CMS80F262x Series

Reference Manual

Enhanced flash memory 1T 8051 microcontrollers

Rev. 1.0.9

Contents

1. Central Processing Unit（CPU）

This 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 accumulator ACC, general register B, stack pointer SP, data pointer DPTR, Program status register PSW, Program counter PC, etc.

1.1 Reset Vector（0000H）

The microcontroller has a word-length system reset vector (0000H). After a reset occurs, the program will restart from 0000H, and the system registers will all be restored to default values. The following program demonstrates how to define the reset vector in FLASH.

Example: define reset vector

```
        ORG     0000H       ; System reset vector
        LJMP    START
        ORG     0010H       ; User program start
START:
        …                   ; User program
        …
        END                 ; End of program
```

1.2 Accumulator（ACC）

ALU is an 8Bit wide arithmetic logic unit, and all the mathematics and logic operations of the MCU are completed through it. It can add, subtract, shift and logic operations on data; ALU also controls the status bit (in the PSW status register) to indicate the status of the operation result.

The ACC register is an 8Bit register, the result of the ALU operation can be stored here.

1.3 B Register（B）

The B register is used when using multiplication and division instructions. If you don't use multiplication and division instructions, it can also be used as a general-purpose register.

1.4 Stack Pointer Register（SP）

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

The operations that affect the SP are: instructions PUSH, LCALL, ACALL, POP, RET, RETI and entering interrupt.

The PUSH instruction occupies one byte on the stack; LCALL, ACALL and interrupt occupies two bytes on the stack, the POP instruction releases one byte, and the RET/RETI instruction releases two bytes.

Use the PUSH instruction to automatically save the current value of the operated register to RAM.

1.5 Data Pointer Register（DPTR0/DPTR1）

The data pointer is mainly used in MOVX, MOVC instruction, its function is to locate the address of XRAM and ROM. There are two data pointer registers DPTR0 and DPTR1 inside the chip, which are selected by the DPS register.

Each group 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 into XRAM address 0001H
```

1.6 Data Pointer Selection Register（DPS）

Data pointer selection register DPS

0x86	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DPS	ID1	ID0	TSL	AU	--	--	--	SEL
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>: Self-decreasing/self-adding function selection.
00= DPTR0 add 1 or DPTR1 add 1;
01= DPTR0 reduce 1 or DPTR1 add 1;
10= DPTR0 add 1 or DPTR1 reduce 1;
11= DPTR0 reduce 1 or DPTR1 reduce 1.
Bit5 TSL: Flip selection enable;
1= After executing the DPTR instruction, the SEL bit will automatically flip;
0= DPTR related instructions do not affect the SEL bit.
Bit4 AU: Self-add/reduce enable bit;
1= After the MOVX @DPTR or MOVC @DPTR instruction is allowed to run, the self-decrement/self-increment operation (determined by ID1-ID0) is executed.
0= DPTR related instructions do not affect the SEL bit.
Bit3~Bit1 -- Reserved, all must be 0.
Bit0 SEL: Data pointer selection bit;
1= Choose DPTR1;
0= Choose DPTR0.

1.7 Program Status Register（PSW）

Program status register PSW

0xD0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PSW	CY	AC	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 flag;
	1=	Carry;
	0=	No carry.
Bit6	AC:	Auxiliary carry flag (half carry flag);
	1=	Carry;
	0=	No carry.
Bit5	F0:	General flag.
Bit4~Bit3	RS<1:0>:	Working register BANK selection bit;
	00=	Choose Bank0;
	01=	Choose Bank1;
	10=	Choose Bank2;
	11=	Choose Bank3.
Bit2	OV:	Overflow flag
	1=	Overflow in arithmetic or logical operation;
	0=	There is no overflow in arithmetic or logical operations.
Bit1	--	Reserved, must be 0.
Bit0	P:	Check Digit;
	1=	The highest bit of the result has a carry.
	0=	The highest bit of the result is not carried.

1.8 Program Counter（PC）

The program counter (PC) controls the execution order of instructions in the FLASH of the program memory. It can address the entire range of the FLASH. After obtaining the instruction code, the Program counter (PC) will automatically increase by one and point to the address of the next instruction code. But if you perform operations such as jump, conditional jump, subroutine call, initialization reset, interrupt, interrupt return, subroutine return, etc., the PC will load the address related to the instruction instead of the address of the next instruction.

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

1.9 Timing Access Register（TA）

Timing access register TA

0x96	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TA	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 bit.

Some protected registers must be written before the following operations are performed on TA.

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

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

Protected register: WDCON，CLKDIV，SCKSEL，MLOCK，WWCON0，WWCON1，WWCMPD.

2. Memory and Register Map

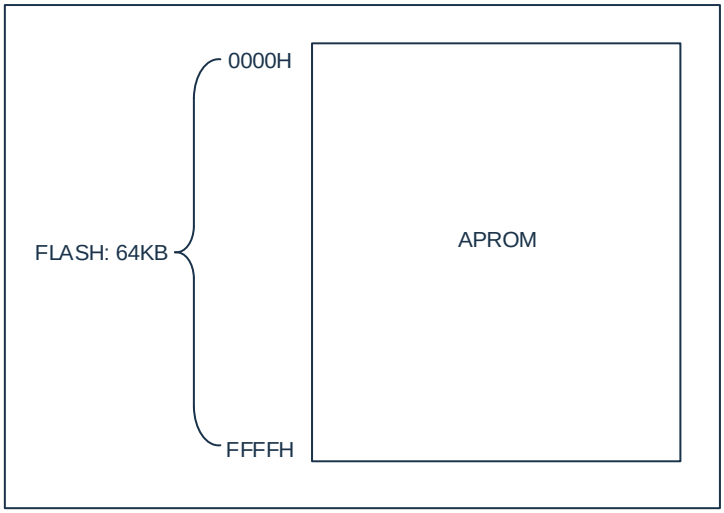
This series of micro-controllers has the following types of memories:

- Up to 64KB of FLASH program memory (APROM area).
- Non-volatile data storage (Data FLASH) up to 1KB.
- General purpose internal data memory (RAM) up to 256B.
- General-purpose external data memory (XRAM) up to 4KB.
- Special function register SFR (BANK0 and BANK1).
- External Special function register XSFR.

2.1 Program Memory APROM

The program memory APROM is used to store source programs and table data, and 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.

The block diagram of the FLASH space allocation structure is shown in the figure below:

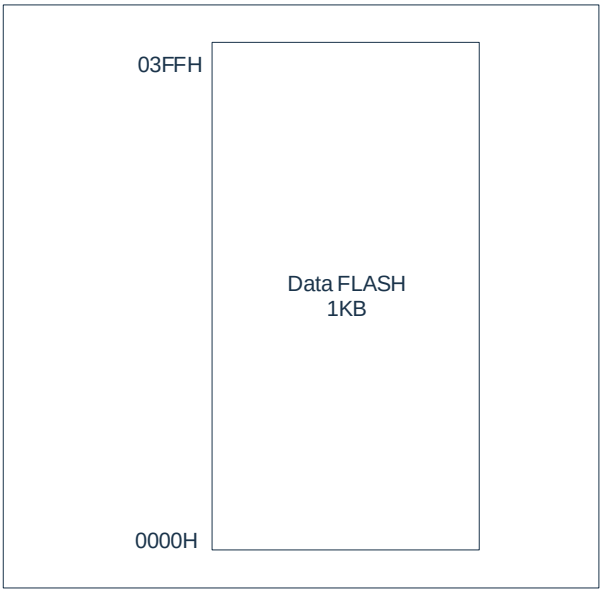


After the chip is reset, the CPU starts to execute from 0000H. Each interrupt is assigned a fixed address in the program memory, and the interrupt causes the CPU to jump to this address and start executing the service program.

For example, external interrupt 1 is assigned the address 0013H. If External Interrupt 1 is used, its service program must start at 0013H. If the interrupt is not used, its service address is used as the storage address of the ordinary program.

2.2 Non-Volatile Data Storage Data FLASH

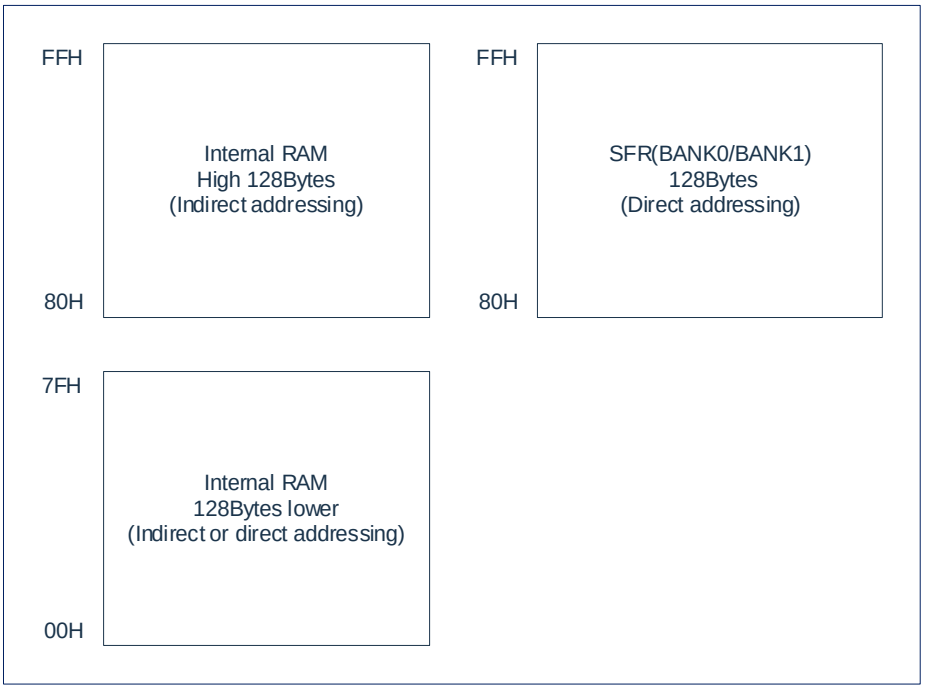
Non-volatile data storage data FLASH can be used to store important data such as constant data, calibration data, protection and safety-related information. The data stored in this area has the characteristic that the data will not be lost when the chip is powered off or suddenly or unexpectedly. The block diagram of the data FLASH space allocation structure is shown in the figure below:



Data FLASH memory read, write, and erase operations are realized through the FLASH control interface.

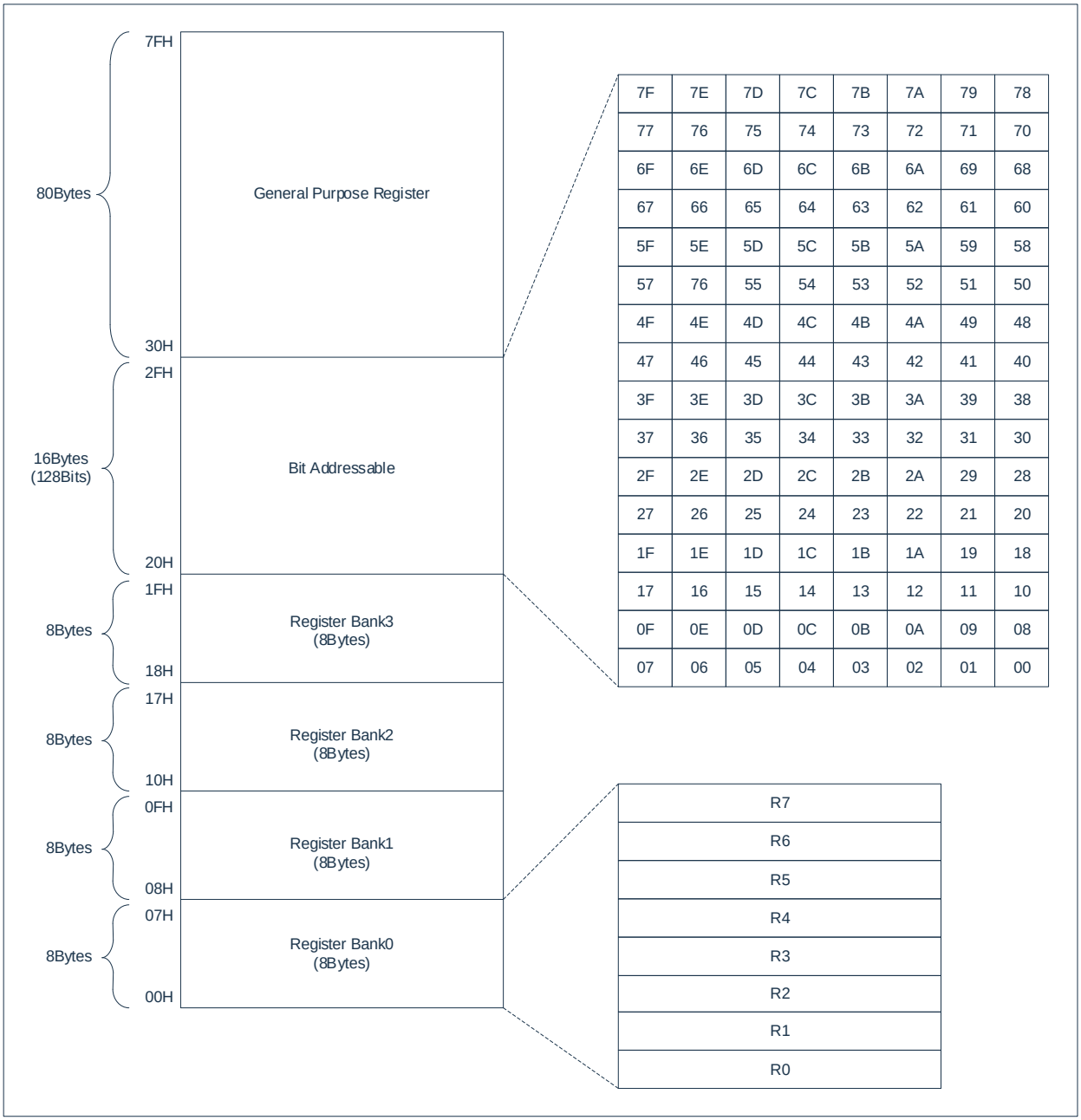
2.3 General Data Memory RAM

The internal data memory is divided into 3 parts: low 128Bytes, high 128Bytes, and Special function register SFR. The structure diagram of RAM space allocation is shown in the figure below:



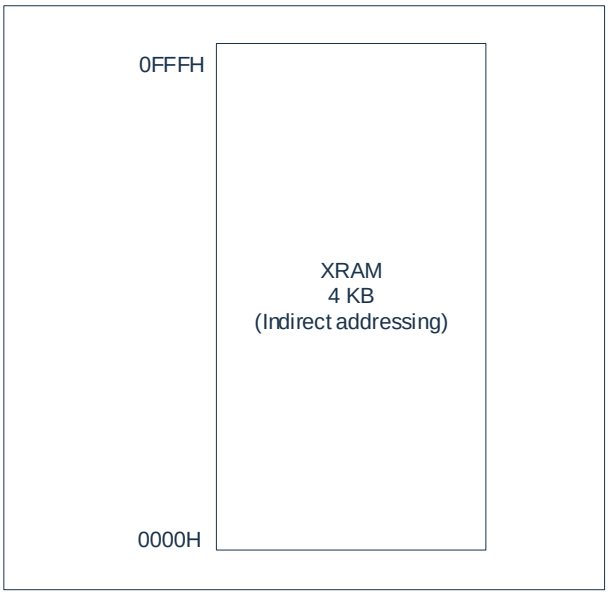
The high 128 bytes and SFR shown in the figure above occupy the same area (80H~FFH), but they are independent. Directly address the storage space (SFR) higher than 7FH and indirectly address the storage space higher than 7FH (high 128 bytes) into different storage spaces. SFR is divided into two pages, BANK0 and BANK1, each page is 128Bytes, occupying the same address area, and entering different storage spaces through the paging register selection.

The low 128 bytes space register allocation shown in the figure above is shown in the figure below. The lowest 32 bytes (00H~1FH) constitute 4 register groups, each group of 8 storage units, with R0~R7 as the unit number, used to store operands and intermediate results. After reset, group 0 is selected by default. If another register group is selected, it needs to be determined by changing the program state. The 16Bytes (20H~2FH) behind the register group constitute a bit-addressable storage space. The RAM unit in this area can be operated either by byte or directly on each bit in the unit. The remaining 80 storage units (30H~7FH), the user can set the stack area and store intermediate data.


7FH
80Bytes
General Purpose Register
30H
2FH
16Bytes
(128Bits)
Bit Addressable
20H
1FH
8Bytes
Register Bank3
(8Bytes)
18H
17H
8Bytes
Register Bank2
(8Bytes)
10H
0FH
8Bytes
Register Bank1
(8Bytes)
08H
07H
8Bytes
Register Bank0
(8Bytes)
00H
7F 7E 7D 7C 7B 7A 79 78
77 76 75 74 73 72 71 70
6F 6E 6D 6C 6B 6A 69 68
67 66 65 64 63 62 61 60
5F 5E 5D 5C 5B 5A 59 58
57 76 55 54 53 52 51 50
4F 4E 4D 4C 4B 4A 49 48
47 46 45 44 43 42 41 40
3F 3E 3D 3C 3B 3A 39 38
37 36 35 34 33 32 31 30
2F 2E 2D 2C 2B 2A 29 28
27 26 25 24 23 22 21 20
1F 1E 1D 1C 1B 1A 19 18
17 16 15 14 13 12 11 10
0F 0E 0D 0C 0B 0A 09 08
07 06 05 04 03 02 01 00
R7
R6
R5
R4
R3
R2
R1
R0

2.4 General External Data Register XRAM

There is a max 4KB XRAM area inside the chip, which is not related to FLASH/RAM. The structure diagram of XRAM space allocation is shown in the figure below:



XRAM/XSFR space access is operated by the DPTR data pointer. DPTR includes two sets of pointers: DPTR0, DPTR1, which are selected by the DPS register. For example, through MOVX indirect addressing operation, the assembly code is as follows:

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

After setting Target-->Memory Model to Large in Keil51, the C compiler will use XRAM as the variable address. Generally use DPTR for XRAM/XSFR operations.

2.5 Special Function Register SFR

Special function register refers to a collection of special-purpose registers, which are essentially on-chip RAM units with special functions, which are discretely distributed in the address range 80H~FFH. Users can perform byte access to them through direct addressing instructions. The low 4 bits of the address are 0000 or 1000 for bit addressing, such as P0, TCON, P1.

SFR is divided into 2 pages, BANK0 and BANK1, and the paging function is controlled by the SFRS register. Addresses 0xD1-0xD7, 0xD9-0xDF, 0xE1-0xE7, 0xE9-0xEF, 0xF1-0xF7, a total of 35 addresses, are different registers in BANK0 and BANK1, accessed through the paging function; other address corresponding in BANK0 and BANK1 are The same register can be accessed through BANK0 and BANK1.

It should be noted that the system does not automatically switch BANK0/1 by the hardware, but executes the switch operation of BANK by writing the paging register (SFRS) by software. Especially in the interrupt service routine, users need to save and restore the paging register (SFRS) by themselves.

SFR paging control register（SFRS）

0x92	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SFRS	SFRS7	SFRS6	SFRS5	SFRS4	SFRS3	SFRS2	SFRS1	SFRS0
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 SFRS<7:0>: Paging selection control;

0x00= BANK0;

0x01= BANK1;

Others= No Access.

The BANK0 register table is as follows:

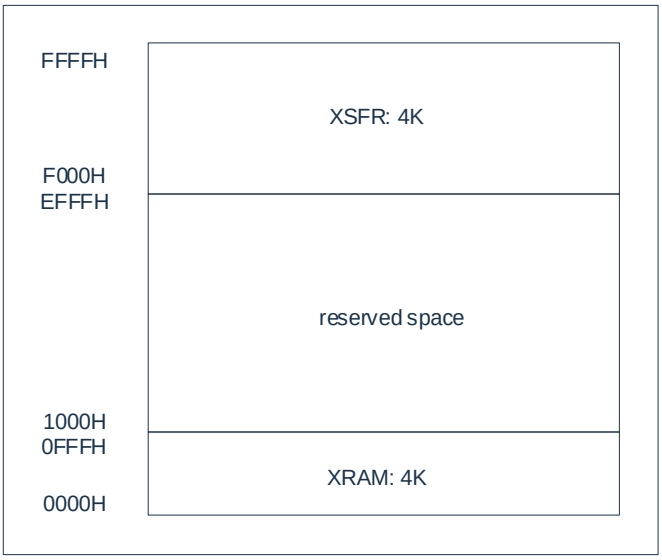
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
0xF8	--	MREGION	MMODE	MDATA	MADRL	MADRH	MSTATUS	MLOCK
0xF0	B	I2CSADR	I2CSCR	I2CSBUF	I2CMSA	I2CMCR	I2CMBUF	I2CMTP
0xE8	--	ADCON2	SCON1	SBUF1	SPCR	SPSR	SPDR	SSCR
0xE0	ACC	--	TL4	TH4	--	WWCON0	WWCMPD	WWCON1
0xD8	P5	--	TL3	TH3	ADRESL	ADRESH	ADCON1	ADCON0
0xD0	PSW	ADCMPC	T34MOD	ADDLYL	ADCMPL	ADCMPH	SCKSEL	SCKSTAU
0xC8	T2CON	T2IF	RLDL	RLDH	TL2	TH2	CCEN	T2IE
0xC0	P4	CAP2CON	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	EIE3	P0EXTIE	P1EXTIE	P2EXTIE	P3EXTIE
0xA0	P2	P1TRIS	P2TRIS	P3TRIS	P4TRIS	P5TRIS	P4EXTIF	P5EXTIF
0x98	SCON0	SBUF	P0TRIS	P4EXTIE	P5EXTIE	--	--	--
0x90	P1	FUNCCR	SFRS	DPX0	--	DPX1	TA	WDCON
0x88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	CLKDIV
0x80	P0	SP	DPL0	DPH0	DPL1	DPH1	DPS	PCON

The BANK1 register table is as follows：

	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
0xF8	--	MREGION	MMODE	MDATA	MADRL	MADRH	MSTATUS	MLOCK
0xF0	B	--	--	--	--	--	--	--
0xE8	--	MD0	MD1	MD2	MD3	MD4	MD5	ARCON
0xE0	ACC	--	FUNCCR1	PCON1	SCON2	SBUF2	SCON3	SBUF3
0xD8	P5	--	--	--	--	--	--	--
0xD0	PSW	--	--	--	--	--	--	--
0xC8	T2CON	T2IF	RLDL	RLDH	TL2	TH2	CCEN	T2IE
0xC0	P4	CAP2CON	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	EIE3	P0EXTIE	P1EXTIE	P2EXTIE	P3EXTIE
0xA0	P2	P1TRIS	P2TRIS	P3TRIS	P4TRIS	P5TRIS	P4EXTIF	P5EXTIF
0x98	SCON0	SBUF	P0TRIS	P4EXTIE	P5EXTIE	--	--	--
0x90	P1	FUNCCR	SFRS	DPX0	--	DPX1	TA	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 register, other function control register. Its addressing range is shown in the figure below:



The external Special function register list is as follows:

Address	Symbol	Direction
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	--	--
F00DH	P0SR	P0 port slope control register
F00EH	P0DS	P0 port data input selection register
F00FH	--	--
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	--	--
F01DH	P1SR	P1 port slope control register

Address	Symbol	Direction
F01EH	P1DS	P1 port data input selection register
F01FH	--	--
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
F026H	P26CFG	P26 port configuration register
F027H	P27CFG	P27 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	--	--
F02DH	P2SR	P2 port slope control register
F02EH	P2DS	P2 port data input selection register
F02FH	--	--
F030H	P30CFG	P30 port configuration register
F031H	P31CFG	P31 port configuration register
F032H	P32CFG	P32 port configuration register
F033H	P33CFG	P33 port configuration register
F034H	P34CFG	P34 port configuration register
F035H	P35CFG	P35 port configuration register
F036H	P36CFG	P36 port configuration register
F037H	P37CFG	P37 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 drive current control register
F03DH	P3SR	P3 port slope control register
F03EH	P3DS	P3 port data input selection register
F03FH	--	--
--	--	--
F040H	P40CFG	P40 port configuration register
F041H	P41CFG	P41 port configuration register
F042H	P42CFG	P42 port configuration register
F043H	P43CFG	P43 port configuration register
F044H	P44CFG	P44 port configuration register
F045H	P45CFG	P45 port configuration register
F046H	P46CFG	P46 port configuration register
F047H	P47CFG	P47 port configuration register
F049H	P4OD	P4 port open drain control register
F04AH	P4UP	P4 port pull-up resistor control register
F04BH	P4RD	P4 port pull-down resistor control register
F04CH	--	--
F04DH	P4SR	P4 port slope control register
F04EH	P4DS	P4 port data input selection register
--	--	--

Address	Symbol	Direction
F050H	P50CFG	P50 port configuration register
F051H	P51CFG	P51 port configuration register
F052H	P52CFG	P52 port configuration register
F053H	P53CFG	P53 port configuration register
F054H	P54CFG	P54 port configuration register
F055H	P55CFG	P55 port configuration register
F056H	--	--
F057H	--	--
F059H	P5OD	P5 port open drain control register
F05AH	P5UP	P5 port pull-up resistor control register
F05BH	P5RD	P5 port pull-down resistor control register
F05CH	--	--
F05DH	P5SR	P5 port slope control register
F05EH	P5DS	P5 port data input selection register
--	--	--
F080H	P00EICFG	P00 port interrupt control register
F081H	P01EICFG	P01 port interrupt control register
F082H	P02EICFG	P02 port interrupt control register
F083H	P03EICFG	P03 port interrupt control register
F084H	P04EICFG	P04 port interrupt control register
F085H	P05EICFG	P05 port 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 port interrupt control register
F08AH	P12EICFG	P12 port interrupt control register
F08BH	P13EICFG	P13 port interrupt control register
F08CH	P14EICFG	P14 port interrupt control register
F08DH	P15EICFG	P15 port interrupt control register
F08EH	P16EICFG	P16 port interrupt control register
F08FH	P17EICFG	P17 port interrupt control register
F090H	P20EICFG	P20 port interrupt control register
F091H	P21EICFG	P21 port interrupt control register
F092H	P22EICFG	P22 port interrupt control register
F093H	P23EICFG	P23 port interrupt control register
F094H	P24EICFG	P24 port interrupt control register
F095H	P25EICFG	P25 port interrupt control register
F096H	P26EICFG	P26 port interrupt control register
F097H	P27EICFG	P27 port 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
F09CH	P34EICFG	P34 port interrupt control register
F09DH	P35EICFG	P35 port interrupt control register
F09EH	P36EICFG	P36 port interrupt control register
F09FH	P37EICFG	P37 port interrupt control register
F0A0H	P40EICFG	P40 port interrupt control register

Address	Symbol	Direction
F0A1H	P41EICFG	P41 port interrupt control register
F0A2H	P42EICFG	P42 port interrupt control register
F0A3H	P43EICFG	P43 port interrupt control register
F0A4H	P44EICFG	P44 port interrupt control register
F0A5H	P45EICFG	P45 port interrupt control register
F0A6H	P46EICFG	P46 port interrupt control register
F0A7H	P47EICFG	P47 port interrupt control register
F0B8H	P50EICFG	P50 port interrupt control register
F0B9H	P51EICFG	P51 port interrupt control register
F0BAH	P52EICFG	P52 port interrupt control register
F0BBH	P53EICFG	P53 port interrupt control register
F0BCH	P54EICFG	P54 port interrupt control register
F0BDH	P55EICFG	P55 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 allocation register
F0C3H	PS_T0G	Timer0 gate input port allocation register
F0C4H	PS_T1	Timer1 external clock input port allocation register
F0C5H	PS_T1G	Timer1 gate input port allocation register
F0C6H	PS_T2	Timer2 external event or gate input port allocation register
F0C7H	PS_T2EX	Timer2 automatic reloading on falling edge input port allocation register
F0C8H	PS_CAP0	Timer2 input capture channel 0 port allocation register
F0C9H	PS_CAP1	Timer2 input capture channel 1 port allocation register
F0CAH	PS_CAP2	Timer2 input capture channel 2 port allocation register
F0CBH	PS_CAP3	Timer2 input capture channel 3 port allocation register
F0CCH	PS_ADET	ADC external trigger input port allocation register
F0CDH	PS_FB	PWM external brake signal port allocation register
--	--	--
F120H	PWMCON	PWM control register
F121H	PWMOE	PWM output enable register
F122H	PWMPINV	PWM output polarity select register
F123H	PWM01PSC	PWM0/1 clock pre-division control register
F124H	PWM23PSC	PWM2/3 clock pre-division control register
F125H	PWM45PSC	PWM4/5 clock pre-division control register
F126H	PWMCNTE	PWM count start control register
F127H	PWMCNTM	PWM count mode select register
F128H	PWMCNTCLR	PWM counter cleared control register
F129H	PWMLOADEN	PWM load enable control register
F12AH	PWM0DIV	PWM0 clock division control register
F12BH	PWM1DIV	PWM1 clock division control register
F12CH	PWM2DIV	PWM2 clock division control register
F12DH	PWM3DIV	PWM3 clock division control register
F12EH	PWM4DIV	PWM4 clock division control register
F12FH	PWM5DIV	PWM5 clock division control register
F130H	PWMP0L	PWM0 Low 8 bits of period data register
F131H	PWMP0H	PWM0 High 8 bits of period data register

Address	Symbol	Direction
F132H	PWMP1L	PWM1 Low 8 bits of period data register
F133H	PWMP1H	PWM1 High 8 bits of period data register
F134H	PWMP2L	PWM2 Low 8 bits of period data register
F135H	PWMP2H	PWM2 High 8 bits of period data register
F136H	PWMP3L	PWM3 Low 8 bits of period data register
F137H	PWMP3H	PWM3 High 8 bits of period data register
F138H	PWMP4L	PWM4 Low 8 bits of period data register
F139H	PWMP4H	PWM4 High 8 bits of period data register
F13AH	PWMP5L	PWM5 Low 8 bits of period data register
F13BH	PWMP5H	PWM5 High 8 bits of period data register
--	--	--
F140H	PWMD0L	PWM0 Low 8 bits of the compare data register
F141H	PWMD0H	PWM0 High 8 bits of the compare data register
F142H	PWMD1L	PWM1 Low 8 bits of the compare data register
F143H	PWMD1H	PWM1 High 8 bits of the compare data register
F144H	PWMD2L	PWM2 Low 8 bits of the compare data register
F145H	PWMD2H	PWM2 High 8 bits of the compare data register
F146H	PWMD3L	PWM3 Low 8 bits of the compare data register
F147H	PWMD3H	PWM3 High 8 bits of the compare data register
F148H	PWMD4L	PWM4 Low 8 bits of the compare data register
F149H	PWMD4H	PWM4 High 8 bits of the compare data register
F14AH	PWMD5L	PWM5 Low 8 bits of the compare data register
F14BH	PWMD5H	PWM5 High 8 bits of the compare data register
--	--	--
F150H	PWMDD0L	PWM0 Low 8 bits of asymmetric down compare data register
F151H	PWMDD0H	PWM0 High 8 bits of asymmetric down compare data register
F152H	PWMDD1L	PWM1 Low 8 bits of asymmetric down compare data register
F153H	PWMDD1H	PWM1 High 8 bits of asymmetric down compare data register
F154H	PWMDD2L	PWM2 Low 8 bits of asymmetric down compare data register
F155H	PWMDD2H	PWM2 High 8 bits of asymmetric down compare data register
F156H	PWMDD3L	PWM3 Low 8 bits of asymmetric down compare data register
F157H	PWMDD3H	PWM3 High 8 bits of asymmetric down compare data register
F158H	PWMDD4L	PWM4 Low 8 bits of asymmetric down compare data register
F159H	PWMDD4H	PWM4 High 8 bits of asymmetric down compare data register
F15AH	PWMDD5L	PWM5 Low 8 bits of asymmetric down compare data register
F15BH	PWMDD5H	PWM5 High 8 bits of asymmetric down compare data register
F15CH	PWMBRKC	PWM Brake recovery control register
F15DH	PWMBRKRDTL	PWM Low 8 bits of delay recovery data register
F15EH	PWMBRKRDTH	PWM High 8 bits of delay recovery data register
--	--	--
F160H	PWMDTE	PWM Programmable dead zone delay control register
F161H	PWM01DT	PWM0/PWM1 Programmable dead zone delay time register

Address	Symbol	Direction
F162H	PWM23DT	PWM2/PWM3 Programmable dead zone delay time register
F163H	PWM45DT	PWM4/PWM5 Programmable dead zone delay time register
F164H	PWMMASKE	PWM mask enable control register
F165H	PWMMASKD	PWM mask data register
F166H	PWMFBKC	PWM brake control register
F167H	PWMFBKD	PWM brake data register
F168H	PWMPIE	PWM period interrupt enable register
F169H	PWMZIE	PWM zero point Interrupt enable register
F16AH	PWMUIE	PWM comparie up Interrupt enable register
F16BH	PWMDIE	PWM compare down Interrupt enable register
F16CH	PWMPIF	PWM period interrupt flag register
F16DH	PWMZIF	PWM zero point interrupt flag register
F16EH	PWMUIF	PWM Compare up interrupt flag register
F16FH	PWMDIF	PWM Compare down interrupt flag register
--	--	--
F500H	C0CON0	comparators 0 control register 0
F501H	C0CON1	comparators 0 control register 1
F502H	C0CON2	comparators 0 control register 2
F503H	C1CON0	comparators 1 control register 0
F504H	C1CON1	comparators 1 control register 1
F505H	C1CON2	comparators 1 control register 2
F506H	CNVRCON	comparators reference voltage control register
F507H	CNFBCON	comparators brake control register
F508H	CNIE	comparators Interrupt enable register
F509H	CNIF	comparators interrupt flag register
F50AH	C0ADJE	comparators 0 adjustment bit select register
F50BH	C1ADJE	comparators 1 adjustment bit select register
F50CH	C0HYS	comparators 0 hysteresis control register
F50DH	C1HYS	comparators 1 hysteresis control register
F50EH	C0CON3	comparators 0 control register 3
F50FH	C1CON3	comparators 1 control register 3
--	--	--
F5C0H	BRTCON	BRT module control register
F5C1H	BRTDL	BRT timer low 8 bits of the data load value
F5C2H	BRTDH	BRT timer high 8 bits of the data load value
--	--	--
F5C4H	BRT1CON	BRT1 module control register
F5C5H	BRT1DL	BRT1 timer low 8 bits of the data load value
F5C6H	BRT1DH	BRT1 timer high 8 bits of the data load value
--	--	--
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>

Address	Symbol	Direction
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	LCDSEG0	LCD SEG0 register
F651H	LCDSEG1	LCD SEG1 register
F652H	LCDSEG2	LCD SEG2 register
F653H	LCDSEG3	LCD SEG3 register
F654H	LCDSEG4	LCD SEG4 register
F655H	LCDSEG5	LCD SEG5 register
F656H	LCDSEG6	LCD SEG6 register
F657H	LCDSEG7	LCD SEG7 register
F658H	LCDSEG8	LCD SEG8 register
F659H	LCDSEG9	LCD SEG9 register
F65AH	LCDSEG10	LCD SEG10 register
F65BH	LCDSEG11	LCD SEG11 register
F65CH	LCDSEG12	LCD SEG12 register
F65DH	LCDSEG13	LCD SEG13 register
F65EH	LCDSEG14	LCD SEG14 register
F65FH	LCDSEG15	LCD SEG15 register
F660H	LCDSEG16	LCD SEG16 register
F661H	LCDSEG17	LCD SEG17 register
F662H	LCDSEG18	LCD SEG18 register
F663H	LCDSEG19	LCD SEG19 register
F664H	LCDSEG20	LCD SEG20 register
F665H	LCDSEG21	LCD SEG21 register
F666H	LCDSEG22	LCD SEG22 register
F667H	LCDSEG23	LCD SEG23 register
F668H	LCDSEG24	LCD SEG24 register
F669H	LCDSEG25	LCD SEG25 register
F66AH	LCDSEG26	LCD SEG26 register
F66BH	LCDSEG27	LCD SEG27 register
F66CH	LCDSEG28	LCD SEG28 register
F66DH	LCDSEG29	LCD SEG29 register
F66EH	LCDSEG30	LCD SEG30 register
F66FH	LCDSEG31	LCD SEG31 register
F670H	LCDSEG32	LCD SEG32 register
F671H	LCDSEG33	LCD SEG33 register
F672H	LCDSEG34	LCD SEG34 register
F673H	LCDSEG35	LCD SEG35 register
F674H	LCDSEG36	LCD SEG36 register
F675H	LCDSEG37	LCD SEG37 register
F676H	LCDSEG38	LCD SEG38 register
F677H	LCDSEG39	LCD SEG39 register
--	--	--
F680H	LCDCON0	LCD control register 0

Address	Symbol	Direction
F681H	LCDCON1	LCD control register 1
F682H	LCDCON2	LCD control register 2
F683H	LCDCON3	LCD control register 3
F684H	LCDCOMEN	LCD COM port enable register（COM7-COM0）
F685H	LCDSEGEN0	LCD SEG port enable register0（SEG7-SEG0）
F686H	LCDSEGEN1	LCD SEG port enable register1（SEG15-SEG8）
F687H	LCDSEGEN2	LCD SEG port enable register2（SEG23-SEG16）
F688H	LCDSEGEN3	LCD SEG port enable register3（SEG31-SEG24）
F689H	LCDSEGEN4	LCD SEG port enable register 4（SEG39-SEG32）
F68AH	LCDCON4	LCD control register 4
--	--	--
F690H	LVDCON	Power monitoring register
--	--	--
F692H	ADCLDO	ADC reference voltage control register
F693H	LVDEICFG	Power Monitor Control Register
F694H	LSECRL	LSE timer data register low 8 bits
F695H	LSECRH	LSE timer data register high 8 bits
F696H	LSECON	LSE timer control register
F697H	XTSCM	Crystal oscillator Stop Detection Control Register
F698H	PS_SCLK	SPI clock input port allocation register
F699H	PS_MOSI	SPI slave input port allocation register
F69AH	PS_MISO	SPI master input port allocation register
F69BH	PS_NSS	SPI chip select input port allocation register
F69CH	PS_SCL	IIC clock input port allocation register
F69DH	PS_SDA	IIC data input port allocation register
F69EH	PS_RXD1	UART1 data input port allocation register
F704H	SMODECON0	Idle Mode Control Register 0
F705H	SMODECON1	Idle Mode Control Register 1
F706H	PCRCDL	Low 8 bits of program CRC operation result data register
F707H	PCRCDH	High 8 bits of program CRC operation result data register
F708H	CRCIN	CRC module data input register
F709H	CRCDL	CRC the low 8 bits of the operation result data register
F70AH	CRCDH	CRC the high 8-bit data register of the operation result
--	--	--
F710H	LEDP01_P00	LED SEG port P00/P01 drive source current select register
F711H	LEDP03_P02	LED SEG port P02/P03 drive source current select register
F712H	LEDP05_P04	LED SEG port P04/P05 drive source current select register
F713H	LEDP07_P06	LED SEG port P06/P07 drive source current select register
F714H	LEDP11_P10	LED SEG port P10/P11 drive source current select register
F715H	LEDP13_P12	LED SEG port P12/P13 drive source current select register
F716H	LEDP15_P14	LED SEG port P14/P15 drive source current select register
F717H	LEDP17_P16	LED SEG port P16/P17 drive source current select register
F718H	LEDP21_P20	LED SEG port P20/P21 drive source current select register
F719H	LEDP23_P22	LED SEG port P22/P23 drive source current select register
F71AH	LEDP25_P24	LED SEG port P24/P25 drive source current select register
F71BH	LEDP27_P26	LED SEG port P26/P27 drive source current select register

Address	Symbol	Direction
F71CH	LEDP31_P30	LED SEG port P30/P31 drive source current select register
F71DH	LEDP33_P32	LED SEG port P32/P33 drive source current select register
F71EH	LED_SEG_CUREN	LED SEG port current drive total enable
--	--	--
F740H	LEDC0DATA0	LED COM0 corresponding SEG7-SEG0 data register
F741H	LEDC0DATA1	LED COM0 corresponding SEG15-SEG8 data register
F742H	LEDC0DATA2	LED COM0 corresponding SEG23-SEG16 data register
F743H	LEDC0DATA3	LED COM0 corresponding SEG27-SEG24 data register
F744H	LEDC1DATA0	LED COM1 corresponding SEG7-SEG0 data register
F745H	LEDC1DATA1	LED COM1 corresponding SEG15-SEG8 data register
F746H	LEDC1DATA2	LED COM1 corresponding SEG23-SEG16 data register
F747H	LEDC1DATA3	LED COM1 corresponding SEG27-SEG24 data register
F748H	LEDC2DATA0	LED COM2 corresponding SEG7-SEG0 data register
F749H	LEDC2DATA1	LED COM2 corresponding SEG15-SEG8 data register
F74AH	LEDC2DATA2	LED COM2 corresponding SEG23-SEG16 data register
F74BH	LEDC2DATA3	LED COM2 corresponding SEG27-SEG24 data register
F74CH	LEDC3DATA0	LED COM3 corresponding SEG7-SEG0 data register
F74DH	LEDC3DATA1	LED COM3 corresponding SEG15-SEG8 data register
F74EH	LEDC3DATA2	LED COM3 corresponding SEG23-SEG16 data register
F74FH	LEDC3DATA3	LED COM3 corresponding SEG27-SEG24 data register
F750H	LEDC4DATA0	LED COM4 corresponding SEG7-SEG0 data register
F751H	LEDC4DATA1	LED COM4 corresponding SEG15-SEG8 data register
F752H	LEDC4DATA2	LED COM4 corresponding SEG23-SEG16data register
F753H	LEDC4DATA3	LED COM4 corresponding SEG27-SEG24data register
F754H	LEDC5DATA0	LED COM5 corresponding SEG7-SEG0 data register
F755H	LEDC5DATA1	LED COM5 corresponding SEG15-SEG8 data register
F756H	LEDC5DATA2	LED COM5 corresponding SEG23-SEG16 data register
F757H	LEDC5DATA3	LED COM5 corresponding SEG27-SEG24 data register
F758H	LEDC6DATA0	LED COM6 corresponding SEG7-SEG0 data register
F759H	LEDC6DATA1	LED COM6 corresponding SEG15-SEG8 data register
F75AH	LEDC6DATA2	LED COM6 corresponding SEG23-SEG16 data register
F75BH	LEDC6DATA3	LED COM6 corresponding SEG27-SEG24 data register
F75CH	LEDC7DATA0	LED COM7 corresponding SEG7-SEG0 data register
F75DH	LEDC7DATA1	LED COM7 corresponding SEG15-SEG8 data register
F75EH	LEDC7DATA2	LED COM7 corresponding SEG23-SEG16 data register
F75FH	LEDC7DATA3	LED COM7 corresponding SEG27-SEG24 data register
F760H	LEDCOMEN	LED COM7~COM0 enable control register
F761H	LEDSEGEN0	LED SEG7-SEG0 enable register 0
F762H	LEDSEGEN1	LED SEG15-SEG8 enable register 1
F763H	LEDSEGEN2	LED SEG23-SEG16 enable register 2
F764H	LEDSEGEN3	LED SEG27-SEG24 enable register 3
F765H	LEDCON	LED control register
F766H	LEDCLKL	LED low 8 bits of clock prescaler data register
F767H	LEDCLKH	LED high 8 bits of clock prescaler data register
F768H	LEDCOMTIME	LED COM port valid time select register

3. Reset

The reset time refers to the time from the chip reset to the chip starting to execute instructions, and its default design value is about 16ms. This time includes the oscillator start-up time and configuration time. Whether the chip is power-on reset or reset caused by other reasons, there will be this reset time. In addition, when the oscillator is selected as an external low-speed crystal oscillation (32.768KHz), the reset time (including the start-up time) is about 1.5s by default (external capacitor 10pF~22pF).

The chip can be reset in the following ways:

- Power-on reset;
- External reset;
- Low voltage reset;
- Watchdog overflow reset;
- Window watchdog reset;
- Software reset;
- Internal CONFIG State protection reset.
- Power-on configuration monitoring reset.

When any of the above resets occurs, all system registers will return to the default state, the program will stop running, and program counter PC will be cleared at the same time. After the reset, the program will start to run from the reset vector 0000H.

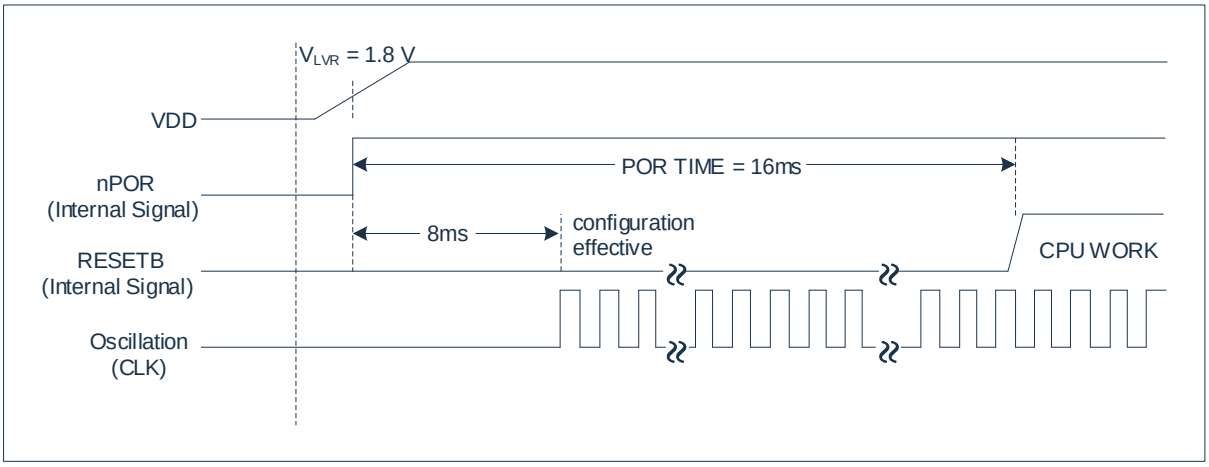
Any reset situation requires a certain response time, and the system provides a complete reset process to ensure the smooth progress of the reset action.

3.1 Power-On Reset

Power-on reset is closely related to LVR operation. The power-on process of the system is in the form of a gradually rising curve, and it takes a certain time to reach the normal level value. The normal sequence of Power-on reset is given below:

- Power on: The system detects the power supply voltage rise and waits for it to stabilize;
- System initialization: all system registers are set to initial values;
- Oscillator starts to work: the oscillator starts to provide the system clock;
- Executing the program: the end of the power-on, the program starts to run.

The Stabilization Time defaults to 16ms. If the configuration selects 32.768KHz crystal oscillator, the stabilization time is about 1.5s. Power-on reset timing diagram is shown in the figure below:



Whether the system is power-on reset can be judged by PORF (WDCON.6) flag bit. The reset types that can set the PORF flag to 1 are: power-on reset, LVR reset, power-on configuration monitoring, external reset, CONFIG protection reset, Window WDT reset.

The relationship between reset flag and reset is as below:

Reset / Flag	Power-on reset	LVR reset	Power-on configuration monitoring	CONFIG Protection reset	Software reset	External reset	WDT reset	Window WDT reset
SWRST	0	0	0	0	1	0	Unaffected	0
PORF	1	1	1	1	Unaffected	1	Unaffected	1
EXTIF	0	0	Unaffected	Unaffected	Unaffected	1	Unaffected	Unaffected
FXTIF	0	0	Unaffected	1	Unaffected	Unaffected	Unaffected	Unaffected
WDTRF	0	0	0	0	Unaffected	0	1	0
WWDTRF	0	0	Unaffected	Unaffected	Unaffected	Unaffected	Unaffected	1

0x97	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WDCON	SWRST	PORF	EXTIF	FIXIF	WDTIF	WDTRF	WDTRE	WDTCLR
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 SWRST: Software reset control bit;
1: Perform system Software reset (write 0 to clear after reset).
0: --

Bit6 PORF: Power-on reset flag;
1: The system is Power-on reset (write 0 to clear, no TA write sequence is required).
0: --

Bit5 EXTIF: External reset flag bit;
1= The system is external reset (write 0 to clear, no TA write sequence is required);
0= --

Bit4 FIXIF: CONFIG status protection bit reset flag bit;
1= The system is reset for the CONFIG state protection bit (write 0 to clear, no TA write sequence is required);
0= --

Bit3 WDTIF: WDT overflow interrupt flag bit;
1= WDT overflow (write 0 to clear);
0= The WDT did not overflow.

Bit2 WDTRF: WDT reset flag;
1= The system is reset by WDT (write 0 to clear);
0= The system is not reset by WDT.

Bit1 WDTRE: WDT reset enable bit;
1= Enable WDT to reset the CPU;
0= Disable WDT to reset CPU.

Bit0 WDTCLR: WDT counter clear bit;
1= Clear the WDT counter (automatically cleared by hardware);
0= Disable WDT counter (writing 0 is invalid).

3.2 External Reset

External reset refers to the reset signal from the external port (NRST), which resets the chip after being input by the Schmitt trigger. A reset request will be triggered if the NRST pin is held low for more than 4 LSI clock cycles, provided that the operating voltage range and stable oscillation conditions are met. After the internal state is initialized and the reset state becomes "1", it takes 16ms to stabilize before the internal RESETB signal becomes "1", and the program starts to execute from the vector address 0000H.

The chip reconfiguration process within the Stabilization Time is the same as the Power-on reset configuration process. External reset pin NRST and its pull-up resistor enable, configured through CONFIG.

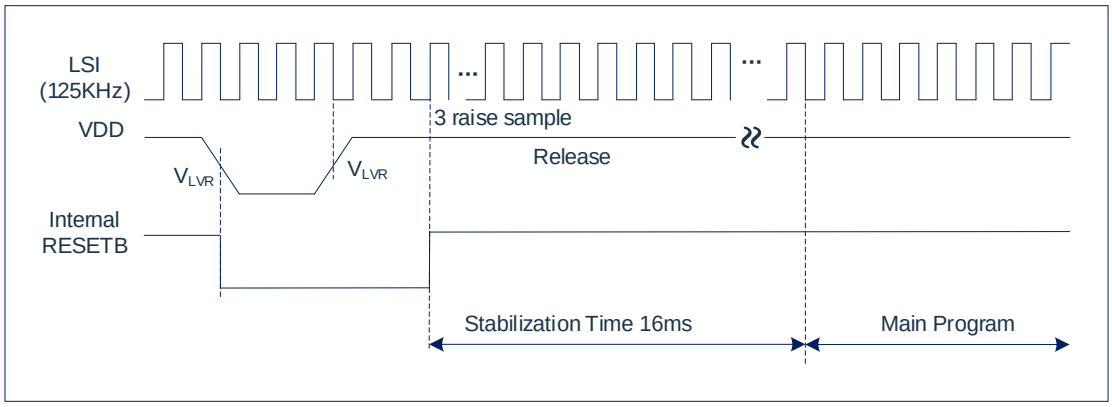
Whether the system is an external reset can be judged by the EXTIF (WDCON.5) flag bit.

3.3 LVR Low Voltage Reset

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

The LVR module detects that VDD<VLVR, and it will request a reset. In sleep mode (STOP) mode, the LVR Low voltage reset function is disabled.

The LVR Low voltage reset timing diagram is shown in the figure below:



The chip reconfiguration process within the Stabilization Time is the same as the power-on reset configuration process.

3.4 Watchdog Reset

Watchdog reset is a protection setting of the system. In the normal state, the Watchdog timer is cleared by the program. If an error occurs, the system is in an unknown state, the Watchdog timer overflows, and the system is reset at this time. After Watchdog reset, the system restarts into normal state.

The WDT counter cannot be addressed. It starts counting when the program is running after the Power-on reset ends. It is recommended to clear the WDT counter first when setting the WDT register to accurately control the WDT overflow time.

The timing of Watchdog reset is as follows:

1) Watchdog timer status: the system detects whether the Watchdog timer overflows, if it overflows, the system resets;
2) Initialization: All system registers are set to the default state;
3) Program: reset is complete, the program starts to run from 0000H;

The clock source of the WDT is provided by the system clock, and the basic time period of the WDT counter is Tsys. Reset the CPU and all registers after the WDT overflows, and the program will start executing from 0000H immediately after 1 Tsys. WDT reset will not re-configure power-on reset. The overflow time of the watchdog can be set by the program, and the overflow time can be selected in the two bits WTS2-WTS0 of the CKCON register. The watchdog overflow time is 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 and can generate interrupts.

3.5 Window Watchdog Reset

Window watchdog reset is also a protection setting for the system. Under normal conditions, the window watchdog timer is cleared by the program within the window period. If an error occurs, the system is in an unknown state, and the window watchdog is cleared outside the window period or the window watchdog timer overflows, and the system is reset at this time. After the window watchdog is reset, the system restarts into a normal state. The window watchdog can also be set to not reset the system and can generate interrupts. See the description below for details.

3.6 Software Reset

Program software reset can be implemented inside the chip. Software reset can relocate the program flow to the reset address 0000H, and then run the program again. The user can write Software reset control bit WDCON[7] (SWRST=1) to realize custom Software reset. Software reset will not re-configure power-on reset.

3.7 CONFIG State Protection Reset

CONFIG State protection reset is a strengthening protection mechanism of the system. During power-on reset, there is a set of 16-bit CONFIG registers inside, and the fixed code (A569H) set in FLASH is loaded. This register will not be operated during normal operation. If the value of the register changes and is not equal to the original fixed code in the case of a specific non-program operation, after several clock samples, the register continues to remain in the state of not being a fixed code, the system will reset.

The reset mechanism prevents the configuration bit from changing under certain conditions, causing the system to enter an unpredictable state.

In normal operation, the clock of the sampling register value is the internal RC fixed clock Fixed_Clock (8MHz, clock source from HSI) and low-power clock (LSI 125KHz). Once the register value is not a fixed code, the LSI oscillator and the LSI oscillator are forced to be enabled. HSI oscillator, and the system clock is switched to the LSI clock, if after 12 Fixed_Clock sampling or 3 LSI clock sampling, the register is still not in a fixed code state, the system will reset.

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

Whether the system is in CONFIG state reset can be judged by the FIXIF (WDCON.4) flag bit.

3.8 Power-On Configuration Monitoring Reset

During 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 enters a certain state that cannot be reconfigured, the internal monitoring circuit starts timing from the configuration, and if it exceeds the set time , The monitoring circuit resets the configuration module and allows the configuration module to perform the configuration process again. To prevent the system from entering an unpredictable state when powering up.

The working clock of the monitoring circuit is LSI (125KHz), and the default monitoring time is 65ms. If 32.768KHz crystal oscillator is selected, the monitoring time is 2.1s.

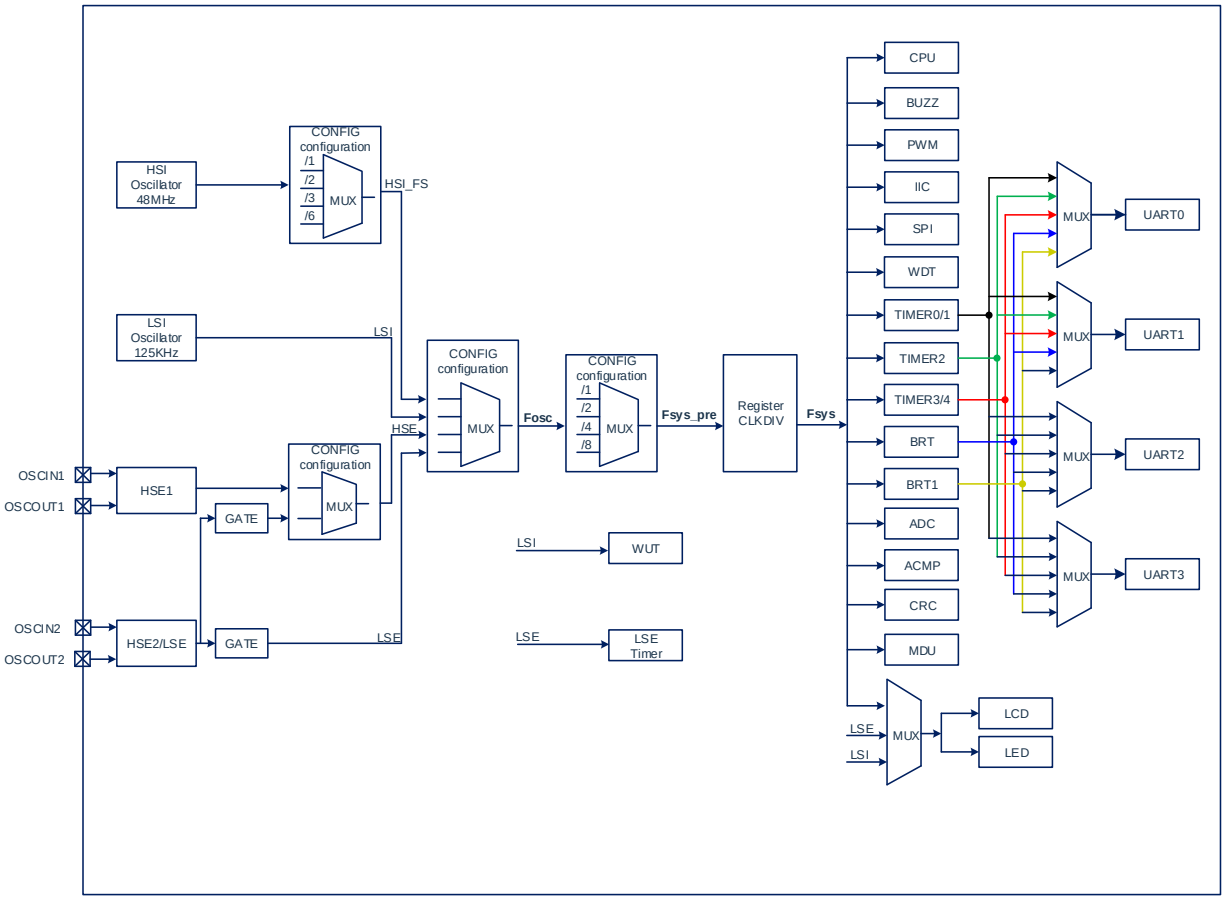
4. Clock Structure

The clock source of the system clock has 4 types, and the clock source and clock division can be selected through the system configuration register settings. The system clock source is as follows;

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

4.1 System Clock Structure

The clock structure block diagram of each Peripherals module system is shown in the figure below:



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 division bit;
00H= Fsys=Fsys_pre;
Others= Fsys=Fsys_pre/（2*CLKDIV）（2,4…510 division）.

Modify the instruction sequence 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 Select Register SCKSEL

0xD6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SCKSEL	--	--	--	SEL	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 SEL: System clock configuration selection;
1= Subject to the clock source configured by CKSEL<1:0>.
0= Subject to the clock source configured by CONFIG.
Bit3 WRITE: Write enable, generate a pulse, and switch the clock;
1= Switch clock;
0= Do not switch clocks
Bit2~Bit0 CKSEL<2:0>: System clock source selection bit;
111= LSI;
110= LSE;
101= HSE;
100= HSI;
Others= Invalid value, access forbidden.

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

```
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 stable status bit;
1= Stable;
0= Not stable.

Bit6 LSE_F: Low-speed external crystal oscillator stable status bit;
1= Stable;
0= Not stable.

Bit5 HSE_F: High-speed external crystal oscillator stable status bit;
1= Stable;
0= Not stable.

Bit4 HSI_F: High-speed internal clock stable status bit;
1= Stable;
0= Not stable.

Bit3~Bit0 -- Reserved, all must be 0.

4.2.4 System Clock Monitor Register XTSCM

F697H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
XTSCM	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:	Stop oscillation detection module enable;
	1=	Enable;
	0=	Disable.
Bit6	SCMIE:	Oscillation stop detection interrupt enable bit (this interrupt and LSE timer interrupt share an interrupt entry);
	1=	Enable;
	0=	Disable.
Bit5~Bit2	--	Reserved, all must be 0.
Bit1	SCMIF:	Stop oscillation interrupt flag bit;
	1=	Mean stop oscillation;
	0=	Cleared to 0 by software, after clearing to 0, it will automatically switch to the HSE/LSE main frequency (only software can clear it to 0).
Bit0	SCMSTA:	Stop oscillation status bit, read only;
	1=	Mean stop oscillation;
	0=	Stop oscillation recover.

Directions:

1) Both SCMIF and SCMSTA can reflect the state of HSE/LSE as system clock. The biggest difference between the two is that when HSE/LSE stops oscillating, SCMSTA will remain in a high level state until HSE/LSE recovers; SCMIF can also reflect that HSE/LSE stops oscillating, but it can generate an interrupt (interrupt enable is required), and SCMIF can also be cleared through the register. After clearing, the main frequency will switch back to HSE/LSE (if it is still in the stop oscillation state at this time, the interrupt will be triggered again).

2) After the oscillation is stopped, the main frequency will be switched from HSE/LSE to HSI. If HSE/LSE is restored, SCMSTA will be automatically cleared, and the main frequency will also be automatically switched back to HSE/LSE from HSI.

4.2.5 Function Clock Control Register

Watchdog overflow time/timer clock source select register CKCON

0x8E	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CKCON	WTS2	WTS1	WTS0	T1M	T0M	--	--	T0CNTM
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 bit;
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 clock source selection bit;
0= Fsys/12;
1= Fsys/4.

Bit3 T0M: Timer0 clock source selection bit;
0= Fsys/12;
1= Fsys/4.

Bit2~Bit1 -- Reserved, all must be 1.

Bit0 T0CNTM: Timer0 count source selection bit;
0= PWM0 output;
1= T0 pin input;

UART0/1Baud rate select register FUNCCR

0x91	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FUNCCR	--	UART1_CKS2	UART1_CKS1	UART1_CKS0	--	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~Bit4 UART1_CKS<2:0>: UART1 timer clock source selection;
000= Timer1 Overflow clock;
001= Timer4 Overflow clock;
010= Timer2 Overflow clock;
011= BRT Overflow clock;
100= BRT1 Overflow clock;
Others= Disable Access.

Bit3 -- Reserved, must be 0.

Bit2~Bit0 UART0_CKS<2:0>: UART0 timer clock source selection;
000= Timer1 Overflow clock;
001= Timer4 Overflow clock;
010= Timer2 Overflow clock;
011= BRT Overflow clock;
100= BRT1 Overflow clock;
Others= Disable Access.

UART2/3Baud rate select register FUNCCR1

0xE2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FUNCCR1	--	UART3_CKS2	UART3_CKS1	UART3_CKS0	--	UART2_CKS2	UART2_CKS1	UART2_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~Bit4 UART3_CKS<2:0>: UART3 timer clock source selection;

000= Timer1 Overflow clock;

001= Timer4 Overflow clock;

010= Timer2 Overflow clock;

011= BRT Overflow clock;

100= BRT1 Overflow clock;

Others= Disable Access.

Bit3 -- Reserved, must be 0.

Bit2~Bit0 UART2_CKS<2:0>: UART2 timer clock source selection;

000= Timer1 Overflow clock;

001= Timer4 Overflow clock;

010= Timer2 Overflow clock;

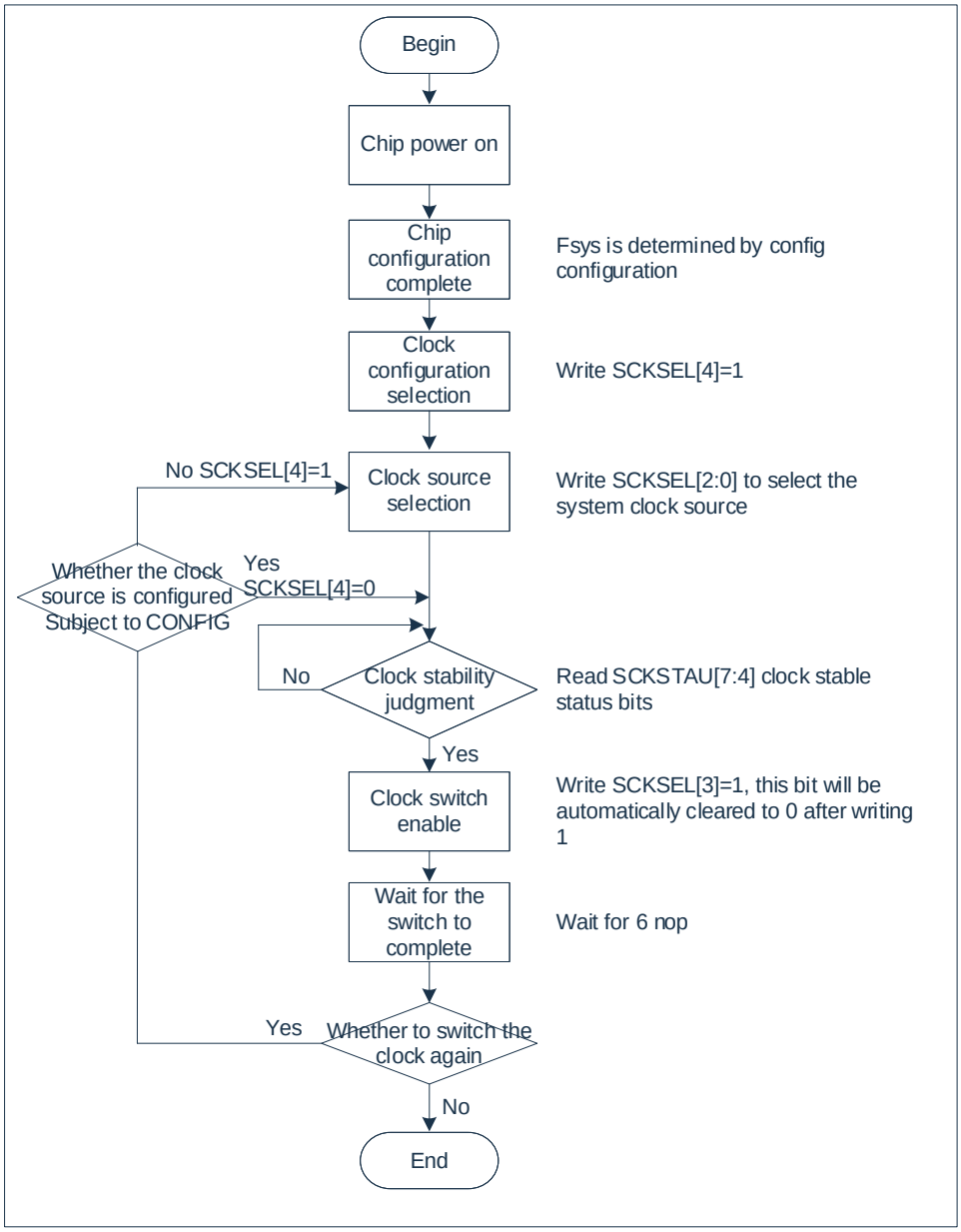
011= BRT Overflow clock;

100= BRT1 Overflow clock;

Others= Disable Access.

4.3 System Clock Switching

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



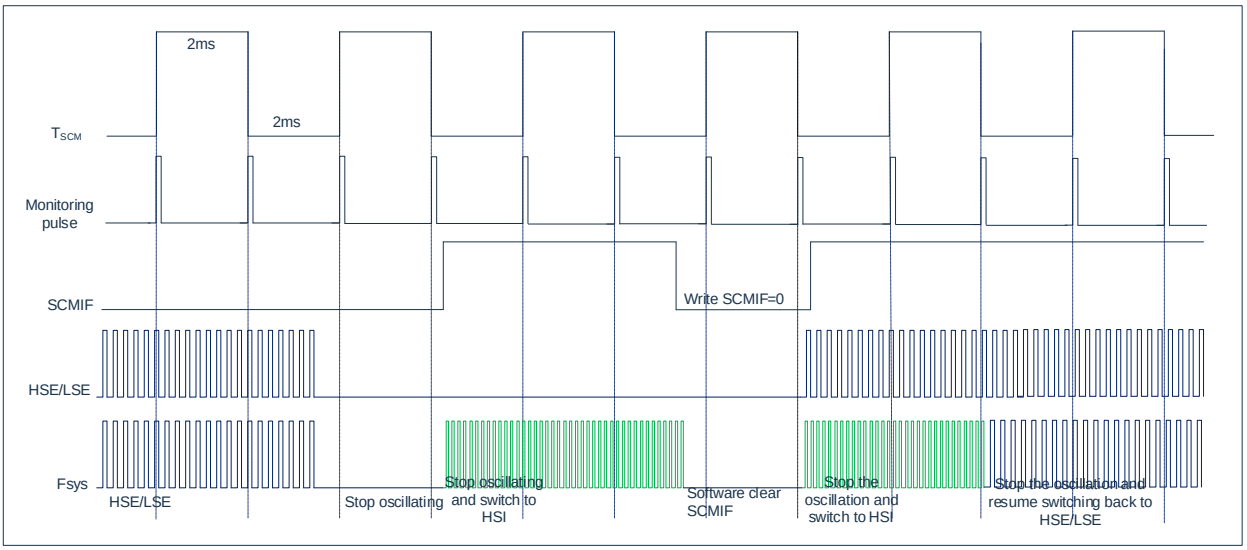
Note: When the system clock source is switched, it must be ensured that the corresponding clock source exists and has been correctly configured.

4.4 System Clock Monitoring

System clock monitoring is a monitoring protection circuit designed to prevent the system from not working due to the stop of the crystal oscillator. When HSE/LSE is used as the system clock, once the HSE/LSE clock is detected to stop, the system will force start the HSI clock source. After the HSI is stable, the system will run at 8MHz. Will automatically switch back from HSI to HSE/LSE.

The SCM module monitors the system clock HSE/LSE every 4ms, and the duty cycle of the T_{SCM} during the monitoring period is 1:1. SCM monitors HSE/LSE oscillation when T_{SCM} is at high level, and processes the monitoring results when T_{SCM} is at low level. If HSE/LSE oscillation is detected, the system clock will be switched to HSI, and oscillation will be interrupted at the same time. The flag bit SCMIF is set. If SCMIF is cleared, even if HSE/LSE has stopped oscillating, the system clock will automatically switch back to HSE/LSE.

The block diagram of the system clock monitoring structure is shown in the figure below:



5. Power Management

Low power consumption modes are divided into two categories:

- IDLE: idle mode

  There are two modes of idle mode 1 and idle mode 2 in this mode, and the mode selection is controlled by the register SMODECON0/SMODECON1.

- STOP: sleep mode

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

enter idle mode: IDLE();

enter sleep mode: STOP();

In different operating modes, the available clocks and wake-up sources are as follows:

Contrast	Normal work	Idle mode		Sleep mode
		Idle mode1	Idle mode2	
Definition	MCU works normally; CP works normally; Peripherals operate normally; LDO on; FLASH on.	MCU idle mode 1; CPU stops working; Digital peripherals operate normally; Analog peripherals operate normally; LDO on; FLASH on.	MCU idle mode 2; CPU stops working; Digital peripherals operate normally; The analog peripheral enable bits will be forced off (except LCD); LDO is closed; FLASH is closed.	MCU sleep mode; CPU stops working; Digital peripherals stop working; Analog peripherals are turned off by software; LDO is closed; FLASH is closed.
Entry conditions	After the system reset is completed, the chip enters the normal working mode	SMODECON0=0x00; SMODECON1=0x00; IDLE(); Enter idle mode 1.	When the system clock is LSI or LSE, SMODECON0=0x55; SMODECON1=0xAA; IDLE();Enter idle mode 2.	STOP(); Enter sleep mode.
Wake up source	--	All interrupts	Interrupts other than ADC and ACMP interrupts	WUT timed wake-up; INT0/1 interrupt wake-up; GPIO interrupt wake-up; LSE timed wake-up; WWDT timed wake-up.
System available clock	--	All clocks	The system clock is LSI or LSE	No clock available
Post wake mode	--	The MCU returns to normal mode, and the program runs normally	The MCU returns to normal mode, and the program runs normally	The MCU returns to normal mode, and the program runs normally
Wake up wait time	--	Run now	Wake-up wait time set by CONFIG	Wake-up wait time set by CONFIG
LVR low voltage reset	Support	Support	Support	Support
External reset	Support	Support	Support	Support
Window watchdog reset	Support	Support	Support	Support
Watchdog reset	Support	Support	Support	Not support

5.1 Power Management Registers

5.1.1 Power Management Register PCON

0x87	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON	SMOD0	SMOD1	--	--	--	SWE	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

	SMOD0:	UART0 baud rate multiplier bit;
Bit7	0=	UART0 baud rate is normal;
	1=	Double the baud rate of UART0.
Bit6	SMOD1:	UART1 baud rate multiplication bit;
	0=	UART1 baud rate is normal;
	1=	UART1 baud rate is doubled.
Bit5~Bit3	--	Reserved, all must be 0's.
Bit2	SWE:	STOP status function wake-up enable bit; (Regardless of the value of SWE, the system can be restarted by power-off reset or enabled external reset)
	0=	Function wake-up is disabled;
	1=	Function wake-up is enabled (can be waked up by external interrupt and timed) wake).
Bit1	STOP:	Sleep state control bit;
	0=	Not enter the dormant state;
	1=	Enter the dormant state (automatically cleared when exiting STOP mode).
Bit0	IDLE:	Idle state control bit;
	0=	Not enter idle state;
	1=	Enter idle state (automatically cleared when exiting IDLE mode).

5.1.2 Idle Mode Control Register SMODECON0

F704H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SMODECON0	SMODE07	SMODE06	SMODE05	SMODE04	SMODE03	SMODE02	SMODE0E1	SMODE00
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	SMODE0<7:0>:	Idle mode control register 0 (wake up from idle mode, this register is cleared by hardware);

5.1.3 Idle Mode Control Register SMODECON1

F705H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SMODECON1	SMODE17	SMODE16	SMODE15	SMODE14	SMODE13	SMODE12	SMODE11	SMODE10
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	SMODE1<7:0>:	Idle mode configuration register 1 (wake up from idle mode, this register is cleared by hardware); When { SMODE1，SMODE0}=0xaa55，the idle mode 2 mode is enabled; When { SMODE1，SMODE0}=other values, the idle mode 1 mode is enabled.

5.2 Power Monitoring Register

This MCU has its own power detection function. If the LVD module is enabled (LVDEN=1) and the voltage monitoring point LVDSEL is set at the same time, voltage detection edge LVDEICFG, when the power supply voltage drops below the LVD setting value or the power supply voltage is higher than the LVD setting value, an trigger will be generated to remind the user.

If the LVD module is enabled before sleeping, the hardware will not close the module circuit after entering sleeping, and the software needs to be closed (LVDEN=0).

5.2.1 Power Monitoring Register LVDCON

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

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

0000=	2.00V;	1000=	3.20V
0001=	2.15V;	1001=	3.40V
0010=	2.30V;	1010=	3.60V
0011=	2.45V;	1011=	3.80V
0100=	2.60V;	1100=	4.00V
0101=	2.75V;	1101=	4.20V
0110=	2.90V;	1110=	4.40V
0111=	3.05V;	1111=	4.60V

Bit3 LVDEN: LVD module enable;

0= Disable;

1= Enable.

Bit2 LVDOUT Supply voltage monitoring bit.

0= The power supply voltage is higher than the monitoring voltage;

1= The supply voltage is lower than the monitoring voltage.

Bit1 -- Reserved, must be 0.

Bit0 LVDINTF: LVD trigger flag bit;

0= No trigger is generated;

1= Trigger generation (cleared by software).

5.2.2 Power Monitor Control Register LVDEICFG

F693H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LVDEICFG	--	--	--	--	--	--	LVDEICFG1	LVDEICFG0
R/W	R	R	R	R	R	R	R/W	R/W
Reset value	0	0	0	0	0	0	0	1

Bit7~Bit2	--	Reserved, all must be 0.
Bit1~Bit0	LVDEICFG<1:0>:	Power monitor trigger edge control bit;
	00=	Trigger disable;
	01=	The power supply voltage is lower than the monitoring voltage to trigger;
	10=	The power supply voltage is higher than the monitoring voltage to trigger;
	11=	Triggered when the supply voltage is lower than the monitoring voltage or the supply voltage is higher than the monitoring voltage.

5.3 IDLE Mode

The idle mode is divided into two modes: idle mode 1 and idle mode 2. Which mode is selected is controlled by the register SMODECON0/SMODECON1. The two modes are configured as follows:

Idle mode 1:

SMODECON0=0X00;

SMODECON1=0X00;

IDLE();

Idle mode 2:

SMODECON0=0X55;

SMODECON1=0XAA;

IDLE();

5.3.1 Idle Mode 1

In this mode, only the CPU clock source is turned off. Therefore, in this state, peripheral functions (such as timers, PWM, and I2C) and clock generators still work normally.

After the system enters idle mode 1, it can be waked-up by any interrupt. After waking-up, it enters the interrupt handler. After the interrupt returns, it continues to execute the instruction after the sleep operation.

If Idle Mode 1 is entered in the Interrupt Service Routine, the system can only be waked-up by a higher priority interrupt.

5.3.2 Idle Mode 2

The chip can be configured in idle mode 2 only when the chip uses LSI or LSE as the master clock. In idle mode 2, the enable bit of the analog peripherals will be forced to close, the CPU instruction fetch is stopped, the LDO is in low power mode, and other digital peripherals can work normally. In this mode the system can be woken up by any interrupt except ADC and ACMP. After the idle mode 2 wakes up, it enters the interrupt handler. After the interrupt returns, it continues to execute the instruction after the sleep operation.

If Idle Mode 2 is entered in the interrupt service routine, the system can only be woken up by a higher priority interrupt.

5.3.2.1 Wake Up Wait State

In idle mode 2, after the interrupt is generated, it still needs to wait for a period of time to wake up the system and execute the next instruction of the program. The waiting time for waking up the CPU is set in the programming CONFIG, and the waiting time can be set to 50us~1s.

If the watchdog reset is enabled in the program before idle mode 2, when the watchdog overflow interrupt is generated, regardless of whether the global interrupt enable bit EA is 1, it will wake up from idle mode 2, and there is no wake-up wait time.

5.3.2.2 Wake-Up Time

The total wake-up time to wake up the system using an external interrupt is:

Power manager stabilization time (200us) + wake-up wait time

The total wake-up time of the system using timing wake-up is:

Power manager stabilization time (200us) + wake-up timer timing + wake-up waiting time

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

5.3.2.3 Reset Operation

In idle mode 2, the system can also be restarted by power-down reset, external reset, WDT reset, and WWDT reset.

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

External reset: The external reset function needs to be enabled, and the related port must be configured as a dedicated reset pin. When the reset pin remains low for more than 4 LSI clock cycles, the system will generate a reset. Releasing the reset pin will cause the system to restart.

WDT reset: After the WDT reset is enabled, the WDT overflow interrupt will be forced to wake up from idle mode 2, and then the system will generate a watchdog reset.

5.4 STOP sleep Mode

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

Note: When the chip is used in this operating mode, the LVR voltage must be set to 1.8V. If the LVR voltage point is selected, it may cause the chip to reset abnormally when waking up from the sleep mode.

5.4.1 Sleep Wake Up

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

1) INT0/1 interrupt

Using the INT0/1 interrupt to wake up the sleep mode, you must turn on the total interrupt enable and the INT0/1 interrupt enable before entering the sleep mode to wake up the system. INT0, INT1 interrupt related registers include IE, IP, TCON, IO multiplexing mapping registers, INT0/1 interrupt wake-up can only wake up from sleep by falling edge interrupt.

2) External (GPIO) interrupt

Using external GPIO interrupt to wake up, you must turn on the total interrupt enable and port interrupt enable before entering the sleep mode to wake up the system. The external GPIO interrupt wake-up can choose rising edge, falling edge, and both edge interrupt wake-up sleep. The interrupt wake-up edge is set by the external interrupt control register PxnEICFG.

3) WUT wakes up regularly

To wake up regularly by WUT, the timed wake-up function must be turned on before going to sleep, and the time from sleep state to wake-up must be set at the same time. The clock source of the timing wake-up circuit is provided by the LSI (low-power oscillator). When the timing wake-up function is enabled, the LSI is automatically turned on in the sleep state.

4) LSE timing wake-up

To wake up regularly by LSE, the LSE module enable, counting enable, and timing wake-up function must be turned on before entering sleep mode, and the time from sleep state to wake-up must be set at the same time.

5) WWDT timing wake-up

To wake up regularly by WWDT, the WWDT module must be enabled before entering sleep mode, WWDT sleep wake-up is enabled, and the time from sleep state to wake-up is set at the same time.

5.4.2 Wake-Up Wait State

Whether it is INT0/1 interrupt, external GPIO interrupt, or WUT timing wake-up, LSE timing wake-up, WWDT timing wake-up sleep mode, an interrupt is generated or after the regular time, we need to wait for some time to wake up the system, the next instruction execution of the program. After the interrupt is generated or the time is up, the system oscillator is started, but the oscillation frequency is not stable yet, the CPU is not working, and the PC still stops in the dormant state. The system needs to wait for a period of time before providing the clock to the CPU. The waiting time for waking up the CPU is set in the programming CONFIG, and the waiting time can be set to 50us~1s. After the wake-up waiting time has elapsed, the MCU considers that the system clock is stable, and then provides the clock to the CPU, and the program continues to execute.

If the internal wake-up timer and external interrupt wake-up function are both enabled, after the system enters the sleep mode, any wake-up method can wake up the CPU. If the internal timer wakes up the oscillator first, and then there is an external interrupt input, after the wake-up wait time has elapsed, the program executes the interrupt handler first and then continues to execute the instruction after the sleep operation.

5.4.3 Sleep Wake-Up Time

Use external interrupts to wake up the system. The total wake-up time is:

Power manager stability time (200us) + wake-up waiting time.

The total wake-up time of the system using timing wake-up is:

power manager stable time (200us) + wake-up timer timing +Wake-up waiting time

(The time given above is Fsys>1MHz)

5.4.4 Reset Operation During Sleep

In sleep mode, the system can also be restarted by power-down reset, external reset or WWDT reset. This restart method has nothing to do with the value of SWE. Even if SWE=0, the system can be restarted through the above reset operation.

Power-off reset: No other conditions are required. After VDD is reduced to 0V, it is powered on to the working voltage again and enters the power-on reset state.

External reset: The external reset function needs to be enabled, and the relevant ports are configured as dedicated reset ports. The reset port remains at a low level of >1us during sleep, the system resets, and the reset port is released, and the system restarts.

5.4.5 Sleep Power Consumption In Debug Mode

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

In the debug mode, after the system enters the sleep state, the related power management circuit and oscillator do not turn off, but continue to turn on. The wake-up operation can also be performed in the debug mode, and the wake-up method is the same as that in the normal mode.

Therefore, in this state, the sleep current obtained by the test is not the true sleep power consumption. It is recommended to close the debug mode after the development of the sleep wake-up function in the debug mode, and then restart the system. The measured current at this time is the actual sleep power consumption.

5.4.6 Sleep Mode Application Example

Before the system enters the sleep mode, if the user needs to obtain a smaller sleep current, please confirm the status of all I/Os. If there are floating I/O ports in the user program, set all floating ports to Output ports, ensure that each input port has a fixed state to avoid when the I/O is in the input state, the port line level is in an uncertain state and increases the sleep current; turn off the ADC module, LSE module, LVD module and others Peripherals to reduce sleep current.

Example: When using timing wake-up, enter sleep processing program (assembler)

```
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
        MOV     P3TRIS,#0FFh
        MOV     P3,#0FFh
        Operation instructions to turn off other functions
        MOV     PCON,#06H       ; Perform a functional wake-up sleep operation,
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP     ;The instruction to execute the sleep operation must be followed by 6 NOP instructions
        Other operation commands after waking up
```

6. Interrupt

6.1 Interrupt Overview

The chip has 25 interrupt sources and interrupt vectors:

Interrupt source	Interrupt described	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
UART1	TI1 or RI1	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
P4EXTIF<7:0>	P4 port external interrupt	11-0x005B	12
P5EXTIF<7:0>	P5 port external interrupt	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
I^2C	I^2C Interrupt	21-0x00AB	22
SPI	SPI interrupt	22-0x00B3	23
UART2	TI2 or RI2	23-0x00BB	24
UART3	TI3 or RI3	24-0x00C3	25
LSE_Timer	LSE timer interrupt	25-0x00CB	26
--	--	26-0x00D3	27
--	--	--	28
WWDT	Window watchdog interrupt	28-0x00E3	29

The chip stipulates two interrupt priority levels, which can realize two-level interrupt nesting. When an interrupt has been responded, if a high-level interrupt sends a request, the latter can interrupt the former to achieve interrupt nesting.

6.2 External Interrupt

6.2.1 INT0/INT1 Interrupt

Each pin of the chip supports 8051 native INT0 and INT1 external interrupts. INT0/INT1 can select falling edge or low level to trigger the interrupt. The related control register is TCON. INT0 and INT1 occupy two interrupt vectors.

6.2.2 GPIO Interrupt

Each GPIO pin of the chip supports external interrupts, and can support falling edge/rising edge/ both edge interrupts. The edge trigger type is configured through the PxNEICFG register. For example, configure port P13 as a falling edge interrupt:

```
P13CFG=0x00;        //Set P13 as GPIO
P1TRIS&=0xF7;       //Set P13 as input port
P13EICFG=0x02;      //Set P13 as falling edge trigger interrupt
```

GPIO interrupt totally occupies 6 interrupt vectors:

P0 port occupies an interrupt vector 0x003B;

P1 port occupies an interrupt vector 0x0043;

P2 port occupies an interrupt vector 0x004B;

P3 port occupies an interrupt vector 0x0053;

P4 port occupies an interrupt vector 0x005B;

P5 port occupies an interrupt vector 0x0063.

When the interrupt is generated, enter the interrupt service routine to determine which port triggered the interrupt first, and then perform the corresponding processing.

6.3 Interrupt And Sleep Wake-Up

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

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

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

After the system is awakened by an external interrupt, it first enters the interrupt service routine to process the interrupt wake-up task. After exiting the interrupt service routine, the system continues to execute the instructions after the sleep operation.

6.4 Interrupt Register

6.4.1 Interrupt Mask Register

6.4.1.1 Interrupt Mask Register IE

Interrupt Mask Register IE is a readable and writable register and can be operated in bits. When an interrupt condition occurs, regardless of the status of the corresponding interrupt enable bit or global enable bit EA, the interrupt flag bit will be set to 1. User software should ensure that the corresponding interrupt flag bit is cleared before enabling an interrupt.

0xA8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IE	EA	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 EA: Global interrupt enable bit;
1= Allow all unmasked interrupts;
0= Disable all interrupts.

Bit6 ES1: UART1 interrupt enable bit;
1= Enable UART1 interrupt;
0= Disable UART1 interrupt.

Bit5 ET2: TIMER2 total interrupt enable bit;
1= Enable all TIMER2 interrupts;
0= Disable all TIMER2 interrupts.

Bit4 ES0: UART0 interrupt enable bit;
1= Enable UART0 interrupt;
0= Disable UART0 interrupt.

Bit3 ET1: TIMER1 interrupt enable bit;
1= Enable TIMER1 interrupt;
0= Disable TIMER1 interrupt.

Bit2 EX1: External interrupt 1 interrupt enable bit;
1= Enable external interrupt 1 interrupt;
0= Disable external interrupt 1 interrupt.

Bit1 ET0: TIMER0 interrupt enable bit;
1= Enable TIMER0 interrupt;
0= Disable TIMER0 interrupt.

Bit0 EX0: External interrupt 0 interrupt enable bit;
1= Enable external interrupt 0 interrupt;
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 interrupt;
0= Disable SPI interrupt.
Bit6 I2CIE: I^2C interrupt enable bit;
1= Enable I^2C interrupt;
0= Disable I^2C interrupt.
Bit5 WDTIE: WDT interrupt enable bit;
1= Enable WDT overflow interrupt;
0= Disable WDT overflow interrupt.
Bit4 ADCIE: ADC interrupt enable bit;
1= Enable ADC interrupt;
0= Disable ADC interrupt.
Bit3 PWMIE: PWM total interrupt enable bit;
1= Allow all PWM interrupts;
0= Disable all PWM interrupts.
Bit2 -- Reserved, must be zero.
Bit1 ET4: Timer4 interrupt enable bit;
1= Enable Timer4 interrupt;
0= Disable Timer4 interrupt.
Bit0 ET3: Timer3 interrupt enable bit;
1= Enable Timer3 interrupt;
0= Disable Timer3 interrupt.

6.4.1.3 Interrupt Mask Register EIE3

0xAB	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIE3	--	--	--	--	--	--	ES3	ES2
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, all must be 0
Bit1 ES3: UART3 interrupt enable bit;
1= Allow UART3 interrupt;
0= Disable UART3 interrupt.
Bit0 ES2: UART2 interrupt enable bit;
1= Allow UART2 interrupt;
0= Disable UART2 interrupt.

6.4.1.4 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 bit;
1= Enable interrupt;
0= Disable interrupt.
Bit6 T2EXIE: Timer2 external load interrupt enable bit;
1= Enable interrupt;
0= Disable interrupt.
Bit5~Bit4 -- Reserved, all must be 0.
Bit3 T2C3IE: Timer2 compare/capture channel 3 interrupt enable bit;
1= Enable interrupt;
0= Disable interrupt.
Bit2 T2C2IE: Timer2 compare/capture channel 2 interrupt enable bit;
1= Enable interrupt;
0= Disable interrupt.
Bit1 T2C1IE: Timer2 compare/capture channel 1 interrupt enable bit;
1= Enable interrupt;
0= Disable interrupt.
Bit0 T2C0IE: Timer2 compare/capture channel 0 interrupt enable bit;
1= Enable interrupt;
0= Disable interrupt.

If the interrupt of Timer2 is enabled, the total interrupt enable bit of Timer2 ET2=1 (IE.5=1)

6.4.1.5 P0 Port 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 bit (i= 0-7);
1= Enable interrupt;
0= Disable interrupt.

6.4.1.6 P1 Port 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 port interrupt enable bit (i= 0-7);
1= Enable interrupt;
0= Disable interrupt.

6.4.1.7 P2 Port Interrupt Control Register P2EXTIE

0xAE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2EXTIE	P27IE	P26IE	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~Bit0 P2iIE: P2i port interrupt enable bit (i= 0-7);
1= Enable interrupt;
0= Disable interrupt.

6.4.1.8 P3 Port Interrupt Control Register P3EXTIE

0xAF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P3EXTIE	P37IE	P36IE	P35IE	P34IE	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~Bit0 P3iIE: P3i port interrupt enable bit (i= 0-7);
1= Enable interrupt;
0= Disable interrupt.

6.4.1.9 P4 Port Interrupt Control Register P4EXTIE

0x9B	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P4EXTIE	P47IE	P46IE	P45IE	P44IE	P43IE	P42IE	P41IE	P40IE
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 P4iIE: P4i port interrupt enable bit (i= 0-7);
1= Enable interrupt;
0= Disable interrupt.

6.4.1.10 P5 Port Interrupt Control Register P5EXTIE

0x9C	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P5EXTIE	--	--	P55IE	P54IE	P53IE	P52IE	P51IE	P50IE
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, all must be 0.

Bit5~Bit0 P5iIE: P5i port interrupt enable bit (i=0-5);

1= Enable interrupt;

0= Disable interrupt.

6.4.2 Interrupt Priority Control Register

6.4.2.1 Interrupt Priority Control Register IP

Interrupt Priority Control Register IP is a readable and writable register, which can be operated by bit.

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 PS1: UART1 interrupt priority control bit;
1= Set to high-level interrupt;
0= Set as low-level interrupt.

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

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

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

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

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

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

6.4.2.2 Interrupt Priority Control Register EIP1

0xB9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP1	PACMP	--	PP5	PP4	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 as high-level interrupt;
0= Set as low-level interrupt.
Bit6 -- Reserved, must be 0.
Bit5 PP5: P5 port interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.
Bit4 PP4: P4 port interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.
Bit3 PP3: P3 port interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.
Bit2 PP2: P2 port interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.
Bit1 PP1: P1 port interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.
Bit0 PP0: P0 port interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.

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

6.4.2.4 Interrupt Priority Control Register EIP3

0xBB	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP3	--	--	PWWDT	--	--	PLSE	PUART3	PUART2
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, all must be 0.
Bit5	PWWDT	WWDT interrupt priority control bit;
	1=	Set as high-level interrupt;
	0=	Set as low-level interrupt.
Bit4~ Bit3	--	Reserved, must be 0.
Bit2	PLSE:	LSE interrupt priority control bit;
	1=	Set as high-level interrupt;
	0=	Set as low-level interrupt.
Bit1	PUART3:	UART3 interrupt priority control bit;
	1=	Set as high-level interrupt;
	0=	Set as low-level interrupt.
Bit0	PUART2:	UART2 interrupt priority control bit;
	1=	Set as high-level interrupt;
	0=	Set as low-level interrupt.

6.4.3 Interrupt Flag Register

6.4.3.1 Timer0/1、INT0/1 Interrupt Flag 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 overflows, the hardware is automatically cleared when entering the interrupt service routine, or software Cleared;
0= Timer1 counter does not overflow.

Bit6 TR1: Timer1 running control bit;
1= Timer1 is started;
0= Timer1 is closed.

Bit5 TF0: Timer0 counter overflow interrupt flag bit;
1= Timer0 counter overflows, it is automatically cleared by hardware when entering the interrupt service routine, or can be cleared by software;
0= Timer0 counter has no overflow.

Bit4 TR0: Timer0 running control bit;
1= Timer0 starts.
0= Timer0 is turned off.

Bit3 IE1: External Interrupt 1 flag bit;
1= External Interrupt 1 generates an interrupt, which is automatically cleared by hardware when entering the interrupt service routine, or cleared by software;
0= External Interrupt 1 does not generate an interrupt.

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

Bit1 IE0: External interrupt 0 flag;
1= External interrupt 0 generates an interrupt, which is automatically cleared by hardware when entering the interrupt service routine, or can be cleared by software;
0= External interrupt 0 does not generate an interrupt.

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

6.4.3.2 Timer2 Interrupt Flag 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 overflows and needs to be cleared by software;
0= The Timer2 counter does not overflow.

Bit6 T2EXIF: Timer2 external load flag;
1= The T2EX port of Timer2 generates a falling edge, which needs to be cleared by software;
0= --

Bit5~Bit4 -- Reserved, all must be 0.

Bit3 T2C3IF: Timer2 compare/capture channel 3 flag;
1= Timer2 compare channel 3 {CCH3:CCL3}={TH2:TL2} or capture channel 3 generates a capture operation, which needs to be cleared by software.
0= --

Bit2 T2C2IF: Timer2 compare/capture channel 2 flag;
1= Timer2 compare channel 2 {CCH2:CCL2}={TH2:TL2} or capture channel 2 generates a capture operation, which needs to be cleared by software.
0= --

Bit1 T2C1IF: Timer2 compare/capture channel 1 flag;
1= Timer2 compare channel 1 {CCH1:CCL1}={TH2:TL2} or capture channel 1 generates a capture operation, which needs to be cleared by software.
0= --

Bit0 T2C0IF: Timer2 compare/capture channel 0 flag;
1= Timer2 compares channel 0 {RLDH:RLDL}={TH2:TL2} or captures channel 0 to generate a capture operation, which needs to be cleared by software.
0= --

6.4.3.3 Peripheral Interrupt Flag 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 general interrupt indicator bit, read-only;
1= SPI generates an interrupt (this bit is automatically cleared after clearing the specific interrupt flag bit);
0= SPI does not generate an interrupt.
Bit6 I2CIF: I2C general interrupt indicator bit, read-only;
1= I2C generates an interrupt (this bit is automatically cleared after clearing the specific interrupt flag bit);
0= I2C does not generate an interrupt.
Bit5 -- Reserved, must be 0.
Bit4 ADCIF: ADC interrupt flag bit;
1= ADC conversion is completed and needs to be cleared by software;
0= ADC conversion is not completed.
Bit3 PWMIF: PWM general interrupt indicator bit, read-only;
1= PWM generates an interrupt, (this bit is automatically cleared after clearing the specific interrupt flag bit);
0= PWM does not generate an interrupt.
Bit2 -- Reserved, must be 0.
Bit1 TF4: Timer4 timer overflow interrupt flag bit;
1= Timer4 timer overflows, it is automatically cleared by hardware when entering the interrupt service routine, or can be cleared by software;
0= Timer4 timer has no overflow.
Bit0 TF3: Timer3 timer overflow interrupt flag bit;
1= Timer3 timer overflows, it is automatically cleared by hardware when entering the interrupt service routine, or it can be cleared by software;
0= Timer3 timer has no overflow.

6.4.3.4 SPI Interrupt Flag 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 complete interrupt flag bit, read only;
1= SPI transmission is complete (read SPSR first, then clear after reading/writing SPDR);
0= The SPI transmission is not complete.
Bit6 WCOL: SPI write conflict interrupt flag bit, read only;
1= Write SPDR operation conflict occurs when SPI transmission is not completed (read SPSR first, then clear after reading/writing SPDR);
0= No write conflicts.
Bit5~Bit1 -- Reserved, all must be 0.
Bit0 SSCEN: SPI master mode NSS output control bit.
1= When SPI is in idle state, NSS outputs high level;
0= NSS outputs the contents of register SSCR.

6.4.3.5 I²C Master Mode Interrupt Flag Register 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 of the entire master module, including I2CMSR);
0= The interrupt flag bit is cleared to 0 in I2C master mode.
I2CMIF: Interrupt flag bit in I2C master mode;
1= In master mode, send/receive is complete, or a transmission error occurs. (software clear, write 0 to clear);
0= No interrupt was generated.

Bit6~Bit0 Control and 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, all must be 0.
Bit2 SENDFIN: I2C slave mode send operation completed flag, read only;
1= The master device no longer needs data, TREQ is no longer set to 1, and the data transfer has been completed. (Automatically cleared after reading I2CSCR).
0= --
Bit1 TREQ: I2C slave mode ready to send flag, read only;
1= As the transmitting device has been addressed, the master device is ready to receive data. (Automatically cleared after writing to I2CSBUF).
0= --
Bit0 RREQ: I2C slave mode receive completion flag, read only;
1= Received. (automatically cleared after reading I2CSBUF);
0= Not received.

The related status bits of the I2C slave mode are also interrupt flag bits

Note: I2C master mode interrupts and slave mode interrupts share the same interrupt vector (00ABH).

6.4.3.7 UART Control Register SCONn

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SCONn	UnSM0	UnSM1	UnSM2	UnREN	UnTB8	UnRB8	TIn	RIn
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; register SCON1 address 0xEA.

BANK1: Register SCON2 address 0xE4; register SCON3 address 0xE6.

Bit7~Bit2 U1SM0、U1SM1、U1SM2、U1REN、U1TB8、U1RB8: UART1 related control bits, see UARTn function description for details.

Bit1 TIn: Send interrupt flag (requires software to clear);
1= Indicates that the send buffer is empty, and the next frame of data can be sent.
0= --

Bit0 RIn: Receive interrupt flag bit (requires software to clear);
1= Indicates that the receive buffer is full, and the next frame of data can be received after reading.
0= --

TIn and RIn occupy the same interrupt vector, and need to be inquired to determine whether it is a receiving interrupt or a sending 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 port interrupt flag bit (i=0-7);
1= P0i port generates an interrupt, which needs to be cleared by software;
0= No interrupt is generated at port P0i.

6.4.3.9 P1 Port 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 port interrupt flag bit (i=0-7);
1= P1i port generates an interrupt, which needs to be cleared by software;
0= No interrupt is generated on port P1i.

6.4.3.10 P2 Port Interrupt Flag Register P2EXTIF

0xB6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2EXTIF	P27IF	P26IF	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~Bit0 P2iIF: P2i port interrupt flag bit (i=0-7);
1= P2i port generates an interrupt and needs to be cleared by software;
0= P2i port does not generate an interrupt.

6.4.3.11 P3 Port Interrupt Flag Register P3EXTIF

0xB7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P3EXTIF	P37IF	P36IF	P35IF	P34IF	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~Bit0 P3iIF: P3i port interrupt flag bit (i=0-7);
1= P3i port generates an interrupt and needs to be cleared by software;
0= P3i port does not generate an interrupt.

6.4.3.12 P4 Port Interrupt Flag Register P4EXTIF

0xA6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P4EXTIF	P47IF	P46IF	P45IF	P44IF	P43IF	P42IF	P41IF	P40IF
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 P4iIF: P4i port interrupt flag bit (i=0-7);
1= P4i port generates interrupt and needs to be cleared by software;
0= P4i port does not generate interrupt.

6.4.3.13 P5 Port Interrupt Flag Register P5EXTIF

0xA7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P5EXTIF	--	--	P55IF	P54IF	P53IF	P52IF	P51IF	P50IF
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, all must be 0.
Bit5~Bit0 P5iIF: P5i port interrupt flag bit (i=0-5);
1= P5i port generates an interrupt, which needs to be cleared by software;
0= No interrupt is generated on the P5i port.

6.4.4 Clear Operation Of Interrupt Flag

Interrupt flag bit is divided into the following types:

- Hardware automatic clear (need to enter the interrupt service program)
- Software clear
- Read/write operation clear

1) Flag bit automatically cleared by hardware

The bits that support automatic hardware clearing are the interrupt flag bits generated by INT0, INT1, T0, T1, T3, and T4. The condition for the hardware to automatically clear the flag is: turn on the total interrupt enable bit EA=1, and turn on the corresponding interrupt enable bit. After the interrupt is generated, the system enters the corresponding interrupt service routine, and the flag bit is automatically cleared. If the interrupt enable is turned off, these flags can also be cleared by software.

2) Flag bit cleared by software

There are flag bits in the system that can only be cleared by software. These flags will not be automatically cleared after entering the interrupt service routine, and need to be cleared by software by writing 0. Otherwise, after exiting the interrupt service program, it will enter the interrupt service program again.

3) Flag bits cleared by read and write operations

There is a flag bit in the system, instead of writing 0 to clear the flag bit, you need to read/write other registers to clear the flag bit. For example, the transfer complete flag bit SPISIF in the SPI interrupt flag register, after setting it to 1, you need to read SPSR first, and then clear it after reading/writing SPDR.

The software clearing operation requires attention: when multiple interrupt flags are in the same register and the moments when 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 bit register contains the upward comparison interrupts of the PG0-PG5 channels. These interrupt flag bits are not related to each other. When PG0 generates an up compare interrupt, the value of PWMUIF is 0x01. After entering the interrupt service routine, perform a read-modify-write operation to clear the bit

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

This operation is specifically implemented by first reading the value of PWMUIF back to the CPU, and then perform calculations again, and finally send back to PWMUIF. If the interrupt flag bit PWMUIF[1] of PG1 is set to 1 after the CPU is read, and PWMUIF[1] is 0 when it is read, it will be sent back to PWMUIF[1] after the calculation is also 0, and PG1 will be cleared at this time The up compare interrupt flag that has been generated is PWMUIF[1].

To clear the above type of interrupt flag bit, it is recommended to write 0 directly, and write 1 to other irrelevant flag bits: PWMUIF = 0xFE. This operation has no actual effect on writing 1 to the irrelevant interrupt flag.

6.4.5 Special Interrupt Flag In Debug Mode

There is a flag bit in the system, instead of writing 0 to clear the flag bit, you need to read/write other registers to clear the flag bit.

In debug mode, break point execution, after single-step operation or stop operation, the emulator will read the values of all registers from the system to the emulation software. The read/write operation of the emulator is exactly the same as the read/write operation in normal mode.

Therefore, during the debugging process, after a pause, the interrupt flag bit set to 1 should appear, but it is displayed as 0 in the observation window.

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

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

void SPI_int (void) interrupt SPI_VECTOR     // SPI interrupt service routine
{
        O1  _nop_();                //Set breakpoint 1
            _nop_();
        O2  k = SPSR;               //Set breakpoint 2
          _nop_();
        ...
}
```

When the breakpoint is running, after stopping at breakpoint 1, the SPI completes the send operation, and the send completion interrupt has been generated, so SPSR.7=1, at this time the emulator has completed the operation of reading all registers (including reading SPSR).

Execute breakpoint operation again and stop at breakpoint 2. At this time, the emulator once again completes the operation of reading all registers (including SPDR), so SPSR.7=0 at this time. The above situation will also occur when single stepping twice, so you need to pay attention to it in debug mode.

7. I/O Port

7.1 GPIO Function

Chip has six groups of I/O ports: PORT0, PORT1, PORT2, PORT3, PORT4, PORT5.

PORTx is a bidirectional port. Its corresponding data direction register is PxTRIS. Set a bit of PxTRIS to 1 (=1) to configure the corresponding pin as an output. Clear a bit of PxTRIS (= 0) to configure the corresponding PORTx pin as an input.

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

When PORTx is used as an output port, reading the Px register is related to the setting of the PxDS register. A bit of PxDS is 1, the corresponding bit of Px read is the status of the pin, a bit of PxDS is cleared (=0), the corresponding bit of Px read is the state of the port data latch; When the PORTx as input port, the status of the pin is read by reading the Px register, which has nothing to do with 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 set to 0. I/O pins configured as analog inputs always read 0.

The registers related to PORTx include Px, PxTRIS, PxOD, PxUP, PxRD, PxDR, PxSR, 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; Register P4 address: 0xC0; Register P5 address: 0xD8.

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

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

0= Port pin level<VIL(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; Register P4TRIS address: 0xA4; Register P5TRIS address: 0xA5.

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

1= The pin is configured as an output;

0= The pin is configured as an input (three-state).

Note:

- After the port is set as an output port, the data read from the port is the value of the output register.
- After the port is set as an input port, the <read-modify-write> type of instruction for the port is actually an operation on the output register.

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; Register P4OD address: F049H; Register P5OD address: F059H.

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

1= The pin is configured as an open-drain state (the output is an open-drain output);

0= The pin is configured as 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; Register P4UP address: F04AH; Register P5UP address: F05AH.

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

1= Pin pull-up resistor is turn on;

0= Pin pull-up resistor is turn 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; Register P4RD address: F04BH; Register P5RD address: F05BH.

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

1= Pin pull-down resistor is turn on;

0= Pin pull-down resistor is turn off.

Note: The control of the pull-down resistor has nothing to do with the configuration of GPIO and the multiplexing function, and is controlled separately by the PxRD register.

7.1.6 PORTx Drive Current Control Register PxDR

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxDR	PxDR7	PxDR6	PxDR5	PxDR4	PxDR3	PxDR2	PxDR1	PxDR0
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 P0DR address: F00CH; Register P1DR address: F01CH; Register P2DR address: F02CH;

Register P3DR address: F03CH; Register P4DR address: F04CH; Register P5DR address: F05CH.

Bit7~Bit0 PxDR<7:0>: Drive current control bit (valid when the port is configured to output).

1= The drive is weak;

0= The drive is strong.

7.1.7 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; Register P4SR address: F04DH; Register P5SR address: F05DH.

Bit7~Bit0 PxSR<7:0>: Px slope control register (valid when the port is configured as output state);

1= Px pin is slow slope;

0= Px pin is fast slope.

7.1.8 PORTx Data Input Selection 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; Register P4DS address: F04EH; Register P5DS address: F05EH.

Bit7~Bit0 PxDS<7:0>: Data input selection bit, when configured as GPIO, it affects the reading of the value of Px register;

1= Output/input mode reads the pin status;
(when the port is set to output state The Smit circuit also remains open);

0= Output mode: read as the data latch status;
input mode: read as the pin status.

Note: If you need to read the pin status when the port is a multiplex function input structure, you need to set the port direction control to input mode.

7.2 Multiplexed Function

7.2.1 Port Multiplexed Function Table

Pins share multiple functions, and each I/O port can be flexibly configured with digital functions or designated 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), and the communication input function is also selected by the communication input function allocation register (PS_XX) designation.

Digital functional configuration shown in the following table:

	External input	Reuse function configuration number						
		0	1	2	3	4	5	6
P00	T1G，ADET	GPIO	ANA	-	-	-	PG3	-
P01	T0G	GPIO	ANA	-	-	-	PG2	-
P02	T0	GPIO	ANA	-	-	-	PG1	CC2
P03	T1	GPIO	ANA	-	-	-	PG0	C0_O
P04	T2EX，CAP0	GPIO	ANA	-	SCLK	SCL	PG5	-
P05	T2	GPIO	ANA	TXD2	MOSI	SDA	PG4	C1_O
P06	-	GPIO	ANA	RXD2	MISO	-	-	C0_O
P07	-	GPIO	ANA	-	NSS(NSSO0)	-	-	-
P10	ADET	GPIO	ANA	-	-	-	-	-
P11	-	GPIO	ANA	RXD1	MISO	-	-	-
P12	-	GPIO	ANA	-	SCLK	SCL	-	CLO
P13	-	GPIO	ANA	TXD1	MOSI	SDA	-	-
P14	-	GPIO	ANA	RXD1	MISO	SCL	-	-
P15	-	GPIO	ANA	-	NSS(NSSO1)	-	-	-
P16	-	GPIO	ANA	-	-	-	-	CC0
P17	T1G	GPIO	ANA	-	-	-	-	CC1
P20	T1，ADET	GPIO	ANA	RXD0	-	-	-	BUZZ
P21	-	GPIO	ANA	TXD0	-	-	PG1	-
P22	-	GPIO	ANA	RXD1	MOSI	-	PG0	-
P23	-	GPIO	ANA	TXD1	MISO	-	PG3	-
P24		GPIO	ANA	-	SCLK	-	PG2	-
P25	-	GPIO	ANA	-	NSS(NSSO2)	-	FB0	-
P26	-	GPIO	ANA	-	-	-	PG5	CLO
P27	CAP1	GPIO	ANA	-	-	-	PG4	-
P30	CAP2，ADET	GPIO	ANA	-	-	-	PG0	-
P31	-	GPIO	ANA	-	NSS(NSSO3)		PG1	-
P32	-	GPIO	ANA	-	-	-	PG2	-
P33	-	GPIO	ANA	-	-	-	PG3	-
P34	-	GPIO	ANA	-	-	-	PG4	-
P35	-	GPIO	ANA	-	-	-	PG5	-
P36	-	GPIO	ANA	-	-	-	-	-
P37	-	GPIO	ANA	-	-	-	-	C0_O
P40	ADET	GPIO	ANA	-	-	-	PG0	-
P41	-	GPIO	ANA	-	-	-	PG1	-
P42	-	GPIO	ANA	-	-	-	PG2	-

P43	-	GPIO	ANA	-	SCLK	SCL	PG3	CLO
P44	-	GPIO	ANA	TXD3	MOSI	SDA	PG4	-
P45	-	GPIO	ANA	RXD3	MISO	SCL	PG5	CC0
P46	-	GPIO	ANA	-	-	-	FB1	C0_O
P47	CAP3	GPIO	ANA	-	NSS(NSSO3)	-	-	-
P50	ADET	GPIO	ANA	-	-	-	PG0	C1_O
P51	-	GPIO	ANA	-	-	-	PG1	-
P52	T0G	GPIO	-	-	-	-	PG4	-
P53	T0	GPIO	ANA	-	-	-	PG5	BUZZ
P54	T1	GPIO	ANA	-	-	-	-	-
P55	T1G	GPIO	ANA	-	-	-	-	CC3

LED port, analog port, the CONFIG configuration port as shown:

	GPIO(0)		ANA(1)				CONFIG
	LEDSEG	LEDCOM	ADC	LCDSEG	LCDCOM	ACMP	
P00	LED_S16	-	-	LCD_S16	-	-	-
P01	LED_S17	-	-	LCD_S17	-	-	-
P02	LED_S18	-	-	LCD_S18	-	-	DSCK2
P03	LED_S19	-	-	LCD_S19	-	-	DSDA2
P04	LED_S20	-	-	LCD_S20	-	C0P0	-
P05	LED_S21	-	-	LCD_S21	-	C0N0	-
P06	LED_S22	-	-	LCD_S22	-	C1P0	OSCIN1
P07	LED_S23	-	-	LCD_S23	-	C1N	OSCOUT1
P10	LED_S0	-	-	LCD_S0	-	-	-
P11	LED_S1	-	-	LCD_S1	-	-	DSDA1
P12	LED_S2	-	AN17	LCD_S2	-	-	-
P13	LED_S3	-	AN18	LCD_S3	-	-	DSCK1
P14	LED_S4	-	AN0	LCD_S4	-	-	-
P15	LED_S5	-	AN1	LCD_S5	-	-	-
P16	LED_S6	-	AN2	LCD_S6	-	-	-
P17	LED_S7	-	AN3	LCD_S7	-	-	-
P20	LED_S8	-	AN4	LCD_S8	-	-	-
P21	LED_S9	-	AN5	LCD_S9	-	-	-
P22	LED_S10	-	AN6	LCD_S10	-	-	-
P23	LED_S11	-	AN7	LCD_S11	-	-	-
P24	LED_S12	-	AN19	LCD_S12	-	-	-
P25	LED_S13	-	AN20	LCD_S13	-	-	-
P26	LED_S14	-	AN21	LCD_S14	-	-	-
P27	LED_S15	-	AN22	LCD_S15	-	-	-
P30	LED_S27	LED_C0	-	LCD_S39	LCD_C0	-	-
P31	LED_S26	LED_C1	-	LCD_S38	LCD_C1	-	-
P32	LED_S25	LED_C2	-	LCD_S37	LCD_C2	-	-
P33	LED_S24	LED_C3	-	LCD_S36	LCD_C3	-	-
P34	-	LED_C4	AN8	LCD_S35	LCD_C4	-	-

P35	-	LED_C5	AN9	LCD_S34	LCD_C5	-	-
P36	-	LED_C6	AN10	LCD_S33	LCD_C6	-	-
P37	-	LED_C7	AN11	LCD_S32	LCD_C7	-	-
P40	-	-	AN12	-	-	C0P4	-
P41	-	-	AN13	-	-	C0P5	-
P42	-	-	AN14	-	-	C0P6	-
P43	-	-	AN15	-	-	C0P7	-
P44	-	-	AN16	-	-	C0N1	-
P45	-	-	-	LCD_S31	-	-	-
P46	-	-	-	LCD_S30	-	-	NRST
P47	-	-	-	LCD_S29	-	-	-
P50	-	-	-	LCD_S24	-	-	OSCIN2
P51	-	-	-	LCD_S25	-	-	OSCOUT2
P52	-	-	-	-	-	-	NRST
P53	-	-	-	LCD_S26	-	-	-
P54	-	-	-	LCD_S27	-	-	-
P55	-	-	-	LCD_S28	-	-	-

7.2.2 Port Multiplexing Function Configuration Register

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, all must be 0.

Bit2~Bit0 PxnCFG<2:0>: Function configuration bits, the default is GPIO function. See the port function configuration description for details;

000= GPIO function;

001= Analog function (ANA);

010= UART function;

011= SPI function;

100= IIC function;

101= PWM function;

110= Others;

111= Access is prohibited.

PxnCFG x=0-5,n=0-7

Each port has a function configuration register PxnCFG, and through each port can be set to the corresponding digital function PxnCFG. For example: To set P20 as the BEEP buzzer function, the configuration is:P20CFG = 0x06;

when the port is used as a multiplex function, there is no need to configure the port direction register PxTRIS.

- SCL, SDA pull-up resistor register can be configured, forced to open open-drain output.
- The RXD0 and RXD1 registers can configure the pull-up resistor or force the pull-up resistor to be turned on in the synchronous mode.

Other multiplexing functions are forced to turn off the pull-up resistor and open-drain output by hardware, that is, the pull-up resistor PxUP or the open-drain output PxOD is invalid by software. When the port is multiplexed with 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 Port Input Function Allocation Register

There are digital functions with only input status inside the chip, such as INT0/INT1... etc. This type of digital input function has nothing to do with the port multiplexing status. 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 allocation register is as follows:

Register	Address	Function	Function 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 allocation register
PS_T0G	F0C3H	T0G	Timer0 gate control input port allocation register
PS_T1	F0C4H	T1	Timer1 external clock input port allocation register
PS_T1G	F0C5H	T1G	Timer1 gate control input port allocation register
PS_T2	F0C6H	T2	Timer2 external event or gate control input port allocation register
PS_T2EX	F0C7H	T2EX	Timer2 falling edge auto reload input port allocation register
PS_CAP0	F0C8H	CAP0	Timer2 input capture channel 0 port allocation register
PS_CAP1	F0C9H	CAP1	Timer2 input capture channel 1 port allocation register
PS_CAP2	F0CAH	CAP2	Timer2 input capture channel 2 port allocation register
PS_CAP3	F0CBH	CAP3	Timer2 input capture channel 3 port allocation register
PS_ADET	F0CCH	ADET	ADC external trigger input Port allocation register

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

	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 chip, the value of the unused reserved, prohibited);

0x00= Assigned to port P00;
0x01= Assigned to port P01;
…… ……
0x20= Assigned to port P20;
0x21= Assigned to port P21;
…… ……
0x40= Assigned to port P40;
0x41= Assigned to port P41;
…… ……
0xFF= Not assigned to port;

1) Each port can be configured For INT0, INT1 interrupt function (other input function port assignment is fixed). If multiple ports are configured for the same digital function at the same time, the priority will decrease in the order of P00, P01,..., P54, P55. If P03 and P20 are configured as T1 functions at the same time, the P03 configuration is valid and the P20 configuration is invalid.

2) The input function assignment structure supports multiple input functions assigned to the same port. For example, INT0 and T0 can be allocated to port P02 at the same time, the configuration is as follows:

```
P02CFG = 0x00;      //P00 port is configured as GPIO function
P0TRIS = 0x00;      //P02 is used as GPIO input function
PS_INT0 = 0x02;     //P02 port is configured as INT0 function
PS_T0 = 0x02;       // P02 port is configured as T0 function
```

3) The input function allocation structure is relatively independent and can support simultaneous use with other multiplexed function ports. At this time, there is no need to configure the direction register of the corresponding port. For example, RXD0 and INT0 can be allocated to Port P20 is configured as follows:

```
P20CFG = 0x02;      //P20 is multiplexed as RXD0 function of UART0
PS_INT0 = 0x20;     //P20 port is configured as INT0 function
```

4) The input function configuration structure can also be used simultaneously with the port external interrupt function. If you can assign T1G and GPIO interrupt functions to port P00 at the same time, the configuration is as follows:

```
P00CFG = 0x00;      //P00 port is configured as GPIO function
P0TRIS = 0x00;    //P00 is used as GPIO input function
PS_T1G = 0x00;      //P00 port is configured as T1G function
P00EICFG = 0x01;    //P00 port is configured as rising edge trigger interrupt
P0EXTIE = 0x01;     //Enable P00 port external interrupt
```

7.2.4 Communication Input Function Allocation Register

Port as the communication port (UART1/SPI/IIC), there are multiple input ports to choose from, and different port inputs can be selected by setting the following registers. The communication input function port allocation register is as follows:

Register	Address	Function	Function description
PS_SCLK	F698H	SCLK	SPI clock port allocation register
PS_MOSI	F699H	MOSI	SPI slave input port allocation register
PS_MISO	F69AH	MISO	SPI master input port allocation register
PS_NSS	F69BH	NSS	SPI slave chip select input port allocation register
PS_SCL	F69CH	SCL	IIC clock input port allocation register
PS_SDA	F69DH	SDA	IIC data input port allocation register
PS_RXD1	F69EH	RXD1	UART1 data input port allocation register

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

	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	W	W	W	W	W	W	W	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 (according to the actual port of the chip, see the multiplexing function allocation table);

0x00= Assigned to port P00;

0x01= Assigned to port P01;

…… ……

0x20= Assigned to port P20;

0x21= Assigned to port P21;

…… ……

0x40= Assigned to port P40;

0x41= Assigned to port P41;

…… ……

0xFF= Not assigned to port;

Note: This series of registers are write-only registers. Setting these registers requires direct assignment. And, or operations are not supported, and reading is invalid.

If P11 is configured as RXD1, the configuration is as follows:

```
PS_RXD1 =0X11;      //Select P11 as the RXD1 pin
P11CFG =0X02;       // P11 is multiplexed as UART1's RXD1 function
```

7.2.5 Port External Interrupt Control Register

When using external interrupts, the port needs to be configured as a GPIO function and the direction is set as an input port. Or the multiplexing function is an input port (such as RXD0, RXD1), and each port can be configured as a GPIO interrupt function.

PORTx external interrupt control register PxNEICFG

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxNEICFG	--	--	--	--	--	--	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 PxNEICFG<1:0>: PxN external interrupt control bit;

00= External interrupt disabled;

01= Rising edge trigger interrupt;

10= Falling edge trigger interrupt;

11= Rising edge or falling edge trigger interrupt.

There are 8 external interrupt control registers of Px, including Px0EICFG~Px7EICFG, which control the external interrupts of Px0~Px7 respectively.

If configure P00 to trigger the interrupt on the falling edge, the configuration is as follows:

```
P00CFG =0x00;       //Configure P00 as a GPIO function
P0TRIS =0x00;       //Configure P00 as an input
P00EICFG = 0x02;    //Configure P00 as a falling edge trigger interrupt
EA = 1;             / /Global interrupt enable
P0EXTIE = 0x01;     //Allow the P00 external interrupt function
```

7.2.6 Multiplex Function Application Note

1) The input of the multiplexing function is relatively independent of the external interrupt (GPIO interrupt) of the port and the structure of the port input function.
   For example, configure the P20 port as RXD0, and configure the P20 GPIO interrupt trigger mode as rising edge trigger and interrupt enable. When the P20 input changes from low to high, it will trigger the P20 GPIO interrupt.
2) The input structure of the digital signal is not affected by the system configuration status.
   For example, if the P52 port is powered on and configured as an external reset port, the input module of this port will be opened. If P52 is configured as INT0 in the program and the interrupt enable is turned on, the interrupt service routine will be executed before the reset signal sampling time is valid, and then the reset operation will be generated.
3) It should be noted that in the debug mode, if the multiplexing function is configured on the DSDA port, the input function is also valid. It is recommended that the related multiplexing function is not configured to the DSDA port in the debugging mode.
4) When the port is used as an analog function, when the function configuration register is set to 0x01, the hardware closes the digital circuit to reduce power consumption, and the GPIO function-related register setting is invalid.
5) The port input/communication input function has priority restrictions. If there are two or more ports configured with the same input function at the same time, configure the selection according to the priority order of P00, P01,..., P54, P55 from high to low.
6) The output function of the communication port has no priority limit. If there are multiple ports configured with the same output function, the function will be output at these ports at the same time.
7) RXD1 of UART1 needs to be selected by the port allocation register when it is used as an input function, and has nothing to do with the port allocation register when it is used as a synchronous output function. That is, when RXD1 is used as a synchronous output function, multiple pins can be selected as RXD1 output at the same time.
8) When the SCLK of SPI is used as the clock input of the slave, it needs to be selected by the port distribution register. When it is used as the clock output of the master, it has nothing to do with the port distribution register. It is recommended to configure the port distribution register when SCLK is used as an output or input.
9) The SCL of IIC needs to be selected by the port allocation register when it is used as the clock input of the slave. When it is 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 when it is used as an output or input.

8. Watch Dog Timer(WDT)

8.1 Overview

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

When the watchdog timer counts to the set overflow value, a watchdog overflow interrupt flag bit (WDTIF=1) is generated. If the global interrupt is enabled (EA=1) and the watchdog timer interrupt is enabled (EIE2[5]=1), the CPU will execute the interrupt service routine and clear the watchdog counter by writing the register WDCON[0]=1 . After the watchdog counter is cleared, the counter starts counting from 0 again until the next timer overflows.

When the watchdog timer overflows, if the watchdog overflow reset is enabled (WDCON[1]=1) and the watchdog counter is not cleared, a watchdog overflow reset will be generated. The watchdog overflow reset is a protection setting of the system. When the system runs to an unknown state, the watchdog can be used to reset the system, so as to prevent the system from entering an infinite loop. The watchdog time-out reset is detailed in the reset chapter.

8.2 Related Registers

8.2.1 Watchdog Control Register WDCON

0x97	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WDCON	SWRST	PORF	EXTIF	FIXIF	WDTIF	WDTRF	WDTRE	WDTCLR
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 SWRST: Software reset control bit;
1: Execute system software reset (write 0 to clear after reset).
0: --

Bit6 PORF: Power-on reset flag bit;
1: System is power-on reset (write 0 to clear, no TA write sequence is required).
0: --

Bit5 EXTIF: External reset flag bit;
1= System is external reset (write 0 to clear, no TA write sequence is required).
0= --

Bit4 FIXIF: CONFIG status protection bit reset flag bit;
1= System is reset for the CONFIG state protection bit (write 0 to clear, no TA write sequence is required).
0= --

Bit3 WDTIF: WDT overflow interrupt flag bit;
1= WDT overflow (write 0 to clear);
0= WDT does not overflow.

Bit2 WDTRF: WDT reset flag bit;
1= System is reset by WDT (write 0 to clear);
0= System is not reset by WDT.

Bit1 WDTRE: WDT reset enable bit;
1= Enable WDT to reset CPU;
0= Disable WDT to reset CPU.

Bit0 WDTCLR: WDT counter clear bit;
1= Clear WDT counter (hardware automatic clearing);
0= Disable WDT counter (writing 0 is invalid).

Note:
1. If the WDT configuration in CONFIG is: ENABLE, the WDT is always enabled, regardless of the state of the WDTRE control bit. And the overflow reset function of WDT is forcibly turned on.
2. If the WDT configuration in CONFIG is: SOFTWARE CONTROL, you can use the WDTRE control bit to enable or disable WDT.

Modify the instruction sequence 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	--	--	T0CNTM
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 bit;
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 clock source selection bit;
0= Fsys/12;
1= Fsys/4.

Bit3 T0M: Timer0 clock source selection bit;
0= Fsys/12;
1= Fsys/4.

Bit2~Bit1 -- Reserved, both must be 1.

Bit0 T0CNTM: Timer0 count source selection bit;
0= PWM0 output;
1= T0 pin input.

8.3 WDT Interrupt

Watchdog timer can be enabled or disabled through the EIE2 register, and the high/low priority is set through the EIP2 register. The interrupt related 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 interrupt;
0= Disable SPI interrupt.
Bit6 I2CIE: I^2C interrupt enable bit;
1= Enable I^2C interrupt;
0= Disable I^2C interrupt.
Bit5 WDTIE: WDT interrupt enable bit;
1= Enable WDT overflow interrupt;
0= Disable WDT overflow interrupt.
Bit4 ADCIE: ADC interrupt enable bit;
1= Enable ADC interrupt;
0= Disable ADC interrupt.
Bit3 PWMIE: PWM total 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 interrupt;
0= Disable Timer4 interrupt.
Bit0 ET3: Timer3 interrupt enable bit;
1= Enable Timer3 interrupt;
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 as high-level interrupt;
0= Set as low-level interrupt.

Bit6 PI2C: I^2C interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.

Bit5 PWDT: WDT interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.

Bit4 PADC: ADC interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.

Bit3 PPWM: PWM interrupt priority control bit
1= Set as high-level interrupt;
0= Set as low-level interrupt.

Bit2 --: Reserved, must be 0.

Bit1 PT4: TIMER4 interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.

Bit0 PT3: TIMER3 interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.

9. Window Watchdog Timer(WWDT)

9.1 Overview

The window watchdog timer is a 5-bit countdown timer with optional window comparison time, clock source provided by LSI, and optional frequency division. The timer can generate interrupts, wake up the system from sleep mode, and reset the chip.

9.2 Related Registers

9.2.1 WWDT Control Register WWCON0

0xE5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WWCON0	WWDTPSC3	WWDTPSC2	WWDTPSC1	WWDTPSC0	WWDTEN	WWDTRE	WWDTCLR	WWDTRF
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

This register is protected by TA.

Bit7~4 WWDTPSC<3:0>: Window watchdog prescaler bit;

0000= $F_{LSI}/2^{8}$ 1000= $F_{LSI}/2^{16}$

0001= $F_{LSI}/2^{9}$ 1001= $F_{LSI}/2^{17}$

0010= $F_{LSI}/2^{10}$ 1010= $F_{LSI}/2^{18}$

0011= $F_{LSI}/2^{11}$ 1011= $F_{LSI}/2^{19}$

0100= $F_{LSI}/2^{12}$ 1100= $F_{LSI}/2^{20}$

0101= $F_{LSI}/2^{13}$ 1101= $F_{LSI}/2^{21}$

0110= $F_{LSI}/2^{14}$ 1110= $F_{LSI}/2^{22}$

0111= $F_{LSI}/2^{15}$ 1111= $F_{LSI}/2^{22}$

Bit3 WWDTEN: Window watchdog module enable bit;

1= Enable;

0= Disable.

Bit2 WWDTRE: Window watchdog reset enable bit;

1= Enable;

0= Disable.

Bit1 WWDTCLR: Window watchdog clear bit;

1= Clear the timer (write 1 to clear the timer, and automatic clear by the hardware);

0= Writing 0 is invalid.

Bit0 WWDTRF: Window watchdog reset flag bit;

1= Generate reset;

0= Writing 0 to clear reset flag.

9.2.2 WWDT Control Register WWCON1

0xE7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WWCON1	FORCE3	FORCE2	FORCE1	FORCE0	MODE	WWDTSLE	WWDTIE	WWDTIF
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

This register is protected by TA.

Bit7~4 FORCE<3:0>: Window watchdog overflow forced reset enable bit;

A/F= Force window watchdog reset enable;

Others= The watchdog reset enable is determined by WWDTEN and WWDTRE together.

Bit3 MODE: Window watchdog mode selection bit;

1= Window feeding dog mode (0<counter value<CMPDAT value interval feeding the dog will not generate reset, and reset in other cases);

0= Feed dog mode at any time (counter decrements from 1 to 0 to generate reset).

Bit2 WWDTSLE: Window watchdog sleep wake-up enable bit

1= Enable;

0= Disable.

Bit1 WWDTIE: Window watchdog compare interrupt enable bit;

1= Enable;

0= Disable.

Bit0 WWDTIF: Window watchdog compare overflow flag;

1= Compare overflow (enable WWDTIE to generate an interrupt);

0= Writing 0 to clear reset flag.

9.2.3 WWDT Compare Value Register WWCMPD

0xE6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WWCMPD	--	--	--	CMPDAT4	CMPDAT3	CMPDAT2	CMPDAT1	CMPDAT0
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

This register is protected by TA.

Bit7~5 -- Reserved, all must be 0.

Bit4~0 CMPDAT<4:0>: Window Compare value

9.3 WWDT Interrupt And Sleep Wake-up

Window watchdog timer can be enabled or disabled through the WWCON1 register, as described above. The high/low priority is set by the EIP3 register, and its interrupt related bits are as follows.

9.3.1 Interrupt Mask Register EIP3

0xBB	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP3	--	--	PWWDT	--	--	--	--	--
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, all must be 0.
Bit5	PWWDT	WWDT interrupt priority control bit;
	1=	Set as high-level interrupt;
	0=	Set as low-level interrupt.
Bit4~Bit0	--	Reserved, all must be 0.

In any one of the dog feeding modes, when the window watchdog timer counts to the window comparison value, the hardware will set the comparison overflow flag bit WWCON1[0] to 1. If the global interrupt is enabled (EA=1) and the window watchdog compare interrupt is enabled (WWCON1[1]=1), the CPU will execute the interrupt service routine. The calculation formula of the window comparison time is as follows:

Window comparison time $=\frac{PSC}{125}$ ×(0x1F-WWCMPD[4:0]) ms

In the window comparison time formula, PSC is the frequency division coefficient of the window watchdog, which is set by WWCON0[7:4].

Using the window watchdog comparison to wake up the sleep mode, it is necessary to turn on the window watchdog module enable bit WWDTEN and the sleep wake-up enable bit WWDTSLE before sleeping, and set the comparison value "WWCMPD[4:0]". If the global interrupt enable and watchdog compare interrupt enable are turned on before sleep, and after wake-up from sleep, the interrupt service routine will be executed first, and then the next instruction of the sleep instruction will be executed after the interrupt returns.

9.4 Function Description

The window watchdog has two feeding dog modes : window feeding dog mode and feeding dog mode at any time. When WWCON1[3] is set to 1, it is the window feeding dog mode, and when it is 0, it is the feeding dog mode at any time.

- Window feeding dog mode

The window watchdog timer starts to count down from 0x1F. When the timer counts to the set window comparison value WWCMPD [4 : 0], the comparison overflow flag bit WWCON1 is 1. At this time, the clearing timer operation can be performed (WWCON0 sets 1 to clear the current value of the window watchdog timer) to make the timer count again from 0x1F. If the window watchdog timer is cleared before the count value reaches the window comparison value or the timer keeps counting to 0, the window watchdog reset will be generated under the condition that the window watchdog reset is enabled (WWCON0[2]=1), and the hardware sets the window watchdog reset flag bit WWCON0[0] to 1. That is, the window feeding dog mode needs to clear the window watchdog timer during the window period (0 < count < WWCMPD).

- Feeding dog mode at any time

The window watchdog timer starts counting down from 0x1F. Before the timer counts to 0, the timer clearing operation can be performed to make the timer count from 0x1F again. If the timer counts to 0, the window watchdog reset will be generated under the window watchdog reset enable condition, and the hardware will set the window watchdog reset flag to 1. That is, the feeding dog mode at any time can clear the window watchdog timer at any time (0 < count value).

When the window watchdog timer overflow forced reset function is enabled (WWCON1[7:4]=0xa/0xf), regardless of whether the user configures the window watchdog mode enable and reset enable, the window watchdog timer will be started. A system reset occurs when the timer counts down from 0x1F to 0. The window watchdog overflow forced reset function is also effective in sleep mode.

10. Timer 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 working modes, and Timer 0 has four working modes. They provide basic timing and event counting operations.

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

In the "counter mode", the timer 0 timing register will increase whenever it detects a falling edge on the corresponding input pin (T0 or PWM0); the timer 1 timing register will increase whenever it detects a falling edge on the corresponding input pin (T1).

10.1 Overview

Timer 0 and Timer 1 are fully compatible with standard 8051 timers.

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

Mode	M1	M0	Function 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] form 8-bit automatic reload timer/counter, reload from THx
3	1	1	TL0, TH0 are two 8-bit timer/counters, Timer1 stop counting

Register THx and TLx are special function registers, it has the function of storing the actual timer value. THx and TLx can be cascaded into 13-bit or 16-bit registers through mode options. Each time an internal clock pulse is received or a state transition occurs on the external timer pin, the value of the register is increased by 1. The timer will start counting from the value loaded in the preset register until the timer overflows, at which time an internal interrupt signal will be generated. If the automatic reload mode of the timer is selected, the timer value will be reset to the initial value of the preload register and continue counting, otherwise the timer value will be reset to zero. Note that in order to get the maximum calculation range of the timer/counter, the preset register must be cleared first.

10.2 Related registers

10.2.1 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/counting selection bit;
1= Counting;
0= Timing.

Bit5~Bit4 T1M<1:0>: Timer 1 mode selection 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 counting.

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

Bit2 CT0: Timer 0 timing/counting selection bit;
1= Counting;
0= Timing.

Bit1~ Bit0 T0M<1:0>: Timer 0 mode selection bits;
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.

10.2.2 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= Timer1 counter overflows, and the hardware is automatically cleared when entering the interrupt service routine;
0= Timer1 counter does not overflow.

Bit6 TR1: Timer1 running control bit;
1= Timer1 is started;
0= Timer1 is closed.

Bit5 TF0: Timer0 counter overflow interrupt flag bit;
1= Timer0 counter overflows, and the hardware is automatically cleared when entering the interrupt service routine;
0= Timer0 counter does not overflow.

Bit4 TR0: Timer0 running control bit;
1= Timer0 is started;
0= Timer0 is closed.

Bit3 IE1: External interrupt 1 flag bit;
1= External interrupt 1 generates an interrupt, and the hardware is automatically cleared when entering the interrupt service routine;
0= External interrupt 1 does not generate an interrupt.

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

Bit1 IE0: External interrupt 0 flag bit;
1= External interrupt 0 generates an interrupt, and the hardware is automatically cleared when entering the interrupt service routine;
0= External interrupt 0 does not generate an interrupt.

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

10.2.3 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 bit data register (also used as counter low bit).

10.2.4 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 used as counter high bit).

10.2.5 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 used as counter low bit).

10.2.6 Timer1 Data Register High Bit 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 used as counter high bit).

10.2.7 Function Clock Control Register CKCON

0x8E	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CKCON	WTS2	WTS1	WTS0	T1M	T0M	--	--	T0CNTM
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 bit;
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 clock source selection bit;
0= Fsys/12;
1= Fsys/4.

Bit3 T0M: Timer0 clock source selection bit;
0= Fsys/12;
1= Fsys/4.

Bit2~Bit1 -- Reserved, both must be 1.

Bit0 T0CNTM: Timer0 count source selection bit;
0= PWM0 output;
1= T0 pin input.

10.3 Timer0/1 Interrupt

Timer0/1 can be enabled or disabled through the IE register, and the high/low priority can also be set by the IP register. The interrupt related bits are as follows:

10.3.1 Interrupt Mask Register IE

0xA8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IE	EA	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 EA: Global interrupt enable Bit;
1= Allow all interrupts that are not masked;
0= Disable all interrupts.

Bit6 ES1: UART1 interrupt enable bit;
1= Enable UART1 interrupt;
0= Disable UART1 interrupt.

Bit5 ET2: TIMER2 total interrupt enable bit;
1= Enable all TIMER2 interrupts;
0= Disable all TIMER2 interrupts.

Bit4 ES0: UART0 interrupt enable bit;
1= Enable UART0 interrupt;
0= Disable UART0 interrupt.

Bit3 ET1: TIMER1 interrupt enable bit;
1= Enable TIMER1 interrupt;
0= Disable TIMER1 interrupt.

Bit2 EX1: External interrupt 1 interrupt enable bit;
1= Enable external interrupt 1 interrupt;
0= Disable external interrupt 1 interrupt.

Bit1 ET0: TIMER0 interrupt enable bit;
1= Enable TIMER0 interrupt;
0= Disable TIMER0 interrupt.

Bit0 EX0: External interrupt 0 interrupt enable bit;
1= Enable external interrupt 0 interrupt;
0= Disable external interrupt 0 interrupt.

10.3.2 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 PS1: UART1 interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.
Bit5 PT2: TIMER2 interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.
Bit4 PS0: UART0 interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.
Bit3 PT1: TIMER1 interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.
Bit2 PX1: External interrupt 1 interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.
Bit1 PT0: TIMER0 interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.
Bit0 PX0: External interrupt 0 interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.

10.3.3 Timer0/1、INT0/1 interrupt flag 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 overflows, and the hardware is automatically cleared when entering the interrupt service routine;
0= Timer1 counter does not overflow.

Bit6 TR1: Timer1 running control bit;
1= Timer1 is started;
0= Timer1 is closed.

Bit5 TF0: Timer0 counter overflow interrupt flag bit;
1= Timer0 counter overflows, and the hardware is automatically cleared when entering the interrupt service routine;
0= Timer0 counter does not overflow.

Bit4 TR0: Timer0 running control bit;
1= Timer0 is started;
0= Timer0 is closed.

Bit3 IE1: External interrupt 1 flag bit;
1= External interrupt 1 generates an interrupt, and the hardware is automatically cleared when entering the interrupt service routine;
0= External interrupt 1 does not generate an interrupt.

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

Bit1 IE0: External interrupt 0 flag bit;
1= External interrupt 0 generates an interrupt, and the hardware is automatically cleared when entering the interrupt service routine;
0= External interrupt 0 does not generate an interrupt.

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

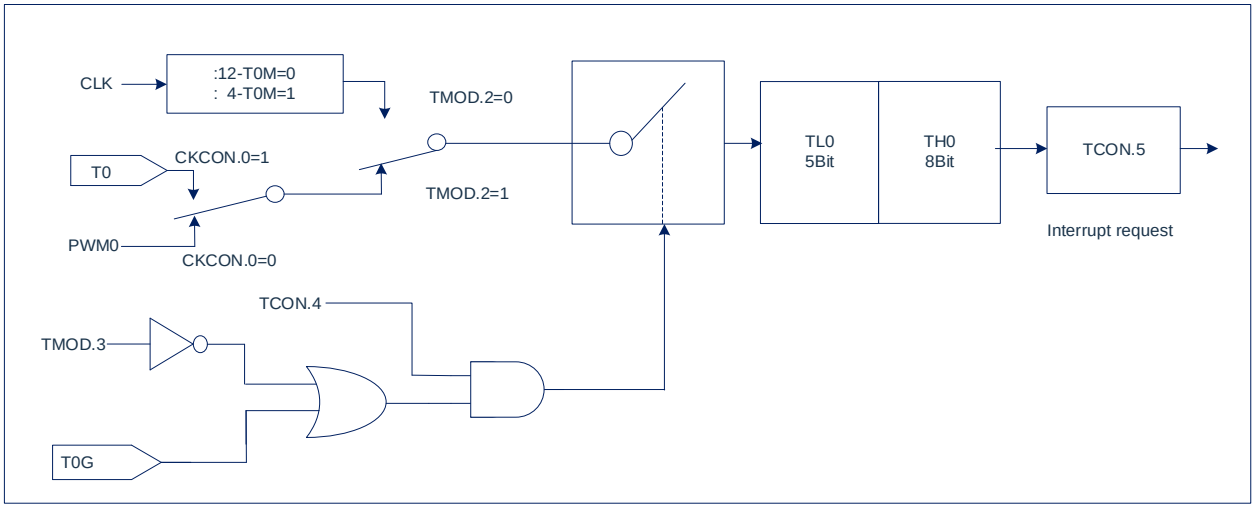
The flag bit that generates an interrupt can be cleared by software, and the result is the same as cleared by hardware. In other words, you can generate interrupts by software (it is not recommended to generate interrupts by writing flag bits) or cancel pending interrupts.

The TF0 and TF1 flags can be cleared by writing 0 when the interrupt is not enabled.

10.4 Timer0 Working Mode

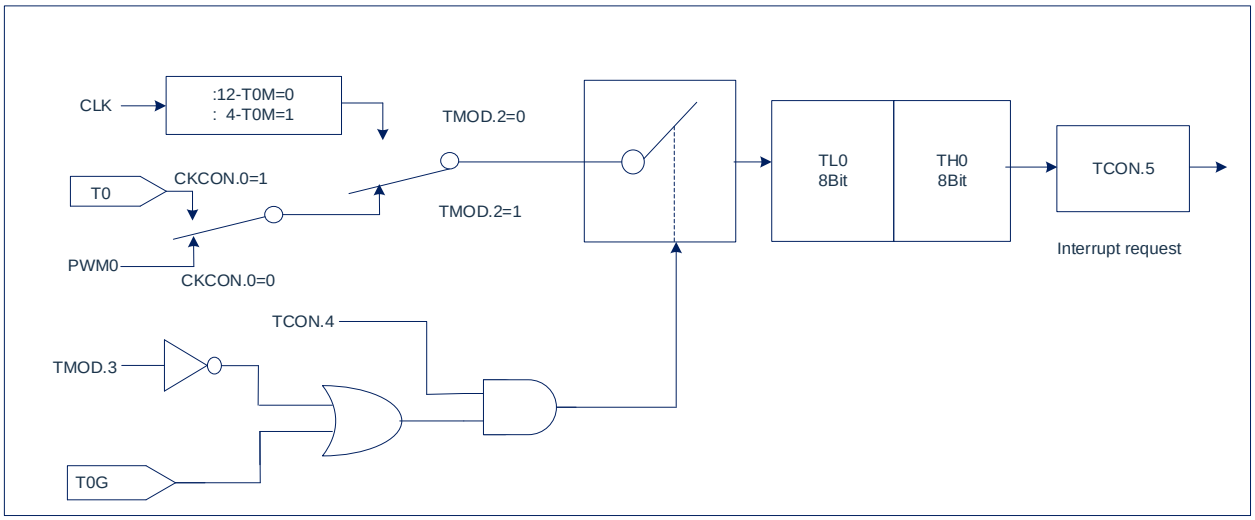
10.4.1 T0 - Mode 0 (13-bit Timer/Counter Mode)

In this mode, Timer0 is 13-bit register. When all the bits of the counter are turned 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 counting input is enabled to timer 0. (Set TMOD.3=1 to allow timer 0 to be controlled by external pin T0G for pulse width measurement). The 13-bit register consists of the low 5 bits of TH0 and TL0. The high 3 bits of TL0 should be ignored. The structure diagram of Timer0 mode 0 is shown in the following figure:



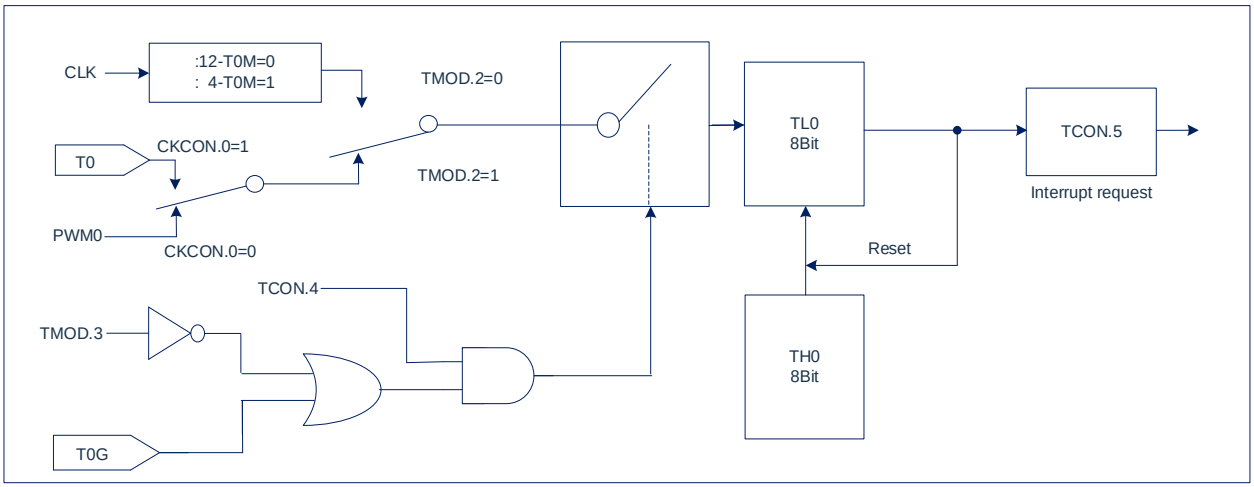
10.4.2 T0 - Mode 1 (16-bit Timer/Counter Mode)

Mode 1 is the same as mode 0, except that all 16 bits of the timer 0 data register run in mode 1. Timer0 mode 1 block diagram as shown below:



10.4.3 T0 - Mode 2 (8-bit Auto Reload Timer/Counter Mode)

The timer register in mode 2 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 to TL0 by software. The value of TH0 remains unchanged during the reload process. The structure diagram of Timer0 Mode 2 is shown in the figure below:



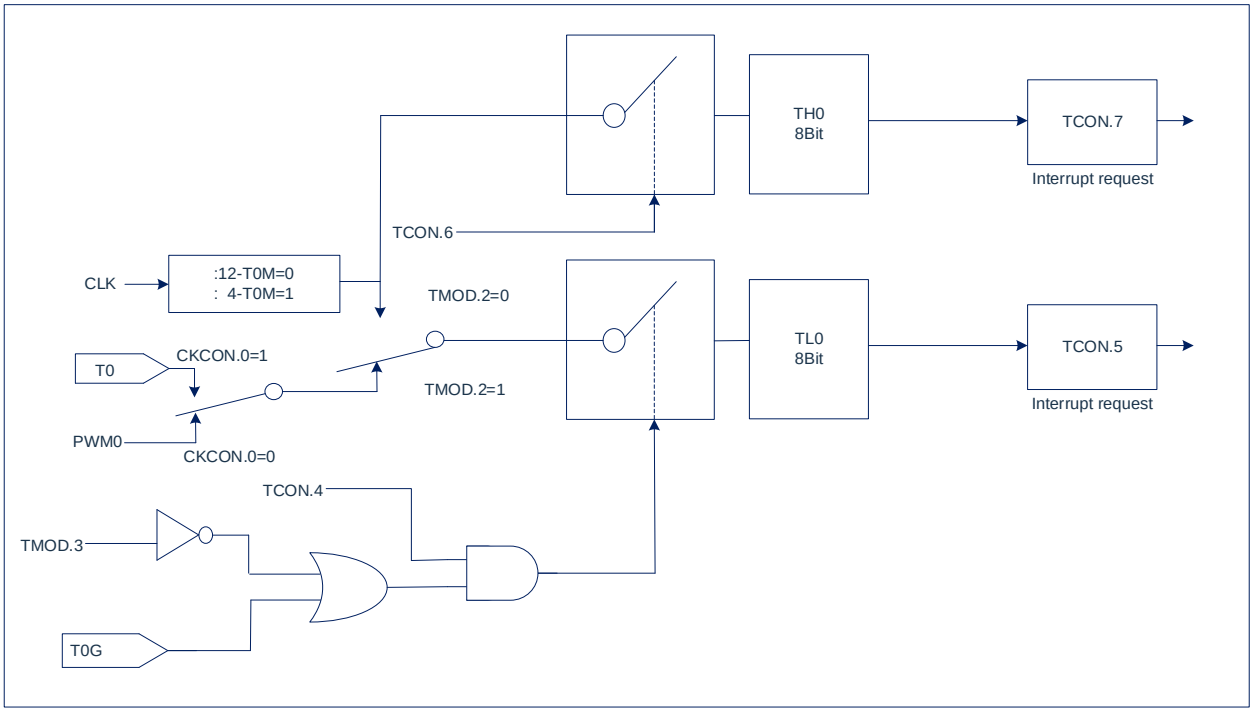
10.4.4 T0 - Mode 3 (Two Separate 8-bit Timers/Counters Mode)

Timer 0 in Mode 3 sets TL0 and TH0 as two independent counters. The logic of Timer 0 Mode 3 is shown in the figure below.

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

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

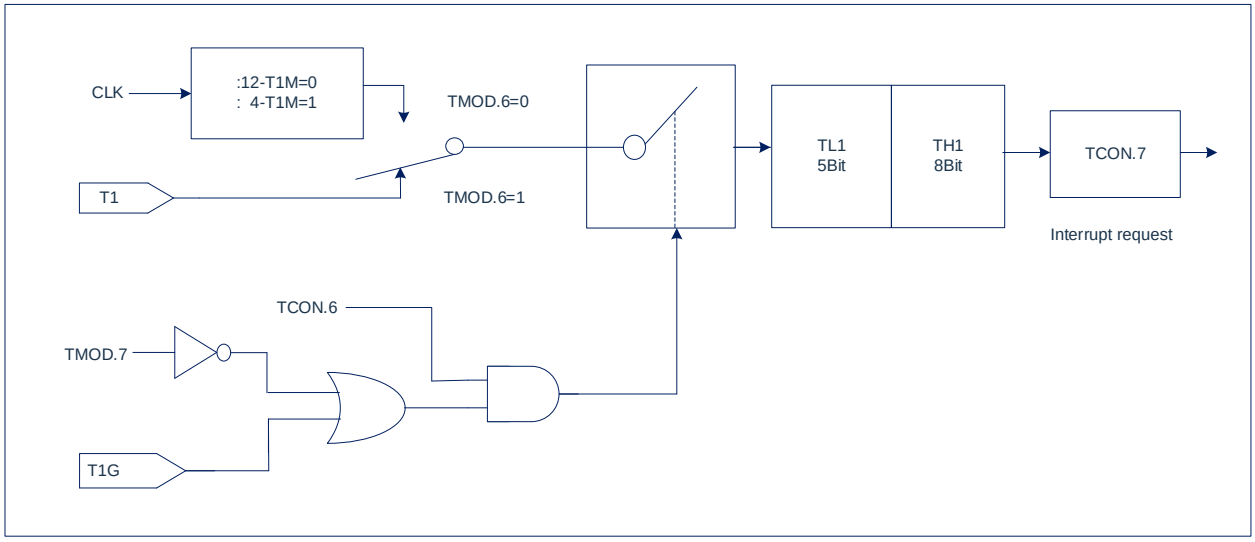
Mode 3 can be used when two 8-bit timers/counters are required. When timer 0 is in mode 3, timer 1 can be turned off by switching to its own mode 3, or it can still be used as a baud rate generator by the serial channel, or in any case that does not require timer 1 interrupts. In application. The structure diagram of Timer0 Mode 3 is shown in the figure below:



10.5 Timer1 Working Mode

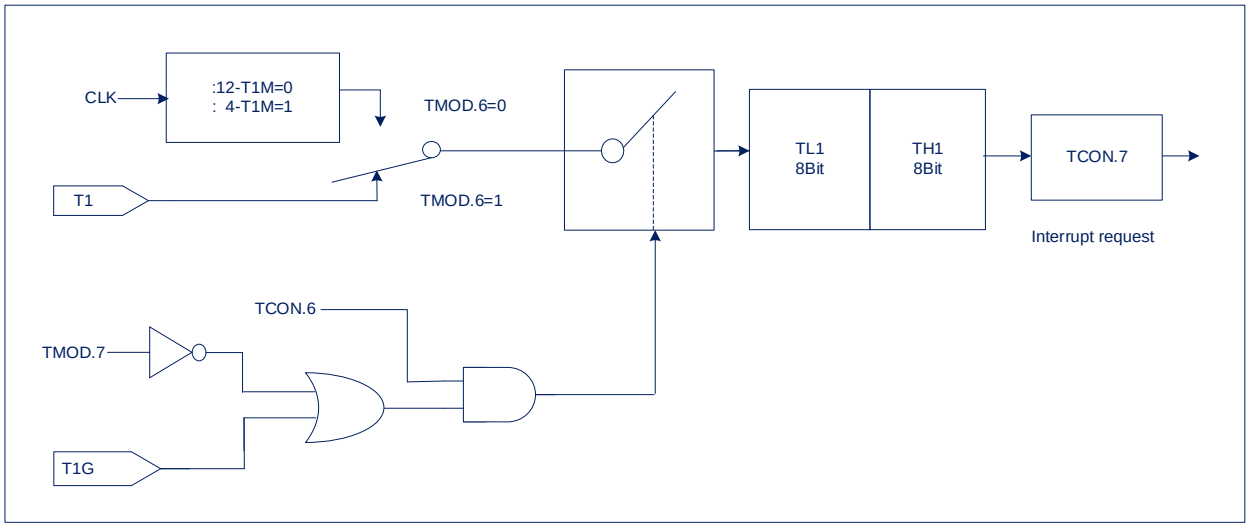
10.5.1 T1 - Mode 0 (13-bit Timer/Counting Mode)

In this mode, Timer 1 is a 13-bit register. When all the bits of the counter are turned 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 counting input is enabled to timer 1. (Setting TMOD.7=1 allows Timer 1 to be controlled by the external pin T1G for pulse width measurement). The 13-bit register consists of 8 bits of TH1 and the lower 5 bits of TL1. The upper three bits of TL1 should be ignored. The structure diagram of Timer1 mode 0 is shown in the figure below:



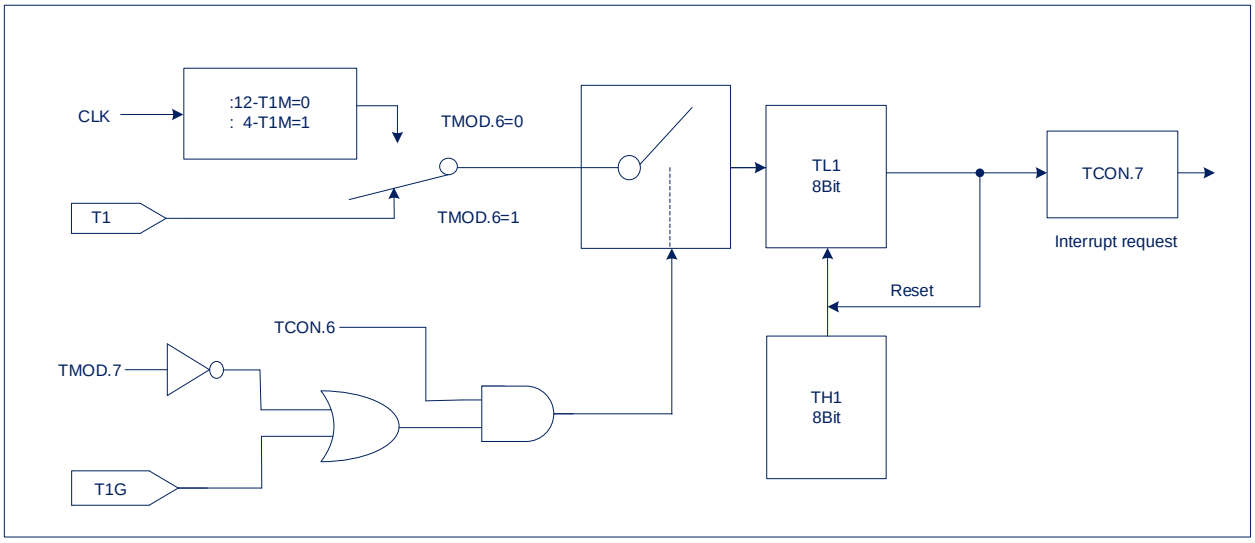
10.5.2 T1 - Mode 1 (16-bit Timer/Counting Mode)

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



10.5.3 T1 - Mode 2 (8-bit Auto Reload Timer/counter 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 sets TF1 to 1, but also reloads the contents of TH1 to TL1 by software. The value of TH1 remains unchanged during the reload process. The structure diagram of Timer1 Mode 2 is shown in the figure below:



10.5.4 T1 - Mode 3(Stop Counting)

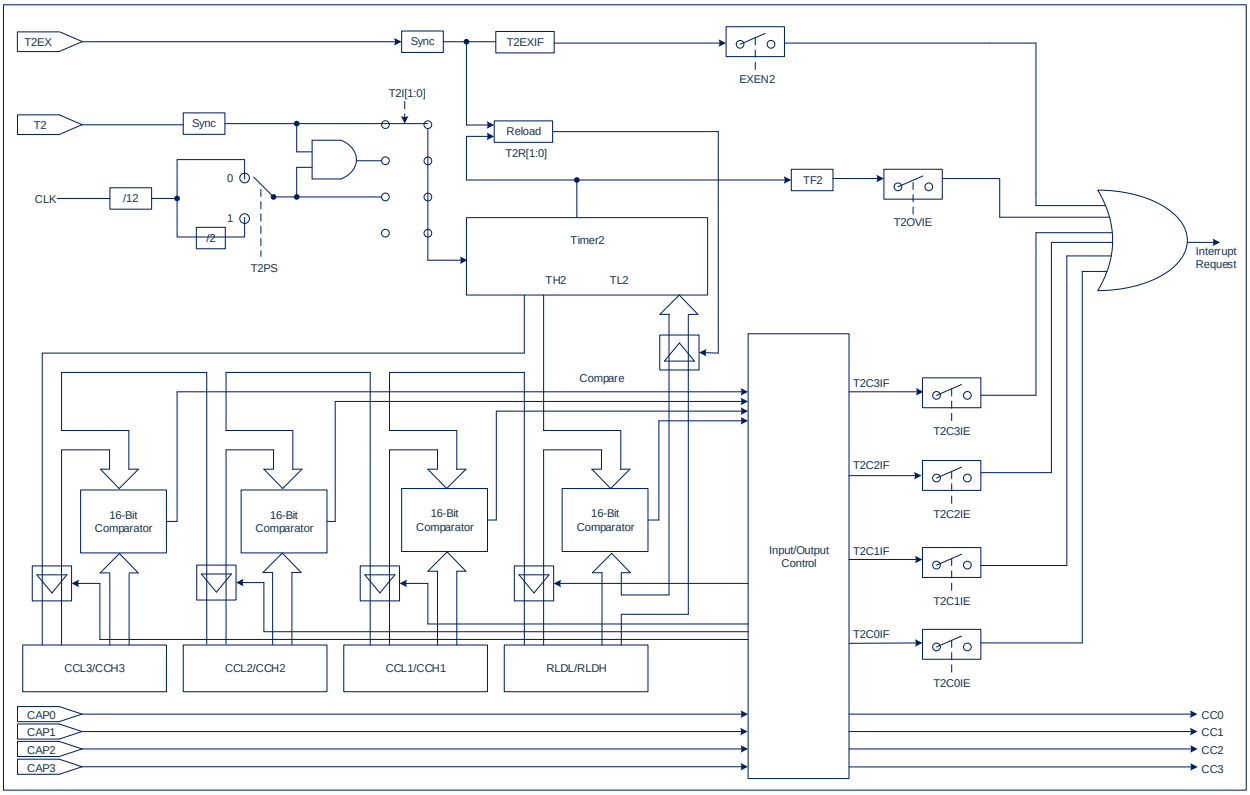
Timer 1 in mode 3 stops counting, and its effect is the same as setting TR1=0.

11. Timer 2 (Timer2)

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

11.1 Overview

The structure diagram of Timer 2 with additional compare/capture/reload register function is shown in the figure below:



11.2 Related Registers

11.2.1 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 single edge selection and compare interrupt time selection bit;
capture channel 0 mode:
1= Capture rising edge to RLDL/RLDH register;
0= Capture falling edge to RLDL/RLDH register.
Compare channel 0 mode:
1= Interrupt generated when TL2/TH2 and RLDL/RLDH are not equal to equal to each other;
0= Interrupt generated when TL2/TH2 and RLDL/RLDH are equal to not equal to each other;

Bit5 CAPES: Capture channel 1-3 input Single edge selection (valid for capture channels 1-3 together).
0= Rising edge is captured to the CCL1/CCH1-CCL3/CCH3 register;
1= Falling edge is captured to the CCL1/CCH1-CCL3/CCH3 register.
Initial capture edge selection in capture mode 2
0= Rising edge capture;
1= Falling edge capture.

Bit4~Bit3 T2R<1:0>: Timer2 loading mode selection bit;
0x= Reloading prohibited;
10= Loading mode 1: automatic reloading when Timer2 overflows;
11= Loading mode 2: reloading on the falling edge of T2EX pin.

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

Bit1~Bit0 T2I<1:0>: Timer2 clock input selection bit;
00= Timer2 stop;
01= System clock frequency division (selected by T2PS control frequency division);
10= External pin T2 as event input (event counting mode);
11= External pin T2 is used as gate control input (Gated timing mode).

11.2.2 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 used as counter low bit).

11.2.3 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 used as counter high bit).

11.2.4 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 bit.

11.2.5 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 bit.

11.2.6 Timer2 Compare/Capture 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 compare/capture channel 1 register low bit.

11.2.7 Timer2 Compare/Capture 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 compare/capture channel 1 register high bit.

11.2.8 Timer2 Compare/Capture 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 compare/capture channel 2 register low bit.

11.2.9 Timer2 Compare/Capture 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 compare/capture channel 2 register high bit.

11.2.10 Timer2 Compare/Capture 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 compare/capture channel 3 register low bit.

11.2.11 Timer2 Compare/Capture 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 compare/capture channel 3 register high bit.

11.2.12 Timer2 Compare/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 bit;
00= Capture/compare prohibited;
01= Capture operation is triggered on the rising or falling edge of channel 3 (CAPES selection) ;
10= Compare mode enable;
11= Capture operation is triggered when writing CCL3 or the both-edge trigger of channel 3.

Bit5~Bit4 CMH2-CML2: Capture/compare mode control bit;
00= Capture/comparison disabled;
01= Capture operation is triggered on the rising or falling edge of channel 2 (CAPES selection);
10= Comparison mode enable;
11= Capture operation is triggered when writing CCL2 or the both-edge trigger of channel 2.

Bit3~Bit2 CMH1-CML1: Capture/compare mode control bit;
00= Capture/compare disabled;
01= Capture operation is triggered on the rising or falling edge of channel 1 (CAPES selection);
10= Comparison mode enable;
11= Capture operation is triggered when writing CCL1 or the both-edge trigger of channel 1.

Bit1~Bit0 CMH0-CML0: Capture/compare mode control bit;
00= Capture/compare disabled;
01= Capture operation is triggered on the rising or falling edge of channel 0 (I3FR selection);
10= Comparison mode enable;
11= Capture operation is triggered when writing RLDL or the both-edge trigger of channel 0.

11.2.13 Timer2 Capture Mode 2 Control Register CAP2CON

0xC1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CAP2CON	CAP2ERR	--	--	--	CAP2LOCK	CAP2CLR	CAP2ST	CAP2EN
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 CAP2ERR: Timer overflow flag bit during Timer2 capture (hardware set and clear, read-only bit);
1= In Timer2 capture mode 2, the counter overflows during the capture process;
0= Timer2 CAP2EN/CAP2ST bit is 0, or the timer does not overflow during capture.
Bit6~Bit4 -- Reserved, all must be 0.
Bit3 CAP2LOCK: Timer2 capture mode 2 read lock function control bit;
1= Enable;
0= Disable.
Bit2 CAP2CLR: Timer2 capture completes clear the timer enable control bit;
1= Enable;
0= Disable.
Bit1 CAP2ST: Timer2 capture mode 2 start bit;
1= Start capture;
0= Stop capture.
Bit0 CAP2EN: Timer2 capture mode 2 enable control bit;
1= Enable;
0= Disable.

11.3 Timer2 Interrupt

Timer 2 can be enabled or disabled by the register IE, the total interrupt can also be set high/low priority by the IP register. Timer2 has 4 types of interrupts:

- Timer overflow interrupt.
- Interrupt on the falling edge of external pin T2EX.
- Compare interrupt.
- Capture interrupt.

To set the Timer2 interrupt, you need to configure the global interrupt enable bit (EA=1), the Timer2 total interrupt enable bit (ET2=1), and the corresponding interrupt type enable bit (T2IE) of Timer2. The 4 types of interrupts of Timer2 share an interrupt vector. After entering the interrupt service routine, it is necessary to determine the related flag bit to determine which type has generated the interrupt

11.3.1 Interrupt Related Registers

11.3.1.1 Interrupt Mask Register IE

0xA8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IE	EA	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 EA: Global interrupt enable Bit;
1= Allow all interrupts that are not masked;
0= Disable all interrupts.
Bit6 ES1: UART1 interrupt enable bit;
1= Enable UART1 interrupt;
0= Disable UART1 interrupt.
Bit5 ET2: TIMER2 total interrupt enable bit;
1= Enable all TIMER2 interrupts;
0= Disable all TIMER2 interrupts.
Bit4 ES0: UART0 interrupt enable bit;
1= Enable UART0 interrupt;
0= Disable UART0 interrupt.
Bit3 ET1: TIMER1 interrupt enable bit;
1= Enable TIMER1 interrupt;
0= Disable TIMER1 interrupt.
Bit2 EX1: External interrupt 1 interrupt enable bit;
1= Enable external interrupt 1 interrupt;
0= Disable external interrupt 1 interrupt.
Bit1 ET0: TIMER0 interrupt enable bit;
1= Enable TIMER0 interrupt;
0= Disable TIMER0 interrupt.
Bit0 EX0: External interrupt 0 interrupt enable bit;
1= Enable external interrupt 0 interrupt;
0= Disable external interrupt 0 interrupt.

11.3.1.2 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 bit;
1= Enable interrupt;
0= Disable interrupt.
Bit6 T2EXIE: Timer2 external load interrupt enable bit;
1= Enable interrupt;
0= Disable interrupt.
Bit5~Bit4 -- Reserved, all must be 0.
Bit3 T2C3IE: Timer2 compare/capture channel 3 interrupt enable bit;
1= Enable interrupt;
0= Disable interrupt.
Bit2 T2C2IE: Timer2 compare/capture channel 2 interrupt enable bit;
1= Enable interrupt;
0= Disable interrupt.
Bit1 T2C1IE: Timer2 compare/capture channel 1 interrupt enable bit;
1= Enable interrupt;
0= Disable interrupt.
Bit0 T2C0IE: Timer2 compare/capture channel 0 interrupt enable bit;
1= Enable interrupt;
0= Disable interrupt.

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

11.3.1.3 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 PS1: UART1 interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.

Bit5 PT2: TIMER2 interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.

Bit4 PS0: UART0 interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.

Bit3 PT1: TIMER1 interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.

Bit2 PX1: External interrupt 1 interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.

Bit1 PT0: TIMER0 interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.

Bit0 PX0: External interrupt 0 interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.

11.3.1.4 Timer2 Interrupt Flag 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 overflows and needs to be cleared by software;
0= Timer2 counter does not overflow.

Bit6 T2EXIF: Timer2 external load flag bit;
1= Timer2 T2EX port generates a falling edge, which needs to be cleared by software;
0= --

Bit5~Bit4 -- Reserved, all must be 0.

Bit3 T2C3IF: Timer2 compare/capture channel 3 flag bit;
1= Timer2 compare channel 3 {CCH3:CCL3}={TH2:TL2} or capture channel 3 has a capture operation, which needs to be cleared by software.
0= --

Bit2 T2C2IF: Timer2 compare/capture channel 2 flag bit;
1= Timer2 compare channel 2 {CCH2:CCL2}={TH2:TL2} or capture channel 2 has a capture operation, which needs to be cleared by software.
0= --

Bit1 T2C1IF: Timer2 compare/capture channel 1 flag bit;
1= Timer2 compare channel 1 {CCH1:CCL1}={TH2:TL2} or capture channel 1 has a capture operation, which needs to be cleared by software.
0= --

Bit0 T2C0IF: Timer2 compare/capture channel 0 flag bit;
1= Timer2 compare channel 0{RLDH:RLDL}={TH2:TL2} or capture channel 0 has a capture operation, which needs to be cleared by software.
0= --

11.3.2 Timer Interruption

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

11.3.3 External Trigger Interrupt

External pin T2EX falling edge trigger interrupt enable bit is set by register T2IE[6], and the interrupt flag bit is checked by register T2IF[6]. When the T2EX pin falls, the external load interrupt flag bit T2EXIF will be set to 1.

11.3.4 Compare Interrupt

All 4 compare channels support compare interrupts. The compare interrupt enable bit is set by the register T2IE[3:0], and the interrupt flag bit is checked by the register T2IF[3:0].

The comparison channel 0 can select the time when the comparison interrupt is generated, and when an interrupt is generated, the interrupt flag T2C0IF of the comparison channel 0 will be set to 1.

When I3FR = 0, TL2/TH2 and RLDL/RLDH will generate an interrupt from unequal to equal time;

When I3FR = 1, TL2/TH2 and RLDL/RLDH will be interrupted when they are equal to unequal;

The comparison channel 1~3 cannot select the interrupt generation time, and it is fixed to generate interrupts at the time when TL2/TH2 and CCxL/CCxH are not equal to equal. If an interrupt is generated, the corresponding compare channel interrupt flag T2CxIF will be set to 1.

11.3.5 Capture Interrupt

4 capture channels all support external capture interrupt. The capture interrupt enable bit is set by the register T2IE[3:0], and the interrupt flag bit is checked by the register T2IF[3:0]. When a capture operation occurs, the interrupt flag T2CxIF of the corresponding capture channel is set to 1.

Note that the write capture mode will not generate an interrupt.

11.4 Timer2 Function Description

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

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

Set different modes of Timer 2, which can be used for the generation of various digital signals and event capture, such as pulse generation, pulse width modulation, pulse width measurement, etc.

11.4.1 Timing Mode

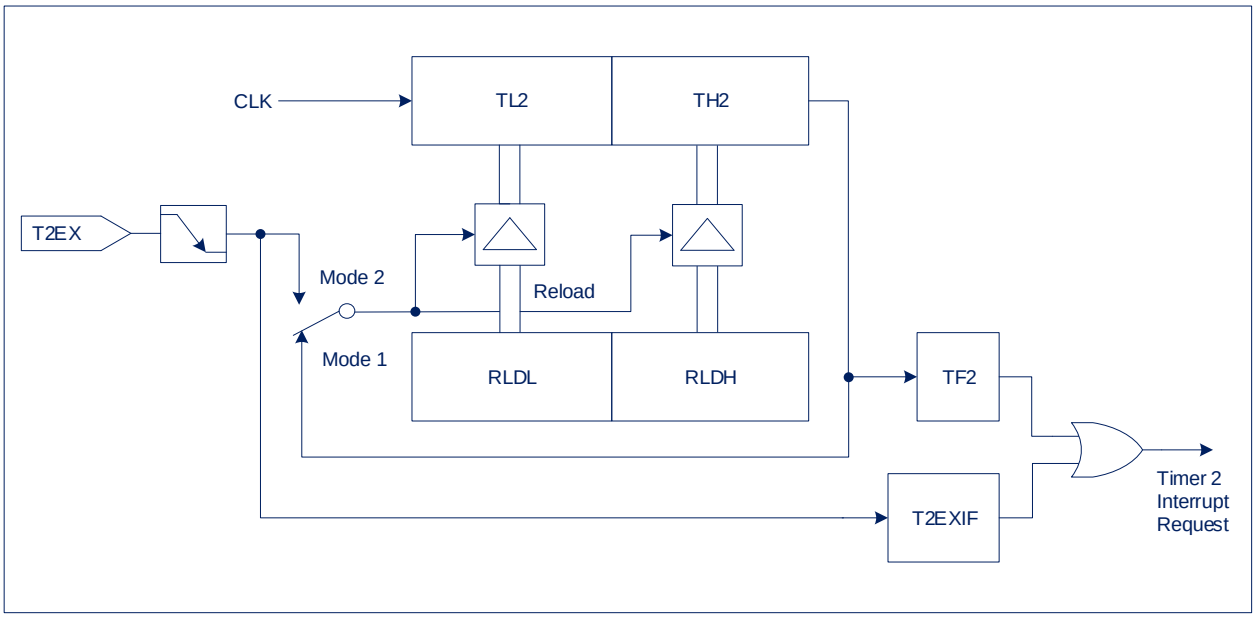
When used as a timer function, the clock source is derived from the system clock. The prescaler provides 1/12 or 1/24 system frequency selection, and the value of the prescaler is selected by the T2PS bit of the register T2CON. So the 16-bit timer register (consisting of TH2 and TL2) increments every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

11.4.2 Reload Mode

The reload mode of Timer 2 is selected by the T2R0 and T2R1 bits of the register T2CON. The block diagram of the reload structure is shown in the figure below.

In load mode 1: When the Timer2 counter rolls over from all 1s to 0 (counter overflows), not only the overflow interrupt flag bit TF2 is set to 1, but the Timer2 register automatically loads the 16-bit value from the RLDL/RLDH register, thereby overwriting the count value 0x0000, the required RLDL/RLDH value can be preset by 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 external load interrupt flag bit T2EXIF is set to 1, and Timer2 automatically loads the 16-bit value of the RLDL/RLDH register as the initial value of the count.



11.4.3 Gated Timing Mode

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

11.4.4 Event Counting Mode

Timer2 is used as the event counting function, the timer counter adds 1 to the falling edge of the external input pin T2. The external input signal is sampled in each system clock cycle. When the sampled input shows a high level in one cycle and a low level in the next cycle, the count is increased. In the next cycle, when the T2 pin changes from high to low, the new count value is updated to the timer data register.

11.4.5 Compare Mode

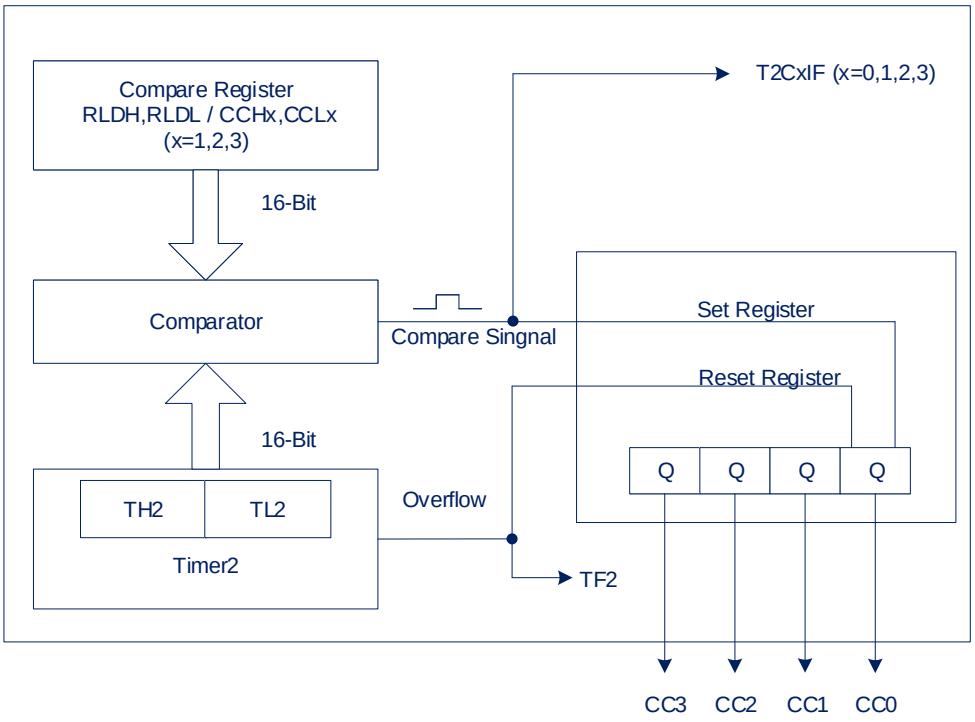
The comparison function includes two modes: comparison mode 0 and comparison mode 1, which are selected by the T2CM bit in the special function register T2CON. These two compare modes can generate periodic signals and can change the duty cycle control mode. They are often used in pulse width modulation (PWM) and control situations that require continuous square wave generation, covering a wider range of applications.

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

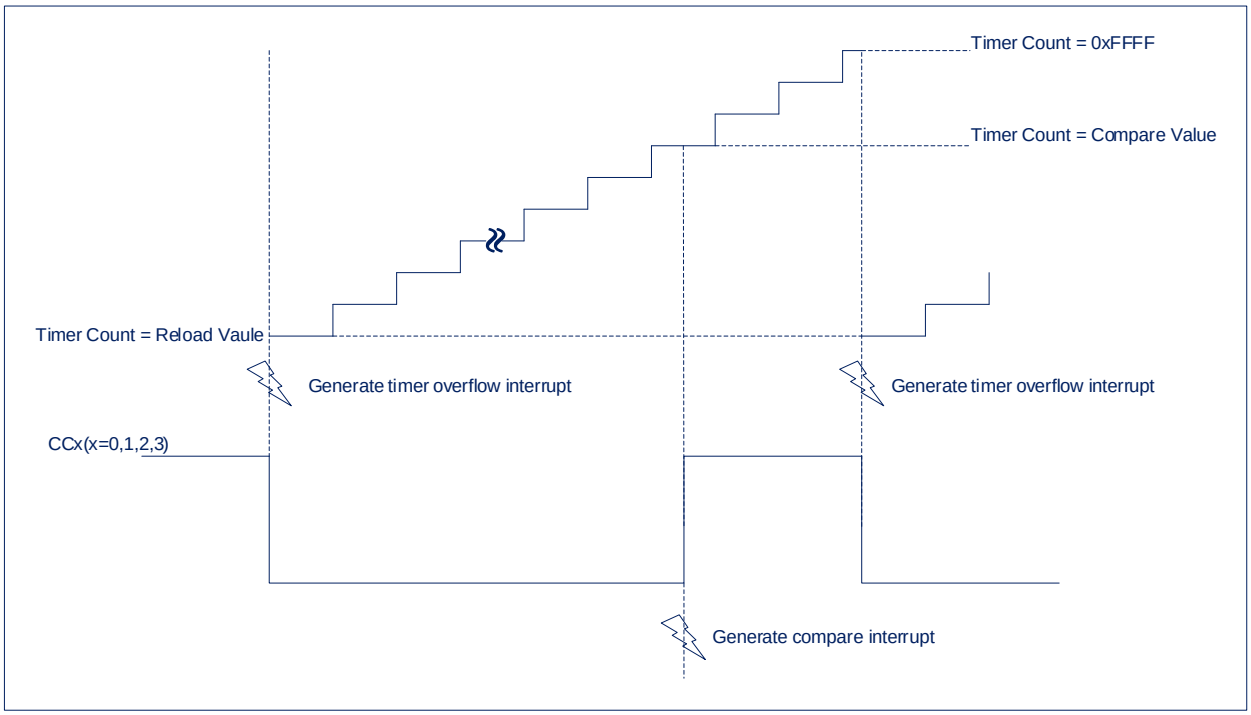
The 16-bit stored value stored in the compare register is compared with the count value of the timer. If the count value in the data register matches the stored value, a transition of the output signal will be generated on the corresponding port pin. Interrupt flag bit.

11.4.5.1 Compare Mode 0

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



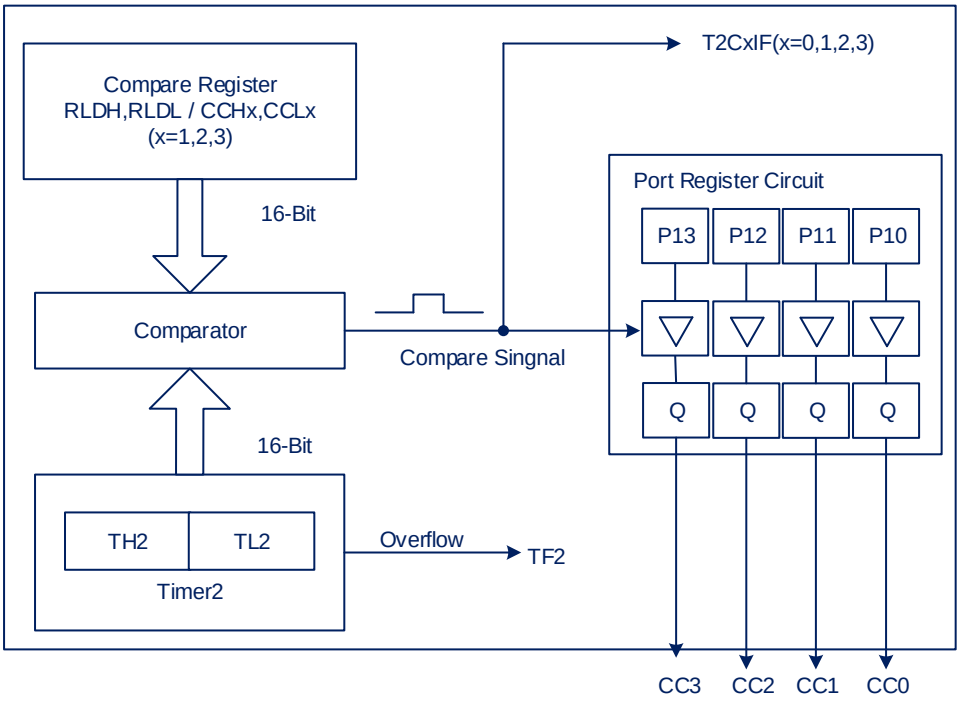
the output structure block diagram of comparison mode 0 is shown in the following figure:



11.4.5.2 Compare Mode 1

In compare mode 1, it is usually used when the output signal has nothing to do with the constant signal period, and the software determines the output signal adaptively transition occasions.

If mode 1 is enabled, the software will write to the corresponding output register of the CC3 port, and the new value will 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 one of two ways to change the output signal or keep its old value. The block diagram of comparison mode 1 is shown in the figure below:



11.4.6 Capture Mode

Each of the four 16-bit registers {RLDH, RLDL}, {CCH1, CCL1}, {CCH2, CCL2}, {CCH3, CCL3} that function as capture functions can be used to lock the current value of {TH2, TL2} 16-bit value. This function provides three different capture modes.

In Mode 0, an external event can lock the contents of Timer 2 into the capture register.

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

Capture channels 0~3 respectively select capture input pins CAP0~CAP3 as input source signals.

In mode 2, the input pin CAP1 is captured as the input source signal. Capture mode 2 is used to capture the signal of the input channel CAP1 three times in a row, and the capture values of the three times are loaded into the CCL1/CCH1, CCL2/CCH2, CCL3/CCH3 registers in turn.

When Capture Mode 2 is enabled, Channel 1, Channel 2, and Channel 3 will be forced to function as Capture Mode 2, and Channel 0 can do any function without being affected.

11.4.6.1 Capture Mode 0

In capture mode 0, the positive transition, negative transition or positive and negative transition on the capture channel 0~3 (CAP0~CAP3) will generate a capture event. When a capture event occurs, the count value of the timer is locked in the corresponding capture register.

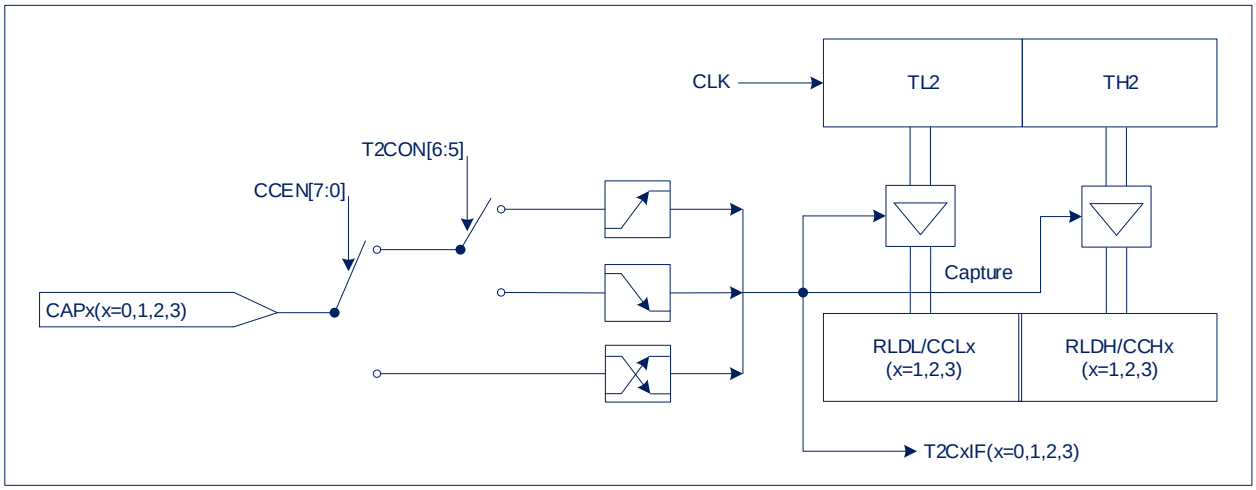
Whether the positive transition triggers the capture operation or the negative transition triggers the capture operation on the capture channel 0 depends on the I3FR bit of T2CON. I3FR=0, negative transition triggers capture; I3FR=1, positive transition triggers capture.

Whether the positive transition triggers the capture operation or the negative transition triggers the capture operation on the capture channels 1~3 depends on the CAPES bit of T2CON. CAPES=0, positive transition triggers capture; CAPES=1, negative transition triggers capture. The selected jump mode of capture channel 1~3 is that the same

capture channel 0~3 supports double jump capture operation at the same time. If the corresponding work mode control bit of the CCEN register is selected as 11, the channel supports the capture operation of double jump. It should be noted that this working mode also supports capture mode 1, that is, write operations can generate capture actions.

In capture mode 0, the external capture events of capture channels 0~3 can generate interrupts.

The structure diagram of capture mode 0 is shown in the following figure:

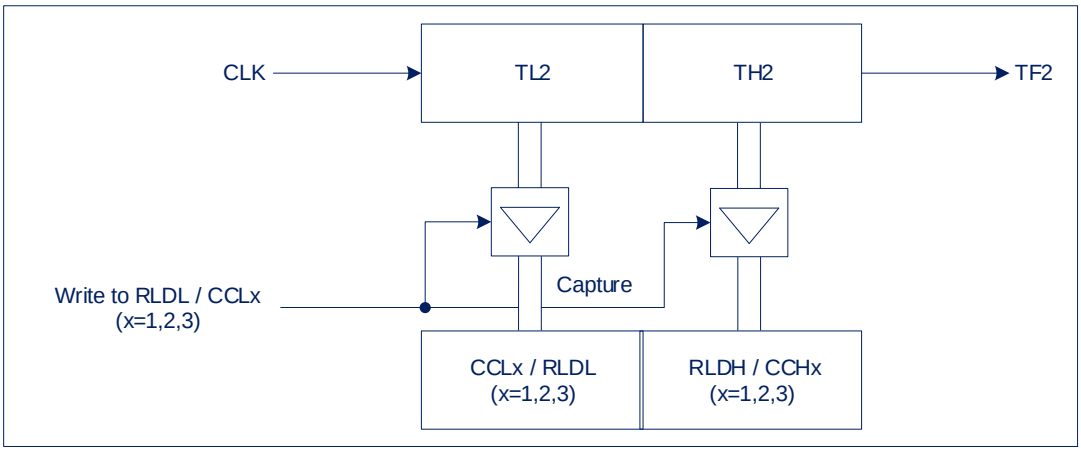


11.4.6.2 Capture Mode 1

In capture mode 1, the capture operation event is the execution of the instruction to write the low byte of the capture register. The write register signal (for example, write RLDL) starts the capture operation, and the written value has nothing to do with this function. After the write instruction is executed, the contents of Timer 2 will be locked into the corresponding capture register.

In capture mode 1, the capture event of capture channel 0~3 will not generate an interrupt request flag.

The structure diagram of capture mode 1 is shown in the figure below:



11.4.6.3 Capture Mode 2

In capture mode 2, the input signal of capture channel CAP1 is captured three times in a row. The three captures are completed as a complete capture event, the capture event is generated, and the three capture values are loaded into the CCL1/CCH1, CCL2/CCH2, CCL3/CCH3 registers in turn.

The capture edge is selected by T2CON[5], T2CON[5]=1 indicates that the first capture is a falling edge capture, the second capture is a rising edge, and the third capture is a falling edge capture; T2CON[5]=0 indicates that the first capture is a rising edge capture, the second capture is a falling edge, and the third capture is a rising edge capture. When capturing consecutively, the third capture edge of the previous capture is the first capture edge of the next capture.

The enable of capture mode 2 is controlled by register CAP2CON[0], the capture start is controlled by CAP2CON[1], the interrupt flag T2IF[1] is set to 1 after the completion of the three captures, the timer overflows CAP2CON[7] in the second or third capture sets to 1 by the hardware, until the next complete capture process without timer overflow, CAP2CON[7] is set to 0 by the hardware.

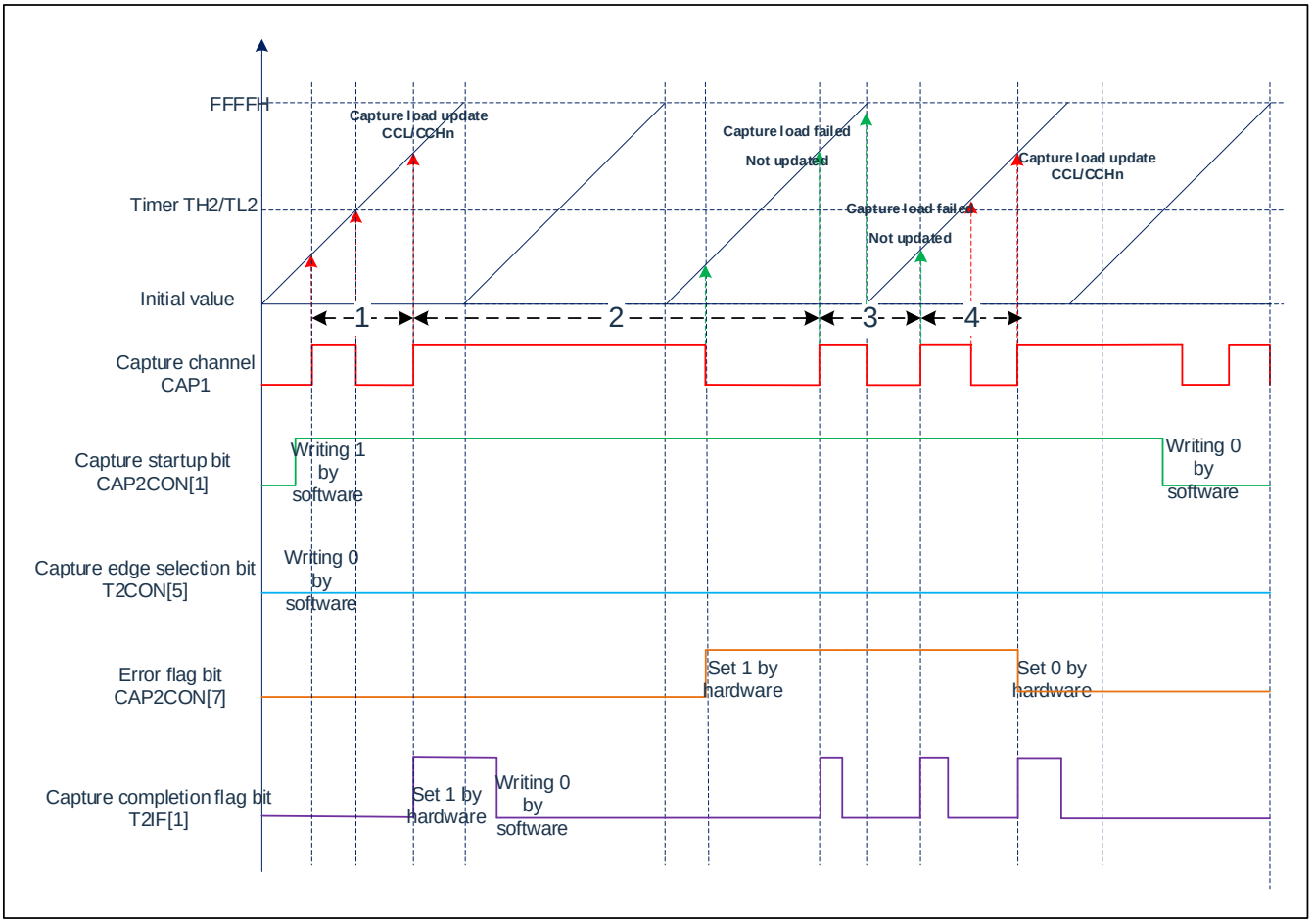
When the read lock function is turned on (CAP2CON[3]=1), the values of the CCL1/CCH1, CCL2/CCH2, CCL3/CCH3 registers are the values that were last captured by the system when writing CAP2CON[3]=1, and the values of the registers will be locked and will not be updated with a new capture. When the read lock function is off (CAP2CON[3]=0), the values of the CCL1/CCH1, CCL2/CCH2, CCL3/CCH3 registers are updated in real time to the new captured values. Therefore, it is recommended that when reading the captured data, turn on the read lock function, and wait for the data reading to complete before turning off the read lock function.

In capture mode 2, you can set the clear timer counter after the capture event is generated (CAP2CON[2]=1). If the enable capture completes clearing the timer, the timer will start counting again from 0 after the third capture of each capture cycle is completed, that is, except for the first capture when CCL1/CCH1 may not be 0, the CCL1/CCH1 of each subsequent capture is 0.

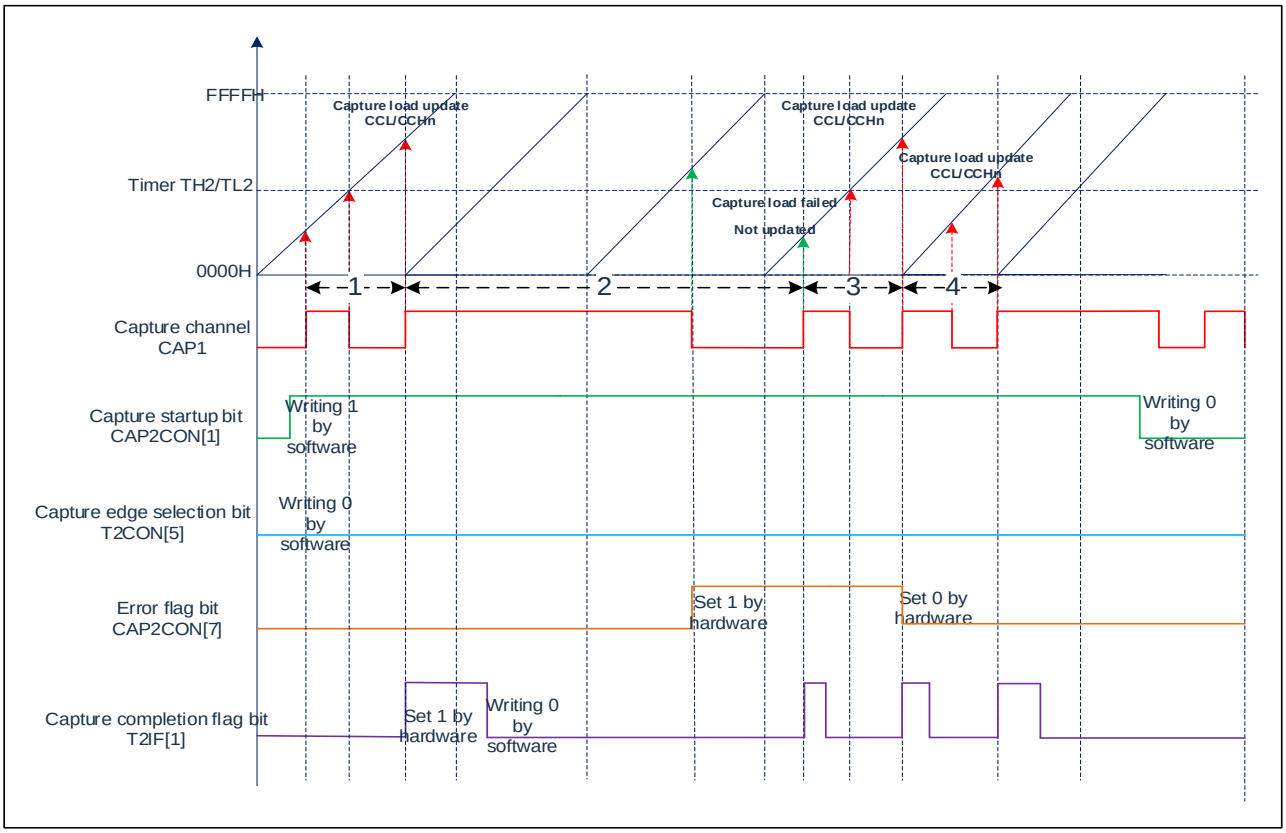
Note: The capture completion timer function is enabled in capture mode 2, and it should not be shared with the T2EX falling-edge auto-reload mode (it is impossible to confirm whether the captured value is the value after the reload, resulting in

inaccurate capture results).

When CAP2CON[2]=0, the block diagram of capture mode 2 is shown in the following figure:



When CAP2CON[2]=1, the block diagram of capture mode 2 is shown in the following figure:



12. Timer3/4

Timer 3/4 is similar to timer 0/1, and is two 16-bit timers. Timer 3 has four working modes, and Timer 4 has three working modes. Compared with Timer0/1, Timer3/4 only provides timing operation.

When the timer is started, the value of the register is incremented every 12 or 4 system cycles.

12.1 Overview

Timer 3 and Timer 4 are composed of two 8-bit registers {TH3, TL3} and {TH4, TL4}. Timers 3 and 4 work in the same four modes. Timer3 and Timer4 modes are described as follows:

Mode	M1	M0	Function description
0	0	0	THx[7:0]，TLx[4:0] form a 13-bit timer
1	0	1	THx[7:0]，TLx[7:0] form a 16-bit timer
2	1	0	TLx[7:0] form an 8-bit auto-reload timer, reload from THx
3	1	1	TL3 and TH3 are two 8-bit timers, Timer4 stops counting

12.2 Related Registers

12.2.1 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 running control bit;
1= Timer4 is started;
0= Timer4 is closed.

Bit6 T4M: Timer4 clock selection bit;
1= Fsys/4;
0= Fsys/12.

Bit5~Bit4 T4M<1:0>: Timer4 mode selection bit;
00= Mode 0, 13-bit timer;
01= Mode 1, 16-bit timer;
10= Mode 2, 8-bit auto-reload timer;
11= Mode 3, stop counting.

Bit3 TR3: Timer3 running control bit;
1= Timer3 is started;
0= Timer3 is closed.

Bit2 T3M: Timer3 clock selection bit;
1= Fsys/4;
0= Fsys/12.

Bit1~Bit0 T3M<1:0>: Timer3 mode selection bit;
00= Mode 0, 13-bit timer;
01= Mode 1, 16-bit timer;
10= Mode 2, 8-bit auto-reload timer;
11= Mode 3, two independent 8-bit timer.

12.2.2 Timer3 Data Register Low 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 (also used as timer low-bit).

12.2.3 Timer3 Data Register High 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 used as timer high-bit)..

12.2.4 Timer4 Data Register Low 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 used as timer low-bit).

12.2.5 Timer4 Data Register High 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 used as timer high-bit).

12.3 Timer3/4 Interrupt

Timer 3/4 can be enabled or disabled through the EIE2 register, and the high/low priority can also be set through the EIP2 register. The interrupt related bits are as follows:

12.3.1 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 interrupt;
0= Disable SPI interrupt.
Bit6 I2CIE I^2C interrupt enable bit;
1= Enable I^2C interrupt;
0= Disable I^2C interrupt.
Bit5 WDTIE: WDT interrupt enable bit;
1= Enable WDT overflow interrupt;
0= Disable WDT overflow interrupt.
Bit4 ADCIE: ADC interrupt enable bit;
1= Enable ADC interrupt;
0= Disable ADC interrupt.
Bit3 PWMIE: PWM total interrupt enable bit;
1= Enable all PWM interrupt;
0= Disable all PWM interrupt.
Bit2 -- Reserved, must be 0.
Bit1 ET4: Timer4 interrupt enable bit;
1= Enable Timer4 interrupt;
0= Disable Timer4 interrupt.
Bit0 ET3: Timer3 interrupt enable bit;
1= Enable Timer3 interrupt;
0= Disable Timer3 interrupt.

12.3.2 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

	PSPI:	SPI interrupt priority control bit;
Bit7	1=	Set as high-level interrupt;
	0=	Set as low-level interrupt.
	PI2C:	I^2C interrupt priority control bit;
Bit6	1=	Set as high-level interrupt;
	0=	Set as low-level interrupt.
	PWDT:	WDT interrupt priority control bit;
Bit5	1=	Set as high-level interrupt;
	0=	Set as low-level interrupt.
	PADC:	ADC interrupt priority control bit;
Bit4	1=	Set as high-level interrupt;
	0=	Set as low-level interrupt.
	PPWM:	PWM interrupt priority control bit;
Bit3	1=	Set as high-level interrupt;
	0=	Set as low-level interrupt.
Bit2	--	Reserved, must be 0.
	PT4:	TIMER4 interrupt priority control bit;
Bit1	1=	Set as high-level interrupt;
	0=	Set as low-level interrupt.
Bit0	PT3:	TIMER3 interrupt priority control bit;
	1=	Set as high-level interrupt;
	0=	Set as low-level interrupt.

12.3.3 Peripheral Interrupt Flag 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 total interrupt flag bit, read only;
1= SPI generate an interrupt, (after clearing the specific interrupt flag bit, this bit is automatically cleared);
0= SPI does not generate an interrupt.

Bit6 I2CIF: I^2C total interrupt flag bit, read only;
1= I^2C generate an interrupt, (after clearing the specific interrupt flag bit, this bit is automatically cleared);
0= I^2C does not generate an interrupt.

Bit5 -- Reserved, must be 0.

Bit4 ADCIF: ADC interrupt flag bit;
1= ADC conversion is completed and needs to be cleared by software;
0= ADC conversion is not completed.

Bit3 PWMIF: PWM total interrupt flag bit, read only;
1= PWM generate an interrupt, (after clearing the specific interrupt flag bit, this bit is automatically cleared);
0= PWM does not generate an interrupt.

Bit2 -- Reserved, must be 0.

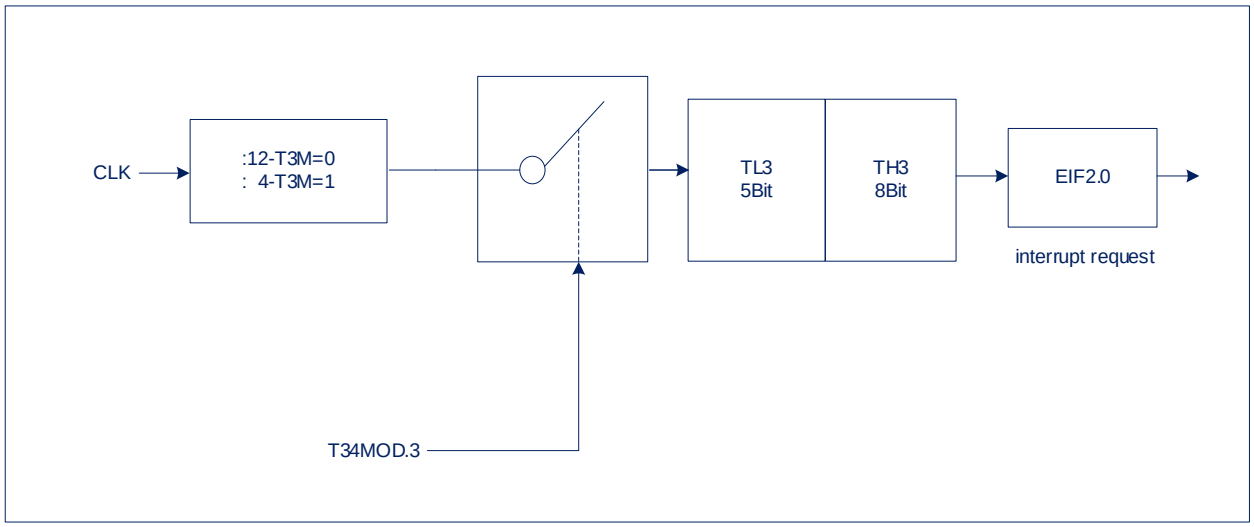
Bit1 TF4: Timer4 timer overflow interrupt flag bit;
1= Timer4 timer overflow, it is automatically cleared by hardware when entering the interrupt service routine, and can also be cleared by software;
0= Timer4 timer does not overflow.

Bit0 TF3: Timer3 timer overflow interrupt flag bit;
1= Timer3 timer overflow, it is automatically cleared by hardware when entering the interrupt service routine, and can also be cleared by software;
0= Timer3 timer does not overflow.

12.4 Timer3 Working Mode

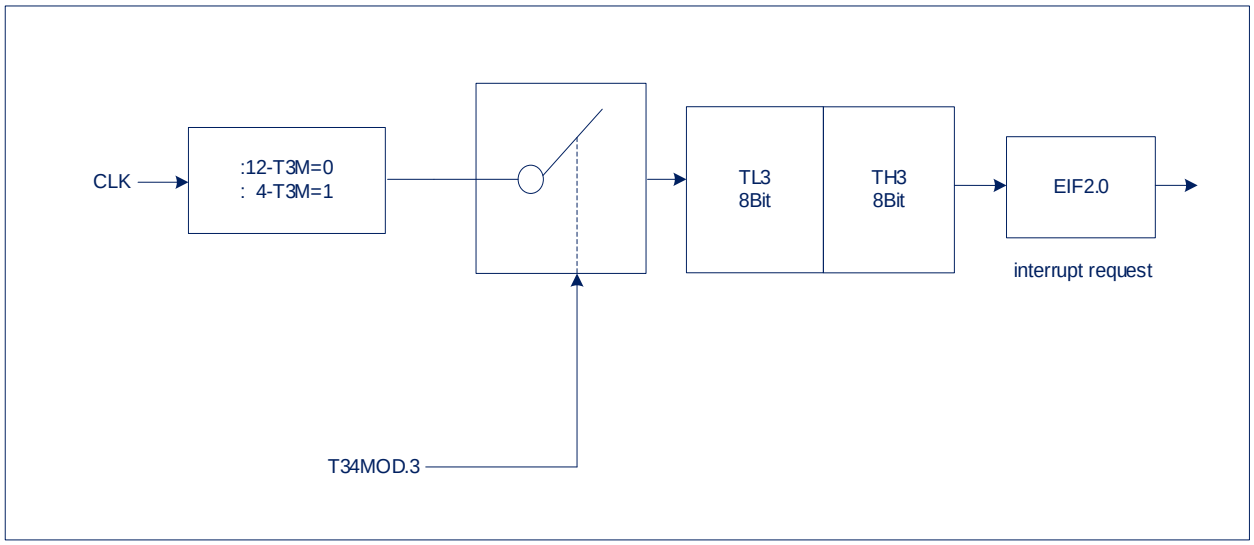
12.4.1 T3 -Mode 0 (13-bit Timing Mode)

In this mode, Timer3 is a 13-bit register. When all the bits of the timer are turned from 1 to 0, the timer 3 interrupt flag TF3 is set to 1. The 13-bit register is composed of TH3 and TL3 low 5 bits. The upper 3 bits of TL3 should be ignored. The structure diagram of Timer3 Mode 0 is shown in the figure below:



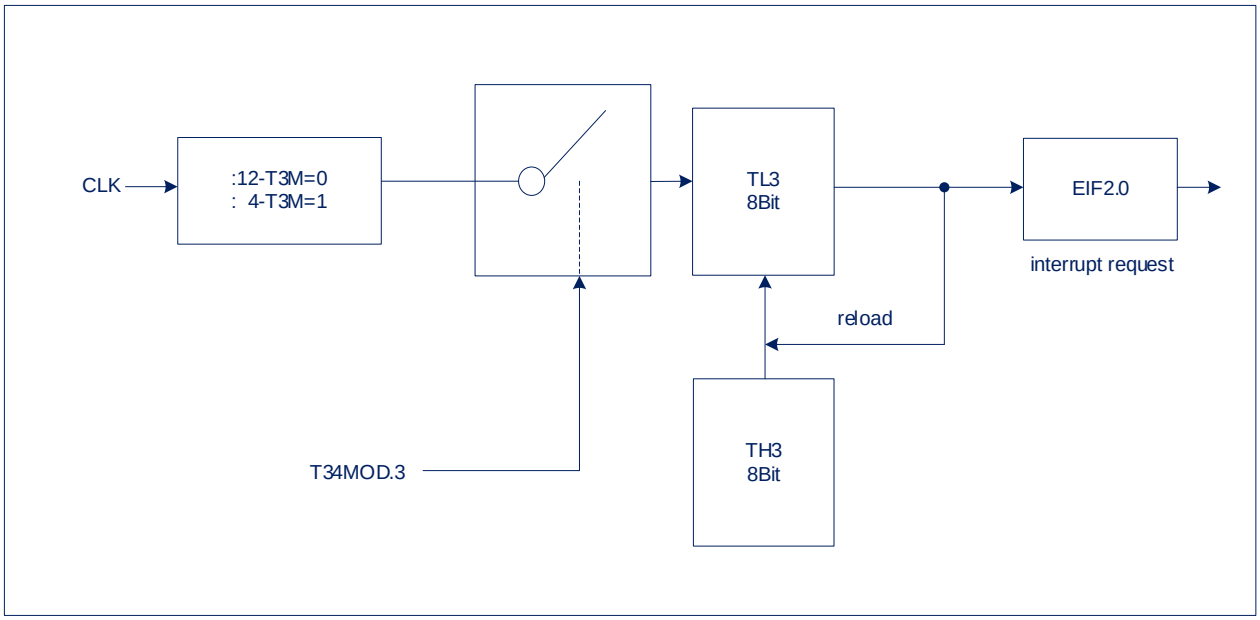
12.4.2 T3 -Mode 1 (16-bit Timing Mode)

Mode 1 is the same as Mode 0, except that all 16 bits of the Timer 3 register run in Mode 1. The structure diagram of Timer3 mode 1 is shown in the figure below:



12.4.3 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 shown. The overflow from TL3 not only sets TF3 to 1, but also reloads the contents of TH3 to TL3 by software. The value of TH3 remains unchanged during the reloading. The structure diagram of Timer3 Mode 2 is shown in the figure below:



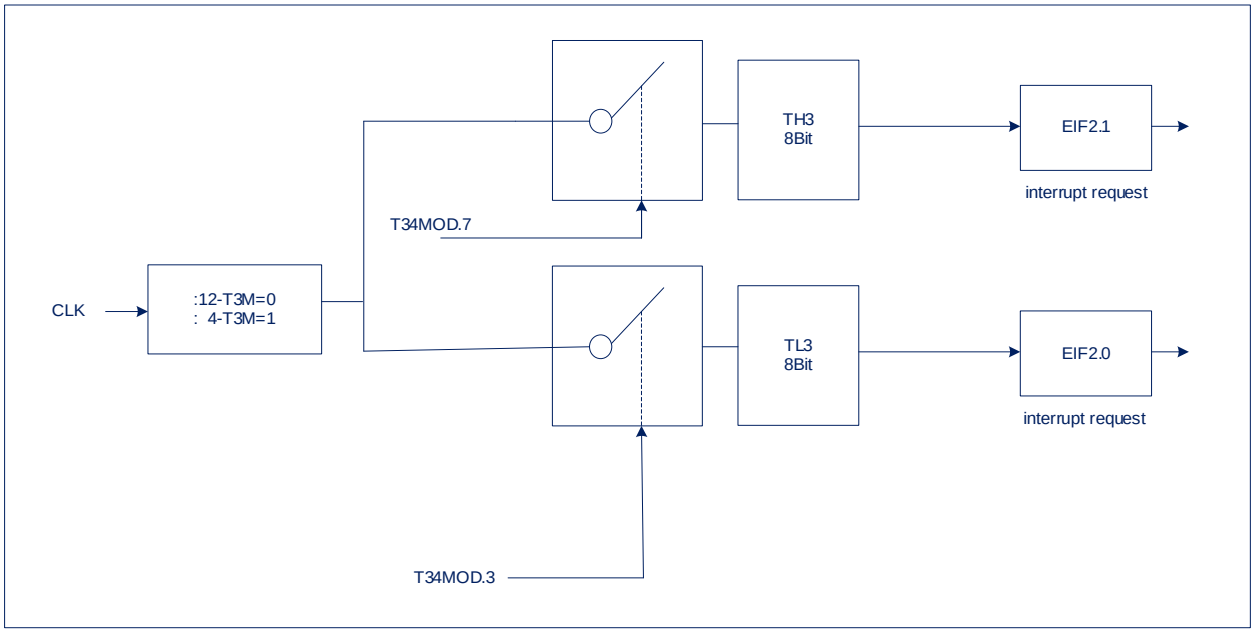
12.4.4 T3 -Mode 3 (Two Independent 8-bit Timers)

Timer 3 in Mode 3 sets TL3 and TH3 as two independent timers. The logic of Timer 3 Mode 3 is shown in the figure below.

TL3 works as an 8-bit timer and uses timer 3 control bits: such as TR3, and TF3.

TH3 works as an 8-bit timer, and uses the TR4 and TF4 flags of timer 4 and controls the timer 4 interrupt.

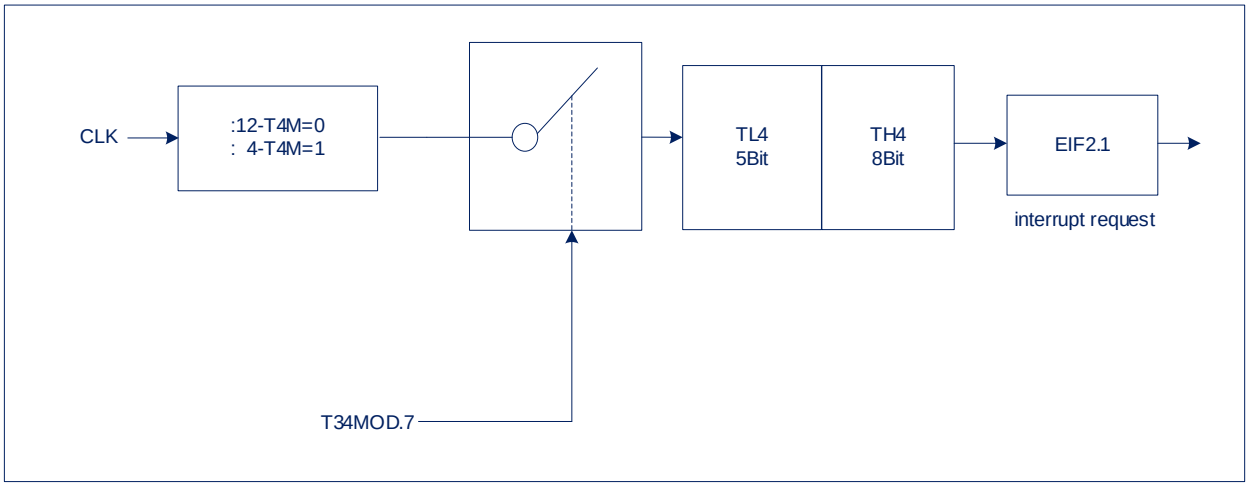
Mode 3 can be used when two 8-bit timers are required. When timer 3 is in mode 3, timer 4 can be turned off by switching to its own mode 3, or it can still be used as a baud rate generator by the serial channel, or in any case that does not require timer 4 interrupts. In application. The structure diagram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



12.5 Timer4 Working Mode

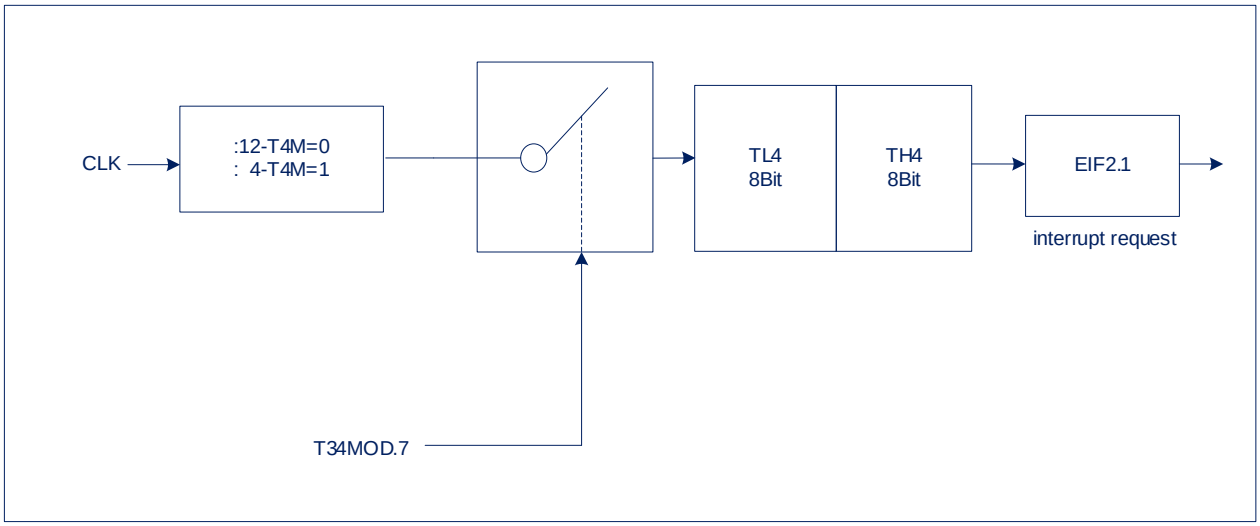
12.5.1 T4 -Mode 0 (13-bit Timing Mode)）

In this mode, Timer4 is a 13-bit register. When all the bits of the timer are turned from 1 to 0, the timer4 interrupt flag TF4 is set to 1. The 13-bit register is composed of TH4 and TL4 low 5 bits. The upper 3 bits of TL4 should be ignored. The structure diagram of Timer4 Mode 0 is shown in the figure below:



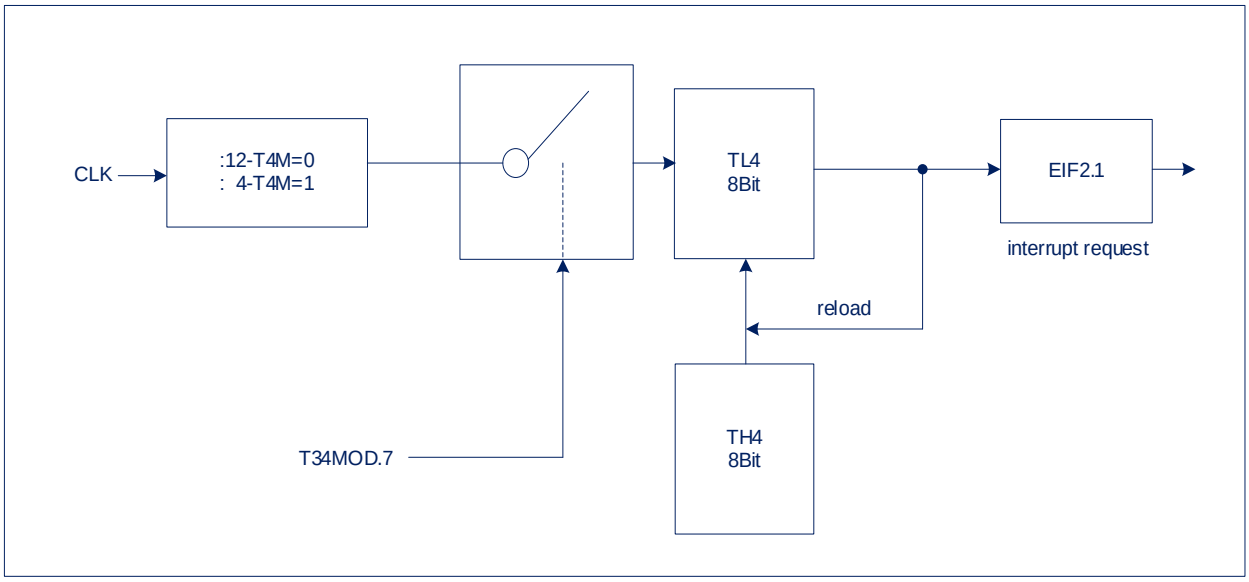
12.5.2 T4 -Mode 1 (16-bit Timing Mode)

Mode 1 is the same as Mode 0, except that all 16 bits of the Timer 4 register run in Mode 1. The structure diagram of Timer4 mode 1 is shown in the figure below:



12.5.3 T4 -Mode 2 (8-bit Auto-reload Timing Mode)

The timer4 register in mode 2 is an 8-bit timer (TL4) with auto-reload mode, as shown in the figure below shown. The overflow from TL4 not only sets TF4 to 1, but also reloads the contents of TH4 to TL4 by software. The value of TH4 remains unchanged during the reloading. The structure diagram of Timer4 Mode 2 is shown in the figure below:



12.5.4 T4 -Mode 3 (Stop Counting)

In mode 3, timer4 stops counting, and its effect is the same as setting TR4=0.

13. LSE_Timer

13.1 Overview

LSE timer is a 16-bit up-counting timer with a clock source from the external low-speed clock LSE. When using the LSE timer function, you should first set the LSE module to be enabled, 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 at the rising edge of the LSE clock. When the count value is equal to the timing value, the interrupt flag bit LSECON[0] is set to 1, and the counter starts counting from 0 again. The timing value is set by the register {LSECRH[7:0], LSECRL[7:0]}.

If the LSE timing function is configured before the sleep, the LSE oscillator and LSE timer can continue to work when the chip sleeping without being affected. If the LSE timer wake-up function is set before the sleep, when the count value is equal to the timing value, the system will be wake up.

13.2 Related Registers

13.2.1 LSE Timer Data Register Low 8 bits 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 low 8 bits.

13.2.2 LSE Timer Data Register High 8 bits 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 high 8 bits.

13.2.3 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 is used for timer counting enable control;
1= Enable;
0= Disable.

Bit4 LSESTA: LSE stable status bit, read-only;
1= LSE is stable;
0= LSE is not stable.

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

Bit2~Bit1 -- Reserved, must be 0.

Bit0 LSEIF: LSE is used as timer interrupt flag bit (cleared by software);
1= Generate interrupt;
0= No interrupt is generated or the interrupt is cleared.

13.3 Interrupt and Sleep Wakeup

LSE timer can enable or disable interrupt through LSECON register, set high/low priority through EIP3 register, and the related bits of interrupt are as follows.

0xBB	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP3	--	--	PWWDT	--	--	PLSE	PUART3	PUART2
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	PWWDT	WWDT interrupt priority control bit;
	1=	Set as high-level interrupt;
	0=	Set as low-level interrupt.
Bit4~ Bit3	--	Reserved, must be 0.
Bit2	PLSE:	LSE interrupt priority control bit;
	1=	Set as high-level interrupt;
	0=	Set as low-level interrupt.
Bit1	PUART3:	UART3 interrupt priority control bit;
	1=	Set as high-level interrupt;
	0=	Set as low-level interrupt.
Bit0	PUART2:	UART2 interrupt priority control bit;
	1=	Set as high-level interrupt;
	0=	Set as low-level interrupt.

When the count value of the LSE timer is equal to the timing value, the timer interrupt flag bit LSEIF is set to 1. If the global interrupt is enabled (EA=1) and the LSE timer interrupt is enabled (LSEIE=1), the CPU will execute the interrupt service routine.

By use the LSE timer interrupt to wake up the sleep mode, you need to turn on LSEEN, LSECNT, LSEWIEN before the sleep, and set the time from sleep state to wakeup$\{LSECRH[7:0], LSECRL[7:0]\}$. If the global interrupt enable and LSE interrupt enable are turned on before the sleep, after the sleep wakes up, the interrupt service routine will be executed first, and the next instruction of the sleep instruction will be executed after the interrupt returns.

13.4 Function 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 stable status bit LSESTA=1, and then configure the LSE timing value {LSECRH[7:0], LSECRL[7 :0]}, finally set LSECNT=1, enable LSE counting, and turn on the LSE counting function. The LSE timer starts counting from 0. When the count value is equal to the timing value, the interrupt flag is set to 1, and the timing value is updated to the value in the timer data register (that is, the LSE timing value is the last time before the count value and the timing value are equal Write the value of {LSECRH[7:0], LSECRL[7:0]}). The minimum timer value of the timer is 1. If the timer value is set to 0, the timer defaults to 1 as the timer value. The formula for calculating the timing time of the LSE timer is as follows:

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

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

14. Wake Up Time（WUT）

14.1 Overview

Wake Up Timer is a 12-bit, up-counting timer used for wake-up from sleep and a clock source from the internal low-speed clock LSI. It can be used to wake up the system in sleep mode. Configure the timing wake-up time before the system enters sleep, and enable the timing wake-up function. When the chip enters the sleep mode, the WUT starts counting, and when the count value is equal to the set value, the chip enters the sleep wake-up waiting state.

14.2 Related Registers

14.2.1 WUTCRH Register

0xBD	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WUTCRH	WUTEN	TIMER_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: Timing wake-up function enable bit;
1= Timing wake-up function is turned on;
0= Timing wake-up function is turned off.
Bit6 TIMER_OV: Timer overflow status bit;
1= Counter overflow;
0= Cleared by software.
Bit5~Bit4 WUTPS<1:0>: Timing wake-up counter clock division bits;
00= F/1;
01= F/8;
10= F/32;
11= F/256.
Bit3~Bit0 WUTD<11:8>: The high 4 bits of the timing wake-up time data.

14.2.2 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 low 8 bits of timing wake-up time data.

14.3 Function Description

Principle of the internal wake-up timer is: after the system enters the sleep mode, the CPU and all peripheral circuits stop working, but the internal low-power oscillator LSI starts to work, and its oscillation clock is 125KHz (T_{LSI}≈8us). Provide clock for WUT counter.

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

Bit 7 of the WUTCRH register is the internal timing wake-up enable bit:

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

{WUTCRH[3:0] and WUTCRL[7:0]} form a 12-bit timing wake-up data register. After entering sleep mode, the WUT counter starts timing. When the value of the WUT counter equals the value of the timing wake-up data register, the system oscillator is started, and enter the wake-up waiting state.

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

15. Baud Rate Timer（BRT/BRT1）

15.1 Overview

There are two 16-bit baud rate timers BRT and BRT1 inside the chip, and these two timer functions are exactly the same, mainly providing the clock for the UART module.

15.2 Related Registers

15.2.1 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, must be 0.
Bit2~Bit0	BRTCKDIV<2:0>	BRT timer prescaler 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.

15.2.2 BRT Timer Data Load 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 low 8 bits;

15.2.3 BRT Timer Data Load 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 high 8 bits;

15.2.4 BRT1 Module Control Register BRT1CON

F5C4H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BRT1CON	BRT1EN	--	--	--	--	BRT1CKDIV2	BRT1CKDIV1	BRT1CKDIV0
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 BRT1EN: BRT1 timer enable bit;
1= Enable;
0= Disable.
Bit6~Bit3 -- Reserved, must be 0.
Bit2~Bit0 BRT1CKDIV<2:0> BRT1 timer prescaler 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.

15.2.5 BRT1 Timer Data Load Low 8-bit Register BRT1DL

F5C5H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BRT1DL	BRT1DL7	BRT1DL6	BRT1DL5	BRT1DL4	BRT1DL3	BRT1DL2	BRT1DL1	BRT1DL0
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 BRT1DL<7:0>: BRT1 timer load value low 8 bits;

15.2.6 BRT1 Timer Data Load High 8-bit Register BRT1DH

F5C6H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BRT1DH	BRT1DH7	BRT1DH6	BRT1DH5	BRT1DH4	BRT1DH3	BRT1DH2	BRT1DH1	BRT1DH0
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 BRT1DH<7:0>: BRT1 timer load value high 8 bits;

15.3 Function Description

There is a 16-bit up-counter inside BRT/BRT1. Its clock comes from the prescaler circuit. The prescaler clock is determined by the timer prescaler selection bit BRTCKDIV/BRT1CKDIV. The initial value of the counter is determined by {BRTDH, BRTDL}/{BRT1DH, BRT1DL} to load.

When the timer enables bit BRTEN=1/BRT1EN=1 is turned on, the counter starts to work. When the value of the 16-bit count er is equal to FFFFH, the BRT/BRT1 counter overflows. And after the overflow, the initial value of the count is automatically loaded into the counter, and then counts again.

The overflow signal of the BRT/BRT1 counter is specially provided to the UART module as the clock source of the baud rate. When the overflow occurs, no interrupt will be generated, and there is no corresponding interrupt structure. In the debug mode of BRT/BRT1, its clock will not stop. If the UART module has started to send or receive data, even if the chip enters the suspended state, the UART will complete the entire process of sending or receiving.

BRT timer overflow rate:

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

BRT1 timer overflow rate:

$$BRT1ov=\frac{Fsys}{(65536-\{BRT1DH,BRT1DL\})\times 2^{BRT1CKDIV}}$$

16. Cyclic Redundancy Check Unit（CRC）

16.1 Overview

To ensure safety during operation, the IEC61508 standard requires confirmation data even during CPU operation. This general-purpose CRC module can perform CRC operations as a peripheral function while the CPU is running. The general-purpose CRC module performs CRC checking by specifying the data to be confirmed by the program, and can be used for multi-purpose checking not limited to the code flash area.

CRC generator polynomial uses " $X^{16}+X^{12}+X^5+1$" of CRC16-CCITT.

16.2 Related Registers

16.2.1 CRC Data Input Register CRCIN

F708H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CRCIN	CRCIN7	CRCIN6	CRCIN5	CRCIN4	CRCIN3	CRCIN2	CRCIN1	CRCIN0
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 CRCIN<7:0> Input the 8-bit data that needs CRC transportation.

16.2.2 CRC Operation Result Low 8-bit Data Register CRCDL

F709H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CRCDL	CRCD7	CRCD6	CRCD5	CRCD4	CRCD3	CRCD2	CRCD1	CRCD0
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 CRCD<7:0> CRC operation result low 8-bit data

16.2.3 CRC Operation Result High 8-bit Data Register CRCDH

F70AH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CRCDH	CRCD15	CRCD14	CRCD13	CRCD12	CRCD11	CRCD10	CRCD9	CRCD8
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 CRCD<15:8> CRC operation result high 8-bit data

16.3 Function Description

After writing the CRCIN register, one system clock passes, and the CRC operation result is saved to the CRCDL/CRCDH register. If necessary, read the previous operation data before writing overwrite, otherwise it will be overwritten by the new operation result.

For example: send data "12345678H", write the value to the CRCIN register in the order of "12H", "34H", "56H", "78H", after writing, read from the CRCDL/CRCDH register as CRCDL=0xF0, CRCDH=0x67, that is, the result of the CRC operation on the bit sequence of the data "12345678H" is 0x67F0. The register operation is as follows:

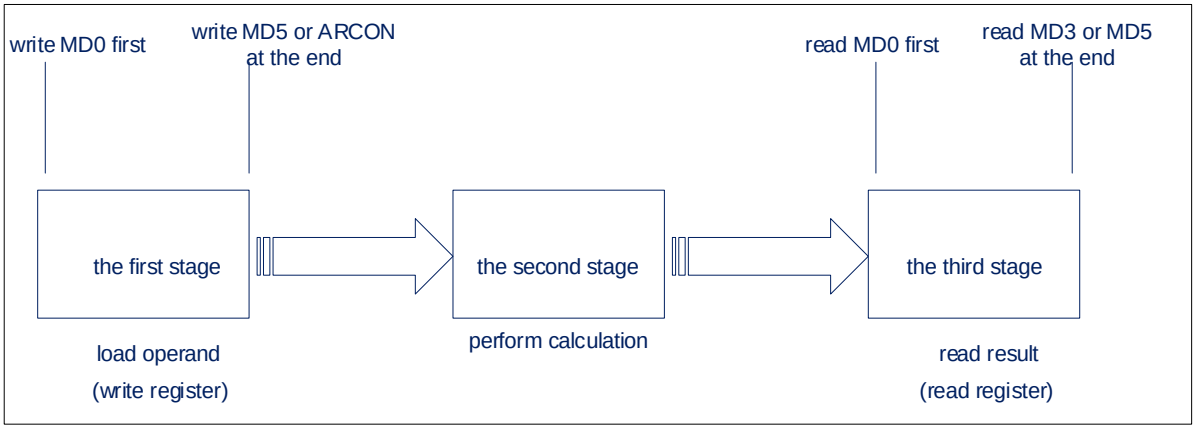
```
CRCIN=0x12;     // Send the first number
CRCIN=0x34;     // Send the second number
CRCIN=0x56;     // Send the third number
CRCIN=0x78;     // Send the fourth number
resl=CRCDL;     // Read the low 8 bits of the CRC operation result to the variable resl
resh=CRCDH;     // Read the high 8 bits of the CRC operation result to the variable resh
```

17. Multiplication And Division Unit（MDU）

17.1 Overview

The MDU (Multiplication/Division Unit) module provides 32bit/16bit division, 16bit/16bit division, 16bit*16bit multiplication, 32bit shift operation, 32bit normalization operation functions, all operations are unsigned integer operations, and the shift operation supports 32-bit data left or right shift operation. The operation of the MDU module is controlled by 7 registers (MD0/MD1/MD2/MD3/MD4/MD5/ARCON). MD0~MD5 are used to store the operand before the operation, the result and the remainder after the operation, and ARCON is the control register.

The operation of the MDU module can be divided into three stages. The first stage is to load the operand (write register), the second stage is to execute the operation, and the third stage is to read the result (read register). The first and third stages need to operate registers and depend on the CPU to run, while the second stage can run independently of the CPU. When writing the MD5 or ARCON register to start the MDU operation, after the MDU has passed a fixed time, read the register to get the operation result. The structural block diagram of the operation phase of the MDU module is as follows:



17.2 Related Registers

The operation of the MDU module is controlled by registers MD0, MD1, MD2, MD3, MD4, MD5, and ARCON. And each register is described as follows:

17.2.1 Operation Register MD0

0XE9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD0	MD07	MD06	MD05	MD04	MD03	MD02	MD01	MD00
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 MD0<7:0>: 32bit/16bit division operation: write bit7-bit0 as the dividend, read bit7-bit0 as the quotient;
16bit/16bit division operation: write bit7-bit0 as the dividend, read bit7-bit0 as the quotient;
16bit*16bit multiplication operation: write bit7-bit0 as the first multiplier, read bit7-bit0 as the product;
Shift operation: bit7-bit0 of write/read data;
Normalization operation: bit7-bit0 of write/read data.

17.2.2 Operation Register MD1

0XEA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD1	MD17	MD16	MD15	MD14	MD13	MD12	MD11	MD10
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 MD1<7:0>: 32bit/16bit division operation: write as bit15-bit8 of dividend, read as bit15-bit8 of quotient;
16bit/16bit division operation: write as bit15-bit8 of dividend, read as bit15-bit8 of quotient;
16bit*16bit multiplication operation: write bit15-bit8 of the first multiplier, read bit15-bit8 of the product;
Shift operation: bit15-bit8 of write/read data;
Normalization operation: bit15-bit8 of write/read data.

17.2.3 Operation Register MD2

0XEB	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD2	MD27	MD26	MD25	MD24	MD23	MD22	MD21	MD20
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 MD2<7:0>: 32bit/16bit division operation: write bit23-bit16 as dividend, read bit23-bit16 as quotient;
16bit*16bit multiplication operation: read bit23-bit16 of the product;
Shift operation: bit23-bit16 of write/read data;
Normalization operation: bit23-bit16 of write/read data.

17.2.4 Operation Register MD3

0XEC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD3	MD37	MD36	MD35	MD34	MD33	MD32	MD31	MD30
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 MD3<7:0>: 32bit/16bit division operation: write bit31-bit24 as dividend, read bit31-bit24 as quotient;
16bit*16bit multiplication operation: read bit31-bit24 of the product;
Shift operation: bit31-bit24 of write/read data;
Normalization operation: bit31-bit24 of write/read data.

17.2.5 Operation Register MD4

0XED	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD4	MD47	MD46	MD45	MD44	MD43	MD42	MD41	MD40
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 MD4<7:0>: 32bit/16bit division operation: write bit7-bit0 of the divisor, read bit7-bit0 of the remainder;
16bit/16bit division operation: write bit7-bit0 of the divisor, read bit7-bit0 of the remainder;
16bit*16bit multiplication operation: write as bit7-bit0 of the second multiplier.

17.2.6 Operation Register MD5

0XEE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD5	MD57	MD56	MD55	MD54	MD53	MD52	MD51	MD50
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 MD5<7:0>: 32bit/16bit division operation: write bit15-bit8 of the divisor, read bit15-bit8 of the remainder;
16bit/16bit division operation: write bit15-bit8 of the divisor, read bit15-bit8 of the remainder;
16bit*16bit multiplication operation: write as bit15-bit8 of the second multiplier.

17.2.7 Operation Register ARCON

0XEF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ARCON	MDEF	MDOV	SLR	SC4	SC3	SC2	SC1	SC0
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 MDEF: Error flag bit (this bit set to 1 by hardware)

1= Set to 1 when the MD0~MD5 register is written during the data loading process of the MDU module;

0= Read ARCON register, this bit is cleared to 0.

Bit6 MDOV: Overflow flag bit (this bit set to 1 by hardware)

1= Divide by 0;

Or the multiplier operation result is greater than 0xffff;

Or normalization operation MD37=1;

0= This bit is 0 when the condition is not matched to 1 (Clear this bit by writing MD0 register, software writing 0 is invalid).

Bit5 SLR: Shift operation direction control bit

1= Right shift;

0= Left shift.

Bit4~Bit0 SC<4:0>: Shift number of digits/normalization number of digits

Pre-write 00000 to start the normalization operation function, and store the normalized number of digits after the normalization operation is completed;

Pre-write ≠ 00000 (storing shift number of digits) starts the shift operation function.

17.3 Function Description

The division and multiplication operation types of the MDU module are determined by the order of writing MD0~MD5, and the operation types of the shift and normalization functions are controlled by the register ARCON. The multiplication and division operation sequence of the MDU module is shown in the following table:

Operation stage	Sequence of operations	32bit/16bit	16bit/16bit	16bit*16bit
first stage	first write	MD0 (bit7-bit0 of the dividend)	MD0 (bit7-bit0 of the dividend)	MD0 (bit7-bit0 of the multiplier 0)
		MD1 (bit15-bit8 of the dividend)	MD1 (bit15-bit8 of the dividend)	MD4 (bit7-bit0 of the multiplier 1)
		MD2 (bit23-bit16 of the dividend)	--	--
		MD3 (bit31-bit24 of the dividend)	--	--
		MD4 (bit7-bit0 of the divisor)	MD4 (bit7-bit0 of the divisor)	MD1 (bit15-bit8 of the multiplier 0)
	last write	MD5 (bit15-bit8 of the divisor)	MD5 (bit15-bit8 of the divisor)	MD5 (bit15-bit8 of the multiplier 1)
third stage	first read	MD0 (bit7-bit0 of the quotient)	MD0 (bit7-bit0 of the quotient)	MD0 (bit7-bit0 of the product)
		MD1 (bit15-bit8 of the quotient)	MD1 (bit15-bit8 of the quotient)	MD1 (bit15-bit8 of the product)
		MD2 (bit23-bit16 of the quotient)	--	--
		MD3 (bit31-bit24 of the quotient)	--	--
		MD4 (bit7-bit0 of the remainder)	MD4 (bit7-bit0 of the remainder)	MD2 (bit23-bit16 of the product)
	last read	MD5 (bit15-bit8 of the remainder)	MD5 (bit15-bit8 of the remainder)	MD3 (bit31-bit24 of the product)

All operation process of the MDU is completed by hardware, and the conversion rate is fast. It can save a lot of time for program run. The operation time of the second stage of the five operations of the MDU module is shown in the following table:

Function	The number of digits of the operation result	The number of digits of the remainder	Operation time (second stage)
32bit/16bit division	32bit	16bit	16 Tsys
16bit/16bit division	16 bit	16bit	8 Tsys
16bit*16bit multiplication	32bit	--	8 Tsys
32bit shift operation	32bit	--	2~17 Tsys
32bit normalization operation	32bit	--	1~17 Tsys

The Tsys in the above the table is the clock cycle of the MDU module. The calculation time is the operation time of the MDU module, and excluding the time of write and read registers. The operation time of shift operation and normalization operation is depending on the number of digits of the shift and operands. The min time of shift operation is 2 Tsys, and the max is 17 Tsys. The min time of normalization operation is 1 Tsys, and the max is 17 Tsys.

17.3.1 32bit/16bit Division Operation

32bit/16bit divider operation steps are as follows:

1) write the register MD0 (bit7-bit0 of the dividend);
2) write the register MD1 (bit15-bit8 of the dividend);
3) write the register MD2 (bit23-bit16 of the dividend);
4) write the register MD3 (bit31-bit24 of the dividend);
5) write the register MD4 (bit7-bit0 of the divisor);
6) write the register MD5 (bit15-bit8 of the divisor), starts the division operation when the writing is completed;
7) Waiting for 16 clock cycle of the MDU module to ensure the calculation is completed;
8) read the register MD0 (bit7-bit0 of the quotient);
9) read the register MD1 (bit15-bit8 of the quotient);
10) read the register MD2 (bit23-bit16 of the quotient);
11) read the register MD3 (bit31-bit24 of the quotient);
12) read the register MD4 (bit7-bit0 of the remainder);
13) read the register MD5 (bit15-bit8 of the remainder), and read to complete a division operation.

After the operation is completed, if the operation result is not read, the MD0 can be rewritten to start the next operation.

For example, when the dividend is 0X87654321 and the divisor is 0X1234, write the MD0=0X21, MD1=0X43, MD2=0X65, MD3=0X87, MD4=0X34, MD5=0X12, and after the MDU operation is completed, read the result is: MD0=0X23, MD1 =0X70, MD2=0X07, MD3=0X00, MD4=0X05, MD5=0X06.

17.3.2 16bit/16bit Division Operation

16bit/16bit divider operation steps are as follows:

1) write the register MD0 (bit7-bit0 of the dividend);
2) write the register MD1 (bit15-bit8 of the dividend);
3) write the register MD4 (bit7-bit0 of the divisor);
4) write the register MD5 (bit15-bit8 of the divisor), starts the division operation when the writing is completed;
5) Waiting for 8 clock cycle of the MDU module;
6) read the register MD0 (bit7-bit0 of the quotient);
7) read the register MD1 (bit15-bit8 of the quotient);
8) read the register MD4 (bit7-bit0 of the remainder);
9) read the register MD5 (bit15-bit8 of the remainder), and read to complete a division operation.

After the operation is completed, if the operation result is not read, the MD0 can be rewritten to start the next operation.

For example, when the dividend is 0X4321, and the divisor is 0X1234, write the MD0=0X21, MD1=0X43, MD4=0X34, MD5=0X12, and after the operation of the MDU is completed, read the result is: MD0=0X03, MD1=0X00, MD4=0X85, MD5=0X0C.

17.3.3 16bit*16bit Multiplication Operation

16bit*16bit multiplier operation steps are as follows:

1) write the register MD0 (bit7-bit0 of the first multiplier);
2) write the register MD4 (bit7-bit0 of the second multiplier);
3) write the register MD1 (bit15-bit8 of the first multiplier);
4) write the register MD5 (bit15-bit8 of the second multiplier), starts the multiplier operation when the writing is completed;
5) waiting for 8 clock cycle of the MDU module;
6) read the register MD0 (bit7-bit0 of the product);
7) read the register MD1 (bit15-bit8 of the product);
8) read the register MD2 (bit23-bit16 of the product);
9) read the register MD3 (bit31-bit24 of the product), and read to complete a multiplication operation.

After the operation is completed, if the operation result is not read, the MD0 can be rewritten to start the next operation.

For example, when the first multiplier is 0X8765, and the second multiplier is 0X1234, write the MD0=0X65, MD4=0X34, MD1=0X87, MD5=0X12, and after the operation of the MDU is completed, read the result is: MD0=0X84, MD1=0X9A, MD2=0XA0, MD3=0X09.

17.3.4 32bit Shift Operation

32-bit shift operation steps are as follows:

1) write the register MD0 (bit7-bit0 of the operand);
2) write the register MD1 (bit15-bit8 of the operand);
3) write the register MD2 (bit23-bit16 of the operand);
4) write the register MD3 (bit31-bit24 of the operand);
5) write the register ARCON, starts the shift operation when the writing is completed;
6) waiting for 17 clock cycle of the MDU module to ensure the calculation is completed;
7) read the register MD0 (bit7-bit0 of the shift result);
8) read the register MD1 (bit15-bit8 of the shift result);
9) read the register MD2 (bit23-bit16 of the shift result);
10) read the register MD3 (bit31-bit24 of the shift result), and read to complete a multiplication operation.

After the operation is completed, if the operation result is not read, the MD0 can be rewritten to start the next operation.

For example, when the operand is 0X12345678, and shift right 5-bit, write the MD0=0X78, MD1=0X56, MD2=0X34, MD3=0X12, ARCON=0X25, and after the operation of the MDU is completed, read the result is: MD0=0XB3, MD1=0XA2, MD2=0X91, MD3=0X00.

17.3.5 32bit Normalization Operation

The normalization operation is to shift the operand to the left until the highest bit is 1 to end the shift. The 32-bit normalization operation steps are as follows:

1) write the register MD0 (bit7-bit0 of the operand);
2) write the register MD1 (bit15-bit8 of the operand);
3) write the register MD2 (bit23-bit16 of the operand);
4) write the register MD3 (bit31-bit24 of the operand);
5) write the register ARCON=0x00, starts the normalization operation when the writing is completed;
6) waiting for 17 clock cycle of the MDU module to ensure the calculation is completed;
7) read the register MD0 (bit7-bit0 of the normalized result);
8) read the register MD1 (bit15-bit8 of the normalized result);
9) read the register MD2 (bit23-bit16 of the normalized result);
10) read the register MD3 (bit31-bit24 of the normalized result), and read to complete a normalization operation.

After the operation is completed, if the operation result is not read, the MD0 can be rewritten to start the next operation.

For example, when the operand is 0X12345678, write the MD0=0X78, MD1=0X56, MD2=0X34, MD3=0X12, ARCON=0X00, and after the operation of the MDU is completed, read the result is: MD0=0XC0, MD1=0XB3, MD2=0XA2, MD3=0X91, ARCON=0X03.

18. Buzzer Driver（BUZZER）

18.1 Overview

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

18.2 Related Registers

18.2.1 BUZZER Control Register BUZCON

0xBF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BUZCON	BUZEN	--	--	--	--	--	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	BUZEN:	BUZZER enable bit;
	1=	Enable;
	0=	Disable.
Bit6~Bit2	--	Reserved, must be 0.
Bit1~Bit0	BUZCKS<1:0>:	BUZZER frequency division ratio selection bits;;
	00=	Fsys/8;
	01=	Fsys/16;
	10=	Fsys/32;
	11=	Fsys/64.

18.2.2 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 selection bit;
	0x00=	No square wave output;
	others=	Fbuz =Fsys/(2*CLKDIV*BUZCKS).

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

18.3 Function Description

When using the buzzer, you need to configure the corresponding port as a buzzer drive output port first. For example, to configure P24 as a buzzer drive output port, the configuration is as follows:

```
P24CFG = 0x18;          // P24 is configured as a buzzer drive output port
```

By configuring the corresponding registers of the buzzer driver module, you can set the buzzer driver output port to output differently. Frequency of. For example:

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

   output frequency of the buzzer driver is: Fbuz=8MHz/(2*125)/16= 2KHz

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

   output frequency of the buzzer driver is: Fbuz=16MHz/(2*125)/64= 1KHz

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

   output frequency of the buzzer driver is: Fbuz=24MHz/(2*94)/64= 2KHz

Choose different system clock frequency and buzzer driver clock division ratio can obtain the different output frequency. The buzzer driver 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

19. PWM Module

19.1 Overview

19.1.1 Function

The enhanced PWM module supports 6-channel PWM generators, which can be configured as independent 6-channel PWM outputs (PG0-PG5),，or 3 pairs of complementary with dead-zone programming generators. PWM output,（PG0-PG1，PG2-PG3，PG4-PG5）.

Each pair of PWM shares an 8-bit prescaler, and there are 6 groups of clock dividers, providing 5 kinds of frequency division coefficients (1, 1/2, 1/4, 1/8, 1/16). Each PWM output is controlled by an independent 16-bit counter, and another 16-bit comparator is used to trim the duty cycle. The 6-channel PWM generator provides 25 interrupt flags. The cycle or duty ratio of the relevant PWM channel is consistent with the counter, and the interrupt flag will be generated. Each PWM channel has a separate enable bit.

Each PWM can be set to output in one-shot mode (generating a PWM signal cycle) or loop mode (continuous output of PWM waveform)

19.1.2 Characteristics

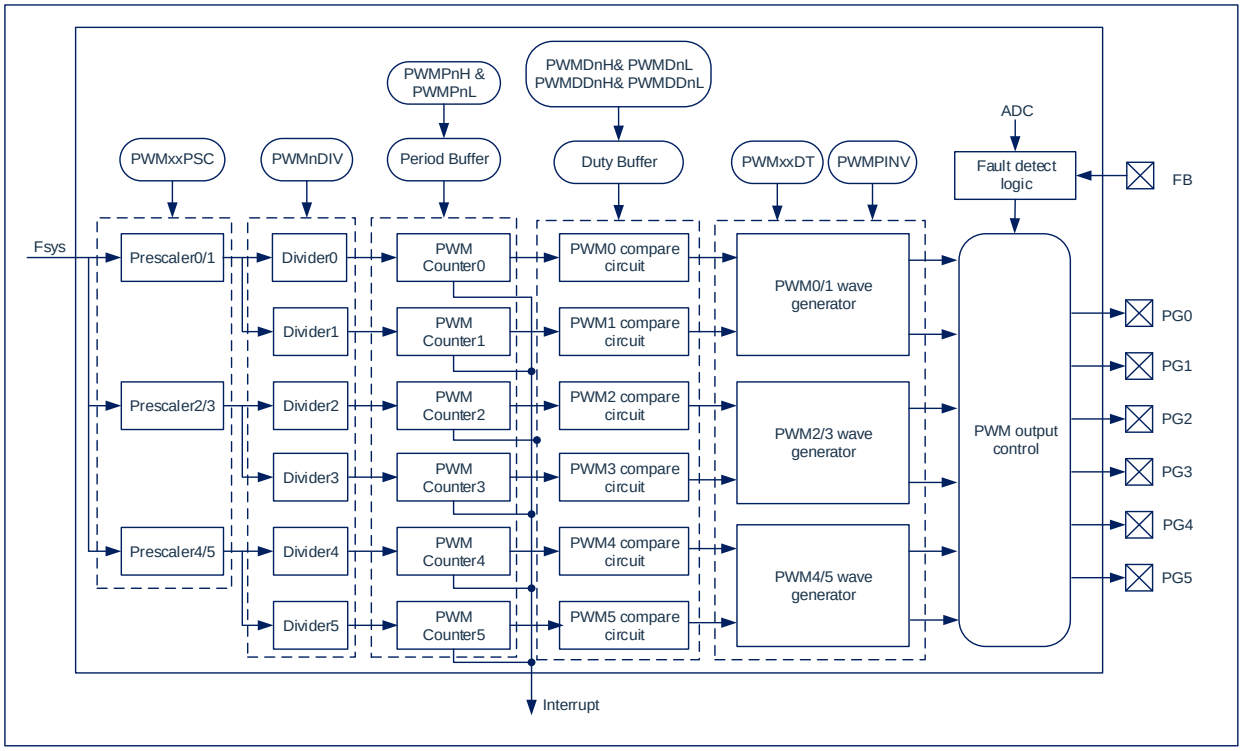
Enhanced PWM module has the following characteristics:

- 6 independent 16-bit PWM control mode
  - 6 independent outputs: PG0、PG1、PG2、PG3、PG4、PG5;
  - 3 groups of complementary PWM pair outputs: (PG0-PG1), (PG2-PG3), (PG4-PG5), programmable dead time can be inserted
  - 3 groups of synchronous PWM pair output: (PG0-PG1), (PG2-PG3), (PG4-PG5), each group of PWM pair pins are synchronized
- Support group control, PG0, PG2, PG4 output synchronization, PG1, PG3, PG5 output synchronization
- Support one-shot mode or auto-load mode
- Support two modes of edge alignment and center alignment
- The center alignment mode supports symmetric counting and asymmetric counting
- Support Programmable Dead-Time Generator, in the complementary PWM mode.
- Each PWM has independent polarity control
- Hardware brake protection and recovery function (external FB trigger, software trigger, ADC comparison event trigger)
- The PWM edge or cycle can trigger the start of AD conversion.

19.2 Configuration

19.2.1 Functional Block Diagram

The enhanced PWM function block diagram is shown in the following figure：



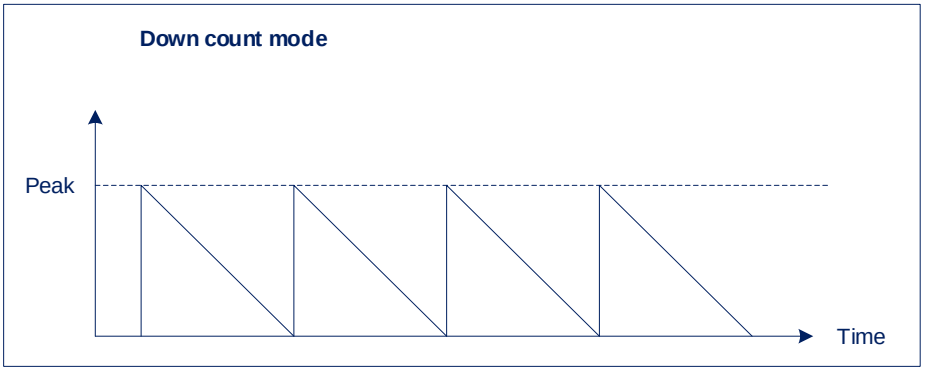
19.2.2 Description Of Each Functional Module

The enhanced PWM module is composed of PWM counter module, output comparison unit, waveform generator, fault detection and output controller.

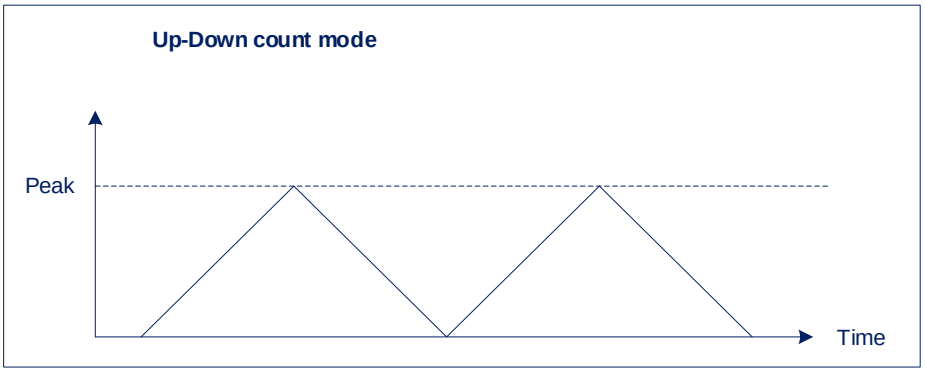
PWM counter:

The system clock is input to the enhanced PWM module, and the system clock is divided by the prescaler and the clock divider to obtain the count clock of 6 PWM counters; the 16-bit control register composed of the period register (PWMPnH, PWMPnL) is used to set Counting period of 6 PWM counters. In order to prevent arbitrarily modifying the period setting of the PWM during the operation of the PWM, a buffer register (Period Buffer) is used. If the PWM is set to continuous operation mode (PWMnCNTM=1), the value of the period register will be automatically loaded into the buffer register (Period Buffer) at the zero point of each PWM.

PWM counter has two counting modes: Down count mode and UP-Down count mode. The count down mode is shown in the following figure:



The UP-Down count mode is shown in the figure below：



OCU：

The output compare unit (OCU) is composed of duty cycle registers (PWMDnH, PWMDnL), which are used to set the PWM duty cycle of 6 channels. Similarly, in order to prevent the duty cycle setting of the PWM from being arbitrarily modified during the operation of the PWM, a buffer register (Duty Buffer) is used to compare with the PWM counter to invert the output level. If the PWM is set to continuous operation mode (PWMnCNTM=1), the value of the duty cycle register will be automatically loaded into the buffer register (Duty Buffer) at the zero point of each PWM.

WFG：

The waveform generator is composed of a dead zone control unit and an output polarity control unit. For the complementary output with dead time, PWM01DT/ PWM23DT/ PWM45DT are used to set the dead time of PWM; in combination with the polarity control register (PWMPINV) to control the output polarity of PWM.

Fault detection (brake function)：

The fault detection module is embedded in the enhanced PWM circuit and is configured as input fault detection to protect the system from damage to the device. Once a valid fault signal input is detected, the output of the PWM is forcibly turned off. In order to adapt to different driving requirements, the level of shutdown can be configured through the PWM brake data register: PWMFBKD.

mask output：

For special applications like square wave motor control, mask output is particularly important. Each channel of PWM has separate mask control bits and mask data bits, which are set by mask control register PWMMASKE and mask data register PWMMASKD.

When mask output disables PWMnMASKE=0, PWMn outputs normal PWM waveform;

When the mask output enables PWMnMASKE=1, PWMn outputs the data of the mask register PWMnMASKD.

output controller：

The output controller is used to control the output state of the PWM. The PWM output enable control register PWMOE is used to set the output enable of each channel. When a fault occurs and the PWM needs to be forcibly turned off, the MCU can output the corresponding level according to the settings in the brake data register PWMFBKD to meet the needs of different peripherals.

19.2.3 Description Of Related IO Ports

Before using the enhanced PWM module, you need to configure the corresponding port as a PWM channel. The PWM channel is marked with PG0~PG5 in the multiplexing function assignment table, corresponding to PWM channels 0~5. It can be seen that different PWM channels can correspond to the same port, and the same PWM channel can be assigned to different ports. This characteristic enables the enhanced PWM function to adapt to different types of packages and flexible PCB layout requirements.

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

```
P13CFG=0x12;// Configure P13 to PG0 channel
P14CFG=0x13;// Configure P14 to PG1 channel
P15CFG=0x14;// Configure P15 to PG2 channel
P16CFG=0x15;// Configure P16 to PG3 channel
P17CFG=0x16;// Configure P17 to PG4 channel
P22CFG=0x17;// Configure P22 to PG5 channel
```

19.3 Enhanced PWM Operation

19.3.1 Load Update Mode

There are two counter loading modes: one-shot mode and auto load mode. In one-shot mode, the period and duty cycle related data is loaded once at the beginning of the counter. In auto load mode, the period and duty cycle related data are automatically loaded at zero during the PWM period.

Due to the double-buffer structure of PWM, in the process of PWM operation, change the related running registers: PWMPnL/PWMPnH/ PWMDnL/PWMDnH/PWMDDnL/PWMDDnH/ The value of the PWM output waveform will not change immediately, when the value of these registers at zero point will be loaded into the corresponding cache. Such a structure will not immediately change the output waveform of the current PWM cycle after changing the cycle duty ratio data, and will only make corresponding changes in the PWM waveform in the next cycle. That is, any change in PWM related data will not affect the current one full PWM cycle.

In high-speed applications, it is possible that the load point has arrived, but the operation of writing to the run register is not completed. At this time, it is not expected that part of the running data has been loaded, and another part of the running data has not been loaded. For this high-speed application, the PWM module provides a load enable bit.

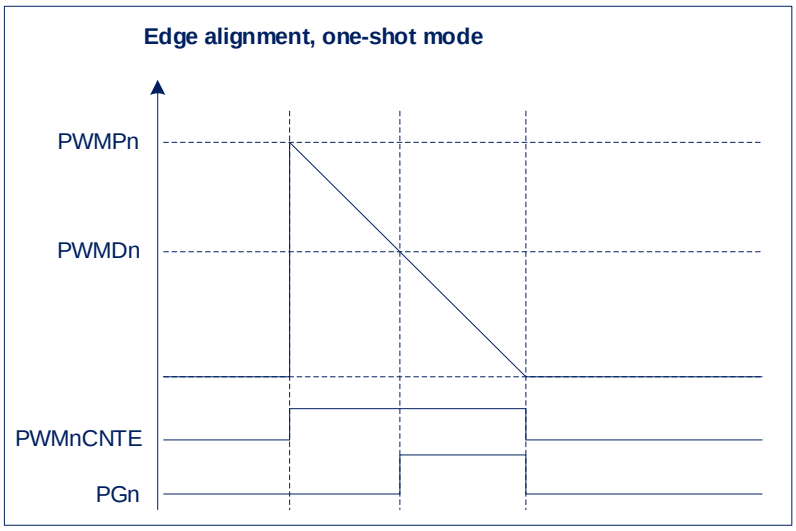
After changing the related running register, the enable bit PWMnLE of the load register PWMLOADEN needs to be set to 1, and the PWMnLE bit is automatically cleared after the period and duty cycle are loaded. That is, it can be judged whether to load the value of the related register into the actual circuit by reading this bit. If PWMnLE=0, it means that it has been loaded, which will affect the PWM waveform being output; if PWMnLE=1, it means that it has not been loaded, the current PWM waveform has not changed, and the value of the register that will be changed before the next loading point is loaded . If you change the value of the related running register again, you also need to set PWMnLE=1 to 1 again.

Note: When PWMnLE=1, changes to the contents of the period and duty cycle registers may cause unpredictable results. It is recommended to change the contents of the period and duty cycle registers first, then set the load enable bit PWMnLE to 1, and finally wait for the load to complete (PWMnLE=0).

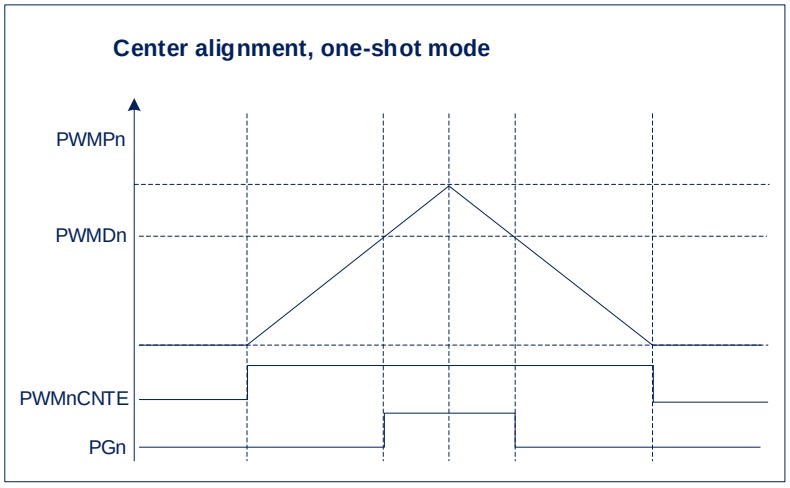
19.3.2 One-shot Count Mode

The one-shot count mode is a mode in which the PWM counter only works for one PWM cycle, and then the PWM counter stops running. When the one-shot mode is completed, the PWM count enable control bit is cleared by hardware (PWMnCNTE=0). If the one-shot mode is enabled again, the PWM count enable control bit must be enabled (PWMnCNTE=1). One-shot count mode can be selected by PWM counter mode control register PWMCNTM.

In edge alignment, the timing diagram of one-shot mode is shown in the following figure：



In center alignment, the timing diagram of single operation mode is shown in the following figure：



19.3.3 Edge Alignment Mode

In the edge alignment mode, the PWM counter adopts the down count mode: the initial value of the 16-bit PWM counter CNTn is PWMPn, which starts to count down until the count value becomes 0. At this time, the MCU automatically changes the value of the period register. Load into CNTn and start counting for the next PWM cycle.

When the value of CNTn is equal to the value of the duty cycle register PWMDn, PGn outputs a high level; CNTn continues to count down to 0, and PGn will output a low level at this time (when the PWM is selected as the inverting output, the output level is just the opposite of the above description).

The related parameters of edge alignment are as follows：

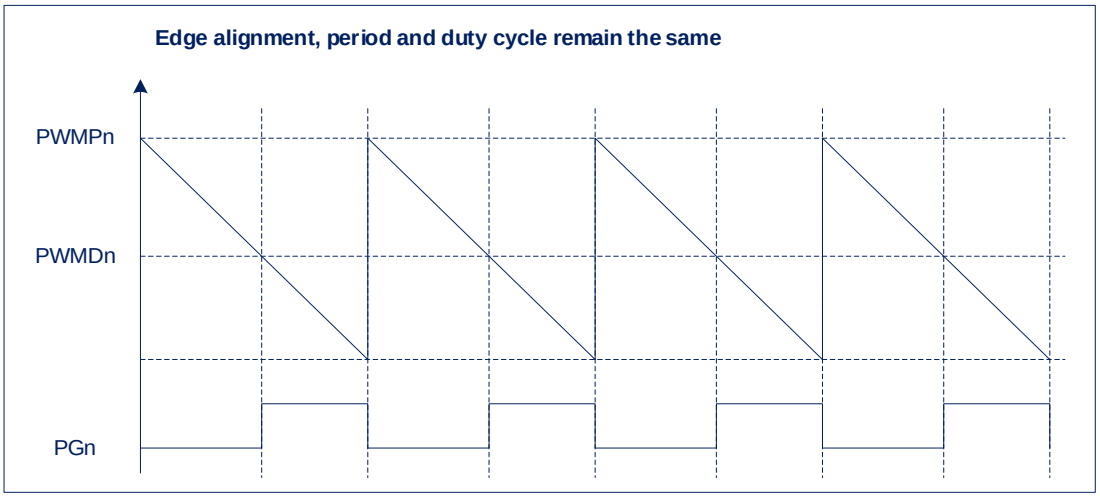
$$\text{High level time} = (\text{PWMDn}+1) \times \text{Tpwm}$$

$$\text{Period} = (\text{PWMPn}+1) \times \text{Tpwm}$$

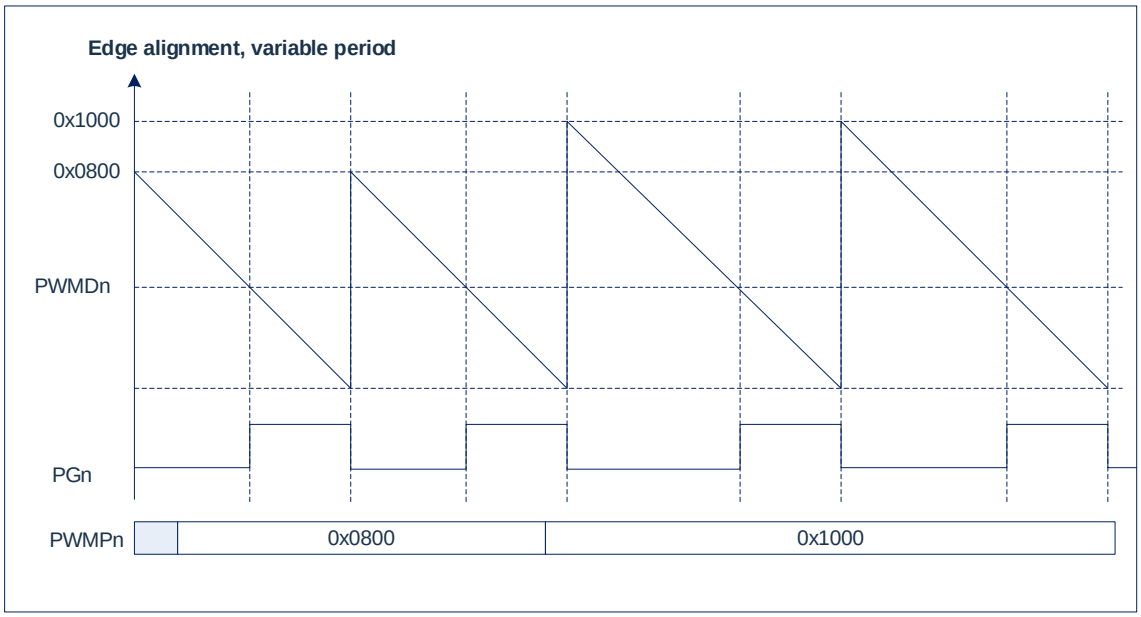
$$\text{Duty cycle} = \frac{\text{PWMDn}+1}{\text{PWMPn}+1}$$

When PWMDn=0，The duty cycle is 0%.

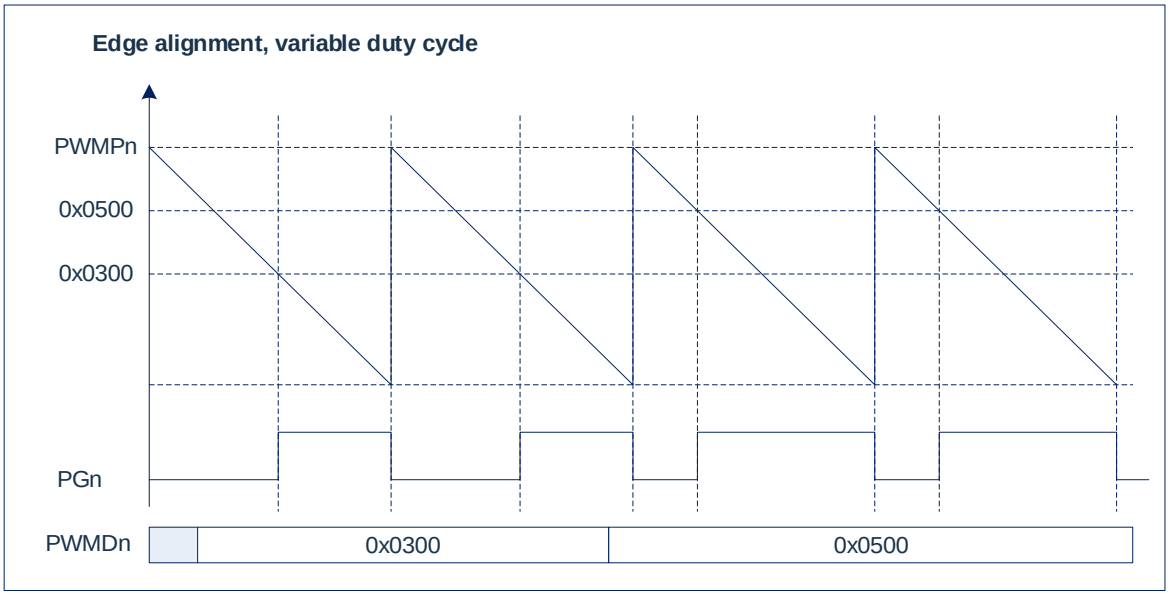
The timing diagram of edge alignment, constant period and duty cycle is shown in the following figure：



The edge alignment, cycle change timing diagram is shown in the following figure：



Edge alignment, duty cycle change timing diagram is shown in the figure below：



19.3.4 Center Alignment Mode

19.3.4.1 Symmetric Counting

In the center alignment symmetrical counting mode, the PWM counter adopts the up-down count mode. The 16-bit PWM counter CNTn starts to count up from 0. When CNTn = PWMPn, it automatically starts to count down until 0, and the subsequent PWM cycle repeats such a counting operation.

On the up-counting edge, when the value of CNTn is equal to the value of the duty cycle register PWMDn, the level of PGn flips and becomes a high level; At the down-counting edge, when the value of CNTn is equal to the value of the duty cycle register PWMDn, the output level of PGn is flipped and becomes a low level (When PWM is selected as inverting output, the output level is just opposite to the above description).

The related parameters in symmetric counting are as follows：

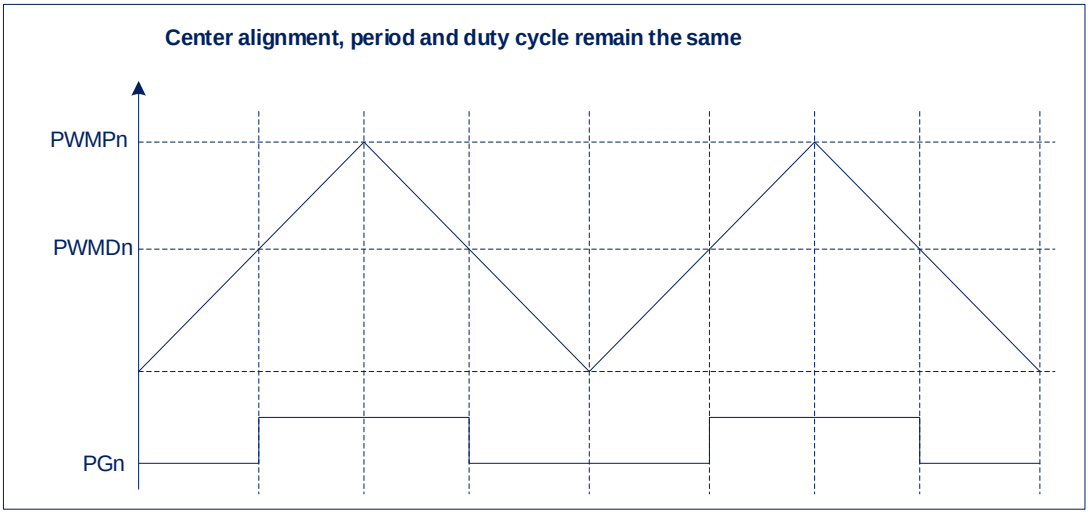
$$\text{High level time} = (\text{PWMPn}\times 2 - \text{PWMDn}\times 2 - 1)\times \text{Tpwm}$$

$$\text{Period} = (\text{PWMPn})\times 2\times \text{Tpwm}$$

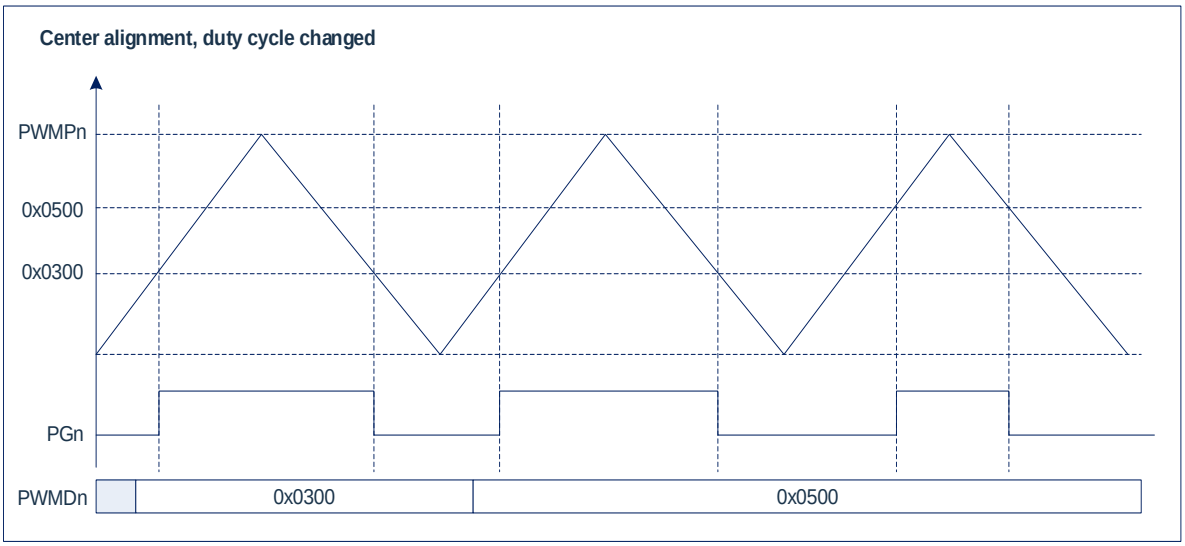
$$\text{Duty ratio} = \frac{\text{PWMPn}\times 2 - \text{PWMDn}\times 2 - 1}{\text{PWMPn}\times 2}$$

When PWMDn=0，100% duty cycle;

The timing diagram of center alignment symmetrical counting with constant period and duty cycle is shown in the following figure：



Center alignment symmetrical counting, the timing diagram of duty cycle change is shown in the figure below：



19.3.4.1 Asymmetric Counting

Center alignment asymmetric PWM mode is a very important feature in motor control, and the PWM counter still works in Up-Down count mode..

In this mode, there are two compare registers: PWMDn, PWMDDn. The 16-bit PWM counter CNTn starts counting up from 0. When CNTn = PWMDn, the output level of PGn is flipped from low level to high level. After that, CNTn continues to count up to PWMPn, and then CNTn starts to count down. During the counting process, when CNTn=PWMDDn, PGn turns to low level, and then continues to count down to 0. Turning on the asymmetric PWM mode requires setting the control bit ASYMEN to 1.

The related parameters in asymmetric mode are as follows：

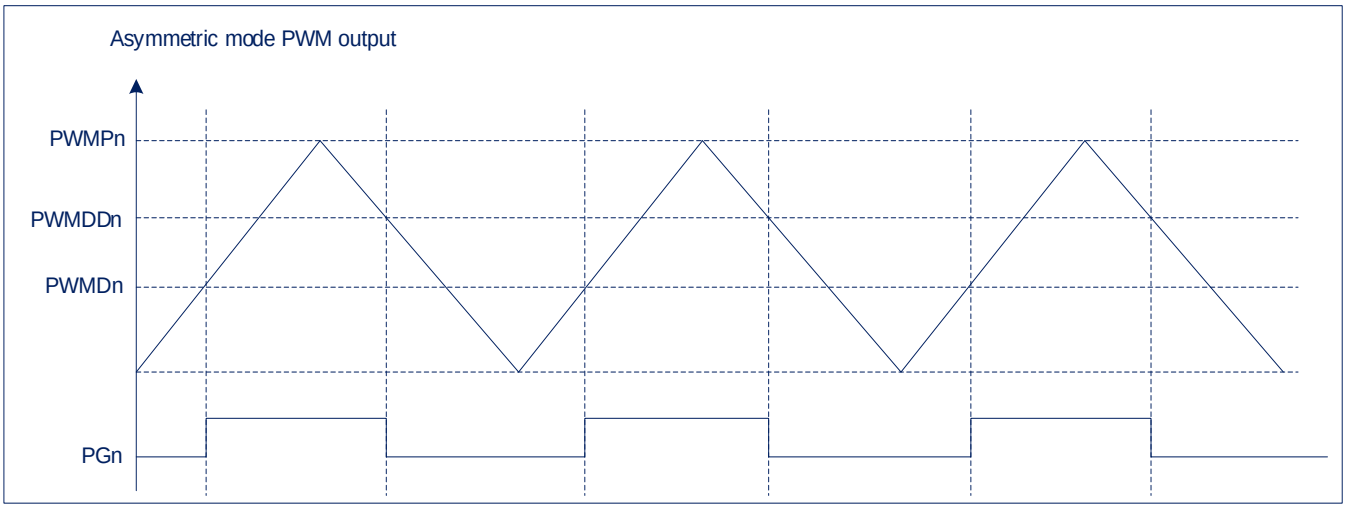
$$\text{High level time} = （PWMPn\times2\text{-}PWMDn\text{-}PWMDDn\text{-}1）\times Tpwm$$

$$\text{Period} = （PWMPn）\times2\times Tpwm$$

$$\text{Duty cycle} = \frac{PWMPn\times2\text{-}PWMDn\text{-}PWMDDn\text{-}1}{PWMPn\times2}$$

When the PWMDn=0 and PWMDDn=0， the duty cycle is 100%

The timing diagram for center alignment asymmetric mode is shown below：

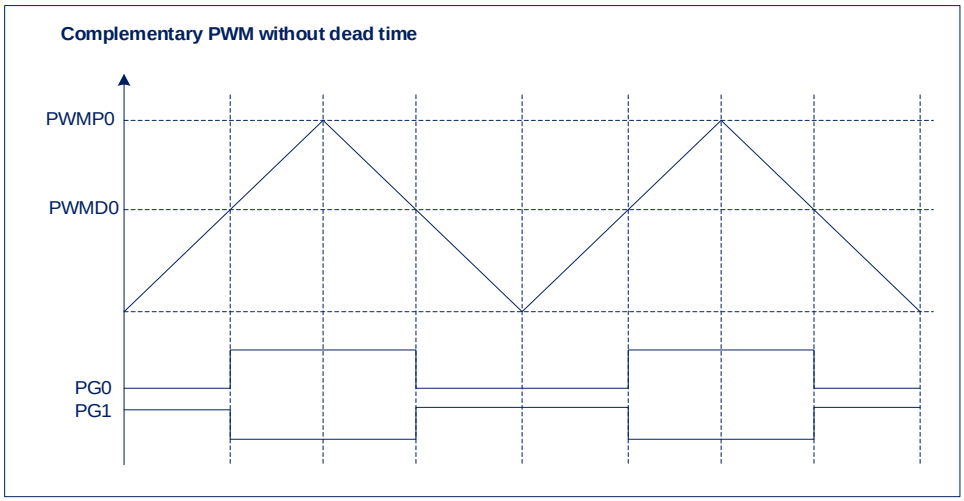


19.3.5 Complementary Mode With Dead Time

In practical motor control applications, the PWM signal used to drive the inverter bridge needs to have a complementary output mode, that is, the drive signal of the upper bridge arm is exactly opposite to the drive signal of the lower bridge arm.

In the enhanced PWM module, the 6-channel PWM can be set to 3 pairs of complementary signals: PWM0 and PWM1, PWM2 and PWM3, PWM4 and PWM5, The period and duty ratio of PWM1, PWM3 and PWM5 are determined by the related registers of PWM0, PWM2 and PWM4 respectively.

The timing diagram of complementary mode without dead time is shown in the following figure：



In motor control applications, the ideal PWM signal is the level flip at the same time. Due to the delay in the turn-on and turn-off of the MOS tube, it is easy to cause the power supply to be connected to the ground, thereby damaging the power tube. In order to avoid this phenomenon, PWM with dead time is particularly important. In complementary mode, each complementary PWM pair supports insertion of dead time, and the inserted dead time is as follows：

PWM0/1 dead-time：（PWM01DT+1）*T_{PWM0}

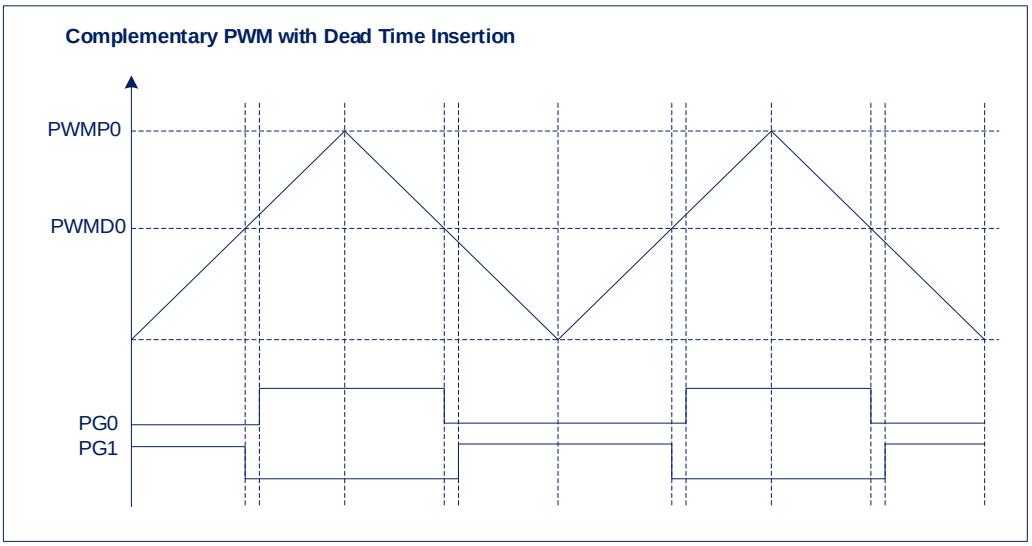
PWM2/3 dead-time：（PWM23DT+1）*T_{PWM2}

PWM4/5 dead-time：（PWM45DT+1）*T_{PWM4}

T_{PWM0}/T_{PWM2}/$T_{PWM4}$are the clock source periods of PG0/PG2/PG4 respectively.

Note: Complementary mode is supported for both center alignment and edge alignment.

The complementary PWM waveform with dead time inserted is shown in the figure below：



19.3.6 Brake Function

The signal sources that can trigger PWM braking are as follows：

- Software trigger
- ADC comparison result output
- External trigger port FB (high/low level trigger)

The software brake can be configured through the brake control register PWMFBCK, the external trigger port FB triggers the PWM brake enable, the trigger type (high level or low level trigger), and the ADC comparator result can be configured through the ADC comparator control register ADCMPC to control the PWM brake enable.

PWM brake (fault protection) related flag bits：

- Fault flag bit PWMFBF (cleared by software)
  After detecting a valid brake trigger source signal, the fault interrupt flag PWMFBF is set to 1 and needs to be cleared by software.
- Fault signal flag bit BRKAF (read only)
  The fault signal flag bit BRKAF is set to 1, and BRKAF is automatically cleared to 0 after the brake signal is cancelled. BRKAF is a read-only bit.
- The fault-protected output status flag BRKOSF (read only)
  BRKOSF=1, indicating that the PWMn channel outputs the PWMFBKD data state;
  BRKOSF=0, indicating that PWMn is in normal output state.

Indicates whether the PWM output is in the braking state or the normal state. BRKOSF will be set to 1 when a valid brake signal is detected. In software recovery mode, executing the brake state clearing operation (BRKCLR=1) will affect the state of this bit.

PWM brake recovery mode can be divided into 4 kinds to meet the needs of different fault protection occasions. The recovery conditions of the 4 brake recovery modes are described in the following table:

brake recovery mode	register PWMBRKC[1:0] setting mode	counter status of braking	recovery conditions				recovery point
			cancel the brake signal	clear brake status	counter enable	delay	
stop mode	00	stop	need	need	need	unnecessary	restart
pause mode	01	keep counting	need	need	unnecessary	unnecessary	After clearing the brake state, the latest loading point
recovery mode	10	keep counting	need	unnecessary	unnecessary	unnecessary	last load point
delayed recovery	11	keep counting	need	unnecessary	unnecessary	need	After the delay time expires, the latest loading point

Note: After the brake protection is generated, the PWMn channel outputs the data in PWMFBKD (each channel can independently set the output high/low level).

stop mode：The fault protection and fault interrupt flags are generated, the PWMCNTE bit is cleared, and the counter operation is stopped. To restore the output, the brake signal needs to be removed, and the fault state clearing operation (PWMBRKC[3]=1) is performed, and then the PWMCNTE bit is set to 1 again.

Pause mode：The failsafe and fault interrupt flags are generated, but the counter continues to run. To restore the output, the brake signal needs to be cancelled, and after the fault state clearing operation (PWMBRKC[3]=1) is executed, the normal output will be restored at the latest loading update point.

recovery mode：The failsafe and fault interrupt flags are generated, but the counter continues to run. After the brake signal is canceled, it will automatically resume normal output at the latest loading update point. No fault state clearing operation is required.

Delayed recovery mode：The failsafe and fault interrupt flags are generated, but the counter continues to run. After the brake signal is cancelled, the PWM will return to normal output at the latest loading update point after a delay for a period of time. No fault state clearing operation is required.

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

$$Tdelay = BRKRDT[9:0]*T_{CLK}(TCLK\ is\ the\ system\ clock\ cycle)$$

It is necessary to pay attention to distinguish 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 canceled before the output can be restored; If it is a pulse signal, the PWM output will resume the output after the latest load update point after triggering the brake, unless the brake pulse signal is generated again during the period.

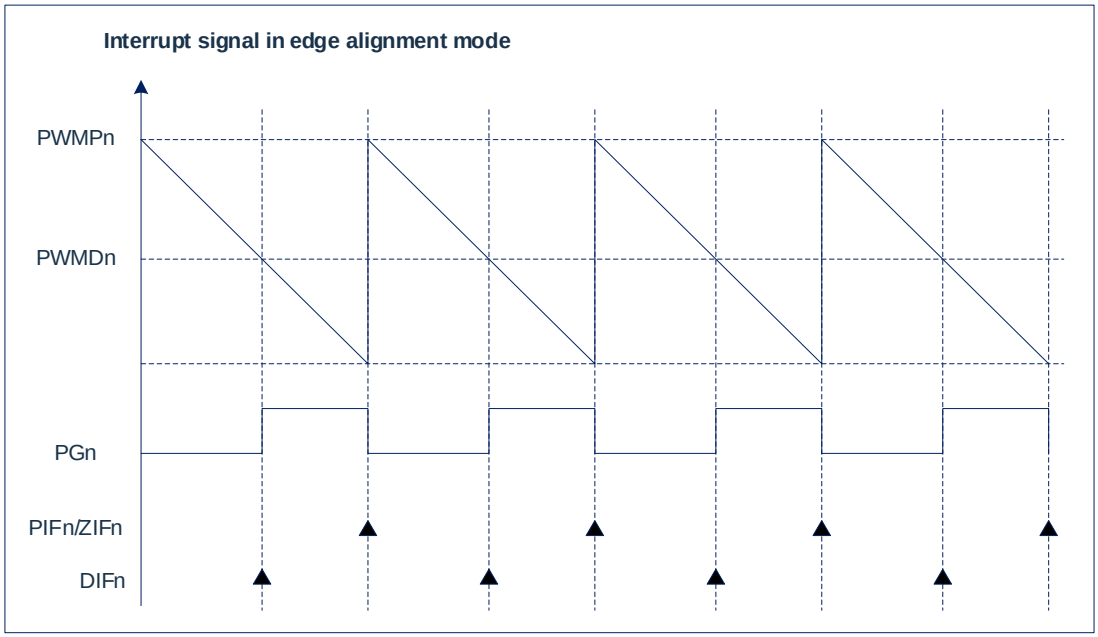
In the case of level braking, when the brake signal generates PWMFBF, it is set to 1. If the software writes PWMFBF to 0 when the brake signal is not canceled, then the PWMFBF will remain 0 until the brake signal is canceled and the brake signal is generated again. is 1. In order to avoid clearing PWMFBF when the brake signal is not canceled, you can check whether the brake signal is canceled by checking the BRKAF bit.

19.3.7 Interrupt Function

The enhanced PWM has a total of 25 interrupt flags, including 6 cycle interrupt flags, 6 zero interrupt flags, 6 up-comparison interrupt flags, 6 down-comparison interrupt flags, and 1 brake interrupt flag. It does not matter whether the corresponding interrupt enable bit is turned on or not. To enable any type of PWM interrupt, you need to open the global interrupt enable bit (EA=1) and the PWM total interrupt enable bit PWMIE, in order to successfully configure the PWM interrupt function. All interrupts of PWM share one interrupt vector entry, so after entering the interrupt service routine, the user can judge which type of interrupt is generated through the interrupt flag bit.

The interrupt mechanism of the enhanced PWM is very flexible. For the center alignment method, there are as many as 4 types of interrupts: zero point interrupt, up-comparison interrupt, period interrupt, and down-comparison interrupt. For edge alignment, there are 3 types of interrupts: cycle interrupt, comparison interrupt, and zero-point interrupt, among which cycle interrupt and zero-point interrupt are the same.

For example, the interrupt signal timing diagram of edge alignment mode is shown in the following figure：



19.4 PWM Related Registers

19.4.1 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 prescaler, clock divider enable bit;

1= Disable (PWMmnPSC, PWMmnDIV are cleared to 0);

0= Enable .

Bit5~Bit4 PWMMODE<1:0>: PWM mode control bits;

00= Independent mode;

01= Complementary mode;

10= Synchronous mode;

11= Reserved.

Bit3 GROUPEN: PWM group function enable bit;

1= PG0 controls PG2, PG4; PG1 controls PG3, PG5;

0= All PWM channel signals are independent of each other.

Bit2 ASYMEN: Asymmetric counting enable bit in PWM center alignment mode;

1= Asymmetric counting enable;

0= Symmetric counting enable.

Bit1 CNTTYPE: PWM counting alignment selection bit;

1= Center alignment;

0= Edge alignment.

Bit0 -- Reserved, must be 0.

19.4.2 PWM Output Enable Control Register PWMOE

F121H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMOE	--	--	PWM5OE	PWM4OE	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~Bit6 -- Reserved, must be both 0.

Bit5 PWM5OE: PWM channel 5 output enable bit;
1= Enable;
0= Disable.

Bit4 PWM4OE: PWM channel 4 output enable bit;
1= Enable;
0= Disable.

Bit3 PWM3OE: PWM channel 3 output enable bit;
1= Enable;
0= Disable.

Bit2 PWM2OE: PWM channel 2 output enable bit;
1= Enable;
0= Disable.

Bit1 PWM1OE: PWM channel 1 output enable bit;
1= Enable;
0= Disable.

Bit0 PWM0OE: PWM channel 0 output enable bit;
1= Enable;
0= Disable.

19.4.3 PWM0/PWM1 Clock Prescaler 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 prescaler control bit;
00= Prescaler clock stop, PWM0/1 counter stop ;
Other= (PWM01PSC+1) Frequency division of the system clock.

19.4.4 PWM2/PWM3 Clock Prescaler 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 prescaler control bit;
00= Prescaler clock stop, PWM2/3 counter stop ;
Other= (PWM23PSC+1) Frequency division of the system clock.

19.4.5 PWM4/PWM5 Clock Prescaler Control Register PWM45PSC

F125H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM45PSC	PWM45PSC7	PWM45PSC6	PWM45PSC5	PWM45PSC4	PWM45PSC3	PWM45PSC2	PWM45PSC1	PWM45PSC0
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 PWM45PSC<7:0>: PWM channel 4/5 prescaler control bit;
00= Prescaler clock stop, PWM4/5 counter stop ;
Other= (PWM45PSC+1) Frequency division of the system clock.

19.4.6 PWM Clock Division Control Register PWMnDIV(n=0-5)

	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

Register PWMnDIV (n=0-5) address:F12AH，F12BH，F12CH，F12DH，F12EH，F12FH.

Bit7~Bit3 -- Reserved, must be 0's.
Bit2~Bit0 PWMnDIV<2:0>: PWM channel n clock division 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 pre-clock).

19.4.7 PWM Data Load Enable Control Register PWMLOADEN

F129H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMLOADEN	--	--	PWM5LE	PWM4LE	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~Bit6 -- Reserved, must be both 0.
Bit5~Bit0 PWMnLE: Data loading enable bit of PWM channel n (n=0-5) (hardware cleared after loading is completed); When PWMnLE=1, changes to the contents of the period and duty cycle registers may cause unpredictable results.
1= Enable loading period, duty cycle data (PERIODn, CMPn, CMPDn).
0= Writing 0 is invalid.

19.4.8 PWM Output Polarity Control Register PWMPINV

F122H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMPINV	--	--	PWM5PINV	PWM4PINV	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~Bit6 -- Reserved, must be both 0.

Bit5~Bit0 PWMnPINV: PWM channel n output polarity control bit (n=0-5);

1= Reverse output;

0= Normal output.

19.4.9 PWM Counter Mode Control Register PWMCNTM

F127H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMCNTM	--	--	PWM5CNTM	PWM4CNTM	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~Bit6 -- Reserved, must be 0.

Bit5~Bit0 PWMnCNTM: PWM channel n counter mode control bit (n=0-5);

1= Automatic loading mode;

0= One-shot mode.

19.4.10 PWM Counter Enable Control Register PWMCNTE

F126H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMCNTE	--	--	PWM5CNTE	PWM4CNTE	PWM3CNTE	PWM2CNTE	PWM1CNTE	PWM0CNTE
R/W	R	R	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 PWMnCNTE: PWM channel n counter enable control bit (n=0-5);

1= PWMn counter is turned on (PWMn starts to output);

0= PWMn counter is stopped (software write 0 to stop the counter and clear the counter value) .

(This bit is cleared to 0 by hardware when the brake is triggered; this bit is cleared to 0 by hardware when the one-shot mode is completed)

19.4.11 PWM Counter Mode Control Register PWMCNTCLR

F128H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMCNTCLR	--	--	PWM5CNTCLR	PWM4CNTCLR	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~Bit6 -- Reserved, must be both 0.

Bit5~Bit0 PWMnCNTCLR: PWM channel n counter clearing control bit (n=0-5) (hardware automatic clearing);

1= PWMn counter clearing;

0= Writing 0 is invalid.

19.4.12 PWM Period Data Register Low 8 bits PWMPnL (n=0-5)

	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

Register PWMPnL (n=0-5) address:F130H，F132H，F134H，F136H，F138H，F13AH.

Bit7~Bit0 PWMPnL<7:0>: The low 8 bits of the PWM channel n period data register.

19.4.13 PWM Period Data Register High 8 bits PWMPnH (n=0-5)

	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

Register PWMPnH (n=0-5) address:F131H，F133H，F135H，F137H，F139H，F13BH.

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

19.4.14 PWM Compare Data Register Low 8 bits PWMDnL (n=0-5)

	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

Register PWMDnL (n=0-5) address:F140H，F142H，F144H，F146H，F148H，F14AH.

Bit7~Bit0 PWMDnL<7:0>: The low 8 bits of the PWM channel n compare data (duty cycle data) register

19.4.15 PWM Compare Data Register High 8 bits PWMDnH (n=0-5)

	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

Register PWMDnH (n=0-5) address: F141H，F143H，F145H，F147H，F149H，F14BH.

Bit7~Bit0 PWMDnH<7:0>: The high 8 bits of the PWM channel n compare data (duty cycle data) register.

19.4.16 PWM Down Compare Data Register Low 8 bits PWMDDnL (n=0-5)

	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

Register PWMDDnL (n=0-5) address:F150H，F152H，F154H，F156H，F158H，F15AH.

Bit7~Bit0 PWMDDnL<7:0>: The low 8 bits of the PWM channel n downward comparison data (duty cycle data under asymmetric counting) register.

19.4.17 PWM Down Compare Data Register High 8 bits PWMDDnH (n=0-5)

	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

Register PWMDDnH (n=0-5) address:F151H，F153H，F155H，F157H，F159H，F15BH.

Bit7~Bit0 PWMDDnH<7:0>: The high 8 bits of the PWM channel n downward comparison data (duty cycle data under asymmetric counting) register.

19.4.18 PWM Programmable Dead-band Delay Control Register PWMDTE

F160H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMDTE	--	--	--	--	--	PWM45DTE	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~Bit3 -- Reserved, must be both 0.

Bit2 PWM45DTE: PWM4/5 channel dead-band delay enable bit;

1= Enable;

0= Disable.

Bit1 PWM23DTE: PWM2/3 channel dead-band delay enable bit;

1= Enable;

0= Disable.

Bit0 PWM01DTE: PWM0/1 channel dead-band delay enable bit;

1= Enable;

0= Disable.

19.4.19 PWM0/PWM1 Programmable Dead-band Delay Time 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-band delay data register.

19.4.20 PWM2/PWM3 Programmable Dead-band Delay Time 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-band delay data register.

19.4.21 PWM4/PWM5 Programmable Dead-band Delay Time Register PWM45DT

F163H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM45DT	PWM45DT7	PWM45DT6	PWM45DT5	PWM45DT4	PWM45DT3	PWM45DT2	PWM45DT1	PWM45DT0
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 PWM45DT<7:0>: PWM channel 4/5 dead-band delay data register.

19.4.22 PWM Mask Enable Control Register PWMMASKE

F164H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMMASKE	--	--	PWM5MASKE	PWM4MASKE	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~Bit6 -- Reserved, all must be 0.

Bit5~Bit0 PWMnMASKE: PWM channel n mask control enable bit (n=0-5);

1= PWMn channel enable mask data output;

0= PWMn channel disable mask data output (normal output PWM waveform).

19.4.23 PWM Mask Data Register PWMMASKD

F165H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMMASKD	--	--	PWM5MASKD	PWM4MASKD	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~Bit6 -- Reserved, all must be 0.

Bit5~Bit0 PWMnMASKD: PWM channel n mask data bit (n=0-5);

1= PWMn channel output high;

0= PWMn channel output low.

19.4.24 PWM Brake Control Register PWMFBKC

F166H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMFBKC	PWMFBIE	PWMFBF	BRKAF	PWMFBKSW	PWMFBES	--	PWMFBEN	--
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 interrupt mask bit;
1= Enable interrupt;
0= Disable interrupt.
Bit6 PWMFBF: PWM brake flag bit (write 0 to clear);
1= Brake operation (PWM output brake data register value);
0= No brake operation.
Bit5 BRKAF: EPWM fault signal flag (read only)
1= A fault signal is generated or the brake signal remains active;
0= No malfunction occurred.
Bit4 PWMFBKSW: PWM software brake signal start bit;
1= PWM generates software brake signal;
0= Disabled .
Bit3 PWMFBES: PWM external hardware brake channel (FB) trigger level selection bit;
1= high level ;
0= low level.
Bit2 -- Reserved , must be 0.
Bit1 PWMFBEN: PWM external hardware brake channel (FB1) enable bit;
1= Enable ;
0= Disable .
Bit0 -- Reserved , must be 0.

19.4.25 PWM Brake Data Register PWMFBKD

F167H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMFBKD	--	--	PWM5FBKD	PWM4FBKD	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~Bit6 -- Reserved, all must be 0.
Bit5~Bit0 PWMnFBKD: PWM channel n brake data bit (n=0-5);
1= PWMn channel outputs high after brake operation occurs.
0= PWMn channel outputs low after braking operation occurs.

19.4.26 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	W	R/W	R/W	R/W
Reset value	0	0	0	0	0	1	0	0

Bit7 BRKOSF: EPWM fault protection output status flag (read only)
0= EPWMn channel is in normal output state
1= The EPWMn channel is the data state of the output BRKODn

Bit6~Bit4 BRKRCS<2:0>: EPWM fault recovery loading point selection bit;
000= The load point of EPWM0 is restored;
001= The load point of EPWM1 is restored;
010= The load point of EPWM2 is restored;
011= The load point of EPWM3 is restored;
100= The load point of EPWM4 is restored;
101= The load point of EPWM5 is restored;
other = Reserved

Bit3 BRKCLR: EPWM fault protection clear bit (write only)
0= Invalid
1= Clear failsafe state
Note: Only when BRKAF=0 can write 1 to perform fault clearing operation, otherwise the operation is invalid.

Bit2 BRKEN: EPWM fault protection enable bit
0= Enable ;
1= Disable .

Bit1~Bit0 BRKMS<1:0>: Failsafe mode selection bits
00= Stop mode
01= Pause mode
10= Recovery mode
11= Delayed recovery mode

19.4.27 PWM Delay Recovery Data Register Low 8 bits 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>: Fault protection recovery delay data low 8 bits (only delay recovery mode is valid)

19.4.28 PWM Delay 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>: Fault protection recovery delay data high 2 bits (only delay recovery mode is valid)

Delay time = $BRKRDT[9:0] \times T_{CLK}$

19.5 PWM Interrupt Related Registers

19.5.1 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 interrupt;
0= Disable SPI interrupt.
Bit6 I2CIE: I^2C interrupt enable bit;
1= Enable I^2C interrupt;
0= Disable I^2C interrupt.
Bit5 WDTIE: WDT interrupt enable bit;
1= Enable WDT overflow interrupt;
0= Disable WDT overflow interrupt.
Bit4 ADCIE: ADC interrupt enable bit;
1= Enable ADC interrupt;
0= Disable ADC interrupt.
Bit3 PWMIE: PWM total interrupt enable bit;
1= Enable all PWM interrupts;
0= Disable all PWM interrupts.
Bit2 --: Reserved, must be zero.
Bit1 ET4: Timer4 interrupt enable bit;
1= Enable Timer4 interrupt;
0= Disable Timer4 interrupt.
Bit0 ET3: Timer3 interrupt enable bit;
1= Enable Timer3 interrupt;
0= Disable Timer3 interrupt.

19.5.2 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 as high-level interrupt;
0= Set as low-level interrupt.

Bit6 PI2C: I^2C interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.

Bit5 PWDT: WDT interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.

Bit4 PADC: ADC interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.

Bit3 PPWM: PWM interrupt priority control bit
1= Set as high-level interrupt;
0= Set as low-level interrupt.

Bit2 -- Reserved, must be zero.

Bit1 PT4: TIMER4 interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.

Bit0 PT3: TIMER3 interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.

19.5.3 PWM Period Interrupt Mask Register PWMPIE

F168H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMPIE	--	--	PWM5PIE	PWM4PIE	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~Bit6 -- Reserved, all must be 0.

Bit5~Bit0 PWMnPIE: PWM channel n-period interrupt mask bit (n=0-5);
1= Enable interrupt;
0= Disable interrupt.

19.5.4 PWM Zero Interrupt Mask Register PWMZIE

F169H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMZIE	--	--	PWM5ZIE	PWM4ZIE	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~Bit6 -- Reserved, all must be 0.

Bit5~Bit0 PWMnZIE: PWM channel n zero interrupt mask bit (n=0-5);

1= Enable interrupt;

0= Disable interrupt.

19.5.5 PWM Compare Up Interrupt Enable Register PWMUIE

F16AH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMUIE	--	--	PWM5UIE	PWM4UIE	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~Bit6 -- Reserved, all must be 0.

Bit5~Bit0 PWMnUIE: PWM channel n upward comparison interrupt mask bit (n=0-5);

1= Enable interrupt;

0= Disable interrupt.

19.5.6 PWM Compare Down Interrupt Enable Register PWMDIE

F16BH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMDIE	--	--	PWM5DIE	PWM4DIE	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~Bit6 -- Reserved, all must be 0.

Bit5~Bit0 PWMnDIE: PWM channel n downward comparison interrupt mask bit (n=0-5);

1= Enable interrupt;

0= Disable interrupt.

19.5.7 PWM Period Interrupt Flag Register PWMPIF

F16CH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMPIF	--	--	PWM5PIF	PWM4PIF	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~Bit6 -- Reserved, all must be 0.

Bit5~Bit0 PWMnPIF: PWM channel n cycle interrupt flag bit (n=0-5);

1= Generate interrupt (cleared by software);

0= Not generate interrupt.

19.5.8 PWM Zero Interrupt Flag Register PWMZIF

F16DH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMZIF	--	--	PWM5ZIF	PWM4ZIF	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~Bit6 -- Reserved, all must be 0.

Bit5~Bit0 PWMnZIF: PWM channel n zero interrupt flag bit (n=0-5);

1= Generate interrupt (cleared by software);

0= Not generate interrupt.

19.5.9 PWM Up Compare Interrupt Flag Register PWMUIF

F16EH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMUIF	--	--	PWM5UIF	PWM4UIF	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~Bit6 -- Reserved, all must be 0.

Bit5~Bit0 PWMnUIF: PWM channel n upward comparison interrupt flag bit (n=0-5);

1= Generate interrupt (cleared by software);

0= Not generate interrupt.

19.5.10 PWM Down Compare Interrupt Flag Register PWMDIF

F16FH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMDIF	--	--	PWM5DIF	PWM4DIF	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~Bit6 -- Reserved, all must be 0.

Bit5~Bit0 PWMnDIF: PWM channel n downward comparison interrupt flag bit (n=0-5);

1= Generate interrupt (cleared by software);

0= Not generate interrupt.

20. Hardware LCD Driver

20.1 Overview

The hardware LCD driver includes a controller, a duty cycle generator, and COM and SEG output ports.

The hardware LCD driver supports two modes of traditional resistance and fast charging, and the bias resistance can be selected from 60KΩ, 225KΩ, and 900KΩ. The fast-charging mode is a new design method. When the 225KΩ or 900KΩ resistance mode is selected, there will be a period of time to select 60KΩ, and then automatically switch to 225KΩ, 900KΩ after this period of time, which can reduce the system power consumption and also can guarantee the brightness of LCD.

The clock source of LCD driver supports Fsys, LSI, LSE three kinds; Among them, Fsys can reach up to 48MHz, LSI and LSE can drive LCD under sleep condition. When the LSI drives the LCD in the sleep state, the LSI timing wake-up function needs to be turned on.

20.2 Characteristics

The LCD driver has the following characteristics:

- Max supported LCD channels: 8COM x 32SEG、6COM x 34SEG、5COM x 35SEG、4COM x 36SEG.
- Support contrast adjustment.
- Optional bias voltage: 1/2、1/3、1/4.
- The duty cycle is optional: 1/4、1/5、1/6、1/8.
- Three kinds of clock sources are optional: system clock, LSI, LSE.
- It supports two modes: traditional resistance and fast charging.
- Choice of fast charging time.

20.3 Related Registers

20.3.1 LCD Control Register LCDCON0

F680H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCDCON0	LCDEN	--	LCDDM1	LCDDM0	COM_SEL	COM_MOD	DUTY1	DUTY0
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 control bit;
1= LCD enable;
0= LCD disable.
Bit6 -- Reserved, must be zero.
Bit5~Bit4 LCDDM<1:0>: LCD display mode;
1x= Normal output;
01= SEG fully open;
00= SEG fully closed.
Bit3 COM_SEL: COM port selection (valid when DUTY[1:0]=11);
See the COM selection instruction table for control details.
Bit2 COM_MOD: Mode selection;
1= MODE1;
0= MODE0.
Bit1~Bit0 DUTY<1:0>: LCD duty cycle selection bits;
11= 1/4DUTY;
10= 1/5DUTY;
01= 1/6DUTY;
00= 1/8DUTY.

COM Selection Description Table

DUTY	COM_MOD	COM_SEL	ICOM0	ICOM1	ICOM2	ICOM3	ICOM4	ICOM5	ICOM6	ICOM7	Valid SEG port
11	1	1	LCD_C3	LCD_C2	LCD_C1	LCD_C0	-	-	-	-	LCD_S0-35
		0	LCD_C7	LCD_C6	LCD_C5	LCD_C4	-	-	-	-	LCD_S39-36 LCD_S0-31
	0	1	LCD_C4	LCD_C5	LCD_C6	LCD_C7	-	-	-	-	LCD_S39-36 LCD_S0-31
		0	LCD_C0	LCD_C1	LCD_C2	LCD_C3	-	-	-	-	LCD_S0-35
10	1	-	LCD_C7	LCD_C6	LCD_C5	LCD_C4	LCD_C3	-	-	-	LCD_S39-37 LCD_S0-31
	0	-	LCD_C0	LCD_C1	LCD_C2	LCD_C3	LCD_C4	-	-	-	LCD_S0-34
01	1	-	LCD_C7	LCD_C6	LCD_C5	LCD_C4	LCD_C3	LCD_C2	-	-	LCD_S39-38 LCD_S0-31
	0	-	LCD_C0	LCD_C1	LCD_C2	LCD_C3	LCD_C4	LCD_C5	-	-	LCD_S0-33
00	1	-	LCD_C7	LCD_C6	LCD_C5	LCD_C4	LCD_C3	LCD_C2	LCD_C1	LCD_C0	LCD_S0-31
	0	-	LCD_C0	LCD_C1	LCD_C2	LCD_C3	LCD_C4	LCD_C5	LCD_C6	LCD_C7	LCD_S0-31

Note: In the table above ICOM0-ICOM7 are internal COM driver for the LCD output signal.

LCD_C0-7, LCD_S0-39 are the ports to which the internal driving signals of the LCD are finally mapped.

20.3.2 LCD Control Register LCDCON1

F681H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCDCON1	LCDTEN	--	BIAS1	BIAS0	LCDTVS3	LCDTVS2	LCDTVS1	LCDTVS0
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 LCDTEN: LCD power supply voltage selection;
1= LCD voltage is provided by internal power supply V_{LCD};
0= LCD voltage is provided by VDD.
Bit6 -- Reserved, must be zero.
Bit5~Bit4 BIAS<1:0>: LCD display bias voltage setting;
1X= 1/4 V_{LCD};
01= 1/3 V_{LCD};
00= 1/2 V_{LCD}.
Bit3~Bit0 LCDTVS<3:0>: LCD internal voltage selection;
0000~1110= V_{LCD}=(15+LCDTVS<3:0>)*VDD/30.
1111= Reserved, prohibit selection.

Note: BIAS and DUTY are independent of each other. Regardless of the bias voltage setting, you can select 1/4, 1/5, 1/6, 1/8 duty cycle.

20.3.3 LCD Control Register LCDCON2

F682H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCDCON2	LSI_EN	--	CLKSEL1	CLKSEL0	LCDPSC3	LCDPSC2	LCDPSC1	LCDPSC0
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 LSI_EN LCD control LSI clock enable selection;
1= When the LCD is enabled, the LCD control LSI clock is enabled
0= LCD control LSI clock enable is invalid
Bit6 -- Reserved, all must be 0.
Bit5~Bit4 CLKSEL<1:0>: LCD clock source F_{LCD} selection;
1X= LSI (125KHz);
01= LSE (32.768KHz);
00= Fsys (system clock).
Bit3~Bit0 LCDPSC<3:0>: LCD clock division ratio selection;

0000=	F_{LCD}/64;	1000=	F_{LCD}/16384;
0001=	F_{LCD}/128;	1001=	F_{LCD}/32768;
0010=	F_{LCD}/256;	1010=	F_{LCD}/65536;
0011=	F_{LCD}/512;	1011=	F_{LCD}/65536;
0100=	F_{LCD}/1024;	1100=	F_{LCD}/65536;
0101=	F_{LCD}/2048;	1101=	F_{LCD}/65536;
0110=	F_{LCD}/4096;	1110=	F_{LCD}/65536;
0111=	F_{LCD}/8192;	1111=	F_{LCD}/65536.

20.3.4 LCD Control Register LCDCON3

F683H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCDCON3	--	--	LCDRM1	LCDRM0	--	FCMODE	FCCTLM1	FCCTLM0
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, all must be 0.
Bit5~Bit4 LCDRM<1:0>: LCD voltage divider resistance selection;
00= 60KΩ;
01= 225KΩ;
1X= 900KΩ.
Bit3 -- Reserved, must be 0.
Bit2 FCMODE: Charging mode selection;
1= fast charging mode (when 225KΩ/900KΩ voltage divider resistor is selected, this mode is valid);
0= Traditional resistance mode.
Bit1~Bit0 FCCTLM<1:0>: Fast charge mode time control bit;
00= 1/8 COM cycle;
01= 1/16 COM cycle;
10= 1/32 COM cycle;
11= 1/64 COM cycle.

20.3.5 LCD Control Register LCDCON4

F68AH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCDCON4	--	--	--	S_LCDBUFOL_DR			S_LCDBUF	EN_LCDBUF
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~Bit5 -- Reserved, all must be 0.
Bit4~Bit2 S_LCDBUFOL_DR Drive capability gear selection
000-111= 100uA~3mA
Bit1 S_LCDBUF LCD open-loop and closed-loop output selection;
1= Open loop BUFFER output
0= Closed-loop BUFFER output
Bit0 EN_LCDBUF Output mode selection
1= BUFFER output
0= Low power mode output

20.3.6 COM Port Enable Control Register LCDCOMEN

F684H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCDCOMEN	COMEN7	COMEN6	COMEN5	COMEN4	COMEN3	COMEN2	COMEN1	COMEN0
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 COMEN<7:0>: LCD_C7-LCD_C0 port enable control bit;
1= Enable ;
0= Disable .

20.3.7 SEG Port Enable Control Register LCDSEGEN0

F685H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCDSEGEN0	SEGEN7	SEGEN6	SEGEN5	SEGEN4	SEGEN3	SEGEN2	SEGEN1	SEGEN0
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 SEGEN<7:0>: LCD_S7-LCD_S0 port enable control bit;
1= Enable ;
0= Disable .

20.3.8 SEG Port Enable Control Register LCDSEGEN1

F686H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCDSEGEN1	SEGEN15	SEGEN14	SEGEN13	SEGEN12	SEGEN11	SEGEN10	SEGEN9	SEGEN8
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 SEGEN<15:8>: LCD_S15-LCD_S8 port enable control bit;
1= Enable ;
0= Disable .

20.3.9 SEG Port Enable Control Register LCDSEGEN2

F687H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCDSEGEN2	SEGEN23	SEGEN22	SEGEN21	SEGEN20	SEGEN19	SEGEN18	SEGEN17	SEGEN16
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 SEGEN<23:16>: LCD_S23-LCD_S16 port enable control bit;
1= Enable ;
0= Disable .

20.3.10 SEG Port Enable Control Register LCDSEGEN3

F688H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCDSEGEN3	SEGEN31	SEGEN30	SEGEN29	SEGEN28	SEGEN27	SEGEN26	SEGEN25	SEGEN24
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 SEGEN<31:24>: LCD_S31-LCD_S24 port enable control bit;
1= Enable ;
0= Disable .

20.3.11 SEG Port Enable Control Register LCDSEGEN4

F689H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCDSEGEN4	SEGEN39	SEGEN38	SEGEN37	SEGEN36	SEGEN35	SEGEN34	SEGEN33	SEGEN32
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 SEGEN<39:32>: LCD_S39-LCD_S32 port enable control bit;
1= Enable ;
0= Disable .

20.3.12 SEG Data Register LCDSEGn(n=0-39)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCDSEGn	ICOM7	ICOM6	ICOM5	ICOM4	ICOM3	ICOM2	ICOM1	ICOM0
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

LCDSEG0~LCDSEG39 addresses are:F650H~F677H.

Bit7~Bit0 ICOM<7:0>: LCD_Sn port data output;
1= High level;
0= Low level.

20.4 COM-SEG Data Sheet

The different DUTY of the hardware LCD driver corresponds to the following data sheet.

20.4.1 1/4DUTY

		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
		-	-	-	-	ICOM3	ICOM2	ICOM1	ICOM0
LCDSEG0	F650H	-	-	-	-	SEG0	SEG0	SEG0	SEG0
LCDSEG1	F651H	-	-	-	-	SEG1	SEG1	SEG1	SEG1
LCDSEG2	F652H	-	-	-	-	SEG2	SEG2	SEG2	SEG2
LCDSEG3	F653H	-	-	-	-	SEG3	SEG3	SEG3	SEG3
LCDSEG4	F654H	-	-	-	-	SEG4	SEG4	SEG4	SEG4
LCDSEG5	F655H	-	-	-	-	SEG5	SEG5	SEG5	SEG5
LCDSEG6	F656H	-	-	-	-	SEG6	SEG6	SEG6	SEG6
LCDSEG7	F657H	-	-	-	-	SEG7	SEG7	SEG7	SEG7
LCDSEG8	F658H	-	-	-	-	SEG8	SEG8	SEG8	SEG8
LCDSEG9	F659H	-	-	-	-	SEG9	SEG9	SEG9	SEG9
LCDSEG10	F65AH	-	-	-	-	SEG10	SEG10	SEG10	SEG10
LCDSEG11	F65BH	-	-	-	-	SEG11	SEG11	SEG11	SEG11
LCDSEG12	F65CH	-	-	-	-	SEG12	SEG12	SEG12	SEG12
LCDSEG13	F65DH	-	-	-	-	SEG13	SEG13	SEG13	SEG13
LCDSEG14	F65EH	-	-	-	-	SEG14	SEG14	SEG14	SEG14
LCDSEG15	F65FH	-	-	-	-	SEG15	SEG15	SEG15	SEG15
LCDSEG16	F660H	-	-	-	-	SEG16	SEG16	SEG16	SEG16
LCDSEG17	F661H	-	-	-	-	SEG17	SEG17	SEG17	SEG17
LCDSEG18	F662H	-	-	-	-	SEG18	SEG18	SEG18	SEG18
LCDSEG19	F663H	-	-	-	-	SEG19	SEG19	SEG19	SEG19
LCDSEG20	F664H	-	-	-	-	SEG20	SEG20	SEG20	SEG20
LCDSEG21	F665H	-	-	-	-	SEG21	SEG21	SEG21	SEG21
LCDSEG22	F666H	-	-	-	-	SEG22	SEG22	SEG22	SEG22
LCDSEG23	F667H	-	-	-	-	SEG23	SEG23	SEG23	SEG23
LCDSEG24	F668H	-	-	-	-	SEG24	SEG24	SEG24	SEG24
LCDSEG25	F669H	-	-	-	-	SEG25	SEG25	SEG25	SEG25
LCDSEG26	F66AH	-	-	-	-	SEG26	SEG26	SEG26	SEG26
LCDSEG27	F66BH	-	-	-	-	SEG27	SEG27	SEG27	SEG27
LCDSEG28	F66CH	-	-	-	-	SEG28	SEG28	SEG28	SEG28
LCDSEG29	F66DH	-	-	-	-	SEG29	SEG29	SEG29	SEG29
LCDSEG30	F66EH	-	-	-	-	SEG30	SEG30	SEG30	SEG30
LCDSEG31	F66FH	-	-	-	-	SEG31	SEG31	SEG31	SEG31
LCDSEG32	F670H	-	-	-	-	SEG32	SEG32	SEG32	SEG32
LCDSEG33	F671H	-	-	-	-	SEG33	SEG33	SEG33	SEG33
LCDSEG34	F672H	-	-	-	-	SEG34	SEG34	SEG34	SEG34
LCDSEG35	F673H	-	-	-	-	SEG35	SEG35	SEG35	SEG35

20.4.2 1/5DUTY

		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
		-	-	-	ICOM4	ICOM3	ICOM2	ICOM1	ICOM0
LCDSEG0	F650H	-	-	-	SEG0	SEG0	SEG0	SEG0	SEG0
LCDSEG1	F651H	-	-	-	SEG1	SEG1	SEG1	SEG1	SEG1
LCDSEG2	F652H	-	-	-	SEG2	SEG2	SEG2	SEG2	SEG2
LCDSEG3	F653H	-	-	-	SEG3	SEG3	SEG3	SEG3	SEG3
LCDSEG4	F654H	-	-	-	SEG4	SEG4	SEG4	SEG4	SEG4
LCDSEG5	F655H	-	-	-	SEG5	SEG5	SEG5	SEG5	SEG5
LCDSEG6	F656H	-	-	-	SEG6	SEG6	SEG6	SEG6	SEG6
LCDSEG7	F657H	-	-	-	SEG7	SEG7	SEG7	SEG7	SEG7
LCDSEG8	F658H	-	-	-	SEG8	SEG8	SEG8	SEG8	SEG8
LCDSEG9	F659H	-	-	-	SEG9	SEG9	SEG9	SEG9	SEG9
LCDSEG10	F65AH	-	-	-	SEG10	SEG10	SEG10	SEG10	SEG10
LCDSEG11	F65BH	-	-	-	SEG11	SEG11	SEG11	SEG11	SEG11
LCDSEG12	F65CH	-	-	-	SEG12	SEG12	SEG12	SEG12	SEG12
LCDSEG13	F65DH	-	-	-	SEG13	SEG13	SEG13	SEG13	SEG13
LCDSEG14	F65EH	-	-	-	SEG14	SEG14	SEG14	SEG14	SEG14
LCDSEG15	F65FH	-	-	-	SEG15	SEG15	SEG15	SEG15	SEG15
LCDSEG16	F660H	-	-	-	SEG16	SEG16	SEG16	SEG16	SEG16
LCDSEG17	F661H	-	-	-	SEG17	SEG17	SEG17	SEG17	SEG17
LCDSEG18	F662H	-	-	-	SEG18	SEG18	SEG18	SEG18	SEG18
LCDSEG19	F663H	-	-	-	SEG19	SEG19	SEG19	SEG19	SEG19
LCDSEG20	F664H	-	-	-	SEG20	SEG20	SEG20	SEG20	SEG20
LCDSEG21	F665H	-	-	-	SEG21	SEG21	SEG21	SEG21	SEG21
LCDSEG22	F666H	-	-	-	SEG22	SEG22	SEG22	SEG22	SEG22
LCDSEG23	F667H	-	-	-	SEG23	SEG23	SEG23	SEG23	SEG23
LCDSEG24	F668H	-	-	-	SEG24	SEG24	SEG24	SEG24	SEG24
LCDSEG25	F669H	-	-	-	SEG25	SEG25	SEG25	SEG25	SEG25
LCDSEG26	F66AH	-	-	-	SEG26	SEG26	SEG26	SEG26	SEG26
LCDSEG27	F66BH	-	-	-	SEG27	SEG27	SEG27	SEG27	SEG27
LCDSEG28	F66CH	-	-	-	SEG28	SEG28	SEG28	SEG28	SEG28
LCDSEG29	F66DH	-	-	-	SEG29	SEG29	SEG29	SEG29	SEG29
LCDSEG30	F66EH	-	-	-	SEG30	SEG30	SEG30	SEG30	SEG30
LCDSEG31	F66FH	-	-	-	SEG31	SEG31	SEG31	SEG31	SEG31
LCDSEG32	F670H	-	-	-	SEG32	SEG32	SEG32	SEG32	SEG32
LCDSEG33	F671H	-	-	-	SEG33	SEG33	SEG33	SEG33	SEG33
LCDSEG34	F672H	-	-	-	SEG34	SEG34	SEG34	SEG34	SEG34

20.4.3 1/6DUTY

		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
		-	-	ICOM5	ICOM4	ICOM3	ICOM2	ICOM1	ICOM0
LCDSEG0	F650H	-	-	SEG0	SEG0	SEG0	SEG0	SEG0	SEG0
LCDSEG1	F651H	-	-	SEG1	SEG1	SEG1	SEG1	SEG1	SEG1
LCDSEG2	F652H	-	-	SEG2	SEG2	SEG2	SEG2	SEG2	SEG2
LCDSEG3	F653H	-	-	SEG3	SEG3	SEG3	SEG3	SEG3	SEG3
LCDSEG4	F654H	-	-	SEG4	SEG4	SEG4	SEG4	SEG4	SEG4
LCDSEG5	F655H	-	-	SEG5	SEG5	SEG5	SEG5	SEG5	SEG5
LCDSEG6	F656H	-	-	SEG6	SEG6	SEG6	SEG6	SEG6	SEG6
LCDSEG7	F657H	-	-	SEG7	SEG7	SEG7	SEG7	SEG7	SEG7
LCDSEG8	F658H	-	-	SEG8	SEG8	SEG8	SEG8	SEG8	SEG8
LCDSEG9	F659H	-	-	SEG9	SEG9	SEG9	SEG9	SEG9	SEG9
LCDSEG10	F65AH	-	-	SEG10	SEG10	SEG10	SEG10	SEG10	SEG10
LCDSEG11	F65BH	-	-	SEG11	SEG11	SEG11	SEG11	SEG11	SEG11
LCDSEG12	F65CH	-	-	SEG12	SEG12	SEG12	SEG12	SEG12	SEG12
LCDSEG13	F65DH	-	-	SEG13	SEG13	SEG13	SEG13	SEG13	SEG13
LCDSEG14	F65EH	-	-	SEG14	SEG14	SEG14	SEG14	SEG14	SEG14
LCDSEG15	F65FH	-	-	SEG15	SEG15	SEG15	SEG15	SEG15	SEG15
LCDSEG16	F660H	-	-	SEG16	SEG16	SEG16	SEG16	SEG16	SEG16
LCDSEG17	F661H	-	-	SEG17	SEG17	SEG17	SEG17	SEG17	SEG17
LCDSEG18	F662H	-	-	SEG18	SEG18	SEG18	SEG18	SEG18	SEG18
LCDSEG19	F663H	-	-	SEG19	SEG19	SEG19	SEG19	SEG19	SEG19
LCDSEG20	F664H	-	-	SEG20	SEG20	SEG20	SEG20	SEG20	SEG20
LCDSEG21	F665H	-	-	SEG21	SEG21	SEG21	SEG21	SEG21	SEG21
LCDSEG22	F666H	-	-	SEG22	SEG22	SEG22	SEG22	SEG22	SEG22
LCDSEG23	F667H	-	-	SEG23	SEG23	SEG23	SEG23	SEG23	SEG23
LCDSEG24	F668H	-	-	SEG24	SEG24	SEG24	SEG24	SEG24	SEG24
LCDSEG25	F669H	-	-	SEG25	SEG25	SEG25	SEG25	SEG25	SEG25
LCDSEG26	F66AH	-	-	SEG26	SEG26	SEG26	SEG26	SEG26	SEG26
LCDSEG27	F66BH	-	-	SEG27	SEG27	SEG27	SEG27	SEG27	SEG27
LCDSEG28	F66CH	-	-	SEG28	SEG28	SEG28	SEG28	SEG28	SEG28
LCDSEG29	F66DH	-	-	SEG29	SEG29	SEG29	SEG29	SEG29	SEG29
LCDSEG30	F66EH	-	-	SEG30	SEG30	SEG30	SEG30	SEG30	SEG30
LCDSEG31	F66FH	-	-	SEG31	SEG31	SEG31	SEG31	SEG31	SEG31
LCDSEG32	F670H	-	-	SEG32	SEG32	SEG32	SEG32	SEG32	SEG32
LCDSEG33	F671H	-	-	SEG33	SEG33	SEG33	SEG33	SEG33	SEG33

20.4.4 1/8DUTY

		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
		ICOM7	ICOM6	ICOM5	ICOM4	ICOM3	ICOM2	ICOM1	ICOM0
LCDSEG0	F650H	SEG0	SEG0	SEG0	SEG0	SEG0	SEG0	SEG0	SEG0
LCDSEG1	F651H	SEG1	SEG1	SEG1	SEG1	SEG1	SEG1	SEG1	SEG1
LCDSEG2	F652H	SEG2	SEG2	SEG2	SEG2	SEG2	SEG2	SEG2	SEG2
LCDSEG3	F653H	SEG3	SEG3	SEG3	SEG3	SEG3	SEG3	SEG3	SEG3
LCDSEG4	F654H	SEG4	SEG4	SEG4	SEG4	SEG4	SEG4	SEG4	SEG4
LCDSEG5	F655H	SEG5	SEG5	SEG5	SEG5	SEG5	SEG5	SEG5	SEG5
LCDSEG6	F656H	SEG6	SEG6	SEG6	SEG6	SEG6	SEG6	SEG6	SEG6
LCDSEG7	F657H	SEG7	SEG7	SEG7	SEG7	SEG7	SEG7	SEG7	SEG7
LCDSEG8	F658H	SEG8	SEG8	SEG8	SEG8	SEG8	SEG8	SEG8	SEG8
LCDSEG9	F659H	SEG9	SEG9	SEG9	SEG9	SEG9	SEG9	SEG9	SEG9
LCDSEG10	F65AH	SEG10	SEG10	SEG10	SEG10	SEG10	SEG10	SEG10	SEG10
LCDSEG11	F65BH	SEG11	SEG11	SEG11	SEG11	SEG11	SEG11	SEG11	SEG11
LCDSEG12	F65CH	SEG12	SEG12	SEG12	SEG12	SEG12	SEG12	SEG12	SEG12
LCDSEG13	F65DH	SEG13	SEG13	SEG13	SEG13	SEG13	SEG13	SEG13	SEG13
LCDSEG14	F65EH	SEG14	SEG14	SEG14	SEG14	SEG14	SEG14	SEG14	SEG14
LCDSEG15	F65FH	SEG15	SEG15	SEG15	SEG15	SEG15	SEG15	SEG15	SEG15
LCDSEG16	F660H	SEG16	SEG16	SEG16	SEG16	SEG16	SEG16	SEG16	SEG16
LCDSEG17	F661H	SEG17	SEG17	SEG17	SEG17	SEG17	SEG17	SEG17	SEG17
LCDSEG18	F662H	SEG18	SEG18	SEG18	SEG18	SEG18	SEG18	SEG18	SEG18
LCDSEG19	F663H	SEG19	SEG19	SEG19	SEG19	SEG19	SEG19	SEG19	SEG19
LCDSEG20	F664H	SEG20	SEG20	SEG20	SEG20	SEG20	SEG20	SEG20	SEG20
LCDSEG21	F665H	SEG21	SEG21	SEG21	SEG21	SEG21	SEG21	SEG21	SEG21
LCDSEG22	F666H	SEG22	SEG22	SEG22	SEG22	SEG22	SEG22	SEG22	SEG22
LCDSEG23	F667H	SEG23	SEG23	SEG23	SEG23	SEG23	SEG23	SEG23	SEG23
LCDSEG24	F668H	SEG24	SEG24	SEG24	SEG24	SEG24	SEG24	SEG24	SEG24
LCDSEG25	F669H	SEG25	SEG25	SEG25	SEG25	SEG25	SEG25	SEG25	SEG25
LCDSEG26	F66AH	SEG26	SEG26	SEG26	SEG26	SEG26	SEG26	SEG26	SEG26
LCDSEG27	F66BH	SEG27	SEG27	SEG27	SEG27	SEG27	SEG27	SEG27	SEG27
LCDSEG28	F66CH	SEG28	SEG28	SEG28	SEG28	SEG28	SEG28	SEG28	SEG28
LCDSEG29	F66DH	SEG29	SEG29	SEG29	SEG29	SEG29	SEG29	SEG29	SEG29
LCDSEG30	F66EH	SEG30	SEG30	SEG30	SEG30	SEG30	SEG30	SEG30	SEG30
LCDSEG31	F66FH	SEG31	SEG31	SEG31	SEG31	SEG31	SEG31	SEG31	SEG31

21. Hardware LED Driver

21.1 Overview

The chip integrates a hardware LED display drive circuit, which can facilitate users to realize LED display drive.

21.2 Characteristics

The hardware LED driver has the following characteristics:

- 1/4、1/5、1/6、1/8 four kinds of DUTY are optional.
- System clock, LSI, LSE three clock sources are optional.
- 16-bit clock source divider controller.
- Supports up to 8 COM ports and 24 SEG ports.
- Two drive modes of common cathode and common anode for COM port are optional.
- The COM port current 50 mA and 150mA are two options (V_{OL}=1.5V@VDD=5V).
- The SEG port current 16 selectable, the max current can reach 40mA (V_{OH}=3.5V@VDD=5V).

21.3 Related Registers

21.3.1 LED Control Register LEDCON

F765H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDCON	LED_EN	DUTY1	DUTY0	CC_CA	COM_SEL	MODE	CLKSEL1	CLKSEL0
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	LED_EN:	LED enable control bit;
	1=	LED enable;
	0=	LED disable.
Bit6~Bit5	DUTY<1:0>:	LED duty cycle selection bit;
	11=	1/4DUTY;
	10=	1/5DUTY;
	01=	1/6DUTY;
	00=	1/8DUTY;
Bit4	CC_CA:	LED drive mode selection bit ;
	1=	Common anode drive mode;
	0=	Common cathode drive mode.
Bit3	COM_SEL:	COM port selection (valid when DUTY<1:0>=11);
		See the COM selection instruction table for control details.
Bit2	MODE:	Mode selection;
	1=	MODE1;
	0=	MODE0.
Bit1~Bit0	CLKSEL<1:0>:	LED clock source F_{LED} selection;
	11=	LSI;
	10=	LSI;
	01=	LSE;
	00=	Fsys (system clock)

COM Selection Description Table

DUTY	COM_MOD	COM_SEL	ICOM0	ICOM1	ICOM2	ICOM3	ICOM4	ICOM5	ICOM6	ICOM7	Valid SEG port
11	1	1	LED_C3	LED_C2	LED_C1	LED_C0	-	-	-	-	LED_S0-23
		0	LED_C7	LED_C6	LED_C5	LED_C4	-	-	-	-	LED_S27-24 LED_S0-23
	0	1	LED_C4	LED_C5	LED_C6	LED_C7	-	-	-	-	LED_S0-23
		0	LED_C0	LED_C1	LED_C2	LED_C3	-	-	-	-	LED_S0-23
10	1	-	LED_C7	LED_C6	LED_C5	LED_C4	LED_C3	-	-	-	LED_S27-25 LED_S0-23
	0	-	LED_C0	LED_C1	LED_C2	LED_C3	LED_C4	-	-	-	LED_S0-23
01	1	-	LED_C7	LED_C6	LED_C5	LED_C4	LED_C3	LED_C2	-	-	LED_S27-26 LED_S0-23
	0	-	LED_C0	LED_C1	LED_C2	LED_C3	LED_C4	LED_C5	-	-	LED_S0-23
00	1	-	LED_C7	LED_C6	LED_C5	LED_C4	LED_C3	LED_C2	LED_C1	LED_C0	LED_S0-23
	0	-	LED_C0	LED_C1	LED_C2	LED_C3	LED_C4	LED_C5	LED_C6	LED_C7	LED_S0-23

Note: ICOM0-ICOM7 in the table above are the LED internal COM drive output signals.

LED_C0-LED_C7, LED_S0-LED_S28 are the pin names to which the internal drive signal of the LED is finally mapped

21.3.2 LED Clock Prescaler Data Register Low 8 bits LEDCLKL

F766H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDCLKL	CLK7	CLK6	CLK5	CLK4	CLK3	CLK2	CLK1	CLK0
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 CLK<7:0>: LED clock divider low 8 bits

21.3.3 LED Clock Prescaler Data Register High 8 bits LEDCLKH

F767H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDCLKH	CLK15	CLK14	CLK13	CLK12	CLK11	CLK10	CLK9	CLK8
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 CLK<15:8>: LED clock divider high 8 bits

The clock frequency of the LED driver: $F_{LED\_CLK} = F_{LED}$ / （CLK<15:0>+1）.

21.3.4 COM Port Effective Time Selection Register LEDCOMTIME

F768H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDCOMTIME	COMT7	COMT6	COMT5	COMT4	COMT3	COMT2	COMT1	COMT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	1	1	0	0	0	1	1

Bit7~Bit0 COMT<7:0>: COM port effective time setting.
Note: It is prohibited to set to 0x00 and 0xFF;
COM time = (COMT<7:0> + 1) * T_{LED_CLK} .

21.3.5 COM Port Enable Control Register LEDCOMEN

F760H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDCOMEN	COMEN7	COMEN6	COMEN5	COMEN4	COMEN3	COMEN2	COMEN1	COMEN0
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 COMEN<7:0>: LED_C7-LED_C0 port enable control bit
1= Enable ;
0= Disable .

21.3.6 SEG Port Enable Control Register LEDSEGEN0

F761H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDSEGEN0	SEGEN7	SEGEN6	SEGEN5	SEGEN4	SEGEN3	SEGEN2	SEGEN1	SEGEN0
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 SEGEN<7:0>: LED_S7-LED_S0 port enable control bit
1= Enable ;
0= Disable .

21.3.7 SEG Port Enable Control Register LEDSEGEN1

F762H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDSEGEN1	SEGEN15	SEGEN14	SEGEN13	SEGEN12	SEGEN11	SEGEN10	SEGEN9	SEGEN8
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 SEGEN<15:8>: LED_S15-LED_S8 port enable control bit
1= Enable ;
0= Disable .

21.3.8 SEG Port Enable Control Register LEDSEGEN2

F763H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDSEGEN2	SEGEN23	SEGEN22	SEGEN21	SEGEN20	SEGEN19	SEGEN18	SEGEN17	SEGEN16
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 SEGEN<23:16>: LED_S23-LED_S16 port enable control bit
1= Enable ;
0= Disable .

21.3.9 SEG Port Enable Control Register LEDSEGEN3

F764H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDSEGEN3	--	--	--	--	SEGEN27	SEGEN26	SEGEN25	SEGEN24
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~Bit0 SEGEN<27:24>: LED_S27-LED_S24 port enable control bit
1= Enable ;
0= Disable .

21.3.10 COM0 Corresponds To The SEG Data Register LEDC0DATAn（n=0-3）

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDC0DATAn	SEG[8n+7]	SEG[8n+6]	SEG[8n+5]	SEG[8n+4]	SEG[8n+3]	SEG[8n+2]	SEG[8n+1]	SEG[8n]
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

address of LEDC0DATA0 is: F740H; the address of LEDC0DATA1 is: F741H;

address of LEDC0DATA2 is: F742H; the address of LEDC0DATA3 is: F743H;

Bit7~Bit0 SEG<8n+7:8n>: When the COM0 port is valid, SEG[8n+ 7]-SEG[8n] port data output;

1= High level;

0= Low level.

21.3.11 COM1 Corresponds To The SEG Data Register LEDC1DATAn（n=0-3）

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDC1DATAn	SEG[8n+7]	SEG[8n+6]	SEG[8n+5]	SEG[8n+4]	SEG[8n+3]	SEG[8n+2]	SEG[8n+1]	SEG[8n]
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

address of LEDC1DATA0 is: F744H; the address of LEDC1DATA1 is: F745H;

address of LEDC1DATA2 is: F746H; the address of LEDC1DATA3 is: F747H;

Bit7~Bit0 SEG<8n+7:8n>: When the COM1 port is valid, SEG[8n+ 7]-SEG[8n] port data output;

1= High level;

0= Low level.

21.3.12 COM2 Corresponds To The SEG Data Register LEDC2DATAn（n=0-3）

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDC2DATAn	SEG[8n+7]	SEG[8n+6]	SEG[8n+5]	SEG[8n+4]	SEG[8n+3]	SEG[8n+2]	SEG[8n+1]	SEG[8n]
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

address of LEDC2DATA0 is: F748H; the address of LEDC2DATA1 is: F749H;

address of LEDC2DATA2 is: F74AH; the address of LEDC2DATA3 is: F74BH;

Bit7~Bit0 SEG<8n+7:8n>: When the COM2 port is valid, SEG[8n+ 7]-SEG[8n] port data output;

1= High level;

0= Low level.

21.3.13 COM3 Corresponds To The SEG Data Register LEDC3DATAn（n=0-3）

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDC3DATAn	SEG[8n+7]	SEG[8n+6]	SEG[8n+5]	SEG[8n+4]	SEG[8n+3]	SEG[8n+2]	SEG[8n+1]	SEG[8n]
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

address of LEDC3DATA0 is: F74CH; the address of LEDC3DATA1 is: F74DH;

address of LEDC3DATA2 is: F74EH; the address of LEDC3DATA3 is: F74FH;

Bit7~Bit0 SEG<8n+7:8n>: When the COM3 port is valid, SEG[8n+ 7]-SEG[8n] port data output;

1= High level;

0= Low level.

21.3.14 COM4 Corresponds To The SEG Data Register LEDC4DATAn（n=0-3）

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDC4DATAn	SEG[8n+7]	SEG[8n+6]	SEG[8n+5]	SEG[8n+4]	SEG[8n+3]	SEG[8n+2]	SEG[8n+1]	SEG[8n]
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

address of LEDC4DATA0 is: F750H; the address of LEDC4DATA1 is: F751H;

address of LEDC4DATA2 is: F752H; the address of LEDC4DATA3 is: F753H;

Bit7~Bit0 SEG<8n+7:8n>: When the COM4 port is valid, SEG[8n+ 7]-SEG[8n] port data output;

1= High level;

0= Low level.

21.3.15 COM5 Corresponds To The SEG Data Register LEDC5DATAn（n=0-3）

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDC5DATAn	SEG[8n+7]	SEG[8n+6]	SEG[8n+5]	SEG[8n+4]	SEG[8n+3]	SEG[8n+2]	SEG[8n+1]	SEG[8n]
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

address of LEDC5DATA0 is: F754H; the address of LEDC5DATA1 is: F755H;

address of LEDC5DATA2 is: F756H; the address of LEDC5DATA3 is: F757H;

Bit7~Bit0 SEG<8n+7:8n>: When the COM5 port is valid, SEG[8n+ 7]-SEG[8n] port data output;

1= High level;

0= Low level.

21.3.16 COM6 Corresponds To The SEG Data Register LEDC6DATAn（n=0-3）

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDC6DATAn	SEG[8n+7]	SEG[8n+6]	SEG[8n+5]	SEG[8n+4]	SEG[8n+3]	SEG[8n+2]	SEG[8n+1]	SEG[8n]
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

address of LEDC6DATA0 is: F758H; the address of LEDC6DATA1 is: F759H;

address of LEDC6DATA2 is: F75AH; the address of LEDC6DATA3 is: F75BH;

Bit7~Bit0 SEG<8n+7:8n>: When the COM6 port is valid, SEG[8n+ 7]-SEG[8n] port data output;

1= High level;

0= Low level.

21.3.17 COM7 Corresponds To The SEG Data Register LEDC7DATAn（n=0-3）

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDC7DATAn	SEG[8n+7]	SEG[8n+6]	SEG[8n+5]	SEG[8n+4]	SEG[8n+3]	SEG[8n+2]	SEG[8n+1]	SEG[8n]
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

address of LEDC7DATA0 is: F75CH; the address of LEDC7DATA1 is: F75DH;

address of LEDC7DATA2 is: F75EH; the address of LEDC7DATA3 is: F75FH;

Bit7~Bit0 SEG<8n+7:8n>: When the COM7 port is valid, SEG[8n+ 7]-SEG[8n] port data output;

1= High level;

0= Low level.

21.3.18 SEG Port P0 Drive Current Selection Register

--	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LED_P0x+1_P0x	DRC7	DRC6	DRC5	DRC4	DRC3	DRC2	DRC1	DRC0
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

Register LED_P01_P00 address: F710H;Register LED_P03_P02 address: F711H;
Register LED_P05_P04 address: F712H;Register LED_P07_P06 address: F713H.

Bit7~Bit4 DRC<7:4>: P0x+1 Source current drive selection control bit;

0000=	0mA;	1000=	21.6mA;
0001=	2.7mA;	1001=	24.3mA;
0010=	5.4mA;	1010=	27mA;
0011=	8.1mA;	1011=	29.7mA;
0100=	10.9mA;	1100=	32.4mA;
0101=	13.5mA;	1101=	35.1mA;
0110=	16.2mA;	1110=	37.8mA;
0111=	18.9mA;	1111=	40.5mA.

Bit3~Bit0 DRC<3:0>: P0x Source current drive selection control bit;

0000=	0mA;	1000=	21.6mA;
0001=	2.7mA;	1001=	24.3mA;
0010=	5.4mA;	1010=	27mA;
0011=	8.1mA;	1011=	29.7mA;
0100=	10.9mA;	1100=	32.4mA;
0101=	13.5mA;	1101=	35.1mA;
0110=	16.2mA;	1110=	37.8mA;
0111=	18.9mA;	1111=	40.5mA.

21.3.19 SEG Port P1 Drive Current Selection Register

--	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LED_P1x+1_P1x	DRC7	DRC6	DRC5	DRC4	DRC3	DRC2	DRC1	DRC0
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

Register LED_P11_P10 address: F714H;Register LED_P13_P12 address: F715H;
Register LED_P15_P14 address: F716H;Register LED_P17_P16 address: F717H.

Bit7~Bit4 DRC<7:4>: P1x+1 Source current drive selection control bit;

0000=	0mA;	1000=	21.6mA;
0001=	2.7mA;	1001=	24.3mA;
0010=	5.4mA;	1010=	27mA;
0011=	8.1mA;	1011=	29.7mA;
0100=	10.9mA;	1100=	32.4mA;
0101=	13.5mA;	1101=	35.1mA;
0110=	16.2mA;	1110=	37.8mA;
0111=	18.9mA;	1111=	40.5mA.

Bit3~Bit0 DRC<3:0>: P1x Source current drive selection control bit;

0000=	0mA;	1000=	21.6mA;
0001=	2.7mA;	1001=	24.3mA;
0010=	5.4mA;	1010=	27mA;
0011=	8.1mA;	1011=	29.7mA;
0100=	10.9mA;	1100=	32.4mA;
0101=	13.5mA;	1101=	35.1mA;
0110=	16.2mA;	1110=	37.8mA;
0111=	18.9mA;	1111=	40.5mA.

21.3.20 SEG Port P2 Drive Current Selection Register

--	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LED_P2x+1_P2x	DRC7	DRC6	DRC5	DRC4	DRC3	DRC2	DRC1	DRC0
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

Register LED_P21_P20 address: F718H;Register LED_P23_P22 address: F719H;

Register LED_P25_P24 address: F71AH;Register LED_P27_P26 address: F71BH.

Bit7~Bit4 DRC<7:4>: P2x+1 Source current drive selection control bit;

0000=	0mA;	1000=	21.6mA;
0001=	2.7mA;	1001=	24.3mA;
0010=	5.4mA;	1010=	27mA;
0011=	8.1mA;	1011=	29.7mA;
0100=	10.9mA;	1100=	32.4mA;
0101=	13.5mA;	1101=	35.1mA;
0110=	16.2mA;	1110=	37.8mA;
0111=	18.9mA;	1111=	40.5mA.

Bit3~Bit0 DRC<3:0>: P2x Source current drive selection control bit;

0000=	0mA;	1000=	21.6mA;
0001=	2.7mA;	1001=	24.3mA;
0010=	5.4mA;	1010=	27mA;
0011=	8.1mA;	1011=	29.7mA;
0100=	10.9mA;	1100=	32.4mA;
0101=	13.5mA;	1101=	35.1mA;
0110=	16.2mA;	1110=	37.8mA;
0111=	18.9mA;	1111=	40.5mA.

21.3.21 SEG Port P3 Drive Current Selection Register

--	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LED_P3x+1_P3x	DRC7	DRC6	DRC5	DRC4	DRC3	DRC2	DRC1	DRC0
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

Register LED_P31_P30 address: F71CH;Register LED_P33_P32 address: F71DH.

Bit7~Bit4 DRC<7:4>: P3x+1 Source current drive selection control bit;

0000=	0mA;	1000=	21.6mA;
0001=	2.7mA;	1001=	24.3mA;
0010=	5.4mA;	1010=	27mA;
0011=	8.1mA;	1011=	29.7mA;
0100=	10.9mA;	1100=	32.4mA;
0101=	13.5mA;	1101=	35.1mA;
0110=	16.2mA;	1110=	37.8mA;
0111=	18.9mA;	1111=	40.5mA.

Bit3~Bit0 DRC<3:0>: P3x Source current drive selection control bit;

0000=	0mA;	1000=	21.6mA;
0001=	2.7mA;	1001=	24.3mA;
0010=	5.4mA;	1010=	27mA;
0011=	8.1mA;	1011=	29.7mA;
0100=	10.9mA;	1100=	32.4mA;
0101=	13.5mA;	1101=	35.1mA;
0110=	16.2mA;	1110=	37.8mA;
0111=	18.9mA;	1111=	40.5mA.

21.3.22 LED_SEG Port Source Current Drive Enable Register

F71EH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LED_SEG_CUREN	MODE7	MODE6	MODE5	MODE4	MODE3	MODE2	MODE1	MODE0
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 MODE<7:0>: LED_SEG port source current drive enable;

0x5A= The drive current of the SEG port is individually controllable by the register;

Other value= Invalid

21.3.23 P3 Port Drive Current Control Register P3DR

F03CH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P3DR	P3DR7	P3DR6	P3DR5	P3DR4	P3DR3	P3DR2	P3DR1	P3DR0
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 P3DR7: P37 drive current selection;
1= 150mA;
0= 50mA.

Bit6 P3DR6: P36 drive current selection;
1= 150mA;
0= 50mA.

Bit5 P3DR5: P35 drive current selection;
1= 150mA;
0= 50mA.

Bit4 P3DR4: P34 drive current selection;
1= 150mA;
0= 50mA.

Bit3 P3DR3: P33 drive current selection;
1= 150mA;
0= 50mA.

Bit2 P3DR2: P32 drive current selection;
1= 150mA;
0= 50mA.

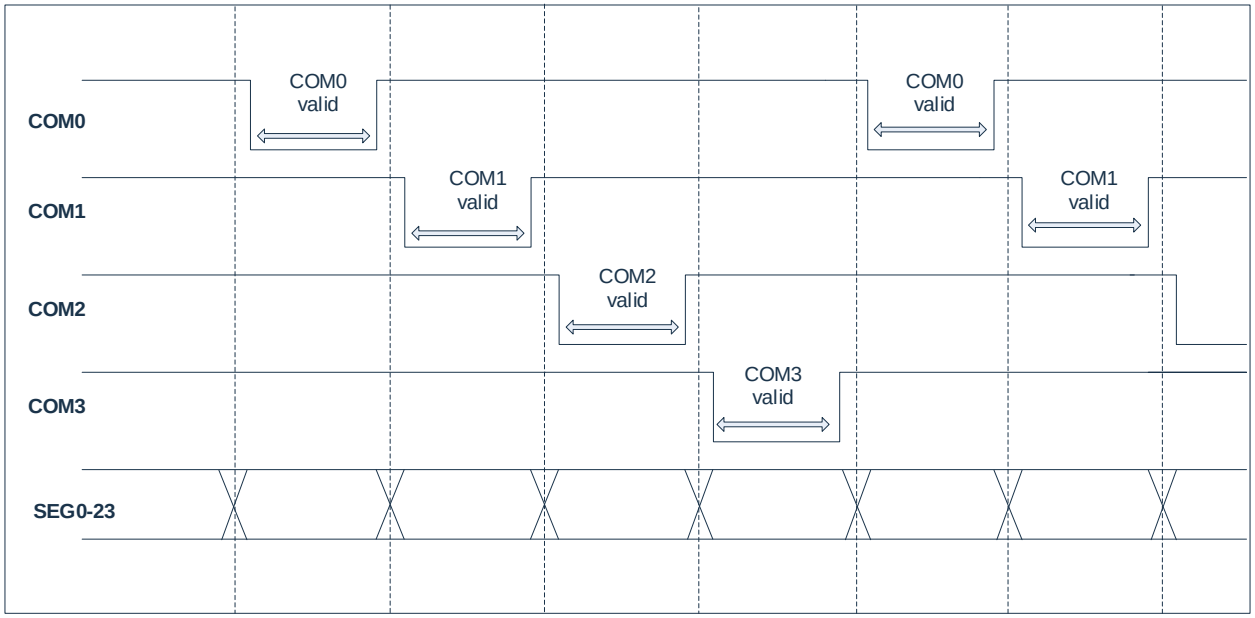
Bit1 P3DR1: P31 drive current selection;
1= 150mA;
0= 50mA.

Bit0 P3DR0: P30 drive current selection;
1= 150mA;
0= 50mA.

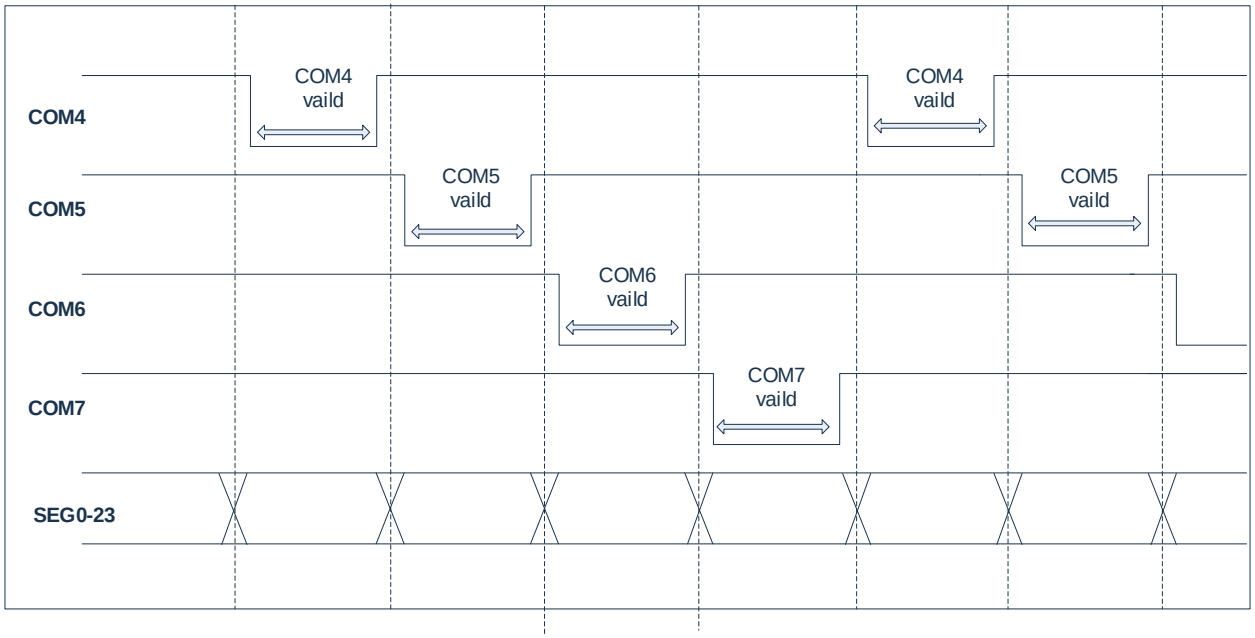
21.4 LED Driver Output Waveform

According to the related configuration registers of the LED driver, the corresponding LED driver output waveform can be set. Take the LED configuration as 1/4 DUTY as an example, set the LED common cathode drive and common anode drive respectively, when the output mode of the COM port is different, the waveform diagram is as shown in the figure below.

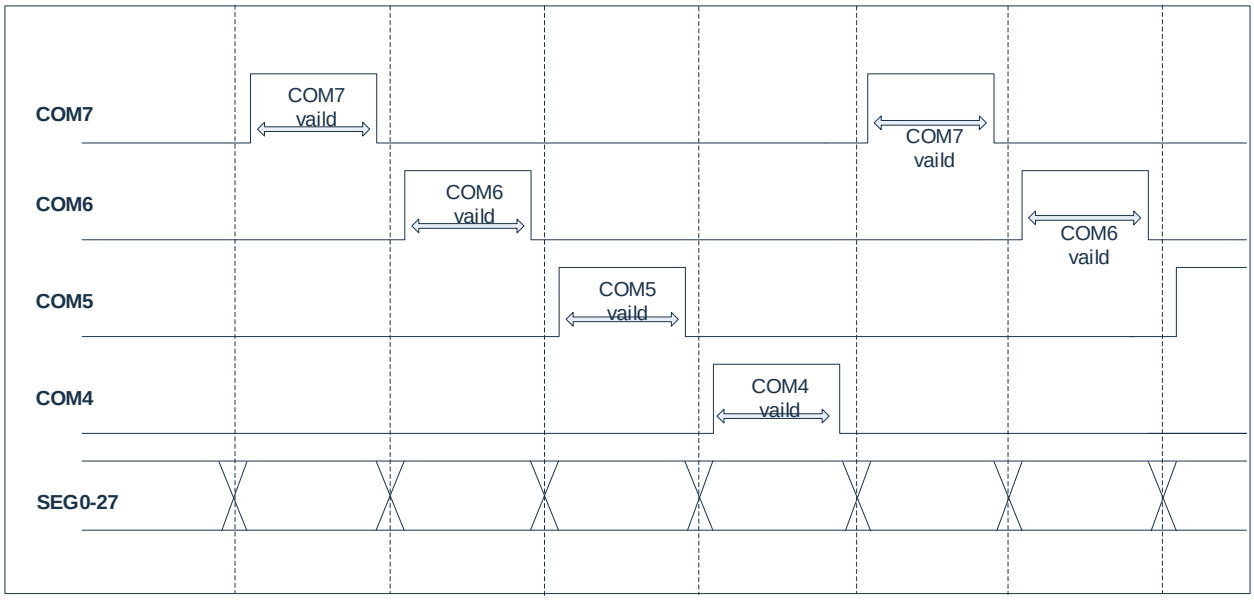
1) LED configuration 1/4DUTY, common cathode drive mode, COM_SEL=0, MODE0 mode, the waveform is shown in the figure below:



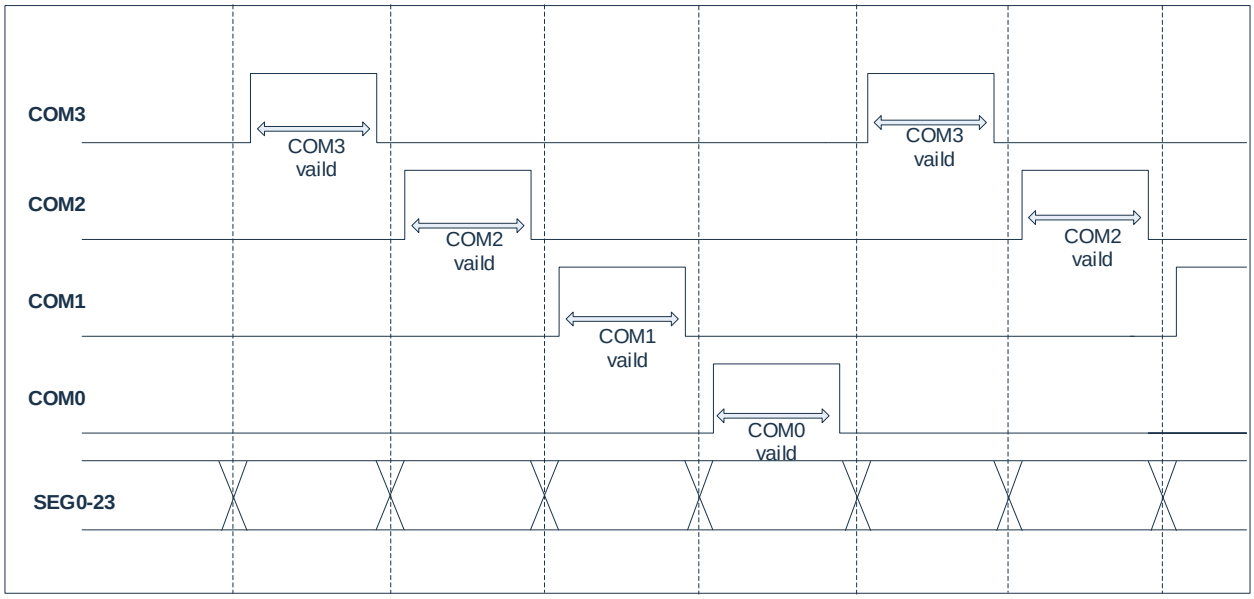
2) LED configuration 1/4DUTY, common cathode drive mode, COM_SEL=1, MODE0 mode, the waveform is shown in the figure below:



3) LED configuration 1/4DUTY, were positive drive mode, COM_SEL = 0, MODE1 mode, the waveform as shown below:



4) LED configuration 1/4DUTY, were positive drive mode, COM_SEL = 1, MODE1 mode, the waveform as shown below:



22. SPI Module

22.1 Overview

This SPI is a fully configurable SPI master/slave device, allowing users to configure the polarity and phase of the serial clock signal SCLK. The serial clock line (SCLK) is synchronized with the shift and sampling of information on two independent serial data lines, and SPI data is sent and received at the same time. SPI allows MCU to communicate with serial peripherals. It can also communicate between processors in a multi-host system. It is a technology-independent design that can be implemented in various process technologies.

The SPI system is flexible enough to directly connect with many standard product peripherals from multiple manufacturers. In order to accommodate most of the available synchronous serial peripherals, the clock control logic allows the clock polarity and phase to be selected. The system can be configured as a master device or a slave device. When the SPI is configured as a master device, the software selects one of eight different bit rates for the serial clock, up to the system clock divided by 4（Fsys/4）.

The SPI slave chip is selected to address the SPI slave device to exchange serial data. When SPI is used as a master device, SPI is automatically driven by the slave selection control register SSCR. The SPI controller includes logic error detection to support inter-processor communication. For example, the write conflict detector can indicate when to write data to the serial shift register during transmission.

SPI has the following characteristics:

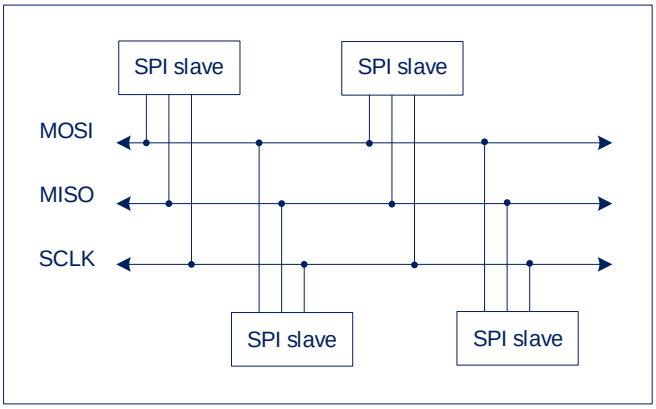
- Full-duplex synchronous serial data transmission.
- Support master/slave mode.
- Support multi-host system.
- System error detection.
- Generated the interrupt.
- Support speed up to 1/4 of the system clock (FSYS≤24MHz).
- The bit rate generates 1/4、1/8、1/16、1/32、1/64、1/128、1/256、1/512.
- Supports four transmission formats.
- The simple interface allows easy connection to the microcontroller.

22.2 SPI Port Configuration

To use the SPI function, you need to configure the relevant port as an SPI channel, and select the corresponding port input through the communication input port register. For example, configure P14, P15, P16, and P17 as SPI communication ports. The configuration code is as follows:

```
P14CFG = 0x0E;      // Select P14 as the NSS channel of SPI
P15CFG = 0x0F;      // Select P15 as the SCLK channel of SPI
P16CFG = 0x10;      // Select P16 as the MOSI channel of SPI
P17CFG = 0x11;      // Select P17 as the MISO channel of SPI
```

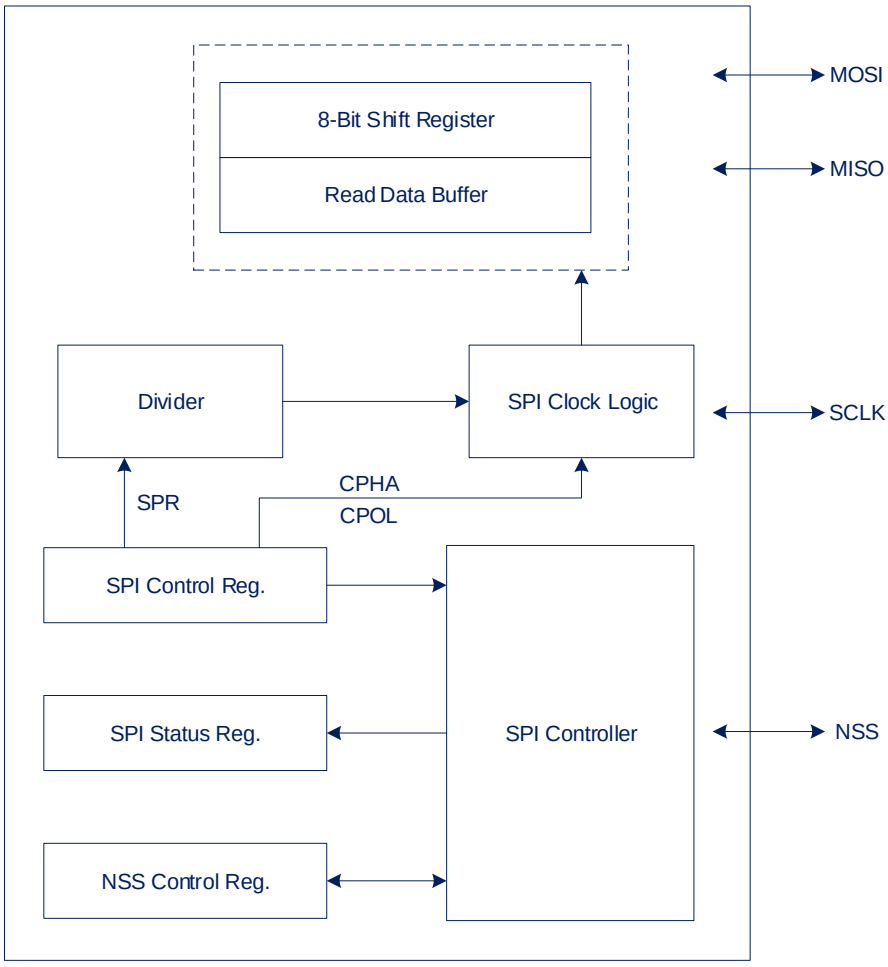
Configured as SCLK, MOSI, MISO and NSS ports, the pull-up resistor and open-drain output switch are forcibly closed. The schematic diagram of the multi-machine SPI communication structure is shown in the figure below:



22.3 SPI Hardware Description

When SPI transmission occurs, when one data pin is shifted out of an 8-bit character, another data pin is shifted into other 8-bit characters. The 8-bit shift register in the master device and the other 8-bit shift register in the slave device are connected as a circular 16-bit shift register. When a transfer occurs, the distributed shift register is shifted by 8 bits, which is effective The characters of the master and slave are exchanged.

The central element in the SPI system is a module containing a shift register and a buffer for reading data. The system has a single buffer in the sending direction and a double buffer in the receiving direction. This means that new data cannot be written to the shifter until the previous data is transmitted; however, the received data is transferred to the parallel read data buffer, so the shifter can freely receive the second serial character. As long as the first character is read from the read data buffer before the next serial character is ready for transmission, there will be no overwriting. The SPI control block diagram is shown in the figure below:



The pins associated with SPI are: NSS，SCLK，MOSI，MISO.

The NSS output pin in the master mode is used to select the slave device, and the NSS input pin in the slave mode is used to enable transmission.

In master mode, the SCLK pin is used as the SPI clock signal reference. When the host device starts the transfer, eight clock cycles are automatically generated on the SCLK pin.

When SPI is configured as a slave device, the SI pin is the input data line of the slave device, and SO is the output data line of the slave device.

When SPI is configured as a master device, MI pin is the input data line of the master device, and MO is the output data line of the master device.

22.4 SPI Related Registers

22.4.1 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: Bit [2] of SPI clock frequency selection bit.

Bit4 MSTR: SPI mode selection bit;

1= Master mode;

0= Slave mode.

Bit3 CPOL: SPI clock polarity selection bit;

1= High when SCLK is idle;

0= Low when SCLK is idle.

Bit2 CPHA: SPI clock phase selection bit.

Bit1~Bit0 SPR<1:0>: SPI clock frequency selection bits[1:0]

(For frequency control, see the table below)

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

22.4.2 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: SPI data sent or received .

Write: Write the data to be sent (the sending order is from high to low).

Read: Data has been received.

22.4.3 SPI Slave Device Selection Control Register SSCR

The slave device selection control register SSCR can be read or written at any time. It is used to configure which slave device selection output should be driven when confirming the SPI master transmission. When the SPI master transfer starts, the contents of the SSCR register will be automatically assigned to the NSS pin.

0xEF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SSCR	--	--	--	--	--	--	--	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~Bit1 -- Reserved, all must be 0.

Bit0 NSSO0: SPI slave device selection control bit（(Master chip select output NSS is NSSO0).

0= NSSO0 outputs 0 when the SPI master transfer starts.

1= NSSO0 outputs 1 when the SPI master transfer starts.

22.4.4 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 completed interrupt flag bit, read-only;

1= SPI transmission completed (first read SPSR, and then read/write SPDR to clear it);

0= SPI transmission is not completed.

Bit6 WCOL: SPI write conflict interrupt flag bit, read only;

1= Write SPDR operation conflict occurs when SPI transmission is not completed (first read SPSR, and then read/write SPDR to clear it);

0= No write conflict.

Bit5~Bit1 -- Reserved, all must be 0.

Bit0 SSCEN: SPI master mode NSS output control bit.

1= SPI is idle state, NSS outputs high level;

0= NSS outputs the content of register SSCR.

The SPI status register (SPSR) contains a flag indicating the completion of the transfer or the occurrence of a system error. When the corresponding event occurs and is cleared by software in sequence, all the flags will be automatically set. By reading SPSR and then accessing SPDR, SPISIF and WCOL will be cleared automatically.

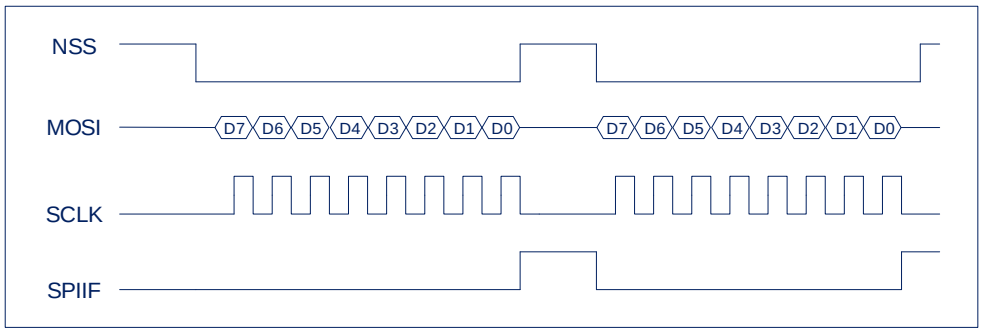
The SSCEN bit is the enable bit for automatic slave selection output. When SSCEN is set to 1, the NSS line outputs the contents of the SSCR register when the transmission is in progress, and the NSS is high when the transmission is idle. When the SSCEN bit is cleared, the NSS line always displays the contents of the SSCR register.

22.5 SPI Master Mode

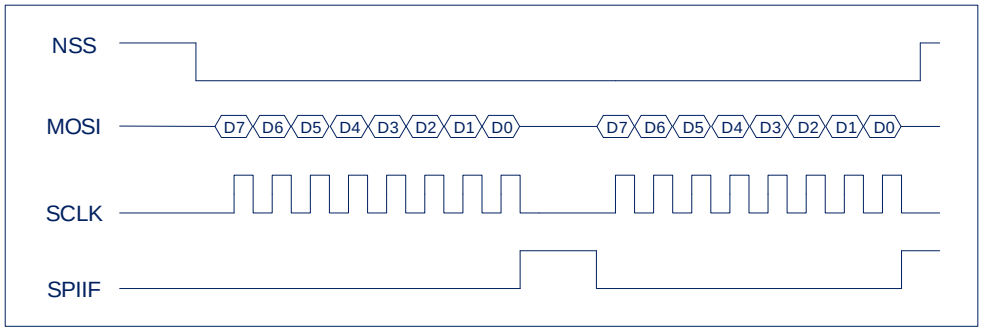
When SPI is configured as master mode, transfer is started by writing to the SPDR register. When a new byte is written to the SPDR register, the SPI starts to transfer. The serial clock SCLK is generated by SPI. In the master mode, SPI is enabled and SCLK is output.

SPI in master mode can select SPI slave device through NSS line. NSS line-the slave select output line is loaded with the contents of the SSCR register. The SSCEN bit of the SPSR register selects between automatic NSS line control and software control. Set SSCEN in the master mode. When SSCEN is set to 1, the NSS line outputs the contents of the SSCR register when the transfer is in progress, and NSS is high when the transfer is idle. When the SSCEN bit is cleared, the NSS line is controlled by software and always displays the contents of the SSCR register, regardless of whether the transfer is in progress or the SPI is in an idle state.

When SSCEN=1, configure the SPI clock polarity CPOL=0, clock phase CPHA=0, and use the slave selection line as shown in the following figure:



When SSCEN=0, configure the SPI clock polarity CPOL=0, clock phase CPHA=0, the slave selection line is used as shown in the figure below:



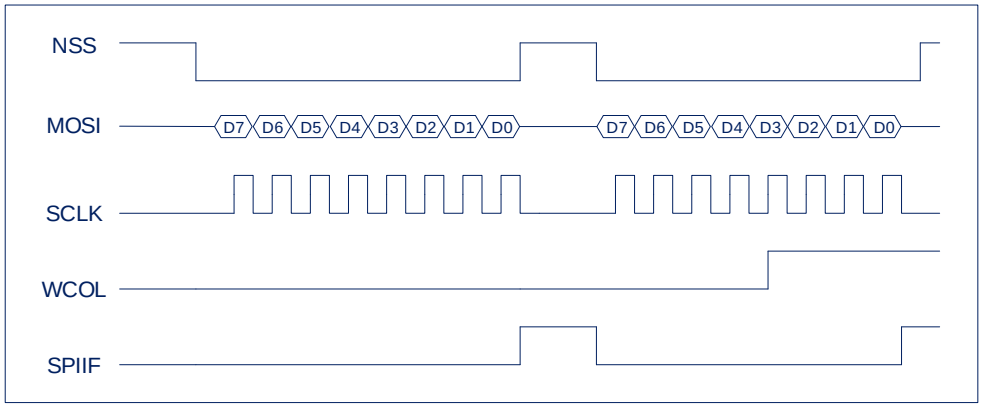
22.5.1 Write Conflict Error

If the SPI data register is written during transmission, a write conflict will occur. Transmission continues uninterrupted, and the write data that caused the error will not be 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 hardware. To clear the WCOL bit, 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 control mode, the write conflict error when the SPI clock polarity CPOL=0 and the clock phase CPHA=0 are configured as shown in the figure below:



The specific conditions for the write conflict are: during data transmission, when NSS is low, the first from the time when a data is sent to the eighth falling edge of SCLK, if SPDR is written during this period, a write conflict will occur and WCOL will be set to 1.

Note: When starting to send data, after writing SPDR, NSS does not change to low level immediately, it needs to wait at most one SPI clock before starting to be low. After NSS is low, it needs to wait for a system clock to start sending the first data, and then it enters the real data transmission state. During the period from writing SPDR to entering the real data transmission state, writing to SPDR again does not produce a write conflict. But this operation will update the data to be sent. If there are multiple write operations to SPDR, the data sent will be the last value written to SPDR.

Since SPI has only one transmit buffer, it is recommended to determine whether the last data has been sent before writing SPDR, and write the SPDR register after confirming that the transmission is completed to prevent write conflicts.

22.6 SPI Slave Mode

When configured as an SPI slave device, SPI transmission is initiated by the external SPI master module by using the SPI slave selection input and generates the SCLK serial clock.

Before the transmission starts, it is necessary to determine which SPI slave will be used to exchange data. NSS is used (clear = 0), the clock signal connected to the SCLK line will cause the SPI slave device to transfer the contents of the receive shift register of the MOSI line, and drive the MISO line with the contents of the transmitter shift register. When all 8 bits are shifted in/out, the 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 transmission error-write conflict error.

22.6.1 Addressed Error

In slave mode, only write conflict errors can be detected by SPI.

When the SPDR register write operation is performed while the SPI transfer is in progress, a write conflict error will occur.

In slave mode, when CPHA is cleared, as long as the NSS slave select line is driven low, even if all bits have been transmitted, a write conflict error may occur. This is because the start of the transfer is not explicitly specified, and the NSS being driven low after the full byte transfer may indicate the start of the next byte transfer.

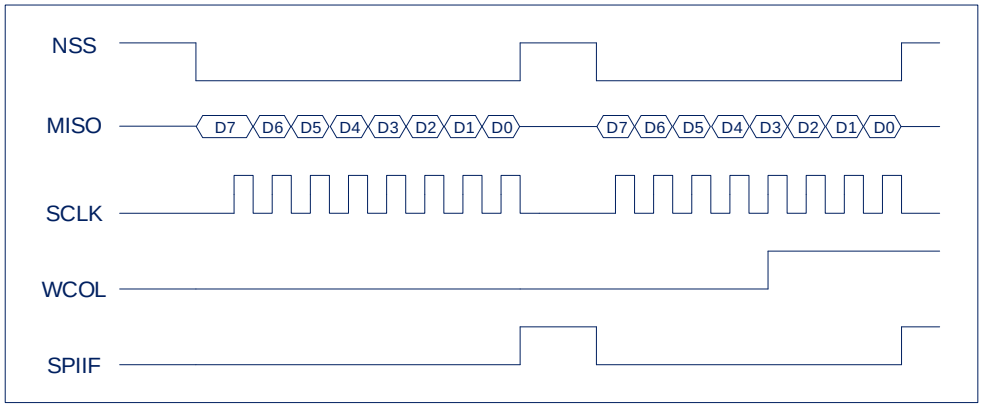
22.6.2 Write Conflict Error

If you write to the SPI data register during transmission, a write conflict will occur. Transmission continues uninterrupted, and the write data that caused the error will not be written to the shifter. Write conflicts are indicated by the WCOL flag in the SPSR register.

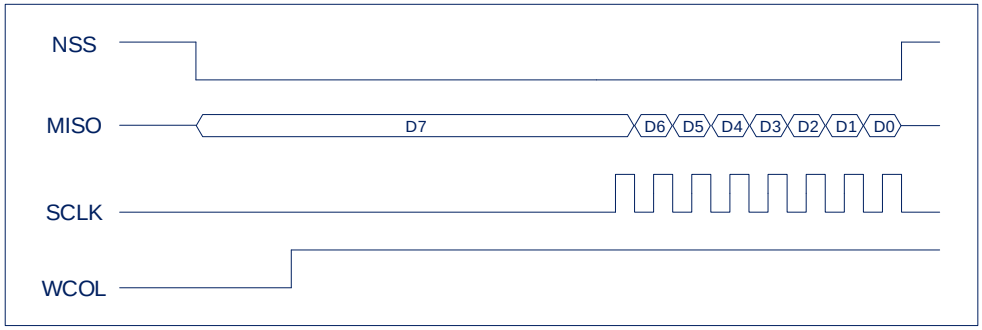
When a WCOL error occurs, the WCOL flag is automatically set to 1 by hardware. To clear the WCOL bit, the user should perform the following sequence:

- read the contents of the SPSR register;
- access the SPDR register (read or write).

The write conflict error during transmission in SPI slave mode is shown in the following figure:



In case CPHA is cleared, WCOL generation can also be caused by writing to the SPDR register when any NSS line is cleared. At this time, the SPI master does not generate the serial clock SCLK. can be completed. This is because the start of the transfer is not explicitly specified, and the NSS being driven low after the 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 conflict error as shown in the following figure:



In addition, after writing SPDR in the slave mode, the NSS controlled by the master does not immediately become low. When NSS is low, you need to wait for the second edge of SCLK to start before entering the real data transmission state.

During the period from writing SPDR to the beginning of sending the first data, writing to SPDR again will not cause a write conflict. But this operation will update the data to be sent. If there are multiple write operations to SPDR, the data sent will be the last value written to SPDR.

During the period when the first data is sent to the second edge of SCLK, writing SPDR again will not cause a write conflict, nor will it update the data being sent. That is, the operation of writing SPDR this time is ignored.

Since SPI has only one transmit buffer, it is recommended to determine whether the last data has been sent before writing SPDR, and write the SPDR register after confirming that the transmission is completed to prevent write conflicts.

22.7 SPI Clock Control Logic

22.7.1 SPI Clock Phase And Polarity Control

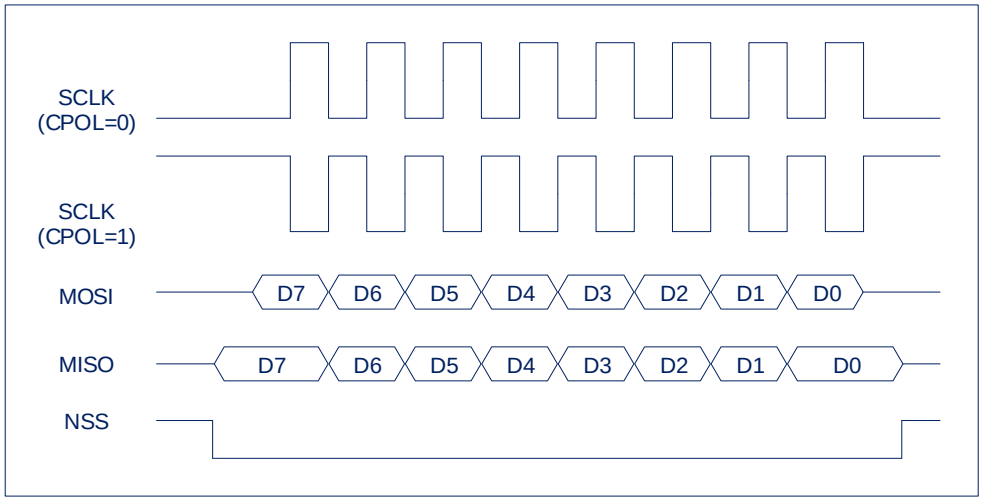
The software can select any of the four combinations using two control bits (phase and polarity of the serial clock SCLK) in the SPI control register (SPCR). The clock polarity is specified by the CPOL control bit. When the transmission is idle, the CPOL control bit selecting high or low level has no significant effect on the transmission format. The clock phase (CPHA) control bit selects one of two fundamentally different transmission formats. The clock phase and polarity of the host 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 peripheral slaves with different requirements. The flexibility of the SPI system allows direct connection with almost all existing synchronous serial peripherals.

22.7.2 SPI Transmission Format

During SPI transmission, data is simultaneously sent (serial shift out) and received (serial shift in). The serial clock line is synchronized with the shift and sampling of the two serial data lines. The slave select line allows individual selection of slave SPI devices; unselected slave devices will not interfere with SPI bus activity. On the SPI master device, the slave select line can be selectively used to indicate multi-master bus contention.

22.7.3 CPHA=0 Transmission Format

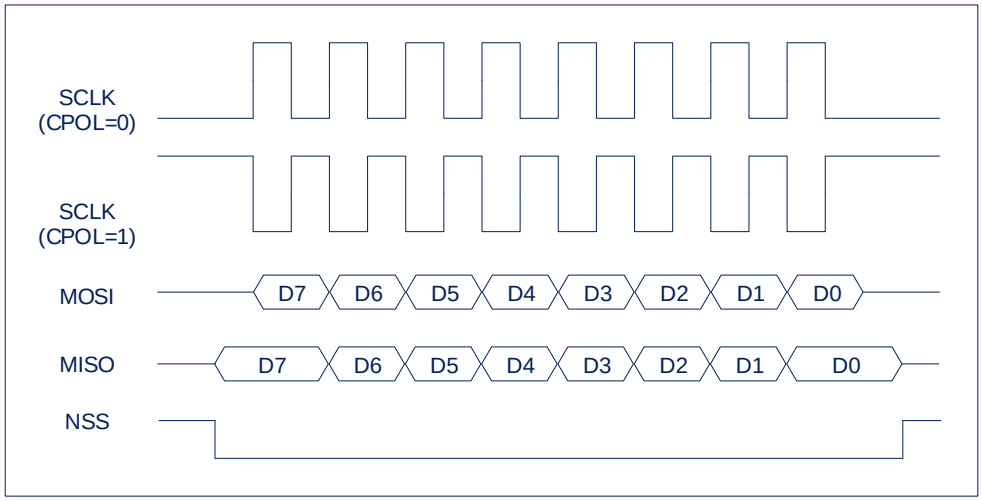
The following figure shows the timing diagram of SPI transmission with. SCLK shows two waveforms: one for CPOL equal to 0 and the other for CPOL equal to 1. The figure can be described as a master device or slave device timing diagram through SCLK. The master input/slave output (MISO) and master output/slave input (MOSI) pins are directly connected between the host and the slave. The MISO signal is the slave output, and the MOSI signal is the master output. The NSS line is the slave selection input of the slave; the NSS pin of the master is not shown, but it is assumed to be invalid. The NSS pin of the host must be high. This sequence diagram functionally describes how to transmit; it should not be used as a substitute for data sheet parameter information.



When CPHA=0, the NSS line must be de-set and reset between each successive serial byte. In addition, if the slave writes data to the SPI data register (SPDR) when NSS is at low level, a write conflict error will occur. When CPHA = 1, the NSS line may remain low between consecutive transmissions (it can always remain low). In a system with a single fixed master and a single slave driving the MISO data line, this format is sometimes preferred.

22.7.4 CPHA=1 Transmission Format

The following figure is the timing diagram of SPI transmission with CPHA=1. SCLK displays two waveforms: one for CPOL=0 and the other for CPOL=1. Since the SCLK, MISO, and MOSI pins are directly connected between the master and the slave, this diagram can be interpreted as a master or slave timing diagram. The MISO signal is the slave output, and the MOSI signal is the master output. The NSS line is the slave selection input of the slave; the NSS pin of the master is not shown, but it is assumed to be invalid. The NSS pin of the host must be high or must be reconfigured as a general-purpose output that does not affect the SPI.



22.8 SPI Data Transmission

22.8.1 SPI Transmission Start

All SPI transmissions are started and controlled by the master SPI device. As a slave device, SPI will consider the transmission start at the first SCLK edge or the falling edge of NSS according to the selected CPHA format. When CPHA = 0, the falling edge of NSS indicates the beginning of the transfer. When CPHA = 1, the first edge on SCLK indicates the beginning of the transfer. No matter which CPHA mode, the transmission can be aborted by making the NSS line high, but it will reset the SPI slave logic and counter. The selected SCLK rate has no effect on the operation of the slave, because the master's clock is controlling the transfer.

When the SPI is configured as a master, the transfer is initiated by software written to SPDR.

22.8.2 SPI Transmission Ends

When the SPIF flag is set to 1, the SPI transmission 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 time of the end period, only the fastest rate is considered in the discussion of the end period. When SPI is configured as a master, SPIF is set at the end of the eighth SCLK cycle. When CPHA is equal to 1, SCLK is inactive during the last half of the eighth SCLK cycle.

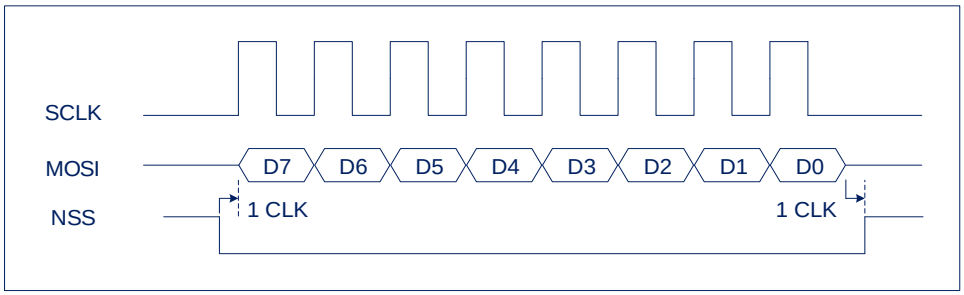
Because the SCLK line can be asynchronous with the MCU clock of the slave, and the slave cannot access as much information as the master can access the SCLK cycle, so when the SPI runs as a slave, the end cycle is different. For example, when CPHA = 1, the last SCLK edge occurs in the middle of the eighth SCLK cycle, and the slave cannot know when the last SCLK cycle ends. For these reasons, the slave considers that the transmission is complete after the last bit of the serial data is sampled, which corresponds to the middle of the eighth SCLK cycle.

The SPIF flag is set at the end of the transmission, but when the NSS line is still low, the slave is not allowed to write new data into the SPDR.

22.9 SPI Timing Diagram

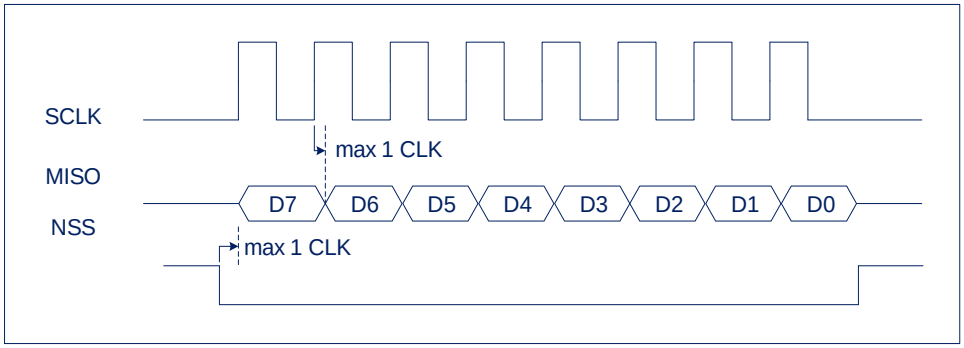
22.9.1 Master Mode Transmission

When the SPI clock polarity CPOL=0, clock phase CPHA=1, in the SPI master mode, after NSS is low, a system clock CLK, MOSI starts to output, and MOSI data is clocked in SCLK The rising edge of the output. The timing diagram of master mode is shown in the following figure:



22.9.2 Slave Mode Transmission

When the SPI clock polarity CPOL=0 and clock phase CPHA=1, the data on MISO starts to output after the falling edge of the NSS line. The max difference between the MSIO data output and the falling edge of NSS is 1 system clock CLK. The timing diagram of the slave mode is shown in the following figure:



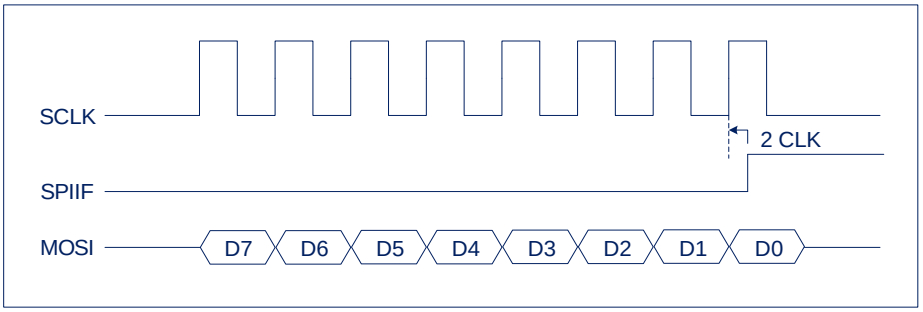
22.10 SPI Interrupt

The interrupt number of SPI is 22, and its interrupt vector is 0x00B3. To enable the SPI interrupt, the enable bit SPIIE must be set to 1, and the overall interrupt enable bit EA must be set to 1.

If the SPI related interrupt enable is turned on and the SPI general interrupt indicator bit SPIIF=1, the CPU will enter the interrupt service routine. The SPIIF operation attribute is read-only and has nothing to do with the state of SPIIE.

After any one of the transmission completion flag SPISIF and write conflict WCOL in the SPI status register SPSR is set to 1, the SPI total interrupt indicator bit SPIIF will be set to 1. Only when these 3 flag bits are all 0, SPIIF is automatically cleared to 0.

When the SPI clock polarity CPOL=0 and the clock phase CPHA=1, in the SPI master mode, SPIIF will generate 2 system clocks CLK after the 8th SCLK rising edge of each frame of data. The timing diagram is shown in the figure below:



22.10.1 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 interrupt;
0= Disable SPI interrupt.
Bit6 I2CIE: I^2C interrupt enable bit;
1= Enable I^2C interrupt;
0= Disable I^2C interrupt.
Bit5 WDTIE: WDT interrupt enable bit;
1= Enable WDT overflow interrupt;
0= Disable WDT overflow interrupt.
Bit4 ADCIE ADC interrupt enable bit;
1= Enable ADC interrupt;
0= Disable ADC interrupt.
Bit3 PWMIE: PWM total interrupt enable bit;
1= Allow all PWM interrupts;
0= Disable all PWM interrupts.
Bit2 -- Reserved, must be zero.
Bit1 ET4: Timer4 interrupt enable bit;
1= Enable Timer4 interrupt;
0= Disable Timer4 interrupt.
Bit0 ET3: Timer3 interrupt enable bit;
1= Enable Timer3 interrupt;
0= Disable Timer3 interrupt.

22.10.2 Interrupt Priority Control Register EIP2

0xB8	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 as high-level interrupt;
0= Set as low-level interrupt.

Bit6 PI2C: I^2C interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.

Bit5 PWDT: WDT interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.

Bit4 PADC: ADC interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.

Bit3 PPWM: PWM interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.

Bit2 --: Reserved, must be zero.

Bit1 PT4: TIMER4 interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.

Bit0 PT3: TIMER3 interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.

22.10.3 Peripheral Interrupt Flag 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 general interrupt indicator bit, read-only;
1= SPI generates an interrupt (this bit is automatically cleared after clearing the specific interrupt flag bit);
0= SPI does not generate an interrupt.

Bit6 I2CIF: I^2C general interrupt indicator bit, read-only;
1= I^2C generates an interrupt (this bit is automatically cleared after clearing the specific interrupt flag bit);
0= I^2C does not generate an interrupt.

Bit5 --: Reserved, must be 0.

Bit4 ADCIF: ADC interrupt flag bit;
1= ADC conversion is completed and needs to be cleared by software;
0= ADC conversion is not completed.

Bit3 PWMIF: PWM general interrupt indicator bit, read-only;
1= PWM generates an interrupt, (this bit is automatically cleared after clearing the specific interrupt flag bit);
0= PWM does not generate an interrupt.

Bit2 --: Reserved, must be 0.

Bit1 TF4: Timer4 timer overflow interrupt flag bit;
1= Timer4 timer overflows, it is automatically cleared by hardware when entering the interrupt service routine, or can be cleared by software;
0= Timer4 timer has no overflow.

Bit0 TF3: Timer3 timer overflow interrupt flag bit;
1= Timer3 timer overflows, it is automatically cleared by hardware when entering the interrupt service routine, or it can be cleared by software;
0= Timer3 timer has no overflow.

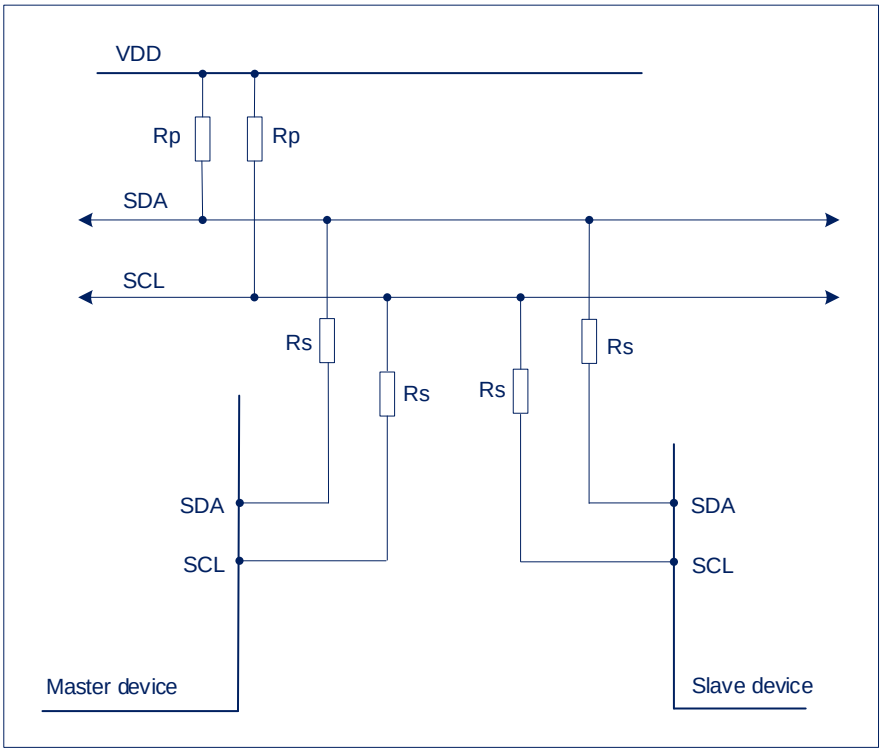
23. I²C Module

23.1 Overview

This module provides the interface between the microcontroller and the I²C bus. The connection diagram is shown in the figure below, and supports arbitration and clock synchronization to allow operation in a multi-master system. I²C supports normal and fast modes.

The I2C module has the following characteristics：

- supports 4 working modes: master sending, master receiving, slave sending, and slave receiving.
- Support 2 transmission speed modes:
  - standard (up to 100Kb/s);
  - fast (up to 400Kb/s);
- perform arbitration and clock synchronization.
- Support multi-master system.
- Host mode supports 7-bit addressing mode and 10-bit addressing mode on I2C bus (software support).
- Slave mode supports 7-bit addressing mode on I2C bus.
- Interrupt generation.
- Allows operation in a wide range of clock frequencies (built-in 8-bit timer).



23.2 I²C Port Configuration

If the I²C function is used, the corresponding port should first be configured as SCL and SDA channels. For example, configure the P04 and P05 ports as I²C functions:

```
PS_SCL = 0x04;     // Select port P04 as the SCL pin
PS_SDA = 0x05;     / select port P05 as the SDA pin
P04CFG = 0x04;     // P04 multiplexes the SCL function
P05CFG = 0x04;     // P05 multiplexes the SDA function
```

After configuring the I2C channel, this group of ports is in an open-drain state by default. You can configure whether to enable SCL, the internal pull-up resistor of the SDA port through PxUP, or add a pull-up resistor outside the chip.

In the master control mode, the IIC outputs SCL to the slave. After sending the address or data, the slave needs to pull down SCL and send back the corresponding response signal to the master. The host needs to read back the state of the SCL port line to check whether the slave has released the SCL to determine whether it needs to send the next frame of data. If the pull-up resistor or board-level parasitic capacitance connected to SCL is larger, the readback time will be longer, which will affect the communication speed of IIC. For details, please refer to the IIC application manual.

23.3 I²C Master Control Mode

There are six registers for connecting with the master: control, status, slave address, send data, receive data and timer period register.

Register		address
write	read	
slave address register I2CMSA	slave address register I2CMSA	0xF4
master mode control register I2CMCR	master mode status register I2CMSR	0xF5
master control sending data register I2CMBUF	master receiving data register I2CMBUF	0xF6
timing cycle register I2CMTP	timing cycle register I2CMTP	0xF7

Master mode control register I2CMCR and the master mode status register I2CMSR share a register address, but they are physically two different registers.

The master control sending data register and the master control receiving data register share a register address. The write operation accesses the sending register I2CMBUF, and the read operation accesses the receiving register I2CMBUF.

During write operation, it is used as a control register to write, and read operation is used as a status register to read.

23.3.1 I²C Master Mode Timing Period Register

To generate a wide range of SCL frequencies, the module has a built-in 8-bit timer. for standard and fast transfers.

When TIMER_PRD ≠ 0，ideal clock period for SCL:2* (1+TIMER_PRD)* 10* Tsys

When TIMER_PRD ＝ 0，ideal clock period for SCL:：3* 10* Tsys

For the specific calculation formula of SCL, please refer to the IIC application manual.

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, all must be 0.
Bit6~Bit0	MTP<6:0>:	Bit6-bit0 of the period timer register of standard and fast mode: TIMER_PRD[6:0].

23.3.2 I²C Master Mode Control And Status Register

Control register includes 4 bits: RUN, START, STOP, and ACK bits. The START bit will generate a START or REPEATED START condition. The STOP bit determines whether the data transmission stops at the end of the cycle or continues. In order to generate a single transmission cycle, the slave address register writes the required address, the R/S bit is set to 0, and the control register writes ACK=x, STOP=1, START=1, RUN=1 (I2CMCR=xxx0_x111x) to perform operations and stop. When the operation is completed (or an error occurs), an interrupt is generated. Data can be read from the receive data register.

When I²C works in master mode, the ACK bit must be set to 1. This will cause the I²C bus controller to automatically send an answer after each byte. When the I²C bus controller no longer needs the slave to send data, this bit must be cleared to 0.

Master control mode control register

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 control module (I2C registers of the entire master control module, including I2CMSR);
0= I²C clear the interrupt flag bit in master control mode.
Bit6~Bit5 -- Reserved.
Bit4 -- Reserved, must be 0.
Bit3 ACK: Acknowledge enable bit;
1= Enable;
0= Disable.
Bit2 STOP: Stop enable bit;
1= Enable;
0= Disable.
Bit1 START: Start enable bit;
1= Enable;
0= Disable.
Bit0 RUN: Run enable bit;
1= Enable;
0= Disable.

Various operations in the master control mode can be realized through the following control bit combination list:

START：Send start signal.

SEND：Send data or address.

RECEIVE：Receive data.

STOP：Send end signal.

Combination of control bits (IDLE state)

R/S	ACK	STOP	START	RUN	OPERATION
0	-	0	1	1	START followed by SEND (the master stays in the sending mode)
0	-	1	1	1	START followed by SEND and STOP
1	0	0	1	1	Receiving after START adopts response (the master stays in receiver mode)
1	0	1	1	1	START followed by RECEIVE and STOP
1	1	0	1	1	START followed by RECEIVE (master unit remains in receiver mode)
1	1	1	1	1	Forbidden combination
0	0	0	0	1	Forbidden combination

1. Combination of control bits (master send state)

R/S	ACK	STOP	START	RUN	OPERATION
-	-	0	0	1	SEND operation
-	-	1	0	0	Stop
-	-	1	0	1	SEND 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 RECAIVE operation (The master stays in receiver mode)
1	0	1	1	1	Repeat START, followed by SEND and STOP conditions
1	1	0	1	1	Repeat START condition followed by RECEIVE (The master stays in receiver mode)
1	1	1	1	1	Forbidden combination

Combination of control bits (master control receive state)

R/S	ACK	STOP	START	RUN	OPERATION
-	0	0	0	1	RECEIVE operation with response (The master stays in receiver mode)
-	-	1	0	0	STOP
-	0	1	0	1	RECEIVE followed by STOP
-	1	0	0	1	RECEIVE operation (main unit remains in receiver mode)
-	1	1	0	1	Forbidden combination
1	0	0	1	1	Repeat the START, followed by the response RECEIVE operation (The master stays 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 (The master stays in receiver mode)
0	-	0	1	1	Repeat START followed by SEND (The master stays 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: Interrupt flag bit in I2C master mode;
1= In master mode, send/receive is completed or a transmission error occurs. (software clear, write 0 to clear);
0= No interrupt is generated.

Bit6 BUS_BUSY: I2C bus busy flag in master mode/slave mode;
1= The I2C bus is busy and cannot make a transfer (set by a start bit on the bus, cleared by a stop condition).
0= --

Bit5 IDLE: I^2C master control mode idle flag bit;
1= Idle state;
0= Working state.

Bit4 ARB_LOST: I^2C master mode arbitration flag;
1= Lost bus control.
0= -

Bit3 DATA_ACK: I^2C master mode sending data acknowledgement flag bit;
1= Last time there is no response to the data sent.
0= -

Bit2 ADD_ACK: I^2C master mode addressing acknowledgement flag bit;
1= There is no acknowledgement in the last addressing.
0= -

Bit1 ERROR: I^2C master mode error flag;
1= No response from the addressed slave/No response to data sent/I^2C bus arbitration conflict.
0= -

Bit0 BUSY: I^2C main control module busy flag bit;
1= I^2C module is transmitting data.
0= --

23.3.3 I²C Slave Address Register

The slave address register is composed of 8 bits: 7-bit address bits (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 SA<6:0>: Slave address in I2C master mode.

Bit0 R/S: Receive/send state selection bit after sending slave address in I2C master mode;

1= Receive data after correct addressing;

0= Send data after correct addressing.

23.3.4 I²C Master Mode Sending And Receiving Data Register

The sending data register consists of eight data bits, which will be sent on the bus during the next sending or burst sending operation. The first sending bit is MD7 (MSB).

Master mode data buffer 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.

23.4 I²C Slave Mode

There are five registers for connecting to the target device: own 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

23.4.1 I²C Self Address Register I2CSADR

The self address register consists of seven address bits that identify the I2C core on the I2C bus. This register can read and write addresses.

Self 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 SA<6:0>: I²C self address of slave mode.

23.4.2 I²C Slave Mode Control And Status Register I2CSCR/I2CSSR

Slave mode control register and slave mode status register occupy a register address, use different operations to distinguish access to these two registers:

Write operation: write I2CSCR (write only)

Read operation: Read the I2CSSR (read-only)

The control register consists of two bits: the RSTS and DA bits. The RSTS bit controls the reset of the entire I2C slave module. When the I2C bus encounters some problems, the software enables this 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	--	--	--	--	--	--	DA
R/W	W	R	R	R	R	R	R	W
Reset value	0	0	0	0	0	0	0	0

Bit7 RSTS: I²C slave module reset control bit;

1= Reset slave module;

0= No effect.

Bit6~Bit1 -- Reserved, all must be 0.

Bit0 DA: I²C slave mode enable bit;

1= Enable;

0= Disable.

The status register consists of three bits: SENDFIN bit, RREQ bit, and TREQ bit. The sent SENDFIN bit indicates that the

master I²C controller has completed the data reception during the I2CS single or continuous sending operation. The receive request RREQ bit indicates that the I2CS device has received a data byte from the I2C master and the I2CS device should read a data byte from the receive data register I2CSBUF. The send request TREQ bit instructs the I2CS device to be addressed as a slave and that the I2CS device should write a data byte to the transmit data register I2CSBUF. If the I2C interrupt enable is turned on, any one of the three flag bits is set to 1 to generate an interrupt.

In the slave control mode, the bus busy flag is judged by Bit6 (BUS_BUSY) of the master control mode status register I2CMSR. When the bus is idle, the I2CMSR register is 0x20. When the start condition is generated until the stop condition is generated, the I2CMSR register is 0x60. When the stop condition is generated, the I2CMSR register is 0x20.

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, all must be 0.

Bit2 SENDFIN: In I2C slave mode, the send operation is completed flag, read only.

1= The master device no longer needs data, TREQ is no longer set to 1, and the data transfer has been completed. (automatically cleared after reading I2CSCR).

0= --

Bit1 TREQ: I²C slave mode ready to send flag bit, read only.

1= As the sending device has been addressed, the master device is ready to receive data. (It is automatically cleared after writing I2CSBUF).

0= --

Bit0 RREQ: I²C slave mode receiving completion flag bit, read only.

1= Receive completed. (It is automatically cleared after reading I2CSBUF).

0= Receive not completed.

23.4.3 I²C Slave Mode To Send And Receive Buffer Register 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>: I²C sent or received data;

Write: Write the data to be sent (the sending order is from the high bit to the low bit);

Read: Data has been received.

23.5 I²C Interrupt

The interrupt number of I2C is 21, and its interrupt vector is 0x00AB. To enable I2C interrupts, the enable bit I2CIE must be set to 1, and the total interrupt enable bit EA must be set to 1.

If the I2C related interrupt enable is turned on,I²C and the I2C total interrupt indicator bit I2CIF=1, the CPU will enter the interrupt service routine. The I2CIF operation attribute is read-only and has nothing to do with the status of I2CIE.

I2C master mode interrupt flag bit I2CMIF, send operation completed flag bit SENDFIN in slave mode, slave mode ready to send flag bit TREQ, slave mode receive completion flag bit RREQ when any one is 1, the I2C total interrupt indicator bit I2CIF will be Will be set to 1. Only when these 4 flag bits are all 0, I2CIF is automatically cleared to 0.

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

23.5.2 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 as high-level interrupt;
0= Set as low-level interrupt.

Bit6 PI2C: I^2C interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.

Bit5 PWDT: WDT interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.

Bit4 PADC: ADC interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.

Bit3 PPWM: PWM interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.

Bit2 --: Reserved, must be zero.

Bit1 PT4: TIMER4 interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.

Bit0 PT3: TIMER3 interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.

23.5.3 Peripheral Interrupt Flag 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 total interrupt indication bit, read only;
1= SPI generates an interrupt, (after clearing the specific interrupt flag bit, this bit is automatically cleared);
0= SPI does not generate interrupts.

Bit6 I2CIF: I^2C total interrupt indication bit, read only;
1= I^2C generates an interrupt, (after clearing the specific interrupt flag bit, this bit is automatically cleared);
0= I^2C does not generate an interrupt.

Bit5 --: Reserved, must be 0.

Bit4 ADCIF: ADC interrupt flag bit;
1= ADC conversion is completed and needs to be cleared by software;
0= ADC conversion is not completed.

Bit3 PWMIF: PWM total interrupt indication bit, read only;
1= PWM generates an interrupt (this bit is automatically cleared after clearing the specific interrupt flag bit);
0= PWM does not generate an interrupt.

Bit2 --: Reserved, must be 0.

Bit1 TF4: Timer4 timer overflow interrupt flag bit;
1= Timer4 timer overflows, it is automatically cleared by hardware when entering the interrupt service routine, and can also be cleared by software;
0= Timer4 timer has no overflow.

Bit0 TF3: Timer3 timer overflow interrupt flag bit;
1= Timer3 timer overflows, it is automatically cleared by hardware when entering the interrupt service routine, and can also be cleared by software;
0= Timer3 timer has no overflow.

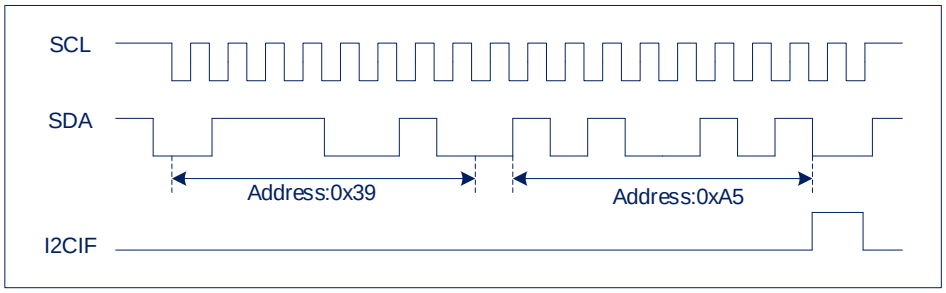
23.6 I²C Slave Mode Transmission Mode

The default I2C address of all the waveforms presented in this section is 0x39 ("00111001").

23.6.1 Single Received

The figure below shows the signal sequence received by I2C during a single data period. single receive sequence:

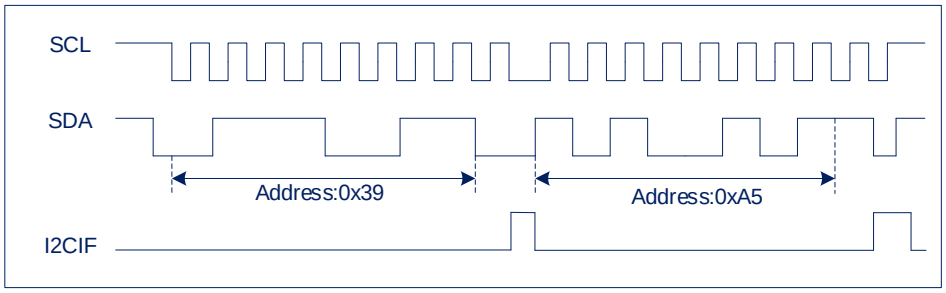
- Start condition;
- I2C is addressed by the I2C host as a receiver;
- The address is confirmed by I2C;
- The data is received by I2C;
- The data is confirmed by I2C;
- Stop condition.



23.6.2 Single Send

The following figure shows the signal sequence sent by I2C during a single data period. Single send sequence:

- Starting conditions;
- I2C is addressed by the I2C host as a transmitter;
- The address is confirmed by I2C;
- The data is transmitted by I2C;
- The data is not confirmed by the I2C host;
- Stop condition.



23.6.3 Continuous Received

The following figure shows the signal sequence received by I2C during continuous data reception. Continuous receiving sequence:

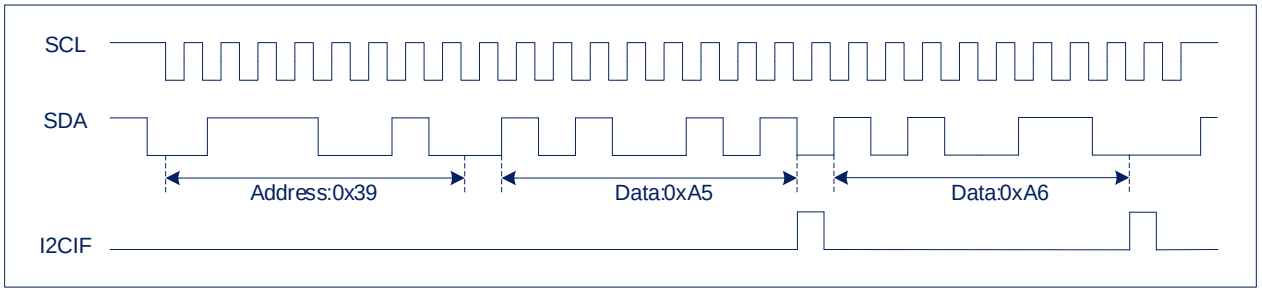
- Starting conditions.
- I2C is addressed by the I2C master as a receiver.
- The address is confirmed by I2C.

1) The data is received by I2C.

2) The data is confirmed by I2C.

- Stop condition.

Sequences 1) and 2) are repeated until the stop condition occurs.



23.6.4 Continuous Send

The following figure shows the signal sequence sent by I2C during continuous data transmission. Continuous sending sequence:

- Sending conditions.
- I2C is addressed by the I2C master as a sender.
- The address is confirmed by I2C.

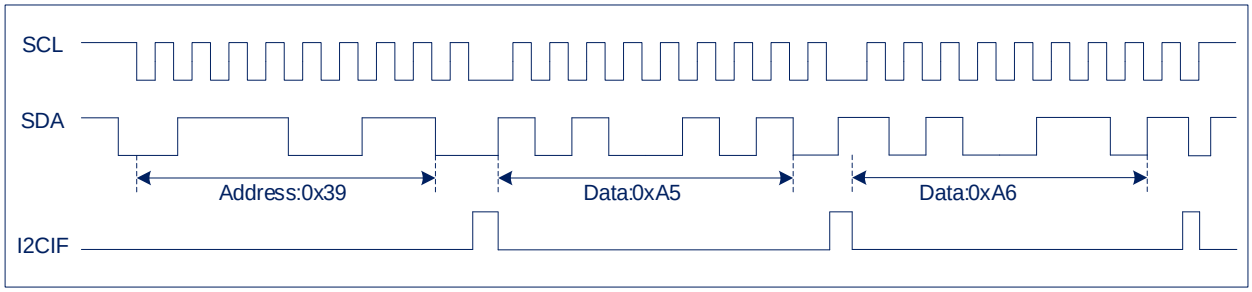
1) The data is sent by I2C.

2) The data I2C host confirms the data.

3) The final data is not confirmed by the I2C host.

- Stop condition.

Repeat sequence 1) and 2) until the last data sent is not confirmed by the I2C master 3).



24. UARTn module

24.1 Overview

The Universal Synchronous Asynchronous Receiver Transmitter (UART0 / UART1 / UART2 / UART3) provide a flexible method for full-duplex data exchange with external devices.

There are two physically independent receiving and sending buffers SBUFn in UARTn, and the operation of receiving buffer or sending buffer is distinguished by reading and writing commands to SBUFn. When writing SBUFn, the data is loaded into the send buffer , read the contents in the receive buffer when reading SBUFn.

UARTn has four operating modes: a synchronous mode and three asynchronous modes. Modes 2 and 3 have multi-machine communication function, which can be enabled by setting the SMn2 bit in the SCONn register to 1. The master processor first sends the address byte that identifies the target slave. The address byte is different from the data byte, because the 9th bit in the address byte is 1 and the data byte is 0. When SMn2=1, the slave will not be interrupted by the data byte. The address byte will interrupt all slaves. The addressed slave will clear its SMn2 bit, and is ready to receive the upcoming data byte. The unaddressed slave sets SMn2 to 1 and ignores the incoming data.

24.2 UARTn Port Configuration

Before using the UARTn module, you need to configure the corresponding ports to the TXDn and RXDn channels of UARTn. For example, the port configuration of UART0/2/3 is as follows:

```
P20CFG = 0x02; //P20 is multiplexed as RXD0 function
P21CFG = 0x02; //P21 is multiplexed as TXD0 function
P06CFG = 0x02; //P06 is multiplexed as RXD2 function
P05CFG = 0x02 ; //P05 is multiplexed as TXD2 function
P45CFG = 0x02; //P45 is multiplexed as RXD3 function
P44CFG = 0x02; //P44 is multiplexed as TXD3 function
```

UART1 port RXD1 can select P11, P14, P22 through PS _ RXD1(RXD1 pin Only one of them can be selected), TXD1 port can select P13, P23 through the port configuration register (can be selected at the same time, or select one. If selected at the same time, P13/P23 will output the corresponding waveform). For example, select P11 as the RXD1 pin of UART1 and P13 as the TXD1 pin, and the configuration is as follows:

```
PS_RXD1 = 0x11;//Select P11 as the RXD1 pin
P11CFG = 0x02; //P11 is multiplexed as RXD1 function
P13CFG = 0x02; //P13 When multiplexing for TXD1 function
```

When using, it is recommended to set the working mode first, and then configure the corresponding port as a serial port.

24.3 UARTn Baud Rate

UARTn is in mode 0, the baud rate is fixed to the system clock divided by 12 (Fsys/12); in mode 2, the baud rate is fixed to the system clock divided by 32 or 64 (Fsys/32, Fsys/64); In mode 1 and mode 3, the baud rate is generated by the Timer1 or Timer4 or Timer2 or BRT or BRT1 module. Which timer the chip selects as the baud rate clock source is determined by the register FUNCCR/FUNCCR1.

24.3.1 Baud Rate Clock Source

When UARTn is in Mode 1 and Mode 3, the clock source of baud rate is selected as follows :

1) Clock source selection of UART0 baud rate:

   FUNCCR[2:0]=000, select Timer1 as the baud rate generator of UART0;;

   FUNCCR[2:0]=001, select Timer4 as the baud rate generator of UART0;

   FUNCCR[2:0]=010, select Timer2 as the baud rate generator of UART0;

   FUNCCR[2:0]=011, select BRT as the baud rate generator of UART0;

   FUNCCR[2:0]=100, select BRT1 as the baud rate generator of UART0.

2) Clock source selection of UART1 baud rate:

   FUNCCR[6:4]=000, select Timer1 as the baud rate generator of UART1;

   FUNCCR[6:4]=001, select Timer4 as the baud rate generator of UART1

   FUNCCR[6:4]=010, select Timer2 as the baud rate generator of UART1;

   FUNCCR[6:4]=011, select BRT as the baud rate generator of UART1;

   FUNCCR[6:4] =100, select BRT1 as the baud rate generator of UART1.

3) Clock source selection of UART2 baud rate:

   FUNCCR1[2:0]=000, select Timer1 as the baud rate generator of UART2;

   FUNCCR1[2:0]=001, select Timer4 as the baud rate generator of UART2;

   FUNCCR1[2:0]=010, select Timer2 as the baud rate generator of UART2;

   FUNCCR1[2:0]=011, select BRT as the baud rate generator of UART2;

   FUNCCR1[2:0] =100, select BRT1 as the baud rate generator of UART2.

4) Clock source selection of UART3 baud rate:

   FUNCCR1[6:4]=000, choose Timer1 as the baud rate generator of UART3;

   FUNCCR1[6:4]=001, choose Timer4 as the baud rate generator of UART3;

   FUNCCR1[6:4]=010, select Timer2 as the baud rate generator of UART3;

   FUNCCR1[6:4]=011, select BRT as the baud rate generator of UART3;

   FUNCCR1[6:4] =100, select BRT1 as the baud rate generator of UART3.

24.3.2 Baud Rate Calculation

When UARTn is in mode 1 and mode 3, the baud rate calculation formula for different clock sources is as follows:

1）The formula of the baud rate when Timer1 or Timer4 works in 8-bit auto reload mode:

$$BaudRate=\frac{Fsys\times 2^{SMODn}}{32\times(4\times 3^{1-TxM})\times(256-THx)}(x=1,4)$$

SMODn is the baud rate selection bit, which is set by the register PCON/PCON1. T1M is the timer 1 clock selection bit, which is set by the register CKCON[4], and T4M is the timer 4 clock selection bit, which is set by the register T34MOD[6]. That is, the TH1/TH4 value of Timer1 or Timer4 at the corresponding baud rate should be set to:

$$THx=256-\frac{Fsys\times 2^{SMODn}}{32\times(4\times 3^{1-TxM})\times BaudRate}(x=1,4)$$

2）The formula of the baud rate when Timer2 works in overflow auto reload mode:

$$BaudRate=\frac{Fsys\times 2^{SMODn}}{32\times(12\times 2^{T2PS})\times(65536-\{RLDH,RLDL\})}$$

T2PS is the timer 2 clock prescaler selection bit, which is set by register T2CON[7]. That is, the value of ' { RLDH, RLDL } ' of Timer2 at the corresponding baud rate should be set as follow:

$$\{RLDH,RLDL\}=65536-\frac{Fsys\times 2^{SMODn}}{32\times(12\times 2^{T2PS})\times BaudRate}$$

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 the BRT timer prescaler selection bit, which is set by the register BRTCON. That is, at the corresponding baud rate, the value of "{BRTDH, BRTDL}" of BRT should be set as :

$$\{BRTDH,BRTDL\}=65536-\frac{Fsys\times 2^{SMODn}}{32\times 2^{BRTCKDIV}\times BaudRate}$$

4）When BRT1 is used as a baud rate generator, the baud rate formula:

$$BaudRate=\frac{Fsys\times 2^{SMODn}}{32\times(65536-\{BRTDH1,BRTDL1\})\times 2^{BRT1CKDIV}}$$

BRT1CKDIV is the BRT1 timer prescaler selection bit, which is set by the register BRT1CON. That is, at the corresponding baud rate, the value of "{BRTDH1, BRTDL1}" of BRT1 should be set as :

：

$$\{BRTDH1,BRTDL1\}=65536-\frac{Fsys\times 2^{SMODn}}{32\times 2^{BRT1CKDIV}\times BaudRate}$$

24.3.3 Baud Rate Error

When UARTn is in mode 1 and mode 3, the errors under different baud rate clock sources and different baud rates are as follows:

Table 1) and 2) are part of the baud rate related information in the 8-bit auto reload mode of Timer 1/Timer 4 in the variable baud rate mode. Table 3) and 4) are part of the baud rate related information when the overflow rate of the BRT/BRT1 timer is used as the UART clock source in the variable baud rate mode.

1）SMODn=0，T1M=1/T4M=1:

Baud rate	Fsys=8MHz			Fsys=16MHz			Fsys=24MHz			Fsys=48MHz		
bps	{TH1, TH4}	Actual Rate	% Error	{TH1, TH4}	Actual Rate	% Error	{TH1, TH4}	Actual Rate	% Error	{TH1, TH4}	Actual 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}	Actual Rate	% Error	{TH1, TH4}	Actual Rate	% Error	{TH1, TH4}	Actual Rate	% Error	{TH1, TH4}	Actual 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/BRT1CKDIV=0

Baud rate	Fsys=8MHz			Fsys=16MHz			Fsys=24MHz			Fsys=48MHz		
bps	{BRTH, BRTL}	Actual Rate	% Error	{BRTH, BRTL}	Actual Rate	% Error	{BRTH, BRTL}	Actual Rate	% Error	{BRTH, BRTL}	Actual 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/BRT1CKDIV=0

Baud rate	Fsys=8MHz			Fsys=16MHz			Fsys=24MHz			Fsys=48MHz		
bps	{BRTH, BRTL}	Actual Rate	% Error	{BRTH, BRTL}	Actual Rate	% Error	{BRTH, BRTL}	Actual Rate	% Error	{BRTH, BRTL}	Actual 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

24.4 UARTn Registers

UARTn has the same function as the standard 8051 UART. The related registers are: FUNCCR, FUNCCR1, SBUFn, SCONn, PCON, PCON1, IE, EIE3, IP, EIP3. The UARTn data buffer (SBUFn) consists of two independent registers: send and receive registers. The data written in SBUFn will set this data in the UARTn output register and start transmission; the data read in SBUFn will read data from the UARTn receiving register. SCON0 register supports bit addressing operation, SCON1/2/3 register does not support bit addressing operation, please pay attention when using assembly language. The baud rate is doubled by setting the registers PCON and PCON1.

24.4.1 UART0/1 Baud Rate Selection Register FUNCCR

0x91	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FUNCCR	--	UART1_CKS2	UART1_CKS1	UART1_CKS0	--	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

Register in BANK0

Bit7	--	Reserved, must be 0.
Bit6-Bit4	UART1_CKS<2:0>:	Timer clock source selection for UART1
	000 =	Timer1 overflow clock;
	001 =	Timer4 overflow clock;
	010 =	Timer2 overflow clock;
	011 =	BRT overflow clock;
	100 =	BRT1Clock;
	Other =	Prohibit access
Bit3	--	Reserved, must be 0.
Bit2-Bit0	UART0_CKS<2:0>:	Timer clock source selection for UART0
	000 =	Timer1 overflow clock;
	001 =	Timer4 overflow clock;
	010 =	Timer2 overflow clock;
	011 =	BRT overflow clock;
	100 =	BRT1 overflow Clock;
	Other =	Prohibit access

24.4.2 UART2/3 Baud Rate Selection Register FUNCCR1

0xE2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FUNCCR1	--	UART3_CKS2	UART3_CKS1	UART3_CKS0	--	UART2_CKS2	UART2_CKS1	UART2_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

Register in BANK1

Bit7 -- Reserved, must be 0.

Bit6~Bit4 UART3_CKS<2:0>: Timer clock source selection for UART3

000 = Timer1 overflow clock;

001 = Timer4 overflow clock;

010 = Timer2 overflow clock;

011 = BRT overflow clock; overflow clock

100 = BRT1Clock;

Other = Prohibit access.

Bit3 -- Reserved, must be 0.

Bit2~Bit0 UART2_CKS<2:0>: Timer clock source selection for UART2

000= Timer1 overflow clock;

001= Timer4 overflow clock;

010= Timer2 overflow clock;

011= BRT overflow clock;

100= BRT1 overflow clock;

Other= Prohibit access.

24.4.3 UARTn Buffer Register SBUFn

	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

BANK0: register SBUF0 address 0x99; register SBUF1 address 0xEB.

BANK1: register SBUF2 address 0xE5; register SBUF3 address 0xE7.

Bit7~Bit0 BUFFERn<7:0>: Buffer data register.

Write: UARTn starts to send data.

Read: Read the received data.

24.4.4 UART Control Register SCONn

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SCONn	UnSM0	UnSM1	UnSM2	UnREN	UnTB8	UnRB8	TIn	RIn
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; register SCON1 address 0xEA.

BANK1: register SCON2 address 0xE4; register SCON3 address 0xE6.

Bit7~Bit6 UnSM0-UnSM1: UARTn mode selection bit;
- 00= Master control synchronous mode;
- 01= 8-bit asynchronous mode, with variable baud rate;
- 10= 9-bit asynchronous mode, with baud rate Fsys/32 or Fsys/ 64;
- 11= 9-bit asynchronous mode, the baud rate is variable.

Bit5 UnSM2: Multi-machine communication control bit;
- 1= Enable;
- 0= Disable.

Bit4 UnREN: Receive enable bit;
- 1= Enable;
- 0= Disable.

Bit3 UnTB8: The 9th bit of the send data, mainly used for sending in 9-bit asynchronous mode;
- 1= The 9th bit is 1;
- 0= The 9th bit is 0.

Bit2 UnRB8: The 9th bit of the receive data, mainly used for receiving in 9-bit asynchronous mode;
- 1= Received the 9th bit of data received is 1.
- 0= Received the 9th bit of data received is 0.

Bit1 TIn: Send interrupt flag bit (requires software to clear);
- 1= That the send buffer is empty, and the next frame data can be sent.
- 0= --

Bit0 RIn: Receive interrupt flag bit (requires software to clear);
- 1= That the receive buffer is full, and the next frame of data can be received after reading.
- 0= --

UARTn mode is as follows:

SMn0	SMn1	Mode	Description	Baud rate
0	0	0	Shift register	Fsys/12
0	1	1	8-Bit UART	controlled by Timer4/Timer1/Timer2/BRT/BRT1.
1	0	2	9-Bit UART	SMODn=0:Fsys/64;SMODn=1:Fsys/32.
1	1	3	9-Bit UART	controlled by Timer4/Timer1/Timer2/BRT/BRT1.

24.4.5 PCON Register

0x87	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON	SMOD0	SMOD1	--	--	--	SWE	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

Register in BANK0

Bit7 SMOD0: UART0 baud rate double bit;
1= UART0 baud rate doubled;
0= UART0 baud rate is normal.

Bit6 SMOD1: UART1 baud rate double bit;
1= UART1 baud rate doubled;
0= UART1 baud rate is normal.

Bit5~Bit3 -- Reserved, must be 0.

Bit2 SWE: STOP status function wake-up enable bit;
(Regardless of the value of SWE, the system can be restarted by power-off reset or enabled external reset)
0= Function wake-up is disabled;
1= Function wake-up is enabled (can be waked up by external interrupt and timed) wake).

Bit1 STOP: Sleep status control bit;
1= Enter sleep status (automatically clear when exiting STOP mode);
0= Not enter sleep status.

Bit0 IDLE: Idle state control bit;
1= Enter idle state (automatically clear when exiting IDLE mode);
0= Not enter idle state

24.4.6 PCON1 Register

0xE3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON1	--	--	--	--	--	--	SMOD3	SMOD2
R/W	R	R	R	R	R	R	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Register in BANK1

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

Bit1 SMOD3: UART3 baud rate double bit;
1= UART3 baud rate doubled;
0= UART3 baud rate is normal.

Bit0 SMOD2: UART2 baud rate double bit;
1= UART2 baud rate doubled;
0= UART2 baud rate is normal.

24.5 UARTn Interrupt

The interrupt number of UART0 is 4, and its interrupt vector is 0x0023.

The interrupt number of UART1 is 6, and its interrupt vector is 0x0033.

The interrupt number of UART2 is 23, and its interrupt vector is 0x00BB.

The interrupt number of UART3 is 24, and its interrupt vector is 0x00C3.

To enable the UARTn interrupt, the enable bit ESn must be set to 1, and the total interrupt enable bit EA must be set to 1. If UARTn related interrupt enable is turned on, when TIn=1 or RIn=1, the CPU will enter the corresponding interrupt service routine. TIn/RIn has nothing to do with the state of ESn and needs to be cleared by software. For detailed description, refer to register SCONn.

24.5.1 Interrupt Mask Register IE

0xA8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IE	EA	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 EA: Global interrupt enable bit;
1= Enable all unmasked interrupts;
0= Disable all interrupts.

Bit6 ES1: UART1 interrupt enable bit;
1= Enable UART1 interrupt;
0= Disable UART1 interrupt.

Bit5 ET2: TIMER2 total interrupt enable bit;
1= Enable all TIMER2 interrupts;
0= Disable all TIMER2 interrupts.

Bit4 ES0: UART0 interrupt enable bit;
1= Enable UART0 interrupt;
0= Disable UART0 interrupt.

Bit3 ET1: TIMER1 interrupt enable bit;
1= Enable TIMER1 interrupt;
0= Disable TIMER1 interrupt.

Bit2 EX1: External interrupt 1 interrupt enable bit;
1= Enable external interrupt 1 interrupt;
0= Disable external interrupt 1 interrupt.

Bit1 ET0: TIMER0 interrupt enable bit;
1= Enable TIMER0 interrupt;
0= Disable TIMER0 interrupt.

Bit0 EX0: External interrupt 0 interrupt enable bit;
1= Enable external interrupt 0 interrupt;
0= Disable external interrupt 0 interrupt.

24.5.2 Interrupt Mask Register EIE3

0xAB	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIE3	--	--	--	--	--	--	ES3	ES2
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, all must be 0.
Bit1 ES3: UART3 interrupt enable bit;
1= Enable UART3 interrupt;
0= Disable UART3 interrupt.
Bit0 ES2: UART2 interrupt enable bit;
1= Enable UART2 interrupt;
0= Disable UART2 interrupt.

24.5.3 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 PS1: UART1 interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.
Bit5 PT2: TIMER2 interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.
Bit4 PS0: UART0 interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.
Bit3 PT1: TIMER1 interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.
Bit2 PX1: External interrupt 1 interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.
Bit1 PT0: TIMER0 interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.
Bit0 PX0: External interrupt 0 interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.

24.5.4 Interrupt Priority Control Register EIP3

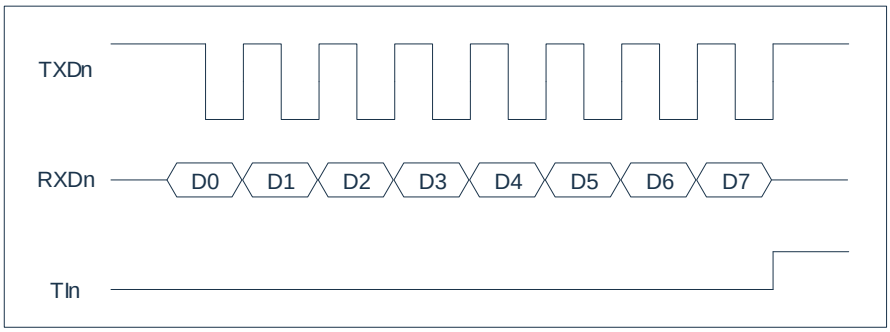
0xBB	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP3	--	--	PWWDT	--	--	PLSE	PUART3	PUART2
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit6	--	Reserved, must be 0.
Bit5	PWWDT	WWDT interrupt priority control bit;
	1=	Set as high-level interrupt;
	0=	Set as low-level interrupt.
Bit4~ Bit3	--	Reserved, must be 0.
Bit2	PLSE:	LSE interrupt priority control bit;
	1=	Set as high-level interrupt;
	0=	Set as low-level interrupt.
Bit1	PUART3:	UART3 interrupt priority control bit;
	1=	Set as high-level interrupt;
	0=	Set as low-level interrupt.
Bit0	PUART2:	UART2 interrupt priority control bit;
	1=	Set as high-level interrupt;
	0=	Set as low-level interrupt.

24.6 UARTn Interrupt

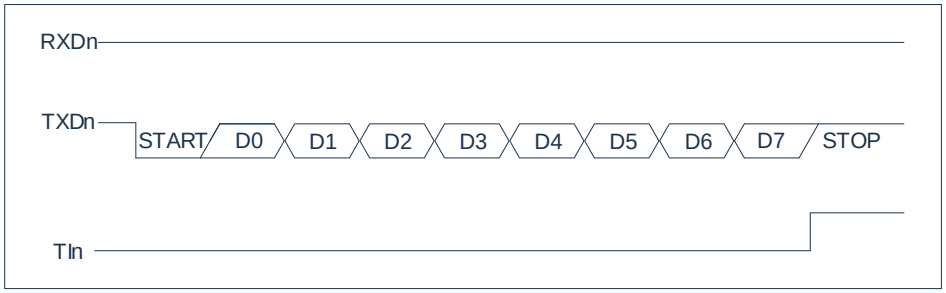
24.6.1 Mode 0-Synchronous Mode

The pin RXDn is used as input or output, and TXDn is used as clock output. The TXDn output is a shift clock. The baud rate is fixed at 1/12 of the system clock frequency. 8-bit data is transmitted LSB first. Initialize the reception by setting the flag in SCONn, set as: RIn = 0 and RENn = 1. The timing diagram of Mode 0 is shown in the figure below:



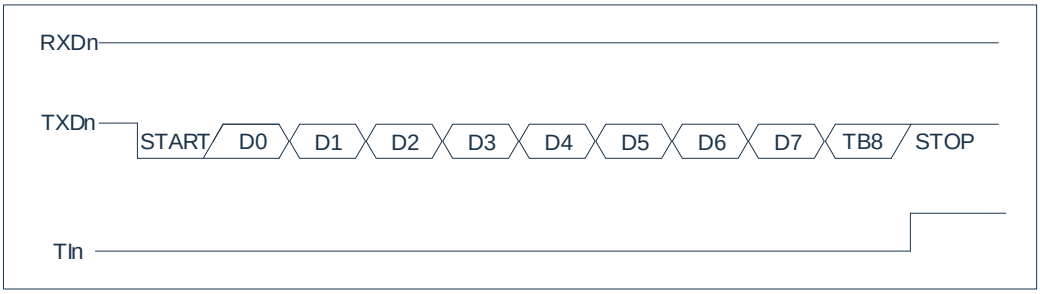
24.6.2 Mode 1-8-bit Asynchronous Mode (Variable Baud Rate)

The pin RXDn is used as input, TXDn is used as serial output. Send 10 bits: start bit (always 0), 8-bit data (LSB first) and stop bit (always 1). When receiving, the start bit is transmitted synchronously, 8 data bits can be obtained by reading SBUFn, and the stop bit is set by the flag RBn8 in SCONn. The baud rate is variable and depends on the TIMER1/TIMER2/TIMER4/BRT/BRT1 mode. The timing diagram of Mode 1 is shown in the figure below:



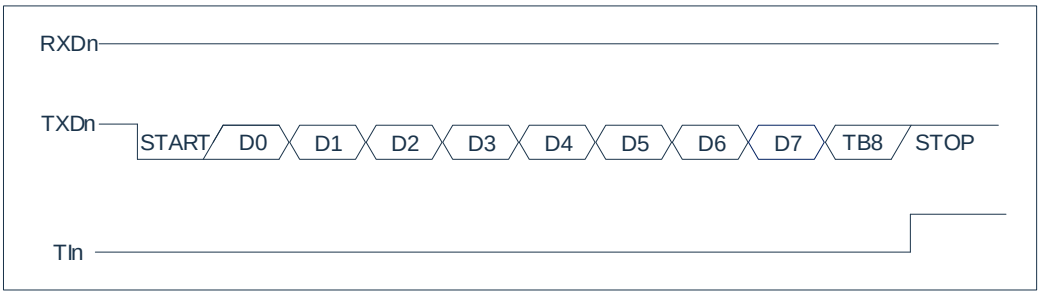
24.6.3 Mode 2-9-bit Asynchronous Mode (Fixed Baud Rate)

This mode is similar to Mode 1, but there are two differences. In this mode,the baud rate is fixed at 1/32 or 1/64 of the CLK clock frequency, 11-bit transmission and reception: start bit (0), 8-bit data (LSB first), programmable ninth bit and stop bit (1). The ninth bit can be used to control the parity check of the UARTn interface: when sending, the bit TBn8 in SCONn is output as the ninth bit, and when receiving, the ninth bit affects the RBn8 in SCONn. The timing diagram of Mode 3 is shown in the figure below:



24.6.4 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/BRT1 mode. The timing diagram of Mode 4 is shown in the following figure:



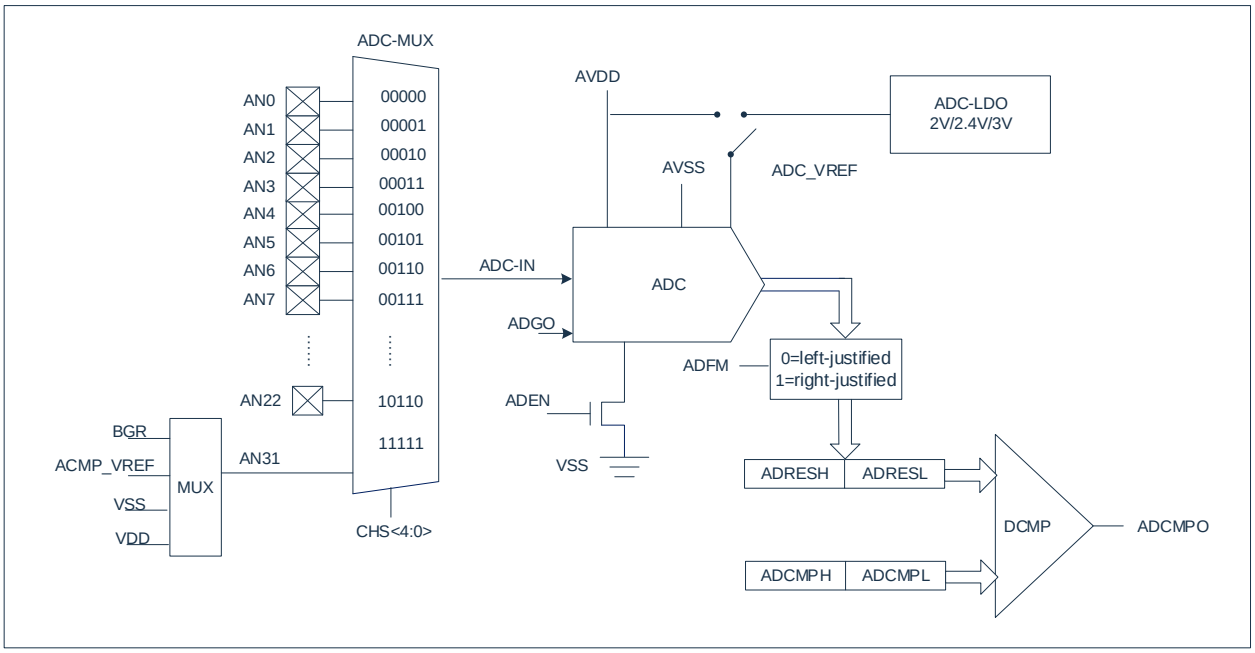
25. Analog to Digital Converter (ADC)

25.1 Overview

The analog to digital converter (ADC) can convert the analog input signal into a 12-bit binary number representing the signal. The block diagram of the ADC structure is shown in the figure below.

The port analog input signal and internal analog signal are connected to the input of the analog to digital converter after passing through the multiplexer. The analog to digital converter uses the successive approximation method to produce a 12-bit binary result, and save the result in the ADC result register (ADRESL and ADRESH), the ADC can generate an interrupt after the conversion is completed. The ADC conversion result is compared with the value of the ADC comparison data register (ADCMPL and ADCMPH), and the comparison result is stored in the ADCMPO flag.

ADC reference voltage is always generated internally, and can be provided by AVDD or by internal ADC-LDO.



25.2 ADC Configuration

When configuring and using ADC, the following factors must be considered:

- Port configuration.
- Channel selection.
- ADC conversion clock source.
- Interrupt control.
- The storage format of the result.

25.2.1 Port Configuration

ADC can convert both analog signals and digital signals. When converting analog signals, the corresponding port needs to be configured as an analog port.

Note: Applying analog voltage to pins defined as digital inputs may cause overcurrent in the input buffer.

25.2.2 Channel Selection

The ADCHS bit in the ADCON1 register determines which channel is connected to the analog to digital converter.

If the channel is changed, a certain delay is required before the next conversion starts. The ADC delay time is shown in the table below:

Delay time	Working voltage
500ns	2.5~4.5V
200ns	4.5~5.5V

25.2.3 ADC Reference Voltage

ADC reference voltage is provided by the chip's VDD by default, or it can be provided by the internal ADC-LDO. ADC-LDO can choose 3 kinds of voltage output: 2.0V/2.4V/3.0V.

25.2.4 Clock Conversion

The ADCKS bit of the ADCON1 register can be set by software to select the conversion clock source.

The time to complete 1-bit conversion is defined as T_{ADCK}. A complete 12-bit conversion requires 18.5 T_{ADCK} cycles (the time that ADGO continues to be high to complete a conversion). Must comply with the corresponding T_{ADCK} specification in order to obtain the correct conversion results, the following table is an example of correct selection of ADC clock.

Fsys	F_{ADCK} (T_A=25˚C)	
	V_{REF}=V_{REF1}=AVDD (AVDD=VDD)	V_{REF}=V_{REF2}=2.0V V_{REF}=V_{REF3}=2.4V V_{REF}=V_{REF4}=3.0V
8MHz	F_{sys}/4	F_{sys}/16
16MHz	F_{sys}/8	F_{sys}/32
24MHz	F_{sys}/16	F_{sys}/64
48MHz	F_{sys}/32	F_{sys}/128

Note: Any change in the system clock frequency will change the ADC clock frequency, which will negatively affect the ADC conversion result.

25.2.5 Formatization Of Results

The result of 12-bit A/D conversion can be in two formats: left alignment or right alignment. The output format is controlled by the ADFM bit in the ADCON0 register.

When ADFM=0, the AD conversion result is aligned to the left;

when ADFM=1, the AD conversion result is aligned to the right.

25.3 ADC Hardware Trigger Start

In addition to software trigger ADC conversion, the ADC module also provides a hardware trigger start method. One is the edge trigger mode of the external port, and the other is the edge or period trigger mode of the PWM.

Using hardware to trigger ADC needs to set ADCEX to 1, that is enable external trigger ADC function. The hardware trigger signal will set the ADGO bit to 1 after a certain delay, and it will be automatically cleared after the conversion is completed. After the hardware trigger function is turned on, the software trigger function will not be turned off. In the ADC idle state, writing 1 to the ADGO bit can also start AD conversion.

25.3.1 External Port Edge Trigger ADC

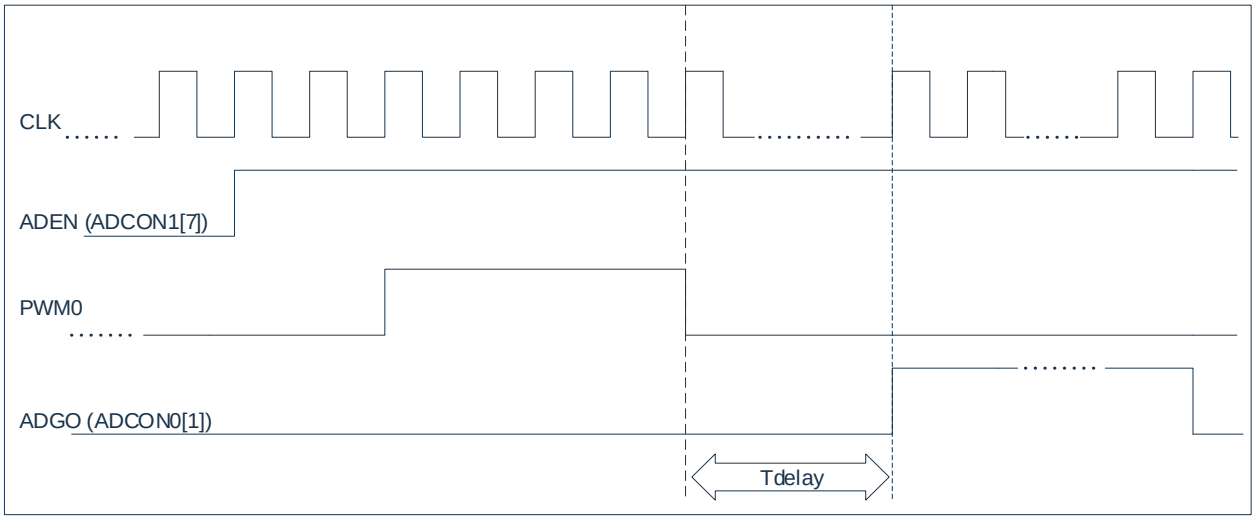
ADET pin edge automatically trigger ADC conversion. At this time, ADTGS[1:0] needs to be 11 (select external port edge trigger), and ADEGS[1:0] can select which edge trigger.

25.3.2 PWM Trigger ADC

PWM can choose to trigger ADC conversion by edge or period/zero point. ADTGS[1:0] select PWM channel (PG0, PG2, PG4), ADEGS[1:0] can select edge type or period type trigger mode.

25.3.3 Hardware Trigger Start Delay

After the hardware trigger signal is generated, AD conversion is not started immediately, and ADGO is set to 1 after a certain delay. The delay is determined by ADDLY[9:0]. The delay time of the hardware trigger signal: (ADDLY+3)*Tsys, the structure diagram of the delay trigger is shown in the figure below:



25.4 ADC Result Comparison

ADC module provides a group of digital comparators for comparing the result of ADC with pre-loaded { ADCMPH, ADCMPL} value size. The result of each ADC conversion will be compared with the preset value ADCMP. The result of the comparison is stored in the ADCMPO flag bit. After the conversion is completed, the flag bit will be automatically updated. The ADCMPPS bit can change the polarity of the output result.

ADC comparison results can trigger enhanced PWM fault brake, which requires ADFBEN set to 1.

When the enhanced PWM function is turned on and ADFBEN=1, the AD conversion result is compared with the preset value {ADCMPH, ADCMPL}. If the comparison result ADCMPO is 1, the PWM will immediately generate a fault brake operation and clear the start-bit of all PWM channels and terminate all PWM channel output.

25.5 ADC Working Principle

25.5.1 Start Conversion

To enable the ADC module, the ADEN bit of ADCON1 register must be 1, and then set the ADGO bit of the ADCON0 register to 1 to start analog to digital conversion (ADGO cannot be set to 1 when ADEN is 0).

25.5.2 Completion Of The Conversion

When the conversion is completed, the ADC module will:

- Clear the ADGO bit;
- Set the ADCIF flag bit to 1;
- Update the ADRESH:ADRESL register with the new conversion result.

25.5.3 Termination Of Conversion

If it is necessary to terminate the conversion before the conversion is completed, the uncompleted analog to digital conversion result will not be updated to the ADRESH:ADRESL register. Therefore, the ADRESH:ADRESL register will hold the value obtained from the last conversion.

Note: Device reset will force all registers to enter the reset state. Therefore, the reset will shut down the ADC module and terminate any pending conversions.

25.5.4 A/D Conversion Steps

The configuration steps for using ADC for analog-to-digital conversion are as follows:

1) Port configuration:
   - Disable pin output driver (see PxTRIS register);
   - Configure the pin as an analog input pin.
2) Configure ADC interrupt (optional):
   - Clear ADC interrupt flag bit;
   - Enable ADC interrupt;
   - Allow peripheral interrupt;
   - Allow global interrupt.
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 sampling time.
5) Set ADGO to 1 to start conversion.
6) Wait for the ADC conversion to complete by one of the following methods:
   - Query the ADGO bit;
   - Wait for the ADC interrupt (allow interrupts).
7) Read the ADC result.
8) Clear the ADC interrupt flag bit (if interrupts are enabled, this operation is required).

Note: If the user attempts to resume sequential code execution after waking the device from sleep mode, the global interrupt must be disabled.

25.5.5 Enter Sleep Mode During Conversion

When the system is about to enters the sleep mode, it is recommended to wait for the ADC ongoing conversion to complete before entering the sleep mode.

If the ADC enters to sleep mode while the conversion is in progress, the conversion is terminated. The conversion operation needs to be performed again after waking up.

25.6 Related Registers

There are mainly 10 registers related to AD conversion, namely:

- AD control register ADCON0, ADCON1, ADCON2, ADCLDO;
- Comparator control register ADCMPC;
- Delay data register ADDLYL;
- AD result data register ADRESH/L;
- Comparator data register ADCMPH/L.

25.6.1 AD Control Register ADCON0

0xDF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON0	ADCHS4	ADFM	--	AN31SEL2	AN31SEL1	AN31SEL0	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	ADCHS4	The 4th bit of ADC analog channel selection bit;
	1=	Refer to the description of register ADCON1 for channel selection;
	0=	-
Bit6	ADFM:	ADC conversion result format selection bit;
	1=	Align to the right;
	0=	Align to the left.
Bit5	--	Reserved
Bit4~Bit2	AN31SEL<2:0>:	ADC channel 31 input source selection bits;
	000=	BGR (1.2V);
	001=	ACMP_VREF (the negative reference voltage of the comparator, see the ACMP chapter for details);
	010=	Reserved
	011=	Reserved
	100=	Reserved
	101=	VSS (ADC reference ground);
	110=	Reserved, forbidden to use;
	111=	VDD (ADC default reference voltage).
Bit1	ADGO:	ADC conversion start bit (when the bit is set to 1, ADEN must be 1, otherwise the operation is invalid);
	1=	Write: start ADC conversion, (the bit will be 1 when ADC is triggered by hardware); Read: ADC is converting.
	0=	Write: invalid. Read: ADC is idle/conversion completed; During ADC conversion (ADGO=1), any software and hardware trigger signals will be ignored.
Bit0	--	Reserved, must be 0.

25.6.2 AD Control Register ADCON1

0xDE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON1	ADEN	ADCKS2	ADCKS1	ADCKS0	ADCHS3	ADCHS2	ADCHS1	ADCHS0
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= Disable ADC, no working current is consumed.

Bit6~Bit4 ADCKS<2:0>: ADC conversion clock selection 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 ADCHS<3:0>: Low 4 bits of the analog channel selection bit, and ADCHS<4> form the 5-bit channel selection bit, ADCHS<4:0>;
00000= AN0; 10000= AN16;
00001= AN1; 10001= AN17;
00010= AN2; 10010= AN18;
00011= AN3; 10011= AN19;
00100= AN4; 10100= AN20;
00101= AN5; 10101= AN21;
00110= AN6; 10110= AN22;
00111= AN7; Others= Prohibited Access;
01000= AN8; 11111 = Se the description of ADCON0.AN31SEL.
01001= AN9;
01010= AN10;
01011= AN11;
01100= AN12;
01101= AN13;
01110= AN14;
01111= AN15;

25.6.3 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 selection bits;
	00=	PG0 (PWM0);
	01=	PG2 (PWM2);
	10=	PG4 (PWM4);
	11=	Port pin (ADET).
Bit3~Bit2	ADEGS<1:0>:	ADC hardware trigger edge selection bits;
	00=	Falling edge;
	01=	Rising edge;
	10=	Period point of PWM period;
	11=	Zero point of PWM period.
Bit1~Bit0	--	Reserved, all must be 0.

25.6.4 AD Comparator Control Register ADCMPC

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 selection bit;
	1=	If ADRES<ADCMP, then ADCMPO=1;
	0=	If ADRES>=ADCMP, then ADCMPO=1.
Bit5	--	Reserved, must be 0.
Bit4	ADCMPO:	ADC comparator output bit. This bit outputs the result of the ADC comparator output, and it will be updated every time the ADC conversion ends.
Bit3~Bit2	--	Reserved, must be 0.
Bit1~Bit0	ADDLY<9:8>:	ADC hardware trigger delay data[9:8] bits.

25.6.5 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 low 8 bits.

25.6.6 AD Data Register High ADRESH, ADFM=0 (left align)

0xDD	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADRESH	ADRES11	ADRES10	ADRES9	ADRES8	ADRES7	ADRES6	ADRES5	ADRES4
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 ADRES<11:4>: ADC result register bits.
The 11-4 digits of the 12-bit conversion result.

25.6.7 AD Data Register Lower ADRESL, ADFM=0 (left alignment)

0xDC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADRESL	ADRES3	ADRES2	ADRES1	ADRES0	--	--	--	--
R/W	R	R	R	R	--	--	--	--
Reset value	X	X	X	X	--	--	--	--

Bit7~Bit4 ADRES<3:0>: ADC result register bits.
Bits 3-0 of the 12-bit conversion result.

Bit3~Bit0 Not used.

25.6.8 AD Data Register High ADRESH, ADFM=1 (right alignment)

0xDD	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADRESH	--	--	--	--	ADRES11	ADRES10	ADRES9	ADRES8
R/W	--	--	--	--	R	R	R	R
Reset value	--	--	--	--	X	X	X	X

Bit7~Bit4 Not used.

Bit3~Bit0 ADRES<11:8>: ADC result register bits.
The 11th to 8th bits of the 12-bit conversion result.

25.6.9 AD Data Register Low Bit ADRESL, ADFM = 1 (right alignment)

0xDC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADRESL	ADRES7	ADRES6	ADRES5	ADRES4	ADRES3	ADRES2	ADRES1	ADRES0
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 ADRES<7:0>: ADC result register bits.
Bits 7-0 of the 12-bit conversion result.

25.6.10 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>: High 8 bits of ADC comparator data

25.6.11 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 Not used.

Bit3~Bit0 ADCMP<3:0>: Low 4 bits of ADC comparator data.

25.6.12 AD Reference Voltage Control Register

F692H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCLDO	LDOEN	VSEL1	VSEL0	--	--	--	--	--
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 is enabled;

1= LDO is enabled, the reference voltage can only select the voltage corresponding to VSEL[1:0];

0= LDO is disabled, the reference voltage is the chip voltage.

Bit6~Bit5 VSEL<1:0>: ADC reference voltage selection bits;

00= 2.0V;

01= 2.0V;

10= 2.4V;

11= 3.0V.

Bit4 -- Reserved, must be 0.

Bit3~Bit0 -- Reserved, must be 0.

25.7 ADC Interrupt

The ADC module allows an interrupt to be generated after the completion of the analog-to-digital conversion. 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 by software, and the ADCIF bit will be set to 1 after each conversion, regardless of whether ADC interrupt is allowed. The ADC interrupt enable and priority can be set by the following relevant register bits.

25.7.1 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 interrupt;
0= Disable SPI interrupt.
Bit6 I2CIE: I^2C interrupt enable bit;
1= Enable I^2C interrupt;
0= Disable I^2C interrupt.
Bit5 WDTIE: WDT interrupt enable bit;
1= Enable WDT overflow interrupt;
0= Disable WDT overflow interrupt.
Bit4 ADCIE: ADC interrupt enable bit;
1= Enable ADC interrupt;
0= Disable ADC interrupt.
Bit3 PWMIE: PWM total 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 interrupt;
0= Disable Timer4 interrupt.
Bit0 ET3: Timer3 interrupt enable bit;
1= Enable Timer3 interrupt;
0= Disable Timer3 interrupt.

25.7.2 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 as high-level interrupt;
0= Set as low-level interrupt.

Bit6 PI2C: I^2C interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.

Bit5 PWDT: WDT interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.

Bit4 PADC: ADC interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.

Bit3 PPWM: PWM interrupt priority control bit
1= Set as high-level interrupt;
0= Set as low-level interrupt.

Bit2 --: Reserved, must be 0.

Bit1 PT4: TIMER4 interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.

Bit0 PT3: TIMER3 interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.

25.7.3 Peripheral Interrupt Flag 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 total interrupt flag bit, read-only;
1= SPI generates an interrupt (after clearing the specific interrupt flag bit, this bit is automatically cleared);
0= SPI does not generate an interrupt.

Bit6 I2CIF: I^2C total interrupt flag bit, read-only;
1= I^2C generates an interrupt and needs to be cleared by software (after clearing the specific interrupt flag bit, this bit is automatically cleared);
0= I^2C does not generate an interrupt.

Bit5 --: Reserved, must be 0.

Bit4 ADCIF: ADC interrupt flag bit;
1= ADC conversion is completed and needs to be cleared by software;
0= ADC conversion is not completed.

Bit3 PWMIF: PWM total interrupt flag bit, read-only;
1= PWM generates an interrupt (this bit is automatically cleared after clearing the specific interrupt flag bit);
0= PWM does not generate an interrupt.

Bit2 --: Reserved, must be 0.

Bit1 TF4: Timer4 timer overflow interrupt flag bit;
1= Timer4 timer overflows, it is automatically cleared by hardware when entering the interrupt service routine, and can also be cleared by software;
0= Timer4 timer has no overflow.

Bit0 TF3: Timer3 timer overflow interrupt flag bit;
1= Timer3 timer overflows, it is automatically cleared by hardware when entering the interrupt service routine, and can also be cleared by software;
0= Timer3 timer has no overflow.

26. Analog Comparator（ACMP0/1）

The chip contains two analog comparators, ACMP0 and ACMP1. When the positive terminal voltage is greater than the negative terminal voltage, the comparator outputs logic 1, otherwise it outputs 0, which can also be changed by the output polarity selection bit. When the output value of the comparator changes, each comparator can generate an interrupt.

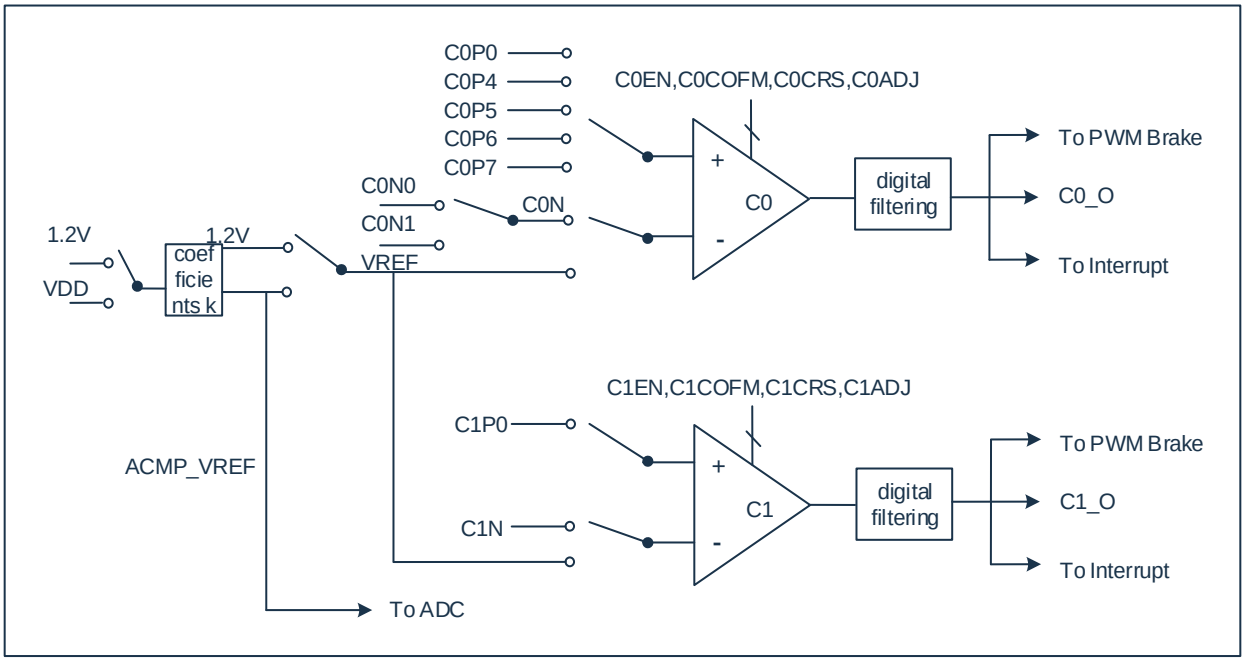
26.1 Comparator Characteristics

The comparator has the following characteristics:

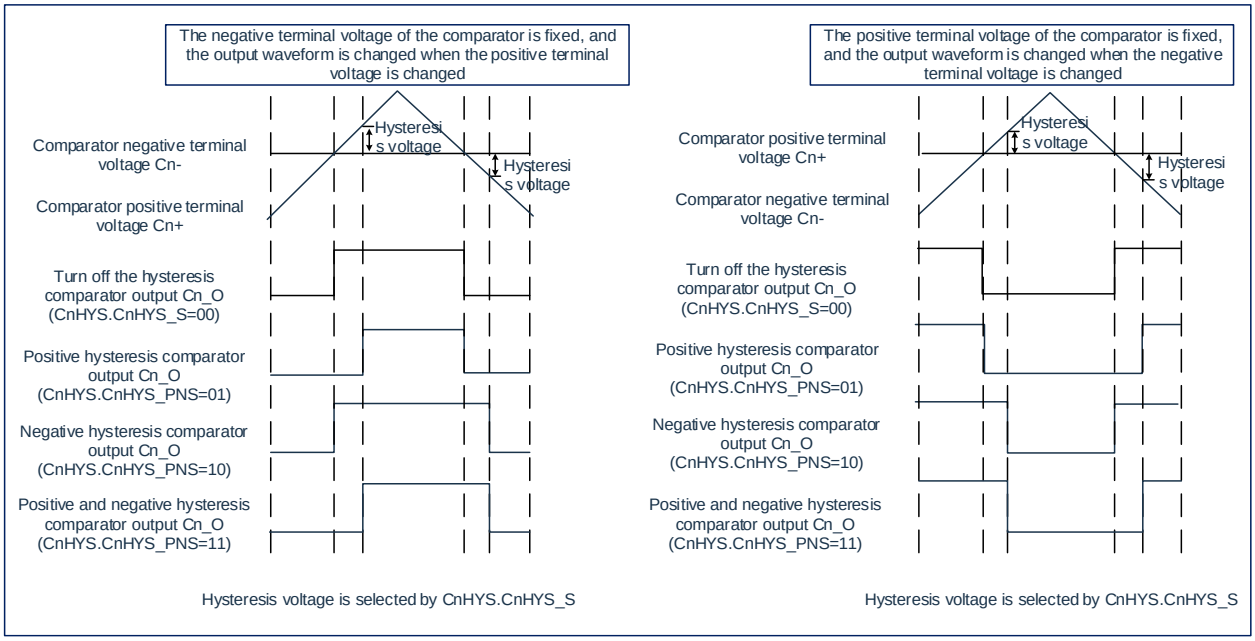
- Positive terminal of comparator 0 can select 5 port inputs;
  The positive terminal of comparator 1 can select 1 port input.
- The negative terminal of comparator 0 can select port input C0N0/C0N1 and internal reference voltage VREF; the negative terminal of comparator 1 can select port input C1N and internal reference voltage VREF.
- The internal reference voltage can choose the internal Bandgap (1.2V) and ACMP_VREF output.
- ACMP_VREF reference source voltage divider range: k=(2/20)~(17/20), a total of 16 gear selections.
- The output filter time can be selected: 0~512*Tsys.
- Support unilateral (positive/negative) and bilateral (positive and negative) hysteresis control.
- The hysteresis voltage can be 10/20/60mV.
- Offset voltage supports software trimming.
- The output can be used as an enhanced PWM brake trigger signal.
- An output change can generate an interrupt.
- Output lockable

26.2 Comparators Structure

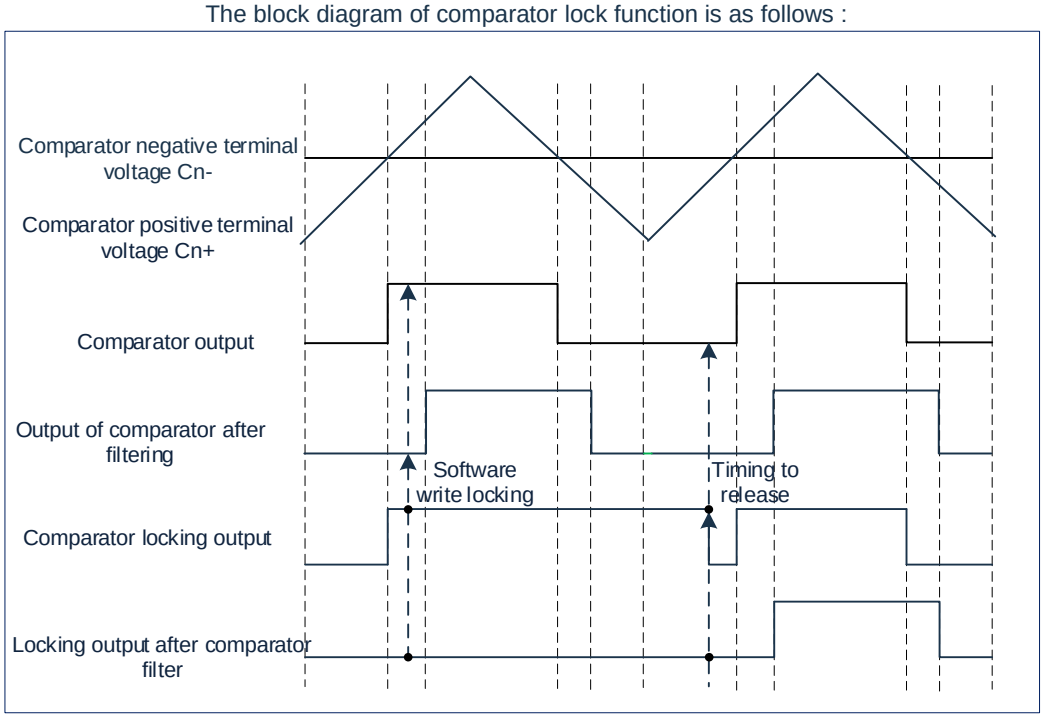
The block diagrams shown below:



comparator with hysteresis control diagram is shown below:



The block diagram of comparator lock function is as follows :



26.3 Related Registers

26.3.1 Comparator 0 Control Register C0CON0

F500H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
C0CON0	C0EN	C0COFM	C0N2G	C0NS1	C0NS0	C0PS2	C0PS1	C0PS0
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 C0EN: Comparator 0 enable bit;
1= Enable;
0= Disable.

Bit6 C0COFM: Comparator 0 trimming mode enable bit;
1= Enable trimming mode;
0= Disable trimming mode.

Bit5 C0N2G: Comparator 0 trimming mode negative terminal grounding enable bit (this bit is valid when C0CRS=0)
1= The negative terminal channel is closed and the internal negative terminal is grounded;
0= The negative terminal channel is enabled, and the signal is input from the negative terminal.

Bit4~Bit3 C0NS<1:0>: Comparator 0 negative terminal channel selection bit;
00= Comparator 0 negative terminal port
01= Internal voltage (Bandgap or ACMP_VREF);
1x= Reserved, prohibited to use.

Bit2~Bit0 C0PS<2:0>: Comparator 0 positive channel selection bit C0PS<2:0>;
000= C0P0;
001= Reserved, forbidden to use;
010= Reserved, forbidden to use;
011= Reserved, forbidden to use;
100= C0P4;
101= C0P5;
110= C0P6;
111= C0P7.

26.3.2 Comparator 1 Control Register C1CON0

F503H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
C1CON0	C1EN	C1COFM	C1N2G	C1NS1	C1NS0	C1PS2	C1PS1	C1PS0
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 C1EN: Comparator 1 enable bit;
1= Enable;
0= Disable.
Bit6 C1COFM: Comparator 1 trimming mode enable bit;
1= Enable trimming mode;
0= Disable trimming mode.
Bit5 C1N2G: Comparator 1 trimming mode negative terminal grounding enable bit (this bit is valid when C1CRS=0)
1= The negative terminal channel is closed and the internal negative terminal is grounded;
0= The negative terminal channel is enabled, and the signal is input from the negative terminal.
Bit4~Bit3 C1NS<1:0>: Comparator 1 negative terminal channel selection bit;
00= C1N;
01= Internal voltage (Bandgap or ACMP_VREF);
1x= Reserved, prohibited to use.
Bit2~Bit0 C1PS<2:0>: Comparator 1 positive channel selection bit C1PS<2:0>;
000= C1P0;
001= Reserved, prohibited to use;
010= Reserved, prohibited to use;
011= Reserved, prohibited to use;
100= Reserved, prohibited to use;
101= Reserved, prohibited to use;
110= Reserved, prohibited to use;;
111= Reserved, prohibited to use.

26.3.3 Comparator Control Register CnCON1

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CnCON1	CnOUT	CnCRS	--	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 bit, read-only;
Bit6 CnCRS: Comparator n trimming mode input terminal selection;
1= Connect the positive and negative terminals together and input from the positive terminal;
0= Connect the positive and negative terminals together, from the negative terminal enter.
Bit5 -- Reserved, must be 0.
Bit4~Bit0 CnADJ<4:0>: Comparator n offset voltage trimming bits.

26.3.4 Comparator 0 Control Register C0CON2

F502H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
C0CON2	--	C0NS	C0POS	C0FE	C0FS3	C0FS2	C0FS1	C0FS0
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	C0NS	Comparator 0 negative terminal port selection
	1=	C0N1
	0=	C0N0
Bit5	C0POS:	Comparator 0 output polarity selection bit (it may cause the interrupt flag to be set when switching);
	1=	Inverted output;
	0=	Normal output .
Bit4	C0FE:	Comparator 0 output filter enable bit;
	1=	Enable filtering;
	0=	Disable filtering.
Bit3~Bit0	C0FS<3:0>:	Comparator 0 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;
	Others=	(0~1)*Tsys.

26.3.5 Comparator 1 Control Register C1CON2

F505H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
C1CON2	--	--	C1POS	C1FE	C1FS3	C1FS2	C1FS1	C1FS0
R/W	R	R	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 C1POS: Comparator 1 output polarity selection bit (it may cause the interrupt flag to be set when switching);

1= Inverted output;

0= Normal output.

Bit4 C1FE: Comparator 1 output filter enable bit;

1= Enable filtering;

0= Disable filtering.

Bit3~Bit0 C1FS<3:0>: Comparator 1 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;

Others= (0~1)*Tsys.

26.3.6 Comparator Trimming 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 trimming mode selection;

AAH = Determined by CnADJ<4:0> in CnCON1 register;

Others = Determined by CONFIG related bits.

It is recommended to start the comparator after setting the parameters of the comparator, otherwise, the output jump of the comparator may be detected by mistake during the setting process.

26.3.7 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 selection 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.

26.3.8 Comparator Reference Voltage Control Register CNVRCON

F506H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CNVRCON	--	--	CNDIVS	CNSVR	CNVS3	CNVS2	CNVS1	CNVS0
R/W	R	R	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, all must be 0.

Bit5 CNDIVS: ACMP_VREF reference source selection bit;

1= Select 1.2V (Bandgap) for voltage division;

0= Select VDD for voltage division.

Bit4 CNSVR: Comparator negative terminal internal voltage VREF selection bit;

1= Select ACMP_VREF (the voltage divider circuit is turned on, independent of the comparator module);

0= Select 1.2V (Bandgap).

Bit3~Bit0 CNVS<3:0>: ACMP_VREF reference source voltage divider coefficient k selection bits;

0000-1111= 2/20 ~ 17/20.

26.3.9 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 control PWM brake enable bit;
0= Disable;
1= Enable.
Bit6 C0FBPEN: Comparator 0 output level control PWM brake enable bit;
0= Disable;
1= Enable.
Bit5 C1FBPS: Comparator 1 output level control PWM brake level selection bit;
0= High level;
1= Low level;
Bit4 C0FBPS: Comparator 0 output level control PWM brake level selection bit;
0= High level;
1= Low level;
Bit3 C1FBEN: Comparator 1 output event control PWM brake enable bit;
0= Disable;
1= Enable.
Bit2 C0FBEN: Comparator 0 output event control PWM brake enable bit;
0= Disable;
1= Enable.
Bit1 C1FBLS: Comparator 1 output event control PWM brake edge selection bit;
0= Rising edge;
1= Falling edge.
Bit0 C0FBLS: Comparator 0 output event control PWM brake edge selection bit;
0= Rising edge;
1= Falling edge.

26.3.10 Comparator Lock Function Register CnCON3

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CnCON3	-	-	CTM1	CTM0	-	-	-	SSET
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

C0CON3address: F50EH; C1CON3address: F50FH.

Bit7~Bit6 -- Reserved, must all be 0.
Bit5~Bit4 CTM<1:0>: Lock recovery timer selection;
0x1= Select timer 0 overflow recovery ;
0x2= Select timer 1 overflow recovery.
Others value= Forbidden (cannot be restored).
Bit3~Bit1 -- Reserved, must all be 0.
Bit0 SSET: Lock enable bit
1= Enable(comparator keeps the current output) ;
0= Disable(comparator normal output).

26.4 Comparator Interrupt

Comparator 0 and Comparator 1 can both set interrupts, and both share an interrupt vector entry. After entering the interrupt service routine, the user can use the interrupt flag bit to determine which type of interrupt is generated. Comparator interrupt priority and interrupt enable can be set by the following related register bits.

26.4.1 Interrupt Priority Control Register EIP1

0xB9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP1	PACMP	--	PP5	PP4	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 as high-level interrupt;
0= Set as low-level interrupt.
Bit6 -- Reserved, must be zero.
Bit5 PP5: Port P5 interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.
Bit4 PP4: Port 4 interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.
Bit3 PP3: Port P3 interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.
Bit2 PP2: Port P2 interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.
Bit1 PP1: Port P1 interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.
Bit0 PP0: Port P0 interrupt priority control bit;
1= Set as high-level interrupt;
0= Set as low-level interrupt.

26.4.2 Comparator Interrupt Enable Register CNIE

F508H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CNIE	--	--	--	--	--	--	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, all 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.

26.4.3 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, all must be 0.

Bit1 C1IF: Comparator 1 interrupt flag bit (write 0 to clear);

1= Comparison 1 output has changed.

0= -

Bit0 C0IF: Comparator 0 interrupt flag bit (write 0 to clear);

1= Compare 0 output has changed.

0= --

27. Flash Memory

27.1 Overview

FLASH memory includes program memory (APROM) and non-volatile data memory (Data FLASH). The maximum program memory space is 64KB, divided into 128 sectors, each sector contains 512B. The max data memory space is 1KB, divided into 2 sectors, each sector contains 512B.

The FLASH memory can be accessed through the relevant special function register (SFR) to realize the IAP function. The SFR registers used to access the FLASH space are as follows:

- MLOCK
- MSTATUS
- MDATA
- MADRL
- MADRH
- PCRCDL
- PCRCDH
- MREGION
- MMODE

The MLOCK register is used to enable memory operations, the MSTATUS register is used to indicate the status of the FLASH operation and to set the accessed address, the MDATA register forms a byte to save the 8-bit data to be read/written, and the MADRL/MADRH register stores the address of the accessed MDATA unit, the PCRCDL/PCRCDH register stores the result of CRC operation, the MREGION register is used for memory area selection, and MMODE is used for memory operation mode selection.

Through the interface of the memory module, the memory can be read/written/erased/CRC checked operation. The memory allows byte read and write, and the write time is controlled by the on-chip timer. Before writing new data, make sure that the data in this address has been erased.The write and erase voltage is generated by the on-chip charge pump. The rated operating voltage of this charge pump is within the voltage range of the device,It is used to perform byte operations.

Flash memory erase operation only supports sector erase, not byte erase. Before modifying the data of an address, it is recommended to save other data, then erase the current sector, and finally write the data.

27.2 Related Registers

27.2.1 FLASH Protection Lock Register MLOCK

0xFF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MLOCK	MLOCK7	MOCK6	MLOCK5	MLOCK4	MLOCK3	MLOCK2	MLOCK1	MLOCK0
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 MLOCK<7:0>: Memory operation enable bit

AAH= Memory-related R/W/E/CRC operations are allowed;

55H= Memory-related R/W/E/CRC operations are allowed, and MADDR is incremented by 1 after writing MDATA;

Others= Memory-related R/W/E/CRC operations are not allowed.

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

```
MOV     TA,#AAH
MOV     TA,#55H
MOV     MLOCK,#AAH
```

27.2.2 FLASH Status Register MSTATUS

0xFE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MSTATUS	MLOCKF	ERROR	CRCASEL	CRCCLR	START	-	-	-
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 MLOCKF: Memory operation enable status indication bit (this bit is 1 when MLOCK is enabled, otherwise it is 0);

1= Enable, FLASH can be operated by register ;

0= Disable, FLASH cannot be operated.

Bit6 ERROR: Operation error flag bit (write 0 to clear) ;

1= Before the programming operation starts, check that the data in the programming address is not ' FFH ' (un erased) and write operations terminate immediately.

0= --

Bit5 CRCASEL: CRC check end address selection bit ;

1= Select the end address ;

0= Select the starting address.

Bit4 CRCCLR: CRC operation result clear bit;

1= Clear CRC operation results register(This bit is automatically cleared by hardware);

0= --.

Bit3 START: Operation start control bit;

1= Start the memory R/W/E/CRC check operation (after the operation is completed, it can be automatically cleared by hardware);

0= Write: Terminate or not start the program memory R/W/E verify operation;
Read: Operation completed or operation not started.

Bit2~Bit0 -- Reserved, must be 0.

Note: The CRCASEL must be cleared after the CRC is finished.

27.2.3 FLASH Memory Data Register MDATA

0xFB	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 to be read or written to the program memory.

27.2.4 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	0	0	0	0	0	0

Bit7~Bit0 MADRL<7:0>: Specify the low 8 bits of the address for memory read/write operations.

27.2.5 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>: Specify the high 8 bits of the address for memory read/write operations.

27.2.6 Program CRC Operation Result Data Register Low 8 bits PCRCDL

F706H	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> Program CRC operation result low 8 bits data

27.2.7 Program CRC Operation Result Data Register High 8 bits PCRCDH

F707H	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> Program CRC operation result high 8 bits data

27.2.8 FLASH Memory Area Control Register MREGION

0xF9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MREGION	REGION7	REGION6	REGION5	REGION4	REGION3	REGION2	REGION1	REGION0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	1	0	1	0	1	0	1	0

Bit7~Bit0 REGION<7:0> Flash area selection bit ;
55H= Select the APROM area ;
AAH= Select the DATA area ;
69H= Select user CONFIG area (read only, write and erase are forbidden) ;
others= Invalid, will exit all ongoing operations.

27.2.9 FLASH Memory Mode Control Register MMODE

0xFA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MMODE	MODE7	MODE6	MODE5	MODE4	MODE3	MODE2	MODE1	MODE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	1	1	0	1	0	0	1

Bit7~Bit0 MODE<7:0> Flash mode selection bit;
55H= Erase operation;
AAH= Write operation;
69H= Read operation;
96H= CRC check;
Others= Invalid, will exit all in-progress operations.

+

27.3 Function Description

During the FLASH memory read/write/erase operation, the CPU is in a suspended state, and when the operation is completed, the CPU continues to run instructions.

Note: FLASH read / write / erase / CRC check must ensure that MLOCK, MREGION, MMODE are effectively configured to start the operation. During the operation, if any register of MLOCK, MREGION and MMODE is invalid, the operation will be prohibited.

27.3.1 FLASH Read Operation

The read operation steps of FLASH memory are as follows :

1) Enable memory access operations:
   TA = 0xAA;
   TA = 0x55;
   MLOCK=0xAA;
2) Set the memory address to be accessed:
   Set the address by the MADRL/MADRH.
3) Set the corresponding area of the memory address to be accessed:
   Set the read area by the MREGION register.
4) Set the read command:
   MMODE=0x69.
5) Start read operation:
   MSTATUS[3] is set to 1.
6) After waiting for six NOP command, determine whether the read operation is over :
   MSTATUS[3] is cleared to 0 by hardware after the read operation.
7) Reading results :
   MDATA stored the data corresponding to the address.
8) Disable memory access operations:
   TA = 0xAA;
   TA = 0x55;
   MLOCK=0x00.

27.3.2 FLASH Write Operation

The write operation steps of FLASH memory are as follows :

1) Enable memory access operations:
   TA = 0xAA;
   TA = 0x55;
   MLOCK=0xAA;
2) Set the memory address to be accessed:
   Set the address by the MADRL/MADRH.
3) Set the data to be written:
   MDATA stores the data to be written.
4) Set the corresponding area of the memory address to be accessed:
   Set the write area by the MREGION register.
5) Set the write command:
   MMODE=0xAA.
6) Start write operation:
   MSTATUS[3] is set to 1.
7) After waiting for six NOP command, determine whether the write operation is over :
   MSTATUS[3] is cleared to 0 by hardware after the write operation.
9) Disable memory access operations:
   TA = 0xAA;
   TA = 0x55;
   MLOCK=0x00.

FLASH memory address self-adding-writing operation steps are as follows:

1) Enable memory access operations:
   TA = 0xAA;
   TA = 0x55;
   MLOCK=0x55;
2) Set the first memory address to be accessed:
   Set the address via MADRL/MADRH.
3) Set the corresponding area of the memory address to be accessed:
   Set the write area through the MREGION register.
4) Set the write command:
   MMODE=0xAA.
5) Set the data to be written:
   MDATA stores the data to be written.
6) After waiting for six NOP command, determine whether the write operation is over :
   MSTATUS[3] is cleared to 0 by hardware after the write operation.
7) Repeat steps 5) and 6) until all data is written.
8) Disable memory access operations:
   TA = 0xAA;
   TA = 0x55;
   MLOCK=0x00.

27.3.3 FLASH Erase Operation

The erase operation steps of FLASH memory are as follows :

1) Enable memory access operations:

   TA = 0xAA;

   TA = 0x55;

   MLOCK=0xAA;

2) Set the memory address to erase:

   Set the address by the MADRL/MADRH.

3) Set the area corresponding to the memory address to be erased:

   Set the erase area by the MREGION register .

4) Set the erase command:

   MMODE=0x55.

5) Start erase operation:

   MSTATUS[3] is set to 1.

6) After waiting for six NOP command, determine whether the write operation is over :

   MSTATUS[3] is cleared to 0 by hardware after the write operation.

7) Disable memory access operations:

   TA = 0xAA;

   TA = 0x55;

   MLOCK=0x00.

27.3.4 CRC Check

The program CRC check command is set by the register MMODE, the start address and the end address can be freely configured by the register MADRL/MADRH, and the result is stored in the register PCRCDL/PCRCDH. During the CRC check process of the program space, the CPU stops working, and the CPU continues to run after the CRC calculation is completed. The CRC check is checked in bytes, in order from the initial address to the end address. The CRC check operation steps are as follows:

1) Enable memory access operations:
   TA = 0xAA;
   TA = 0x55;
   MLOCK=0xAA;
2) Clear the program CRC check result before:
   PCRCDL=0x00;PCRCDH=0x00(or clear PCRCDL/PCRCDH by writing MSTATUS[4] to 1)
3) Set the start and end addresses of the program CRC check:
   MSTATUS[5]=0, set the starting address by the MADRL/MADRH;
   MSTATUS[5]=1, set the end address by the MADRL/MADRH.
4) Select the area of CRC check :
   Set the check area through the MREGION register.
5) Select CRC check command :
   MMODE=0x96.
6) Start CRC check:
   MSTATUS[3] is set to 1.
7) After waiting for six NOP command, determine whether the write operation is over :
   MSTATUS[3] is cleared to 0 by hardware after the CRC check.
8) Read the program CRC check result:
   PCRCDL stores the low 8 bits of the program CRC operation result;
   PCRCDH stores the high 8 bits of the program CRC operation result.
9) The CRC check end address selection bit is cleared to 0
   MSTATUS[5] is cleared to 0 by software after the CRC check.
10) Disable memory access operations:
   TA = 0xAA;
   TA = 0x55;
   MLOCK=0x00.

28. Unique ID（UID）

28.1 Overview

Each chip has a different 96-bit unique identification number, that is, unique identification. It has been set at the factory and cannot be modified by the user.

28.2 UID Registers 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>

29. User Configuration

The system configuration register (CONFIG) is the FLASH option of the MCU's initial conditions, and the program cannot access and operate it. It contains the following:

1. WDT (watchdog working mode selection)

◆ ENABLE	Force to open WDT
◆ SOFTWARE CONTROL(default)	WDT working mode is controlled by the WDTRE bit of the WDCON register

2. PROTECT

◆ ENABLE	FLASH code encryption, the read code is 00H. And it is forbidden to enter debugging mode.
◆ DISABLE(default)	FLASH code is not encrypted

3. FLASH_DATA_PROTECT

◆ DISABLE	FLASH data area is not encrypted
◆ ENABLE(default)）	FLASH data area is encrypted, and after encryption, the value read by the burning emulator is 00H

4. LVR(low voltage reset)

- ◆ 1.8V(default)
- ◆ 2.0V
- ◆ 2.5V
- ◆ 3.5V

5. DEBUG (debug mode)

◆ DISABLE(default)	debug mode is disabled, DSCK1/2 and DSDA1/2 pins are used as normal IO ports
◆ ENABLE	debug mode is enabled, DSCK1/2 and DSDA1/2 pins are configured as debug ports, and the pins correspond Other functions are disabled.

6. DEBUG_SEL(debug IO port selection)

- ◆ DSCK1/DSDA1(default)
- ◆ DSCK2/DSDA2

7. OSC (oscillation mode)

◆ HSI(default)	48MHz
◆ HSE	reference HSE_SEL
◆ LSE(32.768KHz)	if this item is selected, HSE_SEL must keep the default value.
◆ LSI(125KHz)	125KHz

8. HSE_SEL(crystal oscillator port selection)

◆ OSCIN1/OSCOUT1 AS HSE(default)	P06/P07
◆ OSCIN2/OSCOUT2 AS HSE	P50/P51

9. SYS_PRESCALE (system clock prescaler selection)

- ◆ F_{OSC}/1(default)
- ◆ F_{OSC}/2
- ◆ F_{OSC}/4
- ◆ F_{OSC}/8

10. HSI_FS (internal RC oscillator frequency division selection)

◆ F_{HSI}/1	48MHz
◆ F_{HSI}/2	24MHz
◆ F_{HSI}/3	16MHz

- F_{HSI}/6(default) 8MHz

11. EXT_RESET(external reset configuration)

- DISABLE(default) external reset disable
- ENABLE external reset enable
- ENABLE(OPEN PULLUP) external reset enables and opens internal pull-up resistor of reset port

12. EXT_RESETSEL (external reset port selection)

- P46 pin selection for external reset port
- P52 pin selection for external reset port

13. WAKE UP_WAIT TIME(The default time for wake-up from sleep and wait for oscillator to stabilize is 1.0s)

- 50us
- 100us
- 500us
- 1ms
- 5ms
- 10ms
- 500ms
- 1.0s(default)

14. 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)

15. WRITE_PROTECT Program partition protection (protectable intervals, all default intervals are not protected)

- 0000H-0FFFH(protection/unprotection)
- 1000H-1FFFH(protection/unprotection)
- 2000H-2FFFH(protection/unprotection)
- 3000H-3FFFH(protection/unprotection)
- 4000H-4FFFH(protection/unprotection)
- 5000H-5FFFH(protection/unprotection)
- 6000H-6FFFH(protection/unprotection)
- 7000H-7FFFH(protection/unprotection)
- 8000H-8FFFH(protection/unprotection)
- 9000H-9FFFH(protection/unprotection)
- A000H-AFFFH(protection/unprotection)
- B000H-BFFFH(protection/unprotection)
- C000H-CFFFH(protection/unprotection)
- D000H-DFFFH(protection/unprotection)
- E000H-EFFFH(protection/unprotection)
- F000H-FFFFH(protection/unprotection)

16. WRITE_PROTECT DATA partition protection (protectable intervals, all default intervals are not protected)

- 0000H-01FFH(protection/unprotection)
- 0000H-03FFH(protection/unprotection)

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 frequency is selected as FHSI/1, and the system clock prescaler is selected as FOSC/1, and all three conditions are met, if the memory waiting clock is selected as 1*System Clock (1T), the actual memory waiting clock is selected as 2T, and the machine cycle= T_{SYS}/2.

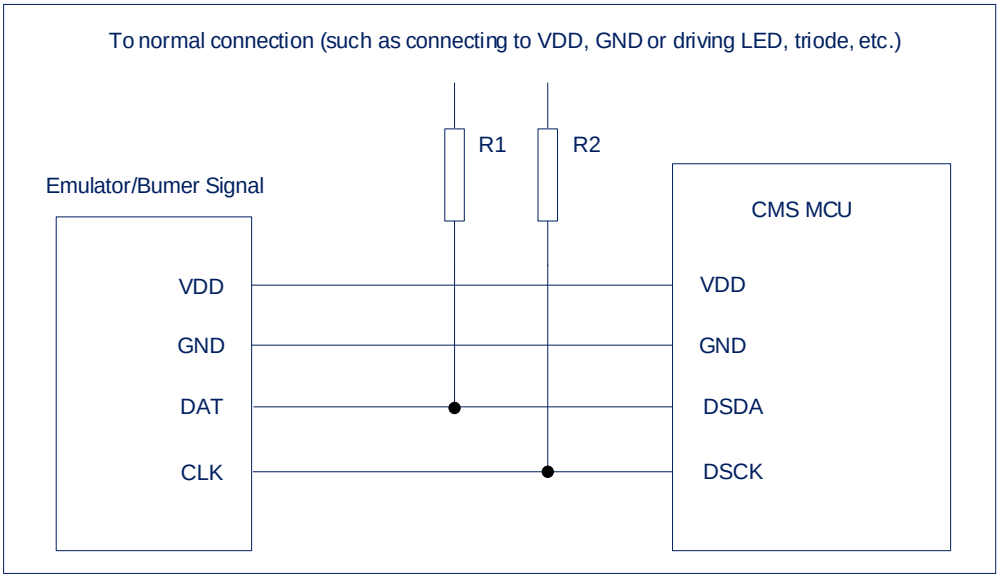
30. Online Programming And Debugging

30.1 Online Programming Mode

The chip can be serially programmed in the final application circuit. Programming can be done simply through the following 4 wires:

- power line
- ground line
- data line
- clock line

In-line serial programming allows users to use unprogrammed devices to manufacture circuit boards and program the chip only before the product is delivered, so that the latest version of firmware or customized firmware can be programmed into the chip. A typical online serial programming connection method is shown in the figure below:



In the above figure, R1 and R2 are electrical isolation devices, which are often replaced by resistors. The resistance values are as follows:R1≥4.7K、R2≥4.7K.

Note that during programming and debugging, DSDA is forbidden to connect a pull-down resistor. If the actual circuit requires a pull-down resistor, it is recommended to use a jumper structure to disconnect the pull-down resistor during programming/debugging, and then connect the pull-down resistor after completion.

30.2 Online Debugging Mode

The chip supports 2-wire (DSCK1/DSDA1 or DSCK2/DSDA2) online debugging function. 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 pay attention to the following points:

- In the debugging state, DSCK1/DSDA1 or DSCK2/DSDA2 are used as dedicated debugging ports, and their GPIO and multiplexing functions cannot be realized.
- When entering the sleep mode/idle mode in the debug state, the system power supply and oscillator will not stop working, and the sleep wake-up function can be simulated in this state. If you need to pay attention to power consumption, it is recommended to turn off the debugging function before testing the actual sleep current of the chip.
- Pause in the debug state, other functional peripherals continue to run, WDT, Timer0/1/2/3/4 counters will stop. But if Timer1/4 is used as the baud rate generator of UART0/1/2/3, Timer1/4 will continue to run in the pause state. Peripherals that continue to run in the paused state may generate interrupts, so be careful when debugging.
- It is not recommended to use the WDT/WWDT reset and software reset functions in the debugging state, because the chip and the debugger may lose connection during reset.

31. Instruction description

There are 5 types of assembly instructions: arithmetic operations, logic operations, data transfer operations, Boolean operations, and program branch instructions, all of which are compatible with the standard 8051.

31.1 Symbol Description

Symbol	Description
Rn	working register R0-R7
Direct	internal data memory RAM unit address (00H-FFH) or special function register SFR address
@Ri	indirect addressing register (@R0 or @R1)
#data	8-bit binary constant
#data16	16-bit binary constant in the instruction
Bit	address in internal data memory RAM or special function register SFR
Addr16	16-bit address, address range 0-64KB address space
Addr11	11-bit address, address range 0-2KB address space
Rel	relative address
A	accumulator

31.2 Instruction List

Mnemonic		Description
operation type		
ADD	A,Rn	Accumulator plus register
ADD	A,direct	Accumulator plus direct addressing unit
ADD	A,@Ri	Accumulator plus indirect addressing RAM
ADD	A,#data	Accumulator plus immediate
ADDC	A,Rn	Accumulator plus register and carry flag
ADDC	A,direct	Accumulator plus direct addressing unit and carry flag
ADDC	A,@Ri	Accumulator plus indirect addressing RAM and carry flag
ADDC	A,#data	Accumulator plus immediate value and carry flag
SUBB	A,Rn	Accumulator minus register and carry flag
SUBB	A,direct	Accumulator minus direct addressing unit and carry flag
SUBB	A,@Ri	Accumulator minus indirect addressing RAM and carry flag
SUBB	A,#data	Accumulator minus immediate value and carry flag
INC	A	Accumulator plus 1
INC	Rn	Register plus 1
INC	direct	Direct addressing unit plus 1
INC	@Ri	Indirect addressing 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	Indirect addressing RAM minus 1
MUL	A,B	Accumulator multiply register B
DIV	A,B	Accumulator divided by register B
DA	A	Decimal adjustment
logic operation class		
ANL	A,Rn	Accumulators and registers
ANL	A,direct	Accumulator and Direct Addressing Unit
ANL	A,@Ri	Accumulator and indirect addressing RAM
ANL	A,#data	Accumulator and immediate data
ANL	direct,A	Direct addressing unit and accumulator
ANL	direct,#data	Direct addressing unit and immediate data
ORL	A,Rn	Accumulator or register
ORL	A, direct	Accumulator or direct addressing unit
ORL	A,@Ri	Accumulator or indirect addressing RAM
ORL	A, #data	Accumulator or immediate
ORL	direct,A	Direct addressing unit or accumulator
ORL	direct,#data	Direct addressing unit or immediate data
XRL	A,Rn	Accumulator XOR Register
XRL	A,direct	Accumulator XOR Direct Addressing Unit
XRL	A,@Ri	Accumulator XOR indirect addressing RAM
XRL	A,#data	Accumulator XOR immediate
XRL	direct,A	Direct addressing unit exclusive XOR accumulator
XRL	direct,#data	direct addressing unit XOR immediate data
CLR	A	Accumulator is cleared to 0
CPL	A	Inverted accumulator
RL	A	Accumulator rotate left
RLC	A	Accumulator with carry flag to rotate left

Mnemonic		Description
RR	A	Accumulator rotate right
RRC	A	Accumulator with carry flag right cyclic shift
SWAP	A	Swap the high 4 bits and low 4 bits of the accumulator
data transmission type		
MOV	A,Rn	Register transfer to accumulator
MOV	A,direct	Direct addressing unit transfer to accumulator
MOV	A,@Ri	Indirect addressing RAM to accumulator
MOV	A,#data	Immediately send accumulator
MOV	Rn,A	Accumulator send register
MOV	Rn,direct	Direct addressing unit to register
MOV	Rn,#data	Immediate data transfer register
MOV	direct,A	The accumulator sends the direct addressing unit
MOV	direct,Rn	register sends the direct addressing unit
MOV	direct1,direct2	Direct address unit transfer to direct address unit
MOV	direct,@Ri	Indirect addressing RAM to direct addressing unit
MOV	direct,#data	Immediate data direct addressing unit
MOV	@Ri,A	The accumulator sends indirect addressing RAM
MOV	@Ri,direct	Direct addressing unit to indirect addressing RAM
MOV	@Ri,#data	Immediate data transmission and indirect addressing RAM
MOV	DPTR,#data16	16-bit immediate data to send data pointer
MOVC	A,@A+DPTR	The look-up table data is sent to the accumulator (DPTR is the base address)
MOVC	A,@A+PC	Look-up table data sent to accumulator (PC is the base address)
MOVX	A,@Ri	External RAM unit to accumulator (8-bit address)
MOVX	A,@DPTR	External RAM unit to accumulator (16-bit address)
MOVX	@Ri,A	The accumulator sends the external RAM unit (8-bit address)
MOVX	@DPTR,A	The accumulator sends the external RAM unit (16-bit address)
PUSH	direct	The direct addressing unit is pushed onto the top of the stack
POP	direct	Direct addressing unit popped from the top of the stack
XCH	A,Rn	Accumulator and register swap
XCH	A, direct	Accumulator and direct addressing unit RAM swap
XCH	A,@Ri	Accumulator and indirect addressing unit RAM swap
XCHD	A,@Ri	The accumulator and the indirect addressing unit RAM swap the lower 4 bits
Boolean operation type		
CLR	C	C clear
CLR	bit	Direct addressing bit is cleared
SETB	C	C set
SETB	bit	Direct addressing bit
CPL	C	C negate
CPL	bit	Direct addressing bit inversion
ANL	C,bit	C logic and direct addressing bit
ANL	C,/bit	The inverse of C logic and direct addressing bits
ORL	C,bit	C logic or direct addressing bit
ORL	C,/bit	The inverse of C logic or direct addressing bit
MOV	C,bit	Direct addressing bit to C
MOV	bit,C	C send direct addressing bit
Program jump class		
ACALL	addr11	Absolute call within 2K address range
LCALL	addr16	Long call in 64K address range
RET		Subroutine return

Mnemonic		Description
RETI		Interrupt return
AJMP	addr11	Absolute transfer within 2K address range
LJMP	addr16	Long transfer within 64K address range
SJMP	rel	Relatively short transfer
JMP	@A+DPTR	Relatively long transfer
JZ	rel	Accumulator is 0 transfer
JNZ	rel	The accumulator is not 0 transfer
JC	rel	C is 1 transfer
JNC	rel	C is 0 transfer
JB	bit,rel	Direct addressing bit is 1 transfer
JNB	bit,rel	Direct addressing bit is 0 transfer
JBC	bit,rel	The direct addressing bit is transferred by 1, and the bit is cleared
CJNE	A,direct,rel	Accumulator and direct addressing unit unequal transfer
CJNE	A,#data,rel	Accumulator and immediate data unequal transfer
CJNE	Rn,#data,rel	Unequal transfer between register and immediate
CJNE	@Ri,#data,rel	Indirect addressing unit RAM and immediate data unequal transfer
DJNZ	Rn,rel	Register minus 1 is not transferred to 0
DJNZ	direct,rel	Direct addressing unit minus 1 is not transferred to 0
NOP		Null instruction
read-modify-write instruction（Read-Modify-Write）		
ANL		Logic (ANL direct, A and ANL direct, #data)
ORL		Logical OR (ORL direct, A and ORL direct, #data)
XRL		Logical XOR (XRL direct, A and XRL direct, #data)
JBC		The direct address bit is transferred by 1, and the bit is cleared (JBC bit, rel)
CPL		Invert (CPL bit)
INC		Plus 1 (INC direct)
DEC		Subtract 1. (DEC direct)
DJNZ		Subtract 1 and not transfer to 0 (DJNZ direct, rel)
MOV	bit,C	C send direct addressing bit
CLR	bit	Direct addressing bit is cleared
SETB	bit	Direct addressing bit

32. Version Revision Description

Revision	Date	Modify content
V1.00	Jun 2021	initial version
V1.01	Apr 2022	Adjust the example of the ADC clock frequency, adjust the I2C description. Adjust the port multiplexing function description, and adding the precautions of the BUZZER. Changing some registers bit description, and optimizing some text expressions.
V1.02	May 2022	1) 7.1 GPIO Function：Adjust GPIO group number 2) Modify register description: 11.2.3 Timer2 Data Register High bit TH2 13.3 Interrupt and Sleep Wakeup 20.3.3 LCD Control Register LCDCON2 21.3.18 SEG Port P0 Drive Current Selection Register 21.3.19 SEG Port P1 Drive Current Selection Register 21.3.20 SEG Port P2 Drive Current Selection Register 21.3.21 SEG Port P3 Drive Current Selection Register
V1.03	Jul 2022	1) 3.1 Power-On Reset: add the relationship table between reset flag and reset 2) 5.2 Power Monitoring Register and related chapters: adjust LVD flag description
V1.04	Aug 2022	Corrected 25.6.1 register description
V1.0.5	Jun 2023	Added some remarks to section 27.2.2
V1.0.6	July 2023	27.3.4 CRC check: add the step of clearing the CRC check end address selecton bit
V1.0.7	Jan 2024	1) Added precautions in 5.4 STOP sleep Mode 2) Correct cover content
V1.0.8	Mar 2024	1) Modify the format of the 20.3.3 register description 2) Update cover information 3) Modified the LVR low voltage reset timing diagram in section 3.3
V1.0.9	Jan 2025	Modified the cover page Revised the description of external reset in section 3.2/5.3.2.3