsys		
5:4	TCERH	**Full chip erasure timing control signal (default time is 50ms)** 0x0: Default value 0x1:51ms 0x2:52ms 0x3:53ms	RW	0x0
3:2	TKP	**TKP 2 Read periodic timing control signals** 0x0: Default value 0x1: Tsys 0x2:2*Tsys 0x3:3*Tsys	RW	0x0
1:0	TKH	**Read signal high level timing control signal** 0x0: Default value 0x1: Tsys 0x2:2*Tsys 0x3:3*Tsys	RW	0x0

Note: TKH should be configured with a value greater than or equal to TKP (see the specific configuration table for TKP and TKH configuration)

16.4.6. FLASH_CRCLEN

Addr = 0xA5 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	LEN	**Data length for CRC operation** The data size (in bytes) of the CRC check operation.	RW	0x0

Note: When FLASH_CRCLEN is configured with a value of 0, the CRC operation cannot be triggered!

16.4.7. FLASH_PSWD

Addr = 0xA6 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	PASSWORD	**FLASH operates protected password registers** The operation protects the password, which needs to be configured before performing the operation of FLASH_CON, and the password is 0xB9	WO	0x0

16.4.8. FLASH_ADDR

Addr = 0xA7 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	ADDR	**FLASH's address register** The address of FLASH memory is 16 bits, and the effective bit is 12 bits. When you need to operate, you need to write to the register twice, the first time to write the high 8 bits, and the second time to write the low 8 bits	WO	0x0

Note:

1. When doing FLASH sector erase, the address needs to be aligned (128 bytes), otherwise the operation cannot be carried out.

2. When FLASH burning, FLASH Control has base address offset (offset is 0x8000), the conversion formula is as follows:

FLASH_ADDR = (W_ADDR - 0x8000)/2

Where W_ADDR is the Flash physical address, FLASH_ADDR is the register, the Flash bit width is 16bit, so the address needs to be divided by 2.

Suppose the address to be stored is 0x9000, then FLASH_ADDR = (0x9000-0x8000)/2=0x800.

16.4.9. FLASH_TRIM

Addr = 0xAA (SFR)

Bit(s)	Name	Description	R/W	Reset
7:6	-	-	-	-
5	MODESEL	**MODESEL signal in test mode**	RW	0x0
4:3	TRIM	**TRIM signal in test mode**	RW	0x2
2:1	VRDCGSEL	**VRDCGSEL signal in test mode**	RW	0x1
0	TRF	**TRF signal in test mode**	RW	0x0

Note: under normal use don't need to configure the register (only for the use of the test, use can cause problems in the test mode)

16.5. Instructions for Use

- Configure the address of the FLASH that needs to be operated (align the corresponding address to trigger the corresponding operation)
- Configure the data of the FLASH that needs to be operated
- Configure FLASH_PASSWORD register to enable
- Configure FLASH_CON to trigger the corresponding FLASH operation
- Wait until the pending corresponding to FLASH_STA is high to end a FLASH operation

17. Analog-to-digital Converter (ADC)

17.1. Feature Overview

The module is a 12bit successive approximation ADC controller. The ADC supports a variety of operation modes: single conversion/two channel alternating trigger, all GPIO and internal channels can be selected for conversion, ADC startup includes software startup, and advanced Timer1/2 trigger startup.

The input clock of the ADC must not exceed 4Mhz, which is generated by the system clock through frequency division.

- 12 bit ADC with quantitative, optional internal or external reference voltage reference voltage for the ADC reference

Voltage, internal reference voltage is 1.2V and 2.4V optional; External reference voltage is VCC by default

- The Analog test signal can be input to any I/O port through the path selection of the ADC
- The conversion speed can reach 200Ksps
- Sampling time is adjustable, adjustable range: 5 ~ 256 ADC clock
- Support single channel configuration delay trigger
- Comparator offset can be calibrated using analog/digital
- Supports dual channels with arbitration trigger (hardware delay feature is not enabled on either channel)
- 1 way software trigger and 7 way hardware trigger are supported
- All GPIO and internal channels are supported as analog conversion channels

17.2. Basic Features

17.2.1. Single channel trigger mode

The single channel trigger mode is to enable only CHAN0EN/CHAN1EN in ADC_CFG0. If only channel 0 is enabled, only the trigger source of channel 0 of ADC_CHS0 configured can trigger the conversion of ADC, and the ADC uses the analog conversion channel of ADC_CHS0 at this time. Of course, only channel 1 is enabled.

17.2.2. Dual channel trigger mode

In the dual channel can trigger mode is also ADC_CFG0 CHAN0EN/CHAN1EN, in this mode, channel 0 priority is higher than 1 priority, in this time will appear in the following two cases:

- Arbitration: When ADC_CHS0 and ADC_CHS1 configuration of the trigger source to trigger an ADC at the same time, the first response ADC_CHS0 configuration trigger source, at the same time, the corresponding simulation in adopting ADC_CHS0 sampling channel, again after the response of channel zero trigger response after 1 trigger, Response of channel 1 in ADC_CHS1 trigger when use the corresponding simulation sampling channel
- Line: Transformed the channel 0, if again channel 0 trigger at this time will be ignored, not response won't latch, if at this time to channel 1 will latch to trigger the request of channel 1, wait until the channel 0 transformation after the conversion of channel 1, again, of course, the channel 1 conversion is also in the same way, to channel 1 trigger will be ignored, But if the channel zero trigger, corresponding request will latch, wait until after the conversion

channel 1 to channel 0.

17.2.3. Single channel trigger delay model

Single channel of trigger delay model is only enabled ADC_CFG0 CHAN0EN & DLY0EN/CHAN1EN & DLY1EN, in this mode when the channel 0 after the trigger source to the ADC_CHS0 configuration, not immediately start the ADC conversion, Need to wait for ADC_CFG3 configured in delay time before going to convert the ADC, at the same time, in this mode, the ADC two channel no arbitration mechanism, so in this mode is not recommended for switching to another channel.

17.2.4. Analog/digital calibration alignment

Simulation calibration needs to make to CALIBEN ADC_CFG0, then channel 0 configured to single channel trigger mode, a software when triggered, will transform the results after low six wrote comp_trim_vdd, then closed CALIBEN, and clear channel 0 in ADC_STA complete marks and analog calibration, Complete the analog calibration.

Digital calibration only need ADC_CFG1 CPCALIBST, waiting for the number in the ADC_STA calibration after the sign bit, clear digital calibration after mark a complete digital calibration.

17.3. The module block diagram

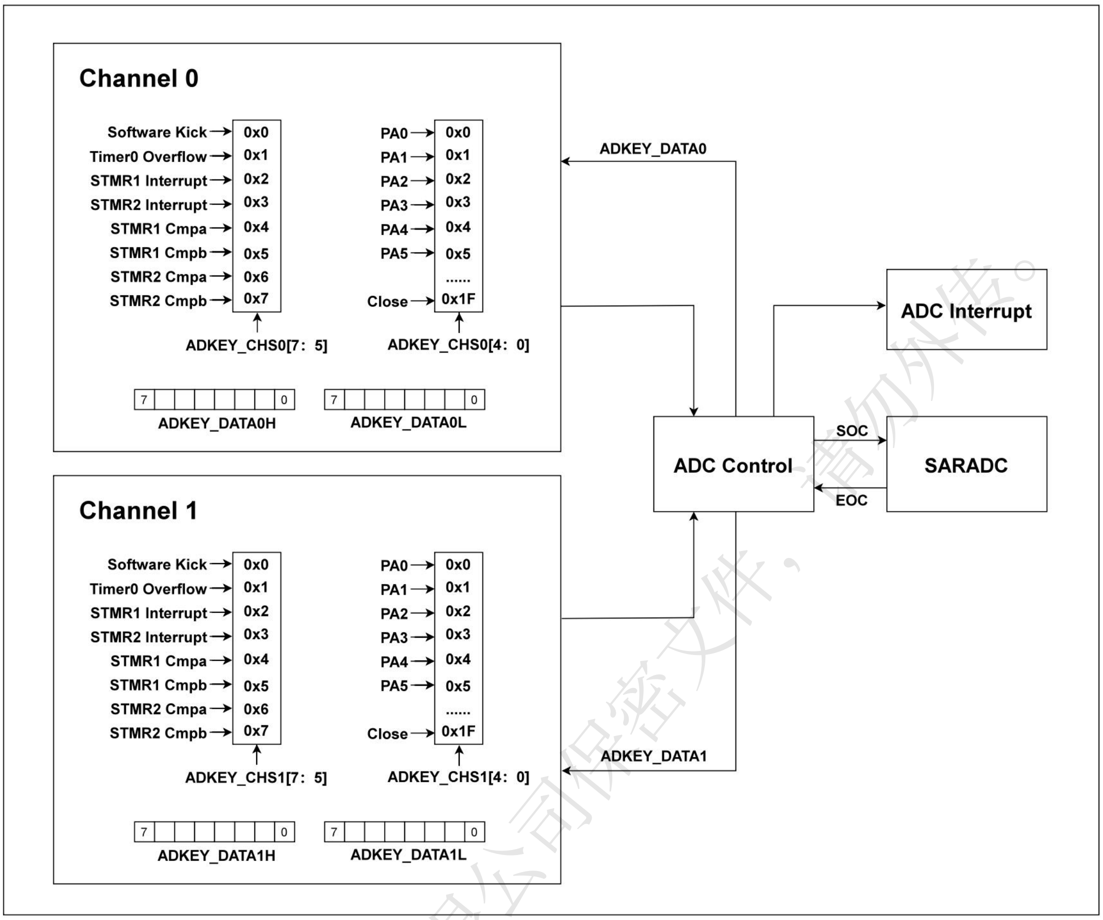


Figure 17-1 ADC structure diagram

17.4. List of registers

Table 17-1 ADC register list

Address	Register Name	Description
0x91 (SFR)	ADC_CFG0	ADC configuration 0 register
0x92 (SFR)	ADC_CFG1	ADC configuration 1 register
0x9A (SFR)	ADC_CFG2	ADC configuration 2 register
0xEE (SFR)	ADC_CFG3	ADC configuration 3 register
0x93 (SFR)	ADC_STA	ADC state register

0x94 (SFR)	ADC_DATAH0	ADC channel0 data high 8bit register
0x95 (SFR)	ADC_DATAL0	ADC channel0 data low 4bit register
0x96 (SFR)	ADC_DATAH1	ADC channel1 data high 8bit register
0x97 (SFR)	ADC_DATAL1	ADC channel1 data low 4bit register
0x98 (SFR)	ADC_CHS0	ADC channel0 select register
0x99 (SFR)	ADC_CHS1	ADC channel1 select register

17.5. Register details

17.5.1. ADC_CFG0

Addr = 0x91 (SFR)

Bit(s)	Name	Description	R/W	Reset
7	CALIBEN	Enable hardware calibration Turn on channel 0 as calibration result, fill back comp_trim_vdd	RW	0x0
6	DLY1EN	Channel 1 ADC trigger delay function enabled 0x0: Off 0x1: Open	RW	0x0
5	DLY0EN	Channel 0 ADC trigger delay function enabled 0x0: Off 0x1: Open	RW	0x0
4	ADCEN	A/D conversion enabled 0x0: Off 0x1: Open	RW	0x0
3	CHAN1EN	Channel 1 Convert enable 0x0: Off 0x1: Open	RW	0x0
2	CHAN0EN	Channel 0 conversion enabled 0x0: Off	RW	0x0

		0x1: Open		
1	ADST1	**ADC channel 1 triggers conversion** Write 1 began to trigger the channel 1	WO	0x0
0	ADST0	**ADC channel 0 triggers conversion** Write 1 to start triggering channel 0 for conversion	WO	0x0

Note: The trigger delay function does not come with arbitration, so you cannot use both channels for conversion when using the trigger delay function!

17.5.2. ADC_CFG1

Addr = 0x92 (SFR)

Bit(s)	Name	Description	R/W	Reset
7	-	-	-	-
6	CPCALIBST	**Digital calibration function trigger bit** When the digital calibration function is triggered, write 1 to trigger. After triggering, the ADC module will automatically calibrate. After calibration, the calibration value will be filled in to comp_trim_vdd	WO	0x0
5:2	ADCPRE	**ADC clock divider** When the frequency ratio is n+ 1,0, the default is 2 frequency, the maximum input clock frequency of ADC is 4MHz, Fadc=Fsys/ADCPRE, where Fadc is the input frequency of ADC, Fsys is the main frequency of the system clock 0x0:2 frequency division 0x1:2 split frequency 0x2:3 split frequency 0x3:4 split frequency	RW	0x7

		0x4:5 split frequency ... 0 xe: 15 points and frequency 0xF: split frequency 16		
1	CHAN1IE	**Interrupt enabled for ADC channel 1** 0x0: Off 0x1: Open	RW	0x0
0	CHAN0IE	**Interrupt enabled for ADC channel 0** 0x0: Off 0x1: Open	RW	0x0

17.5.3. ADC_CFG2

Addr = 0x9A (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	FSMPCYC	**Sampling time configuration** Configuration ratio is n+1 clock 0x0:1 ADC clocks 0x1:2 ADC clocks 0x2:3 ADC clocks ... 0 XFF: 256 ADC clock	RW	0x4

Note: In order for the ADC to work properly, the minimum sampling time is configured to be 0x3

17.5.4. ADC_CFG3

Addr = 0xEE (SFR)

Bit(s)	Name	Description	R/W	Reset
7:4	-	-	-	-

3:0	DLYCYC	**Trigger the delay time configuration** The configuration ratio is 4n+3 ADC clocks 0x0:3 ADC clocks 0x1:7 ADC clocks 0x2:11 ADC clocks ... 0 xf: 67 ADC clock	RW	0x0

17.5.5. ADC_STA

Addr = 0x93 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:5	-	-	-	-
4	EODTADCFLAG	**Digital calibration completion flag** 0x0: Digital calibration is not completed/not enabled 0x1: Digital calibration is complete Write 1 clear 0	RW	0x0
3	EOATADCFLAG	**Analog calibration completion flag** 0x0: Analog calibration not completed/not enabled 0x1: Simulation calibration is completed Write 1 clear 0	RW	0x0
2	CHAN1OVPEND	**Channel 1 Conversion completion flag** 0x0: Channel 1 conversion is not complete/Channel 1 is not enabled 0x1: Channel 1 conversion is complete Write 1 clear 0	RW	0x0
1	CHAN0OVPEND	**Channel 0 conversion complete flag** 0x0: Channel 0 conversion is not complete/channel 1 is not enabled 0x1: The channel 0 conversion is completed	RW	0x0

		Write 1 clear 0		
0	BUSY	ADC busy/idle flag 0x0: ADC idle 0x1: ADC in conversion	RO	0x0

17.5.6. ADC_DATAH0

Addr = 0x94 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	DATA0H	Channel 0's 12-bit A/D conversion results in 8 bits higher	RW	0x0

17.5.7. ADC_DATAL0

Addr = 0x95 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:4	-	-	-	-
3:0	DATA0L	Channel 0's 12-bit A/D conversion result is 4 bits lower	RW	0x0

17.5.8. ADC_DATAH1

Addr = 0x96 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:0	DATA1H	Channel 1 of 12 high 8-bit A/D conversion results	RW	0x0

17.5.9. ADC_DATAL1

Addr = 0x97 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:4	-	-	-	-
3:0	DATA1L	Channel 1's 12-bit A/D conversion result is 4 bits lower	RW	0x0

17.5.10. ADC_CHS0

Addr = 0x98 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:5	TRGSEL0	**Hardware trigger source selection** 0x0: Select software trigger 0x1: Timer0ov 0x2: Advanced Timer1 Interrupt 0x3: Advanced Timer2 interrupt 0x4: Advanced Timer1 CMPA_ACK 0x5: Advanced Timer1 CMPB_ACK 0x6: Advanced Timer2 CMPA_ACK 0x7: Advanced Timer2 CMPB_ACK	RW	0x0
4:0	CHANSEL0	**Analog conversion channel selection for channel 0** 0x0~0x7 select P00~P07 channel respectively, 0x8~0xD select P10~P15 respectively: 0x0: P00 0x1: P01 0x2: P02 0x3: P03 0x4: P04 0x5: P05	RW	0x1F

		0x6: P06 0x7: P07 0x8: reserved 0x9: P11 0xA: P12 0xB: P13 0xC: P14 0xD: P15 0xE: PTAT 0xF: VBG06_ADC 0x10: VCCA 5 partial pressure value 0x11: analog test out All other codes are not selected/reserved		

17.5.11. ADC_CHS1

Addr = 0x99 (SFR)

Bit(s)	Name	Description	R/W	Reset
7:5	TRGSEL1	**Hardware trigger source selection** 0x0: Select software trigger 0x1: Timer0ov 0x2: Advanced Timer1 Interrupt 0x3: Advanced Timer2 interrupt 0x4: Advanced Timer1 CMPA_ACK 0x5: Advanced Timer1 CMPB_ACK 0x6: Advanced Timer2 CMPA_ACK 0x7: Advanced Timer2 CMPB_ACK	RW	0x0
4:0	CHANSEL1	**Analog conversion channel selection for channel 1** 0x0~0x7 select P00~P07 channel respectively, 0x8~0xD select P10~P15 respectively:	RW	0x1F

		0x0: P00 0x1: P01 0x2: P02 0x3: P03 0x4: P04 0x5: P05 0x6: P06 0x7: P07 0x8: reserved 0x9: P11 0xA: P12 0xB: P13 0xC: P14 0xD: P15 0xE: PTAT 0xF: VBG06_ADC 0x10: VCCA 5 partial pressure value 0x11: analog test out All other codes are not selected/reserved		

17.6. Instructions for Using the process

- Configure the ADCPRE of ADC_CFG1 and set the ADC clock frequency division
- Configure ADC_CFG0 to enable channel conversion
- Configure ADC_CFG2 to configure the sampling time
- Configure ADC_CHS0/ADC_CHS1 to configure the analog conversion channel and trigger source
- Configure ADC_CFG0 to enable the ADC
- Wait 20us (no ADC conversion can be triggered during the waiting period) to trigger the ADC

- Wait for the pending of the corresponding channel in ADC_STA to be
- Conversion result of ADC_DATAH0/ADC_DATAH1

18. Analog comparator (CMP0/1)

18.1. Features Overview

The comparator features are as follows:

- DACMP mainly includes 2 8 bit flash Dacs, 2 comparators, 1 channel 20mA constant current source output
- The positive end of each comparator can choose two port (AIO) inputs and one PGA input, and the negative end can choose two port (AIO) inputs and DAC inputs. The positive end of comparator 0 also supports threshold short circuit (VCCA-VTH or PAD-VTH) protection input, and comparator 1 supports CCS sampling voltage input
- DAC reference voltage can be selected internal 1.2V reference, output is 1.2/240* (1~240)
- Built-in 2-channel threshold short circuit protection (optional VCCA-VTH or P03 PAD-VTH), in which the VTH gear adjustment output voltage can be selected as 80mv/200mv/320mv/480mv
- 1 channel 20mA constant current source output with 6bit calibration accuracy of 2.5%, TYP output range: -40.5~+42%
- Both can support P input pair tube or N input pair end input, support offset voltage calibration of P input tube, calibration step is 2mv under typ, calibration range is -13mv~+13mv

- Support digital filtering, filtering time a total of 32 stalls selection, step size of 1 microsecond
- Support digital hysteresis control, hysteresis sampling interval 16 gears are optional, step size 1 microsecond and 16 microsecond optional
- Digital hysteresis voltage can be selected in 1.2/240* (1~240)
- Support offset voltage software adjustment
- The output can be used as the brake trigger signal of enhanced PWM
- Support output change to generate interrupts
- Support for comparator to wake up from sleep state
- Comparator enable can be controlled by internal PWM

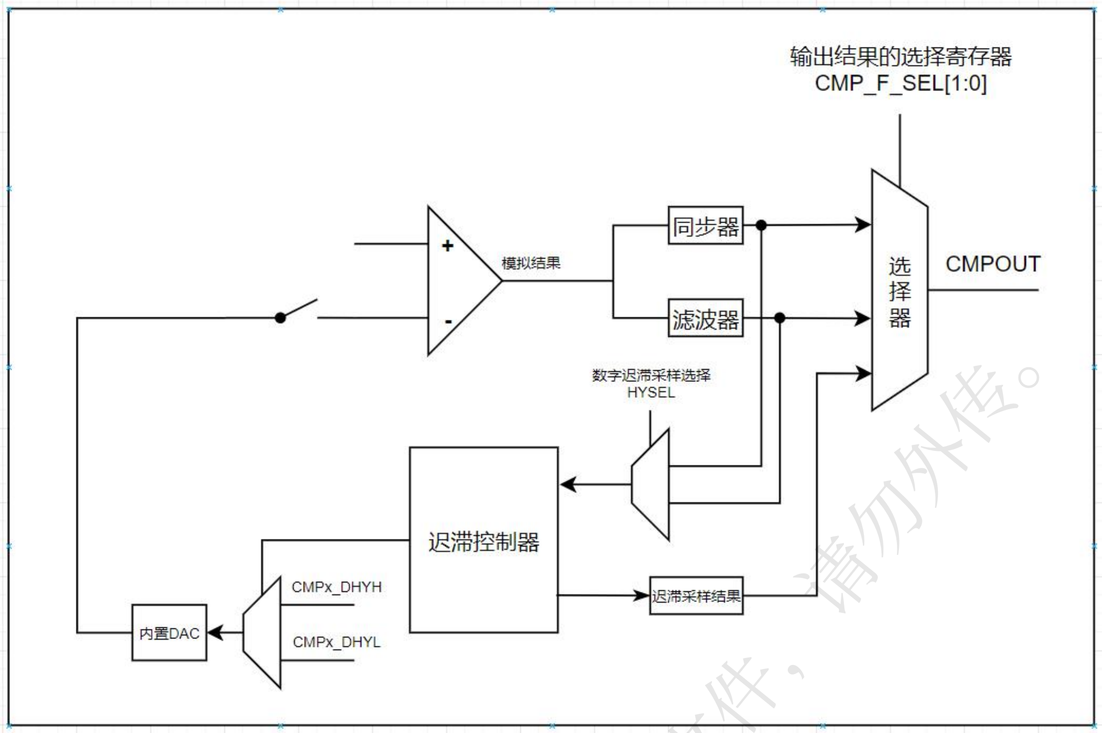
18.2. Functional Instructions

18.2.1. Description of digital filter function

Comparator, the logic of digital filter is: simulation of the comparator output change must continue after several cycles, the comparator digital results will change.

After the analog output of the comparator change, internal counter starting from 0 count, count each plus one sample an analog output, until the count to the configuration values (CMPx_CON2 [away]) or to find analog comparator output value has changed.

18.2.2. Description of digital hysteresis function



Comparator digital delay implementation approach is: sampling comparator output value, to determine the output is 0 or 1, choose according to judgment changed the DAC input (from CMPx_DHYH and choose a CMPx_DHYL, sampling to 0 select CMPx_DHYL, choose CMPx_DHYH when sampling to 1), so as to achieve the effect of hysteresis.

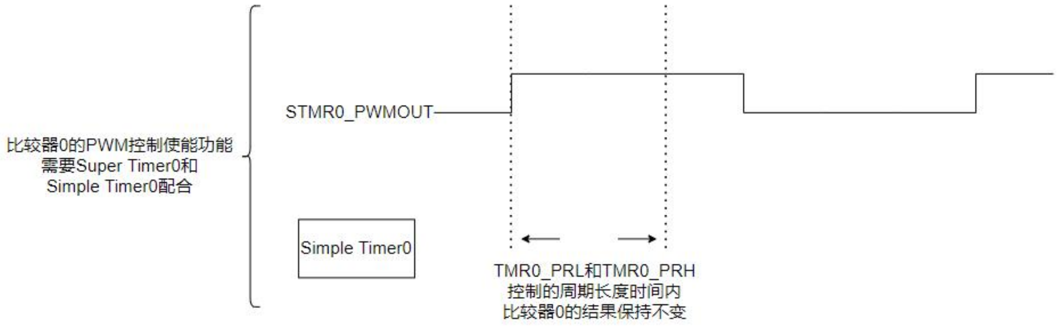
The hysteresis sampling interval can be configured (CMPx_CON3[3:0])

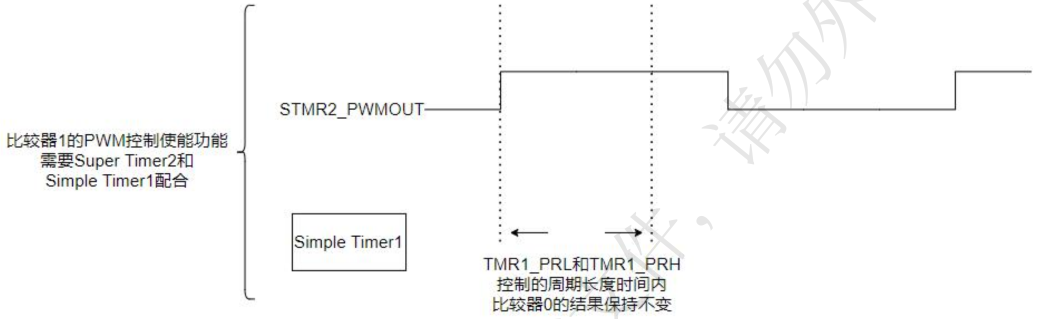
The result of hysteresis sampling can be used as the output of the comparator (CMP_F_SEL[1:0] option)

The hysteresis sampling can choose to acquire the result of the synchronizer or the result of the filter (HYSEL of CMPx_CON3)

18.2.3. The comparator can make PWM control

PWM control enabling is mainly applied to scenarios where PWM control power devices is required. When the PWM flip of the power device is controlled, the power supply may fluctuate greatly, and the results of the comparator may be unreliable. Therefore, the results of the comparator will remain unchanged for a period of time after the PWM flip. This period of time is timed by the timer, and the corresponding relationship is as follows:





When using PWM to control the enable mode, the Simple timer needs to turn on the counting mode. The counting period is the length of time that the comparator's result remains unchanged, after which the comparator's digital circuit will sample the analog comparator's result.

18.3. The module block diagram

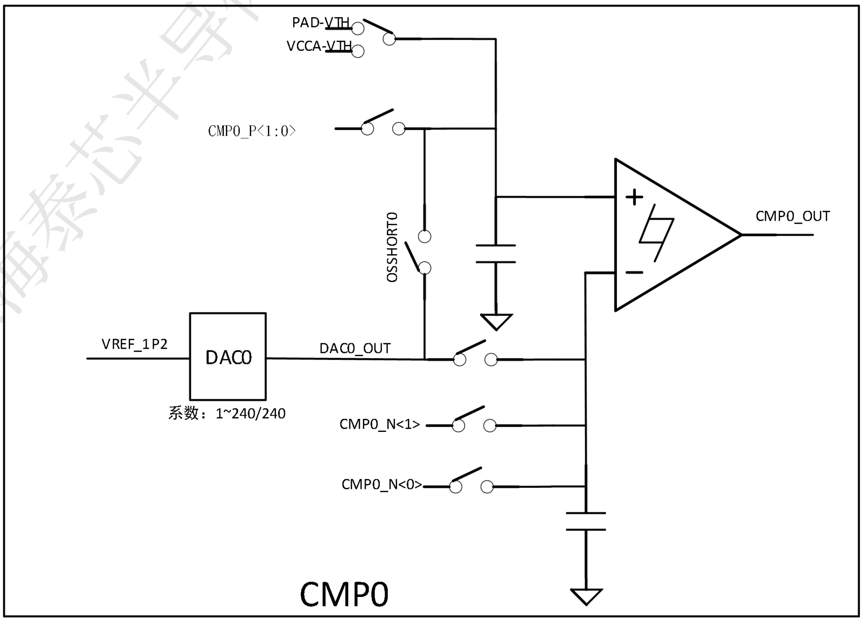


Figure 18.1 Internal block diagram of a comparator 0 simulation

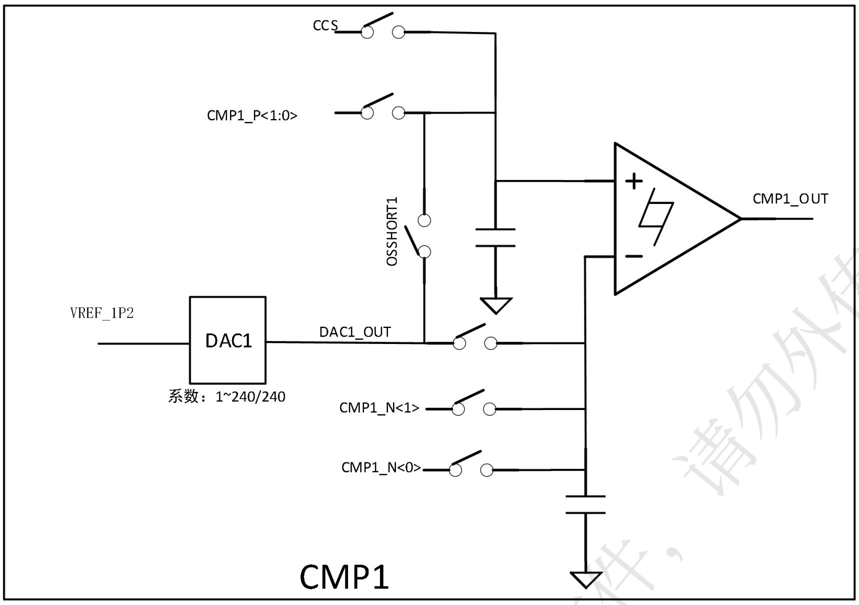


Figure 18.2 Internal block diagram of comparator 1 simulation

18.4. Pin correspondence table

Table 18-1 List of CMP registers

Input side		CMPx_CON1 [CHPSEL]	CMPx_CON1 [CHNSEL]	GPIO
CMP0	CMP0_N[0]	/	0x01	P02
	CMP0_N[1]	/	0x02	P14
	CMP0_P[0]	0x01	/	P11
	CMP0_P[1]	0x02	/	P12
CMP1	CMP1_N[0]	/	0x01	P00
	CMP1_N[1]	/	0x02	P01
	CMP1_P[0]	0x01	/	P03
	CMP1_P[1]	0x02	/	P13

Note: In the table CMPx_CON1 refers to CMP0_CON1 and CMP1_CON1

18.5. List of registers

Table 18-2 List of CMP registers

Address	Register Name	Description
0x36 (XSFR)	CMP0_CON0	CMP0 configuration 0 register
0x37 (XSFR)	CMP0_CON1	CMP0 configuration 1 register
0x38 (XSFR)	CMP1_CON0	CMP1 configuration 0 register
0x39 (XSFR)	CMP1_CON1	CMP1 configuration 1 register
0x3A (XSFR)	CMP_CON	CMP configuration common register
0x3B (XSFR)	CMP_STA	CMP status register
0x74 (XSFR)	CMP0_CON2	CMP0 configuration 2 register
0x75 (XSFR)	CMP0_CON3	CMP0 configuration 3 register
0x76 (XSFR)	CMP0_CON4	CMP0 configuration 4 register
0x77 (XSFR)	CMP0_DAC0	DAC0 register of CMP0
0x78 (XSFR)	CMP0_DAC1	DAC1 register of CMP0
0x79 (XSFR)	CMP1_CON2	CMP1 configuration 2 register
0x7A (XSFR)	CMP1_CON3	CMP1 configuration 3 register
0x7b (XSFR)	CMP1_CON4	CMP1 configuration 4 register
0x7C (XSFR)	CMP1_DAC0	DAC0 register of CMP1
0x7D (XSFR)	CMP1_DAC1	DAC1 register of CMP1

18.6. Register details

18.6.1. CMP0_CON0

Addr = 0x36 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7:5	CMPINTS	Comparator output results in an interrupt triggering mode control register 0x0: Rising edge 0x1: Falling edge 0x2: Double edge 0x3: High level 0x4: Low level	RW	0x0
4	-	-	-	-
3	INVENA	The comparator result is enabled by taking the inverse 0x0: Comparator result is not negated 0x1: The comparator is negated	RW	0x0
2	CMPOUT	Comparison results from the comparator	RO	0x0
1	INTENA	Interrupt the enable signal 0x0: Off 0x1: Open	RW	0x0
0	ENA	Simulation part of the comparator can make Enabling the digital part also needs to be enabled for the comparator to work properly 0x0: Off 0x1: Open	RW	0x0

18.6.2. CMP0_CON1

Addr = 0x37 (XSFR)

Bit(s)	Name	Description	R/W	Reset

7	VTHVCCEN	VCC short circuit protection threshold input enable signal (VCC -0.08/0.2/0.32/0.48) 0x0: Closed 0x1: Open	RW	0x0
6	VTHPADEN	PAD short circuit protection threshold input enable signal (PAD -0.08/0.2/0.32/0.48) 0x0: Closed 0x1: Open	RW	0x0
5:4	CMPVTHS	The comparator has a built-in threshold selection 0x0: VCC (PAD) -80mv 0 x1: VCC (PAD) - 200 mv 0 x2: VCC (PAD) - 320 mv 0 x3: VCC (PAD) - 480 mv	RW	0x0
3:2	CHPSEL	The comparator is the input channel selection 0 x0: closed 0 x1: CMP0_P < 0 > - > P11 0x2：CMP0_P<1> -> P12 0 x3: VTH_OUT_VCC/VTH_OUT_PAD	RW	0x0
1:0	CHNSEL	Input channel selection at the negative end of the comparator 0x0: Off 0x1：CMP0_N<0> -> P02 0x2：CMP0_N<1> -> P14 0x3: DAC0_OUT	RW	0x0

18.6.3. CMP0_CON2

Addr = 0x74 (XSFR)

Bit(s)	Name	Description	R/W	Reset

7:6	CMPFSEL	**Comparator result selection bit** 0x0: Comparator outputs the filtered result of the digital filter 0x1: The comparator outputs the result after synchronization 0x2: Comparator output hysteresis sampling result	RW	0x0
5	OUTPUT_EN	**Controls whether the comparator result outputs the control bit** 0x0: Do not output, the result remains the value it was before turning off output 0x1: Output	RW	0x0
4:0	FILTNUM	**Filter clock number Settings** The maximum value is 32, and the step size is 1 microsecond, that is, the filter clock frequency is 1M 0x00: Filter for 1 clock cycle 0x01: Filter for 2 clock cycles ... 0x1F: Filter 32 clock cycles	RW	0x0

18.6.4. CMP0_CON3

Addr = 0x75 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7	HYSEL	**Hysel 4 Select whether the hysteresis signal goes through the filter** 0x0: The filtered signal 0x1: Sync signal before filtering	RW	0x0
6	HYSEN	**Comparator digital hysteresis function enable bit** The threshold is controlled through two registers of the DAC	RW	0x0

		0x0: Turn off hysteresis 0x1: Turn on hysteresis		
5	CMPENS	**Comparator output control selection** 0x0: Whether the comparator outputs is determined by OUTPUT_EN 0x1: Whether the comparator outputs is controlled by PWM	RW	0x0
4	HYSRCSEL	**Hysteresis count source selection** 0x0:1M clock 0x1:64K clock	RW	0x0
3:0	CMPOCOUNTREG	**Hysteresis counter to set the hysteresis sampling interval** Step sizes of 16 microseconds (64K clock) and 1 microsecond (1M clock) are optional 0x0: Sample 1 cycle apart 0x1: Sample 2 cycles apart ... 0xF: Sample 16 cycles apart	RW	0x0

18.6.5. CMP0_CON4

Addr = 0x76 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7	DACTSEN[0]	**The output of the DAC of comparator 0 tests the enable signal** 0x0: Off 0x1: Open	RW	0x0
6	NONLY	**Nonly 2 Turn off the P-tube enable** 0x0: Open the P-tube 0x1: Close the P-tube	RW	0x1
5	OSSHORT	**Short circuit enable** Short the positive and negative ends of the comparator for comparator calibration	RW	0x0

		0x0: Positive and negative ends are not shorted 0x1: Short the plus and minus ends		
4	PONLY	**Ponly 2 Turn off the N-tube enable** 0x0: Turn N-tube on 0x1: Close N-tube	RW	0x0
3:0	TRIM	**Comparator 0 corrects the value** When the chip is powered on, the factory calibration value will be filled in automatically, or the user can fill in the calibration value	RW	-

Note: When the common mode voltage (the voltage at both ends of the comparator) is relatively low (less than 1/2VDD), the N tube should be closed and the P tube should be opened. When the common mode voltage is relatively high, the P tube should be closed and the N tube should be opened. If the voltage is between 0 and VCC, the two should be opened at the same time. Note: CMP0 only opens P tube by default, CMP1 only opens N tube by default.

18.6.6. CMP0_DAC0

Addr = 0x77 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7:0	CMP0_DAC0	**The DAC0 register for comparator 0** If digital hysteresis is not turned on, this register is the input of the built-in DAC of comparator 0. If digital hysteresis is turned on, the input value of DAC is selected between CMP0_DAC0 and CMP0_DAC1. When the comparison output is 1, CMP0_DAC0 is selected, and when the comparator output is 0, CMP0_DAC1 is selected Step=5mV,0x00~0xF0 corresponding to the conversion output is 0~1.2V, 0xF0~0xFF conversion output is 1.2V	RW	0x0

18.6.7. CMP0_DAC1

Addr = 0x78 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7:0	CMP0_DAC1	The DAC1 register for comparator 0 Effective when enabling digital hysteresis Step=5mV,0x00~0xF0 corresponds to conversion output of 0~1.2V, 0xF0~0xFF conversion output of 1.2V	RW	0x0

18.6.8. CMP1_CON0

Addr = 0x38 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7:5	CMPINTS	Comparator output results in an interrupt triggering mode control register 0x0: Rising edge 0x1: Falling edge 0x2: Double edge 0x3: High level 0x4: Low level	RW	0x0
4	-	-	-	-
3	INVENA	The comparator result is enabled by taking the inverse 0x0: Comparator result is not negated 0x1: Comparator result negates	RW	0x0
2	CMPOUT	Comparison results from the comparator	RO	0x0
1	INTENA	Interrupt the enable signal 0x0: Off 0x1: Open	RW	0x0
0	ENA	The analog part of the comparator enabled Enabling the digital part also needs to be	RW	0x0

		enabled for the comparator to work properly 0x0: Off 0x1: Open		

18.6.9. CMP1_CON1

Addr = 0x39 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7:4	-	-	-	-
3:2	CHPSEL	**Input channel selection on the positive end of the comparator** 0x0: Off 0x1：CMP1_P<0> -> P03 0 x2: CMP1_P < 1 > - > P13 0x3: CCS_FB	RW	0x0
1:0	CHNSEL	**Input channel selection at the negative end of the comparator** 0x0: Off 0x1：CMP1_N<0> -> P00 0x2：CMP1_N<1> -> P01 0x3: DAC0_OUT	RW	0x0

18.6.10. CMP1_CON2

Addr = 0x79 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7:6	CMPFSEL	**Comparator result selection bit** 0x0: Comparator outputs the filtered result of the digital filter 0x1: The comparator outputs the result after synchronization 0x2: Comparator output hysteresis sampling	RW	0x0

		result		
5	OUTPUTEN	**Controls whether the comparator result outputs the control bit** 0x0: Do not output, the result remains the value it was before turning off output 0x1: Output	RW	0x0
4:0	FILTNUM	**Filter clock number Settings** The maximum value is 32, and the step size is 1 microsecond, that is, the filter clock frequency is 1M 0x00: Filter for 1 clock cycle 0x01: Filter for 2 clock cycles ... 0x1F: Filter 32 clock cycles	RW	0x0

18.6.11. CMP1_CON3

Addr = 0x7A (XSFR)

Bit(s)	Name	Description	R/W	Reset
7	HYSEL	**Hysel 4 Select whether the hysteresis signal goes through the filter** 0x0: The filtered signal 0x1: Sync signal before filtering	RW	0x0
6	HYSEN	**Comparator digital hysteresis function enable bit** The threshold is controlled through two registers of the DAC 0x0: Turn off hysteresis 0x1: Turn on hysteresis	RW	0x0
5	CMPENS	**Comparator output control selection** 0x0: Whether the comparator outputs is determined by OUTPUT_EN	RW	0x0

		0x1: Whether the comparator outputs is controlled by PWM		
4	HYSRCSEL	**Hysteresis count source selection** 0x0:1M clock 0x1:64K clock	RW	0x0
3:0	CMP0COUNTREG	**Hysteresis counter** Used to set the hysteresis sampling interval in steps of 16 microseconds (64K clock) and 1 microsecond (1M clock) optional 0x0: Interval 1 cycle of sampling 0x1: Sample 2 cycles apart ... 0xF: Sample 16 cycles apart	RW	0x0

18.6.12. CMP1_CON4

Addr = 0x7B (XSFR)

Bit(s)	Name	Description	R/W	Reset
7	DACTSEN[1]	**The output of the DAC of comparator 1 tests the enable signal** 0x0: Off 0x1: Open	RW	0x0
6	NONLY	**Nonly 2 Turn off the P-tube enable** 0x0: Open the P-tube 0x1: Close the P-tube	RW	0x0
5	OSSHORT	**Short circuit enable** Short the positive and negative ends of the comparator for comparator calibration 0x0: Do not short the positive and negative ends 0x1: Short the plus and minus ends	RW	0x0
4	PONLY	**Ponly 2 Turn off the N-tube enable** 0x0: Turn N-tube on	RW	0x1

		0x1: Close N-tube		
3:0	TRIM	**Comparator 0 corrects the value** When the chip is powered on, the factory calibration value will be filled in automatically, or the user can fill in the calibration value	RW	-

Note: When the common mode voltage (the voltage at both ends of the comparator) is relatively low (less than 1/2VDD), the N tube should be closed and the P tube should be opened. When the common mode voltage is relatively high, the P tube should be closed and the N tube should be opened. If the voltage is between 0 and VCC, the two should be opened at the same time. Note: CMP0 only opens P tube by default, CMP1 only opens N tube by default.

18.6.13. CMP1_DAC0

Addr = 0x7C (XSFR)

Bit(s)	Name	Description	R/W	Reset
7:0	CMP0_DAC0	**The DAC0 register for comparator 0** If digital hysteresis is not turned on, this register is the input of the built-in DAC of comparator 1. If digital hysteresis is turned on, the input value of DAC is selected between CMP1_DAC0 and CMP1_DAC1. When the comparison output is 1, CMP1_DAC0 is selected, and when the comparator output is 0, CMP1_DAC1 is selected Step=5mV,0x00~0xF0 corresponding to the conversion output is 0~1.2V, 0xF0~0xFF conversion output is 1.2V	RW	0x0

18.6.14. CMP1_DAC1

Addr = 0x7D (XSFR)

Bit(s)	Name	Description	R/W	Reset

7:0	CMP0_DAC1	**The DAC1 register for comparator 1** Effective when enabling digital hysteresis Step=5mV,0x00~0xF0 corresponds to conversion output of 0~1.2V, 0xF0~0xFF conversion output of 1.2V	RW	0x0

18.6.15. CMP_CON

Addr = 0x3A (XSFR)

Bit(s)	Name	Description	R/W	Reset
7	REFEN	**The reference voltage of the comparator refers to the current enable signal** 0x0: Off 0x1: Open	RW	0x0
6	CCSEN	**Enabling signal for constant current source** 0x0: Off 0x1: Open	RW	0x0
5:0	TRIMIB	**Constant current source current regulation signal (step=2.5%)** 0x00:11.8 mA 0x0F: 20mA 0x1F: 28.4mA	RW	0x0

18.6.16. CMP_STA

Addr = 0x3B (XSFR)

Bit(s)	Name	Description	R/W	Reset
7	CMP1WKUPEN	**Comparator 1 Digital signal wakeup enabled** Used to wake up CPU 0x0: Off	RW	0x0

		0x1: Open		
6	CMP0WKUPEN	**Comparator 0 digital signal wakeup enabled** Used to wake up the CPU 0x0: Off 0x1: Open	RW	0x0
5	CMP1ANAINVEN	**Comparator 1 simulates WAKEUP negation** 0x0: Do not negate 0x1: Take the negation	RW	0x0
4	CMP0ANAINVEN	**Comparator 0 simulates WAKEUP negation** 0x0: No negation 0x1: Take the negation	RW	0x0
3	CMP1ANAWKP	**Comparator 1 Analog signal WAKEUP enabled** Used to wake up the CPU 0x0: Off 0x1: Open	RW	0x0
2	CMP0ANAWKP	**Comparator 0 emulated signal WAKEUP enabled** Used to wake up the CPU 0x0: Off 0x1: Open	RW	0x0
1	CMP1PNDCLR	**Comparator 1 Interrupt flag bit** Write 1 to clear	RW	0x0
0	CMP0PNDCLR	**Comparator 0 interrupt flag bit** Write 1 to clear	RW	0x0

19. I2C module

19.1. Feature Overview

I2C module features special:

- Support host mode and slave mode
- Support for host arbitration

19.2. Feature Description

In the definition of I2C protocol, there are four working modes: host send, host receive, slave send, slave receive. There is also broadcast mode, which works in a similar way to host send.

19.2.1. Host send

In host send mode, the host should provide a clock. You can enter host mode by setting STA (I2CCON.5) register to 1. When the module detects that the bus is idle, it will send a start bit. When the start bit is successfully sent (the clock is kept high and the data line is pulled low), the SI register will be set to 1 and the status code (I2STAT) will be set to 08H. The software should then write the slave address and write command (SLA+W) to I2DAT. Next the SI bit should be zeroed out by the software, triggering the sending of the SLA+W.

When the SLA+W is successfully sent and an ACK is returned from the device, the SI is set again, and the I2STAT is now 18H. At this point, proceed to the next step as needed.

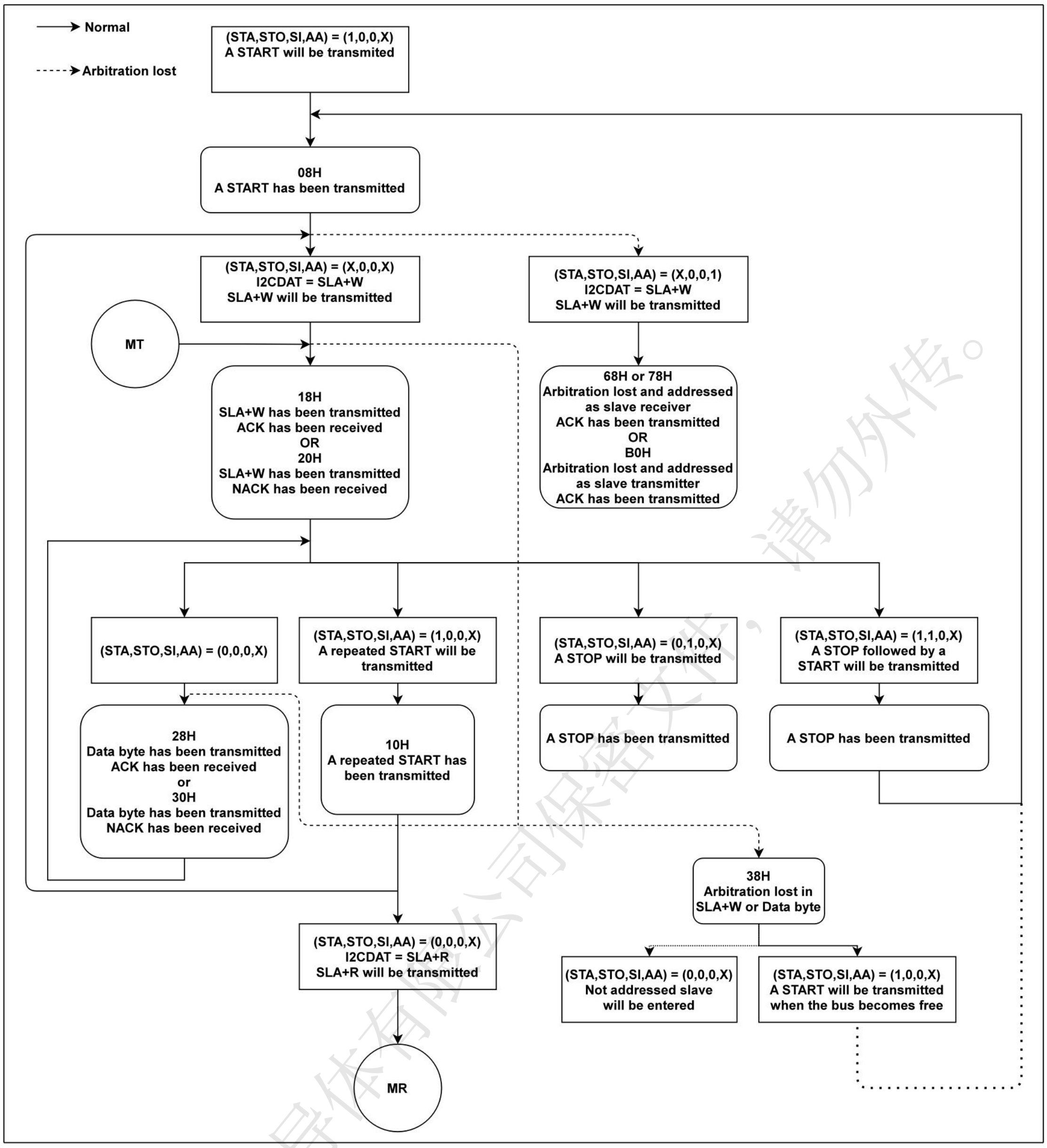

Normal
Arbitration lost
(STA,STO,SI,AA) = (1,0,0,X)
A START will be transmited
08H
A START has been transmitted
(STA,STO,SI,AA) = (X,0,0,X)
I2CDAT = SLA+W
SLA+W will be transmitted
(STA,STO,SI,AA) = (X,0,0,1)
I2CDAT = SLA+W
SLA+W will be transmitted
MT
18H
SLA+W has been transmitted
ACK has been received
OR
20H
SLA+W has been transmitted
NACK has been received
68H or 78H
Arbitration lost and addressed
as slave receiver
ACK has been transmitted
OR
B0H
Arbitration lost and addressed
as slave transmitter
ACK has been transmitted
(STA,STO,SI,AA) = (0,0,0,X)
(STA,STO,SI,AA) = (1,0,0,X)
A repeated START will be
transmitted
(STA,STO,SI,AA) = (0,1,0,X)
A STOP will be transmitted
(STA,STO,SI,AA) = (1,1,0,X)
A STOP followed by a
START will be transmitted
28H
Data byte has been transmitted
ACK has been received
or
30H
Data byte has been transmitted
NACK has been received
10H
A repeated START has
been transmitted
A STOP has been transmitted
A STOP has been transmitted
38H
Arbitration lost in
SLA+W or Data byte
(STA,STO,SI,AA) = (0,0,0,X)
I2CDAT = SLA+R
SLA+R will be transmitted
(STA,STO,SI,AA) = (0,0,0,X)
Not addressed slave
will be entered
(STA,STO,SI,AA) = (1,0,0,X)
A START will be transmitted
when the bus becomes free
MR

Figure 19.1 I2C host send flow diagram

19.2.2. Host receive

In the host receiving mode, the start bit is sent as the host sending mode, except that the data written to I2DAT becomes (SLA+R), and the SI will be set after successful transmission and ACK reception. At this time, the I2STAT value is 40H.

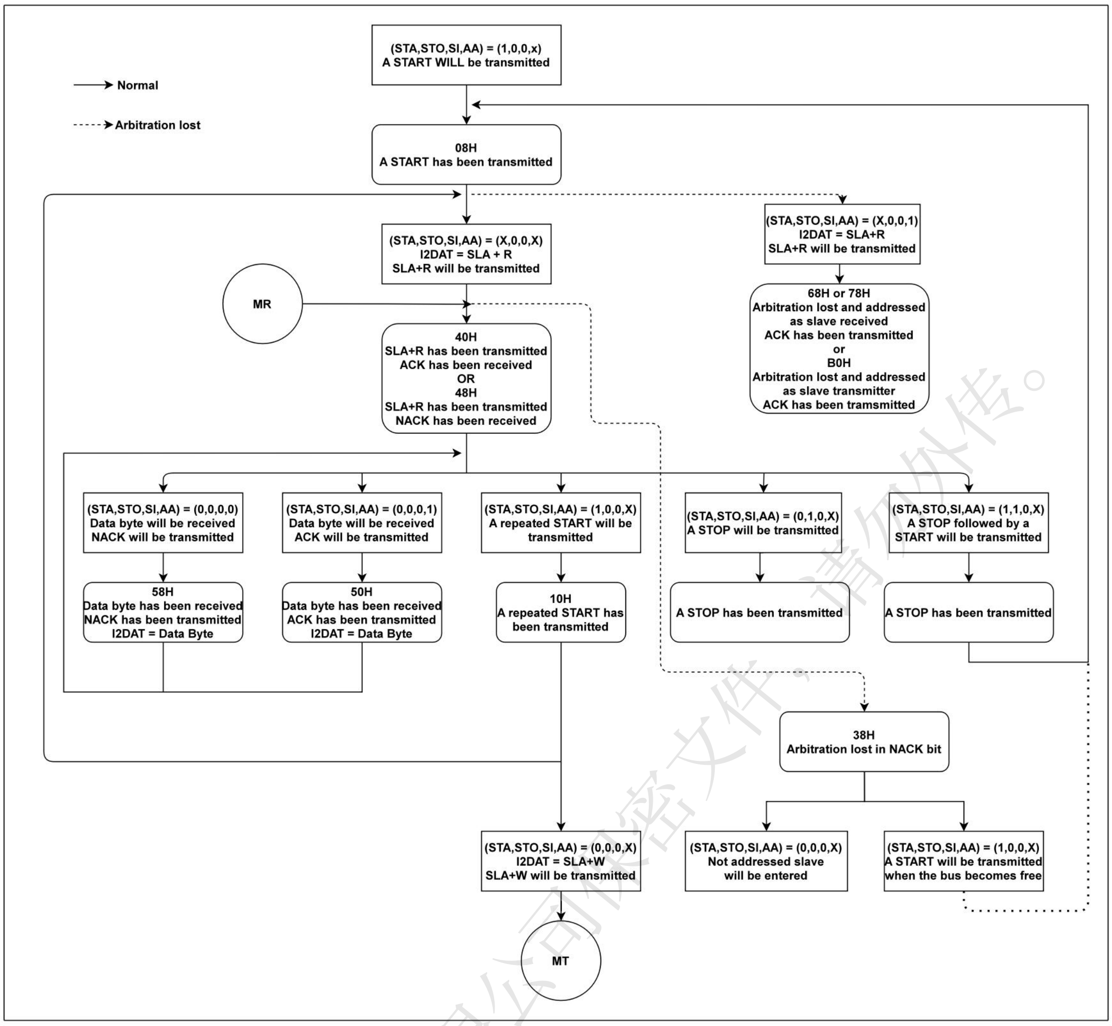

(STA,STO,SI,AA) = (1,0,0,x)
A START WILL be transmitted
Normal
Arbitration lost
08H
A START has been transmitted
(STA,STO,SI,AA) = (X,0,0,X)
I2DAT = SLA + R
SLA+R will be transmitted
(STA,STO,SI,AA) = (X,0,0,1)
I2DAT = SLA+R
SLA+R will be transmitted
MR
40H
SLA+R has been transmitted
ACK has been received
OR
48H
SLA+R has been transmitted
NACK has been received
68H or 78H
Arbitration lost and addressed
as slave received
ACK has been transmitted
or
B0H
Arbitration lost and addressed
as slave transmitter
ACK has been tramsmitted
(STA,STO,SI,AA) = (0,0,0,0)
Data byte will be received
NACK will be transmitted
(STA,STO,SI,AA) = (0,0,0,1)
Data byte will be received
ACK will be transmitted
(STA,STO,SI,AA) = (1,0,0,X)
A repeated START will be
transmitted
(STA,STO,SI,AA) = (0,1,0,X)
A STOP will be transmitted
(STA,STO,SI,AA) = (1,1,0,X)
A STOP followed by a
START will be transmitted
58H
Data byte has been received
NACK has been transmitted
I2DAT = Data Byte
50H
Data byte has been received
ACK has been transmitted
I2DAT = Data Byte
10H
A repeated START has
been transmitted
A STOP has been transmitted
A STOP has been transmitted
38H
Arbitration lost in NACK bit
(STA,STO,SI,AA) = (0,0,0,X)
I2DAT = SLA+W
SLA+W will be transmitted
(STA,STO,SI,AA) = (0,0,0,X)
Not addressed slave
will be entered
(STA,STO,SI,AA) = (1,0,0,X)
A START will be transmitted
when the bus becomes free
MT

Figure 19.2 I2C host receiving flow chart

19.2.3. Slave receiver

In the slave receiving mode, the slave address of the module to be set should be written into I2ADDR before the transmission starts, and the AA register should be set to 1. Only after the AA register is set to 1, the module will reply ACK after receiving the local address.

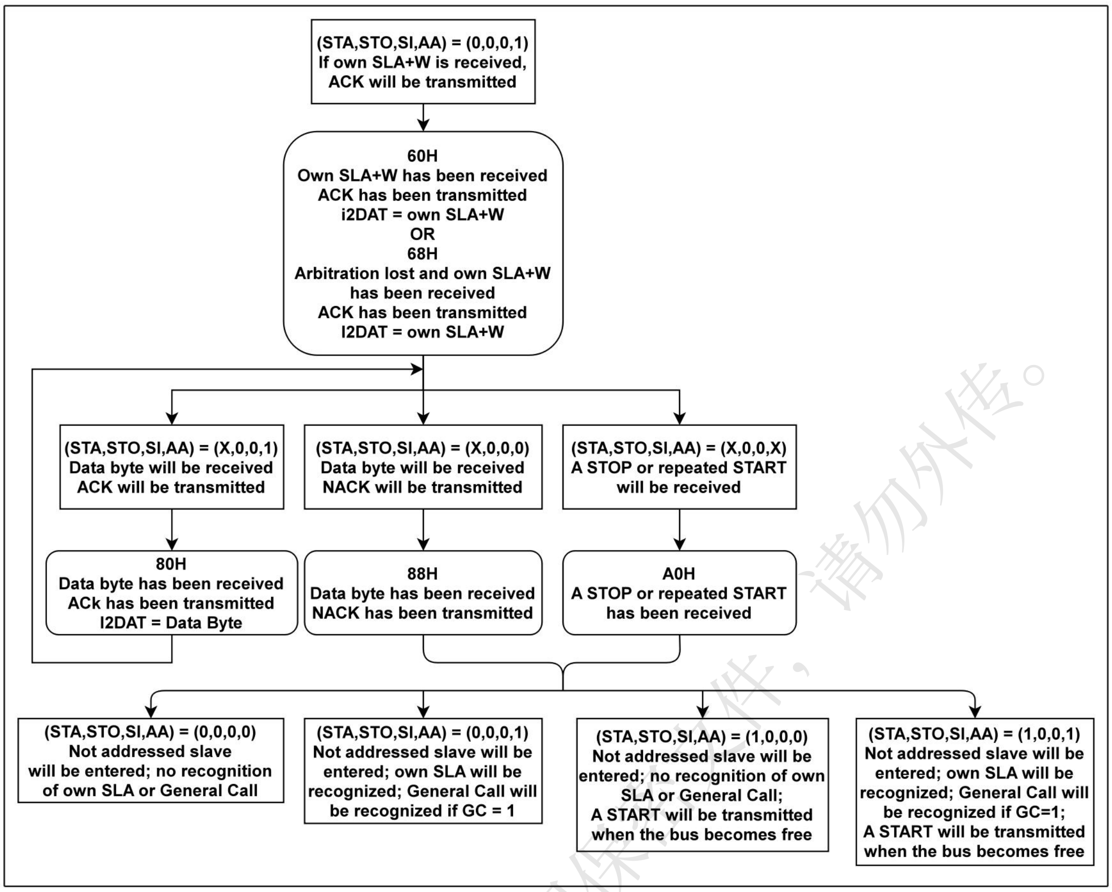

(STA,STO,SI,AA) = (0,0,0,1)
If own SLA+W is received,
ACK will be transmitted
60H
Own SLA+W has been received
ACK has been transmitted
i2DAT = own SLA+W
OR
68H
Arbitration lost and own SLA+W
has been received
ACK has been transmitted
I2DAT = own SLA+W
(STA,STO,SI,AA) = (X,0,0,1)
Data byte will be received
ACK will be transmitted
(STA,STO,SI,AA) = (X,0,0,0)
Data byte will be received
NACK will be transmitted
(STA,STO,SI,AA) = (X,0,0,X)
A STOP or repeated START
will be received
80H
Data byte has been received
ACk has been transmitted
I2DAT = Data Byte
88H
Data byte has been received
NACK has been transmitted
A0H
A STOP or repeated START
has been received
(STA,STO,SI,AA) = (0,0,0,0)
Not addressed slave
will be entered; no recognition
of own SLA or General Call
(STA,STO,SI,AA) = (0,0,0,1)
Not addressed slave will be
entered; own SLA will be
recognized; General Call will
be recognized if GC = 1
(STA,STO,SI,AA) = (1,0,0,0)
Not addressed slave will be
entered; no recognition of own
SLA or General Call;
A START will be transmitted
when the bus becomes free
(STA,STO,SI,AA) = (1,0,0,1)
Not addressed slave will be
entered; own SLA will be
recognized; General Call will
be recognized if GC=1;
A START will be transmitted
when the bus becomes free

Figure 19.3 Flowchart of I2C slave reception

19.2.4. Slave send

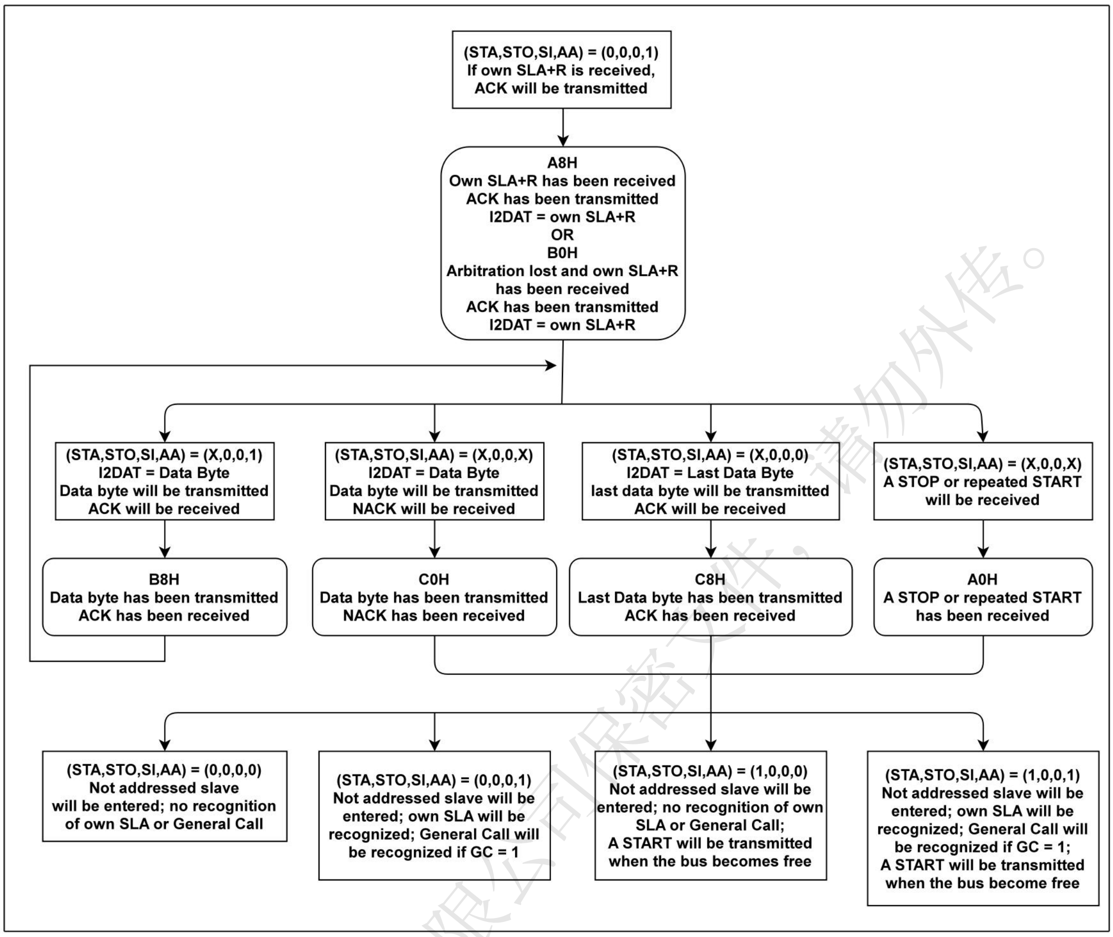

(STA,STO,SI,AA) = (0,0,0,1)
If own SLA+R is received,
ACK will be transmitted
A8H
Own SLA+R has been received
ACK has been transmitted
I2DAT = own SLA+R
OR
B0H
Arbitration lost and own SLA+R
has been received
ACK has been transmitted
I2DAT = own SLA+R
(STA,STO,SI,AA) = (X,0,0,1)
I2DAT = Data Byte
Data byte will be transmitted
ACK will be received
(STA,STO,SI,AA) = (X,0,0,X)
I2DAT = Data Byte
Data byte will be transmitted
NACK will be received
(STA,STO,SI,AA) = (X,0,0,0)
I2DAT = Last Data Byte
last data byte will be transmitted
ACK will be received
(STA,STO,SI,AA) = (X,0,0,X)
A STOP or repeated START
will be received
B8H
Data byte has been transmitted
ACK has been received
C0H
Data byte has been transmitted
NACK has been received
C8H
Last Data byte has been transmitted
ACK has been received
A0H
A STOP or repeated START
has been received
(STA,STO,SI,AA) = (0,0,0,0)
Not addressed slave
will be entered; no recognition
of own SLA or General Call
(STA,STO,SI,AA) = (0,0,0,1)
Not addressed slave will be
entered; own SLA will be
recognized; General Call will
be recognized if GC = 1
(STA,STO,SI,AA) = (1,0,0,0)
Not addressed slave will be
entered; no recognition of own
SLA or General Call;
A START will be transmitted
when the bus becomes free
(STA,STO,SI,AA) = (1,0,0,1)
Not addressed slave will be
entered; own SLA will be
recognized; General Call will
be recognized if GC = 1;
A START will be transmitted
when the bus become free

Figure 19.4 I2C send from slave flow chart

19.2.5. Broadcast mode

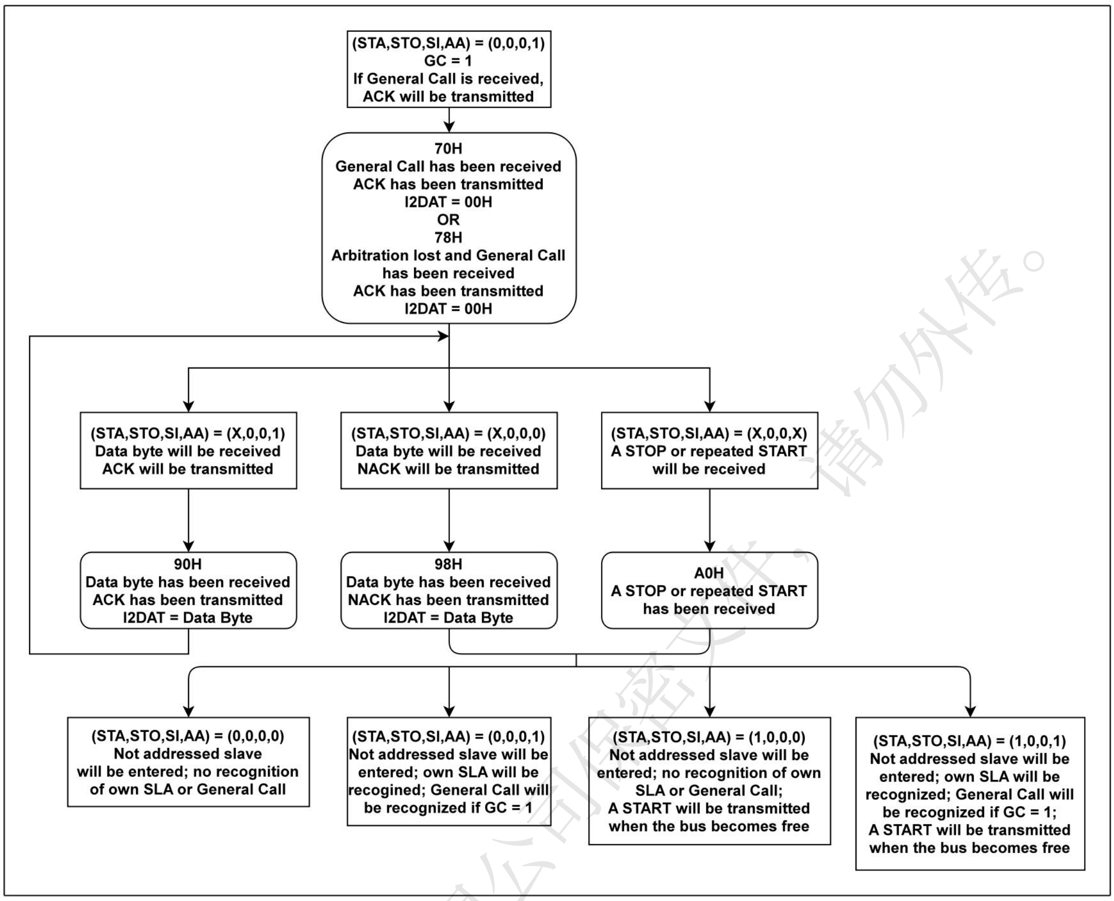

(STA,STO,SI,AA) = (0,0,0,1)
GC = 1
If General Call is received,
ACK will be transmitted
70H
General Call has been received
ACK has been transmitted
I2DAT = 00H
OR
78H
Arbitration lost and General Call
has been received
ACK has been transmitted
I2DAT = 00H
(STA,STO,SI,AA) = (X,0,0,1)
Data byte will be received
ACK will be transmitted
(STA,STO,SI,AA) = (X,0,0,0)
Data byte will be received
NACK will be transmitted
(STA,STO,SI,AA) = (X,0,0,X)
A STOP or repeated START
will be received
90H
Data byte has been received
ACK has been transmitted
I2DAT = Data Byte
98H
Data byte has been received
NACK has been transmitted
I2DAT = Data Byte
A0H
A STOP or repeated START
has been received
(STA,STO,SI,AA) = (0,0,0,0)
Not addressed slave
will be entered; no recognition
of own SLA or General Call
(STA,STO,SI,AA) = (0,0,0,1)
Not addressed slave will be
entered; own SLA will be
recogined; General Call will
be recognized if GC = 1
(STA,STO,SI,AA) = (1,0,0,0)
Not addressed slave will be
entered; no recognition of own
SLA or General Call;
A START will be transmitted
when the bus becomes free
(STA,STO,SI,AA) = (1,0,0,1)
Not addressed slave will be
entered; own SLA will be
recognized; General Call will
be recognized if GC = 1;
A START will be transmitted
when the bus becomes free

Figure 19.5 I2C broadcast mode flowchart

19.3. List of registers

Table 19.1 List of IIC registers

Address	Register Name	Description
0x70 (XSFR)	I2C0_CON	I2C control register
0x71 (XSFR)	I2C0_DATA	I2C data register
0x72 (XSFR)	I2C0_ADR	I2C address register
0x73 (XSFR)	I2C0_STA	I2C status register

19.4. Register details

19.4.1. I2C0_CON

Addr = 0x70 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7	ENS1	**IIC enable bit** 0x0: ″sdao″ and ″sclo″ output 1 and ignore ″sdai″ and ″scli″ inputs 0x1: Enable the iic module	RW	0x0
6	STA	**IIC start bit** 0x0: No operation 0x1: Checks the iic bus and sends a start bit if the bus is idle and the module is in host mode	RW	0x0
5	STO	**The IIC stop bit** 0x0: No operation 0x1: When the module is in host mode, a stop bit is sent	RW	0x0
4	SICLR	**SI clear bit** Write 1 to clear SI, read constant to 0	WO	0x0
3	AA	**IIC reply control bit** 0x0: NACK is sent when the following occurs ● 1byte is received in host receive mode ● 1byte is received in slave receiving mode 0x1: Send ACK when the following occurs ● Receive the local slave address ● Receive the broadcast address with the broadcast address bit enabled ● 1byte is received in the case of host reception	RW	0x0

		1byte is received in the case of slave reception		
2:0	CR	**Baud rate control bit** The I2C baud rate is calculated from the following formula :baud(kHz)= sysclk/x, where the value of x is determined by the CR register, are: 0x00:256 0x01:224 0x02:160 0x03:80 0x04:1024 0x05:120 0x06:60 0x07: When CR is set to 0x07, the baud rate is determined by the pwm frequency of Timer0 and is calculated as the pwm frequency of Timer0 divided by 8	RW	0x0

19.4.2. I2C0_STA

Addr = 0x73 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7:3	STA	**Module status flag bit**	RO	0x1F
2	SI	**SI flag bit** All but the f8H state will set the SI, and SI_CLR writes 1 and reads DATA will clear the SI	RO	0x0
1	INT	**Interrupt flag bit** When interrupt is enabled, set with SI	RO	0x0
0	INTEN	**Interrupt the enable bit**	RW	0x0

19.4.3. I2C0_ADR

Addr = 0x72 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7:1	ADR	Adrs for this module	RW	0x0
0	GC	Broadcast address control bits 0x0: Ignore broadcast address 0x1: Respond to broadcast address	RW	0x0

19.4.4. I2C0_DATA

Addr = 0x71 (XSFR)

Bit(s)	Name	Description	R/W	Reset
7:0	DAT	The data register	RW	0x0

19.5. Instructions for using the process