CMS80F751x Reference Manual

Enhanced Flash 8-bit 1T 8051- Microcontroller

Rev. 1.0.6

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 Initiation
START:
        …                   ;User Program
        …
        END                 ;End of Proceedings
```

1.2 BOOT Partition

The size of the program area is 32K*8Bit, and the program is divided into BOOT area and APROM area. The size of the BOOT area is allocated by the user configuration register.

The chip is powered on, if the program starts from the BOOT area, it needs to meet: the address space allocation method is 1/2/3 (set BOOT_1K/BOOT_2K/BOOT_4K through the CONFIG), otherwise the program will start from APROM area.

Take the BOOT area 1K space as an example: CONFIG configures BOOT_1K, after the chip is powered on and configured, the program starts to run from address 7C00H. If the program needs to switch between the BOOT area and the APROM area, write 0xAA/0x55 to the BOOT area control register BOOTCON (see register description for details), and then perform a software reset or generate a watchdog reset.

During power-on reset, external reset, and voltage reset, the BOOTCON reset value is 0x00. Software reset and watchdog reset cannot clear this register.

BOOT Control Register（BOOTCON）

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

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

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

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

Other= Invalid.

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

1) BOOTCON register needs to write AAH

```
MOV   DPTR,# BOOTCON
MOV   A,#0AAH
MOVX  @DPTR,A
```

2) Perform software reset

```
MOV   TA,#0AAH
MOV   TA,#055H
MOV   WDCON,#080H
```

For example: Use the software reset method, and then switch from the APROM area to the BOOT area, the configuration is as follows:

1) BOOTCON register needs to write 55H

```
MOV   DPTR，# BOOTCON
MOV   A,#055H
MOVX  @DPTR,A
```

2) Perform software reset

```
MOV   TA,#0AAH
MOV   TA,#055H
MOV   WDCON,#080H
```

Note: When the BOOT function is valid, the APROM program needs to ensure that the PC will not overflow (overflow means that the PC exceeds the address range of the APROM). If the PC overflows, the system may operate abnormally.

1.3 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 result of the operation.

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

1.4 B Register（B）

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

1.5 Stack Pointer Register（SP）

SP Register points to the address of the stack, after a reset, it goes to its initial values 0x07, it means that the stack area starts with the 08H of the RAM address. The value of the SP can be modified. If the stack area is set to 0xC0, the value of SP needs to be set to 0xBF after the system is reset.

Operations affecting SP:PUSH, LCALL, ACALL, POP, RET, RETI and interrupt access.

The PUSH instruction occupies one byte in the stack, LCALL, ACALL and interrupt access occupy two.

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

1.6 Data Pointer Register（DPTR0/DPTR1）

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

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

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

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

1.7 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-subtract/ self-add function selection.

00= DPTR0 add 1 or DPTR1 add 1;

01= DPTR0 minus 1 or DPTR1 add 1;

10= DPTR0 add 1 or DPTR1 minus 1;

11= DPTR0 minus 1or DPTR1 minus 1.

Bit5 TSL: Flip selection enable;

1= After the DPTR command is executed, the SEL will flip automatically;

0= DPTR related instructions do not affect SEL.

Bit4 AU: Self-add / self-subtract enable;

1= After the MOVX @ DPTR or MOVC @ DPTR instructions are allowed to run, the operation of self-add / self-subtract (depends on ID1 - ID0) is performed.

0= DPTR related instructions does not affect SEL.

Bit3~Bit1 -- Reserved, all must be 0.

Bit0 SEL: Data pointer selection;

1= Select DPTR1;

0= Select DPTR0.

1.8 Program Status Register（PSW）

Program Status Register PSW

0xD0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PSW	CY	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 bit);
1= Carry;
0= No carry.

Bit5 F0: General Purpose Flag.

Bit4~Bit3 RS<1:0>: Work Register BANK Select;
00= Select Bank0;
01= Select Bank1;
10= Select Bank2;
11= Select Bank3.

Bit2 OV: Overflow Flag;
1= Arithmetic or logical operation has overflow;
0= Arithmetic or logical operation has no overflow.

Bit1 -- Reserved, must be 0.

Bit0 P: Parity;
1= The highest bit of the result is carried.
0= The highest bit of the result does not carry.

1.9 Program Counter（PC）

Program Counter(PC) controls the order in which instructions are executed in the program memory Flash, it can address the entire range of Flash. After obtaining the script, the PC will automatically add 1 to point to the address of the next script. But if operations such as jump, conditional jump, subroutine call, initialization reset, interrupt, interrupt return, subroutine return are performed, 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 will be discarded during the current instruction execution, and an empty instruction operation cycle will be inserted before the correct instruction can be obtained. On the contrary, the next instruction will be executed sequentially.

1.10 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.

Some protected registers must perform the following operations on TA before they can be written.

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

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

Protected register: WDCON，CLKDIV，SCKSEL.

2. Memory And Register Map

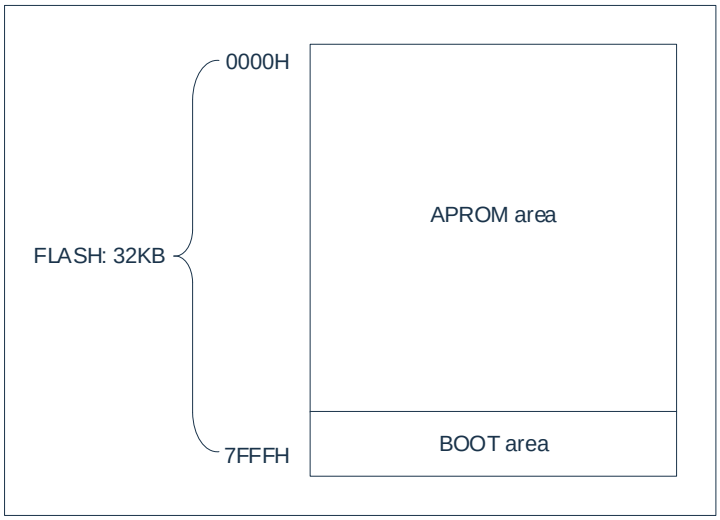
This series of micro-controllers has the following types of memories:

- Maximum 32KB FLASH program memory (shared by APROM area and BOOT area).
- Maximum 1KB non-volatile data memory (Data FLASH).
- Maximum 256B general-purpose internal data memory (RAM).
- Maximum 2KB general-purpose external data memory (XRAM).
- Special function register SFR (BANK0 and BANK1).
- External special function register XSFR.

2.1 Program Memory FLASH

The program memory FLASH is used to store the source program and table data, and the program counter PC is used as the address pointer. The PC is a 16-bit program counter, so the address space that can be addressed is 64KB, but this chip has only 32K bytes of program storage space.

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

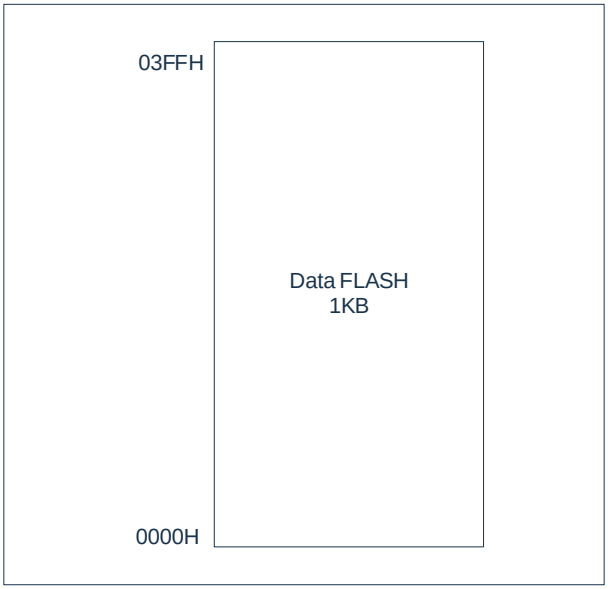


After the chip is reset, the CPU starts to execute from 0000H. Each interrupt is assigned a fixed address in the program memory, and the interrupt causes the CPU to jump to this address and start executing the service program.

For example, external interrupt 1 is assigned address 0013H. If external interrupt 1 is used, its service program must start from 0013H. If this interrupt is not used, its service address will be used as a normal program storage address.

2.2 Non-volatile Data Memory Data FLASH

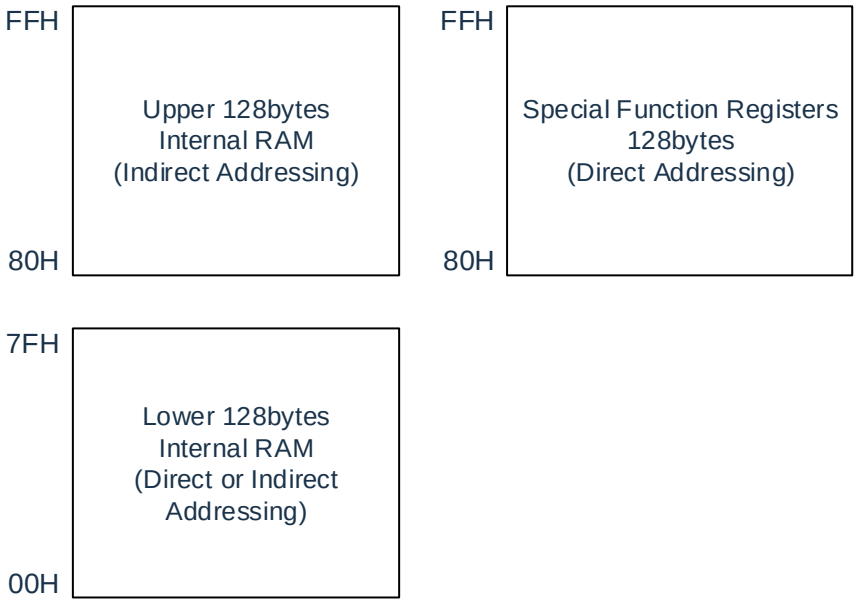
The non-volatile data memory Data FLASH can be used to store important data such as constant data, calibration data, protection and safety-related information. The data stored in this area has the characteristic that the data will not be lost when the chip is powered off or suddenly or unexpectedly.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



The reading, writing, and erasing operations of Data FLASH memory are realized through the FLASH control interface.

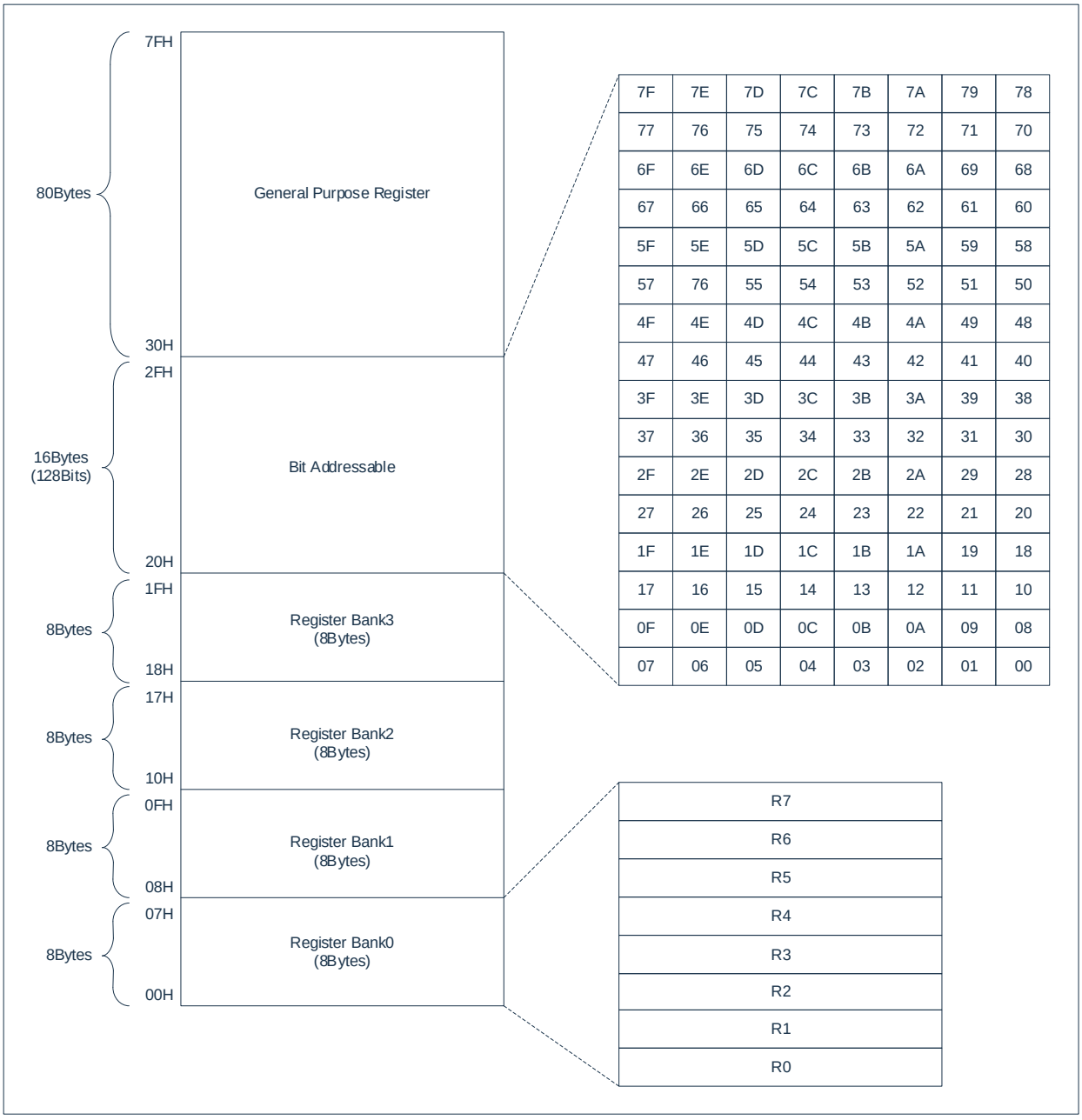
2.3 General Data Register RAM

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:



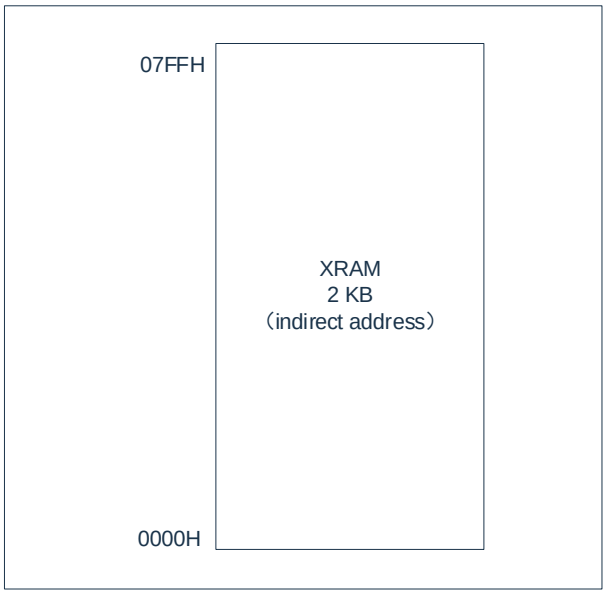
The high 128Bytes and SFR shown in the figure above occupy the same area (80H~FFH), but they are independent. Directly address the storage space (SFR) higher than 7FH and indirectly address the storage space higher than 7FH (high 128Bytes) into different storage spaces. SFR is divided into two pages,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

The lower 128Bytes 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 by byte or directly on each bit in the unit. For the remaining 80 storage units (30H~7FH), users 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 maximum 2KB 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 the XRAM address 01H, the upper 8 bits are determined by DPH0/1
```

When Target-->Memory Model is set to Large in Keil51, the C compiler will use XRAM as the variable address. General use the DPTR performs XRAM/XSFR operations.

2.5 Special Function Register Table SFR

Special function registers refer 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 lower four bits of the address are 0000 or 1000 for bit addressing, such as P0, TCON, and 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 have 35 addresses, which are different registers in BANK0 and BANK1, which are accessed through the paging function; other addresses correspond to BANK0 and BANK1. The same register can be accessed through BANK0 and BANK1.

It should be noticed 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 are required 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;

Other= No Access

The BANK0 register table is as follows:

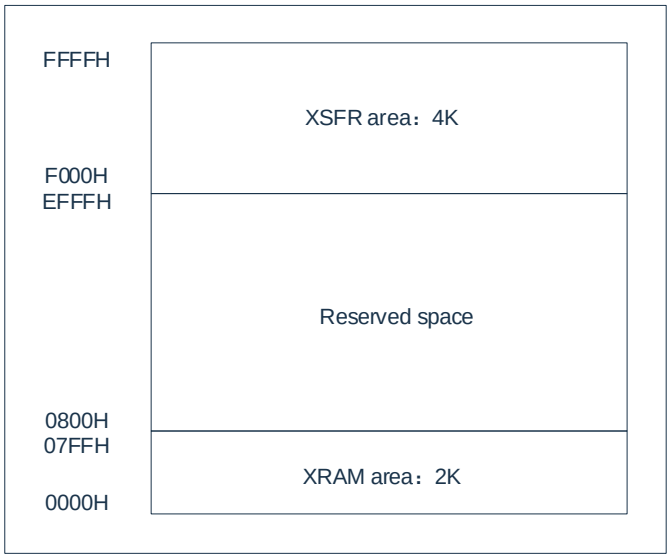
BANK0	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
0xF8	--	PCRCDL	PCRCDH	MLOCK	MADRL	MADRH	MDATA	MCTRL
0xF0	B	I2CSADR	I2CSCR	I2CSBUF	I2CMSA	I2CMCR	I2CMBUF	I2CMTP
0xE8	--	ADCON2	SCON1	SBUF1	SPCR	SPSR	SPDR	SSCR
0xE0	ACC	--	TL4	TH4	--	--	--	--
0xD8	P5	ADCCHS	TL3	TH3	ADRESL	ADRESH	ADCON1	ADCON0
0xD0	PSW	ADCMPC	T34MOD	ADDLYL	ADCMPL	ADCMPH	SCKSEL	SCKSTAU
0xC8	T2CON	T2IF	RLDL	RLDH	TL2	TH2	CCEN	T2IE
0xC0	--	--	CCL1	CCH1	CCL2	CCH2	CCL3	CCH3
0xB8	IP	EIP1	EIP2	EIP3	WUTCRL	WUTCRH	BUZDIV	BUZCON
0xB0	--	--	EIF2	--	P0EXTIF	P1EXTIF	P2EXTIF	--
0xA8	IE	--	EIE2	--	P0EXTIE	P1EXTIE	P2EXTIE	--
0xA0	P2	P1TRIS	P2TRIS	--	--	P5TRIS	--	P5EXTIF
0x98	SCON0	SBUF0	P0TRIS	--	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:

BANK1	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
0xF8	--	PCRCDL	PCRCDH	MLOCK	MADRL	MADRH	MDATA	MCTRL
0xF0	B	--	--	--	--	--	--	--
0xE8	--	MD0	MD1	MD2	MD3	MD4	MD5	ARCON
0xE0	ACC	--	--	--	--	--	--	--
0xD8	P5	--	--	--	--	--	--	--
0xD0	PSW	--	--	--	--	--	--	--
0xC8	T2CON	T2IF	RLDL	RLDH	TL2	TH2	CCEN	T2IE
0xC0	--	--	CCL1	CCH1	CCL2	CCH2	CCL3	CCH3
0xB8	IP	EIP1	EIP2	EIP3	WUTCRL	WUTCRH	BUZDIV	BUZCON
0xB0	--	--	EIF2	--	P0EXTIF	P1EXTIF	P2EXTIF	--
0xA8	IE	--	EIE2	--	P0EXTIE	P1EXTIE	P2EXTIE	--
0xA0	P2	P1TRIS	P2TRIS	--	--	P5TRIS	--	P5EXTIF
0x98	SCON0	SBUF0	P0TRIS	--	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 address space and XRAM. It mainly includes: port control register and other function control registers. Seeking the address range is as follows:



The list of external special function registers is as follows:

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

Address	Registers	Register Description
F01DH	P1SR	P1 port slope control register
F01EH	P1DS	P1 port data input selection register
--	--	--
F020H	P20CFG	P20 port configuration register
F021H	P21CFG	P21 port configuration register
F022H	P22CFG	P22 port configuration register
F023H	P23CFG	P23 port configuration register
F024H	P24CFG	P24 port configuration register
F025H	P25CFG	P25 port configuration register
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
--	--	--
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

Address	Registers	Register Description
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
--	--	--
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
F0BEH	--	--
F0BFH	--	--
--	--	--
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 control input port allocation register
F0C4H	PS_T1	Timer1 external clock input port allocation register
F0C5H	PS_T1G	Timer1 gate control input port allocation register
F0C6H	PS_T2	Timer2 external event or gate control input port allocation register
F0C7H	PS_T2EX	Timer2 falling edge automatically reloads the 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_FB0	PWM external brake signal 0 port allocation register
F0CEH	PS_FB1	PWM external brake signal 1 port allocation register
--	--	--
F120H	PWMCON	PWM control register
F121H	PWMOE	PWM output enable register
F122H	PWMPINV	PWM output polarity selection register
F123H	PWM01PSC	PWM0/1 clock prescaler control register
F124H	PWM23PSC	PWM2/3 clock prescaler control register
F125H	PWM45PSC	PWM4/5 clock prescaler control register
F126H	PWMCNTE	PWM count start control register
F127H	PWMCNTM	PWM count mode selection register
F128H	PWMCNTCLR	PWM counter clear control register
F129H	PWMLOADEN	PWM load enable control register

Address	Registers	Register Description
F12AH	PWM0DIV	PWM0 frequency division control register
F12BH	PWM1DIV	PWM1 frequency division control register
F12CH	PWM2DIV	PWM2 frequency division control register
F12DH	PWM3DIV	PWM3 frequency division control register
F12EH	PWM4DIV	PWM4 frequency division control register
F12FH	PWM5DIV	PWM5 frequency division control register
F130H	PWMP0L	PWM0 Cycle low 8-bit register
F131H	PWMP0H	PWM0 Cycle high 8-bit register
F132H	PWMP1L	PWM1 Cycle low 8-bit register
F133H	PWMP1H	PWM1 Cycle high 8-bit register
F134H	PWMP2L	PWM2 Cycle low 8-bit register
F135H	PWMP2H	PWM2 Cycle high 8-bit register
F136H	PWMP3L	PWM3 Cycle low 8-bit register
F137H	PWMP3H	PWM3 Cycle high 8-bit register
F138H	PWMP4L	PWM4 Cycle low 8-bit register
F139H	PWMP4H	PWM4 Cycle high 8-bit register
F13AH	PWMP5L	PWM5 Cycle low 8-bit register
F13BH	PWMP5H	PWM5 Cycle high 8-bit register
--	--	--
F140H	PWMD0L	PWM0 Compare Data low 8-bit register
F141H	PWMD0H	PWM0 Compare Data high 8-bit register
F142H	PWMD1L	PWM1 Compare Data low 8-bit register
F143H	PWMD1H	PWM1 Compare Data high 8-bit register
F144H	PWMD2L	PWM2 Compare Data low 8-bit register
F145H	PWMD2H	PWM2 Compare Data high 8-bit register
F146H	PWMD3L	PWM3 Compare Data low 8-bit register
F147H	PWMD3H	PWM3 Compare Data high 8-bit register
F148H	PWMD4L	PWM4 Compare Data low 8-bit register
F149H	PWMD4H	PWM4 Compare Data high 8-bit register
F14AH	PWMD5L	PWM5 Compare Data low 8-bit register
F14BH	PWMD5H	PWM5 Compare Data high 8-bit register
--	--	--
F150H	PWMDD0L	PWM0 Down Compare Data Low 8-bit Register
F151H	PWMDD0H	PWM0 Down Compare Data High 8-bit Register
F152H	PWMDD1L	PWM1 Down Compare Data Low 8-bit Register
F153H	PWMDD1H	PWM1 Down Compare Data High 8-bit Register
F154H	PWMDD2L	PWM2 Down Compare Data Low 8-bit Register
F155H	PWMDD2H	PWM2 Down Compare Data High 8-bit Register
F156H	PWMDD3L	PWM3 Down Compare Data Low 8-bit Register
F157H	PWMDD3H	PWM3 Down Compare Data High 8-bit Register
F158H	PWMDD4L	PWM4 Down Compare Data Low 8-bit Register
F159H	PWMDD4H	PWM4 Down Compare Data High 8-bit Register
F15AH	PWMDD5L	PWM5 Down Compare Data Low 8-bit Register
F15BH	PWMDD5H	PWM5 Down Compare Data High 8-bit Register
--	--	--
F160H	PWMDTE	PWM Dead Time Enable Control Register
F161H	PWM01DT	PWM0/1 Dead-band Delay Data Register

Address	Registers	Register Description
F162H	PWM23DT	PWM2/3 Dead-band Delay Data Register
F163H	PWM45DT	PWM4/5 Dead-band Delay Data 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 Periodic interrupt mask register
F169H	PWMZIE	PWM Zero Interrupt Mask Register
F16AH	PWMUIE	PWM Compare Up Interrupt Mask Register
F16BH	PWMDIE	PWM Compare Down Interrupt Mask Register
F16CH	PWMPIF	PWM Periodic Interrupt Flag Register
F16DH	PWMZIF	PWM Zero Interrupt Flag Register
F16EH	PWMUIF	PWM Compare Up Interrupt Flag Register
F16FH	PWMDIF	PWM Compare Down Interrupt Flag Register
--	--	--
F500H	C0CON0	Comparator 0 Control Register 0
F501H	C0CON1	Comparator 0 Control Register 1
F502H	C0CON2	Comparator 0 Control Register 2
F503H	C1CON0	Comparator 1 Control Register 0
F504H	C1CON1	Comparator 1 Control Register 1
F505H	C1CON2	Comparator 1 Control Register 2
F506H	CNVRCON	Comparator Reference Voltage Control Register
F507H	CNFBCON	Comparator Brake Control Register
F508H	CNIE	Comparator Interrupt Mask Register
F509H	CNIF	Comparator Interrupt Flag Register
F50AH	C0ADJE	Comparator 0 Adjustment Bit Selection Register
F50BH	--	--
F50CH	C0HYS	Comparator 0 Hysteresis Control Register
F50DH	C1HYS	Comparator 1 Hysteresis Control Register
--	--	--
F520H	OP0CON0	OPAMP 0 Control Register 0
F521H	OP0CON1	OPAMP 0 Control Register 1
F522H	--	--
F523H	OP1CON0	OPAMP 1 Control Register 0
F524H	OP1CON1	OPAMP 1 Control Register 1
F525H	--	Unused
F526H	OP0ADJE	OPAMP 0 Adjustment Bit Selection Register
F527H	OP1ADJE	OPAMP 1 Adjustment Bit Selection Register
--	--	--
F5C0H	BRTCON	BRT module control register
F5C1H	BRTDL	BRT timer data load value lower 8 bits
F5C2H	BRTDH	BRT timer data load value high 8 bits
--	--	--
F5E0H	UID0	UID<7:0>
F5E1H	UID1	UID<15:8>
F5E2H	UID2	UID<23:16>
F5E3H	UID3	UID<31:24>

Address	Registers	Register Description
F5E4H	UID4	UID<39:32>
F5E5H	UID5	UID<47:40>
F5E6H	UID6	UID<55:48>
F5E7H	UID7	UID<63:56>
F5E8H	UID8	UID<71:64>
F5E9H	UID9	UID<79:72>
F5EAH	UID10	UID<87:80>
F5EBH	UID11	UID<95:88>
--	--	--
F650H	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
--	--	--
F680H	LCDCON0	LCD Control Register 0
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 Register 0（SEG7-SEG0）
F686H	LCDSEGEN1	LCD SEG Port Enable Register 1（SEG15-SEG8）
F687H	LCDSEGEN2	LCD SEG Port Enable Register 2（SEG19-SEG16）
--	--	--
F690H	LVDCON	Power monitoring register
F691H	BOOTCON	BOOT control register
F692H	ADCLDO	ADC reference voltage control register
F693H	TS_REG	Temperature sensor register
F694H	LSECRL	LSE timer data register low 8-bit
F695H	LSECRH	LSE timer data register high 8-bit-
F696H	LSECON	LSE timer control register
F697H	XT_SCM	LSE/HSE clock stop oscillation detection control register

Address	Registers	Register Description
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
F69FH	PS_RXD0	UART0 data input port allocation register
--	--	--
F708H	CRCIN	CRC module data input register
F709H	CRCDL	CRC operation result low 8-bit data register
F70AH	CRCDH	CRC operation result high 8-bit data register
--	--	--
F711H	LEDSDRP0H	LED SEG port P04-P07 drive source current selection register
F712H	LEDSDRP1L	LED SEG port P10-P13 drive source current selection register
F713H	LEDSDRP1H	LED SEG port P14-P17 drive source current selection register
F714H	LEDSDRP2L	LED SEG port P20-P23 drive source current selection register
F715H	LEDSDRP2H	LED SEG port P24-P27 drive source current selection register
--	--	--
F740H	LEDC0DATA0	LED COM0 corresponds to SEG7-SEG0 data register
F741H	LEDC0DATA1	LED COM0 corresponds to SEG15-SEG8 data register
F742H	LEDC0DATA2	LED COM0 corresponds to SEG19-SEG16 data register
--	--	--
F744H	LEDC1DATA0	LED COM1 corresponds to SEG7-SEG0 data register
F745H	LEDC1DATA1	LED COM1 corresponds to SEG15-SEG8 data register
F746H	LEDC1DATA2	LED COM1 corresponds to SEG19-SEG16 data register
--	--	--
F748H	LEDC2DATA0	LED COM2 corresponds to SEG7-SEG0 data register
F749H	LEDC2DATA1	LED COM2 corresponds to SEG15-SEG8 data register
F74AH	LEDC2DATA2	LED COM2 corresponds to SEG19-SEG16 data register
--	--	--
F74CH	LEDC3DATA0	LED COM3 corresponds to SEG7-SEG0 data register
F74DH	LEDC3DATA1	LED COM3 corresponds to SEG15-SEG8 data register
F74EH	LEDC3DATA2	LED COM3 corresponds to SEG19-SEG16 data register
--	--	--
F750H	LEDC4DATA0	LED COM4 corresponds to SEG7-SEG0 data register
F751H	LEDC4DATA1	LED COM4 corresponds to SEG15-SEG8 data register
F752H	LEDC4DATA2	LED COM4 corresponds to SEG19-SEG16 data register
--	--	--
F754H	LEDC5DATA0	LED COM5 corresponds to SEG7-SEG0 data register
F755H	LEDC5DATA1	LED COM5 corresponds to SEG15-SEG8 data register
F756H	LEDC5DATA2	LED COM5 corresponds to SEG19-SEG16 data register
--	--	--
F758H	LEDC6DATA0	LED COM6 corresponds to SEG7-SEG0 data register
F759H	LEDC6DATA1	LED COM6 corresponds to SEG15-SEG8 data register
F75AH	LEDC6DATA2	LED COM6 corresponds to SEG19-SEG16 data register
--	--	--

Address	Registers	Register Description
F75CH	LEDC7DATA0	LED COM7 corresponds to SEG7-SEG0 data register
F75DH	LEDC7DATA1	LED COM7 corresponds to SEG15-SEG8 data register
F75EH	LEDC7DATA2	LED COM7 corresponds to SEG19-SEG16 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 SEG19-SEG16 enable register 2
--	--	--
F765H	LEDCON	LED control register
F766H	LEDCLKL	LED clock prescaler data register low 8-bit
F767H	LEDCLKH	LED clock prescaler data register high 8-bit
F768H	LEDCOMTIME	LED COM port effective time selection register

3. Reset

The reset time (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. This reset time will exist regardless of whether the chip is reset by power-on or other reasons. 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;
- Software reset;
- 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 the program counter PC will be cleared at the same time. After the reset, the program will start to run from the reset vector 0000H.

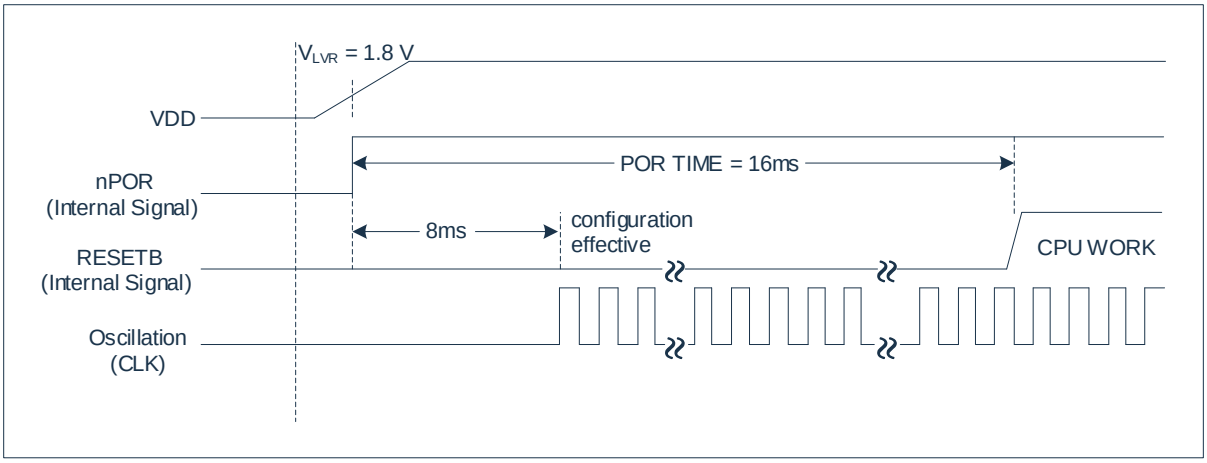
Any kind of reset situation requires a certain response time, and the system provides a complete reset process to ensure the smooth progress of the reset action.

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;
- Execute the program: after 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. The power-on reset sequence diagram is shown in the following figure:



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

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

Bit7 SWRST: Software reset control;
1: Perform a 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~Bit4 --: Reserved, all must be 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;

1= Enable WDT to reset the CPU;

0= Disable WDT to reset CPU.

Bit0 WDTCLR: WDT counter clear;

1= Clear the WDT counter (automatically cleared by hardware);

0= Disable the 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. If the NRST pin is held low for more than 16us (three rising edges of the internal LSI clock sampling) during the operating voltage range and stable oscillation, a reset will be requested.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.

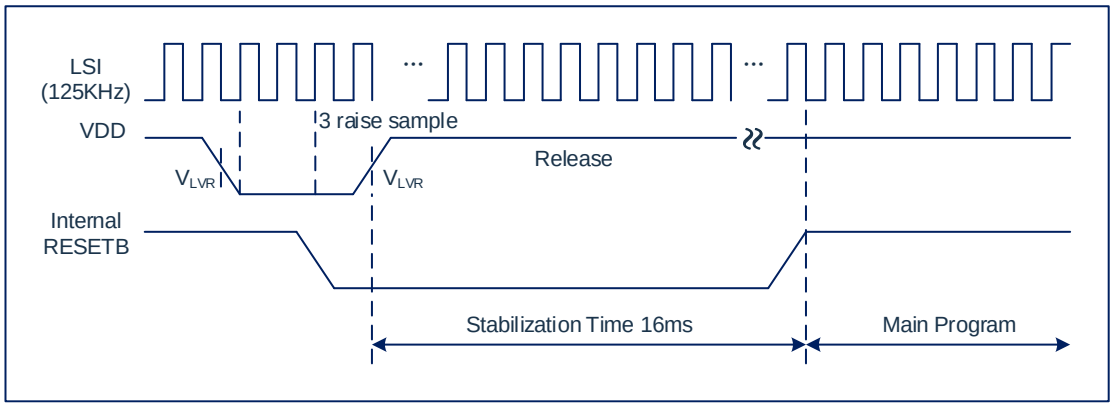
During the Stabilization Time, the chip is reconfigured, which is the same as the power-on reset configuration process. The external reset pin NRST and its pull-up resistor are enabled through CONFIG configuration.

3.3 LVR Low Voltage Reset

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

The LVR module detects VDD<VLVR, a reset is requested. In the sleep mode (STOP) mode, the LVR low voltage reset function is disabled.

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



During the Stabilization Time, the chip is reconfigured, which is the same as the power-on reset configuration process.

3.4 Watchdog Reset (WDT)

The watchdog reset is a protection setting for 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 and the watchdog timer overflows and the system is reset. After the watchdog resets, the system restarts to the normal state.

The WDT counters are not addressable and start counting when the Power-on Reset is running. It is recommended to clear the WDT counter when setting the WDT register to accurately control the WDT time-out.

The timing of the watchdog reset is as follows:

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

The clock source of 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 WDS2-WTS0 of the CKCON register. The watchdog overflow time is shown in the following table:

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

The WDT can also be set to not reset the system and can generate an interrupt.

3.5 Software Reset

Program software reset can be implemented inside the chip. Software reset can relocate the program flow to reset address 000H, and then run the program again. The user can write the software reset control bit WDCON[7] (SWRST=1) to realize a custom software reset. The software reset will not restart the power-on reset configuration.

3.6 CONFIG Status Protection Reset

The CONFIG state protection reset is an enhanced protection mechanism for the system. On a power-on reset, there is a set of 16-bit CONFIG registers internally that load the fixed code (A569H) set in the FLASH, which is not manipulated during normal operation. If in the case of a specific non-program operation, the value of this register changes and is not equal to the original fixed code, after several clock samples, the register continues to remain in a state of not being fixed code, and the system will generate a reset.

This reset mechanism prevents the configuration bits from changing under certain conditions, causing the system to enter an unpredicted state.

In normal operation, the clock of the sampling register value is the internal RC fixed clock Fixed Clock (8MHz, the clock source is from HSI) and the 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.

In order to prevent the oscillator from oscillating under certain conditions, two clocks are used for sampling.

3.7 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. If it exceeds the set time , The monitoring circuit resets the configuration module and allows the configuration module to perform the configuration process again. This prevents the system from entering an unpredictable state when it is powered on.

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

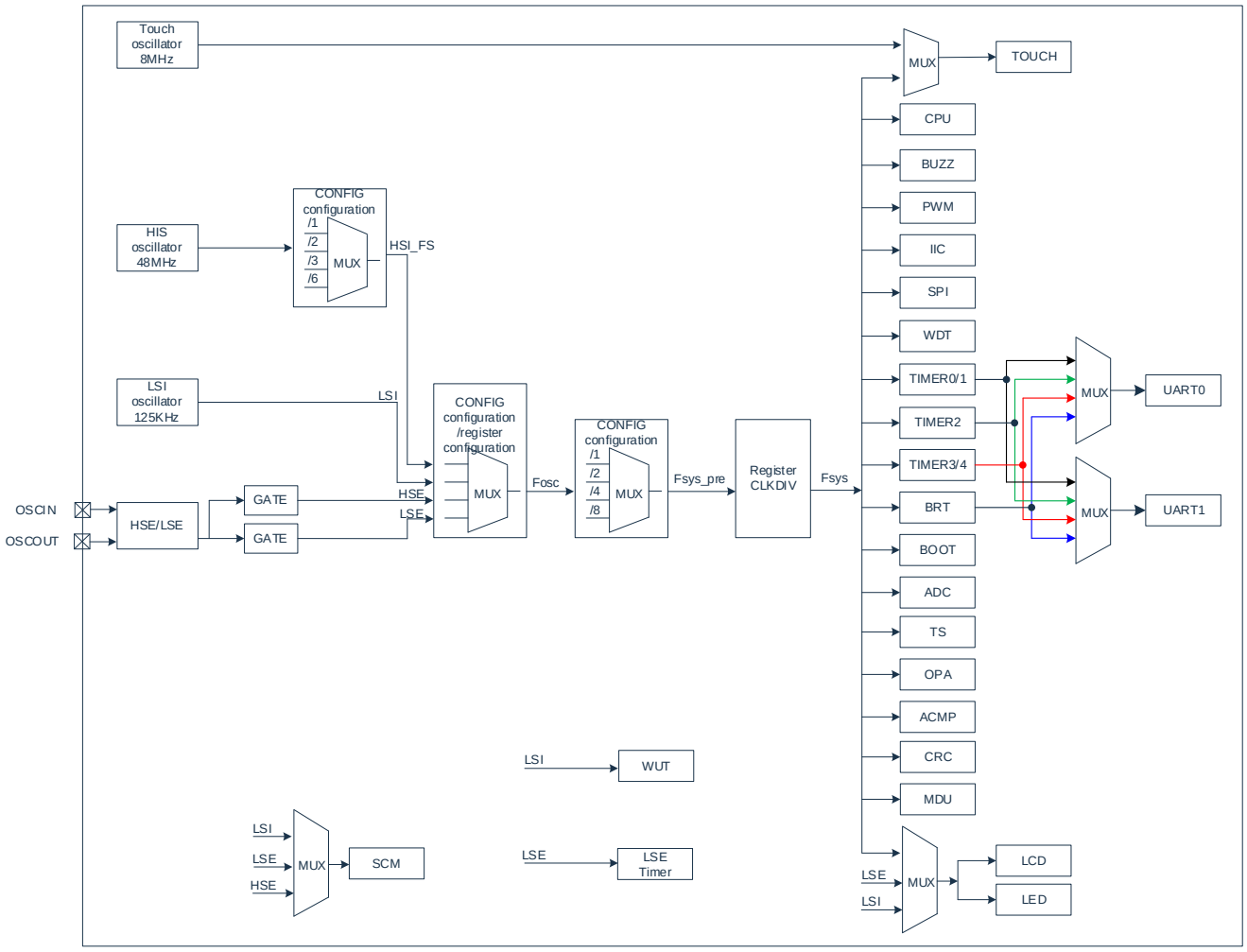
The clock source of the system clock has 4 types, and the clock source and clock frequency division can be selected through the setting of the system configuration register or the user register. The system clock sources are as follows:

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

The default clock source of the chip is HSI, and the system clock will run in HSI after the chip reset is completed. If you need to change the system clock source, you can set it through the system configuration register (you need to operate with the burning tool and the host computer software), or through the user register (set the relevant registers according to the operation steps, see below for details). When the external high-speed and low-speed oscillators are selected as the system clock, the vibration stop monitoring function is supported, see below for details.

4.1 System Clock Structure

The block diagram of the system clock structure of each peripheral module 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>: Frequency division bit of system clock Fsys;

00H= Fsys=Fsys_pre;

other= Fsys=Fsys_pre/（2*CLKDIV）（2,4…510 frequency 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 Switch Register SCKSEL

0xD6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SCKSEL	--	--	--	SEL	WRITE	CKSEL2	CKSEL1	CKSEL0
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 -- No Access.

Bit6~Bit5 -- Reserved, all must be 0.

Bit4 SEL: System clock configuration selection;

1= Write: subject to the clock source configured by CKSEL<2:0>

Read: This bit has been written with 1 (after writing 1, wait for 6 system clock cycles <6*NOP> to ensure that the clock source control selection changes)

0= Read: Subject to the clock source configured in CONFIG (default selection after power-on).

Write: Write 0 operation is prohibited

(After power-on allows writing 1 operation, then writing 0 is prohibited, that is, only one CONFIG configuration clock source is allowed to change to CKSEL<2:0> control clock source. If you need to be controlled by CONFIG again, you need to perform power-on reset configuration again) ;

Bit3 WRITE: Write enable, generate a pulse to switch the clock (only valid for the clock source selected by CKSEL<2:0>);

1= Switch clock (you need to wait for the target clock source to be switched to stable position 1 before writing 1);

0= Do not switch the clock.

Bit2~Bit0 CKSEL<2:0>: System clock source selection;

111= LSI;

110= LSE;

101= HSE;

100= HSI;

Other= Invalid value, access is forbidden.

After the clock source is switched, the system will switch successfully within a few system clock cycles. It is recommended that the program execute 6*NOP before executing other instructions.

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	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit7 LSI_F: Low-speed internal stable state;
1= Stable;
0= Not stable.

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

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

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

Bit3 -- No Access.

Bit2~Bit0 -- Reserved, all must be 0.

4.2.4 System Clock Monitor Register SCM

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

Bit7 SCMEN: The vibration stop detection module is enabled;
1= Enable;
0= Disable.

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

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

Bit1 SCMIF: Stop vibration interrupt flag;
1= Indicates stop vibration;
0= Cleared by software. After clearing, it will automatically switch to the HSE/LSE frequency (only cleared by software).

Bit0 SCMSTA: Oscillation status bit, read only;
1= Indicates stop vibration;
0= Stop vibration and resume.

Note:

1) Both SCMIF and SCMSTA can reflect the status of HSE/LSE as the 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 HSE/LSE stop oscillating, but it can generate interrupts (need to turn on the interrupt enable), SCMIF can also be cleared through the register. After clearing, the main frequency will switch back to HSE/LSE (if the oscillation is still stopped 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 the HSE/LSE is restored, SCMSTA will be automatically cleared, and the main frequency will be automatically switched back to HSE/LSE by HSI.

4.2.5 Function Clock Control Register

Watchdog overflow time/timer clock source selection register CKCON

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

Bit7~Bit5 WTS<2:0>: WDT overflow time selection;

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;

0= Fsys/12;

1= Fsys/4.

Bit3 T0M: Timer0 clock source selection;

0= Fsys/12;

1= Fsys/4.

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

UART0/1 Baud Rate Selection Register FUNCCR

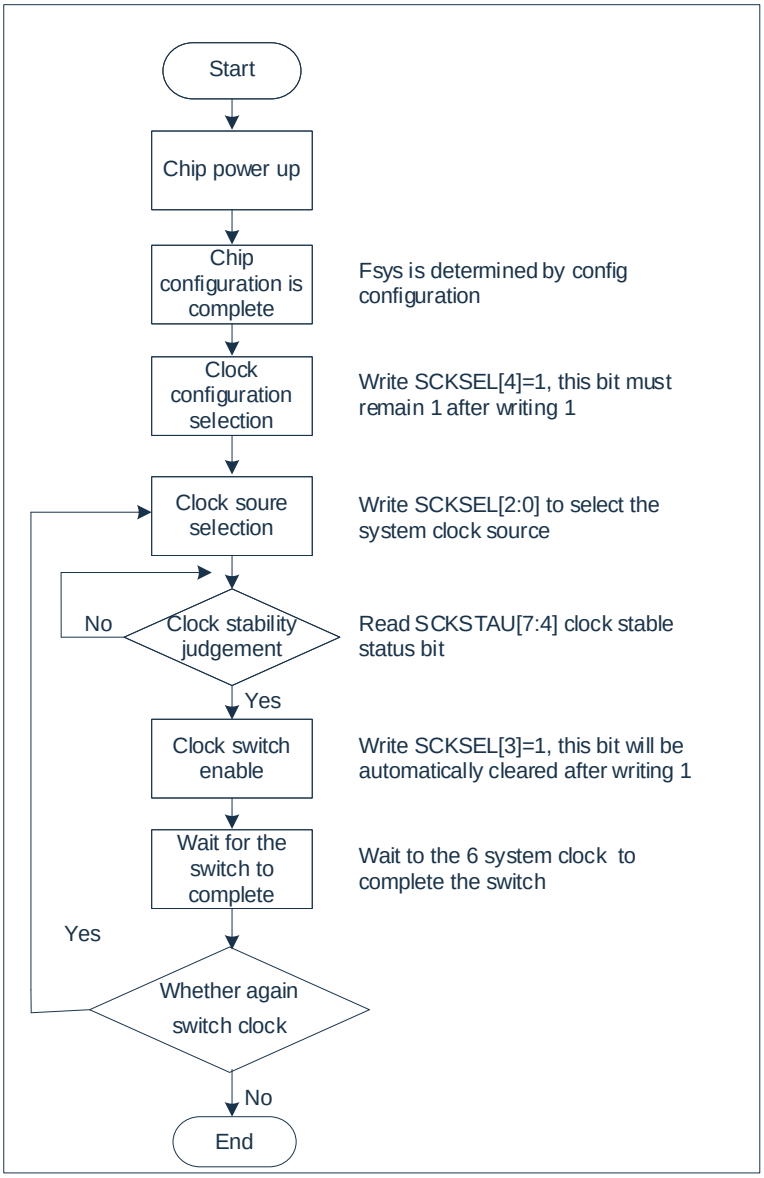
0x91	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FUNCCR	--	UART1_CKS2	UART1_CKS1	UART1_CKS0	--	UART0_CKS2	UART0_CKS1	UART0_CKS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit7	--	Reserved, must be 0.
Bit6~Bit4	UART1_CKS<2:0>:	Timer clock source selection of UART1;
	000=	Timer1 overflow clock;
	001=	Timer4 overflow clock;
	010=	Timer2 overflow clock;
	011=	BRT Overflow clock;
	Other=	No Access.
Bit3	--	Reserved, must be 0.
Bit2~Bit0	UART0_CKS<2:0>:	Timer clock source selection of UART0;
	000=	Timer1 overflow clock;
	001=	Timer4 overflow clock;
	010=	Timer2 overflow clock;
	011=	BRT Overflow clock;
	Other=	No Access.

4.3 System Clock Switching

The crystal oscillator port of the chip can only be connected with HSE or LSE, so it is forbidden to use the HSE/LSE mutual switching function. When the current chip selects the external HSE clock (config_03H_Bit7=0), it is forbidden to use LSE related functions. When the current chip selects the external LSE clock (config_03H_Bit7=1), it is forbidden to use HSE related functions. If the user misuses, the error configuration status bit of the register SCKSTAU[4] will be set to 1, and the user needs to change the code or configuration after judging that the error flag is set.

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



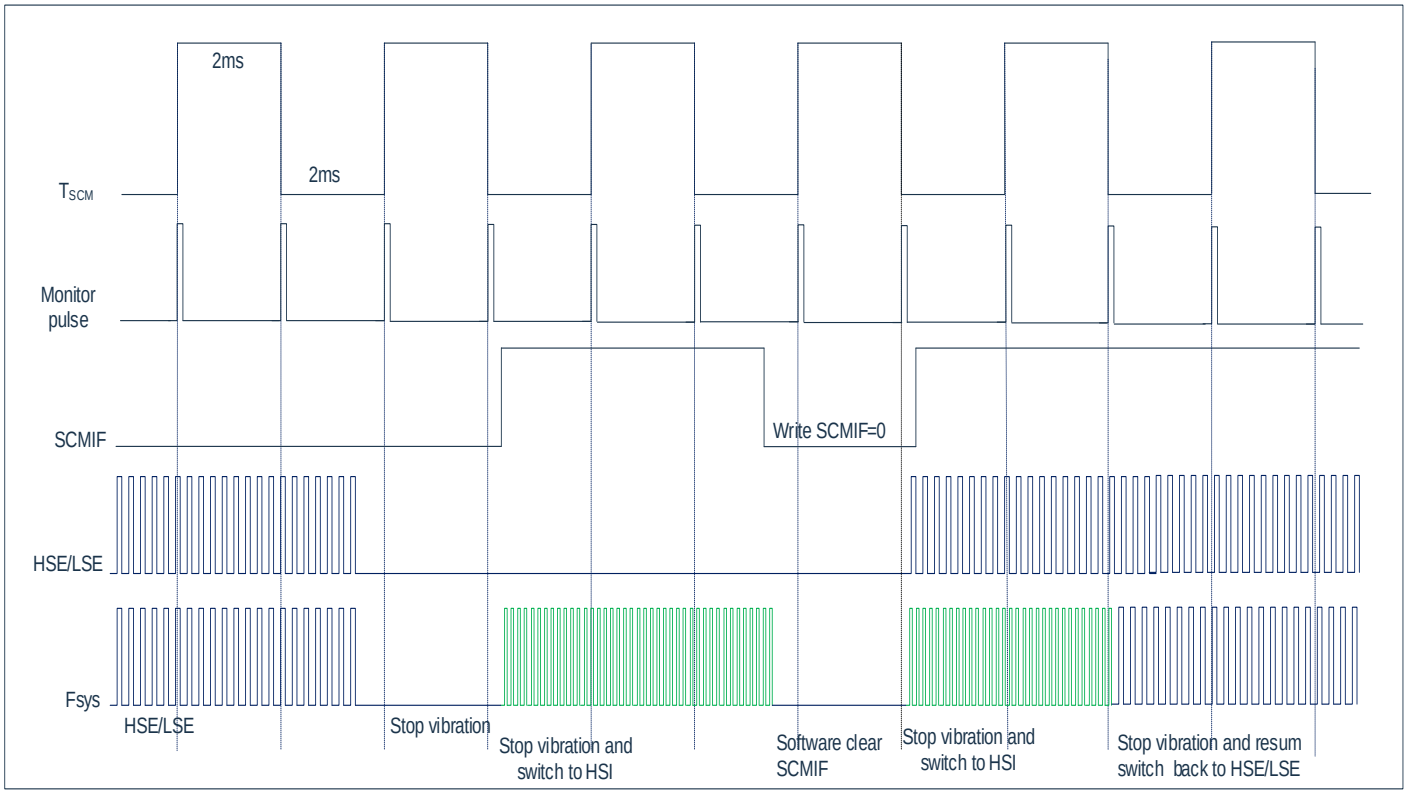
Note: When the system clock source is switched, it must be ensured that the corresponding clock source exists and has been correctly configured.

4.4 System Clock Monitoring

System clock monitoring (SCM: 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 and 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 TSCM during the monitoring period is 1:1. SCM monitors HSE/LSE oscillation when TSCM is at high level, and processes the monitoring results when TSCM 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

The low power modes are divided into 2 types:

- IDLE: Idle mode
- STOP: Sleep mode

When the user is developing the program, it is strongly recommended to use the IDLE and STOP macros to control the system mode of the microcontroller. Do not set it directly IDLE and STOP bits. The macro is as follows:

Enter idle mode: IDLE();

Enter sleep mode: STOP();

5.1 Power Management Register PCON

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

Bit7 SMOD0: UART0 baud rate multiplication;
0= UART0 baud rate is normal;
1= UART0 doubles the baud rate.

Bit6 SMOD1: UART1 baud rate multiplication;
0= UART1 baud rate is normal;
1= UART1 doubles the baud rate.

Bit5~Bit3 -- Reserved, all must be 0.

Bit2 SWE: STOP status function wake-up enable;
(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= Allow Function wake-up(wake-up by external interrupt and timed wake-up).

Bit1 STOP: Sleep mode control;
0= has not entered the sleep state;
1= Enters the sleep state (exit STOP auto-zero).

Bit0 IDLE: idle state control;
0= has not entered the idle state;
1= Enters the idle state (exit IDLE mode is automatically cleared).

5.2 Power Monitoring Register LVDCON

The 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, when the power supply voltage drops below the LVD setting value, an interrupt will be generated to remind the user.

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

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

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

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

Bit3 LVDEN: LVD Enable;

0= Prohibit;

1= Enable.

Bit2 -- Reserved, must be 0.

Bit1 LVDINTE: LVD interrupt enable;

0= LVD interrupt is disabled;

1= LVD interrupt enable.

Bit0 LVDINTF: LVD interrupt flag;

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

1= The power supply voltage is lower than the monitoring voltage (cleared by software).

5.3 IDLE Mode

In this mode, only the CPU clock source is turned off. Therefore, in this state, peripheral functions (such as timer, PWM, and I2C) and clock generator (HSI/Crystal oscillator) still work normally.

After the system enters the idle mode, it can be awakened by any interrupt. After waking up, it enters the interrupt handler. After the interrupt returns, the next instruction is executed.

If the idle mode is entered in the interrupt service routine, only the higher priority interrupts can wake up the system.

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.

5.4.1 Sleep And 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 from sleep mode:

1) INT0/1 interrupt

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

2) External (GPIO) interrupt

When using external GPIO interrupt to wake up, the total interrupt enable and port interrupt enable must be turned on before entering sleep to wake up the system. External GPIO interrupt wake-up can choose rising edge, falling edge, and dual-edge interrupt to wake up sleep. The interrupt wake-up edge is set by the external interrupt control register PxnEICFG.

3) WUT timed wake-up

To wake up by WUT regularly, the timing wake-up function must be turned on before entering 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 turned on, the LSI is automatically turned on in the sleep state.

4) LSE timed wake-up

Wake up by LSE regularly, you must enable the LSE module , count module, and timer wake-up functions before entering sleep, and set the time from sleep to wake-up at the same time.

5.4.2 Wake Waiting Status

Whether it is INT0/1 interrupt, external GPIO interrupt, WUT timed wake up, LSE timed wake up sleep mode, after the interrupt is generated or the timing time is up, you need to wait for a period of time to wake up the system and execute the next instruction 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 waiting 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 And Wake-Up Time

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

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

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

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

(The conditions for the time given above are Fsys > 1MHz)

5.4.4 Reset Operation During Sleep

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

Power-down reset: No other conditions are required. After VDD drops to 0V, power on again to the working voltage and enter the power-on reset state.

External reset: The external reset function needs to be turned on, and the related ports are configured as dedicated reset ports. The reset port maintains a low level of >1us during sleep, and the system generates a reset. Release the reset port, 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 current measured 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/O. If there are floating I/O ports in the user's plan, set all the floating ports as output ports to ensure Each input port has a fixed state to avoid that 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 other peripherals to reduce small sleep current.

Example: When using timing wake-up, the processing program (assembler) that goes to sleep

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

6. Interrupt

6.1 Interruption Overview

The chip has 22 interrupt sources and interrupt vectors:

Interrupt source	Interrupt description	Interrupt vector	Same priority sequence
INT0	External Interruption 0	0-0x0003	1
Timer0	Timer0 interruption	1-0x000B	2
INT1	External Interruption 1	2-0x0013	3
Timer1	Timer1 Interruption	3-0x001B	4
UART0	TI0 or RI0	4-0x0023	5
Timer2	Timer2 Interruption	5-0x002B	6
UART1	TI1 or RI1	6-0x0033	7
P0EXTIF<7:0>	P0 External Interruption	7-0x003B	8
P1EXTIF<7:0>	P1 External Interruption	8-0x0043	9
P2EXTIF<7:0>	P2 External Interruption	9-0x004B	10
--	--	10-0x0053	11
--	--	11-0x005B	12
P5EXTIF<7:0>	P5 External Interruption	12-0x0063	13
--	--	13-0x006B	14
ACMP	Comparator Interruption	14-0x0073	15
Timer3	Timer3 Interruption	15-0x007B	16
Timer4	Timer4 Interruption	16-0x0083	17
--	--	17-0x008B	18
PWM	PWM Interruption	18-0x0093	19
ADC	ADC Interruption	19-0x009B	20
WDT	WDT Interruption	20-0x00A3	21
I^2C	I2C Interruption	21-0x00AB	22
SPI	SPI Interruption	22-0x00B3	23
--	--	23-0x00BB	24
--	--	24-0x00C3	25
LSE_Timer/SCM	LSE timer Interruption/SCM	25-0x00CB	26
LVD	LVD power failure Interruption	26-0x00D3	27
TOUCH	Touch Interruption	27-0x00DB	28

The LSE timer LSE_Timer interrupt and the crystal oscillator stop oscillation monitoring SCM interrupt share an interrupt vector entry, but they have independent interrupt enable bits.

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

6.2 External Interruption

6.2.1 INT0/INT1 Interruption

The chip supports 8051 native INT0, INT1 external interrupt. INT0/INT1(P0 .0 / P0.1) can choose to descend along or low level trigger interrupt, the relevant control register is TCON. INT0 and INT1 occupy two interrupt vectors.

6.2.2 GPIO Interruption

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

```
P13CFG=0x00;        //Place P13 as GPIO
P1TRIS&=0xF7;       //Place P13 as input
P13EICFG=0x02;      //Set P13 to descent edge trigger interruption
```

interruptions of GPIO total occupy 4 interrupt vectors:

P0 occupy 1 interrupt vector 0x003B;

P1 occupy 1 interrupt vector 0x0043;

P2 occupy 1 interrupt vector 0x004B;

P5 occupy 1 interrupt vector 0x0063.

Entering the interrupt service program and judging which port triggers the interrupt and then proceed accordingly when the interrupt occurs.

6.3 Interruption And Wake-Up From Sleep

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. 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/double 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 Interruption Register

6.4.1 Interruption Mask Register

6.4.1.1 Interruption Mask Register IE

Interrupt Mask Register IE is a readable and writable register that can be operated by bit. When an interrupt condition occurs, the interrupt flag will be set, regardless of the corresponding interrupt enable bit or the state of the global enable bit EA (in the IE register). User software should ensure that the corresponding interrupt flag bit is cleared before allowing 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;
1= Enable all unmasked interrupts;
0= Disable all interrupt.

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

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

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

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

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

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

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

6.4.1.2 Interruption 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;
1= Enable SPI interrupt;
0= Disable SPI interrupt;

Bit6 I2CIE: I^2C interrupt enable;
1= Enable I^2C interrupt;
0= Disable I^2C interrupt;

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

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

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

Bit2 -- Reserved, must to be 0.

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

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

6.4.1.3 Timer2 Interruption 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;
1= Enable interrupt;
0= Disable interrupt.

Bit6 T2EXIE: Timer2 external load interrupt enable;
1= Enable interrupt;
0= Disable interrupt.

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

Bit3 T2C3IE: Timer2 compare channel3 interrupt enable;
1= Enable interrupt;
0= Disable interrupt.

Bit2 T2C2IE: Timer2 compare channel2 interrupt enable;
1= Enable interrupt;
0= Disable interrupt.

Bit1 T2C1IE: Timer2 compare channel1 interrupt enable;
1= Enable interrupt;
0= Disable interrupt.

Bit0 T2C0IE: Timer2 compare channel0 interrupt enable;
1= Enable interrupt;
0= Disable interrupt.

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

6.4.1.4 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（i=0-7）;
1= Allow interrupt;
0= Prohibit interrupt .

6.4.1.5 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（i=0-7）;

1= Allow interrupt;

0= Prohibit interrupt.

6.4.1.6 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（i=0-7）;

1= Allow interrupt;

0= Prohibit interrupt.

6.4.1.7 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（i=0-5）;

1= Allow interrupt;

0= Prohibit 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 that can be operated bit 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;
1= Set to high priority interrupt;
0= Set to low priority interrupt.

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

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

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

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

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

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

6.4.2.2 Interrupt Priority Control Register EIP1

0xB9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP1	PACMP	--	PP5	--	--	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;
1= Set to high level interrupt;
0= Set to low level interrupt.
Bit6 -- Reserved, must to be 0.
Bit5 PP5: P5 port interrupt priority control;
1= Set to high level interrupt;
0= Set to low level interrupt.
Bit4~Bit3 -- Reserved, all must to be 0.
Bit2 PP2: P2 port interrupt priority control;
1= Set to high level interrupt;
0= Set to low level interrupt.
Bit1 PP1: P1 port interrupt priority control;
1= Set to high level interrupt;
0= Set to low level interrupt.
Bit0 PP0: P0 port interrupt priority control;
1= Set to high level interrupt;
0= Set to 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;
1= Set to high level interrupt;
0= Set to low level interrupt.
Bit6 PI2C: I2C interrupt priority control;
1= Set to high level interrupt;
0= Set to low level interrupt.
Bit5 PWDT: WDT interrupt priority control;
1= Set to high level interrupt;
0= Set to low level interrupt.
Bit4 PADC: ADC interrupt priority control;
1= Set to high level interrupt;

0= Set to low level interrupt.

Bit3 PPWM: PWM interrupt priority control;

1= Set to high level interrupt;

0= Set to low level interrupt.

Bit2 -- Reserved, must to be 0.

Bit1 PT4: TIMER4 interrupt priority control;

1= Set to high level interrupt;

0= Set to low level interrupt.

Bit0 PT3: TIMER3 interrupt priority control;

1= Set to high level interrupt;

0= Set to low level interrupt.

6.4.2.4 Interrupt Priority Control Register EIP3

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

Bit7~Bit5 -- Reserved, all must be 0.

Bit4 PTOUCH: TOUCH interrupt priority control;

1= Set to high priority interrupt;

0= Set to low priority interrupt.

Bit3 PLVD: LVD interrupt priority control;

1= Set to high priority interrupt;

0= Set to low priority interrupt.

Bit2 PLSE: LSE interrupt priority control;

1= Set to high priority interrupt;

0= Set to low priority interrupt.

Bit1~Bit0 -- Reserved, all must be 0.

6.4.3 Interrupt Flag Bit Register

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

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

Bit7 TF1: Timer1 counter overflow interrupt flag;
1= Timer1 counter overflow, hardware automatically cleared when entering the interrupt service program, or software cleared;
0= Timer1 counter no overflow.

Bit6 TR1: Timer1 run control;
1= Timer1 start;
0= Timer1 close.

Bit5 TF0: Timer0 counter overflow interrupt flag;
1= Timer0 counter overflow, hardware automatically cleared when entering the interrupt service program, or software cleared;
0= Timer0 counter no overflow.

Bit4 TR0: Timer0 run control;
1= Timer0 start;
0= Timer0 close.

Bit3 IE1: External interrupt 1 flag;
1= External interrupt 1 produces an interrupt, automatically clearing the hardware or software when entering the interrupt service program;
0= External interrupt 1 did not generate an interrupt.

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

Bit1 IE0: External interrupt 0 flag;
1= External interrupt 0 produces interrupt, hardware automatically clears zero when entering interrupt service program, and software can also clear zero;
0= External interrupt 0 did not generate an interrupt.

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

6.4.3.2 Timer2 Interrupt Flag Bit Register T2IF

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

Bit7 TF2: Timer2 counter overflow interrupt flag;
1= Timer2 counter overflow, which needs to be cleared by software;
0= Timer2 counter has no 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 -- Reserve, all must be 0.

Bit3 T2C3IF: Timer2 compare/capture channel3 flag;
1= Timer2 compares channel3 {CCH3: CCL3}={TH2:TL2} or capture channel3 to generate capture operation, which needs to be cleared by software.
0= --

Bit2 T2C2IF: Timer2 compare/capture channel2 flag;
1= Timer2 compares channel2 {CCH2: CCL2}={TH2:TL2} or capture channel2 to generate capture operation, which needs to be cleared by software.
0= --

Bit1 T2C1IF: Timer2 compare/capture channel1 flag;
1= Timer2 compares channel1 {CCH1: CCL1}={TH2:TL2} or capture channel1 to generate capture operation, which needs to be cleared by software.
0= --

Bit0 T2C0IF: Timer2 compare/capture channel0 flag;
1= Timer2 compares channel0 {RLDH: RLDL}={TH2:TL2} or capture channel0 to generate capture operation, which needs to be cleared by software.
0= --

6.4.3.3 External Interrupt Flag Bit Register EIF2

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

Bit7 SPIIF: SPI total interrupt indicator bit, read only;

1= SPI produces an interrupt (this bit is automatically cleared after clearing the specific interrupt flag;

0= SPI did not cause an interruption.

Bit6 I2CIF: I^2C total interrupt indicator position, read only;

1= I^2C produces an interrupt (after clearing the specific interrupt flag bit, this bit is automatically cleared);

0= There was no interruption to the I^2C.

Bit5 --: Reserved, must to be 0.

Bit4 ADCIF: ADC interrupt flag;

1= ADC conversion is completed, software clearance is required;

0= ADC conversion not completed.

Bit3 PWMIF: PWM total interrupt indicator, read only;

1= PWM produces an interrupt;

0= PWM did not interrupt.

Bit2 --: Reserved, must to be 0.

Bit1 TF4: Timer4 counter overflow interrupt flag;

1= The Timer4 counter overflows and is automatically cleared by hardware when entering the interrupt service routine, or it can be cleared by software;

0= Timer4 counter has no overflow.

Bit0 TF3: Timer3 counter overflow interrupt flag;

1= The Timer3 counter overflows and is automatically cleared by hardware when entering the interrupt service routine, or it can be cleared by software;

0= Timer3 counter has no overflow.

6.4.3.4 SPI Interrupt Flag Bit Register SPSR

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

Bit7 SPISIF: SPI complete flag, Read only;
1= SPI transfer to complete (Read SPSR first, then clear after reading/writing SPDR);
0= SPI not transferred.

Bit6 WCOL: SPI Write collision error flag Read only;
1= SPI Write collision error (Read SPSR first, then clear after reading/writing SPDR);
0= No Write collision error.

Bit5~Bit1 -- Reserved, all must be 0.

Bit0 SSCEN: SPI master mode NSS output control.
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 Bit 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: I^2C active module reset control;
1= Reset the main control module (I^2C registers of the entire main control module, including I2CMSR);
0= The interrupt flag bit is cleared to 0 in I^2C master mode.

I2CMIF: Interrupted flag bit in I^2C master mode;
1= In master mode, sending / receiving is complete, or a transmission error occurs. (Software clearance, write 0 clearance);
0= No interruption occurred.

Bit6~Bit0 In the I^2C master control mode, the control and sign positions are described in the I2CM description.

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: I^2C sends the operation flag bit from the moving mode, read only;

1= The main control device no longer needs data. TREQ is no longer set to 1 and this data transfer has been completed. (automatically cleared after reading I2 CSCR).

0= ---

Bit1 TREQ: I^2C from moving mode ready to send flag, read only;

1= As the sending device has been addressed or the main control device is ready to receive data. (automatically cleared after writing I^2CSBUF).

0= ---

Bit0 RREQ: I^2Creceives the sign from the moving mode and reads only;

1= Received. (Automatic clearing after reading I^2CSBUF);

0= Not received.

The related state bit of the I^2C slave mode is also the interrupt flag bit.

Note: I^2C master mode interrupt and slave mode interrupt 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.

Bit7~Bit2 U1SM0、U1SM1、U1SM2、U1REN、U1TB8、U1RB8: UART1related control bits, see UARTn function description for details

Bit1 TIn: Send interrupt flag (requires software to clear);

1= It means that the sending buffer is empty, and the frame data can be sent.

0= --

Bit0 RIn: Receive interrupt flag (requires software to clear);

1= It means that the receiving 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 queried to determine whether it is a receiving interrupt or a sending interrupt.

6.4.3.8 P0 Interrupt Flag Bit 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 (i=0-7);

1= P0i has an interruption and needs software clearance;

0= There was no interruption to the P0i port.

6.4.3.9 P1 Interrupt Flag Bit 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(i=0-7);

1= P1ihas an interruption and needs software clearance;

0= There was no interruption to the P1i port.

6.4.3.10 P2 Interrupt Flag Bit 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 (i=0-7);

1= P2i When the port is interrupted, the software needs to be cleared;

0= P2i There was no interruption to the P2i port.

6.4.3.11 P5 Interrupt Flag Bit 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 (i=0-5);

1= P5i When the port is interrupted, the software needs to be cleared ;

0= P5i There was no interruption in the port.

6.4.4 Clear Operation Of Interrupt Flag

The clearing operation of the interrupt flag bit is divided into the following types:

- Hardware is automatically cleared (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 program, and the flag bit is automatically cleared. If the interrupt enable is turned off, these flag bits 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 program, 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 bit cleared by read and write operations

There are flag bits 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 transmission completion flag bit SPISIF in the SPI interrupt flag register needs to be read after setting to 1, and then cleared 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 upward comparison interrupt, the value of PWMUIF is 0x01. After entering the interrupt service routine, perform a read-modify-write operation to clear this bit.

PWMUIF &= 0xFE;

This operation is specifically implemented as first reading the value of PWMUIF back to the CPU, then performing calculations, and finally sending it 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 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 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 Bits In Debug Mode

The flag bit in the system does not write 0 to the flag bit zero, but needs to read/write other registers to clear the flag bit.

In the test state, the Breakpoint is executed, and after the single step operation or stop operation, the emulator will read the value of all registers from the system to the simulation software. The read/write operation of the emulator is exactly the same as read/write in normal mode.

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

Example: The transmission complete 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 program
{
        O1   _nop_();          //Set Breakpoint 1
             _nop_();
        O2   k = SPSR;         //Set Breakpoint 2
         _nop_();
        ...
}
```

When the Breakpoint is running, after the SPI stops at Breakpoint 1, the SPI completes the send operation and has generated a send completion interrupt, so SPSR .7 = 1, at which point the emulator has completed reading all the hosting operations(including reading SPSR).

Run the Breakpoint again and stop at Breakpoint 2. At this point, the emulator completes reading all registers(including SPDR) again, so SPSR .7 = 0 at this time. The above situation can also occur twice in a single step, and you need to pay attention in debugging mode.

7. I/O Port

7.1 GPIO Function

The chip has four I/O ports: PORT0、PORT1、PORT2、PORT5.

PORTx is a bidirectional port. Its corresponding data direction register is PxTRIS. Setting a position (= 1) of PxTRIS will configure the corresponding pin as an output. Clearing a bit (= 0) of PxTRIS configures the corresponding PORTx pin as an input.

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

When the PORTx is used as an output port, reading the Px register is related to the setting of the PxDS register. Setting a position (= 1) of PxDS, the corresponding bit of Px read is the state of the pin, clearing a bit (= 0) of PxDS, the corresponding bit of Px read is the state of the port data latch; when the PORTx is used as an input port, reading the Px register reads the state of the pin, and having no connection with the setting of the PxDS register

When the PORTx pin is used 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 are always read as 0.

PORTx related registers include Px, PxTRIS, PxOD, PxUP, PxRD, PxDS, etc.

7.1.1 PORTx Data Register Px

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

Register P0 Address: 0x80; Register P1 Address: 0x90; Register P2 Address: 0xA0; Register P5 Address: 0xD8.

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

1= Port pin level >VIH;（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 P5TRIS Address: 0xA5.

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

1= The pin is configured as an output;

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

Note:
After the port is set as the output port, the data of the port is read as the value of the output register.
After the port is set as the input port, the <read-modify-write> type of instruction to the port is actually the operation of 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 P5OD Address: F059H.

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

1= Pin is configured for open drain state(output is open drain output);

0= Pin is configured for normal state (output is 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 P5UP Address: F05AH.

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

1= Pin pull-up resistor is turned on;

0= Pin pull-up resistor is turned 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: F05BH.

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

1= Pin pull-down resistor is turned on;

0= Pin pull-down resistor is turned off.

Note: The pull-down resistor control is independent of the GPIO's configuration and multiplexing functions, and is controlled separately by the PxRD register.

7.1.6 PORTx Slope Control Register PxSR

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

Register P0SR Address: F00DH; Register P1SR Address: F01DH;

Register P2SR Address: F02DH; Register P5SR Address: F05DH;

Bit7~Bit0 PxSR<7:0>: Px Slope Control Register (valid when the port is configured for output status).

1= Px pin is slow slope;

0= Px pin is fast slope.

7.1.7 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 P5DS Address: F05EH;

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

1= Read the pin status in both output/input mode;
(The Schmitt circuit remains open when the port is set to output mode);

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

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

7.2 Reuse Function

7.2.1 Port Reuse Function Table

The 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). The communication input function is also selected by the communication input function allocation register (PS_XX) designation.

Digital function configuration is shown in the table below:

	External Input	Digital Function Configuration							
		0	1	2	3	4	5	6	7
P00	-	GPIO	ANA	SCL	SCLK	CC0	PG0	TXD1	RXD1
P01	-	GPIO	ANA	SDA	MOSI	CC1	PG1	TXD1	RXD1
P02	T0G/T1G/T2G	GPIO	ANA	-	MISO	CC2	PG2	TXD1	RXD1
P03	T0/T1/T2	GPIO	ANA	-	NSS(SSO0)	CC3	PG3	TXD1	RXD1
P04	-	GPIO	ANA	TXD0	NSS(SSO1)	-	PG4	TXD1	RXD1
P05	-	GPIO	ANA	RXD0	NSS(SSO2)	-	PG5	TXD1	RXD1
P06	ADET/INT0	GPIO	ANA	-	NSS(SSO3)	BUZZ	FB0	TXD1	RXD1
P07	INT0/INT1	GPIO	ANA	-	-	CL0	FB1	TXD1	RXD1
P10	-	GPIO	ANA	SCL	SCLK	-	PG0	TXD1	RXD1
P11	-	GPIO	ANA	SDA	MOSI	-	PG1	TXD1	RXD1
P12	T0G/T1G/T2G	GPIO	ANA	-	MISO	-	PG2	TXD1	RXD1
P13	T0/T1/T2	GPIO	ANA	-	NSS(SSO0)	-	PG3	TXD1	RXD1
P14	CAP0	GPIO	ANA	TXD0	NSS(SSO1)	C0_O	PG4	TXD1	RXD1
P15	CAP1	GPIO	ANA	RXD0	NSS(SSO2)	C1_O	PG5	TXD1	RXD1
P16	CAP2	GPIO	ANA	-	NSS(SSO3)	BUZZ	FB0	TXD1	RXD1
P17	CAP3	GPIO	ANA	-	-	CL0	FB1	TXD1	RXD1
P20	-	GPIO	ANA	SCL	SCLK	CC0	PG0	TXD1	RXD1
P21	-	GPIO	ANA	SDA	MOSI	CC1	PG1	TXD1	RXD1
P22	T0G/T1G/T2G	GPIO	ANA	-	MISO	CC2	PG2	TXD1	RXD1
P23	T0/T1/T2	GPIO	ANA	-	NSS(SSO0)	CC3	PG3	TXD1	RXD1
P24	-	GPIO	ANA	TXD0	NSS(SSO1)	C0_O	PG4	TXD1	RXD1
P25	-	GPIO	ANA	RXD0	NSS(SSO2)	C1_O	PG5	TXD1	RXD1
P26	ADET/INT0	GPIO	ANA	-	NSS(SSO3)	C0_O	FB0	TXD1	RXD1
P27	INT0/INT1	GPIO	ANA	-	-	C1_O	FB1	TXD1	RXD1
P50	-	GPIO	ANA	SCL	SCLK	-	PG0	TXD1	RXD1
P51	-	GPIO	ANA	SDA	MOSI	-	PG1	TXD1	RXD1
P52	-	GPIO	ANA	-	MISO	-	PG2	TXD1	RXD1
P53	-	GPIO	ANA	-	NSS(SSO0)	-	PG3	TXD1	RXD1
P54	-	GPIO	ANA	TXD0	NSS(SSO1)	-	PG4	TXD1	RXD1
P55	-	GPIO	ANA	RXD0	NSS(SSO2)	-	PG5	TXD1	RXD1

LED port allocation, analog module, CONFIG configuration port are as follows:

	GPIO(0)		ANA(1)						CONFIG
	LEDSEG	LEDCOM	ADC	TOUCH	LCDSEG	LCDCOM	ACMP	OP	
P00	-	COM0	AN0	TK0	-	COM0	-	-	-
P01	-	COM1	AN1	TK1	-	COM1	-	-	-
P02	-	COM2	AN2	TK2	-	COM2	-	-	-
P03	-	COM3	AN3	TK3	-	COM3	-	-	-
P04	SEG0	COM4	AN4	TK4	SEG0	COM4	-	-	-
P05	SEG1	COM5	AN5	TK5	SEG1	COM5	-	-	-
P06	SEG2	COM6	AN6	TK6	SEG2	COM6	-	-	-
P07	SEG3	COM7	AN7	TK7	SEG3	COM7	-	-	-
P10	SEG4	-	AN8	TK8	SEG4	-	-	OP0_O	-
P11	SEG5	-	AN9	TK9	SEG5	-	-	OP0_N	-
P12	SEG6	-	AN10	TK10	SEG6	-	-	OP0_P	-
P13	SEG7	-	AN11	TK11	SEG7	-	-	OP1_O	-
P14	SEG8	-	AN12	TK12	SEG8	-	-	OP1_N	-
P15	SEG9	-	AN13	TK13	SEG9	-	-	OP1_P	-
P16	SEG10	-	AN14	TK14	SEG10	-	C0N	-	-
P17	SEG11	-	AN15	TK15	SEG11	-	C1N	-	-
P20	SEG12	-	AN16	TK16	SEG12	-	C0P0/C1P0	-	-
P21	SEG13	-	AN17	TK17	SEG13	-	C0P1/C1P1	-	-
P22	SEG14	-	AN18	TK18	SEG14	-	C0P2/C1P2	-	-
P23	SEG15	-	AN19	TK19	SEG15	-	C0P3/C1P3	-	-
P24	SEG16	-	AN20	TK20	SEG16	-	-	-	-
P25	SEG17	-	AN21	TK21	SEG17	-	-	-	-
P26	SEG18	-	AN22	TK22	SEG18	-	-	-	-
P27	SEG19	-	AN23	TK23	SEG19	-	-	-	-
P50	-	-	AN39	TK39	-	-	-	-	NRST
P51	-	-	AN40	TK40	-	-	-	-	OSCIN
P52	-	-	AN41	TK41	-	-	-	-	OSCOUT
P53	-	-	AN42	TK42	-	-	-	-	DSDA
P54	-	-	AN43	TK43	-	-	-	-	DSCK
P55	-	-	AN44	TK44	-	-	-	-	-

7.2.2 Port Reuse Function Configuration Register

PORTx Function Configuration Registers 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, please refer to Reuse Function Explanation;

000= GPIO function;

001= Analog function（ANA）;

Others= Reuse function.

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

Each port has a function configuration register PxnCFG, and each port can be set to the corresponding digital function through PxnCFG. For example: To set P16 as the BEEP buzzer function, configure as: P16CFG = 0x04;

The port direction register PxTRIS does not need to be configured when the port is used as a multiplexing function.

- SCL and SDA pull-up resistance registers can be configured to force open drain output.
- RXD0 and RXD1 synchronization modes force pull-up.

Other multiplexing functions are forced to close the pull-up resistance and open drain output by hardware, that is, set the pull-up resistance PxUP or open drain output PxOD by software to be invalid.

When the port is multiplexed into SCL and SDA functions, the hardware forces the port to be open-drain output, and the pull-up resistance PxUP can be set through 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 reuse 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 automatically reloads the 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_FB0	F0CDH	FB0	PWM external brake signal FB0
PS_FB1	F0CEH	FB1	PWM external brake signal FB1

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

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

0x00= Allocate to P00 port;

0x01= Allocate to P01 port;

…… ……

0x20= Allocate to P20 port;

0x21= Allocate to P21 port;

…… ……

0x50= Allocate to P50 port;

0x51= Allocate to P51 port;

…… ……

0xFF= No allocation to the port;

1) 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 P13 are configured as T1 functions at the same time, the P03 configuration is valid and the P13 configuration is invalid.

2) This input function allocation structure supports multiple input functions to be allocated to the same port. For example, T0 and T1 can be allocated to port P03 at the same time, the configuration is as follows:

```
P03CFG = 0x00;      // P03 port is configured as GPIO function
P0TRIS = 0x00;      // P03 is used as GPIO input function
PS_T0 = 0x03;       // P03 port is configured as T0 function
PS_T1= 0x03;        // P03 port is configured as T1 function
```

3) The input function configuration structure can also be used simultaneously with the port external interrupt function. If the T1G and GPIO interrupt functions can be allocated to port P02 at the same time, the configuration is as follows:

```
P00CFG = 0x00;      // P02 port is configured as GPIO function
P0TRIS = 0x00;      // P02 is used as GPIO input function
PS_T1G = 0x02;      // P02 port is configured as T1G function
P02EICFG = 0x01;    // P02 port is configured as rising edge trigger interrupt
P0EXTIE = 0x04;     // Allow P02 port external interrupt
```

7.2.4 Communication Input Function Allocation Register

When the port is used as a communication port (UART0/UART1/SPI/IIC), there are multiple input ports available, and different port inputs can be selected by setting the following registers. The communication input function port allocation registers are 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 selection 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
PS_RXD0	F69FH	RXD0	UART0 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	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

(depending on the actual port of the chip, see Reuse Function Allocation Table);

0x00= Allocate to P00 port;

0x01= Allocate to P01 port;

…… ……

0x20= Allocate to P20 port;

0x21= Allocate to P21 port;

…… ……

0x50= Allocate to P50 port;

0x51= Allocate to P51 port;

…… ……

0xFF= No allocation to the port

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

```
PS_RXD1 =0x11;      // P11 is selected as RXD1 pin
P11CFG =0x07;       // P11 is multiplexed as RXD1 function of UART1
```

7.2.5 External Port Interrupt Control Register

When using an external interrupt, you need to configure the port as a GPIO function and set the direction 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;

00= External interrupt disabled;

01= Rising edge trigger interrupt;

10= Falling edge trigger interrupt;

11= Both rising or falling edges trigger interrupts;

There are 8 external interrupt control registers of Px, including Px0EICFG ~ Px7EICFG, which control the external interrupt of Px0 ~ Px7respectively, x = 0,1,2,5.

If the P00 falling edge is configured to trigger an interrupt, the configuration is as follows:

```
P00CFG =0x00;       // P00 is configured as GPIO function
P0TRIS =0x00;       // P00 is configured as input
P00EICFG = 0x02;    // P00 is configured as falling edge trigger interrupt
EA = 1;             // Enable global interrupt
P0EXTIE = 0x01;     // Enable P00 external interrupt function
```

7.2.6 Reuse Functions Application Notes

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

   For example, the P20 port is configured as RXD1, and the GPIO interrupt trigger mode of P20 is raised edge trigger and interrupt enable. When the P20 input changes from low level to high level, the GPIO interrupt of P20 will be triggered.
3) The input structure of digital signals is not affected by system configuration.

   For example, the port input module is opened by configuring the P50 power on as an external reset port. If P50 is configured as RXD1 in the program at the same time, and the interrupt can be opened, then the interrupt service program is executed before the sampling time of reset signal is valid, and then the reset operation is generated.
4) It should be noticed that in the debugging mode, if the reuse function is configured to the DSDA port, its input function is also effective. It is recommended that in the debugging mode, the relevant reuse function should not be configured to the DSDA port.
5) When the port is used as an analog function, when the function configuration register is set to 0x01, the hardware closes the digital circuit to reduce power consumption, and the GPIO function-related register setting is invalid.
6) Port input / communication input function has priority limitation. If two or more ports are equipped with the same input function at the same time, then according to P00, P01, ..., P54, P55 from high to low priority order to configure the selection.
7) The output function of the communication port has no priority limitation. If there are multiple ports configured with the same output function, the function will be output at these ports at the same time.
8) RXD0/1 of UART0/1 is selected by port allocation register as input function and is independent of port allocation register as synchronous output function. When RXD1 is used as synchronous output function, multiple pins can be selected as RXD0 / 1 output at the same time.
9) The SCLK of SPI is selected by port allocation register when it is used as slave clock input, and it is independent of port allocation register when it is used as master clock output. It is recommended that SCLK configure port allocation register both as output and input.
10) When the SCL of IIC is used as the clock input of the slave, it needs to be selected by the port allocation register. When it is used as the clock output of the master, it is related to the port allocation register. It is recommended that the SCL be configured with the port allocation register whether it is used as the output or input.

8. Watchdog Timer（WDT）

8.1 Overview

Watchdog 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, the watchdog overflow interrupt flag (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 program and clear the watchdog counter by writing the register WDCON[0]=1. After the watchdog counter is cleared, the counter starts counting again from 0 until the next timer overflow.

When the watchdog timer overflows, if the watchdog overflow reset is enabled (WDCON[1]=1), and the watchdog counter is not cleared, the 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 system can be reset by the watchdog, so as to avoid the system entering an infinite dead cycle. See the Dog Overflow reset section for details.

8.2 Related Register

8.2.1 Watchdog Control Register WDCON

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

Bit7	SWRST:	Software reset control;
	1:	Perform system software reset (write 0 after reset).
	0:	--
Bit6	PORF:	Power-on reset;
	1:	The system is power-on reset (write 0 to clear, TA write timing is not required).
	0:	--
Bit5~Bit4	--	Reserved, all must be 0.
Bit3	WDTIF:	WDT overflow interrupt flag;
	1=	WDT overflow (write 0 to clear);
	0=	WDT has no 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;
	1=	Enable WDT to reset CPU;
	0=	Disable WDT to reset CPU.
Bit0	WDTCLR:	WDT Counter clear;
	1=	Clear WDT Counter(hardware reset automatically);
	0=	Forbid WDT Counter (write 0 is invalid).

Note:

1. If WDT in CONFIG is configured to Enable, WDT is always enabled regardless of the state of WDTRE control bits ,and the overflow reset function of WDT is forcibly turned on.
2. If WDT in CONFIG is configured to SOFTWARE CONTROL, WDT can be enabled or disabled by using WDTRE control bits.

Modifying the command sequence required by WDCON (no other commands can be inserted in the middle) :

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

8.2.2 Watchdog Overflow Control Register CKCON

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

Bit7~Bit5 WTS<2:0>: WDT overflow time select;
000= 217*Tsys;
001= 218*Tsys;
010= 219*Tsys;
011= 220*Tsys;
100= 221*Tsys;
101= 222*Tsys;
110= 224*Tsys;
111= 226*Tsys.

Bit4 T1M: Timer1 clock source selection;
0= Fsys/12;
1= Fsys/4.

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

Bit2~Bit0 -- Reserved, all must be 0.

8.3 WDT Interrupt

The watchdog timer can enable or close interrupts through the EIE2 register, and set high/low priority through the EIP2 register, the interrupt related bits are as follows.

8.3.1 Interruption Mask Register EIE2

0xAA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIE2	SPIIE	I^2CIE	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;
1= Enable SPI interrupt;
0= Disable SPI interrupt;

Bit6 I2CIE: I2C interrupt enable;
1= Enable I2C interrupt;
0= Disable I2C interrupt;

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

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

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

Bit2 -- Reserved, must to be 0.

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

Bit0 ET3: Timer3 interrupt enable;
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	PI^{2}C	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;
1= Set to high level interrupt;
0= Set to low level interrupt.

Bit6 PI2C: I2C interrupt priority control;
1= Set to high level interrupt;
0= Set to low level interrupt.

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

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

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

Bit2 -- Reserved, must to be 0.

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

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

9. Timer 0/1（Timer0/1）

Timer 0 is similar to the type and structure of Timer1 and is two 16-bit timers. Timer1 has three modes of operation, and Timer0 has four modes of operation. They provide basic timing and event count operations.

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

-In "counter mode", the timing register increases whenever a falling edge on the corresponding input pin (T0 or T1) is detected.

9.1 Overview

Timer0 and Timer1 are fully compatible with the standard 8051 timer.

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

Mode	M1	M0	Function Description
0	0	0	THx[7:0],TLx[4:0] Composition of 13-bit time/counter
1	0	1	THx[7:0],TLx[7:0] Composition of 16-bit time/counter
2	1	0	TLx[7:0] Composition of 8-bit automatic reload time/counter, reload from THx
3	1	1	TL0,TH0 is two 8-bit time/counter, Timer1 stops counting

Registers THx and TLx are special function registers that have the ability to store actual timer values. THx and TLx can be cascaded into 13-bit or 16-bit registers by pattern options. Each time an internal clock pulse or an external timer pin is received, the value of the register adds 1 to the state jump. The timer will count from the values loaded in the preset register until the timer is full of Overflow, which will generate an internal interrupt signal. If selected as the automatic overloading mode of the timer, the timer's value will be reset to the initial value of the preload register and continue to count, otherwise the timer's value will be reset to zero. Note that in order to get the maximum range of time/counter calculations, the preset register must be cleared first.

9.2 Related Register

9.2.1 Timer0/1 Mode Register TMOD

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

Bit7 Timer1 gate control;

GATE1:

1= Enable;

0= Disable.

Bit6 CT1: Timer1 timer/counter selection;

1= Counting;

0= Timing.

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

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: Timer0 gate control

1= Enable;

0= Disable.

Bit2 CT0: Timer 0 timer/counter selection;

1= Counting;

0= Timing.

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

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 separate 8-bit timer/counter.

9.2.2 Timer0/1 Control Register TCON

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

Bit7 TF1: Timer1 counter overflow interrupt flag;
1= Timer1 counter overflow, hardware automatically cleared when entering the interrupt service program;
0= Timer1 counter no overflow.

Bit6 TR1: Timer1 run control;
1= Timer1 start;
0= Timer1 close.

Bit5 TF0: Timer0 counter overflow interrupt flag;
1= Timer0 counter overflow, hardware automatically cleared when entering the interrupt service program;
0= Timer0 counter no overflow.

Bit4 TR0: Timer0 run control;
1= Timer0 start;
0= Timer0 close.

Bit3 IE1: External interrupt 1 flag;
1= External interrupt 1 produces an interrupt, automatically clearing the hardware or software when entering the interrupt service program;
0= External interrupt 1 did not generate an interrupt.

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

Bit1 IE0: External interrupt 0 flag;
1= External interrupt 0 produces interrupt, hardware automatically clears zero when entering interrupt service program;
0= External interrupt 0 did not generate an interrupt.

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

9.2.3 Timer0 Low Bit Data Register 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>: Timer0 low bit data register (also as the counter low byte)

9.2.4 Timer0 High Bit Data Register 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>: Timer0 high bit data register (also as the counter high byte)

9.2.5 Timer1 Low Bit Data Register 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>: Timer1 low bit data register(also as the counter low byte).

9.2.6 Timer1 High Bit Data Register 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>: Timer1 high bit data register(also as the counter high byte).

9.2.7 Function Clock Control Register CKCON

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

Bit7~Bit5 WTS<2:0>: WDT overflow time select;
000= 217*Tsys;
001= 218*Tsys;
010= 219*Tsys;
011= 220*Tsys;
100= 221*Tsys;
101= 222*Tsys;
110= 224*Tsys;
111= 226*Tsys.

Bit4 T1M: Timer1 clock source selection;
0= Fsys/12;
1= Fsys/4.

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

Bit2~Bit0 -- Reserved, all must be 0.

9.3 Timer0/1 Interrupt

Timer0/1 can enable or disable interrupt by IE register, can also set high or low priority by IP register, and the bytes of interrupt are as follows:

9.3.1 Interruption 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;
1= Enable all unmasked interrupts;
0= Disable all interrupt.

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

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

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

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

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

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

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

9.3.2 Interrupt Priority Control Register IP

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

Bit7 -- Reserved, must be 0.

Bit6 PS1: UART1 interrupt priority control;
1= Set to high priority interrupt;
0= Set to low priority interrupt.

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

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

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

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

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

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

9.3.3 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;
1= Timer1 counter overflow, hardware automatically cleared when entering the interrupt service program, or software cleared;
0= Timer1 counter no overflow.

Bit6 TR1: Timer1 run control;
1= Timer1 start;
0= Timer1 close.

Bit5 TF0: Timer0 counter overflow interrupt flag;
1= Timer0 counter overflow, hardware automatically cleared when entering the interrupt service program, or software cleared;
0= Timer0 counter no overflow.

Bit4 TR0: Timer0 run control;
1= Timer0 start;
0= Timer0 close.

Bit3 IE1: External interrupt 1 flag;
1= External interrupt 1 produces an interrupt, automatically clearing the hardware or software when entering the interrupt service program;
0= External interrupt 1 did not generate an interrupt.

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

Bit1 IE0: External interrupt 0 flag;
1= External interrupt 0 produces interrupt, hardware automatically clears zero when entering interrupt service program, and software can also clear zero;
0= External interrupt 0 did not generate an interrupt.

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

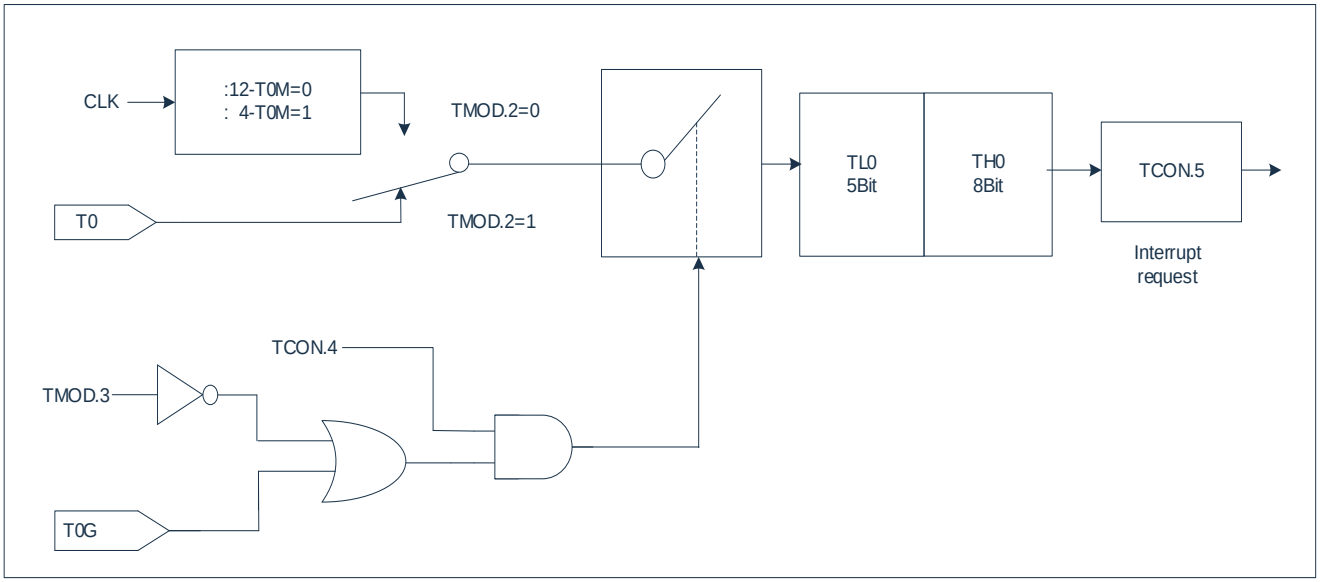
The flag that generates the interrupt can be cleared by software, the same as the result of clearing by hardware. In other words, interrupts can be generated by software (it is not recommended to generate interrupts by writing flag bits) or cancel pending interrupts.

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

9.4 Timer0 Operating Mode

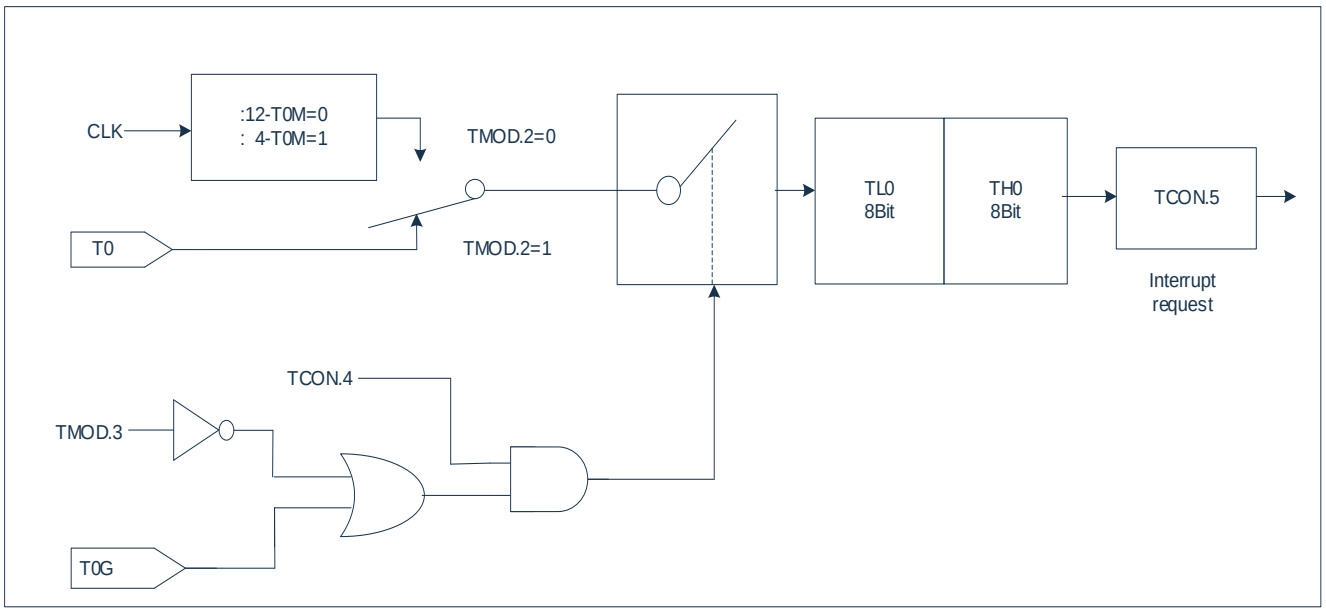
9.4.1 T0 -Mode0（13-Bit Timing/Counting Mode）

In this mode, the Timer0 is a 13-bit register. When all bits of the counter flip from 1 to 0, the Timer0 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 enables to the Timer0. (TMOD.3 = 1 is set to allow timer 0 to be controlled by external pin T0G for pulse width measurement). The 13-bit register is composed of 5-bit lower TL0 and TH0. The upper 3 bits of TL0 should be ignored. Timer0 Mode0 structure diagram is shown as follows :



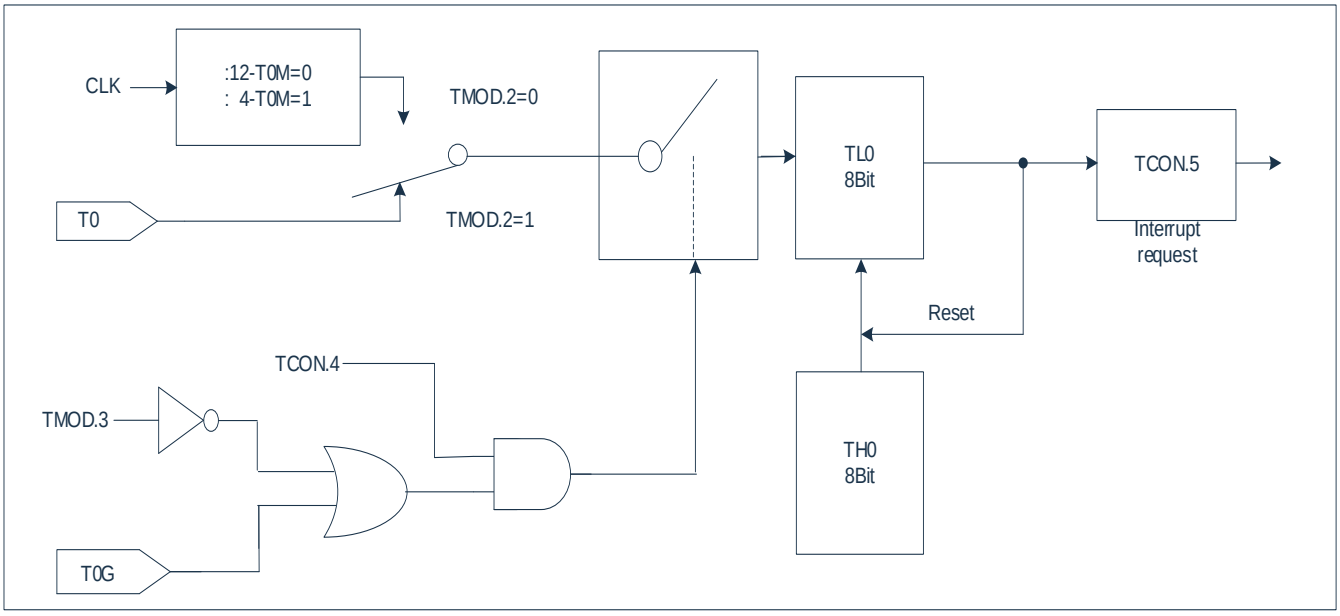
9.4.2 T0 -Mode1（16-Bit Timing/Counting Mode）

Mode1 is the same as Mode0, except that all 16 bits of the Timer0 Data Register are running in Mode1. The structure diagram of Timer0 Mode1 is shown in the figure below:



9.4.3 T0 -Mode2（8-Bit Auto Reload Timing/Counting Mode）

The timer register in Mode 2 is an 8-bit counter (TL0) equipped with automatic reload mode, as shown in the following figure. The overflow from TL0 not only makes TF0 set to 1, but also reloads the content of TH0 from software to TL0. The TH0 value remains unchanged during reloading. The structure diagram of Timer0 Mode 2 is as follows :



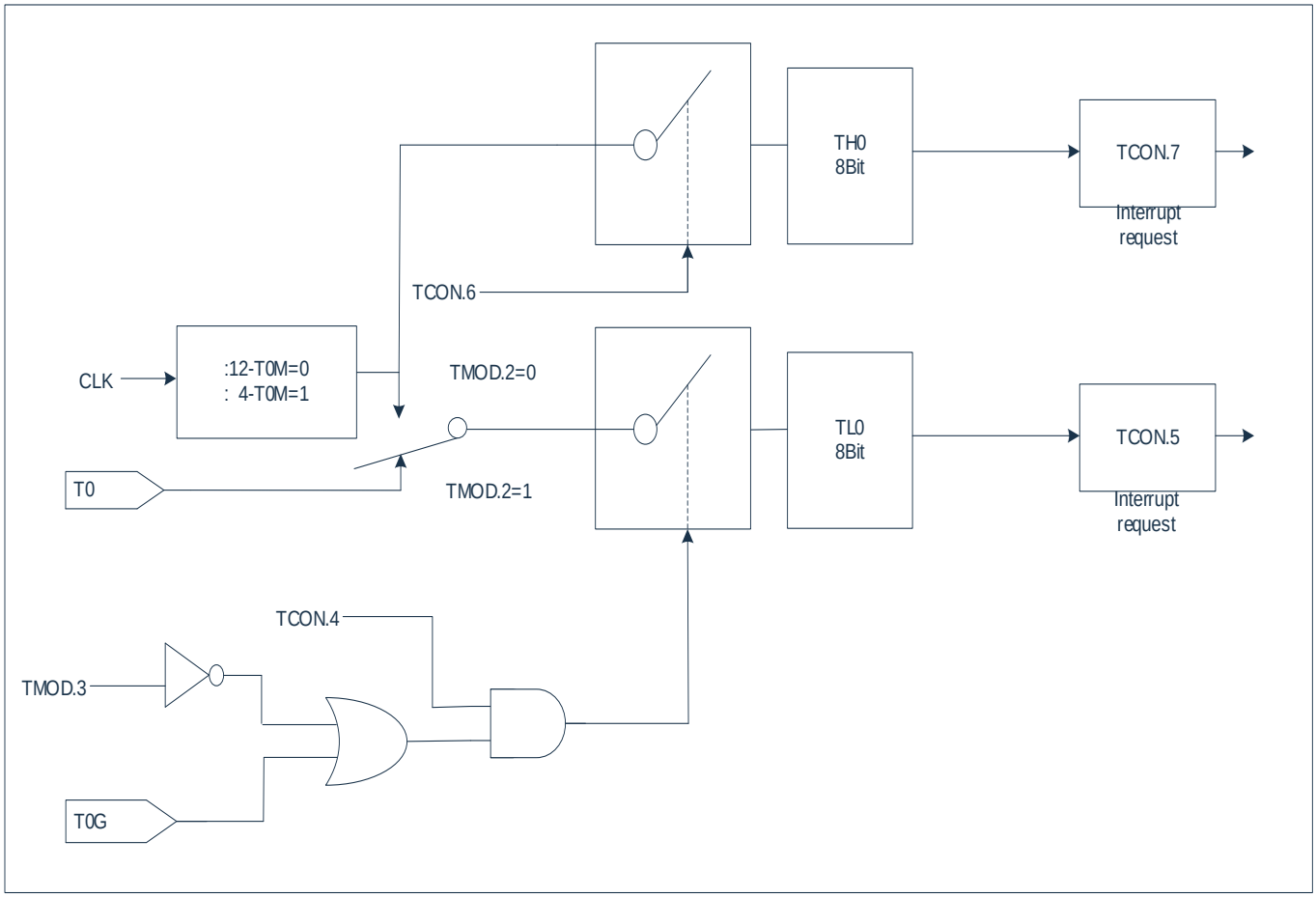
9.4.4 T0 -Mode3（Two Separate 8-Bit Timer/Counters）

Timer0 in Mode3 sets TL0 and TH0 to two independent counters. The logic of Timer0 Mode3 is shown below.

TL0 can operate as a timer or counter and uses the control bits of Timer0: CT0, TR0, GATE0, and TF0.

TH0 can only work as a timer and uses the TR1 and TF1 flags of Timer1 to control the Timer1 interrupt.

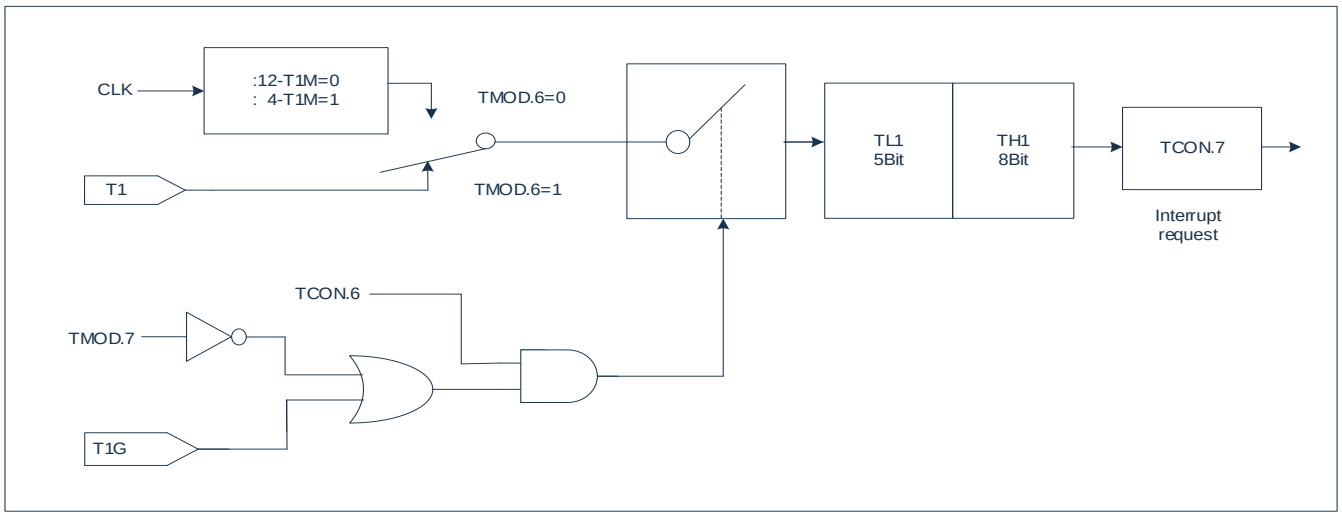
Mode3 is available when two 8-bit timer/counters are required. When Timer0 is in Mode3, Timer1 can be turned off by switching to its own Mode3, or it can still be used as a Baud Rate Generator by the serial channel, or if no Timer1 interrupt is required in the application. The structure block diagram of Timer0 Mode 3 is shown below.



9.5 Timer1 Operating Mode

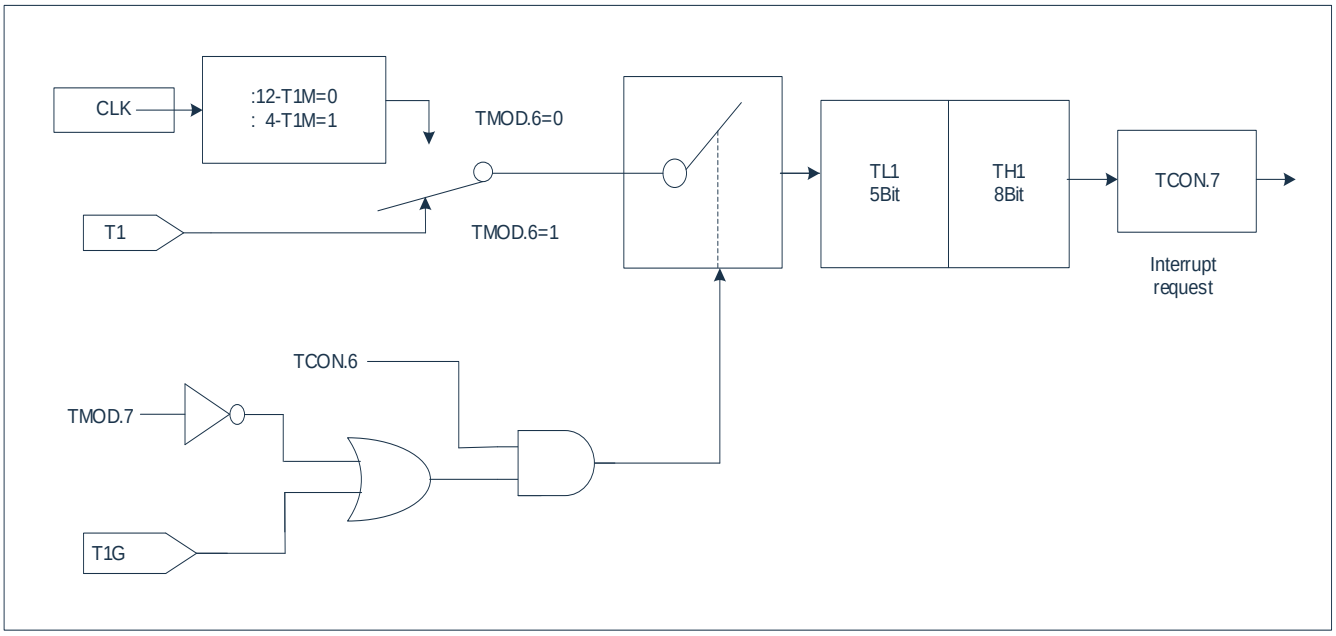
9.5.1 T1 -Mode0（13-Bit Timing/Counting Mode）

In this mode, Timer1 is a 13-bit register. When all bits of the counter are turned from 1 to 0, the Timer1 interrupt flag TF1 is set 1. When TCON.6 = 1 and TMOD.7 = 0 or when TCON.6 = 1, TMOD.7 = 1 and T1G = 1, count input enablesTimer1. (TMOD. 7 = 1 is set to allow Timer1 to be controlled by external pin T1G for pulse width measurement). The 13-bit register is composed of TH18 bits and TL1 low 5 bits. The upper 3 bits of TL1 should be ignored. The structure diagram of Timer1 Mode0 is as follows :



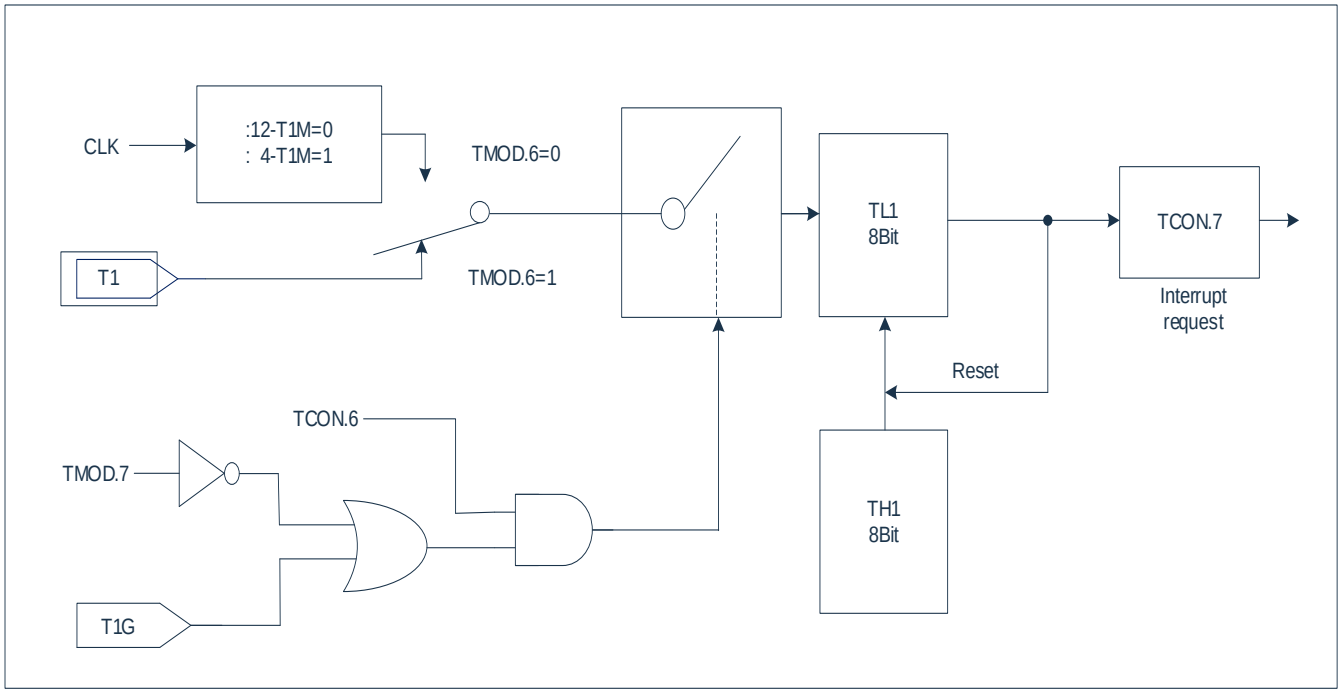
9.5.2 T1 -Mode1（16-Bit Timing/Counting Mode）

Mode1 is the same as Mode0, except that all 16 bits of the Timer1 data register are running in Mode1. The structure diagram of Timer1 Mode1 is shown in the figure below:



9.5.3 T1 -Mode2（8-Bit Auto Reload Timing/Counting Mode）

The timer register in Mode2 is an 8-bit counter (TL1) equipped with automatic reload mode, as shown in the following figure. The overflow from TL1 not only makes TF1 set to 1, but also reloads the content of TH1 from software to TL1. The TH1 value remains unchanged during reloading. The structure diagram of Timer1 Mode 2 is as follows :



9.5.4 T1 -Mode3（Stop Counting）

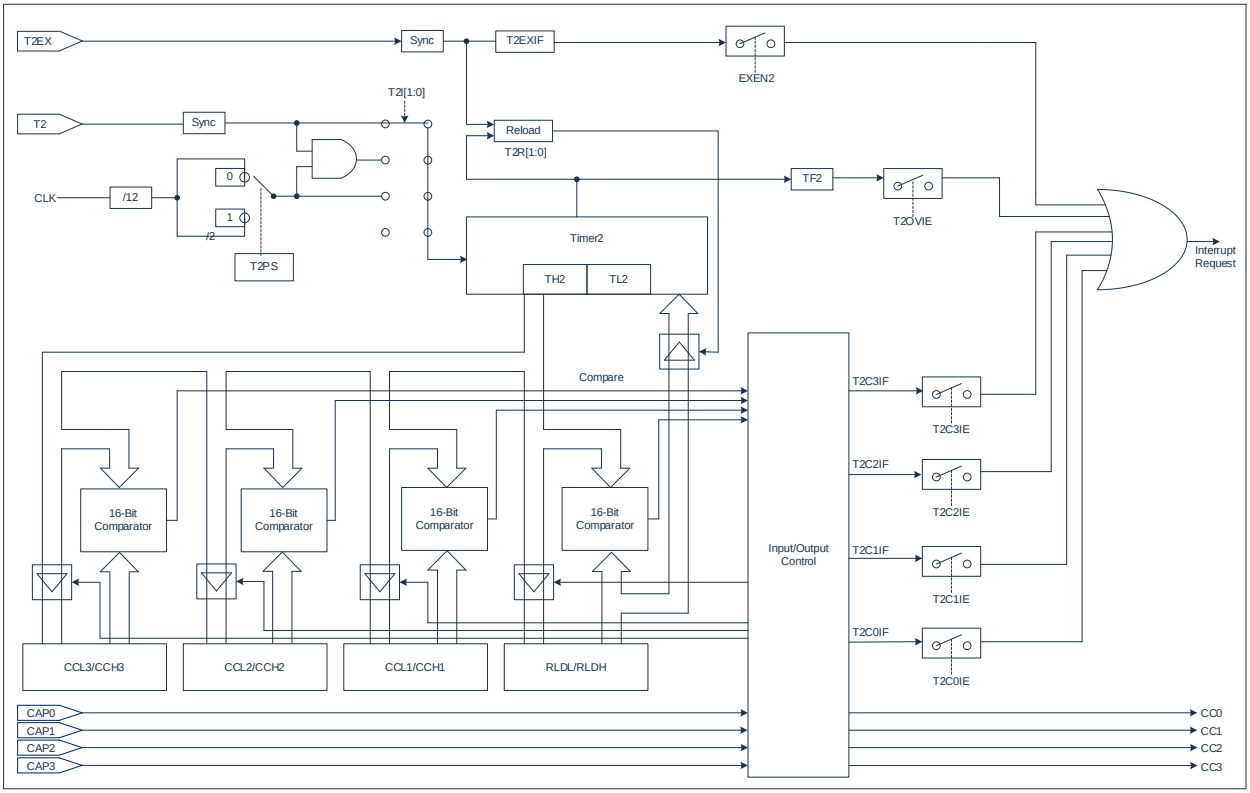
The Timer1 in Mode3 stops counting, with the same effect as setting TR1 = 0.

10. Timer 2（Timer2）

Timer 2 with additional comparison / capture / reload function is one of the most core peripheral units. It can be used to generate various digital signals and capture events, such as pulse generation, pulse width modulation, pulse width measurement, etc.

10.1 Overview

The structure diagram of timer 2 with additional comparison / capture / reload register function is shown as follows :



10.2 Related Register

10.2.1 Timer2 Control Register T2CON

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

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

Bit6 I3FR: Capture channel0 input single edge selection and compare interrupt time selection;
Capture channel0 mode:
1= Rising edge capture to RLDL/RLDH register;
0= Falling edge capture to RLDL/RLDH register.
Compare channel0 mode:
1= The interrupt is generated when TL2/TH2 and RLDL/RLDH are unequal to equal moment;
0= The interrupt is generated when TL2/TH2 and RLDL/RLDH are equal to unequal moment.

Bit5 CAPES: Capture channel1-3 input single edge selection (for capture channel1-3 together).
0= Rising edge capture to CCL1/CCH1-CCL3/CCH3 register;
1= Falling edge capture to CCL1/CCH1-CCL3/CCH3 register;

Bit4~Bit3 T2R<1:0>: Timer2 load mode selection;
0x= Prohibit reloading;
10= Load mode1:Timer2 overflow reload;
11= Load mode2:T2EX falling edge reload.

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

Bit1~Bit0 T2I<1:0>: Timer2 clock input selection;
00= Timer2 stop;
01= Frequency division of system clock (selected by T2PS control frequency division);
10= External pin T2 for event input (event counting mode) ;
11= External pin T2 for gating input (gating timing mode).

10.2.2 Timer2 Low Bit Data Register TL2

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

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

10.2.3 Timer2 High Bit Data Register 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>: Timer2 high bit data register (also as the counter high byte).

10.2.4 Timer2 Compare/Capture/Reload Register Low 8-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>: Timer2 compare/capture/reload register low 8-bit.

10.2.5 Timer2 Compare/Capture/Reload Register High 8-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>: Timer2 compare/capture/reload register high 8-bit

10.2.6 Timer2 Compare/Capture Channel1 Register Low 8-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>: Timer2 compare/capture channel1 register low 8-bit.

10.2.7 Timer2 Compare/Capture Channel1 Register High 8-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>: Timer2 compare/capture channel1 register high 8-bit.

10.2.8 Timer2 Compare/Capture Channel2 Register Low 8-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>: Timer2 compare/capture channel2 register low 8-bit.

10.2.9 Timer2 Compare/Capture Channel2 Register High 8-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>: Timer2 compare/capture channel2 register high 8-bit

10.2.10 Timer2 Compare/Capture Channel3 Register Low 8-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>: Timer2 compare/capture channel3 register low 8-bit.

10.2.11 Timer2 Compare/Capture Channel3 Register High 8-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>: Timer2 compare/capture channel3 register high 8-bit.

10.2.12 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;
00= Compare/capture disable;
01= Capture operation is triggered on the rising edge or falling edge of channel3 (CAPES selection);
10= Compare mode enable;
11= Capture operation is triggered on writing CCL3 or the double-edge of channel3.

Bit5~Bit4 CMH2-CML2: Capture/compare mode control;
00= Compare/capture disable;
01= Capture operation is triggered on the rising edge or falling edge of channel2 (CAPES selection);
10= Compare mode enable;
11= Capture operation is triggered on writing CCL2 or the double-edge of channel2.

Bit3~Bit2 CMH1-CML1: Capture/compare mode control;
00= Compare/capture disable;
01= Capture operation is triggered on the rising edge or falling edge of channel1 (CAPES selection);
10= Compare mode enable;
11= Capture operation is triggered on writing CCL1 or the double-edge of channel1.

Bit1~Bit0 CMH0-CML0: Capture/compare mode control;
00= Compare/capture disable;
01= Capture operation is triggered on the rising edge or falling edge of channel1 (I3FR selection);
10= Compare mode enable;
11= Capture operation is triggered on writing RLDL or the double-edge of channel1.

10.3 Timer2 Interrupt

Timer2 can enable or close the total interrupt through register IE, and can also set high / low priority through IP register. Timer2 has four types of interruption :

- Timed overflow interrupt.
- External pin T2EX falling edge interrupt.
- 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 four types of interrupts of Timer2 share an interrupt vector. After entering the interrupt service routine, need to determine the relevant flag bit to determine which type has generated the interrupt.

10.3.1 Interrupt Related Register

10.3.1.1 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;
1= Enable all unmasked interrupts;
0= Disable all interrupt.

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

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

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

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

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

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

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

10.3.1.2 Timer2 Interrupt Mask Register T2IE

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

Bit7 T2OVIE: Timer2 overflow interrupt enable;
1= Enable interrupt;
0= Disable interrupt.
Bit6 T2EXIE: Timer2 external load interrupt enable;
1= Enable interrupt;
0= Disable interrupt.
Bit5~Bit4 -- Reserved, all must be 0.
Bit3 T2C3IE: Timer2 compare channel3 interrupt enable;
1= Enable interrupt;
0= Disable interrupt.
Bit2 T2C2IE: Timer2 compare channel2 interrupt enable;
1= Enable interrupt;
0= Disable interrupt.
Bit1 T2C1IE: Timer2 compare channel1 interrupt enable;
1= Enable interrupt;
0= Disable interrupt.
Bit0 T2C0IE: Timer2 compare channel0 interrupt enable;
1= Enable interrupt;
0= Disable interrupt.

To turn on the interruption of Timer2, the total interruption of Timer2 needs to be turned on to enable ET2 = 1 (IE.5 = 1).

10.3.1.3 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;

1= Set to high priority interrupt;

0= Set to low priority interrupt.

Bit5 PT2: TIMER2 interrupt priority control;

1= Set to high priority interrupt;

0= Set to low priority interrupt.

Bit4 PS0: UART0 interrupt priority control;

1= Set to high priority interrupt;

0= Set to low priority interrupt.

Bit3 PT1: TIMER1 interrupt priority control;

1= Set to high priority interrupt;

0= Set to low priority interrupt.

Bit2 PX1: External interrupt 1 priority control;

1= Set to high priority interrupt;

0= Set to low priority interrupt.

Bit1 PT0: TIMER0 interrupt priority control;

1= Set to high priority interrupt;

0= Set to low priority interrupt.

Bit0 PX0: External interrupt 0 priority control;

1= Set to high priority interrupt;

0= Set to low priority interrupt.

10.3.1.4 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;
1= Timer2 counter overflow, which needs to be cleared by software;
0= Timer2 counter has no 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 -- Reserve, all must be 0.

Bit3 T2C3IF: Timer2 compare/capture channel3 flag;
1= Timer2 compares channel3 {CCH3: CCL3}={TH2:TL2} or capture channel3 to generate capture operation, which needs to be cleared by software.
0= --

Bit2 T2C2IF: Timer2 compare/capture channel2 flag;
1= Timer2 compares channel2 {CCH2: CCL2}={TH2:TL2} or capture channel2 to generate capture operation, which needs to be cleared by software.
0= --

Bit1 T2C1IF: Timer2 compare/capture channel1 flag;
1= Timer2 compares channel1 {CCH1: CCL1}={TH2:TL2} or capture channel1 to generate capture operation, which needs to be cleared by software.
0= --

Bit0 T2C0IF: Timer2 compare/capture channel0 flag;
1= Timer2 compares channel0 {RLDH: RLDL}={TH2:TL2} or capture channel0 to generate capture operation, which needs to be cleared by software.
0= --

10.3.2 Timer Interrupt

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

10.3.3 External Trigger Interrupt

The falling edge of the external pin T2EX triggers the interrupt enable bit by register T2IE[6], and the interrupt flag bit is viewed by register T2IF[6]. When the T2EX pin falls along, the external load interrupt flag bit T2EXIF will be set.

10.3.4 Compare Interrupt

The 4 compare channels all 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 channel0 can select the moment 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;the interrupt is generated when TL2/TH2 and RLDL/RLDH are unequal to equal.

When I3FR = 1;the interrupt is generated when TL2/TH2 and RLDL/RLDH 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.

10.3.5 Capture Interrupt

The 4 capture channels all support external capture interrupts. 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.

10.4 Timer2 Function Description

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

- Timing mode.
- Reload mode.
- Gate control timing mode.
- Event counting mode.
- Compare mode.
- Capture mode.

Setting different modes of timer 2 can be used to generate various digital signals and capture events, such as pulse generation, pulse width modulation, pulse width measurement, etc.

10.4.1 Timing Mode

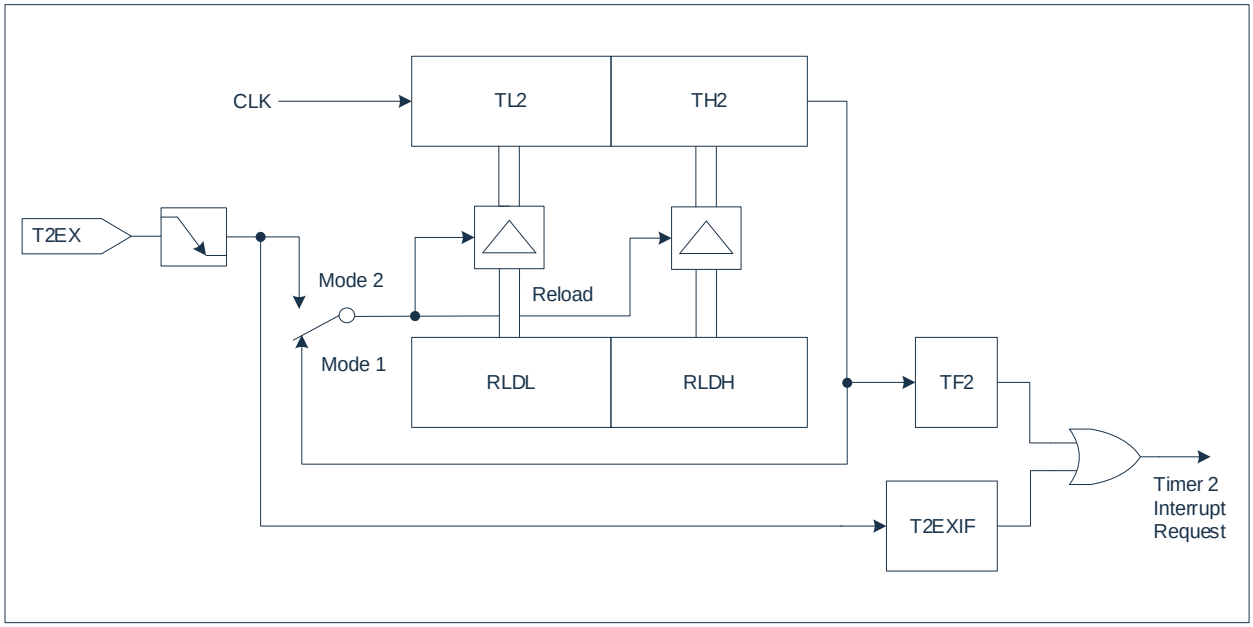
When used as a timer function, the clock source comes from the system clock. The prescaler provides 1/12 or 1/24 system frequency selection. The value of the prescaler is selected by the T2PS bit of the register T2CON. Therefore, the 16-bit timer register (consisting of TH2 and TL2) increments every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

10.4.2 Reload Mode

The reload mode is selected by the T2R0 bit and T2R1 bit of the register T2CON,the reload structure diagram is shown in the figure below.

In loading mode 1 : When the Timer2 counter flips from all 1 to 0 (the counter overflows), not only the overflow interrupt flag TF2 is set to 1, but also the Timer2 register automatically loads the 16-bit value from the RLDL / RLDH register to cover the count value 0x0000, and 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.



10.4.3 Gate Control Timing Mode

When Timer2 is used as a gate control timer function, the external input pin T2 is used as the gated input of Timer2. 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.

10.4.4 Event Counting Mode

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

10.4.5 Compare Mode

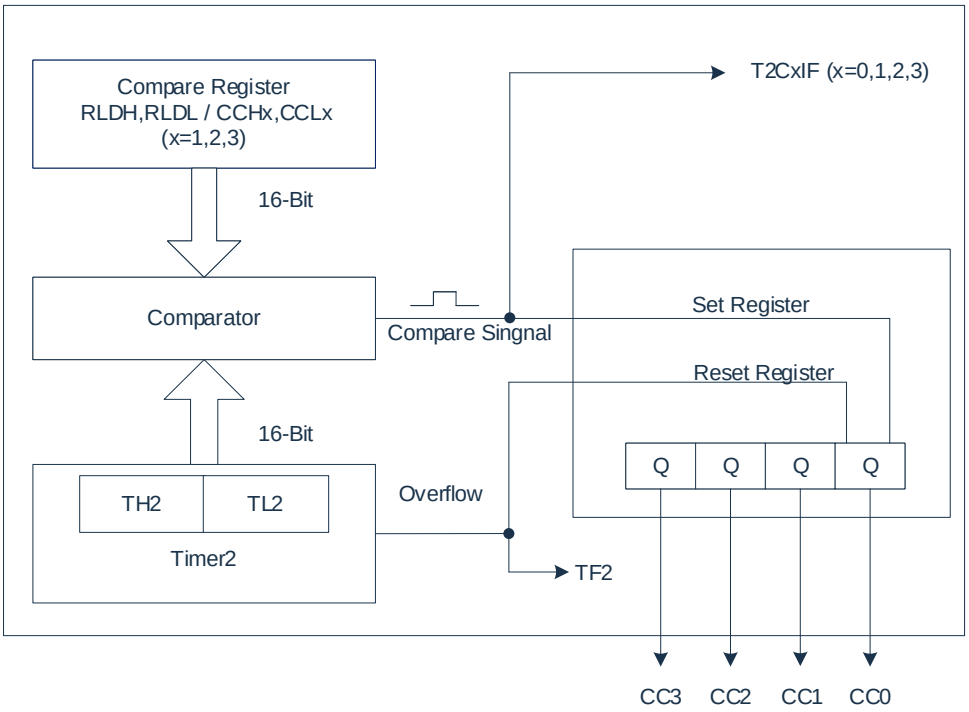
The compare function includes two modes : compare mode 0 and compare mode 1, which are selected by the T2CM bit in the special function register T2CON. These two compare modes can generate periodic signals and change the duty cycle control mode, which are often used in pulse width modulation (PWM) and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

The output channels of compare function are CC0, CC1, CC2 and CC3. The output signals corresponding to 16 bit compare registers { RLDH, RLDL }, { CCH1, CCL1 }, { CCH2, CCL2 }, { CCH3, CCL3 } and data registers { TH2, TL2 }.

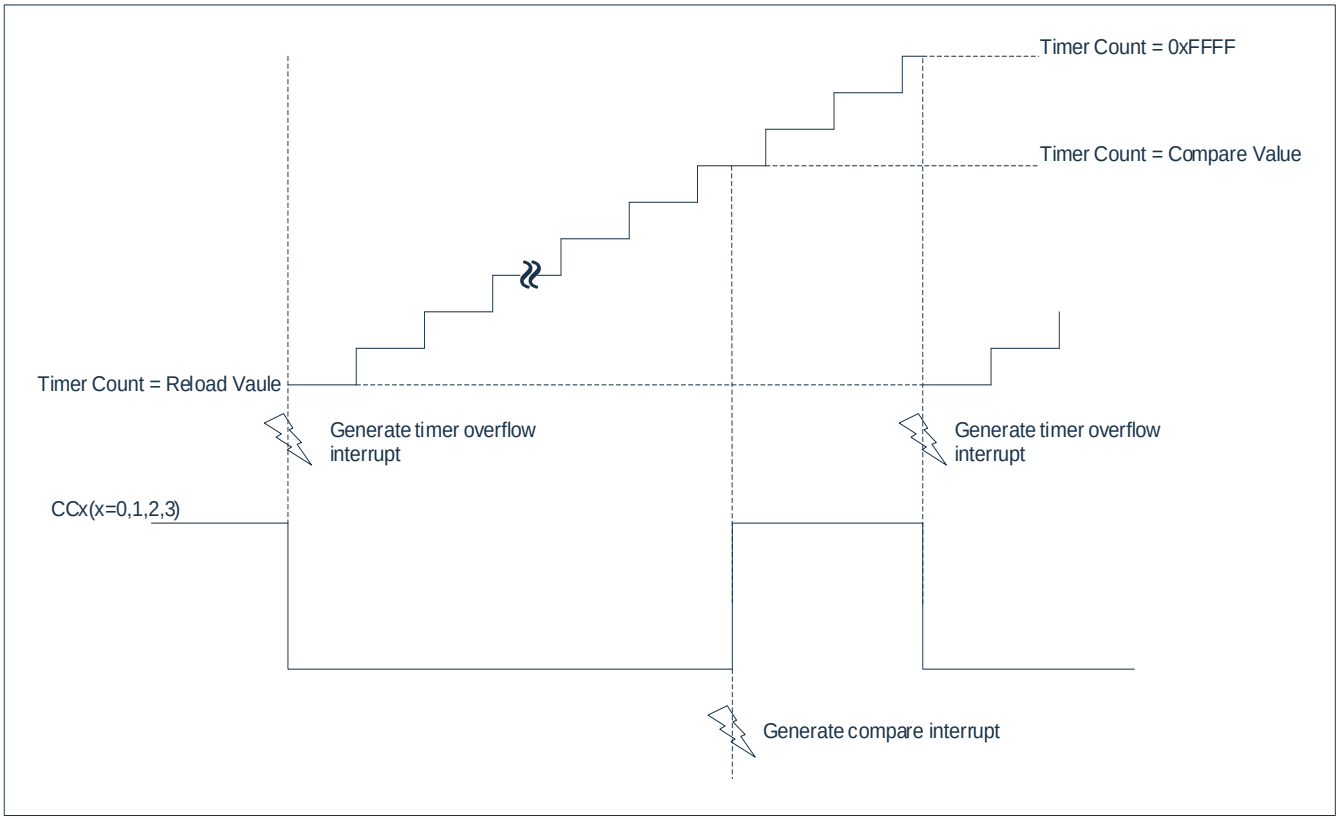
The 16-bit storage value stored in the compare register is compared with the counter value of the timer. If the counter value in the data register matches the storage value, the jump of the output signal will be generated on the corresponding port pin and the interrupt flag will be generated.

10.4.5.1 Compare Mode0

In mode 0, when the timer count is equal to the compare register.an output signal changes from low to high. It goes back to a low level on timer overflow. Figure below shows a functional diagram of a port register in compare mode 0. The compare output channel is directly controlled by two events: timer overflow and compare operation. The structure diagram of compare mode 0 is shown as follows :



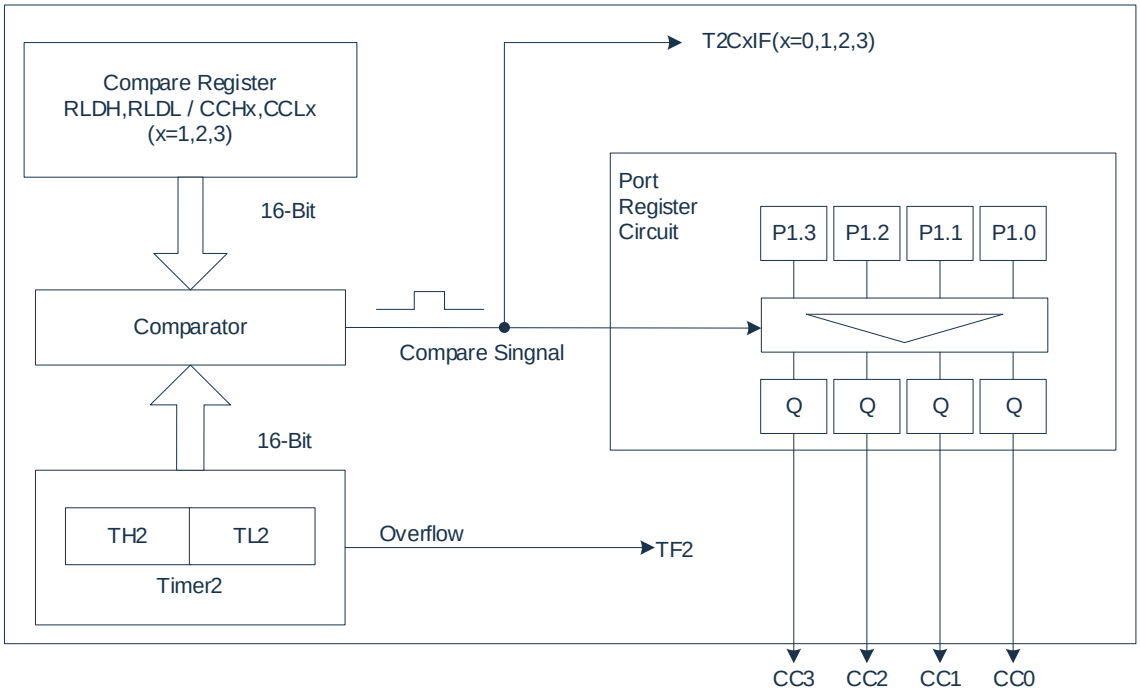
The output structure diagram of compare mode 0 is shown as follows :



10.4.5.2 Compare Mode1

In comparison mode 1, it is usually used when the output signal is independent of the constant signal period and the software adaptively determines the output signal jump.

If mode 1 is enabled, the software will write to the corresponding output register of the CCx 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 as follows:



10.4.6 Capture Mode

Each of the four 16-bit compare/capture registers {RLDH, RLDL}, {CCH1, CCL1}, {CCH2, CCL2}, {CCH3, CCL3} can be used to latch the current 16-bit value of the Timer 2 registers {TH2, TL2} . Two different modes are provided for this function.

In mode 0, an external event latches Timer 2 contents to a dedicated capture register.

In mode 1, a capture will occur upon writing to the low order byte {RLDL/CCL1/CCL2/CCL3 }of the dedicated 16-bit capture register. This mode is provided to allow software reading of Timer 2 contents {TH2, TL2} at runtime.

Capture channels 0~3 select the acquisition input pins CAP0~CAP3 as the input source signals.

10.4.6.1 Capture Mode0

In capture mode0, positive, negative or positive and negative jumps on capture channel0~3 (CAP0~CAP3) will produce capture events. When the capture event occurs, the count lock of the timer exists in the corresponding capture register.

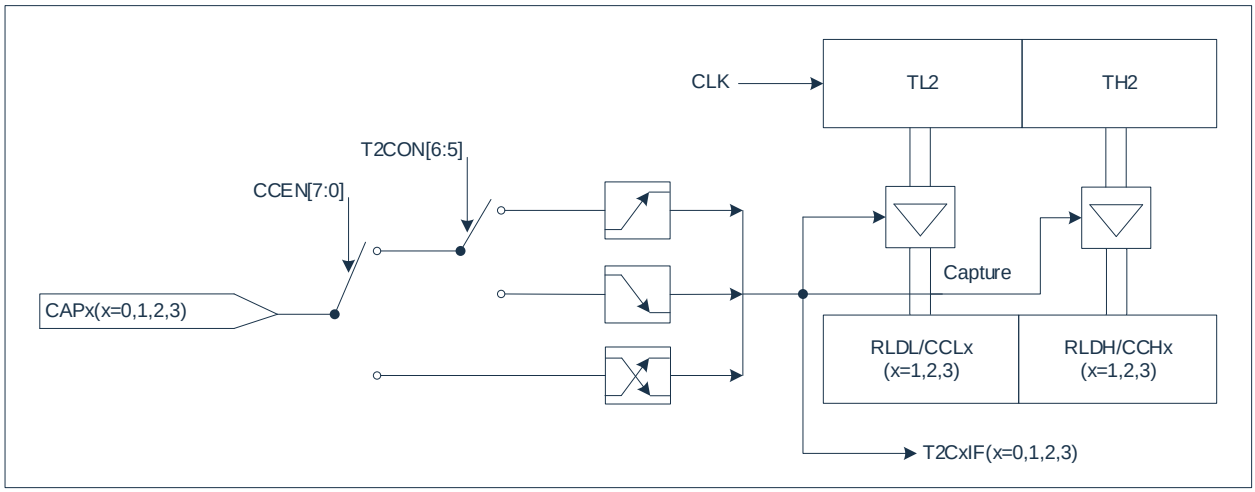
The positive jump trigger capture or negative jump trigger capture on capture channel 0 depends on the I3FR bit of T2CON. I3FR=0, negative jump trigger capture ; I3FR=1, positive jump triggers capture.

Capture channel1~3 is the positive jump trigger capture or negative jump trigger capture depends on the T2CON CAPES bit. CAPES=0, positive jump trigger capture ; CAPES=1, negative jump trigger capture. Capture channels 1~3 of the selected jump mode is the same.

The capture channels 0~3 also support the capture operation of double jumps. When the corresponding mode control bit of CCEN register is selected as 11, the channel supports double jump capture operation. It should be noted that capture mode 1 is also supported under this working mode, that is, writing operation can generate capture action.

In capture mode0, the external capture events of capture channels 0~3 can be interrupted.

The structure diagram of capture mode0 is shown as follows :

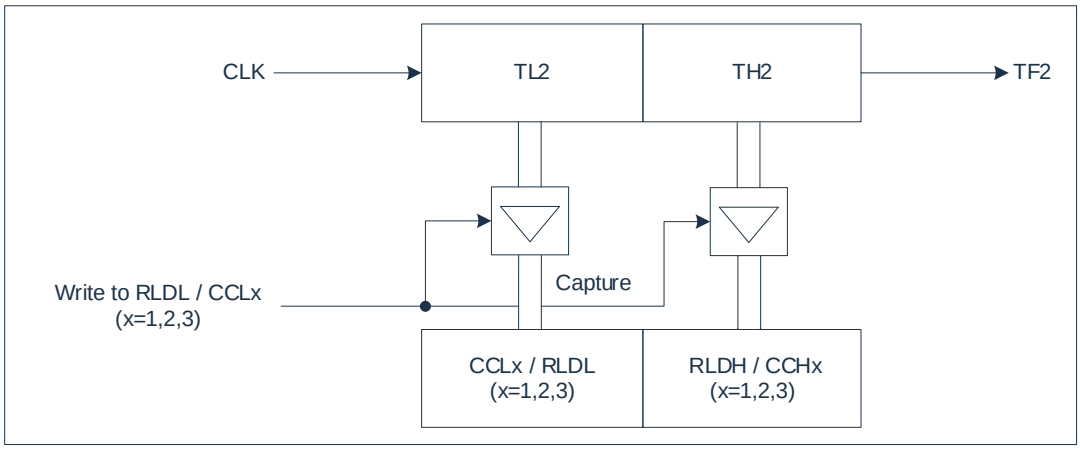


10.4.6.2 Capture Mode1

In capture mode1, the capture operation event is the execution of low byte instructions that write the capture register. The write register signal (for example, write RLDL) starts the capture operation, and the write value is not related to this function. Once the write instruction is executed, the contents of Timer2 are locked into the corresponding capture register.

Under capture mode 1, the capture event of capture channels 0~3 will not produce the interrupt request flag.

The structure diagram of capture mode1 is shown as follows :



11. Timer 3/4（Timer3/4）

The Timer3/4 is similar to the Timer0/1, which is two 16-bit timers. Timer3 has four working modes, Timer4 has three working modes. Compared with Timer0/1, Timer3/4 only provides timing operation.

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

11.1 Overview

Timer3 and Timer4 are composed of two 8-bit registers {TH3, TL3} and {TH4, TL4}, respectively. The Timer3 and Timer4 modes are as follows:

Mode	M1	M0	Function Description
0	0	0	{THx[7:0], TLx[4:0] }as a 13-bit timer
1	0	1	{THx[7:0], TLx[7:0] } as a 16-bit timer
2	1	0	TLx[7:0] as an 8-bit Auto-Reload timer , reloading from THx
3	1	1	{TL3}, {TH3 } as two separate 8-bit timers, Timer4 stops timing.

11.2 Related Register

11.2.1 Timer3/4 Control Register T34MOD

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

Bit7 TR4: Timer4 run control;
1= Timer4 start;
0= Timer4 closes.

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

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

Bit3 TR3: Timer3 run control;
1= Timer3 start;
0= Timer3 closes.

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

Bit1~Bit0 T3M<1:0>: Timer4 mode selection;
00= Mode 0, 13-bit timer;
01= Mode 1, 16-bit timer;
10= Mode 2, 8-bit automatic reload timer;
11= Mode 3, two separate 8-bit timer.

11.2.2 Timer3 Low Bit Data Register 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>: Timer3 low 8-bit data register(also as timer low byte).

11.2.3 Timer3 High Bit Data Register 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>: Timer3 high 8-bit data register(also as timer high byte).

11.2.4 Timer4 Low Bit Data Register 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>: Timer4 low 8-bit data register(also as timer low byte).

11.2.5 Timer4 High Bit Data Register 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>: Timer4 high 8-bit data register(also as timer high byte).

11.3 Timer3/4 Interrupt

The Timer3/4 can enable or disable the interrupt through the EIE2 register, and the high/low priority level can be set through the EIP2 register. The interrupt correlation bits are as follows :

11.3.1 Interrupt Mask Register EIE2

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

Bit7 SPIIE: SPI interrupt enable;
1= Enable SPI interrupt;
0= Disable SPI interrupt;

Bit6 I2CIE I^2C interrupt enable;
1= Enable I^2C interrupt;
0= Disable I^2C interrupt;

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

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

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

Bit2 -- Reserved, must to be 0.

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

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

11.3.2 Interrupt Priority Control Register EIP2

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

Bit7 PSPI: SPI interrupt priority control;
1= Set to high level interrupt;
0= Set to low level interrupt.

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

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

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

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

Bit2 -- Reserved, must to be 0.

Bit1 PT4: TIMER4 interrupt priority
1= Set to high level interrupt;
0= Set to low level interrupt.

Bit0 PT3: TIMER3 interrupt priority
1= Set to high level interrupt;
0= Set to low level interrupt.

11.3.3 External Interrupt Flag Bit Register EIF2

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

Bit7 SPIIF: SPI total interrupt indicator, read only;

1= SPI produces an interrupt (this bit is automatically cleared after clearing the specific interrupt flag;

0= SPI did not cause an interruption.

Bit6 I2CIF: I^2C total interrupt indicator position, read only;

1= I^2C produces an interrupt (after clearing the specific interrupt flag bit, this bit is automatically cleared);

0= There was no interruption to the I^2C.

Bit5 -- Reserved, must to be 0.

Bit4 ADCIF: ADC interrupt flag;

1= ADC conversion is completed, software clearance is required;

0= ADC conversion not completed.

Bit3 PWMIF: PWM total interrupt indicator, read only;

1= PWM produces an interrupt;

0= PWM did not interrupt.

Bit2 -- Reserved, must to be 0.

Bit1 TF4: Timer4 counter overflow interrupt flag;

1= The Timer4 counter verflows and is automatically cleared by hardware when entering the interrupt service routine, or it can be cleared by software;

0= Timer4 counter has no overflow.

Bit0 TF3: Timer3 counter overflow interrupt flag;

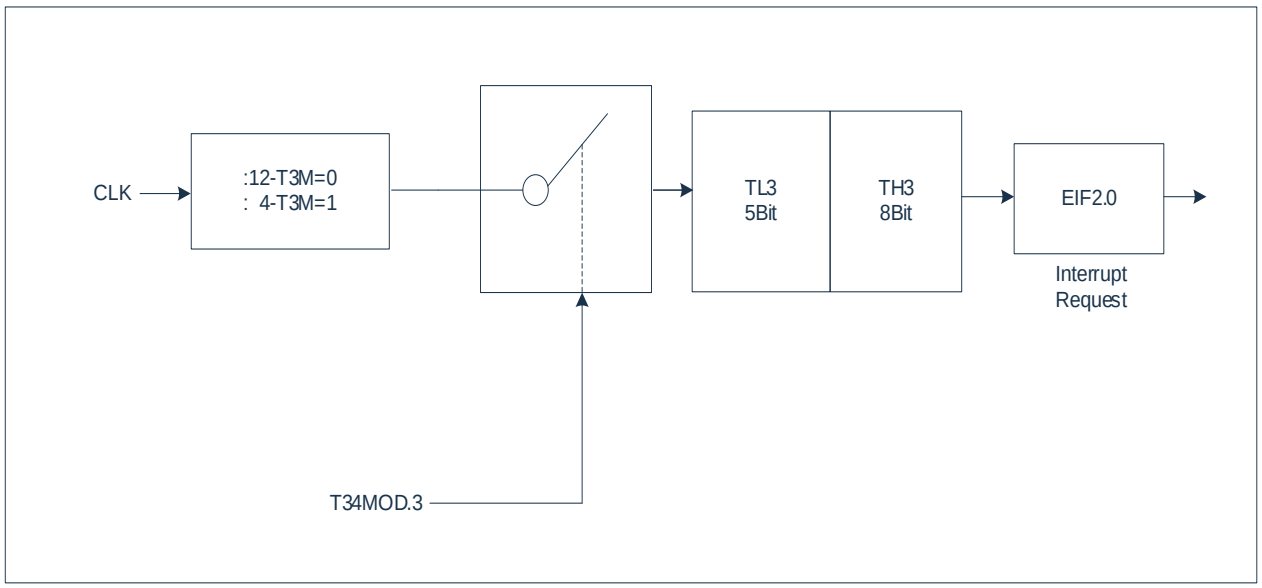
1= The Timer3 counter overflows and is automatically cleared by hardware when entering the interrupt service routine, or it can be cleared by software;

0= Timer3 counter has no overflow.

11.4 Timer3 Operation Mode

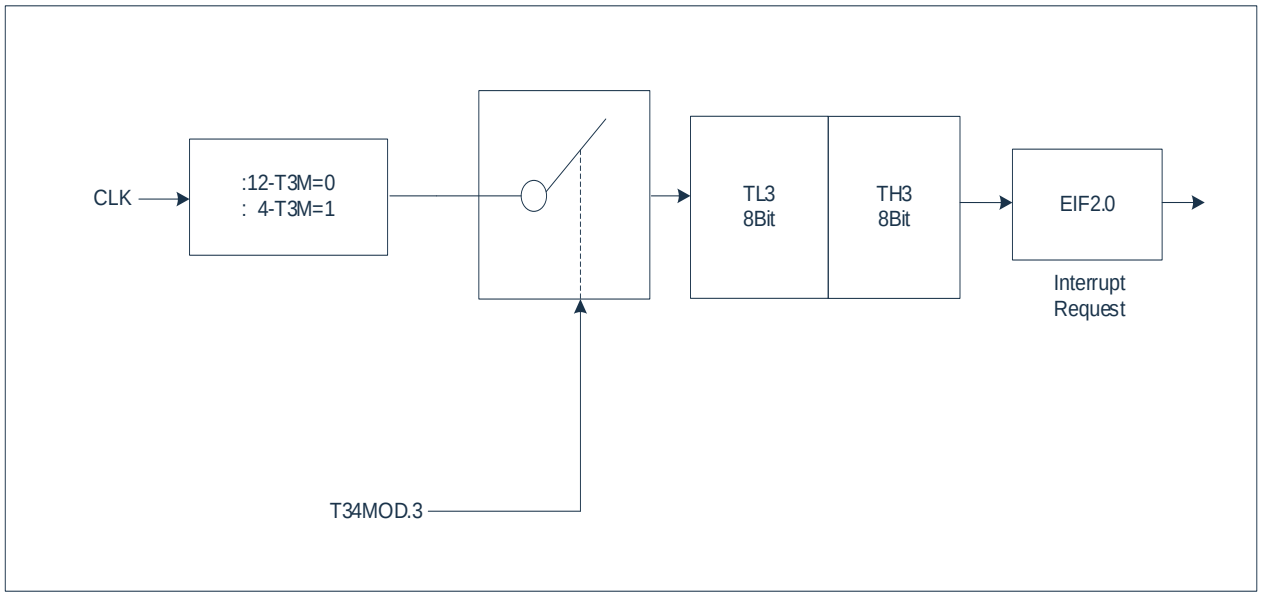
11.4.1 T3 -Mode0（13-Bit Timing Mode）

In this mode, the Timer3 is a 13-bit register. When all bits of the counter flip from 1 to 0, the Timer0 interrupt flag TF3 is set to 1. The 13-bit register is composed of 5-bit lower TL3 and TH3. The upper 3 bits of TL3 should be ignored. Timer3 Mode0 structure diagram is shown as follows :



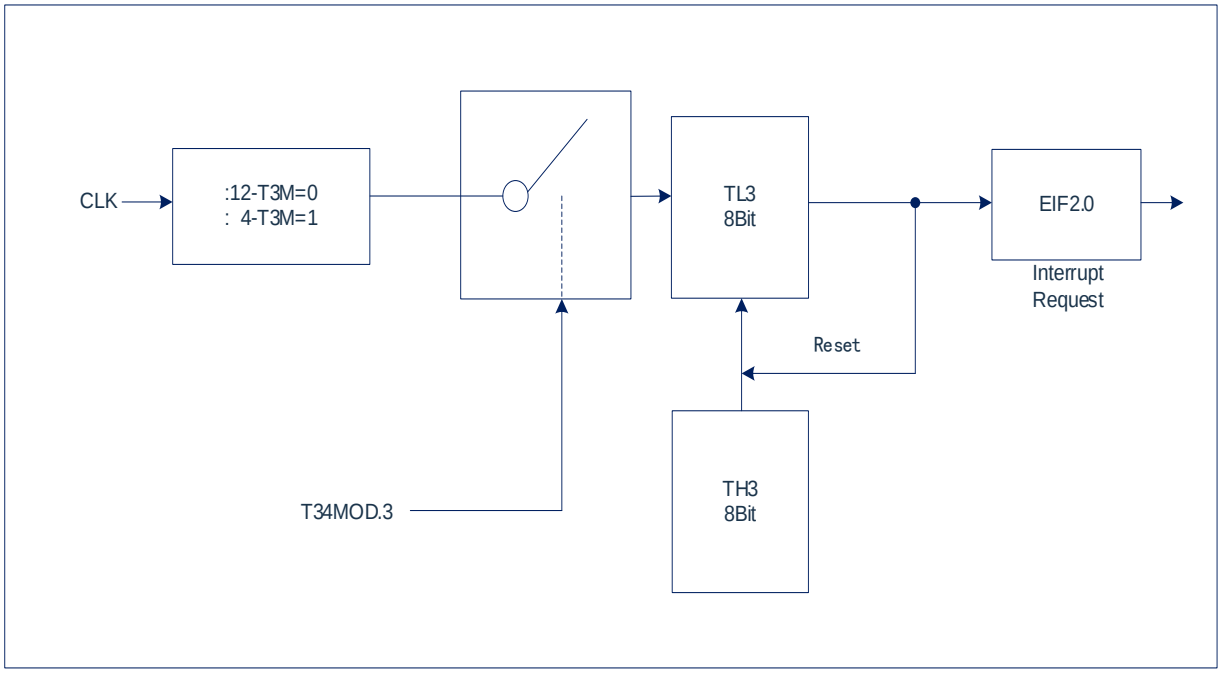
11.4.2 T3 -Mode1（16-Bit Timing Mode）

Mode1 is the same as Mode0, except that all 16 bits of the Timer3 Data Register are running in Mode1. The structure diagram of Timer3 Mode1 is shown in the figure below:



11.4.3 T3 -Mode2（8-Bit Auto Reload Timing Mode）

The Timer3 register in Mode2 is an 8-bit timer (TL3) equipped with automatic reload mode, as shown in the following figure. The overflow from TL3 not only makes TF3 set 1, but also reloads TH3 content from software to TL3. The TH3 value remains unchanged during reloading. The structure diagram of Timer3 Mode2 is as follows :



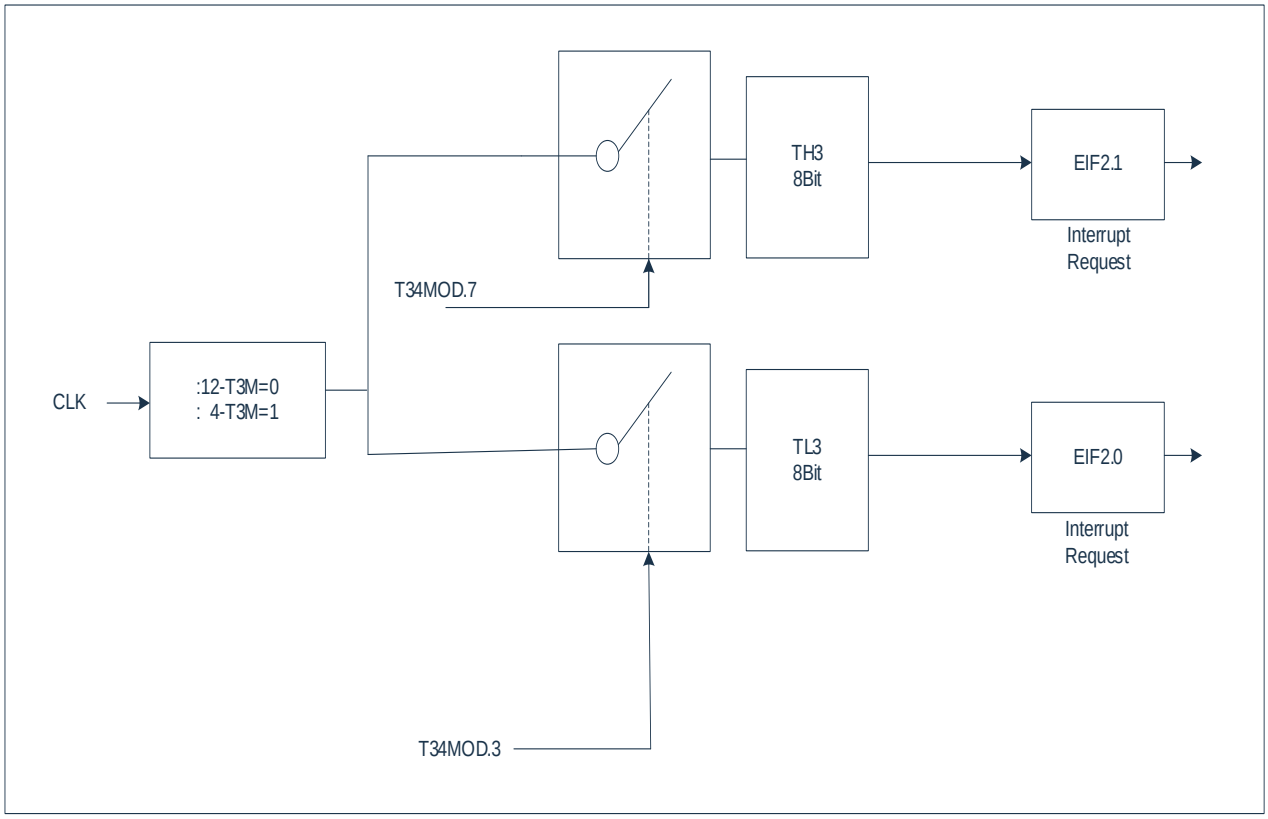
11.4.4 T3 -Mode3（Two Separate 8-Bit Timer）

Timer3 in Mode3 sets TL3 and TH3 to two independent timers. The logic of Timer3 Mode3 is shown below.

TL3 can operate as a timer and uses the control bits of Timer3:TR3, and TF3.

TH0 can operate as a timer and uses the TR4 and TF4 flags of Timer4 to control the Timer4 interrupt.

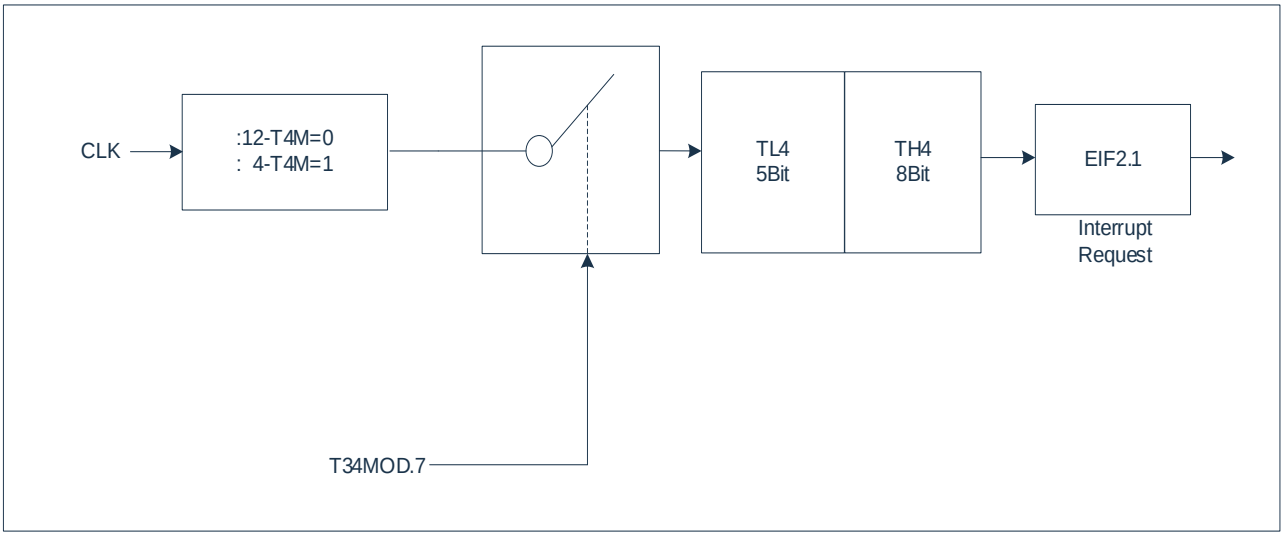
Mode3 is available when two 8-bit timer are required. When Timer3 is in Mode3, Timer4 can be turned off by switching to its own Mode3, or it can still be used as a Baud Rate generator by the serial channel, or if no Timer4 interrupt is required in the application. The structure block diagram of Timer3 Mode 3 is shown below.



11.5 Timer4 Operation Mode

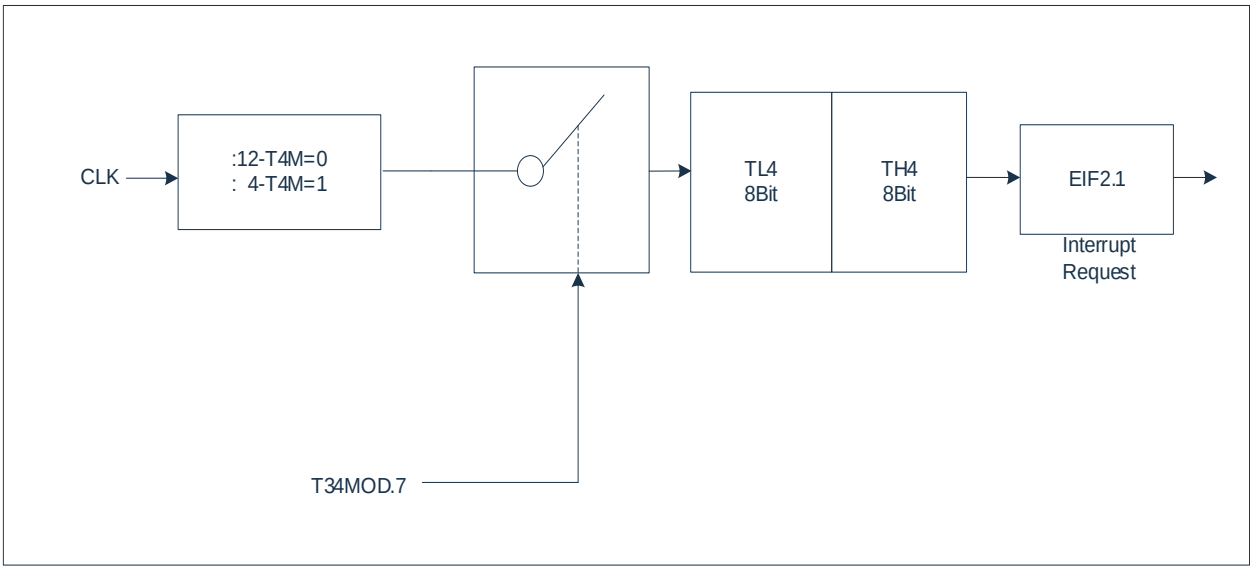
11.5.1 T4 -Mode0（13-Bit Timing Mode）

In this mode, the Timer4 is a 13-bit register. When all bits of the counter flip from 1 to 0, the Timer4 interrupt flag TF4 is set to 1. The 13-bit register is composed of 5-bit lower TL4 and TH4. The upper 3 bits of TL4 should be ignored. Timer4 Mode0 structure diagram is shown as follows :



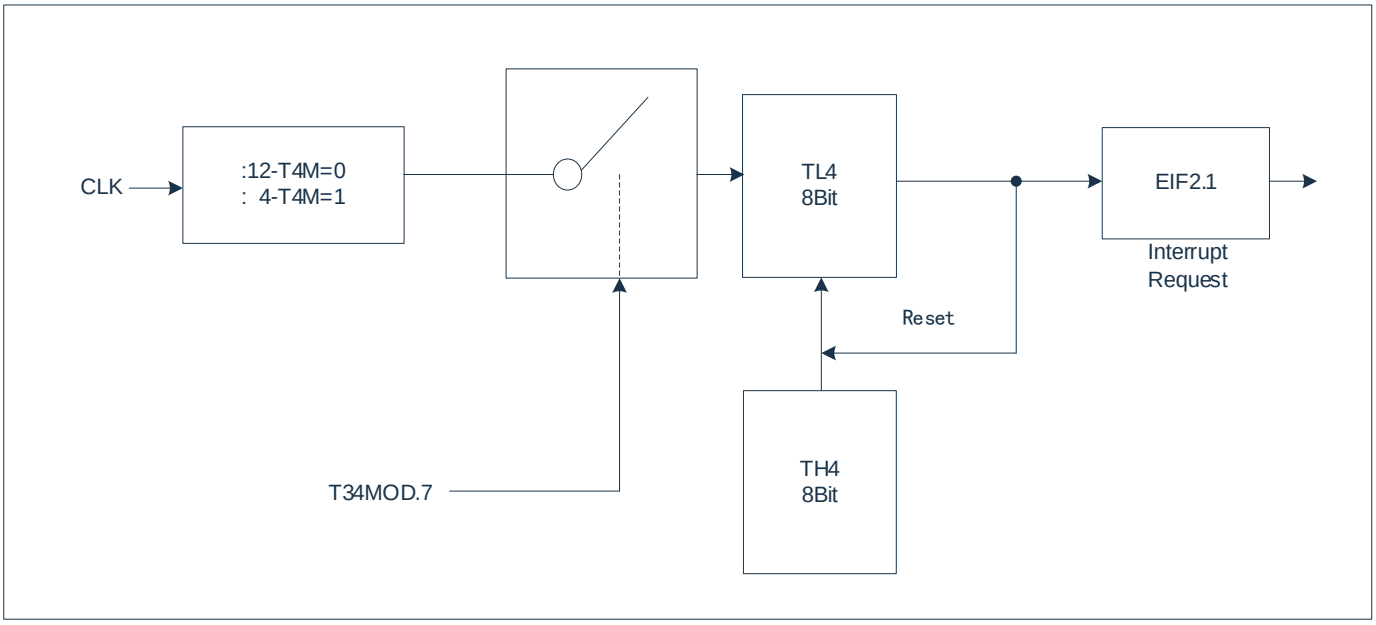
11.5.2 T4 -Mode1（16-Bit Timing Mode）

Mode1 is the same as Mode0, except that all 16 bits of the Timer4 Data Register are running in Mode1. The structure diagram of Timer4 Mode1 is shown in the figure below:



11.5.3 T4- Mode2（8-Bit Auto Reload Timing Mode）

The Timer4 register in Mode2 is an 8-bit timer (TL4) equipped with automatic reload mode, as shown in the following figure. The overflow from TL4 not only makes TF4 set 1, but also reloads TH4 content from software to TL3. The TH4 value remains unchanged during reloading. The structure diagram of Timer4 Mode2 is as follows :



11.5.4 T4- Mode3（Stop Counting）

The Timer4 in Mode3 stops counting, with the same effect as setting TR4 = 0.

12. LSE_Timer

12.1 Overview

The 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 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 {LSECRH[7:0]，LSECRL[7:0]}.

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

12.2 Related Registers

12.2.1 LSE Timer Data Register Low 8-bit LSECRL

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

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

12.2.2 LSE Timer Data Register High 8-bit 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.

12.2.3 LSE Timer Control Register LSECON

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

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

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

Bit5 LSECNTEN: LSE is used for timer counting enable control;
1= Enable;
0= Disable;

Bit4 LSESTA: LSE stable state, read only;
1= LSE is stable;
0= LSE is not stable;

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

Bit2~Bit1 -- Reserved, all must be 0;

Bit0 LSEIF: LSE is used for timer interrupt flag (cleared by software);
1= Produce an interrupt;
0= No Interrupted or the interrupt is cleared.

12.3 Interrupt And Sleep Wake-up

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

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

Bit7~Bit5 -- Reserved, all must be 0.

Bit4 PTOUCH: TOUCH interrupt priority control;

1= Set to high priority interrupt;

0= Set to low priority interrupt.

Bit3 PLVD: LVD interrupt priority control;

1= Set to high priority interrupt;

0= Set to low priority interrupt.

Bit2 PLSE: LSE interrupt priority control;

1= Set to high priority interrupt;

0= Set to low priority interrupt.

Bit1~Bit0 -- Reserved, all must be 0.

When the count value of the LSE timer is equal to timing value, the interrupt flag bit LSEIF is set to 1. If the master interruption enable bit (EA=1) is set to 1, and the LSE timer interruption is enabled (LSEIE=1), then the CPU will enter the interrupt service program.

To use LSE timer interrupt to wake-up the sleep mode, you need to turn on LSEEN, LSECNT, LSEWUEN before sleep, and set the time from sleep to wakeup {LSECRH[7:0], LSECRL[7:0]}. If the master interrupt enable and LSE interrupt enable are turn on before 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.

12.4 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 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 timing value, the interrupt flag bit 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 timing value of the timer is 1. If the timing value is set to 0, the timer defaults to 1 as the timing value. The formula for calculating the timing time of LSE timer is as follows:

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

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

13. Wake-Up Timer（WUT）

13.1 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 regularly 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, WUT starts counting and when the count value is equal to the set value, the chip enters the sleep wake-up waiting state.

13.2 Related Registers

13.2.1 WUTCRH Register

0xBD	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WUTCRH	WUTEN	--	WUTPS1	WUTPS0	WUTD11	WUTD10	WUTD9	WUTD8
R/W	R/W	R	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;
1= Timing wake-up function is turn on;
0= Timing wake-up function is turn off;
Bit6 -- Reserved, must be 0;
Bit5~Bit4 WUTPS<1:0>: Timing wake-up counter clock divider;
00= F/1;
01= F/8;
10= F/32;
11= F/256.
Bit3~Bit0 WUTD<11:8>: High 4 bits of timing wake-up time data;

13.2.2 WUTCRL Register

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

Bit7~Bit0 WUTD<7:0>: Low 8 bits of timing wake-up time data.

13.3 Function Description

The principle of the internal wake-up timer is: after system enters the sleep mode, the CPU and all peripheral circuits stop working, and internal low power consumption oscillation 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 和 WUTRCL.

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

- WUTEN=1: Timing wake-up function is turn on;
- WUTEN=0: Timing wake-up function is turn off;

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

14. Baud Rate Timer（BRT）

14.1 Overview

There is a 16-bit baud rate timer BRT inside the chip, which mainly provides the clock for the UART module.

14.2 Related Registers

14.2.1 BRT Module Control Register BRTCON

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

Bit7 BRTEN: BRT timer enable;
1= Enable;
0= Disable;

Bit6~Bit3 -- Reserved, all must be 0;

Bit2~Bit0 BRTCKDIV<2:0> BRT timer prescaler selection;
000= Fsys/1;
001= Fsys/2;
010= Fsys/4;
011= Fsys/8;
100= Fsys/16;
101= Fsys/32;
110= Fsys/64;
111= Fsys/128.

14.2.2 BRT Timer Data Is Loading The 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-bit;

14.2.3 BRT Timer Data Is Loading The 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-bit;

14.3 Function Description

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

When the timer enable bit BRTEN=1, the counter starts to work. When the value of 16-bit counter is equal to FFFFH, the BRT counter overflows. And after overflow, the initial value of the count is automatically loaded into the counter, and then the count is restarted.

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

BRT timer overflow rate:

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

15. Cycle Redundancy Check（CRC）

15.1 Overview

In order to ensure safety during operation, the IEC61508 standard requires confirmation of data even when the CPU is running. This general-purpose CRC module can perform CRC calculations as a peripheral function during CPU operation. The general-purpose CRC module specifies the data to be confirmed by the program for CRC checking, not limited to the code flash area but can be used for multi-purpose checking.

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

15.2 Related Registers

15.2.1 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> Enter 8-bit data that requires CRC operation.

15.2.2 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.

15.2.3 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.

15.3 Function Description

After writing the CRCIN register, and after a system clock, save the CRC operation result to the CRCDL/CRCDH register. If necessary, read the previous calculation data before writing overwise, otherwise it will be overwritten by the new calculation result.

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

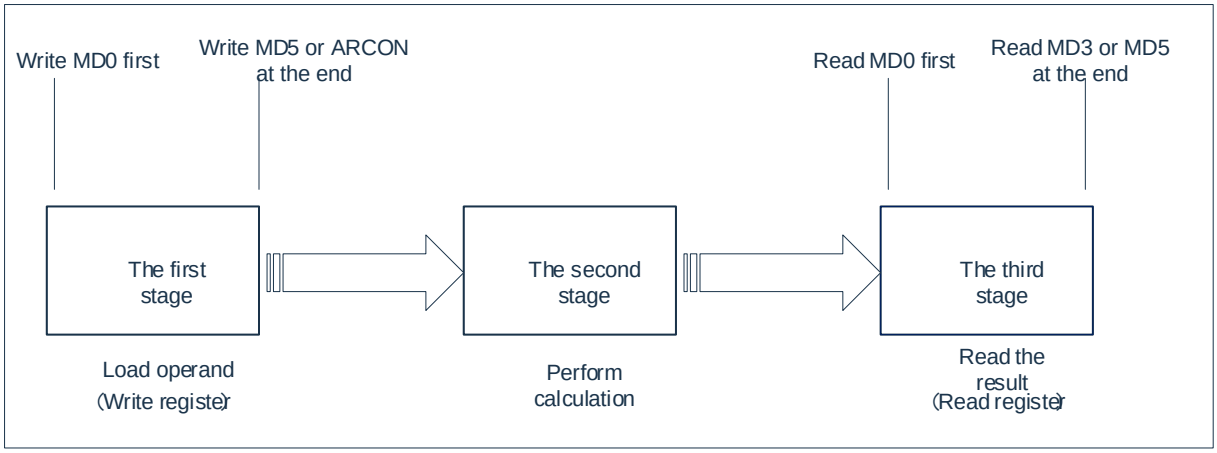
```
CRCIN=0x12;     // send the first number
CRCIN=0x34;     // send the second number
CRCIN=0x56;     // send the third number
CRCIN=0x78;     // send the fourth number
resl=CRCDL;     // read the low 8-bit of the CRC operation result to the variable resl
resh=CRCDH;     // read the high 8-bit of the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

16. Multiplication/Division Unit（MDU）

16.1 Overview

MDU（Multiplication/Division Unit） module provides the 32bit/16bit division, 16bit/16bit division, 16bit*16bit multiplication, 32bit shift operation, 32bit normalization operation functions. All operations are unsigned integer operation, among which shift operation support 32bit data left shift or right shift operation. The operation of 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 starts, the result after the operation and the remainder, and the 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, but the second stage can run independently of the CPU. when writing the MD5 or ARCON register to starts the MDU operation, the MDU reads the register to get the result after a fixed time. The diagram of the MDU module operation stage is as follow:



16.2 Related Registers

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

16.2.1 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: write/read data bit7-bit0;
Normalization operation: write/read data bit7-bit0.

16.2.2 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 bit15-bit8 as the dividend, read bit15-bit8 as the quotient;
16bit/16bit division operation: write bit15-bit8 as the dividend, read bit15-bit8 as the quotient;
16bit*16bit multiplication operation: write bit15-bit8 as the first multiplier, read bit15-bit8 as the product;
Shift operation: write/read data bit15-bit8;
Normalization operation: write/read data bit15-bit8.

16.2.3 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 the dividend, read bit23-bit16 as the quotient;
16bit*16bit multiplication operation: read bit23-bit16 as the product;
Shift operation: write/read data bit23-bit16;
Normalization operation: write/read data bit23-bit16.

16.2.4 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 the dividend, read bit31-bit24 as the quotient;
16bit*16bit multiplication operation: read bit31-bit24 as the product;
Shift operation: write/read data bit31-bit24;
Normalization operation: write/read data bit31-bit24.

16.2.5 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 as the divisor, read bit7-bit0 as the remainder;
16bit/16bit division operation: write bit7-bit0 as the divisor, read bit7-bit0 as the remainder;
16bit*16bit multiplication operation: write bit7-bit0 as the second multiplier.

16.2.6 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 as the divisor, read bit15-bit8 as the remainder;

16bit/16bit division operation: write bit15-bit8 as the divisor, read bit15-bit8 as the remainder;

16bit*16bit multiplication operation: write bit15-bit8 as the second multiplier.

16.2.7 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: Failure flag (the bit is set by hardware)

1= When the MD0~MD5 register is written during the MDU module loading data, it is set to 1;

0= Read the ARCON register and clear the bit to 0.

Bit6 MDOV: Overflow flag (the bit is set by hardware)

1= The divisor is 0;

Or the multiplier operation result is more than the 0xffff;

Or normalization operation MD37=1;

0= The bit is 0 when the condition does not match to 1 (clear this bit by writing MD0 register, and writing 0 by software is invalid).

Bit5 SLR: Shift operation direction control

1= Right shift;

0= Left shift.

Bit4~Bit0 SC<4:0>: Number of shift/number of normalization

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

Pre-write ≠ 00000 (store the number of shift bits) to start the shift operation function.

16.3 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 ARCON register. The multiplication and division operation sequence of the MDU module is shown in the following table:

Operation stage	Operation sequence	32bit/16bit	16bit/16bit	16bit*16bit
The first stage	The first write	MD0 (bit 7-0 of the dividend)	MD0 (bit 7-0 of the dividend)	MD0 (bit 7-0 of the multiplier 0)
		MD1 (bit 15-8 of the dividend)	MD1 (bit 15-8 of the dividend)	MD4 (bit 7-0 of the multiplier 1)
		MD2 (bit 23-16 of the dividend)	--	--
		MD3 (bit 31-24 of the dividend)	--	--
		MD4 (bit 7-0 of the divisor)	MD4 (bit 7-0 of the divisor)	MD1 (bit 15-8 of the multiplier 0)
	The last write	MD5 (bit 15-8 of the divisor)	MD5 (bit 15-8 of the divisor)	MD5 (bit 15-8 of the multiplier 1)
The third stage	The first read	MD0 (bit 7-0 of the quotient)	MD0 (bit 7-0 of the quotient)	MD0 (bit 7-0 of the product)
		MD1 (bit 15-8 of the quotient)	MD1 (bit 15-8 of the quotient)	MD1 (bit 15-8 of the product)
		MD2 (bit 23-16 of the quotient)	--	--
		MD3 (bit 31-24 of the quotient)	--	--
		MD4 (bit 7-0 of the remainder)	MD4 (bit 7-0 of the remainder)	MD2 (bit 23-16 of the product)
	The last read	MD5 (bit 15-8 of the remainder)	MD5 (bit 15-8 of the remainder)	MD3 (bit 31-24 of the product)

All calculation processes of MDU are completed by hardware, and the conversion rate is fast, which can save a lot of time for program operation. The calculation time of the five operations of the MDU module is shown in the following table:

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

Tsys in the above table is the clock cycle of the MDU module; The operation time is the calculation time of the MDU module, excluding the time to read and write registers; The operation time of shift operation and normalization operation depends on the number of shift bits and operands respectively; The minimum operation time of shift operation is 2 Tsys, and the maximum is 17 Tsys; The minimum operation time of normalization operation is 1 Tsys, and the maximum is 17 Tsys.

16.3.1 32bit/16bit Division Operation

The operation steps of the 32bit/16bit divider 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）, start the division operation after writing is completed;
7) Wait for 16 clock cycles of the MDU module to ensure that 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）, read to complete a division operation.

If the operation result has not been read after the operation is completed, MD0 can be rewritten to start the next operation.

For example, when the dividend is 0x87654321, the divisor is 0x1234, write MD0=0x21, MD1=0x43, MD2=0x65, MD3=0x87, MD4=0x34, MD5=0x12, the read result after MDU calculation is: MD0=0x23, MD1=0x70, MD2=0x07, MD3=0x00, MD4=0x05, MD5=0x06.

16.3.2 16bit/16bit Division Operation

The operation steps of the 16bit/16bit divider 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）, start the division operation after writing is completed;
5) Wait for 8 clock cycles 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）, read to complete a division operation.

If the operation result has not been read after the operation is completed, MD0 can be rewritten to start the next operation.

For example, when the dividend is 0x4321, the divisor is 0x1234, write MD0=0x21, MD1=0x43, MD4=0x34, MD5=0x12, the read result after MDU calculation is: MD0=0x03, MD1=0x00, MD4=0x85, MD5=0x0C.

16.3.3 16bit*16bit Multiplication Operation

The operation steps of the 16bit/16bit multiplier 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）, start the multiplication operation after writing is completed;
5) Wait for 8 clock cycles 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）, read to complete a multiplication operation.

If the operation result has not been read after the operation is completed, MD0 can be rewritten to start the next operation.

For example, when the first multiplier is 0x8765, the second multiplier is 0x1234, write MD0=0x65, MD4=0x34, MD1=0x87, MD5=0x12, the read result after MDU calculation is: MD0=0x84, MD1=0x9A, MD2=0xA0, MD3=0x09.

16.3.4 32bit Shift Operation

The 32bit 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, start the shift operation after writing is completed;
6) Wait for 17 clock cycles of the MDU module to ensure that 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）, read to complete a shift operation.

If the operation result has not been read after the operation is completed, MD0 can be rewritten to start the next operation.

For example, the operand is 0x12345678, when shifting 5 bits to the right, write MD0=0x78, MD1=0x56, MD2=0x34, MD3=0x12, ARCON=0x25, the read result after MDU calculation is: MD0=0xB3, MD1=0xA2, MD2=0x91, MD3=0x00.

16.3.5 32bit Normalization Operation

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

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, start the normalization operation after writing is completed;
6) Wait for 17 clock cycles of the MDU module to ensure that the calculation is completed;
7) Read the register MD0（bit7-bit0 of the normalization result）;
8) Read the register MD1（bit15-bit8 of the normalization result）;
9) Read the register MD2（bit23-bit16 of the normalization result）;
10) Read the register MD3（bit31-bit24 of the normalization result）, read to complete a normalization operation.

If the operation result has not been read after the operation is completed, MD0 can be rewritten to start the next operation.

For example, the operand is 0x12345678, during the normalization operation, write MD0=0x78, MD1=0x56, MD2=0x34, MD3=0x12, ARCON=0x00, the read result after MDU calculation is: MD0=0xC0, MD1=0xB3, MD2=0xA2, MD3=0x91, ARCON=0x03.

17. BUZZER

17.1 Overview

The buzzer 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 of BUZZER is controlled by the BUZCON register and the BUZDIV register. And its frequency output can cover a wide range.

17.2 Related Registers

17.2.1 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;
1= Enable;
0= Disable;
Bit6~Bit2 -- Reserved, all must be 0.
Bit1~Bit0 BUZCKS<1:0>: BUZZER frequency division ratio selection;
00= Fsys/8;
01= Fsys/16;
10= Fsys/32;
11= Fsys/64.

17.2.2 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;
0x00= No square wave output;
Others= Fbuz =Fsys/(2*CLKDIV*BUZCKS).

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

17.3 Function Description

When using the BUZZER, you need to configure the corresponding port as a BUZZER drive output port. For example, configure P16 as a BUZZER drive output port, the configuration is as follows:

```
P16CFG = 0x04;          // P16 is configured the BUZZER drive output port
```

By configuring the corresponding register of the BUZZER drive module, you can set the BUZZER drive output port to output different frequencies. For example:

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

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

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

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

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

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

Different output frequency can be obtained by selecting the different system clock frequency and BUZZER drive clock frequency division ratio. The BUZZER drive output frequency is shown in the following table:

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

18. Enhanced PWM Module

18.1 Overview

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

Each PWM has an independent 16-bit period register and a 16-bit duty cycle register (compare data register and down compare data register) to configure the period of the PWM output and adjust the duty cycle. And each PWM has an independent clock divider control register, and each pair of PWM shares an 8-bit prescaler control register.

Each PWM can configured the edge-aligned counting or center-aligned counting mode. In center-aligned counting mode, it can also be set to symmetric counting and asymmetric counting. Each PWM can be set to output in single mode (generating a PWM signal cycle) or automatic load (continuous output of PWM waveform), and its output polarity can also be set by output polarity controller.

Enhanced PWM also supports mask output function, hardware brake protection function, and interrupt function. The 6-channel PWM generators provide a total of 25 interrupt flags, including zero-point interrupt, upward comparison interrupt, period interrupt, downward comparison interrupt, and brake interrupt. These interrupts share an interrupt vector entry.

18.2 Characteristic

Enhanced PWM supports the following characteristic:

- 6-channel independent 16-bit PWM control mode.
  - 6-channel independent output: PG0, PG1, PG2, PG3, PG4, PG5;
  - 3-channel complementary PWM pairs output:（PG0-PG1）, PG2-PG3）,（PG4-PG5）, capable of programmable dead-time insertion;
  - Three synchronous PWM pairs output:（PG0-PG1）,（PG2-PG3）,（PG4-PG5）, with each pin in a pair in-phase.
- Support group control, the outputs of PG0 and PG2 and PG4 are synchronized, the outputs of PG1 and PG3 and PG5 are synchronized.
- Support edge-aligned mode and center-aligned mode.
- Support symmetrical and asymmetrical counting in center-aligned mode.
- Support single mode or automatic load mode.
- Each PWM generator has independent polarity control.
- mask output function.
- Hardware brake protections（external FB0/FB1 trigger, ADC comparing event trigger, ACMP output trigger）.
- PWM edge or period triggers to start AD conversion.

18.3 Pin Configuration

It is necessary to configure corresponding ports as PWM channel before using enhanced PWM block, PWM channel is marked with PG0-PG5 on the pin distribution diagram, corresponding to PWM channel 0-5.

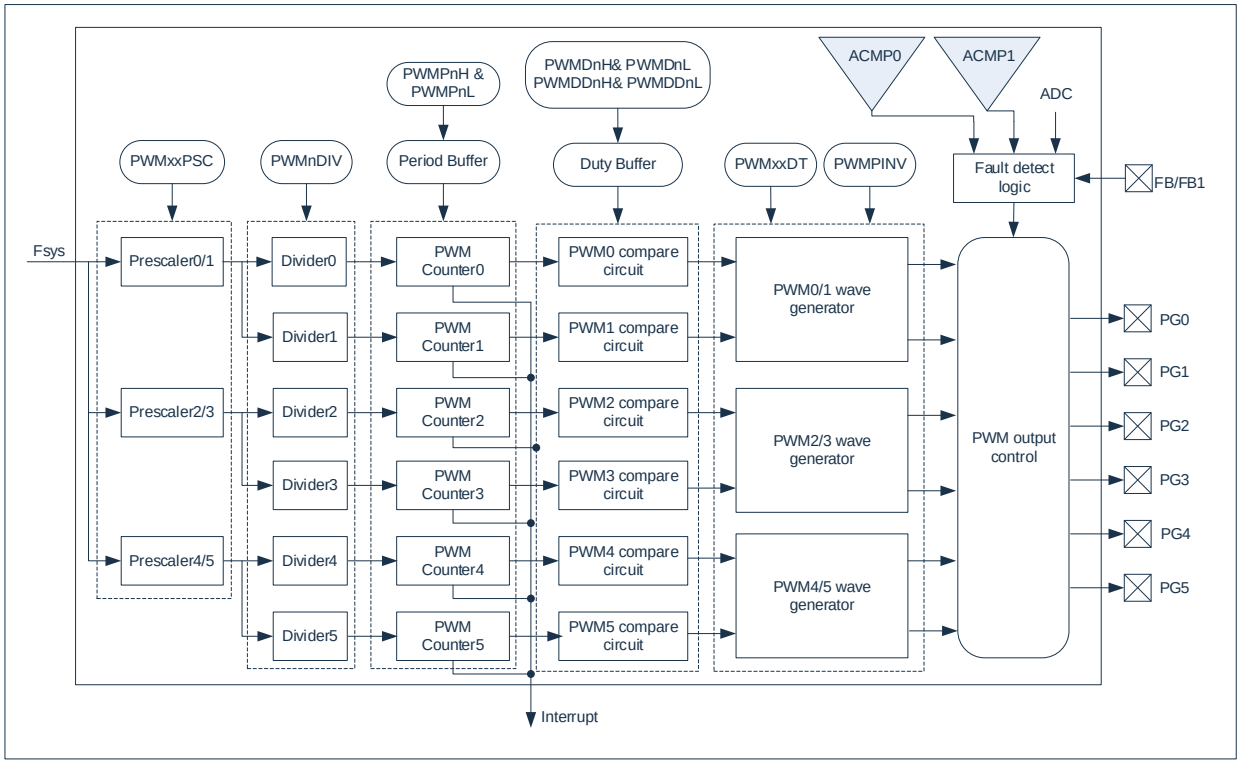
The distribution of PWM channel is controlled by corresponding port configuration register, for example:

```
P00CFG=0x05;    //Configure P00 as PG0 channel
P01CFG=0x05;    //Configure P01 as PG1 channel
P02CFG=0x05;    //Configure P02 as PG2 channel
P03CFG=0x05;    //Configure P03 as PG3 channel
P04CFG=0x05;    //Configure P04 as PG4 channel
P05CFG=0x05;    //Configure P05 as PG5 channel
```

18.4 Function Description

18.4.1 Functional Diagram

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



18.4.2 Edge-aligned

In edge-aligned mode, the 16-bit PWM counter CNTn starts counting down in each period and compares it with the value CMPn latched in the PWMDnH/PWMDnL register. When CNTn=CMPn, PGn outputs a high level and PWMnDIF is set to 1. CNTn continues counting down to 0, and at this time, PGn will output low level and PWMnZIF will be set to 1. When CNTn counts to 0, if PWMnCNTM=1, CMPn and PERIODn will be reloaded.

The relevant parameters of edge-aligned are as follows:

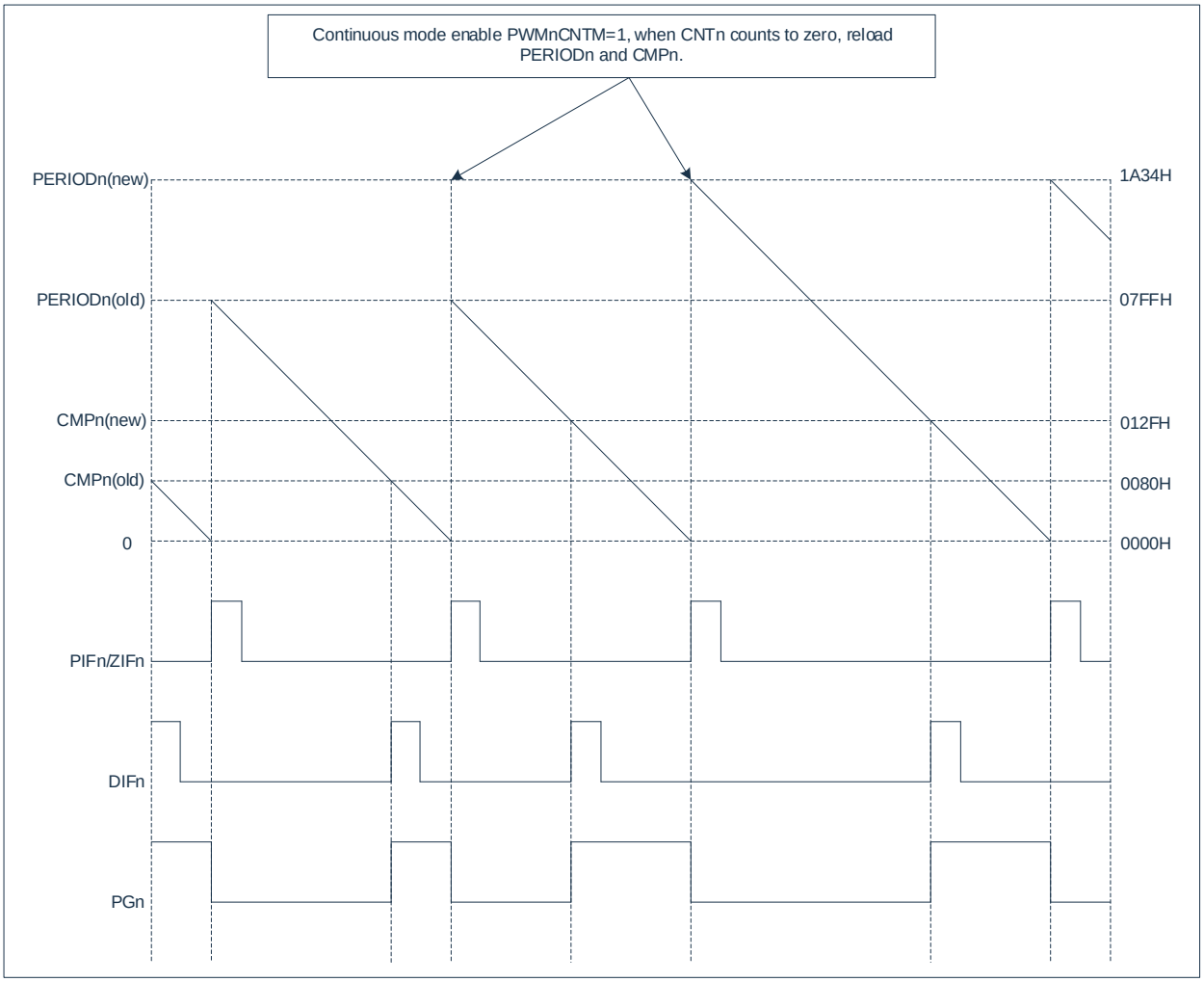
$$High\ level\ voltage\ time = (CMPn+1) \times Tpwm (CMPn \geq 1)$$

$$Period = (PERIODn+1) \times Tpwm$$

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

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

The edge-aligned timing is shown in the figure below:



18.4.3 Center-aligned

In the center-aligned counting mode, symmetric counting and asymmetric counting are supported. To turn on the asymmetric counting mode, you need to set ASYMEN to 1. In the asymmetric counting mode, accurate center-aligned waveforms can be achieved.

18.4.3.1 Symmetric counting

In center-aligned symmetrical counting type,16-bit PWM counter (CNTn) will start counting up from 0. When CNTn=CMPn, PGn will output high level and PWMnUIF is set to 1; After that, CNTn continues to count up until it is equal to PERIODn, and PWMnPIF is set to 1; Then CNTn starts to count down. When CNTn=CMPn, PGn outputs low level and PWMnDIF is set to 1; After that, the counter continues to count down to 0, and PWMnZIF is set to 1.

The relevant parameters of the center-aligned symmetrical counting type are as follows:

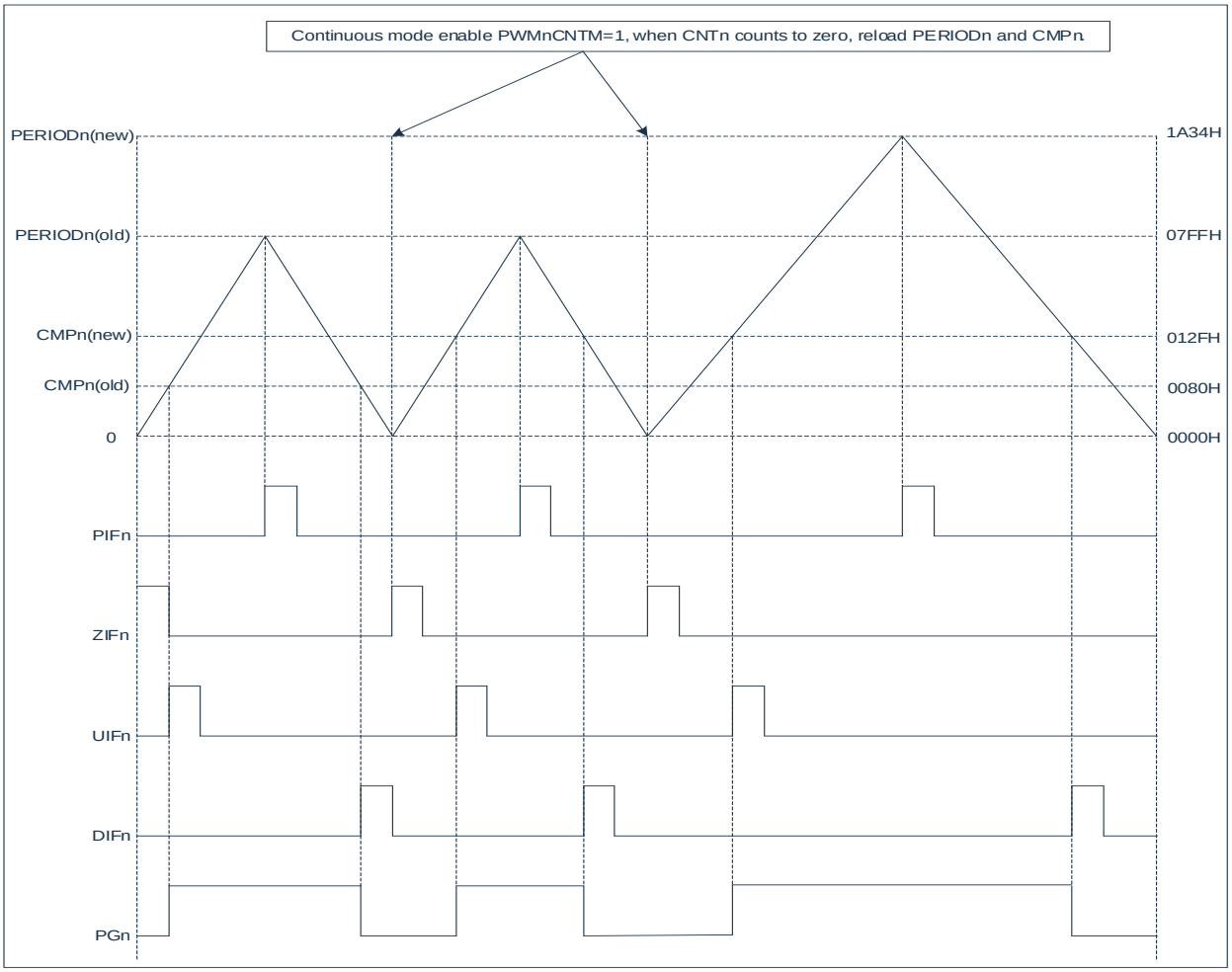
$$High\ level\ voltage\ time = (PERIODn \times 2 - CMPn \times 2 - 1) \times Tpwm;\ (CMPn \geq 1)$$

$$Period = (PERIODn) \times 2 \times Tpwm;$$

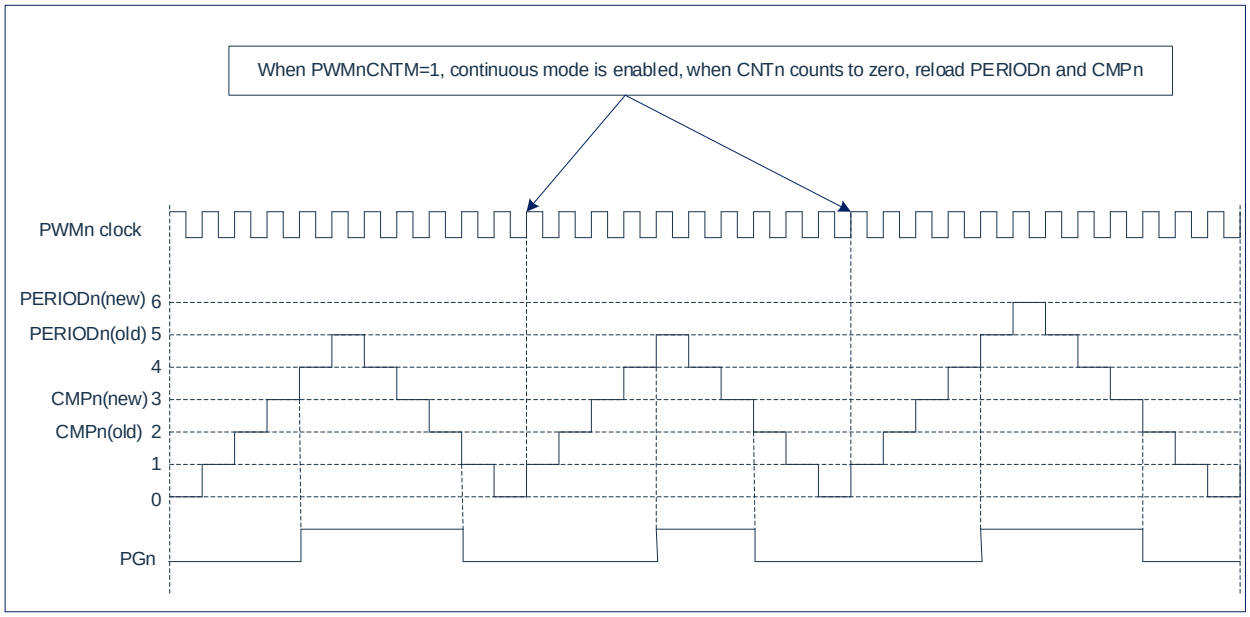
$$Duty\ cycle = \frac{PERIODn \times 2 - CMPn \times 2 - 1}{PERIODn \times 2};\ (CMPn \geq 1)$$

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

The center-aligned symmetric counting timing is shown in the figure below:



The center-aligned counter waveform (symmetric counting) is shown in the figure below:



18.4.3.2 Asymmetric counting

In center-aligned asymmetric counting mode, the 16-bit PWM counter CNTn counts up from 0. When CNTn=CMPn, PGn outputs a high level and PWMnUIF is set to 1; After that, CNTn continues to count up until it is equal to PERIODn, and PWMnPIF is set to 1; Then CNTn starts to count down. When CNTn=CMPDn (PWMDDnH/PWMDDnL) during the down count, PGn outputs low level and PWMnDIF is set to 1. Then continue to count down to 0, and set PWMnZIF to 1.

The relevant parameters of the center-aligned asymmetrical counting type are as follows:

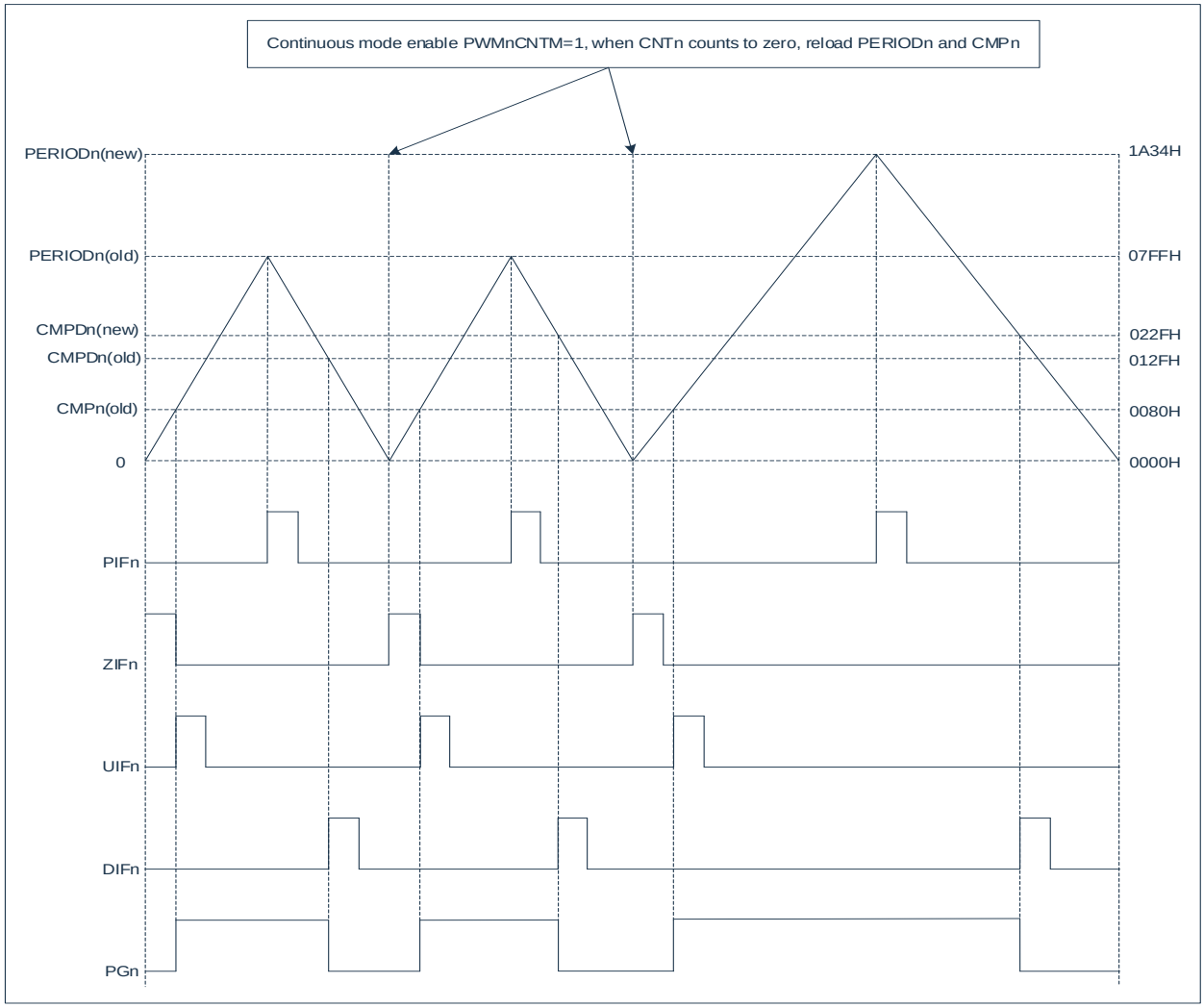
$$High\ level\ voltage\ time = (PERIODn \times 2 - CMPDn - CMPn - 1) \times Tpwm$$

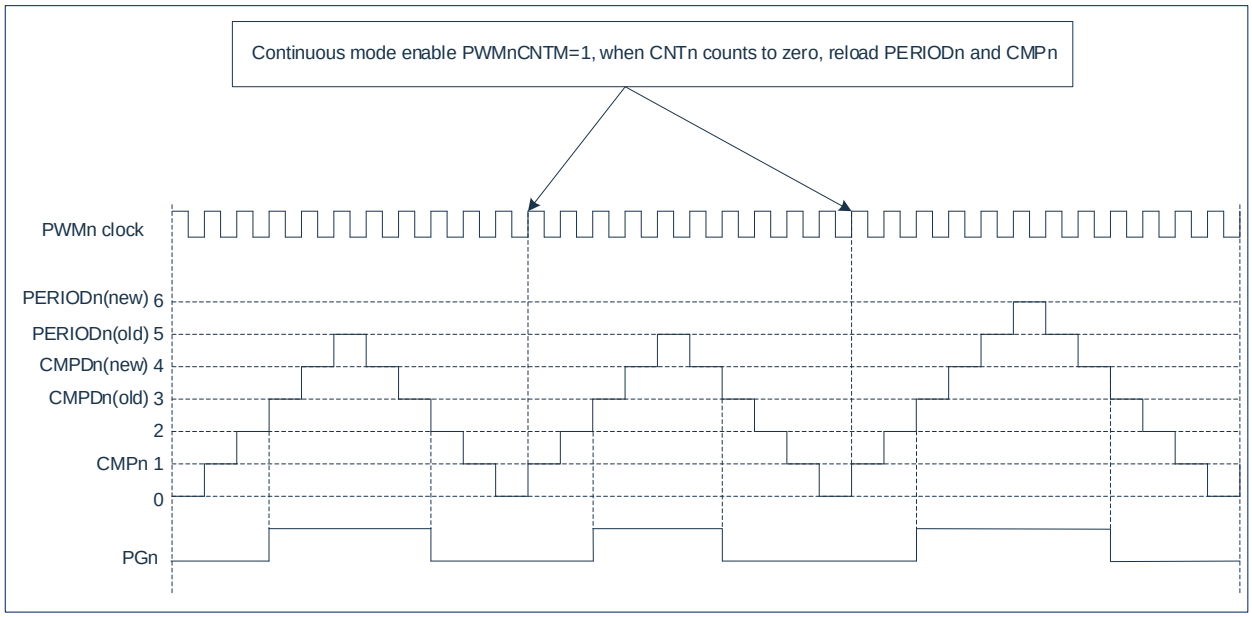
$$Period = (PERIODn) \times 2 \times Tpwm$$

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

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

The center-aligned asymmetric counting timing is shown in the figure below:





18.4.4 Complementary Mode

The 6-channel PWM can be set as 3 complementary pairs. In complementary mode, the period, duty cycle and clock division control of PG1, PG3 and PG5 are determined by the related registers of PG0, PG2 and PG4 respectively. That is, in addition to corresponding output enable control bit （PWMnOE）, The output waveforms of PG1, PG3 and PG5 are no longer controlled by their own registers.

In complementary mode, each of PWM complementary pairs support insert dead zone delay. Insert dead zone time as follows:

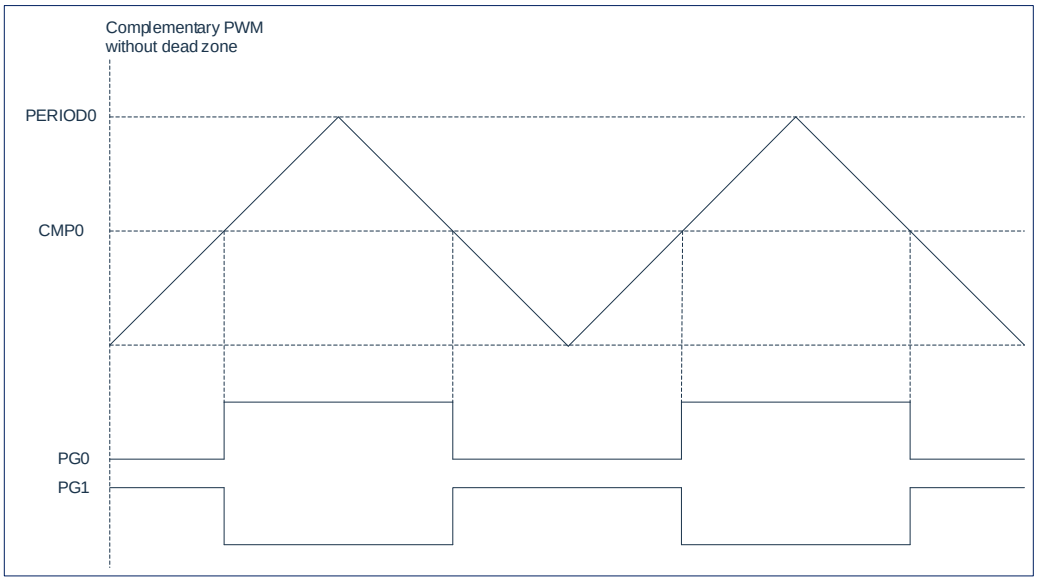
PWM0/1 dead zone time:（PWM01DT+1）*T_{PWM0}；

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

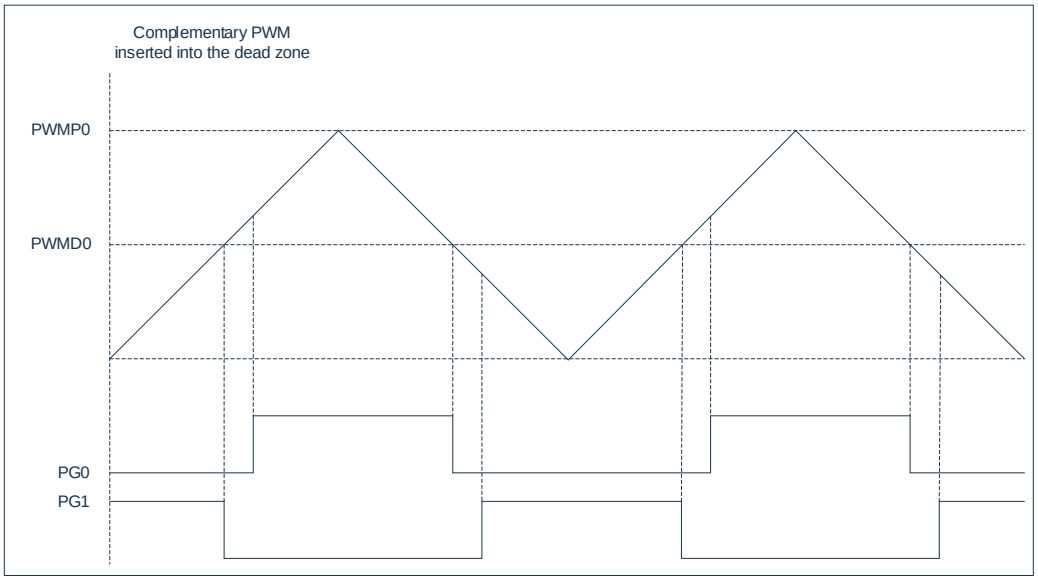
PWM4/5 dead zone time:（PWM45DT+1）*T_{PWM4}；

T_{PWM0}/T_{PWM2}/T_{PWM4} are clock source cycles PG0/PG2/PG4 respectively.

Take PG0/PG1 as an example, the waveform diagram without dead zone in complementary mode is shown in the figure below:



Take PG0/PG1 as an example, the waveform diagram with dead zone in complementary mode is shown in the figure below:



18.4.5 Synchronize Mode

The 6-channel PWM can be set as 3 complementary pairs. In synchronize mode, the period, duty cycle and clock division control of PG1, PG3 and PG5 are determined by the related registers of PG0, PG2 and PG4 respectively. That is, in addition to corresponding output enable control bit （PWMnOE）, The output waveforms of PG1, PG3 and PG5 are no longer controlled by their own registers. The output waveform of PG1 is the same as PG0, the output waveform of PG3 is the same as PG2, and the output waveform of PG5 is the same as PG4.

18.4.6 Mask Output

Enhanced PWM support the mask function. Each channel of PG0-PG5 has a separate control. The corresponding mask enable control is set by mask control register PWMMASKE, and the mask output data is set by mask data register PWMMASKD.

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

When MASKEn=1, the PGn channel outputs the data of MASKDn.

18.4.7 Brake Function

Enhanced PWM can be configured as software or hardware to trigger the brake function through the brake control register PWMFBKC. The hardware trigger brake signal sources are as follows:

- External trigger port FB0;
- External trigger port FB1;
- ADC result compare output;
- Output of ACMP 0;
- Output of ACMP 0.

Through the brake control register PWMFBCK, the external trigger port FB0/FB1 can be configured to trigger PWM brake enable and trigger type（high level or low level trigger）, The ADC comparator result can be configured to control the PWM brake enable through the ADC comparator control register ADCMPC, and the ACMP0/ACMP1 output control PWM brake enable is configured through the comparator brake control register CNFBCON.

After the brake is triggered, the brake flag is set to 1, the counter enable bits of all channels are cleared by hardware, and the PWM outputs the preset brake data. The preset data of PWM brake output is configured through the brake data register PWMFBKD. If you want to restore normal output, you need to clear the brake flag and re-enable the PWM channel counter.

18.5 PWM Related Registers

18.5.1 PWM Control Register PWMCON

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

Bit7 -- Reserved, must be 0;

Bit6 PWMRUN: PWM clock prescaler, clock divider enable;

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

0= Enable.

Bit5~Bit4 PWMMODE<1:0>: Control bit of PWM's mode;

00= Independent mode;

01= Complementary mode;

10= Synchronize mode;

11= Reserved.

Bit3 GROUPEN: PWM group function enable;

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

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

Bit2 ASYMEN: Asymmetric count enable in PWM center-aligned;

1= Asymmetric counting is enabled;

0= Symmetric counting is enabled.

Bit1 CNTTYPE: PWM count alignment select;

1= Center-aligned;

0= Edge-aligned;

Bit0 -- Reserved, must be 0.

18.5.2 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, all must be 0.

Bit5 PWM5OE: Output enable of PWM channel 5;

1= Enable;

0= Disable.

Bit4 PWM4OE: Output enable of PWM channel 4;

1= Enable;

0= Disable.

Bit3 PWM3OE: Output enable of PWM channel 3;

1= Enable;

0= Disable.

Bit2 PWM2OE: Output enable of PWM channel 2;

1= Enable;

0= Disable.

Bit1 PWM1OE: Output enable of PWM channel 1;

1= Enable;

0= Disable.

Bit0 PWM0OE: Output enable of PWM channel 0;

1= Enable;

0= Disable.

18.5.3 PWM0/1 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>: Prescaler control of PWM channel 0/1;

00= Prescaler clock stop, PWM0/1counter stop;

other= System clock（PWM01PSC+1）frequency division.

18.5.4 PWM2/3 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>: Prescaler control of PWM channel 2/3;

00= Prescaler clock stop, PWM2/3 counter stop;

other= System clock（PWM23PSC+1）frequency division.

18.5.5 PWM4/5 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>: Prescaler control of PWM channel 4/5;

00= Prescaler clock stop, PWM4/5 counter stop;

other= System clock（PWM45PSC+1）frequency division.

18.5.6 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, all must be 0.

Bit2~Bit0 PWMnDIV<2:0>: Clock frequency division control of PWM channel n;

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 prescalered clock）.

18.5.7 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, all must be 0.

Bit5~Bit0 PWMnLE: Data load enable control register of PWM channel n(n=0-5)（hardware reset after loading）;

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

0= Writing 0 is invalid.

18.5.8 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, all must be 0.

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

1= Reverse output;

0= Normal output.

18.5.9 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, all must be 0.

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

1= Automatic load mode;

0= One-shot mode.

18.5.10 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/W	R/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 PWMnCNTE: PWM channel n is counter enable control (n=0-5);

1= PWMn counter start (PWMn begin output);

0= PWMn counter stop (The software writes 0 to stop the counter and clear the counter value).（This bit is cleared to 0 by hardware when the brake is triggered; And it is cleared to 0 by hardware when the single-shot mode is completed）

18.5.11 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, all must be 0.

Bit5~Bit0 PWMnCNTCLR: PWM channel n is counter clear 0 control(n=0-5)（Hardware clear 0 automatically）;

1= PWMn counter clear 0;

0= Writing 0 is invalid.

18.5.12 PWM Period Data Low 8-bit Register 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>: PWM channel n is period data low 8-bit register.

18.5.13 PWM Period Data High 8-bit Register 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>: PWM channel n is period data high 8-bit register.

18.5.14 PWM Compare Data Low 8-bit Register 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>: PWM channel n is compare date(duty cycle data) low 8-bit register.

18.5.15 PWM Compare Data High 8-bit Register 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>: PWM channel n is compare date(duty cycle data) high 8-bit register.

18.5.16 PWM Compare Data Low 8-bit Registers Down 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>: PWM channel n is compare data low 8-bit registers down (duty cycle data in asymmetric counting).

18.5.17 PWM Compare Data High 8-bit Registers Down 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>: PWM channel n is compare data high 8-bit registers down (duty cycle data in asymmetric counting).

18.5.18 PWM Dead Zone Enable 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, all must be 0.

Bit2 PWM45DTE: PWM4/5 channel dead zone delay enable;

1= Enable;

0= Disable.

Bit1 PWM23DTE: PWM2/3 channel dead zone delay enable;

1= Enable;

0= Disable.

Bit0 PWM01DTE: PWM0/1 channel dead zone delay enable;

1= Enable;

0= Disable.

18.5.19 PWM0/1 Dead Zone Delay Data Register PWM01DT

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

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

18.5.20 PWM2/3 Dead Zone Delay Data Register PWM23DT

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

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

18.5.21 PWM4/5 Dead Zone Delay Data 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 zone delay data register.

18.5.22 PWM Mask 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 is mask control enable (n=0-5);

1= PWMn channel enable mask data output;

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

18.5.23 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 (n=0-5);

1= PWMn channel output is high;

0= PWMn channel output is low;

18.5.24 PWM Brake Control Register PWMFBKC

F166H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMFBKC	PWMFBIE	PWMFBF	PWM5FBCCE	PWMFBKSW	PWMFB1ES	PWMFB0ES	PWMFB1EN	PWMFB0EN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 PWMFBIE: PWM's brake interrupt mask;
1= Enable interrupt;
0= Disable interrupt.

Bit6 PWMFBF: PWM's Brake flag (writing 0 is clear);
1= A brake operation is generated (PWM output brake data register value);
0= No braking operation occurred.

Bit5 PWMFBCCE: Whether to clear all channel counter selection bit when PWM braking;
1= The value of the counter will not be cleared when braking;
0= Clear the channel counter value when braking.

Bit4 PWMFBKSW: PWM software brake signal start;
1= PWM generates software brake signal;
0= Prohibit.

Bit3 PWMFB1ES: PWM external hardware brake channel 1 (FB1) trigger level select;
1= high level;
0= low level.

Bit2 PWMFB0ES: PWM external hardware brake channel 1 (FB0) trigger level select;
1= high level;;
0= low level.

Bit1 PWMFB1EN: PWM external hardware brake channel 1 (FB1) enable;
1= Enable;
0= Disable.

Bit0 PWMFB0EN: PWM external hardware brake channel 1 (FB0) enable;
1= Enable;
0= Disable.

18.5.25 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 (n=0-5);
1= PWMn channel outputs high after braking operation;
0= PWMn channel outputs low after braking operation.

18.6 PWM Interrupt

Enhanced PWM has a total of 25 interrupt flags, including 6 period interrupt flags, 6 zero-point interrupt flags, 6 upward comparison interrupt flags, 6 downward comparison interrupt flags, 1 brake interrupt flag. And the generation of interrupt flag bits, 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 enable the global interrupt enable bit (EA=1) and the PWM total interrupt enable bit PWMIE to successfully configure the PWM interrupt function. All PWM interrupts share an 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 enables and priority of the enhanced PWM can be set by the following relevant register bits.

18.6.1 Interruption Mask Register EIE2

0xAA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIE2	SPIIE	I^2CIE	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;
1= Enable SPI interrupt;
0= Disable SPI interrupt;

Bit6 I^2CIE: I^2C interrupt enable;
1= Enable I^2C interrupt;
0= Disable I^2C interrupt;

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

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

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

Bit2 -- Reserved, must to be 0.

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

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

18.6.2 Interrupt Priority Control Register EIP2

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

Bit7 PSPI: SPI interrupt priority control;
1= Set to high level interrupt;
0= Set to low level interrupt.

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

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

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

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

Bit2 -- Reserved, must to be 0.

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

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

18.6.3 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(n=0-5);
1= Enable interrupt;
0= Disable interrupt.

18.6.4 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 (n=0-5);

1= Enable interrupt;

0= Disable interrupt.

18.6.5 PWM Up Compare Interrupt Mask 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 up compare interrupt mask (n=0-5);

1= Enable interrupt;

0= Disable interrupt.

18.6.6 PWM Down Compare Interrupt Mask 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 down compare interrupt mask (n=0-5);

1= Enable interrupt;

0= Disable interrupt.

18.6.7 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 period interrupt flag(n=0-5);

1= Generate an interruption (software clear);

0= No interruption.

18.6.8 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 (n=0-5);

1= Generate an interruption (software clear);

0= No interruption.

18.6.9 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 up compare interrupt flag bit (n=0-5);

1= Generate an interruption (software clear);

0= No interruption.

18.6.10 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 down compare interrupt flag (n=0-5);

1= Generate an interruption (software clear);

0= No interruption.

19. Hardware LCD Driver

19.1 Overview

Hardware LCD driver include a controller, a duty cycle generator and COM and SEG output ports.

Hardware LCD driver support two modes: traditional resistance and fast charging，the bias resistance can be selected from 60KΩ, 225KΩ, and 900KΩ.Fast charging mode is a new way of design，when selecting 225KΩ, 900KΩ resistance mode, 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 ensure the brightness of the 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.

19.2 Characteristic

The LCD driver supports the following features:

- Maximum supported LCD channel: 8COM x 16SEG、6COM x 18SEG、5COM x 19SEG、4COM x 20SEG.
- Support contrast adjustment.
- Bias voltage optional: 1/2、1/3、1/4.
- Duty cycle optional: 1/4、1/5、1/6、1/8.
- Three kinds of clock sources are optional: Fsys、LSI、LSE.
- Support two modes: traditional resistance and fast charging.
- Fast charging time is optional.
- In LCD mode, LSI enable independent.
- Supports driving LCD in sleep mode.

19.3 Related Register

19.3.1 LCD Control Register LCDCON0

F680H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCDCON0	LCDEN	--	LCDDM1	LCDDM0	--	--	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;
1= LCD Enable;
0= LCD Disable.
Bit6 -- Reserved, must be 0.
Bit5~Bit4 LCDDM<1:0>: LCD display mode;
1x= Normal output;
01= SEG all open;
00= SEG all off.
Bit3~Bit2 -- Reserved, must be 0.
Bit1~Bit0 DUTY<1:0>: LCD duty cycle selection;
11= 1/4DUTY;
10= 1/5DUTY;
01= 1/6DUTY;
00= 1/8DUTY.

COM Selection Description Table

DUTY	ICOM0	ICOM1	ICOM2	ICOM3	ICOM4	ICOM5	ICOM6	ICOM7	Effective SEG port
11	LCD_C0	LCD_C1	LCD_C2	LCD_C3	-	-	-	-	LCD_S0-LCD_S19
10	LCD_C0	LCD_C1	LCD_C2	LCD_C3	LCD_C4	-	-	-	LCD_S1-LCD_S19
01	LCD_C0	LCD_C1	LCD_C2	LCD_C3	LCD_C4	LCD_C5	-	-	LCD_S2-LCD_S19
00	LCD_C0	LCD_C1	LCD_C2	LCD_C3	LCD_C4	LCD_C5	LCD_C6	LCD_C7	LCD_S4-LCD_S19

Note: In the above table, ICOM0-ICOM7 are the internal COM drive output signals of the LCD.

LCD_C0-LCD_C7, LCD_S0-LCD_S19 are the ports to which the internal driving signals of the LCD are finally mapped.

19.3.2 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 0.

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, no choice

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.

19.3.3 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: LSI clock enable bit in LCD mode;

1= In LCD mode, LSI is enabled (LCDCON0.LCDEN needs to be 1);

0= In LCD mode, LSI is disabled

Bit6 -- Reserved, 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.

19.3.4 LCD Control Register LCDCON3

F683H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCDCON3	--	--	LCDRM1	LCDRM0	--	FCMODE	FCCTL1	FCCTL0
R/W	R/W	R/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 LCD_RM<1:0>: LCD divider resistor 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 resistive mode.

Bit1~Bit0 FCCTLM<1:0>: Fast charging mode time control;

00= 1/8 COM cycle;

01= 1/16 COM cycle;

10= 1/32 COM cycle;

11= 1/64 COM cycle.

19.3.5 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;

1= Enable;

0= Disable.

19.3.6 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;

1= Enable;

0= Disable.

19.3.7 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;

1= Enable;

0= Disable.

19.3.8 SEG Port Enable Control Register LCDSEGEN2

F687H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LCDSEGEN2	--	--	--	--	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~Bit4 -- Reserved, all must be 0

Bit4~Bit0 SEGEN<19:16>: LCD_S19-LCD_S16 port enable control;

1= Enable;

0= Disable.

19.3.9 SEG Data Register 6(n=0-19)

	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~LCDSEG19 address: F650H~F663H.

Bit7~Bit0 ICOM<7:0>: LCD_Sn Port data output;

1= High level;

0= Low level.

19.4 COM -SEG data-sheet

The hardware LCD driver mode is configured to different DUTY corresponding to the following data sheet.

19.4.1 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

19.4.2 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

19.4.3 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

19.4.4 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

20. Hardware LED Driver

20.1 Overview

The chip integrates a hardware LED display drive circuit, which can facilitate users to realize LED display drive.

20.2 Characteristic

The LED driver supports the following features:

- 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.
- Support up to 8 COM ports and 20 SEG ports.
- Two drive modes of common cathode and common anode for COM port are optional.
- COM port current 50mA, 150mA two gears optional（VOL=1.5V@VDD=5V）.
- SEG port current 16 levels are optional, the maximum current can reach 40mA（VOH=3.5V@VDD=5V）.

20.3 Related Register

20.3.1 LED Control Register LEDCON

F765H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDCON	LED_EN	DUTY1	DUTY0	CC_CA	--	--	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;
1= LED Enable;
0= LED Disable.

Bit6~Bit5 DUTY<1:0>: LED duty cycle selection;
11= 1/4DUTY;
10= 1/5DUTY;
01= 1/6DUTY;
00= 1/8DUTY;

Bit4 CC_CA: LED drive mode selection;
1= common anode drive mode;
0= common cathode drive mode.

Bit3~Bit2 -- Reserved, all must be 0

Bit1~Bit0 CLKSEL: LED clock source FLED selection;
11= LSI;
10= LSI;
01= LSE;
00= Fsys（System clock）.

COM Selection Description Table

DUTY	ICOM0	ICOM1	ICOM2	ICOM3	ICOM4	ICOM5	ICOM6	ICOM7	Effective SEG port
11	LED_C0	LED_C1	LED_C2	LED_C3	-	-	-	-	LED_S0-LED_S19
10	LED_C0	LED_C1	LED_C2	LED_C3	LED_C4	-	-	-	LED_S1-LED_S19
01	LED_C0	LED_C1	LED_C2	LED_C3	LED_C4	LED_C5	-	-	LED_S2-LED_S19
00	LED_C0	LED_C1	LED_C2	LED_C3	LED_C4	LED_C5	LED_C6	LED_C7	LED_S4-LED_S19

Note: In the above table, ICOM0-ICOM7 are the internal COM drive output signals of the LED.

LED_C0-LED_C7, LED_S0-LED_S28 are the ports to which the internal driving signals of the LED are finally mapped.

20.3.2 LED Clock Prescaler Data Register Low 8-bit 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.

20.3.3 LED Clock Prescaler Data Register High 8-bit 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.

LED driver clock frequency: $F_{LED\_CLK} = F_{LED}$ / （CLK<15:0>+1）.

20.3.4 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	0	0	0	0	0	0	0

Bit7~Bit0 COMT<7:0>: COM port effective time setting

Note: Prohibit setting to 0x00 and 0xFF;

COM time = (COMT<7:0> + 1) * T_{LED_CLK} .

20.3.5 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;
1= Enable;
0= Disable.

20.3.6 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;
1= Enable;
0= Disable.

20.3.7 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;
1= Enable;
0= Disable.

20.3.8 SEG Port Enable Control Register LEDSEGEN2

F763H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDSEGEN2	--	--	--	--	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~Bit4 -- Reserved, all must be 0
Bit3~Bit0 SEGEN<19:16>: LED_S19-LED_S16 port enable control;
1= Enable;
0= Disable.

20.3.9 COM0 Corresponds To SEG Data Register LEDC0DATAn（n=0-2）

	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

LEDC0DATA0 address: F740H; LEDC0DATA1 address: F741H; LEDC0DATA2 address: F742H.

When n = 0/1:

Bit7~Bit0 SEG<8n+7:8n>: When COM0 port is valid, SEG[8n+7]-SEG[8n] port data output;

1= High level;

0= Low level.

When n = 2:

Bit7~Bit4 -- Reserved, all must be 0

Bit3~Bit0 SEG<8n+7:8n>: When COM0 port is valid, SEG[8n+7]-SEG[8n] port data output;

1= High level;

0= Low level.

20.3.10 COM1 Corresponds To SEG Data Register LEDC1DATAn（n=0-2）

	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

LEDC1DATA0 address: F744H; LEDC1DATA1 address: F745H; LEDC1DATA2 address: F746H.

When n = 0/1:

Bit7~Bit0 SEG<8n+7:8n>: When COM1 port is valid, SEG[8n+7]-SEG[8n] port data output;

1= High level;

0= Low level.

When n = 2:

Bit7~Bit4 -- Reserved, all must be 0

Bit3~Bit0 SEG<8n+7:8n>: When COM1 port is valid, SEG[8n+7]-SEG[8n] port data output;

1= High level;

0= Low level.

20.3.11 COM2 Corresponds To SEG Data Register LEDC2DATAn（n=0-2）

	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

LEDC2DATA0 address: F748H; LEDC2DATA1 address: F749H; LEDC2DATA2 address: F74AH.

When n = 0/1:

Bit7~Bit0 SEG<8n+7:8n>: When COM2 port is valid, SEG[8n+7]-SEG[8n] port data output;

1= High level;

0= Low level.

When n = 2:

Bit7~Bit4 -- Reserved, all must be 0

Bit3~Bit0 SEG<8n+7:8n>: When COM2 port is valid, SEG[8n+7]-SEG[8n] port data output;

1= High level;

0= Low level.

20.3.12 COM3 Corresponds To SEG Data Register LEDC3DATAn（n=0-2）

	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

LEDC3DATA0 address: F74CH; LEDC3DATA1 address: F74DH; LEDC3DATA2 address: F74EH.

When n = 0/1:

Bit7~Bit0 SEG<8n+7:8n>: When COM3 port is valid, SEG[8n+7]-SEG[8n] port data output;

1= High level;

0= Low level.

When n = 2:

Bit7~Bit4 -- Reserved, all must be 0

Bit3~Bit0 SEG<8n+7:8n>: When COM3 port is valid, SEG[8n+7]-SEG[8n] port data output;

1= High level;

0= Low level.

20.3.13 COM4 Corresponds To SEG Data Register LEDC4DATAn（n=0-2）

	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

LEDC4DATA0 address: F750H; LEDC4DATA1 address: F751H; LEDC4DATA2 address: F752H.

When n = 0/1:

Bit7~Bit0 SEG<8n+7:8n>: When COM4 port is valid, SEG[8n+7]-SEG[8n] port data output;

1= High level;

0= Low level.

When n = 2:

Bit7~Bit4 -- Reserved, all must be 0

Bit3~Bit0 SEG<8n+7:8n>: When COM4 port is valid, SEG[8n+7]-SEG[8n] port data output;

1= High level;

0= Low level.

20.3.14 COM5 Corresponds To SEG Data Register LEDC5DATAn（n=0-2）

	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

LEDC5DATA0 address: F754H; LEDC5DATA1 address: F755H; LEDC5DATA2 address: F756H.

When n = 0/1:

Bit7~Bit0 SEG<8n+7:8n>: When COM5 port is valid, SEG[8n+7]-SEG[8n] port data output;

1= High level;

0= Low level.

When n = 2:

Bit7~Bit4 -- Reserved, all must be 0

Bit3~Bit0 SEG<8n+7:8n>: When COM5 port is valid, SEG[8n+7]-SEG[8n] port data output;

1= High level;

0= Low level.

20.3.15 COM6 Corresponds To SEG Data Register LEDC6DATAn（n=0-2）

	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

LEDC6DATA0 address: F758H; LEDC6DATA1 address: F759H; LEDC6DATA2 address: F75AH.

When n = 0/1:

Bit7~Bit0 SEG<8n+7:8n>: When COM6 port is valid, SEG[8n+7]-SEG[8n] port data output;

1= High level;

0= Low level.

When n = 2:

Bit7~Bit4 -- Reserved, all must be 0

Bit3~Bit0 SEG<8n+7:8n>: When COM6 port is valid, SEG[8n+7]-SEG[8n] port data output;

1= High level;

0= Low level.

20.3.16 COM7 Corresponds To SEG Data Register LEDC7DATAn（n=0-2）

	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

LEDC7DATA0 address: F75CH; LEDC7DATA1 address: F75DH; LEDC7DATA2 address: F75EH.

When n = 0/1:

Bit7~Bit0 SEG<8n+7:8n>: When COM7 port is valid, SEG[8n+7]-SEG[8n] port data output;

1= High level;

0= Low level.

When n = 2:

Bit7~Bit4 -- Reserved, all must be 0

Bit3~Bit0 SEG<8n+7:8n>: When COM7 port is valid, SEG[8n+7]-SEG[8n] port data output;

1= High level;

0= Low level.

20.3.17 SEG Port P04-P07 Drive Current Control Register LEDSDRP0H

F711H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDSDRP0H	--	--	--	--	DRC3	DRC2	DRC1	DRC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	1	1	1	1

Bit7~Bit4 -- Reserved, all must be 0

Bit3~Bit0 DRC<3:0>: Source current drive selection control (control P04/P05/P06/P07 four SEG ports)

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.

20.3.18 SEG Port P10-P13 Drive Current Control Register LEDSDRP1L

F712H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDSDRP1L	--	--	--	--	DRC3	DRC2	DRC1	DRC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	1	1	1	1

Bit7~Bit4 -- Reserved, all must be 0

Bit3~Bit0 DRC<3:0>: Source current drive selection control (control P10/P11/P12/P13 four SEG ports);

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.

20.3.19 SEG Port P14-P17 Drive Current Control Register LEDSDRP1H

F713H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDSDRP1H	--	--	--	--	DRC3	DRC2	DRC1	DRC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	1	1	1	1

Bit7~Bit4 -- Reserved, all must be 0

Bit3~Bit0 DRC<3:0>: Source current drive selection control (control P14/P15/P16/P17 four SEG ports);

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.

20.3.20 SEG Port P20-P23 Drive Current Control Register LEDSDRP2L

F714H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDSDRP2L	--	--	--	--	DRC3	DRC2	DRC1	DRC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	1	1	1	1

Bit7~Bit4 -- Reserved, all must be 0

Bit3~Bit0 DRC<3:0>: Source current drive selection control (control P20/P21/P22/P23 four SEG ports);

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.

20.3.21 SEG Port P24-P27 Drive Current Control Register LEDSDRP2H

F715H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDSDRP2H	--	--	--	--	DRC3	DRC2	DRC1	DRC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	1	1	1	1

Bit7~Bit4 -- Reserved, all must be 0

Bit3~Bit0 DRC<3:0>: Source current drive selection control (control P24/P25/P26/P27 four SEG ports);

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.

20.3.22 COM Port Current Sink Selection Register P0DR

F00CH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0DR	P0DR7	P0DR6	P0DR5	P0DR4	P0DR3	P0DR2	P0DR1	P0DR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit7 P0DR7: P07 drive current selection;
1= 150mA;
0= 50mA.

Bit6 P0DR6: P06 drive current selection;
1= 150mA;
0= 50mA.

Bit5 P0DR5: P05 drive current selection;
1= 150mA;
0= 50mA.

Bit4 P0DR4: P04 drive current selection;
1= 150mA;
0= 50mA.

Bit3 P0DR3: P03 drive current selection;
1= 150mA;
0= 50mA.

Bit2 P0DR2: P02 drive current selection;
1= 150mA;
0= 50mA.

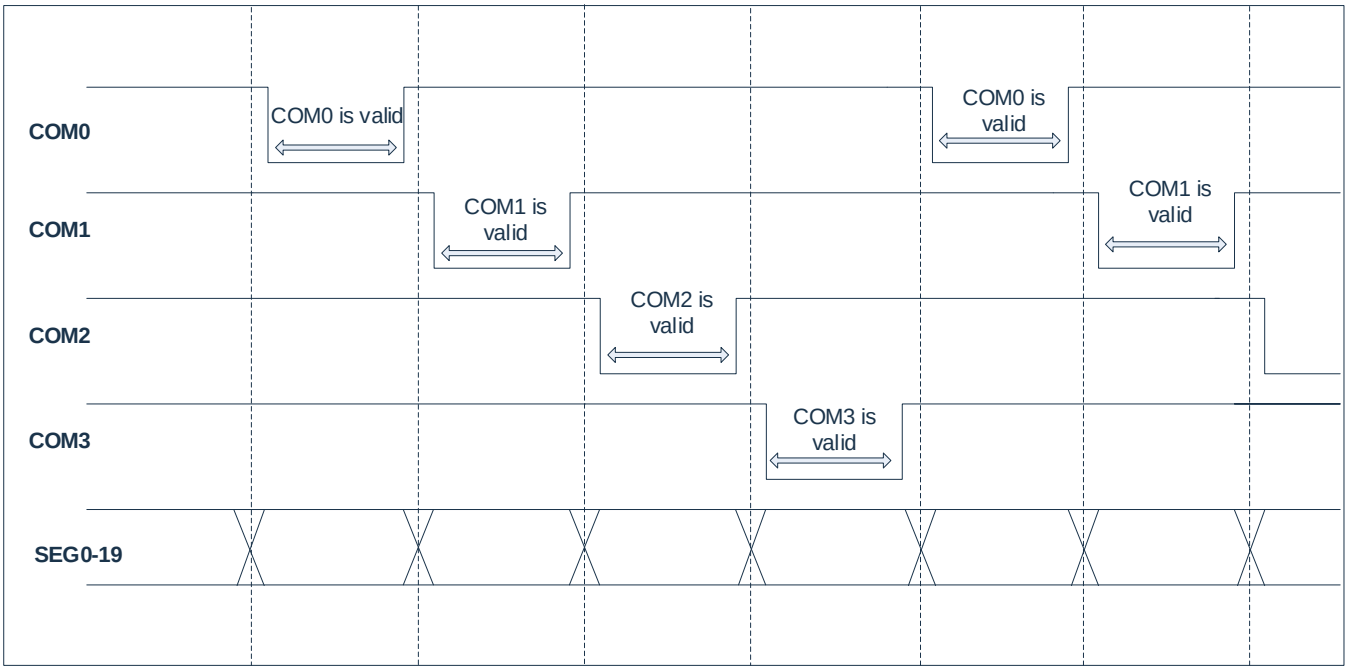
Bit1 P0DR1: P01 drive current selection;
1= 150mA;
0= 50mA.

Bit0 P0DR0: P00 drive current selection;
1= 150mA;
0= 50mA.

20.4 LED Driver Output Waveform

According to the relevant configuration registers of the LED driver, the corresponding LED driver output waveform can be set.

LED configuration 1/4DUTY, common cathode drive mode, the waveform is shown in the figure below:



21. SPI

21.1 Overview

The SPI is a fully configurable SPI master/slave device, which allows user to configure polarity and phase of serial clock signal SCK. A serial clock line (SCK) synchronizes shifting and sampling of the information on the two independent serial data lines. SPI data are simultaneously send and receive. The SPI allows the microcontroller to communicate with serial peripheral devices，It is also capable of inter-processor communications in a multi-master system，the SPI is a technology independent design that can be implemented in a variety of process technologies.

The SPI system is flexible enough to interface directly with numerous standard product peripherals from several manufacturers. Clock control logic allows a selection of clock polarity and clock phase to accommodate most available synchronous serial peripheral devices. The system can be configured as a master or a slave device, When the SPI is configured as a master, software selects one of four different bit rates for the serial clock, data rates as high as System clock divided by four (CLK/4).

The SPI slave chip is selected to address the SPI slave device to exchange serially shifted 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:

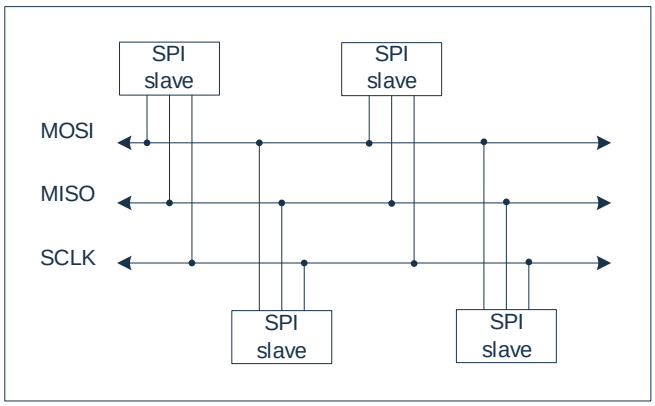
- Full duplex synchronous serial data transfer;
- Master mode and Slave mode supported;
- Multi-master system supported;
- System error detection;
- Interrupt generation;
- Supports speeds up to 1/4 of system clock（F_{SYS}≤24MHZ）
- Bit rates generated 1/4, 1/8, 1/ 16, 1/32, 1/64, 1/128, 1/256, 1/512 of system clock;
- Four transfer formats supported;
- Simple interface allows easy connection to microcontrollers.

21.2 SPI Port Configuration

To use the SPI function, you need to configure the related port as SPI channels, and select the corresponding port input through the communication input port register. For example, configure P00, P01, P02, P03 as SPI communication ports. The configuration code is as follows:

```
PS_SCLK = 0x00;   //Select P00 as the SCLK channel of SPI
PS_MOSI = 0x01;   //Select P01 as the MOSI channel of SPI
PS_MISO = 0x02;   //Select P02 as the MISO channel of SPI
PS_NSS = 0x03;    //Select P03 as the NSS channel of SPI
P00CFG = 0x03;    //P00 is reused as SCLK function
P01CFG = 0x03;    //P01 is reused as MOSI function
P02CFG = 0x03;    //P02 is reused as MISO function
P03CFG = 0x03;    //P03 is reused as NSS function
```

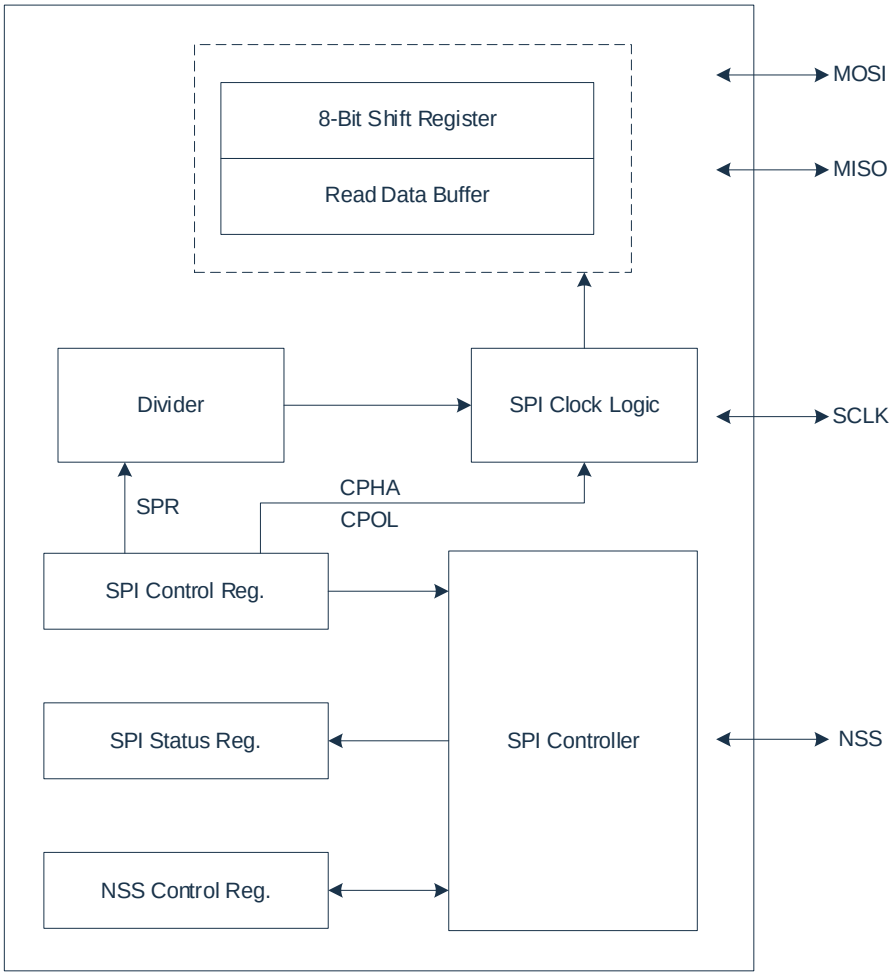
Configured as SCLK, MOSI, MISO and NSS ports, the pull-up resistor and open-drain output switch are forcibly closed. The schematic diagram of the multi-machine SPI communication structure is shown in the figure below:



21.3 SPI Hardware Description

When an SPI transfer occurs, an 8-bit character is shifted out on data pin while a different 8-bit character is simultaneously shifted in a second data pin. An 8-bit shift register in the master and another 8- bit shift register in the slave are connected as a circular 16-bit shift register. When a transfer occurs, this distributed shift register is shifted 8-bit positions; thus, the characters in the master and slave are effectively exchanged.

The central element in the SPI system is the block containing the shift register and the read data buffer. The system is single buffered in the transmit direction and double buffered in the receive direction. This fact means new data for transmission cannot be written to the shifter until the previous transaction is complete; however, received data is transferred into a parallel read data buffer so the shifter is free to accept a second serial character. As long as the first character is read out of the read data buffer before the next serial character is ready to be transferred, no overrun condition will occur. The SPI control block diagram is shown in the figure below:



The pins associated with the SPI: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 the SPI is configured as a slave the SI pin is the slave input data line, and the SO is the slave output data line.

When the SPI is configured as a master, the MI pin is the master input data line, and the MO is the master output data line.

21.4 SPI Related Register

21.4.1 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 mode enable;

1= Enable;

0= Disable.

Bit5 SPR2: SPI clock rate select :[2].

Bit4 MSTR: SPI mode select;

1= master mode;

0= slave mode.

Bit3 CPOL: SPI clock polarity select;

1= SCLK is high in idle state;;

0= SCLK is low in idle state.

Bit2 CPHA: SPI clock phase select.

Bit1~Bit0 SPR<1:0>: SPI clock rate select :[1:0]

(See the table below for frequency control details)

SPR2-SPR0 control SPI clock division

SPR2	SPR1	SPR0	System clock division
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

21.4.2 SPI Data Register SPDR

0xEE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPDR	SPIDATA7	SPIDATA6	SPIDATA5	SPIDATA4	SPIDATA3	SPIDATA2	SPIDATA1	SPIDATA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit7~Bit0 SPIDATA<7:0>: Data send or received by the SPI.

write: Write the data to be send(Send order from high to low).

read: Received data.

21.4.3 Slave Select Control Register SSCR

Slave selection control register SSCR can be read or written at any time, it is used to configure which slave select output should be driven while SPI master transfer. Contents of SSCR register is automatically assigned on NSS pin when SPI master transmission starts.

0xEF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SSCR	--	--	--	--	NSSO3	NSSO2	NSSO1	NSSO0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	1	1	1	1	1	1	1	1

Bit7~Bit4 -- Reserved, all must be one.

Bit3~Bit0 NSSOx: SPI slave device selection control bit (master chip select output NSS is NSSOx, x=0-3)

0= When SPI master transmission starts, NSSOx output 0.

1= When SPI master transmission starts, NSSOx output 1.

21.4.4 SPI State 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 complete flag, Read only;
1= SPI transfer to complete (Read SPSR first, then clear after reading/writing SPDR);
0= SPI not transferred.
Bit6 WCOL: SPI write collision error flag, Read only;
1= SPI Write collision error (Read SPSR first, then clear after reading/writing SPDR);
0= No write collision error.
Bit5~Bit1 -- Reserved, all must be 0.
Bit0 SSCEN: SPI master mode NSS Output control.
1= NSS is high in SPI idle state;
0= The contents of the NSS output register SSCR.

SPI status register (SPSR) contains flags indicating the completion of transfer or occurrence of system errors. All flags are set automatically when the corresponding event occur and cleared by software sequence. SPIF and WCOL are automatically cleared by reading SPSR followed by an access of the SPDR.

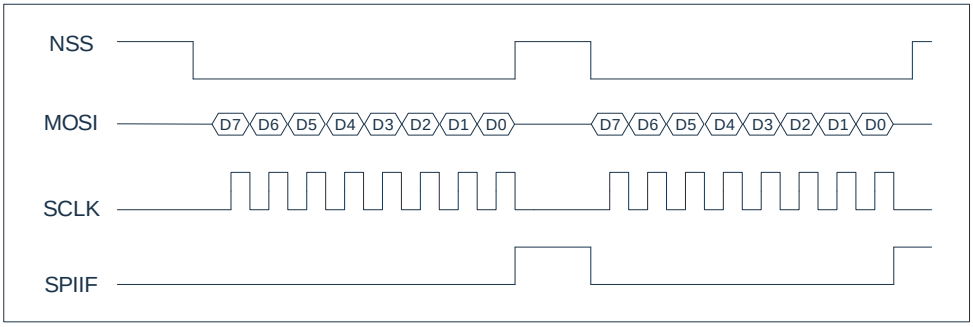
The SSCEN bit is the enable bit for automatic slave selection output. When SSCEN is set to 1, the NSS line outputs the contents of the SSCR register when the transmission is in progress, and the NSS is high when the transmission is idle. When the SSCEN bit is cleared, the NSS line always display the contents of the SSCR register.

21.5 SPI Master Model

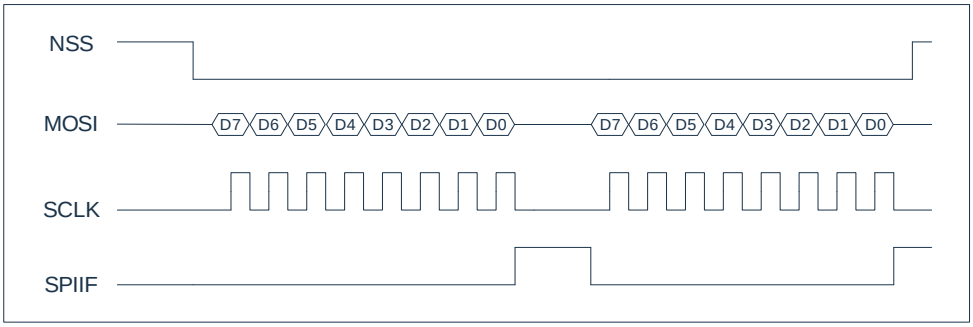
When the SPI is configured in master mode, the transfer is initiated by writing to the SPDR register. When a new byte is written into the SPDR register, the SPI starts to transfer. The serial clock SCLK is generated by SPI. In the master mode, SPI is enabled and SCLK is output.

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 shows 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 the slave selection line used as shown in the figure below:



When SSCEN=0, configure the SPI clock polarity CPOL=0, clock phase CPHA=0, and the slave selection line used as shown in the figure below:



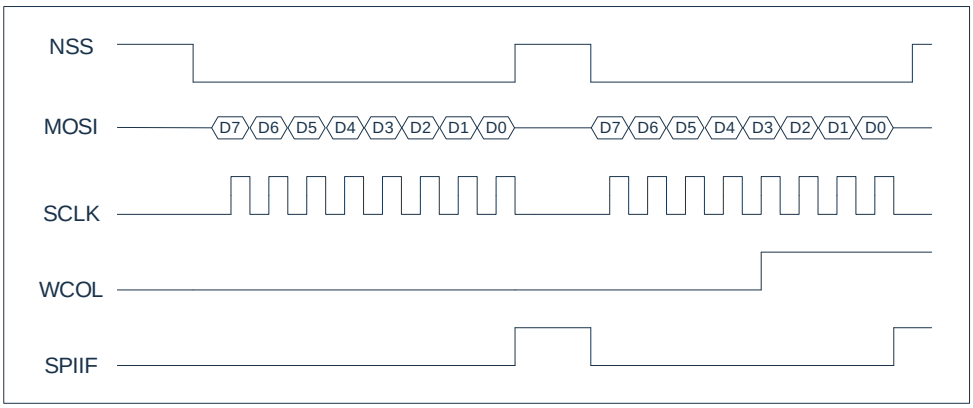
21.5.1 Write Collision Error

A write collision occurs if the SPI data register is written while a transfer is in progress. The transfer continues undisturbed, and the write data that caused the error is not written to the shifter. The write wollision is indicated by the WCOL flag in SPSR register.

The WCOL flag is set automatically by hardware, when the WCOL error condition occurs. To clear the WCOL bit, user should execute the following sequence:

- read contents of the SPSR register;
- perform access to the SPDR register (read or write)

In the SPI master control mode, configure the SPI clock polarity CPOL=0, the clock phase CPHA=0, the write conflict error is shown in the figure below:



The conditions for writing conflicts: during data transmission, when NSS is low, the first data starts transmitting to the 8th SCLK falling edge. If SPDR is written during this time, a write collision will occur and WCOL will be set.

Warning: When starting to send data, if NSS does not immediately go low after writing SPDR, needs to wait for up to one SPI clock before it starts low. After NSS is low, it needs to wait for a system clock to start sending the first data, and then enter the real data transmission state. Writing to SPDR again does not create a write collision while writing SPDR to enter the true data transfer state. However, this operation updates the data to be send. If there are multiple writes to SPDR, the data send will be the last value written to SPDR.

Since the SPI has only one transmit buffer, it is recommended to judge whether the last data has been send before writing the SPDR, and then confirm that the SPDR register is written after the transmission is completed to prevent a write conflict.

21.6 SPI Slave Mode

When configured as SPI slave the transfer is initiated by external SPI master module by assertion of the SPI slave select input, and generation of the SCK serial clock.

Before the start of the transfer, it is necessary to determine which SPI slave will be used to exchange data. The NSS is asserted (cleared = 0), the clock signal connected to the SCLK line will cause the SPI slave to shift into receiver shift register contents of the MOSI line, and drives the MISO line with contents of the transmitter shift register. When all 8 bits are shifted in/out the SPI generates the interrupt request by setting the IRQ output. The contents of the shift register drive the MISO line.

In SPI slave mode only one transfer error is possible

- Write collision error.

21.6.1 Addressed Error

In slave mode only the write collision error can be detected by the SPI.

The write collision error occurs when the SPDR register write is performed while the SPI transfer is in progress.

In slave mode when the CPHA is cleared, the write collision error may occur as long as the NSS slave select line is driven low, even if all bits are already transferred. This is because there is not clearly specified the transfer beginning, and NSS driven low after full byte transfer may indicate beginning of the next byte transfer.

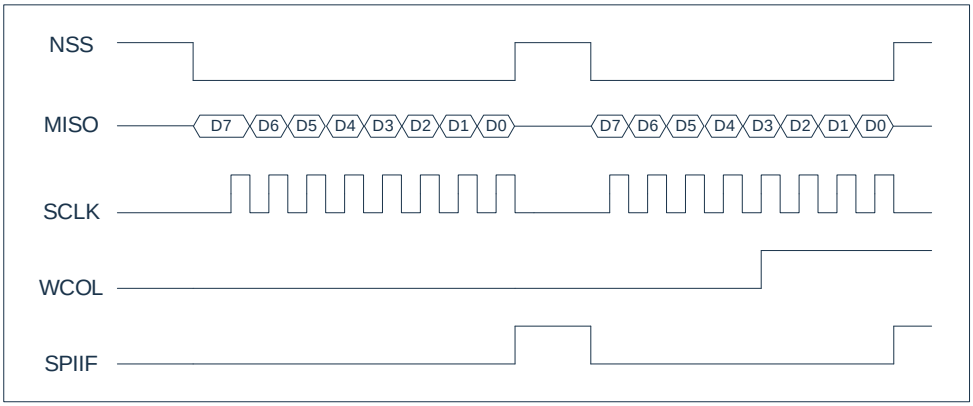
21.6.2 Write Collision Error

A write collision occurs if the SPI data register is written while a transfer is in progress. The transfer continues undisturbed, and the write data that caused the error is not written to the shifter. The write collision is indicated by the WCOL flag in SPSR register.

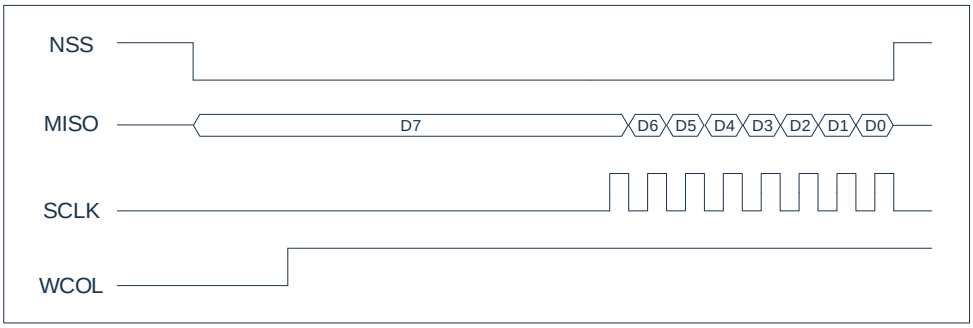
The WCOL flag is set automatically by hardware, when the WCOL error condition occurs. To clear the WCLO bit, user should execute the following sequence:

- read contents of the SPSR register;
- perform access to the SPDR register (read or write).

Write collision error during transmission in SPI slave mode as shown in the figure below:



In case CPHA is cleared, WCOL generation can also be caused by writing to SPDR register when any NSS line is cleared. At this time, the SPI master can also complete without generating the serial clock SCLK. 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 collision error as shown in the figure below:



After the SPDR is written in the slave mode, the master controlled NSS does not go low immediately. When NSS is low, it needs to wait for the second edge of SCLK to start to enter the real data transmission state.

Writing to SPDR does not create a write collision during the SPDR to start sending the first data. However, this operation updates the data to be sent. If there are multiple writes to SPDR, the data sent will be the last value written to SPDR.

During the start of the first data transmission to the second edge of SCLK, writing to SPDR again does not create a write collision and does not update the data being transmitted. That is, the operation of writing the SPDR is ignored.

Since the SPI has only one transmit buffer, it is recommended to judge whether the last data is sent before writing SPDR, and then confirm that the SPDR register is written after the transmission is completed to prevent write collision.

21.7 SPI Clock Control Logic

21.7.1 SPI Clock Phase And Polarity Control

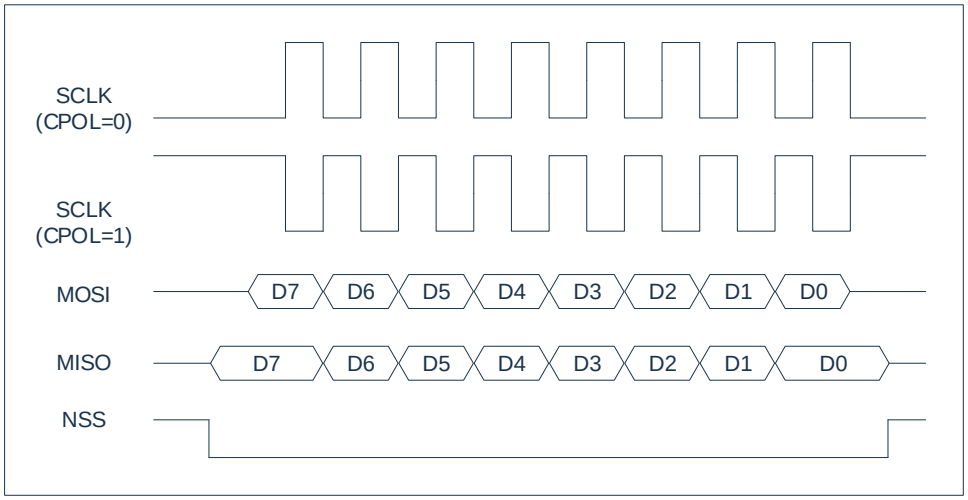
Software can select any of four combinations of serial clock (SCK) phase and polarity using two bits in the SPI control register (SPCR). The clock polarity is specified by the CPOL control bit, When the transmission is idle, the CPOL control bit selection high or low has no significant impact on the transmission format. The clock phase (CPHA) control bit selects one of two fundamentally different transfer formats. The clock phase and polarity should be identical for the master SPI device and the communicating slave device. In some cases, the phase and polarity are changed between transfers to allow a master device to communicate with peripheral slaves having different requirements. The flexibility of the SPI system on the SPI allows direct interface to almost any existing synchronous serial peripheral.

21.7.2 SPI Transport Format

During an SPI transfer, data is simultaneously transmitted (shifted out serially) and received (shifted in serially). A serial clock line synchronizes shifting and sampling of the information on the two serial data lines. A slave select line allows individual selection of a slave SPI device; slave devices that are not selected do not interfere with SPI bus activities. On a master SPI device, the slave select line can optionally be used to indicate a multiple-master bus contention.

21.7.3 CPHA=0 Transport Format

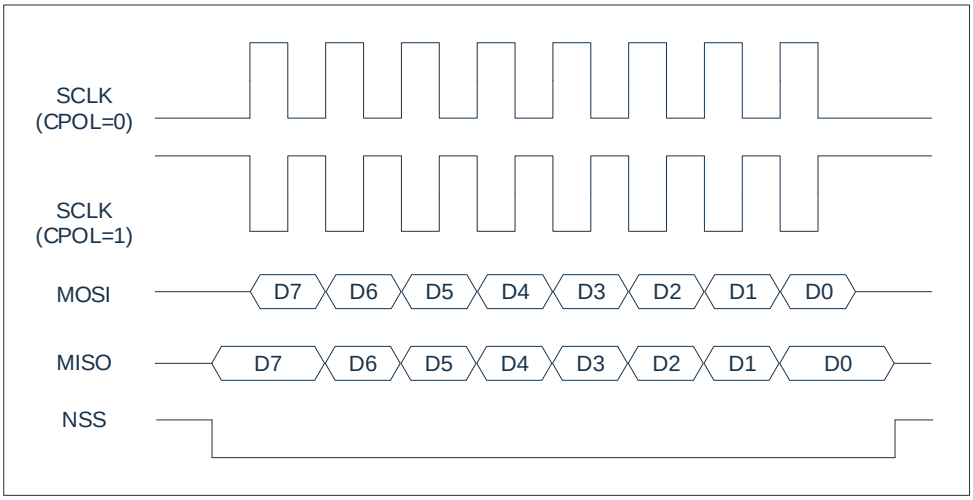
Figure below shows a timing diagram of an SPI transfer where CPHA is 0. Two waveforms are shown for SCLK: one for CPOL equals 0 and another for CPOL equals 1. The diagram may be interpreted as a master or slave timing diagram since the SCLK, master in/slave out (MISO), and master out/slave in (MOSI) pins are directly connected between the master and the slave. The MISO signal is the output from the slave, and the MOSI signal is the output from the master. The NSS line is the slave select input to the slave; the NSS pin of the master is not shown but is assumed to be inactive. The NSS pin of the master must be high. This timing diagram functionally depicts how a transfer takes place; it should not be used as a replacement for data-sheet parametric information.



When CPHA = 0, the NSS line must be deasserted and reasserted between each successive serial byte. Also, if the slave writes data to the SPI data register (SPDR) while SS is active low, a write-collision error result. When CPHA = 1, the NSS line may remain active low between successive transfers (can be tied low at all times). This format is sometimes preferred in systems having a single fixed master and a single slave driving the MISO data line.

21.7.4 CPHA=1 Transport Format

Figure below is a timing diagram of an SPI transfer where CPHA = 1. Two waveforms are shown for SCLK: one for CPOL= 0 and another for CPOL = 1. The diagram may be interpreted as a master or slave timing diagram since the SCLK, MISO, and MOSI pins are directly connected between the master and the slave. The MISO signal is the output from the slave, and the MOSI signal is the output from the master. The NSS line is the slave select input to the slave; then NSS pin of the master is not shown but is assumed to be inactive. The NSS pin of the master must be high or must be reconfigured as a general-purpose output not affecting the SPI.



21.8 SPI Data Transmission

21.8.1 SPI Transfer Start

All SPI transfers are started and controlled by a master SPI device. As a slave, the transfer to begin with the first SCLK edge or the falling edge of NSS, depending on the CPHA format selected. When CPHA = 0, the falling edge of NSS indicates the beginning of a transfer. When CPHA = 1, the first edge on the SCLK indicates the start of the transfer. In either CPHA format, a transfer can be aborted by taking the NSS line high, which causes the SPI slave logic and bit counters to be reset. The SCLK rate selected has no effect on slave operations since the clock from the master is controlling transfers.

When the SPI is configured as a master, transfers are started by a software write to the SPDR.

21.8.2 SPI Transfer End

An SPI transfer is technically complete when the SPIF flag is set, but, depending on the configuration of the SPI system, there may be additional tasks. Because the SPI bit rate does not affect timing of the ending period, only the fastest rate is considered in discussions of the ending period. When the SPI is configured as a master, SPIF is set at the end of the eighth SCK cycle. When CPHA equals 1, SCLK is inactive for the last half of the eighth SCLK cycle.

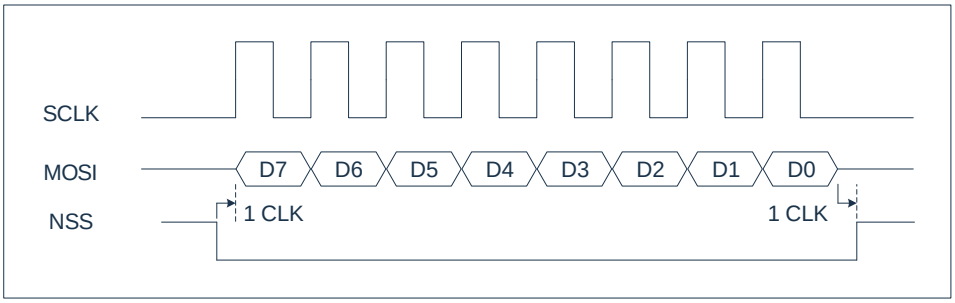
When the SPI is operating as a slave, the ending period is different because the SCLK line can be asynchronous to the MCU clocks of the slave and because the slave does not have access to as much information about SCLK cycles as the master. For example, when CPHA = 1, where the last SCLK edge occurs in the middle of the eighth SCLK cycle, the slave has no way of knowing when the end of the last SCLK cycle is. For these reasons, the slave considers the transfer complete after the last bit of serial data has been sampled, which corresponds to the middle of the eighth SCLK cycle.

The SPIF flag is set at the end of a transfer, but the slave is not permitted to write new data to the SPDR while the NSS line is still low.

21.9 SPI Timing Diagram

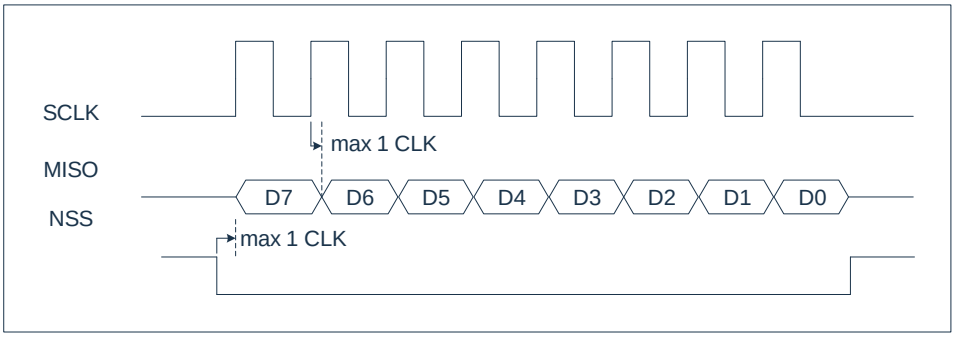
21.9.1 Master Mode Transmission

When the clock polarity of SPI is CPOL=0 and the clock phase CPHA=1, the system clock CLK after NSS is low in SPI master control mode, MOSI starts to output, and the data of MOSI is output on the rising edge of the SCLK clock. The timing diagram of the main control mode is shown in the figure below:



21.9.2 Slave Mode Transmission

When the SPI clock polarity CPOL=0 and the clock phase CPHA=1, the data on MISO starts to output after the falling edge of the NSS line. The maximum difference between the MISO data output and the falling edge of NSS is 1 system clock CLK. The timing diagram of the slave mode is shown in the figure below:



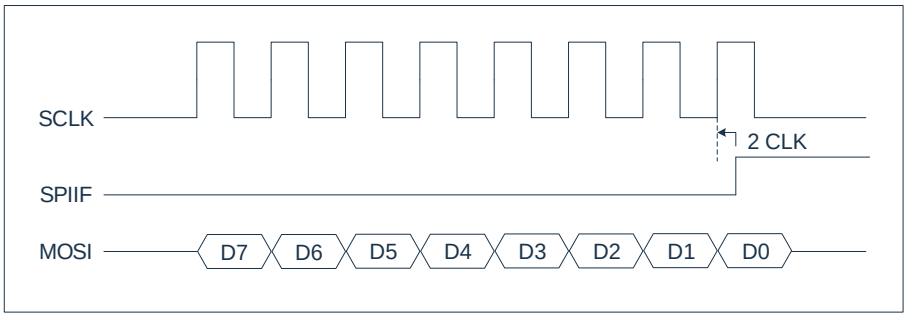
21.10 SPI Interrupt

SPI's interrupt number is 22, The interrupt vector is 0x00B3. If the SPI interrupt is to be enabled, the SPIIE must be set to 1, and set the total interrupt EA.

If the SPI related interrupt enable is turned on and the SPI total interrupt indication bit SPIIF = 1, the CPU will enter the interrupt service routine. SPIIF operation attributes are read-only and independent of the state of SPIIE.

After the SPI status register SPSR has the transfer completion flag SPISIF and write collision WCOL, the SPI total interrupt indication bit, SPIIF, will be set. SPIIF is automatically cleared only when all three flags are 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:



21.10.1 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;
1= Enable SPI interrupt;
0= Disable SPI interrupt;
Bit6 I2CIE: I^2C interrupt enable;
1= Enable I^2C interrupt;
0= Disable I^2C interrupt;
Bit5 WDTIE: WDT interrupt enable;
1= Enable WDT overflow interrupt;
0= Disable WDT overflow interrupt.
Bit4 ADCIE ADC interrupt enable;
1= Enable ADC interrupt;
0= Disable ADC interrupt.
Bit3 PWMIE: PWM global interrupt enable;
1= Enable PWM all interruptions;
0= Disable PWM all interruptions.
Bit2 -- Reserved, must to be 0.
Bit1 ET4: Timer4 interrupt enable;
1= Enable Timer4 interrupt;

0= Disable Timer4 interrupt.

Bit0 ET3: Timer3 interrupt enable;

1= Enable Timer3 interrupt;

0= Disable Timer3 interrupt.

21.10.2 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;

1= Set to high level interrupt;

0= Set to low level interrupt.

Bit6 PI2C: I^2C interrupt priority control;

1= Set to high level interrupt;

0= Set to low level interrupt.

Bit5 PWDT: WDT interrupt priority control;

1= Set to high level interrupt;

0= Set to low level interrupt.

Bit4 PADC: ADC interrupt priority control;

1= Set to high level interrupt;

0= Set to low level interrupt.

Bit3 PPWM: PWM interrupt priority control;

1= Set to high level interrupt;

0= Set to low level interrupt.

Bit2 -- Reserved, must to be 0.

Bit1 PT4: TIMER4 interrupt priority control;

1= Set to high level interrupt;

0= Set to low level interrupt.

Bit0 PT3: TIMER3 interrupt priority control;

1= Set to high level interrupt;

0= Set to low level interrupt.

21.10.3 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 total interrupt indicator, read only;

1= SPI produces an interrupt (this bit is automatically cleared after clearing the specific interrupt flag;

0= SPI did not cause an interruption.

Bit6 I2CIF: I^2C total interrupt indicator position, read only;

1= I^2C produces an interrupt (after clearing the specific interrupt flag bit, this bit is automatically cleared);

0= There was no interruption to the I^2C.

Bit5 -- Reserved, must to be 0.

Bit4 ADCIF: ADC interrupt flag;

1= ADC conversion is completed, software clearance is required;

0= ADC conversion not completed.

Bit3 PWMIF: PWM total interrupt indicator, read only;

1= PWM produces an interrupt;

0= PWM did not interrupt.

Bit2 -- Reserved, must to be 0.

Bit1 TF4: Timer4 counter overflow interrupt flag;

1= The Timer4 counter verflows and is automatically cleared by hardware when entering the interrupt service routine, or it can be cleared by software;

0= Timer4 counter has no overflow.

Bit0 TF3: Timer3 counter overflow interrupt flag;

1= The Timer3 counter overflows and is automatically cleared by hardware when entering the interrupt service routine, or it can be cleared by software;

0= Timer3 counter has no overflow.

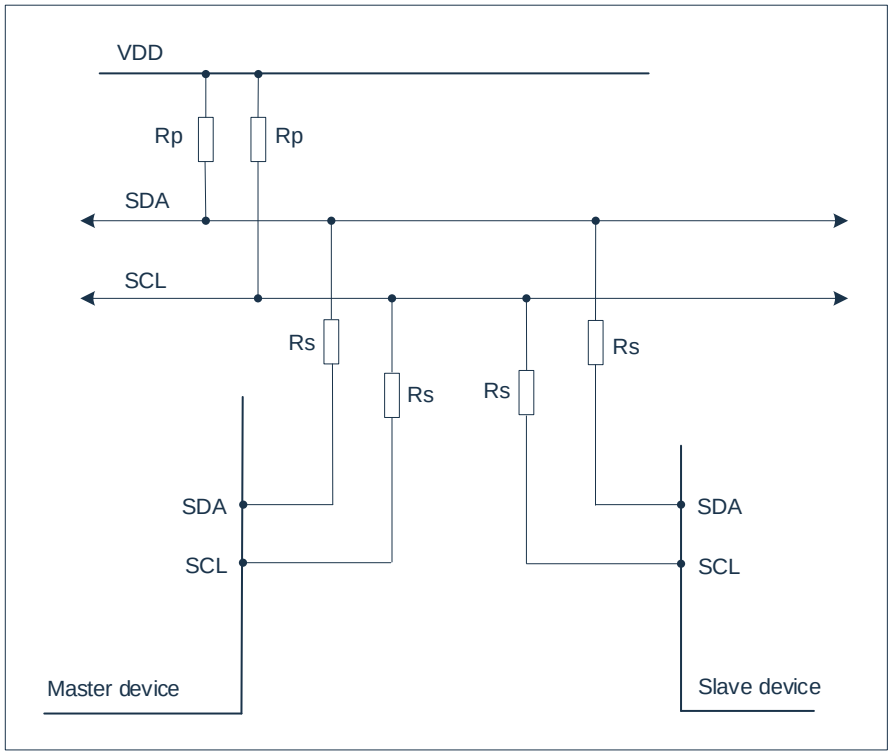
22. I²C Module

22.1 Overview

This module provides the interface between the microprocessor and the I²C bus. The connection diagram is shown in the figure below, and supports arbitration and clock synchronization so that it can run in multi-master system. I²C supports normal and fast modes.

The I²C module has the following features:

- Supports four transfer mode: master transmitter, master receiver, slave transmitter and slave receiver.
- Supports two transfer speed mode:
  - Standard speed(up to 100Kbs);
  - Fast speed (up to 400Kbs);
- Supports arbitration and clock synchronization.
- Supports multi-master system.
- Master mode supports 7-bit addressing mode and 10-bit addressing mode.
- Slave mode supports 7-bit addressing mode.
- Supports interrupt.
- Operate at a wide range of clock frequency(8-bit internal timer).



22.2 I²C Port Configuration

If you use the I²C function, you should first configure the corresponding ports as SCL and SDA channels. For example, configure P00 and P01 ports as I²C functions:

```
PS_SCL = 0x00;     //Select port P00 as the SCL pin
PS_SDA = 0x01;     //Select port P01 as the SDA pin
P00CFG = 0x02;     //P00 multiplexed SCL function
P01CFG = 0x02;     //P01 multiplexed SDA function
```

After configuring I2C channel, this group of ports is in open drain state by default. You can configure whether to enable SCL, internal pull-up resistance of SDA port, or add pull-up resistance outside the chip through PxUP.

In the master control mode, IIC outputs SCL to the slave. After sending the address or data, the slave needs to pull the SCL down and send back the corresponding response signal to the host. The host needs to read back the SCL port line status to detect whether the slave releases the SCL to determine whether the next frame data transmission is required. If the pull-up resistance or board-level parasitic capacitance of SCL is larger, the reading back time will be longer, which will affect the communication speed of IIC. Please refer to IIC application manual for details.

22.3 I²C Master Mode

The six register that is connected with host machine are as follows: control, status, address of slave address, send data, receive data and period register.

Register		Address
Write	Read	
Slave address register I2CMSA	Slave address register I2CMSA	0xF4
Control register in master mode I2CMCR	Control register in master mode I2CMSR	0xF5
Sending data register in master mode I2CMBUF	Receive data register in master mode I2CMBUF	0xF6
Period timing register I2CMTP	Period timing register I2CMTP	0xF7

In master mode, control register and status register use the same address, but they are physically different register.

In master mode, sending data register and receiving data register use the same register address, write operation access sending data register(I2CMBUF), Read operation access receiving data register (I2CMBUF).

Write operation is executed by control register. Read operation is executed by status register.

22.3.1 I²C Period Timer Register In Master Mode

In order to generate a wide range of SCL frequency, this model have 8-bit timer which is used to standard and fast transmission.

When TIMER_PRD ≠ 0，Ideal clock cycle of SCL = 2* (1+TIMER_PRD)* 10* Tsys

When TIMER_PRD＝0，Ideal clock cycle of SCL = 3* 10* Tsys

Refer to IIC Application Manual for the specific calculation formula of SCL.

0xF7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CMTP	--	MTP6	MTP5	MTP4	MTP3	MTP2	MTP1	MTP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	1

Bit7 -- Reserved, must be 0.

Bit6~Bit0 MTP<6:0>: The 6th and 0th bit of period timing register in standard and fast mode: TIMER_PRD[6:0].

22.3.2 I²C Control And Status Register In Master Mode

Control register include 4-bit: RUN, START, STOP and ACK. The bit of START generate conditions of START or REPEATED START. The bit of STOP decides whether the transmission of data is stopped at the end of period. The work flow of generate a single sending period is as follows: Address register of the slave machine write address, then the bit of R/S is set to 0, finally, Control register write code statements such as ACK=x、STOP=1、START=1、RUN=1 (I2CMCR=xxx0_x111x) which is used to perform and stop action. When the action is complete or some failure is occurred, an interruption is generated. These data can be received by receiving data register.

When I²C is working in master mode, the bit of ACK must be set to 1. Accordingly, The I²C bus controller can automatic response after sending each byte. The bit of ACK must be set to 0 when the I²C bus controller don't need data which is sent by slave machine.

Control register in master mode

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: the bit of I²C active recovery control;

1= Reset control module(I²C register of master control module, include I2CMSR);

0= Interrupt flag is set to 0 in I²C master control module.

Bit6~Bit5 -- Reserved

Bit4 -- Reserved，Must be 0.

Bit3 ACK: Response enable;

1= Enable;

0= Disable.

Bit2 STOP: Stop enable;

1= Enable;

0= Disable.

Bit1 START: Start enable;

1= Enable;

0= Disable.

Bit0 RUN: Run enable;

1= Enable;

0= Disable.

The combination of the following control bit list can realize all kinds of operations in master control module:

START: send start signal.

SEND: send data or address.

RECEIVE: receive data.

STOP: send end signal.

The combination of control bit (IDLE state)

R/S	ACK	STOP	START	RUN	OPERATION
0	-	0	1	1	SEND after START(master machine maintain in send mode)
0	-	1	1	1	END and STOP after START
1	0	0	1	1	Non-responsive reception after START (master machine maintain in receive mode)
1	0	1	1	1	RECEIVE and STOP after START
1	1	0	1	1	RECEIVE after START(master machine maintain in receive mode)
1	1	1	1	1	No combination
0	0	0	0	1	No combination

The combination of control bit(sending state in main control module)

R/S	ACK	STOP	START	RUN	OPERATION
-	-	0	0	1	SEND
-	-	1	0	0	STOP
-	-	1	0	1	STOP after SEND
0	-	0	1	1	SEND after repeated START
0	-	1	1	1	Repeat START, followed by SEND and STOP
1	0	0	1	1	Repeat START, followed by RECAIVE (master machine maintain in receive mode)
1	0	1	1	1	Repeat START, followed by SEND and STOP
1	1	0	1	1	Repeat START, followed by RECAIVE (master machine maintain in receive mode)
1	1	1	1	1	No combination

The combination of control bit(receiving state in main control module)

R/S	ACK	STOP	START	RUN	OPERATION
-	0	0	0	1	RECEIVE(master machine maintain in receive mode)
-	-	1	0	0	STOP
-	0	1	0	1	STOP after RECEIVE
-	1	0	0	1	RECEIVE (master machine maintain in receive mode)
-	1	1	0	1	No combination
1	0	0	1	1	Repeat START, followed by RECAIVE(master machine maintain in receive mode)
1	0	1	1	1	Repeat START, followed by SEND and STOP
1	1	0	1	1	Repeat START, followed by RECAIVE(master machine maintain in eceive mode)
0	-	0	1	1	Repeat START, followed by SEND(master machine maintain in send mode)
0	-	1	1	1	Repeat START, followed by SEND and STOP

Status register in master mode 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 in I^2C master mode;

1= Complete send/receive in master mode, or occur send failure. (write 0 to reset);

0= No interrupt

Bit6 BUS_BUSY: The I^2C bus busy flag bit in master mode/salve mode;

1= The I^2C bus is working, that can not send data(this start bit of bus is set to 1 and the stop condition is set to 0)

0= --

Bit5 IDLE: Idle flag in I^2C master mode;

1= Idle;

0= Work.

Bit4 ARB_LOST: Arbitration flag in I^2C master mode;

1= Lose control of bus.

0= --

Bit3 DATA_ACK: The response of sending data in I^2C master control mode;

1= No response of the last sending data.

0= --

Bit2 ADD_ACK: Addressing flag in I^2C master control mode;

1= No response of the last addressing.

0= --

Bit1 ERROR: I^2C master mode error flag

1= No response of addressing slave machine and arbitration conflict.

0= --

Bit0 BUSY: Busy flag in I²C master control mode;

1= Sending data.

0= --

22.3.3 I²C Slave Address Register

The salve address register is composed by 8 bits: address bit of 7 bits(A6-A0) and receive/send bit(R/S). R/S determines whether the next operation is to receive(1) or to send(0).

Salve address register in master control mode 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 control mode.

Bit0 R/S: Receiving/sending status selection bit after sending address of salve machine in I²C master control mode.

1= Receiving data after correct addressing.

0= Sending data after correct addressing.

22.3.4 Sending And Receiving Data Register In I²C Master Control Mode

The sending data register consists of 8 bits that will be sent on the bus at the next transmission or burst operation, and the first transmission bit is MD7 (MSB) .

Data cache register in master control mode 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>: Receiving/sending data in I²C master control mode.

22.4 I²C Slave Mode

There are five registers that is used to connect object device: address, control, send data and receive data.

Register		Address
Write	Read	
Own Address register (I2CSADR)	Own Address register I2CSADR	0xF1
Control register I2CSCR	Status register I2CSSR	0xF2
Sending data I2CSBUF	Receiving data I2CSBUF	0xF3

22.4.1 I²C Own Address Register I2CSADR

The own address register is made up of 7 bits that identify the I²C core on the I²C bus. This register can read and write addresses.

own address register I2CSADR

0xF1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CSADR	--	SA6	SA5	SA4	SA3	SA2	SA1	SA0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 -- Reserved, must be 0.

Bit6~Bit0 SA<6:0>: Own address in I²C salve mode.

22.4.2 I²C Control Register And Status Register Of I²C Slave Mode I2CSCR/I2CSSR

In slave mode, control register and status register occupy a register address and use different operations to distinguish access to the two registers.

Write: write I2CSCR(only write)

Read: read I2CSSR(only read)

The control register consists of two bits: RSTS and DA. The RSTS controls the reset of the entire I²C salve module. When the I²C encounters some problems, the I2CS can be reinitialized by using RSTS. The DA can enable and disable I2CS device. Read the address place the status register on the data bus.

Control register in slave mode 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: Reset control bit of I²C slave mode;

1= Reset slave module;

0= No effect.

Bit6~Bit1 -- Reserved, must be 0.

Bit0 DA: I²C slave mode enable;

1= Enable;

0= Disable.

The status register consists of three bits: SENDFIN, RREQ and TREQ. The SENDFIN bit indicates that the host I²C controller has completed receiving data during the single or continuous transmission operation of I2CS. The RREQ bit indicates that the I2CS device has received data from the I²C host machine, then the I2CS device should read a byte from the receiving register(I2CSBUF). The TREQ bit indicates that the I2CS device is addressed as a slave transmitter, and the I2CS write a byte to the sending register(I2CSBUF) . If I²C interrupt is enabled, any one of the 3 flag bits which is set to 1 will cause interruption.

In the slave mode, the bus busy flag bit is determined by Bit6 (BUS_BUSY) of the master mode status register I2CMSR. When the bus is idle, I2CMSR is 0x20. When the start condition is generated to the stop condition is generated, the I2CMSR register is 0x60. When the stop condition is generated, I2CMSR is 0x20.

Status register in slave mode 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: Sending complete flag In I²C slave mode(only read).

1= The master control device no longer needs data, the TREQ is no longer set to 1, and the data transmission has been completed. (automatically cleared after reading I2CSCR).

0= --

Bit1 TREQ: Prepared sending flag In I²C slave mode(only read).

1= As a transmitting device, it has been addressed or the main control device is ready to receive data. (automatically cleared after writing I2CSBUF).

0= --

Bit0 RREQ: Receiving complete flag In I²C slave mode(only read).

1= Complete receive (automatically cleared after reading I2CSBUF).

0= Uncompleted receive.

22.4.3 Sending And Receiving Cached Register Of I²C Slave Mode I2CSBUF

0xF3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CSBUF	I2CSBUF7	I2CSBUF6	I2CSBUF5	I2CSBUF4	I2CSBUF3	I2CSBUF2	I2CSBUF1	I2CSBUF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 I2CSBUF<7:0>: Data sent or received by I²C;

write: Write data (order from high to low).

read: Read received data.

22.5 I²C interrupt

The interruption number of I²C is 21, where the interruption vector is 0x00AB. To enable I²C interrupt, the enable bit(I2CIE) is set to 1, and the master interruption enable bit(EA) is set to 1.

If all the interruption enable is turned on and master interruption enable(I2CIF) is set to 1, then the CPU will enter the interrupt service program. The I2CIF property is read-only and has nothing to do with the state of I2CIE.

Interrupt flag bit(I2CMIF) in master mode, SENDFIN in slave mode, TREQ in slave mode and RREQ in slave mode, when any one of them is 1, the master interrupt flag bit(I2CIF) of I²C is set to 1. When the 4 flag bits are all 0, the I2CIF will automatically clear 0.

22.5.1 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;
1= Enable SPI interrupt;
0= Disable SPI interrupt;

Bit6 I2CIE: I²C interrupt enable;
1= Enable I²C interrupt;
0= Disable I²C interrupt;

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

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

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

Bit2 --: Reserved, must to be 0.

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

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

22.5.2 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;
1= Set to high level interrupt;
0= Set to low level interrupt.

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

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

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

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

Bit2 -- Reserved, must to be 0.

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

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

22.5.3 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 indicator, read only;

1= SPI produces an interrupt (this bit is automatically cleared after clearing the specific interrupt flag;

0= SPI did not cause an interruption.

Bit6 I2CIF: I^2C total interrupt indicator position, read only;

1= I^2C produces an interrupt (after clearing the specific interrupt flag bit, this bit is automatically cleared);

0= There was no interruption to the I^2C.

Bit5 -- Reserved, must to be 0.

Bit4 ADCIF: ADC interrupt flag bit;

1= ADC conversion is completed, software clearance is required;

0= ADC conversion not completed.

Bit3 PWMIF: PWM total interrupt indicator, read only;

1= PWM produces an interrupt;

0= PWM did not interrupt.

Bit2 -- Reserved, must to be 0.

Bit1 TF4: Timer4 counter overflow interrupt flag;

1= The Timer4 counter overflows and is automatically cleared by hardware when entering the interrupt service routine, or it can be cleared by software;

0= Timer4 counter has no overflow.

Bit0 TF3: Timer3 counter overflow interrupt flag;

1= The Timer3 counter overflows and is automatically cleared by hardware when entering the interrupt service routine, or it can be cleared by software;

0= Timer3 counter has no overflow.

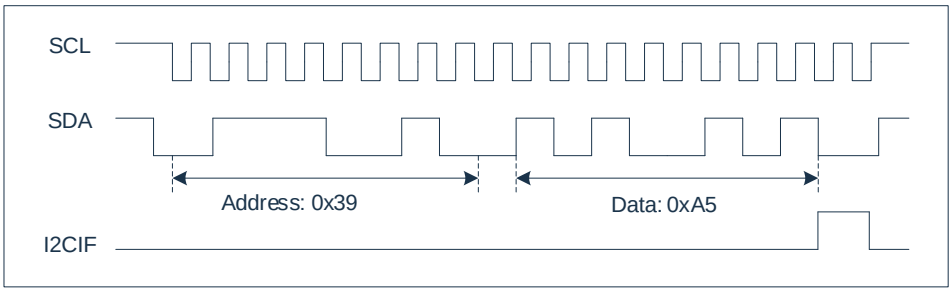
22.6 I²C Transmission Method of Slave Mode

The default I²C address of all the waveform presented in this section is 0x39 ("00111001").

22.6.1 Single Receiving

The following figure shows the sequence of signals received by I²C during single data. Single receiving sequence is as follows:

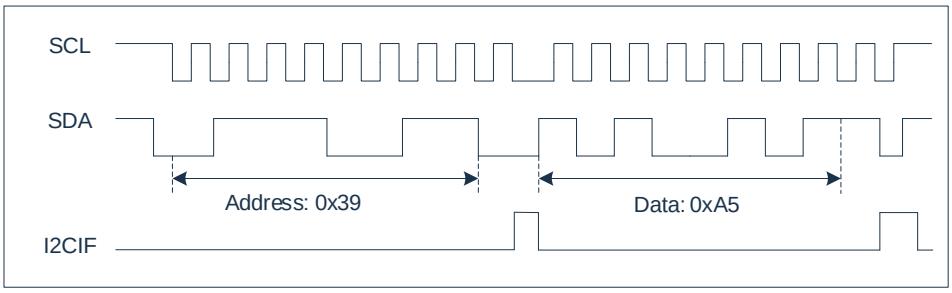
- Starting conditions.
- Addressing by I²C host machine.
- The address is confirmed by I²C.
- The data is received by I²C.
- The data is confirmed by I²C.
- Stop conditions.



22.6.2 Single Sending

The following figure shows the sequence of signals sent by I²C during single data. Single sending sequence is as follows:

- Starting conditions.
- Addressing by I²C host machine.
- The address is confirmed by I²C.
- The data is transmitted by I²C.
- The data is not confirmed by I²C.
- Stop conditions.

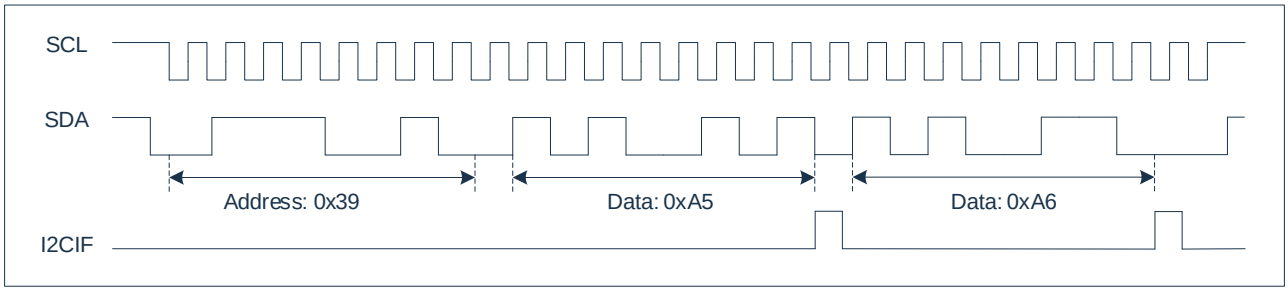


22.6.3 Continuous Receiving

The following figure shows the sequence of signals received by I²C during burst data. Continuous receive sequence:

- Starting conditions.
- Addressing by I²C host machine.
- The address is confirmed by I²C.
  - (1) The data is received by I²C.
  - (2) The data is confirmed by I²C.
- Stop conditions.

Repeat sequence (1) and sequence (2) until the stop conditions occur.

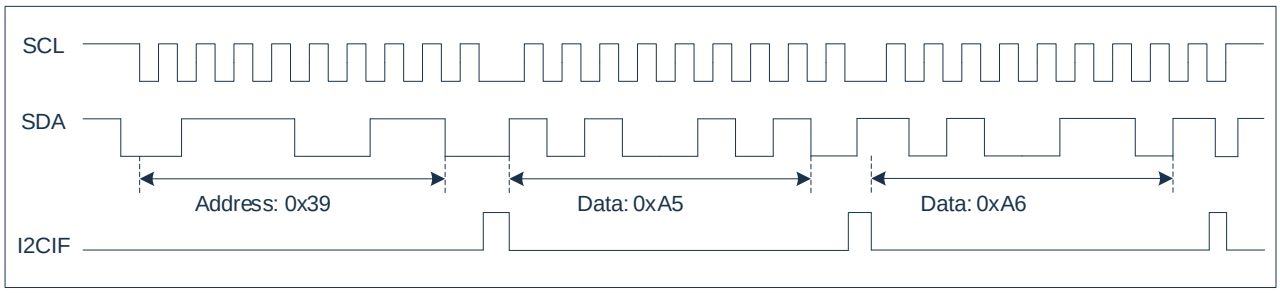


22.6.4 Continuous Sending

The following figure shows the signal sequence sent by I²C during continuous data transmission. Continuous transmission sequence:

- Starting conditions.
- Addressing by I²C host machine.
- The address is confirmed by I²C.
  - (1) The data is sent by I²C.
  - (2) The data is confirmed by I²C.
  - (3) The last data is not confirmed by I²C.
- Stop conditions.

Repeat sequence (1) and sequence (2) until sequence (3) occur.



23. UARTn Moudle

23.1 Overview

The Universal Synchronous Asynchronous Receiver Transmitter (UART0 / UART1) provides a flexible method for full-duplex data exchange with external devices.

There are two physically independent receiving and sending buffers inside UARTn.SBUFn, which can be used to distinguish between receiving and sending buffers by reading and writing commands to SBUFn. When writing SBUFn, the data is loaded into the sending buffer; when reading SBUFn, reading the content in the receiving buffer.

The serial port can operate in 4 modes: one synchronous and three asynchronous modes. Mode 2 and 3 has a special feature for multi-processor communications. This feature is enabled by setting SMn2 bit in SCONn register. The master firstly sends out an address byte, which identifies the target slave. An address byte differs from a data byte in that the 9th bit is 1 in an address byte and 0 in a data byte. When SMn2 = 1, slave will not be interrupted by a data byte. An address byte will interrupt all slaves. The addressed slave will clear its SMn2 bit and prepare to receive the data bytes that will be coming. The slaves that were not being addressed will set “1” to SMn2 bit and ignore the incoming data.

23.2 UARTn Port Configuration

You need to configure the corresponding ports as TXDn and RXDn channels of UARTn before using the UARTn module. For example, the port configuration of UART0 is as follows:

```
PS_RXD0 = 0x05; //Select P05 as the RXD0 pin
P05CFG = 0x02;  //P05 is multiplexed as RXD0 function
P04CFG = 0x02;  //P04 is multiplexed as TXD0 function
```

The port RXDn of UARTn can be selected by PS_RXDn (only one of the RXDn pins can be selected), and the TXDn port can be selected by the port configuration register (either can be selected at the same time, or one of them. If selected at the same time, both will output the corresponding waveform).

When using, it is recommended to set the working mode first, and then configure the corresponding port as a serial port.

23.3 UARTn Baud Rate

When 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 timer Timer1 or Timer4 or Timer2 or BRT module. Which timer the chip selects as the baud rate clock source is determined by the register FUNCCR.

23.3.1 Baud Rate Clock Source

When UARTn is in mode 1 and mode 3, the baud rate clock source is selected as follows:

1）UART0 baud rate clock source selection:

When FUNCCR[2:0]=000, select Timer1 as the baud rate generator of UART0;

When FUNCCR[2:0]=001, select Timer4 as the baud rate generator of UART0;

When FUNCCR[2:0]=010, select Timer2 as the baud rate generator of UART0;

When FUNCCR[2:0]=011, select BRT as the baud rate generator of UART0.

2）UART1 baud rate clock source selection:

When FUNCCR[6:4]=000, select Timer1 as the baud rate generator of UART1;

When FUNCCR[6:4]=001, select Timer4 as the baud rate generator of UART1;

When FUNCCR[6:4]=010, select Timer2 as the baud rate generator of UART1;

When FUNCCR[6:4]=011, select BRT as the baud rate generator of UART1;

23.3.2 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:

$$Baud\ Rate=\frac{Fsys\times 2^{SMODn}}{32\times(4\times 3^{1-TxM})\times(256-THx)}(x=1,4)$$

SMODn is the baud rate selection bit, which is set by the register PCON. 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]. Under corresponding baud rate, TH1/TH4 of Timer1 or Timer4 should be set as:

$$THx=256-\frac{Fsys\times 2^{SMODn}}{32\times(4\times 3^{1-TxM})\times Baud\ Rate}(x=1,4)$$

2）The formula of the baud rate when Timer2 works in overflow auto-reload mode:

$$Baud\ Rate=\frac{Fsys\times 2^{SMODn}}{32\times(12\times 2^{T2PS})\times(65536-\{RLDH,RLDL\})}$$

T2PS is the timer 2 clock prescaler selection bit, which is set by register T2CON[7]. The value of Timer2 at the corresponding baud rate should be set to:

$$\{RLDH,RLDL\}=65536-\frac{Fsys\times 2^{SMODn}}{32\times(12\times 2^{T2PS})\times Baud\ Rate}$$

3）When BRT is used as a baud rate generator, the baud rate formula:

$$Baud\ Rate=\frac{Fsys\times 2^{SMODn}}{32\times(65536-\{BRTDH,BRTDL\})\times 2^{BRTCKDIV}}$$

BRTCKDIV is the BRT timer prescaler selection bit, which is set by the register BRTCON. The value of BRT under the corresponding baud rate should be set to:

$$\{BRTDH,BRTDL\}=65536-\frac{Fsys\times 2^{SMODn}}{32\times 2^{BRTCKDIV}\times BaudRate}$$

23.3.3 Baud Rate Deviation

When UARTn is in mode 1 and mode 3, select different baud rate clock sources, the error is as follows under different baud rates:

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 BRT timer overflow rate 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

Baud rate	Fsys=8MHz			Fsys=16MHz			Fsys=24MHz			Fsys=48MHz		
bps	{BRT, 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

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

23.4 UARTn Registers

The UARTn has the same functionality as a standard 8051 UART. The related registers are: FUNCCR、SBUFn、SCONn、PCON、IE、IP、EIP3. The UARTn data buffer (SBUFn) consists of two separate registers: transmit and receive registers. A data written into the SBUFn will be set in UARTn output register and starts a transmission. Reading SBUFn will read data from the UARTn receive register. SCON0 register support bit addressing operation, but SCON1 register doesn't support this function. It should be paid attention when using assembly language. The baud rate is doubled by setting the register PCON.

23.4.1 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;
	else=	No 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;
	else=	No Access.

23.4.2 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: SBUF0 register address 0x99; SBUF1 register address 0xEB.

Bit7~Bit0	BUFFERn<7:0>:	Data buffer register.
	write:	UARTn starts to send data.
	read:	Read received data.

23.4.3 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: SCON0 register address: 0x98; SCON1 register address: 0xEA.

Bit7~Bit6 UnSM0- UnSM1: Control bit for multi-processor communications;
00= Master control synchronous mode;
01= 8-bit asynchronous mode, baud rate is variable;
10= 9-bit asynchronous mode, baud rate is Fsys/32 or Fsys/64;
11= 9-bit asynchronous mode, baud rate is variable.

Bit5 UnSM2: Enable bit for multi-processor communications;
1= Enable;
0= Disable.

Bit4 UnREN: Receive enable;
1= Enable;
0= Disable.

Bit3 UnTB8: The 9th bit of transmitting data, mainly used for transmitting of 9-bit
1= The 9th bit is 1;
0= The 9th bit is 0.

Bit2 UnRB8: The 9th bit of receiving data, mainly used for receiving of 9-bit
1= The 9th bit of receiving data is 1;
0= The 9th bit of receiving data is 0.

Bit1 TIn: Transmit interrupt flag (needs software clear);
1= Transmit buffer is empty, it can transmit next data.
0= --

Bit0 RIn: Receive interrupt flag (needs software clear);
1= Receive buffer is full, it could read data from receive register.
0= --

UARTn mode as shown in the following table:

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
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

23.4.4 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;

1= UART0 baud rate doubling;

0= UART0 baud rate normal.

Bit6 SMOD1: UART1 baud rate double;

1= UART1 baud rate doubling;

0= UART1 baud rate normal.

Bit5~Bit3 -- Reserved, all must be 0.

Bit2 SWE: Wake-up enable bit for STOP status;

(Regardless of the value of SWE, the system can be restarted by power-off reset or enabled external reset)

0= Prohibit function of wake-up;

1= Allow function wake-up (wake-up by external interrupt and timed wake-up)

Bit1 STOP: Sleep state control;

1= Enter sleep state (automatically clear when exit STOP mode);

0= Not enter sleep state.

Bit0 IDLE: Idle state control;

1= Enter idle state (automatically clear when exit IDLE);

0= Not enter idle state.

23.5 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.

Enable UARTn interrupt needs to set 1 to ESn bit and set 1 to overall interrupt enable bit: EA. If related interrupt enabling is turned on, Tin = 1 or RIn = 1, CPU will enter related ISR (Interrupt Service Routines). TIn/RIn has no relation with ESn, and needs software clear. For detailed description, refer to register SCONn.

23.5.1 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;
1= Enable all unmasked interrupts;
0= Disable all interrupt.

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

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

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

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

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

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

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

23.5.2 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;
1= Set to high priority interrupt;
0= Set to low priority interrupt.

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

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

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

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

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

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

23.5.3 Interrupt Priority Control Register EIP3

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

Bit7~Bit5 -- Reserved, all must be 0.

Bit4 PTOUCH: TOUCH interrupt priority control;

1= Set to high priority interrupt;

0= Set to low priority interrupt.

Bit3 PLVD: LVD interrupt priority control;

1= Set to high priority interrupt;

0= Set to low priority interrupt.

Bit2 PLSE: LSE interrupt priority control;

1= Set to high priority interrupt;

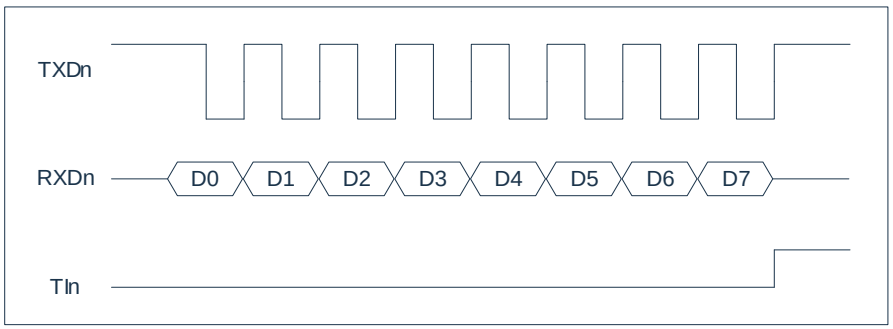
0= Set to low priority interrupt.

Bit1~Bit0 -- Reserved, all must be 0.

23.6 UARTn Mode

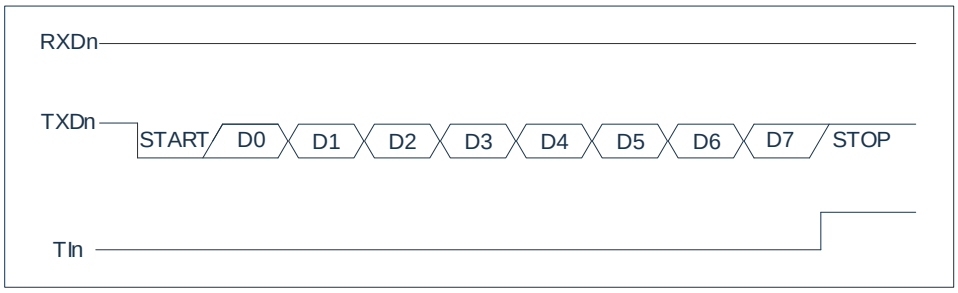
23.6.1 Mode 0 - Synchronous Mode

Pin RXDn serves as input and TXDn as output. TXDn output is a shift clock. The baud rate is fixed at 1/12 of the CLK clock frequency. Eight bits are transmitted with 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:



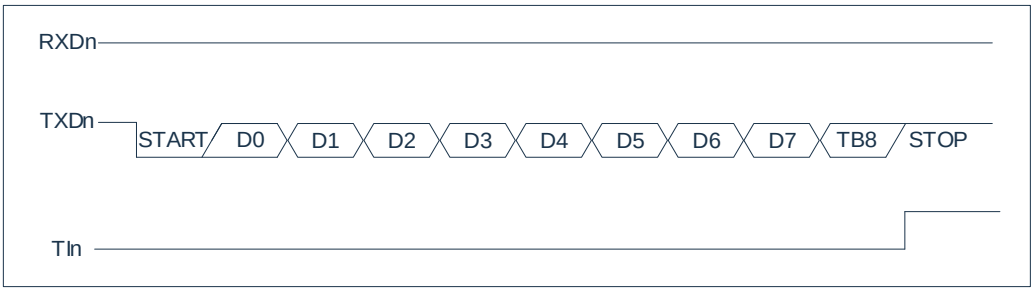
23.6.2 Mode 1 - 8-bit Asynchronous Mode (Variable Baud Rate)

Pin RXDn serves as input, and TXDn serves as serial output. 10 bits are transmitted: a start bit (always 0), 8 data bits (LSB first), and a stop bit (always 1). When receive, a start bit synchronizes the transmission, 8 data bits are got by reading SBUFn, and stop bit sets the flag RBn8 in the register: SCONn. The baud rate is variable and depends on the mode of Timer1/Timer2/Timer4/BRT. The timing diagram of Mode 1 is shown in the figure below:



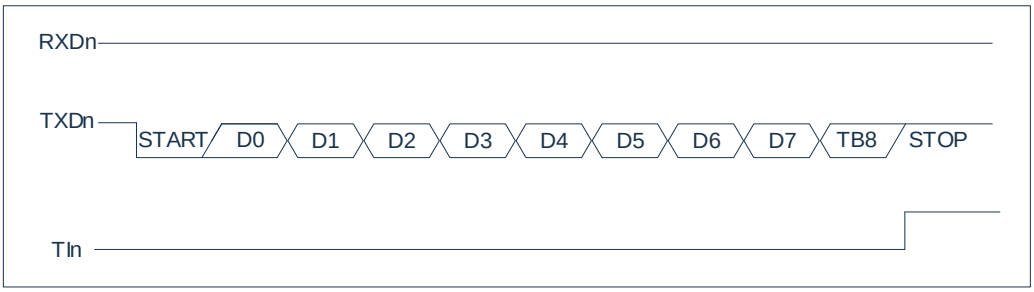
23.6.3 Mode 2 - 9-bit Asynchronous Mode (Fixed Baud Rate)

This mode is similar to Mode 1 with two differences. The baud rate is fixed at 1/32 or 1/64 of CLK clock frequency, and 11 bits are transmitted or received: a start bit (0), 8 data bits (LSB first), a programmable 9th bit, and a stop bit (1). The 9th bit can be used to control the parity checking of the UARTn interface: at transmission, bit TBn8 in SCONn is output as the 9th bit, and at receive, the 9th bit affects RBn8 in SCONn. The timing diagram of Mode 2 is shown in the figure below:



23.6.4 Mode 3 - 9-bit Asynchronous Mode (Variable Baud Rate)

The only difference between Mode 2 and Mode 3 is that the baud rate is variable in Mode 3. When REN0=1, data receiving is enabled. The baud rate is variable and depends on the mode of Timer1/Timer2/Timer4/BRT. The timing diagram of Mode 3 is shown in the figure below:



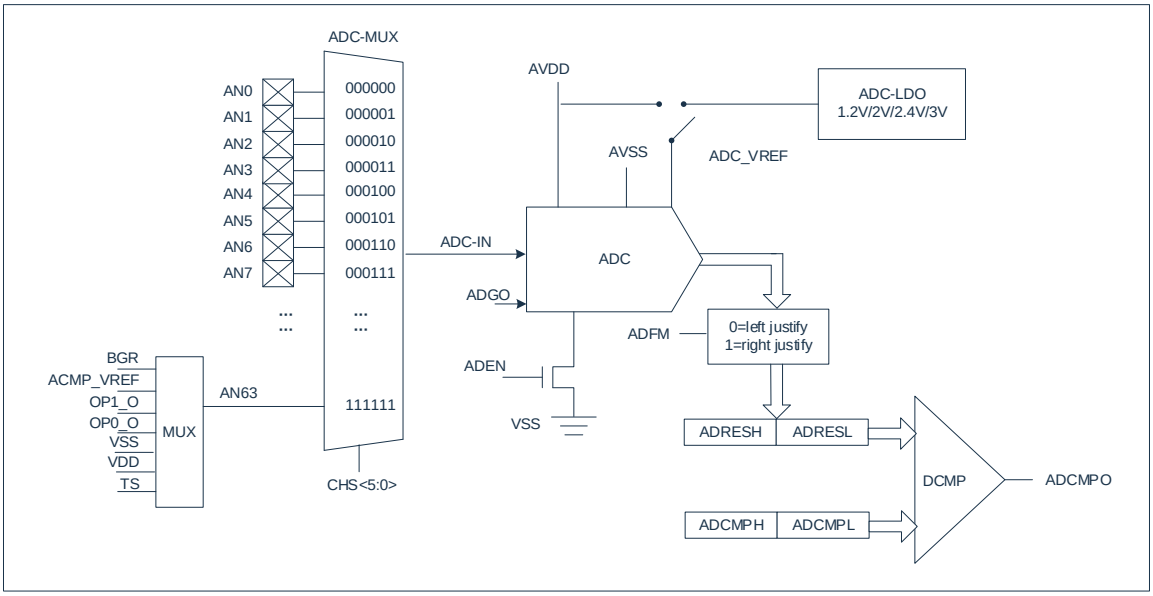
24. Analog To Digital Conversion (ADC)

24.1 Overview

The ADC can convert the analog input signal into a 12-bit binary representing the signal, the block diagram of ADC structure is shown in the figure below.

The port analog input signal and internal analog signal is connected to the input of the analog to digital converter after passing through the multiplexer. ADC uses a 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 complete. The ADC conversion result is compared with the value of the ADC compare data register (ADCMPL and ADCMPH), the result of the comparison is stored in the ADCMPO flag.

The ADC reference voltage is always internally generated and can be provided by AVDD or by the internal ADC-LDO.



24.2 ADC Configuration

When configuring and using the ADC, the following factors must be considered:

- Port configuration.
- Channel selection.
- ADC conversion clock source.
- Interrupt control.
- Result storage format.

24.2.1 Port Configuration

The ADC can convert analog signals and convert digital signals. When converting analog signals, should be configured as an analog port by configuring the corresponding.

Note: Applying an analog voltage to a pin defined as a digital input may cause an overcurrent in the input buffer.

24.2.2 Channel Selection

The register ADCCHS determines which channel to connect to the ADC.

When channel is changed, a certain delay is required before the next conversion starts. The ADC delay time is shown in the following table:

Delay time	Operating voltage
500ns	2.5~4.5V
200ns	4.5~5.5V

24.2.3 ADC Reference Voltage

The 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 4 kinds of voltage output: 1.2V/2.0V/2.4V/3.0V.

24.2.4 Conversion Clock

The ADCKS bit of the ADCON1 register can be set by software to select the clock source for conversion.

The time to complete one bit conversion is defined as T_{ADCK}. A complete 12-bit conversion requires 18.5 T_{ADCK} cycles (the duration of ADGO to complete a conversion is high). The correct conversion results can only be obtained if the corresponding T_{ADCK} specifications are met. The following table is an example of selecting the ADC clock correctly.

Fsys	F_{ADCK} (T_A=25℃)		
	V_{REF}=V_{REF1}=AVDD (AVDD=VDD)	V_{REF}=V_{REF2}=1.2V	V_{REF}=V_{REF3}=2.0V V_{REF}=V_{REF4}=2.4V V_{REF}=V_{REF5}=3.0V
8MHz	F_{sys}/4	F_{sys}/256	F_{sys}/16
16MHz	F_{sys}/8	forbidden	F_{sys}/32
24MHz	F_{sys}/16	forbidden	F_{sys}/64
48MHz	F_{sys}/32	forbidden	F_{sys}/128

Note: Any change in the system clock frequency will change the frequency of the ADC clock, which will negatively affect the ADC conversion result.

24.2.5 Result Formatting

The result of a 12-bit A/D conversion can be in two formats: left aligned or right aligned. The output format is controlled by the ADFM bit of the ADCON0 register.

When ADFM=0, the AD conversion result is left aligned;

When ADFM=1, the AD conversion result is right aligned.

24.3 ADC Hardware Triggered Start

In addition to the software-triggered AD conversion, the ADC module also can be triggered by the hardware, one for the external port edge trigger mode, one It is an edge or period trigger of PWM.

Using hardware to trigger the ADC requires ADCEX to be set, that is, enable the 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. When the hardware trigger function is enabled, the software trigger function will not be turned off. When the ADC is idle, writing 1 to the ADGO bit can also start AD conversion.

24.3.1 External Port Edge Trigger ADC

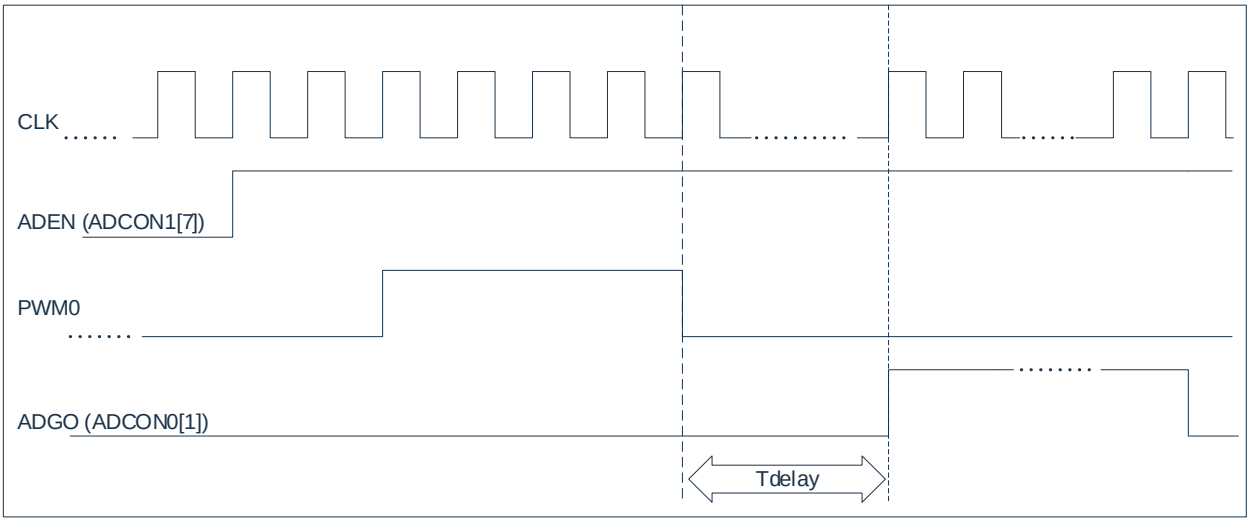
The ADET pin edge automatically triggers an AD conversion. At this time ADTGS[1:0] needs to be 11 (select external port edge trigger), and ADEGS[1:0] can select which edge trigger.

24.3.2 PWM Triggered ADC

PWM can choose to trigger ADC conversion by edge or cycle point/zero point. ADTGS[1:0] selects PWM channel (PG0, PG2, PG4), ADEGS[1:0] can select edge type or period type trigger mode.

24.3.3 Hardware Trigger Delay Before Starting

After the hardware trigger signal is generated, the AD conversion is not started immediately. It takes a certain delay before the ADGO value is set to 1. Delay by ADDLY[9:0] decided. Delay time of hardware trigger signal: (ADDLY+3) *Tsys, the structure diagram of the delay trigger is shown in the figure below:



24.4 ADC Results Of Comparison

ADC block provides a set of digital comparators for comparison between the results of ADC and the numbers pre-loaded in {ADCMPH,ADCMPL}. The conversion results of ADC will be compared with preset value ADCMP every time, the result of comparison will be stored in ADCMPO flag bit, this flag bit will be updated automatically after the conversion. ADCMPPS bit could change the polarity of the output results.

ADC comparison result could trigger PWM fault brake, it requires to set ADFBEN to 1 if enabling this function.

When the enhanced PWM function is enabled and ADFBEN=1, the conversion results of ADC will be compared with preset value {ADCMPH,ADCMPL}, if the comparison result is 1, PWM generates fault brake action, resets all the PWM channels and terminates all the outputs of PWM channels.

24.5 ADC Working Principle

24.5.1 Start Conversion

To enable the ADC module, the ADEN bit of the ADCON1 register must be set and then the analog-to-digital conversion is initiated by setting the ADGO bit of the ADCON0 register (ADGO cannot be set when ADEN is '0').

24.5.2 Complete Conversion

When the conversion is complete, the ADC module will:

- Clear ADGO bit;
- Set ADCIF bit to 1;
- Update the ADRESH: ADRESL register with the new result of the conversion.

24.5.3 Terminate Conversion

If the conversion must be terminated before the conversion is complete, the uncompleted analog-to-digital conversion results are not updated to the ADRESH:ADRESL register. Therefore, the ADRESH:ADRESL register will retain the value obtained from the last conversion.

Note: Device Reset will force all registers to the Reset state. Therefore, reset will turn off the ADC module and terminate any pending conversions.

24.5.4 A/D Conversion Step

The configuration steps for analog to digital conversion using ADC are as follows:

1) PIN Configuration:
   - Use a pin as an output driver is forbidden (PxTRIS register);
   - Configure pins as analog input pin.
2) Configure ADC interrupt (optional):
   - Clear ADC interrupt flag bit;
   - Enable ADC interrupt;
   - Enable peripheral interrupt;
   - Enable global interrupt.
3) Configute ADC module:
   - Choose ADC conversion clock;
   - Choose ADC input channel;
   - Choose the format of results;
   - Initiate ADC module.
4) Waiting for required sampling time.
5) Set ADGO to 1 to initiate conversion.
6) Waiting for the end of conversion by one of the following ways:
   - Check ADGO bit;
   - Waiting for the ADC interrupt (enable interrupt).
7) Read ADC results.
8) Reset the ADC flag bit of interrupt (This step is required to be done if interrupt is enabled).

Note: If the user tries to resume sequential code execution after waking the device from sleep mode, the global interrupt must be disabled.

24.5.5 Enter Sleep Mode During Conversion

When the system is about to sleep, it's recommended to wait for the conversion is completed, and then getting into sleep mode.

If the system is going to be in sleep mode while ADC is converting, then the conversion is terminated, it will convert again after weak-up.

24.6 Related Register

The following nine registers are related to AD conversion:

- AD control registers:ADCON0、ADCON1、ADCON2、ADCCHS、ADCLDO;
- Comparator control register: ADCMPC;
- Delay data register: ADDLYL;
- AD results data registers: ADRESH/L;
- Comparator data register: ADCMPH/L.

24.6.1 AD Control Register ADCON0

0xDF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON0	--	ADFM	ANACH3	ANACH2	ANACH1	ANACH0	ADGO	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 -- Reserved, must to be 0.

Bit6 ADFM: Format select bit of ADC conversion result;

1= Right-aligned;

0= Left-aligned.

Bit5~Bit2 ANACH<3:0>: ADC channel 63 input source selection bit;

0000= BGR (1.2V);

0001= ACMP_VREF (The negative terminal reference voltage of the comparator, see the ACMP chapter for details);

0010= OP1_O;

0011= OP0_O;

0100= TS_ANA (Temperature sensor voltage);

0101= VSS (ADC reference ground);

0110= Reserved, disabling use;

0111= VDD (ADC default reference voltage);

Others= Reserved, disabling use.

Bit1 ADGO: ADC conversion start (ADEN must be set to be 1 when this bit is 1, otherwise it's invalid operation).

1= write: start ADC conversion, (This bit will be set to be 1 when hardware triggers ADC).

read: ADC is converting.

0= write: invalid;

read: ADC enters idle/conversion complete;

During ADC's conversion time (ADGO=1), any trigger signals of hardware or software will be ignored.

Bit0 -- Reserved, must to be 0.

24.6.2 AD Control Register ADCON1

0xDE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON1	ADEN	ADCKS2	ADCKS1	ADCKS0	--	--	--	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	1	0	0	0	0	0	0

Bit7 ADEN: ADC enable;
1= Enable ADC;
0= Disable ADC, no operating current is consumed.

Bit6~Bit4 ADCKS<2:0>: ADC conversion clock select.
000= Fsys/2; 100= Fsys/32;
001= Fsys/4; 101= Fsys/64;
010= Fsys/8; 110= Fsys/128;
011= Fsys/16; 111= Fsys/256.

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

24.6.3 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.
1= Enable;
0= Disable.

Bit6 -- Reserved, must to be 0;

Bit5~Bit4 ADTGS<1:0>: ADC hardware trigger source select.
00= PG0(PWM0)
01= PG2 (PWM2)
10= PG4 (PWM4)
11= Port pin (ADET)

Bit3~Bit2 ADEGS<1:0>: ADC hardware trigger edge select;
00= Falling edge;
01= Rising edge;
10= The period points of the PWM period;
11= The zero point of the PWM period.

Bit1~Bit0 -- Reserved, all must to be 0.

24.6.4 AD Channel Selection Register ADCCHS

0xD9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCCHS	--	--	CHS5	CHS4	CHS3	CHS2	CHS1	CHS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 -- Reserved, must to be 0;

Bit5~Bit0 CHS<5:0>: Analog channel select;

000000=	AIN0;	010000=	AIN16;	100000=	AIN32;
000001=	AIN1;	010001=	AIN17;	100001=	AIN33;
000010=	AIN2;	010010=	AIN18;	100010=	AIN34;
000011=	AIN3;	010011=	AIN19;	100011=	AIN35;
000100=	AIN4;	010100=	AIN20;	100100=	AIN36;
000101=	AIN5;	010101=	AIN21;	100101=	AIN37;
000110=	AIN6;	010110=	AIN22;	100110=	AIN38;
000111=	AIN7;	010111=	AIN23;	100111=	AIN39;
001000=	AIN8;	011000=	AIN24;	101000=	AIN40;
001001=	AIN9;	011001=	AIN25;	101001=	AIN41;
001010=	AIN10;	011010=	AIN26;	101010=	AIN42;
001011=	AIN11;	011011=	AIN27;	101011=	AIN43;
001100=	AIN12;	011100=	AIN28;	101100=	AIN44;
001101=	AIN13;	011101=	AIN29;	101101=	AIN45;
001110=	AIN14;	011110=	AIN30;	Others=	Access barred;
001111=	AIN15;	011111=	AIN31;	111111=	See ADCON0.ANACH instruction.

24.6.5 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: The result of ADC comparator controls PWM brake enable;
1= Enable;
0= Disable.

Bit6 ADCMPPS: The output polarity of ADC comparator select;
1= If ADRES<ADCMP, ADCMPO=1;
0= If ADRES>=ADCMP, ADCMPO=1.

Bit5 -- Reserved, must to be 0.

Bit4 ADCMPO: ADC comparator output bit.
This bit outputs ADC comparator's result; it'll be updated every time ADC 's conversion is completed.

Bit3~Bit2 -- Reserved, all must to be 0.

Bit1~Bit0 ADDLY<9:8>: ADC hardware trigger delay data [9:8] bits.

24.6.6 AD Hardware Trigger Delay Data Register ADDLYL

0xD3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADDLYL	ADDLY7	ADDLY6	ADDLY5	ADDLY4	ADDLY3	ADDLY2	ADDLY1	ADDLY0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 ADDLY<7:0>: ADC hardware trigger delay data low 8-bit.

24.6.7 AD Data Register High Bits ADRESH，ADFM=0(Left-aligned)

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 results register bits.
Number 11 to 4 bit of 12-bit conversion results.

24.6.8 AD Data Register Low Bits ADRESL，ADFM=0(Left-aligned)

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 results register.
Number 3 to 0 bit of 12-bit conversion results.

Bit3~Bit0 Unused.

24.6.9 AD Data Register High ADRESH，ADFM=1(Right-aligned)

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 Unused.

Bit3~Bit0 ADRES<11:8>: ADC results register.
Number 11 to 8 bit of 12-bit conversion results.

24.6.10 AD Data Register Low ADRESH，ADFM=1(Right-aligned)

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 results register.
Number 7 to 0 bit of 12-bit conversion results.

24.6.11 AD Comparator Data Register ADCMPH

0xD5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCMPH	D11	D10	D9	D8	D7	D6	D5	D4
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	1	1	1	1	1	1	1	1

Bit7~Bit0 ADCMP<11:4>: High 8-bit of ADC comparator’s data. ADC比较器数据高8位.

24.6.12 AD Comparator Data Register ADCMPL

0xD4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCMPL	--	--	--	--	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	1	1	1	1	1	1	1	1

Bit7~Bit4 Unused.

Bit3~Bit0 ADCMP<3:0>: Lowe 4-bit of ADC comparator's data.

24.6.13 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 enable;

1= Enable LDO, the reference voltage can only select the voltage corresponding to VSEL [1:0];

0= Disable LDO, the reference voltage is the power supply voltage of the chip.

Bit6~Bit5 VSEL<1:0>: ADC reference voltage selection;

00= 1.2V;

01= 2.0V;

10= 2.4V;

11= 3.0V.

Bit4 -- Reserved, must to be 0.

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

24.7 ADC Interrupt

The ADC module allows an interrupt to be generated after the analog to digital conversion is completed. The ADC Interrupt Enable bit is the ADCIE bit in the EIE2 register. The ADC Interrupt Flag is the ADCIF bit in the EIF2 register. The ADCIF bit must be cleared in software. The ADCIF bit is set after each conversion and is independent of whether the ADC interrupt is enabled. The ADC interrupt enable and priority can be set by the following relevant register bits.

24.7.1 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;
1= Enable SPI interrupt;
0= Disable SPI interrupt;
Bit6 I2CIE: I^2C interrupt enable;
1= Enable I^2C interrupt;
0= Disable I^2C interrupt;
Bit5 WDTIE: WDT interrupt enable;
1= Enable WDT overflow interrupt;
0= Disable WDT overflow interrupt.
Bit4 ADCIE: ADC interrupt enable;
1= Enable ADC interrupt;
0= Disable ADC interrupt.
Bit3 PWMIE: PWM global interrupt enable;
1= Enable PWM all interruptions;
0= Disable PWM all interruptions.
Bit2 -- Reserved, must to be 0.
Bit1 ET4: Timer4 interrupt enable;
1= Enable Timer4 interrupt;
0= Disable Timer4 interrupt.
Bit0 ET3: Timer3 interrupt enable;
1= Enable Timer3 interrupt;
0= Disable Timer3 interrupt.

24.7.2 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;
1= Set to high level interrupt;
0= Set to low level interrupt.

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

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

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

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

Bit2 --: Reserved, must to be 0.

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

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

24.7.3 External Interrupt Flag Bit Register EIF2

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

Bit7 SPIIF: SPI total interrupt indicator, read only;

1= SPI produces an interrupt (this bit is automatically cleared after clearing the specific interrupt flag;

0= SPI did not cause an interruption.

Bit6 I2CIF: I^2C total interrupt indicator position, read only;

1= I^2C produces an interrupt (after clearing the specific interrupt flag bit, this bit is automatically cleared);

0= There was no interruption to the I^2C.

Bit5 -- Reserved, must to be 0.

Bit4 ADCIF: ADC interrupt flag bit;

1= ADC conversion is completed, software clearance is required;

0= ADC conversion not completed.

Bit3 PWMIF: PWM total interrupt indicator, read only;

1= PWM produces an interrupt;

0= PWM did not interrupt.

Bit2 -- Reserved, must to be 0.

Bit1 TF4: Timer4 counter overflow interrupt flag;

1= The Timer4 counter overflows and is automatically cleared by hardware when entering the interrupt service routine, or it can be cleared by software;

0= Timer4 counter has no overflow.

Bit0 TF3: Timer3 counter overflow interrupt flag;

1= The Timer3 counter overflows and is automatically cleared by hardware when entering the interrupt service routine, or it can be cleared by software;

0= Timer3 counter has no overflow.

25. Temperature Sensor

25.1 Overview

The chip contains a temperature sensor whose output analog varies with the chip temperature. The temperature change can be indirectly obtained by ADC sampling and converting the analog signal output by the sensor.

25.2 Register Description

25.2.1 Temperature Sensor Control Register TS_REG

0xF693	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TS_REG	TS_EN	TS_SEL	--	--	TS_TRIM3	TS_TRIM2	TS_TRIM1	TS_TRIM0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7	TS_EN:	Temperature sensor enable;
	1=	Enable;
	0=	Disable.
Bit6	TS_SEL:	Temperature sensor adjust bit selection;
	1=	Select register configuration;
	0=	Select config configuration.
Bit5~Bit4	--	Reserved, all must be 0.
Bit3~Bit0	TS_TRIM<3:0>:	Temperature sensor register configuration adjust.

25.3 Functional Description

25.3.1 Configuration

The temperature sensor will generate a corresponding analog quantity (TS_ANA) as the chip temperature changes. The analog quantity can be sampling through ADC, and the value of the ADC conversion result register can be read to obtain the chip temperature indirectly. The configuration of the temperature sensor is as follows:

1) Set the temperature sensor detection enable TS_REG [7] = 1.

2) Set the temperature sensor adjust bit selection and adjust bit value.

3) Set the temperature detection channel ADCCHS [5:0] = 111111, ADCON0[5:2] = 0100.

4) Set the ADC related configuration and turn on the ADC conversion enable.

5) Wait for the ADC conversion to complete and read the register data.

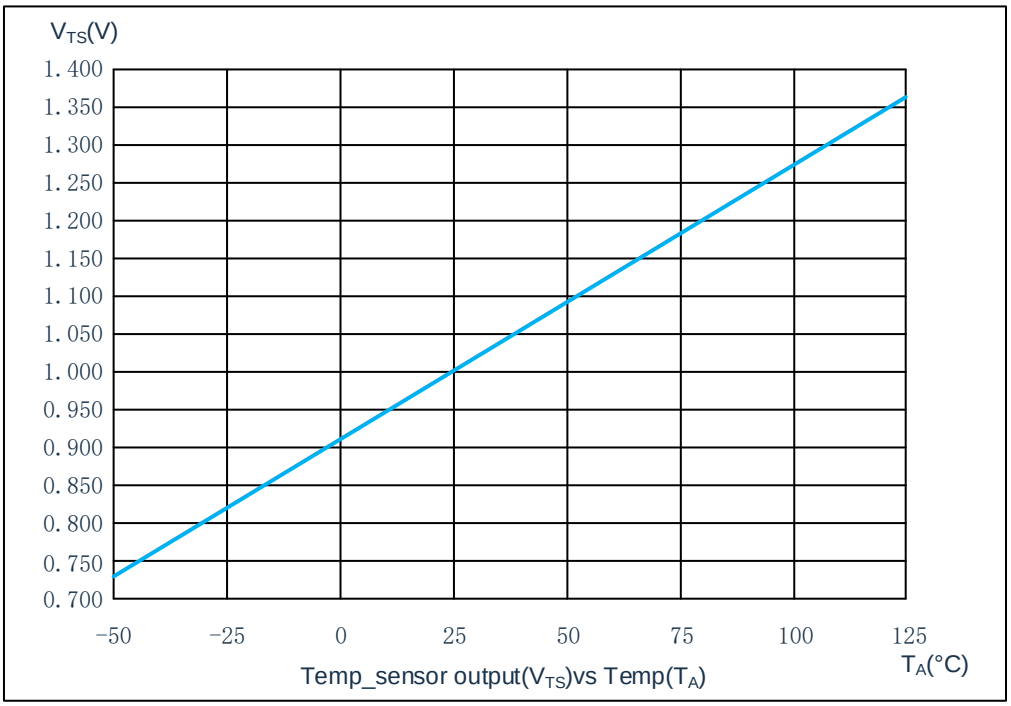
25.3.2 Functional Characteristic

The temperature sensor has the following characteristics:

1) When the temperature changes from -40 °C to 125 °C, the analog signal voltage range of the temperature sensor is 0.7V ~ 1.4V;
2) When the temperature changes from -40 °C to 125 °C, the slope of analog quantity changing with temperature is K:3.5±0.2 mV/℃;
3) When the temperature is 25 °C, the voltage is 1 ± 0.01V;
4) The analog signal quantity and temperature are approximately linearly changing, and the changing curve is shown in the figure below. In the figure below, T_A is the temperature (unit, ℃), and VTS is the output voltage of the temperature sensor (unit, V).

The variation curve of the analog output of the temperature sensor with temperature is shown in the following figure:

Temp_sensor output(V_{TS})vs Temp(T_A)								
T_A	−50	−25	0	25	50	75	100	125
V_{TS}	0.734	0.821	0.909	0.997	1.085	1.173	1.263	1.354



25.3.3 Calculation Formula

The conversion result of temperature sensor is:

$$RES_{AD}=\frac{V_{TS}}{V_{REF}}\times 4096$$

In the above formula, RES_{AD} is the measured 12-bit AD conversion value, V_{REF} is the reference voltage of ADC (unit: V), V_{TS} is the output voltage of analog signal (unit: V), and ΔV is the output voltage of analog signal.

The temperature (T) calculation reference formula is as follows:

$$T=(\frac{V_{REF}\times RES_{AD}}{4096}-0.909)\div K$$

In the above formula, K is the slope of analog quantity changing with temperature (unit, V/℃).

26. TOUCH

The touch module is an integrated circuit designed to realize the human touch interface, which can replace the mechanical light touch button to realize the operation interface of waterproof and dustproof, sealing and isolation, solid and beautiful.

technical parameter:

- Up to 30 touch buttons available.
- No external touch capacitor required.
- Effective touch response time is less than 100 ms.

26.1 Precautions For Using The Touch Module

- The ground wire of the touch button detection part should be separately connected to an independent ground, and another point is connected to the common ground of the whole machine.
- Avoid high voltage, high current, high frequency operation of the main board and touch circuit board overlap placement. If it is unavoidable, try to stay away from the high-voltage and high-current period area or add shielding on the main board.
- The connection between the Induction disc and the touch chip should be as short and thin as possible. If PCB process allows the use of 0.1mm line width as much as possible.
- The connection between the Induction disc and the touch chip should not cross the signal wire with strong interference and high frequency.
- Induction disc to touch the chip around the connection wire 0.5mm do not wiring other signal lines.

27. 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.

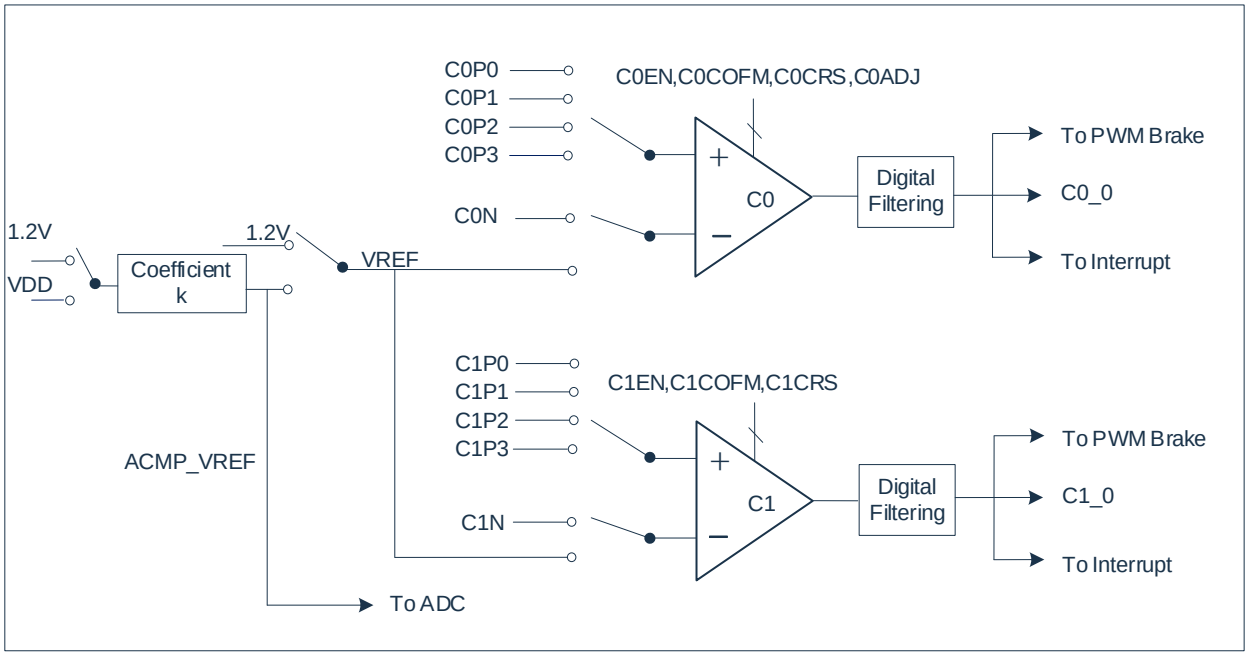
27.1 Comparator Characteristics

The comparator has the following characteristics:

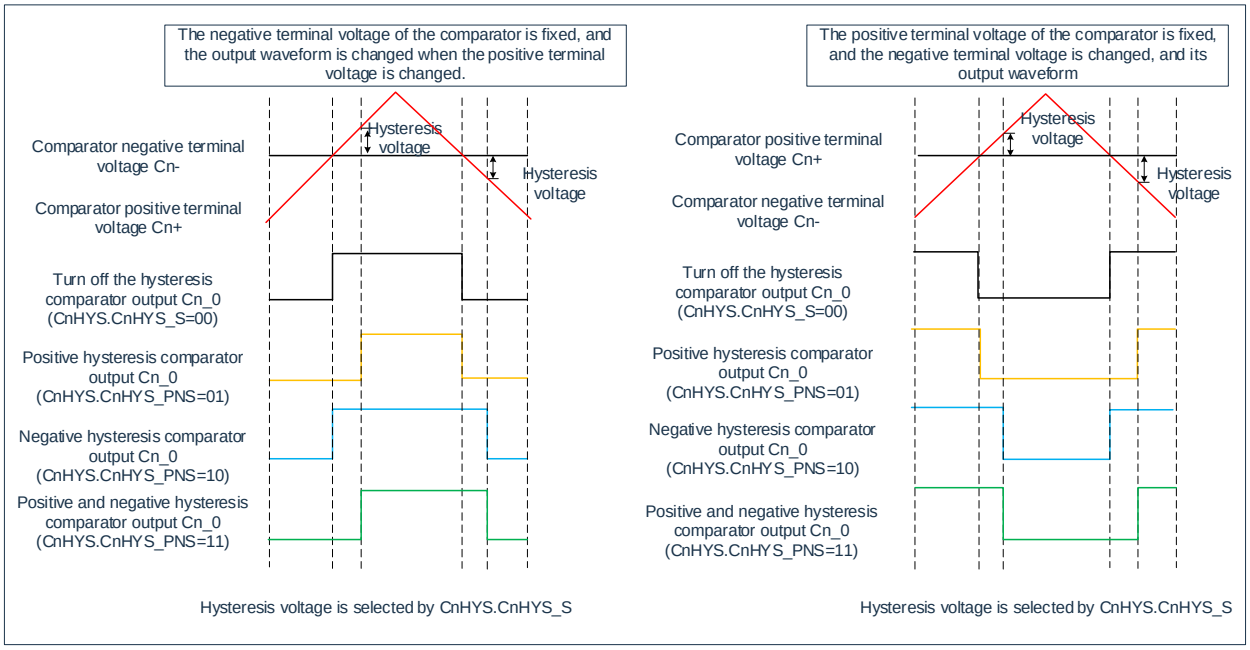
- The positive terminal can choose 4 ports inputs.
- The negative terminal can select port input and internal reference voltage VREF.
- The internal reference voltage can choose the internal Bandgap (1.2V) and ACMP_VREF output.
- ACMP_VREF reference source voltage divider range: k= (2/20) ~ (17/20), a total of 16 gear selections.
- Output filter is available and filter time is selectable: 0 ~ 512 * Tsys.
- Support unilateral (positive / negative) and bilateral (positive / negative) hysteresis control.
- Hysteresis voltage can be selected 10 / 20 / 60mV.
- Software supports offset voltage adjustment.
- Output could be used as brake signal of PWM
- The change of output could generate related interrupt

27.2 Comparator Structure

The structure diagram of the comparator is shown as follows:



The comparator hysteresis control block diagram is shown below:



27.3 Related Register

27.3.1 Comparator Control Register CnCON0

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CnCON0	CnEN	CnCOFM	CnN2G	CnNS1	CnNS0	CnPS2	CnPS1	CnPS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

C0CON0 address: F500H; C1CON0 address: F503H.

Bit7 CnEN: Comparator n enable;

1= Enable;

0= Disable.

Bit6 CnCOFM: comparator n adjustment mode enable;

1= Enable;

0= Disable.

Bit5 CnN2G: Comparator n adjustment mode negative terminal grounding enable;

(This bit is valid when CnCRS=0)

1= Negative terminal channel is closed, negative terminal of the comparator connects inner grounding.

0= Negative terminal channel is enabled, signal input from negative terminal.

Bit4~Bit3 CnNS<1:0>: Comparator n negative terminal channel select;

00= Comparator n negative terminal;

01= Internal voltage (Bandgap or ACMP_VREF);

1x= Reserved, disabling use.

Bit2~Bit0 CnPS<2:0>: Comparator n positive terminal channel select;

000= CnP0;

001= CnP1;

010= CnP2;

011= CnP3;

1xx= Reserved, disabling use.

27.3.2 Comparator Control Register C0CON1

F501H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
C0CON1	C0OUT	C0CRS	--	C0ADJ4	C0ADJ3	C0ADJ2	C0ADJ1	C0ADJ0
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

Bit7 C0OUT: Comparator 0 result (read only).

Bit6 C0CRS: Comparator 0 adjustment mode input port select;

1= The positive and negative terminal are connected together and input from the positive terminal;

0= The positive and negative terminal are connected together and input from the negative terminal.

Bit5 -- Reserved, must to be 0.

Bit4~ Bit0 C0ADJ<4:0>: Comparator 0 offset voltage adjust.

27.3.1 Comparator Control Register C1CON1

F504H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
C1CON1	C1OUT	--	--	--	--	--	--	--
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

Bit7 C1OUT: Comparator 1 result (read only).

Bit6 -- Reserved, must to be 0.

Bit5 -- Reserved, must to be 0.

Bit4~ Bit0 -- Reserved, disabling use.

27.3.2 Comparator Control Register CnCON2

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CnCON2	--	--	CnPOS	CnFE	CnFS3	CnFS2	CnFS1	CnFS0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

C0CON2 address:F502H; C1CON2 address:F505H.

Bit7~Bit6 -- Reserved, must to be 0.

Bit5 CnPOS: Comparator output polarity select (It may cause the interrupt flag to be set when switching);

1= Inverting output;

0= Normal output.

Bit4 CnFE: Comparator output filtering enable;

1= Enable;

0= Disable.

Bit3~Bit0 CnFS<3:0>: Comparator output filter time select.

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;

Other= (0~1)*Tsys.

27.3.3 Comparator Adjust Bit Select Register C0ADJE

F50AH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
C0ADJE	C0ADJE7	C0ADJE6	C0ADJE5	C0ADJE4	C0ADJE3	C0ADJE2	C0ADJE1	C0ADJE0
R/W	R/W	R/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 C0ADJE<7:0>: Comparator 0 offset voltage adjustment mode select;

AAH = Determined by C0ADJ<4:0> in the CnCON0 register;

Other= Determined by the CONFIG.

It is recommended to set the comparator parameters, and then start the comparator, otherwise there may be a situation in which the comparator output jumps due to false detection during the set process.

27.3.4 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, all must to be 0.

Bit3~Bit2 CnHYS_PNS<1:0> Positive and negative hysteresis selection bit;

00= Hysteresis closing;

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;

00= Hysteresis closing;

01= 10mV;

10= 20mV;

11= 60mV.

27.3.5 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, must to be 0.

Bit5 CNDIVS: ACMP_VREF reference voltage source selection;

1= Select 1.2V (Bandgap) for voltage division;

0= Select VDD for voltage division.

Bit4 CNSVR: The internal voltage VREF selection bit of the negative terminal of the comparator;

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 voltage source voltage division coefficient k selection;

0000-1111= 2/20 ~ 17/20.

27.3.6 Comparator Brake Control Register CNFBCON

F507H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CNFBCON	--	--	--	--	C1FBEN	C0FBEN	C1FBLS	C0FBLS
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~Bit4 -- Reserved, all must to be 0.

Bit3 C1FBEN: Comparator 1 output controlling PWM brake enable;

0= Disable;

1= Enable.

Bit2 C0FBEN: Comparator 0 output controlling PWM brake enable;

0= Disable;

1= Enable.

Bit1 C1FBLS: Comparator 1 output controlling PWM brake edge select;

0= Rising edge;

1= Falling edge.

Bit0 C0FBLS: Comparator 0 output controlling PWM brake edge select;

0= Rising edge;

1= Falling edge.

27.4 Comparator Interrupt

Both Comparator 0 and Comparator 1 can set interrupts, and both share an interrupt vector entry. After entering the interrupt service program, 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 relevant register bits.

27.4.1 Interrupt Priority Control Register EIP1

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

27.4.2 Comparator Interrupt Mask 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 to be 0.
Bit1 C1IE: Comparator 1 interrupt enable;
0= Disable
1= Enable.
Bit0 C0IE: Comparator 0 interrupt enable;
0= Disable;
1= Enable.

27.4.3 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 to be 0.

Bit1 C1IF: Comparator 1 interrupt flag (writing 0 to clear);

1= Compare 1 output changes.

0= --

Bit0 C0IF: Comparator 0 interrupt flag (writing 0 to clear);

1= Compare 0 output changes.

0= --

28. OP0/1

The chip contains two operational amplifier modules, OP0 and OP1, basic signal amplification and signal calculation functions can be realized with a small number of peripheral components.

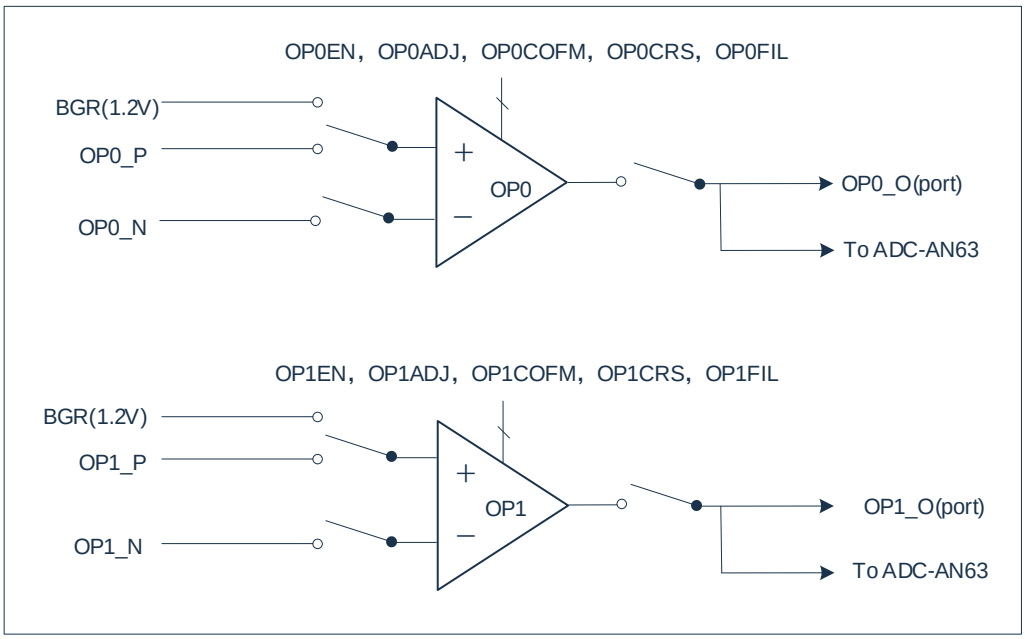
28.1 OP Amp Characteristics

Its characteristics are as follows:

- Each of the op amps is multiplexed with the GPIO port.
- The positive terminal supports internal 1.2V voltage input.
- Two modes of comparison / op-amp are supported.
- The output of the operational amplifier can be connected to the internal 63 channels of the ADC for measurement
- Software support offset voltage adjustment.

28.2 OP Amp Structure

The block diagram of the calculation structure is shown in the figure below:



28.3 Related Register

28.3.1 Op Amp Control Register OPnCON0

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OPnCON0	OPnEN	OPnCOFM	OPnFIL	OPnOS	OPnNS1	OPnNS0	OPnPS1	OPnPS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	1	0	0	0	0	0

OP0CON0 address: F520H; OP1CON0 address: F523H.

Bit7 OPnEN: OPn enable;
1= Enable;
0= Disable.

Bit6 OPnCOFM: OPn adjustment mode enable;
1= Enable;
0= Disable.

Bit5 OPnFIL: Pond working mode selection;
1= Operational amplifier mode (OPnCOFM must be 0);
0= Comparison mode (OPnCOFM must be 0).

Bit4 OPnOS: Opn output channel select;
1= OPn_O;
0= Disable.

Bit3~Bit2 OPnNS<3:2>: Opn negative channel select;
00= OPn_N;
Other= Disable.

Bit1~Bit0 OPnPS<1:0>: Opn positive channel select;
00= OPn_P;
01= BGR (1.2V);
Other= Disable.

28.3.2 Op Amp Control Register OPnCON1

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OPnCON1	OPnDOUT	OPnCRS	--	OPnADJ4	OPnADJ3	OPnADJ2	OPnADJ1	OPnADJ0
R/W	R	R/W	R	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	1	0	0	0	0

OP0CON1 address: F521H; OP1CON1 address: F524H.

Bit7 OPnDOUT: OPn adjustment result/comparison mode output, read only.

Bit6 OPnCRS: OPn adjustment mode input selects;
1= Positive input(only OPn_P can be selected);
0= Negative input.

Bit5 -- Reserved, must to be 0.

Bit4~Bit0 OPnADJ<4:0>: OPn offset voltage adjustment bits.

28.3.3 Op Amp Adjustment Bit Selection Register OPnADJE

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OPnADJE	OPnADJE7	OPnADJE6	OPnADJE5	OPnADJE4	OPnADJE3	OPnADJE2	OPnADJE1	OPnADJE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

OP0ADJE address: F526H; OP1ADJE address: F527H.

Bit7~Bit0 OPnADJE<7:0>: OPn offset voltage adjustment mode select.

AAH = Determined by OPnADJ<4:0> of the OPnCON1 register.

Other= Determined by the CONFIG.

29. Flash Memory

29.1 Overview

FLASH memory includes program memory (APROM/BOOT) and non-volatile data memory (Data FLASH). The maximum program memory space is 32KB, divided into 64 sectors, each sector contains 512B. The maximum 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, and the program space can also be CRC checked through the special function register (SFR). The SFR registers used to access the FLASH space are as follows:

- MLOCK
- MDATA
- MADRL
- MADRH
- PCRCDL
- PCRCDH
- MCTRL

The MLOCK register is used to enable the operation of the memory, the MDATA register forms a byte to store the 8-bit data to be read/written, the MADRL/MADRH register stores the address of the accessed MDATA unit or the address of the CRC check, PCRCDL/ PCRCDH the register is used to maintain the running result of the program CRC, and the MCTRL register is used for memory operation control.

Through the memory module interface, the memory can be read/written/erased. The memory allows byte reads and writes, and the write time is controlled by the on-chip timer. Before writing new data, make sure that the data in the address has been erased. The writing and erasing voltages are generated by an on-chip charge pump. The rated operating voltage of this charge pump is within the voltage range of the device for byte operations.

Flash memory erasing operation only supports sector erasing, not byte erasing. Before modifying the data of a certain address, it is recommended to save other data first, then erase the current sector, and finally write the data.

The chip supports the CRC check of the program space code. The check code is generated by using the polynomial CRC16-CCITT “$X^{16}+X^{12}+X^5+1$”.

29.2 Related Register

29.2.1 FLASH Protection Lock Register MLOCK

0xFB	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MLOCK	MLOCK7	MOCK6	MLOCK5	MLOCK4	MLOCK3	MLOCK2	MLOCK1	MLOCK0
R/W	W	W	W	W	W	W	W	W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 MLOCK<7:0>: Memory operation enable bit (this register only supports write operation, read as 00H);

AAH= Allow memory related R/W/E operations;

00H/FFH= Operation not allowed;

Other= Write is prohibited.

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

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

29.2.2 FLASH Memory Data Register MDATA

0xFE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MDATA	MDATA7	MDATA6	MDATA5	MDATA4	MDATA3	MDATA2	MDATA1	MDATA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 MDATA<7:0>: Program memory to read or write data.

29.2.3 FLASH Memory Low Address Register MADRL

0xFC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MADRL	MADRL7	MADRL6	MADRL5	MADRL4	MADRL3	MADRL2	MADRL1	MADRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	1	1	1	1	1	1

Bit7~Bit0 MADRL<7:0>: Specify the low 8-bit of the address for memory read/write operations.

29.2.4 FLASH Memory High 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 higr 8-bit of the address for memory read/write operations.

29.2.5 Program CRC Operation Result Data Register Low 8-bit PCRCDL

0xF9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCRCDL	PCRCD<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 PCRCD<7:0> The low 8-bit of the program CRC calculation result

29.2.6 Program CRC Operation Result Data Register High 8-bit PCRCDH

0xFA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCRCDH	PCRCD<15:8>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 PCRCD<15:8> The high 8-bit of the program CRC calculation result

29.2.7 FLASH Memory Control Register MCTRL

0xFF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MCTRL	--	--	MERR	MREG	MMODE1	MMODE0	CRCADR	MSTART
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	1	0	0	1	0	0	0	0

Bit7~Bit6 -- Reserved.

Bit5 MERR: Operation error flag (write 0 clear);

1= Before the programming operation, the write operation is immediately terminated when the data in the test programming address is not "FFH" (not erased).

0= ---

Bit4 MREG: Flash area selection bit;

1= Select data area (Low 10-bit address is valid);

0= Select program area (Low 15-bit address is valid).

Bit3~Bit2 MMODE<1:0>: Operating mode selection:

11= Erase mode (Range of erase operation: the entire sector where the current address is located);

10= Write mode;

01= CRC mode;

00= Read mode.

Bit1 CRCADR: Program CRC check address selection;

1= End address selection;

0= Start address selection.

Bit0 MSTART: Start operation control;

1= Start program memory R/W/E/CRC operation (Automatically cleared by hardware after the operation is completed);

0= Write: Terminate or not start program memory R/W/E/CRC operation;
Read: operation completed or operation not started.

Note: The CRCADR must be cleared after the CRC is finished

29.3 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.

The operation memory instruction must be followed by 6 NOP instructions, for example:

```
MOV MCTRL, #09H                ; Start of write operation
NOP
NOP
NOP
NOP
NOP
NOP
MOV MCTRL, #01H                ; Start of read operation
NOP
NOP
NOP
NOP
NOP
NOP
```

The program CRC check command is set by the register MCTRL [3:2], the start address and the end address can be freely configured through the register MADRL/MADRH, and the result is stored in the register PCRCDL/PCRCDH.

This CRC operation can only access the program storage space, but not the data storage space. In the program space CRC check process, the CPU stops working and waits for the CPU to continue running after the CRC calculation is completed. The CRC check is performed in byte mode, from the initial address to the end address. After the current CRC check is performed, the MMODE [1:0] =00 of MCTRL must be set. The CRC check operation steps are as follows:

1) Enable access to program memory registers:

   TA = 0xAA;

   TA = 0x55;

   MLOCK=0xAA; // The default value is 00

2) Check result before clearing program CRC:

   PCRCDL=0x00;

   PCRCDH=0x00;

3) Set the start and end addresses of the program CRC check:

   MCTRL [1] =0, set the starting address through MADRL/MADRH;

   MCTRL [1] =1, set the end address through MADRL/MADRH.

4) Start program CRC check command:

   MCTRL=0x05;

5) Wait for the end of the program CRC check:

   After the CRC check is complete, the MCTRL[0] hardware clears 0.

6) Read the program CRC check result:

PCRCDL stores the lower 8 bits data of the CRC operation result of the program;

PCRCDH stores the higher 8 bits data of the CRC operation result of the program.

The CRC check end address selection bit is cleared to 0.

7) MCTRL[1] is cleared to 0 by software after the CRC check.

30. Unique ID（UID）

30.1 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.

30.2 UID Register Description

UID0

F5E0H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID0	UID7	UID6	UID5	UID4	UID3	UID2	UID1	UID0
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 UID<7:0>

UID1

F5E1H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID1	UID15	UID14	UID13	UID12	UID11	UID10	UID9	UID8
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 UID<15:8>

UID2

F5E2H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID2	UID23	UID22	UID21	UID20	UID19	UID18	UID17	UID16
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 UID<23:16>

UID3

F5E3H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID3	UID31	UID30	UID29	UID28	UID27	UID26	UID25	UID24
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 UID<31:24>

UID4

F5E4H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID4	UID39	UID38	UID37	UID36	UID35	UID34	UID33	UID32
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 UID<39:32>

UID5

F5E5H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID5	UID47	UID46	UID45	UID44	UID43	UID42	UID41	UID40
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 UID<47:40>

UID6

F5E6H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID6	UID55	UID54	UID53	UID52	UID51	UID50	UID49	UID48
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 UID<55:48>

UID7

F5E7H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID7	UID63	UID62	UID61	UID60	UID59	UID58	UID57	UID56
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 UID<63:56>

UID8

F5E8H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID8	UID71	UID70	UID69	UID68	UID67	UID66	UID65	UID64
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 UID<71:64>

UID9

F5E9H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID9	UID79	UID78	UID77	UID76	UID75	UID74	UID73	UID72
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 UID<79:72>

UID10（0xF5EA）

F5EAH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID10	UID87	UID86	UID85	UID84	UID83	UID82	UID81	UID80
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 UID<87:80>

UID11

F5EBH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID11	UID95	UID94	UID93	UID92	UID91	UID90	UID89	UID88
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 UID<95:88>

31. User Configuration

The system configuration register (CONFIG) is the FLASH option for the MCU initial condition, cannot be accessed and operated by the program. It contains the following:

1. WDT（Watchdog working mode selection）
   - ENABLE — Forced to open WDT
   - SOFTWARE CONTROL（default） — WDT mode of operation is controlled by the WDTRE bit in the WDCON register.
2. PROTECT
   - ENABLE — FLASH The code is encrypted and the code read is 00H. And it is forbidden to enter debug 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, the value read by the emulator after encryption is 00H
4. LVR（Low voltage reset）
   - 1.8V（default）
   - 2.0V
   - 2.5V
   - 3.5V
5. DEBUG（Debug mode）
   - DISABLE（default） — The debugging mode is forbidden, DSCK, DSDA pins are used as ordinary IO ports
   - ENABLE — Debug mode is enabled, DSDA and DSCK are configured as debug ports, and other functions corresponding to the pins are turned off.
6. OSC（Oscillation mode）
   - HSI（default） — 48MHz
   - HSE
   - LSE(32.768KHz)
   - LSI(125KHz) — 125KHz
7. HSELSE_SEL Selection of Oscillation Mode of Crystal Oscillator
   - HSE
   - LSE
8. SYS_PRESCALE（System clock prescaler selection）
   - F_{OSC}/1（default） — （The system clock selects HSI, and if HSI_FS=F_{HSI}/1, then F_{CPU}=F_{HSI}/2）
   - F_{OSC}/2
   - F_{OSC}/4
   - F_{OSC}/8
9. 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
10. EXT_RESET（External reset configuration）
    - DISABLE（default） — External reset disabled
    - ENABLE — External reset enable

- ENABLE(OPEN PULLUP) External reset enable and open reset internal pull-up resistor

11. WAKE UP_WAIT TIME（Sleep and wake-up and wait for the oscillator to stabilize the default time is 1.0s）
    - 50us
    - 100us
    - 500us
    - 1ms
    - 5ms
    - 10ms
    - 500ms
    - 1.0s（default）
12. CPU_WAITCLOCK（Memory wait clock selection）
    - 1*System Clock (1T)（default）
    - 2*System Clock (2T)
    - 3*System Clock (3T)
    - 4*System Clock (4T)
    - 5*System Clock (5T)
    - 6*System Clock (6T)
    - 7*System Clock (7T)
    - 8*System Clock (8T)
13. WRITE_PROTECT Program partition protection (protectable interval, all default intervals are not protected)
    - 0000H-07FFH（Protect/unprotect） 4000H-47FFH（Protect/unprotect）
    - 0800H-0FFFH（Protect/unprotect） 4800H-4FFFH（Protect/unprotect）
    - 1000H-17FFH（Protect/unprotect） 5000H-57FFH（Protect/unprotect）
    - 1800H-1FFFH（Protect/unprotect） 5800H-5FFFH（Protect/unprotect）
    - 2000H-27FFH（Protect/unprotect） 6000H-67FFH（Protect/unprotect）
    - 2800H-2FFFH（Protect/unprotect） 6800H-6FFFH（Protect/unprotect）
    - 3000H-37FFH（Protect/unprotect） 7000H-77FFH（Protect/unprotect）
    - 3800H-3FFFH（Protect/unprotect） 7800H-7FFFH（Protect/unprotect）
14. BOOT space selection
    - BOOT_DIS（default） Prohibited in BOOT area
    - BOOT_1K 1K space in BOOT area
    - BOOT_2K 2K space in BOOT area
    - BOOT_4K 4K space in BOOT area

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=TSYS/2.
3) WRITE_PROTECT program partition protection is effective for Data Flash when setting 0000H-07FFH interval protection.

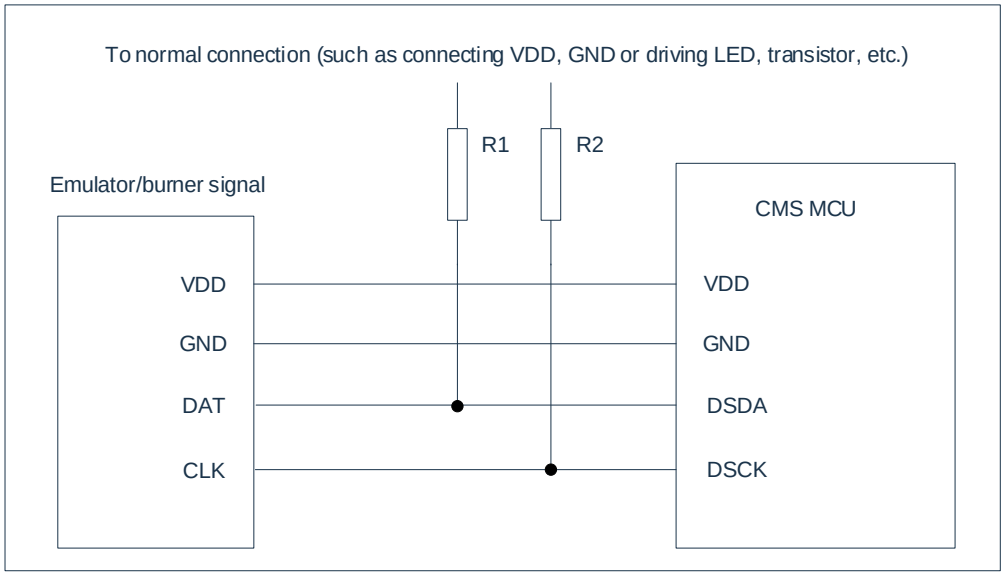
32. Online Programming And Debugging

32.1 Online Programming Mode

The chip can be serially programmed in the final application circuit. Programming can be done simply through the following 4 wires:

- VDD
- GND
- DAT
- CLK

Online 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. The 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.

32.2 Online Debug Mode

The chip supports 2-wire (DSCK, DSDA) online debugging function. If you use the online 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, the DSCK and DSDA ports 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 and then test 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, Timer1/4 will continue to run in the pause state. Peripherals that continue to run in the paused state may generate interrupts, so you need to pay attention when debugging.
- It is recommended not to use WDT reset and software reset function in debugging state, because the chip may lose connection with the debugger during resetting.

33. Instruction Description

Assembly instructions include 5 categories: Arithmetic operations, logic operations, data transfer operations, Boolean operations and program branch instructions,

All of these instructions are compatible with standard 8051.

33.1 Symbol Description

Symbol	Description
Rn	Working register R0-R7
Direct	The unit address of the internal data memory RAM (00H-FFH) or the address in the special function register SFR
@Ri	Indirect internal or external RAM location addressed by register (@R0 or @R1)
#data	8-bit binary constant
#data16	16-bit binary constant in the instruction
Bit	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

33.2 List Of Instruction

Instruction		Description
Operation type		
ADD	A,Rn	Add register to accumulator.
ADD	A,direct	Add directly addressed data to accumulator.
ADD	A,@Ri	Add indirectly addressed data to accumulator.
ADD	A,#data	Add immediate data to accumulator.
ADDC	A,Rn	Add register to accumulate or with carry.
ADDC	A,direct	Add directly addressed data to accumulator with carry.
ADDC	A,@Ri	Add indirectly addressed data to accumulator with carry.
ADDC	A,#data	Add immediate data to accumulator with carry.
SUBB	A,Rn	Subtract register from accumulator with borrow.
SUBB	A,direct	Subtract directly addressed data from accumulate or with borrow.
SUBB	A,@Ri	Subtract directly addressed data from accumulator with borrow.
SUBB	A,#data	Subtract mediate data from accumulate or with borrow.
INC	A	Increment accumulator.
INC	Rn	Increment register.
INC	direct	Increment directly addressed location.
INC	@Ri	Increment indirectly addressed location.
INC	DPTR	Increment data pointer.
DEC	A	Decrement accumulator.
DEC	Rn	Decrement register.
DEC	direct	Decrement directly addressed location.
DEC	@Ri	Increment indirectly addressed location.
MUL	A,B	Multiply A and B.
DIV	A,B	Divide A by B.
DA	A	Decimally adjust accumulator.
Logical operation type		
ANL	A,Rn	AND register to accumulator.
ANL	A,direct	AND directly addressed data to accumulator.
ANL	A,@Ri	AND indirectly addressed data to accumulator.
ANL	A,#data	AND immediate data to accumulator.
ANL	direct,A	AND accumulator to directly addressed location.
ANL	direct,#data	AND immediate data to directly addressed location.
ORL	A,Rn	OR register to accumulator.
ORL	A, direct	OR directly addressed data to accumulator.
ORL	A,@Ri	OR indirectly addressed data to accumulator.
ORL	A, #data	OR immediate data to accumulator.
ORL	direct,A	OR accumulator to directly addressed location.
ORL	direct,#data	OR immediate data to directly addressed location.
XRL	A,Rn	Exclusive OR (XOR) register to accumulator.
XRL	A,direct	XOR directly addressed data to accumulator.
XRL	A,@Ri	XOR indirectly addressed data to accumulator.
XRL	A,#data	XOR immediate data to accumulator.
XRL	direct,A	XOR accumulator to directly addressed location.
XRL	direct,#data	XOR immediate data directly addressed location.

Instruction		Description
CLR	A	Clear accumulator.
CPL	A	Complement accumulator.
RL	A	Rotate accumulator left.
RLC	A	Rotate accumulator left through carry.
RR	A	Rotate accumulator right.
RRC	A	Rotate accumulator right through carry.
SWAP	A	Swap nibbles within the accumulator.
Data transmission type		
MOV	A,Rn	Move register to accumulator.
MOV	A,direct	Move directly addressed data to accumulator.
MOV	A,@Ri	Move indirectly addressed data to accumulator.
MOV	A,#data	Move immediate data to accumulator.
MOV	Rn,A	Move accumulator to register.
MOV	Rn,direct	Move directly addressed data to register.
MOV	Rn,#data	Move immediate data to register.
MOV	direct,A	Move accumulator to direct.
MOV	direct,Rn	Move register to direct.
MOV	direct1,direct2	Move directly addressed data to directly addressed location.
MOV	direct,@Ri	Move indirectly addressed data to directly addressed location.
MOV	direct,#data	Move immediate data to directly addressed location.
MOV	@Ri,A	Move accumulator to indirectly addressed location.
MOV	@Ri,direct	Move directly addressed data to indirectly addressed location.
MOV	@Ri,#data	Move immediate data to indirectly addressed location.
MOV	DPTR,#data16	Load data pointer with a 16-bitimmediate.
MOVC	A,@A+DPTR	Load accumulator with a code byte relative to DPTR.
MOVC	A,@A+PC	Load accumulator with a code byte relative to PC.
MOVX	A,@Ri	Move external RAM(8-bitaddress) to accumulator.
MOVX	A,@DPTR	Move external RAM(16-bitaddress) to accumulator.
MOVX	@Ri,A	Move accumulator to external RAM (8-bitaddress).
MOVX	@DPTR,A	Move accumulator to external RAM(16-bitaddress).
PUSH	direct	Push directly addressed data onto stack.
POP	direct	Pop directly addressed data location from stack.
XCH	A,Rn	Exchange register with accumulator.
XCH	A, direct	Exchange directly addressed location with accumulator.
XCH	A,@Ri	Exchange indirect RAM with accumulator.
XCHD	A,@Ri	Exchange low-order nibbles of indirect and accumulator.
Boolean type		
CLR	C	Clear carry flag.
CLR	bit	Clear directly addressed bit.
SETB	C	Set carry flag.
SETB	bit	Set directly addressed bit.
CPL	C	Complement carry flag.
CPL	bit	Complement directly addressed bit.
ANL	C,bit	AND directly addressed bit to carry flag.
ANL	C,/bit	AND complement of directly addressed bit to carry.
ORL	C,bit	OR directly addressed bit to carry flag.

Instruction		Description
ORL	C,/bit	OR complement of directly addressed bit to carry.
MOV	C,bit	Move directly addressed bit to carry flag.
MOV	bit,C	Move carry flag to directly addressed bit.
Program jump type		
ACALL	addr11	Absolute subroutine call in 2k address range
LCALL	addr16	Long subroutine call in 64k address range
RET		Return from subroutine
RETI		Return from interrupt
AJMP	addr11	Absolute jump in 2k address range
LJMP	addr16	Long jump in 64k address range
SJMP	rel	Short jump (relative address)
JMP	@A+DPTR	Jump indirect relative to the DPTR
JZ	rel	Jump if accumulator is zero
JNZ	rel	Jump if accumulator is not zero
JC	rel	Jump if carry flag is set
JNC	rel	Jump if carry flag is not set
JB	bit,rel	Jump if directly addressed bit is set
JNB	bit,rel	Jump if directly addressed bit is not set
JBC	bit,rel	Jump if directly addressed bit is set and clear bit
CJNE	A,direct,rel	Compare directly addressed data to accumulator and jump if not equal
CJNE	A,#data,rel	Compare immediate data to accumulator and jump if not equal
CJNE	Rn,#data,rel	Compare immediate data to register and jump if not equal
CJNE	@Ri,#data,rel	Compare immediate to indirect and jump if not equal
DJNZ	Rn,rel	Decrement register and jump if not zero
DJNZ	direct,rel	Decrement directly addressed location and jump if not zero
NOP		No operation for one cycle
Read-modify-write instruction （Read-Modify-Write）		
ANL		Logical AND. (ANL direct, A and ANL direct, #data)
ORL		Logical OR. (ORL direct, A and ORL direct, #data)
XRL		Logical exclusive OR. (XRL direct, A and XRL direct, #data)
JBC		Jump if bit = 1 and clear it. (JBC bit, rel)
CPL		Complement bit. (CPL bit)
INC		Increment. (INC direct)
DEC		Decrement. (DEC direct)
DJNZ		Decrement and jump if not zero. (DJNZ direct, rel)
MOV	bit,C	Move carry to bit. (MOV bit, C)
CLR	bit	Clear bit. (CLR bit)
SETB	bit	Set bit. (SETB bit)

34. Revision History

Revision	Date	Modify content
V1.00	October 2019	initial version
V1.01	June 2020	Update font and temperature sensor chapter description
V1.02	February 2021	Modify the title of 2.3 and reset value of some registers, and delete the default GPIO function of PxnCFG
V1.03	June 2021	Delete CMS80F251x series description
V1.05	February 2023	Adjust ADC clock frequency example, adjust I2C description, delete FLASH operation time related description, adjust port multiplexing function description, add BUZZER precautions, change some register bit description, optimize some text expression, correct conversion clock parameters
V1.0.6	July 2023	1) Added some remarks to section 29.2.7 2) Added the step "Clear the CRC end address select bit" to the function description in section 29.3