CMS8S78xx series

Reference Manual

Enhanced flash 8-bit 1T 8051 microcontroller

Rev. 1.0.9

Table of Content

1. Central Processing Unit (CPU)

The series is a microcontroller with 8-bit 8051 frame structure. The CPU is the core component of the microcontroller, which is composed of arithmetic units, controllers, and special register groups. The arithmetic unit module mainly implements data arithmetic and logic operations, bit variable processing and data transfer operations; the controller module mainly decodes instructions, and then sends out various control signals; the special register group is mainly used to indicate the memory address of the current instruction to be executed , Store the operand and indicate the state after the instruction is executed. The special register group mainly includes accumulatorACC, general register B, stack pointer SP, data pointer DPTR, Program status registerPSW, Program counterPC, etc.

1.1 Reset Vector (0000H)

The microcontroller has a word-long system reset vector (0000H), after which the program will restart execution at 0000H and the system registers will revert to their default values. The following program demonstrates how to define a reset vector in FLASH.

Example: Define a reset vector

```
        ORG     0000H       ; System reset vector
        LJMP    START
        ORG     0010H       ; The user program starts
START:
        …                   ; User programs
        …
        END                 ; The program ends
```

1.2 BOOT Partition

The size of the program area space is 16K*8Bit, where the program is divided into BOOT area and APROM area, and the BOOT area size is allocated by the user configuration register.

When the chip is powered on, if the program is started from the BOOT area, it needs to be satisfied: the address space allocation method is 1/2/3 (set BOOT_1K/BOOT_2K/BOOT_4K through CONFIG), otherwise the program will be launched from the APROM area.

Take the 1K space in the BOOT area as an example: confIG configuration BOOT_1K, after the chip is powered on configuration, the program starts running from address 3C00H. If the program needs to switch between the BOOT region and the APROM region, it is necessary to write a 0xAA/0x55 to the BOOT region control register BOOTCON (see register description for details), and then perform a software reset or generate a watchdog reset.

When power-on reset, external reset, voltage reset, bootcon reset value is 0x00, software reset and watchdog reset can not clear the register.

BOOT Control Register (BOOTCON)

F691H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BOOTCON	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 D<7:0>: BOOT region control bit (this register can only be written when the chip is configured for BOOT_1K/BOOT_2K/BOOT_4K);

0x55= If you switch from the APROM area to the BOOT area, you need to write 0x55 to it, and then perform a software reset or generate a watchdog reset;

0xAA= If you switch from the BOOT area to the APOROM area, you need to write a 0xAA to it, and then perform a software reset or generate a watchdog reset;

Other values= Invalid.

For example, after the chip is powered on and booted from the BOOT area, use the software reset method to switch to the APROM area, and the configuration is as follows:

1) The BOOTCON register needs to write AAH

```
MOVDPTR   ,# BOOTCON
MOVE,#0AAH
MOVX@DPTR,A
```

2) Perform a software reset

```
MOVTA,#0AAH
MOVTA,#055H
MOVWDCON   ,#080H
```

For example, use the software reset method to switch from the APROM area to the BOOT area, and configure it as follows:

1) The BOOTCON register needs to write 55H

```
MOVDPTR，#   BOOTCON
MOVA,#055H
MOVX@DPTR,A
```

2) Perform a software reset

```
MOVTA,#0AAH
MOVTA,#055H
MOVWDCON   ,#080H
```

Note: When the BOOT function is active, the program in the A PROM needs to ensure that the PC does not overflow (overflow means that the PC is outside the address range of the A PROM) if P C occurs In the case of an overflow, the system may be running abnormally.

1.3 Accumulator (ACC)

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

The ACC register is an 8Bit register where the results of ALU operations can be stored.

1.4 B Register(B)

The B register is used when using multiplication and division instructions. If the multiplication and division instruction is not used, it can also be used as a universal register.

1.5 Stack Pointer Register (SP)

The SP register points to the address of the stack, and the default value after reset is 0x07, which means that the area of the stack starts at 08H of the RAM address. The value of the SP can be modified, and if the stack region is set to start at 0xC0, the value of the SP needs to be set to 0xBF after the system reset.

Operations that affect SP are: instruction PUSH, LCALL, ACALL, POP, RET, RETI, and entering interrupts.

PUSH instructions occupy one byte in the stack, LCALL, ACALL, and interrupt occupy two bytes in the stack, POP instructions release one byte, and RET/RETI instructions free two bytes.

Using the PUSH instruction automatically saves the current value of the operated register to RAM.

1.6 Data Pointer Register (DPTR0/DPTR1)

The data pointer is mainly used in MOVX, MOVC instructions, and its role is to locate the addresses of XRAM and ROM. Inside the chip, there are two data pointer registers DPTR0 and DPTR1, selected by the DPS register.

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

For example, the assembly code for operating XRAM is as follows:

```
MOV     DPTR,#0001H
MOV     A,#5AH
MOVX    @DPTR,A          ; Write the data in A to XRAM address 0001H
```

1.7 Data Pointer Selection Register (DPS)

The data pointer selects register DPS

0x86	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DPS	ID1	ID0	TSL	AT	--	--	--	SALT
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit6 ID<1:0>: Subtract/add function selection.
00= DPTR0 plus 1 or DPTR1 plus 1;
01= DPTR0 minus 1 or DPTR1 plus 1;
10= DPTR0 plus 1 or DPTR1 minus 1;
11= DPTR0 minus 1 or DPTR1 minus 1.
Bit5 TSL: Flip selection enable;
1= After executing the DPTR instruction, the SEL bit is automatically flipped;
0= DPTR-related instructions do not affect the SEL bits.
Bit4 AT: Self-addition/subtraction enable bits;
1= Allows MOVX @DPTR or MOVC @DPTR instructions to run, perform subtractive/self-addition operations (determined by ID1-ID0).
0= DPTR-related directives do not affect DPTR itself.
Bit3~Bit1 --: Reserved, must be 0.
Bit0 SALT: Data pointer selection bit;
1= Select DPTR1;
0= Select DPTR0.

1.8 Program Status Register (PSW)

Program status register PSW

0xD0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PSW	CY	AND	F0	RS1	RS0	OV	--	P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
Reset value	0	0	0	0	0	0	0	0

Bit7 CY: Carry marker bits;
1= There are carry-ons;
0= No carry.
Bit6 AND: Auxiliary carry markers (half-carry markers);
1= There are carry-ons;
0= No carry.
Bit5 F0: Generic flag bits.
Bit4~Bit3 RS<1:0>: Working register BANK select bit;
00= Select Bank0;
01= Select Bank1;
10= Select Bank2;
11= Select Bank3.
Bit2 OV: Overflow flag bit;
1= There is an overflow in arithmetic or logical operations;
0= There is no overflow of arithmetic or logical operations.
Bit1 --: Reserved, must be 0.
Bit0 P: Check bit;
1= The highest level of the result occurred.
0= The highest bit of the result did not occur carry.

1.9 Program Counter (PC)

The program counter (PC) controls the order of instruction execution in the program memory FLASH, it can address the entire flash range, after obtaining the instruction code, the program counter (PC) will automatically add one, pointing to the address of the next instruction code. However, when performing operations such as jumps, conditional jumps, subroutine calls, initial resets, interrupts, interrupt returns, subprogram returns, etc., the PC loads the address associated with 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 an empty instruction operation cycle will be inserted before the correct instruction can be obtained. Instead, the next instruction is executed sequentially.

1.10 Timing Access Register (TA)

Timing access register TA

0x96	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
HE	TA7	TA6	TA5	TA4	TA3	TA2	TA1	TA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 TA<7:0>: Timing access control bits.

Some protected registers must be written to the TA before they can be performed as follows.

```
MOV   HE, #0AAH
MOV   TA, #055H
```

No other instructions can be inserted in the middle, and this sequence needs to be re-executed when modified again.

Protected registers: WDCON，CLKDIV，SCKSEL，MLOCK。

2. Memory and Register Mapping

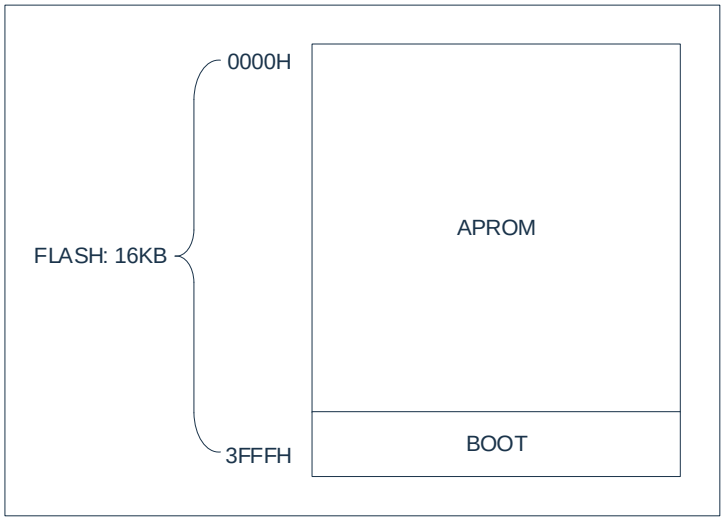
This series of Microcontrollers has the following types of memory:

- Flash program memory up to 16KB (shared by APROM and BOOT).
- Non-volatile data memory (Data FLASH) up to 1KB.
- Up to 256B of General Purpose Internal Data Memory (RAM).
- Up to 1 KB of General Purpose External Data Memory (XRAM).
- Special function register SFR.
- External special function register XSFR.

2.1 Program Storage Flash

Program memory FLASH is used to store source program and table data, and the program counter PC is used as an address pointer. The PC is a 16-bit program counter, so the address space that can be addressed is 64KB. But this chip only has 16K bytes of program storage space.

The FLASH space allocation block diagram is shown in the following figure:

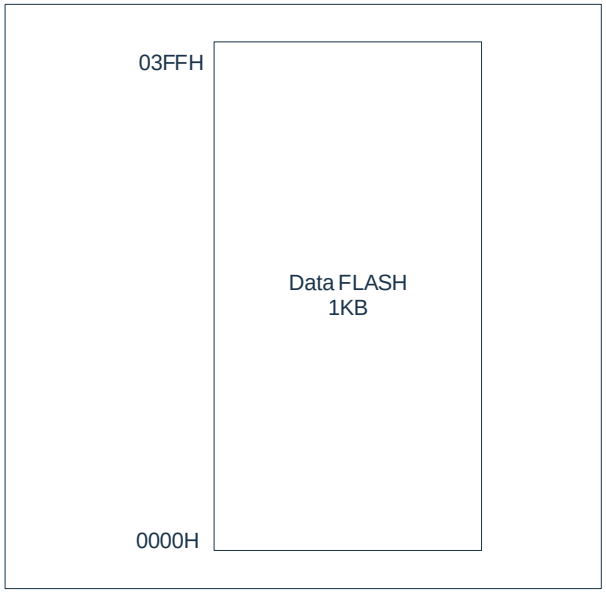


After the chip is reset, the CPU starts executing from 0000H. Each interrupt is assigned a fixed address in the program memory, and the interrupt causes the CPU to jump to that address to start executing the service program.

For example, external interrupt 1, which is assigned the address 0013H, if external interrupt 1 is used, its service program must start at the 0013H location. If the interrupt is not used, its service address is used as a normal program store address.

2.2 Non-volatile Data Memory Data FLASH

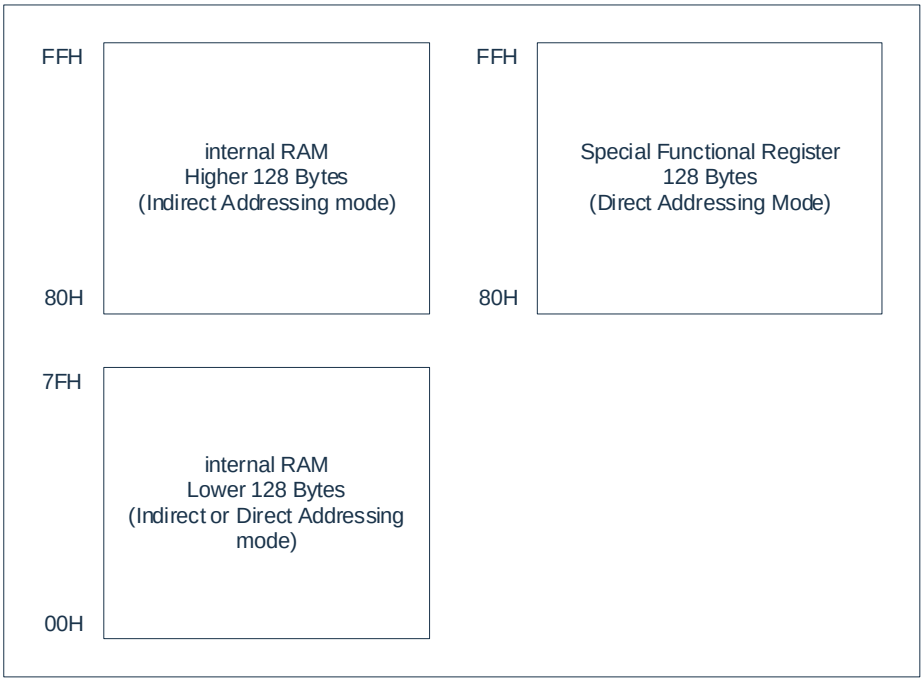
The non-volatile data memory Data Flash can be used to store important data such as constant data, calibration data, protection safety-related information, etc. The data stored in this area has the characteristic that the data is not lost in the event of a chip power outage or a sudden or unexpected power outage. Data FLASH space allocation block diagram is shown in the following figure:



The read, write and erase operations of the Data FLASH memory are implemented through the FLASH control interface.

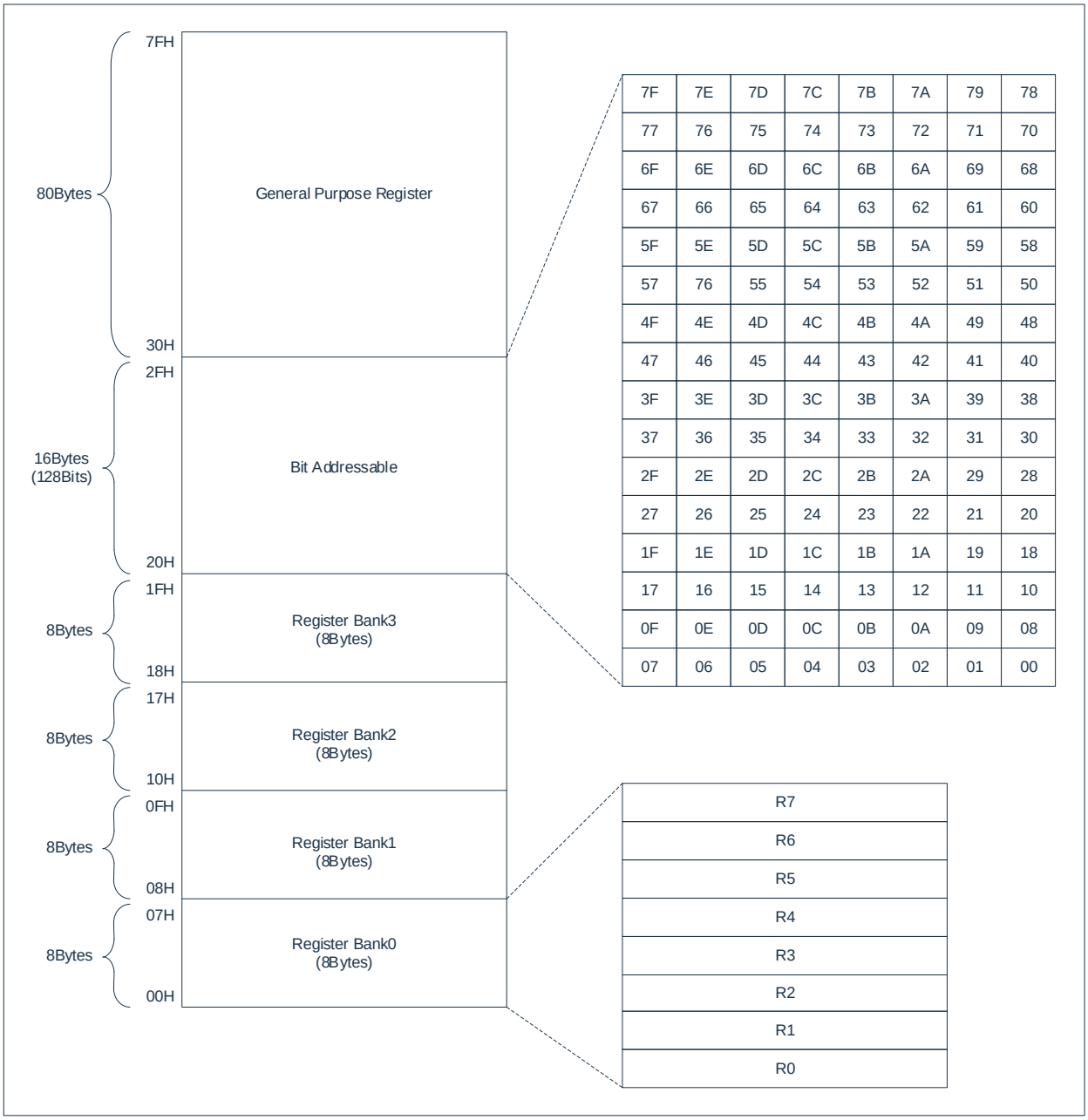
2.3 General Data Register RAM

The internal data memory is divided into three parts: low 128Bytes, high 128Bytes, and special function register SFR. The RAM space allocation block diagram is shown in the following figure:



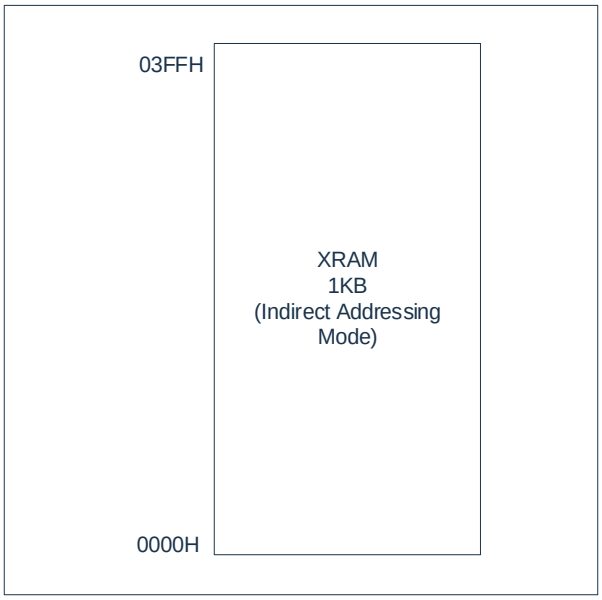
The high 128 Bytes shown above and SFR occupy the same area (80H to FFH), but they are independent. Storage spaces with direct addressing above 7FH (SFR) and indirect addressing above 7FH (128 Bytes high) go into different storage spaces.

The low 128Bytes spatial register allocation shown in the figure above is shown in the following figure. The lowest 32 bytes (00H to 1FH) form 4 register groups, each group of 8 storage units, with R0 ~ R7 as the unit number, used to save the operands and intermediate results. After reset, the 0 group is selected by default, and if a different register group is selected, it is decided by changing the program state. The 16 Bytes (20H to 2FH) behind the register bank form a bit-addressable storage space in which the RAM cells can operate either byte-byte or directly on each bit in the cell. With the remaining 80 storage units (30H to 7FH), the user can set the stack area and store intermediate data.



2.4 General External Data Register XRAM

There is a maximum 1KB XRAM area inside the chip, this area is not connected to FLASH/RAM, and the XRAM space allocation block diagram is shown in the following figure:



XRAM/XSFR spatial access operates through DPTR data pointers, which consist of two sets of pointers: DPTR0, DPTR1, selected by the DPS registers. For example, through movx indirection operations, the assembly code is as follows:

```
MOV     R0.#01H
MOV     A,#5AH
MOVX    @R0,A       ; Write the data in A to XRAM address 01H, the high 8-bit address is determined by DPH0/1
```

After target--> Memory Model is set to Large in Keil51, the C compiler will take XRAM as the variable address. XRAM/XSFR operations are generally performed with DPTR.

2.5 Special Function Register SFR

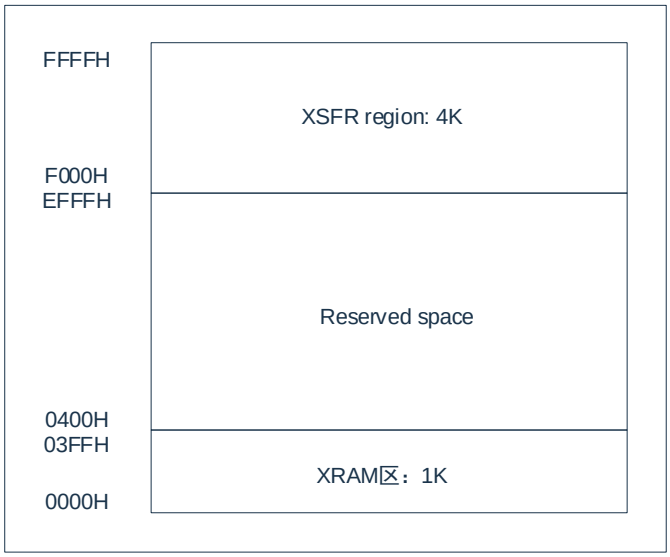
Special function registers refer to a collection of registers with special purposes, essentially some on-chip RAM units with special functions, discretely distributed in the address range of 80H to FFH. Users can byte access them through direct addressing instructions, and addresses four bits lower than 0000 or 1000 can be addressed bitwise, such as P0, TCON, P1.

The SFR register table is as follows:

	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
0xF8	--	PCRCDL	PCRCDH	MLOCK	MADRL	MADRH	MDATA	MCTRL
0xF0	B	I2CSADR	I2CSCR	I2CSBUF	I2CMSA	I2CMCR	I2CMBUF	I2CMTP
0xE8	--	ADCON2	--	--	SPCR	SPSR	SPDR	SSCR
0xE0	ACC	--	TL4	TH4	--	--	--	--
0xD8	--	ADCCHS	TL3	TH3	ADRESL	ADRESH	ADCON1	ADCON0
0xD0	PSW	ADCMPC	T34MOD	ADDLYL	ADCMPL	ADCMPH	SCKSEL	SCKSTAU
0xC8	T2CON	T2IF	RLDL	RLDH	TL2	TH2	CCEN	T2IE
0xC0	--	--	CCL1	CCH1	CCL2	CCH2	CCL3	CCH3
0xB8	IP	EIP1	EIP2	EIP3	WUTCRL	WUTCRH	BUZDIV	BUZCON
0xB0	P3	--	EIF2	--	P0EXTIF	P1EXTIF	P2EXTIF	P3EXTIF
0xA8	IE	--	EIE2	--	P0EXTIE	P1EXTIE	P2EXTIE	P3EXTIE
0xA0	P2	P1TRIS	P2TRIS	P3TRIS	--	--	--	--
0x98	SCON0	SBUF0	P0TRIS	--	--	--	--	--
0x90	P1	FUNCCR	--	--	--	--	HE	WDCON
0x88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	CLKDIV
0x80	P0	SP	DPL0	DPH0	DPL1	DPH1	DPS	PCON

2.6 External Special Function Register XSFR

XSFR is a special register shared by the addressing space and XRAM, mainly including: port control registers, other function control registers. Its addressing range is shown in the following figure:



The list of external special function registers is as follows:

address	register	Register description
F000H	P00CFG	P00 port configuration register
F001H	P01CFG	P01 port configuration register
F002H	P02CFG	P02 port configuration register
F003H	P03CFG	P03 port configuration register
F004H	P04CFG	P04 port configuration register
F005H	P05CFG	P05 port configuration register
F006H	P06CFG	P06 port configuration register
F007H	P07CFG	P07 port configuration register
F009H	P0OD	P0-port open-drain control register
F00AH	P0UP	P0 port pull-up resistor control register
F00BH	P0RD	P0 port pull-down resistor control register
F00CH	P0DR	P0 ports drive current selection registers
F00DH	P0SR	P0 port slope control register
F00EH	P0DS	P0 port data input select register
--	--	--
F010H	P10CFG	P10 port configuration register
F011H	P11CFG	P11 port configuration register
F012H	P12CFG	P12 port configuration register
F013H	P13CFG	P13 port configuration register
F014H	P14CFG	P14 port configuration register
F015H	P15CFG	P15 port configuration register
F016H	P16CFG	P16 port configuration register
F017H	P17CFG	P17 port configuration register
F019H	P1OD	P1 port open-drain control register
F01AH	P1UP	P1 port pull-up resistor control register
F01BH	P1RD	P1 port pull-down resistor control register
F01CH	P1DR	Port P1 drives the current selection register

F01DH	P1SR	P1 port slope control register
F01EH	P1DS	Port P1 data input select register
--	--	--
F020H	P20CFG	P20 port configuration register
F021H	P21CFG	P21 port configuration register
F022H	P22CFG	P22 port configuration register
F023H	P23CFG	P23 port configuration register
F024H	P24CFG	P24 port configuration register
F025H	P25CFG	P25 port configuration register
--	--	--
--	--	--
F029H	P2OD	P2 port open-drain control register
F02AH	P2UP	P2 port pull-up resistor control register
F02BH	P2RD	P2 port pull-down resistor control register
F02CH	P2DR	P2 port drives the current selection register
F02DH	P2SR	P2 port slope control register
F02EH	P2DS	P2 port data input select register
--	--	--
F030H	P30CFG	P30 port configuration register
F031H	P31CFG	P31 port configuration register
F032H	P32CFG	P32 port configuration register
F033H	P33CFG	P33 port configuration register
--	--	--
--	--	--
--	--	--
--	--	--
F039H	P3OD	P3 port open-drain control register
F03AH	P3UP	P3 port pull-up resistor control register
F03BH	P3RD	P3 port pull-down resistor control register
F03CH	P3DR	P3 port drives the current selection register
F03DH	P3SR	P3 port slope control register
F03EH	P3DS	P3 port data input select register
--	--	--
F080H	P00EICFG	P00 interrupt control register
F081H	P01EICFG	P01 interrupt control register
F082H	P02EICFG	P02 port interrupt control register
F083H	P03EICFG	P03 interrupt control register
F084H	P04EICFG	P04 port interrupt control register
F085H	P05EICFG	P05 interrupt control register
F086H	P06EICFG	P06 port interrupt control register
F087H	P07EICFG	P07 port interrupt control register
F088H	P10EICFG	P10 port interrupt control register
F089H	P11EICFG	P11 interrupt control register
F08AH	P12EICFG	P12 interrupt control register
F08BH	P13EICFG	P13 port interrupt control register
F08CH	P14EICFG	P14 interrupt control register
F08DH	P15EICFG	P15-port interrupt control register
F08EH	P16EICFG	P16 port interrupt control register

F08FH	P17EICFG	P17 interrupt control register
F090H	P20EICFG	P20 port interrupt control register
F091H	P21EICFG	P21 interrupt control register
F092H	P22EICFG	P22 port interrupt control register
F093H	P23EICFG	P23 interrupt control register
F094H	P24EICFG	P24 interrupt control register
F095H	P25EICFG	P25 interrupt control register
--	--	--
--	--	--
F098H	P30EICFG	P30 port interrupt control register
F099H	P31EICFG	P31 port interrupt control register
F09AH	P32EICFG	P32 port interrupt control register
F09BH	P33EICFG	P33 port interrupt control register
--	--	--
--	--	--
--	--	--
--	--	--
F0C0H	PS_INT0	External interrupt 0 input port allocation register
F0C1H	PS_INT1	External interrupt 1 input port allocation register
F0C2H	PS_T0	Timer0 external clock input port assignment register
F0C3H	PS_T0G	Timer0 gated input port assignment register
F0C4H	PS_T1	Timer1 external clock input port assignment register
F0C5H	PS_T1G	Timer1 gated input port assignment register
F0C6H	PS_T2	Timer2 external event or gated input port assignment register
F0C7H	PS_T2EX	Timer2 drops along the autoreload input port allocation register
F0C8H	PS_CAP0	The Timer2 input captures channel 0 port assignment registers
F0C9H	PS_CAP1	The Timer2 input captures the channel 1 port assignment register
F0CAH	PS_CAP2	The Timer2 input captures the channel 2 port assignment register
F0CBH	PS_CAP3	The Timer2 input captures the channel 3 port assignment register
F0CCH	PS_ADET	The ADC's external trigger input port allocates registers
F0CDH	PS_FB0	PWM's external brake signal 0-port distribution register
F0CAndH	PS_FB1	PWM's external brake signal 1-port distribution register
--	--	--
F120H	PWMCON	PWM control registers
F121H	PWMOE	PWM output enable register
F122H	PWMPINV	PWM output polarity selection register
F123H	PWM01PSC	PWM0/1 clock prescaler control register
F124H	PWM23PSC	PWM2/3 clock prescale control register
--	--	--
F126H	PWMCNTE	The PWM count starts the control register
F127H	PWMCNTM	PWM counting mode select register
F128H	PWMCNTCLR	PWM counter clears control registers
F129H	PWMLOADEN	PWM loads the enable control register
F12AH	PWM0DIV	PWM0 clock divider control register
F12BH	PWM1DIV	PWM1 clock divider control register

F12CH	PWM2DIV	PWM2 clock divider control register
F12DH	PWM3DIV	PWM3 clock divider control register
--	--	--
---	--	--
F130H	PWMP0L	The PWM0 cycle data register is 8 bits lower
F131H	PWMP0H	The PWM0 cycle data register is 8 bits high
F132H	PWMP1L	The PWM1 cycle data register is 8 bits lower
F133H	PWMP1H	The PWM1 cycle data register is 8 bits high
F134H	PWMP2L	The PWM2 cycle data register is 8 bits lower
F135H	PWMP2H	The PWM2 cycle data register is 8 bits high
F136H	PWMP3L	The PWM3 cycle data register is 8 bits lower
F137H	PWMP3H	The PWM3 cycle data register is 8 bits high
--	--	--
F140H	PWMD0L	PWM0 compares 8 bits lower to the data register
F141H	PWMD0H	PWM0 compares the data register 8 bits higher
F142H	PWMD1L	PWM1 compares 8 bits lower to the data register
F143H	PWMD1H	PWM1 compares the data registers 8 bits higher
F144H	PWMD2L	PWM2 compares 8 bits lower than the data register
F145H	PWMD2H	PWM2 compares the data registers 8 bits higher
F146H	PWMD3L	PWM3 compares 8 bits lower than the data register
F147H	PWMD3H	PWM3 compares the data registers 8 bits higher
--	--	--
F150H	PWMDD0L	PWM0 compares down to 8 bits lower the data register
F151H	PWMDD0H	The PWM0 compares down to the data register 8 bits higher
F152H	PWMDD1L	PWM1 compares down to 8 bits lower than the data register
F153H	PWMDD1H	PWM1 compares down to the data register 8 bits higher
F154H	PWMDD2L	PWM2 compares down to 8 bits lower the data register
F155H	PWMDD2H	PWM2 compares down to the data register 8 bits higher
F156H	PWMDD3L	PWM3 compares down to 8 bits lower the data register
F157H	PWMDD3H	PWM3 compares down to the data register 8 bits higher
--	--	--
F15CH	PWMBRKC	PWM brake recovery control register
F15DH	PWMBRKRDTL	PWM delay recovery data registers 8 bits lower
F15EH	PWMBRKRDTH	PWM delay recovery data registers 8 bits high
--	--	--
F160H	PWMDTE	The PWM dead-band enable control register
F161H	PWM01DT	PWM0/1 dead-zone delay data register
F162H	PWM23DT	PWM2/3 dead-zone delay data register
--	--	--
F164H	PWMMASKE	PWM mask control registers
F165H	PWMMASKD	PWM mask data registers
F166H	PWMFBKC	PWM brake control registers
F167H	PWMFBKD	PWM brake data registers
F168H	PWMPIE	The PWM cycle interrupts the shield registers
F169H	PWMZIE	PWM zero-point interrupt mask register
F16AH	PWMUIE	PWM up compares interrupt mask registers
F16BH	PWMDIE	PWM compares the interrupt mask registers downwards

F16CH	PWMPIF	PWM cycle interrupt flag register
F16DH	PWMZIF	PWM zero-point interrupt flag register
F16EH	PWMUIF	PWM up compares the interrupt flag registers
F16FH	PWMDIF	PWM compares the interrupt flag registers downwards
--	--	--
F500H	C0CON0	Comparator 0 control register 0
F501H	C0CON1	Comparator 0 control register 1
F502H	C0CON2	Comparator 0 control register 2
F503H	C1CON0	Comparator 1 control register 0
F504H	C1CON1	Comparator 1 control register 1
F505H	C1CON2	Comparator 1 control register 2
F506H	CNVRCON	Comparator reference voltage control register
F507H	CNFBCON	Comparator brake control registers
F508H	MISS	Comparator interrupt mask registers
F509H	CNIF	Comparator interrupt flag register
F50AH	C0ADJE	Comparator 0 adjusts the bit selection register
F50BH	C1ADJE	Comparator 1 adjusts the bit selection register
F50CH	C0HYS	Comparator 0 hysteresis control register
F50DH	C1HYS	Comparator 1 hysteresis control register
--	--	--
F5C0H	BRTCON	The BRT module controls the registers
F5C1H	BRTDL	The BRT timer data load value is 8 bits lower
F5C2H	BRTDH	The BRT timer data load value is 8 bits higher
--	--	--
F5E0H	UID0	UID<7:0>
F5E1H	UID1	UID<15:8>
F5E2H	UID2	UID<23:16>
F5E3H	UID3	UID<31:24>
F5E4H	UID4	UID<39:32>
F5E5H	UID5	UID<47:40>
F5E6H	UID6	UID<55:48>
F5E7H	UID7	UID<63:56>
F5E8H	UID8	UID<71:64>
F5E9H	UID9	UID<79:72>
F5EAH	UID10	UID<87:80>
F5EBH	UID11	UID<95:88>
--	--	--
F650H	LCDCON0	LCD control register 0
F651H	LCD_S0	LCD COM/SEG selects register 0
F652H	LCD_S1	LCD COM/SEG select register 1
F653H	LCD_S2	LCD COM/SEG selection register 2
F654H	LCDEN0	LCD function selects register 0
F655H	LCDEN1	LCD function select register 1
F656H	LCDEN2	LCD function selects register 2
--	--	--
F690H	LVDCON	Power supply monitor registers
F691H	BOOTCON	BOOT control registers
F692H	ADCLDO	ADC reference voltage control register

F693H	TS_REG	Temperature sensor register
F694H	LSECRL	The LSE timer data register is 8 bits lower
F695H	LSECRH	The LSE timer data register is 8 bits high
F696H	LSECON	LSE timer control registers
F697H	XT_SCM	LSE/HSE clock stop detection control register
F698H	PS_SCLK	The SPI clock input port assigns registers
F699H	PS_MOSI	SPI slave input port assignment registers
F69AH	PS_MISO	SPI host input port assignment register
F69BH	PS_NSS	SPI chip-selected input port assignment register
F69CH	PS_SCL	IIC clock input port assignment register
F69DH	PS_SDA	IIC data input port assignment registers
--	--	--
F69FH	PS_RXD0	UART0 data input port assignment register
--	--	--
F710H	LEDSDRP0L	LED SEG port P00-P03 drives the current selection register
F711H	LEDSDRP0H	LED SEG port P04-P07 drives the current selection register
F712H	LEDSDRP1L	LED SEG port P10-P13 drives the pull current selection register
F713H	LEDSDRP1H	Led SEG ports P14-P17 drive the current-selector register
F714H	LEDSDRP2L	LED SEG ports P20-P23 drive the current-pull selector register
F715H	LEDSDRP2H	LED SEG ports P24-P27 drive the current selection register
F716H	LEDSDRP3L	LED SEG port P30-P 33 drives the pull current selection register
--	--	--

3. Reset

Reset Time refers to the time from the time the chip resets to the time when the chip starts executing instructions, and its default design value is about 16ms. This time includes oscillator start time, configuration time. This reset time will exist whether the chip is powered on reset or otherwise caused by a reset. In addition, when the oscillator is selected as an external low-speed crystal oscillation (32.768KHz), the reset time (including the oscillation time) is about 1.5s (external capacitor 10pF~22pF) by default.

The chip can be reset in the following ways:

- Power-on reset;
- External reset;
- Low voltage reset;
- watchdog overflow reset;
- Software reset
- CONFIG status protection reset.
- Power-on configuration monitoring reset

When any of the above resets occur, all system registers will return to their default state, the program will stop running, the program counter PC will be cleared to zero, and the program will run from the reset vector 0000H after the reset is completed.

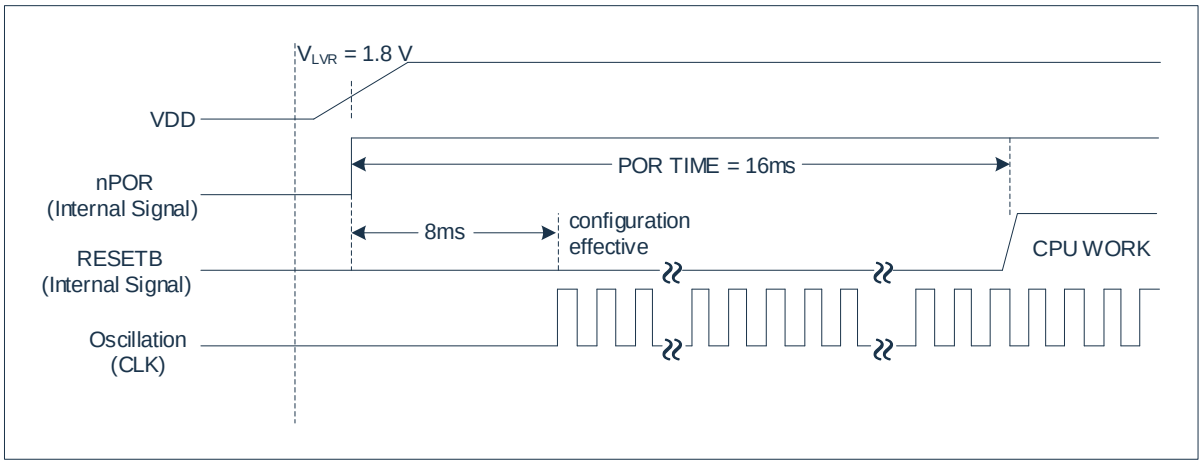
Any kind of reset situation requires a certain response time, and the system provides a perfect reset process to ensure the smooth progress of the reset operation.

3.1 Power-on Reset

Power-on reset is closely related to LVR operation. The process of powering up the system takes a gradually rising curve and takes some time to reach the normal level. The normal timing of the power-on reset is given below:

- Power-up: the system detects that the supply voltage is rising and waits for it to stabilize;
- System initialization: all system registers are set to initial values;
- The oscillator begins to work: the oscillator begins to provide the system clock;
- Execute the program: The power-up ends and the program starts running.

Stabilization Time defaults to 16ms, but if the configuration selects a 32.768KHz crystal oscillator, the settling time is about 1.5s. The power-on reset timing diagram is shown in the following figure:



Whether the system is power-on reset can be determined by the PORF (WDCON.6) flag bit. The types of resets that can be placed with a PORF flag of 1 are: power-on reset, LVR low voltage reset, power-on configuration monitoring reset, external reset, CONFIG status protection reset.

0x97	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WDCON	SWRST	PORF	--	--	WDTIF	WDTRF	WDTRE	WDTCLR
R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W
Reset value	0	1	0	0	0	0	0	0

Bit7 SWRST: Software reset control bit;
1: Perform a system software reset (write 0 clear after reset).
0: --
Bit6 PORF: Power-on reset marker;
1: The system resets the system at power-on (write 0 clear, do not need TA to write timing).
0: --
Bit5~Bit4 -- Reserved, must be 0.
Bit3 WDTIF: WDT overflow interrupt flag bit;
1= WDT overflow (write 0 cleared);
0= WDT does not overflow.
Bit2 WDTRF: WDT reset marker bit;
1= The system is reset by WDT (write 0 cleared);
0= The system is not reset by WDT.
Bit1 WDTRE: WDT reset enable bit;
1= Enable WDT reset CPU;
0= Disable WDT reset cpu.
Bit0 WDTCLR: WDT counter clear bit;
1= Clear WDT counter (hardware auto-zero);
0= Disables the WDT counter (write 0 is invalid).

3.2 External Reset

External reset refers to a reset signal from an external port (NRST) that resets the chip after being input by a Schmitt trigger. If the NRST pin remains low above about 16us (internal LSI clock sampled with 3 rising edges) during operating voltage range and stable oscillation, a reset is requested. After the internal state is initialized and reset state changes to "1", it takes 16ms of settling time for the internal RESETB signal to become "1", and the program starts at vector address 0000H.

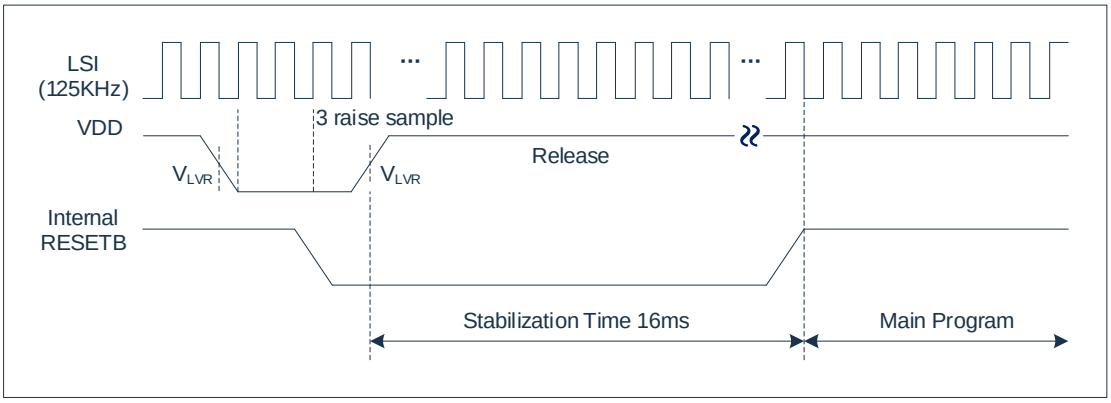
The process of reconfiguring the chip during Stabilization Time is the same as the configuration process for power-on reset. The external reset pin NRST and its pull-up resistor enable, configured via CONFIG.

3.3 LVR Low-voltage Reset

A low-voltage reset (LVR) function is integrated inside the chip, and when the system voltage VDD falls below the LVR voltage, the LVR is triggered and the system resets. The voltage point that triggers the reset can be set in CONFIG.

LVR module will request reset when it detects VDD<VLVR. In sleep mode (STOP) mode, lvr low-voltage reset disables.

The LVR low-voltage reset timing diagram is shown in the following figure:



The process of reconfiguring the chip during Stabilization Time is the same as the configuration process for power-on reset.

3.4 Watchdog Reset

Watchdog reset is a protective setting of the system. In normal condition, the watchdog timer is cleared to zero by the program. If an error occurs, the system is in an unknown state, the watchdog timer overflows, and the system resets. After the watchdog is reset, the system reboots into a normal state.

WDT counter is not addressable, the program starts counting after the power-on reset is completed, and it is recommended to clear the WDT counter first when setting the WDT register in order to accurately control the overflow time of the WDT.

The timing of the watchdog reset is as follows:

1) Watchdog timer status: the system detects whether the watchdog timer overflows, if overflowing, the system resets;
2) Initialization: all system registers are set to the default state;
3) Program: The reset is complete and the program starts running from 0000H.

The clock source for the WDT is provided by the system clock, and the timing base period of the WDT counter is Tsys. After the WDT overflow resets the CPU with all registers, the program executes immediately after 1 Tsys from 0000H. The WDT reset does not reconfigure the power-on reset. The overflow time of the watchdog can be set by the program, and the overflow time can be selected in the CKCON register WDS2-WTS0. Watchdog overflow times are shown in the table below:

WTS[2:0]	Watchdog Interval	Number of clocks	OVT@Fsys=16MHz	OVT@Fsys=48MHz
000	2^{17}	131072	8.192ms	2.731ms
001	2^{18}	262144	16.384ms	5.461ms
010	2^{19}	524288	32.768ms	10.923ms
011	2^{20}	1048576	65.536ms	21.845ms
100	2^{21}	2097152	131.072ms	43.691ms
101	2^{22}	4194304	262.144ms	87.381ms
110	2^{24}	16777216	1.048s	349.525ms
111	2^{26}	67108864	4.194s	1.398s

WDT can also be set to not reset the system, which can produce an interrupt.

3.5 Software Reset

Inside the chip, the program software resets, which relocates the program flow to the reset address 0000H and then runs the program again. The user-writable software reset control bit WDCON[7] (SWRST=1) implements a custom software reset. The software reset does not reconfigure the power-on reset.

3.6 CONFIG Status Protection Reset

CONFIG state protection reset is an enhanced protection mechanism of the system. During power-on reset, there is an internal set of 16-bit CONFIG registers that load the fixed code set in flash (A569H) and do not operate during normal operation. If, in the case of a particular non-program operation, the value of the register changes and is not equal to the original fixed code, and after several clock samples, the register continues to remain in a state that is not fixed code, the system will reset.

This reset mechanism prevents changes in configuration bits under certain conditions, so that the system enters an unpredicted state.

In normal operation, the clock of the sample register value is the internal RC fixed clock Fixed_Clock (8MHz, clock source from HSI) and low power clock (LSI 125KHz), once the value of the register is not a fixed code, force the LSI oscillator and HSI oscillator to be enabled, and the system clock switches to the LSI clock, if after 12 Fixed_Clock sampling or 3 LSI clock sampling, the register remains not in a fixed code state, The system generates a reset.

Under certain conditions, in order to prevent the oscillator from stopping, two clocks are used for sampling.

3.7 Power-on Configuration Monitor Reset

In the power-on configuration process, there is a configuration monitoring circuit inside the chip, if the power-on configuration time is too long, or the power-on configuration into a certain state can not be reconfigured, the internal monitoring circuit from the configuration to start timing, if more than the setting time, the monitoring circuit reset configuration module, so that the configuration module reconfiguration process. In case the system enters an unpredicted state when powered up.

The monitoring circuit operates at LSI (125KHz), the default monitoring time is 65ms, and if the 32.768KHz crystal oscillator is selected, the monitoring time is 2.1s.

4. Clock Structure

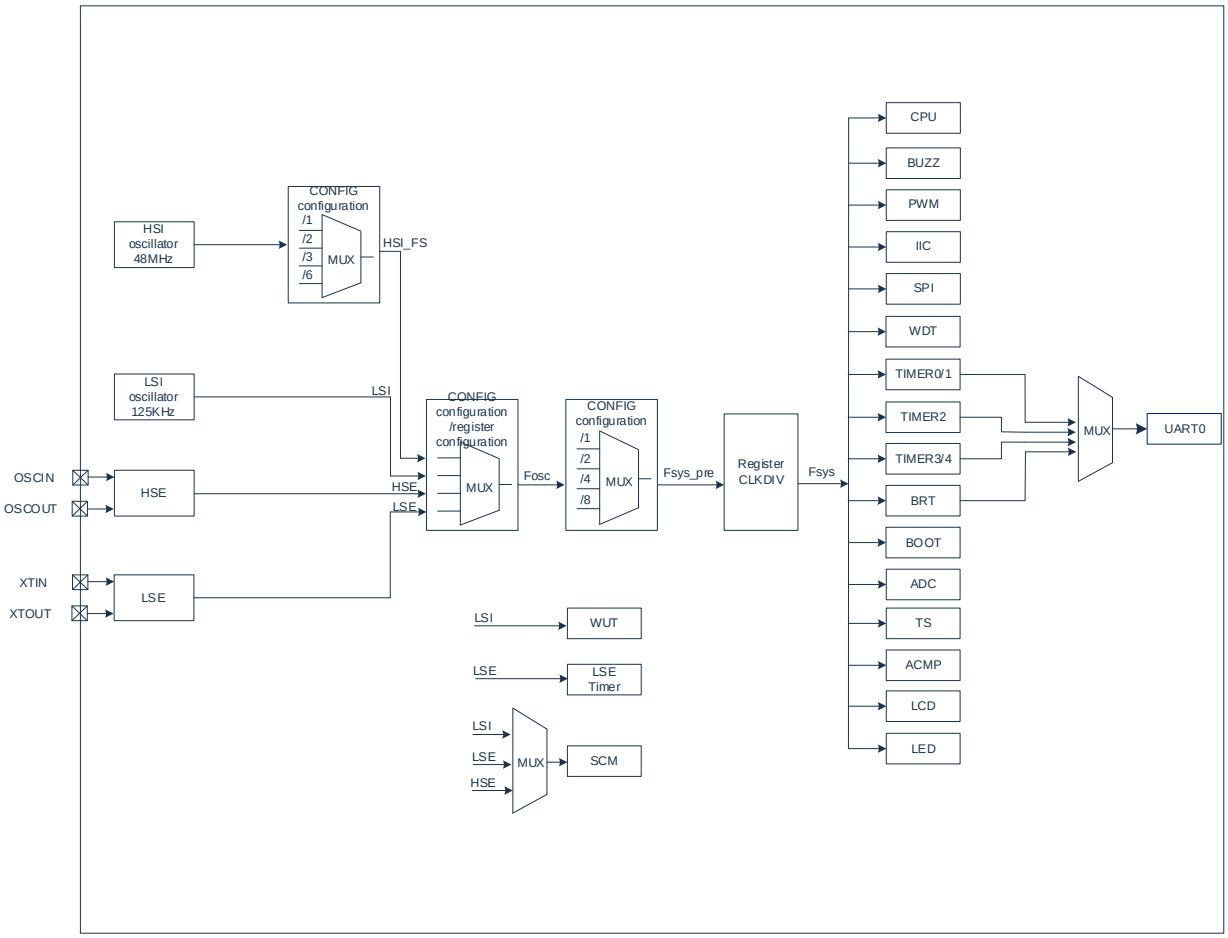
There are four types of clock sources for system clocks, and clock source and clock divider can be selected by setting the system configuration register or user register. The system clock sources are as follows:

- Internal high-speed oscillation HSI (48MHz).
- External high-speed oscillation HSE (8MHz/16MHz).
- External low-speed oscillation LSE (32.768KHz).
- Internal low-speed oscillation LSI (125KHz).

The default clock source of the chip is HSI, and the system clock runs in HSI after the chip reset is complete. If you need to change the system clock source, you can set the system configuration register (with the programming tool and host computer software), or through the user register configuration (follow the procedure to set the relevant register, as detailed below). When using external high-speed and low-speed oscillators as the system clock, the oscillation stop monitoring function is supported, as detailed below.

4.1 System Clock Structure

The system clock block diagram of each peripheral module is shown in the following figure:



4.2 Related Registers

4.2.1 Oscillator Control Register CLKDIV

0x8F	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CLKDIV	CLKDIV7	CLKDIV6	CLKDIV5	CLKDIV4	CLKDIV3	CLKDIV2	CLKDIV1	CLKDIV0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 CLKDIV<7:0>: System clock Fsys divider;
00H= Fsys=Fsys_pre;
Other = Fsys=Fsys_pre/（2*CLKDIV）（2,4... 510 division).

Modify the sequence of instructions required by CLKDIV (no other instructions can be inserted in the middle):

```
MOV     TA,#0AAH
MOV     TA,#055H
MOV     CLKDIV,#02H
```

4.2.2 System Clock Switching Register SCKSEL

0xD6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SCKSEL	--	--	--	SALT	WRITE	CKSEL2	CKSEL1	CKSEL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit5 --: Reserved, all must be 0.
Bit4 SALT: System clock configuration selection;
1= Based on the clock source configured > CKSEL < 2:0;
0= The clock source configured by CONFIG is the standard (power-up default selection).
Bit3 WRITE: Write enable, generate a pulse, perform clock switching;
1= Switch clocks (you need to wait for the switching target clock source to stabilize bit 1 before you can write 1);
0= Do not switch clocks.
Bit2~Bit0 CKSEL<2:0>: System clock source select bit;
111= LSI;
110= LSE;
101= HSE;
100= HSI;
Other = Invalid value, access prohibited.

After the clock source is switched, the system will successfully switch within several system clock cycles, and it is recommended that the program execute 6 NOPs before executing other instructions.

Modify the sequence of instructions required by SCKSEL (no other instructions can be inserted in between):

```
MOV     TA,#0AAH
MOV     TA,#055H
MOV     SCKSEL,#05H
```

4.2.3 System Clock Status Register SCKSTAU

0xD7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SCKSTAU	LSI_F	LSE_F	HSE_F	HSI_F	--	--	--	--
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 LSI_F: Low-speed internal steady-state bit;
1= Stability;
0= Not stable.

Bit6 LSE_F: Steady state bit of low-speed external crystal;
1= Stability;
0= Not stable.

Bit5 HSE_F: High-speed external crystal stabilized state bit;
1= Stability;
0= Not stable.

Bit4 HSI_F: High-speed internal clock steady-state bit;
1= Stability;
0= Not stable.

Bit3 -- Forbidden Access

Bit2 -- Forbidden Access

Bit1 -- Forbidden Access

Bit0 -- Forbidden Access

4.2.4 System Clock Monitor Register SCM

F697H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
XT_SCM	SCMEN	SCMIE	--	--	--	--	SCMIF	SCMSTA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
Reset value	0	0	0	0	0	0	0	0

Bit7 SCMEN: Oscillation stop detection module enable;
1= Enable;
0= Disable.

Bit6 SCMIE: Stop detects interrupt enable bits (the interrupt and LSE timer interrupt share an interrupt entry);
1= Enable;
0= Disable.

Bit5~Bit2 -- Reserved, all must be 0.

Bit1 SCMIF: Stop interrupt flag bit;
1= Indicates oscillation stopping;
0= The software clears 0, and after clearing 0, it will automatically switch to the HSE/LSE frequency (only the software can clear 0).

Bit0 SCMSTA: Stop status bit, read-only;
1= Indicates oscillation stopping;
0= Shutdown recovery.

Note:

1) Both SCMIF and SCMSTA can reflect the state of the HSE/LSE system clock. The biggest difference between the two is that when the HSE/LSE is stopped, the SCMSTA will remain at a high level until the HSE/LSE is restored; SCMIF can also reflect HSE/LSE shutdown, but it can produce an interrupt (interrupt enable is required), or it can clear the SCMIF through the register, and the frequency will switch back to HSE/LSE after clearing (if it is still in a stalled state at this time, the interrupt will be triggered again).

2) After the oscillation is stopped, the main frequency will be cut to HSI by HSE/LSE, if HSE/LSE is restored, SCMSTA will automatically clear zero, and the main frequency will also be automatically switched back to HSE/LSE by HSI.

4.2.5 Function Clock Control Registers

Watchdog overflow time/timer clock source selection register CKCON

0x8E	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CKCON	WTS2	WTS1	WTS0	T1M	T0M	--	--	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	1	1	1

Bit7~Bit5 WTS<2:0>: WDT overflow time selection bits;
000= 2^{17}*Tsys;
001= 2^{18}*Tsys;
010= 2^{19}*Tsys;
011= 2^{20}*Tsys;
100= 2^{21}*Tsys
101= 2^{22}*Tsys;
110= 2^{24}*Tsys;
111= 2^{26}*Tsys。

Bit4 T1M: Timer1's clock source select bit;
0= Fsys/12;
1= Fsys/4。

Bit3 T0M: Clock source select bit of Timer0;
0= Fsys/12;
1= Fsys/4。

Bit2~Bit0 -- Reserved, both must be 1.

UART0 baud rate selection register FUNCCR

0x91	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FUNCCR	--	--	--	--	--	UART0_CKS2	UART0_CKS1	UART0_CKS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

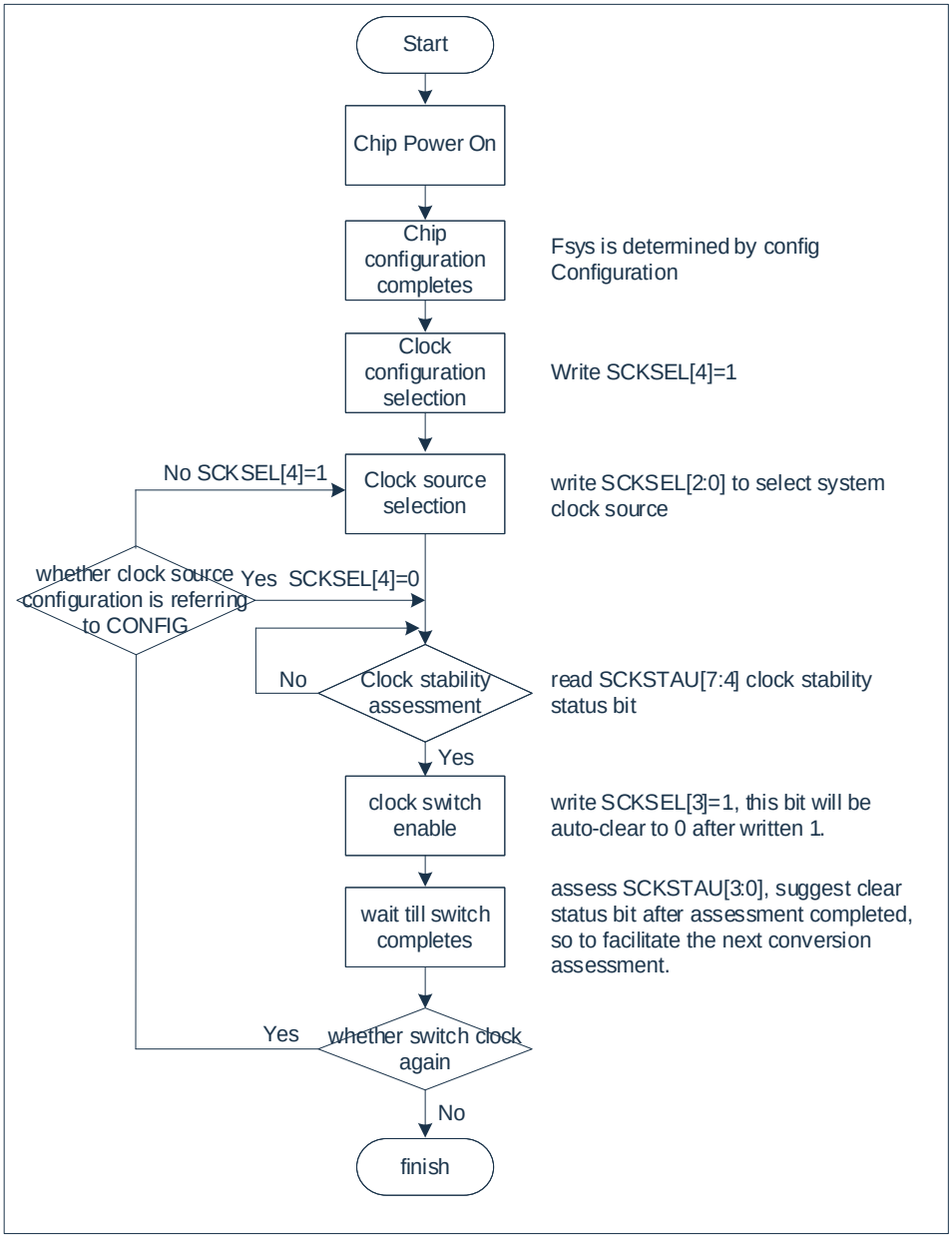
Bit7 -- Reserved, must be 0.

Bit6~Bit3 -- Reserved, must be 0.

Bit2~Bit0 UART0_CKS<2:0>: Timer clock source selection for UART0;
000= Overflow clock for Timer1;
001= Overflow clock for Timer4;
010= Overflow clock for Timer2;
011= BRT overflow clock;
Other = Forbidden Access.

4.3 System clock switching

The system clock switching steps are shown in the following figure:



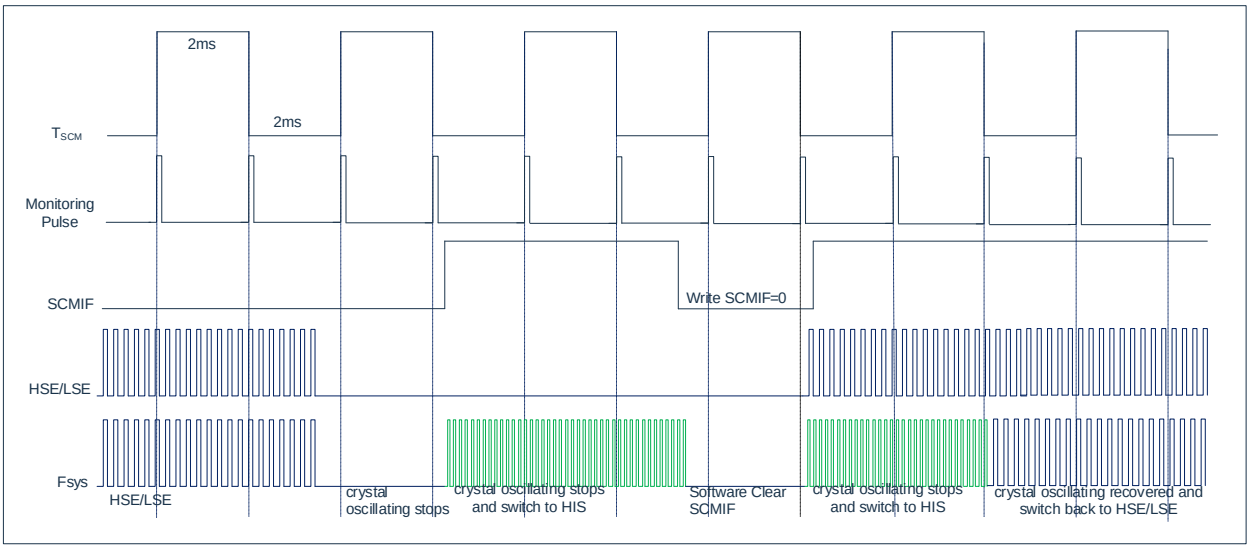
Note: When switching the system clock source, ensure that the corresponding clock source exists and is configured correctly.

4.4 System Clock Monitoring

System Clock Monitoring (SCM: system clock monitoring) is a monitoring and protection circuit designed to prevent the failure of the system due to crystal oscillation suspension. When using HSE/LSE as the system clock, once the HSE/LSE clock stops, the system will force the HSI clock source to start, and after the HSI is stabilized, the system will run at 8MHz main frequency, and then if the HSE/LSE clock is restored and stable, the system clock will automatically switch back from the HSI back to HSE/LSE.

The SCM module monitors the system clock HSE/LSE every 4ms, and the duty cycle TSCM has a duty cycle of 1:1. SCM performs oscillation-stop monitoring of HSE/LSE during period when TSCM is high, and T_{SCM} processes monitoring results during low level if HSE/ is detected LSE stops, the system clock is switched to HSI, and the stop interrupt flag SCMIF is set to 1. If SCMIF is cleared, the system clock will automatically switch back to HSE/LSE even if the HSE/LSE has stopped.

The system clock monitoring block diagram is shown in the following figure:



5. Power Management

Low-power modes fall into 2 categories:

- IDLE: Idle mode
- STOP: Sleep mode

When users use C language for program development, it is strongly recommended to use IDLE and STOP macros to control the system mode, and do not directly set THE IDLE and STOP bits. The macros are as follows:

Enter idle mode: IDLE();

Enter sleep mode: STOP();

5.1 Power Management Register PCON

0x87	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON	SMOD0	--	--	--	--	THEIR	STOP	IDLE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7	SMOD0:	UART0 baud rate multiplier;
	0=	UART0 baud rate is normal;
	1=	The UART0 baud rate doubles.
Bit6~Bit3	--	Reserved, all must be 0.
Bit2	THEIR:	STOP status function wake-up enable bit; (The system can be restarted by a power-down reset or an enabled external reset regardless of the SWE value)
	0=	Disables functional wake-up;
	1=	Allows function wake-up (wake-up by external interrupts and timed wake-ups).
Bit1	STOP:	Sleep state control bit;
	0=	Not in hibernation;
	1=	Enters a hibernate state (exits STOP mode to automatically clear zero).
Bit0	IDLE:	Idle state control bit;
	0=	Not in the idle state;
	1=	Enter the idle state (exit IDLE mode to clear automatically).

5.2 Power Supply Monitor Register LVDCON

The MCU comes with a power supply detection function. If the LVD module enable (LVDEN=1) is set and the voltage monitoring point LVDSEL is set, when the power supply voltage drops below the LVD setpoint, an interrupt will be generated to alert the user.

If the LVD module is enabled before hibernation, the hardware will not close the module circuit after entering hibernation, and a software shutdown is required (LVDEN=0).

0xF690	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LVDCON	LVDSEL3	LVDSEL2	LVDSEL1	LVDSEL0	LVDEN	--	LVDINTE	LVDINTF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit4 LVDSEL<3:0>: LVD voltage monitoring points;

0000= 2.00V; 1000= 3.21V;
0001= 2.16V; 1001= 3.42V;
0010= 2.31V; 1010= 3.62V;
0011= 2.45V; 1011= 3.81V;
0100= 2.60V; 1100= 4.00V;
0101= 2.73V; 1101= 4.20V;
0110= 2.88V; 1110= 4.43V;
0111= 2.98V; 1111= 4.60V。

Bit3 LVDEN: LVD enable;
0= Disable;
1= Enable.

Bit2 -- Reserved, must be 0.

Bit1 LVDINTE: LVD interrupt enable bit;
0= LVD interrupt disabled;
1= LVD interrupt enablement.

Bit0 LVDINTF: LVD interrupt flag bit;
0= The supply voltage is higher than the monitoring voltage;
1= The supply voltage is lower than the monitor voltage (software zero).

5.3 IDLE Idle Mode

In this mode, only the CPU clock source is turned off. As a result, peripheral functions such as timers, PWM, I2C and clock generators (HSI/crystal drivers) are still functioning in this state.

After the system enters idle mode, it can be woken up by any interrupt, enter the interrupt handler after waking, and continue to perform the post-hibernation operation after the interrupt returns.

If idle mode is entered in the interrupt service program, the system can only be woken up by a higher priority interrupt.

5.4 STOP Sleep Mode

In this mode, all circuits except the LVD module and LSE module are shut down (the LVD/LSE module must be closed by software), the system is in a low-power mode, and the digital circuits are not working.

5.4.1 Sleep Wakes up

After entering the sleep mode, you can turn on the sleep wake function (SWE=1 need to be set) to wake up the sleep mode. There are several ways to wake up sleep mode:

1) INT0/1 interrupt

With INT0/1 interrupt wake-up sleep mode, the global interrupt enable must be turned on with the INT0/1 interrupt enable before entering hibernation to wake up the system. INT0, INT1 interrupt-related registers include IE, IP, TCON, IO multiplexed mapping registers, INT0/1 interrupt wake-up can only drop along the interrupt wake-up sleep.

2) External (GPIO) Interrupt

With an external GPIO interrupt wake-up, the global interrupt enable and the port interrupt enable must be turned on before entering hibernation to wake up the system. External GPIO interrupt wake-up can be selected for rising-edge, falling-edge, dual-edge interrupt wake-up sleep, and the interrupt wake-up edge is set by the external interrupt control register PxnEICFG.

3) WUT timed wake-up

To wake up by WUT, the timed wake function must be turned on before entering hibernation, and the hibernation state must be set to the time of wake-up. The clock source of the timed wake-up circuit is provided by the LSI (Low Power Oscillator), and the timed wake-up function is turned on automatically when the timed wake-up function is turned on in the sleep state.

4) LSE timed wake-up

To wake up by LSE timing, the LSE module enable, count enable, and timed wake function must be turned on before entering hibernation, and the hibernation state must be set to wake up time.

5.4.2 Wake-up Wait State

Whether it is an INT0/1 interrupt, an external GPIO interrupt, or a WUT timed wake-up, LSE timed wake-up sleep mode, after the interrupt is generated or the configured time arrives, it takes a period of time to wake up the system and execute the next instruction of the program. After the interrupt is generated or the configured time expires, the system oscillator starts, but the oscillation frequency is not stable, the CPU is not working, the PC still stops in hibernation, and the system needs to wait for a period of time to provide the clock to the CPU. The wait time to wake up the CPU is set in the flashing CONFIG, and the wait time can be set to 50us~1s. After the wake-up wait time, the MCU believes that the system clock has stabilized, and then provides the clock to the CPU, and the program continues to execute.

If both the internal wake-up timer and the external interrupt wake-up function are turned on, either wake-up mode can wake up the CPU after the system enters sleep mode. If the internal timer wakes up the oscillator first and then has an external interrupt input, after the wake-up wait time has elapsed, the program executes the interrupt handler and then continues with the instructions after the hibernation operation.

5.4.3 Sleep Wake-up Time

The total wake-up time of the system with an external interrupt wake-up system is:

Power Manager Settling Time (200us) + Wake-up Wait Time

The total wake-up time of the system with timed wake-up is:

Power Manager Settling Time (200us) + Timing of wake-up timer + Wake-up wait time

(The above given time condition is Fsys>1MHz)

5.4.4 Reset Operation Under Sleep

In sleep mode, the system can also be restarted by power-down reset or external reset, independent of the value of SWE, even if SWE=0 can also restart the system by the above reset operation.

Power-down reset: No other conditions are required, VDD is reduced to 0V and then powered back on to the working voltage and enters the power-on reset state.

External reset: you need to open the external reset function, the relevant port is configured as a dedicated reset port, the reset port remains low >1us when sleeping, the system generates a reset, release the reset port, then the system restarts.

5.4.5 Sleep Power Consumption in Debug Mode

The sleep state in debug mode does not reflect the actual chip sleep state.

In debug mode, after the system enters a sleep state, the associated power management circuit, the oscillator does not turn off, but continues to turn on. Wake-up operations can also be performed in debug mode in the same way as normal mode.

Therefore, in this state, the sleep current obtained by the test is not the real sleep power consumption. It is recommended to turn off debug mode after the development of the sleep wake function is completed in debug mode, and then restart the system, at which time the measured current is the actual sleep power consumption.

5.4.6 Example of a Sleep Mode Application

Before the system enters the sleep mode, if the user needs to obtain a small sleep current, please confirm the state of all I/O, if there is a suspended I / O port in the user solution, set all the suspended ports as output ports, to ensure that each input port has a fixed state, to avoid I / O as the input state, the port line level is in a static state and increase the sleep current; Power down ADC modules, LSE modules, LVD modules, and other peripherals to reduce sleep current.

Example: A processing procedure (assembler) that goes into hibernation when using a timed wake-up

```
SLEEP_MODE：
        MOV     WUTCRL,#31h
        MOV     WUTCRH,#80h
        MOV     P0TRIS,#0FFh
        MOV     P0,#0FFh
        MOV     P1TRIS,#0FFh
        MOV     P1,#0FFh
        MOV     P2TRIS,#0FFh
        MOV     P2,#0FFh
        Instructions to turn off other functions
        MOV     PCON,#06H       ; Perform a hibernation operation that can be woken up
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP     ; The instruction to perform the sleep operation must be followed by 6 NOP instructions
        Other action instructions after wake-up
```

6. Interrupt

6.1 Interrupt Overview

The chip has 20 interrupt sources and interrupt vectors:

Interrupt source	Interrupt description	Interrupt vector	Sibling priority sequence
INT0	External interrupt 0	0-0x0003	1
Timer0	Timer 0 interrupt	1-0x000B	2
INT1	External interrupt 1	2-0x0013	3
Timer1	Timer 1 interrupt	3-0x001B	4
UART0	TI0 or RI0	4-0x0023	5
Timer2	Timer 2 interrupt	5-0x002B	6
--	--	6-0x0033	7
P0EXTIF<7:0>	P0 port external interrupt	7-0x003B	8
P1EXTIF<7:0>	P1 port external interrupt	8-0x0043	9
P2EXTIF<7:0>	P2 port external interrupt	9-0x004B	10
P3EXTIF<7:0>	P3 port external interrupt	10-0x0053	11
--	--	11-0x005B	12
--	--	12-0x0063	13
--	--	13-0x006B	14
ACMP	Comparator interrupt	14-0x0073	15
Timer3	Timer 3 interrupt	15-0x007B	16
Timer4	Timer 4 interrupt	16-0x0083	17
--	--	17-0x008B	18
PWM	PWM interrupt	18-0x0093	19
ADC	ADC interrupt	19-0x009B	20
WDT	WDT interrupt	20-0x00A3	21
I2C	I2C interrupt	21-0x00AB	22
SPI	SPI interrupt	22-0x00B3	23
--	--	23-0x00BB	24
--	--	24-0x00C3	25
LSE_Timer/SCM	LSE timer interrupt/SCM	25-0x00CB	26
LVD	LVD power-down interrupt	26-0x00D3	27
--	--	27-0x00DB	28

LSE timers LSE_Timer interrupts and oscillation stop monitoring SCM interrupts share a single interrupt vector entry, but they have independent interrupt enable bits.

The chip specifies two interrupt priorities, allowing for two levels of interrupt nesting. When an interrupt has already responded, if a high-level interrupt is requested, the latter can interrupt the former, implementing interrupt nesting.

6.2 External Interrupts

6.2.1 INT0/INT1 Interrupt

The chip supports the 8051 native INT0, INT1 external interrupt, INT0/INT1 can choose to falling edge or low level trigger interrupt, the relevant control register is TCON. INT0 and INT1 occupy two interrupt vectors.

6.2.2 GPIO Interrupt

Each GPIO pin of the chip supports an external interrupt and can support falling/rising/dual edge interrupts, with the edge trigger type configured through the Px_NEICFG registers. For example, configure the P13 port as a falling edge interrupt:

P13CFG=0x00; //Set P13 to GPIO

P1TRIS&=0xF7; Set P13 as the input port

P13EICFG=0x02; //Set P13 as a falling edge trigger interrupt

GPIO interrupts occupy a total of 4 interrupt vectors:

Port P0 occupies an interrupt vector 0x003B;

Port P1 occupies an interrupt vector 0x0043;

Port P2 occupies an interrupt vector 0x004B;

Port P3 occupies an interrupt vector 0x0053.

When an interrupt occurs, the interrupt service program can first determine which port triggered the interrupt, and then process it accordingly.

6.3 Interrupt With Sleep Wake-up

After the system enters sleep mode (STOP wakeable mode), each external interrupt can be set to wake up the system.

INT0/INT1 interrupt wake-up system needs to turn on the corresponding interrupt enable and global interrupt enable, and the wake-up mode is the falling edge wake-up (INT0/INT1 wake-up mode and interrupt trigger mode select bit IT0/IT1 are independent).

GPIO interrupt wake-up system, it is recommended to set the corresponding port interrupt trigger edge mode before entering sleep mode (GPIO wake-up mode is the same as interrupt trigger edge mode, you can choose rising edge/ falling edge /double edge wake-up), and turn on the corresponding interrupt enable and global interrupt enable.

After the system is woken up by an external interrupt, it first enters the interrupt service program to handle the interrupt wake task, and after exiting the interrupt service program, the system continues to perform instructions after the hibernation operation.

6.4 Interrupt Register

6.4.1 Interrupt Mask Registers

6.4.1.1 Interrupt Mask Register IE

Interrupt mask register IE is a read-write register that can be operated bitwise. When an interrupt condition arises, the interrupt flag bit will be set to 1 regardless of the state of the corresponding interrupt enable bit or the global enable bit EA. The user software should ensure that the corresponding interrupt flag bits are cleared to zero before enabling an interrupt.

0xA8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IE	SHE	--	ET2	ES0	ET1	EX1	ET0	EX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 SHE: Global interrupt enable bits;
1= Enable all unblocked interrupts;
0= Disable all interrupts.
Bit6 -- Reserved, must be 0.
Bit5 ET2: TIMER2 Global interrupt enable bits;
1= Enable all interrupts of TIMER2;
0= All interrupts of TIMER2 are disabled.
Bit4 ES0: UART0 interrupt enable bit;
1= Enable UART0 interrupts;
0= Disable UART0 Interrupt.
Bit3 ET1: TIMER1 interrupt enable bit;
1= Enable TIMER1 interrupts;
0= Disable TIMER1 Interrupt.
Bit2 EX1: External interrupt 1 interrupt enable bits;
1= Enable external interrupt 1 interrupt;
0= Disable external interrupt 1 interrupt.
Bit1 ET0: TIMER0 interrupt enable bits;
1= Enable TIMER0 interrupts;
0= Disable TIMER 0 Interrupt.
Bit0 EX0: External interrupt 0 interrupt enable bit;
1= Enable external interrupt 0 interrupts;
0= Disable external interrupt 0 interrupt.

6.4.1.2 Interrupt Mask Register EIE2

0xAA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIE2	SPIIE	I2CIE	WDTIE	ADCIE	PWMIE	--	ET4	ET3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 SPIIE: SPI interrupt enable bit;
1= Enable SPI interrupts;
0= Disable SPI Interrupt.
Bit6 I2CIE: I2C Interrupt enable bit;
1= Enabl I2C nterrupts;
0= Disable I2C Interrupt.
Bit5 WDTIE: WDT interrupt enable bit;
1= Enable WDT overflow interrupts;
0= Disable WDT overflow interrupts.
Bit4 ADCIE: ADC interrupt enable bit;
1= Enable ADC interrupts;
0= Disable ADC interrupts.
Bit3 PWMIE: PWM global interrupt enable bit;
1= Enable all PWM interrupts;
0= Disable all PWM interrupts.
Bit2 -- Reserved, must be 0.
Bit1 ET4: Timer4 interrupt enable bit;
1= Enable Timer4 interrupts;
0= Disable Timer4 Interrupt.
Bit0 ET3: Timer3 interrupt enable bit;
1= Enable Timer3 interrupts;
0= Disable Timer3 Interrupt.

6.4.1.3 Timer2 Interrupt Mask Register T2IE

0xCF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2IE	T2OVIE	T2EXIE	--	--	T2C3IE	T2C2IE	T2C1IE	T2C0IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 T2OVIE: Timer2 overflow interrupt enable bits;
1= Interrupts enabled;
0= Disable Interrupt.
Bit6 T2EXIE: Timer2 external loading interrupt enable bits;
1= Interrupts enabled;
0= Disable Interrupt.
Bit5~Bit4 -- Reserved, must be 0.
Bit3 T2C3IE: Timer2 compares channel 3 interrupt enable bits;
1= Interrupts enabled;
0= Disable Interrupt.
Bit2 T2C2IE: Timer2 compares channel 2 interrupt enable bits;
1= Interrupts enabled;
0= Disable Interrupt.
Bit1 T2C1IE: Timer2 compares channel 1 interrupt enable bits;
1= Interrupts enabled;
0= Disable Interrupt.
Bit0 T2C0IE: Timer2 compares channel 0 interrupt Enable bits;
1= Interrupts enabled;
0= Disable Interrupt.

If you want to enable the interrupt of Timer2, you also need to turn on the global interrupt enable bit ET2=1 of Timer2 (IE.5=1)

6.4.1.4 P0 Interrupt Control Register P0EXTIE

0xAC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0EXTIE	P07IE	P06IE	P05IE	P04IE	P03IE	P02IE	P01IE	P00IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 P0iIE: P0i port interrupt Enable bits (i=0-7);
1= Interrupts enabled;
0= Disable Interrupt.

6.4.1.5 Port P1 Interrupt Control Register P1EXTIE

0xAD	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P1EXTIE	P17IE	P16IE	P15IE	P14IE	P13IE	P12IE	P11IE	P10IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 P1iIE: P1i interrupt enable bits (i=0-7);
1= Interrupts enabled;
0= Disable Interrupt.

6.4.1.6 P2 Interrupt Control Register P2EXTIE

0xAE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2EXTIE	--	--	P25IE	P24IE	P23IE	P22IE	P21IE	P20IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit6 -- Reserved, must be 0.
Bit5~Bit0 P2iIE: P2i port interrupt Enable bits (i=0-5);
1= Interrupts enabled;
0= Disable Interrupt.

6.4.1.7 P3 Port Interrupt Control Register P3EXTIE

0xAF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P3EXTIE	--	--	--	--	P33IE	P32IE	P31IE	P30IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit4 -- Reserved, must be 0.
Bit3~Bit0 P3iIE: P3i port interrupt Enable bits (i=0-3);
1= Interrupts enabled;
0= Disable Interrupt.

6.4.2 Interrupt Priority Controls the Register

6.4.2.1 Interrupt Priority Control Register IP

Interrupt priority control register IP is a read-write register that can be operated bitwise.

0xB8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IP	--	--	PT2	PS0	PT1	PX1	PT0	PX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit6 -- Reserved, must be 0.

Bit5 PT2: TIMER2 interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

Bit4 PS0: UART0 interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

Bit3 PT1: TIMER1 interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

Bit2 PX1: External interrupt 1 interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

Bit1 PT0: TIMER0 interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

Bit0 PX0: External interrupt 0 interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

6.4.2.2 Interrupt Priority Control Register EIP1

0xB9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP1	PACMP	--	--	--	PP3	PP2	PP1	PP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 PACMP: Analog comparator interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.
Bit6~Bit4 -- Reserved, must be 0.
Bit3 PP3: P3 port interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.
Bit2 PP2: P2 port interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.
Bit1 PP1: P1 port interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.
Bit0 PP0: P0 port interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

6.4.2.3 Interrupt Priority Control Rregister EIP2

0xBA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP2	PSPI	PI2C	PWDT	PADC	PPWM	--	PT4	PT3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 PSPI: SPI interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.
Bit6 PI2C: I2C interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.
Bit5 PWDT: WDT interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.
Bit4 PADC: ADC interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.
Bit3 PPWM: PWM interrupt priority control bit
1= Set to High-level Interrupt;
0= Set to low-level interrupt.
Bit2 -- Reserved, must be 0.
Bit1 PT4: TIMER4 interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

Bit0 PT3: TIMER3 interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

6.4.2.4 Interrupt Priority Control Register EIP3

0xBB	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP3	--	--	--	--	PLVD	PLSE	--	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit4 -- Reserved, all must be 0.
Bit3 PLVD: LVD interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.
Bit2 PLSE: LSE interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.
Bit1~Bit0 -- Reserved, all must be 0.

6.4.3 Interrupt Flag Bit Register

6.4.3.1 Timer0/1, INT0/1 Interrupt Flag Bit Register TCON

0x88	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 TF1: Timer1 counter overflow interrupt flag bit;
1= Timer1 counter overflow, when entering the interrupt service program, the hardware is automatically cleared, and the software can also be cleared;
0= The Timer1 counter has no overflow.

Bit6 TR1: Timer1 operational control bit;
1= Timer1 starts;
0= Timer1 closes.

Bit5 TF0: Timer0 counter overflow interrupt flag bit;
1= Timer0 counter overflow, when entering the interrupt service program, the hardware is automatically cleared, and the software can also be cleared;
0= The Timer0 counter has no overflow.

Bit4 TR0: Timer0 operational control bit;
1= Timer0 starts.
0= Timer0 closes.

Bit3 IE1: External interrupt 1 flag;
1= External interrupt 1 generates an interrupt, the hardware is automatically cleared when entering the interrupt service program, and the software can also be cleared;
0= External interrupt 1 did not produce an interrupt.

Bit2 IT1: External interrupt 1 trigger mode control bit;
1= Falling edge trigger;
0= Low level triggering.

Bit1 IE0: External interrupt 0 flag;
1= External interrupt 0 generates an interrupt, the hardware is automatically cleared when entering the interrupt service program, and the software can also be cleared;
0= External interrupt 0 did not produce an interrupt.

Bit0 IT0: External interrupt 0 trigger mode control bit;
1= Falling edge trigger;
0= Low level triggering.

6.4.3.2 Timer2 Interrupt Flag Bit Register T2IF

0xC9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2IF	TF2	T2EXIF	--	--	T2C3IF	T2C2IF	T2C1IF	T2C0IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 TF2: Timer2 counter overflow interrupt flag bit;
1= Timer2 counter overflow, software zeroing is required;
0= The Timer2 counter has no overflow.

Bit6 T2EXIF: Timer2 externally loaded flag bits;
1= The T2EX port of Timer2 generates a falling edge, which requires software clearance;
0= --

Bit5~Bit4 -- Reserved, must be 0.

Bit3 T2C3IF: Timer2 Compare/Capture Channel 3 Flag Bits;
1= Timer2 Compare channel 3 {CCH3:CCL3}={TH2:TL2} or capture channel 3 produces a capture operation that requires software zeroing.
0= --

Bit2 T2C2IF: Timer2 Compare/Capture Channel 2 Flag Bits;
1= Timer2 Compare channel 2 {CCH2:CCL2}={TH2:TL2} or capture channel 2 to produce a capture operation that requires software zeroing.
0= --

Bit1 T2C1IF: Timer2 Compare/Capture Channel 1 Flag Bits;
1= Timer2 Compare channel 1 {CCH1:CCL1}={TH2:TL2} or capture channel 1 to produce a capture operation that requires software zeroing.
0= --

Bit0 T2C0IF: Timer2 Compare/Capture Channel 0 Flag Bits;
1= Timer2 Compare channel 0{RLDH:RLDL}={TH2:TL2} or capture channel 0 produces a capture operation that requires software zeroing.
0= --

6.4.3.3 Peripheral Interrupt Flag Bit Register EIF2

0xB2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIF2	SPIIF	I2CIF	--	ADCIF	PWMIF	--	TF4	TF3
R/W	R	R	--	R/W	R	--	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 SPIIF: SPI global interrupt indicator bit, read-only;
1= SPI generates an interrupt, (this bit is automatically cleared after the specific interrupt flag is cleared);
0= The SPI did not produce an interrupt.
Bit6 I2CIF: I2C global interrupt indicator bit, read-only;
1= I2C produces an interrupt, (after clearing the specific interrupt flag, this bit is automatically cleared);
0= I2C did not produce an interrupt.
Bit5 -- Reserved, must be 0.
Bit4 ADCIF: ADC interrupt flag bit;
1= ADC conversion is completed, and software zeroing is required;
0= The ADC conversion was not completed.
Bit3 PWMIF: PWM global interrupt indicator bit, read-only;
1= PWM generates an interrupt, (after the specific interrupt flag is cleared, this bit is automatically cleared);
0= The PWM did not produce an interrupt.
Bit2 -- Reserved, must be 0.
Bit1 TF4: Timer4 timer overflow interrupt flag bit;
1= Timer4 timer overflow, the hardware is automatically cleared when entering the interrupt service program, and the software can also be cleared;
0= The Timer4 timer has no overflow.
Bit0 TF3: Timer3 timer overflow interrupt flag bit;
1= Timer3 timer overflow, when entering the interrupt service program, the hardware is automatically cleared, and the software can also be cleared;
0= The Timer3 timer has no overflow.

6.4.3.4 SPI Interrupt Flag Bit Register SPSR

0xED	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPSR	SPISIF	WCOL	--	--	--	--	--	SSCEN
R/W	R	R	--	--	--	--	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 SPISIF: SPI transmission completion interrupt flag bit, read-only;
1= SPI transmission is completed (read SPSR first, then read/write SPDR and then clear zero);
0= The SPI was not transmitted.
Bit6 WCOL: SPI write violation interrupt flag bit, read-only;
1= When the SPI transfer is not completed, a collision of the write SPDR operation occurs (read the SPSR first, then clear the SPDR after reading/writing the SPDR);
0= No write conflicts.
Bit5~Bit1 -- Reserved, must be 0.
Bit0 SSCEN: SPI master mode NSS output control bit.
1= When the SPI is idle, the NSS output is high;
0= NSS output registers the contents of the SSCR.

6.4.3.5 I2C Master Mode Interrupt Flag Registers I2CMCR/I2CMSR

0xF5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CMCR	RSTS	--	--	--	ACK	STOP	START	RUN
I2CMSR	I2CMIF	BUS_BUSY	IDLE	ARB_LOST	DATA_ACK	ADDR_ACK	ERROR	BUSY
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 RSTS: I2C master module reset control bit;
1= Reset the master module (I2C registers for the entire master module, including I2CMSR);
0= The interrupt flag bit in I2C master mode is clear to 0.
I2CMIF: I2C Master mode interrupt flag bit;
1= In master mode, send/receive completes, or a transmission error occurs. (Software zero, write 0 to clear);
0= No interrupt was generated.
Bit6~Bit0 Controland flag bits in I2C master mode, see I2CM description for details.

6.4.3.6 I2C Slave Mode Status Register I2CSSR

0xF2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CSSR	--	--	--	--	--	SENDFIN	TREQ	RREQ
R/W	--	--	--	--	--	R	R	R
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit3 -- Reserved, must be 0.
Bit2 SENDFIN: I2C slave mode send operation completion flag bit, read-only;
1= The data is no longer required by the master device, the TREQ is no longer set to 1, and the data transfer has been completed. (Automatic zeroing after reading I2CSCR).
0= --
Bit1 TREQ: I2C Slave mode prepares to send flag bits, read-only;
1= As the transmitting device has been addressed, the master device is ready to receive data. (Auto zero after writing I2CSBUF).
0= --
Bit0 RREQ: I2C slave mode receives completion flag bits, read-only;
1= Received. (Automatic zeroing after reading I2CSBUF);
0= Not received.

The relevant status bits for I2C slave mode are also interrupt flag bits

Note: The I2C master mode interrupt shares the same interrupt vector (00ABH) as the slave mode interrupt

6.4.3.7 UART Control Register SCON0

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SCON0	U0SM0	U0SM1	U0SM2	U0REN	U0TB8	U0RB8	TI0	RI0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

BANK0: Register SCON0 address 0x98.

Bit7~Bit2 U0SM0、U0SM1, U0SM2, U0REN, U0TB8, U0RB8: See the UART0 function description for details

Bit1 TI0: Send interrupt flag bits (requires software zeroing);

1= Indicates that the send buffer is empty, and you can send the frame data.

0= --

Bit0 RI0: Receive interrupt flag bits (requires software zeroing);

1= Indicates that the receive buffer is full, and the next frame of data can be received after reading.

0= --

TI0 and RI0 occupy the same interrupt vector and require queries to determine whether to receive an interrupt or send an interrupt.

6.4.3.8 P0 Port Interrupt Flag Register P0EXTIF

0xB4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0EXTIF	P07IF	P06IF	P05IF	P04IF	P03IF	P02IF	P01IF	P00IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 P0iIF: P0i interrupt flag bit (i=0-7);

1= P0i port produces an interrupt, which requires software clearance;

0= There is no interrupt in the P0i port.

6.4.3.9 Port P1 Interrupt Flag Register P1EXTIF

0xB5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P1EXTIF	P17IF	P16IF	P15IF	P14IF	P13IF	P12IF	P11IF	P10IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 P1iIF: P1i interrupt flag bit (i=0-7);

1= P1i port produces an interrupt, which requires software clearance;

0= There is no interrupt in the P1i port.

6.4.3.10 P2 Port Interrupt Flag Bit Register P2EXTIF

0xB6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2EXTIF	--	--	P25IF	P24IF	P23IF	P22IF	P21IF	P20IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit6 -- Reserved, must be 0.

Bit5~Bit0 P2iIF: P2i interrupt flag bit (i=0-7);

1= P2i port produces an interrupt, which requires software clearance;

0= There is no interrupt in the P2i port.

6.4.3.11 P3 Port Interrupt Flag Bit Register P3EXTIF

0xB7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P3EXTIF	--	--	--	--	P33IF	P32IF	P31IF	P30IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit6 -- Reserved, must be 0.

Bit5~Bit0 P3iIF: P3i interrupt flag bit (i=0-3);

1= P3i port produces an interrupt, which requires software clearance;

0= The P3i port did not produce an interrupt.

6.4.4 The clear operation for the interrupt flag bit

The clear operation of the interrupt flag is divided into the following categories:

- Automatic hardware cleanup (requires entry into interrupt service)
- Software cleanup
- Read/write operations are cleared

1) The hardware automatically clears the flag bits

The bits that support hardware auto-clearing are the interrupt flag bits generated by IN0, INT1, T0, T1, T3, and T4. The condition for the hardware to automatically clear the flag is: open the global interrupt enable bit EA=1, and open the corresponding interrupt enable bit, after the interrupt is generated, the system enters the corresponding interrupt service program, and the flag bit is automatically cleared. If interrupt enables shutdown, these flags can also be cleared using software.

2) The flag bits that the software clears

There are flags in the system that can only be cleared with software. These flags are not automatically cleared after entering the interrupt service program and require the software to write 0 to clear. Otherwise, after exiting the interrupt service program, you will enter the interrupt service program again.

3) The flag bit cleared by the R/W operations

The flag bit in the system is not written to zero to the flag bit, but requires reading/writing other registers to clear the flag bit. For example, if the transmission completion flag bit SPISIF in the SPI interrupt flag register is set to 1, you need to read spsr first, and then read/write SPDR and then clear zero.

Software cleanup operations require attention: when multiple interrupt flags are in the same register, and the moments these flags are generated are not related to each other, it is not recommended to use read-modify-write operations. For example, the PWMUIF interrupt flag register, which contains the upward comparison interrupt of the PG0-PG5 channel, these interrupt flag bits are not related to each other. When PG0 produces an up-to-compare interrupt, the value of PWMUIF is 0x01, and a read-modify-write operation clears the bit after entering the interrupt service program

```
PWMUIF &= 0xFE;
```

This operation is implemented by reading the value of PWMUIF back to the CPU, then performing the operation, and finally sending it back to PWMUIF. If the interrupt flag bit PWMUIF[1] of PG1 is set to 1 after the CPU read, and PWMUIF[1] is 0 when read, the operation is performed and sent back to PWMUIF[1] is also 0, at which point the upward interrupt flag bit PWMUIF[1] that PG1 has generated will be cleared.

Clear the interrupt flag bits of the above type, it is recommended to write 0 directly, other unrelated flag bits write 1: PWMUIF = 0xFE. This operation has no practical effect on unrelated interrupt flag write 1.

6.4.5 Special Interrupt Flag Bits in Debug Mode

The flag bit in the system is not written to zero to the flag bit, but requires reading/writing other registers to clear the flag bit.

In debug mode, after breakpoint execution, step-through, or stop operation, the emulator reads out all register values from the system to the emulation software, and the emulator reads/writes exactly the same as in normal mode.

So during debugging, after a pause occurs, the break flag bit of a set of 1 should appear, but it is displayed as 0 in the observation window.

Example: The transmit completion flag bit SPISIF in the SPI interrupt flag register in debug mode

```
...// Set the port and interrupt enable
SPDR = 0x56; Send SPDR data
delay();
...

void SPI_int (void) interrupt SPI_VECTOR// SPI interrupt service program
{
        O1_nop_(); Set breakpoint 1
            _nop_();
        O2k = SPSR; Set breakpoint 2
         _nop_();
        ...
}
```

When the breakpoint is running, stop after breakpoint 1, SPI completes the transmit operation, and the transmit completion interrupt has been generated, so SPSR.7=1, at which point the emulator has completed the operation of reading all the registers at once (including reading SPSR),

Perform the breakpoint run again, stop after breakpoint 2, at which point the emulator again finishes reading all registers (including SPDR), so SPSR.7=0. The above situation can also occur when you step into it twice, which requires attention in debug mode.

7. I/O Port

7.1 GPIO Function

The chip has four sets of I/O ports: PORT0, PORT1, PORT2, PORT3.

PORTx is a bidirectional port. Its corresponding data direction register is PxTRIS. A bit set to 1 (=1) of the PxTRIS allows the corresponding pin to be configured as an output. Zeroing one bit (= 0) of the PxTRIS configures the corresponding PORTx pin as the input.

When PORTx is used as an output port, the write Px register will write to the port latch, and all write operations are read-modify-write operations. Therefore, writing a port means reading the pin level of that port, then modifying the value read, and finally writing the changed value to the port data latch.

When PORTx is used as an output port, the Px register is read, which is related to the setting of the PxDS register. One position of PxDS 1 (=1), the corresponding bit of Px read is the state of the pin, one bit of PxDS is cleared to zero out (=0), and the corresponding bit of Px read is the state of the port data latch; When PORTx is used as an input port, the Px register reads the state of the pin, regardless of the setting of the PxDS register.

When using the PORTx pin as an analog input, the user must ensure that the bits in the PxTRIS register remain in the set 0 state. I/O pins configured as analog inputs are always read as 0.

Registers related to the PORTx port are Px, PxTRIS, PxOD, PxUP, PxRD, PxDS, etc.

7.1.1 PORTx Data Register Px

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Px	Px7	Px6	Px5	Px4	Px3	Px2	Px1	Px0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	X	X	X	X	X	X	X	X

Register P0 Address: 0x80; Register P1 Address: 0x90; Register P2 Address: 0xA0; Register P3 address: 0xB0.

Bit7~Bit0 Px<7:0>: Px I/O pin bits;

1= Port pin level >V_{IH} (forward threshold voltage);

0= The port pin level < V_{IL} (negative threshold voltage).

7.1.2 PORTx Direction Register PxTRIS

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxTRIS	PxTRIS7	PxTRIS6	PxTRIS5	PxTRIS4	PxTRIS3	PxTRIS2	PxTRIS1	PxTRIS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Register P0TRIS Address: 0x9A; Register P1TRIS Address: 0xA1; Register P2TRIS Address: 0xA2; Register P3TRIS address: 0xA3.

Bit7~Bit0 PxTRIS<7:0>: Three-state control bit;

1= The pins are configured as outputs;

0= The pins are configured as inputs (tri-state).

Note:

1. When a port is set to an output port, the data that reads the port is the value of the output register.
2. After the port is set to the input port, the < read-modify-write instructions of the > type to the port are actually operations on the output registers.

7.1.3 PORTx Open-drain Control Register PxOD

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxOD	PxOD7	PxOD6	PxOD5	PxOD4	PxOD3	PxOD2	PxOD1	PxOD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Register P0OD Address: F009H; Register P1OD address: F019H; Register P2OD address: F029H; Register P3OD address: F039H.

Bit7~Bit0 PxOD<7:0>: Open-drain control bit;

1= The pins are configured for an open-drain state (the output is an open-drain output);

0= The pins are configured for a normal state (the output is a push-pull output).

7.1.4 PORTx Pull-up Resistor Control Register PxUP

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxUP	PxUP7	PxUP6	PxUP5	PxUP4	PxUP3	PxUP2	PxUP1	PxUP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Register P0UP Address: F00AH; Register P1UP Address: F01AH; Register P2UP Address: F02AH; Register P3UP Address: F03AH.

Bit7~Bit0 PxUP<7:0>: Pull-up resistor control bit;

1= Pin pull-up resistor on;

0= The pin pull-up resistor is off.

7.1.5 PORTx pull-down resistor control register PxRD

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxRD	PxRD7	PxRD6	PxRD5	PxRD4	PxRD3	PxRD2	PxRD1	PxRD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Register P0RD address: F00BH; Register P1RD address: F01BH; Register P2RD address: F02BH; Register P3RD address: F03BH.

Bit7~Bit0 PxRD<7:0>: Pull-down resistor control bit;

1= Pin pull-down resistor open;

0= Pin pull-down resistor off.

Note: The control of the pull-down resistor is independent of the configuration and multiplexing function of the GPIO and is controlled separately by the PxRD registers.

7.1.6 PORTx Slope Control Register PxSR

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxSR	PxSR7	PxSR6	PxSR5	PxSR4	PxSR3	PxSR2	PxSR1	PxSR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Register P0SR Address: F00DH; Register P1SR Address: F01DH; Register P2SR Address: F02DH; Register P3SR address: F03DH.

Bit7~Bit0 PxSR<7:0>: Px slope control register (in effect when the port is configured as an output state);
1= The Px pin is slow slope;
0= The Px pin is fast slope.

7.1.7 The PORTx Data Input Selects Register PxDS

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxDS	PxDS7	PxDS6	PxDS5	PxDS4	PxDS3	PxDS2	PxDS1	PxDS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Register P0DS Address: F00EH; Register P1DS Address: F01EH; Register P2DS Address: F02EH; Register P3DS address: F03EH.

Bit7~Bit0 PxDS<7:0>: Data input selection bits that affect the value of the reading Px register when configured for GPIO;
1= Both output/input modes read pin status;
(The Smit circuit also remains open when the port is set to output);
0= Output mode: read to data latch state;
Input Mode: Reads to pin state.

Note: If you need to read the pin state of the port when it is a multiplexing function input structure, you need to set the port direction control to input mode.

7.2 Multiplexed Functions

7.2.1 Port multiplexing feature table

Pins are shared in a variety of functions, and each I/O port can be flexibly configured with digital functions or specified analog functions. The digital function of the external input is selected by the port input function allocation register (PS_XX); The multiplexing function is selected by the port multiplexing function configuration register (PxnCFG), where the communication input function is also specified by the communication input function allocation register (PS_XX).

The digital function configuration is shown in the following table:

	External input	Digital function configuration							
		0	1	2	3	4	5	6	7
P00	T0G/CAP0	GPIO	ANA	MOSI	-	PG0	CC0	-	-
P01	CAP1	GPIO	ANA	-	-	PG1	CC1	-	-
P02	-	GPIO	ANA	MOSI	-	PG2	C1_O	-	-
P03	-	GPIO	ANA	MISO	SDA	PG3	BUZZ	-	-
P04	-	GPIO	ANA	SCK	SCL	-	-	-	-
P05	CUSTOM	GPIO	ANA	NSS（SSO0）	-	-	-	-	-
P06	-	GPIO	ANA	MOSI	-	FB0	-	-	-
P07	-	GPIO	ANA	-	-	FB1	-	-	-
P10	T0G	GPIO	ANA	MISO	SDA	-	C0_The	-	-
P11	-	GPIO	ANA	SCK	SCL	FB0	-	-	-
P12	T2EX	GPIO	ANA	NSS（SSO1）	-	FB1	-	-	-
P13	CUSTOM	GPIO	ANA	-	RXD	-	CLO	-	-
P14	CAPE 3	GPIO	ANA	-	TXD	PG3	CC3	-	-
P15	T1G/CAP2	GPIO	ANA	-	-	PG2	CC2	-	-
P16	T2	GPIO	ANA	-	-	PG1	-	-	-
P17	T2EX	GPIO	ANA	-	-	PG0	BUZZ	-	-
P20	-	GPIO	ANA	-	-	PG0	C1_The	-	-
P21	T1G	GPIO	ANA	-	RXD	PG1	-	-	-
P22	T0G	GPIO	ANA	-	TXD	PG2	-	-	-
P23	-	GPIO	ANA	-	-	PG3	BUZZ	-	-
P24	T1/NRST	GPIO	ANA	-	-	-	C1_O	-	-
P25	T0/NRST	GPIO	ANA	-	-	-	-	-	-
P30	INT0	GPIO	ANA	-	-	-	-	-	-
P31	INT1	GPIO	ANA	-	-	-	-	-	-
P32	-	GPIO	ANA	-	-	-	-	-	-
P33	-	GPIO	ANA	-	-	-	-	-	-

The analog module, CONFIG configuration ports are shown in the following table:

	ANA(1)				CONFIG
	ADC	LCDSEG	LCDCOM	ACMP	
P00	AN0	SEG0	COM0	C1P2	-
P01	AN1	-	-	-	-
P02	AN2	SEG21	COM21	-	-
P03	AN3	SEG22	COM22	-	-
P04	AN4	SEG1	COM1	-	-
P05	AN5	SEG23	COM23	-	-
P06	AN6	SEG2	COM2	C1P3	-
P07	AN7	-	-	-	-
P10	AN8	SEG3	COM3	-	-
P11	AN9	SEG4	COM4	C0P0	-
P12	AN10	SEG5	COM5	C0N	-
P13	AN11	SEG6	COM6	-	-
P14	AN12	SEG7	COM7	-	-
P15	AN13	SEG8	COM8	-	-
P16	AN14	SEG9	COM9	-	-
P17	AN15	SEG10	COM10	-	-
P20	AN16	SEG11	COM11	-	-
P21	AN17	SEG12	COM12	C1P0	-
P22	AN18	SEG13	COM13	C1N	-
P23	AN19	SEG14	COM14	C1P1	-
P24	AN20	SEG15	COM15	-	-
P25	AN21	SEG16	COM16	-	-
P30	AN22	SEG17	COM17	-	XTOUT
P31	AN23	SEG18	COM18	-	XTIN
P32	AN24	SEG19	COM19	-	OSCIN
P33	AN25	SEG20	COM20	-	OSCOUT

7.2.2 Port Multiplexing Feature Configuration Register

The PORTx function configuration register PxnCFG

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxnCFG	--	--	--	--	--	PxnCFG2	PxnCFG1	PxnCFG0
R/W	--	--	--	--	--	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	1

Bit7~Bit3 -- Reserved, must be 0.

Bit2~Bit0 PxnCFG<2:0>: Feature configuration bit, which defaults to GPIO function. For details, see port function configuration instructions;

000= GPIO function;

001= Analog Function (ANA);

Other = Multiplexing function;

PxnCFG x=0/1/2/5,n=0-7

Each port has a function configuration register, PxnCFG, through which each port can be set to the corresponding digital function. For example: to set P17 as the BEEP buzzer function, configure it as: P17CFG = 0x05;

When the port is multiplexed, there is no need to configure the port direction register PxTRIS.

- SCL and SDA pull-up resistance registers can be configured to force open drain output.

- RXD0 and RXD1 synchronization modes force pull-up.

Other multiplexing functions are hardware forced to close the pull-up resistor, turn off the open-drain output, that is, the pull-up resistor PxUP or the open-drain output PxOD is invalid through the software.

When the port is multiplexed to SCL and SDA functions, the hardware forces the port to be an open-drain output, and the pull-up resistor PxUP can be set by software.

7.2.3 The Port Input Function Allocation Registers

Inside the chip there are digital functions with only the input state, such as INT0/INT1... etc., this type of digital input function is independent of the port multiplexing state. As long as the assigned port supports digital input (such as RXD0 as a digital input and GPIO as an input function), the port supports this function.

The input function port assignment registers are as follows:

register	address	function	Feature description
PS_INT0	F0C0H	INT0	External interrupt 0 input port allocation register
PS_INT1	F0C1H	INT1	External interrupt 1 input port allocation register
PS_T0	F0C2H	T0	Timer0 external clock input port assignment register
PS_T0G	F0C3H	T0G	Timer0 gated input port assignment register
PS_T1	F0C4H	T1	Timer1 external clock input port assignment register
PS_T1G	F0C5H	T1G	Timer1 gated input port assignment register
PS_T2	F0C6H	T2	Timer2 external event or gated input port assignment register
PS_T2EX	F0C7H	T2EX	Timer2 drops along the autoreload input port allocation register
PS_CAP0	F0C8H	CAP0	The Timer2 input captures channel 0 port assignment registers
PS_CAP1	F0C9H	CAP1	The Timer2 input captures the channel 1 port assignment register

PS_CAP2	F0CAH	CAP2	The Timer2 input captures the channel 2 port assignment register
PS_CAP3	F0CBH	CAP3	The Timer2 input captures the channel 3 port assignment register
PS_ADET	F0CCH	CUSTOM	The ADC's external trigger input port allocates registers
PS_FB0	F0CDH	FB0	PWM external brake signal FB0
PS_FB1	F0CAndH	FB1	PWM external brake signal FB1

PS_XX input function port allocation register PS_XX (as described in the table above)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS_XX	--	PS_XX6	PS_XX5	PS_XX4	PS_XX3	PS_XX2	PS_XX1	PS_XX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	1	1	1	1	1	1	1	1

Bit7 -- Reserved, must be 0.

Bit6~Bit0 PS_XX<6:0>: The input function assigns control bits

(Subject to the actual port of the chip, the unused value is retained and prohibited);

0x00= Assigned to P00;

0x01= Assigned to port P01;

…… ……

0x20= Assigned to P20 ports;

0x21= Assigned to P21;

…… ……

…… ……

0xFF= Not assigned to a port;

1) If multiple ports are configured at the same time for the same numeric function, the priority is P00, P01,..... The order of them is decremented. If both P05 and P13 are configured as ADET functions, the P05 configuration is valid and the P13 configuration is invalid.

2) The input function assignment structure allows multiple input functions to be assigned to the same port. For example, T1G and CAP2 can be assigned to P15 port at the same time, and the configuration is as follows:

P15CFG = 0x00; //P15 port configured for GPIO capability

P1TRIS = 0x00; //P15 for GPIO input function

PS_T1G = 0x15; The P15 port is configured for T1G function

PS_CAP2 = 0x15; //P15 port configured for CAP2 function

3) This input function configuration structure can also be used in conjunction with the interrupt function outside the port. If you can assign the T1G and GPIO interrupt functions to the P15 port at the same time, configure it as follows:

P15CFG = 0x00; //P15 port configured for GPIO capability

P1TRIS = 0x00; P15 for GPIO input function

PS_T1G = 0x15; //P15 ports are configured for T1G capability

P15EICFG = 0x01; //P15 ports are configured to trigger interrupts on the rising edge

P1EXTIE = 0x20;/Allows port external interrupts for P15

7.2.4 Communication Input Function Allocation Registers

When the port is used as a communication port (UART0/SPI/IIC), multiple input ports are selectable, and different port inputs can be selected by setting the following registers. The communication input function port assignment registers are as follows:

register	address	function	Feature description
PS_SCLK	F698H	SCLK	SPI clock port assignment registers
PS_MOSI	F699H	MOSI	SPI slave input port assignment registers
PS_MISO	F69AH	MISO	SPI host input port assignment register
PS_NSS	F69BH	NSS	SPI allocates registers from the on-chip select input port
PS_SCL	F69CH	SCL	IIC clock input port assignment register
PS_SDA	F69DH	SDA	IIC data input port assignment registers
PS_RXD0	F69FH	RXD0	UART0 data input port assignment register

Communication input function port assignment register PS_XX (as described in the table above)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS_XX	--	PS_XX6	PS_XX5	PS_XX4	PS_XX3	PS_XX2	PS_XX1	PS_XX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	1	1	1	1	1	1	1	1

Bit7	--	Reserved, must be 0.
Bit6~Bit0	PS_XX<6:0>:	Input function allocation control bit (subject to the actual port of the chip, see the multiplexing function allocation table;
	0x00=	Assigned to P00;
	0x01=	Assigned to port P01;
	……	……
	0x20=	Assigned to P20 ports;
	0x21=	Assigned to P21;
	……	……
	0xFF=	Not assigned to a port.

If P13 is configured as RXD0, configure it as follows:

PS_RXD0 =0x13; Select the P13 as the RXD0 pin

P13CFG =0x03; P13 is multiplexed to the RXD0 function of UART0

7.2.5 Port external interrupt control registers

When using an external interrupt, the port needs to be configured as GPIO function and the direction is set to the input port. Alternatively, the multiplexing function is the input port (such as RXD0), each port can be configured as a GPIO interrupt function.

PORTx external interrupt control register Px_NEICFG

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Px_NEICFG	--	--	--	--	--	--	Px1EICFG1	Px0EICFG0
R/W	--	--	--	--	--	--	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit2 -- Reserved, all must be 0.

Bit1~Bit0 Px_NEICFG<1:0>: Px_N external interrupt control bit;

00= No external interrupts;

01= Rising edge triggers interrupts;

10= Falling edge triggers interrupt;

11= Both the rising or falling edge trigger an interrupt.

Px has 8 external interrupt control registers, including Px0EICFG~Px7EICFG, which control the external interrupts of Px0~Px7, x=0,1,2,5.

If you configure the P00 falling edge to trigger an interrupt, configure it as follows:

P00CFG =0x00; Configure P00 for GPIO functionality

P0TRIS =0x00; Configure P00 as input

P00EICFG = 0x02; Configure P00 to trigger interrupts on the falling edge

EA = 1; Global interrupt enablement

P0EXTIE = 0x01; Enables P00 external interrupt function

7.2.6 Multiplexing Features Application Notes

1) The multiplexing function configuration register is configured as an analog function by default (0x01), and if the digital function is used, the value of the register needs to be set to 0x00.
2) The input of the multiplexing function is relatively independent of the structure of the port's external interrupt (GPIO interrupt) and port input function.

   For example, the P21 port is configured as RXD0, and the GPIO interrupt trigger mode of the P21 is to trigger the rising edge and interrupt enable, and when the P21 input changes from low to high, the GPIO interrupt of P21 will be triggered.
3) The input structure of the digital signal is not affected by the configuration state of the system.
4) It should be noted that in debug mode, such as the multiplexing function is configured to the DSDA port, its input function is also valid, it is recommended that in the debugging mode, do not configure the relevant reuse function to the DSDA port.
5) When the port is used as an analog function, when the function configuration register is set to 0x01, the hardware turns off the digital circuit to reduce power consumption, and the GPIO function-related register setting is invalid.
6) The port input/communication input function has priority restrictions, if two or more ports are configured with the same input function at the same time, it is P00, P01,..... Configure the selection in order of priority from high to low.
7) The output capability of a communication port has no priority restrictions, and if multiple ports are configured with the same output capability, the functionality outputs simultaneously on those ports.
8) The RXD0 of UART0 is selected by the port allocation register as an input function, and the port allocation register is independent of the synchronous output function. That is, when RXD0 is used as a synchronous output function, multiple pins can be selected as RXD0 outputs at the same time.
9) When the SPI's SCLK is used as the clock input of the slave, it needs to be selected by the port allocation register, and the clock output of the master is independent of the port allocation register, and it is recommended that the SCLK be configured as both an output and an input.
10) When the SCL of IIC is used as the clock input of the slave, it needs to be selected by the port allocation register. When it is used as the clock output of the master, it is related to the port allocation register. It is recommended that the SCL be configured with the port allocation register whether it is used as the output or input.

8. Watchdog Timer (WDT)

8.1 Overview

The Watch Dog Timer is an on-chip timer with configurable overflow time and clock source provided by the system clock Fsys.

When the watchdog timer counts to the configured overflow value, a watchdog overflow interrupt flag bit (WDTIF=1) is generated. If the global interrupt is enabled (EA=1) and the watchdog timer is disabled (EIE2[5]=1), the CPU executes the interrupt service program, clearing the watchdog counter through the write register WDCON[0]=1. After the watchdog counter clears, the counter starts counting from 0 again until the next timer overflows.

When the watchdog timer overflows, if the watchdog overflow reset enable (WDCON[1]=1) and the watchdog counter is not cleared, an overflow reset of the watchdog is generated. Watchdog overflow reset is a protective setting of the system, when the system is running to an unknown state, the watchdog can be used to reset the system, thereby avoiding the system from entering an indefinite dead loop. Watchdog overflow reset is detailed in the Reset section.

8.2 Related Registers

8.2.1 Watchdog Control Register WDCON

0x97	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WDCON	SWRST	PORF	--	--	WDTIF	WDTRF	WDTRE	WDTCLR
R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W
Reset value	0	1	0	0	0	0	0	0

Bit7	SWRST:	Software reset control bit;
	1:	Perform a system software reset (write 0 clear after reset).
	0:	--
Bit6	PORF:	Power-on reset marker;
	1:	The system resets the system at power-on (write 0 clear, do not need TA to write timing).
	0:	--
Bit5~Bit4	--	Reserved, must be 0.
Bit3	WDTIF:	WDT overflow interrupt flag bit;
	1=	WDT overflow (write 0 cleared);
	0=	WDT does not overflow.
Bit2	WDTRF:	WDT reset marker bit;
	1=	The system is reset by WDT (write 0 cleared);
	0=	The system is not reset by WDT.
Bit1	WDTRE:	WDT reset enable bit;
	1=	Enable WDT reset CPU;
	0=	Disable WDT reset cpu.
Bit0	WDTCLR:	WDT counter clear bit;
	1=	Clear WDT counter (hardware auto-zero);
	0=	Disables the WDT counter (write 0 is invalid).

Note:

1. If the WDT in CONFIG is configured as: ENABLE, the WDT is always enabled, regardless of the state of the WDTRE control bit. And the overflow reset function of WDT is forced on.
2. If WDT in CONFIG is configured as: SOFTWARE CONTROL, WDTRE can be enabled or disabled using the WDTRE control bit.

Modify the sequence of instructions required by WDCON (no other instructions can be inserted in the middle):

```
MOV     TA,#0AAH
MOV     TA,#055H
ORL     WDCON,#01H
```

8.2.2 Watchdog overflow control register CKCON

0x8E	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CKCON	WTS2	WTS1	WTS0	T1M	T0M	--	--	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	1	1	1

Bit7~Bit5 WTS<2:0>: WDT overflow time selection bits;
000= 2^{17}*Tsys;
001= 2^{18}*Tsys;
010= 2^{19}*Tsys;
011= 2^{20}*Tsys;
100= 2^{21}*Tsys
101= 2^{22}*Tsys;
110= 2^{24}*Tsys;
111= 2^{26}*Tsys。

Bit4 T1M: Timer1's clock source select bit;
0= Fsys/12;
1= Fsys/4。

Bit3 T0M: Clock source select bit of Timer0;
0= Fsys/12;
1= Fsys/4。

Bit2~Bit0 -- Reserved, must be 1.

8.3 WDT Interrupt

The watchdog timer can enable or disable interrupts via the EIE2 register, and the high/low priority is set through the EIP2 register, where the relevant bits are as follows.

8.3.1 Interrupt Mask Register EIE2

0xAA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIE2	SPIIE	I2CIE	WDTIE	ADCIE	PWMIE	--	ET4	ET3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 SPIIE: SPI interrupt enable bit;
1= Enable SPI interrupts;
0= Disable SPI Interrupt.
Bit6 I2CIE: I2C Interrupt enable bit;
1= Allow I2C Interrupts;
0= Disable I2C Interrupt.
Bit5 WDTIE: WDT interrupt enable bit;
1= Enable WDT overflow interrupts;
0= Disable WDT overflow interrupts.
Bit4 ADCIE: ADC interrupt enable bit;
1= Enable ADC interrupts;
0= Disable ADC interrupts.
Bit3 PWMIE: PWM global interrupt enable bit;
1= Enable all PWM interrupts;
0= Disable all PWM interrupts.
Bit2 --: Reserved, must be 0.
Bit1 ET4: Timer4 interrupt enable bit;
1= Enable Timer4 interrupts;
0= Disable Timer4 Interrupt.
Bit0 ET3: Timer3 interrupt enable bit;
1= Enable Timer3 interrupts;
0= Disable Timer3 Interrupt.

8.3.2 Interrupt priority control register EIP2

0xBA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP2	PSPI	PI2C	PWDT	PADC	PPWM	--	PT4	PT3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 PSPI: SPI interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

Bit6 PI2C: I2C Interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

Bit5 PWDT: WDT interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

Bit4 PADC: ADC interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

Bit3 PPWM: PWM interrupt priority control bit
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

Bit2 --: Reserved, must be 0.

Bit1 PT4: TIMER4 interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

Bit0 PT3: TIMER3 interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

9. Timer Counter 0/1 (Timer0/1)

Timer 0 is similar in type and structure to Timer 1 and is two 16-bit timers. Timer 1 has three modes of operation and Timer 0 has four modes of operation. They provide basic timing and event counting operations.

In "timer mode", the timing register is incremented every 12 or 4 system cycles when the timer clock is enabled.

In "counter mode", the timing register increments whenever it detects a falling edge on the corresponding input pin (T0 or T1).

9.1 Overview

Timer 0 and Timer 1 are fully compatible with the standard 8051 timer.

Each timer consists of two 8-bit registers: {TH0(0x8C): TL0(0x8A)} and {TH1(0x8D):TL1(0x8B)}. Timers 0 and 1 operate in four identical modes. The Timer0 and Timer1 modes are described below.

Mode	M1	M0	Feature description
0	0	0	THx [7:0], TLx [4:0] form a 13-bit timer/counter
1	0	1	THx [7:0], TLx [7:0] form a 16-bit timer/counter
2	1	0	TLx [7:0] consists of an 8-bit auto-reload timer/counter that is reloaded from THx
3	1	1	TL0, TH0 are two 8-bit timers/counters, and Timer1 stops counting

The registers THx and TLx are special function registers with the function of storing the actual timer values. THx and TLx can be cascaded into 13-bit or 16-bit registers via mode options. Each time an internal clock pulse or an external timer pin is received, the value of the register is incremented by 1. The timer starts counting from the value contained in the preset register until the timer is full till overflow, at which point an internal interrupt signal is generated. If auto-overload mode is selected for the timer, the value of the timer is reset to the initial value of the preload register and continues counting, otherwise the value of the timer is reset to zero. Note that in order to get the maximum calculation range of the timing/counter, the preset registers must first be cleared to zero.

9.2 Related Registers

9.2.1 Timer0/1 mode register TMOD

0x89	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMOD	GATE1	CT1	T1M1	T1M0	GATE0	CT0	T0M1	T0M0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 GATE1: Timer 1 gate control bit;
1= Enable;
0= Disable.

Bit6 CT1: Timer 1 timing/count select bits;
1= Count;
0= Timing.

Bit5~Bit4 T1M<1:0>: Timer 1 mode select bit;
00= Mode 0, 13-bit timer/counter;
01= Mode 1, 16-bit timer/counter;
10= Mode 2, 8-bit automatic reload timer/counter;
11= Mode 3, Stop Count.

Bit3 GATE0: Timer 0 gate control bit;
1= Enable;
0= Disable.

Bit2 CT0: Timer 0 Timing/Count Select Bits;
1= Count;
0= Timing.

Bit1~ Bit0 T0M<1:0>: Timer 0 mode select bit;
00= Mode 0, 13-bit timer/counter;
01= Mode 1, 16-bit timer/counter;
10= Mode 2, 8-bit automatic reload timer/counter;
11= Mode 3, two independent 8-bit timers/counters.

9.2.2 Timer0/1 control register TCON

0x88	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 TF1: Timer1 counter overflow interrupt flag bit;
1= The Timer1 counter overflows and enters the interrupt service program hardware to automatically zero;
0= The Timer1 counter has no overflow.

Bit6 TR1: Timer1 operational control bit;
1= Timer1 starts;
0= Timer1 is off.

Bit5 TF0: Timer0 counter overflow interrupt flag bit;
1= The Timer0 counter overflows, entering the interrupt service program hardware to automatically zero;
0= The Timer0 counter has no overflow.

Bit4 TR0: Timer0 operational control bit;
1= Timer0 starts;
0= Timer0 closes.

Bit3 IE1: External interrupt 1 flag;
1= External interrupt 1 generates an interrupt, and the hardware of the interrupt service program is automatically cleared;
0= External interrupt 1 did not produce an interrupt.

Bit2 IT1: External interrupt 1 trigger mode control bit;
1= Falling edge trigger;
0= Low level triggering.

Bit1 IE0: External interrupt 0 flag;
1= External interrupt 0 generates an interrupt, and the hardware entering the interrupt service program is automatically cleared;
0= External interrupt 0 did not produce an interrupt.

Bit0 IT0: External interrupt 0 trigger mode control bit;
1= Falling edge trigger;
0= Low level triggering.

9.2.3 Timer0 data register low bit TL0

0x8A	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TL0	TL07	TL06	TL05	TL04	TL03	TL02	TL01	TL00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~ Bit0 TL0<7:0>: Timer 0 low data register (also as counter low).

9.2.4 Timer0 data register high bit TH0

0x8C	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TH0	TH07	TH06	TH05	TH04	TH03	TH02	TH01	TH00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 TH0<7:0>: Timer 0 high bit data register (also as counter high bit).

9.2.5 Timer1 data register low bit TL1

0x8B	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TL1	TL17	TL16	TL15	TL14	TL13	TL12	TL11	TL10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 TL1<7:0>: Timer 1 low bit data register (also as counter low bit).

9.2.6 Timer1 data register high TH1

0x8D	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TH1	TH17	TH16	TH15	TH14	TH13	TH12	TH11	TH10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 TH1<7:0>: Timer 1 high-bit data register (also as counter high-bit).

9.2.7 Function clock control register CKCON

0x8E	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CKCON	WTS2	WTS1	WTS0	T1M	T0M	--	--	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	1	1	1

Bit7~Bit5 WTS<2:0>: WDT overflow time selection bits;
000= 2^{17}*Tsys;
001= 2^{18}*Tsys;
010= 2^{19}*Tsys;
011= 2^{20}*Tsys;
100= 2^{21}*Tsys
101= 2^{22}*Tsys;
110= 2^{24}*Tsys;
111= 2^{26}*Tsys。

Bit4 T1M: Timer1's clock source select bit;
0= Fsys/12;
1= Fsys/4。

Bit3 T0M: Clock source select bit of Timer0;
0= Fsys/12;
1= Fsys/4。

Bit2~Bit0 -- Reserved, must be 1.

9.3 Timer0/1 Interrupt

Timer0/1 can enable or disable interrupts via the IE register, and can also set high/low priority via the IP register, where the relevant bits are described as following:

9.3.1 Interrupt Mask register IE

0xA8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IE	SHE	--	ET2	ES0	ET1	EX1	ET0	EX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 SHE: Global interrupt enable bits;
1= Enable all unblocked interrupts;
0= Disable all interrupts.
Bit6 -- Reserved, must be 0.
Bit5 ET2: TIMER2 Global interrupt enable bits;
1= Enable timer2 all interrupts;
0= Disable all TIMER2 interrupts.
Bit4 ES0: UART0 interrupt enable bit;
1= Enable UART0 interrupts;
0= Disable UART0 Interrupt.
Bit3 ET1: TIMER1 interrupt enable bit;
1= Enable TIMER1 interrupts;
0= Disable TIMER1 Interrupt.
Bit2 EX1: External interrupt 1 interrupt enable bits;
1= Enable external interrupt 1 interrupt;
0= Disable external interrupt 1 interrupt.
Bit1 ET0: TIMER0 interrupt enable bits;
1= Enable TIMER0 interrupts;
0= Disable TIMER0 Interrupts.
Bit0 EX0: External interrupt 0 interrupt enable bit;
1= Enable external interrupt 0 interrupts;
0= Disable external interrupt 0 interrupt.

9.3.2 Interrupt priority control register IP

0xB8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IP	--	PS1	PT2	PS0	PT1	PX1	PT0	PX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 -- Reserved, must be 0.

Bit6 -- Reserved, must be 0.

Bit5 PT2: TIMER2 interrupt priority control bit;

1= Set to High-level Interrupt;

0= Set to low-level interrupt.

Bit4 PS0: UART0 interrupt priority control bit;

1= Set to High-level Interrupt;

0= Set to low-level interrupt.

Bit3 PT1: TIMER1 interrupt priority control bit;

1= Set to High-level Interrupt;

0= Set to low-level interrupt.

Bit2 PX1: External interrupt 1 interrupt priority control bit;

1= Set to High-level Interrupt;

0= Set to low-level interrupt.

Bit1 PT0: TIMER0 interrupt priority control bit;

1= Set to High-level Interrupt;

0= Set to low-level interrupt.

Bit0 PX0: External interrupt 0 interrupt priority control bit;

1= Set to High-level Interrupt;

0= Set to low-level interrupt.

9.3.3 Timer0/1, INT0/1 interrupt flag bit register TCON

0x88	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 TF1: Timer1 counter overflow interrupt flag bit;
1= Timer1 counter overflow, when entering the interrupt service program, the hardware is automatically cleared, and the software can also be cleared;
0= The Timer1 counter has no overflow.

Bit6 TR1: Timer1 operational control bit;
1= Timer1 starts;
0= Timer1 is off.

Bit5 TF0: Timer0 counter overflow interrupt flag bit;
1= Timer0 counter overflow, when entering the interrupt service program, the hardware is automatically cleared, and the software can also be cleared;
0= The Timer0 counter has no overflow.

Bit4 TR0: Timer0 operational control bit;
1= Timer0 starts;
0= Timer0 closes.

Bit3 IE1: External interrupt 1 flag;
1= External interrupt 1 generates an interrupt, the hardware is automatically cleared when entering the interrupt service program, and the software can also be cleared;
0= External interrupt 1 did not produce an interrupt.

Bit2 IT1: External interrupt 1 trigger mode control bit;
1= Falling edge trigger;
0= Low level triggering.

Bit1 IE0: External interrupt 0 flag;
1= External interrupt 0 generates an interrupt, the hardware is automatically cleared when entering the interrupt service program, and the software can also be cleared;
0= External interrupt 0 did not produce an interrupt.

Bit0 IT0: External interrupt 0 trigger mode control bit;
1= Falling edge trigger;
0= Low level triggering.

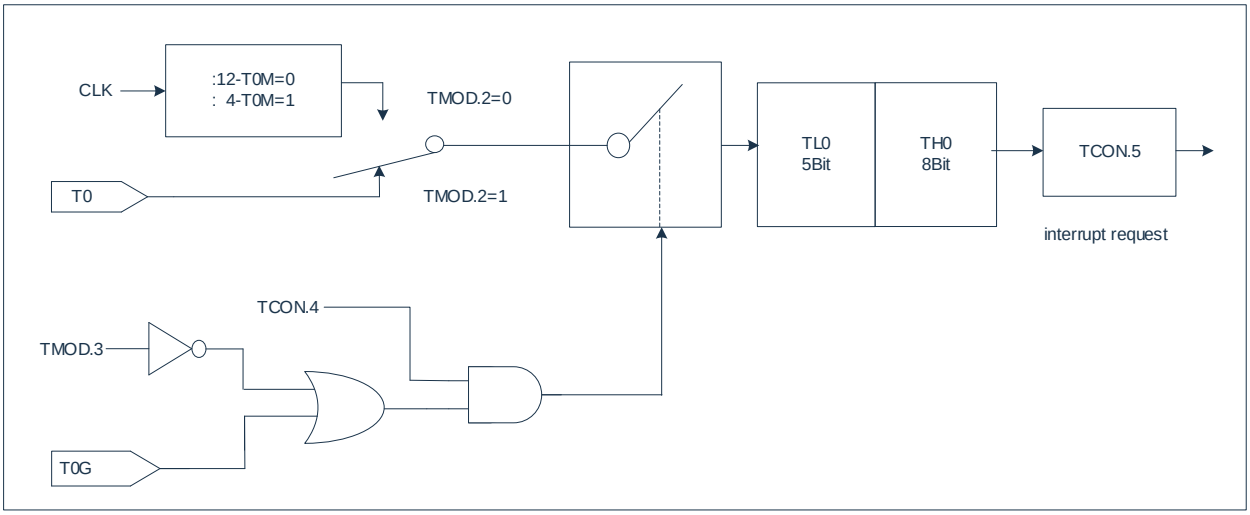
Flag bits that produce interrupts can be cleared by software, the same result as cleared by hardware. That is, an interrupt can be generated by software (it is not recommended to generate an interrupt by writing flag bits) or a pending interrupt can be canceled.

TF0, the TF1 flag bit can be cleared by writing 0 without enabling interrupt.

9.4 Timer0 Working Mode

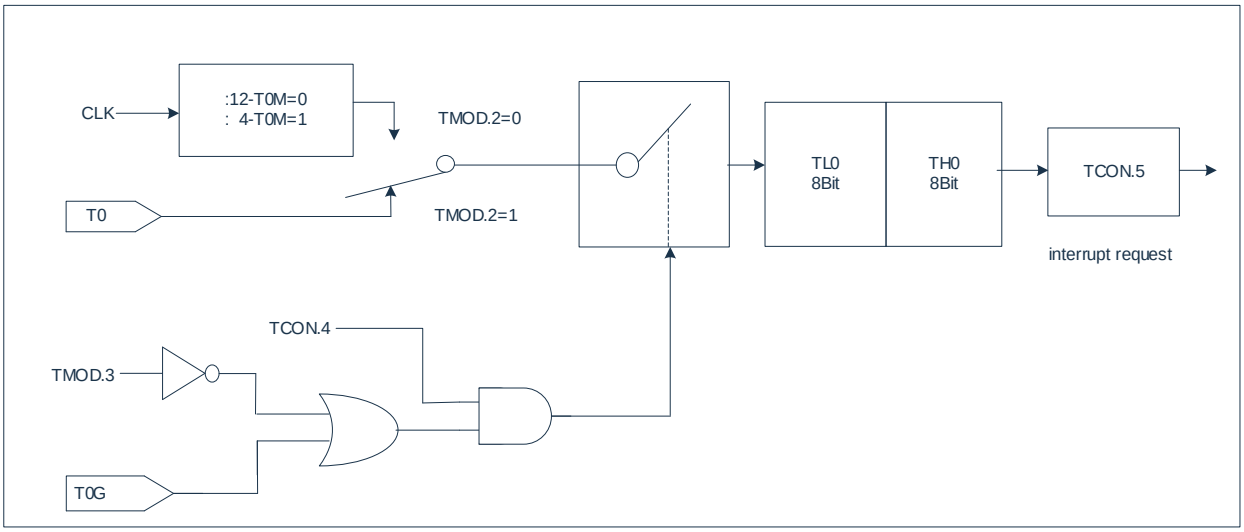
9.4.1 T0 - Mode 0 (13-bit Timing/Counting Mode)

In this mode, timer 0 is a 13-bit register. When all the bits of the counter are flipped from 1 to 0, the timer 0 interrupt flag TF0 is set to 1. When TCON.4=1 and TMOD.3=0 or TCON.4=1, TMOD.3=1, T0G=1, the count input is enabled to timer 0. (Setting TMOD.3=1 allows timer 0 to be controlled by an external pin, T0G, for pulse width measurements). The 13-bit register consists of TH0 and TL0 low 5 bits. TL0 high 3 bits should be ignored. Timer0 Mode 0 block diagram is shown in the following figure:



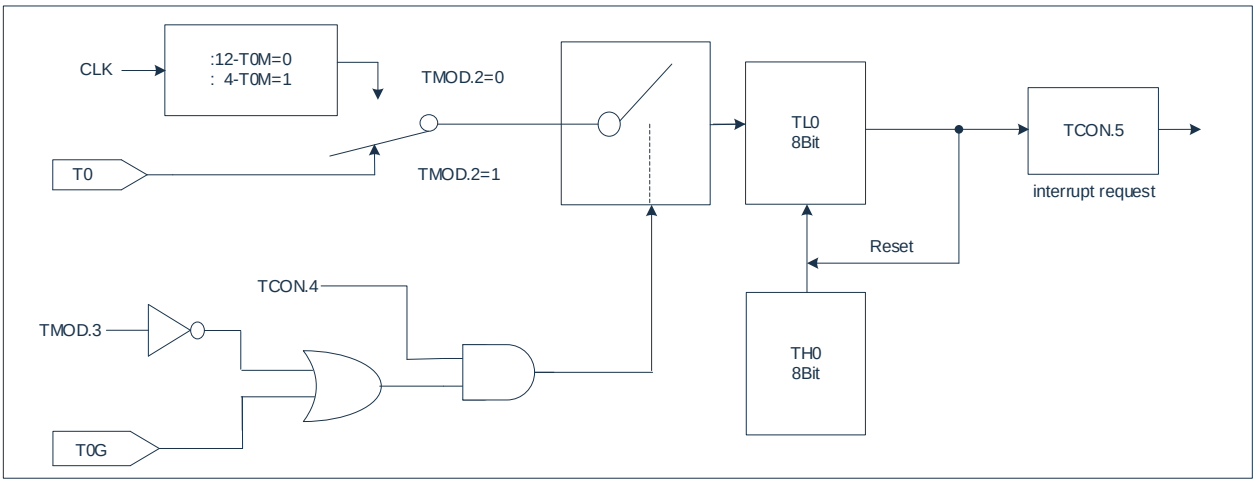
9.4.2 T0 - Mode 1 (16-bit Timing/Counting Mode)

Mode 1 is the same as mode 0, except that the timer 0 data register 16 bits are all running in mode 1. Timer0 Mode 1 block diagram is shown in the following figure:



9.4.3 T0 - Mode 2 (8-bit Auto-reload Timing/Counting Mode)

The mode 2 timer register is an 8-bit counter (TL0) with auto reload mode, as shown in the figure below. The overflow from TL0 not only sets TF0 to 1, but also reloads the contents of TH0 from software to TL0. The value of TH0 remains unchanged during Reloading. Timer0 Mode 2 block diagram is shown in the following figure:



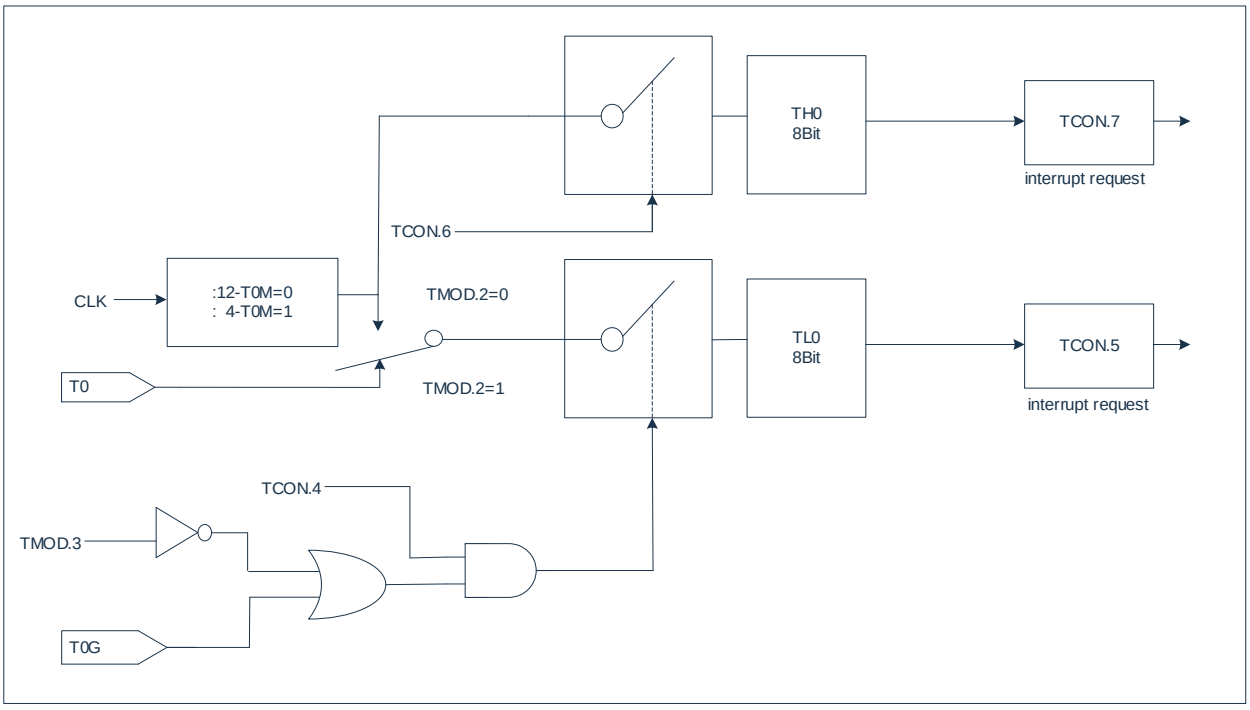
9.4.4 T0 - Mode 3 (Two Separate 8-bit Timers/Counters)

Timer 0 in mode 3 sets TL0 and TH0 to two independent counters. The logic of timer 0 mode 3 is shown in the following figure.

TL0 can operate as a timer or counter and use the control bits of timer 0: such as CT0, TR0, GATE0, and TF0.

TH0 can only operate as a timer, and uses the TR1 and TF1 flags of timer 1 and controls the interrupt of timer 1.

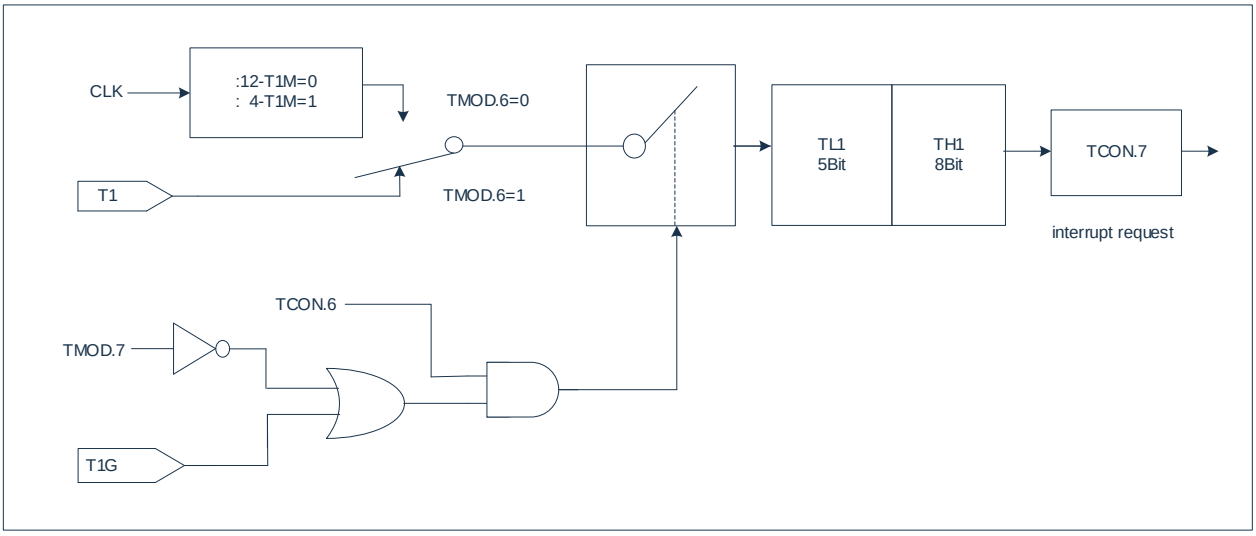
Mode 3 can be used when two 8-bit timers/counters are required. When timer 0 is in mode 3, timer 1 can turn it off by switching to its own mode 3, or it can still be used as a baud rate generator by the serial channel, or in any application that does not require an interrupt from timer 1. Timer0 Mode 3 block diagram is shown in the following figure:



9.5 Timer1 Working Mode

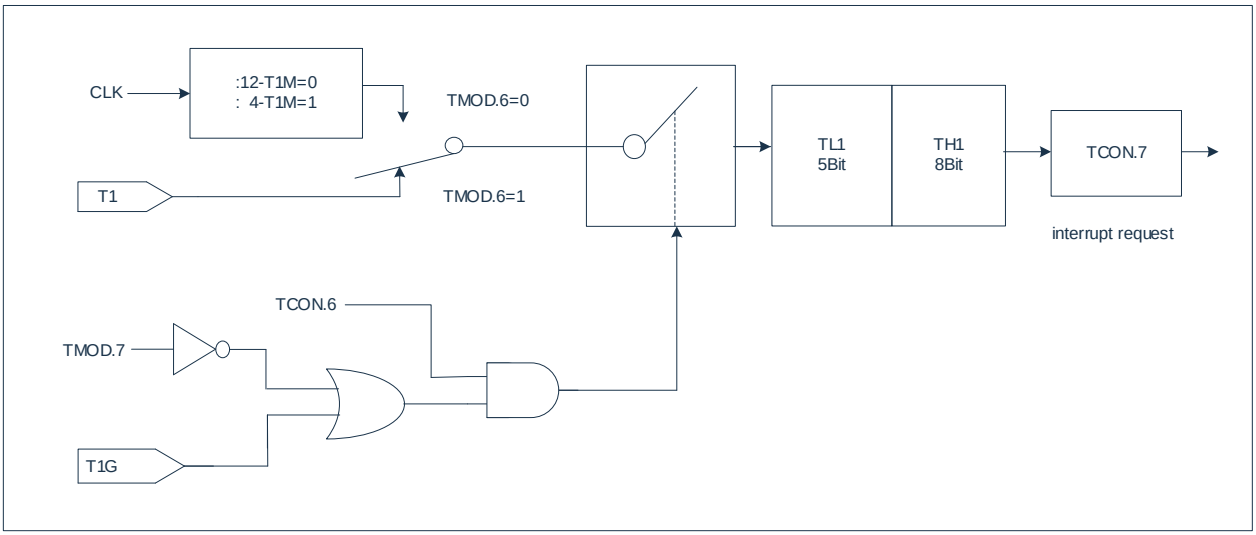
9.5.1 T1 - Mode 0 (13-bit Timing/Counting Mode)

In this mode, timer 1 is a 13-bit register. When all the bits of the counter are flipped from 1 to 0, the timer 1 interrupt flag TF1 is set to 1. When TCON.6=1 and TMOD.7=0 or when TCON.6=1, TMOD.7=1, and T1G=1, the count input is enabled to timer 1. (Setting TMOD.7=1 allows timer 1 to be controlled by an external pin, T1G, for pulse width measurements). The 13-bit register consists of TH1 8 bits and TL1 low 5 bits. TL1 high three bits should be ignored. Timer1 Mode 0 block diagram is shown in the following figure:



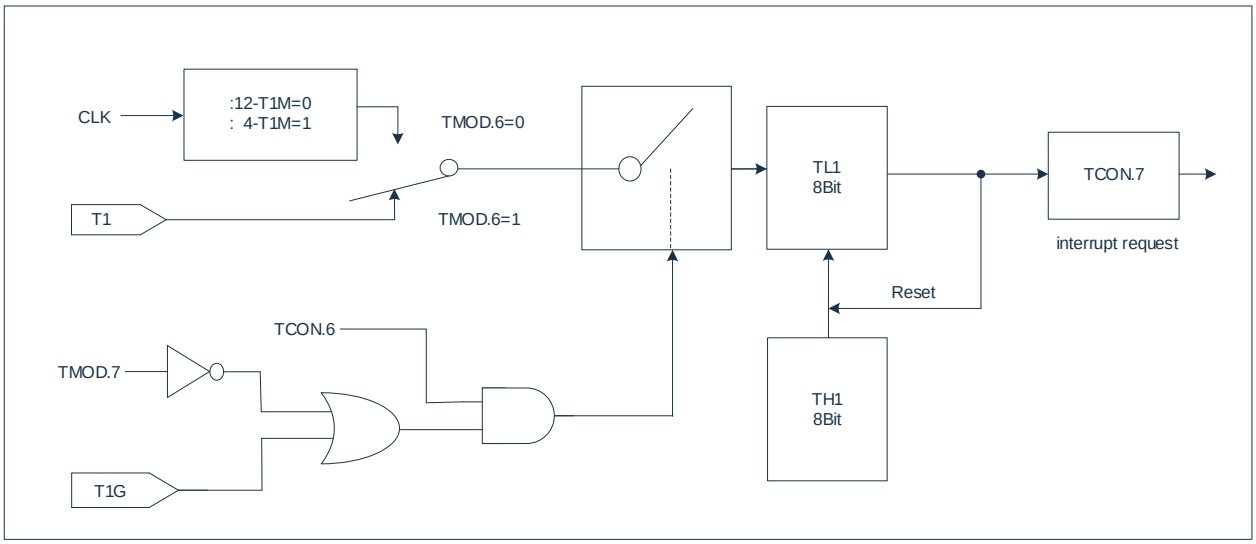
9.5.2 T1 - Mode 1 (16-bit Timing/Counting Mode)

Mode 1 is the same as mode 0, except that the timer 1 registers 16 bits are all running in mode 1. Timer1 Mode 1 block diagram is shown in the following figure:



9.5.3 T1 - Mode 2 (8-bit Auto Reload Timing/Counting Mode)

The timer 1 register in mode 2 is an 8-bit counter (TL1) with auto-reload mode, as shown in the figure below. The overflow from TL1 not only makes TF1 1, but also reloads the contents of TH1 from software to TL1. The value of TH1 remains unchanged during Reloading. Timer1 Mode 2 block diagram is shown in the following figure:



9.5.4 T1 - Mode 3 (Stop Count)

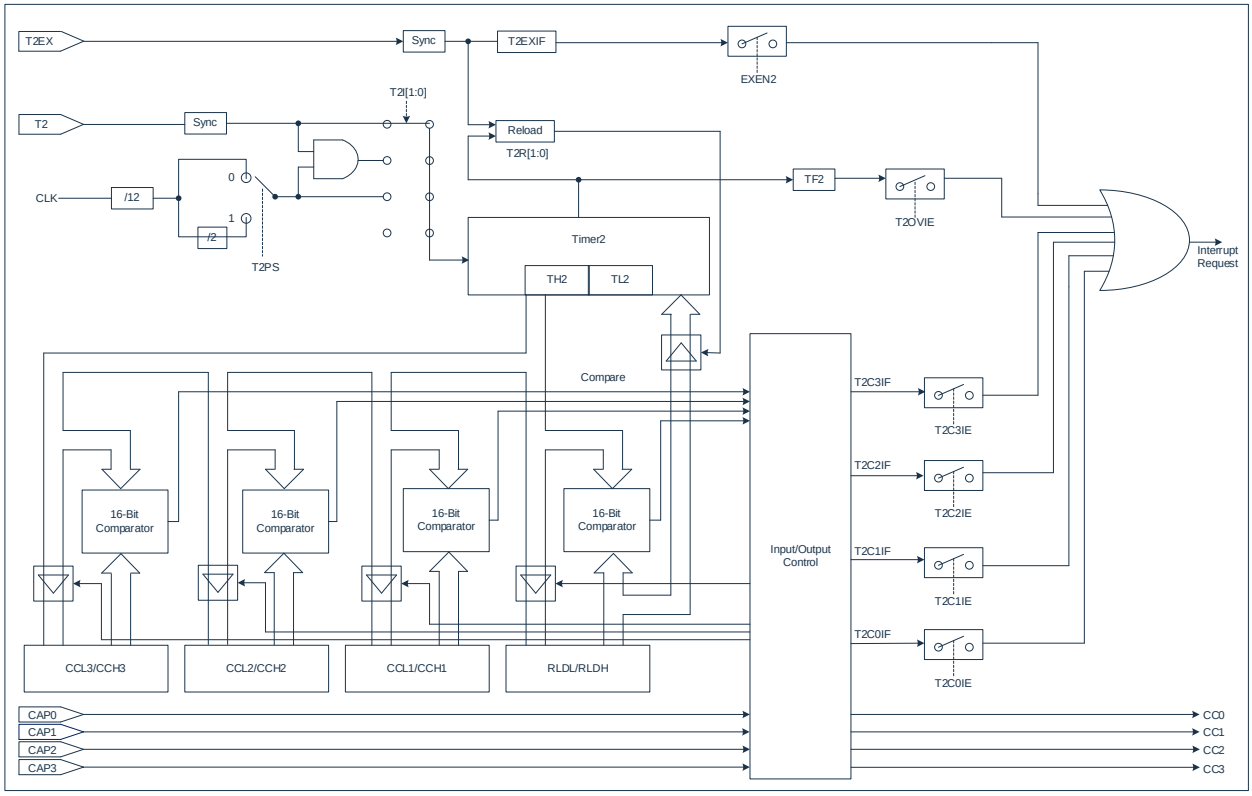
Timer 1 in mode 3 stops counting with the same effect as setting TR1=0.

10. Timer Counter 2 (Timer2)

Timer 2 with additional compare/capture/reload functionality is one of the core peripheral units. It can be used for the generation of various digital signals and event capture, such as pulse generation, pulse width modulation, pulse width measurement, etc.

10.1 Overview

Block diagram of timer 2 with additional compare/capture/reload register function as shown in the following figure:



10.2 Related Registers

10.2.1 Timer2 Control Register T2CON

0xC8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2CON	T2PS	I3FR	CAPES	T2R1	T2R0	T2CM	T2I1	T2I0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 T2PS: Timer2 clock prescaler selection bit;
1= Fsys/24;
0= Fsys/12。

Bit6 I3FR: Capture channel 0 input one-edge selection with comparison interrupt moment selection bit;
Capture channel 0 mode:
1= Rising edge capture to RLDL/RLDH registers;
0= The falling edge is captured to the RLDL/RLDH register.
Compare channel 0 modes:
1= TL2/TH2 and RLDL/RLDH never wait until the same moment to produce an interrupt;
0= TL2/TH2 and RLDL/RLDH are interrupted from the moment of equality to inequality;

Bit5 CAPES: Capture channels 1-3 input one-edge edge selection (in effect for capture channels 1-3).
0= The rising edge is captured to the CCL1/CCH1-CCL3/CCH3 registers;
1= The falling edge is captured to the CCL1/CCH1-CCL3/CCH3 registers.

Bit4~Bit3 T2R<1:0>: Timer2 Load mode select bit;
0x= Reloading is disabled;
10= Loading mode 1: Automatically Reload when Timer2 overflows;
11= Loading mode 2: Reload on the descending edge of the T2EX pin.

Bit2 T2CM: Comparison mode selection;
1= Comparison mode 1;
0= Compare mode 0.

Bit1~Bit0 T2I<1:0>: Timer2 clock input select bit;
00= Timer2 stops;
01= Division of the system clock (selected by T2PS control crossover);
10= External pin T2 for event input (event count mode);
11= The external pin T2 is the gated input (gated timing mode).

10.2.2 Timer2 Data Register Low Bit TL2

0xCC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TL2	TL27	TL26	TL25	TL24	TL23	TL22	TL21	TL20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 TL2<7:0>: Timer 2 low bit data register (also as counter low bit).

10.2.3 Timer2 Data Register High Bit TH2

0xCD	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TH2	TH27	TH26	TH25	TH24	TH23	TH22	TH21	TH20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 TH2<7:0>: Timer 2 high-bit data register (also as counter low).

10.2.4 Timer2 Compare/Capture/Auto Reload Register Low Bit RLDL

0xCA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RLDL	RLDL7	RLDL6	RLDL5	RLDL4	RLDL3	RLDL2	RLDL1	RLDL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 RLDL<7:0>: Timer 2 compare/capture/auto reload register low.

10.2.5 Timer2 Compare/Capture/Auto Reload Register High Bit RLDH

0xCB	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RLDH	RLDH7	RLDH6	RLDH5	RLDH4	RLDH3	RLDH2	RLDH1	RLDH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 RLDH<7:0>: Timer 2 compare/capture/auto reload register high bits.

10.2.6 Timer2 Compares/Captures Channel 1 Register Low-bit CCL1

0xC2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CCL1	CCL17	CCL16	CCL15	CCL14	CCL13	CCL12	CCL11	CCL10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 CCL1<7:0>: Timer 2 compares/captures channel 1 register low.

10.2.7 Timer2 Compares/Captures Channel 1 Register High-bit CCH1

0xC3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CCH1	CCH17	CCH16	CCH15	CCH14	CCH13	CCH12	CCH11	CCH10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 CCH1<7:0>: Timer 2 compares/captures channel 1 register high bits.

10.2.8 Timer2 Compares/Captures Channel 2 Register Low-bit CCL2

0xC4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CCL2	CCL27	CCL26	CCL25	CCL24	CCL23	CCL22	CCL21	CCL20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 CCL2<7:0>: Timer 2 compares/captures channel 2 registers low.

10.2.9 Timer2 Compares/Captures Channel 2 Register High-bit CCH2

0xC5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CCH2	CCH27	CCH26	CCH25	CCH24	CCH23	CCH22	CCH21	CCH20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 CCH2<7:0>: Timer 2 compares/captures channel 2 register bits.

10.2.10 Timer2 Compares/Captures Channel 3 Register Low-bit CCL3

0xC6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CCL3	CCL37	CCL36	CCL35	CCL34	CCL33	CCL32	CCL31	CCL30
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 CCL3<7:0>: Timer 2 compares/captures channel 3 registers low.

10.2.11 Timer2 Compares/Captures Channel 3 Register High-bit CCH3

0xC7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CCH3	CCH37	CCH36	CCH35	CCH34	CCH33	CCH32	CCH31	CCH30
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 CCH3<7:0>: Timer 2 compares/captures channel 3 register high bits.

10.2.12 Timer2 Compares the Capture Control Register CCEN

0xCE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CCEN	CMH3	CML3	CMH2	CML2	CMH1	CML1	CMH0	CML0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit6 CMH3-CML3: Capture/Compare Mode Control Bits;
- 00= Capture/Compare Disabled;
- 01= The capture operation is triggered on the rising or falling edge of channel 3 (CAPES selection);
- 10= Comparison mode enable;
- 11= The capture operation is triggered when writing CCL3 or the double edge of channel 3.

Bit5~Bit4 CMH2-CML2: Capture/Compare Mode Control Bits;
- 00= Capture/Compare Disabled;
- 01= The capture operation is triggered on the rising or falling edge of channel 2 (CAPES selection);
- 10= Comparison mode enable;
- 11= The capture operation is triggered when writing CCL2 or the double edge of channel 2.

Bit3~Bit2 CMH1-CML1: Capture/Compare Mode Control Bits;
- 00= Capture/Compare Disabled;
- 01= The capture operation is triggered on the rising or falling edge of channel 1 (CAPES selection);
- 10= Comparison mode enable;
- 11= The capture operation is triggered when writing CCL1 or the double edge of channel 1.

Bit1~Bit0 CMH0-CML0: Capture/Compare Mode Control Bits;
- 00= Capture/Compare Disabled;
- 01= The capture operation is triggered on the rising or falling edge of channel 0 (I3FR selection);
- 10= Comparison mode enable;
- 11= The capture operation is triggered when writing the RLDL or the double edge of channel 0.

10.3 Timer2 Interrupts

Timer 2 can be enabled or disabled by register IE, and high/low priority can also be set via IP registers. Timer2 has 4 interrupt types:

- A timed overflow interrupt.
- The external pin T2EX drops along the interrupt.
- Compare interrupts.
- Capture interrupts.

To set the Timer2 interrupt, configure the global interrupt enable bit (EA=1), the Timer2 global interrupt enable bit (ET2=1), and the corresponding interrupt type enable bit (T2IE) for Timer2. The four types of interrupts in Timer2 all share an interrupt vector, and after entering the interrupt service program, you need to determine the relevant flag bits to determine which type of interrupt is generated.

10.3.1 Interrupt Correlation Registers

10.3.1.1 Interrupt Mask Register IE

0xA8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IE	SHE	ES1	ET2	ES0	ET1	EX1	ET0	EX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 SHE: Global interrupt enable bits;
1= Enable all unblocked interrupts;
0= Disable all interrupts.
Bit6 -- Reserved, must be 0.
Bit5 ET2: TIMER2 Global interrupt enable bits;
1= Enable all interrupts of TIMER2;
0= All interrupts of TIMER2 are disabled.
Bit4 ES0: UART0 interrupt enable bit;
1= Enable UART0 interrupts;
0= Disable UART0 Interrupt.
Bit3 ET1: TIMER1 interrupt enable bit;
1= Enable TIMER1 interrupts;
0= Disable TIMER1 Interrupt.
Bit2 EX1: External interrupt 1 interrupt enable bits;
1= Enable external interrupt 1 interrupt;
0= Disable external interrupt 1 interrupt.
Bit1 ET0: TIMER0 interrupt enable bits;
1= Enable TIMER0 interrupts;
0= Disable TIMER0 Interrupts.
Bit0 EX0: External interrupt 0 interrupt enable bit;
1= Enable external interrupt 0 interrupts;
0= Disable external interrupt 0 interrupt.

10.3.1.2 Timer2 Interrupt Mask Register T2IE

0xCF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2IE	T2OVIE	T2EXIE	--	--	T2C3IE	T2C2IE	T2C1IE	T2C0IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 T2OVIE: Timer2 overflow interrupt enable bits;
1= Interrupts enabled;
0= Disable Interrupt.
Bit6 T2EXIE: Timer2 external loading interrupt enable bits;
1= Interrupts enabled;
0= Disable Interrupt.
Bit5~Bit4 -- Reserved, must be 0.
Bit3 T2C3IE: Timer2 compares channel 3 interrupt enable bits;
1= Interrupts enabled;
0= Disable Interrupt.
Bit2 T2C2IE: Timer2 compares channel 2 interrupt enable bits;
1= Interrupts enabled;
0= Disable Interrupt.
Bit1 T2C1IE: Timer2 compares channel 1 interrupt enable bits;
1= Interrupts enabled;
0= Disable Interrupt.
Bit0 T2C0IE: Timer2 compares channel 0 interrupt Enable bits;
1= Interrupts enabled;
0= Disable Interrupt.

If you want to enable the interrupt of Timer2, you also need to enable the global interrupt enable bit OF TIME2=1 (IE.5=1).

10.3.1.3 Interrupt Priority Control Register IP

0xB8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IP	--	--	PT2	PS0	PT1	PX1	PT0	PX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 -- Reserved, must be 0.
Bit6 -- Reserved, must be 0.
Bit5 PT2: TIMER2 interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.
Bit4 PS0: UART0 interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.
Bit3 PT1: TIMER1 interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.
Bit2 PX1: External interrupt 1 interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.
Bit1 PT0: TIMER0 interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.
Bit0 PX0: External interrupt 0 interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

10.3.1.4 Timer2 Interrupt Flag Bit Register T2IF

0xC9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2IF	TF2	T2EXIF	--	--	T2C3IF	T2C2IF	T2C1IF	T2C0IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 TF2: Timer2 counter overflow interrupt flag bit;
1= Timer2 counter overflow, software zeroing is required;
0= The Timer2 counter has no overflow.
Bit6 T2EXIF: Timer2 externally loaded flag bits;
1= The T2EX port of Timer2 generates a falling edge, which requires software clearance;
0= --
Bit5~Bit4 -- Reserved, must be 0.
Bit3 T2C3IF: Timer2 Compare/Capture Channel 3 Flag Bits;
1= Timer2 Compare channel 3 {CCH3:CCL3}={TH2:TL2} or capture channel 3 produces a capture operation that requires software zeroing.
0= --
Bit2 T2C2IF: Timer2 Compare/Capture Channel 2 Flag Bits;
1= Timer2 Compare channel 2 {CCH2:CCL2}={TH2:TL2} or capture channel 2 to produce a capture operation that requires software zeroing.
0= --
Bit1 T2C1IF: Timer2 Compare/Capture Channel 1 Flag Bits;
1= Timer2 Compare channel 1 {CCH1:CCL1}={TH2:TL2} or capture channel 1 to produce a capture operation that requires software zeroing.
0= --
Bit0 T2C0IF: Timer2 Compare/Capture Channel 0 Flag Bits;
1= Timer2 Compare channel 0{RLDH:RLDL}={TH2:TL2} or capture channel 0 produces a capture operation that requires software zeroing.
0= --

10.3.2 Timer Interrupts

The timer interrupt enable bit is set by register T2IE[7], and the interrupt flag bit is viewed by register T2IF[7]. When the Timer2 timer overflows, the timer overflow interrupt flag bit TF2 will be set to 1.

10.3.3 Externally Triggered Interrupts

The external pin T2EX falling edge trigger interrupt enable bit is set by register T2IE[6], and the interrupt flag bit is viewed by register T2IF[6]. When the T2EX pin drops the edge, the external load interrupt flag bit T2EXIF will be placed at 1.

10.3.4 Compare Interrupts

All four comparison channels support comparison interrupts. The comparison interrupt enable bit is set by register T2IE[3:0] and the interrupt flag bit is viewed by register T2IF[3:0].

Comparing channel 0 can choose to compare the moment when the interrupt occurred, and if an interrupt is generated, the interrupt flag T2C0IF of the comparison channel 0 is set to 1.

When I3FR = 0, TL2/TH2 and RLDL/RLDH never wait until the same moment to produce an interrupt;

When I3FR = 1, TL2/TH2 and RLDL/RLDH are interrupted from the moment of equality to inequality;

Comparing channels 1 to 3 can not choose the interrupt generation time, fixed to TL2/TH2 and CCxL/CCxH never wait until the same time to produce an interrupt. If an interrupt is generated, the corresponding comparison channel interrupt flag T2CxIF is set to 1.

10.3.5 Capture Interrupts

All four capture channels support external capture interrupts. The capture interrupt enable bit is set by register T2IE[3:0], and the interrupt flag bit is viewed by register T2IF[3:0]. When a capture operation is generated, the interrupt flag T2CxIF of the corresponding capture channel is set to 1.

Note Write capture does not produce interrupts.

10.4 Timer2 Feature Description

Timer 2 is a 16-bit up counting timer with a clock source from the system clock. Timer2 can be configured with the following functional modes:

- Timing mode.
- Reload mode.
- Gating timing mode.
- Event counting mode.
- Compare mode.
- Capture mode.

Different modes of timer 2 can be set for generation and event capture of various digital signals, such as pulse generation, pulse width modulation, pulse width measurement, etc.

10.4.1 Timing Mode

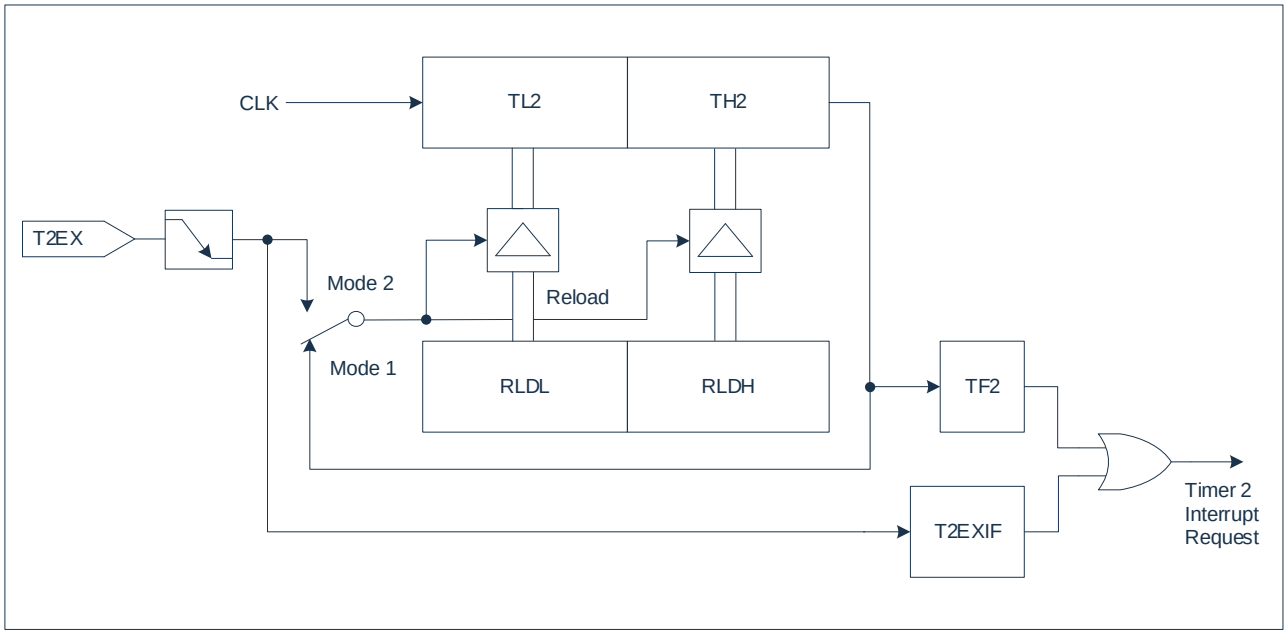
When used as a timer function, the clock source comes from the system clock. The prescaler provides 1/12 or 1/24 system frequency selection, and the value of the prescaler is selected by the T2PS bit of register T2CON. Thus, the 16-bit timer register (consisting of TH2 and TL2) is incremented every 12 clock cycles or every 24 clock cycles.

10.4.2 Reload Mode

The reload mode of timer 2 is selected by the T2R0 and T2R1 bits of register T2CON, as shown in the reload block diagram below.

In load mode 1: When the Timer2 counter is flipped from all 1 to 0 (counter overflow), not only the interrupt flag bit TF2 is set to 1, but also The Timer2 register automatically loads 16 bit value from the RLDL/RLDH register, thus covering the 0x0000 of the count value, and the required RLDL/RLDH value can be preset by the software.

In load mode 2: The 16-bit reload operation from the RLDL/RLDH register is triggered by the falling edge of the corresponding T2EX input pin. When the falling edge of T2EX is detected, the interrupt flag bit T2EXIF is set to 1 externally, while Timer2 automatically loads the 16-bit value of the RLDL/RLDH register as the initial value of the count.



10.4.3 Gated Timing Mode

When Timer2 is used as a gated timer function, the external input pin T2 acts as the gated input to timer 2. If the T2 pin is high, the internal clock input is gated to the timer. A low T2 pin terminates the counting. This function is often used to measure pulse width.

10.4.4 Event Counting Mode

When Timer2 is used as an event counting function, the timer counter adds 1 to the falling edge of the external input pin T2. The external input signal is sampled at each system clock cycle, and the count increases when the sampling input shows a high level for one cycle and a low level for the next. When a change from high to low on the T2 pin is detected in the following cycle, the new count value is updated to the timer data register.

10.4.5 Compare Mode

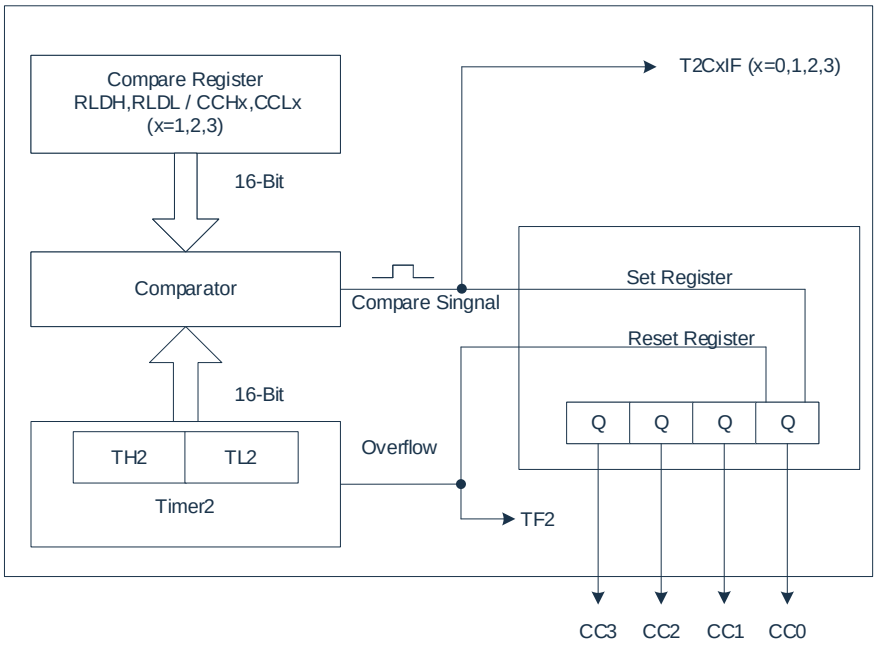
The comparison function consists of two modes: comparison mode 0 and comparison mode 1, selected by the T2CM bit in the special function register T2CON. These two comparison modes generate periodic signals and change the duty cycle control mode, and are often used for pulse width modulation (PWM) and control applications where continuous square waves need to be generated, covering a wide range of applications.

The output channels of the comparison function are CC0, CC1, CC2, CC3. Corresponding to the 16-bit comparison register {RLDH,RLDL},{CCH1,CCL1},{CCH2,CCL2},{CCH3,CCL3} and the data register {TH2,TL2} output signal.

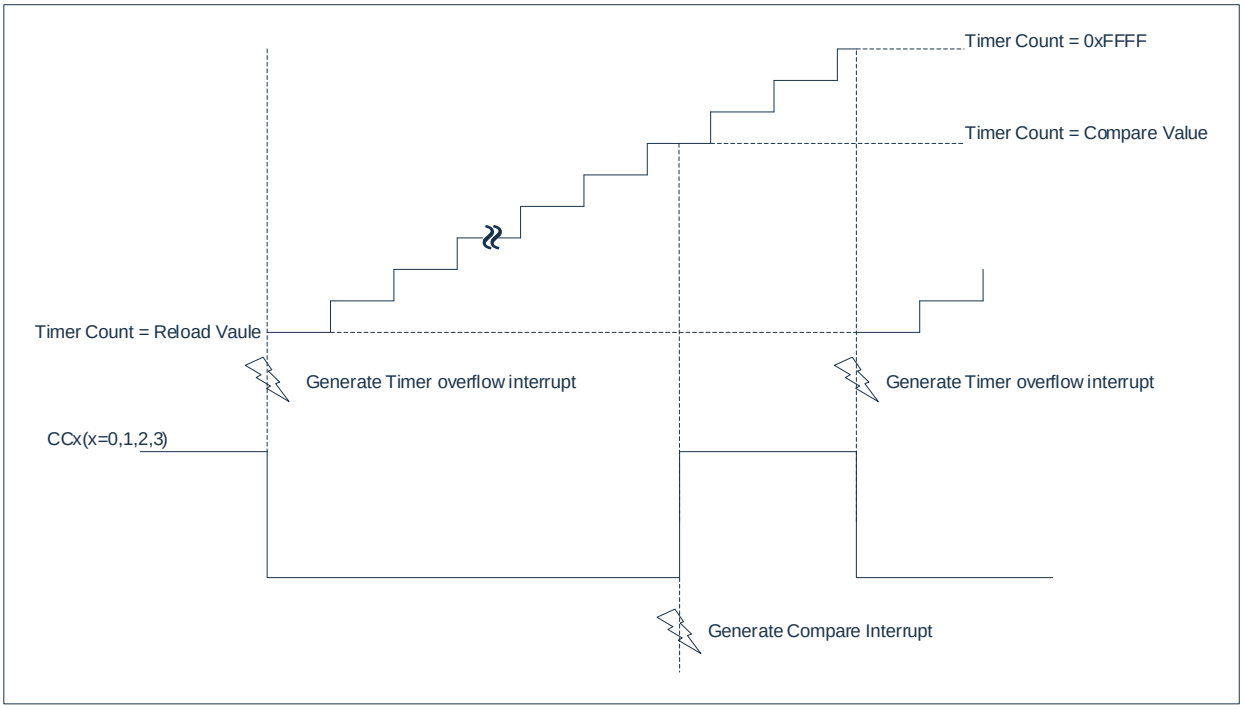
The 16-bit stored value stored in the comparison register is compared to the count value of the timer, and if the count value in the data register matches the stored value, a jump in the output signal is generated on the corresponding port pin and an interrupt flag bit is generated.

10.4.5.1 Compare Mode 0

In mode 0, when the timer's count value and the comparison register are equal, the comparison output signal changes from low to high. When the timer count value overflows, the comparison output signal goes low. The comparison output channel is directly controlled by two events: the timer overflow and the comparison operation. The block diagram of the comparison mode 0 is shown in the following figure:



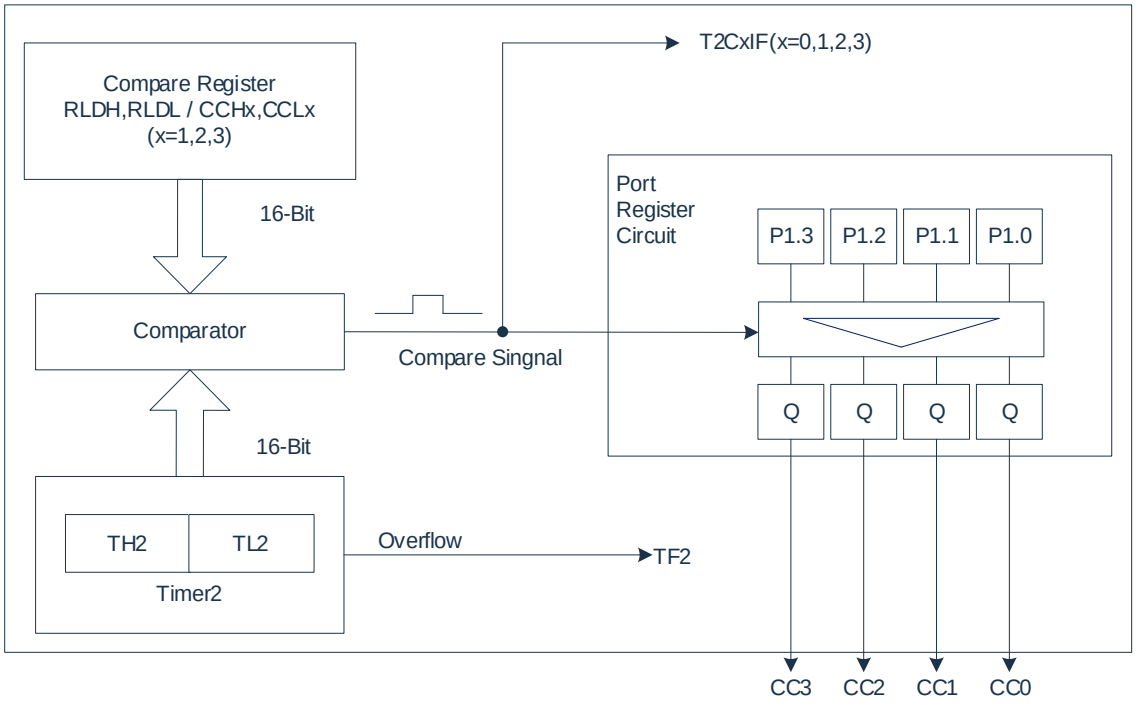
The output block diagram for comparision mode 0 is shown in the following figure:



10.4.5.2 Comparison Mode 1

In comparison mode 1, it is typically used where the output signal is independent of a constant signal cycle, where the software adaptively determines the output signal transition.

If mode 1 is enabled, the software writes to the corresponding output register of the CCx port, and the new value does not appear on the output pin until the next comparison match occurs. When the timer 2 counter matches the stored comparison value, the user can choose whether the output signal changes the new value or keeps its old value in one of two ways. The block diagram of comparision Mode 1 is shown in the following figure:



10.4.6 Capture Mode

Each of the four 16-bit registers {RLDH,RLDL}, {CCH1,CCL1}, {CCH2,CCL2}, {CCH3,CCL3} can be used to latch the current 16-bit value of {TH2,TL2}. This feature provides two different capture modes.

In mode 0, an external event can latch the contents of timer 2 into the capture register.

In mode 1, the capture operation occurs when a low-bit byte (RLDL/CCL1/CCL2/CCL3) is written to the 16-bit capture register. This mode allows the software to read the contents of {TH2,TL2} at runtime.

Capture channels 0 to 3 select the capture input pins CAP0 to CAP3 as the input source signal.

10.4.6.1 Capture Mode 0

In capture mode 0, positive, negative, or positive and negative transactions on capture channels 0 to 3 (CAP0 to CAP3) will produce capture events. When a capture event occurs, the timer's count value lock is stored in the corresponding capture register.

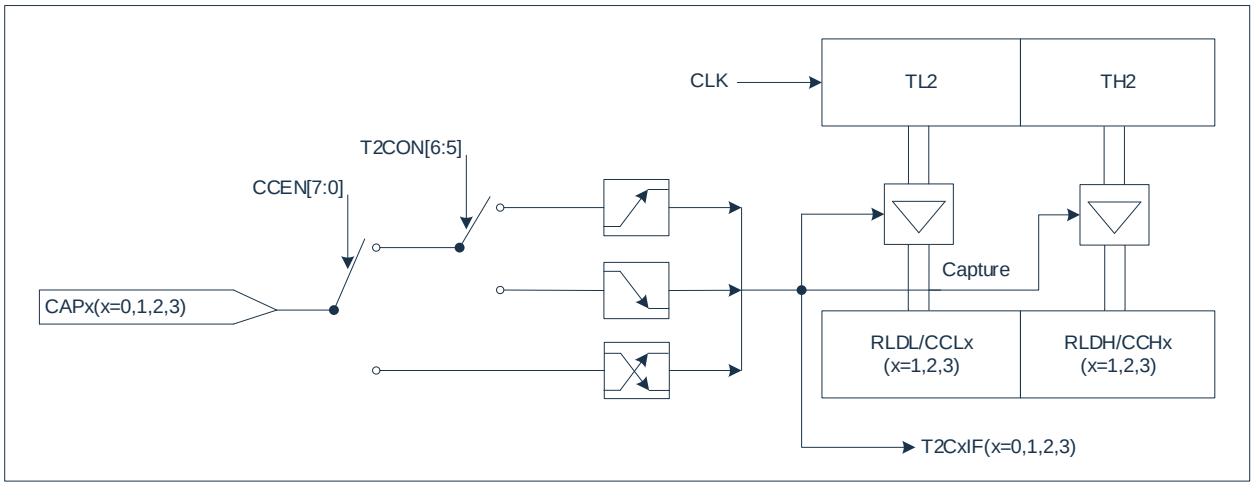
Whether a positive transaction triggers capture operation or a negative transaction triggers capture operation on capture channel 0 depends on the I3FR bit of T2CON. I3FR=0, negative transaction trigger capture; I3FR=1, positive transaction trigger capture.

Whether a positive transaction trigger capture operation or a negative transaction trigger capture operation on capture channels 1 to 3 depends on the CAPES bit of the T2CON. CAPES=0, positive transaction trigger capture; CAPES= 1, negative transaction trigger capture. The transition mode for the selection of capture channels 1 to 3 is the same

Capture channels 0 to 3 support double- transactions capture operations at the same time. Select the corresponding operating mode control bit of the CCEN register to 11, and the channel supports double- transaction capture operation. It should be noted that this mode of operation also supports capture mode 1, that is, the write operation can produce a capture action.

In capture mode 0, external capture events from capture channel 0 to 3 can produce interrupt.

The block diagram of Capture Mode 0 is shown in the following figure:

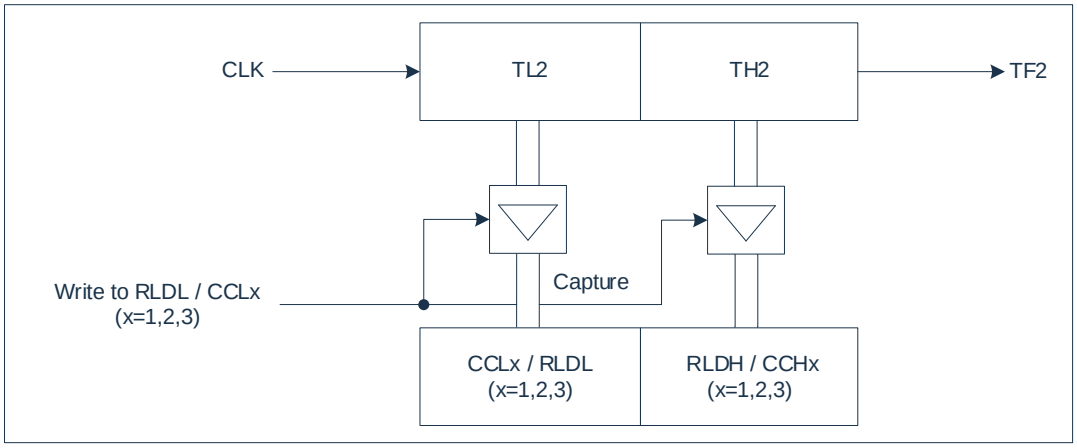


10.4.6.2 Capture Mode 1

In capture mode 1, the capture operation event is the execution of a write byte instruction to the capture register. A write register signal, such as a write RLDL, initiates a capture operation, and the value written is independent of this function. After the write instruction is executed, the contents of timer 2 are latched into the corresponding capture register.

In capture mode 1, capture events for capture channels 0 to 3 do not produce interrupt request flags.

The block diagram of Capture Mode 1 is shown in the following figure:



11. Timer 3/4 (Timer3/4)

Timer 3/4 is similar to timer 0/1 in that it is two 16-bit timers. Timer 3 has four modes of operation and Timer 4 has three modes of operation. In contrast to Timer0/1, Timer3/4 only provides timer operations.

With the timer activated, the value of the register is incremented every 12 or 4 system cycles.

11.1 Overview

Timer 3 and timer 4 consist of two 8-bit registers {TH3,TL3} and {TH4,TL4}. Timers 3,4 operate in four identical modes. The Timer3 and Timer4 modes are described below:

mode	M1	M0	Feature description
0	0	0	THx [7:0], TLx [4:0] make up a 13-bit timer
1	0	1	THx [7:0], TLx [7:0] make up a 16-bit timer
2	1	0	TLx [7:0] consists of an 8-bit auto-reload timer that is reinstalled from THx
3	1	1	TL3, TH3 are two 8-bit timers, and Timer4 stop counting

11.2 Related Registers

11.2.1 Timer3/4 Control Register T34MOD

0xD2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T34MOD	TR4	T4M	T4M1	T4M0	TR3	T3M	T3M1	T3M0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 TR4: Timer4 operational control bit;
1= Timer4 starts;
0= Timer4 is off.

Bit6 T4M: Timer 4 clock select bits;
1= Fsys/4;
0= Fsys/12。

Bit5~Bit4 T4M<1:0>: Timer 4 mode select bit;
00= Mode 0, 13-bit timer;
01= Mode 1, 16-bit timer;
10= Mode 2, 8-bit automatic reload timer;
11= Mode 3, Stop Count.

Bit3 TR3: Timer3 operational control bit;
1= Timer3 starts;
0= Timer3 is off.

Bit2 T3M: Timer 3 clock select bits;
1= Fsys/4;
0= Fsys/12。

Bit1~Bit0 T3M<1:0>: Timer 3 mode select bit;
00= Mode 0, 13-bit timer;
01= Mode 1, 16-bit timer;
10= Mode 2, 8-bit automatic reload timer;
11= Mode 3, two independent 8-bit timers.

11.2.2 Timer3 data register low bit TL3

0xDA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TL3	TL37	TL36	TL35	TL34	TL33	TL32	TL31	TL30
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 TL3<7:0>: Timer 3 low bit data register (while acting as timer low bit).

11.2.3 Timer3 data register high bit TH3

0xDB	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TH3	TH37	TH36	TH35	TH34	TH33	TH32	TH31	TH30
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 TH3<7:0>: Timer 3 high bit data register (also as timer high bit).

11.2.4 Timer4 data register low bit TL4

0xE2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TL4	TL47	TL46	TL45	TL44	TL43	TL42	TL41	TL40
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 TL4<7:0>: Timer 4 low bit data register (also as timer low bit).

11.2.5 Timer4 data register high bit TH4

0xE3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TH4	TH47	TH46	TH45	TH44	TH43	TH42	TH41	TH40
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 TH4<7:0>: Timer 4 high bit data register (also as timer high bit).

11.3 Timer3/4 Interrupt

Timer 3/4 can enable or disable interrupts via the EIE2 register, and high/low priority can also be set via the EIP2 register, where the relevant bits are described as following:

11.3.1 Interrupt mask register EIE2

0xAA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIE2	SPIIE	I2CIE	WDTIE	ADCIE	PWMIE	--	ET4	ET3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 SPIIE: SPI interrupt enable bit;
1= Enable SPI interrupts;
0= Disable SPI Interrupt.
Bit6 I2CIE I2C Interrupt enable bit;
1= Allow I2C Interrupts;
0= Disable I2C Interrupt.
Bit5 WDTIE: WDT interrupt enable bit;
1= Enable WDT overflow interrupts;
0= Disable WDT overflow interrupts.
Bit4 ADCIE: ADC interrupt enable bit;
1= Enable ADC interrupts;
0= Disable ADC interrupts.
Bit3 PWMIE: PWM global interrupt enable bit;
1= Enable all PWM interrupts;
0= Disable all PWM interrupts.
Bit2 -- Reserved, must be 0.
Bit1 ET4: Timer4 interrupt enable bit;
1= Enable Timer4 interrupts;
0= Disable Timer4 Interrupt.
Bit0 ET3: Timer3 interrupt enable bit;
1= Enable Timer3 interrupts;
0= Disable Timer3 Interrupt.

11.3.2 Interrupt Priority Control Register EIP2

0xBA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP2	PSPI	PI2C	PWDT	PADC	PPWM	--	PT4	PT3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 PSPI: SPI interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

Bit6 PI2C: I2C Interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

Bit5 PWDT: WDT interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

Bit4 PADC: ADC interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

Bit3 PPWM: PWM interrupt priority control bit
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

Bit2 --: Reserved, must be 0.

Bit1 PT4: TIMER4 interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

Bit0 PT3: TIMER3 interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

11.3.3 Peripheral Interrupt Flag Bit Register EIF2

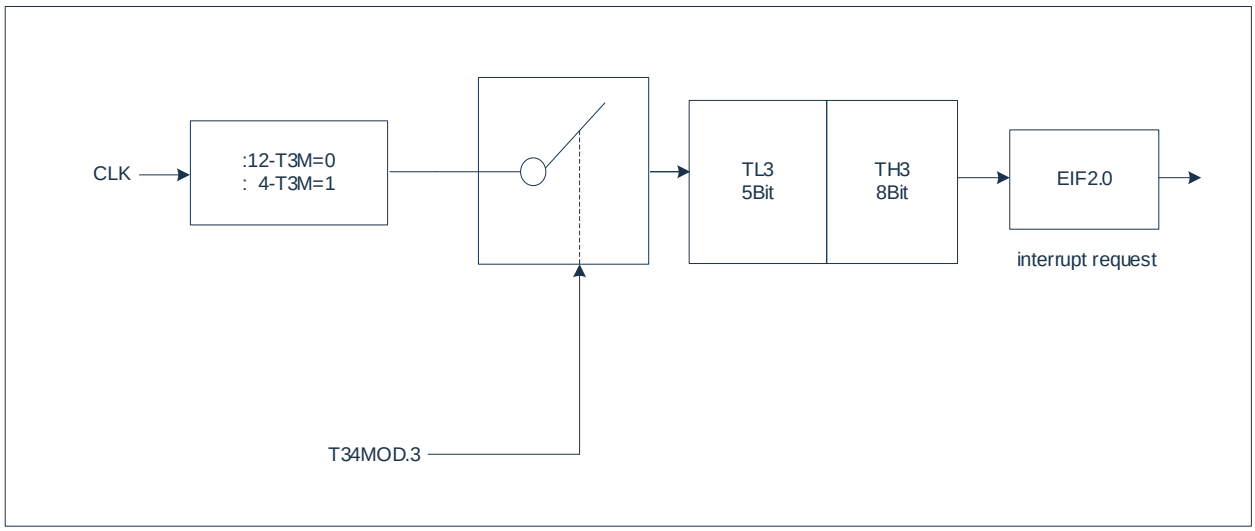
0xB2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIF2	SPIIF	I2CIF	--	ADCIF	PWMIF	--	TF4	TF3
R/W	R	R	--	R/W	R	--	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 SPIIF: SPI global interrupt indicator bit, read-only;
1= SPI generates an interrupt, (this bit is automatically cleared after the specific interrupt flag is cleared);
0= The SPI did not produce an interrupt.
Bit6 I2CIF: I2C global interrupt indicator bit, read-only;
1= I2C produces an interrupt, (after clearing the specific interrupt flag, this bit is automatically cleared);
0= I2C did not produce an interrupt.
Bit5 -- Reserved, must be 0.
Bit4 ADCIF: ADC interrupt flag bit;
1= ADC conversion is completed, and software zeroing is required;
0= The ADC conversion was not completed.
Bit3 PWMIF: PWM global interrupt indicator bit, read-only;
1= PWM generates an interrupt, (after the specific interrupt flag is cleared, this bit is automatically cleared);
0= The PWM did not produce an interrupt.
Bit2 -- Reserved, must be 0.
Bit1 TF4: Timer4 timer overflow interrupt flag bit;
1= Timer4 timer overflow, the hardware is automatically cleared when entering the interrupt service program, and the software can also be cleared;
0= The Timer4 timer has no overflow.
Bit0 TF3: Timer3 timer overflow interrupt flag bit;
1= Timer3 timer overflow, when entering the interrupt service program, the hardware is automatically cleared, and the software can also be cleared;
0= The Timer3 timer has no overflow.

11.4 Timer3 Working Mode

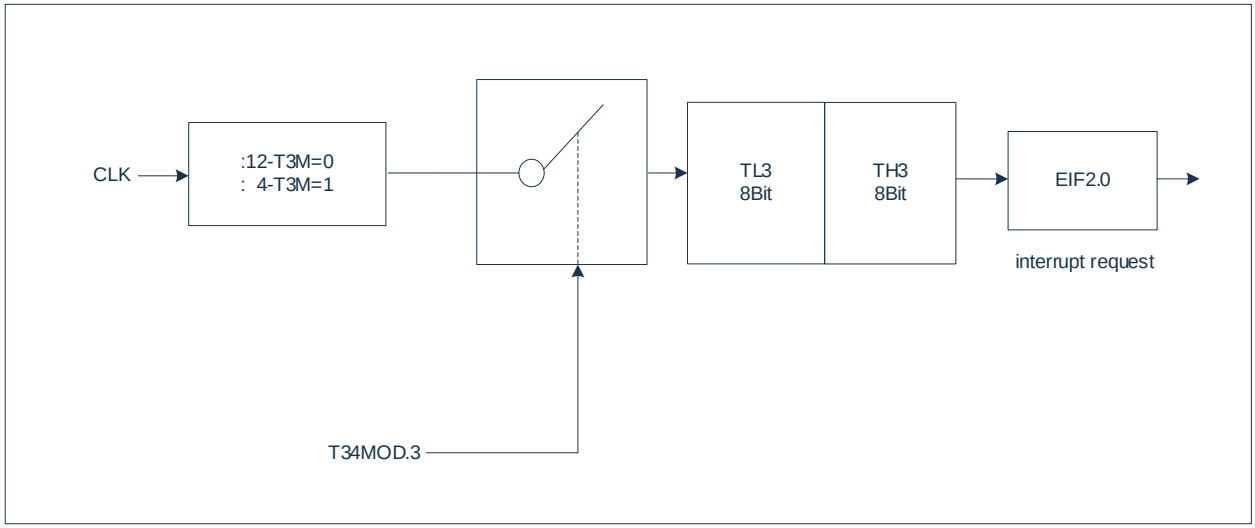
11.4.1 T3 - Mode 0 (13-bit Timing Mode)

In this mode, timer 3 is a 13-bit register. When all the bits of the timer are flipped from 1 to 0, the timer 3 interrupt flag TF3 is set to 1. The 13-bit register consists of TH3 and TL3 low 5 bits. TL3 high 3 bits should be ignored. The block diagram of the Timer3 mode 0 is shown in the following figure:



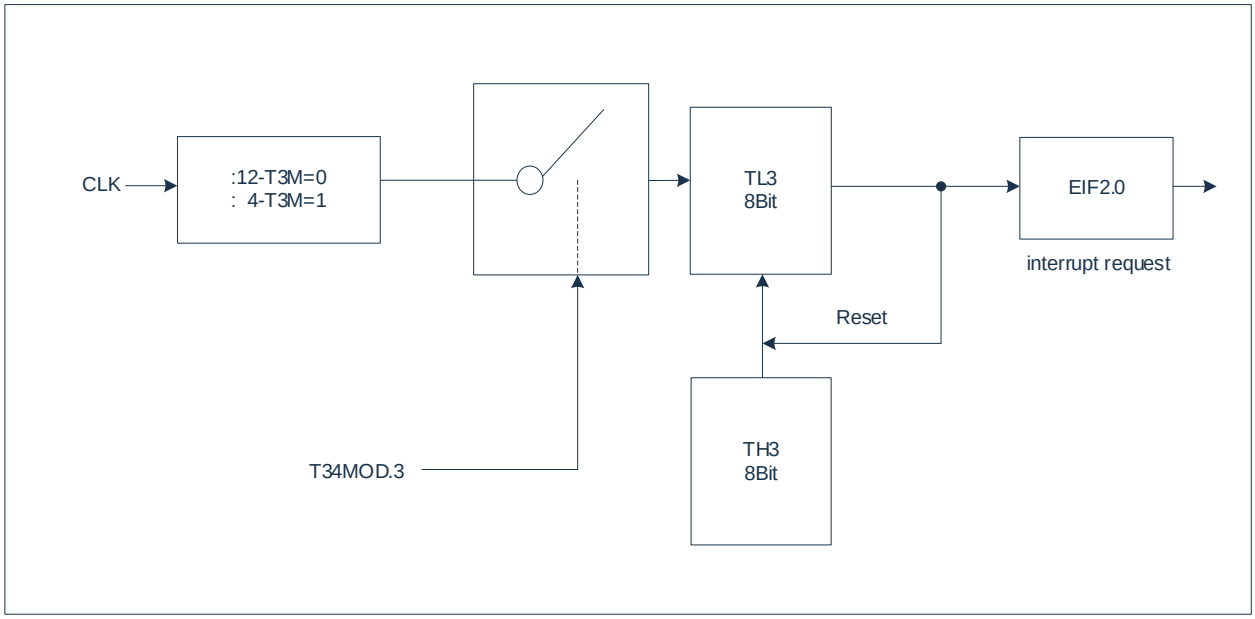
11.4.2 T3 - Mode 1 (16-bit Timing Mode)

Mode 1 is the same as mode 0, except that timer 3 registers 16 bits are all running in mode 1. The block diagram of Timer3 Mode 1 is shown in the following figure:



11.4.3 T3 - Mode 2 (8-bit Auto Reload Timing Mode)

The timer 3 register in mode 2 is an 8-bit timer (TL3) with auto reload mode, as shown in the figure below. The overflow from TL3 not only puts TF3 at 1, but also reloads the contents of TH3 from software to TL3. The value of TH3 remains unchanged during Reloading. The block diagram of the Timer3 mode 2 is shown in the following figure:



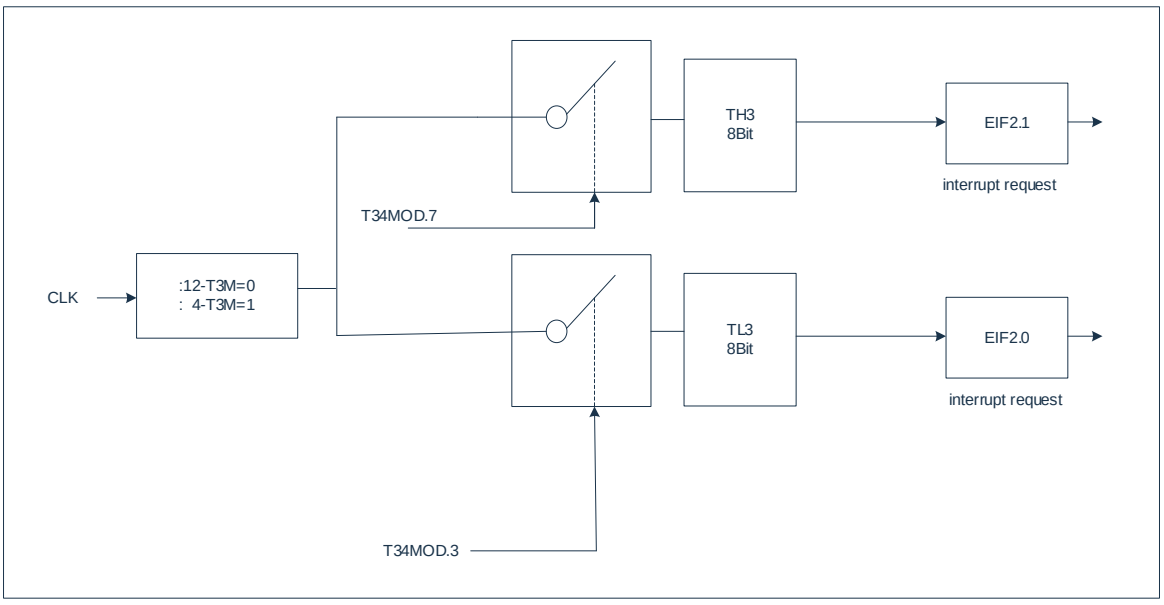
11.4.4 T3 - Mode 3 (Two Separate 8-bit Timers)

Timer 3 in mode 3 sets TL3 and TH3 to two independent timers. The logic of timer 3 mode 3 is shown in the following figure.

TL3 operates as an 8-bit timer and uses the control bits of timer 3, such as TR3, and TF3.

TH3 operates as an 8-bit timer and uses the TR4 and TF4 flags of Timer 4 and controls Timer 4 interrupts.

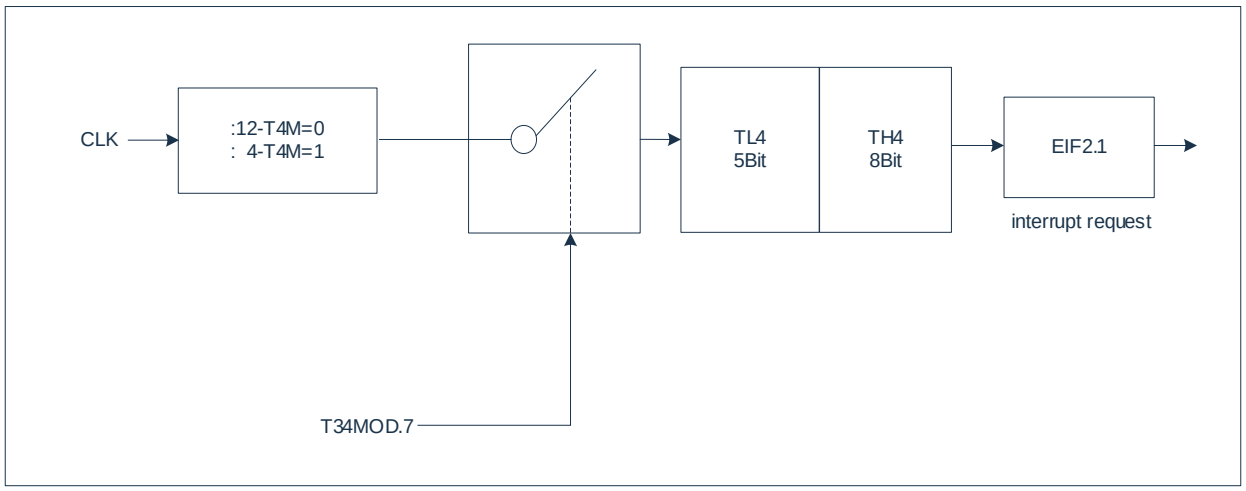
Mode 3 can be used when two 8-bit timers are required. When Timer 3 is in Mode 3, Timer 4 can turn it off by switching to its own Mode 3, or it can still be used as a baud rate generator by the serial channel, or in any application that does not require a timer 4 interrupt. The block diagram of Timer3 mode 3 is shown in the following figure:



11.5 Timer4 Working Mode

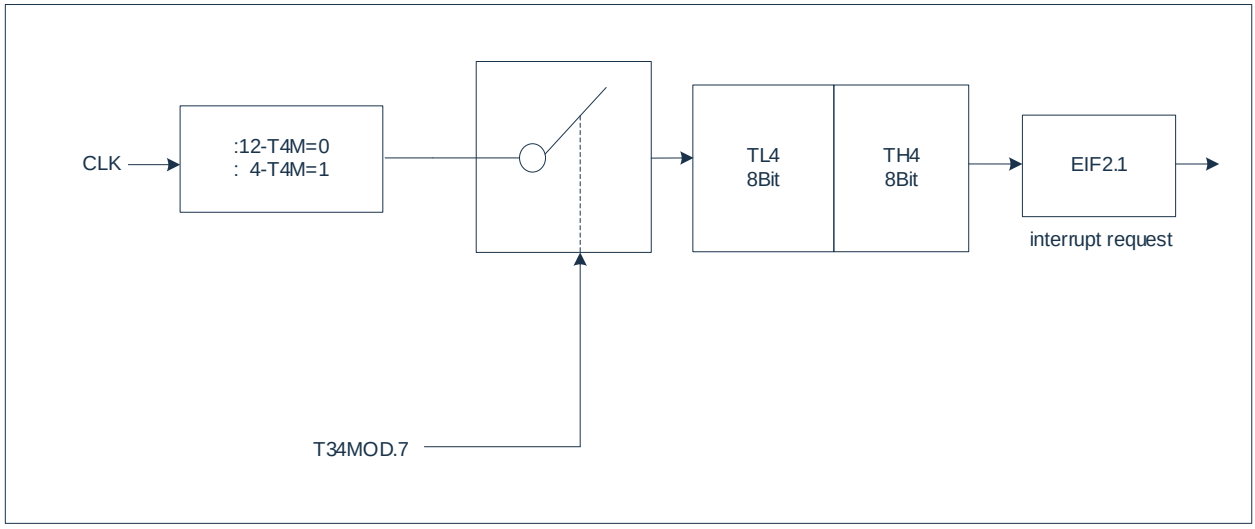
11.5.1 T4 - Mode 0 (13-bit Timing Mode)

In this mode, timer 4 is a 13-bit register. When all the bits of the timer are flipped from 1 to 0, the timer 4 interrupt flag TF4 is set to 1. The 13-bit register consists of TH4 8 bits and TL4 low 5 bits. TL4 high three bits should be ignored. The block diagram of Timer4 mode 0 is shown in the following figure:



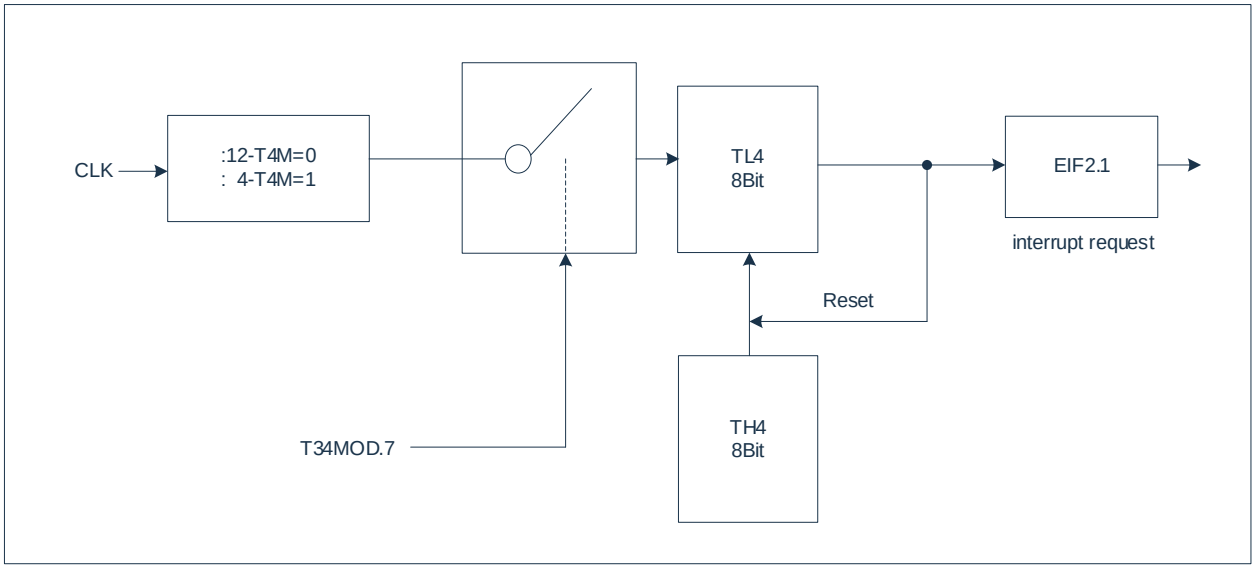
11.5.2 T4 - Mode 1 (16-bit timing mode)

Mode 1 is the same as Mode 0, except that the timer 4 registers 16 bits are all running in mode 1. The block diagram of Timer4 mode 1 is shown in the following figure:



11.5.3 T4- Mode 2 (8-bit auto Reload Timing Mode)

The timer 4 register in mode 2 is an 8-bit timer (TL4) with an auto-reload mode, as shown in the figure below. The overflow from TL4 not only makes TF4 1, but also reloads the contents of TH4 from software to TL4. The value of TH4 remains unchanged during Reloading. The block diagram of Timer4 mode 2 is shown in the following figure:



11.5.4 T4 - Mode 3 (Stop Count)

Timer4 in mode 3 stops counting with the same effect as setting TR4=0.

12. LSE Timer(LSE_Timer)

12.1 Overview

The LSE timer is a clock source from an external low-speed clock LSE, a 16-bit up-counting timer. When using the LSE timer function, you should first set the LSE module to enable, wait for the LSE clock to stabilize (about 1.5s), and then set the LSE count enable. The counter adds 1 to the count value on the rising edge of the LSE clock, and when the count value is equal to the timing value, the interrupt flag LSECON[0] is set to 1, and the counter starts counting from 0 again. The timing value is set by registers {LSECRH[7:0], LSECRL[7:0]}.

If the LSE timing function is configured before hibernation, the LSE oscillator and LSE timer can continue to operate without being affected while the chip is asleep. If the LSE timed wake function is set before sleep, the system will wake up when the count value is equal to the timed value.

12.2 Related Registers

12.2.1 LSE Timer Data Register Low 8 bit LSECRL

F694H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LSECRL	LSED7	LSED6	LSED5	LSED4	LSED3	LSED2	LSED1	LSED0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	1	1	1	1	1	1	1	1

Bit7~Bit0 LSED<7:0>: LSE timing/wake-up time data is 8 bits lower.

12.2.2 LSE Timer Data Registers are 8 Bits High LSECRH

F695H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LSECRH	LSED15	LSED14	LSED13	LSED12	LSED11	LSED10	LSED9	LSED8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	1	1	1	1	1	1	1	1

Bit7~Bit0 LSED<15:8>: LSE timing/wake-up time data is 8 bits higher.

12.2.3 LSE Timer Control Register LSECON

F696H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LSECON	LSEEN	LSEWUEN	LSECNTEN	LSESTA	LSEIE	--	--	LSEIF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 LSEEN: LSE module enable control;
1= Enable;
0= Disable.

Bit6 LSEWUEN: LSE timer wake-up enable control;
1= Enable;
0= Disable.

Bit5 LSECNTEN: LSE as timer count enable control;
1= Enable;
0= Disable.

Bit4 LSESTA: LSE steady-state bit, read-only;
1= LSE stability;
0= The LSE is not stable.

Bit3 LSEIE: LSE as timer interrupt enable control;
1= Enable;
0= Disable.

Bit2~Bit1 -- Reserved, must be 0.

Bit0 LSEIF: LSE as timer interrupt flag bit (software clear 0);
1= An interrupt is generated.
0= No interrupts were generated or the breaks were cleared to zero out.

12.3 Interrupt With Sleep Wake-up

The LSE timer can enable or disable interrupts via LSECON registers, setting high/low priority via EIP3 registers, where the relevant bits are described as following.

0xBB	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP3	--	--	--	--	PLVD	PLSE	--	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit5	--	Reserved, all must be 0.
Bit4	--	Reserved, all must be 0.
Bit3	PLVD:	LVD interrupt priority control bit;
	1=	Set to High-level Interrupt;
	0=	Set to low-level interrupt.
Bit2	PLSE:	LSE interrupt priority control bit;
	1=	Set to High-level Interrupt;
	0=	Set to low-level interrupt.
Bit1~Bit0	--	Reserved, all must be 0.

When the count value of the LSE timer is equal to the timer value, the off flag bit LSEIF in the timer is set to 1. If the global interrupt is enabled (EA=1) and the LSE timer interrupt is enabled (LSEIE=1), the CPU executes the interrupt service program.

Using LSE timed interrupt wake-up sleep mode, you need to enable LSEEN, LSECNT, LSEWUEN before hibernation, and set the post-hibernation state to wake up time. If the global interrupt enable and LSE interrupt enable are turned on before hibernation, after hibernation wakes up, the interrupt service program is executed first, and the next instruction of the hibernation instruction is executed after the interrupt returns. { LSECRH[7:0]，LSECRL[7:0]}

12.4 Feature Description

To use the LSE timer function, you need to set LSEEN=1 to enable the LSE timer function module, and then wait for the LSE clock steady state bit LSESTA=1, then configure the LSE timing value {LSECRH[7:0], LSECRL[7:0]}, and finally set LSECNT=1 to enable LSE count and turn on LSE counting function. The LSE timer counts from 0, interrupt flag bit set to 1 when the count value is equal to the timing value, and update the timing value to the value in the timer data register (i.e. the LSE timing value is the last time {LSECRH[7:0], LSECRL[7:0]} value is written before the LSE timing value is equal to the count value and the timing value). The minimum timer value is 1, if the timer value is set to 0, the timer defaults to 1 as the timer value. The LSE timer configured time calculation formula is as follows:

LSE configured time $=\frac{1}{32.768}\times(\{\text{LSECRH[7:0]，LSECRL[7:0]}\}+1)$ ms

LSEEN, LSECNTEN, LSESTA any bit of the LSE timer is 0, and the count value of the LSE will be cleared.

13. Wake-up Timer (WUT)

13.1 Overview

Wake Up Timer is a clock source from the internal low-speed clock LSI, a 12-bit, up-counting timer for sleep wake-up, and after the timer count and the configured 12-bit count value are equal, the timing overflow state bit set to 1 which can be cleared to 0 by software. Can be used to wake up the system at a configured time in sleep mode. Configure the timed wake-up time before the system goes to sleep and enable the timed wake-up function. When the chip enters sleep mode, the WUT starts counting, and when the count value is equal to the configured value, the chip enters a sleep wake-up wait state.

13.2 Related Registers

13.2.1 WUTCRH Register

0xBD	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WUTCRH	WUTEN	HOURS_OV	WUTPS1	WUTPS0	WUTD11	WUTD10	WUTD9	WUTD8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7	WUTEN:	Timed wake-up function enable bit;
	1=	The timed wake-up function is turned on;
	0=	The timed wake-up function is disabled.
Bit6	TIMER_OV:	Timing overflow status bits;
	1=	Timed overflow
	0=	Software clear 0
Bit5~Bit4	WUTPS<1:0>:	Timed wake-up counter clock divider;
	00=	F/1;
	01=	F/8;
	10=	F/32;
	11=	F/256。
Bit3~Bit0	WUTD<11:8>:	The timed wake-up time data is 4 bits higher.

13.2.2 WUTCRL Register

0xBC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WUTCRL	WUTD7	WUTD6	WUTD5	WUTD4	WUTD3	WUTD2	WUTD1	WUTD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0	WUTD<7:0>:	The timed wake-up time data is 8 bits lower.

13.3 Feature Description

The internal wake-up timer works on the principle that after the system enters sleep mode, the CPU stops working with all peripheral circuitry, and the internal low-power oscillator LSI begins to operate, and its oscillation clock is 125KHz (T_{LSI}≈ 8us). Provides a clock for the WUT counter.

There are two internal wake-up timer registers: WUTCRH and WUTRCL.

Bit7 of the WUTCRH register is an internal timed wake-up enable bit:

- WUTEN=1: Turn on the timed wake-up function;
- WUTEN=0: Turns off timed wake-up.

{WUTCRH[3:0] and WUTCRL[7:0]} form a 12-bit timed wake-up data register, after entering sleep mode, the HUT counter starts to time, when the value of the WUT counter is equal to the value of the timed wake-up data register, start the system oscillator, enter the wake-up waiting state.

Timed wake-up time:$T=(WUTD[11:0]+1)\times WUTPS\times T_{LSI}$

14. Baud Rate Timer (BRT)

14.1 Overview

The chip has a 16-bit baud rate timer BRT, which mainly provides a clock for the UART module.

14.2 Related Registers

14.2.1 BRT Module Control Register BRTCon

F5C0H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BRTCON	BRTEN	--	--	--	--	BRTCKDIV2	BRTCKDIV1	BRTCKDIV0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7	BRTEN:	BRT timer enable bit;
	1=	Enable;
	0=	Disable.
Bit6~Bit3	--	Reserved, both 0;
Bit2~Bit0	BRTCKDIV<2:0>	BRT timer prescale selection bit;
	000=	Fsys/1;
	001=	Fsys/2;
	010=	Fsys/4;
	011=	Fsys/8;
	100=	Fsys/16;
	101=	Fsys/32;
	110=	Fsys/64;
	111=	Fsys/128。

14.2.2 The BRT Timer Data is Loaded With a Low 8-bit Register BRTDL

F5C1H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BRTDL	BRTDL7	BRTDL6	BRTDL5	BRTDL4	BRTDL3	BRTDL2	BRTDL1	BRTDL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 BRTDL<7:0>: BRT timer load value 8 bits lower;

14.2.3 The BRT Timer Data is Loaded With a High 8-bit Register BRTDH

F5C2H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BRTDH	BRTDH7	BRTDH6	BRTDH5	BRTDH4	BRTDH3	BRTDH2	BRTDH1	BRTDH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 BRTDH<7:0>: BRT timer load value 8 bits higher;

14.3 Feature Description

The BRT has a 16-bit increment counter, the clock is derived from the pre-division circuit, the pre-division clock is determined by the timer pre-division select bit BRTCKDIV, and the initial value of the counter is loaded by {BRTDH, BRTDL}.

When the timer enable bit BRTEN=1 is turned on, the counter starts working. When the value of the 16-bit counter is equal to FFFFH, the BRT counter overflows. After the overflow, the initial value of the count is automatically loaded into the counter and then the count is re-counted.

The overflow signal of the BRT counter is specially provided to the UART module as a clock source for the baud rate, and there is no interrupt when overflowing, and there is no corresponding interrupt structure. BrT in debug mode, its clock does not stop, if the UART module has begun to send or receive data, even if the chip into a suspended state, the UART will complete the entire process of sending or receiving.

BRT timer overflow rate:

$$BRTov=\frac{Fsys}{(65536-\{BRTDH,BRTDL\})\times 2^{BRTCKDIV}}$$

15. Buzzer Driver (BUZZER)

15.1 Overview

The buzzer drive module consists of an 8-bit counter, a clock driver, and a control register. The buzzer drives a 50% duty-square wave with a frequency set by registers BUZCON and BUZDIV, and its frequency output covers a wide range.

15.2 Related Registers

15.2.1 BUZZER Control Register BUZCON

0xBF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BUZCON	BUSEN	--	--	--	--	--	BUZCKS1	BUZCKS0
R/W	R/W	R	R	R	R	R	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 BUSEN: BUZZER enable bit;
1= Enable;
0= Disable.
Bit6~Bit2 -- Reserved, must be 0.
Bit1~Bit0 BUZCKS<1:0>: BUZZER divide-by-side ratio select bits;
00= Fsys/8;
01= Fsys/16;
10= Fsys/32;
11= Fsys/64。

15.2.2 BUZZER Frequency Control Register BUZDIV

0xBE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BUZDIV	BUZDIV7	BUZDIV6	BUZDIV5	BUZDIV4	BUZDIV3	BUZDIV2	BUZDIV1	BUZDIV0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 BUZDIV<7:0>: BUZZER frequency select bit;
0x00= No square wave output ;
Other = Fbuz =Fsys/(2*CLKDIV*BUZCKS)。

Note: It is not recommended to modify BUZDIV during BUZEN=1

15.3 Feature Description

When using a buzzer, you need to configure the corresponding port as a buzzer-driven output. For example, configure P17 as a buzzer drive output as follows:

P17CFG = 0x05; The P1 7 is configured as a buzzer-driven output

By configuring the Related Registers of the buzzer drive module, it is possible to set the different frequencies at which the buzzer drive outputs. For example:

1) Set Fsys= 8MHz, BUZCKS < 1:0>=01, and BUZDIV=125

   The buzzer drive output frequency is: Fbuz = 8MHz / (2 * 125) / 16 = 2KHz

2) Set Fsys=16MHz, BUZCKS < 1:0>=11, and BUZDIV=125

   The buzzer drive output frequency is: Fbuz = 16MHz / (2 * 125) / 64 = 1KHz

3) Set Fsys=24MHz, BUZCKS < 1:0>=11, and BUZDIV=94

   The buzzer drive output frequency is: Fbuz = 24MHz / (2 * 94) / 64 = 2KHz

Different system clock frequencies and buzzer-driven clock divide ratios can be selected to obtain different output frequencies. The buzzer drive output frequency is shown in the following table:

BUZCKS<1:0>	Fbuz@Fsys=8MHz	Fbuz@Fsys=16MHz	Fbuz@Fsys=24MHz	Fbuz@Fsys=48MHz
00	2KHz~500KHz	4KHz~1MHz	6KHz~1.5MHz	12KHz~3MHz
01	1KHz~250KHz	2KHz~500KHz	3KHz~750KHz	6KHz~1.5MHz
10	0.5KHz~125KHz	1KHz~250KHz	1.5KHz~375KHz	3KHz~750KHz
11	0.25KHz~62.5KHz	0.5KHz~125KHz	0.75KHz~187.5KHz	1.5KHz~375KHz

16. Enhanced PWM Module

16.1 Overview

The enhanced PWM module supports four PWM generators, which can be configured as four independent PWM outputs (PG0-PG3), or as 2 sets of synchronous PWM outputs, or 2 pairs of complementary PWM outputs with programmable dead-zone generators, where PG0-PG1 and PG2-PG3 are pairs 。

Each PWM has its own 16-bit cycle register, a 16-bit duty cycle register (compare data register and compare down data register) to configure the cycle of the PWM output and adjust the duty cycle. Each PWM has its own clock-divider control registers, and each pair of PWMs shares an 8-bit prescale control register.

Each PWM can be configured as edge alignment count or center alignment count mode, and center alignment count mode can also be set to symmetric count or asymmetric count. Each PWM can be set in single mode (generating a PWM signal cycle) or autoloaded (continuous output PWM waveform) output, and its output polarity can also be set by the output polarity controller.

The enhanced PWM also supports mask output function, hardware brake protection function, and interrupt function. The 4-way PWM generator provides a total of 25 interrupt flags, including zero interrupt, up-compare interrupt, cycle interrupt, down-compare interrupt, brake interrupt, which share a single interrupt vector entry.

16.2 Characteristic

The Enhanced PWM Module has the following features:

- 4 independent 16-bit PWM control modes.
  - 4 independent outputs: PG0, PG1, PG2, PG3;
  - 2 sets of complementary PWM pairs can be inserted into the output :(PG0-PG1) and (PG2-PG3) with programmable dead-zone;
  - 2 sets of synchronous PWM pairs are :(PG0-PG1) and (PG2-PG3) each group of PWM pairs are pin synchronized.
- Support group control, PG0, PG2 output synchronization, PG1, PG3 output synchronization.
- Support edge alignment, center alignment 2 modes.
- Center alignment mode supports both symmetric and asymmetric counts.
- Support single-shot mode or auto-load mode.
- Each PWM has independent polarity control.
- Mask output function.
- Hardware brake protection and recovery functions (external FB0/FB1 trigger, ADC comparison event trigger, ACMP output trigger).
- The PWM edge or period can trigger the initiation of the AD conversion.

16.3 Port Configuration

Before using the enhanced PWM module, the corresponding port needs to be configured as a PWM channel, and the PWM channel is marked with PG0~PG3 on the multiplexing function allocation table, corresponding to PWM channel 0~3.

The allocation of PWM channels is controlled by the corresponding port configuration registers, for example:

P00CFG=0x04; Configure the P00 as a PG0 channel

P01CFG=0x04; Configure P01 as PG1 channel

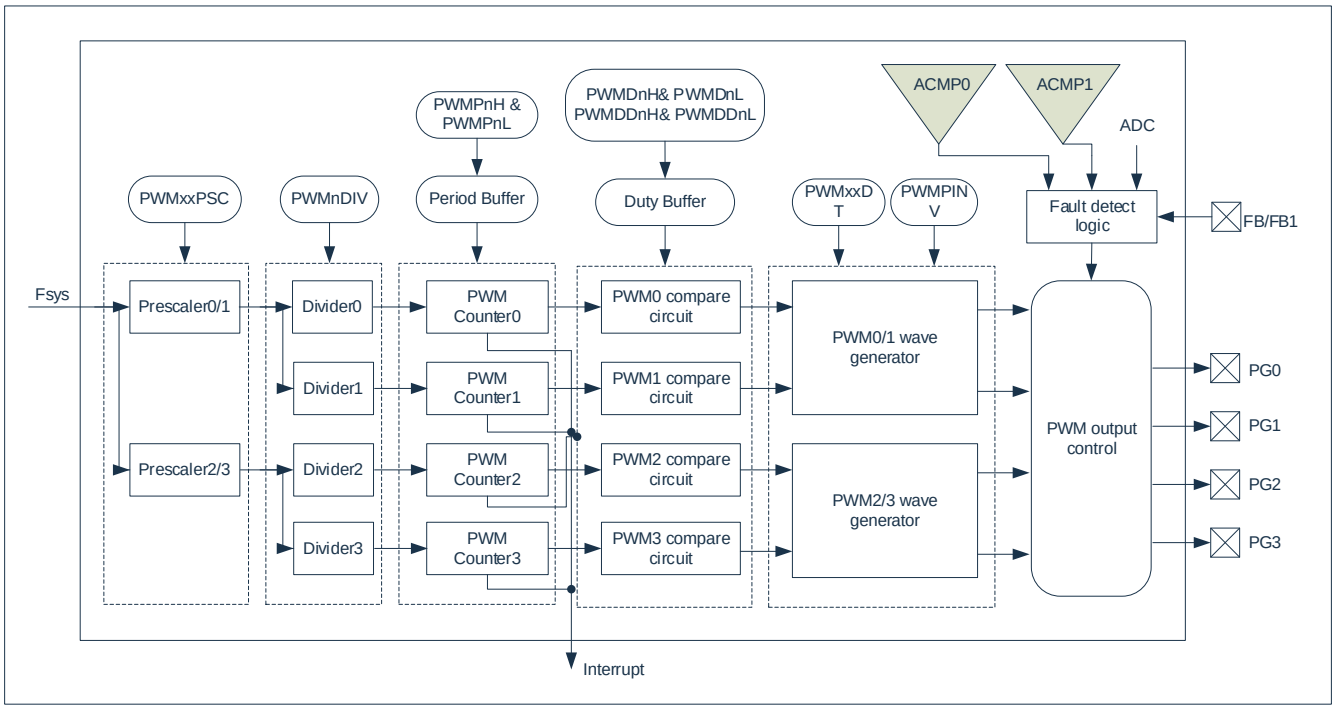
P02CFG=0x04; Configure P02 as a PG2 channel

P03CFG=0x04; Configure the P03 as a PG3 channel

16.4 Feature Description

16.4.1 Functional Block Diagram

The enhanced PWM consists of a clock control module, a PWM counter module, an output comparison unit, a waveform generator, a brake protection module (fault detection) and an output controller, and its block diagram is shown in the following figure:



16.4.2 Edge Alignment

In edge alignment mode, the 16-bit PWM counter CNTn starts counting down at the beginning of each cycle and compares with the value C in the PWMDnH/PWMDnL register MPn is compared, when CNTn= CMPn, PGn output high, PWMnDIF set to 1. CNTn continues to count down to 0, at which point PGn will output low and PWMnZIF will be set to 1. When CNTncounts to zero, CMPn and PERIODn are reloaded if PWMnCNTM=1.

The relevant parameters for edge alignment are as follows:

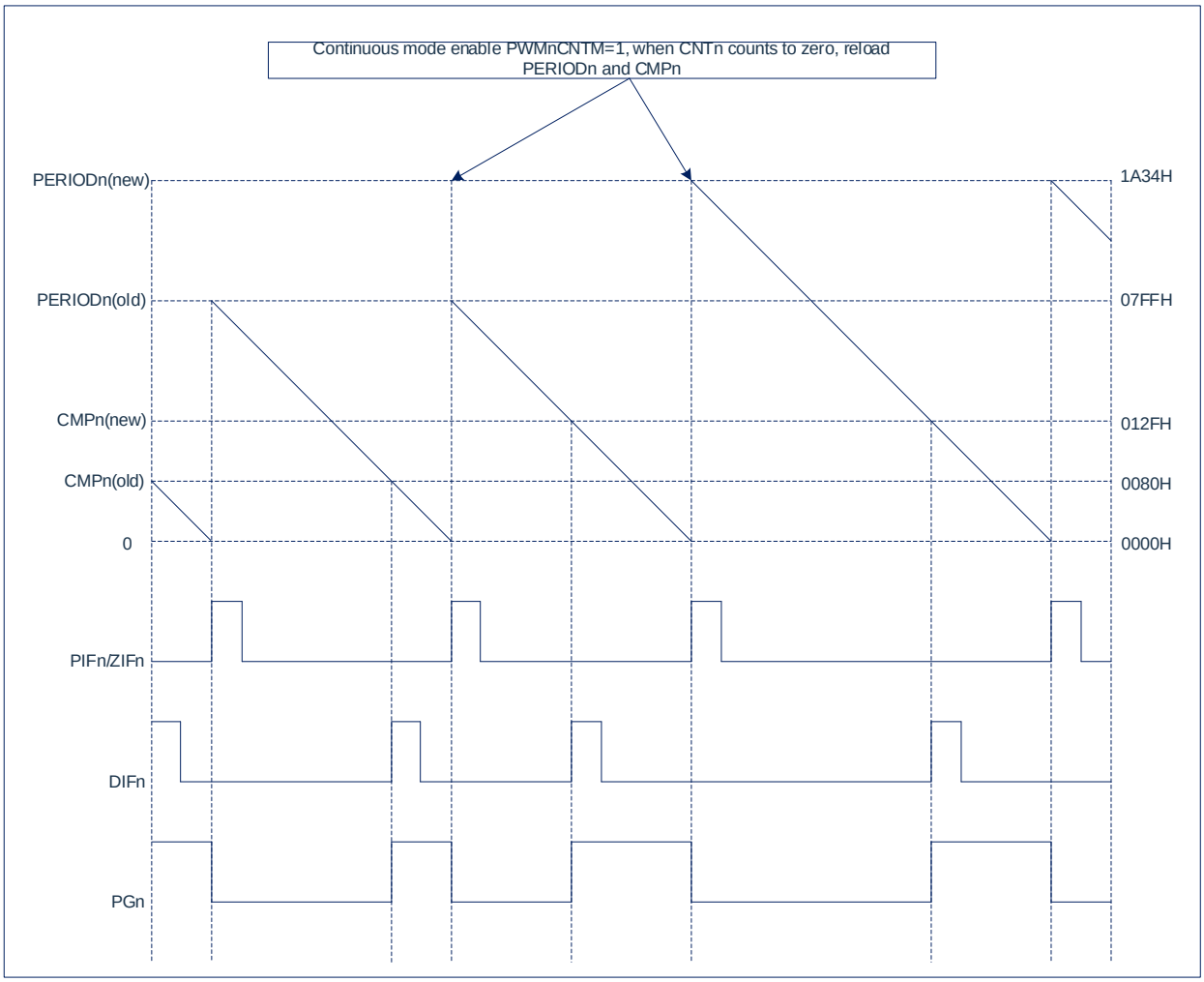
High voltage duration=（*CMPn+1*）×*Tpwm*($CMPn \geq 1$)

Cycle=（*PERIODn+1*）×*Tpwm*

$$\text{duty cycle} = \frac{\text{CMPn+1}}{\text{PERIODn+1}} \quad (\text{CMPn} \geq 1)$$

At CMPn=0, the duty cycle is 0%.

Edge alignment timing is shown in the following figure:



16.4.3 Center alignment

In center-aligned counting mode, both symmetric and asymmetric counting are supported. To enable the asymmetric counting method, the ASYMEN needs to be placed at 1, and the asymmetric counting method can achieve accurate center alignment waveforms.

16.4.3.1 Symmetrical count

In the center-aligned symmetrical counting mode, the 16-bit PWM counter CNTn counts upwards from 0, and when CNTn=CMPn, the PGn output is high, and the PWMnUIF is set to 1; CNTn then continues to count upwards to equal to PERIODn, PWMnPIF sets 1; Then the CNTn starts counting down, and when cnTn=CMPn in the process of counting down, the PGn output is low and the PWMnDIF is set to 1. After that continue counting down to 0, PWMnZIF sets 1.

The relevant parameters of the center alignment symmetry count method are as follows:

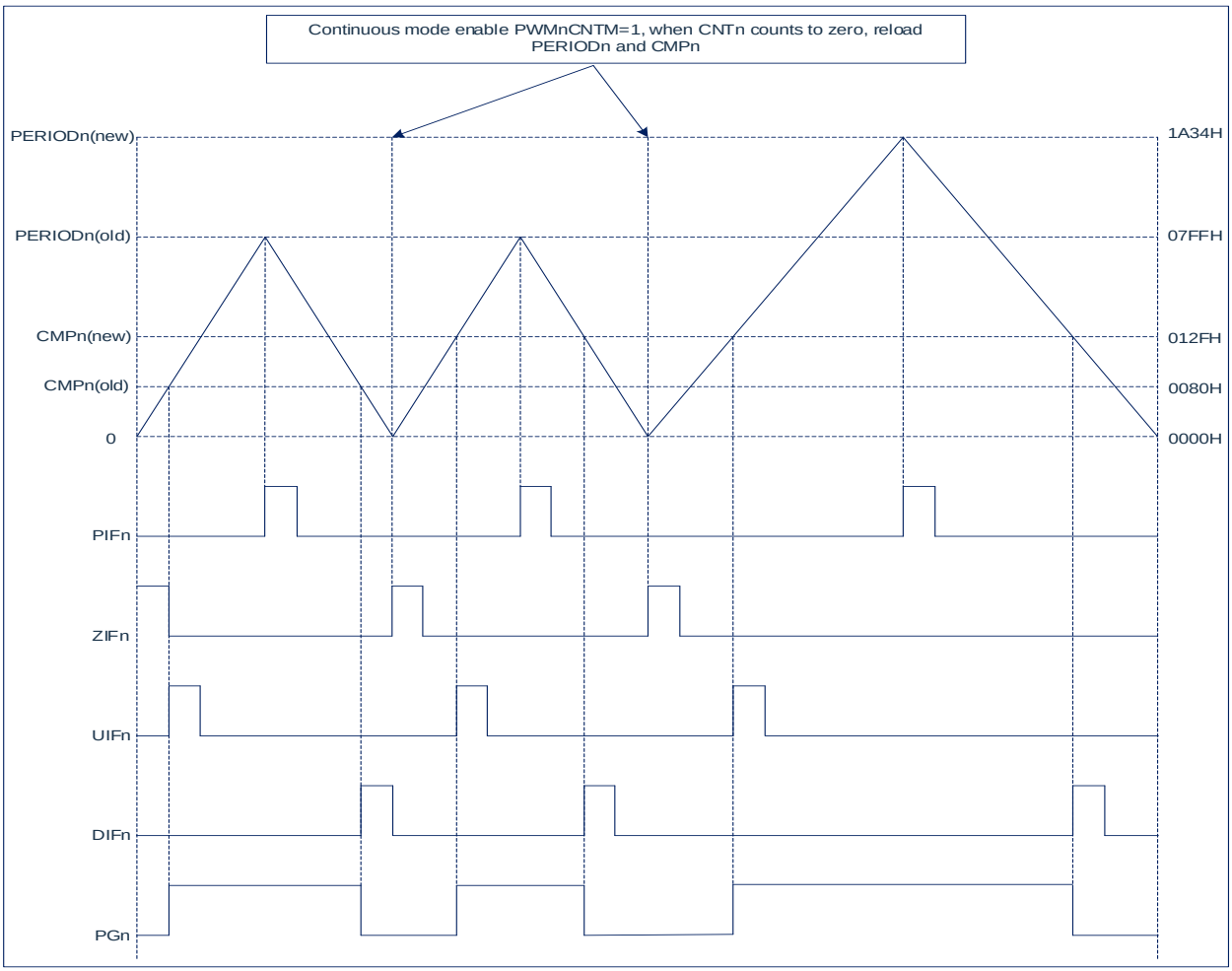
High voltage duration = （*PERIODn×2-CMPn×2-1*）×*Tpwm* ;$(CMPn \geq 1)$

Cycle = （*PERIODn*）×2×*Tpwm* ;

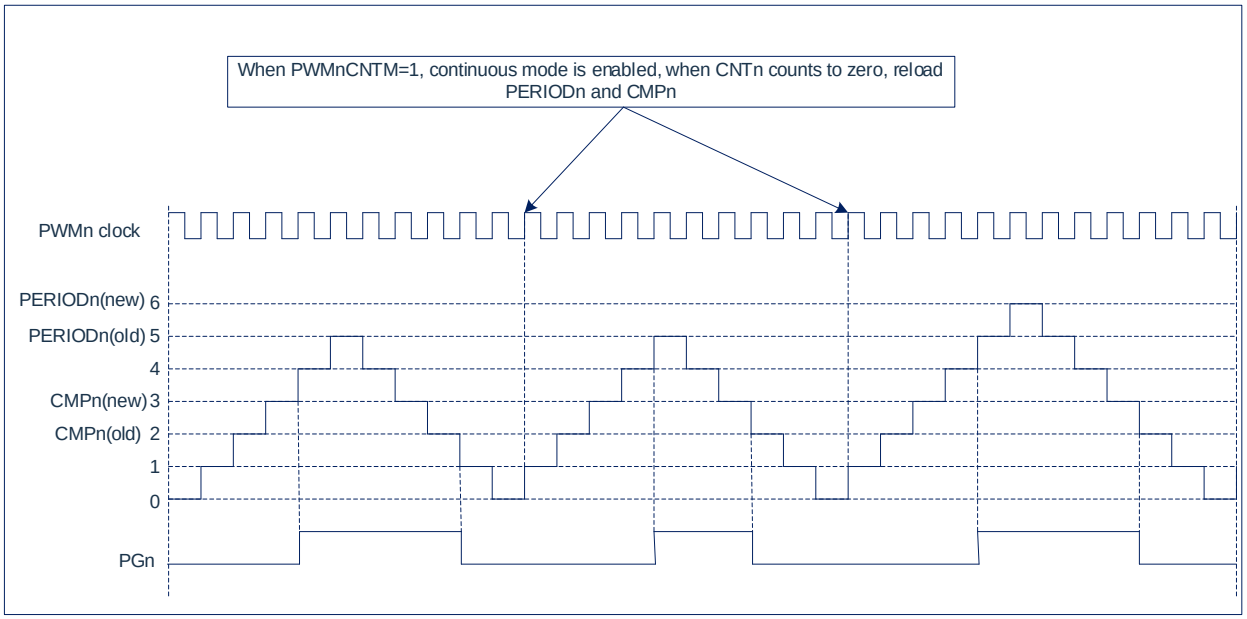
$$\text{Duty Cycle} = \frac{PERIODn \times 2 - CMPn \times 2 - 1}{PERIODn \times 2} ; (CMPn \geq 1)$$

When CMPn=0, the duty cycle is 100%;

Center-aligned symmetrical count timing is shown in the following figure:



The center-aligned counter waveform (symmetrical count) is shown in the following figure:



16.4.3.2 Asymmetric Count

In the center-aligned asymmetric counting mode, the 16-bit PWM counter CNTn counts upwards from 0, and when CNTn=CMPn, the PGn output is high, and the PWMnUIF is set to 1; CNTn then continues to count upwards to equal to PERIODn, PWMnPIF sets 1; Then the CNTn starts counting down, and when cnTn = CMPDn (PWMDDnH/PWMDDnL) during the downward counting process, the PGn output is low and PWMnDIF is set to 1. After that continue counting down to 0, PWMnZIF sets 1.

The relevant parameters for the center-aligned asymmetric count are as follows:

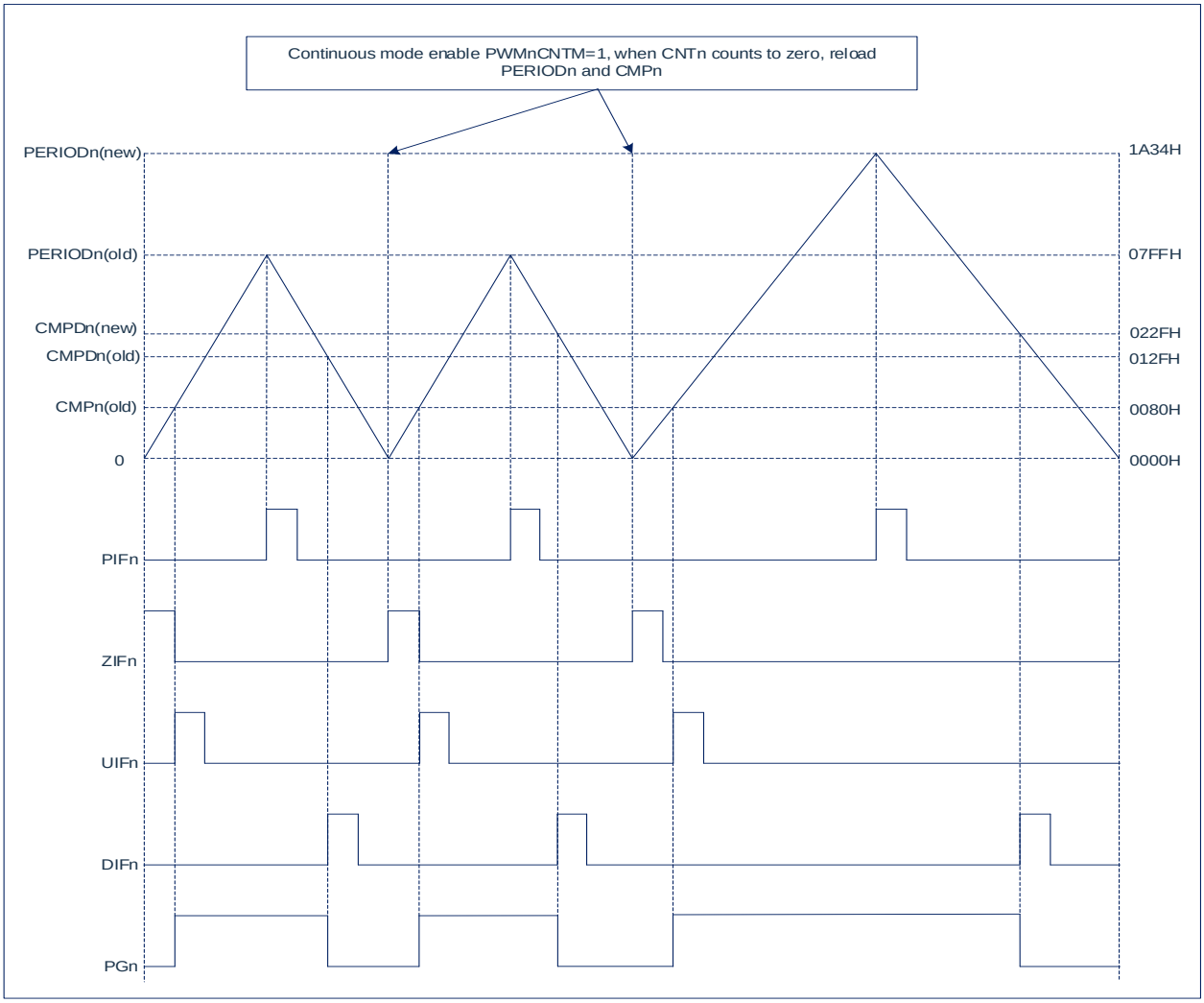
$$\text{High voltage duration} = (PERIODn \times 2 - CMPDn - CMPn - 1) \times Tpwm$$

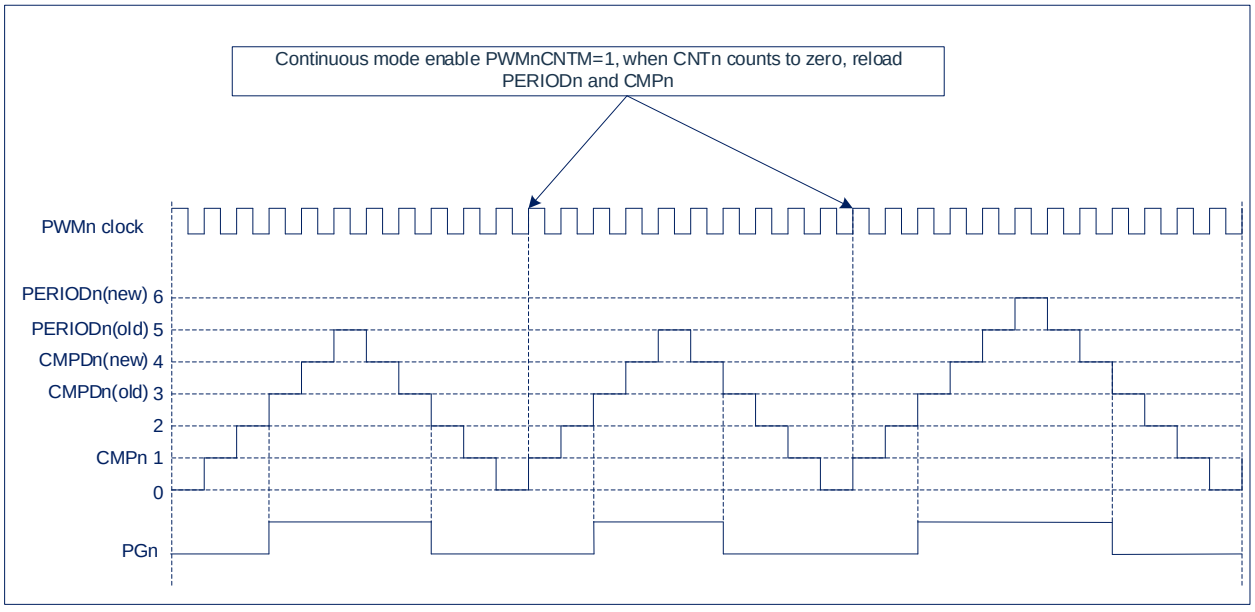
$$\text{Cycle} = (PERIODn) \times 2 \times Tpwm$$

$$\text{Duty Cycle} = \frac{PERIODn \times 2 - CMPDn - CMPn - 1}{PERIODn \times 2}$$

When CMPn=0 and CMPDn=0, the duty cycle is 100%;

Center-aligned asymmetric count timing is shown in the following figure:





16.4.4 Complementary Model

The 4-way PWM can be set up as 2 sets of complementary PWM pairs. In complementary mode, the period, cycle, duty cycle, and clock division control of PG1 and PG3 are determined by PG0, PG2-related registers, respectively, in addition to the corresponding output enable control bit (PWMnOE), PG1, PG1, The PG3 output waveform is no longer controlled by its own registers.

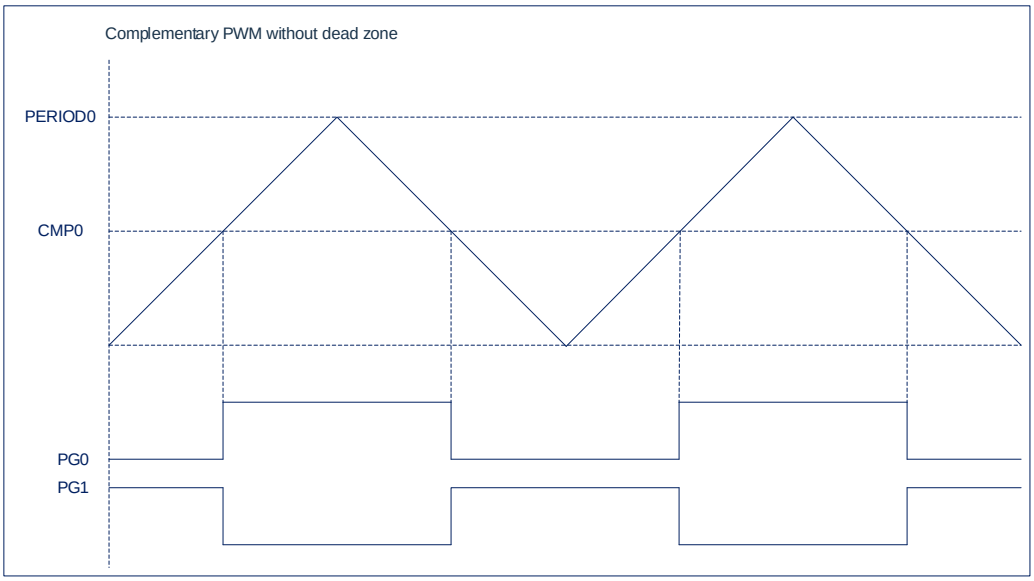
In complementary mode, each set of complementary PWM pairs supports inserting a dead-band delay, and the inserted dead-zone time is as follows:

PWM0/1 Dead-zone: (PWM01DT+1)*T_{PWM0} ;

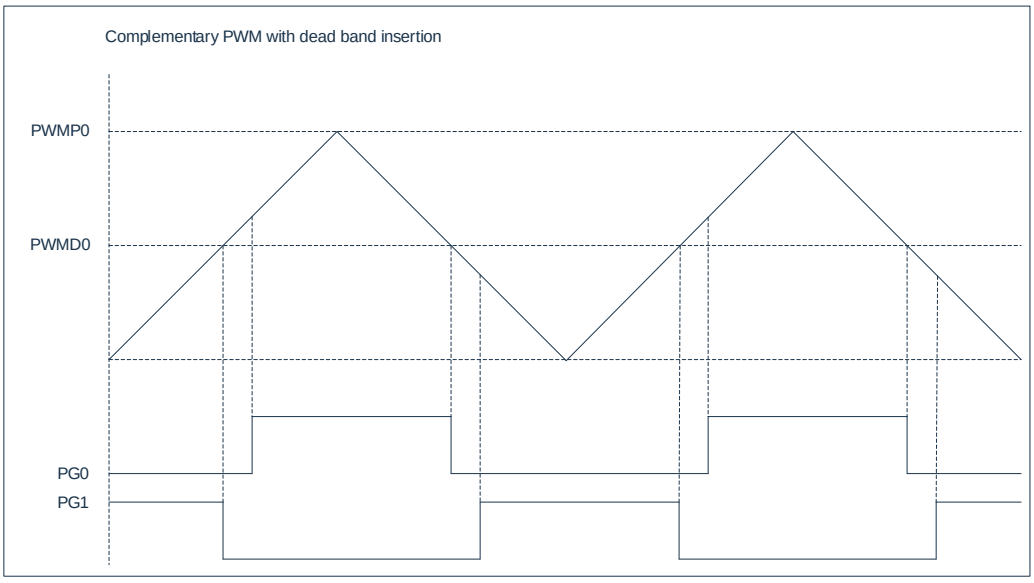
PWM2/3 Dead-zone: (PWM23DT+1)*T_{PWM2} ;

T_{PWM0}/T_{PWM2} are the clock source periods of PG0/PG2, respectively.

Taking PG0/PG1 as an example, the waveform plot without dead zone in complementary mode is shown in the following figure:



TakingPG0/PG1 as an example, the waveform graph with a dead zone in complementary mode is shown in the following figure:



16.4.5 Synchronous Mode

4-way PWM can be set up as 2 sets of synchronous PWM pairs. In synchronous mode, the period, duty cycle and clock division control of PG1 and PG3 are determined by PG0 and PG2 related registers, that is, in addition to the corresponding output enable control bit (PWMnOE), the PG1 and PG3 output waveforms are no longer controlled by their own registers, and the PG1 output wave is similar to PG0 and PG 3 Output waveform pG2.

16.4.6 Mask Output

Enhanced PWM supports masking functionality. PG0-PG 3 Each channel has a separate control, the corresponding mask enable control is set by masking the register PWMMASKE, and the mask output data is set by the mask data register PWMMASKD.

When MASKEn=0, the PGn channel outputs a normal PWM waveform;

When MASKEn=1, the PGn channel outputs the data for M ASKDn.

16.4.7 Brake Function

The enhanced PWM can be configured with a software or hardware trigger braking function via the brake control register PWMFBKC. There are several sources of hardware-triggered brake signals as follows:

- External trigger port FB0 (high/low level);
- External trigger port FB1 (high/low level);
- ADC result comparison output;
- ACMP0 output (high/low level rising edge falling edge);
- ACMP1 output (high/low level rising edge falling edge).

PWM brake related Flag bits:

- Fault flag bit PWMFBF (Software Clear 0).

  After detecting a valid brake trigger source signal, the fault interrupt flag PWMFBF is set to 1 and must be cleared by the software.
- Fault signal flag bit BRKAF (read only).

  The fault signal flag is BRKAF set to 1, and after the brake signal is withdrawn, BRKAF automatically clears 0. BRKAF is read-only.
- Fault protection output status flag bit BRKOSF (read-only).

  BRKOSF=1 indicates the output PWMFBKD data status of the EPWMn channel;

  BRKOSF=0 indicates that EPWMn is the normal output state.

Indicates whether the EPWM output is in braking or normal state. BRKOSF will set 1 when a valid brake signal is detected. In software recovery mode, performing a brake state clear operation (BRKCLR=1) affects the state of that bit.

PWM Brake Recovery Mode:

The fail-safe mode can be divided into 4 types to meet the needs of different fault protection occasions.

Brake mode	Register PWMBRKC[1:0]	Counter status when braking	Recovery conditions				Recovery points
			Undo the brake signal	Clear the brake state	Counter enable	Delay	
			PWMFBKC[6]	PWMBRKC[3]	PWMCNTE	PWMBRKRDT	
Stop mode	00	Stop it	need	need	need	No, you don't	resume
Pause mode	01	Continue counting	need	need	No, you don't	No, you don't	After clearing the brake state, the most recent loading point
Recovery mode	10	Continue counting	need	No, you don't	No, you don't	No, you don't	The most recent load point
Delayed recovery	11	Continue counting	need	No, you don't	No, you don't	need	After the delay time arrives, The most recent load point

Note: After generating brake protection, the EPWMn channel outputs the data in PWMFBKD, and each channel can individually set the output high/low level.

Stop Mode: Generate fault protection and fault interrupt flags, clear the PWMCNTE bit to zero, and stop the counter operation. Restoring the output requires the brake signal to be revoked, and a fault state clear operation is performed (PWMBRKC[3]=1), and then the PWMCNTE bit is set to 1 again.

Pause Mode: Failsafe and fault interrupt flags are generated, but the counter continues to operate. The recovery output requires the brake signal to be revoked, and after performing the fault state clear operation (PWMBRKC[3]=1), the normal output is restored at the most recent load update point.

Recovery Mode: Failsafe and fault interrupt flags are generated, but the counters continue to operate. After the brake signal is revoked, the normal output is automatically restored at the last load update point. Failure state cleanup operations are not required.

Attention needs to be paid to distinguishing whether the brake signal is a pulse signal or a level signal: if the brake source is a level signal, you need to wait for the brake to be undo before you can restore the output; If it is a pulse signal, the EPWM output resumes the output at the last load update point after the brake is triggered, unless the brake pulse signal is generated again during the period.

Delayed recovery mode: Failsafe and fault interrupt flags are generated, but the counters continue to operate. The brake signal is revoked after a delay for a period of time when the EP WM returns to normal output at the most recent load update point. Failure state cleanup operations are not required.

The delay time can be set freely and can be controlled by registers {PWMBRKRDTH[1:0], PWMBRKRDTL [7:0]} (BRKRDT[9:0]). The delay time is as follows:

Tdelay = BRKRDT[9:0]*TCLK

Attention needs to be paid to distinguishing whether the brake signal is a pulse signal or a level signal: if the brake source is a level signal, you need to wait for the brake to be undo before you can restore the output; If it is a pulse signal, the EPWM output waits for the completion delay to resume the output after the last load update point, unless the brake pulse signal is generated again during the period

External trigger ports FB0/FB1 can be configured to trigger PWM brake enable via brake control register PWMFBCK, trigger type (high or low level trigger), and register A via the ADC comparator DCMPC can be configured with ADC comparator result control PWM brake enable, and ACMP0/ACMP1 can be configured via comparator brake control register CNFBCON

Output control PWM brake enable.

When the brake is triggered, the brake flag bit set to 1, the counter enable bits for all channels are hardware cleared, and the PWM outputs preset brake data. There are four brake modes to choose from.

16.5 PWM-related Registers

16.5.1 PWM Control Register PWMCON

F120H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMCON	--	PWMRUN	PWMMODE1	PWMMODE0	GROUPEN	ASYMEN	CNTTYPE	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 -- Reserved, must be 0.
Bit6 PWMRUN: PWM clock pre-division, clock division enable bit;
1= Prohibition (PWMmnPSC, PWMmnDIV are cleared 0);
0= Enable.
Bit5~Bit4 PWMMODE<1:0>: Mode control bit of PWM;
00= Standalone mode;
01= Complementary models;
10= Synchronous mode;
11= Retain.
Bit3 GROUPEN: PWM teaming function enable bit;
1= PG0 controls PG2; PG1 controls PG3;
0= All PWM channel signals are independent of each other.
Bit2 ASYMEN: Asymmetric count enable bits in PWM center alignment;
1= Asymmetric count enable;
0= Symmetry count enables.
Bit1 CNTTYPE: PWM count alignment selection bits;
1= Center alignment;
0= Edge alignment.
Bit0 -- Reserved, must be 0.

16.5.2 PWM Output Enable Control Register PWMOE

F121H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMOE	--	--	--	--	PWM3OE	PWM2OE	PWM1OE	PWM0OE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit4 -- Reserved, must be 0.
Bit3 PWM3OE: Output enable bit of PWM channel 3;
1= Enable;
0= Disable.
Bit2 PWM2OE: Output enable bit of PWM channel 2;
1= Enable;
0= Disable.
Bit1 PWM1OE: Output enable bit of PWM channel 1;
1= Enable;
0= Disable.
Bit0 PWM0OE: Output enable bit of PWM channel 0;
1= Enable;
0= Disable.

16.5.3 PWM0/1 Clock Prescale Control Register PWM01PSC

F123H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM01PSC	PWM01PSC7	PWM01PSC6	PWM01PSC5	PWM01PSC4	PWM01PSC3	PWM01PSC2	PWM01PSC1	PWM01PSC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 PWM01PSC<7:0>: PWM channel 0/1 prescale control bit;
00= The prescaler clock stops, the counter of PWM0/1 stops;
Other = The system clock is divided (PWM01PSC+1).

16.5.4 PWM2/3 Clock Prescale Control Register PWM23PSC

F124H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM23PSC	PWM23PSC7	PWM23PSC6	PWM23PSC5	PWM23PSC4	PWM23PSC3	PWM23PSC2	PWM23PSC1	PWM23PSC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 PWM23PSC<7:0>: PWM channel 2/3 prescale control bit;
00= Prescale clock stop, PWM2/3 counter stop;
Other = (PWM23PSC+1) division of the system clock.

16.5.5 PWM Clock Divider Control Register PWMnDIV (n=0-3).

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMnDIV	--	--	--	--	--	PWMnDIV2	PWMnDIV1	PWMnDIV0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

RegisterSPWMnDIV (n=0-3) Address: F12AH, F12BH, F12CH, F12DH.

Bit7~Bit3 -- Reserved, must be 0.
Bit2~Bit0 PWMnDIV<2:0>: PWM channel n clock divider control bit;
000= Fpwmn-PSC/2;
001= Fpwmn-PSC/4;
010= Fpwmn-PSC/8;
011= Fpwmn-PSC/16;
100= Fpwmn-PSC;
Other = Fsys (system clock);
(PSC is the clock after prescale).

16.5.6 PWM Data Loading Enable Control Register PWMLOADEN

F129H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMLOADEN	--	--	--	--	PWM3LE	PWM2LE	PWM1LE	PWM0LE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit4 -- Reserved, must be 0.

Bit3~Bit0 PWMnLE: Data loading enable bits (n=0-3) of PWM channel n (hardware clearing after loading is completed);

1= Enable load cycle, duty cycle data (PERIODn, CMPn, CMPDn).

0= Writing 0 is invalid.

16.5.7 PWM Output Polarity Control Register PWMPINV

F122H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMPINV	--	--	--	--	PWM3PINV	PWM2PINV	PWM1PINV	PWM0PINV
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit4 -- Reserved, must be 0.

Bit3~Bit0 PWMnPINV: PWM channel n output polarity control bit (n=0-3);

1= Reverse output;

0= Normal output.

16.5.8 PWM Counter Mode Control Register PWMCNTM

F127H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMCNTM	--	--	--	--	PWM3CNTM	PWM2CNTM	PWM1CNTM	PWM0CNTM
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit4 -- Reserved, must be 0.

Bit3~Bit0 PWMnCNTM: PWM channel n counter mode control bit (n=0-3);

1= Auto loading mode;

0= One-shot mode.

16.5.9 PWM Counter Enable Control Register PWMCNTE

F126H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMCNTE	--	--	--	--	PWM3CNTE	PWM2CNTE	PWM1CNTE	PWM0CNTE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit4 -- Reserved, must be 0.

Bit3~Bit0 PWMnCNTE: PWM channel n counter enable control bit (n=0-3);

1= PWMn counter on (PWMn starts output);

0= The PWMn counter stops (the software writes 0 and the counter stops and clears the counter value).

(The brake triggers the bit hardware to clear 0; Single-shot mode completes the bit hardware clearance 0)

16.5.10 PWM Counter Mode Control Register PWMCNTCLR

F128H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMCNTCLR	--	--	--	--	PWM3CNTCLR	PWM2CNTCLR	PWM1CNTCLR	PWM0CNTCLR
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit4 -- Reserved, must be 0.

Bit3~Bit0 PWMnCNTCLR: PWM channel n counter zeroing control bit (n=0-3) (hardware automatic zeroing);

1= PWMn counter clears;

0= Writing 0 is invalid.

16.5.11 PWM Cycle Data Register Low 8 Bits PWMPnL (n=0-3)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMPnL	PWMPnL7	PWMPnL6	PWMPnL5	PWMPnL4	PWMPnL3	PWMPnL2	PWMPnL1	PWMPnL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Registers PWMPnL (n=0-3) Address: F130H, F132H, F134H, F136H.

Bit7~Bit0 PWMPnL<7:0>: The PWM channel n-period data register is 8 bits lower.

16.5.12 PWM Cycle Data Register High 8 Bits PWMPnH (n=0-3)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMPnH	PWMPnH7	PWMPnH6	PWMPnH5	PWMPnH4	PWMPnH3	PWMPnH2	PWMPnH1	PWMPnH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Registers PWMPnH (n=0-3) Address: F131H, F133H, F135H, F137H.

Bit7~Bit0 PWMPnH<7:0>: The PWM channel n-period data register is 8 bits high.

16.5.13 PWM Compare Data Register Low 8 BitS PWMDnL (n=0-3)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMDnL	PWMDnL7	PWMDnL6	PWMDnL5	PWMDnL4	PWMDnL3	PWMDnL2	PWMDnL1	PWMDnL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Registers PWMDnL (n=0-3) Address: F140H, F142H, F144H, F146H.

Bit7~Bit0 PWMDnL<7:0>: PWM channel n compare data (duty cycle data) registers 8 bits lower.

16.5.14 PWM compare 8 bits higher PWMDnH (n=0-3)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMDnH	PWMDnH7	PWMDnH6	PWMDnH5	PWMDnH4	PWMDnH3	PWMDnH2	PWMDnH1	PWMDnH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Registers PWMDnH (n=0-3) Address: F141H, F143H, F145H, F147H.

Bit7~Bit0 PWMDnH<7:0>: The PWM channel n comparison data (duty cycle data) register is 8 bits higher.

16.5.15 PWM down compare data register 8 bits lower PWMDDnL (n=0-3)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMDDnL	PWMDDnL7	PWMDDnL6	PWMDDnL5	PWMDDnL4	PWMDDnL3	PWMDDnL2	PWMDDnL1	PWMDDnL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Registers PWMDDnL (n=0-3) Address: F150H, F152H, F154H, F156H.

Bit7~Bit0 PWMDDnL<7:0>: PWM channel n down to compare data (duty cycle data at asymmetric counts) register 8 bits lower.

16.5.16 PWM down compare data register 8 bits high PWMDDnH (n=0-3)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMDDnH	PWMDDnH7	PWMDDnH6	PWMDDnH5	PWMDDnH4	PWMDDnH3	PWMDDnH2	PWMDDnH1	PWMDDnH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Registers PWMDDnH (n=0-3) Address: F151H, F153H, F155H, F157H.

Bit7~Bit0 PWMDDnH<7:0>: PWM channel n down-compare data (duty cycle data at asymmetric counts) register 8 bits higher.

16.5.17 PWM dead-zone enable control register PWMDTE

F160H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMDTE	--	--	--	--	--	--	PWM23DTE	PWM01DTE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit2 -- Reserved, must be 0.
Bit1 PWM23DTE: PWM2/3 channel dead-zone delay enable bit;
1= Enable;
0= Disable.
Bit0 PWM01DTE: PWM0/1 channel dead-zone delay enable bit;
1= Enable;
0= Disable.

16.5.18 PWM0/1 Dead-zone Delay Data Register PWM01DT

F161H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM01DT	PWM01DT7	PWM01DT6	PWM01DT5	PWM01DT4	PWM01DT3	PWM01DT2	PWM01DT1	PWM01DT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 PWM01DT<7:0>: PWM channel 0/1 dead-zone delay data register.

16.5.19 PWM2/3 Dead-zone Delay Data Register PWM23DT

F162H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM23DT	PWM23DT7	PWM23DT6	PWM23DT5	PWM23DT4	PWM23DT3	PWM23DT2	PWM23DT1	PWM23DT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 PWM23DT<7:0>: PWM channel 2/3 dead-zone delay data register.

16.5.20 PWM Mask Control Register PWMMASKE

F164H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMMASKE	--	--	--	--	PWM3MASKE	PWM2MASKE	PWM1MASKE	PWM0MASKE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit4 -- Reserved, must be 0.
Bit3~Bit0 PWMnMASKE: PWM channel n mask control enable bit (n=0-3);
1= PWMn channel enables masked data output;
0= The PWMn channel disables masking data output (normal output PWM waveform).

16.5.21 PWM Mask Data Register PWMMASKD

F165H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMMASKD	--	--	--	--	PWM3MASKD	PWM2MASKD	PWM1MASKD	PWM0MASKD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit4 -- Reserved, must be 0.

Bit3~Bit0 PWMnMASKD: PWM channel n mask data bits (n=0-3);

1= PWMn channel output is high;

0= PWMn channel output is low.

16.5.22 PWM Brake Control Register PWMFBKC

F166H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMFBKC	PWMFBIE	PWMFBF	BRKAF	PWMFBKSW	PWMFB1ES	PWMFB0ES	PWMFB1EN	PWMFB0EN
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 PWMFBIE: PWM brake break mask bit;

1= Enable interrupts;

0= Disable Interrupt.

Bit6 PWMFBF: PWM brake marker bit (write 0 to clear);

1= Generated brake operation (value of the PWM output brake data register);

0= No brake action is generated.

Bit5 BRKAF: PWM brake signal marker bit;

1= Generated brake or brake signal to remain effective;

0= No brake action is generated.

Bit4 PWMFBKSW: PWM software brake signal start bit;

1= PWM generates software brake signals;

0= Disable.

Bit3 PWMFB1ES: PWM External Hardware Brake Channel (FB1) Trigger Level Selector Bit;

1= High level;

0= Low level.

Bit2 PWMFB0ES: PWM External Hardware Brake Channel (FB0) Trigger Level Selector Bit;

1= High level;

0= Low level.

Bit1 PWMFB1EN: PWM external hardware brake channel (FB1) enable bit;

1= Enable;

0= Disable.

Bit0 PWMFB0IN: PWM external hardware brake channel (FB0) enable bit;

1= Enable;

0= Disable.

16.5.23 PWM Brake Recovery Control Register PWMBRKC

F15CH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMBRKC	BRKOSF	BRKRCS2	BRKRCS21	BRKRCS20	BRKCLR	BRKEN	BRKMS1	BRKMS0
R/W	R	R/W	R/W	R/W	In	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

	BRKOSF:	EPWM fault-protected output status flag bit (read-only).
Bit7	0=	The EPWMn channel is in the normal output state
	1=	The EPWMn channel is the data state of the output PWMnFBKD
Bit6~Bit4	BRKRCS<2:0>:	EPWM failback load point selection bit;
	000=	Load point recovery for EPWM0;
	001=	Load point recovery for EPWM1;
	010=	EPWM2 load point recovery;
	011=	EPWM3 load point recovery;
	Other =	Retain.
Bit3	BRKCLR:	EPWM fail-safe clear bit (write only).
	0=	
	1=	Clear the fail-safe status Note: Only if BRKAF= 0 can you write 1, perform a fault cleanup operation, otherwise the operation is invalid.
Bit2	BRKEN:	EPWM fault protection enable bit
	0=	Disable
	1=	Enable
Bit1~Bit0	BRKMS<1:0>:	
	00=	Stop mode
	01=	Pause mode
	10=	Recovery mode
	11=	Delayed recovery mode

16.5.24 PWM Delayed Recovery Data Register Low 8 Bit PWMBRKRDTL

F15DH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMBRKRDTL	BRKRDT7	BRKRDT6	BRKRDT5	BRKRDT4	BRKRDT3	BRKRDT2	BRKRDT1	BRKRDT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0	BRKRDT <7:0>:	Fail-safe recovery delay data 8 bits lower (delayed recovery mode only works)

16.5.25 PWM Delayed Recovery Data Register High 2 Bits PWMBRKRDTH

F15EH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMBRKRDTH	--						BRKRDT9	BRKRDT8
R/W	R	R	R	R	R	R	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0	BRKRDT <9:8>:	Fail-safe recovery delay data 2 bits higher (delayed recovery mode only works) Delay time = $BRKRDT[9:0] \times TCLK$

16.5.26 PWM Brake Data Register PWMFBKD

F167H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMFBKD	--	--	--	--	PWM3FBKD	PWM2FBKD	PWM1FBKD	PWM0FBKD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit4 -- Reserved, must be 0.

Bit3~Bit0 PWMnFBKD: PWM channel n brake data bits (n=0-3);

1= The PWMn channel produces a high output after braking operation.

0= The PWMn channel produces a low output after braking operation.

16.6 PWM Interrupt

The enhanced PWM has a total of 17 interrupt flags, of which 4 are cyclic interrupt flags, 4 are zero interrupt flags, 4 are up-compare interrupt flags, and 4 are down-compare interrupt flags. 1 brake interrupt flag, the generation of the interrupt flag is independent of whether the corresponding interrupt enable bit is turned on or not. To enable PWM, any type of interrupt requires the global interrupt enable bit (EA=1) and the PWM global interrupt enable bit PWMIE to successfully configure PWM Interrupt function. All interrupts of PWM share an interrupt vector entry, so after entering the interrupt service program, the user can determine which type of interrupt is generated by the interrupt flag.

The interrupt enable and priority of the enhanced PWM can be set by the relevant register bits as follows.

16.6.1 Interrupt Mask Register EIE2

0xAA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIE2	SPIIE	I2CIE	WDTIE	ADCIE	PWMIE	--	ET4	ET3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 SPIIE: SPI interrupt enable bit;
1= Enable SPI interrupts;
0= Disable SPI Interrupt.

Bit6 I2CIE: I2C Interrupt enable bit;
1= Allow I2C interrupts;
0= Disable I2C Interrupt.

Bit5 WDTIE: WDT interrupt enable bit;
1= Enable WDT overflow interrupts;
0= Disable WDT overflow interrupts.

Bit4 ADCIE: ADC interrupt enable bit;
1= Enable ADC interrupts;
0= Disable ADC interrupts.

Bit3 PWMIE: PWM global interrupt enable bit;
1= Enable all PWM interrupts;
0= Disable all PWM interrupts.

Bit2 --: Reserved, must be 0.

Bit1 ET4: Timer4 interrupt enable bit;
1= Enable Timer4 interrupts;
0= Disable Timer4 Interrupt.

Bit0 ET3: Timer3 interrupt enable bit;
1= Enable Timer3 interrupts;
0= Disable Timer3 Interrupt.

16.6.2 Interrupt Priority Control Register EIP2

0xBA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP2	PSPI	PI2C	PWDT	PADC	PPWM	--	PT4	PT3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 PSPI: SPI interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.
Bit6 PI2C: I2C Interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.
Bit5 PWDT: WDT interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.
Bit4 PADC: ADC interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.
Bit3 PPWM: PWM interrupt priority control bit
1= Set to High-level Interrupt;
0= Set to low-level interrupt.
Bit2 -- Reserved, must be 0.
Bit1 PT4: TIMER4 interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.
Bit0 PT3: TIMER3 interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

16.6.3 PWM Cycle Interrupt Shield Register PWMPIE

F168H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMPIE	--	--	--	--	PWM3PIE	PWM2PIE	PWM1PIE	PWM0PIE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit4 -- Reserved, must be 0.
Bit3~Bit0 PWMnPIE: PWM channel n-period interrupt shielding bit (n=0-3);
1= Enable interrupts;
0= Disable Interrupt.

16.6.4 PWM Zero Interrupt Mask Register PWMZIE

F169H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMZIE	--	--	--	--	PWM3ZIE	PWM2ZIE	PWM1ZIE	PWM0ZIE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit4 -- Reserved, must be 0.

Bit3~Bit0 PWMnZIE: PWM channel n zero interrupt shield bit (n=0-3);

1= Enable interrupts;

0= Disable Interrupt.

16.6.5 PWM up Compare Interrupt Mask Registers PWMUIE

F16AH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMUIE	--	--	--	--	PWM3UIE	PWM2UIE	PWM1UIE	PWM0UIE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit4 -- Reserved, must be 0.

Bit3~Bit0 PWMnUIE: PWM channel n up to compare interrupt shield bits (n=0-3);

1= Enable interrupts;

0= Disable Interrupt.

16.6.6 PWM Down Compare Interrupt Mask Registers PWMDIE

F16BH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMDIE	--	--	--	--	PWM3DIE	PWM2DIE	PWM1DIE	PWM0DIE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit4 -- Reserved, must be 0.

Bit3~Bit0 PWMnDIE: PWM channel n down compared interrupt shield bits (n=0-3);

1= Enable interrupts;

0= Disable Interrupt.

16.6.7 PWM Cycle Interrupt Flag Register PWMPIF

F16CH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMPIF	--	--	--	--	PWM3PIF	PWM2PIF	PWM1PIF	PWM0PIF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit4 -- Reserved, must be 0.

Bit3~Bit0 PWMnPIF: PWM channel n-period interrupt flag bit (n=0-3);

1= generate interrupts (software zeroing);

0= No interrupt was generated.

16.6.8 PWM Zero Interrupt Flag Register PWMZIF

F16DH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMZIF	--	--	--	--	PWM3ZIF	PWM2ZIF	PWM1ZIF	PWM0ZIF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit4 -- Reserved, must be 0.

Bit3~Bit0 PWMnZIF: PWM channel n zero interrupt flag bit (n=0-3);

1= generate interrupts (software zeroing);

0= No interrupt was generated.

16.6.9 PWM Up Compare Interrupt Flag Register PWMUIF

F16EH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMUIF	--	--	--	--	PWM3UIF	PWM2UIF	PWM1UIF	PWM0UIF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit4 -- Reserved, must be 0.

Bit3~Bit0 PWMnUIF: PWM channel n up to compare interrupt flag bits (n=0-3);

1= generate interrupts (software zeroing);

0= No interrupt was generated.

16.6.10 PWM Down Compares Interrupt Flag Register PWMDIF

F16FH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMDIF	--	--	--	--	PWM3DIF	PWM2DIF	PWM1DIF	PWM0DIF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit4 -- Reserved, must be 0.

Bit3~Bit0 PWMnDIF: PWM channel n compares the interrupt flag bits downwards (n=0-3);

1= generate interrupts (software zeroing);

0= No interrupt was generated.

17. LCD Driver

17.1 Overview

With the ability to drive external LCD panels, the COM and SEG pins of the LCD drive are shared with the IO, and the LCD enable control signal is implemented by the software configuration.

17.2 Characteristic

LCD drives have the following characteristics:

- Maximum supported LCD channels: 24COM, 24SEG
- Bias voltage selectable: 1/3.
- Contains two types of Frames.
- "1" in the figure below represents the lit LCD pixel; The signal polarity of the pins is "0" or "1" and is generated by the corresponding I/O common pin data bits.

17.3 LCD Operating Instructions

The waveform cycle of a complete LCD contains two Frames, Namely Frame0 andFrame1.

17.3.1 1/3 Bias Register Operation

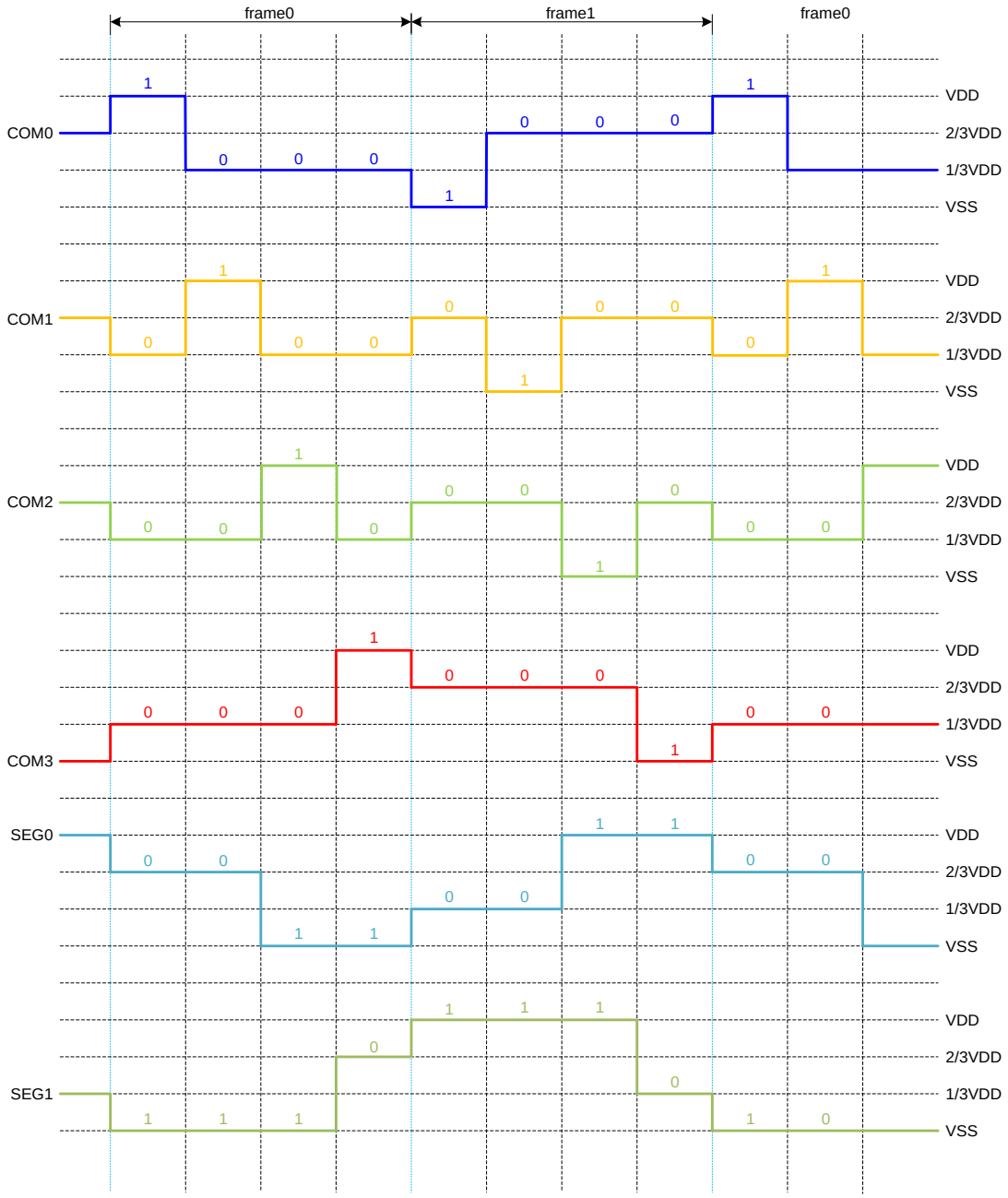
Register operations related to 1/3 Bias of registers are shown in the following table:

Pins	Frame0	Frame1
WITH	Pin output VDD; FRAME = 0; LCD_Sx = 0; LCD_ENx= 1; Pxn = 1;	Pin output (2/3)*VDD; FRAME = 1; LCD_Sx = 0; LCD_ENx= 1; Pxn = 0;
	Pin output (1/3)*VDD; FRAME = 0; LCD_Sx = 0; LCD_ENx= 1; Pxn = 0;	Pin output GND; FRAME = 1; LCD_Sx = 0; LCD_ENx= 1; Pxn = 1;
ITSELF	Pin output (2/3)*VDD; FRAME = 0; LCD_Sx = 1; LCD_ENx= 1; Pxn = 0;	Pin output VDD; FRAME = 1; LCD_Sx = 1; LCD_ENx= 1; Pxn = 1;
	Pin output GND; FRAME = 0; LCD_Sx = 1; LCD_ENx= 1; Pxn = 1;	Pin output (1/3)*VDD; FRAME = 1; LCD_Sx = 1; LCD_ENx= 1; Pxn = 0;

The shaded part of the table represents the operation of the corresponding SEG/COM for the LCD that needs to be lit in the two Frames.

17.3.2 1/3 Bias Timing Diagram

The "1" in the figure represents the lit LCD pixels



17.4 Related Registers

17.4.1 LCD Control Register LCDCON0

F650H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCDCON0	LCDEN	FRAME	--	--	--	--	ISEL1	ISEL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 LCDEN LCD enable bit
1= Enables the LCD module
0= LCD modules are Disabled

Bit6 FRAME Frame0 or Frame
0= Frame 0
1= Frame 1

Bit5~Bit2 --

Bit1~Bit0 LOW[1:0] LCD current select bit
00= 8.3uA
01= 16.7uA
10= 50uA
11= 100uA

17.4.2 LCD COM/SEG Select Register LCD_S0

F651H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCD_S0	LCD_S07	LCD_S06	LCD_S05	LCD_S04	LCD_S03	LCD_S02	LCD_S01	LCD_S00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 LCD_S0[7:0] COM/SEG[7:0] function selection for LCD
1= The corresponding pin is used as a LCD_SEG port
0= The corresponding pin is used as a LCD_COM port

17.4.3 LCD COM/SEG Select Register LCD_S1

F652H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCD_S1	LCD_S15	LCD_S14	LCD_S13	LCD_S12	LCD_S11	LCD_S10	LCD_S9	LCD_S8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 LCD_S1[7:0] LCD COM/SEG[15:8] function selection
1= The corresponding pin is used as a LCD_SEG port
0= The corresponding pin is used as a LCD_COM port

17.4.4 LCD COM/SEG Select Register LCD_S2

F653H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCD_S2	LCD_S23	LCD_S22	LCD_S21	LCD_S20	LCD_S19	LCD_S18	LCD_S17	LCD_S16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 LCD_S2[7:0] LCD's COM/SEG[23:16] feature selection

1= The corresponding pin is used as a LCD_SEG port

0= The corresponding pin is used as a LCD_COM port

17.4.5 LCD Function Select Register LCDEN0

F654H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCDEN0	LCDEN07	LCDEN06	LCDEN05	LCDEN04	LCDEN03	LCDEN02	LCDEN01	LCDEN00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 LCDEN0[7:0] LCD corresponding SEG/COM pin function selection

1= LCD SEG/COM function

0= Other pin functions

17.4.6 LCD Function Select Register LCDEN1

F6554H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCDEN1	LCDEN15	LCDEN14	LCDEN13	LCDEN12	LCDEN11	LCDEN10	LCDEN9	LCDEN8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 LCDEN1[7:0] LCD corresponding SEG/COM pin function selection

1= LCD SEG/COM function

0= Other pin functions

17.4.7 LCD Function Select Register LCDEN2

F655H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCDEN2	LCDEN23	LCDEN22	LCDEN21	LCDEN20	LCDEN19	LCDEN18	LCDEN17	LCDEN16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 LCDEN2[7:0] LCD corresponding SEG/COM pin function selection

1= LCD SEG/COM function

0= Other pin functions

18. LED Driver

18.1 Overview

Software driver LED, can be convenient for users to achieve LED display driver.

18.2 Characteristic

LED drivers have the following characteristics:

- Supports up to 24 COM ports and 24 SEG ports.
- COM/SEG port current drive is optional.
- COM port current 50 mA, 150 mA selectable (V_{OL} = 1.5V@VDD = 5 V).
- The SEG port current is available in four gears, with a maximum current of up to 45mA (V_{OH}=3.5V@VDD=5V).

18.3 Related Registers

18.3.1 SEG Port P00-P03 Drive Current Control Register LEDSDRP0L

F710H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDSDRP0L	--	--	--	--	--	--	LEDSDRP0L1	LEDSDRP0L0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	1	1

	LEDSDRP0L<1:0>	P00, P01, P02, P03 pull current drive
	00=	8.9mA
Bit1~Bit0	01=	17.8mA
	10=	26.8mA
	11=	45mA

18.3.2 SEG Port P04-P07 Drive Current Control Register LEDSDRP0H

F711H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDSDRP0H	--	--	--	--	--	--	LEDSDRP0H1	LEDSDRP0H0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	1	1

	LEDSDRP0H <1:0>	P04, P05, P06, P07 pull current drive
	00=	8.9mA
Bit1~Bit0	01=	17.8mA
	10=	26.8mA
	11=	45mA

18.3.3 SEG Port P10-P13 Drive Current Control Register LEDSDRP1L

F712H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDSDRP1L	--	--	--	--	--	--	LEDSDRP1L1	LEDSDRP1L0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	1	1

Bit1~Bit0 LEDSDRP1L <1:0> P10, P11, P12, P13 pull current drive

00= 8.9mA

01= 17.8mA

10= 26.8mA

11= 45mA

18.3.4 SEG Port P14-P17 Drive Current Control Register LEDSDRP1H

F713H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDSDRP1H	--	--	--	--	--	--	LEDSDRP1H1	LEDSDRP1H0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	1	1

Bit1~Bit0 LEDSDRP1H <1:0> P14, P15, P16, P17 pull current drive

00= 8.9mA

01= 17.8mA

10= 26.8mA

11= 45mA

18.3.5 SEG Port P20-P23 Drive Current Control Register LEDSDRP2L

F714H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDSDRP2L	--	--	--	--	--	--	LEDSDRP2L1	LEDSDRP2L0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	1	1

Bit1~Bit0 LEDSDRP2L <1:0> P20, P21, P22, P23 pull current drive

00= 8.9mA

01= 17.8mA

10= 26.8mA

11= 45mA

18.3.6 SEG Port P24-P27 Drive Current Control Register LEDSDRP2H

F715H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDSDRP2H	--	--	--	--	--	--	LEDSDRP2H1	LEDSDRP2H0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	1	1

Bit1~Bit0 LEDSDRP2H <1:0> P24, P25, P26, P27 pull current drive

00= 8.9mA

01= 17.8mA

10= 26.8mA

11= 45mA

18.3.7 SEG Port P30-P33 Drive Current Control Register LEDSDRP3L

F716H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDSDRP3L	--	--	--	--	--	--	LEDSDRP3L1	LEDSDRP3L0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	1	1

Bit1~Bit0 LEDSDRP3L <1:0> P30, P31, P32, P33 pull current drive

00= 8.9mA

01= 17.8mA

10= 26.8mA

11= 45mA

18.3.8 LED COM Port Sink Current Selection Register PnDR (n=0/1/2/3).

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PnDR	PnDR7	PnDR6	PnDR5	PnDR4	PnDR3	PnDR2	PnDR1	PnDR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

P0DR : F00CH P1DR:F01CH P2DR:F02CH P3DR:F03CH

Bit7 PnDR7 Pn7 drive current selection
1= 150mA
0= 50mA

Bit6 PnDR6 Pn6 drive current selection
1= 150mA
0= 50mA

Bit5 PnDR5 Pn5 drive current selection
1= 150mA
0= 50mA

Bit4 PnDR4 Pn4 drive current selection
1= 150mA
0= 50mA

Bit3 PnDR3 Pn3 drive current selection
1= 150mA
0= 50mA

Bit2 PnDR2 Pn2 drive current selection
1= 150mA
0= 50mA

Bit1 PnDR1 Pn1 drive current selection
1= 150mA
0= 50mA

Bit0 PnDR0 Pn0 drive current selection
1= 150mA
0= 50mA

19. SPI Module

19.1 Overview

This SPI is a fully configurable SPI master/slave device that allows the user to configure the polarity and phase of the serial clock signal SCLK. The serial clock line (SCLK) is synchronized with the shifting and sampling of information on two independent serial data lines, and the SPI data is sent and received simultaneously. SPI allows the MCU to communicate with serial peripherals, it is also capable of interprocessor-to-processor communication in multi-host systems, and is a technology-independent design that can be implemented in a variety of process technologies.

The SPI system is flexible enough to connect directly with many standard product peripherals from several manufacturers. To accommodate most of the available synchronous serial peripherals, clock control logic allows the selection of clock polarity and phase. The system can be configured as a master device or slave device, and when the SPI is configured as a host device, the software chooses one of eight different bit rates for the serial clock, up to the system clock divided by 4 (Fsys/4).

The SPI slave chip selects an addressable SPI slave device to exchange serial data. When the SPI is used as the host device, the SPI auto drive is selected by the slave selection control register SSCR. The SPI controller includes logic error detection to support interprocessor communication, such as a write conflict detector that indicates when data is written to the serial shift register during transfer.

SPI has the following features:

- Full-duplex synchronous serial data transfer.
- Supports master/slave mode.
- Support for multi-host systems.
- System error detection.
- Interrupt generation.
- Supports speeds up to 1/4 of the system clock (F_{SYS}≤24MHz).
- The bit rate produces 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512 of the system clock.
- Four transmission formats are supported.
- The simple interface allows easy connection to the microcontroller.

19.2 SPI Port Configuration

Using the SPI function requires configuring the relevant port as an SPI channel and selecting the corresponding port input through the communication input port registers. For example, P04, P02, P03, P12 are configured as SPI communication ports. The configuration code is as follows:

PS_SCLK = 0x04; Select P04 for the SCLK channel for the SPI

PS_MOSI = 0x02; Select P02 for the MOSI channel for the SPI

PS_MISO = 0x03; Select P03 as the MISO channel for the SPI

PS_NSS = 0x12; Select P12 for the NSS channel for the SPI

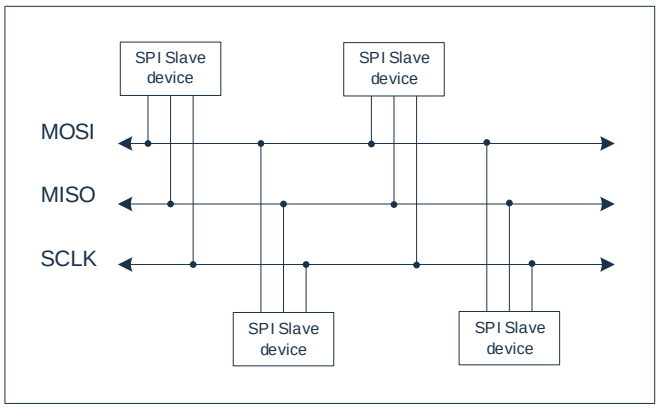
P04CFG = 0x02; P04 multiplexing for SCLK function

P02CFG = 0x02; P02 multiplexing for MOSI function

P03CFG = 0x02; P03 multiplexing for MISO function

P12CFG = 0x02; P12 multiplexing for NSS functions

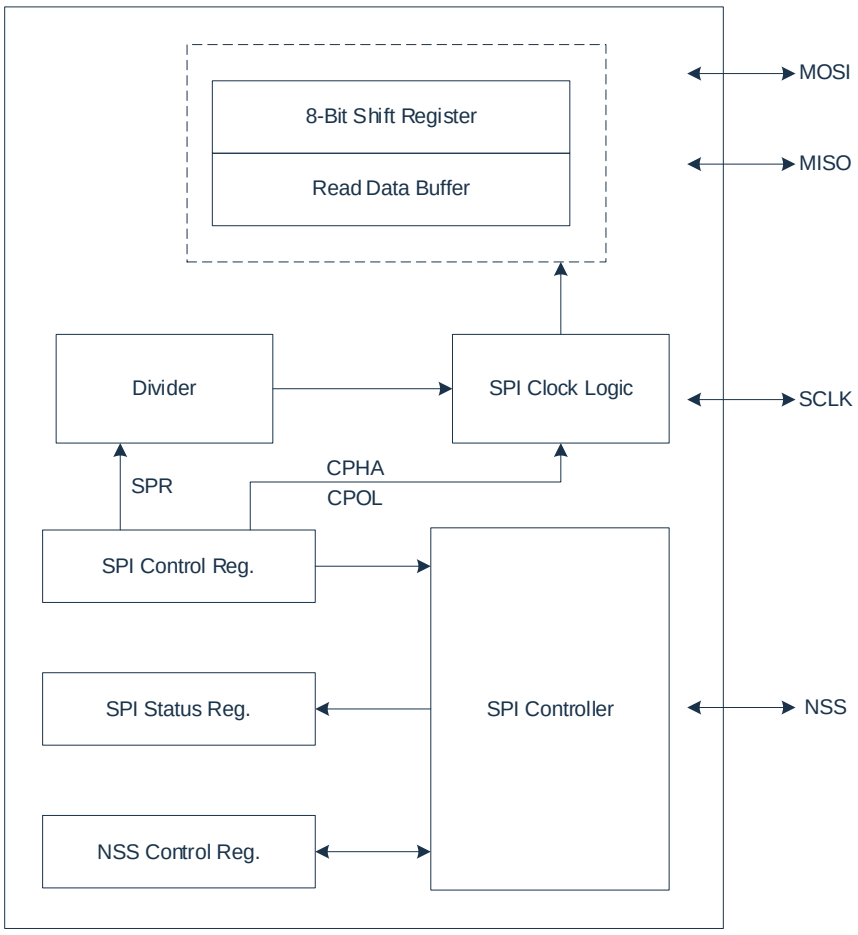
Configured as SCLK, MOSI, MISO, and NSS ports, its pull-up resistor and the open-drain output are forced off. The schematic diagram of the multi-slave SPI communication structure is shown in the following figure:



19.3 SPI Hardware Description

When an SPI transfer occurs, when one data pin moves out of one 8-bit character, the other data pin moves in the other 8-bit character. The 8-bit shift register in the master device and another 8-bit shift register in the slave device are connected as a cyclic 16-bit shift register, and when the transfer occurs, the distributed shift register is shifted by 8 bits, thus effectively swapping the characters of the master slave.

The central element in the SPI system is the module containing the shift registers and the buffer for reading data. The system is single buffer in the transmit direction and double buffer in the receive direction. This means that new data cannot be written to the shifter until the previous data transfer is complete; However, the received data is transmitted to a parallel read data buffer, so the shifter is free to receive a second serial character. As long as the first character is read out of the read data buffer before the next serial character is ready for transmission, there is no overwrite. The SPI control block diagram is shown in the following figure:



The pins associated with SPI are: NSS, SCLK, MOSI, MISO.

The NSS output pins in master mode are used to select slave devices, and the NSS input pins in slave mode are used to enable transmission.

In host mode, the SCLK pin is used as an SPI clock signal reference. When the host device initiates a transfer, eight clock cycles are automatically generated on the SCLK pins.

When the SPI is configured as a slave device, the SI pin is the slave device input data line and the SO is the slave device output data line.

When the SPI is configured as a host device, the MI pin is the host device input data line and the MO is the host device output data line.

19.4 SPI-related Registers

19.4.1 SPI Control Register SPCR

0xEC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPCR	--	SPEN	SPR2	MSTR	CPOL	CPHA	SPR1	SPR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	1	0	0

Bit7	--	Reserved, must be 0.
Bit6	SPEN:	SPI module enable bit;
	1=	Enable;
	0=	Disable.
Bit5	SPR2:	The SPI clock frequency selects bit [2].
Bit4	MSTR:	SPI mode select bit;
	1=	Master mode;
	0=	Slave mode.
Bit3	CPOL:	SPI clock polarity select bit;
	1=	SCLK is high when idle;
	0=	SCLK is low when idle.
Bit2	CPHA:	SPI clock phase select bit.
Bit1~Bit0	SPR<1:0>:	SPI clock frequency select bit [1:0].
		(For details of frequency control, see the table below)

The SPR2-SPR0 controls the SPI clock divider

SPR2	SPR1	SPR0	System clock divider
0	0	0	4
0	0	1	8
0	1	0	16
0	1	1	32
1	0	0	64
1	0	1	128
1	1	0	256
1	1	1	512

19.4.2 SPI Data Register SPDR

0xEE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPDR	SPIDATA7	SPIDATA6	SPIDATA5	SPIDATA4	SPIDATA3	SPIDATA2	SPIDATA1	SPIDATA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0	SPIDATA<7:0>:	Data sent or received by SPI.
	Write operation:	Write the data that will be sent (the order of sending is from high bit to low bit).
	Read operation:	Data that has been received.

19.4.3 SPI Device Select Control Register SSCR

The slave device selection control register SSCR can read or write at any time and is used to configure which slave selection output should be driven when acknowledging the SPI master transfer. When the SPI host transfer starts, the contents of the SSCR register are automatically assigned to the NSS pin.

0xEF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SSCR	--	--	--	--	--	--	NSSO1	NSSO0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	1	1	1	1	1	1	1	1

Bit7~Bit2 -- Reserved, must be 1.

Bit1~Bit0 NSSOx: The SPI selects the control bits from the device (the host chip select output NSS is NSSOx, x=0-1).

0= When the SPI host transfer starts, NSSOx outputs 0.

1= When the SPI host transfer starts, NSSOx outputs 1.

19.4.4 SPI Status Register SPSR

0xED	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPSR	SPISIF	WCOL	--	--	--	--	--	SSCEN
R/W	R	R	--	R	--	--	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 SPISIF: SPI transmission completion interrupt flag bit, read-only;

1= SPI transmission is completed (read SPSR first, then read/write SPDR and then clear zero);

0= The SPI was not transmitted.

Bit6 WCOL: SPI write violation interrupt flag bit, read-only;

1= When the SPI transfer is not completed, a collision of the write SPDR operation occurs (read the SPSR first, then clear the SPDR after reading/writing the SPDR);

0= No write conflicts.

Bit5~Bit1 -- Reserved, must be 0.

Bit0 SSCEN: SPI master mode NSS output control bit.

1= When the SPI is idle, the NSS output is high;

0= NSS output registers the contents of the SSCR.

The SPI Status Register (SPSR) contains flags that indicate that the transfer was complete or that a system error occurred. When the corresponding event occurs and is cleared sequentially by the software, all flags are set automatically. By reading spsr and then accessing spdr, SPISIF and WCOL will be automatically cleared.

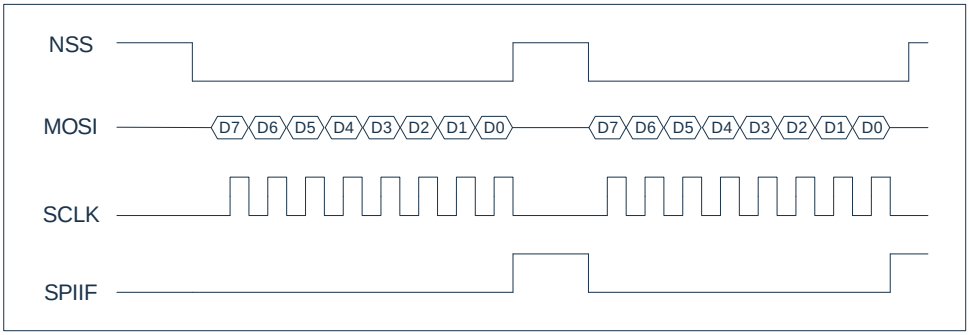
The SSCEN bit is the enable bit of the automatic slave selection output. When SSCEN is set to 1, the NSS line outputs the contents of the SSCR register while the transmission is in progress, and the NSS is high when the transmission is idle. When the SSCEN bit is cleared, the NSS line always displays the contents of the SSCR registers.

19.5 SPI Master Mode

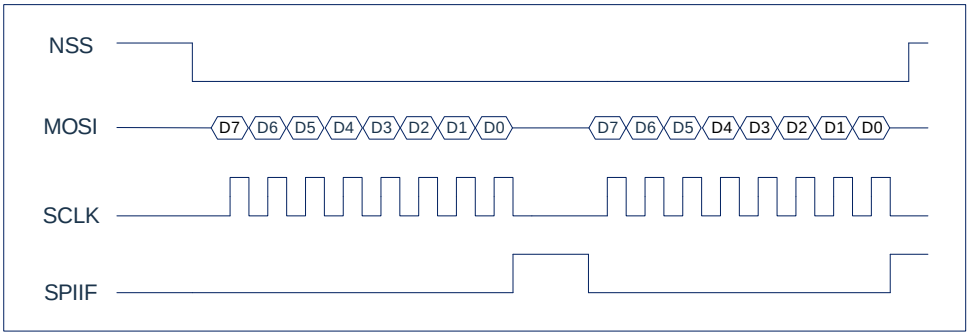
When SPI is configured for host mode, the transfer is initiated by writing to the SPDR registers. When new bytes are written to the SPDR register, the SPI starts transferring. The serial clock SCLK is generated by the SPI, in host mode the SPI is enabled, and the SCLK output.

SPI in master mode can select the SPI slave device via the NSS cable. NSS Line - Slave Selection Output Line loads the contents of the SSCR registers. The SSCEN bit of the SPSR register is selected between automatic NSS line control and software control. Place SSCEN in host mode, when SSCEN is set to 1, the NSS line outputs the contents of the SSCR register while the transmission is in progress, and the NSS is high when the transmission is idle. When the SSCEN bit is cleared, the NSS cable is controlled by the software and always displays the contents of the SSCR registers, regardless of whether the transmission is in progress or the SPI is idle.

When SSCEN=1, configure the clock polarity CPOL=0 and the clock phase CPHA=0 for SPI, as shown in the following figure



When SSCEN=0, configure the clock polarity CPOL=0 and the clock phase CPHA=0 for SPI, as shown in the following figure:



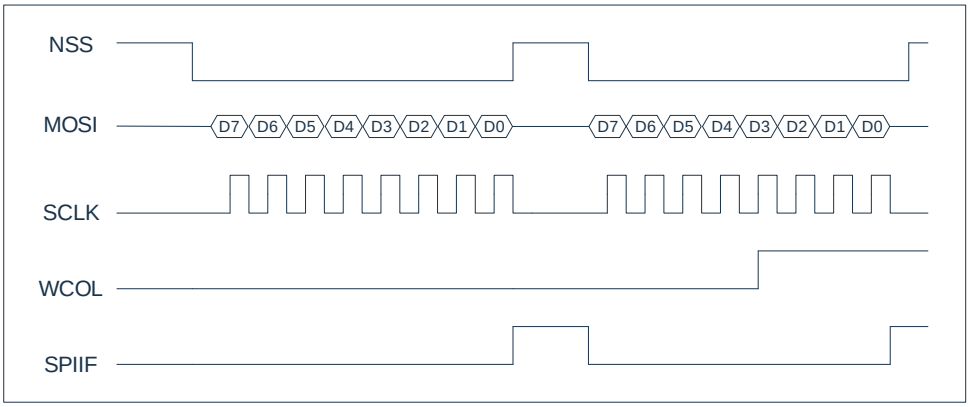
19.5.1 Write Conflict Error

If the SPI data registers are written during the transfer, a write violation occurs. The transfer continues uninterrupted, and the write data that causes the error is not written to the shifter. Write conflicts are indicated by the WCOL flag in the SPSR register.

When a WCOL error occurs, the WCOL flag is automatically set to 1 by the hardware. To clear the WCOL bits, the user should perform the following steps:

- Read the contents of the SPSR register;
- Access the SPDR register (read or write).

In the SPI master mode, the write conflict error when configuring the clock polarity CPOL=0 and clock phase CPHA=0 of the SPI is shown in the following figure:



The specific conditions for the occurrence of write conflicts are: during the data transmission process, when the NSS is low, the first data starts to be sent from the moment to the 8th SCLK falling edge, if you write SPDR during this period, a write conflict will occur, and the WCOL will be set to 1.

Note: When you start sending data, after writing spdr, the NSS does not immediately go low, and you need to wait for at most one SPI clock before it starts to be low. After the NSS is low, it is necessary to wait for a system clock to start sending the first data before entering the real data transfer state. Between writing the SPDR to the time it enters the real data transfer state, writing the SPDR again does not create a write conflict. However, the operation updates the data that is ready to be sent. If there are multiple writes to the SPDR, the data sent will be the last value written to the SPDR.

Since SPI has only one transmit buffer, it is recommended to determine whether the last data was sent before writing the SPDR, and then write the SPDR register after the transmission is completed to prevent write conflicts.

19.6 SPI Slave Mode

When configured as an SPI slave device, SPI transmission is initiated by an external SPI master module by using the SPI slave selection input and generates an SCLK serial clock.

Before the transfer begins, it is necessary to determine which SPI slave will be used to exchange data. The NSS is used (clear = 0), and the clock signal connected to the SCLK line will transfer the SPI slave device to the receiving shift register contents of the MOSI line and drive the MISO line with the contents of the transmitter shift registers. When all 8 bits are moved in/out, SPI generates an interrupt request by setting the IRQ output. The contents of the shift register drive the MISO line.

In SPI slave mode, there can only be one transmit error - write conflict error.

19.6.1 Address Error

In slave mode, only write conflict errors can be detected by SPI.

When an SPDR register write operation is performed while an SPI transfer is in progress, a write violation error occurs.

In slave mode, when CPHA is cleared, a write collision error may occur as long as the NSS slave selection line is driven low, even if all bits have been transmitted. This is because the transfer start is not explicitly specified, and the NSS is driven low after a full-byte transfer may indicate the start of the next byte transfer.

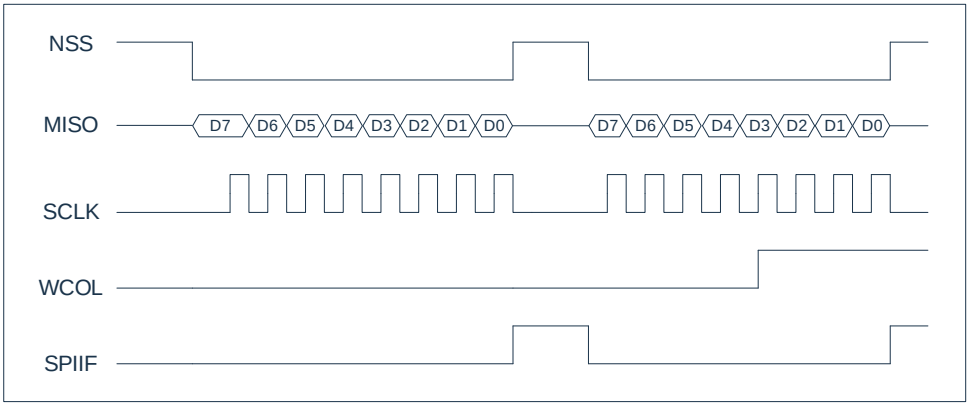
19.6.2 Write Conflict Error

If the SPI data register is written during the transfer, a write violation occurs. The transfer continues uninterrupted, and the write data that causes the error is not written to the shifter. Write conflicts are indicated by the WCOL flag in the SPSR register.

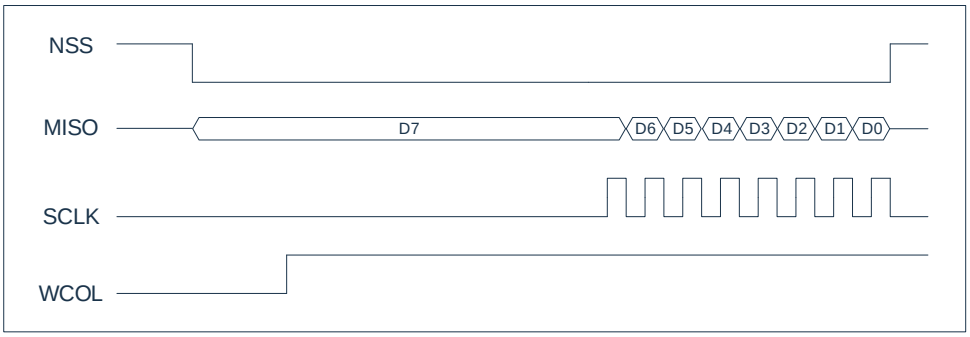
When a WCOL error occurs, the WCOL flag is automatically set to 1 by the hardware. To clear the WCOL bits, the user should perform the following sequence:

- Read the contents of the SPSR register;
- Access the SPDR register (read or write).

The write violation error during transfer in SPI slave mode is shown in the following figure:



In case the CPHA is cleared, WCOL generation can also be caused by SPDR register writes when either NSS line is cleared, at which point the SPI host can also complete without generating a serial clock, SCLK. This is because the transfer start is not explicitly specified, and the NSS is driven low after a full-byte transfer may indicate the start of the next byte transfer. When the NSS transmission line is low and the clock phase CPHA = 0, writing SPDR causes a write collision error as shown in the following figure:



In addition, in slave mode, after writing the SPDR, the host-controlled NSS does not immediately become low. When the NSS is low, you need to wait for the second edge of the SCLK to start before entering the real data transfer state.

Between writing the SPDR and starting to send the first data, writing the SPDR again does not create a write conflict. However, the operation updates the data that is ready to be sent. If there are multiple writes to the SPDR, the data sent will be the last value written to the SPDR.

Writing the SPDR again does not create a write conflict during the start of sending the first data to the second edge of the SCLK, nor does it update the data being sent. That is, the operation of writing the SPDR is ignored.

Since SPI has only one transmit buffer, it is recommended to determine whether the last data was sent before writing the SPDR, and then write the SPDR register after the transmission is completed to prevent write conflicts.

19.7 SPI Clock Control Logic

19.7.1 SPI Clock Phase and Polarity Control

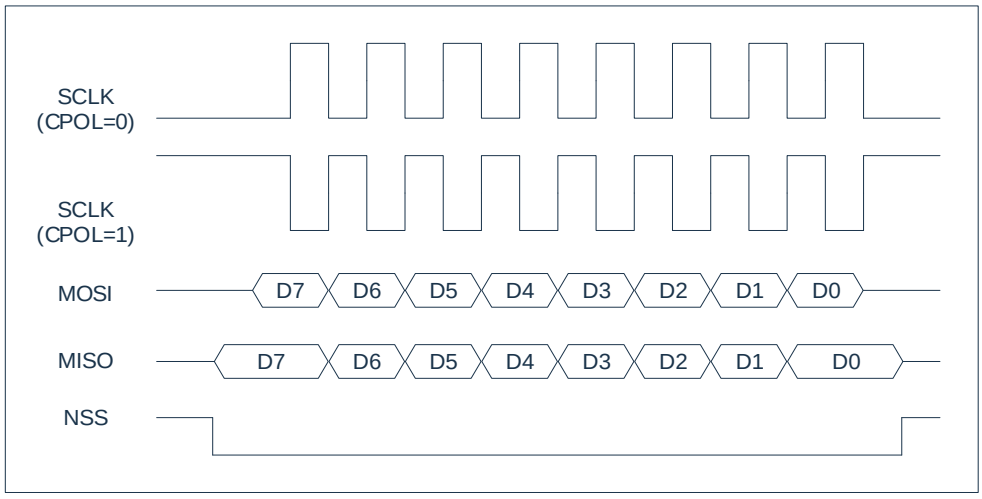
The software can select any of the four combinations using two control bits (phase and polarity of the serial clock SCLK) in the SPI control register (SPCR). Clock polarity is specified by the CPOL control bit, and the CPOL control bit selection high or low level when the transmission is idle has no significant effect on the transmission format. The Clock Phase (CPHA) control bit selects one of two largely different transmission formats. The clock phase and polarity of the master SPI device and the communication slave device should be the same. In some cases, the phase and polarity are changed during transmission to allow the host device to communicate with a peripheral slave with different requirements. The flexibility of the SPI system allows direct connection to almost all existing synchronous serial peripherals.

19.7.2 SPI Transfer Format

During SPI transmission, data is sent simultaneously (serially shift out) and received (serially shift in). The serial clock line is synchronized with the shift and sampling of the two serial data lines. Slave selection line allows the individual selection of slave SPI devices; Slaves that are not selected do not interfere with SPI bus activity. On the SPI host device, the slave selection line may be selectively used to indicate multi-master bus competition.

19.7.3 CPHA=0 Transfer Format

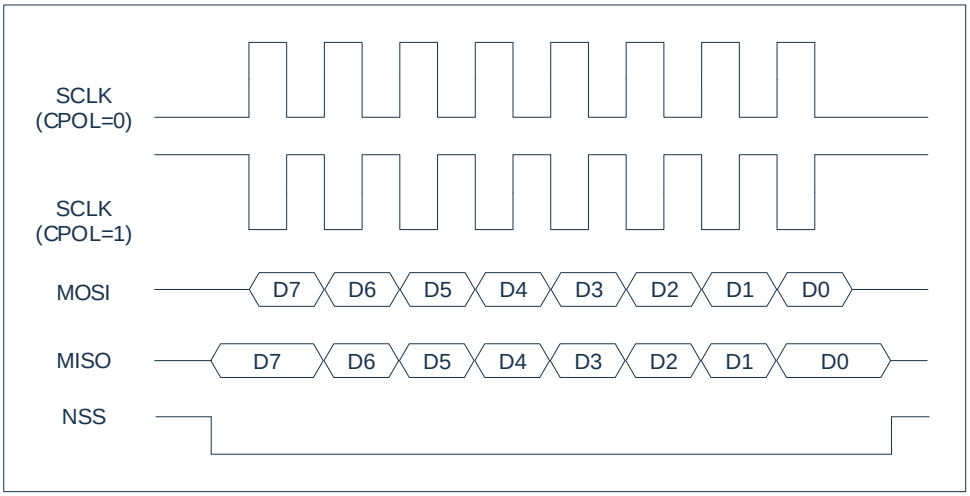
The following figure shows a timing diagram of an SPI transmission with a CPHA of 0. SCLK shows two waveforms: one for CPOL equal to 0 and one for CPOL equal to 1. The diagram can be described as a master device or slave device timing diagram with the master/slave (MISO) and master out/out (MOSI) pins directly connected between the master and slave. The MISO signal is output from the Slave and the MOSI signal is the host output. The slave selection input of the NSS line is slave; The NSS pin of the host is not displayed, but is assumed to be invalid. The NSS pin of the host must be high. This timing diagram functionally describes how the transmission takes place; It should not be used as a substitute for datasheet parameter information.



When CPHA=0, the NSS line must unset and reset between each consecutive serial byte. In addition, if the slave writes data to the SPI Data Register (SPDR) when the NSS is low, a write collision error is generated. When CPHA = 1, the NSS line may remain low between consecutive transmissions (which can always be kept low). In systems with a single fixed master and a single slave driving the MISO data line, this format is sometimes preferred.

19.7.4 CPHA=1 Transfer Format

The following figure is a timing diagram of the SPI transmission with CPHA = 1. SCLK shows two waveforms: one for CPOL=0 and one for CPOL=1. Since the SCLK, MISO, and MOSI pins are directly connected between the master and slave, this diagram can be interpreted as a master or slave timing diagram. The MISO signal is output from the Slave and the MOSI signal is the host output. The slave selection input of the NSS line is slave; The NSS pin of the host is not displayed, but is assumed to be invalid. The NSS pin of the host must be high or must be reconfigured to a general-purpose output that does not affect the SPI.



19.8 SPI Data Transfer

19.8.1 SPI Transfer Starts

All SPI transfers are initiated and controlled by the master SPI device. As a slave device, the SPI will consider the transmission starting at the first SCLK edge or the falling edge of the NSS, depending on the CPHA format chosen. When CPHA = 0, the falling edge of the NSS indicates the start of the transmission. When CPHA = 1, the first edge on SCLK indicates the start of the transfer. Regardless of the CPHA mode, the transmission can be aborted by making the NSS line high, but resetting the SPI slave logic and counter. The selected SCLK rate has no effect on slave operation because the master's clock is controlling the transmission.

When SPI is configured as a host, the transfer is initiated by software that writes to the SPDR.

19.8.2 SPI Transfer ends

When the SPIF flag is set to 1, the SPI transfer is technically completed, but depending on the configuration of the SPI system, there may be other tasks. Since the SPI bit rate does not affect the timing of the end period, only the fastest rate is considered in the discussion during the end period. When the SPI is configured as a host, the SPIF asserts at the end of the eighth SCLK cycle cycle. When the CPHA is equal to 1, the SCLK is inactive in the last half of the eighth SCLK cycle.

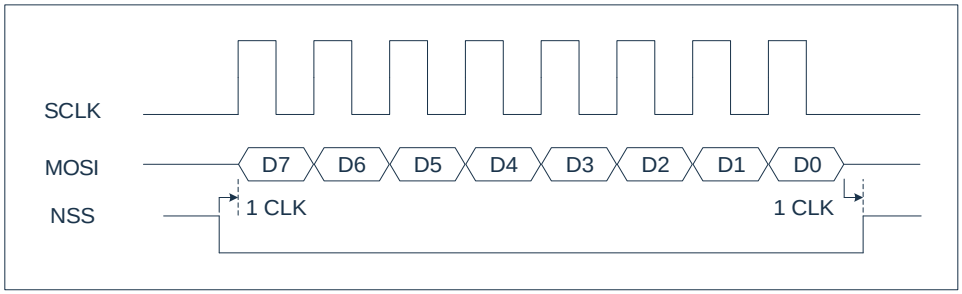
Because the SCLK line can be asynchronous with the slave's MCU clock, and the slave cannot access as much information as the master does to the SCLK cycle, the end cycle is different when the SPI is running as a slave. For example, when CPHA = 1, where the last SCLK edge occurs in the middle of the eighth SCLK cycle, the slave has no way of knowing when the previous SCLK cycle ended. For these reasons, the slave believes that after the last bit of the serial data is sampled, the transmission is complete, which corresponding the middle of the eighth SCLK cycle.

The SPIF flag is set at the end of the transmission, but the NSS line is still low, and the slave does not allow new data to be written to the SPDR.

19.9 SPI Timing Diagram

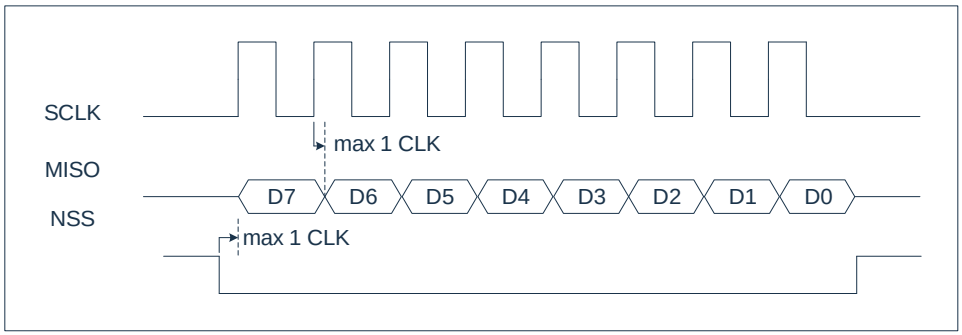
19.9.1 Master Mode Transmission

When the clock polarity of the SPI is CPOL=0 and the clock phase CPHA=1, the NSS in SPI master mode is the clK of the system clock after the low level, the MOSI starts to output, and the DATA of the MOSI is output on the rising edge of the SCLK clock. The master mode timing diagram is shown in the following figure:



19.9.2 Slave Mode Transmission

When the clock polarity of the SPI is CPOL=0 and the clock phase is CPHA=1, the data on miso starts to output after the falling edge of the NSS line. Miso data output differs from the falling edge of the NSS by a maximum of 1 system clock CLK. The slave mode timing diagram is shown in the following figure:



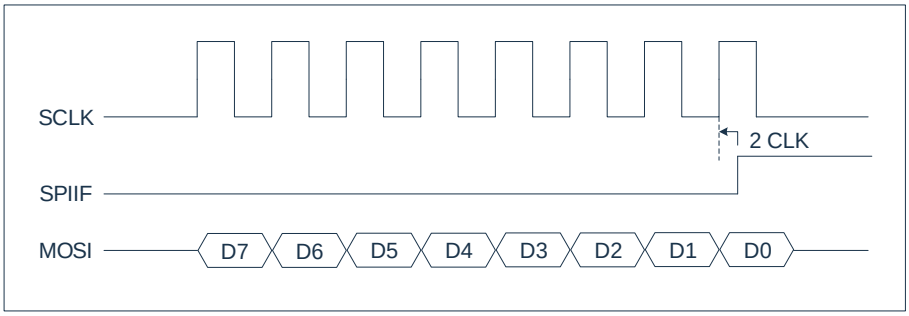
19.10 SPI Interrupt

The interrupt number of the SPI is 22, where the interrupt vector is 0x00B3. To enable an SPI interrupt, it must set its enable bit SPIIE to 1 and the global interrupt enable bit EA to 1.

If the SPI-related interrupt enables are all turned on, the CPU will enter the interrupt service program when the SPI global interrupt indicator bit SPIIF=1. The SPIIF operation properties are read-only and independent of the state of SPIIE.

After the SPI status register SPSR has either of the transmission completion flagSISIF and the write violation WCOL flag 1, the SPI global interrupt indicator bit SPIIF will be set to 1. SPIIF automatically clears 0 only if all three flag bits are 0.

When the clock polarity of the SPI is CPOL=0 and the clock phase is CPHA=1, the SPIIF in the SPI master mode generates clKs after the rising edge of the eighth SCLK clock in each frame of data, and the timing diagram is shown in the following figure:



19.10.1 Interrupt Mask Register EIE2

0xAA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
AndIE2	SPIIE	I2CIE	WDTIE	ADCIE	PWMIE	--	ET4	ET3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 SPIIE: SPI interrupt enable bit;
1= Enable SPI interrupts;
0= Disable SPI Interrupt.
Bit6 I2CIE: I2C interrupt enable bit;
1= Allow I2C interrupts;
0= Disable I2C Interrupt.
Bit5 WDTIE: WDT interrupt enable bit;
1= Enable WDT overflow interrupts;
0= Disable WDT overflow interrupts.
Bit4 ADCIE ADC interrupt enable bit;
1= Enable ADC interrupts;
0= Disable ADC interrupts.
Bit3 PWMIE: PWM global interrupt enable bit;
1= Enable all PWM interrupts;
0= Disable all PWM interrupts.
Bit2 -- Reserved, must be 0.
Bit1 ET4: Timer4 interrupt enable bit;
1= Enable Timer4 interrupts;
0= Disable Timer4 Interrupt.
Bit0 ET3: Timer3 interrupt enable bit;
1= Enable Timer3 interrupts;
0= Disable Timer3 Interrupt.

19.10.2 Interrupt Priority Control Register EIP2

0xB8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ANDTHEP2	PSPI	PI2C	PWDT	PADC	PPWM	--	PT4	PT3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 PSPI: SPI interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.
Bit6 PI2C: I2C interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.
Bit5 PWDT: WDT interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.
Bit4 PADC: ADC interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.
Bit3 PPWM: PWM interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.
Bit2 --: Reserved, must be 0.
Bit1 PT4: TIMER4 interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.
Bit0 PT3: TIMER3 interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

19.10.3 Peripheral Interrupt Flag Bit Register EIF2

0xB2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIF2	SPIIF	I2CIF	--	ADCIF	PWMIF	--	TF4	TF3
R/W	R	R	--	R/W	R	--	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 SPIIF: SPI global interrupt indicator bit, read-only;
1= SPI generates an interrupt, (this bit is automatically cleared after the specific interrupt flag is cleared);
0= The SPI did not produce an interrupt.
Bit6 I2CIF: I2C global interrupt indicator bit, read-only;
1= I2C produces an interrupt, (after clearing the specific interrupt flag, this bit is automatically cleared);
0= I2C did not produce an interrupt.
Bit5 -- Reserved, must be 0.
Bit4 ADCIF: ADC interrupt flag bit;
1= ADC conversion is completed, and software zeroing is required;
0= The ADC conversion was not completed.
Bit3 PWMIF: PWM global interrupt indicator bit, read-only;
1= PWM generates an interrupt, (after the specific interrupt flag is cleared, this bit is automatically cleared);
0= The PWM did not produce an interrupt.
Bit2 -- Reserved, must be 0.
Bit1 TF4: Timer4 timer overflow interrupt flag bit;
1= Timer4 timer overflow, the hardware is automatically cleared when entering the interrupt service program, and the software can also be cleared;
0= The Timer4 timer has no overflow.
Bit0 TF3: Timer3 timer overflow interrupt flag bit;
1= Timer3 timer overflow, when entering the interrupt service program, the hardware is automatically cleared, and the software can also be cleared;
0= The Timer3 timer has no overflow.

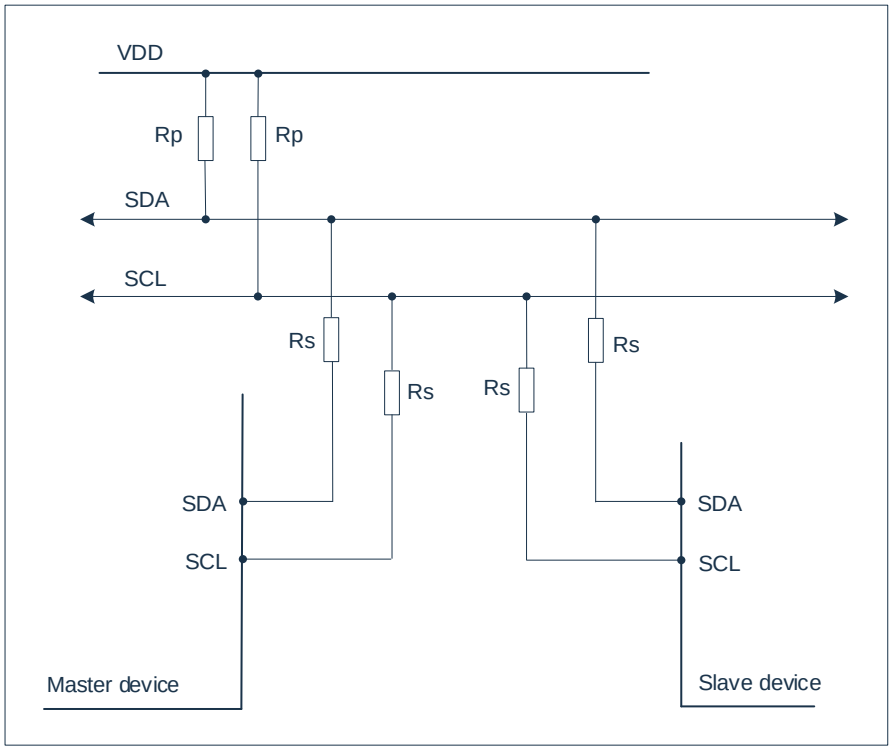
20. I2C Module

20.1 Overview

The module provides an interface between the microcontroller and the I2C bus, as shown in the connection diagram below, and supports arbitration and clock synchronization to allow operation in multi-master systems. I2C supports normal, fast mode.

The I2C module has the following characteristics:

- Support 4 working modes: master transmission, master reception, slave transmission, slave reception.
- Supports 2 transfer speed modes:
  - Standard (up to 100Kb/s);
  - Fast (up to 400Kb/s);
- Perform arbitration and clock synchronization.
- Support for multi-master systems.
- The master mode supports7-bit addressing mode and 10-bit addressing mode on the I2C-bus (software supported).
- Slave mode supports7-bit addressing mode on the I2C-bus.
- Interrupt generation.
- Allows operation over a wide clock frequency range (built-in 8-bit timer).
- I2C timeout detection function.



20.2 I2C Port Configuration

If you use the I2C function, you should first configure the corresponding port as an SCL, SDA channel. For example, configure P04, P03 ports as I2C function:

```
PS_SCL = 0x04;    Select the P04 port as the SCL pin
PS_SDA = 0x03;    /Select P03 port as the SDA pin
P04CFG = 0x03;    P04 Multiplex SCL function
P03CFG = 0x03;    P03 multiplex SDA function
```

After configuring I2C channel, this group of ports is in open drain state by default. You can configure whether to enable SCL, internal pull-up resistance of SDA port, or add pull-up resistance outside the chip through PxUP.

In the master control mode, IIC outputs SCL to the slave. After sending the address or data, the slave needs to pull the SCL down and send back the corresponding response signal to the host. The host needs to read back the SCL port line status to detect whether the slave releases the SCL to determine whether the next frame data transmission is required. If the pull-up resistance or board-level parasitic capacitance of SCL is larger, the reading back time will be longer, which will affect the communication speed of IIC. Please refer to IIC application manual for details.

20.3 I2C Master Mode

There are six registers for connecting to the master: control, status, slave address, transmit data, receive data, and timer cycle registers.

register		address
write	Read	
Slave address register I2CMSA	Slave address register I2CMSA	0xF4
Master mode control register I2CMCR	Master mode status register I2CMSR	0xF5
The master transmits the data register I2CMBUF	The master receives data register I2CMBUF	0xF6
Timing cycle register I2CMTP	Timing cycle register I2CMTP	0xF7

The master mode control register I2CMCR shares a register address with the master mode status register I2CMSR, but is physically two different registers.

The master transmit data register shares a register address with the master receive data register, and the write operation accesses the transmit register I2CMBUF and the read operation accesses the receiving register I2CMBUF.

Write operations are written as control registers, and read operations are read as status registers.

20.3.1 I2C Master Mode Timing Cycle Register

To generate a wide range of SCL frequencies, the module has a built-in 8-bit timer. For standard and fast transfers.

TIMER_PRD ≠ 0, the ideal clock period of the SCL: 2* (1+TIMER_PRD)*10* Tsys

TIMER_PRD = 0, the ideal clock period of the SCL: 3* 10* Tsys

Refer to IIC Application Manual for the specific calculation formula of SCL.

0xF7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CMTP	--	MTP6	MTP5	MTP4	MTP3	MTP2	MTP1	MTP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	1

Bit7 -- Reserved, must be 0.

Bit6~Bit0 MTP<6:0>: Period timing registers in standard and fast mode, bits 6-0: TIMER_PRD [6:0].

20.3.2 I2C Master Mode Control and Status Registers

The control registers include 4 bits: RUN, START, STOP, ACK bits. The START bit will produce the START or RESTART START condition. The STOP bit determines whether the data transfer stops at the end of the cycle, or continues. To generate a single transmission cycle, the slave address register writes to the desired address, the R/S bit is set to 0, and the control register writes to ACK=x, STOP=1, START=1, RUN=1 (I2CMCR=xxx0_x111x) to perform the operation and stop. An interrupt occurs when the operation completes (or an error occurs). Data can be read from the receiving data register.

When I2C is operating in master mode, the ACK bit must be set to 1. This will cause the I2C-Bus controller to automatically send a reply after each byte. When the I2C-bus controller no longer needs to send data from the slave, the bit must clear 0.

Master mode control registers

0xF5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CMCR	RSTS	--	--	--	ACK	STOP	START	RUN
R/W	W	R	R	W	W	W	W	W
Reset value	0	0	1	0	0	0	0	0

Bit7 RSTS: I2C master module reset control bit;
1= Reset the master module (I2C registers for the entire master module, including I2CMSR);
0= The interrupt flag bit in I2C master mode is clear to 0.
Bit6~Bit5 -- Reserved.
Bit4 -- Reserved, must be 0.
Bit3 ACK: Answer enable bit;
1= Enable;
0= Disable.
Bit2 STOP: Stop enable bit;
1= Enable;
0= Disable.
Bit1 START: Start the enable bit;
1= Enable;
0= Disable.
Bit0 RUN: Run the enable bit;
1= Enable;
0= Disable.

Various operations in master mode can be implemented through the following list of control bit combinations:

START: Sends a start signal.

SEND: Send data or address.

RECEIVE: Receives data.

STOP: Send an end signal.

Combination of control bits (IDLE state).

R/S	ACK	STOP	START	RUN	OPERATION
0	-	0	1	1	START followed by SEND (master remains in send mode)
0	-	1	1	1	START is followed by SEND and STOP
1	0	0	1	1	Non-responsive reception after START (master remains in receiver mode)
1	0	1	1	1	START is followed by REVIVE and STOP
1	1	0	1	1	START followed by RECOVER (master remains in receiver mode)
1	1	1	1	1	Combinations are prohibited
0	0	0	0	1	Combinations are prohibited

Combination of control bits (master send status)

R/S	ACK	STOP	START	RUN	OPERATION
-	-	0	0	1	Send operation
-	-	1	0	0	Stop it
-	-	1	0	1	SEND is followed by STOP
0	-	0	1	1	Repeat START followed by SEND
0	-	1	1	1	Repeat START, followed by SEND and STOP
1	0	0	1	1	Repeat the START condition followed by the response RECEIVE operation (Master remains in receiver mode)
1	0	1	1	1	Repeat START, followed by the SEND and STOP conditions
1	1	0	1	1	Repeat the START condition followed by RECEIVE (Master remains in receiver mode)
1	1	1	1	1	Combinations are prohibited

Combination of control bits (master receive state)

R/S	ACK	STOP	START	RUN	OPERATION
-	0	0	0	1	RECEIVE operation with reply (Master remains in receiver mode)
-	-	1	0	0	STOP
-	0	1	0	1	RECEIVE is followed by STOP
-	1	0	0	1	RECEIVE operation (master remains in receiver mode).
-	1	1	0	1	Combinations are prohibited
1	0	0	1	1	The START is repeated, followed by a response RECEIVE operation (Master remains in receiver mode)
1	0	1	1	1	Repeat START, followed by RECEIVE and STOP
1	1	0	1	1	Repeat START followed by RECEIVE (Master remains in receiver mode)
0	-	0	1	1	Repeat START followed by SEND (Master remains in transmitter mode)
0	-	1	1	1	Repeat START, followed by SEND and STOP

Master mode status register I2CMSR

0xF5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CMSR	I2CMIF	BUS_BUSY	IDLE	ARB_LOST	DATA_ACK	ADD_ACK	ERROR	BUSY
R/W	R	R	R	R	R	R	R	R
Reset value	0	0	1	0	0	0	0	0

Bit7 I2CMIF: I2C master mode interrupt flag bit;
1= In master mode, send/receive completes, or a transmission error occurs. (Software zero, write 0 to clear);
0= No interrupt was generated.
Bit6 BUS_BUSY: I2C-bus busy flag bit in master mode/slave mode;
1= The I2C-bus is busy and cannot be transmitted (cleared by the start bit 1 on the bus, stop condition).
0= --
Bit5 IDLE: I2C master mode idle flag bit;
1= is idle;
0= is the working status.
Bit4 ARB_LOST: I2C Master Mode Arbitration Flag Bit;
1= Bus control is lost.
0= --
Bit3 DATA_ACK: I2C master mode transmits data response flag bits;
1= The last time the data was sent was not answered.
0= --
Bit2 ADD_ACK: I2C Master Mode Addressing Answer Flag Bit;
1= The last addressing did not answer.
0= --
Bit1 ERROR: I2C Master mode error flag bit;
1= Addressing slaves does not answer/send data without answer/I2C-bus arbitration conflicts.
0= --
Bit0 BUSY: I2C Master module busy flag bit;
1= The I2C module is transferring data.
0= --

20.3.3 I2C Slave Address Register

The slave address register consists of 8 bits: 7 bits of address (A6-A0) and receive/transmit bits R/S. The R/S bit determines whether the next operation is to receive (1) or send (0).

Master mode slave address register I2CMSA

0xF4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CMSA	SA6	SA5	SA4	SA3	SA2	SA1	SA0	R/S
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit1 AT<6:0>: I2C Master mode slave address.
Bit0 R/S: I2C Master mode after sending slave address after receiving/sending status selection bits;
1= Receive data after correct addressing;
0= The data is sent after correct addressing.

20.3.4 I2C Master Mode Transmit and Receive Data Registers

The transmit data register consists of eight data bits that will be sent on the bus on the next send or burst operation, the first of which is MD7 (MSB).

Master mode data cache register I2CMBUF

0xF6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CMBUF	MD7	MD6	MD5	MD4	MD3	MD2	MD1	MD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 MD<7:0>: Send/receive data in I2C master mode.

20.4 I2C Slave Mode

There are five registers for connecting to the target device: self address, control, status, send data, and receive data registers.

register		address
write	Read	
Self address register I2CSADR	Self address register I2CSADR	0xF1
Control register I2CSCR	Status register I2CSSR	0xF2
Send data I2CSBUF	Receive data I2CSBUF	0xF3

20.4.1 I2C Own Address Register I2CSADR

The own address register consists of seven address bits that identify the I2C core on the I2C bus. This register can R/W addresses.

Own address register I2CSADR

0xF1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CSADR	--	SA6	SA5	SA4	SA3	SA2	SA1	SA0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 -- Reserved, must be 0.

Bit6~Bit0 AT<6:0>: Theown address of the I2C slave mode.

20.4.2 I2C Slave Mode Control and Status Registers I2CSCR/I2CSSR

Slave mode control registers and slave mode status registers occupy a register address, using different operations to access the two registers separately:

Write operation: Write to I2CSCR (write only)

Read operation: Read I2CSSR (read only).

The control register consists of two bits: RSTS and DA bits. The RSTS bit controls the reset of the entire I2C slave module, and when the I^2C bus encounters some problem, the software enables the bit to reinitialize the I2CS. The DA bit enables and disables I2CS device operation. Reading this address places the status register on the data bus.

Slave mode control register I2CSCR

0xF2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CSCR	RSTS	--	--	--	--	--	--	OF
R/W	In	R	R	R	R	R	R	In
Reset value	0	0	0	0	0	0	0	0

Bit7 RSTS: I2C Slave module reset control bit;

1= Reset slave module;

0= No impact.

Bit6~Bit1 -- Reserved, must be 0.

Bit0 FROM: I2C Slave mode enable bit;

1= Enable;

0= Disable.

The status register consists of three bits: sendfin bit, RREQ bit, TREQ bit. The SENDFIN bit of Send Complete indicates that the Master I2C controller has completed the receipt of data during a single or continuous I2CS transmit operation. The Receive Request RREQ bit indicates that the I2CS device has receiveda data byte from the I2C master, and the I2CS device should read a data byte from the receiving data register I2CSBUF. The Send Request TREQ bit indicates that the I2CS device is addressed as a slave transmitter, and the I2CS device should write a byte of data to the transmit data register I2CSBUF. If the I2C interrupt enable is on, any of the 3 flags at 1 will produce an interrupt.

The bus busy flag in slave mode is judged by bit6 (BUS_BUSY) of the master mode status register I2CMSR, which is 0x20 when the bus is idle, I2CMSR register is 0x60 when the start condition is generated and the stop condition is generated, and I2CMSR is 0x20 when the stop condition is generated.

Slave mode status register I2CSSR

0xF2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CSSR	--	--	--	--	--	SENDFIN	TREQ	RREQ
R/W	--	--	--	--	--	R	R	R
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit3 -- Reserved, must be 0.

Bit2 SENDFIN: I2C slave mode sends the operation completion flag bit, read-only.

1= The data is no longer required for the master device, treQ is no longer set to 1, and the data transfer has been completed. (Automatic zeroing after reading I2CSCR).

0= --

Bit1 TREQ: I2C slave mode prepares to send flag bits, read-only.

1= As the transmitting device has been addressed or the master device is ready to receive data. (Auto zero after writing I2CSBUF).

0= --

Bit0 RREQ: I2C slave mode receives completion flag bits, read-only.

1= Received. (Automatic zeroing after reading I2CSBUF).

0= Not received.

20.4.3 I2C Slave Mode Transmit and Receive Buffer Registers I2CSBUF

0xF3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CSBUF	I2CSBUF7	I2CSBUF6	I2CSBUF5	I2CSBUF4	I2CSBUF3	I2CSBUF2	I2CSBUF1	I2CSBUF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 I2CSBUF<7:0>: data sent or received by I2C;

Write operation: Write the data that will be sent (the order of sending is from the high bit to the low bit);

Read operation: Data that has been received.

20.5 I2C Interrupt

The interrupt number for I2C is 21, where the interrupt vector is 0x00AB. The enable I2C interrupt must set its enable bit I2CIE to 1 and the global interrupt enable bit EA to 1.

If the I2C-related interrupt enables are turned on,the CPU will enter the interrupt service program when the I2C global interrupt indicator bit I2CIF=1 is turned on. The I2CIF operation properties are read-only and independent of the state of the I2CIE.

I2C Master mode interrupt flag bit I2CMIF, slave mode transmit operation completion flag bit SENDFIN, slave mode ready to send flag bit TREQ, slave mode receive completion flag bit RREQ Any one is 1,I2C global interrupt indicator bit I2CIF Will be set to 1. I2CIF automatically clears 0 only if all four flag bits are 0.

20.5.1 Interrupt Mask Register EIE2

0xAA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
AndIE2	SPIIE	I2CIE	WDTIE	ADCIE	PWMIE	--	ET4	ET3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 SPIIE: SPI interrupt enable bit;
1= Enable SPI interrupts;
0= Disable SPI Interrupt.
Bit6 I2CIE: I2C interrupt enable bit;
1= Allow I2C interrupts;
0= Disable I2C Interrupt.
Bit5 WDTIE: WDT interrupt enable bit;
1= Enable WDT overflow interrupts;
0= Disable WDT overflow interrupts.
Bit4 ADCIE: ADC interrupt enable bit;
1= Enable ADC interrupts;
0= Disable ADC interrupts.
Bit3 PWMIE: PWM global interrupt enable bit;
1= Enable all PWM interrupts;
0= Disable all PWM interrupts.
Bit2 -- Reserved, must be 0.
Bit1 ET4: Timer4 interrupt enable bit;
1= Enable Timer4 interrupts;
0= Disable Timer4 Interrupt.
Bit0 ET3: Timer3 interrupt enable bit;
1= Enable Timer3 interrupts;
0= Disable Timer3 Interrupt.

20.5.2 Interrupt Priority Control Register EIP2

0xBA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ANDTHEP2	PSPI	PI2C	PWDT	PADC	PPWM	--	PT4	PT3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 PSPI: SPI interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

Bit6 PI2C: I2C interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

Bit5 PWDT: WDT interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

Bit4 PADC: ADC interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

Bit3 PPWM: PWM interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

Bit2 --: Reserved, must be 0.

Bit1 PT4: TIMER4 interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

Bit0 PT3: TIMER3 interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

20.5.3 Peripheral Interrupt Flag Bit Register EIF2

0xB2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIF2	SPIIF	I2CIF	--	ADCIF	PWMIF	--	TF4	TF3
R/W	R	R	--	R/W	R	--	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 SPIIF: SPI global interrupt indicator bit, read-only;
1= SPI generates an interrupt, (this bit is automatically cleared after the specific interrupt flag is cleared);
0= The SPI did not produce an interrupt.
Bit6 I2CIF: I2C global interrupt indicator bit, read-only;
1= I2C produces an interrupt, (after clearing the specific interrupt flag, this bit is automatically cleared);
0= I2C did not produce an interrupt.
Bit5 -- Reserved, must be 0.
Bit4 ADCIF: ADC interrupt flag bit;
1= ADC conversion is completed, and software zeroing is required;
0= The ADC conversion was not completed.
Bit3 PWMIF: PWM global interrupt indicator bit, read-only;
1= PWM generates an interrupt, (after the specific interrupt flag is cleared, this bit is automatically cleared);
0= The PWM did not produce an interrupt.
Bit2 -- Reserved, must be 0.
Bit1 TF4: Timer4 timer overflow interrupt flag bit;
1= Timer4 timer overflow, the hardware is automatically cleared when entering the interrupt service program, and the software can also be cleared;
0= The Timer4 timer has no overflow.
Bit0 TF3: Timer3 timer overflow interrupt flag bit;
1= Timer3 timer overflow, when entering the interrupt service program, the hardware is automatically cleared, and the software can also be cleared;
0= The Timer3 timer has no overflow.

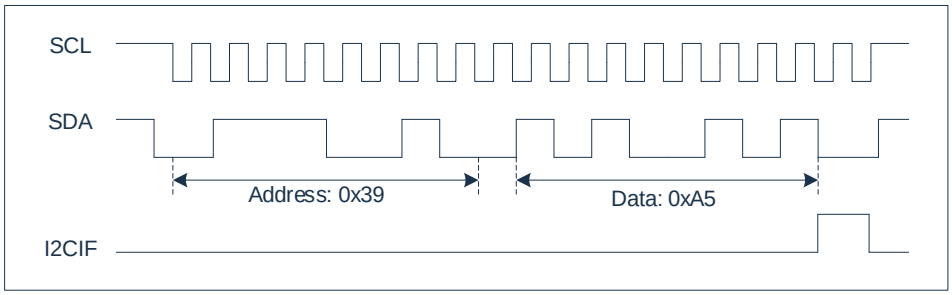
20.6 I2C Slave Mode Transmission Mode

All rendered waveforms in this section default I2C to have their own address 0x39 ("00111001").

20.6.1 Single Receive

The following figure shows the sequence of signals received by I2C during a single data session. Single receive sequence:

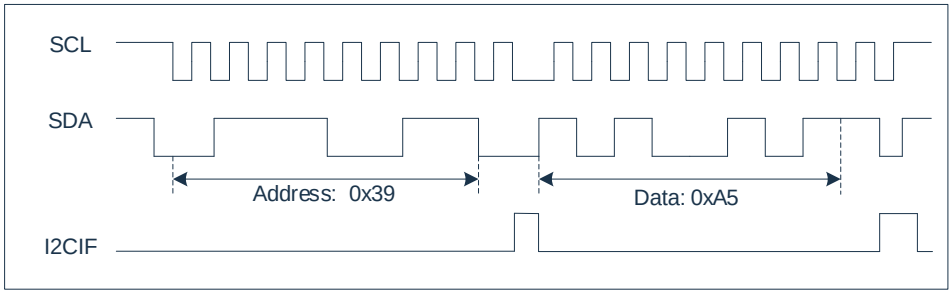
- Starting conditions;
- The I2C is addressed by the I2C master as the receiver;
- The address is confirmed by I2C;
- Data received by I2C;
- Data confirmed by I2C;
- Stop condition.



20.6.2 Single send

The following figure shows the sequence of signals sent by I2C during a single data session. Single send sequence:

- Starting conditions;
- I2C is addressed by the I2C master as a transmitter;
- The address is confirmed by I2C;
- Datais transmitted by I2C;
- Data is not confirmed by the I2C master;
- Stop condition.



20.6.3 Continuous reception

The following figure shows the sequence of signals received by I2C during continuous data reception. Continuous receive sequence:

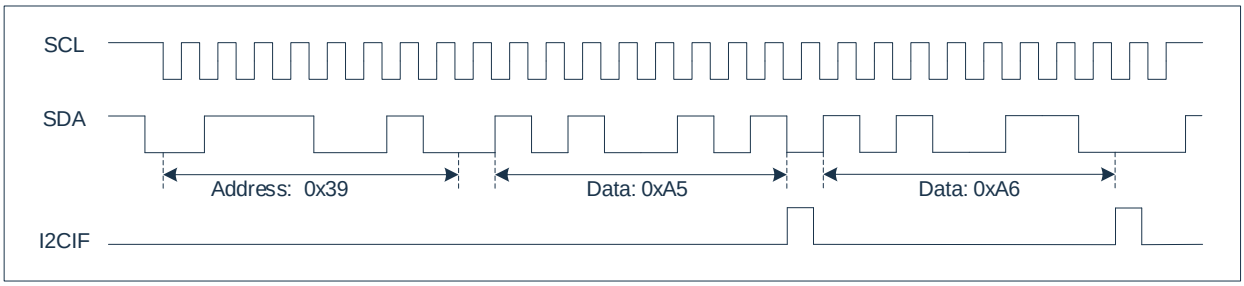
- Start conditions.
- I2C is addressed by the I2C master as a receiver.
- The address is confirmed by I2C.

1) Data is received by I2C.

2) The data is confirmed by I2C.

- Stop condition.

Sequences 1) and 2) repeat until the stop condition occurs.



20.6.4 Continuous Sending

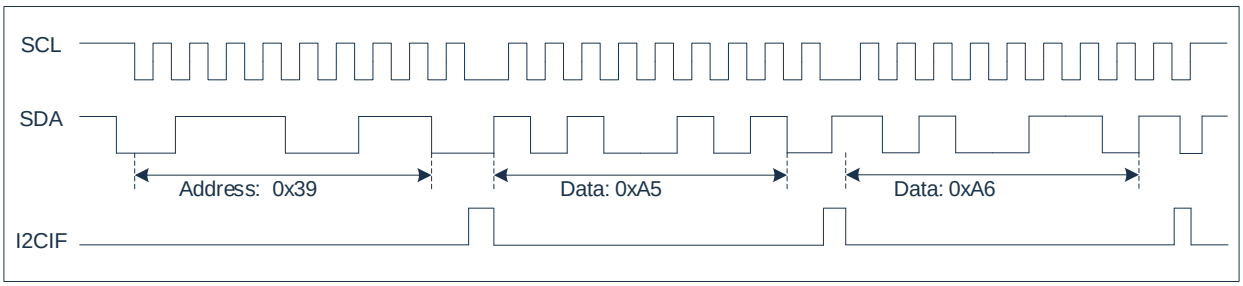
The following figure shows the sequence of signals sent by I2C during continuous data transmission. Consecutive send sequences:

- Send conditions.
- I2C is addressed by the I2C master as a transmitter.
- The address is confirmed by I2C.

1) The data is sent by I2C.
2) Data I2C master acknowledges data.
3) The last data is not confirmed by the I2C master.

- Stop condition.

Repeat sequences 1) and 2) until the last data sent is not confirmed by the I2C master 3).



21. UART0 Module

21.1 Overview

The Universal Synchronous Asynchronous Transceiver (UART0) provides a flexible way to exchange full-duplex data with external devices.

UART0 has two physically separate receive and transmit buffers, SBUF0, which distinguish between operations on a receive buffer or a transmit buffer by reading and writing instructions to SBUF0. When writing SBUF0, the data is loaded into the transmit buffer; When reading SBUF0, read the contents of the receive buffer.

UART0 has four modes of operation: one synchronous mode and three asynchronous modes. Modes 2 and 3 have a multi-Slave communication function that is enabled by placing SMnn2 as 1 in the SCON 0 register. The host processor first sends the address byte that identifies the target slave. Address bytes are different from data bytes in that bit 9 in the address byte is 1 and data bytes are 0. At SMn2=1, the slave is not interrupted by bytes of data. The address byte will interrupt all slaves. The addressed slave will clear its SMn2 bits and prepare to receive incoming bytes of data. The unaddressed slave sets SMn2 to 1 and ignores incoming data.

21.2 UART0 Port Configuration

Before using the UART0 module, the corresponding ports need to be configured as TXD0 and RXD0 channels of UART n. For example, the port configuration of UART0 is as follows:

PS_RXD0 = 0x13; Select P13 as the RXD0 pin

P13CFG = 0x03; P13 is multiplexed to the RXD0 function

P14CFG = 0x03; P14 multiplexed to TXD0 function

Port RXD0 of UART 0 can be selected via PS_RXD0 (only one of the RXD 0 pins can be selected), and port TXD0 can be selected by the port configuration register (either selected at the same time or by selecting one of them). If selected at the same time, the corresponding waveform will be output).

When using, it is recommended to set the working mode first, and then configure the corresponding port as a serial port.

21.3 UART0 Baud Rate

UART0 In mode 0, the baud rate is fixed to the division twelve of the system clock (Fsys/12); In mode 2, the baud rate is fixed to the system clock's division of thirty-two or division sixty-four (Fsys/32, Fsys/64); In modes 1 and 3, the baud rate is generated by the timer Timer1 or Timer4 or Timer2 or BRT module, and the chip chooses which timer to use as the baud rate clock source is determined by the register FUNCCR.

21.3.1 Baud Rate Clock Source

UART0 in modes 1 and 3, the baud rate clock source selection is as follows:

UART0 baud rate clock source selection:

FunccR [2:0] = 000, Timer1 is selected as the baud rate generator for UART0;

FunccR [2:0] = 001, Timer4 is selected as the baud rate generator for UART0;

FunccR [2:0] = 010, Select Timer2 as the baud rate generator for UART0;

FunccR [2:0] = 011, BRT is chosen as the baud rate generator for UART0.

21.3.2 Baud Rate Calculation

UART0 in mode 1 and mode 3, the baud rate calculation formula for different clock sources is as follows:

1) Timer1 or Timer4 works in 8-bit auto-reload mode with baud rate formula:

$$BaudRate=\frac{Fsys\times 2^{SMOn}}{32\times(4\times 3^{1-TxM})\times(256-THx)}\ (x=1,4)$$

SMOD0 is the baud rate select bit, set by the register PCON. T1M is the timer 1 clock select bit, set by register CKCON[4], T4M is the timer 4 clock select bit, set by register T34MOD[6]. That is, the value of Timer 1 or Timer4 at the corresponding baud rate of TH1/TH4 should be set to:

$$THx=256-\frac{Fsys\times 2^{SMODn}}{32\times(4\times 3^{1-TxM})\times BaudRate}\ (x=1,4)$$

2) Timer2 works in overflow auto Reload mode, equation for baud rate:

$$BaudRate=\frac{Fsys\times 2^{SMODn}}{32\times(12\times 2^{T2PS})\times(65536-\{RLDH,RLDL\})}$$

T2PS is a timer 2 clock prescaler selection bit set by register T2CON[7]. That is, the value of Timer2 at the corresponding baud rate should be set to:{RLDH,RLDL}

$$\{RLDH,RLDL\}=65536-\frac{Fsys\times 2^{SMODn}}{32\times(12\times 2^{T2PS})\times BaudRate}$$

3) When BRT is used as a baud rate generator, the baud rate formula:

$$BaudRate=\frac{Fsys\times 2^{SMODn}}{32\times(65536-\{BRTDH,BRTDL\})\times 2^{BRTCKDIV}}$$

BRTCKDIV is a BRT timer prescale selection bit, set by the register BRTCON. That is, the value of the BRT at the corresponding baud rate should be set to:{BRTDH,BRTDL}

$$\{BRTDH,BRTDL\}=65536-\frac{Fsys\times 2^{SMODn}}{32\times 2^{BRTCKDIV}\times BaudRate}$$

21.3.3 Baud Rate Error

UART0 in mode 1 and mode 3, select different baud rate clock sources, the error under different baud rate is as follows:

Table 1) and 2) are some baud rate information in the 8-bit automatic reload mode of timer 1/timer 4 in variable baud rate mode. Table 3) and 4) are some baud rate information related to the BRT timer overflow rate as the UART clock source in variable baud rate mode.

1）SMODn=0，T1M=1/T4M=1

baud rate	Fsys=8MHz			Fsys=16MHz			Fsys=24MHz			Fsys=48MHz		
bps	{TH1, TH4}	Current Rate	% Error	{TH1, TH4}	Current Rate	% Error	{TH1, TH4}	Current Rate	% Error	{TH1, TH4}	Current Rate	% Error
4800	243	4808	-0.16	230	4808	-0.16	217	4808	-0.16	178	4808	-0.16
9600	--	--	--	247	9615	-0.16	236	9375	2.34	217	9615	-0.16
19200	--	--	--	--	--	--	246	18750	2.34	236	18750	2.34
38400	--	--	--	--	--	--	251	37500	2.34	246	37500	2.34
115200	--	--	--	--	--	--	--	--	--	--	--	--
250000	--	--	--	--	--	--	--	--	--	--	--	--
500000	--	--	--	--	--	--	--	--	--	--	--	--

2）SMODn=1，T1M=1/T4M=1

baud rate	Fsys=8MHz			Fsys=16MHz			Fsys=24MHz			Fsys=48MHz		
bps	{TH1, TH4}	Current Rate	% Error	{TH1, TH4}	Current Rate	% Error	{TH1, TH4}	Current Rate	% Error	{TH1, TH4}	Current Rate	% Error
4800	230	4808	-0.16	204	4808	-0.16	178	4808	-0.16	100	4808	-0.16
9600	243	9615	-0.16	230	9615	-0.16	217	9615	-0.16	178	9615	-0.16
19200	--	--	--	243	19230	-0.16	236	18750	2.34	217	19231	-0.16
38400	--	--	--	--	--	--	246	37500	2.34	236	37500	2.34
115200	--	--	--	--	--	--	--	--	--	--	--	--
250000	--	--	--	--	--	--	--	--	--	--	--	--
500000	--	--	--	--	--	--	--	--	--	--	--	--

3）SMODn=0，BRTCKDIV=0

baud rate	Fsys=8MHz			Fsys=16MHz			Fsys=24MHz			Fsys=48MHz		
bps	{BRTH, BRTL}	Current Rate	% Error	{BRTH, BRTL}	Current Rate	% Error	{BRTH, BRTL}	Current Rate	% Error	{BRTH, BRTL}	Current Rate	% Error
4800	65484	4808	-0.16	65432	4808	-0.16	65380	4808	-0.16	65224	4808	-0.16
9600	65510	9615	-0.16	65484	9615	-0.16	65458	9615	-0.16	65380	9615	-0.16
19200	65523	19231	-0.16	65510	19231	-0.16	65497	19231	-0.16	65458	19231	-0.16
38400	--	--	--	65523	38462	-0.16	65516	37500	2.34	65497	38462	-0.16
115200	--	--	--	--	--	--	--	--	--	65523	115385	-0.16
250000	--	--	--	--	--	--	--	--	--	65530	250000	0
500000	--	--	--	--	--	--	--	--	--	65533	500000	0

4）SMODn=1，BRTCKDIV=0

baud rate	Fsys=8MHz			Fsys=16MHz			Fsys=24MHz			Fsys=48MHz		
bps	{BRTH, BRTL}	Current Rate	% Error	{BRTH, BRTL}	Current Rate	% Error	{BRTH, BRTL}	Current Rate	% Error	{BRTH, BRTL}	Current Rate	% Error
4800	65432	4808	-0.16	65328	4808	-0.16	65224	4792	0.16	64911	4800	0
9600	65484	9615	-0.16	65432	9615	-0.16	65380	9615	-0.16	65224	9615	-0.16
19200	65510	19231	-0.16	65484	19231	-0.16	65458	19231	-0.16	65380	19231	-0.16
38400	65523	38462	-0.16	65510	38462	-0.16	65497	38462	-0.16	65458	38462	-0.16
115200	--	--	--	--	--	--	65523	115385	-0.16	65510	115385	-0.16
250000	--	--	--	--	--	--	--	--	--	65524	250000	0
500000	--	--	--	--	--	--	--	--	--	65530	500000	0
1000000	--	--	--	--	--	--	--	--	--	65533	1000000	0

21.4 UART0 Register

The UART0 has the same features as the standard 8051 UART. Its Related Registers are: FUNCCR, SBUF0, SCON0, PCON, IE, IP, EIP3。 The UART0 data buffer (SBUF0) consists of 2 independent registers: the transmit and receive registers. The data written to SBUF0 will be set in the UART0 output register and the transmission will begin; The data read from SBUF0 will be read from the UART0 receiving register. The SCON0 register supports bit addressing operations, and the SCON1 registers do not support bit addressing operations, so be aware when using assembly language. The baud rate is doubled by register PCON settings.

21.4.1 UART0 Baud Rate Selection Register FUNCCR

0x91	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FUNCCR	--	--	---	--		UART0_CKS2	UART0_CKS1	UART0_CKS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7-Bite3	--	Reserved, must be 0.
Bit2-Bit0	UART0_CKS<2:0>:	Timer clock source selection for UART0
	000=	Overflow clock for Timer1;
	001=	Overflow clock for Timer4;
	010=	Overflow clock for Timer2;
	011=	BRT overflow clock;
	Other =	Forbidden Access.

21.4.2 UART0 Buffer Register SBUF0

0x99	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SBUFn	BUFFERn7	BUFFERn6	BUFFERn5	BUFFERn4	BUFFERn3	BUFFERn2	BUFFERn1	BUFFERn0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0	BUFFER0<7:0>:	Buffered data registers.
	Write:	UART0 starts sending data.
	Read:	Reads the received data.

21.4.3 UART Control Register SCON0

0x98	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SCON0	In0SM0	In0SM1	In0SM2	U0REN	In0TB8	In0RB8	TI0	RI0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit6 IN0SM0- IN0SM1: Multi-Slave communication control bit;
00= Master synchronization mode;
01= 8-bit asynchronous mode, variable baud rate;
10= 9-bit asynchronous mode with baudrates of F sys/32 or Fsys/64;
11= 9-bit asynchronous mode with variable baud rate.

Bit5 In0SM2: Multi-machine communication control bit;
1= Enable;
0= Disable.

Bit4 U0REN: Receive enable bits;
1= Enable;
0= Disable.

Bit3 In0TB8: The 9th bit of sending data, mainly used for sending in 9-bit asynchronous mode;
1= The 9th digit is 1;
0= The 9th digit is 0.

Bit2 In0RB8: The 9th bit of receiving data, mainly used for sending in 9-bit asynchronous mode;
1= The 9th bit of data received is 1;
0= Bit 9 received is 0.

Bit1 TI0: Send interrupt flag bits (requires software zeroing);
1= Indicates that the send buffer is empty, and you can send the frame data.
0= --

Bit0 RI0: Receive interrupt flag bits (requires software zeroing);
1= Indicates that the receive buffer is full, and the next frame of data can be received after reading.
0= --

The UART0 schema is as follows:

SM00	SM01	mode	description	baud rate
0	0	0	Shift register	Fsys/12
0	1	1	8-Bit UART	Controlled by Timer4/Timer1/Timer2/BRT
1	0	2	9-Bit UART	SMOD0=0：Fsys/64; SMOD0=1：Fsys/32
1	1	3	9-Bit UART	Controlled by Timer4/Timer1/Timer2/BRT

21.4.4 PCON Registers

0x87	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON	SMOD0	--	--	--	--	THEIR	STOP	IDLE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

B Registers in ANK0

Bit7 SMOD0: UART0 baud rate multiplier;
1= UART0 baud rate doubled;
0= The UART0 baud rate is normal.

Bit6~Bit3 -- Reserved, must be 0.

Bit2 THEIR: STOP status function wake-up enable bit;
(The system can be restarted by a power-down reset or an enabled external reset regardless of the SWE value).
0= Disables functional wake-up;
1= Enables function wake-up (wake-up by external interrupts and timed wake-ups).

Bit1 STOP: Sleep state control bit;
1= Enter the sleep state (exit STOP mode to automatically clear zero);
0= Does not go into hibernation.

Bit0 IDLE: Idle state control bit;
1= Enter the idle state (exit THE IDLE mode to automatically clear the zero);
0= Not idle

21.5 UART0 Interrupt

The interrupt number of UART0 is 4, where the interrupt vector is 0x0023.

To enable the UART0 interrupt, it must set its enable bit ES0 to 1 and the global interrupt enable bit EA to 1. If the interrupt enables associated with UART0 are all turned on, TI0=1 or RI0=1, the CPU will enter the corresponding interrupt service program. TI0/RI0 is independent of the state of ES0 and requires software clearance, which describes reference register SCON0 in detail.

21.5.1 Interrupt Mask Register IE

0xA8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IE	SHE	--	ET2	ES0	ET1	EX1	ET0	EX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 EA: Global interrupt enable bits;
1= Enable all unblocked interrupts;
0= Disable all interrupts.
Bit6 -- Reserved, must be 0.
Bit5 ET2: TIMER2 Global interrupt enable bits;
1= Enable all interrupts of TIMER2;
0= All interrupts of TIMER2 are Disabled.
Bit4 ES0: UART0 interrupt enable bit;
1= Enable UART0 interrupts;
0= Disable UART0 Interrupt.
Bit3 ET1: TIMER1 interrupt enable bit;
1= Enable TIMER1 interrupts;
0= Disable TIMER1 Interrupt.
Bit2 EX1: External interrupt 1 interrupt enable bits;
1= Enable external interrupt 1 interrupt;
0= Disable external interrupt 1 interrupt.
Bit1 AND0: TIMER0 interrupt enable bit;
1= Enable TIMER0 interrupts;
0= Disable TIMER0 Interrupt.
Bit0 EX0: External interrupt 0 interrupt enable bit;
1= Enable external interrupt 0 interrupts;
0= Disable external interrupt 0 interrupt.

21.5.2 Interrupt Priority Control Register IP

0xB8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IP	--	--	PT2	PS0	PT1	PX1	PT0	PX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit6 -- Reserved, must be 0.
Bit5 PT2: TIMER2 interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.
Bit4 PS0: UART0 interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.
Bit3 PT1: TIMER1 interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.
Bit2 PX1: External interrupt 1 interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.
Bit1 PT0: TIMER0 interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.
Bit0 PX0: External interrupt 0 interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

21.5.3 Interrupt Priority Register EIP3

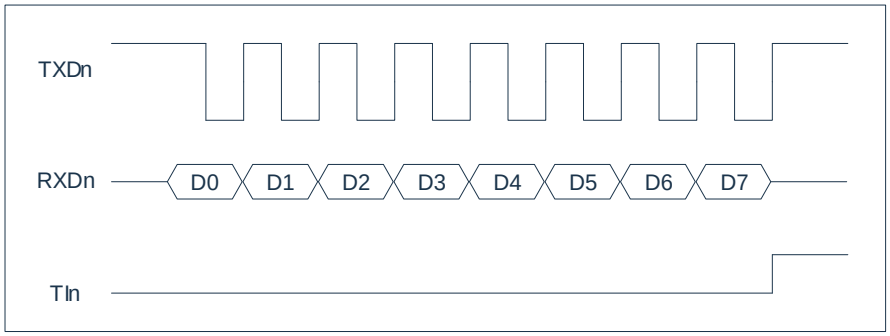
0xBB	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP3	--	--	--	--	PLVD	PLSE	--	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit4 -- Reserved, must be 0.
Bit3 PLVD: LVD interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.
Bit2 PLSE: LSE interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.
Bit1~Bit0 -- Reserved, must be 0.

21.6 UART0 Mode

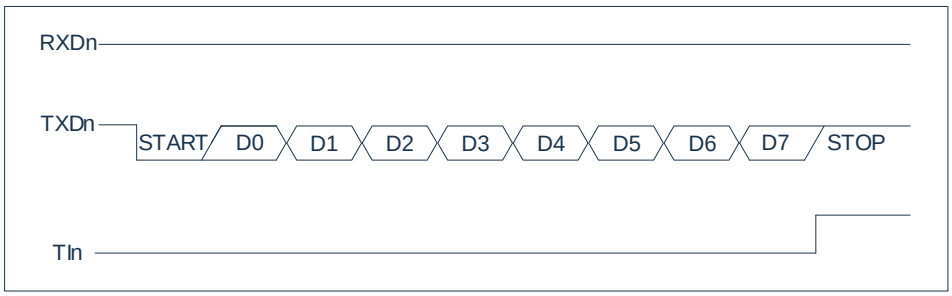
21.6.1 Mode 0 - Synchronous Mode

Pin RXD00 is the input or output and TXD0 is the clock output. The TXD0 output is a shift clock. The baud rate is fixed at 1/12 of the system clock frequency. 8 bits are transmitted preferentially with LSB. The receive is initialized by setting the flag in SCON 0, set to: RI0 = 0 and REN0 = 1. The mode 0 timing diagram is shown in the following figure:



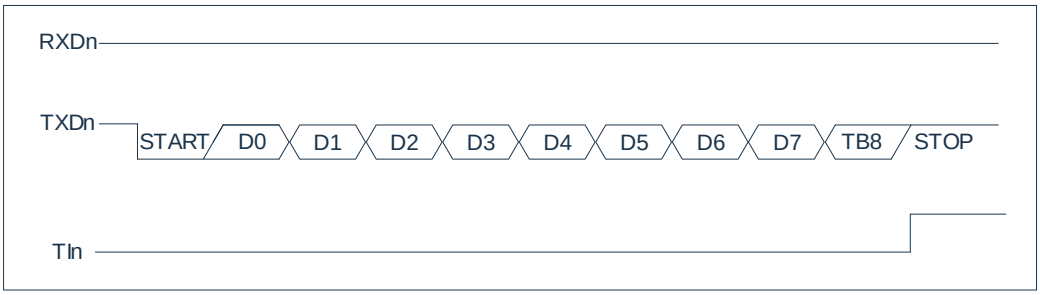
21.6.2 Mode 1-8 Bit Asynchronous Mode (Variable Baud Rate).

Pin RXD0 is used as the input and TXD0 is used as the serial output. Send 10 bits: start bit (always 0), 8 bits of data (LSB first), and stop bit (always 1). When received, the start bit is transmitted synchronously, 8 data bits can be obtained by reading SBUF 0, and the stop bit sets the flag RB0 8 in SCON 0. The baud rate is variable and depends on the TIMER1/TIMER2/TIMER4/BRT mode. The Mode 1 timing diagram is shown in the following figure:



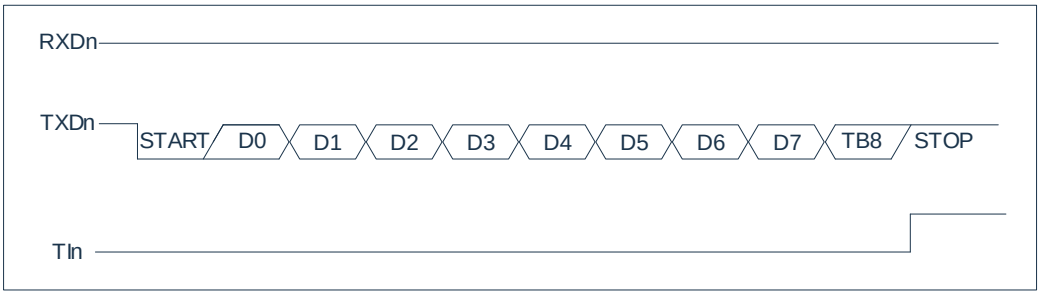
21.6.3 Mode 2-9 Bit Asynchronous Mode (Fixed Baud Rate).

This mode is similar to Mode 1, but differs in two ways. The baud rate is fixed at 1/32 or 1/64 of the CLK clock frequency, with 11 bits of transceiver: start bit (0), 8 bits of data (LSB first), programmable bit 9, and stop bit (1). Bit 9 can be used to control parity of the UART0 interface: bit TB08 in SCON0 acts as the 9th bit output when sending, and bit 9 affects RB08 in SCON0 when receiving. The mode 3 timing diagram is shown in the following figure:



21.6.4 Mode 3-9 Bit Asynchronous Mode (Variable Baud Rate)

The only difference between mode 2 and mode 3 is that the baud rate in mode 3 is variable. When REN0=1, data reception is enabled. The baud rate is variable and depends on the TIMER1/TIMER2/TIMER4/BRT mode. The Mode 4 timing diagram is shown in the following figure:



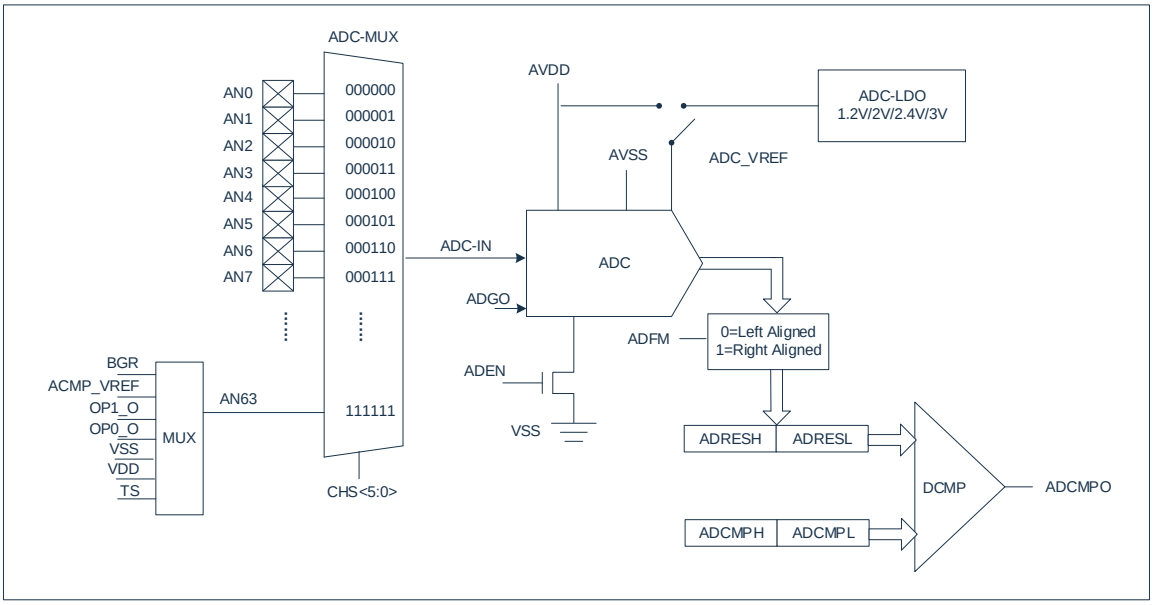
22. Analog-to-digital Converters (ADCs)

22.1 Overview

An analog-to-digital converter (ADC) converts an analog input signal into a 12-bit binary number representing the signal, as shown in the ADC block diagram below.

The port analog input signal and the internal analog signal are connected to the input of the analog-to-digital converter after being multiplexed. The analog-to-digital converter uses successive approximation to produce a 12-bit binary result and saves the result in the ADC result register (ADRESL and ADRSH), and the ADC can generate an interrupt after the conversion is complete. The ADC conversion results are compared to the values of the ADC comparison data registers (ADCMPL and ADCPH), which are stored in the ADCPO flag bits.

The ADC reference voltage is always internally generated and can be provided with avdd or by an internal ADC-LDO.



22.2 ADC Configuration

When configuring and using an ADC, the following factors must be considered:

- Port configuration.
- Channel selection.
- ADC converts the clock source.
- Interrupt control.
- The format in which the results are stored.

22.2.1 Port Configuration

ADC can convert both analog and digital signals. When converting an analog signal, the corresponding port needs to be configured as an analog port.

Note: Applying an analog voltage to a pin defined as a digital input may cause an overcurrent in the input buffer.

22.2.2 Channel Selection

The register ADCCHS bit determines which channel is connected to the analog-to-digital converter.

If you change the channel, you will need a delay before the next conversion starts. The ADC delay time is shown in the following table:

Delay time	Operating voltage
500ns	2.5~4.5V
200ns	4.5~5.5V

22.2.3 ADC Reference Voltage

The reference voltage of the ADC is provided by default by the chip's VDD or by the internal ADC-LDO. The ADC-LDO can choose from four voltage outputs: 1.2V/2.0 V/2.4 V/3.0 V.

22.2.4 Convert the Clock

The converted clock source can be selected by software setting the ADCKS bit of the ADCON1 register.

The time to complete a bit conversion is defined as T_{ADCK}. A full 12-bit conversion requires 18.5 T_{ADCK} cycles (the completion of a conversion ADGO lasts at a high time). The appropriate T_{ADCK} specification must be met to obtain the correct conversion results, and the following table is an example of the correct selection of an ADC clock.

Fsys	F_{ADCK} (T_A=25℃)		
	V_{REF}=V_{REF1}=AVDD (AVDD=VDD)	V_{REF}=V_{REF2}=1.2V	V_{REF}=V_{REF3}=2.0V V_{REF}=V_{REF4}=2.4V V_{REF}=V_{REF5}=3.0V
8MHz	F_{sys}/4	F_{sys}/256	F_{sys}/16
16MHz	F_{sys}/8	Disable	F_{sys}/32
24MHz	F_{sys}/16	Disable	F_{sys}/64
48MHz	F_{sys}/32	Disable	F_{sys}/128

Note: Any change in the system clock frequency will change the frequency of the ADC clock, which can negatively affect the ADC conversion results.

22.2.5 Result Format

The results of the 12-bit A/D conversion can be in two formats: left-aligned or right-aligned. The output format is controlled by the ADFM bit of the ADCON0 register.

When ADFM=0, the AD conversion result is left-aligned;

When ADFM=1, the AD conversion result is right-aligned.

22.3 The ADC Hardware Trigger Start

In addition to software-triggered ADC conversion, the ADC module provides a way for hardware to trigger start. One is the external port edge triggering method, and the other is the edge or periodic triggering mode of the PWM.

Using a hardware trigger ADC requires setting ADCX to 1, even if the ADC function can be triggered externally. The hardware trigger signal will set the ADGO bit to 1 after a certain delay, and will be automatically cleared after conversion. After the hardware trigger function is enabled, the software trigger function is not disabled, and the AD conversion can be initiated by writing 1 to the ADGO bit when the ADC is idle.

22.3.1 The External Port Edge Triggers the ADC

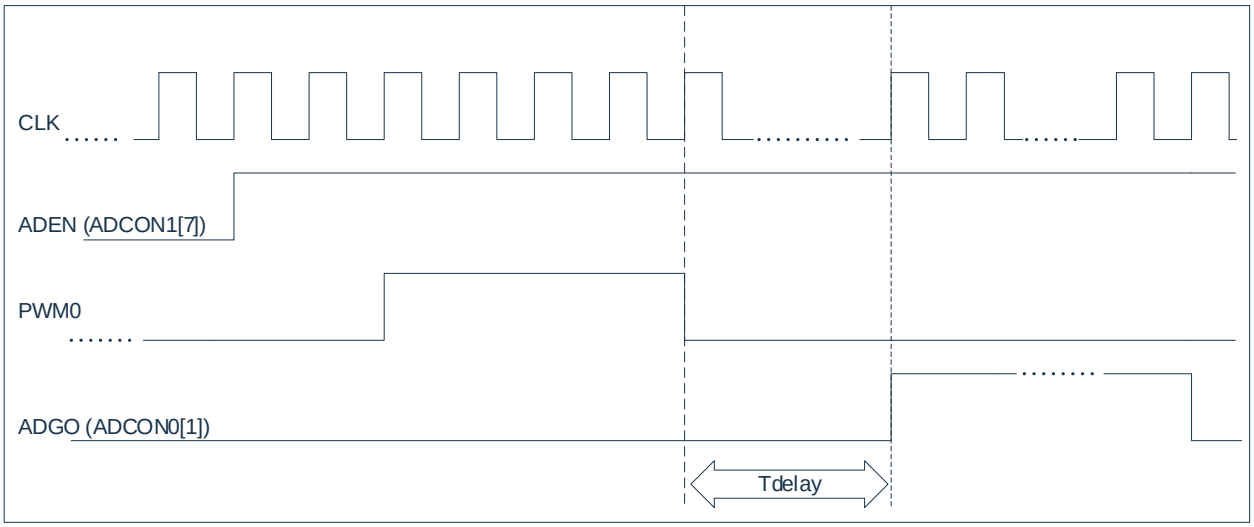
The ADET pin edge automatically triggers the ADC conversion. At this point, ADTGS [1:0] needs to be 11 (select external port edge trigger), ADEGS [1:0] can choose which edge trigger.

22.3.2 PWM Triggers the ADC

The PWM can optionally trigger the ADC conversion by the edge or the zero point. ADTGS [1:0] Select the PWM channel (PG0, PG2, PG4), and ADEGS [1:0] can select the triggering method of edge type or period type.

22.3.3 Hardware Trigger Start Delay

After the hardware trigger signal is generated, the AD conversion is not started immediately, and it takes a certain delay before the ADGO is set to 1. The delay is determined by ADDLY [9:0]. The delay time of the hardware trigger signal: (ADDLY+3)*Tsys, the block diagram of the delayed trigger structure is shown in the following figure:



22.4 ADC Results Comparison

The ADC module provides a set of digital comparators for comparing the results of an ADC with the value size of preloaded {ADCMPH, ADCMPL}. The result of each ADC conversion is compared to the preset value ADCMP, and the result of the comparison is stored in the ADCPO flag bit, which is automatically updated after the conversion is completed. The ADCMPPS bit can change the polarity of the output result.

The ADC comparison results trigger the enhanced PWM fault brake, which requires the ADFBEN to be set to 1.

When the enhanced PWM function is turned on, ADFBEN = 1, the result of the AD conversion is compared with the preset value {ADCMPH, ADCMPL}, if the result of the comparison ADCCMPO is 1, the PWM immediately generates a fault brake operation, clears the start bit of all PWM channels, and terminates all PWM channel outputs.

22.5 How the ADC Works

22.5.1 Start the Conversion

To enable the ADC module, you must first place 1 at ADEN bit of the ADCON1 register and then set 1 at ADGO bit of the ADCON0 register to start analog-to-digital conversion (ADGO cannot be set to 1 when ADEN is 0).

22.5.2 Complete the Conversion

When the conversion is complete, the ADC module will:

- Zero ADGO bit;
- Place the ADCIF flag at bit 1;
- Update the ADRESH:ADRESL register with the new result of the conversion.

22.5.3 Terminate the Conversion

If the conversion must be terminated before it is complete, the analog-to-digital conversion results that have not yet completed are not updated to the ADRSH:ADRESL register. Therefore, the ADRESH:ADRESL register will maintain the value obtained by the last conversion.

Note: A device reset forces all registers to enter a reset state. Therefore, the reset shuts down the ADC module and terminates any pending transitions.

22.5.4 A/D Conversion Steps

The configuration steps for analog-to-digital conversion using an ADC are as follows:

1) Port configuration:
   - Disable pin output drivers (see PxTRIS registers);
   - Configure the pins as analog input pins.
2) Configure ADC interrupt (optional):
   - Clear the ADC interrupt flag bit;
   - Enable ADC interrupts;
   - Enable peripheral interrupts;
   - Enable global interrupts.
3) Configure the ADC module:
   - Select the ADC conversion clock;
   - Select the ADC input channel;
   - Select the format of the result;
   - Start the ADC module.
4) Wait for the required acquisition time.
5) Set ADGO to 1 to start the conversion.
6) Wait for the ADC conversion to finish by one of the following methods:
   - Query ADGO bits;
   - Wait for the ADC to interrupt (interrupt enabled).
7) Read the ADC results.
8) Zero out the ADC interrupt flag bit (this is required if Interrupts are enabled).

Note: If the user attempts to resume sequential code execution after waking the device from sleep mode, the global interrupt must be suppressed.

22.5.5 Go to sleep during the conversion process

When the system enters hibernation, it is recommended that you wait for the ADC to complete the transition in progress before entering hibernation.

If you go hibernation during an ongoing conversion of the ADC, the conversion is terminated. The conversion operation needs to be restarted after waking up.

22.6 Related Registers

There are 11 main registers associated with AD conversion, which are:

- AD control registers ADCON0, ADCON1, ADCON2, ADCCHS, ADCLDO;
- Comparator control register ADCPC;
- Delay data register ADDLYL;
- AD result data register ADRSH/L;
- Comparator data register ADCMPH/L.

22.6.1 AD Control Register ADCON0

0xDF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON0	--	ADFM	ANACH3	ANACH2	ANACH1	ANACH0	ADGO	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 -- Reserved, must be 0.

Bit6 ADFM: ADC conversion result format select bits;
1= Right alignment;
0= Left-aligned.

Bit5~Bit2 ANACH<3:0>: ADC channel 63 input source select bit;
0000= BGR(1.2V);
0001= ACMP_VREF (negative reference voltage of the comparator, see ACMP section for details);
0010= Reserved, prohibited use;
0011= Reserved, prohibited use;
0100= TS_ANA (temperature sensing voltage);
0101= VSS (ADC Reference Area);
0110= Reserved, prohibited use;
0111= VDD (ADC default reference voltage).
Other = Reserved, prohibited.

Bit1 ADGO: ADC converts the start bit (ADEN must be 1 for this bit 1, otherwise the operation is invalid);
1= Write: Starts the ADC conversion, (the hardware also uses this bit 1 when the ADC is triggered);
Read: The ADC is converting.
0= Write: Invalid.
Read: ADC idle/converted;
During the conversion of AD C (ADGO=1), any software and hardware trigger signals are ignored.

Bit0 -- Reserved, must be 0.

22.6.2 AD Control Register ADCON1

0xDE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON1	ADEN	ADCKS2	ADCKS1	ADCKS0	--	--	--	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	1	0	0	0	0	0	0

Bit7 ADEN: ADC enable bit;
1= Enable ADC;
0= ADC is Disabled and does not consume operating current.

Bit6~Bit4 ADCKS<2:0>: ADC conversion clock select bits;
000= Fsys/2; 100= Fsys/32;
001= Fsys/4; 101= Fsys/64;
010= Fsys/8; 110= Fsys/128;
011= Fsys/16; 111= Fsys/256。

Bit3~Bit0 -- Reserved, must be 0.

22.6.3 AD Control Register ADCON2

0xE9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON2	ADCEX	--	ADTGS1	ADTGS0	ADEGS1	ADEGS0	--	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 ADCEX: ADC hardware trigger enable bit;
1= Enable;
0= Disable.

Bit6 -- Reserved, must be 0.

Bit5~Bit4 ADTGS<1:0>: ADC hardware trigger source select bit;
00= PG0 (PWM0);
01= PG2 (PWM2);
10= --
11= Port pin (ADET).

Bit3~Bit2 ADEGS<1:0>: ADC hardware trigger edge selection bit;
00= Descending edges;
01= Rising edge;
10= The periodic point of the PWM cycle;
11= The zero point of the PWM cycle.

Bit1~Bit0 -- Reserved, must be 0.

22.6.4 AD Channel Selection Register ADCCHS

0xD9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCCHS	--	--	CHS5	CHS4	CHS3	CHS2	CHS1	CHS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 -- Reserved, must be 0.

Bit5~Bit0 CHS<5:0>: Analog channel selection bits;

000000=	AN0;	010000=	AN16;	100000=	
000001=	AN1;	010001=	AN17;	100001=	
000010=	AN2;	010010=	AN18;	100010=	
000011=	AN3;	010011=	AN19;	100011=	
000100=	AN4;	010100=	AN20;	100100=	
000101=	AN5;	010101=	AN21;	100101=	
000110=	AN6;	010110=	AN22;	100110=	
000111=	AN7;	010111=	AN23;	100111=	Prohibited access;
001000=	AN8;	011000=	AN24;	101000=	
001001=	AN9;	011001=	AN25;	101001=	
001010=	AN10;	011010=		101010=	
001011=	AN11;	011011=		101011=	
001100=	AN12;	011100=	Prohibited access;	101100=	
001101=	AN13;	011101=		101101=	
001110=	AN14;	011110=		Other =	
001111=	AN15;	011111=		111111=	See ADCON0.ANACH description.

22.6.5 AD Comparator Control Register ADCPC

0xD1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCMPC	ADFBEN	ADCMPPS	--	ADCMPO	--	--	ADDLY9	ADDLY8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 ADFBEN: ADC comparator result control PWM brake enable bit;
1= Enable;
0= Disable.

Bit6 ADCMPPS: ADC comparator output polarity select bit;
1= If ADRES < ADCMP, ADCPO=1;
0= If ADRES > =ADCMP, ADCPO=1.

Bit5 -- Reserved, must be 0.

Bit4 ADCMPO: ADC comparator output bits.
This bit outputs the result of the ADC comparator output, which is updated each time the ADC conversion ends.

Bit3~Bit2 -- Reserved, must be 0.

Bit1~Bit0 ADDLY<9:8>: ADC hardware trigger delay data [9:8] bits.

22.6.6 AD Hardware Trigger Delay Data Register ADDLYL

0xD3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADDLYL	ADDLY7	ADDLY6	ADDLY5	ADDLY4	ADDLY3	ADDLY2	ADDLY1	ADDLY0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 ADDLY<7:0>: ADC hardware trigger delay data is 8 bits lower.

22.6.7 AD Data Register High ADRESH, ADFM=0 (Left-aligned).

0xDD	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADRESH	ADDRESS11	ADDRESS10	ADDRESS9	ADDRESS8	ADDRESS7	ADDRESS6	ADDRESS5	ADDRESS4
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 ADDRESS<11:4>: ADC result register bit.
1Bit 1 2 conversion result of bits 11-4.

22.6.8 AD Data Register Low ADRSL, ADFM=0 (Left-aligned).

0xDC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADRESL	ADDRESS3	ADDRESS2	ADDRESS1	ADDRESS0	--	--	--	--
R/W	R	R	R	R	--	--	--	--
Reset value	X	X	X	X	--	--	--	--

Bit7~Bit4 ADDRESS<3:0>: ADC result register bit.
1Bits 3-0 of the 2-bit conversion result.
Bit3~Bit0 Unused.

22.6.9 AD Data Register High ADRESH, ADFM=1 (Right-aligned).

0xDD	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADRESH	--	--	--	--	ADDRESS11	ADDRESS10	ADDRESS9	ADDRESS8
R/W	--	--	--	--	R	R	R	R
Reset value	--	--	--	--	X	X	X	X

Bit7~Bit4 Unused.
Bit3~Bit0 ADDRESS<11:8>: ADC result register bit.
1Bits 11-8 of the 2-bit conversion result.

22.6.10 AD Data Register Low ADRSL, ADFM = 1 (Right-aligned).

0xDC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADRESL	ADDRESS7	ADDRESS6	ADDRESS5	ADDRESS4	ADDRESS3	ADDRESS2	ADDRESS1	ADDRESS0
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 ADDRESS<7:0>: ADC result register bit.
12 bits converted to bits 7-0 of the result.

22.6.11 AD Comparator Data Register ADCMPH

0xD5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCMPH	D11	D10	D9	D8	D7	D6	D5	D4
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	1	1	1	1	1	1	1	1

Bit7~Bit0 ADCMP<11:4>: The ADC comparator data is 8 bits high.

22.6.12 AD Comparator Data Register ADCMPL

0xD4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCMPL	--	--	--	--	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	1	1	1	1	1	1	1	1

Bit7~Bit4 Unused.
Bit3~Bit0 ADCMP<3:0>: The ADC comparator data is 4 bits lower.

22.6.13 AD Reference Voltage Control Register

F692H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCLDO	LDOEN	VSEL1	VSEL0	LDOOUTEN	--	--	--	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 LDOEN ADC_LDO enabled;
1= LDO enable, the reference voltage can only select the voltage corresponding to VSEL [1:0];
0= LDO prohibits, the reference voltage is the chip supply voltage.
Bit6~Bit5 VSEL<1:0>: ADC reference voltage selection bit;
00= 1.2V ;
01= 2.0V ;
10= 2.4V ;
11= 3.0V 。
Bit4 LDOOUTEN The LDO output is enabled.
1= Enable output (P16 output)
0= Disable output
Bit3~Bit0 -- Reserved, must be 0.

22.7 ADC Interrupt

The ADC module allows an interrupt to be generated after the analog-to-digital conversion is complete. The ADC interrupt enable bit is the ADCIE bit in the EIE2 register, and the ADC interrupt flag bit is the ADCIF bit in the EIF2 register. The ADCIF bit must be cleared with software, and the ADCIF bit is set to 1 after each conversion, regardless of whether the ADC interrupt is enabled. The interrupt enable and priority of the ADC can be set by the following relevant register bits.

22.7.1 Interrupt Mask Register EIE2

0xAA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIE2	SPIIE	I2CIE	WDTIE	ADCIE	PWMIE	--	ET4	ET3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 SPIIE: SPI interrupt enable bit;
1= Enable SPI interrupts;
0= Disable SPI Interrupt.
Bit6 I2CIE: I2C Interrupt enable bit;
1= Allow I2C interrupts;
0= Disable I2C Interrupt.
Bit5 WDTIE: WDT interrupt enable bit;
1= Enable WDT overflow interrupts;
0= Disable WDT overflow interrupts.
Bit4 ADCIE: ADC interrupt enable bit;
1= Enable ADC interrupts;
0= Disable ADC interrupts.
Bit3 PWMIE: PWM global interrupt enable bit;
1= Enable all PWM interrupts;
0= Disable all PWM interrupts.
Bit2 --: Reserved, must be 0.
Bit1 ET4: Timer4 interrupt enable bit;
1= Enable Timer4 interrupts;
0= Disable Timer4 Interrupt.
Bit0 ET3: Timer3 interrupt enable bit;
1= Enable Timer3 interrupts;
0= Disable Timer3 Interrupt.

22.7.2 Interrupt Priority Control Register EIP2

0xBA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP2	PSPI	PI2C	PWDT	PADC	PPWM	--	PT4	PT3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 PSPI: SPI interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

Bit6 PI2C: I2C Interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

Bit5 PWDT: WDT interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

Bit4 PADC: ADC interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

Bit3 PPWM: PWM interrupt priority control bit
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

Bit2 --: Reserved, must be 0.

Bit1 PT4: TIMER4 interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

Bit0 PT3: TIMER3 interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

22.7.3 Peripheral Interrupt Flag Bit Register EIF2

0xB2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIF2	SPIIF	I2CIF	--	ADCIF	PWMIF	--	TF4	TF3
R/W	R	R	--	R/W	R	--	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 SPIIF: SPI global interrupt indicator bit, read-only;
1= SPI generates an interrupt, (this bit is automatically cleared after the specific interrupt flag is cleared);
0= The SPI did not produce an interrupt.
Bit6 I2CIF: I2C global interrupt indicator bit, read-only;
1= I2C produces an interrupt, (after clearing the specific interrupt flag, this bit is automatically cleared);
0= I2C did not produce an interrupt.
Bit5 -- Reserved, must be 0.
Bit4 ADCIF: ADC interrupt flag bit;
1= ADC conversion is completed, and software zeroing is required;
0= The ADC conversion was not completed.
Bit3 PWMIF: PWM global interrupt indicator bit, read-only;
1= PWM generates an interrupt, (after the specific interrupt flag is cleared, this bit is automatically cleared);
0= The PWM did not produce an interrupt.
Bit2 -- Reserved, must be 0.
Bit1 TF4: Timer4 timer overflow interrupt flag bit;
1= Timer4 timer overflow, the hardware is automatically cleared when entering the interrupt service program, and the software can also be cleared;
0= The Timer4 timer has no overflow.
Bit0 TF3: Timer3 timer overflow interrupt flag bit;
1= Timer3 timer overflow, when entering the interrupt service program, the hardware is automatically cleared, and the software can also be cleared;
0= The Timer3 timer has no overflow.

23. Temperature Sensor

23.1 Overview

The chip contains a temperature sensor whose output analog volume varies with the temperature of the chip. The analog signal output by the sensor is acquired and converted by ADC, and temperature changes can be obtained indirectly.

23.2 Register Description

23.2.1 The Temperature Sensor Control Register TS _REG

0xF693	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TS_REG	TS_EN	TS_SEL	--	--	TS_TRIM3	TS_TRIM2	TS_TRIM1	TS_TRIM0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7	TS_EN:	Temperature sensor enable bit;
	1=	Enable;
	0=	Disable.
Bit6	TS_SEL:	Temperature sensor trimming and tuning selection;
	1=	Select register configuration;
	0=	Select config configuration.
Bit5~Bit4	--	Reserved, must be 0.
Bit3~Bit0	TS_TRIM<3:0>:	Temperature sensor registers are configured to trim the tuning.

23.3 Feature Description

23.3.1 Configuration

The temperature sensor generates a corresponding analog amount (TS_ANA) as the chip temperature changes, which can be acquired by the ADC and the value of the ADC conversion result register is read indirectly to obtain the chip temperature condition. The temperature sensor usage configuration is as follows:

1) Set the temperature sensor detection enable TS_REG[7] = 1.
2) Set the temperature sensor trimming selection and trimming value.
3) Set the temperature sensing channel ADCCHS[5:0] = 111111 and ADCON0[5:2] = 0100.
4) Set the ADC-related configuration and turn on the ADC translation enable.
5) Wait for the ADC conversion to complete and read the register data.

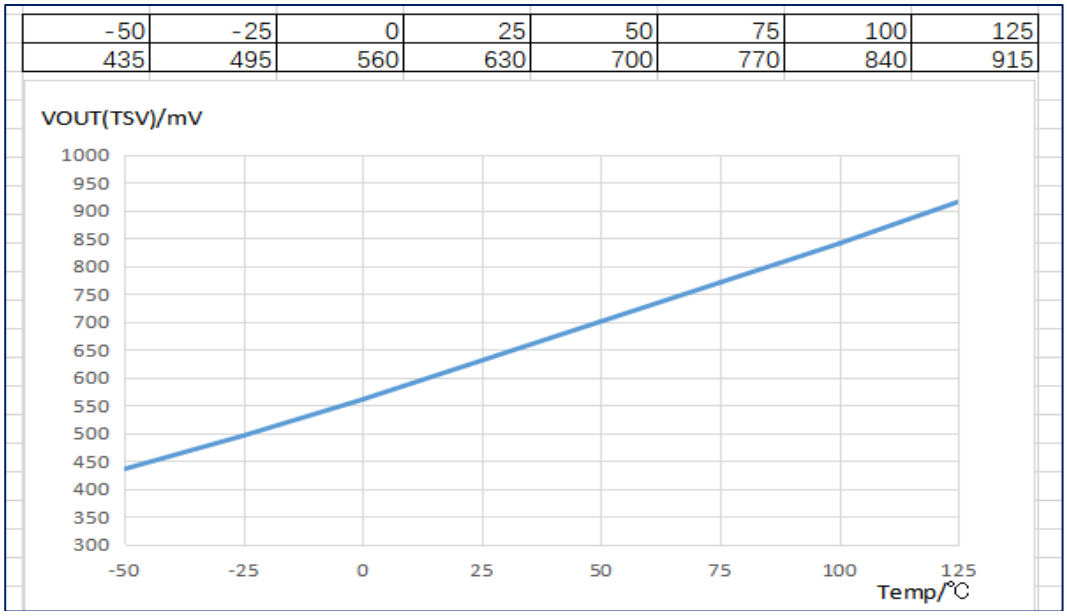
23.3.2 Features

This temperature sensor has the following characteristics:

1) When the temperature changes from - 40℃ to 125℃, the analog signal voltage range generated is: 0.7V~1.4V;
2) When the temperature changes from - 40℃ to 125℃, the slope of analog quantity with temperature is K: 3.5 ± 0.2 mV/℃;
3) When the temperature is 25℃, the voltage value is: 1±0.01V;
4) The analog signal quantity and temperature approximate linear change process, and the change curve is shown in the figure below. In the figure below, T_A is the temperature (unit:℃), and V_{TS} is the output voltage of temperature sensor (unit:V).

The temperature sensor output analog volume varies with temperature as shown in the following figure:

-50	-25	0	25	50	75	100	125
435	495	560	630	700	770	840	915



23.3.3 Calculation Formula

The conversion result of temperature sensor is as follows:

$$RES_{AD}=\frac{V_{TS}}{V_{REF}}\times 4096$$

In the case, RES_{AD} is the measured 12-bit AD conversion value, V_{REF} is the reference voltage of ADC (unit:V), V_{TS} is the output voltage of analog signal (unit:V), ΔV is the output voltage of analog signal.

$$T=\left(\frac{V_{VREF}\times RES_{AD}}{4.096}-0.909\right)\div K$$

In the case, K is the slope of analog quantity change with temperature (unit:V/℃).

24. Analog Comparator (ACMP0/1).

The chip contains two analog comparators, ACMP0 and ACMP1. When the positive voltage is greater than the negative voltage, the comparator outputs logic 1 and vice versa output 0, which can also be changed by the output polarity select bit. When the comparator output value changes, each comparator can generate an interrupt.

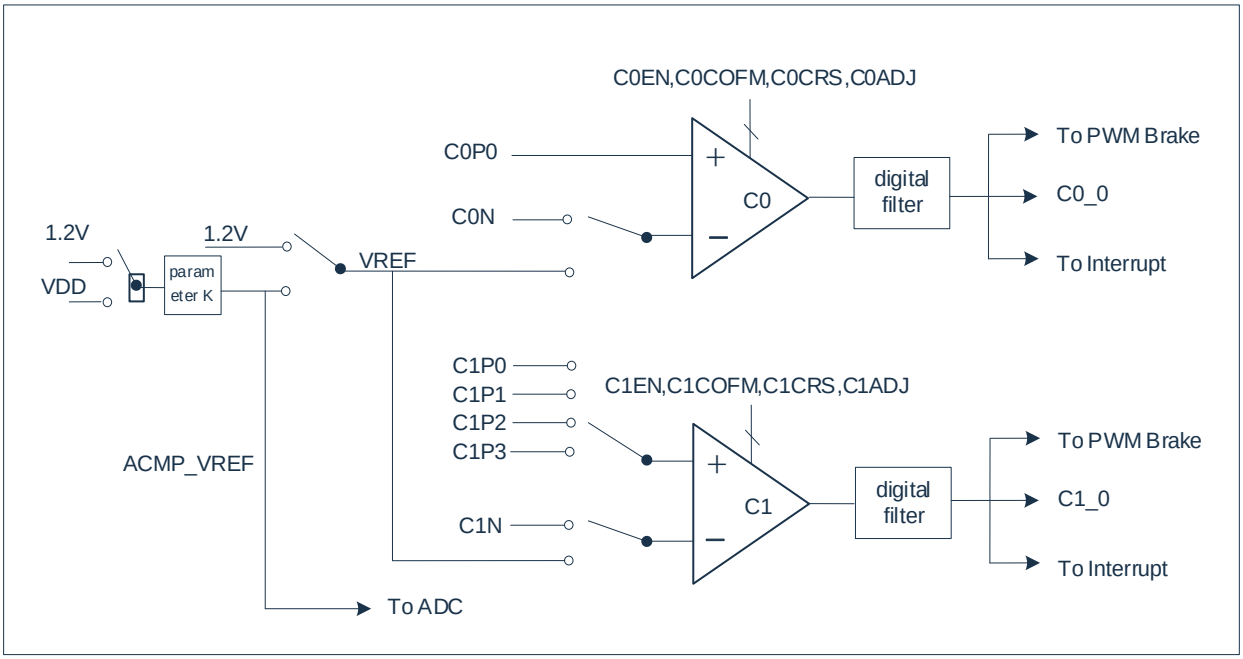
24.1 Comparator Features

The comparator has the following characteristics:

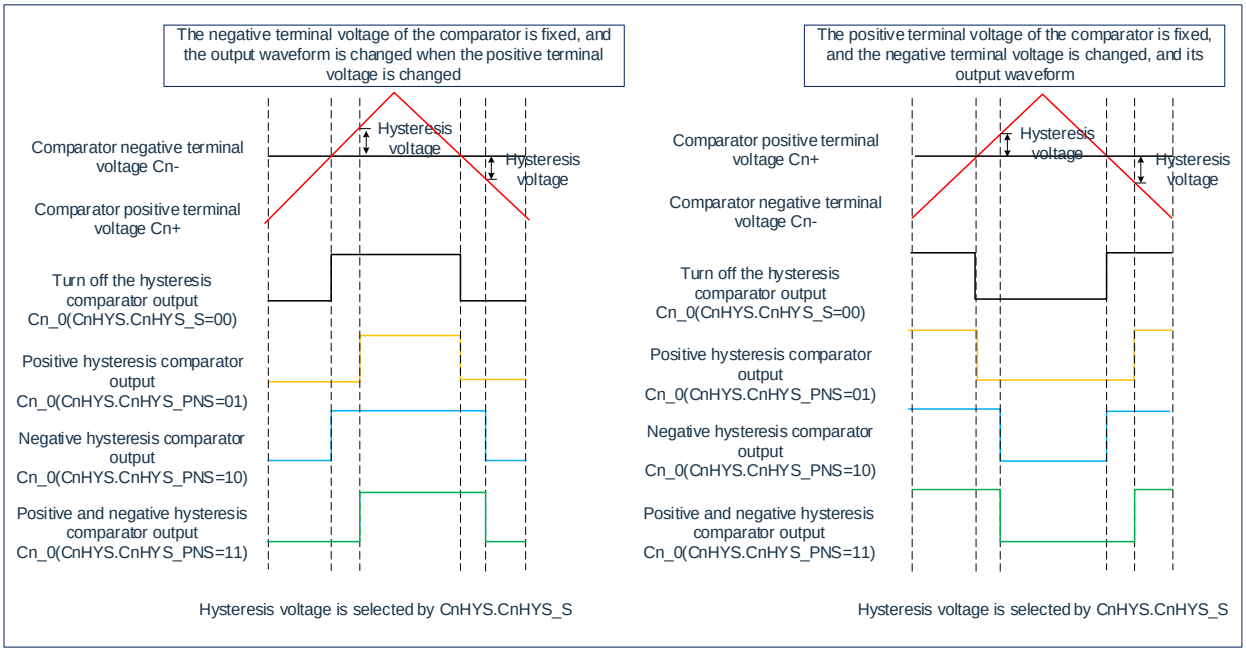
- The ACMP0 positive side can select 1 port input, and the ACMP1 positive side can select 4 port inputs.
- The negative side selects the port input with the internal reference voltage VREF.
- The internal reference voltage selects the internal Bandgap (1.2V) and ACMP_VREF outputs.
- ACMP_VREF reference source voltage dividing range: gear selectable.
- Output filterable time can be selected: 0 ~ 512 * Tsys.
- Supports unilateral (positive/negative) and bilateral (plus/negative) hysteresis control.
- Hysteresis voltage is available at 10/20/60mV.
- The software supports offset voltage trimming.
- The output acts as a brake trigger signal for the enhanced PWM.
- Output changes can produce interrupts.

24.2 Comparator Structure

The comparator block diagram is shown in the following figure:



The comparator hysteresis control block diagram is shown in the following figure:



24.3 Related Registers

24.3.1 Comparator Control Register CnCON0

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CnCON0	Cnen	CnCOFM	CnN2G	CnNS1	CnNS0	CnPS2	CnPS1	CnPS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

C0CON0 address:F500H;C1CON0 address:F503H。

Bit7 CnEN: Comparator n enable bit;
1= Enable;
0= Disable.

Bit6 CnCOFM: Comparator n mode enable bit;
1= Enable adjustment mode;
0= Disables the mode of adjustment.

Bit5 CnN2G: Comparator n adjustment mode negative ground enable bit (this bit is active at CnCRS=0).
1= The negative end channel is closed and the internal negative end is grounded;
0= The negative channel is enabled and the signal is input from the negative side.

Bit4~Bit3 CnNS<1:0>: Comparator n negative channel selection bit;
00= Comparator n negative port
01= Internal voltage (Bandgap or ACMP_VREF);
1x= Reserved, prohibited.

Bit2~Bit0 CnPS<2:0>: Comparator n-positive channel select bit CnPS< 2:0>;
000= CnP0;
001= CnP1;
010= CnP2;
011= CnP3;
1xx= Reserved, prohibited.

24.3.2 Comparator Control Register CnCON1

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CnCON1	CnOUT	CnCRS1	CnCRS0	CnADJ4	CnADJ3	CnADJ2	CnADJ1	CnADJ0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	1	0	0	0	0

C0CON1 Address: F501H; C1CON1 Address: F504H.

Bit7 CnOUT: Comparator n result bits, read-only;

Bit6~Bit5 CnCRS[1:0]: Comparator n adjustment mode input selection;
00= The positive and negative ends are connected together, connecting the N channel;
01= The positive and negative ends are connected together, connecting GND;
1x= The positive and negative ends are connected together to connect the P channel.

Bit4~ Bit0 CnADJ<4:0>: Comparator n offset voltage regulation bit.

24.3.3 Comparator Control Register CnCON2

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CnCON2	--	--	CnPOS	CnFE	CnFS3	CnFS2	CnFS1	CnFS0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

C0CON2 Address: F502H; C1CON2 Address: F505H.

Bit7~Bit6 -- Reserved, must be 0.

Bit5 CnPOS: Comparator n output polarity select bit (bits that may cause interrupt flag bit when switching);

1= Inverting output;

0= Normal output.

Bit4 CnFE: Comparator n output filter enable bit;

1= Enable filtering;

0= Filtering is Disabled.

Bit3~Bit0 CnFS<3:0>: Comparator n output filter time selection bit;

0000= (0~1)*Tsys;

0001= (1~2)*Tsys;

0010= (2~3)*Tsys;

0011= (4~5)*Tsys;

0100= (8~9)*Tsys;

0101= (16~17)*Tsys;

0110= (32~33)*Tsys;

0111= (64~65)*Tsys;

1000= (128~129)*Tsys;

1001= (256~257)*Tsys;

1010= (512~513)*Tsys;

Thether= (0~1)*Tsys。

24.3.4 Comparator Adjustment Bit Selection Register CnADJE

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CnADJE	CnADJE7	CnADJE6	CnADJE5	CnADJE4	CnADJE3	CnADJE2	CnADJE1	CnADJE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

C0ADJE Address: F50AH; C1ADJE address: F50BH.

Bit7~Bit0 CnADJE<7:0>: Comparator n offset voltage regulation mode selection;

AAH = Determined by the CnADJ < 4:0 > in the CnCON1 register;

Other = Determined by the CONFIG correlation bit.

It is recommended to set the parameters of the comparator before starting the comparator, otherwise it may occur that the comparator output jump is detected by mistake during the setup process.

24.3.5 Comparator Hysteresis Control Register CnHYS

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CnHYS	--	--	--	--	CnHYS_PNS1	CnHYS_PNS0	CnHYS_S1	CnHYS_S0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

C0HYS Address: F50CH; C1HYS Address: F50DH.

Bit7~Bit4	--	Reserved, must be 0.
Bit3~Bit2	CnHYS_PNS<1:0>	Positive and negative hysteresis select bits;
	00=	Off hysteresis;
	01=	positive hysteresis (unilateral hysteresis);
	10=	Negative hysteresis (unilateral hysteresis);
	11=	Positive and negative hysteresis (bilateral hysteresis).
Bit1~Bit0	CnHYS_S<1:0>	Hysteresis control bit;
	00=	Off hysteresis;
	01=	10mV;
	10=	20mV;
	11=	60mV。

24.3.6 Comparator Reference Voltage Control Register CNVRCON

F506H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CNVRCON	CNDIVS	CNSVR	CNVS5	CNVS4	CNVS3	CNVS2	CNVS1	CNVS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7	CNDIVS:	ACMP_VREF reference source selection bit;	
	1=	Select 1.2V (Bandgap) for the partial voltage;	
	0=	Select VDD for voltage division.	
Bit6	CNSVR:	Comparator negative internal voltage VREF select bit;	
	1=	Select ACMP_VREF (voltage divider circuit on, independent of the comparator module);	
	0=	Select 1.2V (Bandgap).	
Bit5~Bit4	CNVS<5:4>	ACMP_VREF reference voltage selection bit	
		CNDIVS=0	CNDIVS=1
	00=	[CNVS<3:0>+1]/60*VDD	[CNVS<3:0>+1]/60*1.2V
	01=	0. 5*VDD	0. 5*1.2V
	10=	0. 75*VDD	0. 75*1.2V
	11=	1*VDD	1*1.2V
Bit5~Bit0	CNVS<3:0>	CNVS< 5:4>=00 ACMP_VREF reference partial pressure selection bit	

24.3.7 Comparator Brake Control Register CNFBCON

F507H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CNFBCON	C1FBPEN	C0FBPEN	C1FBPS	C0FBPS	C1FBEN	C0FBEN	C1FBLS	C0FBLS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 C1FBPEN: Comparator 1 output level controls PWM brake enable;
0= Disable;
1= Enable.
Bit6 C0FBPEN: Comparator 0 output level control PWM brake enable bit;
0= Disable;
1= Enable.
Bit5 C1FBPS: Comparator 1 output controls PWM brake level selection;
0= High level;
1= Low level.
Bit4 C0FBPS: Comparator 0 output controls PWM brake level selection bit;
0= High level;
1= Low level.
Bit3 C1FBEEN: Comparator 1 output edge controls PWM brake enable;
0= Disable;
1= Enable.
Bit2 C0FBEEN: Comparator 0 output edge controls PWM brake enable;
0= Disable;
1= Enable.
Bit1 C1FBES: Comparator 1 output controls PWM brake edge selection;
0= Rising edge;
1= Falling edge.
Bit0 C0FBES: Comparator 0 output controls PWM brake edge selection;
0= Rising edge;
1= Falling edge.

24.4 Comparator Interrupt

Both comparator 0 and comparator 1 can set interrupts, both of which share an interrupt vector entry, and when entering the interrupt service program, the user can determine which type of interrupt is generated by the interrupt flag. Comparator interrupt priority and interrupt enable can be set by the following Related Register bits.

24.4.1 Interrupt Priority Control Register EIP1

0xB9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP1	PACMP	--	--	--	PP3	PP2	PP1	PP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 PACMP: Analog comparator interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.
Bit6~Bit4 -- Reserved, must be 0.
Bit3 PP3: P3 port interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.
Bit2 PP2: P2 port interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.
Bit1 PP1: P1 port interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.
Bit0 PP0: P0 port interrupt priority control bit;
1= Set to High-level Interrupt;
0= Set to low-level interrupt.

24.4.2 Comparator Interrupt Mask Register CNIE

F508H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CNO	--	--	--	--	--	--	C1IE	C0IE
R/W	R	R	R	R	R	R	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit2 -- Reserved, must be 0.
Bit1 C1IE: Comparator 1 interrupt enable bit;
0= Disable;
1= Enable.
Bit0 C0IE: Comparator 0 interrupt enable bit;
0= Disable;
1= Enable.

24.4.3 Comparator Interrupt Flag Register CNIF

F509H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CNIF	--	--	--	--	--	--	C1IF	C0IF
R/W	R	R	R	R	R	R	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~ Bit2 -- Reserved, must be 0.

Bit1 C1IF: Comparator 1 interrupt flag bit (write 0 clear);

1= Compare 1 output changes.

0= --

Bit0 C0IF: Comparator 0 interrupt flag bit (write 0 clear);

1= Compare 0 output changes.

0= --

25. Flash Memory

25.1 Overview

FLASH memory contains program memory (APROM/BOOT) and nonvolatile data memory (Data FLASH). The maximum program memory space is 16KB, divided into 32 sectors, each containing 512B. The maximum data memory space is 1KB, divided into 2 sectors, each containing 512B.

The FLASH memory can be accessed via the associated Special Function Register (SFR) for IAP functionality, and the Special Function Register (SFR) can be used CRC validation can also be performed on the program space. The SFR registers used to access the FLASH space are as follows:

- MLOCK
- MDATA
- MADRL
- MADRH
- PCRCDL
- PCRCDH
- MCTRL

MLOCK registers are used to enable memory operation, MDATA registers form a byte to hold 8 bits of data to be read/written, MADRL/MADRH registers hold the address of the accessed MDATA unit or the address of CRC checksum, PCRCDL/PCRCDH registers are used to keep the program CRC running results, and MCTRL registers are used for memory operation control.

The memory module interface allows the memory to be read/written/erased. The memory allows byte reads and writes, and the write time is controlled by the on-chip timer, which ensures that the data in that address has been erased before writing new data. The write and erase voltages are generated by an on-chip charge pump that is rated for operation within the voltage range of the device for byte operation.

The Flash memory erase operation only supports sector erasure, not byte erasure. Before modifying the data for an address, it is recommended that you save other data before erasing the current sector and finally writing the data.

The chip supports CRC checking of program space codes using the polynomial CRC16-CCITT '$X^{16}+X^{12}+X^{5}+1$' to build.

25.2 Related Registers

25.2.1 FLASH Protect Lock Register MLOCK

0xFB	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MLOCK	MLOCK7	MOCK6	MLOCK5	MLOCK4	MLOCK3	MLOCK2	MLOCK1	MLOCK0
R/W	In	In	In	In	In	In	In	In
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 MLOCK<7:0>: Memory operation enable bit (this register only supports write operations, reads are 00H);
AAH= Allows memory-dependent R/W/E operation;
00H/FFH= Operation is not allowed;
Other = Writes are prohibited.

Modify the sequence of instructions required by MLOCK (no other instructions can be inserted in the middle):

```
MOV     TA,#0AAH
MOV     TA,#055H
MOV     MLOCK,#0AAH
```

25.2.2 FLASH Memory Data Register MDATA

0xFE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MDATA	MDATA7	MDATA6	MDATA5	MDATA4	MDATA3	MDATA2	MDATA1	MDATA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 MDATA<7:0>: Data that is read or written to program memory.

25.2.3 FLASH Memory Address Register MADRL

0xFC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MADRL	MADRL7	MADRL6	MADRL5	MADRL4	MADRL3	MADRL2	MADRL1	MADRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	1	1	1	1	1	1

Bit7~Bit0 MADRL<7:0>: Specifies that the address of the memory read/write operation is 8 bits lower.

25.2.4 FLASH Memory Address Register MADRH

0xFD	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MADRH	MADRH7	MADRH6	MADRH5	MADRH4	MADRH3	MADRH2	MADRH1	MADRH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 MADRH<7:0>: Specifies that the address of the memory read/write operation is 8 bits high.

25.2.5 Program CRC Operation Result Data Register Lower 8-bit PCRCDL

0xF9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCRCDL	PCRCD<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 PCRCD<7:0> The program CRC operation results 8 bits lower data

25.2.6 Program CRC Operation Result Data Register Higher 8-bit PCRCDH

0xFA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCRCDH	PCRCD<15:8>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 PCRCD<15:8> The program CRC operation results in 8 bits of data high

25.2.7 FLASH Memory Control Register MCTRL

0xFF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MCTRL	--	--	TAKE	MREG	MMODE1	MMODE0	CRCADR	MSTART
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	1	0	0	1	0	0	0	0

Bit7~Bit6 -- Retain.

Bit5 TAKE: Operation error flag bit (write 0 cleared);

1= Before the programming operation begins, the data in the detection programming address is not "FFH" (not erased) and the write operation terminates immediately.

0= --

Bit4 MREG: Flash area selection bits;

1= Select the data area (low 10-bit address is valid);

0= Select the program area (low 1 4-bit address valid).

Bit3~Bit2 MMODE<1:0>: Operating mode selection bit:

11= Erase operation mode (the scope of the erase operation is: the entire sector where the current address is located);

10= Write operation mode;

01= CRC mode;

00= Read mode of operation.

Bit1 CRCADR: Program CRC checksum address selection bit;

1= End address select bit;

0= Select the bit for the start address.

Bit0 MSTART: Operation start control bit;

1= Start the program memory R/W/E/CRC check operation (after the operation is completed, it can be automatically cleared by the hardware);

0= Write: Terminate or do not start the program memory R/W/E/CRC check operation;
Read: The operation completed or the operation did not start.

Note: The CRCADR must be cleared after the CRC is finished.

25.3 Feature Description

During FLASH memory read/write/erase operation, the CPU is in a paused state, when the operation completes, the CPU continues to run instructions.

The operation memory instruction must be followed by 6 NOP instructions, for example:

```
MOV MCTRL,#09H                ; The write operation begins
NOP
NOP
NOP
NOP
NOP
NOP
MOV MCTRL,#01H                ; The read operation begins
NOP
NOP
NOP
NOP
NOP
NOP
```

The program CRC check command is set by the register MCTRL [3:2], the start and end addresses are freely configurable through the registers MADRL/MADRH, and the results are saved in the register PCRCDL/PCRCDH.

This CRC operation can only access the program storage space, and the data storage space cannot be accessed. During the program space CRC check, the CPU stops working and waits for the CPU to continue running after the CRC calculation is complete. The CRC check is checked bytely, from the initial address to the end address, and after performing the current CRC check, you need to set the MMODE [1:0]=00 of MCTRL. The CRC check procedure is as follows:

1) Enable access to program memory registers:
   TA = 0xAA;
   TA = 0x55;
   MLOCK=0xAA; The default value is 00
2) Check the results before clearing the program CRC:
   PCRCDL=0x00; PCRCDH=0x00。
3) Set the program CRC check start and end addresses:
   MCTRL[1]=0, set the starting address by MADRL/MADRH;
   MCTRL[1]=1, with the end address set via MADRL/MADRH.
4) Start the program CRC check command:
   MCTRL=0x05。
5) Wait for the program CRC check to end:
   After the CRC check is complete, the MCTRL[0] hardware clears 0.
6) Read the program CRC check result:
   PCRCDL stores the lower 8 bits CRC operation result of the program;
   PCRCDH stores the higher 8 bits CRC operation result of the program.
7) The CRC check end address selection bit is cleared to 0.
   MCTRL[1] is cleared to 0 by software after the CRC check.

26. Unique ID (UID)

26.1 Overview

Each chip has a different 96-bit unique identification number, or Unique identification. It has been set at the factory and cannot be modified by the user.

26.2 UID Register Description

UID0

F5E0H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID0	UID7	UID6	UID5	UID4	UID3	UID2	UID1	UID0
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 UID<7:0>

UID1

F5E1H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID1	UID15	UID14	UID13	UID12	UID11	UID10	UID9	UID8
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 UID<15:8>

UID2

F5E2H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID2	UID23	UID22	UID21	UID20	UID19	UID18	UID17	UID16
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 UID<23:16>

UID3

F5E3H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID3	UID31	UID30	UID29	UID28	UID27	UID26	UID25	UID24
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 UID<31:24>

UID4

F5E4H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID4	UID39	UID38	UID37	UID36	UID35	UID34	UID33	UID32
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 UID<39:32>

UID5

F5E5H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID5	UID47	UID46	UID45	UID44	UID43	UID42	UID41	UID40
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 UID<47:40>

UID6

F5E6H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID6	UID55	UID54	UID53	UID52	UID51	UID50	UID49	UID48
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 UID<55:48>

UID7

F5E7H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID7	UID63	UID62	UID61	UID60	UID59	UID58	UID57	UID56
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 UID<63:56>

UID8

F5E8H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID8	UID71	UID70	UID69	UID68	UID67	UID66	UID65	UID64
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 UID<71:64>

UID9

F5E9H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID9	UID79	UID78	UID77	UID76	UID75	UID74	UID73	UID72
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 UID<79:72>

UID10（0xF5EA）

F5EAH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID10	UID87	UID86	UID85	UID84	UID83	UID82	UID81	UID80
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 UID<87:80>

UID11

F5EBH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID11	UID95	UID94	UID93	UID92	UID91	UID90	UID89	UID88
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 UID<95:88>

27. User configuration

The System Configuration Register (CONFIG) is a FLASH option for the initial conditions of the MCU and cannot be accessed or operated by the program. It contains the following:

1. WDT (Watchdog Working Method Selection)
   - ENABLE — Force open WDT
   - SOFTWARE CONTROL (default) — The WDT operation is controlled by the WDTRE bit of the WDCON register
2. PROTECT
   - ENABLE — The FLASH code is encrypted, and the code read out is 00H. And it is forbidden to enter debug mode
   - DISABLE (default) — Flash code is not encrypted
3. FLASH_DATA_PROTECT
   - DISABLE — Flash data areas are not encrypted
   - ENABLE (default) — The FLASH data area is encrypted, and the value read out by the flashing emulator after encryption is 00H
4. LVR (Low Voltage Reset)
   - 1.8V (default).
   - 2.0V
   - 2.5V
   - 3.5V
5. DEBUG (debug mode)
   - DISABLE (default) — Debug mode prohibits, DSCK, DSDA pins are used as ordinary IO ports
   - ENABLE — Debug mode enables, DSCK, DSDA pins are configured as debug ports, and other functions corresponding to the pins are turned off
6. OSC (oscillation mode)
   - HSI (default) — 48MHz
   - HSE
   - LSE(32.768KHz)
   - LSI(125KHz) — 125KHz
7. SYS_PRESCALE (system clock prescale selection)
   - F_{OSC}/1 (default).
   - F_{OSC}/2
   - F_{OSC}/4
   - F_{OSC}/8
8. HSI_FS (internal RC oscillator crossover selection)
   - F_{HSI}/1 — 48MHz
   - F_{HSI}/2 — 24MHz
   - F_{HSI}/3 — 16MHz
   - F_{HSI}/6 (default). — 8MHz
9. EXT_RESET (external reset configuration)
   - DISABLE (default) — External reset prohibits
   - ENABLE — External reset enable

- ENABLE(OPEN PULLUP) — An external reset is enabled and the internal pull-up resistor of the reset port is turned on

10. EXT_RESET_SEL
    - Select P24 pin for external reset port
    - Select P25 pins for the external reset port
11. WAKE UP_WAIT TIME (sleep wake-up waits for oscillator to stabilize by default to 1.0s)
    - 50us
    - 100us
    - 500us
    - 1ms
    - 5ms
    - 10ms
    - 500ms
    - 1.0s (default)
12. CPU_WAITCLOCK (memory wait clock selection)
    - 1*System Clock (1T) (default)
    - 2*System Clock (2T)
    - 3*System Clock (3T)
    - 4*System Clock (4T)
    - 5*System Clock (5T)
    - 6*System Clock (6T)
    - 7*System Clock (7T)
    - 8*System Clock (8T)
13. WRITE_PROTECT program partition protection (protectable areas, all default intervals are unprotected)
    - 0000H-07FFH (Protected/Unprotected)
    - 0800H-0FFFH (Protected/Unprotected)
    - 1000H-17FFH (Protected/Unprotected)
    - 1800H-1FFFH (Protected/Unprotected)
    - 2000H-27FFH (Protected/Unprotected)
    - 2800H-2FFFH (Protected/Unprotected)
    - 3000H-37FFH (Protected/Unprotected)
    - 3800H-3FFFH (Protected/Unprotected)
14. BOOT space selection
    - BOOT_DIS (default) — The BOOT zone is prohibited
    - BOOT_1K — The BOOT area space is 1K
    - BOOT_2K — The BOOT area space is 2K
    - BOOT_4K — The boot area space is 4K

Note:

1) The machine cycle is related to the memory wait clock selection (CPU_WAITCLOCK): machine cycle = T_{SYS}/CPU_WAITCLOCK.
2) When the oscillation mode is selected as HSI, the internal RC oscillator is selected as FHSI/1, and the system clock prescale is selected as F_{OSC}/1, and all three conditions are met, if the memory wait clock is selected as 1*System Clock (1T), the actual memory wait clock is selected as 2T, and the machine cycle = T_{SYS}/2.

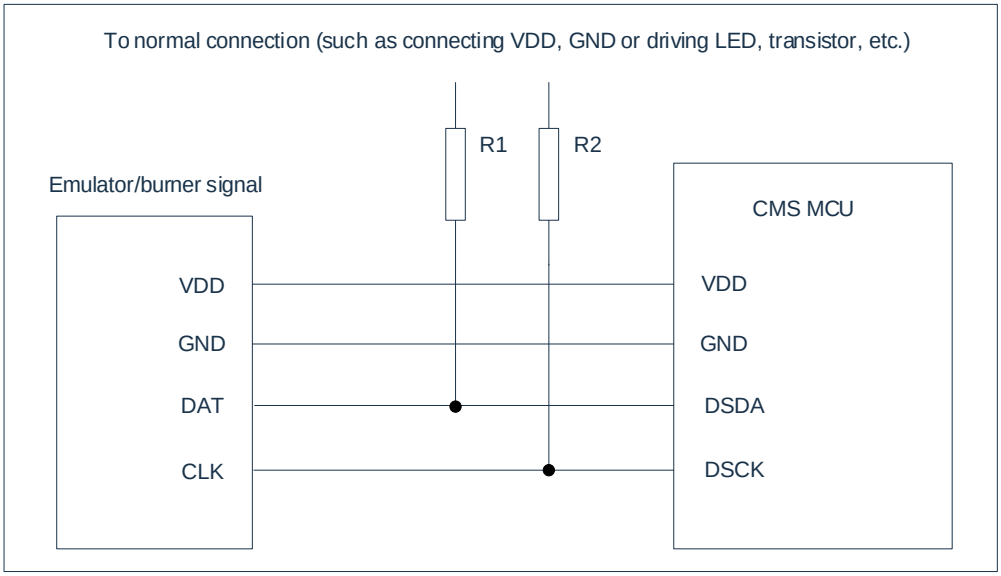
28. In-circuit Programming and Debugging

28.1 Online Programming Mode

The chip can be programmed serially in the end application circuit. Programming can be done simply by the following 4 wires:

- Power cord
- Ground wire
- Data cable
- Clock line

In-line serial programming allows users to manufacture circuit boards using unprogrammed devices and program the chip only before product delivery, allowing the latest version of firmware or custom firmware to be programmed into the chip. A typical online serial programming connection method is shown in the following figure:



In the figure above, R1 and R2 are galvanical isolation devices, often replaced by resistors, and their resistance values are as follows: R1≥4.7K, R2≥4.7K.

Note that the DSDA disables the connection of pull-down resistors during programming and debugging. If the actual circuit needs to be connected to the pull-down resistor, it is recommended to use the jumper structure to disconnect the pull-down resistor during programming/debugging, and then plug in the pull-down resistor after completion.

28.2 Online Debug Mode

The chip supports 2-wire (DSCK, DSDA) in-circuit debugging. If you use the in-circuit debugging function, you need to set DEBUG in the system configuration register to ENABLE. When using debug mode, you need to be aware of the following points:

- Under the debug state, the DSCK and DSDA ports are dedicated debug ports and cannot implement their GPIO and multiplexing functions.
- When the debug state enters sleep mode/idle mode, the system power supply and oscillator do not stop working, and the sleep wake function can be simulated in this state. If you need to focus on power consumption, it is recommended to turn off the debug function and then test the actual sleep current of the chip.
- Pauses in debug state, other functional peripherals continue to run, and the WDT, Timer0/1/2/3/4 counters stop. However, if Timer1/4 is used as the baud rate generator for UART0, Timer1/4 will also continue to run in the paused state. Peripherals that continue to run in a paused state may cause breaks, and you need to be aware when debugging.

29. Instruction description

Assembly instructions consist of a total of 5 categories: arithmetic operations, logical operations, data transfer operations, Boolean operations, and program branch instructions, all of which are compatible with standard 8051.

29.1 Symbol description

Symbol	Description
Rn	Working registers R0-R 7
Direct	The cell address (00H-FFH) of the internal data memoryRAM or the address in the special function register SFR
@Ri	Indirection register (@R0 or @R1).
#data	8-bit binary constant
#datto16	A 16-bit binary constant in the instruction
Bit	Bit address in the internal data memoryRAM or special function register SFR
Addr16	16-bit address with an address range of 0-64KB address space
Addr11	11-bit address, address range 0-2KB address space
Rthe	Relative address
A	accumulator

29.2 List of Instructions

Mnemonics		description
Operation class		
ADD	A,R n	Accumulator plus register
ADD	A,direct	Accumulator plus direct addressing unit
ADD	A,@Rto	Accumulator plus indirect addressingRAM
ADD	A,#data	The accumulator adds the immediate number
ADDC	A,Rn	Accumulator plus registers and carry flags
ADDC	A,direct	Accumulator plus direct addressing unit and carry signs
ADDC	A,@Rto	Accumulator plus indirect addressingRAM and carry flags
ADDC	A,#data	Accumulator plus immediate number and carry signs
SANDBB B	A,R n	Accumulator minus register and carry flag
SANDBB B	A,direct	Accumulator minus direct addressing unit and carry flag
SANDBB B	A,@Ri	Accumulator minus indirect addressingRAM and carry flags
SANDBB B	A,#data	Accumulator minus immediate number and carry signs
INC	A	Accumulator plus 1
INC	Rn	Register plus 1
INC	direct	Direct addressing unit plus 1
INC	@Ri	Indirection RAM plus 1
INC	DPTR	Data pointer plus 1
DEC	A	Accumulator minus 1
DEC	Rn	Register minus 1
DEC	direct	Direct addressing unit minus 1
DEC	@ Ri	Indirection RAM minus 1
MANDL	A,B	Accumulator multiplier by register B
DIV	A,B	The accumulator is divided by register B
DA	A	Decimal adjustment
Logical operation classes		
ANL	A,Rn	Accumulators and registers
ANL	A,direct	Accumulators and direct addressing units
ANL	A,@Rto	Accumulator with indirect addressingRAM
ANL	A,#data	Accumulator with immediate number
ANL	direct,A	Direct addressing unit with accumulator
ANL	direct,#data	Direct addressing units with immediate numbers
ORL	A,R n	Accumulator or register
ORL	A, direct	Accumulator or direct addressing unit
ORL	A,@Rto	Accumulator or indirect addressingRAM
ORL	A, #data	Accumulator or immediate number
ORL	direct,A	Direct addressing unit or accumulator
ORL	diri walkt,#dayour	Direct addressing units or immediate numbers
XRL	A,R n	Accumulator Xor register
XRL	A,direct	Accumulator heterogeneous or direct addressing unit
XRL	A,@Ri	Accumulator Xor indirect addressing RAM
XRL	A,#data	Accumulator different or immediate number
XRL	direct,A	Direct addressing unit Xor accumulator
XRL	direct,#data	Direct addressing units vary or immediately number
CLR	A	Accumulator clear 0
CPL	A	The accumulator is reversed

Mnemonics		description
RL	A	The accumulator is shifted in the left loop
RLC	A	The accumulator is even the carry flag for a left loop shift
RR	A	The accumulator is shifted in the right loop
RR RC	A	The accumulator is connected to the carry mark right loop shift
SWAP	A	The accumulator is swapped 4 bits high and 4 bits low
Data transfer class		
MOV	A,R n	Registers are transferred to the accumulator
MOV	A,direct	The direct addressing unit is transferred to the accumulator
MOV	A,@Ri	Indirection RAM feed accumulator
MOV	A,#data	Immediately count the accumulators
MOV	Rn,A	Accumulator feed register
MOV	Rn,direct	Direct addressing unit feed register
MOV	Rn,#data	Immediately count the send registers
MOV	direc t,A	The accumulator sends a direct addressing unit
MOV	direc t,R n	Registers send direct addressing units
MOV	d irect1,direct2	The direct address unit is transferred to the direct addressing unit
MOV	d irec t,@Ri	Indirect addressing RAM sends direct addressing units
MOV	d irect,#data	Immediately count the number of direct addressed units
MOV	@Ri,A	The accumulator sends an indirectly addressed RAM
MOV	@R i,direct	The direct addressing unit sends the indirect addressing RAM
MOV	@Ri,#datake	Immediately send indirection addressed RAM
MOV	DPTR,#data16	The 16-bit immediate number sends a data pointer
MOVC	A,@A+DPTR	Lookup table data feed accumulator (DPTR is the base address).
MOVC	A,@A+PC	Lookup table data feed accumulator (PC is the base address).
MOVX	A,@Rto	ExternalRAM unit feed accumulator (8-bit address).
MOVX	A,@DPTR	ExternalRAM unit feed accumulator (16-bit address).
MOVX	@Ri,A	The accumulator sends an externalRAM unit (8-bit address).
MOVX	@DPTR,A	The accumulator sends an externalRAM unit (16-bit address).
PUSH	direct	The direct addressing unit is pressed into the top of the stack
POP	direct	A direct addressing unit pops up at the top of the stack
XCH	A,Rn	The accumulator is exchanged with the registers
XCH	A, direct	The accumulator is swapped with the direct addressing unit RAM
XCH	A,@Ri	The accumulator is swapped with the indirect addressing unit RAM
XCHD	A,@Ri	The accumulator swaps 4 bits lower with the indirect addressing unit RAM
Boolean operation class		
CLR	C	C Clear Zero
CLR	bit	The direct address bit is cleared to zero
SETB	C	C set
SETB	bit	Direct addressing location bits
CPL	C	C takes the reverse
CPL	bit	Direct addressing bits are reversed
ANL	C,bit	C logic with direct addressing bits
ANL	C,/bit	C logic is inverse with direct addressing bits
ORL	C,bit	C logical or direct addressing bits
ORL	C,/bit	C logic or the inverse of the direct addressing bit
MOV	C,bit	Direct addressing bits send C
MOV	bit,C	C sends direct addressing bits
Program jump class		

Mnemonics		description
ACALL	add r11	Absolute invocation within the 2K address range
LCALL	addr16	Long calls within 64K address range
RAND		Subroutine returns
RETI		Interrupt returns
AJMP	addr11	Absolute transfer within 2K address range
LJMP	add r16	Long transfer within 64K address range
SJMP	randl	Relatively short transfer
JMP	@A+DPTR	Relatively long transfer
JZ	rthe	The accumulator is 0 transfers
JNZ	rthe	The accumulator is not transferred to 0
JC	rthe	C is 1 transfer
JNC	rthe	C is 0 transfer
JB	band t,r the	The direct addressing bit is 1 transfer
JNB	band t,r the	The direct addressing bit is transferred at 0
JBC	band t,r the	The direct addressing bit is transferred to 1 and clears that bit
CJNE	A,dandrec t,r the	Accumulators are transferred unequally from direct addressing units
CJNE	A,#data,r the	Accumulator with immediate number unequal transfer
CJNE	Rn,#data,r el	Registers are transferred with an immediate number unequal
CJNE	@ Ri,#data,r el	Indirection unit RAM with immediate number unequal transfer
DJNZ	Rn,r el	Register minus 1 does not transfer to 0
DJNZ	dandrec t,r the	Direct addressing unit minus 1 does not transfer to 0
NOP		Empty directive
Read-Modify-Write instructions (R ead-Modify-Write).		
ANL		Logical (direct ANL, A and direct ANL, #data)
ORL		Logical OR (ORL direct, A and ORL direct, #data)
XRL		Logical XRL direct, A and XRL direct, #data).
JBC		The direct addressing bit is transferred to 1 and cleared (JBC bit, rel).
CPL		CPL bit
INC		Increment by 1 (INC direct)
DEC		Decrement by 1. (DEC direct)
DJNZ		Minus 1 is not 0 transfer (DJNZ direct, rel).
MOV	bit,C	C sends direct addressing bits
CLR	bit	The direct address bit is cleared to zero
SETB	bit	Direct addressing location bits

30. Version Revision Notes

The version number	Time	Revision content
V1.00	April 2020	Initial release
V1.01	August 2020	Modify the user configuration instructions
V1.02	January 2023	2.3 General Data Register RAM: Modify title
V1.03	January 2023	Delete UART1 chapter description
V1.04	January 2023	Modify BOOT partition
V1.05	January 2023	1) 20.2 I2C Port Configuration: add description 2) 7.2.2 Port Multiplexing Feature Configuration Register: adjustment instructions 3) 15.2.2 BUZZER Frequency Control Register BUZDIV: add precautions 4) 22.2.4 Convert the Clock: example of optimizing ADC clock 5) 25.3 Feature Description: delete FLASH parameter 6) 17.4.3 LCD COM/SEG Select Register LCD_S1, 17.4.4 LCD COM/SEG Select Register LCD_S2, 17.4.6 LCD Function Select Register LCDEN1: adjustment register description 7) 18.2 Characteristic: correct COM/SEG quantity 8) Correct register description and text errors
V1.06	January 2023	1) 3.1 Power-on Reset: adjust the reset type of settable PORF flag bit to 1 2) 23.3.2 Features, 23.3.3 Calculation Formula: adjustment instructions
V1.0.7	January 2023	Corrected the description errors in section 16.5.22
V1.0.8	Mar 2023	Corrected Bit name in 17.4.7LCD Function Select Register LCDEN2
V1.09	July 2023	1) Added some remarks to section 25.2.7 2) Added the step "Clear the CRC end address select bit" to the function description in section 25.3