CMS8S5880 User Manual

Enhanced Flash 8 bit 1T 8051- Microcontroller

Rev. 1.3.7

Please note the following CMS IP policy

*Zhongwei Semiconductor (Shenzhen) Co., Ltd. (hereinafter referred to as the Company) has applied for a patent and enjoys absolute legal rights and interests. The patent rights related to the Company's MCUs or other products have not been authorized to be licensed, and any company, organization or individual who infringes the Company's patent rights through improper means will take all possible legal actions to curb the infringer's improper infringement and recover the losses suffered by the Company as a result of the infringement or the illegal benefits obtained by the infringer.

＊The name and logo of Zhongwei Semiconductor (Shenzhen) Co., Ltd. are registered trademarks of the Company.

＊The Company reserves the right to further explain the reliability, functionality and design improvements of the products in the data sheet. However, the Company is not responsible for the use of the Specification Contents. The applications mentioned herein are for illustrative purposes only and the Company does not warrant and does not represent that these applications can be applied without further modification, nor does it recommend that its products be used in places that may cause harm to persons due to malfunction or other reasons. The Company's products are not licensed for lifesaving, life-sustaining devices or systems as critical devices. The Company reserves the right to modify the product without prior notice, please refer to the official website www.mcu.com.cn for the latest information.

Contents

1. PRODUCT DESCRIPTION

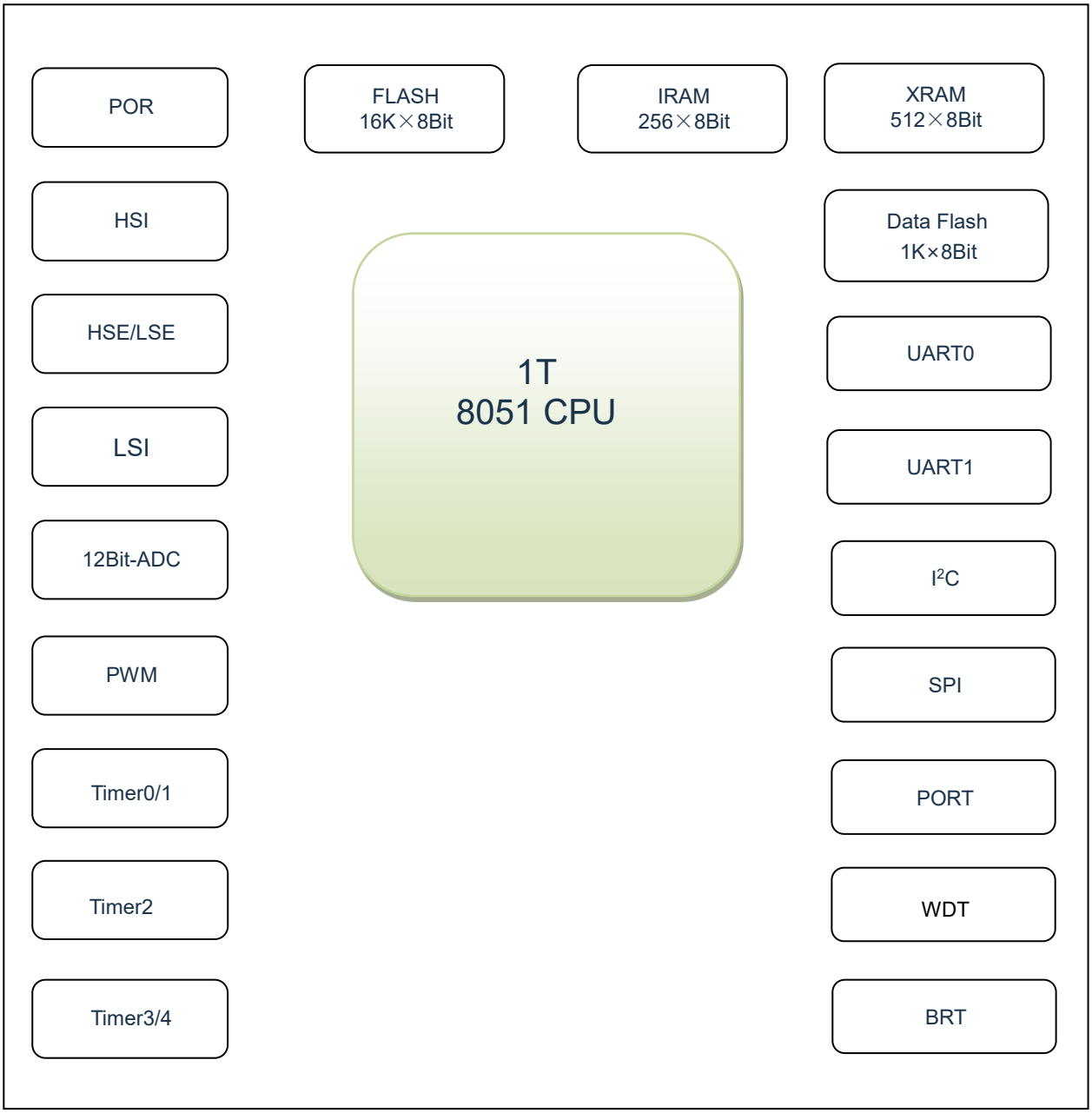
1.1 Functional Characteristics

- **1T command system compatible with MCT-51**
  - Support up to 24MHz main frequency 1T instruction operation
- **RAM**
  - Program FLASH: 16K×8Bit
  - Data FLASH: 1K×8Bit
  - General RAM: 256×8Bit
  - General XRAM: 512×8Bit
- **4 kinds of oscillation modes**
  - HSI-Internal RC oscillation: 8MHz/16MHz /24MHz
  - HSE-External crystal oscillation: 8MHz/16MHz
  - LSE- External crystal oscillation: 32.768KHz
  - LSI-Internal low power oscillation: 125KHz
- **Low Voltage Reset (LVR)**
  - 1.9V/2.1V/2.6V/3.5V
- **GPIO**
  - Up to 18 GPIOs
  - All support up/down resistor function
  - All support edge interrupt
  - (rising/falling/double edge)
  - All support wake-up function
- **Interrupt source**
  - - Supports all external port interrupts
  - - 5 timer interrupts
  - - Other peripheral interrupts
- **Timer**
  - WDT timer (watchdog timer)
  - Timer0/1, Timer2, Timer3/4
  - BRT (standalone serial port baud rate clock generator)
  - WKT (dedicated wake-up timer)
- **Range of operating voltage**
  - 2.1V-4.5V@F_{SYS}=24MHz
- **Range of working temperature**
  - -40℃～105℃
- **Communication module**
  - 1x SPI
  - 1x I^2C
  - 2x UART (UART0 and UART1)
- **Buzzer driver**
  - 50% duty cycle, frequency can be set freely
- **Enhanced EPWM**
  - 6-channel enhanced PWM
  - 6 mutually independent cycle counters
  - Support independent/complementary/synchronous/group mode
  - Support edge-aligned/center-aligned mode
  - Support complementary mode dead time delay function
  - Support mask function and brake function
- **High precision 12-bit ADC**
  - All GPIOs (18I/Os) support AD channels Internal 1.2V reference voltage can be detected
  - Support hardware triggered start conversion function
  - Support a set of result digital comparison function
- **Low power mode**
  - Idle mode (IDLE)
  - Sleep mode (STOP)
- **Support 96-bit unique ID number (UID)**
  - Each chip has an individual ID number

Model Description:

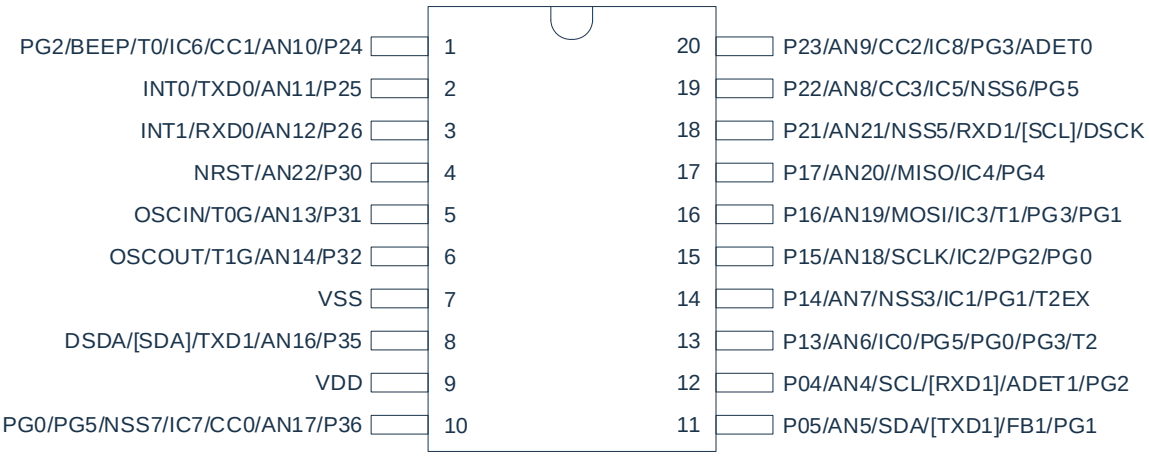
PRODUCT	FLASH	RAM	XRAM	DataFlash	I/O	ADC	PACKAGE
CMS8S5880-TSSOP20	16K×8	256×8	512×8	1K×8	18	12Bit×18	TSSOP20
CMS8S5880-QFN20	16K×8	256×8	512×8	1K×8	18	12Bit×18	QFN20

1.2 System Block Diagram



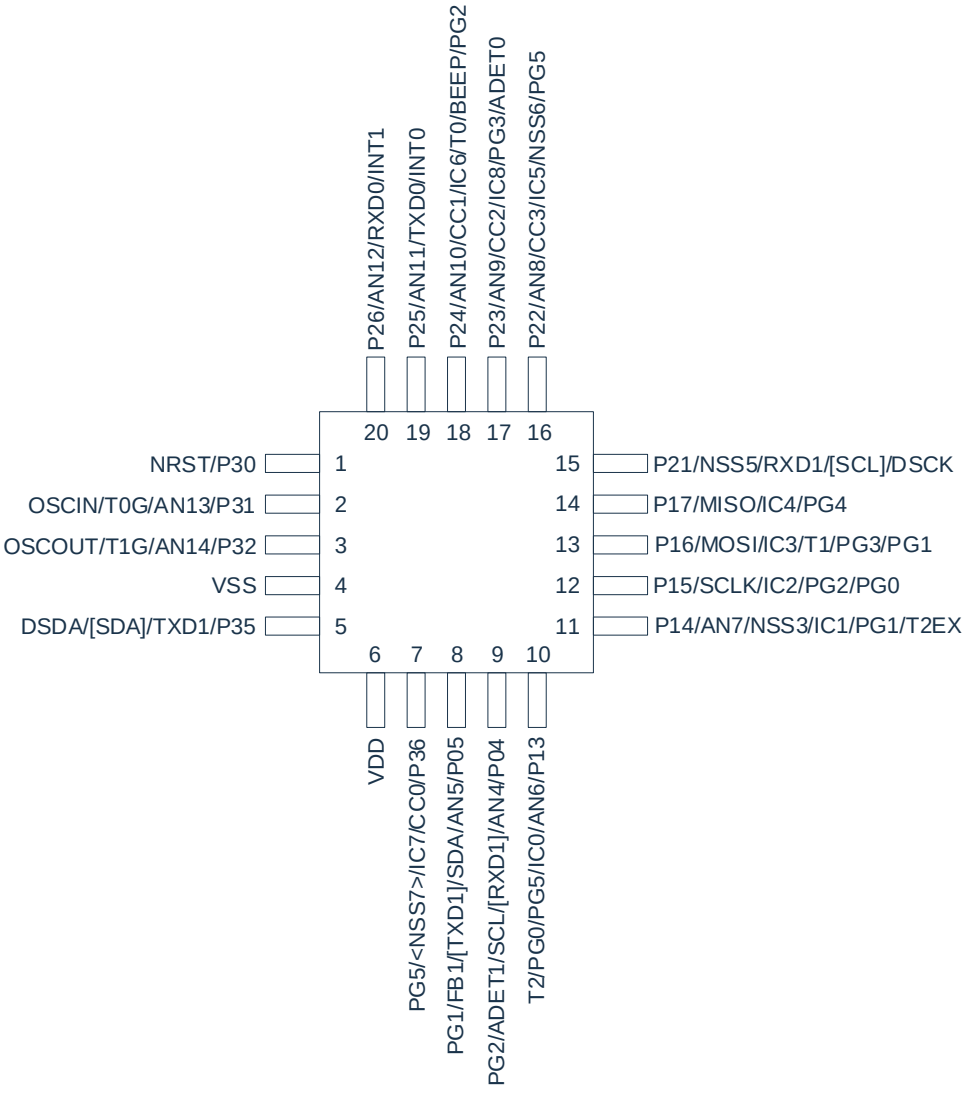
1.3 Pin Distribution

1.3.1 CMS8S5880-TSSOP20 Pin map



CMS8S5880-TSSOP20

1.3.2 CMS8S5880-QFN20 Pin map



CMS8S5880-QFN20

Pin description:

Pin name	Function	I/O Types	Description
P04/AN4/SCL/ADET1//[RXD1]/PG2	P04	I/O	GPIO configures input and output, pull-up and pull-down functions through registers
	AN4	I	ADC channel 4 input
	SCL	I/O	I^2C clock input and output
	ADET1	I	ADC external trigger input 1
	[RXD1]	I	UART1 input
	PG2	O	PWM2 output
P05/AN5/SDA/FB1/PG1/[TXD1]	P05	I/O	GPIO configures input and output, pull-up and pull-down through registers
	AN5	I	ADC channel 5 input
	SDA	I/O	I^2 C data input and output
	FB1	I	PWM brake external input 1
	PG1	O	PWM1 output
	[TXD1]	O	UART1 output
P13/AN6/IC0/PG5/PG0/PG3/T2	P13	I/O	GPIO configures input and output, pull-up and pull-down through registers
	AN6	I	ADC channel 6 input
	IC0	I	Timer2 captures input signal 0
	PG5	O	PWM5 output
	PG0	O	PWM0 output
	PG3	O	PWM3 output
	T2	I	Timer2 external event or gating input
P14/AN7/NSS3/IC1/PG1/T2EX	P14	I/O	GPIO configures input and output, pull-up and pull-down through registers
	AN7	I	ADC channel 7 input
	NSS3	O	SPI master mode chip select output signal 3
	IC1	I	Timer2 captures input signal 1
	PG1	O	PWM1 output
	T2EX	I	Timer2 falling edge auto-reload input
P15/AN18/SCLK/IC2/PG2/PG0	P15	I/O	GPIO configures input and output, pull-up and pull-down through registers
	AN18	I	ADC channel 18 input
	SCLK	I/O	SPI clock input and output pins
	IC2	I	Timer2 captures input signal 2
	PG2	O	PWM2 output
	PG0	O	PWM0 output
P16/AN19/MOSI/IC3/T1/PG3/PG1	P16	I/O	GPIO configures input and output, pull-up and pull-down through registers
	AN19	I	ADC channel 19 input
	MOSI	I/O	SPI master sends slave receives data
	IC3	I	Timer2 captures input signal 3
	T1	I	Timer1 external clock input
	PG3	O	PWM3 output
	PG1	O	PWM1 output
P17/AN20/MISO/IC4/PG4	P17	I/O	GPIO configures input and output, pull-up and pull-down through registers
	AN20	I	ADC channel 20 input
	MISO	I/O	SPI master receives data sent by slave
	IC4	I	Timer2 captures input signal 4

Pin name	Function	I/O Types	Description
	PG4	O	PWM4 output
P21/AN21/NSS5/RXD1/[SCL]/ DSCK	P21	I/O	GPIO configures input and output, pull-up and pull-down through registers
	AN21	I	ADC channel 21 input
	NSS5	O	SPI master mode chip select output signal 5
	RXD1	I	UART1 input
	[SCL]	I/O	I^2C clock input and output
	DSCK	I/O	Programming, debugging clock inputs and outputs
P22/AN8/CC3/IC5/NSS6/ PG5	P22	I/O	GPIO configures input and output, pull-up and pull-down through registers
	AN8	I	ADC channel 8 input
	CC3	O	Timer2 comparison output signal 3
	IC5	I	Timer2 captures input signal 5
	NSS6	O	SPI master mode chip select output signal 6
	PG5	O	PWM5 output
P23/AN9/CC2/IC8/PG3/ ADET0/	P23	I/O	GPIO configures input and output, pull-up and pull-down through registers
	AN9	I	ADC channel 9 input
	CC2	O	Timer2 comparison output signal 2
	IC8	I	Timer2 captures input signal 8
	PG3	O	PWM3 output
	ADET0	I	ADC external trigger input signal 0
P24/AN10/CC1/IC6/T0/BEEP/ PG2	P24	I/O	GPIO configures input and output, pull-up and pull-down through registers
	AN10	I	ADC channel 10 input
	CC1	O	Timer2 comparison output signal 1
	IC6	I	Timer2 captures input signal 6
	T0	I	Timer0 external clock input
	BEEP	O	Buzzer output
	PG2	O	PWM2 output
P25/AN11/TXD0/INT0	P25	I/O	GPIO configures input and output, pull-up and pull-down through registers
	AN11	I	ADC channel 11 input
	TXD0	O	UART0 output
	INT0	I	External interrupt 0 input
P26/AN12/RXD0/INT1	P26	I/O	GPIO, configure input and output through registers, pull-up and pull-down functions
	AN12	I	ADC channel 12 input
	RXD0	I	UART0 input
	INT1	I	External interrupt 1 input
P30/AN22/NRST	P30	I/O	GPIO configures input and output, pull-up and pull-down through registers
	AN22	I	ADC channel 22 input
	NRST	I	External reset input
P31/AN13/T0G/OSCIN	P31	I/O	GPIO configures input and output, pull-up and pull-down through registers
	AN13	I	ADC channel 13 input
	T0G	I	Timer0 gating input
	OSCIN	I	External oscillation input
P32/AN14/T1G/OSCOUT	P32	I/O	GPIO configures input and output, pull-up and pull-down through registers
	AN14	I	ADC channel 14 input

Pin name	Function	I/O Types	Description
	T1G	I	Timer1 gating input
	OSCOUT	O	External oscillation output
P35/AN16/TXD1/[SDA]/DSDA	P35	I/O	GPIO configures input and output, pull-up and pull-down through registers
	AN16	I	ADC channel 16 input
	TXD1	O	UART1 output
	[SDA]	I/O	I^2C data input and output
	DSDA	I/O	Programming and debugging data input and output
P36/AN17/CC0/IC7/NSS7/PG5/PG0	P36	I/O	GPIO configures input and output, pull-up and pull-down through registers
	AN17	I	ADC channel 17 input
	CC0	O	Timer2 compare output signal 0
	IC7	I	Timer2 captures the input signal7
	NSS7	I/O	SPI master chip select output slave chip select input
	PG5	O	PWM5 output
	PG0	O	PWM0 output
VDD		P	Power supply voltage input pin
VSS		P	Ground pin

1.4 System Configuration Register

The system configuration register (CONFIG) is the FLASH option for the MCU initial condition. It can only be programmed by the CMS writer and cannot be accessed and operated by the user. It contains the following:

1. WDT (Watchdog operating mode selection)
   - ENABLE — Forced to enable WDT
   - SOFTWARE CONTROL — WDT operating mode is controlled by the WDTRE bit in the WDCON register.
2. PROTECT (encrypt)
   - ENABLE — The Flash code is encrypted and read as 00H. And it is forbidden to enter debug mode
   - DISABLE — The Flash code is not encrypted
3. FLASH_DATA_PROTECT (encrypt)
   - DISABLE — The FLASH data area is not encrypted
   - ENABLE — The Flash data area is encrypted, the value read by the emulator after encryption is 00H
4. LVR (Low voltage reset)
   - 1.9V
   - 2.1V
   - 2.6V
   - 3.5V
5. DEBUG (Debug mode)
   - DISABLE — Debug mode is forbidden, P21 and P35 are used as ordinary IO ports
   - ENABLE — Debug mode is enabled, P21 and P35 are configured as debug ports (DSCK,DSDA), and other functions corresponding to the pins are turned off.
6. OSC (Oscillation mode)
   - HSI
   - HSE — OSCIN and OSCOUT are configured as oscillation ports
   - LSE(32.768KHz) — OSCIN and OSCOUT are configured as oscillation ports
   - LSI(125KHz)
7. OSC_PRESCALE (Oscillation output prescaler selection)
   - Fosc/1, Fosc/2, Fosc/4, Fosc/8
8. HSI_FS (Internal RC oscillator frequency selection)
   - 8MHz
   - 16MHz
   - 24MHz
9. EXT_RESET (External reset configuration)
   - DISABLE — External reset disabled
   - ENABLE — External reset enable
   - ENABLE(OPEN PULLUP) — External reset enable and open internal pull-up resistor at the reset port
10. WAKE UP_WAIT TIME (the default waiting time of wake up after sleep mode is 1.2s)
    - 50us
    - 5ms
    - 100us
    - 10ms
    - 500us
    - 500ms
    - 1ms
    - 1.2s

1.5 Online Serial Programming

The microcontroller can be serially programmed in the final application circuit. Programming can be simply done by the following four lines:

- Power line
- Ground line
- Data line
- Clock line

This allows the user to fabricate circuit boards using unprogrammed devices and just to program the microcontroller before the product is delivered. Thereby the latest version of firmware or custom firmware can be programmed into the microcontroller.

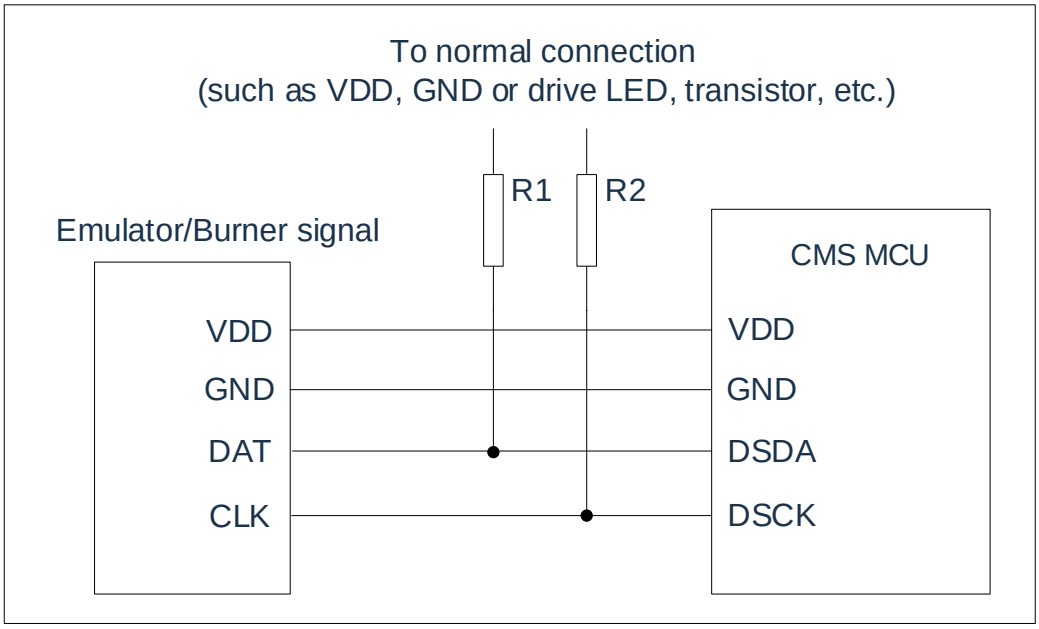


Figure 1-1: Typical online serial programming connection method

In the above figure, R1 and R2 are electrically isolated devices, which are often replaced by resistors. The resistance values are as follows: R1≥4.7K, R2≥4.7K.

See the <**错误!未找到引用源。**> section for details.

2. CENTRAL PROCESSING UNIT (CPU)

2.1 Memory

2.1.1 Program Memory

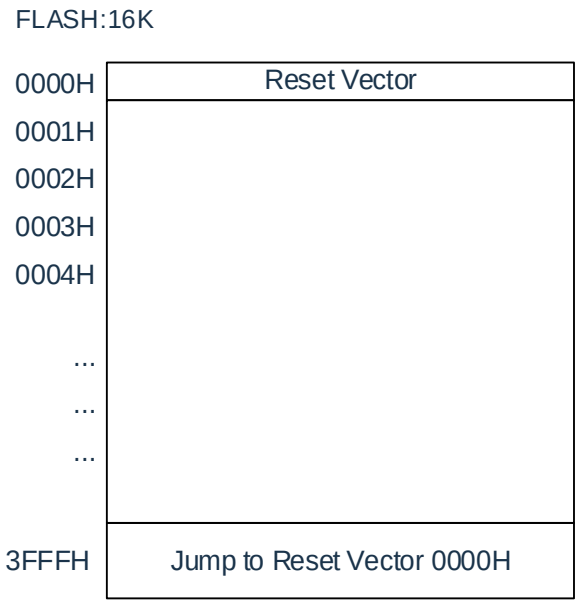


Figure 2-1: Program Memory Space

A 16-bit program counter can reach up to addresses of 64K bytes, But this chip can only store16K bytes.

The above figure shows the lower area of the program memory. After reset, the CPU starts at 0000H. Each interrupt is assigned to a fixed address interrupt in the program memory, which leads the CPU jumps to the address to start excuting the service program, such as external interrupt 1, the assigned address of which is 000BH. If using external interrupt 1, its service program must start from the address of 0013H. If the interrupt is not used, its service address is used as a normal program storage address

2.1.2 Reset Vector (0000H)

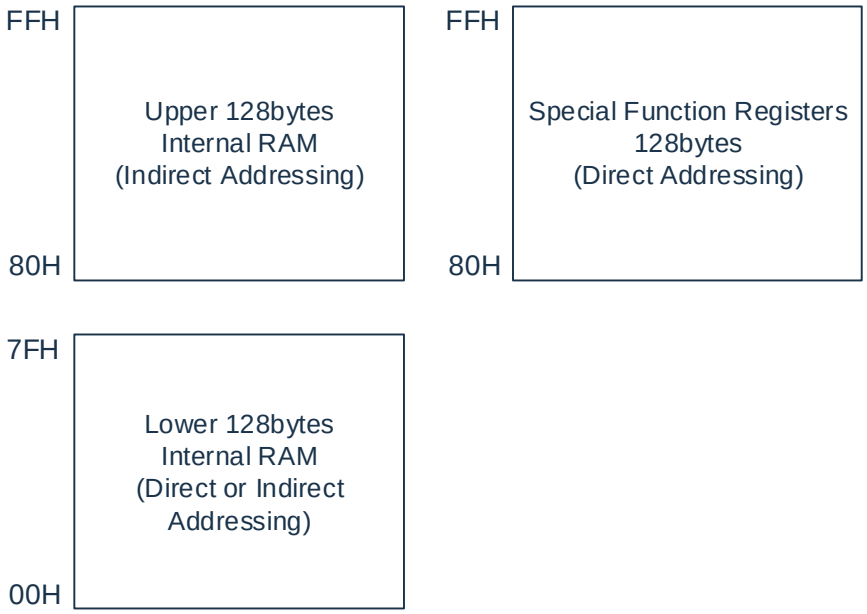
Single chip microcomputer posses a system reset vector with a word length (0000H).

After the reset occurs, the program will start again from 0000H, and the system registers will be recover to the default values. The following program demonstrates show 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
```

2.1.3 Data Memory (IRAM)



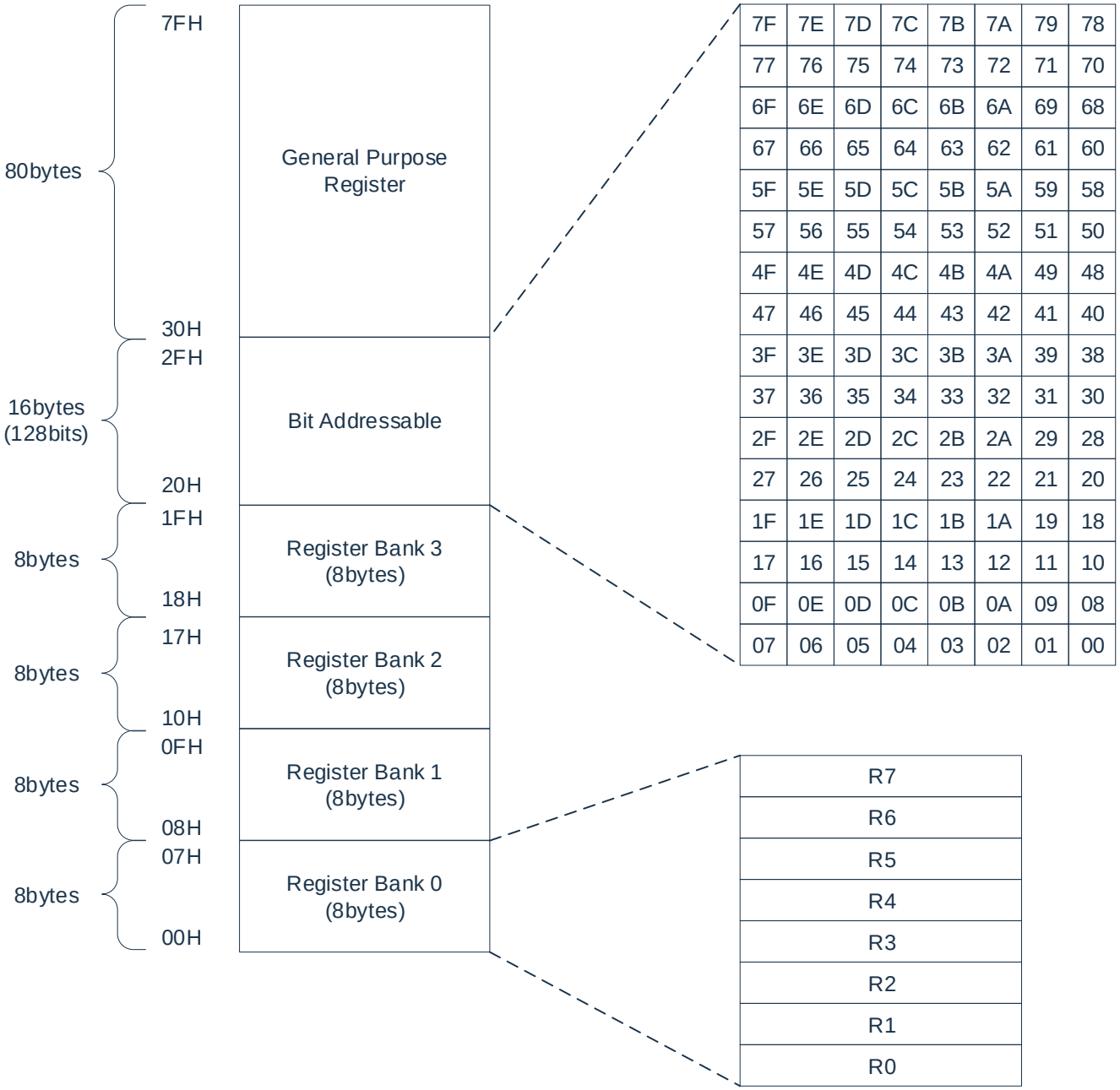
Internal data register is divided into 3 parts: low 128 bytes, high 128 bytes, SFR.

The internal data register address is one byte wide, which means that the address space is only 256 bytes wide. However, in fact 384 bytes wide is available if adjusting using internal RAM addressing method. Direct and indirect addressing memory space that are higher than 7FH enter different memory space, the diagram above indicates high 128 bytes and SFR occupy the same area which is 80H to FFH, but they themselves are independent.

The diagram above shows the low 128 bytes RAM for all 8051 series. The lowest 32bytes compose 4 registers, and the program instructions can call registers from R0 to R7, two bits in the program state determine which register group to use. It's more efficient to use code space in this way, because register instruction is shorter than direct addressing instruction.

The 16bytes after the register group compose a bit-addressable memory space. 8051 instruction setting includes bit instruction width operations, but the 128bits in this area can be directly addressed through these instructions. The address of the area is from 20H to 2FH.

All bytes from low 128 bytes in general register can be accessed directly or indirectly, and high 128 bytes can only be accessed indirectly. These areas are used as data RAM and stack.


7FH
80bytes
General Purpose Register
30H
2FH
16bytes (128bits)
Bit Addressable
20H
1FH
8bytes
Register Bank 3 (8bytes)
18H
17H
8bytes
Register Bank 2 (8bytes)
10H
0FH
8bytes
Register Bank 1 (8bytes)
08H
07H
8bytes
Register Bank 0 (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 56 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

2.1.4 Special Function Register Table (SFR)

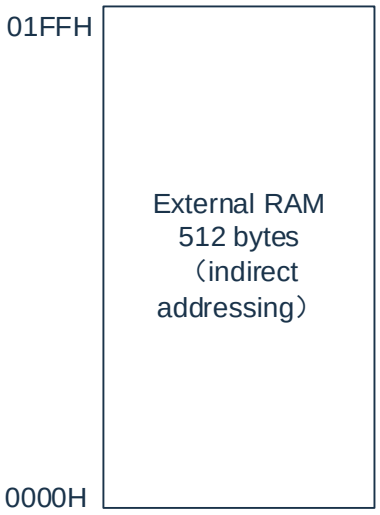
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
0xF8	--	--	--	MLOCK	MADRL	MADRH	MDATA	MCTRL
0xF0	B	I2CSADR	I2CSCR	I2CSBUF	I2CMSA	I2CMCR	I2CMBUF	I2CMTP
0xE8	--	ADCON2	SCON1	SBUF1	SPCR	SPSR	SPDR	SSCR
0xE0	ACC	--	TL4	TH4	--	--	--	--
0xD8	--	--	TL3	TH3	ADRESL	ADRESH	ADCON1	ADCON0
0xD0	PSW	ADCMPC	T34MOD	ADDLYL	ADCMPL	ADCMPH	CAPCR0	CAPCR1
0xC8	T2CON	T2IF	RLDL	RLDH	TL2	TH2	CCEN	T2IE
0xC0	--	--	CCL1	CCH1	CCL2	CCH2	CCL3	CCH3
0xB8	IP	EIP1	EIP2	--	WUTCRL	WUTCRH	BUZDIV	BUZCON
0xB0	P3	--	EIF2	--	P0EXTIF	P1EXTIF	P2EXTIF	P3EXTIF
0xA8	IE	--	EIE2	--	P0EXTIE	P1EXTIE	P2EXTIE	P3EXTIE
0xA0	P2	P1TRIS	P2TRIS	P3TRIS	--	--	--	--
0x98	SCON0	SBUF	P0TRIS	--	--	--	--	--
0x90	P1	FUNCCR	--	DPX0	--	DPX1	TA	WDCON
0x88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	CLKDIV
0x80	P0	SP	DPL0	DPH0	DPL1	DPH1	DPS	PCON

Note: The lower four digits of the SFR address are 0000 or 1000 and can be addressed, such as P0, TCON and P1.

Addresses marked with "--" in the table are not allowed to be accessed.

2.1.5 External Data Memory (XRAM)

There is 512 bytes XRAM inside the chip, this area has no connection with RAM/FLASH and can be read and written through an 8-bit SFR.



XRAM/XSFR operation:

DPTR includes two sets of pointers through the DPTR data pointer operation: DPTR0.DPTR1.

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

The assembly code is as follows:

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

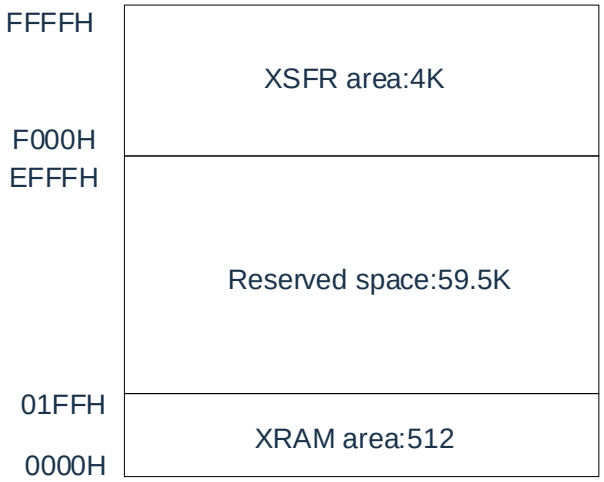
Through the MOVX indirect addressing operation, the assembly code is as follows:

```
MOV     R0,#01H
MOV     A,#5AH
MOVX    @R0,A        ; Write the data in A to XRAM address 01H, the 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. Generally use the DPTR performs XRAM/XSFR operations.

2.1.6 Special Function Register Table (XSFR)

XSFR is a special register sharing the address space with XRAM. It mainly includes: port control register and other function control registers. Its addressing range is as follows:



XSFR list:

Address	Symbol	Description
F004H	P04CFG	P0.4 port configuration register
F005H	P05CFG	P0.5 port configuration register
F006H	--	--
F007H	--	--
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
--	--	Not used
F013H	P13CFG	P1.3 port configuration register
F014H	P14CFG	P1.4 port configuration register
F015H	P15CFG	P1.5 port configuration register
F016H	P16CFG	P1.6 port configuration register
F017H	P17CFG	P1.7 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
--	--	Not used
F020H	--	--
F021H	P21CFG	P2.1 port configuration register
F022H	P22CFG	P2.2 port configuration register
F023H	P23CFG	P2.3 port configuration register
F024H	P24CFG	P2.4 port configuration register
F025H	P25CFG	P2.5 port configuration register
F026H	P26CFG	P2.6 port configuration register
F027H	--	--
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

Address	Symbol	Description
--	--	Not used
F030H	P30CFG	P3.0 port configuration register
F031H	P31CFG	P3.1 port configuration register
F032H	P32CFG	P3.2 port configuration register
F033H	--	--
F034H	--	--
F035H	P35CFG	P3.5 port configuration register
F036H	P36CFG	P3.6 port configuration register
F037H	--	--
F039H	P3OD	P3 port open-drain control register
F03AH	P3UP	P3 port pull-up resistor control register
F03BH	P3RD	P3 port pull-down resistor control register
--	--	Not used
F084H	P04EICFG	P0.4 port interrupt control register
F085H	P05EICFG	P0.5 port interrupt control register
--	--	Not used
F08BH	P13EICFG	P1.3 port interrupt control register
F08CH	P14EICFG	P1.4 port interrupt control register
F08DH	P15EICFG	P1.5 port interrupt control register
F08EH	P16EICFG	P1.6 port interrupt control register
F08FH	P17EICFG	P1.7 port interrupt control register
F090H	--	--
F091H	P21EICFG	P2.1 port interrupt control register
F092H	P22EICFG	P2.2 port interrupt control register
F093H	P23EICFG	P2.3 port interrupt control register
F094H	P24EICFG	P2.4 port interrupt control register
F095H	P25EICFG	P2.5 port interrupt control register
F096H	P26EICFG	P2.6 port interrupt control register
F097H	--	--
F098H	P30EICFG	P3.0 port interrupt control register
F099H	P31EICFG	P3.1 port interrupt control register
F09AH	P32EICFG	P3.2 port interrupt control register
F09BH	--	--
F09CH	--	--
F09DH	P35EICFG	P3.5 port interrupt control register
F09EH	P36EICFG	P3.6 port interrupt control register
F09FH	--	--
--	--	Not used
F120H	PWMCON	PWM control register
F121H	PWMOE	PWM output enable register
F122H	PWMPINV	PWM Output Polarity Selection Register
F123H	PWM01PSC	PWM0/PWM1 Prescaler Control Register
F124H	PWM23PSC	PWM2/PWM3 Prescaler Control Register
F125H	PWM45PSC	PWM4/PWM5 Prescaler Control Register
F126H	PWMCNTE	PWM Count Start Control Register
F127H	PWMCNTM	PWM Count Mode Selection Register
F128H	PWMCNTCLR	PWM counter clear control register

Address	Symbol	Description
F129H	PWMLOADEN	PWM Load Enable Control Register
F12AH	PWM0DIV	PWM0 divider control register
F12BH	PWM1DIV	PWM1 divider control register
F12CH	PWM2DIV	PWM2 divider control register
F12DH	PWM3DIV	PWM3 divider control register
F12EH	PWM4DIV	PWM4 divider control register
F12FH	PWM5DIV	PWM5 divider control register
F130H	PWMP0L	PWM0 period low 8-bit register
F131H	PWMP0H	PWM0 period high 8-bit register
F132H	PWMP1L	PWM1 period low 8-bit register
F133H	PWMP1H	PWM1 period high 8-bit register
F134H	PWMP2L	PWM2 period low 8-bit register
F135H	PWMP2H	PWM2 period high 8-bit register
F136H	PWMP3L	PWM3 period low 8-bit register
F137H	PWMP3H	PWM3 period high 8-bit register
F138H	PWMP4L	PWM4 period low 8-bit register
F139H	PWMP4H	PWM4 period high 8-bit register
F13AH	PWMP5L	PWM5 period low 8-bit register
F13BH	PWMP5H	PWM5 period high 8-bit register
--	--	Not used
F140H	PWMD0L	PWM0 data low 8-bit register
F141H	PWMD0H	PWM0 data high 8-bit register
F142H	PWMD1L	PWM1 data low 8-bit register
F143H	PWMD1H	PWM1 data high 8-bit register
F144H	PWMD2L	PWM2 data low 8-bit register
F145H	PWMD2H	PWM2 data high 8-bit register
F146H	PWMD3L	PWM3 data low 8-bit register
F147H	PWMD3H	PWM3 data high 8-bit register
F148H	PWMD4L	PWM4 data low 8-bit register
F149H	PWMD4H	PWM4 data high 8-bit register
F14AH	PWMD5L	PWM5 data low 8-bit register
F14BH	PWMD5H	PWM5 data high 8-bit register
--	--	Not used
F150H	PWMDD0L	PWM0 asymmetric down compare data low 8-bit register
F151H	PWMDD0H	PWM0 asymmetric down compare data high 8-bit register
F152H	PWMDD1L	PWM1 asymmetric down compare data low 8-bit register
F153H	PWMDD1H	PWM1 asymmetric down compare data high 8-bit register
F154H	PWMDD2L	PWM2 asymmetric down compare data low 8-bit register
F155H	PWMDD2H	PWM2 asymmetric down compare data high 8-bit register
F156H	PWMDD3L	PWM3 asymmetric down compare data low 8-bit register
F157H	PWMDD3H	PWM3 asymmetric down compare data high 8-bit register
F158H	PWMDD4L	PWM4 asymmetric down compare data low 8-bit register
F159H	PWMDD4H	PWM4 asymmetric down compare data high 8-bit register
F15AH	PWMDD5L	PWM5 asymmetric down compare data low 8-bit register
F15BH	PWMDD5H	PWM5 asymmetric down compare data high 8-bit

Address	Symbol	Description
		register
--	--	Not used
F160H	PWMDTE	PWM programmable dead time delay control register
F161H	PWM01DT	PWM0/PWM1 Programmable Dead Time Delay Register
F162H	PWM23DT	PWM2/PWM3 Programmable Dead Time Delay Register
F163H	PWM45DT	PWM4/PWM5 Programmable Dead Time Delay 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 Enable Register
F169H	PWMZIE	PWM Zero Interrupt Enable Register
F16AH	PWMUIE	PWM Up Compare Interrupt Enable Register
F16BH	PWMDIE	PWM Down Compare Interrupt Enable Register
F16CH	PWMPIF	PWM Periodic Interrupt Flag Register
F16DH	PWMZIF	PWM Zero Interrupt Flag Register
F16EH	PWMUIF	PWM Up Compare Interrupt Flag Register
F16FH	PWMDIF	PWM Down Compare Interrupt Flag Register
--	--	Not used
F5C0H	BRTCON	BRT module control register
F5C1H	BRTDL	BRT timer load value low 8 bits
F5C2H	BRTDH	BRT timer load value high 8 bits
--	--	Not used
F5E0H	UID0	UID<7:0>
F5E1H	UID1	UID<15:8>
F5E2H	UID2	UID<23:16>
F5E3H	UID3	UID<31:24>
F5E4H	UID4	UID<39:32>
F5E5H	UID5	UID<47:40>
F5E6H	UID6	UID<55:48>
F5E7H	UID7	UID<63:56>
F5E8 H	UID8	UID<71:64>
F5E9 H	UID9	UID<79:72>
F5EA H	UID10	UID<87:80>
F5EB H	UID11	UID<95:88>

Note: Access to addresses marked with "--" in the table is Disabled.

2.2 Accumulator (ACC)

ALU is an 8-Bit wide arithmetic logic unit, all the mathematical and logical operations of MCU are accomplished through it. It can carry out logical operations such as addition, subtraction and shift on data; ALU also controls the status bit (PSW status register), to indicate the state of the operation result.

ACC is an 8-bit register, the operation results of ALU could be stored in ACC.

2.3 B Register (B)

B Register is used when multiply and division instructions are used or it can be used as a general register when not using multiply and division instructions.

2.4 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 IRAM address. The value of the SP can be modified. If the stack area is set to 0C0, 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.

2.5 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.

2.6 Data Pointer Selection Register (DPS)

Data Pointer Selection Register DPS

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

Bit7~Bit6	ID1-ID0:	Self-subtract/ self-add function selection
	00=	DPTR0 plus 1 or DPTR1 plus 1;
	01=	DPTR0 minus 1 or DPTR1 plus 1;
	10=	DPTR0 plus 1 or DPTR1 minus 1;
	11=	DPTR0 minus 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 bit;
	1=	After the MOVX @ DPTR or MOVC @ DPTR instructions are allowed to run, execute self-add/self subtract(determined by ID1-ID0)
	0=	DPTR related instructions does not affect SEL.
Bit3~Bit1	---	
Bit0	SEL:	Data pointer selection bit;
	1=	Select DPTR1;
	0=	Select DPTR0.

2.7 Program Status Register (PSW)

Program Status Register PSW

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

Bit7 CY: Carry flag bit;
1= carry;
0= no carry.
Bit6 AC: Auxiliary Carry Flag Bit (half carry flag bit);
1= carry;
0= no carry.
Bit5 F0: General Purpose Flag.
Bit4~Bit3 RS1-RS0: Work Register BANK Select Bit;
00= Select Bank0;
01= Select Bank1;
10= Select Bank2;
11= Select Bank3.
Bit2 OV: Overflow Flag Bit;
1= Arithmetic or logical operation has overflow;
0= Arithmetic or logical operation has no overflow.
Bit0 P: Parity Bit;
1= The highest bit of the result is carried.
0= The highest bit of the result does not carry.

2.8 Program Counter (PC)

Program Counter (PC) controls the instruction execution sequence in the program memory Flash, Its addressing range is over the entire Flash. After obtaining the instruction code, the PC will automatically add one and point to the address of the next instruction code. But if executions such as jump, conditional jump, assign value to PCL, subroutine call, initialization reset, interrupt, interrupt return, subroutine return are performed, 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 which is read during the current instruction execution will be discarded, and an empty instruction operation cycle will be inserted before the correct instruction can be obtained. Otherwise, the next instruction will be executed sequentially.

2.9 Timing Access Register (TA)

Timing Access Register TA

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

Bit7~Bit0 TA[7:0]: timing access control bit.

Some protected registers must 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.

2.10 Watch Dog TIMER (WDT)

Watch Dog TIMER is an on-chip timer, WDT timing overflow will result in reset.

2.10.1 WDT Overflow Period

For details, please refer to the <Watchdog Reset (WDT) >.

2.10.2 Watch Dog Control Register WDCON

Watch Dog 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/W	R/W
Reset Value	0	1	---	---	0	0	0	0

Bit7 SWRST: Software reset control bit;
1: Execute system software reset (write 0 to clear after reset).
0: ---
Bit6 PORF: Power-on reset flag bit;
1: The system is power-on reset (write 0 to clear, TA write timing is not required).
0: ---
Bit5~Bit4 Unused.
Bit3 WDTIF: WDT Overflow interrupt flag bit;
1= WDT Overflow (write 0 to clear);
0= WDT No overflow.
Bit2 WDTRF: WDT reset flag bit;
1= The system is reset by WDT (write 0 to clear);
0= The system is not reset by WDT.
Bit1 WDTRE: WDT Reset enable bit;
1= Enable WDT to reset CPU;
0= Disable WDT to reset CPU.
Bit0 WDTCLR: WDT Counter clear bit;
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 Disable, WDT can be enabled or disabled by using WDTRE control bits.

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

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

2.10.3 WDT Overflow Control Register CKCON

0x8E	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CKCON	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~Bit6 WTS1-WTS0: WDT overflow time select bit;
00= 2^{17}*Tsys;
01= 2^{20}*Tsys;
10= 2^{23}*Tsys;
11= 2^{26}*Tsys.
Bit5 Unused.
Bit4 T1M: Clock source selection bit for Timer1.
0= Fsys/12;
1= Fsys/4.
Bit3 T0M: Clock source selection bit for Timer0.
0= Fsys/12;
1= Fsys/4.
Bit2~Bit0 Unused.

2.11 Function Control Register (FUNCCR)

0x91	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FUNCCR	---	---	UART1_PORTS	I2C_PORTS	UART1_CKS1	UART0_CKS1	UART1_CKS	UART0_CKS
R/W	R/W	R/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 bits: Must be 0.

Bit5 UART1_PORTS: UART1_PORTS.

0= Select TXD1, RXD1 function port;

1= Select [TXD1], [RXD1] function port.

Bit4 I2C_PORTS: I^2C port select selection bit;

0= Select SCL, SDA function port;

1= Select [SCL], [SDA] function port.

Bit3 UART1_CKS1: Timer clock source higher selection bits for UART1, {UART1_CKS1, UART1_CKS}:

00= The overflow clock of Timer1;

01= The overflow clock of Timer4;

10= The overflow clock of Timer2;

11= The overflow clock of BRT;

Bit2 UART0_CKS1: Timer clock source higher selection bits for UART0, {UART0_CKS1, UART0_CKS}:

00= The overflow clock of Timer1;

01= The overflow clock of Timer4;

10= The overflow clock of Timer2;

11= The overflow clock of BRT;

Bit1 UART1_CKS: Timer clock source lower selection bits for UART1, see UART1_CKS1 description;

Bit0 UART0_CKS: Timer clock source lower selection bits for UART0, see UART1_CKS0 description;

3. SYSTEM CLOCK

3.1 System Oscillators

The chip has 4 types of oscillation:

- Internal RC oscillation (HSI);
- External high-speed oscillation (HSE);
- External low-speed oscillation (LSE - 32.768 KHz);
- Internal low-power oscillation (LSI).

The default oscillation mode of the chip is internal RC oscillation, and it can be modified to other oscillators in CONFIG.

3.2 Reset Time

Reset Time refers to the time from reset to starting executing instructions of the chip. Its default design value is about 16ms. This time includes oscillator start-up time and configuration time.

Note: This reset time will exist whether the chip is powered-on reset or reset due to other reasons.

In addition, when the oscillator is selected as external low-speed crystal oscillation (32.768KHz), the reset time (including the start-up time) is about 1.5s as default (external capacitor is 10pF~22pF) .

3.3 Clock Structure

3.3.1 Clock Control Register

The oscillator control (CLKDIV) register controls the system clock and frequency selection.

3.3.2 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=Fosc;
other: Fsys =Fosc/ (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
```

3.3.3 Function Clock Control Register CKCON

0x8E	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CKCON	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~Bit0 WTS1-WTS0: WDT overflow time select bit.
00= 2^{17}*Tsys;
01= 2^{20}*Tsys;
10= 2^{23}*Tsys;
11= 2^{26}*Tsys.
Bit5 Unused.
Bit4 T1M: Clock source selection bit for Timer1.
0= Fsys/12;
1= Fsys/4.
Bit3 T0M: Clock source selection bit for Timer0.
0= Fsys/12;
1= Fsys/4.
Bit2~Bit0 Unused.

4. RESET

The chip can be reset in five ways as follows:

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

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

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

4.1 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. The normal timing of power-on reset is given below:

- Power on: the system detects a rise in power supply voltage and waits for it to stabilize;
- system initialization: all system registers are set to initial values;
- The oscillator starts to work: the oscillator starts to provide the system clock;
- Executing the program: The power-on is over and the program starts to run.

Stabilization time defaults to 16ms,and if 32.768KHz crystal oscillator is selected for configuration, the stabilization time is about 1.5s.

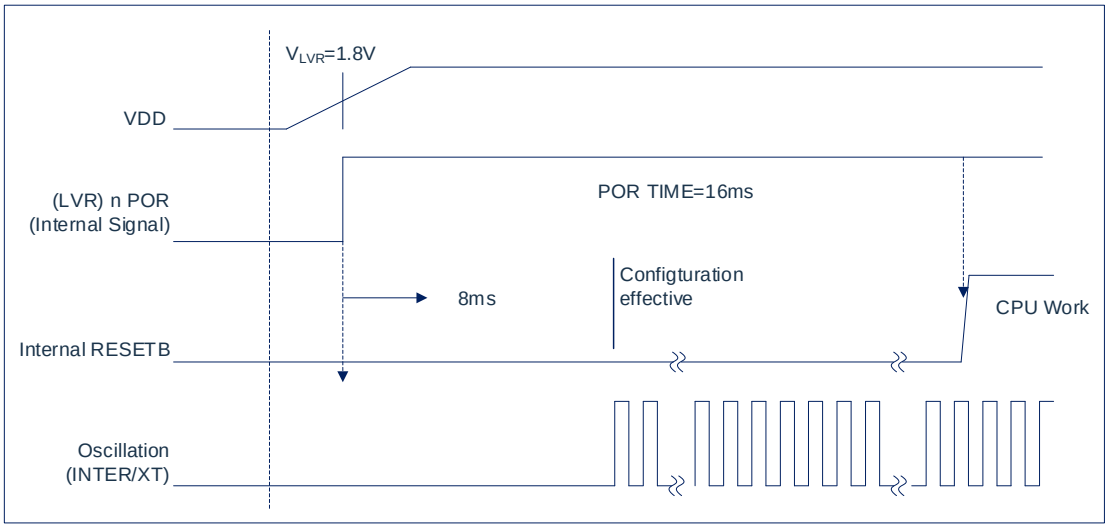


Figure 4-1: Power-on reset timing chart

Whether the system is powered on or reset can be judged by the PORF (WDCON.6) flag.

WDCON Register

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

Bit7 SWRST: Software reset control bit;
1: Execute system software reset (write 0 to clear after reset).
0: ---

Bit6 PORF: Power-on Reset bit;
1: The system is power-on reset/LVR reset/external reset/ CONFIG protection reset (write 0 clear, no TA write timing required);
0: ---

Bit5~ Bit4 Unused.

Bit3~ Bit0 WDT: Related control/flag bits. See the WDT chapter for details.

The reset types that can set the PORF flag to 1 are: power-on reset, LVR reset, external reset, and CONFIG protection reset.

4.2 External Reset

The external reset refers to the reset signal from the external port, and the external reset is input by the Schmitt trigger. If the RESETB pin is held low for more than 16us (three rising edges of the internal LPRC 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 changes to "1", 16ms is required. The settling time of internal RESET becomes "1". The program starts at vector address 0000H. The external reset pin can be configured with an internal pull-up resistor enable in CONFIG.

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

4.3 Low Voltage Reset

The low-voltage reset (LVR) function is integrated inside the chip. When the system voltage V_{DD} 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 $V_{DD}<V_{LVR}$, a reset is requested.

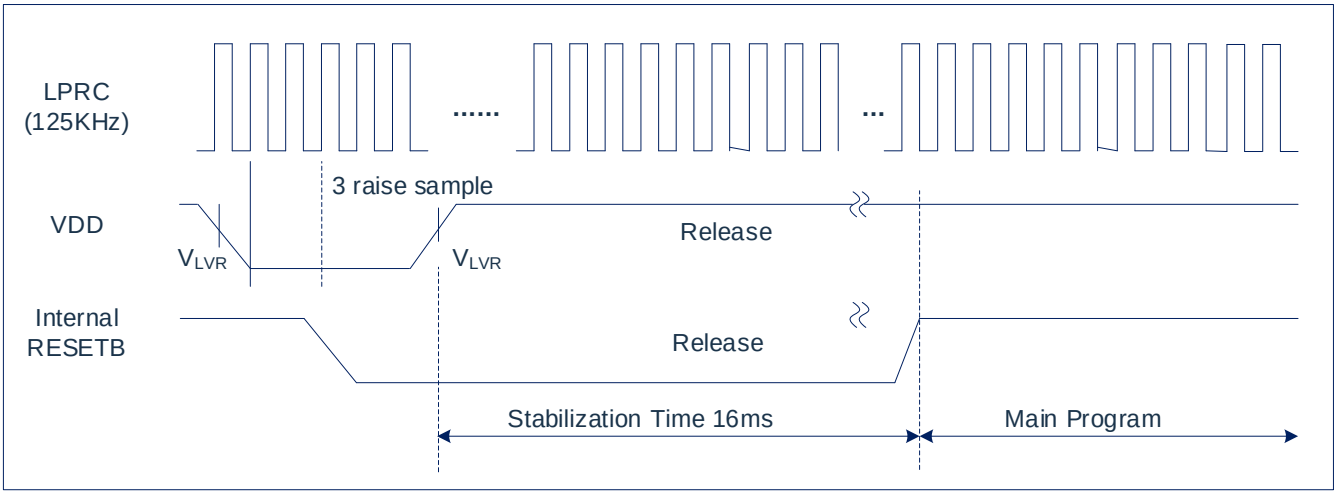


Figure 4-2:LVR Low Voltage Reset Timing Diagram

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

4.4 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:

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

Reset the CPU with all registers when WDT overflows, and the program is executed from 0000H immediately after 1 Tsys. WDT reset does not re-configure the power-on reset.

The clock source for the WDT is provided by the system clock (System Clock), and the clock fundamental period of the WDT counter is Tsys.

The watchdog's overflow time can be set by the program. The overflow time can be selected by two bits in the CKCON register, WDS1-WTS0.

WTS[1:0]	Watchdog Interval	Number of clocks	OVT@Fsys=16MHz
00	2^{17}	131072	8ms
01	2^{20}	1048576	65ms
10	2^{23}	8388608	524ms
11	2^{26}	67108864	4.2s

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

4.5 Software Reset

Program software reset can be implemented inside the chip. Software reset can relocate the program flow to reset address 0000H, and then run the program again. The user can implement a custom software reset. The control bit that implements the software reset SWRST (WDCON.7).

WDCON

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

Bit7 SWRST: Software reset control bit.
1: Execute system software reset (change from 0->1 to generate reset, software reset will not clear to zero, need to write 0 to clear).
0: ---

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

Bit5~ Bit4 Unused.

Bit3~ Bit0 WDT related control/flag bits. See the WDT chapter for details.

4.6 CONFIG Status Protection Reset

The CONFIG state protection reset is an enhanced protection mechanism for the system. During powering on and reset, there is a set of 16-bit CONFIG registers internally that load the fixed code (A569H) set in the FLASH and are not being operated during normal operation. In the case of a specific non-program operation, the value of this register changes and does not equal to the original fixed code, after a number of sampling clocks, if the register continues to remain unfixed code, the system will 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 for sampling register value is internal RC Fixed_Clock (8MHz, clock source from HSI) and low power clock (LSI 125KHz), once the register value is not fixed code, force enable LSI oscillator and HSI oscillator, and system clock switch to LSI clock, if after 12 Fixed_ Clock sampling or 3 LSI clock sampling, the register still maintains the state of not being a fixed code, then the system generates a reset.

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

4.7 Power-on Configuration Monitoring Reset

In the power-on configuration process, there is a configuration monitoring circuit inside the chip. If the configuration time is too long in the power-on, or if the power-on configuration enters a certain state that cannot be reconfigured, the internal monitoring circuit starts timing from the configuration, and if the set time is exceeded, the monitoring circuit resets the configuration module and allows the configuration module to re-configure the process. To prevent the system from entering a non-predictable state at power-on.

The monitoring circuit operating clock is LSI (125KHz), the default monitoring time is 65ms, if you choose 32.768KHz crystal oscillation, the monitoring time is 2.1s.

5. POWER MANAGEMENT

The low power modes are divided into 2 categories:

- IDLE: Idle mode
- STOP: Sleep mode

When a 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 IDLE and STOP bits directly.

- Enter idle mode: IDLE();
- Enter sleep mode: STOP();

5.1 Power Management Register PCON

Power Management Control 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 bit;
	0=	UART0 baud rate is normal;
	1=	UART0 doubles the baud rate.
Bit6	SMOD1:	UART1 baud rate multiplication bit;
	0=	UART1 baud rate is normal;
	1=	UART1 doubles the baud rate.
Bit5	---	
Bit4~Bit3	Reserved bit:	Must be 0.
Bit2	SWE:	STOP status function wake-up enable bit; (The system can be restarted by a power-down reset or an enabled external reset, regardless of the SWE value)
	0=	Disabling of functional wake-up calls;
	1=	Allow function wake-up (can be woken up by external interrupt and timed wake-up)
Bit1	STOP:	The dormant state control bit;
	0=	Not in a dormant state;
	1=	Enters sleep state (automatically cleared by exiting STOP mode).
Bit0	IDLE:	Idle status control bit;
	0=	Not in idle status;
	1=	Enters idle state (automatically cleared when exiting IDLE mode).

5.2 Idle Mode

In this mode, only the CPU clock source is turned off. Therefore, peripheral functions (such as timers, PWM and I^2C) and clock generators (HSI/crystal oscillator driver) still work normally in this state.

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

If you enter idle mode in the interrupt service program, only interrupts with higher priority can wake up the system.

5.3 Sleep Mode

In this mode, all circuits are turned off. The system is in ultra-low power mode and all analog and digital circuits are not working.

5.3.1 Wake From Sleep Mode

After entering sleep mode, the function wake-up can be turned on (SWE=1 needs to be set), which can be woken up by external interrupt or wake-up by timing module (WUT).

- Waking up by external interrupts, the general interrupt enable and port interrupt enable must be turned on before going to sleep to wake up the system.
- When waking up by the timed wake-up module, you need to turn on the timer wake-up function and set the time from the sleep state to wake-up before entering to sleep. The clock source of the timer wakeup circuit is provided by the LSI (low power oscillator). If the timer wake-up function is turned on, the LSI is automatically turned on in the sleep state.

5.3.2 Wake Waiting Status

After waking up, the oscillator starts. Since the oscillation frequency is not stable, the CPU is still not working, and the PC is still in the sleep state. The system needs to wait for a while before providing the clock to the CPU. The waiting time for waking up the CPU is programmed in the CONFIG. The waiting time can be set to 50us~1.2s. After the wake-up waiting time elapses, the clock is provided for the CPU only if MCU considers the system clock is stable,, and the program continues to execute.

If the internal power-down wake-up timer and the external interrupt wake-up function are both turned on, any wake-up mode can wake up the CPU after the system enters power-down mode.

If the internal timer wakes up the oscillator first, after that there is an external interrupt input, then after the wake-up waiting time elapses, the program executes the interrupt handler first and then resumes the next instruction in the sleep operation.

5.3.3 Wake Up Time in Sleep Mode

The total wake-up time for waking up the system using external interrupts is

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

The total wake-up time using the timer wake-up system is

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

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

5.3.4 Reset To Reboot The System

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

Power-down reset: without any other conditions, V_{DD} is reduced to 0V and then re-powered to the operating voltage to enter the power-up reset state.

External reset: external reset function needs to be enabled, and the relevant port is configured as a dedicated reset port. When the reset port is kept low for >1us in sleep state, the system generates a reset, and if the reset port is released, the system restarts.

5.4 Sleep Mode Application Example

Before the system enters the sleep mode, if the user needs to obtain a small sleep current, please confirm the status of all I/O first. If there is a floating I/O port in the user's application, set all the floating ports as output ports to ensure that all the input ports are set to be in fixed status. Each input port has a fixed state to prevent the increasement of the sleep current from the port line voltage level being in an indeterminate state when I/O ports are in input state; turn off other peripheral modules such as AD to reduce the sleep current.

Example: Processing for entering sleep mode when timed wake-up is used (assembly program)

```
SLEEP_MODE:
        MOV     WUTCRL,#31H
        MOV     WUTCRH,#80H
        MOV     P0TRIS,#0FFH
        MOV     P0,#0FFH
        MOV     P1TRIS,#0FFH
        MOV     P1,#0FFH
        MOV     P2TRIS,#0FFH
        MOV     P2,#0FFH
        MOV     P3TRIS,#0FFH
        MOV     P3,#0FFH
        Turn off other operations
        MOV     PCON,#06H       ; Perform wake-able sleep operations,
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP     ; The instruction to execute sleep operation must be followed by 6 NOP instructions
        Other intructions after wake-up
```

5.5 Sleep Power Consumption In Debug Mode

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

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

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

6. I/O PORT

6.1 GPIO Function

The chip has four I/O ports: PORT0, PORT1, PORT2, PORT3. The readable and writable port data registers provide direct access to these ports.

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

Reading the PORTx register reads the state of the pin, 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. The PxTRIS register controls the direction of the PORTx pin even when the PORTx pin is used as an analog input. When using the PORTx pin as an analog input, the user must ensure that the bits in the PxTRIS register remain set to '0'. I/O pins configured as analog inputs are always read as 0.

The registers associated with the PORTx port are Px, PxTRIS, PxOD, PxUP, PxRD, and so on.

6.1.1 PORTx Data Register Px

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

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

Bit7~Bit0 Px<7:0>: Px I/O pin bits.
1= port pin level>V_{IH};
0= port pin level<V_{IL}.

6.1.2 PORTx Direction Register PxTRIS

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

Register P0TRIS Address: 0x9A; Register P1TRIS Address: 0xA1;

Register P2TRIS Address: 0xA2; Register P3TRIS Address: 0xA3.

Bit7~Bit0 PxTRIS<7:0>: Three-state control bits.
1= pin is configured as an output;
0= The pin is configured as an input (tri-state).

Note:
1. After the port is set as the output port, the data of the port is read as the value of the output register.
2. 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.

6.1.3 PORTx Open-Drain Control Register PxOD

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

Register P0OD Address: F009H; Register P1OD Address: F019H;

Register P2OD Address: F029H; Register P3OD Address: F039H.

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

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

0= Pin is configured for normal state (output is push-pull output).

6.1.4 PORTx Pull-up Resistor Control Register PxUP

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

Register P0UP Address: F00AH; Register P1UP Address: F01AH;

Register P2UP Address: F02AH; Register P3UP Address: F03AH.

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

1= Pin pull-up resistor is turned on;

0= Pin pull-up resistor is off.

6.1.5 PORTx Pull-Down Resistor Control Register PxRD

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

Register P0RD Address: F00BH; Register P1RD Address: F01BH;

Register P2RD Address: F02BH; Register P3RD Address: F03BH.

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

1= pin pull-down resistor is turned on;

0= Pin pull-down resistor is 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.

6.2 Multiplexing Functions

6.2.1 Port Configuration Registers

PORTx Function Configuration Register Px_NCFG

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Px_NCFG	---	---	---	---	---	Px2CFG	Px1CFG1	Px0CFG
R/W	---	---	---	---	---	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit7~Bit3 Unused.

Bit2~Bit0 $Px_NCFG<2:0>$: Function configuration bit, default is GPIO function

Please refer to Port Function Configuration for details.

There are 8 function configuration registers of Px, including Px0CFG~Px7CFG, which control the function configuration of Px0~Px7 respectively.

The multiplexing function of the port is determined by the Px[i] CFG register, and the table reads as follows:

PIN	CONFIG	0 (GPIO)	1	2	3	4	5	6
P0.4		ADET1	AN4	[RXD1]	SCL			PG2
P0.5		FB1	AN5	[TXD1]	SDA			PG1
P1.3		IC0/T2	AN6			PG3	PG5	PG0
P1.4		IC1/T2EX	AN7			NSS3	-	PG1
P1.5		IC2	AN18			SCLK	PG0	PG2
P1.6		IC3/T1	AN19			MOSI	PG1	PG3
P1.7		IC4	AN20			MISO		PG4
P2.1	DSCK		AN21	RXD1	[SCL]	NSS5		
P2.2		IC5	AN8		CC3	NSS6		PG5
P2.3		IC8/ADET0	AN9		CC2			PG3
P2.4		IC6/T0	AN10		CC1		BEEP	PG2
P2.5		INT0	AN11	TXD0				
P2.6		INT1	AN12	RXD0				
P3.0	NRST		AN22					
P3.1	OSCIN	T0G	AN13					
P3.2	OSCOUT	T1G	AN14					
P3.5	DSDA		AN16	TXD1	[SDA]			
P3.6		IC7	AN17		CC0	NSS7	PG0	PG5

To use the multiplexing function of GPIO configuration, just set the corresponding port as input port.

To configure as an analog port, simply configure Px[i]CFG to 1.

In this state, the smit input/up/down resistors are off.

Example: If you want to set P2.4 to BEEP buzzer function:

```
Compilation:  MOV   DPDR , #F024H ; F024H is P24CFG's address
              MOV A , #05H
              MOVX @DPTR , A
C:            P24CFG = 0x05;
```

6.2.2 Port External Interrupt Control Register

When using external interrupts, you need to configure the port as a GPIO function and set the direction to the output port, or set reuse function to input port (such as RXD0, RXD1).

PORTx external interrupt control register Px_NEICFG

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

Bit7~Bit2 Unused.

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

00= External interrupt prohibition;

01= Rising edge trigger interrupt;

10= Falling edge trigger interrupt;

11= Trigger interrupt at both rising and Falling edges.

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

7. INTERRUPT

7.1 Overview

The chip has 18 interrupt sources and interrupts vectors:

Interrupt sources	Interrupt description	Interrupt vectors	Same priority sequence
INT0	External Interrupt 0	0-0x0003	1
Timer0	Timer 0 interrupt	1-0x000B	2
INT1	External interrupt 1	2-0x0013	3
Timer1	Timer 1 interrupt	3-0x001B	4
UART0	TI0or RI0	4-0x0023	5
Timer2	Timer 2 interrupt	5-0x002B	6
UART1	TI1or RI1	6-0x0033	7
P0EXTIF<7:0>	P0 port external interrupt	7-0x003B	8
P1EXTIF<7:0>	P1 port external interrupt	8-0x0043	9
P2EXTIF<7:0>	P2 port external interrupt	9-0x004B	10
P3EXTIF<7:0>	P3 port external interrupt	10-0x0053	11
--	--	11-0x005B	12
--	--	12-0x0063	13
--	--	13-0x006B	14
---	--	14-0x0073	15
Timer3	Timer 3 interrupt	15-0x007B	16
Timer4	Timer 4 interrupt	16-0x0083	17
--	--	17-0x008B	18
PWM	PWM interrupt	18-0x0093	19
ADC	ADC interrupt	19-0x009B	20
WDT	WDT interrupt	20-0x00A3	21
I^2C	I^2 C interrupt	21-0x00AB	22
SPI	SPI Interrupt	22-0x00B3	23

The chip specifies two interrupt priorities to achieve 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. However, the same level or lower level interrupt can not interrupt the high-level interrupt response.

7.2 External Interrupt

7.2.1 INT0/INT1 Interrupt

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

7.2.2 GPIO Interrupt

Each GPIO pin of the chip supports external interrupts. And can support falling edge/rising edge/double edge interrupt. To choose which edge to trigger in Px_N EICFG register configuration, example: if you need to configure P1.3 port as falling edge interrupt:

```
P13CFG=0x00.        //set P1.3 as GPIO
P1TRIS&=0xF7.    //set P1.3 as input port
P13EICFG=0x02.  //set P1.3 as falling edge triggered interrupt
```

The interrupts of GPIO occupy a total of 4 interrupt vectors:

- P0.4 - P0.5 occupy an interrupt vector 0x003B;
- P1.3 - P1.7 occupy an interrupt vector 0x0043;
- P2.1 - P2.6 occupy an interrupt vector 0x004B;
- P3.0-P3.2, P3.5-P3.6 occupy an interrupt vector 0x0053.

If an interrupt is generated, entering the interrupt service program can first determine which port triggered the interrupt, and then process it accordingly.

7.2.3 Interrupt and Wake Up from Sleep

Each external interrupt can be set to wake up the system after it enters the sleep mode (STOP wake-able mode).

INT0/INT1 interrupt wake-up system needs to open the corresponding interrupt enable and total interrupts enable, and wake-up mode is falling edge wake-up (INT0/INT1 wake-up mode and interrupt trigger mode selection bit IT0/IT1 are not related).

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

After the system is woken up by an external interrupt, it firstly enters the interrupt service program to handle the interrupt wake-up task, and after exiting the interrupt service program, the system continues to execute the instructions after the sleep operation

7.3 Interrupt Register

7.3.1 Interrupt Mask Register

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.

Interrupt Mask Register IE (0xA8)

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: Overall interrupt enable bit;
1= Enable all un-masked interrupt;
0= Disable all interrupt.

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

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

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

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

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

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

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

Timer2 Interrupt Mask Register T2IE (0xCF)

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

Bit7 T2OVIE: Timer2 overflow interrupt enable bit;
1= Enable interrupt;
0= Disable interrupt.
Bit6 T2EXIE: Timer2 external load interrupt enable bit;
1= Enable interrupt;
0= Disable interrupt.
Bit5 ---
Bit4 --- Timer2 compare3 interrupt enable bit;
Bit3 T2C3IE: Enable interrupt;
1= Disable interrupt.
0= Timer2 compare2 interrupt enable bit;
Bit2 T2C2IE: Enable interrupt;
1= Disable interrupt.
0= Timer2 compare1 interrupt enable bit;
Bit1 T2C1IE: Enable interrupt;
1= Disable interrupt.
0= Timer2 compare0 interrupt enable bit;
Bit0 T2C0IE: Enable interrupt;
1= Disable interrupt.
0=

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

P0 port interrupts control register P0EXTIE (0xAC)

0xAC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0EXTIE	---	---	P05IE	P04IE	--	--	---	---
R/W	R/W	R/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 ---
Bit5~Bit4 P0iIE: P0i port interrupt enable bit (i=0-5);
1= Enable interrupt;
0= Disable interrupt;
Bit3~Bit0 ---

P1 port interrupts control register P1EXTIE (0xAD)

0xAD	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P1EXTIE	P17IE	P16IE	P15IE	P14IE	P13IE	---	---	---
R/W	R/W	R/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 P1iIE: P1iport interrupt enable bit (i =3-7);
1= Enable interrupt;
0= Disable interrupt.
Bit2~Bit0 ---

P2 port interrupt control register P2EXTIE (0xAE)

0xAE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2EXTIE	---	P26IE	P25IE	P24IE	P23IE	P22IE	P21IE	---
R/W	R/W	R/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~Bit1 P2iIE: P2iport interrupt enable bit (i=1-6);
1= Enable interrupt;
0= Disable interrupt.
Bit0 ---

P3 port interrupt control register P3EXTIE (0xAF)

0xAF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P3EXTIE	---	P36IE	P35IE	---	---	P32IE	P31IE	P30IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit7 ---
Bit6~Bit5 P3iIE: P3iport interrupt enable bit (i=5-6);
1= Enable interrupt;
0= Disable interrupt.
Bit4~Bit3 ---
Bit2~Bit0 P3iIE: P3iport interrupt enable bit (i=0-2);
1= Enable interrupt;
0= Disable interrupt.

Interrupt mask register EIE2 (0xAA)

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

Bit7 SPIIE: SPI Interrupt enable bit
1= Enable SPI interrupt;
0= Disable SPI interrupt.
Bit6 I2CIE: I^2C Interrupt enable bit
1= Enable I^2C interrupt;
0= Disable I^2C interrupt.
Bit5 WDTIE: WDT Interrupt enable bit
1= Enable WDT interrupt;
0= Disable WDT interrupt.
Bit4 ADCIE: ADC Interrupt enable bit
1= Enable ADC interrupt;
0= Disable ADC interrupt.
Bit3 PWMIE: PWM global interrupt enable bit
1= Enable PWM global interrupt;
0= Disable PWM global interrupt.
Bit2 ---
Bit1 ET4: Time4 Interrupt enable bit
1= Enable Time4 interrupt;
0= Disable Time4 interrupt.
Bit0 ET3: Time3 Interrupt enable bit
1= Enable Time3 interrupt;
0= Disable Time3 interrupt.

7.3.2 Interrupt Priority Control Register

Interrupt priority control register IP is a readable and writable register that can be operated bit by bit.

Interrupt priority control register IP (0xB8)

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 ---

Bit6 PS1: UART1 interrupt priority control bit;
1= Set to advanced interrupt;
0= Set to low level interrupt;

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

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

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

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

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

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

Ports interrupt priority control register EIP1 (0xB9)

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

Bit7 ---
Bit6~ Bit4 ---
Bit3 PP3: P3 port interrupt priority control bit;
1= Set to advanced interrupt;
0= Set to low level interrupt.
Bit2 PP2: P2 port interrupt priority control bit;
1= Set to advanced interrupt;
0= Set to low level interrupt.
Bit1 PP1: P1 port interrupt priority control bit;
1= Set to advanced interrupt;
0= Set to low level interrupt.
Bit0 PP0: P0 port interrupt priority control bit;
1= Set to advanced interrupt;
0= Set to low level interrupt.

Peripheral interrupt priority control register EIP2 (0xBA)

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

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

7.3.3 Interrupt Flag Bit Register

Timer0/1, INT0/1 interrupt flag bit register TCON, can perform addressing by bit.

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 overflows the interrupt flag bit;
1= Timer1 counter Overflow, enter the interrupt service program hardware automatically zero;
0= Timer1 counter has no overflow.

Bit6 TR1: Timer1 runs the control bit;
1= Timer1 starts;
0= Timer1 stopped.

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

Bit4 TR0: Timer0 runs the control bit;
1= Timer0 starts;
0= Timer0 stopped.

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

Bit2 IT1: External interrupt 1 trigger mode control bit;
1= The falling edge triggers;
0= Low-level trigger.

Bit1 IE0: External interrupt 0 flag bit;
1= External interrupt 0 produces an interrupt, and the hardware of the interrupt service program is automatically cleared.
0= External interrupt 0 did not cause an interrupt.

Bit0 IT0: External interrupt 0 trigger mode control bit;
1= The falling edge triggers;
0= Low-level trigger.

UART0 interrupt flag bit register SCON0, can perform addressing by bit.

0x98	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SCON0	U0SM0	U0SM1	U0SM2	U0REN	U0TB8	U0RB8	TI0	RI0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit7~Bit2 U0SM0, U0SM1, U0SM2, U0REN, U0TB8, U0RB8:UART related control bits, see UART0 function description.

Bit1 TI0: UART0 sends interrupt flag bits;

1= Synchronous mode ends at the eighth bit, or sends the stop bit asynchronously. Software can also clear it to zero.

0= ---

Bit0 RI0: UART0 receives interrupt flag bits;

1= Synchronous mode received the eighth bit, or asynchronous received. Software can also clear it to zero.

0= ---

TI 0 and RI 0 occupy the same interrupt vector (0023H) and require a query to determine whether to receive an interrupt or send an interrupt.

UART1 interrupt flag bit register SCON1, this register can not perform addressing by bit

0xEA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SCON1	U1SM0	U1SM1	U1SM2	U1REN	U1TB8	U1RB8	TI1	RI1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit7~Bit2 U1SM0, U1SM1, U1SM2, U1REN, U1TB8, U1RB8: UART1 related control bit, see UART1 function description

Bit1 TI1: UART1 sends interrupt flag bits.

1= Synchronous mode ends at the eighth bit, or sends the stop bit asynchronously. Software can also clear it to zero.

0= ---

Bit0 RI1: UART1 receives interrupt flag bits;

1= Synchronous mode received the eighth bit, or asynchronous received. Software can also clear it to zero.

0= UART1sends interrupt flag bits.

TI1 and RI1 occupy the same interrupt vector (0033H) and require a query to determine whether to receive an interrupt or send an interrupt.

Timer2 Interrupt Flag Bit Register T2IF (0xC9)

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
Reset Value	0	0	---	---	0	0	0	0

Bit7 TF2: Timer2 counter overflow interrupt flag bit;
1= The Timer2 counter overflows and needs to be cleared by software;
0= Timer2 counter has no overflow.
Bit6 T2EXIF: Timer2 external load flag bit;
1= Timer2's T2EX port generates a falling edge and needs to be cleared by software;
0= ---
Bit5 ---
Bit4 --- Timer2 compares the channel 3 flag bit;
Bit3 T2C3IF: Timer2 compares channel 3 {CCH3:CCL3}={TH2.TL2} and needs to be cleared by software.
1= ---
0= Timer2 compares the channel 2 flag bit;
Bit2 T2C2IF: Timer2 compares channel 2 {CCH2:CCL2}={TH2.TL2} and needs to be cleared by software.
1= ---
0= Timer2 compares the channel 1 flag bit;
Bit1 T2C1IF: Timer2 compares channel 1 {CCH1:CCL1}={TH1.TL1} and needs to be cleared by software.
1= ---
0= Timer2 compares the channel 0 flag bit;
Bit0 T2C0IF: Timer2 compares channel 0 {RLDH:RLDL}={TH2.TL2} and needs to be cleared by software.
1= ---
0=

P0 Interrupt Flag Bit Register P0EXTIF (0xB4)

0xB4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0EXTIF	---	---	P05IF	P04IF	---	---	---	---
R/W	R/W	R/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 ---

Bit5~Bit4 P0iIF: P0i port interrupt flag bit (i=4-5);

1= P0i port generate an interrupt and need to be cleared by software;

0= No interrupt is generated at P0i port.

Bit3~Bit0 ---

P1 Interrupt Flag Bit Register P1EXTIF (0xB5)

0xB5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P1EXTIF	P17IF	P16IF	P15IF	P14IF	P13IF	---	---	---
R/W	R/W	R/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 P1iIF: P1i port interrupt flag bit (i=3-7);

1= P1i port generate an interrupt and need to be cleared by software;

0= No interrupt is generated at P1i port

Bit2~Bit0 ---

P2 Interrupt Flag Bit Register P2EXTIF (0xB6)

0xB6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2EXTIF	---	P26IF	P25IF	P24IF	P23IF	P22IF	P21IF	---
R/W	R/W	R/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~Bit1 P2[i]IF: P2[i] port interrupt flag bit (i=1-6);

1= P2[i] port generate an interrupt and need to be cleared by software;

0= No interrupt is generated at P2[i] port

Bit0 ---

P3 Interrupt Flag Bit Register P3EXTIF (0xB7)

0xB7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P3EXTIF	---	P36IF	P35IF	---	---	P32IF	P31IF	P30IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit7 ---

Bit6~Bit5 P3[i]IF: P3[i] port interrupt flag bit (i=5-6);

1= P3[i] port generate an interrupt and need to be cleared by software;

0= No interrupt is generated at P3[i] port.

Bit4~Bit3 ---

Bit2~Bit0 P3[i]IF: P3[i] port interrupt flag bit (i=0-2);

1= P3[i] port generate an interrupt and need to be cleared by software;

0= No interrupt is generated at P3[i] port.

External Interrupt Flag Bit Register EIF2 (0xB2)

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 instruction, Read only;

1= SPI interrupt.(Automatically cleared after clearing the specific interrupt flag);

0= No SPI interrupt.

Bit6 I2CIF: I^2C Total interrupt instruction, Read only;

1= I^2C interrupt, (Automatically cleared after clearing the specific interrupt flag);

0= No I^2C interrupt.

Bit5 ---

Bit4 ADCIF: ADC interrupt flag;

1= ADC Conversion completed, need software clear;

0= ADC conversion is not complete.

Bit3 PWMIF: PWM Total interrupt instruction, Read only;

1= PWM interrupt, (Automatically cleared after clearing the specific interrupt flag);

0= No PWM interrupt.

Bit2 ---

Bit1 TF4: Timer4 counter overflow interrupt flag;

1= Timer4 enter the interrupt service routine, the hardware automatically clears;

0= Timer4 counter has no overflow.

Bit0 TF3: Timer3 counter overflow interrupt flag;

1= Timer3 enter the interrupt service routine, the hardware automatically clears;

0= Timer3 counter has no overflow.

SPI Interrupt Flag Bit Register SPSR (0xED)

0xED	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPSR	SPISIF	WCOL	---	---	---	---	SSICS	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 conflict error flag Read only;
1= SPI Write conflict error (Read SPSR first, then clear after reading/writing SPDR);
0= No Write conflict error.
Bit5 ---
Bit4 ---
Bit3~Bit2 ---
Bit1 SSICS: Must be 1 in SPI slave mode
Bit0 SSCEN: SPI master control mode NSSx (x=3,5-7)Output control bit.
1= NSSx outputs high when SPI is being in idle state;
0= NSSx outputs the content of register SSCR.

I²C Master 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: I² C active module reset control bit;
1= Reset the master module (I² C registers of the entire master module, including I2CMSR);
0= I² C Interrupt flag bit cleared to 0 in master control mode.
I2CMIF: I² C interrupt flag bit in master mode;
1= Send/receive complete in master mode, or a transmission error occurs. (software clear, write 0 to clear);
0= No interruptions were generated.
Bit6~Bit0 I²C Control and flag bits in master control mode, see I2CM description for details.

I²C Slave 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 ---

Bit2 SENDFIN: Sending complete flag bit in I²C slave mode (read only).

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 bit In I²C slave mode (read only).

1= As the transmitting device is already addressed or the master device is ready to receive data. (automatically cleared to zero after writing I2CSBUF)

0= ---

Bit0 RREQ: Receiving complete flag bit In I²C slave mode (read only).

1= Receive complete. (automatically write zero after reading I2CSBUF)

0= Uncompleted.

The related state bit of the I²C slave mode is also the interrupt flag bit.

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

7.3.4 Clearance of Interrupt Flag Bit

The clearance of interrupt flag bits in the system is divided into the following steps:

1) Hardware auto-clear (requires access to interrupt service program)
2) Software clear
3) Read/write to clear

- Flag bits automatically cleared by hardware

  The bits that support hardware auto clear are the interrupt flag bits generated by INT0, INT1, T0, T1, T3, and T4. The hardware auto clear flag conditions are: turn on the total interrupt enable bit EA=1 and turn on the corresponding interrupt enable bit, after generating an interrupt the system enters the corresponding interrupt service program and the flag bits are cleared automatically. If the interrupt enable is turned off, these flag bits can also be cleared using the software.

- Flag bits cleared by software

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

- Flag bits cleared by read and write operations

  There are flag bits in the system that do not write 0 to that flag bit to clear it, but need to read/write other registers to clear the flag bit. For example, the transfer completion flag bit SPISIF in the SPI interrupt flag register, after setting 1, you need to read SPSR and then read/write SPDR to clear it.

The software clear operation needs to be noted that 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 the read-modify-write operation.

For example, the PWMUIF interrupt flag bit register, which contains the upward comparison interrupts of channels PG0-PG5, these interrupt flag bits are not related to each other. When PG0 generates an upward comparison interrupt, the value of PWMUIF is 0x01, and the bit is cleared by a read-modify-write operation after entering the interrupt service program

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

This operation is implemented by first reading the value of PWMUIF back to the CPU and then performing the operation, and finally sending it back to P WMUIF. If the interrupt flag bit PWMUIF[1] of PG1 is set to 1 after the CPU reads it, and PWMUIF[1] is 0 when it is read, then it is also 0 when it is sent back to PWMUIF[1] after the operation. At this time, the up interrupt flag bit PWMUIF[1] already generated by PG1 is cleared.

To clear the interrupt flag bits of the above types, it is recommended to write 0 directly and 1 for other irrelevant flag bits:

```
PWMUIF = 0xFE;
```

Writing 1 to the irrelevant interrupt flag bits has no practical effect.

7.3.5 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: SPI interrupt flag bit register transmission completed flag bit SPISIF in debug mode

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

void SPI_int (void) interrupt SPI_VECTOR     // SPI interrupting service procedures
{
        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.

8. TIMING COUNTER 0/1 (Timer0/1)

The type and structure of timer 0 is similar to those of timer 1, they are two 16-bit timers. Timer 1 has three operating modes, and Timer 0 has four operating modes .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 the corresponding input pin(T0 or T1) is detected.

8.1 Overview

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

Each timer consists of two 8-bit registers:{TH0(0x8C): TL0(0x8A)} and{TH1(0x8D): TL1(0x8B)}. Timer 0, 1 works in four identical modes. The Timer 0 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 auto-reload time/counter, reload from THx
3	1	1	TL0,TH0is 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.

8.2 Timer 0/1 Register

8.2.1 Timer 0/1 mode register TMOD

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

Bit7 GATE1: Timer 1 gate control bit;
1= Enable;
0= Disable.
Bit6 CT1: Timer 1 timing / count selection bit;
1= Counting;
0= Timing.
Bit5~ Bit4 T1M1, T1M0: Timer 1 mode selection bit;
00= Mode 0, 13-bit time/counter;
01= Mode 1, 16-bit time/counter;
10= Mode 2, 8-bit auto-reload time/counter;
11= Mode 3, Timer1 stops.
Bit3 GATE0: Timer 0 gate control bit
1= Enable;
0= Disable.
Bit2 CT0: Timer 0 timing / counting selection bit;
1= Counting;
0= Timing.
Bit1~ Bit0 T0M1, T0M0: Timer 0 mode selection bit;
00= Mode 0, 13-bit time/counter;
01= Mode 1, 16-bit time/counter;
10= Mode 2, 8-bit auto-reload time/counter;
11= Mode 3, two separate 8-bit time/counter.

8.2.2 Timer0/1 Control Register TCON (Addressable)

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 overflows the interrupt flag bit;
1= Timer1 counter Overflow, enter the interrupt service program hardware automatically zero;
0= Timer1 counter has no overflow.
Bit6 TR1: Timer1 runs the control bit;
1= Timer1 starts;
0= Timer1 stopped.
Bit5 TF0: Timer0 counter overflows the interrupt flag bit;
1= Timer0 enter the interrupt service program hardware automatically zero;
0= Timer0 counter has no overflow.
Bit4 TR0: Timer0 runs the control bit;
1= Timer0 starts;
0= Timer0 stopped.
Bit3 IE1: External interrupt 1 flag bit;
1= External interrupt 1 produces an interrupt, and the hardware of the interrupt service program is automatically cleared;
0= External interrupt 1 did not cause an interrupt.
Bit2 IT1: External interrupt 1 trigger mode control bit;
1= The falling edge triggers;
0= Low-level trigger.
Bit1 IE0: External interrupt 0 flag bit;
1= External interrupt 0 produces an interrupt, and the hardware of the interrupt service program is automatically cleared.
0= External interrupt 0 did not cause an interrupt.
Bit0 IT0: External interrupt 0 trigger mode control bit;
1= The falling edge triggers;
0= Low-level trigger.

8.2.3 Timer 0 Data Register Lower Bit TL0

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

Bit7~ Bit0 TL07-TL00: Timer 0 lower-bit data register (also as lower bit of the counter).

8.2.4 Timer 0 Data Register Higher Bit TH0

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

Bit7~Bit0 TH07-TH00: Timer 0 higher-bit data register (also as higher bit of the counter).

8.2.5 Timer 1 Data Register Lower Bit TL1

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

Bit7~Bit0 TL17-TL10: Timer 1 lower-bit data register (also as lower bit of the counter).

8.2.6 Timer 1 Data Register Higher Bit TH1

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

Bit7~Bit0 TH17-TH10: Timer 1 higher-bit data register (also as higher bit of the counter) .

8.2.7 Function Clock Control Register CKCON

0x8E	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CKCON	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~Bit6 WTS1-WTS0: WDT overflow time selection bit.
00= 2^{17}*Tsys
01= 2^{20}*Tsys
10= 2^{23}*Tsys
11= 2^{26}*Tsys
Bit5 Unused.
Bit4 T1M: Clock source selection bit for Timer1.
1= Fsys/4
0= Fsys/12
Bit3 T0M: Clock source selection bit for Timer0.
1= Fsys/4
0= Fsys/12
Bit2~ Bit0 ---

8.3 Timer 0/1 Interrupt

Timer 0/1 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:

Interrupt mask register IE (0xA8)

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: Overall interrupt enable bit;
1= Enable all un-masked interrupt;
0= Disable all interrupt.
Bit6 ES1: UART1 interrupt enable bit;
1= Enable UART1 interrupt;
0= Disable UART1 interrupt.
Bit5 ET2: TIMER2 interrupt enable bit;
1= Enable TIMER2 interrupt;
0= Disable TIMER2 interrupt.
Bit4 ES0: UART0 interrupt enable bit;
1= Enable UART0 interrupt;
0= Disable UART0 interrupt.
Bit3 ET1: TIMER1 interrupt enable bit;
1= Enable TIMER1 interrupt;
0= Disable TIMER1 interrupt.
Bit2 EX1: External interrupt 1 enable bit;
1= Enable external interrupt 1;
0= Disable external interrupt 1.
Bit1 ET0: TIMER0 interrupt enable bit;
1= Enable TIMER0 interrupt;
0= Disable TIMER0 interrupt.
Bit0 EX0: External interrupt 0 enable bit;
1= Enable external interrupt 0;
0= Disable external interrupt 0.

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 ---

Bit6 PS1: UART1 interrupt priority control bit;
1= Set to advanced interrupt;
0= Set to low level interrupt;

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

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

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

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

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

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

Timer0/1, INT0/1 Interrupt Flag Register TCON, can perform addressing by bit.

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 overflows the interrupt flag bit;
1= Timer1 counter Overflow, enter the interrupt service program hardware automatically zero;
0= Timer1 counter has no overflow.
Bit6 TR1: Timer1 runs the control bit;
1= Timer1 starts;
0= Timer1 stopped.
Bit5 TF0: Timer0 counter overflows the interrupt flag bit;
1= Timer0 enter the interrupt service program hardware automatically zero;
0= Timer0 counter has no overflow.
Bit4 TR0: Timer0 runs the control bit;
1= Timer0 starts;
0= Timer0 stopped.
Bit3 IE1: External interrupt 1 flag bit;
1= External interrupt 1 produces an interrupt, and the hardware of the interrupt service program is automatically cleared;
0= External interrupt 1 did not cause an interrupt.
Bit2 IT1: External interrupt 1 trigger mode control bit;
1= The falling edge triggers;
0= Low-level trigger.
Bit1 IE0: External interrupt 0 flag bit;
1= External interrupt 0 produces an interrupt, and the hardware of the interrupt service program is automatically cleared.
0= External interrupt 0 did not cause an interrupt.
Bit0 IT0: External interrupt 0 trigger mode control bit;
1= The falling edge triggers;
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 when there is no interrupt enabled.

8.4 Timer 0 Operating Mode

8.4.1 T0 - Mode 0 (13-Bit Timing/Counting Mode)

In this mode, Timer 0 is a 13-bit register. The timer 0 interrupt flag TF0 is set when all bits of the counter are toggled from 1 to 0. When TCON.4=1 and TMOD.3=0 or TCON.4=1, TMOD.3=1, T0G=1, the count input is enabled to timer 0. (SettingsTMOD.3=1 allows timer 0 to be controlled by external pin T0G for pulse width measurement). The 13-bit register consists of TH0 and TL0, which are the lower 5 bits.

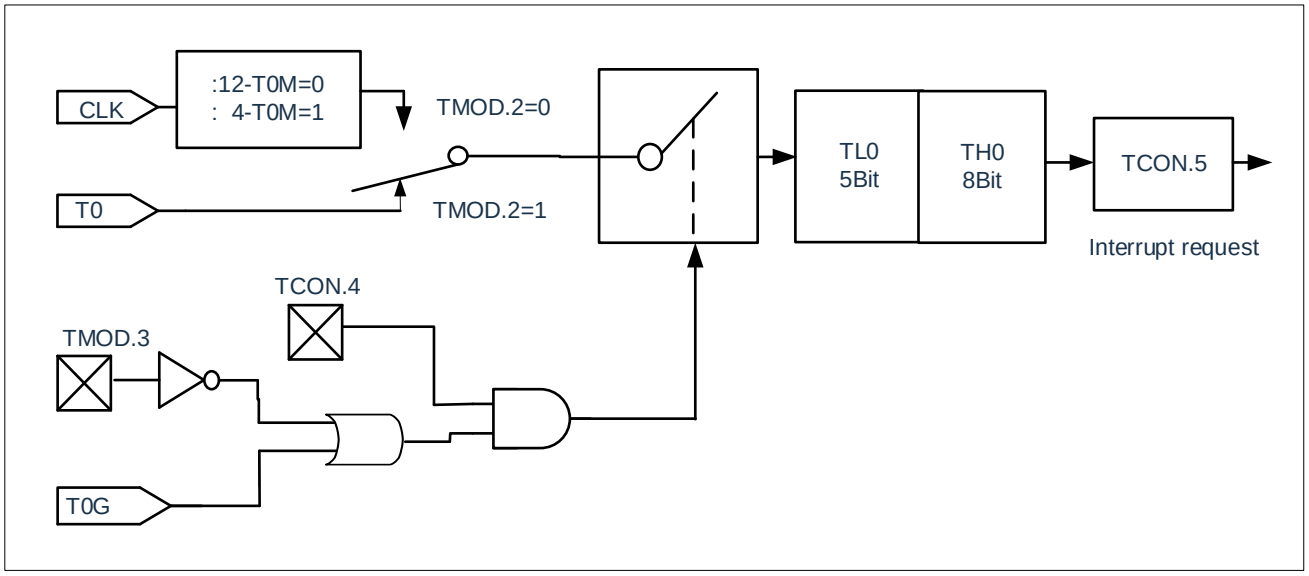


Figure 8-1: Timer0,Mode 0: 13-bit timing/counter

8.4.2 T0 -Mode 1 (16-bit timer/count mode)

Mode 1 is the same as Mode 0 except that Timer 1 Data Register 16 bits are all running in Mode 1. Mode 1 is shown below.

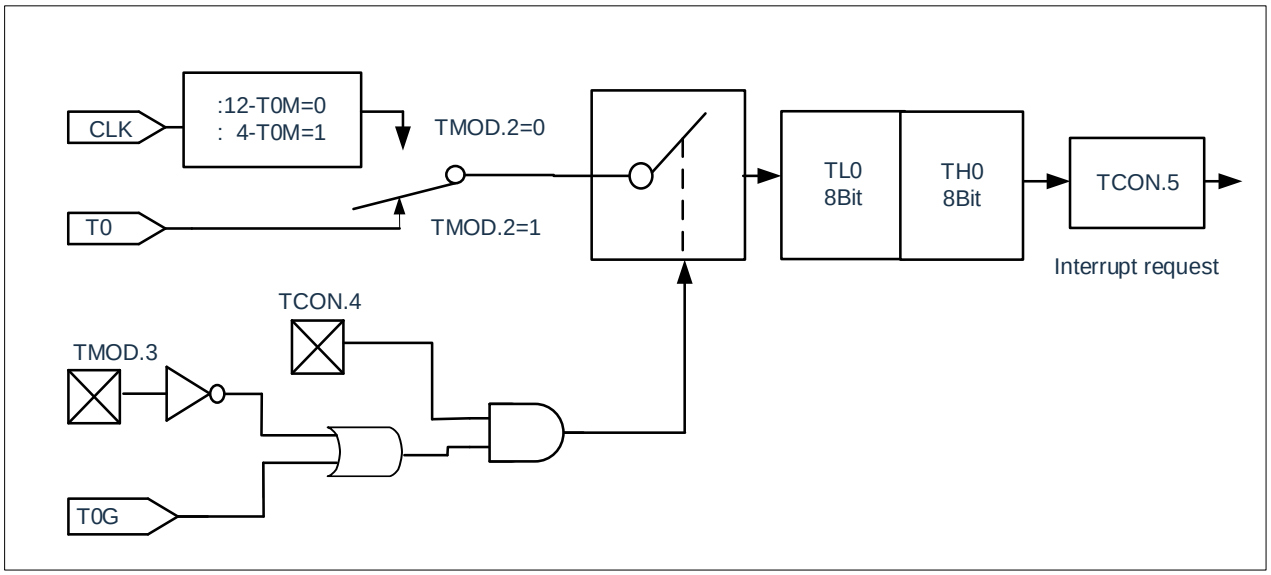


Figure 8-2: Timer0,Mode 1: 16-bit timing/counter

8.4.3 T0 - Mode 2 (8-bit auto reload timing / counting mode)

In the Timer 2,Timer Register is an 8-bit counter (TL0) with auto-reload mode as shown below. An overflow from TL0 not only sets TF0, but also reloads the contents of TH0 from software to TL0. The value of TH0 remains unchanged during reassembly.

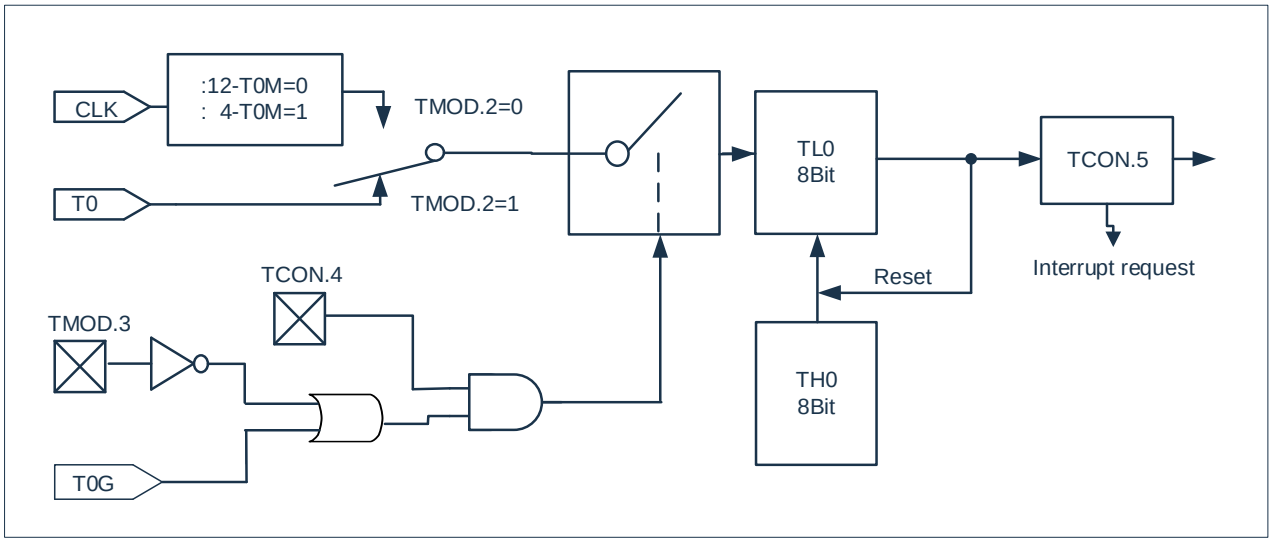


Figure 8-3: Timer0, Mode 2: 8-bit timing/counter

8.4.4 T0 - Mode3 (Two separate 8-bit timer/counters)

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

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

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

Mode 3 is available when two 8-bit timer/counters are required. When Timer 0 is in Mode 3, Timer 1 can be turned off by switching to its own Mode 3, or it can still be used as a Baud Rate Generator by the serial channel, or if no Timer 1 interrupt is required. In the application.

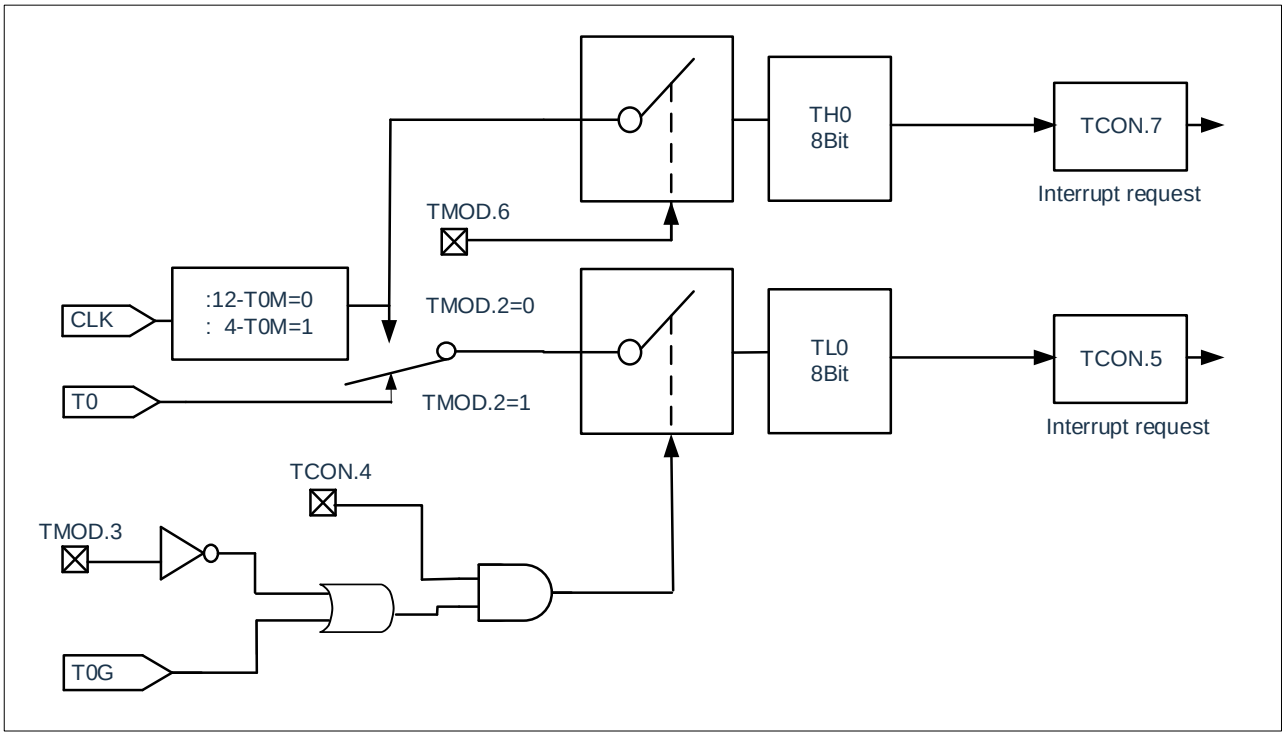


Figure 8-4: Timer0, Mode 3: Two 8-bit timer/counters

8.5 Timer 1 Operation Mode

8.5.1 T1 -Mode0 (13-bit Timer/Count Mode)

In this mode, Timer 1 is a 13-bit register. When all the bits of the counter are inverted from 1 to 0, the Timer 1 interrupt flag TF1 is set. When TCON.6=1 and TMOD.7=0 or when TCON.6=1, TMOD.7=1 and T1G=1, the count input is enabled to Timer 1.(Set TMOD.7=1 to allow Timer 1 to be controlled by external pin T1G for pulse width measurement). The 13-bit register consists of TH1 8 bits and TL1 lower 5 bits. The upper third of TL1 should be ignored.

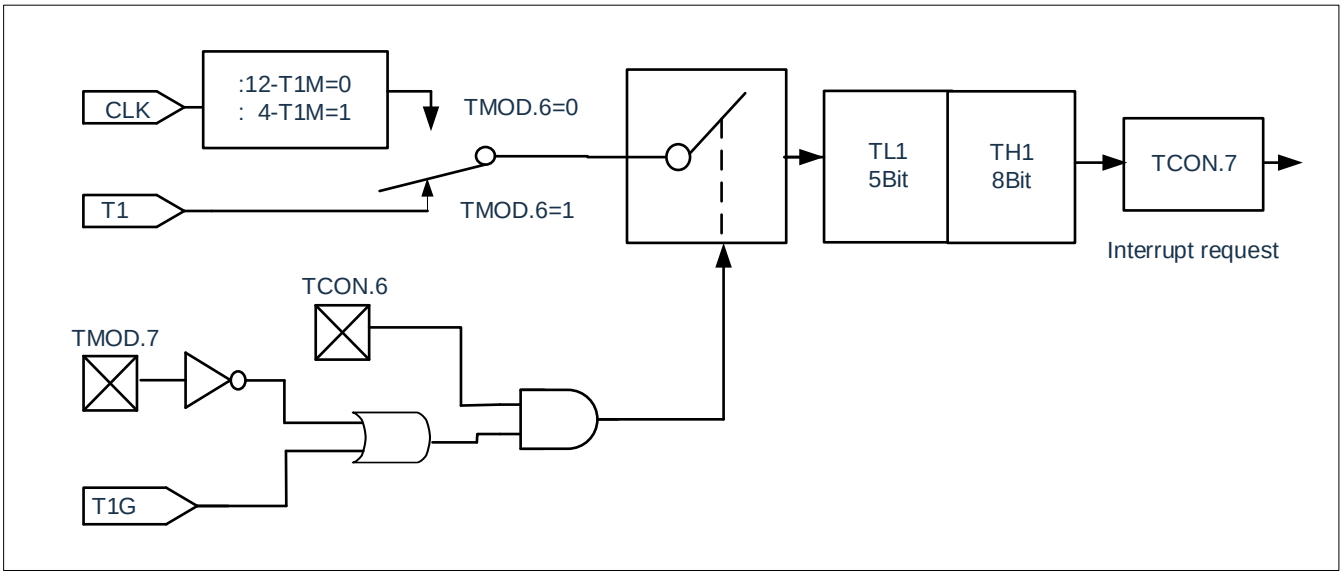


Figure 8-5: Timer1, Mode 0: 13-bit timing/counter

8.5.2 T1 - Mode1 (16-bit Timer/Count Mode)

Mode 1 is the same as Mode 0 except that the Timer 1 Register 16 bits of Mode 1 are all running. Mode 1 is shown below.

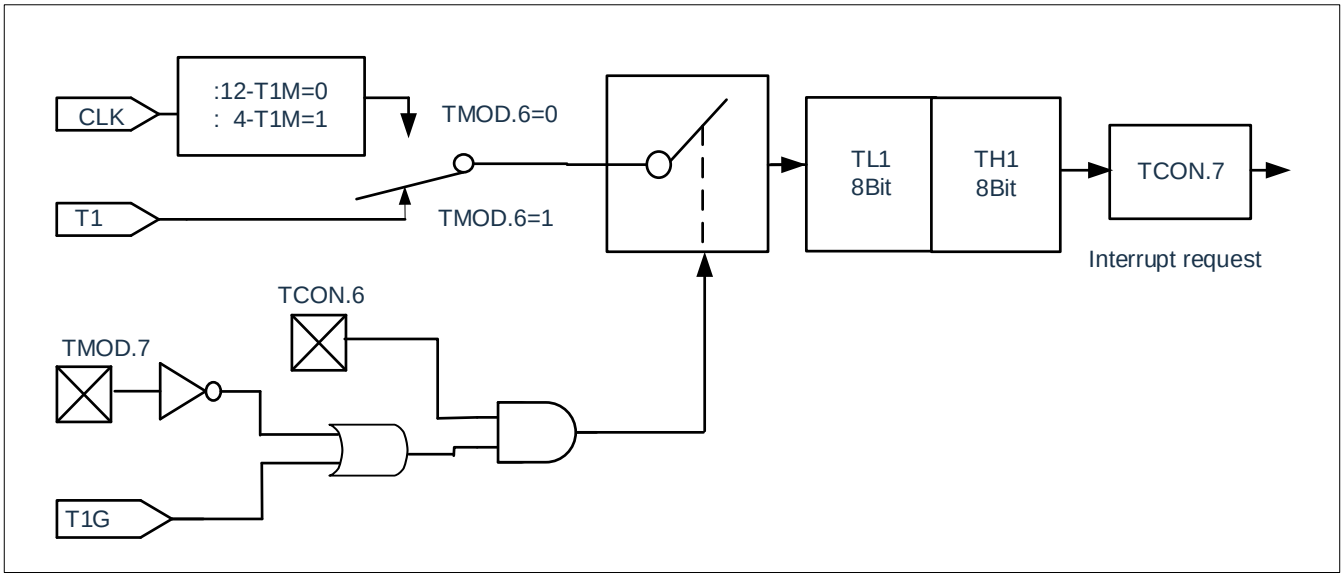


Figure 8-6: Timer1, Mode 1: 16-bit timing/counter

8.5.3 T1 -Mode2 (8-bit Auto Reload Timing/Counting Mode)

In Mode 2, the Timer 1 register is an 8-bit counter (TL1) with auto-reload mode as shown below. An overflow from TL1 not only sets TF1, but also reloads the contents of TH1 from software to TL1. The value of TH1 remains unchanged during reassembly.

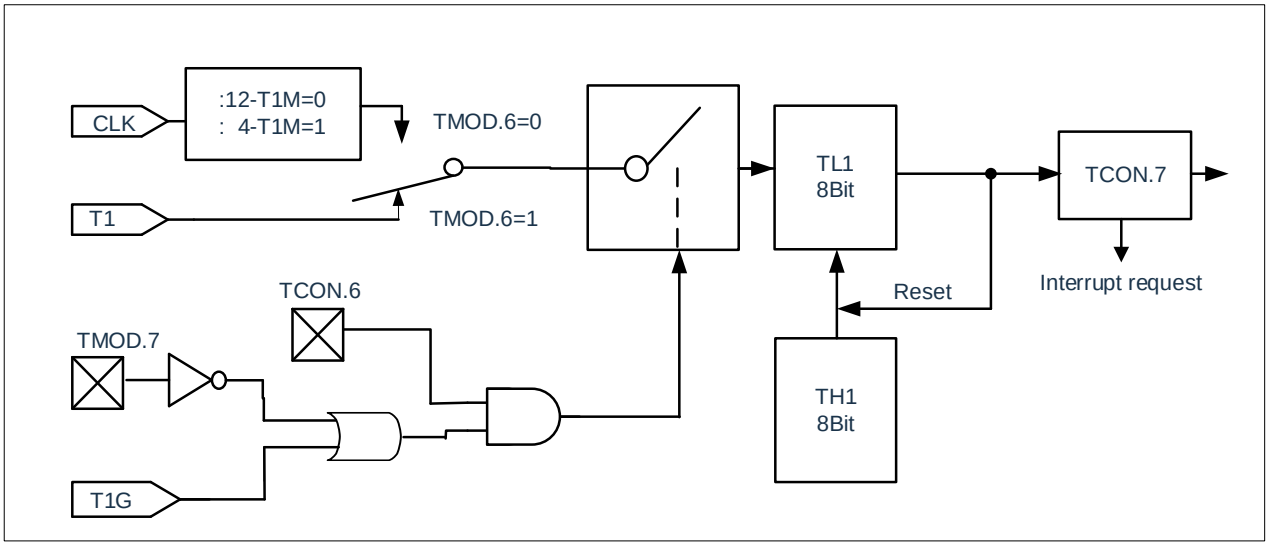


Figure 8-7: Timer1, Mode 2: 8-bit timing/counter (automatic reload)

8.5.4 T1 - Mode3 (Stop Counting)

Timer 1 in Mode 3 stops counting and has the same effect as setting TR1=0.

9. TIMING COUNTER 2 (TIMER2)

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

9.1 Overview

The block diagram below shows the general configuration of Timer 2 with additional compare /capture /reloads registers.

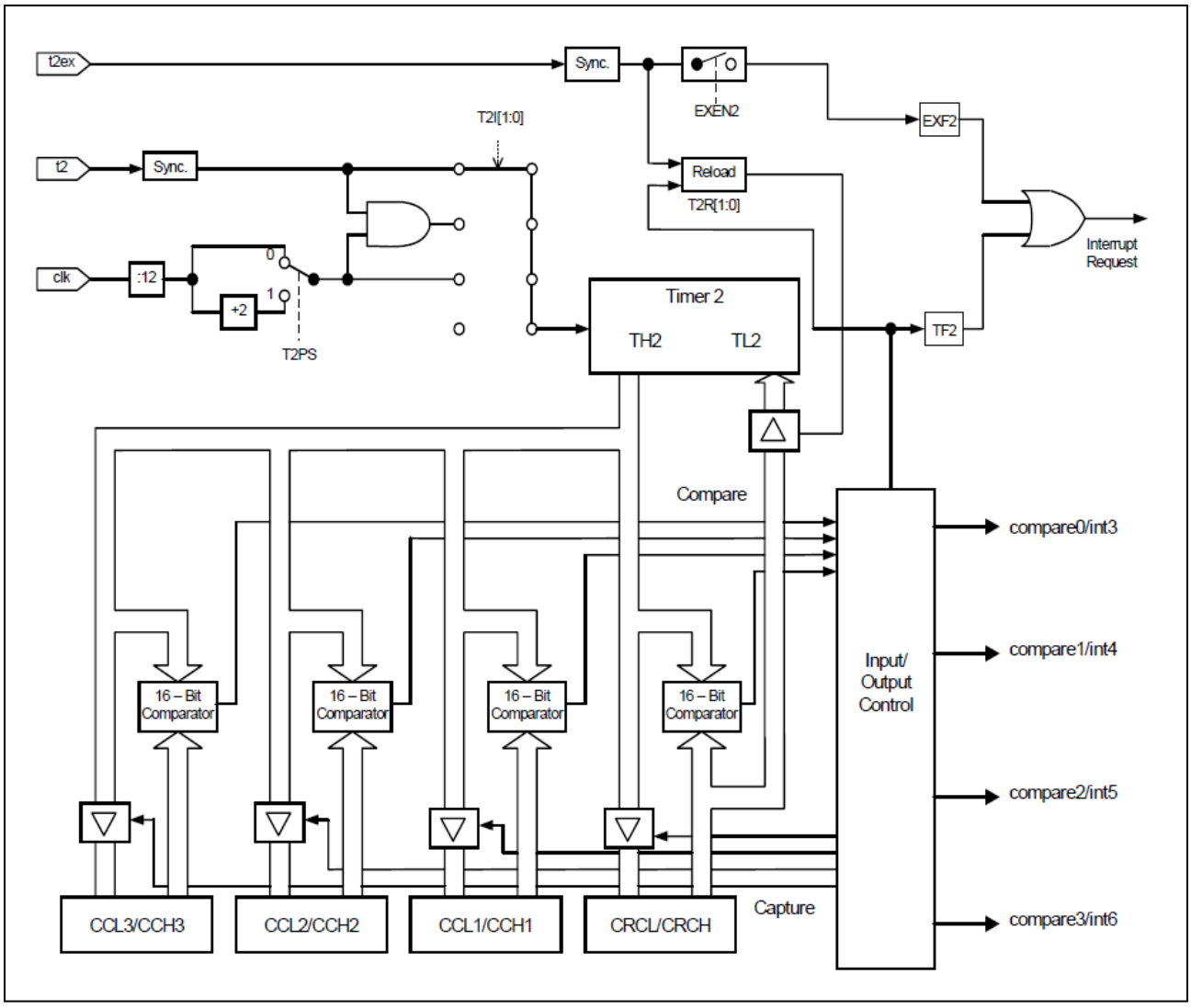


Figure 9-1: Compare/capture Module

9.2 Timer2 Register

Registers of Timer 2 with compare/capture function are as follows:

9.2.1 T2CON

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

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

Bit6 I3FR: Capture channel 0 input single edge and select with compare interrupt moment select bit;
Capture channel 0 mode:
1= Rising edge captured RLDL/RLDH register;
0= Falling edge captured RLDL/RLDH registers.
Compare channel 0 mode:
1= TL2/TH2 and RLDL/RLDH generate interrupts from unequal to equal moments;
0= TL2/TH2 and RLDL/RLDH generate interrupts from equal to unequal moments;

Bit5 CAPES: Capture channel 1-3 input single edge selection (effective for capture channels 1-3 together)
0= Rising edge captured CC0L/CC0H-CC3L/CC3H registers;
1= Falling edge captured CC0L/CC0H-CC3L/CC3H registers.

Bit4~Bit3 T2R1-T2R0: Timer2 Load mode selection bit;
0x= Reinstallation ban;
10= Load mode 1: automatic reloading in case of Timer2 overflow;
11= Load mode 2: Reload on the falling edge of the T2EX pin.

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

Bit1~ Bit0 T2I1-T2I0: Timer2 clock input selection bit;
00= Timer2 stops;
01= Frequency division of the system clock (T2PS controls the frequency division selection);
10= External pin T2 for event input (event counting mode)
11= External pin T2 for gating input (gated timing mode)

9.2.2 Timer2 Data Register Lower Bit TL2

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

Bit7~Bit0 TL27-TL20: Timer 2 Lower-bit data register (also as lower bit of the counter).

9.2.3 Timer2 Data Register Higher Bit TH2

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

Bit7~Bit0 TH27-TH20: Timer 2 higher-bit data register (also as lower bit of the counter).

9.2.4 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 RLDL7- RLDL0: Timer2 compare/capture/reload register low 8-bit

9.2.5 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 RLDH7- RLDH0: Timer2 compare/capture/reload register high 8-bit.

9.2.6 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 CCL17-CCL10: Timer2 compare/capture channel1 register low 8-bit.

9.2.7 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 CCH17-CCH10: Timer2 compare/capture channel1 register high 8-bit.

9.2.8 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 CCL27-CCL20: Timer2 compare/capture channel2 register low 8-bit

9.2.9 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 CCH27-CCH20: Timer2 compare/capture channel2 register high 8-bit

9.2.10 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 CCL37-CCL30: Timer2 compare/capture channel3 register low 8-bit

9.2.11 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 CCH37-CCH30: Timer2 compare/capture channel3 register high 8-bit

9.2.12 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 bits;
- 00= Capture/comparison disable;
- 01= Capture operation triggered on the rising or falling edge of channel 3 (CAPES selection);
- 10= Comparison mode enable;
- 11= The capture operation is triggered on write CCL3 or a double edge trigger on channel 3.

Bit5~Bit4 CMH2-CML2: Capture/compare mode control bits;
- 00= Capture/comparison disable;
- 01= Capture operation triggered on the rising or falling edge of channel 2 (CAPES selection);
- 10= Comparison mode enable;
- 11= The capture operation is triggered on write CCL2 or a double edge trigger on channel 2.

Bit3~Bit2 CMH1-CML1: Capture/compare mode control bits;
- 00= Capture/comparison disable;
- 01= Capture operation triggered on the rising or falling edge of channel 1 (CAPES selection);
- 10= Comparison mode enable;
- 11= The capture operation is triggered on a write to CCL1 or a double edge trigger on channel 1.

Bit1~Bit0 CMH0-CML0: Capture/compare mode control bits;
- 00= Capture/comparison disable;
- 01= Capture operation triggered on the rising or falling edge of channel 0 (I3FR selection);
- 10= Comparison mode enable;
- 11= The capture operation is triggered on write RLDL or a double edge trigger on channel 0.

9.2.13 Timer2 capture channel input choose register0 CAPCR0

0xD6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CAPCR0	CAP0CR7	CAP0CR6	CAP0CR5	CAP0CR4	CAP0CR3	CAP0CR2	CAP0CR1	CAP0CR0
R/W	R/W	R/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 CAP0CR7-CAP0CR4: Capture channel1 input control bit.
0000= Select IC0 0101=Select IC5
0001= Select IC1 0110=Select IC6
0010= Select IC2 0111=Select IC7
0011= Select IC3 1000=Select IC8
0100= Select IC4 others=Select IC0
Bit3~ Bit0 CAP0CR3-CAP0CR0: Capture channel0 input control bit.
0000= Select IC0 0101= Select IC5
0001= Select IC1 0110= Select IC6
0010= Select IC2 0111= Select IC7
0011= Select IC3 1000 =Select IC8
0100= Select IC4 others= Select IC0

9.2.14 Timer2 capture channel input choose register1 CAPCR1

0xD7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CAPCR1	CAP1CR7	CAP1CR6	CAP1CR5	CAP1CR4	CAP1CR3	CAP1CR2	CAP1CR1	CAP1CR0
R/W	R/W	R/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 CAP1CR7-CAP1CR4: Capture channel3 input control bit.
0000= Select IC0 0101= Select IC5
0001= Select IC1 0110= Select IC6
0010= Select IC2 0111= Select IC7
0011= Select IC3 1000= Select IC8
0100= Select IC4 others = Select IC0
Bit3~ Bit0 CAP1CR3-CAP1CR0: Capture channel2 input control bit.
0000= Select IC0 0101= Select IC5
0001= Select IC1 0110= Select IC6
0010= Select IC2 0111= Select IC7
0011= Select IC3 1000= Select IC8
0100= Select IC4 others = Select IC0

9.3 Timer2 Interrupt

Timer 2 Interrupt related bits are shown below. An interrupt can be turned on/off by IE register, and set into high/low priority group by IP register

Interrupt enable register IE (0xA8)

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: Overall interrupt enable bit;
1= Enable all un-masked interrupt;
0= Disable all interrupt.
Bit6 ES1: UART1 interrupt enable bit;
1= Enable UART1 interrupt;
0= Disable UART1 interrupt.
Bit5 ET2: TIMER2 interrupt enable bit;
1= Enable TIMER2 interrupt;
0= Disable TIMER2 interrupt.
Bit4 ES0: UART0 interrupt enable bit;
1= Enable UART0 interrupt;
0= Disable UART0 interrupt.
Bit3 ET1: TIMER1 interrupt enable bit;
1= Enable TIMER1 interrupt;
0= Disable TIMER1 interrupt.
Bit2 EX1: External interrupt 1 enable bit;
1= Enable external interrupt 1;
0= Disable external interrupt 1.
Bit1 ET0: TIMER0 interrupt enable bit;
1= Enable TIMER0 interrupt;
0= Disable TIMER0 interrupt.
Bit0 EX0: External interrupt 0 enable bit;
1= Enable external interrupt 0;
0= Disable external interrupt 0.

Interrupt Prioity 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 ---

Bit6 PS1: UART1 interrupt priority control bit;
1= Set to advanced interrupt;
0= Set to low level interrupt;

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

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

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

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

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

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

Timer2 Interrupt Enable Control Register T2IE

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

Bit7 T2OVIE: Timer2 overflow interrupt enable bit;
1= Enable interrupt;
0= Disable interrupt.
Bit6 T2EXIE: Timer2 external load interrupt enable bit;
1= Enable interrupt;
0= Disable interrupt.
Bit5~Bit4 ---
Bit3 T2C3IE: Timer2 compare3 interrupt enable bit;
1= Enable interrupt;
0= Disable interrupt.
Bit2 T2C2IE: Timer2 compare2 interrupt enable bit;
1= Enable interrupt;
0= Disable interrupt.
Bit1 T2C1IE: Timer2 compare1 interrupt enable bit;
1= Enable interrupt;
0= Disable interrupt.
Bit0 T2C0IE: Timer2 compare0 interrupt enable bit;
1= Enable interrupt;
0= Disable interrupt.

If enable timer2 interrupt, you need to set T2IE relevant bits and set ET2=1 (IE.5=1).

Timer2 Interrupt Flag Register T2IF (0xC9)

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
Reset Value	0	0	---	---	0	0	0	0

Bit7 TF2: Timer2 counter overflow interrupt flag bit;
1= Timer2 counter overflow, requiring software clearing;
0= Timer2 counters are not overflowing.

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

Bit5 ---

Bit4 ---

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

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

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

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

9.3.1 Timer Interrupt

When the Timer2 timer overflows, an interrupt is generated. TF2 flag will be set to 1.

9.3.2 External Trigger Interrupt

An interrupt is also generated at falling edge of the T2EX pin, while T2EXIE bit is set. T2EXIF flag will set to 1.

9.3.3 Compare Interrupt

All 4 comparison channels support comparison interrupts.

Compare Channel 0 selects the moment when the compare interrupt is generated, and generating the interrupt sets the compare channel 0 interrupt flag T2C0IF to 1.

- I3FR = 0 when TL2/TH2 and RLDL/RLDH generate interrupts from unequal to equal moments;
- I3FR = 1 when TL2/TH2 and RLDL/RLDH generate interrupts from equal to unequal moments;

The comparison channels 1-3 cannot select the interrupt generation moment, fixed as TL2/TH2 and CCxL/CCxH generate interrupts from unequal to equal moments. If an interrupt is generated, the interrupt flag T2CxIF of the corresponding comparison channel x is set to 1.

Timer2 all related interrupts share a common interrupt vector, after entering the interrupt service program, you need to judge the related flag bit to determine which situation has generated the interrupt.

9.3.4 Capture interrupt

All four capture channels support external capture interrupts. When a capture operation is generated, the interrupt flag of the corresponding capture channel is set to T2CxIF.

Note: that the write capture method does not generate interrupts.

9.4 Timer2 Function Description

Timer 2 is a 16-bit register that can operate as a timer, event counter or gated timer.

9.4.1 T2 Timing Mode

In timer function, the clock source is derived from system frequency .The prescaler offers the possibility of selecting the 1/12 or 1/24 of system frequency. Thus, the 16-bit timer register (consisted of TH2 and TL2) is either incremented in every 12 clock periods or in every 24 clock periods. The prescaler is selected by bit T2PS of T2CON.

9.4.2 T2 Reload Mode

The reload mode for Timer 2 is selected by the T2R0 and T2R1 bits of T2CON.

In mode 1, When counter 2 is inverted by all 1s to 0 (counter overflow), not only TF2 is set, but timer 2 is registered. The device automatically loads the 16-bit value from the RLDL/RLDH register. The required RLDL/RLDH value can be preset by software. The reload occurs in the same clock cycle as TF2 is set, thus overwriting the count value of 0x0000.

In mode 2, the 16-bit reassembly operation from the RLDL/RLDH register is triggered by the descent edge of the corresponding T2EX input pin. In addition, if T2EXIE=1, the T2EXIF flag will be set to 1. If timer 2 total interrupts enable (ET2=1), an interrupt occurs

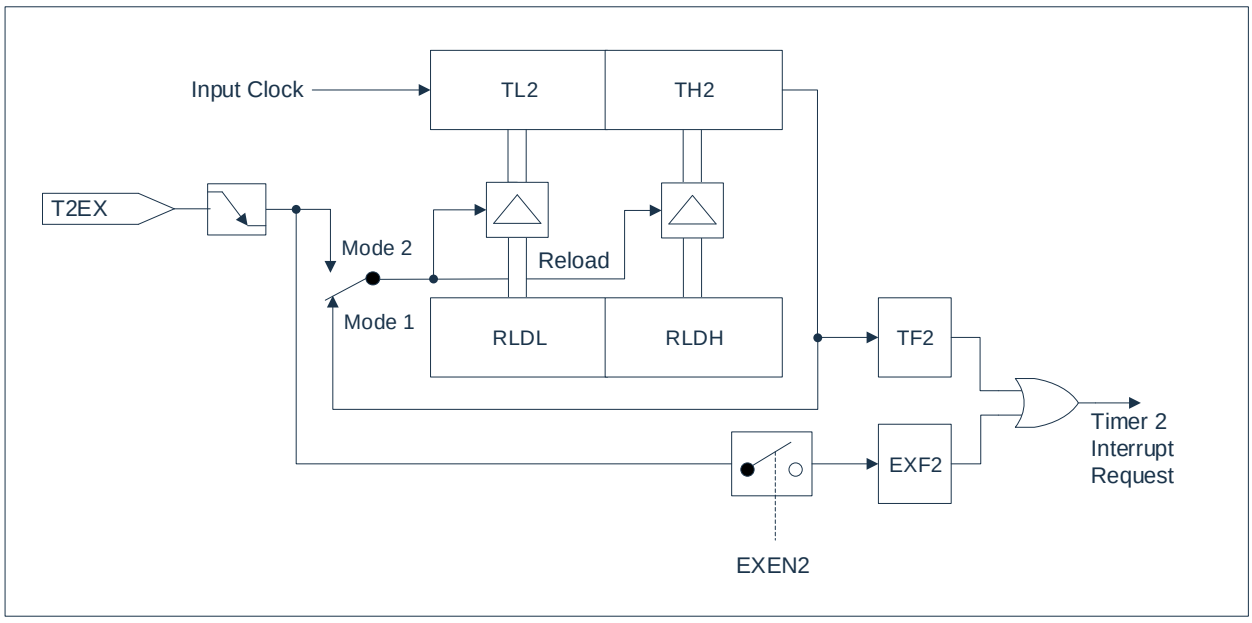


Figure 9-2: Timer2 reload mode block diagram

9.4.3 Timer2 Gated Timing Mode

For gated timer function, the external input pin T2 is used as the input gating for Timer 2. If the T2 pin is high, the internal clock input is selected to the timer, and a low T2 pin terminates the count. This function is often used to measure the pulse width.

9.4.4 Timer2 Event Counting Mode

While in the counter function, Timer 2 is incremented with a 1 to 0 transition on its corresponding external input pin T2. In this function, the external input is sampled at each clock cycle. The count is increased when the sampled input shows high in one cycle and low in the next cycle. The new count value appears in the timer register when the change from high to low on the T2 pin is detected again in the next cycle.

9.5 Compare/PWM function

The compare function of a timer/register combination can be described as follows:

The 16-bit value stored in a compare/capture register is compared with the contents of the timer register. If the count value in the timer register matches the stored value, an appropriate output signal is generated at a corresponding port pin, and an interrupt is requested.

Variation of this time stamp somehow changes the wave of a rectangular output signal at a port pin. This may as a variation of the duty cycle of a periodic signal be used for pulse width modulation(PWM) as well as for a continually controlled generation of any kind of square waveforms.

The channel of the compare function output is CC0, CC1, CC2, CC3. Output signals corresponding to the comparison of the {TH2, TL2} registers of the 16-bit compare registers {RLDH, RLDL}, {CCH1, CCL1}, {CCH2, CCL2}, {CCH3, CCL3} and Timer2, respectively.

The compare function consists of two modes: compare modes 0 and 1 are selected by bit T2CM in the special function register T2CON.Two compare modes are implemented to cover a wide range of possible applications.

9.5.1 Compare Mode 0

In mode 0, upon matching the timer and compare register contents, 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.

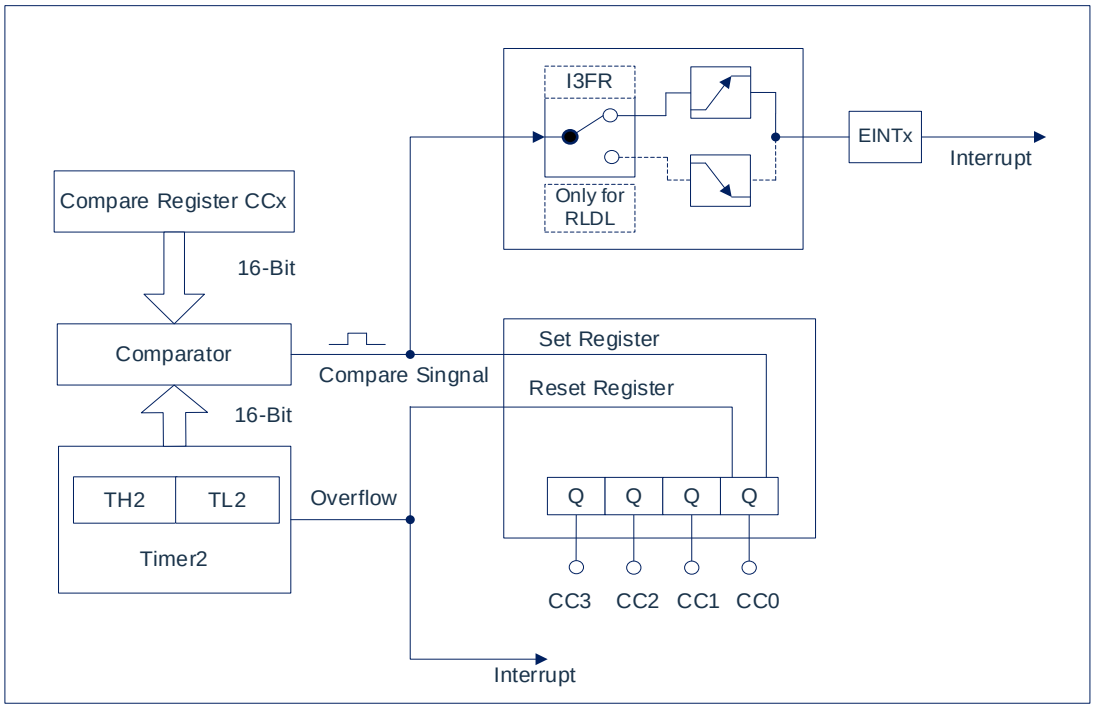


Figure 9-3: Timer2 compare mode 0 block diagram

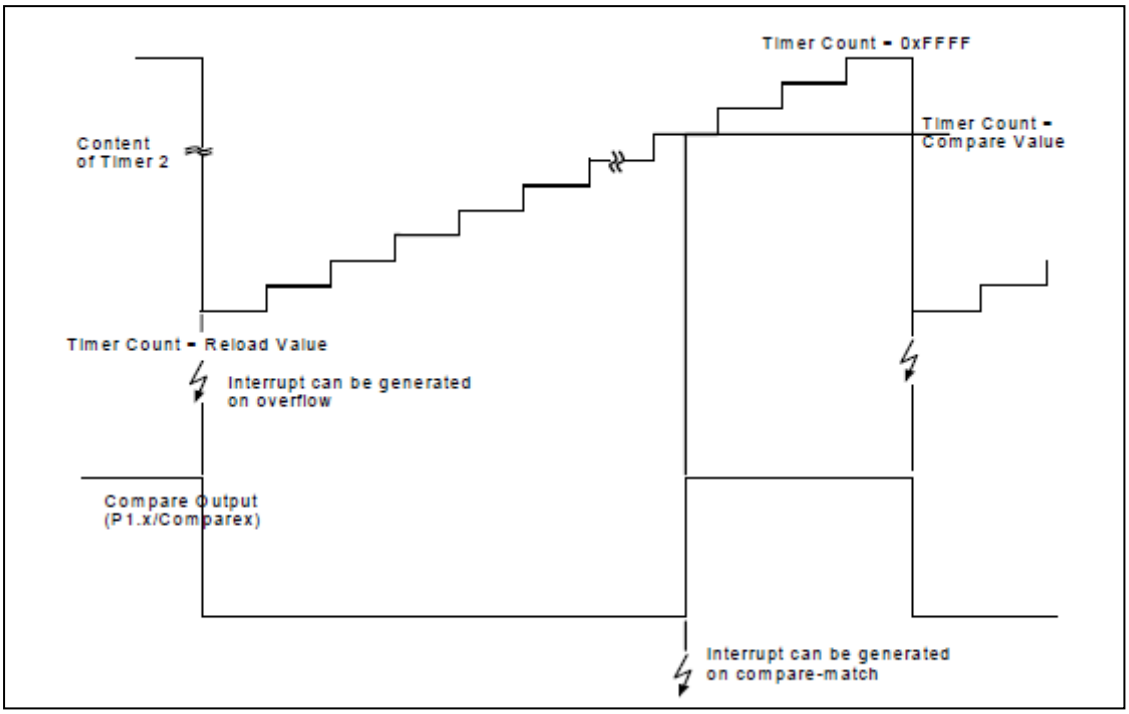


Figure 9-4: Timer 2 compare mode 0 function

9.5.2 Compare Mode 1

In compare mode 1, the software adaptively determines the transition of the output signal. It is commonly used when output signals are not related to a constant signal period. In compare mode 1, both transitions of a signal can be controlled. Compare outputs in this mode can be considered as high speed outputs do not depend on the CPU activity.

If mode 1 is enabled, and the software writes to an appropriate output register of PORT 1, a new value will not appear at the output pin until the next compare match occurs. User can select this way whether the output signal should make a new transition or should keep its old value, until the Timer 2 counter matches the stored compare value. Figure below shows a functional diagram of Timer 2 in compare mode 1.

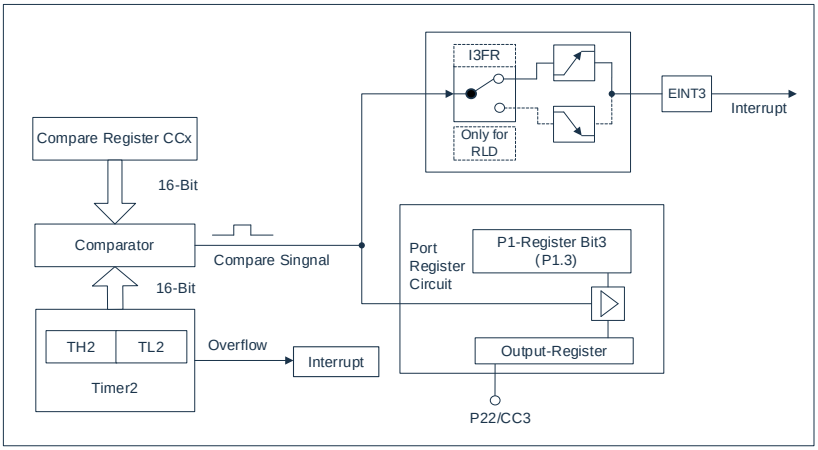


Figure 9-5: Timer 2 compare mode 1 block diagram

Note: Compare mode 1 is only valid for compare channel 3.

9.6 Capture Function

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} on the fly.

Capture channel 0~3 can select any of the input pins IC0-IC8 as the input source signal. The following is the control method for the corresponding capture channel:

Capture channel	Control bit	Input pin
0	CAPCR0[3:0]	=0: IC0 =1: IC1 …. =8: IC8 = others: IC0
1	CAPCR0[7:4]	As above
2	CAPCR1[3:0]	As above
3	CAPCR1[7:4]	As above

9.6.1 Capture Mode 0

The external events that result in capture in this mode are:

- For CC registers 1 to 3: captures positive or negative jumps on channels 1-3, and can also support positive and negative jumps.
- For RLD register: captures positive or negative jumps on channel 0. Positive and negative jumps can also be supported.

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

Whether the capture operation on capture channels 1-3 is triggered by positive or negative jumps depends on the CAPES bit of T2CON. The selected jumping method for capture channels 1-3 is the same

In addition, capture channels 0-3 also support double-jump capture operation by selecting the corresponding operating mode control bit of CCEN register as 11. It should be noted that this working mode also supports capture mode 1, i.e. write operation can generate capture action.

All external capture events for channels 0-3 can generate interrupts.

9.6.2 Capture Mode 1

In mode 1 capture occurs in response to a write instruction to the low order byte of a capture register. The write-to register signal (e.g. write to RLDL) is used to initiate a capture. The value written to the dedicated capture register is irrelevant for this function. The timer 2 contents will be latched into the appropriate capture register in the cycle following the write instruction. In this mode, no interrupt request will be generated.

Figures below show functional diagrams of the timer 2 capture function.

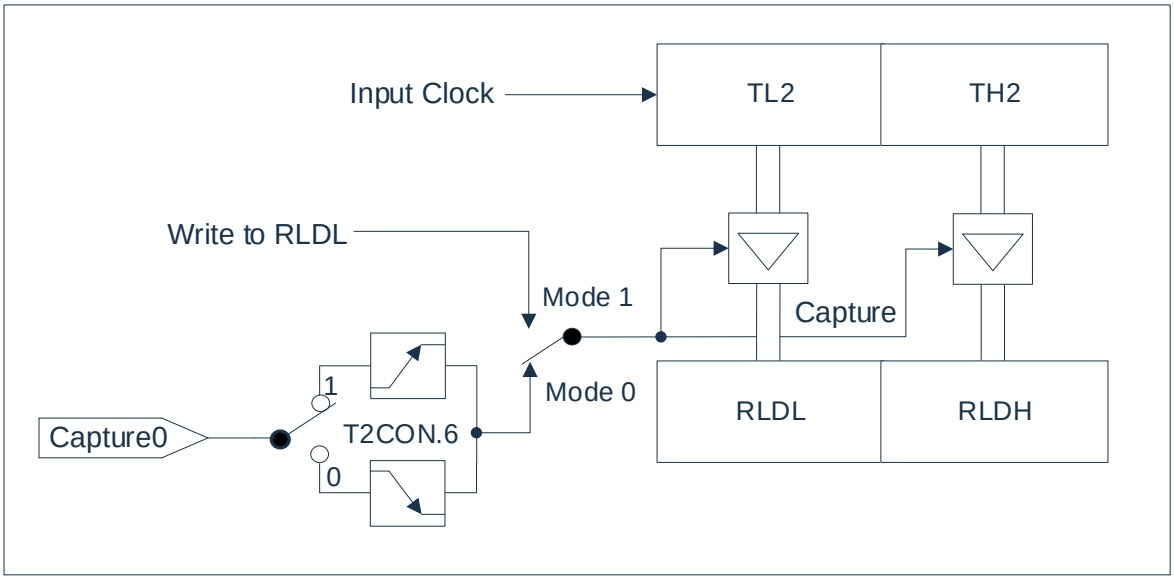


Figure 9-6: Timer 2 capture mode 0 for RLDL and RLDH registers block diagram

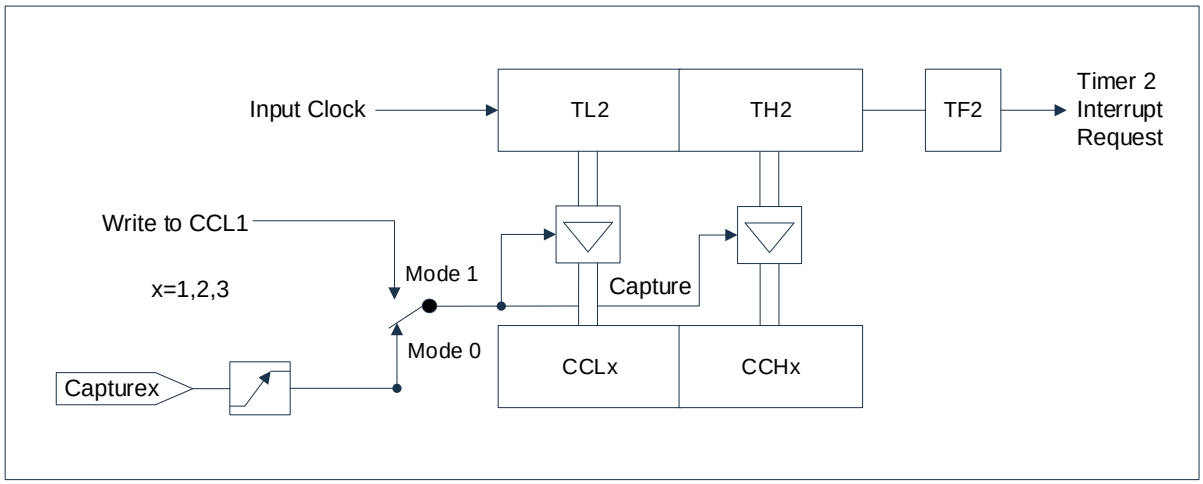


Figure 9-7:Timer 2 capture mode 0 for CCLx and CCHx registers block diagram

10. Timer 3/4

Timer 3/4 are similar to Timer 0/1 and they are two 16-bit timers. Timer 3 has four operating modes, and Timer 4 has three operating modes. Compared with Timer0/1, Timer3/4 only provides timing operations.

The register is incremented in every 12 clock periods or in every 4 clock periods, when timer is enabled.

10.1 Overview

Timer 3 and Timer 4 are composed of two 8-bit registers {TH3, TL3} and {TH4, TL4}, respectively. Timer 3 and 4 are operating in four same modes. The modes of Timer3 and Timer4 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.

10.2 Timer 3/4 Register

10.2.1 Timer 3/4 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: Timer 4 timer enable bit
1= Enable;
0= Disable.
Bit6 T4M: Timer 4 clock selection bit
1= Fsys/4;
0= Fsys/12.
Bit5~Bit4 T4M1, T4M0: Timer 4 mode selection bit
00= Mode 0,13-bit timer;
01= Mode 1,16-bit timer;
10= Mode 2,8-bit Auto-Reload timer;
11= Mode 3,Stop timer.
Bit3 TR3: Timer 3timer enable bit
1= Enable;
0= Disable.
Bit2 T3M: Timer 4 clock selection bit
1= Fsys/4;
0= Fsys/12.
Bit1~Bit0 T3M1, T3M0: Timer 3 mode selection bit
00= Mode 0,13-bit timer;
01= Mode 1, 16-bit timer;
10= Mode 2, 8-bit Auto-Reload timer;
11= Mode 3, two separate 8-bit timers.

10.2.2 Timer 3 Data Register Lower Bit TL3

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

Bit7~Bit0 TL37-TL30: Timer 3 Low 8-bit data register.

10.2.3 Timer 3 Data Register Higher Bit TH3

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

Bit7~Bit0 TH37-TH30: Timer 3 High 8-bit data register. (also as higher bit of the counter)

10.2.4 Timer 4 Data Register Lower Bit TL4

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

Bit7~Bit0 TL47-TL40: Timer 4 Low 8-bit data register (also as lower bit of the counter)

10.2.5 Timer 4 Data Register Higher Bit TH4

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

Bit7~Bit0 TH47-TH40: Timer 4 High 8-bit data register. (also as higher bit of the counter)

10.3 Timer 3/4 Interrupt

Timer 3/4 interrupts can be enabled or disabled by the IE register. The high/low priority can also be set by the IP register. The interrupt related bits are as follows:

Interrupt Mask Register EIE2

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

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

Bit6 I2CIE I2C Interrupt enable bit
1= Enable I^2C interrupt;
0= Disable I^2C interrupt.

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

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

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

Bit2 ---

Bit1 ET4: Time4 Interrupt enable bit
1= Enable Time4 interrupt;
0= Disable Time4 interrupt.

Bit0 ET3: Time3 Interrupt enable bit
1= Enable Time3 interrupt;
0= Disable Time3 interrupt.

Peripheral interrupt status register EIF2 (0xB2)

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 instruction, Read only;
1= SPI interrupt, (Automatically cleared after clearing the specific interrupt flag);
0= No SPI interrupt.
Bit6 I2CIF: I^2C Total interrupt instruction, Read only;
1= I^2Cinterrupt, (Automatically cleared after clearing the specific interrupt flag);
0= No I^2Cinterrupt.
Bit5 ---
Bit4 ADCIF: ADC interrupt flag;
1= ADC Conversion completed, need software clear;
0= ADC conversion is not complete.
Bit3 PWMIF: PWM Total interrupt instruction, Read only;
1= PWM interrupt, (Automatically cleared after clearing the specific interrupt flag);
0= No PWM interrupt.
Bit2 ---
Bit1 TF4: Timer4 counter overflow interrupt flag;
1= Timer4 enter the interrupt service routine, the hardware automatically clears;
0= Timer4 counter has no overflow.
Bit0 TF3: Timer3 counter overflow interrupt flag;
1= Timer3 enter the interrupt service routine, the hardware automatically clears;
0= Timer3 counter has no overflow.

10.4 Timer3 Operation Mode

10.4.1 T3 - Mode0 (13-bit Timing Mode)

In this mode, Timer3 is 13-bit register. When all the '1' of the counter become '0', the interrupt flag bit TF3 is set to '1'. The 13-bit register consists of the lower 5 bits of TL3 and TH3. The higher 3 bits of TL3 should be ignored.

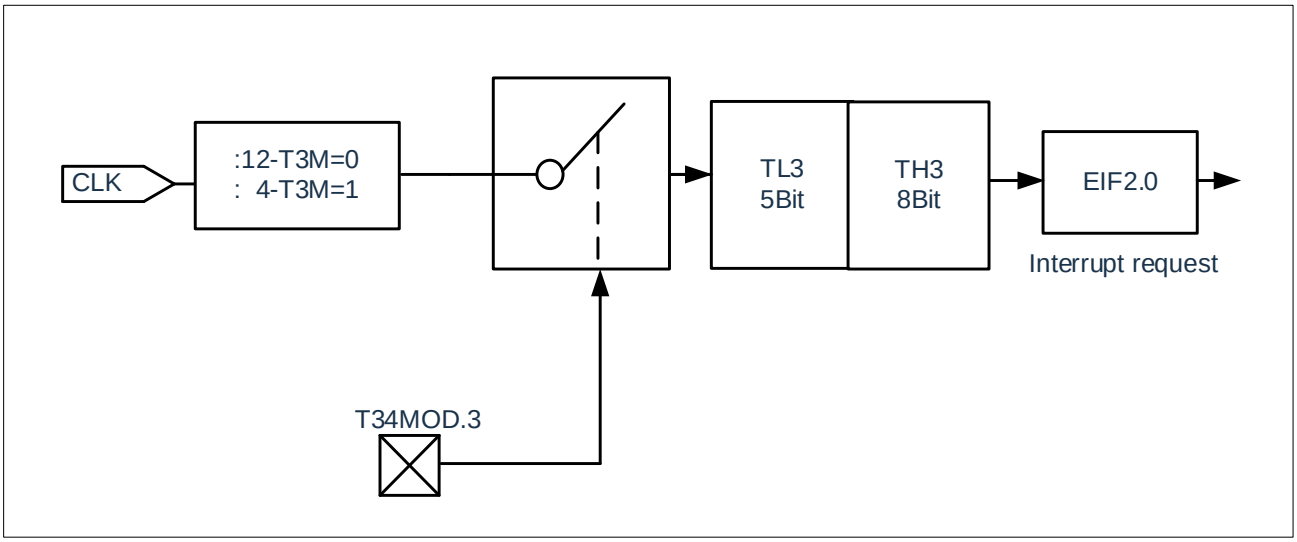


Figure 10-1: Timer3, Mode0: 13-bit Timer

10.4.2 T3 - Mode1 (16 –bit timing mode)

Mode 1 is the same as mode 0, Just in mode 1,all 16 bits of the Timer3 register work. Mode 1 is shown below

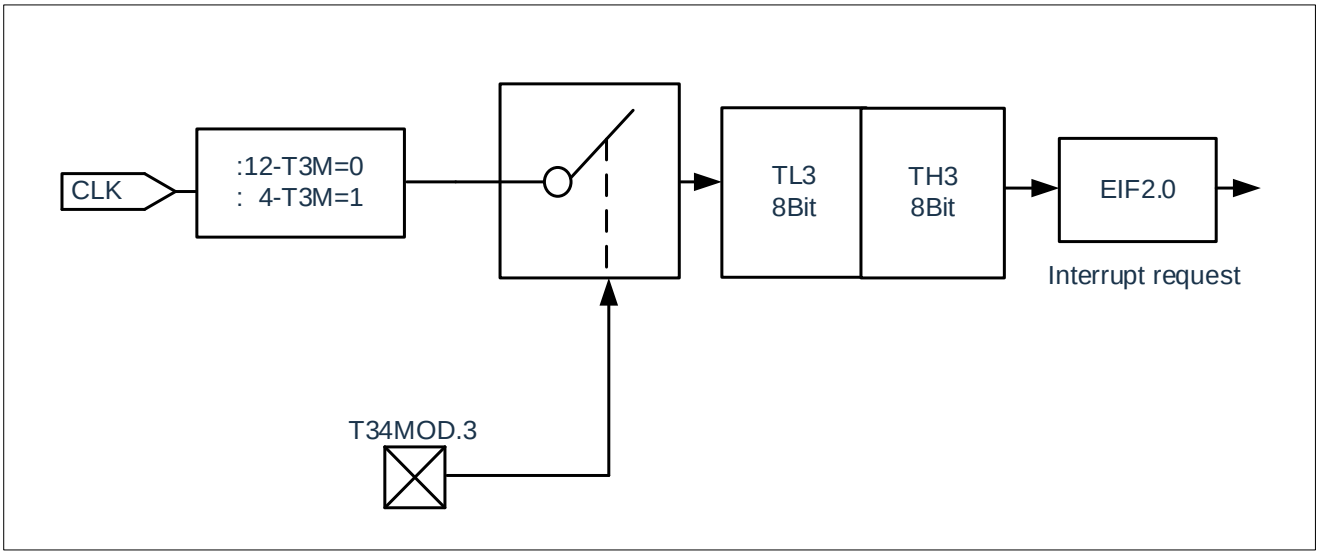


Figure 10-2: Timer3, Mode1: 16-bit Timer

10.4.3 T3 - Mode2 (8-bit automatic reload timing mode)

In mode 2, the Timer3 register is an 8-bit counter with automatic reload mode, as shown below. Overflow from TL3 register. Not only set TF3,but also reload TH3 to TL3,the value of TH3 remains unchanged during the reload process.

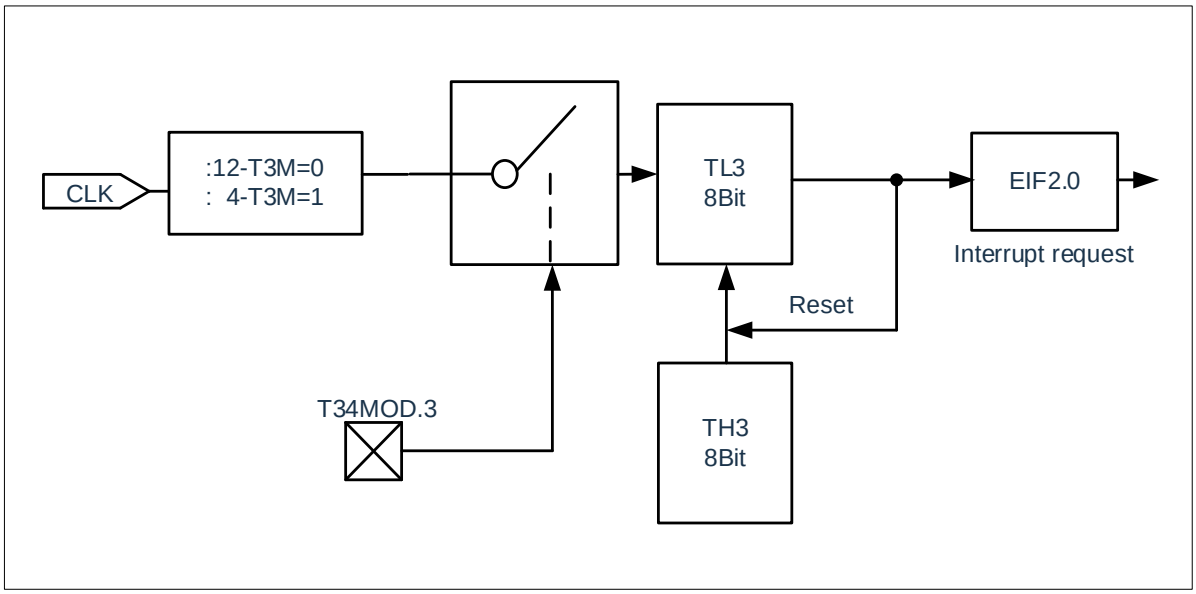


Figure 10-3: Timer3, Mode2: 8-bit Timer/counter (automatic reload)

10.4.4 T3 -Mode3 (Two separate 8-bit timers)

In mode 3, times 3 sets TL3 and TH3 to two independent counters. The logic of mode3 of timer 3 is as follows.

TL3 works as an 8-bit timer, and use the control bit of timer 0: such as TR3 and TF3.

TH3 works as an 8-bit timer, use TR4 and TF4 flags and control the timer 4 interrupt.

When you need to use two 8-bit timers, you can use mode 3.When timer3 is in mode 3, timer4 can switch to itself, Mode 3 is turned off, or can still be used as a baud rate generator by the serial channel, or in any application that does not require Timer 4 interrupts.

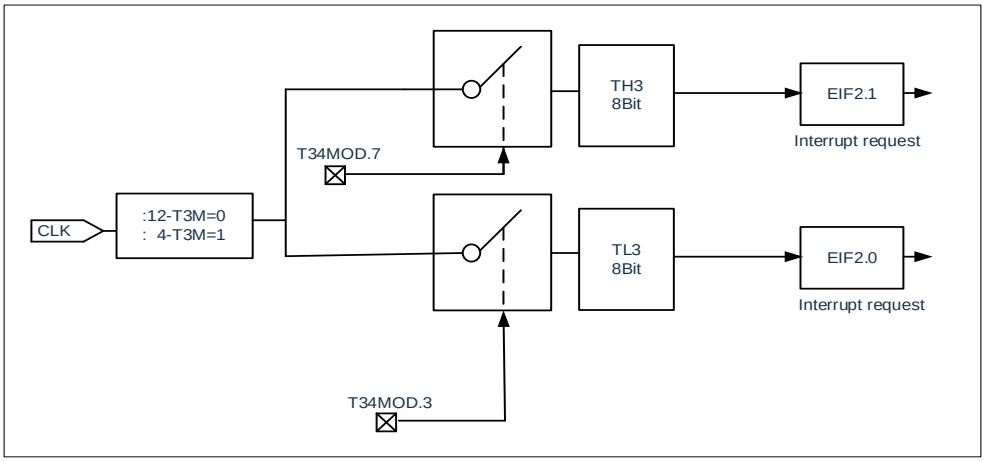


Figure 10-4: Timer3, Mode3: Two 8-bit timer/counter

10.5 Timer4 Operation Mode

10.5.1 T4 -Mode0 (13 –bit timing mode)

In this mode, timer 4 is a 13-bit register. When all the '1' of the counter become '0', the interrupt flag bit TF4 is set to '1'. The 13-bit register consists of the lower 5 bits of TL4 and TH4. The higher 3 bits of TL4 should be ignored.

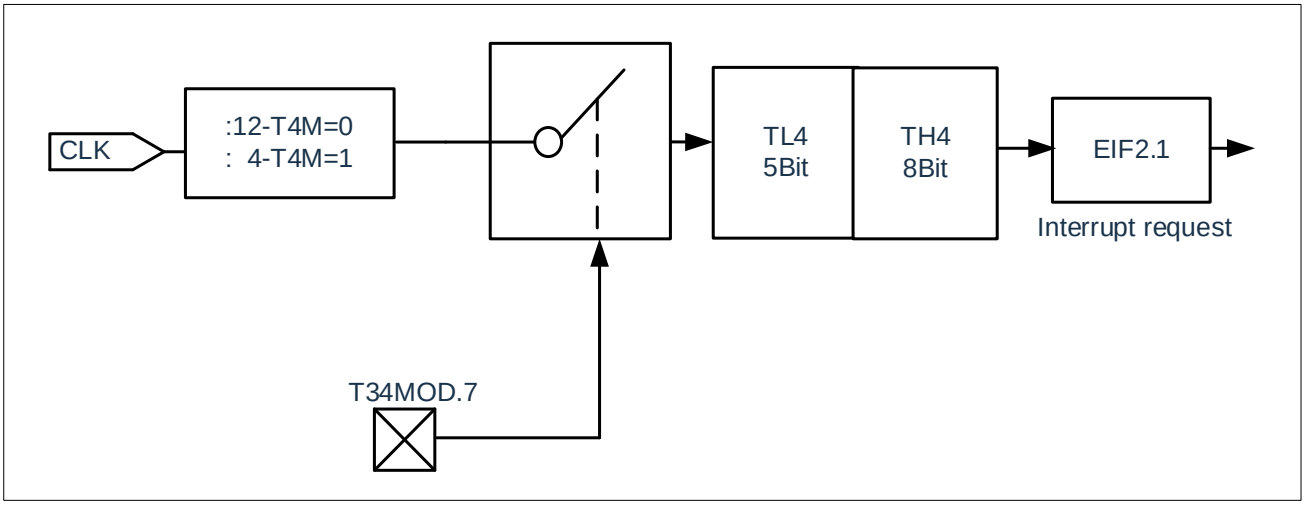


Figure 10-5: Timer4, Mode0: 13-bit timer/counter

10.5.2 T4 –Mode1 (16 –bit Timing Mode)

Mode 1 is the same as mode 0, Just in mode 1,all 16 bits of the Timer4 register work. Mode 1 is shown below.

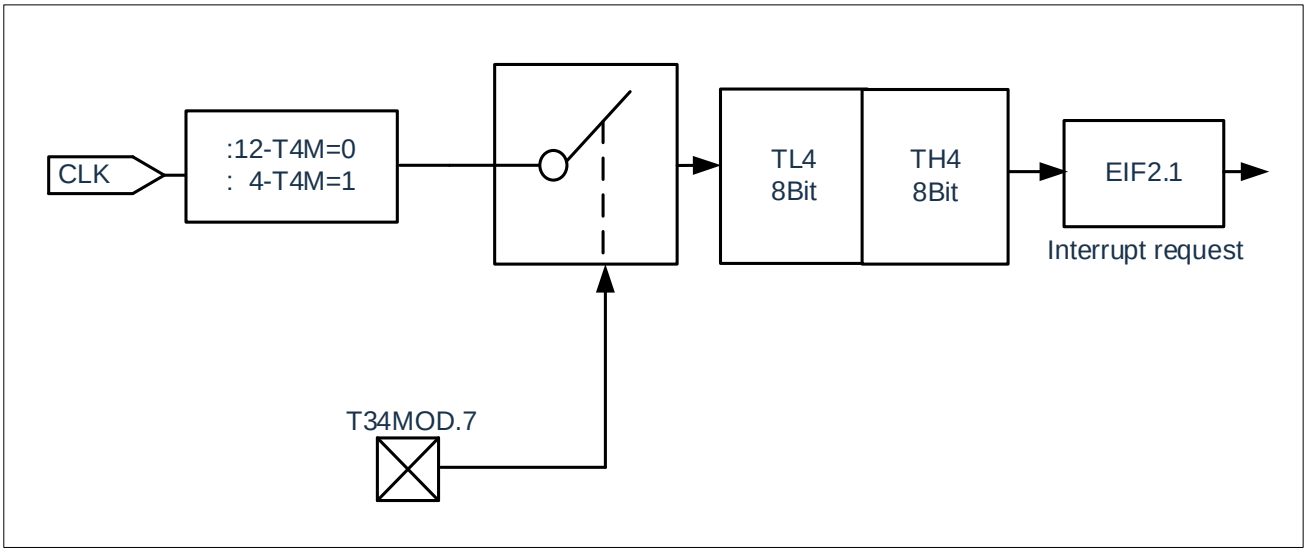


Figure 10-6: Timer4, Mode1: 16-bit timer/counter

10.5.3 T4- Mode2 (8-bit automatic reload timing mode)

In mode 2, the Timer4 register is an 8-bit counter with automatic reload mode, as shown below. Overflow from TL4 register. Not only set TF4,but also reload TH4 to TL4,the value of TH4 remains unchanged during the reload process.

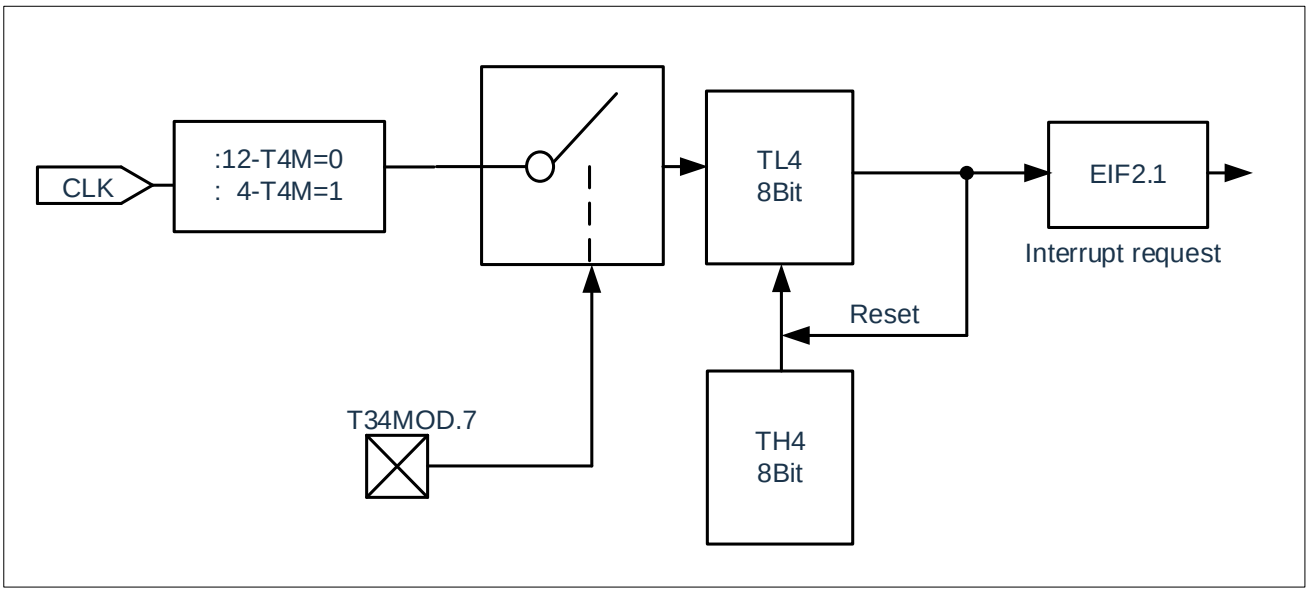


Figure 10-7: Timer4, Mode2: 8-bit timer/counter (automatic reload)

10.5.4 T4- Mode3 (stop counting)

Mode 3 stops counting, the effect is the same as setting TR4 = 0.

11. BUZZER

The buzzer consists of an 8-bit counter, a clock driver, and a control register. It produces a 50% duty cycle square wave, its frequency covers a wide range. The output frequency of BUZZER is controlled by the BUZCON register and the BUZDIV register.

When using the buzzer, you need to configure the relevant port as a buzzer port first:

P24CFG = 0x05; // P2.4 configured as a buzzer port

BUZZER control register BUZCON

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

Bit7 BUZEN: BUZZER enable bit;
1= Enable;
0= Disable.

Bit6~ Bit2 ---

Bit1~ Bit0 BUZCKS<1:0>: BUZZER frequency division ratio;
00= Fsys/8;
01= Fsys/16;
10= Fsys/32;
11= Fsys/64.

BUZZER frequency control register BUZDIV

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

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

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

Example 1: Fsys= 8MHz, BUZCKS<1:0>=01, BUZDIV=125
Fbuz=8MHz/(2*125)/16= 2KHz

Example 2: Fsys=16MHz, BUZCKS<1:0>=11, BUZDIV=125
Fbuz=16MHz/(2*125)/64= 1KHz

Example 3: Fsys=24MHz, BUZCKS<1:0>=11, BUZDIV=94
Fbuz=24MHz/(2*94)/64= 2KHz

Buzzer output frequency list

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

12. Analog To Digital Conversion (ADC)

12.1 ADC Overview

The ADC can convert the analog input signal into a 12-bit binary value which indicates the signal. The analog input signal at the port is connected to the input of the analog to digital converter after passing through the multiliexer.ADC adopts a successive approximation method to produce a 12-bit binary result, and save the result in the ADC result register (ADRESL and ADRESH).

ADC reference voltage is always generatedinternally.ADC can generate an interrupt after the conversion is complete.

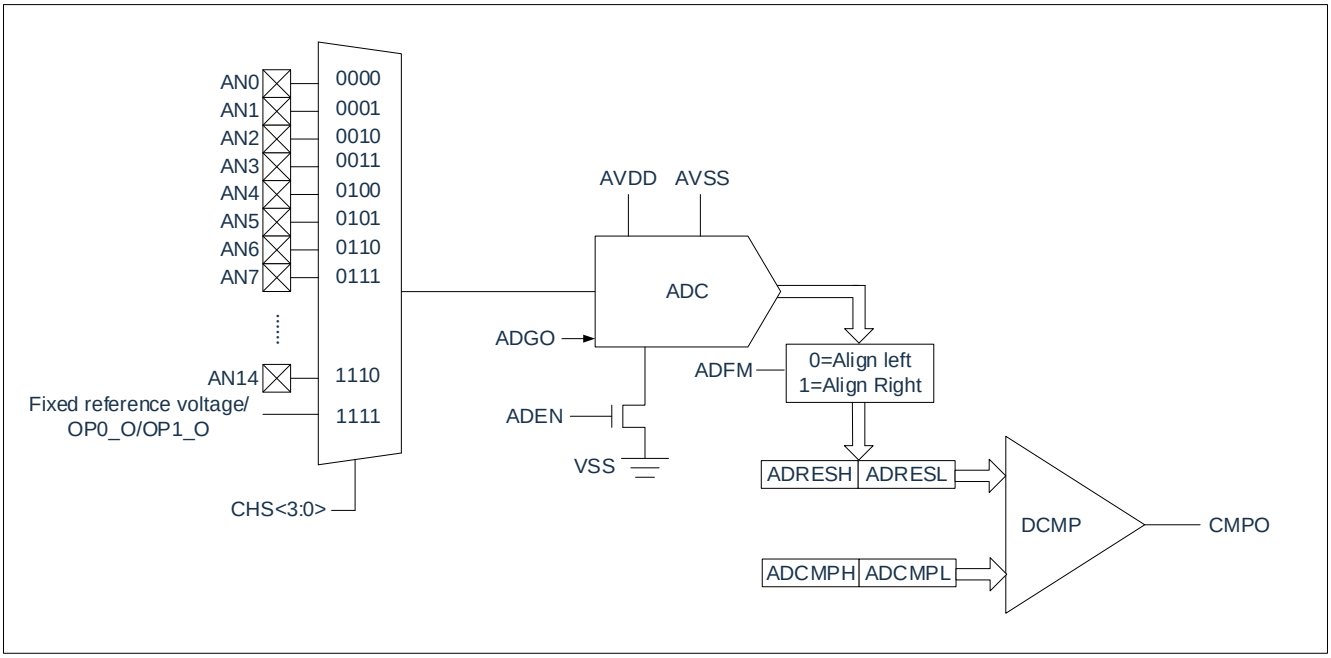


Figure 12-1: ADC Block diagram

12.2 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.

12.2.1 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. See the corresponding port chapter for more information.

Note: Applying an analog voltage to a pin defined as a digital input may cause an over current in the input buffer.

12.2.2 Channel Selection

Which channel is connected to the analog-to-digital converter is determined by the ADCHS bit of the ADCON1 register.

If the channel is changed, a certain delay is required before the next conversion starts. For more information, please refer to “ADC”.

12.2.3 ADC Reference Voltage

The reference voltage of the ADC is always provided by the VDD and GND of the chip.

12.2.4 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 a one-bit conversion is defined as T_{ADCK} . A complete 12-bit conversion takes 18.5 T_{ADCK} cycles (the time to complete a conversion ADGO lasts as high).

The corresponding T_{ADCK} specification must be met in order to obtain the correct conversion results.

Note: Any change in system clock frequency will change the frequency of the ADC clock, which will negatively affect the ADC conversion results.

12.2.5 ADC Interrupt

The ADC module allows an interrupt to be generated after the analog to digital conversion is completed. The ADC Interrupt Flag is the ADCIF bit in the EIF2 register. The ADC Interrupt Enable bit is the ADCIE bit in the EIE2 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.

12.2.6 Result Formatting

The result of a 12-bit A/D conversion can be in two formats: left or right. 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.

12.3 ADC Hardware Triggered Start

In addition to the software-triggered AD conversion, the ADC module also provides a way to trigger 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, even if the ADC function can be externally triggered. After a certain delay, the hardware trigger signal will automatically clear the ADGO bit after the conversion. 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.

12.3.1 External Port Edge Trigger ADC

The ADCETn pin edge automatically triggers an AD conversion. The ADCPX bit selects the input pin for the trigger (ADCET0, ADCET1). at this time ADTGS[1:0] needs to be 11 (select external port edge trigger), and ADEGS[1:0] can select which edge trigger.

12.3.2 PWM Triggered ADC

The PWM selects whether the ADC conversion is triggered by an edge or a period zero/midpoint. ADTGS[1:0] selects the PWM channel (PG0, PG2, PG4), and ADEGS[1:0] can select the edge type or period type trigger mode.

12.3.3 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+4) *Tsys

12.4 ADC Operating Principle

12.4.1 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').

12.4.2 Complete Conversion

When the conversion is complete, the ADC module will:

- Clear ADGO bit;
- Set ADCIF bit 1;
- Update the ADRESH:ADRESL register with the new result of the conversion.

12.4.3 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. Because Thus, the ADRESH:ADRESL register will retain the value obtained from the last conversion.

Note: A device Reset will force all registers to the Reset state. Therefore, reset will turn off the ADC module and terminate any pending conversions.

12.4.4 A/D Conversion Step

The ADC use age guide is given as follows step by step:

1. Pin configuration:
   - Use a pin as a driver is forbidden (PxTRIS register);
   - Configure pins as analog input pin.
2. Configure ADC block:
   - Choose ADC conversion clock;
   - Choose ADC input channel;
   - Choose the format of results;
   - Initiate ADC block
3. Configure ADC interrupt (optional):
   - Reset ADC interrupt flag bit;
   - Enable ADC interrupt;
   - Enable peripheral interrupt;
   - Enable global interrupt.
4. Waiting for the 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 a customer is trying to renew the device's sequence code execution after it's awakened from sleep mode, then global interrupt must be forbidden.

12.4.5 Conversion During Sleep Mode

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'll convert again when the system is awakened. ADGO will remain as one until the conversion is completed.

12.5 ADC Related Register

The following nine registers are related to AD conversion:

- AD control registers: ADCON0, ADCON1, ADCON2;
- Comparator control register: ADCMPC;
- Delay data register: ADDLYL;
- AD results data registers: ADRESH/L;
- Comparator data register: ADCMPH/L.

AD control register: ADCON0

0xDF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON0	ADCHS4	ADFM	AN15SEL1	AN15SEL0	---	---	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	ADCHS<4>:	ADC analog channel selection bit <4>, as described in ADCON0.ADCHS;
	1=	ADCHS<3:0> to select channels AN16-AN31 (non-existent channels can be ignored);
	0=	ADCHS<3:0> selects channels AN0-AN15 (non-existent channels can be ignored).
Bit6	ADFM:	ADC conversion result format selection bit;
	1=	Right alignment;
	0=	Left alignment.
Bit5~Bit4	AN15SEL<1:0>:	ADC channel 15 input source selection bit;
	00=	Fixed reference voltage (1.2V);
	01=	--;
	10=	--;
	11=	--;
Bit3- Bit2	Unused.	
Bit1	ADGO:	ADC conversion start bit (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	---	

AD control register ADCON1

0xDE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON1	ADEN	ADCKS2	ADCKS1	ADCKS0	ADCHS3	ADCHS2	ADCHS1	ADCHS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	1	0	0	0	0	0	0

Bit7 ADEN: ADC enable bit;
1= Enable ADC;
0= Disable ADC, no operating current is consumed.

Bit6~Bit4 ADCKS<2:0>: ADC conversion clock select bit.

000=	Fsys/2	100=	Fsys/32
001=	Fsys/4	101=	Fsys/64
010=	Fsys/8	110=	Fsys/128
011=	Fsys/16	111=	Fsys/256

Bit3~Bit0 ADCHS<3:0>: The analog channel select bits are 4 bits lower and form a 5-bit channel select bit with ADCHS<4>, ADCHS<4:0>.

00000=	--	10000=	AN16
00001=	--	10001=	AN17
00010=	--	10010=	AN18
00011=	--	10011=	AN19
00100=	AN4	10100=	AN20
00101=	AN5	10101=	AN21
00110=	AN6	10110=	AN22
00111=	AN7	Other =	--
01000=	AN8		
01001=	AN9		
01010=	AN10		
01011=	AN11		
01100=	AN12		
01101=	AN13		
01110=	AN14		
01111=	See the description of ADCON0.AN15SEL.		

AD control register ADCON2

0xE9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON2	ADCEX	ADCPX	ADTGS1	ADTGS0	ADEGS1	ADEGS0	---	---
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit7	ADCEX:	ADC hardware trigger enable bit.
	1=	Enable;
	0=	Disable.
Bit6	ADCPX:	Pin select bit of ADC external trigger port.
	1=	ADCET1
	0=	ADCET0
Bit5~Bit4	ADTGS:	ADC hardware trigger source select bit.
	00=	PG0(PWM0)
	01=	PG2 (PWM2)
	10=	PG4 (PWM4)
	11=	Port pin(ADCET0/ADCET1)
Bit3~ Bit2	ADEGS:	ADC hardware trigger edge select bit;
	00=	Falling edge;
	01=	Rising edge;
	10=	Midpoint of PWM period;
	11=	End point of PWM period.
Bit1~Bit0	---	

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 bit;
	1=	Enable;
	0=	Disable.
Bit6	ADCMPPS:	The output polarity of ADC comparator select bit;
	1=	If ADRES<ADCMP, then ADCMPO=1;
	0=	If ADRES>=ADCMP, then ADCMPO=1.
Bit5	---	
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	---	
Bit1~ Bit0	ADDLY[9:8]:	ADC hardware trigger delay data[9:8]bits.

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 lower 8 bits.

AD data register higher 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 bits conversion results.

AD data register lower 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 bits.
Number 3 to 0 bit of 12 bits conversion results.
Bit3~Bit0 Unused.

AD data register higher bits: 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 bits.
Number 11 to 8 bit of 12 bits conversion results.

AD data register lower bits: ADRESL, 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 bits.
Number 7 to 0 bit of 12 bits conversion results.

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>: Higher 8 bits of ADC comparator's data.

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.

Bit 3~Bit0 ADCMP[3:0]: Lower 4 bits of ADC comparator's data.

12.6 ADC Results 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 updated automatically after the conversion. ADCCMPPS 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 PWM function is enabled, when 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.

13. Enhanced PWM Generator

13.1 Overview

The PWM0 supports six PWM generators which can be configured as six independent PWM outputs, (PG0-PG5), or as three complementary PWM pairs (PG0-PG1,PG2-PG3,PG4-PG5)with three programmable dead-time generators.

Every complementary PWM pairs share one 8-bit prescaler, There are six clock dividers providing five divided frequencies (1, 1/2, 1/4, 1/8, 1/16) for each channel. Each PWM output has independent 16-bit counter for PWM period control, and 16-bit comparators for PWM duty control. The six PWM generators provide twenty-four independent PWM interrupt flags, corresponding period or duty cycle of PWM channel is matched with counter, interrupt flag will be generated, each PWM posses its own independent enable bit.

Each PWM generator can be configured as One-shot mode to produce only one PWM cycle signal or loop mode to output PWM waveform continuously.

13.2 Features

Enhanced PWM supports the following features:

- Six independent 16-bit PWM control mode.
  - Six independent outputs - PG0,PG1,PG2,PG3,PG4,PG5;
  - Three complementary PWM pairs: (PG0-PG1), (PG2-PG3), (PG4-PG5),capable of programmable dead-time insertion;
  - Three synchronous PWM pairs: (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 PG5are synchronized
- One-shot(only support edge-aligned type) or Auto-reload mode PWM
- Support edge-aligned mode and center-aligned mode
- Support symmetrical and asymmetrical counting in center-aligned mode
- Support programmable dead-time generator between complementary paired PWMs
- Each PWM generator has independent polarity setting control
- Hardware fault brake protections (external FB1 trigger, support software trigger).
- ADC comparing event triggers hardware fault brake protection
- PWM edge or period triggers to start AD conversion

13.3 Port Configuration

It is necessary to configure corresponding ports as PWM channel before using enhanced PWM block, PWM channel is marked withPG0-PG5 on the pin distribution diagram, corresponding to PWM channel 0-5,respectively.It could be found out that different PWM channel could correspond to the same port, and the same PWM channel could be distribute to different port, this feature makes enhanced PWM function adapt to different type of package and requirement for flexible layout.

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

```
P13CFG=0x06;//Configure P13 as PG0 channel
P14CFG=0x06;//Configure P14 as PG1 channel
P15CFG=0x06;//Configure P15 as PG2 channel
P16CFG=0x06;//Configure P16 as PG3 channel
P17CFG=0x06;//Configure P17 as PG4 channel
P22CFG=0x06;//Configure P22 as PG5 channel
```

13.4 Function description

It supports two kinds of operating mode: edge-aligned, center-aligned

13.4.1 Edge-aligned

In Edge-aligned PWM Output type, the 16-bit PWM counter CNTn will start counting-down from the start of every period, compare with the latched value of CMPn, PGn will output high level voltage when CNTn=CMPn, CMPnDIF is set to 1.CNTn continues counting-down to zero, PGn will output low voltage at this moment, also CMPn and PERIODn will be updated in the condition of PWMnCNTM=1,PIFand set PIF period interrupt flag.

Edge-aligned corresponding parameters:

$$High\ level\ time = (CMPn+1) \times Tpwm \quad (CMPn \geqslant 1)$$

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

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

The duty cycle is 0% when CMPn=0.

The edge alignment timing is shown in the following figure:

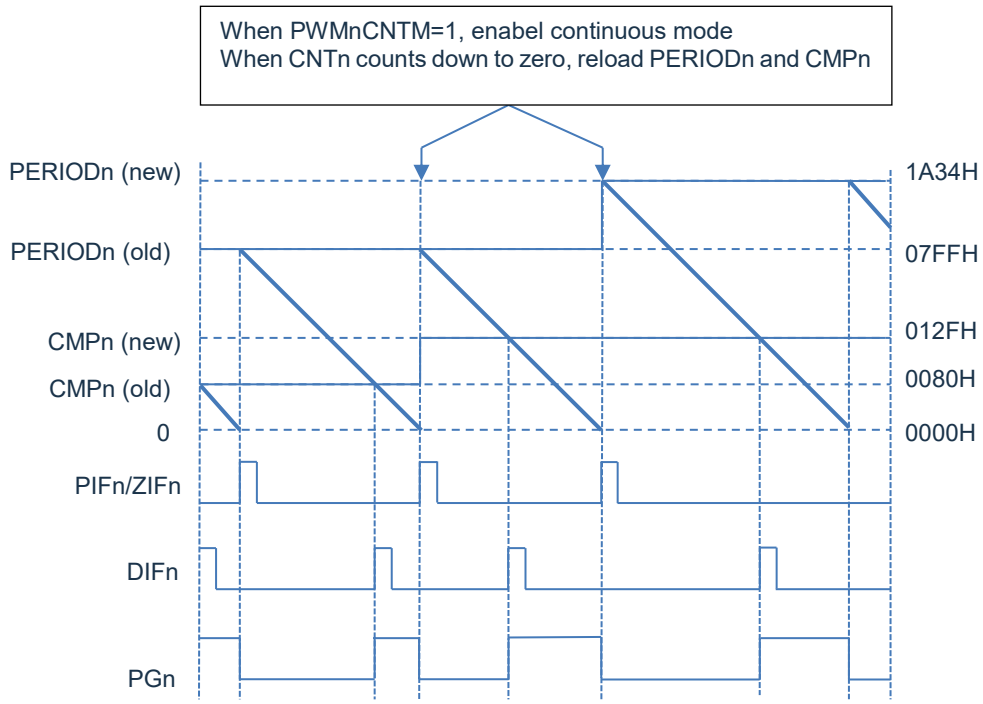


Figure 13-1: edge-aligned timing diagram

13.4.2 Center-aligned

In the center-aligned symmetric counting mode, the 16-bit PWM counter CNTn starts counting up from 0. When CNTn=CMPn, PGn outputs high, after which CNTn continues to count up until it is equal to PERIODn, and then CNTn starts counting down, and in the process of counting down when CNTn=CMPn, PGn outputs low, after which it continues to count down to 0.

The parameters related to the center-aligned symmetric counting method are as follows:

$$high\ level\ time = (PERIODn \times 2 - CMPn \times 2 - 1) \times Tpwm\ ;\ (CMPn \geqslant 1)$$

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

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

100% duty cycle at CMPn=0;

The center-aligned timing (symmetric counting) is shown in the following figure:

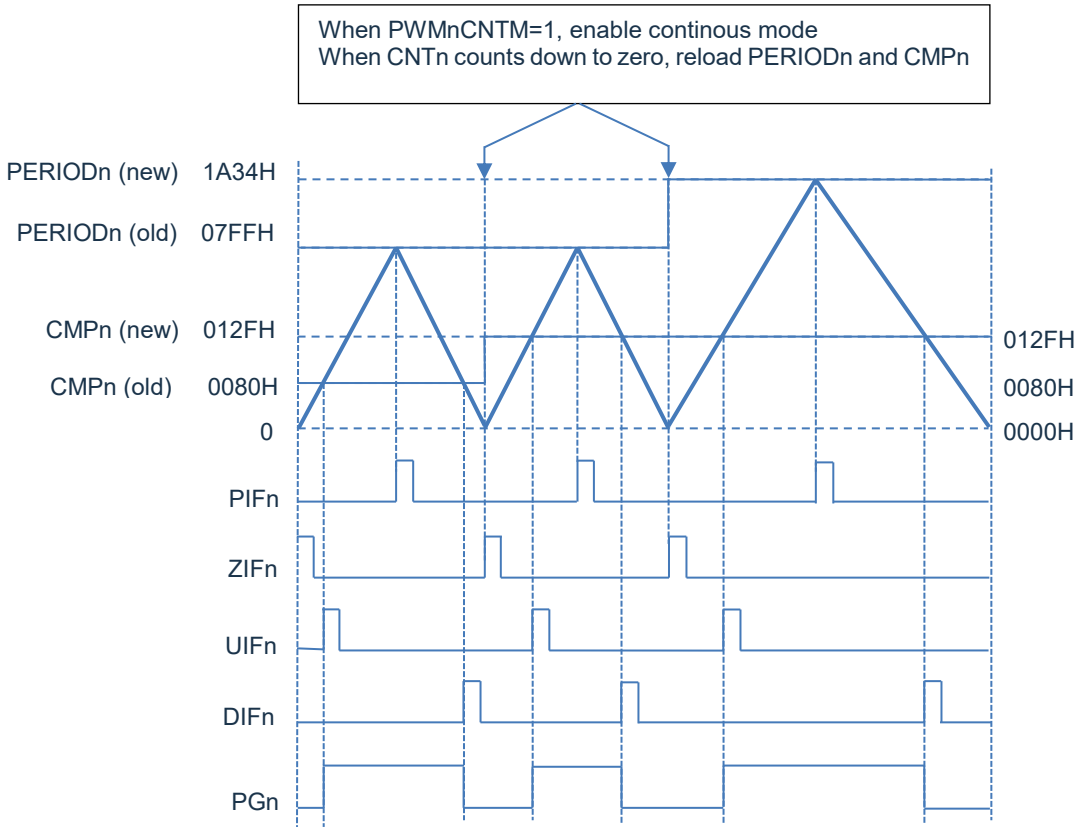


Figure 13-2: Center-aligned timing diagram (symmetrical counting)

Figure 13-3 shows the waveform of center-aligned counter:

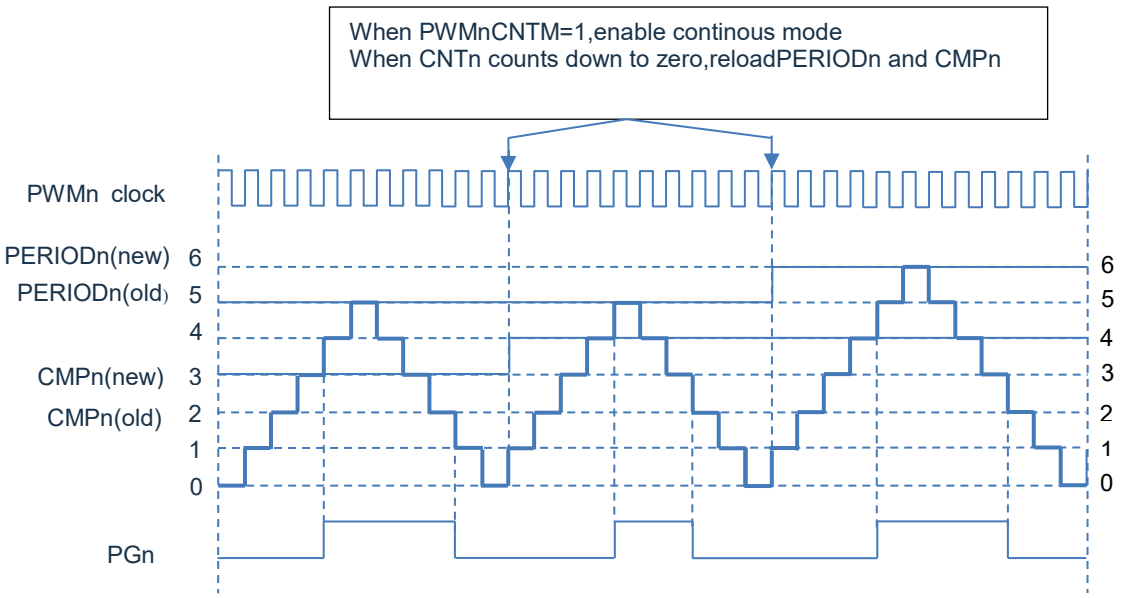


Figure 13-3: Center-aligned counter waveform (symmetrical counting)

In the center-aligned asymmetric counting mode, the 16-bit PWM counter CNTn starts counting upward from 0. When CNTn=CMPn, PGn outputs high, after which CNTn continues to count upward until it is equal to PERIODn, and then CNTn starts counting downward. To turn on the asymmetric counting mode, set ASYMEN to 1. The asymmetric counting mode can achieve accurate center-aligned waveforms.

The parameters related to the center-aligned asymmetric counting method are as follows:

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

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

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

100% duty cycle when CMPn=0 versus CMPDn=0;

The center-aligned timing (asymmetric counting) is shown in the following figure:

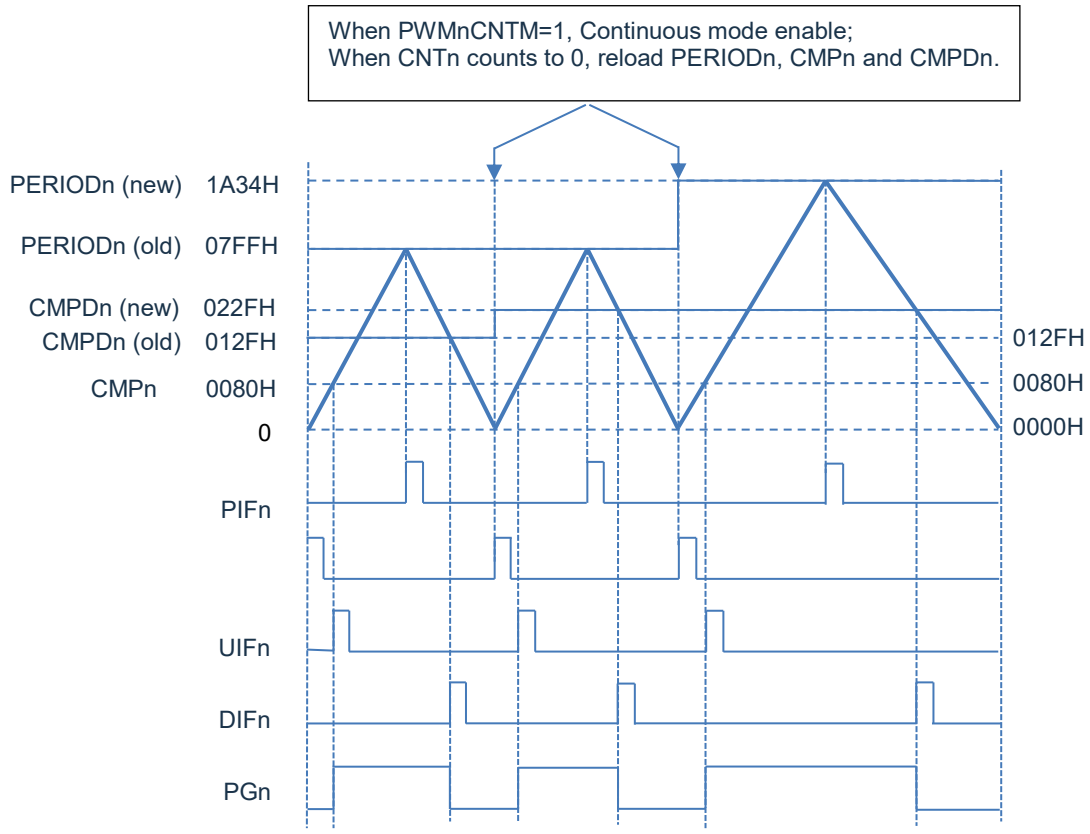


Figure 13-4: Centre alignment timing diagram(Asymmetric counting)

Centre alignment timeline diagram (Asymmetric count) is shown in the figure below:

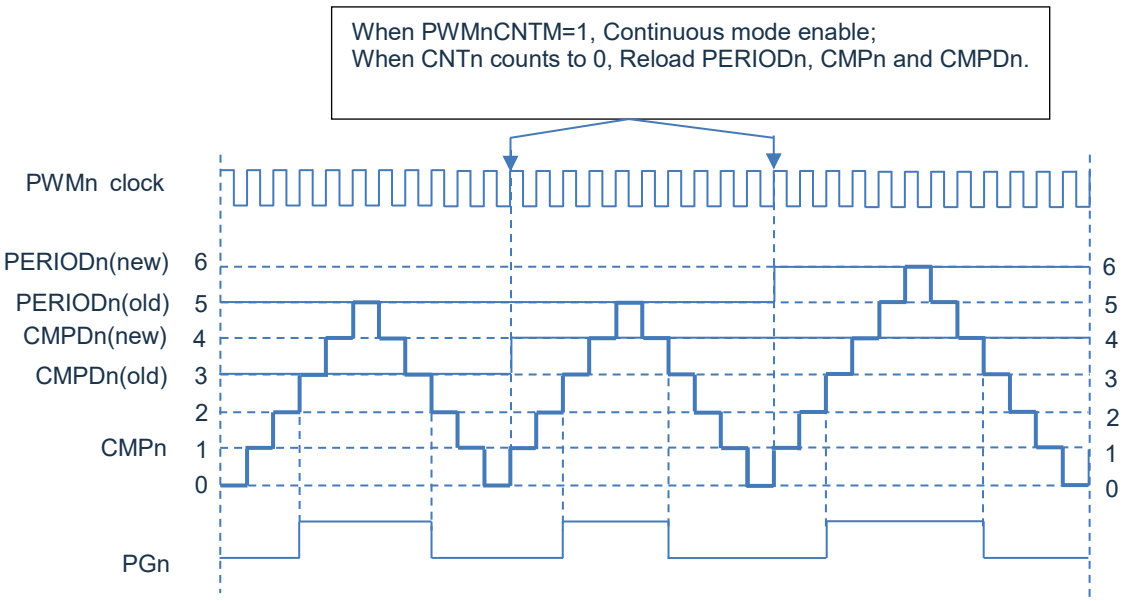


Figure 13-5: Centre alignment timing diagram (Asymmetric count)

13.4.3 Complementary Mode And Dead Zone Delay

6 channel PWM can be set to 3 sets of complementary pairs. In the complementary mode, the period and duty cycle of PWM1, PWM3 and PWM5 are determined by the relevant registers of PWM0, PWM2 and PWM4 respectively. Meanwhile, the dead zone delay register can also affect the duty cycle of PWM complementary pair. At this point, in addition to the corresponding output enabling bit (PWMnOE), PWM1/PWM3/PWM5 output waveform is no longer controlled by its 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 of PWM0/PWM2/PWM4, respectively.

Both center alignment and edge alignment support complementary patterns.

13.4.4 Brake Function

There are several sources that trigger PWM brakes:

1. External trigger port FB1;
2. ADC result compare output;

After the brake is triggered, the brake flag is set to 1, the counter enable bit of all channels is cleared, and the PWM outputs the preset brake data.

To resume normal output, you need to clear the brake flag and re-enable the PWM channel counter.

13.5 PWM Related Register

PWM Control Register PWMCON

F120H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMCON	---	---	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~ Bit6 ---

Bit5~Bit4 PWMMODE: Control bit of PWM's mode;
00= Independent mode;
01= Complementary mode;
10= Synchronize mode;
11= Reserved.

Bit3 GROUPEN: PWM group function enable bit;
1= PG0controlPG2,PG4;PG1 controlPG3,PG5;
0= All PWM Channel signal are independent of each other.

Bit2 ASYMEN: Asymmetric count enable bit in PWM center alignment;
1= Asymmetric counting is enabled;
0= Symmetric counting is enabled.

Bit1 CNTTYPE: PWM count alignment select bit;
1= Center alignment;
0= Edge alignment.

Bit0 ---

PWM Output Enable Control Register PWMOE

F121H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWNOE	---	---	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 ---

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

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

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

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

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

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

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 bit of PWM channel 0/1;
00= Prescaler clock stop, PWM0/1counter stop;
others= System clock (PWM01PSC+1) frequency divide.

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 bit of PWM channel 2/3;
00= Prescaler clock stop, PWM2/3 counter stop;
others= System clock (PWM23PSC+1)frequency divide.

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 bit of PWM channel 4/5;
00= Prescaler clock stop, PWM4/5counter stop;
others= System clock (PWM45PSC+1) frequency divide.

PWM clock frequency 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 ---
Bit2~Bit0 PWMnDIV<2:0>: Clock frequency divide control bit of PWM channel n;
000= Fpwmn-PSC/2;
001= Fpwmn-PSC/4;
010= Fpwmn-PSC/8;
011= Fpwmn-PSC/16;
100= Fpwmn-PSC;
others= Fsys (system clock);
(PSC is the prescalered clock)

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 ---
Bit5~ Bit0 PWMnLE: Data load enable control register of PWM channel n(n=0-5)
(Hardware clear 0 when loading is finished);
1= Enable load period and duty period data (PERIODn, CMPn, CMPDn).
0= Writing 0 is invalid.

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 ---

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

1= Inverse output;

0= Normal output.

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 ---

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

1= Automatic loading mode;

0= Reserved (access disabled)

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 ---

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

1= PWMn counter on (PWMn starts output);

0= PWMn counter stops (software writes 0 to stop the counter and clear the counter value).

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 ---

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

1= PWMn counter clear 0;

0= Writing 0 is invalid.

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.

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

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMPnH	PWMPnH7	PWMPnH6	PWMPnH5	PWMPnH	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.

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 period data) low 8-bit register.

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 period data) high 8-bit register.

PWM Down Compare Data Low 8-bit Register 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 down compare data low 8-bit register (duty period data in asymmetric counting).

PWM Down Compare Data High 8-bit Register 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 down compare data high 8-bit registers (duty period data in asymmetric counting).

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 ---

Bit2 PWM45DTE: Dead zone enable bit of PWM4/5 channel;
1= Enable;
0= Disable.

Bit1 PWM23DTE: Dead zone enable bit of PWM2/3 channel;
1= Enable;
0= Disable.

Bit0 PWM01DTE: Dead zone enable bit of PWM0/1 channel;
1= Enable;
0= Disable.

PWM 0/1 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 is dead zone delay data register.

PWM 2/3 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 is dead zone delay data register.

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 is dead zone delay data register.

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 ---

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

1= PWMn channel enable mask data output;

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

PWM Mask Control 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 ---

Bit5~Bit0 PWMnMASKD: PWM channel n mask data bits (n=0-5);

1= PWMn channel output high;

0= PWMn channel output is low.

PWM Brake Control Register PWMFBKC

F166H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMFBKC	PWMFBIE	PWMFBF	PWM5FBCCE	PWMFBKSW	PWMFB1ES	----	PWMFB1EN	----
R/W	R/W	R/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 bit;
1= Enable interrupt;
0= Disable interrupt.
Bit6 PWMFBF: PWM's Brake mark (writing 0 is clear);
1= Enable (PWM output value of brake data register);
0= Disable.
Bit5 PWMFBCCE: Whether to clear all channel counter selection bits when PWM brakes;
1= Enable;
0= Disable.
Bit4 PWMFBKSW: PWM's Software brake signal start bit;
1= Enable;
0= Disable.
Bit3 PWMFB1ES: PWM's external hardware brake channel 1 (FB1) trigger level select bit;
1= High level trigger;
0= Low level trigger
Bit2 Unused.
Bit1 PWMFB1EN: PWM's external hardware brake channel 1 (FB1) enable bit;
1= Enable;
0= Disable.
Bit0 Unused.

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 Unused.
Bit5~Bit0 PWMnFBKD: PWM channel n brake data bit (n=0-5);
1= PWMn channel's output is high after brake operation.
0= PWMn channel's output is low after brake operation.

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 ---
Bit5~Bit0 PWMnPIE: PWM channel n period interrupt mask bit (n=0-5)
1= Enable interrupt;
0= Disable interrupt.

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 ---

Bit5~Bit0 PWMnZIE: PWM channel n zero interrupt mask bit (n=0-5);

1= Enable interrupt;

0= Disable interrupt.

PWM Up Compare Interrupt Mask Register PWMUIE

F16BH	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 ---

Bit5~Bit0 PWMnUIE: PWM channel n up compare interrupt mask bit (n=0-5);

1= Enable interrupt;

0= Disable interrupt.

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 ---

Bit5~Bit0 PWMnDIE: PWM channel n down compare interrupt mask bit (n=0-5);

1= Enable interrupt;

0= Disable interrupt.

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 ---

Bit5~Bit0 PWMnPIF: PWM channel n period interrupt flag bit (n=0-5);

1= Generate an interrupt (software clear);

0= No interrupt.

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 ---

Bit5~Bit0 PWMnZIF: PWM channel n zero interrupt flag bit (n=0-5);

1= Generate an interrupt (software clear);

0= No interrupt.

PWM Up Compare Interrupt Flag Register PWMUIF

F16FH	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 ---

Bit5~Bit0 PWMnUIF: PWM channel n up compare interrupt flag bit (n=0-5);

1= Generate an interrupt (software clear);

0= No interrupt.

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 ---

Bit5~Bit0 PWMnDIF: PWM channel n down compare interrupt flag bit (n=0-5);

1= Generate an interrupt (software clear);

0= No interrupt.

Note: See the chapter <Interrupt > for details of PWM interrupt flag bit clearing operation cautions.

14. I²C MODULE

14.1 Overview

This model provides interface between microprocessor and I²C bus. The I²C supports arbitration and clock synchronization so that it can run in multi-master system. The I²C supports normal mode and fast mode.

The main features of the model include:

- 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 (high to 400Kbs);
- Supports arbitration and clock synchronization.
- Supports multi-master system.
- Master mode supports 7-bit addressing mode and 10-bit addressing mode on I²C bus.
- Slave mode s supports 7-bit addressing mode on I²C bus.
- Generate interrupt.
- Operate at a wide range of clock frequency(8-bit internal timer).

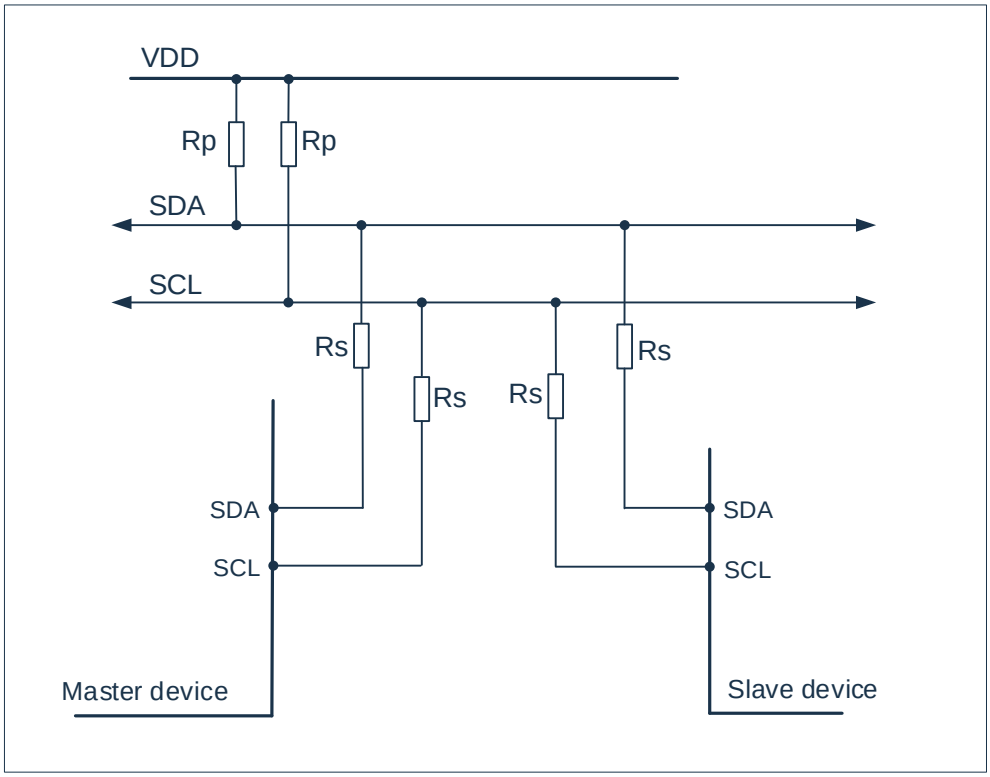


Figure 14-1: I²C Interconnection

14.2 I²C Port Configuration

The chip has a I²C model, and the I²C have multiple channel with general I/O. The I²C channel have two port types: { SCL,SDA} and {[SCL],[SDA]}.

The 4th bit of Function control register deploy I²C channel:

FUNCCR.4=0; Select { SCL,SDA} as the channel of I²C;

FUNCCR.4=1; Select {[SCL],[SDA]} as the channel of I²C;

If you use I²C, you should configure ports of SCL and SDA as the dedicated port of I²C. For instance, if you configure {SCL,SDA} port as the port of I²C:

```
P04CFG=0x03;      // configured as SCL port
P05CFG=0x03;    //configured as SDA port
```

After configuring the I²C channel, the group of ports is open-drain by default. You can configure whether to enable the internal pull-up resistors of SCL and SDA ports through PxUP, or add pull-up resistors outside the chip.

In master mode, the IIC outputs SCL to the slave. After sending address or data, the slave needs to pull down SCL and send back the corresponding answer signal to the host. The host needs to read back the SCL port line status to detect whether the slave releases SCL to determine whether the next data frame needs to be sent. If the pull-up resistor or board-level parasitic capacitance of SCL is larger, it will lead to longer read-back time, which will affect the communication speed of IIC, please refer to the IIC application manual.

14.3 I²C Master Mode

The six register that is connected with host machine are as follows : control, status, address of slave machine, 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	Status register in master mode I2CMSR	0xF5
Transmitting 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, transmitting data register and receiving data register use the same register address,

Write operation access transmitting data register(I2CMBUF), Read operation access receiving data register(I2CMBUF).

Write operation is executed by control register. Read operation is executed by status register.

14.3.1 I²C Timing Period Register In Master Control Mode

To generate a wide range of SCL frequencies, the module has a built-in 8-bit timer. For standard and fast transmission.

Ideal clock period of SCL when TIMER_PRD ≠ 0: 2* (1+TIMER_PRD)* 10* Tsys

Ideal clock period of SCL when TIMER_PRD = 0: 3* 10* Tsys

Refer to the IIC application manual for the specific equations of SCL calculation.

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	Reserve:	Must be zero.
Bit6~ Bit0	MTP6-MTP0:	Bits 6-0 of the period timing register for standard and fast modes: TIMER_PRD[6:0].

14.3.2 I²C Master Mode Control and Status Register

The control register consists of 4 bits: RUN, START, STOP, ACK bits. The START bit will generate a START or REPEATED START condition. The STOP bit determines whether the data transfer will stop at the end of the cycle or continue it. To generate a single transmit cycle, the slave address register is written with the desired address, the R/S bit is set to 0 and the control register is written with ACK=x, STOP=1, START=1, RUN=1 (I2CMCR=xxx0_x111x) to perform the operation and stop. When the operation is completed (or an error occurs), an interrupt is generated. Data can be read from the receive data register.

When the I²C is operating in master mode, the ACK bit must be set to 1. This will cause the I²C bus controller to automatically send an answer after each byte. This bit must be cleared to 0 when the I²C bus controller no longer needs the slave to send data.

Master mode control register

0xF5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CMCR	RSTS	---	---	---	ACK	STOP	START	RUN
R/W	W	R	R	W	W	W	W	W
Reset Value	0	0	0	0	0	0	0	0

Bit7 RSTS: I² C active module reset control bit;
1= Reset the master module (I² C registers of the entire master module, including I2CMSR);
0= I² C Interrupt flag bit cleared to 0 in master control mode.
Bit6~ Bit5 ---
Bit4 Reserve: Must to be 0.
Bit3 ACK: Response enable bit;
1= Enable;
0= Disable.
Bit2 STOP: Stop enable bit;
1= Enable;
0= Disable.
Bit1 START: Start enable bit;
1= Enable;
0= Disable.
Bit0 RUN: Run enable bit;
1= Enable;
0= Disable.

The combination of the following control bit list can realize all kinds of operations in master control module:

START: transmit start signal.

SEND: transmit data or address.

RECEIVE: receive data.

STOP: transmit end signal.

Combination of control bit (IDLE status)

R/S	ACK	STOP	START	RUN	OPERATION
0	-	0	1	1	START followed by SEND (master remains in transmit mode)
0	-	1	1	1	START followed by SEND and STOP
1	0	0	1	1	Non-response is used for reception after START (master remains in receiver mode)
1	0	1	1	1	START followed by RECEIVE and STOP
1	1	0	1	1	START followed by RECEIVE (master remains in receiver mode)
1	1	1	1	1	Disable combinations
0	0	0	0	1	Disable combinations

Combination of control bit (master send status)

R/S	ACK	STOP	START	RUN	OPERATION
-	-	0	0	1	SEND operation
-	-	1	0	0	Stop
-	-	1	0	1	SEND followed by STOP
0	-	0	1	1	Repeat START followed by SEND
0	-	1	1	1	Repeat START, followed by SEND and STOP
1	0	0	1	1	Repeat the START condition followed by the answer RECAIVE operation (Master remains in receiver mode)
1	0	1	1	1	Repeat START, followed by SEND and STOP conditions
1	1	0	1	1	Repeat the START condition followed by RECEIVE (Master remains in receiver mode)
1	1	1	1	1	Disable combinations

Combination of control bit (master receive state)

R/S	ACK	STOP	START	RUN	OPERATION
-	0	0	0	1	RECEIVE operation with answering (Master remains in receiver mode)
-	-	1	0	0	STOP
-	0	1	0	1	RECEIVE followed by STOP
-	1	0	0	1	RECEIVE operation (master remains in receiver mode)
-	1	1	0	1	Disable combinations
1	0	0	1	1	Repeat START, followed by an answer RECEIVE operation (Master remains in receiver mode)
1	0	1	1	1	Repeat START, followed by RECEIVE and STOP
1	1	0	1	1	Repeat START followed by RECEIVE (Master remains in receiver mode)
0	-	0	1	1	Repeat START followed by SEND (master stays in transmitter mode)
0	-	1	1	1	Repeat START, followed by SEND and STOP

Status registers 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 transmit failure.
(write 0 to reset);
0= No interrupt occurred.

Bit6 BUS_BUSY: The I^2C bus busy flag bit in master mode/salve mode;
1= The I^2C bus is working, that cannot transmit 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: Failure flag in I^2C master control mode;
1= No response of addressing slave machine and arbitration conflict.
0= ---

Bit0 BUSY: Busy flag in I^2C master control mode;
1= Sending data.
0= ---

14.3.3 I²C Slave Address Register

The slave address register consists of 8 bits: 7 address bits (A6-A0) and receive/non-transmit bits R/S. The R/S bits determine whether the next operation is receive (1) or transmit (0).

Master mode slave address register I2CMSA

0xF4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CMSA	SA6	SA5	SA4	SA3	SA2	SA1	SA0	R/S
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit7~Bit1 SA6-SA0: I² C Slave address in master mode.

Bit0 R/S: I² C Receive/send status selection bit after sending the slave address in master mode;

1= Receiving data after correct addressing;

0= Send data after correct addressing.

14.3.4 I²C Master Mode Transmit and Receive Data Registers

The transmit data register consists of eight data bits that will be sent on the bus on the next transmit or burst transmit operation, and the first transmission bit is MD7 (MSB).

Master mode data cache register I2CMBUF

0xF6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CMBUF	MD7	MD6	MD5	MD4	MD3	MD2	MD1	MD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit7~Bit0 MD7-MD0: I² C Send/receive data in master mode.

14.4 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	
Self-address register I2CSADR	Self-address register I2CSADR	0xF1
Control register I2CSCR	Status register I2CSSR	0xF2
Send data I2CSBUF	Receive data I2CSBUF	0xF3

14.4.1 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 ---

Bit6~Bit0 SA6-SA0: Own address in I²C salve mode.

14.4.2 I²C Slave Mode Control and Status Register 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 (write only)

Read: read I2CSSR (read only)

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 in I²C slave mode;

1= Reset slave module;

0= No effect.

Bit6~ Bit1 ---

Bit0 DA: Enable bit in I²C slave mode;

1= Enable;

0= Disable.

The status register consists of three bits: the SENDFIN bit, the RREQ bit, and the TREQ bit. The Send Complete SENDFIN bit indicates that the host I²C controller has completed receiving data during an I2CS single or continuous transmit operation. The Receive Request RREQ bit indicates that the I2CS device has received a data byte from the I²C host and that the I2CS device should read a data byte from the Receive Data Register I2CSBUF. The Transmit Request TREQ bit indicates that the I2CS device is addressed as a slave transmitter and the I2CS device shall write a data byte to the transmit data register I2CSBUF. If the I²C interrupt enable is on, an interrupt is generated by setting any of the three flag bits to 1.

The bus busy flag bit in slave mode is judged by Bit6 (BUS_BUSY) of I2CMSR in master mode status register. I2CMSR is 0x20 when the bus is idle, and I2CMSR register is 0x60 when the start condition is generated until the stop condition is generated.

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 ---

Bit2 SENDFIN: Sending complete flag bit In I²C slave mode (read only).

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 bit In I²C slave mode (read only).

1= The master control device is ready to receive data.
(automatically write zero after writing I2CSBUF)

0= ---

Bit0 RREQ: Receiving complete flag bit In I²C slave mode (read only).

1= Receive complete. (automatically write zero after reading I2CSBUF)

0= Receive uncompleted.

14.4.3 I²C Slave mode transmit and receive buffer register I2CSBUF

0xF3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CSBUF	I2CSBUF7	I2CSBUF6	I2CSBUF5	I2CSBUF4	I2CSBUF3	I2CSBUF2	I2CSBUF1	I2CSBUF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

Bit7~ Bit0 I2CSBUF: I²C Sending/receiving data.

Write: Send data (Send order from MSB to LSB).

Read: Read received dates.

14.5 I²C Interrupt

The interrupt number of I²C is 21, where the interrupt vector is 0x00AB. To enable I²C interrupt, the enable bit(I2CIE) is set to 1, and the master interrupt enable bit(EA) is set to 1.

If all the interrupt enable is turned on and master interrupt 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 1.

Interrupt mask register (EIE2)

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

Bit7 SPIIE: SPI Interrupt enable bit
1= Enable SPI interrupt;
0= Disable SPI interrupt.
Bit6 I2CIE: I²C Interrupt enable bit
1= Enable I²C interrupt;
0= Disable I²C interrupt.
Bit5 WDTIE: WDT Interrupt enable bit
1= Enable WDT interrupt;
0= Disable WDT interrupt.
Bit4 ADCIE: ADC Interrupt enable bit
1= Enable ADC interrupt;
0= Disable ADC interrupt.
Bit3 PWMIE: PWM global interrupt enable bit
1= Enable PWM global interrupt;
0= Disable PWM global interrupt.
Bit2 ---
Bit1 ET4: Time4 Interrupt enable bit
1= Enable Time4 interrupt;
0= Disable Time4 interrupt.
Bit0 ET3: Time3 Interrupt enable bit
1= Enable Time3 interrupt;
0= Disable Time3 interrupt.

Peripheral interrupt priority control register EIP2

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

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

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

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

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

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

Bit2 ---

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

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

Peripheral interrupt flag register EIF2 (0xB2)

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 bus interrupt instruction, Read only;
1= SPI interrupt, (Automatically cleared after clearing the specific interrupt flag);
0= No SPI interrupt.

Bit6 I2CIF: I^2C Total interrupt instruction, Read only;
1= I^2C interrupt, (Automatically cleared after clearing the specific interrupt flag);
0= No I^2C interrupt.

Bit5 ---

Bit4 ADCIF: ADC interrupt flag;
1= ADC Conversion completed, need software clear;
0= ADC conversion is not complete.

Bit3 PWMIF: PWM Total interrupt instruction, Read only;
1= PWM interrupt, (Automatically cleared after clearing the specific interrupt flag);
0= No PWM interrupt.

Bit2 ---

Bit1 TF4: Timer4 counter overflow interrupt flag;
1= Timer4 enter the interrupt service routine, the hardware automatically clears, or can be cleared by software;
0= Timer4 counter has no overflow.

Bit0 TF3: Timer3 counter overflow interrupt flag;
1= Timer3 enter the interrupt service routine, the hardware automatically clears, or can be cleared by software;
0= Timer3 counter has no overflow.

14.6 I²C Transmission Mode In Slave Mode

This section describes all available transmission mode in the kernel. The default address of all the waveform is 0x39(“00111001”).

14.6.1 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.

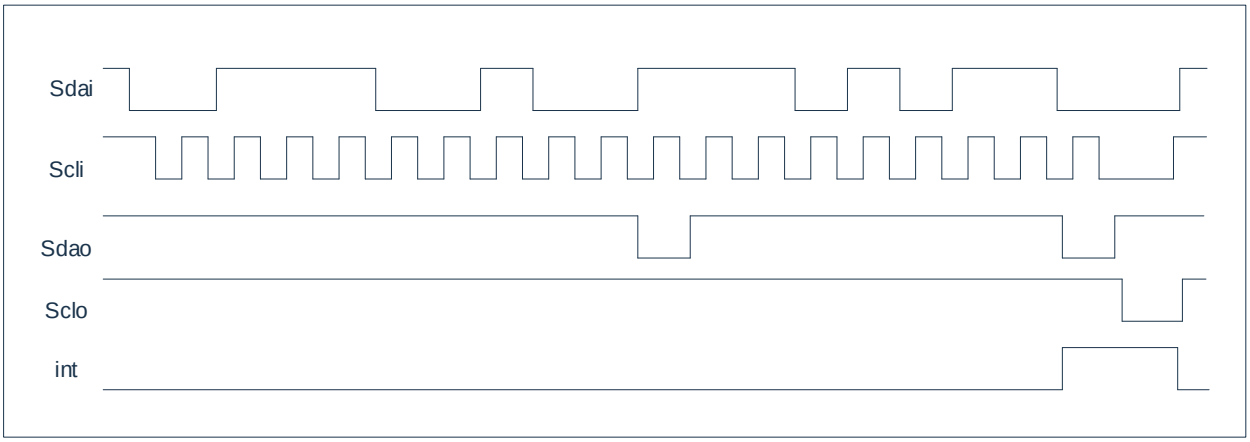


Figure 14-2: Timing diagram of single receiving

14.6.2 Single Sending

The following figure shows the sequence of signals sent by I²C during a single data period. Single transmission sequence:

- 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.

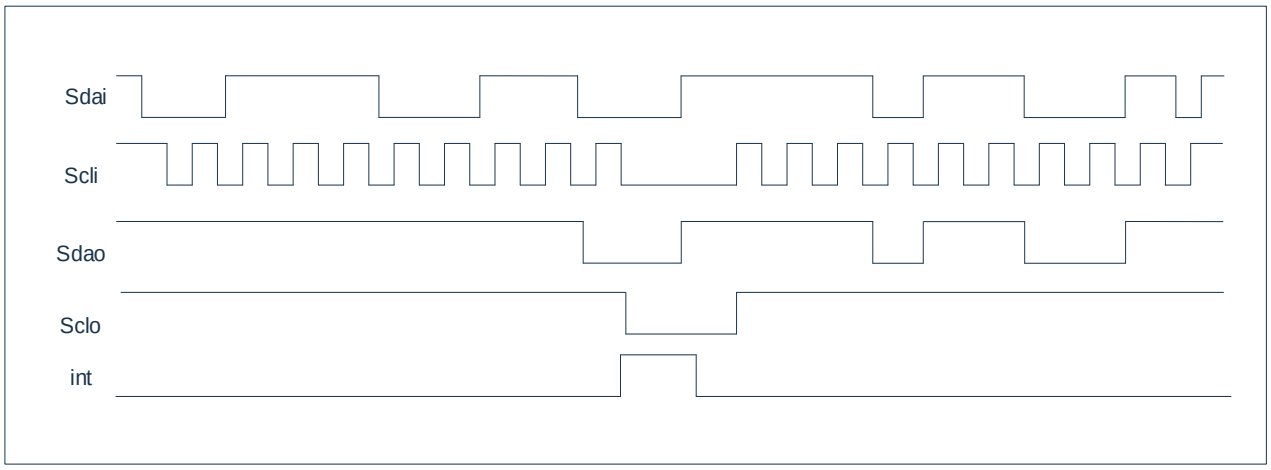


Figure 14-3: Timing diagram of single sending

14.6.3 Continuous Receiving

The following figure shows the sequence of signals received by I²C during burst data. Burst receiving sequence is as follows:

- 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 condition occurs.

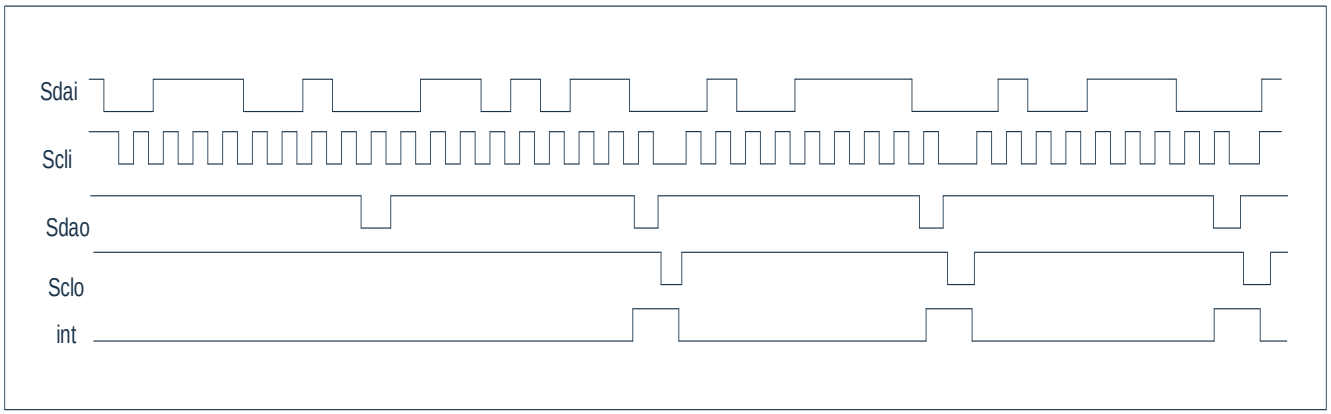


Figure 14-4: Timing diagram of continuous receiving

14.6.4 Continuous Sending

The following figure shows the sequence of signals sended by I²C during burst data. Burst sending sequence is as follows:

- Starting conditions.
- Addressing by I²C host machine.
- The address is confirmed by I²C.
  (1) The data is sended 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) occurs.

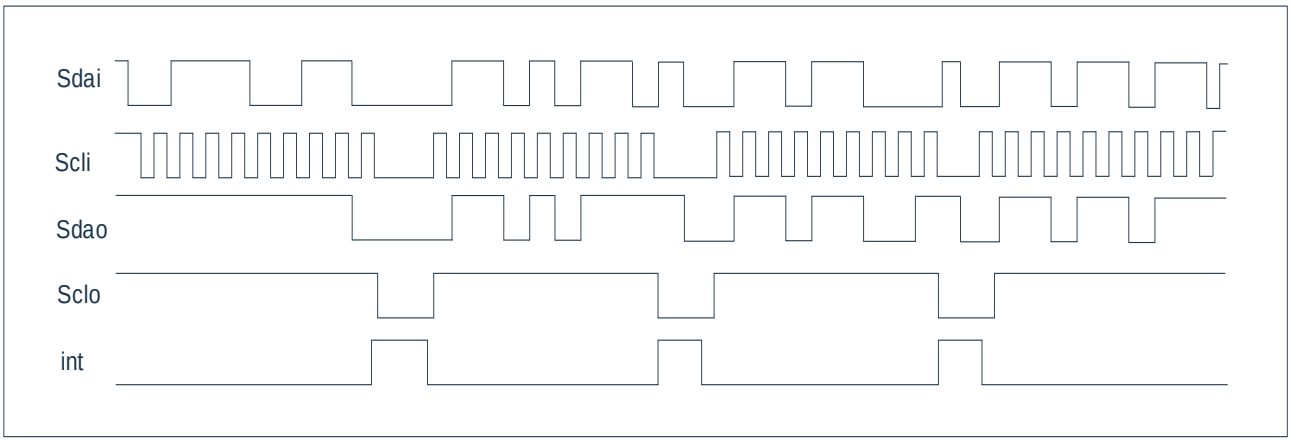


Figure 14-5: Timing diagram of continuous sending

15. BRT MODULE

15.1 Introduction

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

15.2 Function Description

BRT has one 16-bit counter internally. The initial value of the counter is loaded by {BRTDH, BRTDL}, and the counter starts to operate when BRTEN=1. Its clock comes from the prescaler circuit, and the prescaler clock is determined by BRTCKDIV.

The counter works as an add counter, and the BRT counter overflows when the value of the 16-bit counter is equal to FFFFH. The value of {BRTDH, BRTDL} is automatically loaded into the counter after the overflow, and then the counting is done again.

The overflow signal of the BRT counter is provided exclusively to the UART module as a clock source for the baud rate. No interrupts are generated on overflow. There is also no corresponding interrupt structure.

When BRT is in debug mode, its clock will not stop. If the UART module has started to send or receive data, the UART will complete the whole process of sending or receiving even if the chip goes into stop state.

BRT timer overflow rate:

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

15.3 Register Description

BRTCON register

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

BRTEN: BRT timer enable bit;
Bit7 1= Enable;
0= Disable.
Bit6-3 --- ---
Bit2~Bit0 BRTCKDIV<2:0> BRT timer prescaler selection bit;
000= Fsys/1;
001= Fsys/2;
010= Fsys/4;
011= Fsys/8;
100= Fsys/16;
101= Fsys/32;
110= Fsys/64;
111= Fsys/128;

BRTDL register

F5C1H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BRTDL								
R/W	R/W	R/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>: Lower 8 bits of the BRT timer load value;

BRTDH register

F5C2H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BRTDH								
R/W	R/W	R/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>: Higher 8 bits of the BRT timer load value;

The following table gives some of the baud rate related information for the baud rate timer overflow rate as the UART clock source in variable baud rate mode:

SMODn=0, BRTCKDIV=0

	Fsys=8MHz			Fsys=16MHz			Fsys=24MHz		
	{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
9600	65510	9615	-0.16	65484	9615	-0.16	65458	9615	-0.16
19200	65523	19231	-0.16	65510	19231	-0.16	65497	19231	-0.16
38400	---	---	---	65523	38462	-0.16	65516	37500	2.34
115200	---	---	---	---	---	---	---	---	---

SMODn=1, BRTCKDIV=0

	Fsys=8MHz			Fsys=16MHz			Fsys=24MHz		
	{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
9600	65484	9615	-0.16	65432	9615	-0.16	65380	9615	-0.16
19200	65510	19231	-0.16	65484	19231	-0.16	65458	19231	-0.16
38400	65523	38462	-0.16	65510	38462	-0.16	65497	38462	-0.16
115200	---	---	---	---	---	---	65523	115385	-0.16

16. UARTn MODULE (n=0,1)

16.1 Introduction

There are 2 UART modules included in the chip: UART0 and UART1,they have identical functions.

The serial port is full duplex, which means it can transmit and receive synchronously. Receive module is double-buffered, which means it can start to receive a second byte before a previously received byte has been read from the receive register. Data are loaded to transmit register by writing SBUFn register, and it could access a receive register with independent physical addresses by reading SBUFn register.

The serial port can operate in 4 modes: one synchronous and three asynchronous modes. Mode 2 and 3 has special features 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.

16.2 UARTn Port Configuration

It needs to configure related ports as UARTn channel before using UARTn module:

P25CFG = 0x02; //P2.5 is configured as TXD0 channel

P26CFG =0x02; //P2.6 is configured as RXD0 channel, this port is automatically configured as open drain output with pull-up register when using synchronous mode.

P35CFG = 0x02; //P3.5 is configured as TXD1 channel

P21CFG =0x02; //P2.1 is configured as RXD1 channel, this port is automatically configured as suggested that it sets up work mode firstly, then to configure related ports as serial port.

16.3 UARTn Baud Rate

In mode 0, the baud rate is fixed at twelve divisions of the system clock (Fsys/12); in mode 2, the baud rate is fixed at thirty-two divisions of the system clock or sixty-four divisions (Fsys/32, Fsys/64); in modes 1 and 3, the baud rate is generated by Timer1 or Timer4 or Timer2 or BRT module, and the choice of timer as the baud rate clock source is determined by register FUNCCR.

When {FUNCCR[2],FUNCCR[0]}=00, Timer1 is selected as the baud rate generator of UART0;
When {FUNCCR[2],FUNCCR[0]}=01, Timer4 is selected as the baud rate generator of UART0;
When {FUNCCR[2],FUNCCR[0]}=10, Timer2 is selected as the baud rate generator of UART0;
When {FUNCCR[2],FUNCCR[0]}=11, BRT is selected as the baud rate generator of UART0.
When {FUNCCR[3],FUNCCR[1]}=00, Timer1 is selected as the baud rate generator of UART1;
When {FUNCCR[3],FUNCCR[1]}=01, Timer4 is selected as the baud rate generator of UART1;
When {FUNCCR[3],FUNCCR[1]}=10, Timer2 is selected as the baud rate generator of UART1;
When {FUNCCR[3],FUNCCR[1]}=11, BRT is selected as the baud rate generator of UART1.

Equation for baud rate for Timer1 or Timer4 operating in 8-bit auto-reload mode:

$$BaudRate=\frac{Fsys\times 2^{SMODn}}{32\times\left(4\times 3^{1-TxM}\right)\times(256-THx)}\ (x=1,4)$$

SMODn is the baud rate selection bit, set by register PCON. T1M is the Timer1 clock selection bit, set by register CKCON[4], and T4M is the Timer4 clock selection bit, set by register T34MOD[6]. That is, the value of TH1/TH4 for Timer1 or Timer4 at the corresponding baud rate should be set as follow:

$$THx=256-\frac{Fsys\times 2^{SMODn}}{32\times\left(4\times 3^{1-TxM}\right)\times BaudRate}\ (x=1,4)$$

Equation for baud rate for Timer2 operating in overflow auto-reload mode:

$$BaudRate=\frac{Fsys\times 2^{SMODn}}{32\times\left(12\times 2^{T2PS}\right)\times(65536-\{RLDH,RLDL\})}$$

T2PS is Timer2 clock prescaler selection bit, set by register T2CON[7]. That is, the value of Timer2 at the corresponding baud rate{RLDH,RLDL}
should be set to:

$$\{RLDH,RLDL\}=65536-\frac{Fsys\times 2^{SMODn}}{32\times\left(12\times 2^{T2PS}\right)\times BaudRate}$$

BRT Baud rate equation when used as a baud rate generator:

$$BaudRate=\frac{Fsys\times 2^{SMODn}}{32\times(65536-\{BRTDH,BRTDL\})\times 2^{BRTCKDIV}}$$

BRTCKDIV is the BRT timer prescaler selection bit, which is set by register BRTCON. That is, the BRT at the corresponding baud rate{BRTDH,BRTDL} value should be set as follows:

$$\{BRTDH,BRTDL\}=65536-\frac{Fsys\times 2^{SMODn}}{32\times 2^{BRTCKDIV}\times BaudRate}$$

The following table shows the related information of partial baud rate for Timer1/Timer4 8-bit auto-reload mode under variable baud rate mode:

SMODn=0

	Fsys=8MHz				Fsys=16MHz				Fsys=24MHz			
	TH1 or TH4	T1M or T4M	Actual Rate	% Error	TH1 or TH4	T1M or T4M	Actual Rate	% Error	TH1 or TH4	T1M or T4M	Actual Rate	% Error
4800	243	1	4808	-0.16	230	1	4808	-0.16	217	1	4808	-0.16
9600	---	---	---	---	247	1	9615	-0.16	236	1	9375	2.34
19200	---	---	---	---	---	---	---	---	246	1	18750	2.34
38400	----	---	---	---	---	---	---	---	251	1	37500	2.34

SMODn=1

	Fsys=8MHz				Fsys=16MHz				Fsys=24MHz			
	TH1 or TH4	T1M or T4M	Actual Rate	% Error	TH1 or TH4	T1M or T4M	Actual Rate	% Error	TH1 or TH4	T1M or T4M	Actual Rate	% Error
4800	230	1	4808	-0.16	204	1	4808	-0.16	178	1	4808	-0.16
9600	243	1	9615	-0.16	230	1	9615	-0.16	217	1	9615	-0.16
19200	---	---	---	---	243	1	19230	-0.16	236	1	18750	2.34
38400	----	---	---	---	---	---	---	---	246	1	37500	2.34

16.4 UARTn Register

The UARTn has the same functionality as a standard 8051 UART. The UART0 related registers are: SBUFn, SCONn, PCON(0x87), IE(0xA8) and IP(0xB8). The UARTn data buffer (SBUFn) consists of two separateregisters: transmit and receive registers. A data written into the SBUFn will be set in UARTn output register and startsa 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.

UART 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

SBUF register address: 0x99; SBUF1 register address: 0xEB;

Bit7~Bit0 BUFFERn<7:0>: Data buffer register.

Write: UARTn starts to transmit data.

read: Read received data.

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

SCON0 register address: 0x98; SCON1 register address: 0xEA.

Bit7~Bit6 UnSM0- UnSM1: Control bit for multi-processor communications;

00= Master 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 bit;

1= Enable;

0= Disable.

Bit3 UnTB8: The 9th bit of transmitting data, mainly used for transmitting of 9-bit asynchronous mode;

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 asynchronous mode;

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 is listed in the table below:

SMn0	SMn1	Mode	Description	Baud rate
0	0	0	Shift register	Fsys/12
0	1	1	8-Bit UART	variable
1	0	2	9-Bit UART	Fsys/32 or /64
1	1	3	9-Bit UART	variable

UARTn baud rate

Mode	Baud rate
Mode0	Fsys/12
Mode1, 3	Controlled by Timer4/Timer1/Timer2/BRT, see section 16.3.
Mode2	SMODn=0: Fsys/64 SMODn=1: Fsys/32

SMODn bit is in the power management control register 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

Bit7 SMOD0: The UART0 baud rate multiplier bit;
1= Doubling the UART0 baud rate;
0= UART0 baud rate is normal.
Bit6 SMOD1: UART1 baud rate multiplier bit;
1= Doubling the UART1 baud rate;
0= UART1 baud rate is normal.
Bit5 ---
Bit4~Bit3 Reserved bit: Must be 0.
Bit2 SWE: STOP status function wake-up enable bit;
(The system can be restarted by a power-down reset or an enabled external reset, regardless of the SWE value)
0= Disable of functional wake-up calls;
1= Enable function wake-up (can be woken up by external interrupt and timed wake-up)
Bit1 STOP: The dormant state control bit;
1= Entering the sleep state (automatically cleared by exiting STOP mode);
0= Not in the dormant state.
Bit0 IDLE: Idle status control bit;
1= Entering the idle state (automatically cleared when exiting IDLE mode);
0= Not in idle state.

16.5 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. To enable the UARTn interrupt, you must set its enable bit ESn to 1 and set the total interrupt enable bit EA to 1.

If the interrupt enable of UARTn is on and TIn=1 or RIn=1, the CPU will enter the corresponding interrupt service program. TIn/RIn is not related to the status of ESn. And it needs to be cleared by software.

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: Overall interrupt enable bit;
1= Enable all un-masked interrupt;
0= Disable all interrupt.

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

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

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

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

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

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

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

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 ---

Bit6 PS1: UART1 interrupt priority control bit;
1= Set to advanced interrupt;
0= Set to low level interrupt;

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

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

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

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

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

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

UART control register SCONn

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SCONn	SMn0	SMn1	SMn2	RENn	TBn8	RBn8	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

Register SCON0 address:0x98;Register SCON1 address:0xEA.

Bit7~Bit6	SMn0-SMn1:	Control bit for multi-processor communications;
	00=	Master 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	SMn2:	Enable bit for multi-processor communications;
	1=	Enable;
	0=	Disable.
Bit4	RENn:	Receive enable bit;
	1=	Enable;
	0=	Disable.
Bit3	TBn8:	The 9th bit of transmitting data, mainly used for transmitting of 9-bit asynchronous mode;
	1=	The 9th bit is 1;
	0=	The 9th bit is 0.
Bit2	RBn8:	The 9th bit of receiving data, mainly used for receiving of 9-bit asynchronous mode;
	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=	---

16.6 UARTn Mode (n=0/1)

16.6.1 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. Reception is initialized by setting the flags in SCON0 as follows: RIn=0 and RENn=1.

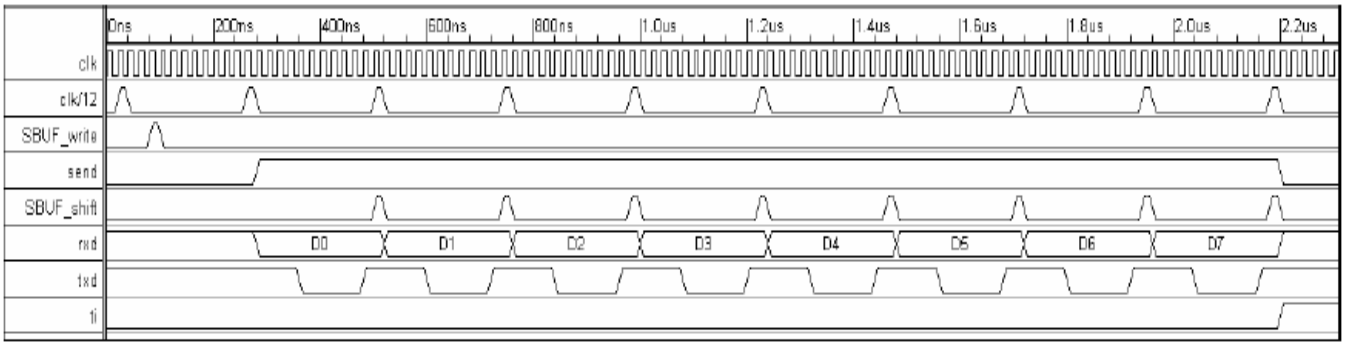


Figure 16-1: UARTn communication mode 0 timing diagram

16.6.2 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 or TIMER4.

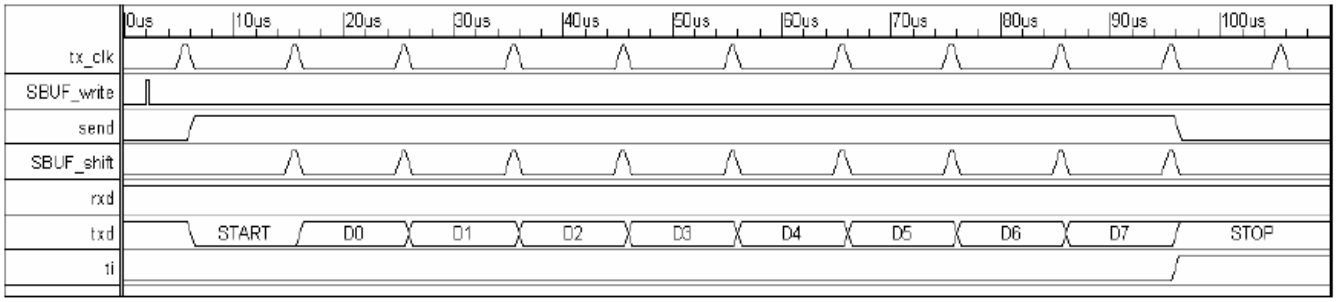


Figure 16-2: UARTn communication mode 1 timing diagram

16.6.3 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, and11 bits are transmitted or received: a start bit (0), 8 data bits (LSB first), a programmable 9th bit, and a stop bit (1). The9th bit can be used to control the parity checking of the UARTn interface: at transmission, bit TBn8 in SCONn is output as the 9thbit, and at receive, the 9th bit affects RBn8 in SCONn.

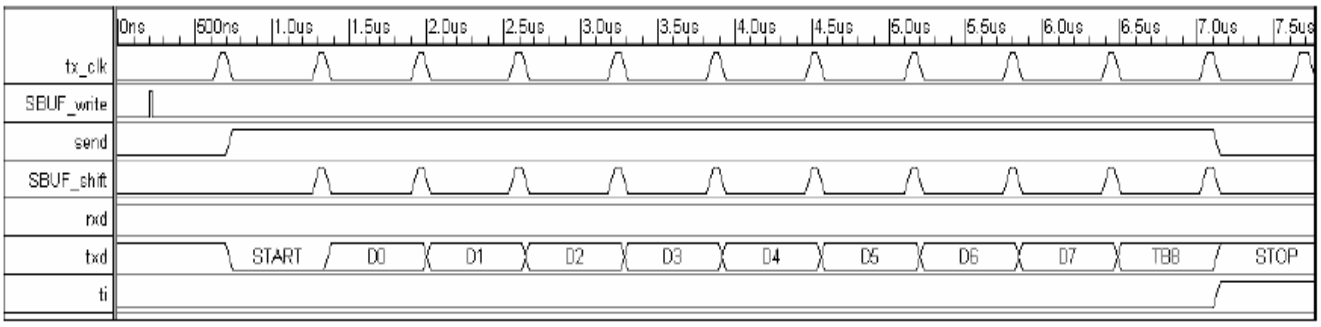


Figure 16-3: UARTn communication mode 2 timing diagram

16.6.4 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,datareceiving is enabled. The baud rate is variable and depends on the mode of TIMER1 or TIMER4.

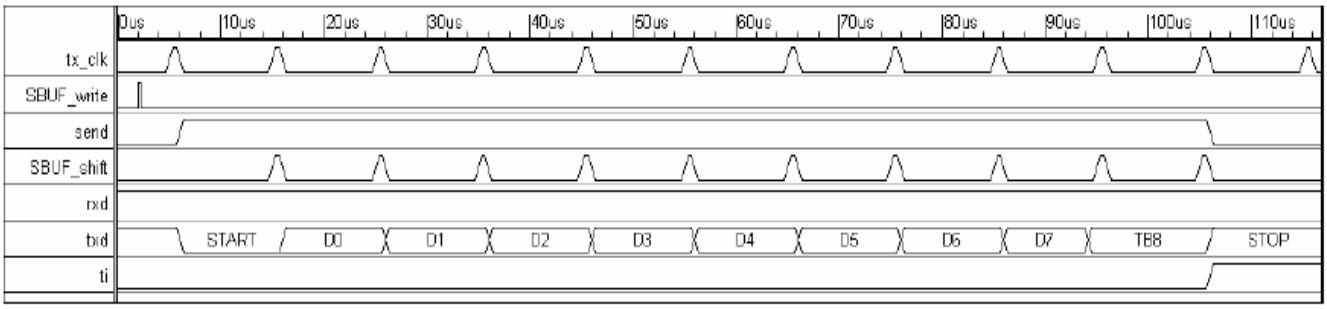


Figure 16-4: UARTn communication mode 3 timing diagram

17. SPI MODULE

17.1 SPI Introduction

The SPI is a fully configurable SPI master/slave device, which allows user to configure polarity and phase of serial clock signal SCLK. The SPI allows the microcontroller to communicate with serial peripheral devices. It is also capable of inter process or communications in a multi-master system. A serial clock line (SCLK) synchronizes shifting and sampling of the information on the two independent serial data lines. SPI data are simultaneously transmitted and received. 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. The system can be configured as a master or a slave device. Data rates as high as System clock divided by four (Fsys/4). Clock control logic allows a selection of clock polarity and a choice of two fundamentally different clocking protocols to accommodate most available synchronous serial peripheral devices. When the SPI is configured as a master, software selects one of four different bit rates for the serial clock.

The SPI automatically drive selected by SSCR (Slave Select Control Register) slave select outputs (NSS3, NSS5-NSS7), and address SPI slave device to exchange serially shifted data. Error-detection logic is included to support inter process or communications. A write conflict detector indicates when an attempt is made to write data to the serial shift register while a transfer is in progress. A multiple-master mode-fault detects or automatically disables SPI output drivers if more than one SPI devices simultaneously attempts to become bus master.

All features listed below are included in the current SPI:

- Full duplex synchronous serial data transfer;
- Master mode and Slave mode supported;
- Multi-master system supported;
- Up to 4 SPI slaves can be addressed;
- System error detection;
- Interrupt generation;
- Supports speeds up to 1/4 of system clock;
- 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.

17.2 SPI Port Configuration

First, configure the related ports as SPI channels ,as follows:

```
P15CFG = 0x04;  // P1.5 is Configured as SCLK channel
P16CFG = 0x04;  //P1.6 is configured as a MOSI channel
P17CFG = 0x04;  // P1.7 is configured as a MISO channel
```

In the SPI Master mode, the signal output of the driven device can be selected, NSS3, NSS5-NSS7can select 4 slave devices at the same time. The SSCR register determines the output enable of NSSx(x=3, 5-7),The relevant ports also need to be configured as NSS ports, For example, configure P1.4 to NSS3 port:

```
P14CFG = 0x04;       //P1.4is Configured as NSS3 channel
SSCR&= 0xF7;         //Enable NSS3 output
```

In the SPI Master mode, Instead of using the internal NSSx to output the selected signal of the driven device, the GPIO function can be used to control the selected signal:

```
SPSR&=0xFE;     // SSCEN =0
```

In the SPI slave mode, The selected signal NSS is input from NSS7:

```
SPSR |= 0x02;        //NSS7 as the NSS input
P36CFG = 0x04;      //P36 port is configured as NSS7 port
```

> Note: Only NSS7 can be configured as a chip selection channel in master mode or slave mode, while other NSS3, nss5-nss6 can only be configured as a chip selection channel in master mode.

Port SCLK、MOSI、MISO and NSSx can be set with pull-up resistors and open-drain output switches through the relevant port registers are as shown below:

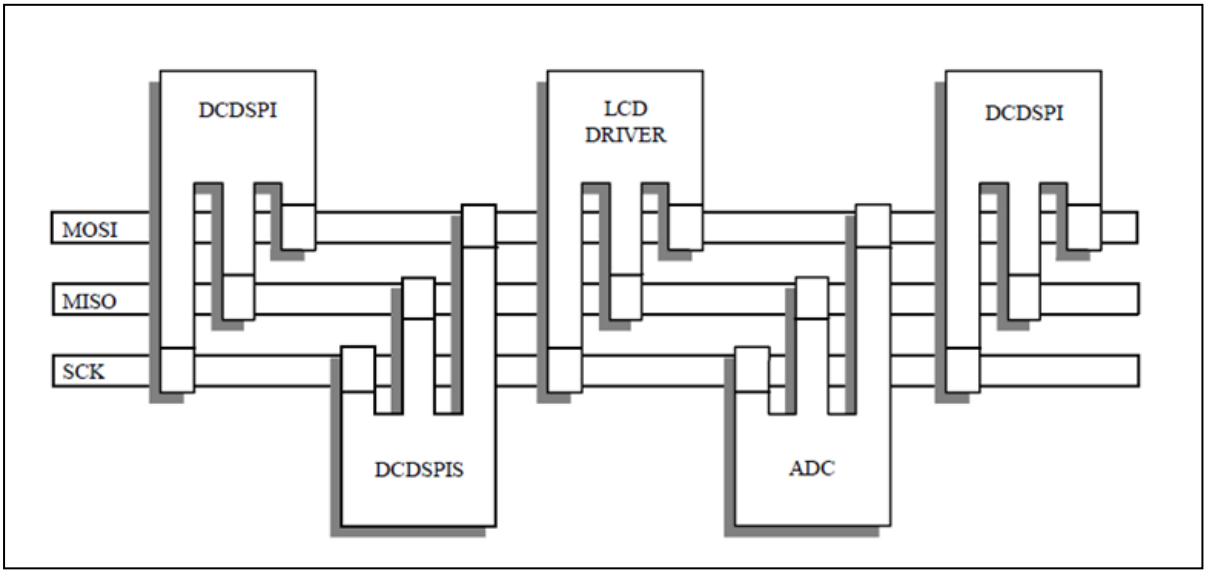


Figure 17-1: Multiple SPI Communication

17.3 SPI Block Diagram

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. Another way to view this transfer is that 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 eight 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.

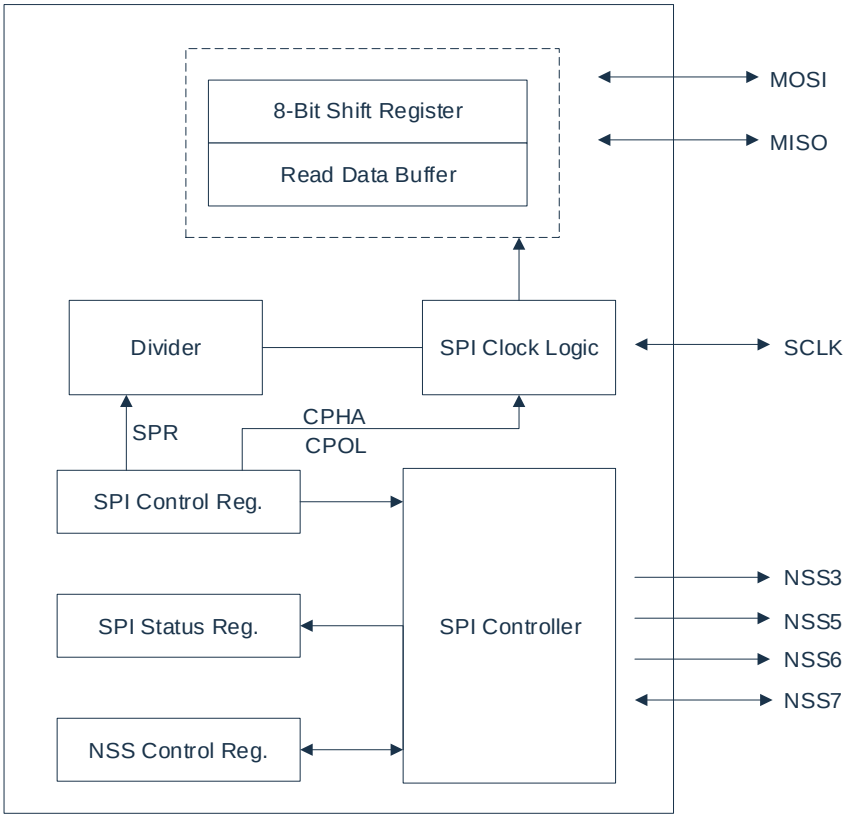


Figure 17-2: SPI Block Diagram

The pins associated with the SPI are: NSSx(x=3,5,6,7),SCLK,MOSI,MISO.

The NSSx input pin in a master mode is used to detect mode-fault errors. A low on this pin indicates that some other device in a multi-master system has become a master and trying to select the DSPI as a slave. The NSSx input pin in a slave mode is used to enable transfer.

In master mode, the SCLK pin is used as a reference for the SPI clock signal. When the host device initiates transmission, eight clock cycles will be 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.

17.4 SPI Register

17.4.1 SPCR -SPI Control Register

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 ---

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 SPR1-SPR0: SPI clock rate select :[1:0]

(See the table below for frequency control details)

SPR2-SPR0 control SPI clock divider

SPR2	SPR1	SPR0	System clock divided by
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

17.4.2 SPDR -SPI Data Register

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: Data sent or received by the SPI.

Write: Write the data to be sent (Send order from MSB to LSB).

Read: Received data.

17.4.3 SSCR –Slave Select Control Register

The control register may 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 NSS3, NSS5-NSS7 pins when SPI master transmission starts.

0xEF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SSCR	SSO7	SSO6	SSO5	---	SSO3	---	---	---
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 SSO[i]: SPI slave select, i=3,5-7
0= When SPI host transmission starts, NSSi output 0.
1= When SPI host transmission starts, NSSi output 1.

17.4.4 SPSR-SPI State Register

0xED	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPSR	SPISIF	WCOL	---	---	---	---	SSICS	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 conflict error flag Read only;
1= SPI Write conflict error (Read SPSR first, then clear after reading/writing SPDR);
0= No Write conflict error.
Bit5 ---
Bit4 ---
Bit3~Bit2 ---
Bit1 SSICS: It must be 1 in SPI slave mode
Bit0 SSCEN: SPI master control mode NSSx (x=3,5-7)Output control bit.
1= NSSx outputs high when SPI is being in idle state;
0= NSSx outputs the content of 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 the automatic slave selection output. When SSCEN is set to 1, the NSSx line outputs the contents of the SSCR register when the transmission is in progress, and the NSSx goes high when the transmission is idle. When the SSCEN bit is cleared to zero, the NSSx line always displays the contents of the SSCR register.

17.5 SPI Master

When the SPI is configured in host mode, the transfer is initiated by writing to the SPDR register. When a new byte is written to the SPDR register, the SPI begins transmission of the nearest BAUD timer overflow. The serial clock, SCLK, is generated by the SPI. In host mode, and the SPI enables the SCLK output.

The SPI in host mode can select one of the four SPI slave devices via the NSSx line. the NSSx line-slave select output line is loaded with the contents of the SSCR register (0x03). the SSCEN bit of the SPSR register selects between automatic NSSx line control and software control. With the SSCEN bit in master mode, when SSCEN is set to 1, the NSSx line outputs the contents of the SSCR register when the transmission is in progress and NSSx is high when the transmission is idle. When the SSCEN bit is cleared to zero, the NSSx line is controlled by software and always displays the contents of the SSCR register, whether the transmission is in progress or the SPI is idle.

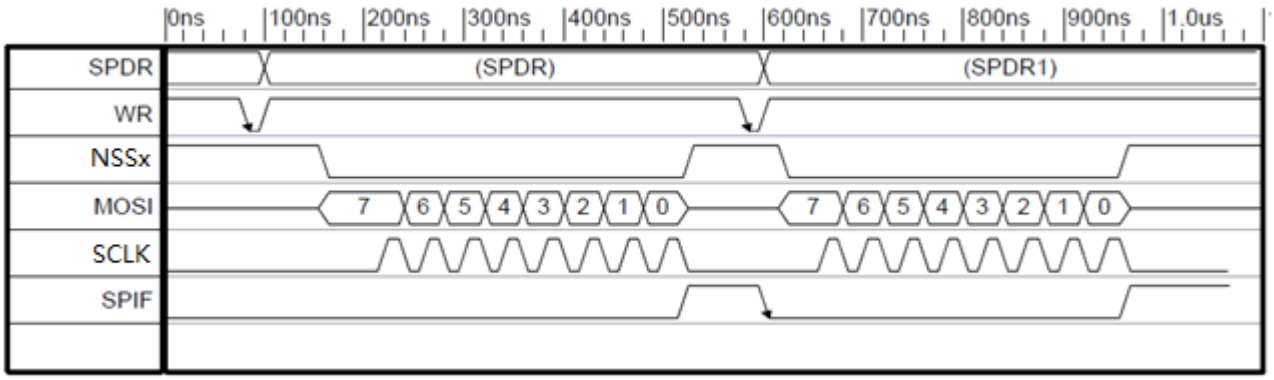


Figure 17-3: Automatic slave selection line usage

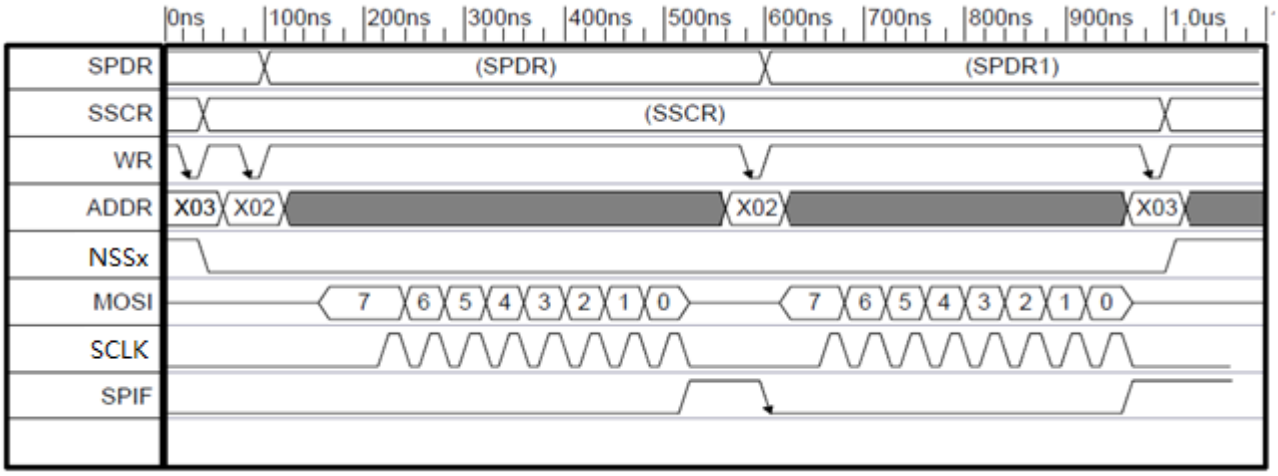


Figure 17-4: Software controlled NSSx line

17.5.1 Write Conflict Error

A write conflict 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 conflict is indicated by the WCOL flag in SPSR (3) 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 steps:

- Read contents of the SPSR register
- Access to the SPDR register (read or write)

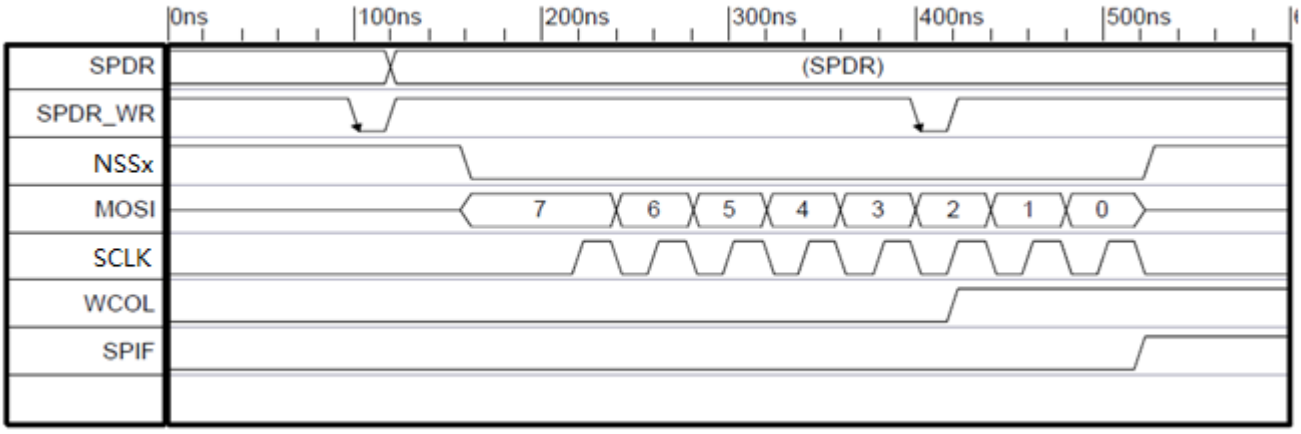


Figure 17-5: Write conflict error in SPI master mode

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 conflict 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 conflict while writing SPDR to enter the true data transfer state. 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.

Since the SPI has only one transmit buffer, it is recommended to judge whether the last data has been sent before writing the SPDR, and then confirm that the SPDR register is written after the transmission is completed to prevent a write conflict.

17.6 SPI Slave

When configured as SPI Slave the DSPI transfer is initiated by external SPI master module by assertion of the SPI Slave Select input, and generation of the SCLK serial clock.

Before transfer starts the SPI master has to assert the Slave Select line to determine which SPI slave will be used to exchange data. The NSS is asserted (cleared = 0), the clock signal connected to the SXCK 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 eight bits are shifted in/out the SPI generates the Interrupt request by setting the IRQ output.

In SPI slave mode only one transfer error is possible – Write Conflict Error.

17.6.1 Addressed Error

In slave mode only the write conflict error can be detected by the DSPI.

The write conflict 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 conflict error may occur as long as the SS 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.

17.6.2 Write Conflict Error

A write conflict occurs if the DSPI 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 conflict is indicated by the WCOL flag in SPSR (3) 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;
- access to the SPDR register (read or write).

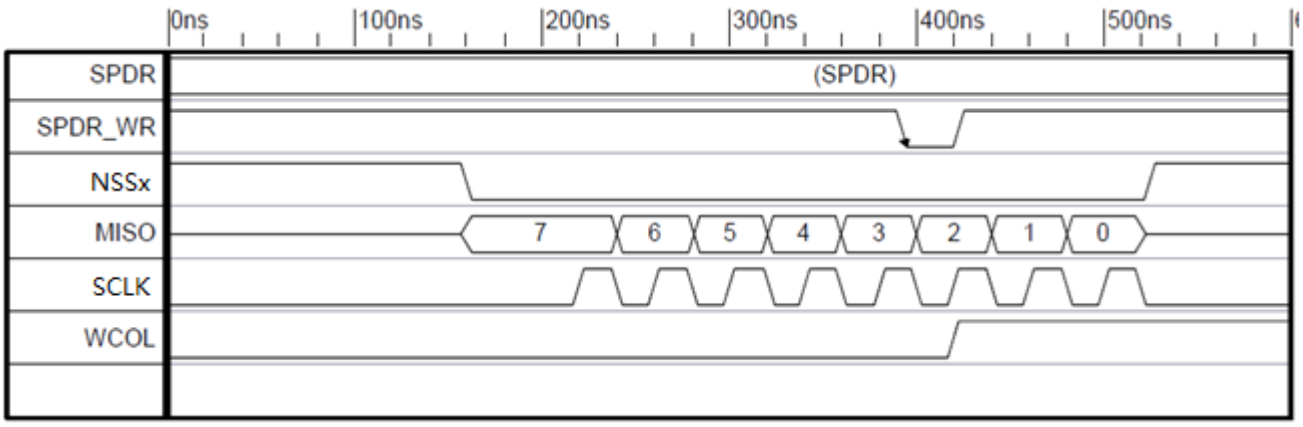


Figure 17-6: Write Conflict Error - SPI Slave mode – SPDR write during transfer

Figure below shows the WCOL generation, in case that the CPHA is cleared. As it is shown the WCOL generation is cause by any S{DR register write with NSS line cleared. It is done even if the SPI master didn't generate the serial clock SCLK. 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.

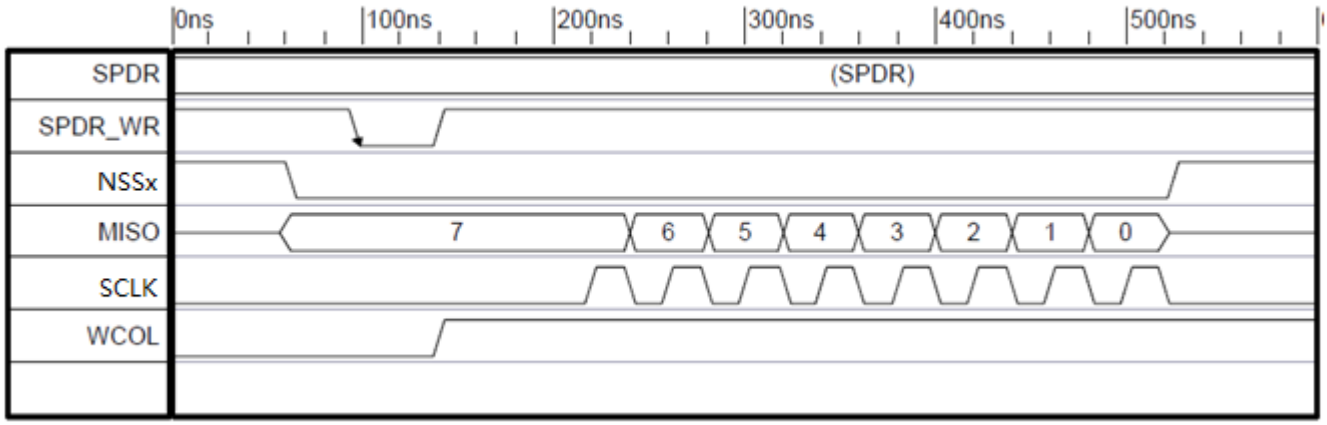


Figure17-7: WCOL Error-SPI Slave mode–SPDR write when CPHA = 0 and NSS = 0

After the SPDR is written in the slave mode, the host-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 conflict 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 conflict 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 conflict.

17.7 SPI Clock Control Logic

17.7.1 SPI Clock Phase and Polarity Control

Software can select any of four combinations of serial clock (SCLK) phase and polarity using two bits in the SPI control register (SPCR). The clock polarity is specified by the CPOL control bit, which selects an active high or active low clock and has no significant effect on the transfer 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.

17.7.2 SPI Transmit 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.

17.7.3 CPHA=0 Transmit 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.

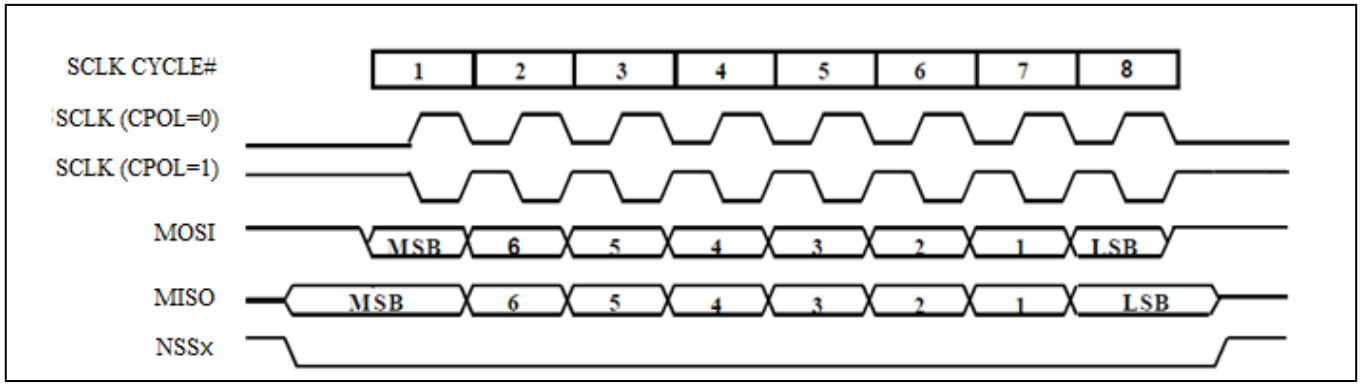


Figure 17-8: CPHA=0 SPI Transmit Format

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-conflict error results. 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.

17.7.4 CPHA=1 Transmit 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 SS line is the slave select input to the slave; the SS pin of the master is not shown but is assumed to be inactive. The SS pin of the master must be high or must be reconfigured as a general-purpose output not affecting the SPI.

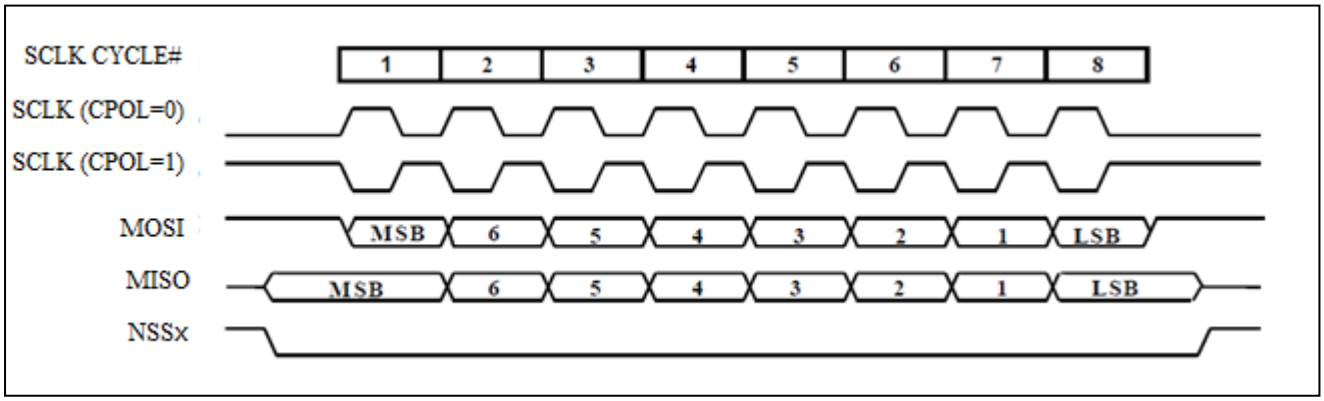


Figure 17-9: CPHA=1 SPI Transmit Format

17.8 SPI Data Transmission

17.8.1 SPI Transmission Start

All SPI transfers are started and controlled by a master SPI device. As a slave, the DSPI considers a 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.

17.8.2 SPI Transmission 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 SCLK 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.

17.9 SPI Timing Diagram

17.9.1 Master Mode Transmission

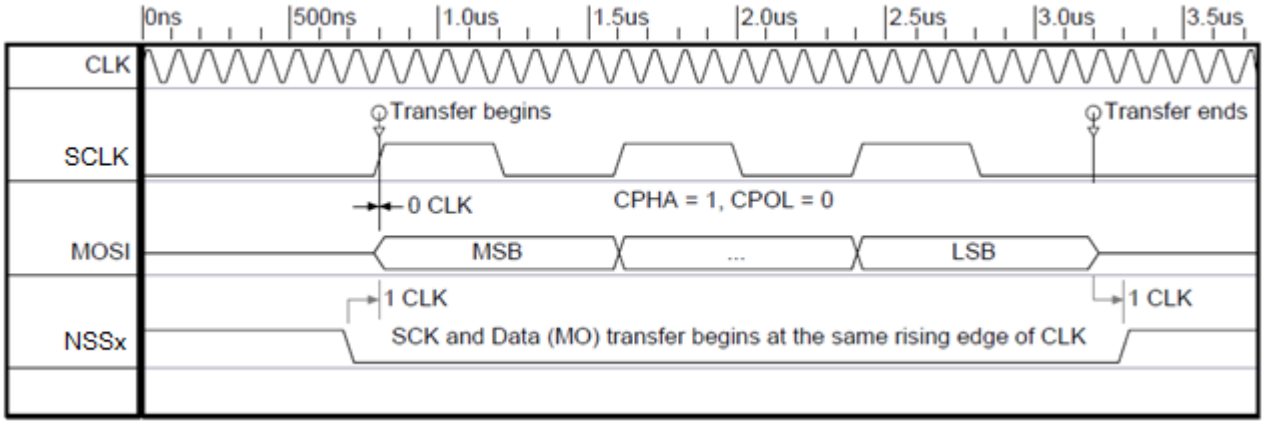


Figure 17-10: Master mode timing diagram

17.9.2 Slave Mode Transmission

At a beginning of transfer in Slave mode the data on serial output (MISO) appears on first rising edge after falling edge on Slave Select (NSS) line.

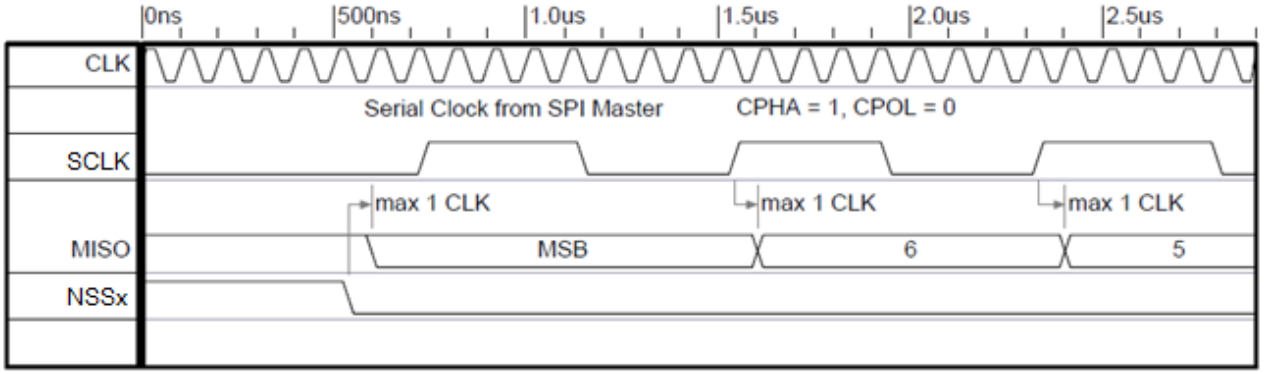


Figure 17-11: Slave mode timing diagram

17.10 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, write conflict WCOL, and mode error MODF, the SPI total interrupt indication bit, SPIIF, will be set. SPIIF is automatically cleared only when all three flags are 0.

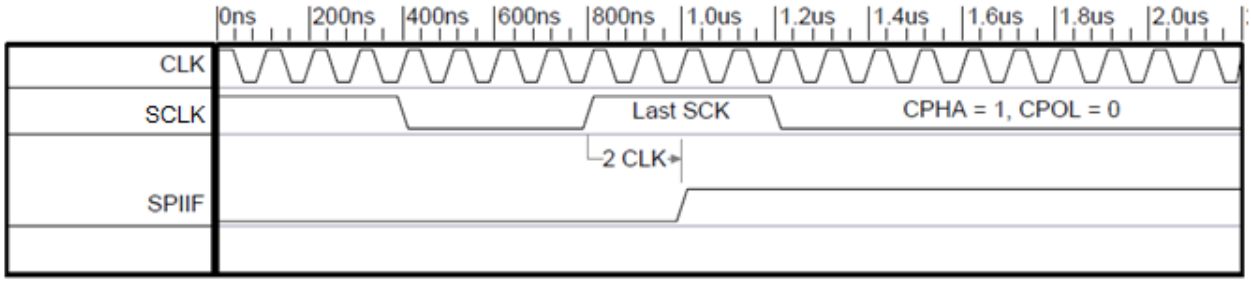


Figure 17-12: Interrupt generation

Interrupt mask register EIE2

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

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

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

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

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

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

Bit2 ---

Bit1 ET4: Time4 Interrupt enable bit
1= Enable Time4 interrupt;
0= Disable Time4 interrupt.

Bit0 ET3: Time3 Interrupt enable bit
1= Enable Time3 interrupt;
0= Disable Time3 interrupt.

Interrupt priority control register EIP2

0xB8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP2	PSPI	PI2C	PWDT	PADC	PPWM	---	PT4	PT3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0

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

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

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

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

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

Bit2 ---

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

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

Peripheral interrupt flag register EIF2 (0xB2)

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 generated interrupt, (this bit is automatically cleared when the specific interrupt flag bit is cleared);
0= SPI does not generate interrupts.
Bit6 I2CIF: I^2C total interrupt indication bit, read-only;
1= I^2C generates an interrupt, (this bit is automatically cleared when the specific interrupt flag bit is cleared);
0= I^2C does not generate an interrupt.
Bit5 ---
Bit4 ADCIF: ADC interrupt flag bit;
1= Completion of ADC conversion, requiring software clearing;
0= ADC conversion is not completed.
Bit3 PWMIF: PWM total interrupt indicator bit, read-only;
1= PWM generates an interrupt, (this bit is automatically cleared when the specific interrupt flag bit is cleared);
0= PWM does not generate interrupts.
Bit2 ---
Bit1 TF4: Timer4 counter overflow interrupt flag bit;
1= Timer4 counter overflow, which is automatically cleared by hardware when entering the interrupt service routine, or can be cleared by software;
0= Timer4 counter without overflow.
Bit0 TF3: Timer3 counter overflow interrupt flag bit;
1= Timer3 counter overflow, which is automatically cleared by hardware when entering the interrupt service program, or by software;
0= Timer3 counter without overflow.

18. FLASH MEMORY

Flash memory is divided into two parts for users: program area and data area.

- The size: program area 16K*8Bit; data area 1K*8Bit.
- The program area is divided into 32 sectors, one sector contains 512 bytes; the data area is 2 sectors.

The memory is readable under normal working conditions. It can also be indirectly addressed by a special function register (SFR). There are four SFR registers for accessing program memory:

- MCTRL
- MDATA
- MADRL
- MADRH
- MLOCK

When operating the memory module interface, the MDATA register is used as a byte to hold the 8-bit data to be read/ written. While the MADR register holds the address of the accessed MDATA unit. The memory allows byte read and write, and the byte write operation writes new data (erased before writing).The write time is controlled by the on-chip timer. The on-chip charge pump generates write and erase voltages that are rated to operate within the device's voltage range for byte operations.

Since the memory type is Flash, the erase operation only supports sector erase and does not support byte erase. Before modifying the data of an address, it is recommended to save the other data, erase the current sector, and then write the data.

The memory can be read/write/erase(R/W/E) through the memory module interface.

Flash memory 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	1	0	1	0	1	1	0

Bit7~Bit0 MLOCK<7:0>: Memory operation enable bit.
AAH: Allow memory-related R/W/E operations
00H/FFH/56H: Operation not allowed
Disable writing other values
(This register only supports write operation, read as 00H)

Flash memory data register MDATA

0xFE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MDATA	MDATA7	MDATA6	MDATA5	MDATA4	MDATA3	MDATA2	MDATA1	MDATA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	X	X	X	X	X	X	X	X

Bit7~Bit0 MDATA<7:0>: Data that is read or written to program memory

Flas 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>: The lower 8 bits of the memory read/write operation address.

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>: The higher 8 bits of the memory read/write operation address.

Flash memory control register MCTRL

0xFF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MCTRL	---	---	MERR	MREG	MMODE1	MMODE0	---	MSTART
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	1	0	0	0	0

Bit7~Bit6 ---

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;

1= Select data area (Low 10-bit address is valid);

0= Select program area (Low 14-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= Reserved;

00= Read mode.

Bit1 Unused, read as 0

Bit0 MSTART: Start operation control;

1= Start program memory R/W/E operation (Automatically cleared by hardware after the operation is completed);

0= Write: Terminate or not start program memory R/W/E operation;
Read: operation completed or operation not started.

When the flash memory is operated, the CPU is in the pause state, and when the operation is completed, the CPU continues to run the command.

6 NOP instructions must be added after the operation memory instruction:

```
MOV MCTRL,#09H          ; Write begins
NOP
NOP
NOP
NOP
NOP
NOP
MOV MCTRL,#01H          ; Read begins
NOP
NOP
NOP
NOP
NOP
NOP
```

19. UNIQUE ID (UID)

19.1 Overview

Each chip has a unique 96-digit identification number(Unique identification). It has been set at the factory and cannot be modified by the use.

19.2 UID Register

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>

20. TIMED WAKE UP IN SLEEP STATE

20.1 Timed Wakeup Control Register

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: Timed wake up function enable
1= Enable
0= Disable
Bit6 Unused
Bit5~Bit4 WUTPS<1:0>: Timed wake up counter clock divider.
00= F/1
01= F/8
10= F/32
11= F/256
Bit3~Bit0 WUTD<11:8>: Timed wake up time data high 4 bits

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	1	1	1	1	1	1	1	1

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

20.2 Principle Of Timed Wake Up

The internal timed wake-up principle is that after the system enters the sleep mode, the CPU and all peripheral circuits stop working, and the internal low-power oscillator LPRC starts to work, and its oscillation clock is 125KHz(T_{LSI}≈8us). Clocks are provided for the WUT (Wake Up Timer) counter.

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

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

- WUTEN=1: Enable the timed wakeup function;
- WUTEN=0: Disable the timed wakeup function;

WUTCRH[3:0] and WUTCRL[7:0] form a 12-bit timer wake-up data register with a reset value of 0FFH. After entering sleep mode, the WUT counter starts timing. When the value of the WUT counter is equal to the value of the timer wake-up data register, the system oscillator is started to enter the wake-up wait state.

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

21. COMPILE AND DEBUG (KEIL C51)

21.1 Entering Debug Mode

The chip online debugging mode occupies two ports and can be connected to Keil through the emulator for online debugging, before that, the Keil-C51 software and drivers must be installed.

The chip does not support emulation debugging by default, the emulation must set the DEBUG bit ENABLE in CONFIG through the special burner to support emulation debugging. (If set to the encrypted state PROTECT=ENABLE, you can not enter the debug mode).

The hardware connection sequence during debugging is as follows:

1. Connect the emulator and PC;
2. Connect the emulator to the target board;
3. Target board power;
4. Then click the DEBUG button in the keil.

The program code will be downloaded to the chip in debug mode. DSCK and DSDA are configured as debug ports and cannot be used as general purpose I/O.

21.2 Chip Status In Debug Mode

The emulator does not provide power supply, so you need to pay attention to the handling of power supply during emulation.

- In the debug state, the DSCK and DSDA ports are used as dedicated debug ports and cannot implement GPIO and multiplexing functions.
- In the debug state, enter the sleep mode/idle mode (STOP/IDLE), the system power and oscillator will not stop working, and the sleep wakeup function can be emulated in this state. If you need to pay attention to the power consumption, it is recommended to turn off the debug function and then test the actual sleep current of the chip.
- In debug state pause, other function peripherals continue to run, WDT, Timer0/1/2/3/4 counters will stop. If Timer1/4 is used as UART0/1's baud rate generator, Timer1/4 will also continue to run in stop state.

21.3 Compilation Instruction

21.3.1 Hybrid Compilation

Insert the settings of the assembly code in the C code. In the list of Keil files, right click on the C file with assembly code, then go to Option for File ‘main.c’ (if the file is main.c).In the window Properties, change the Generate Assembler SRC File and Assembler SRC File options from gray to black, and then compile.

22. ROGRAM DOWNLOAD

22.1 Connect The Debugger to Download

After compiling the program in Keil C51:

1. Set the chip configuration options in Option for Target-->Debug-->setting;
2. Set the encryption option in Option for Target-->Utilities-->setting;
3. Click Download (F8) to download the program to the chip.

22.2 Connect Dedicated Writer to Download

Use a dedicated burner to download.

23. ELECTRICAL PARAMETERS

23.1 Absolute Maximum Rating

Symbol	Item	Min	Max	Unit
T_{ST}	storage temperature	-55	150	℃
T_A	Operating temperature	-40	105	℃
V_{DD}-V_{SS}	Operating voltage	-0.3	4.8	V
V_{IN}	Input voltage	V_{SS}-0.3	V_{DD}+0.3	V
I_{DD}	V_{DD} maximum input current	-	120	mA
I_{SS}	V_{SS} maximum output current	-	120	mA
I_{IO}	Single IO maximum sink current	-	50	mA
	Single IO maximum output current	-	20	mA
	All IO maximum sink current	-	120	mA
	All IO maximum output current	-	120	mA

23.2 DC Electrical Characteristics

(V_{DD}-V_{SS}=2.1～4.5V, T_A=25℃)

Symbol	Item	Test condition	Min	Typ	Max	Unit
V_{DD}	Operating Voltage	HSI=8MHz~24MHz	2.1	-	4.5	V
I_{DD}	Normal mode	V_{DD}=4.5V, HSI=24MHz, All peripherals OFF	-	4	-	mA
		V_{DD}=3V, HSI=24MHz, All peripherals OFF	-	4	-	mA
		V_{DD}=4.5V, HSI=16MHz, All peripherals OFF	-	3	-	mA
		V_{DD}=3V, HSI=16MHz, All peripherals OFF	-	3	-	mA
		V_{DD}=4.5V, HSI=8MHz, All peripherals OFF	-	2	-	mA
		V_{DD}=3V, HSI=8MHz, All peripherals OFF	-	2	-	mA
	IDLE mode	V_{DD}=4.5V, HSI=24MHz, All peripherals OFF	-	2	-	mA
		V_{DD}=3V, HSI=24MHz, All peripherals OFF	-	2	-	mA
		V_{DD}=4.5V, HSI=16MHz, All peripherals OFF	-	1.5	-	mA
		V_{DD}=3V, HSI=16MHz, All peripherals OFF	-	1.5	-	mA
		V_{DD}=4.5V, HSI=8MHz, All peripherals OFF	-	1.1	-	mA
		V_{DD}=3V, HSI=8MHz, All peripherals OFF	-	1.1	-	mA
I_{SLEEP}	Sleep current	WDT=DIS, V_{LVR}=1.9V	-	6	-	uA
I_{LI}	Input leakage	-	-	-	0.1	uA
V_{IL}	Input low level	-	V_{SS}	-	0.3V_{DD}	V
V_{IH}	Input high level	-	0.7V_{DD}	-	V_{DD}	V
V_{OL}	Output low voltage	V_{DD}=4.5V, I_{OL}=9mA	-	-	0.4	V
		V_{DD}=3V, I_{OL}=7mA	-	-	0.4	V
V_{OH}	Output high voltage	V_{DD}=4.5V, I_{OH}=7mA	3.1	-	-	V
		V_{DD}=3V, I_{OH}=4mA	2.1	-	-	V
R_{PH}	Pull-up resistor	-	-	32	-	KΩ
R_{PL}	Pull-down resistor	-	-	32	-	KΩ

23.3 AC Electrical Parameters

23.3.1 Power-up and power-down operation

(T_A = 25℃, excluding 32.768K crystal oscillation start time)

Symbol	Item	Condition	Min	Typ	Max	Unit
T_{RESET}	Reset time	V_{DD}=4.5V	--	16	--	ms
T_{VDDR}	V_{DD} rise rate	V_{DD}=4.5V	2	--	∞	us/V
T_{VDDF}	V_{DD} fall rate	V_{DD}=4.5V	2	--	∞	us/V

23.3.2 External Oscillator

Symbol	Item	Condition	Min	Typ	Max	Unit
V_{HSE}	Operating Voltage	F=8/16MHz,C_{XT}=0-47pF	2.1	-	4.5	V
V_{LSE}	Operating Voltage	F=32.768KHz,C_{XT}=10-22pF	2.1	-	4.5	V

23.3.3 Internal Oscillator

V_{DD}=2.1V-4.5V

Symbol	Item	Test condition	Frequency Error	Min	Typ	Max	Unit
F_{HSI}	Internal high speed 8MHz	T_A=0℃ to 80℃	±1.5%	-	8.0	-	MHz
		T_A= -40℃ to 105℃	±2.5%	-	8.0	-	MHz
	Internal high speed 16MHz	T_A=0℃ to 80℃	±1%	-	16.0	-	MHz
		T_A= -40℃ to 105℃	±2%	-	16.0	-	MHz
	Internal high speed 24MHz	T_A=0℃ to 80℃	±1%	-	24.0	-	MHz
		T_A= -40℃ to 105℃	±1.5%	-	24.0	-	MHz

The chart below shows the HSI=8MHz/16MHz/24MHz offset trend at different temperatures.

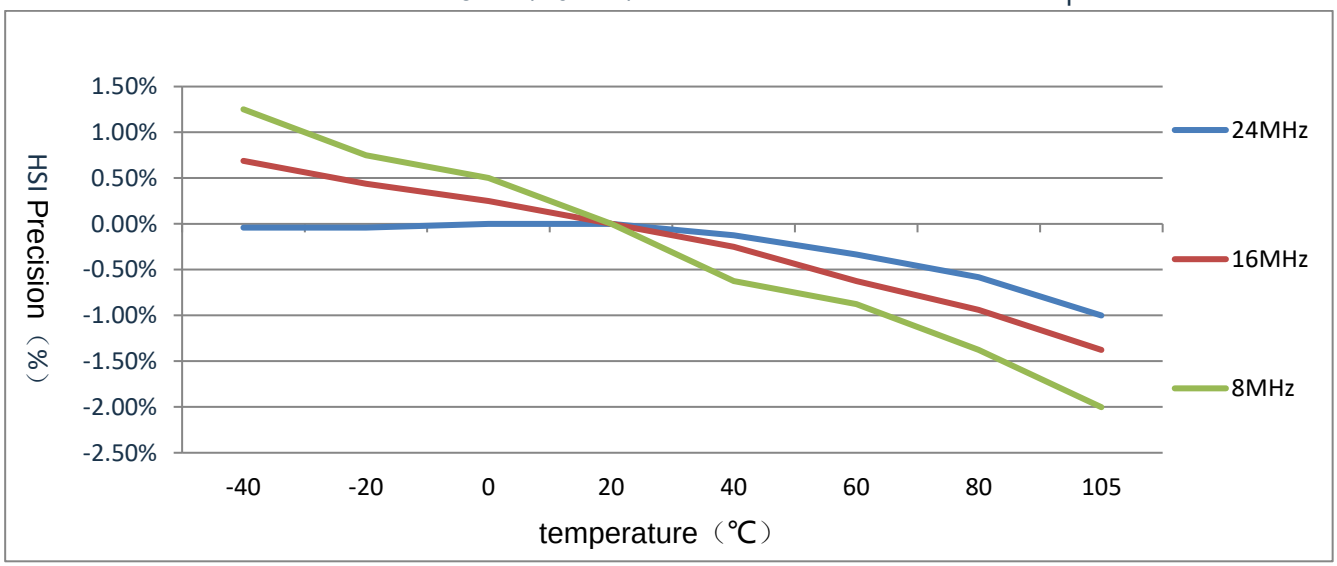


Figure 23-1: Precision versus temperature curve

23.4 Low Voltage Reset Electrical Parameters

Symbol	Item	Min	Typ	Max	Unit
V_{LVR1}	Low pressure detection threshold 1.9V	1.75	1.9	2.05	V
V_{LVR2}	Low pressure detection threshold 2.1V	1.95	2.1	2.25	V
V_{LVR3}	Low pressure detection threshold 2.6V	2.45	2.6	2.75	V
V_{LVR4}	Low pressure detection threshold 3.5V	3.35	3.5	3.65	V

23.5 ADC Electrical Characteristics

Symbol	Item	Min	Typ	Max	Unit
V_{AVDD}	ADC operating Voltage	3.0	-	4.5	V
V_{AIN}	Analog signal input	0	-	V_{AVDD}	V
N_R	Resolution		12		Bit
T_{ADCK}	ADC clock period	0.5	-	16	us
T_{ADC}	AD conversion time	-	18.5	-	T_{ADCK}
F_S	Sampling rate		100		Ksps

23.6 BANDGAP Electrical Characteristics

Symbol	Item	Test condition	Min	Typ	Max	Unit
V_{REF}	Internal reference 1.2V	V_{DD}=2.1-4.5V, T_A= -40℃ to 105℃	1.188	1.2	1.212	V

23.7 FLASH Electrical Characteristics

Symbol	Item	Test condition	Min	Typ	Max	Unit
V_F	Flash working voltage	-	2.1	-	4.5	V
T_F	Flash Operating temperature	-	-40	27	105	℃
$N_{ENDURANCE}$	Number of erases	-	20,000	-	-	Cycle
T_{RET}	Data retention time	25℃	100	-	-	year
T_{ERASE}	Sector erase time	-	-	4.5	-	ms
T_{PROG}	Programming time	-	-	7	-	us
I_{DD1}	Read current	-	-	-	2.5	mA
I_{DD2}	Programming current	-	-	-	3.6	mA
I_{DD3}	Erase current	-	-	-	2	mA

23.8 ESD Electrical Characteristics

Symbol	Item	Test condition	Rating
V_{ESD}	Electrostatic discharge (Human body discharge mode HBM)	T_A = 25℃ JEDEC EIA/JESD22- A114	3B
	Electrostatic discharge (Machine discharge mode MM)	T_A = 25℃ JEDEC EIA/JESD22- A115	C

23.9 Latch-Up Electrical Characteristics

Symbol	Item	Test condition	Test type
LU	Static latch-up class	JEDEC STANDARD NO.78D NOVEMBER 2011	Class I (T_A = 25℃)

24. Instruction

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.

24.1 Symbol Description

Symbol	Description
Rn	Working register R0-R7
Direct	Unit address of internal data memory RAM (00H-FFH) or address in special function register SFR
@Ri	Indirect Addressing Register (@R0 or @R1)
#data	8-bit binary constants
#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
Addr11	11-bit address, address range 0-2KB
Rel	Relative Address
A	Accumulator

24.2 Instruction Set

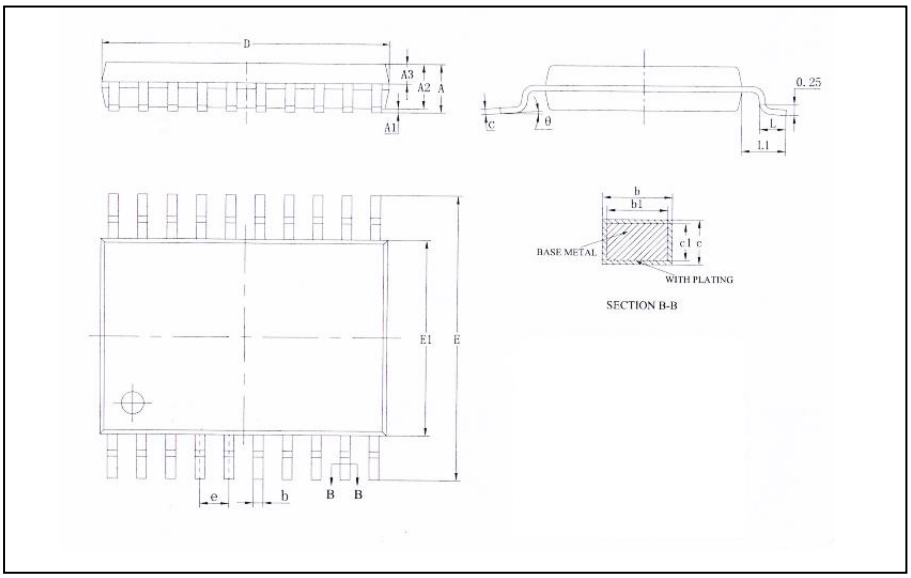
Instruction		Description
Operation type		
ADD	A,Rn	Add register to accumulator.
ADD	A,direct	Add directly addressed data to accumulator.
ADD	A,@Ri	Add indirectly addressed data to accumulator.
ADD	A,#data	Add immediate data to accumulator.
ADDC	A,Rn	Add register to accumulate or with carry.
ADDC	A,direct	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 in directly addressed data from accumulator with borrow.
SUBB	A,#data	Subtract immediate 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	Decrement indirectly addressed location.
MUL	AB	Multiply A and B.
DIV	AB	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.
CLR	A	Clear accumulator.
CPL	A	Complement accumulator.

Instruction		Description
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 dato to102n directly 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 fromstack.
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 bitto carry flag.
ANL	C,/bit	AND complement of directly addressed bit to carry.
ORL	C,bit	OR directly addressed bit to carry flag.
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		

Instruction		Description
ACALL	addr11	Absolute subroutine call
LCALL	addr16	Long subroutine call
RET		Return from subroutine
RETI		Return from interrupt
AJMP	addr11	Absolute jump
LJMP	addr16	Long jump
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 lacation 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)

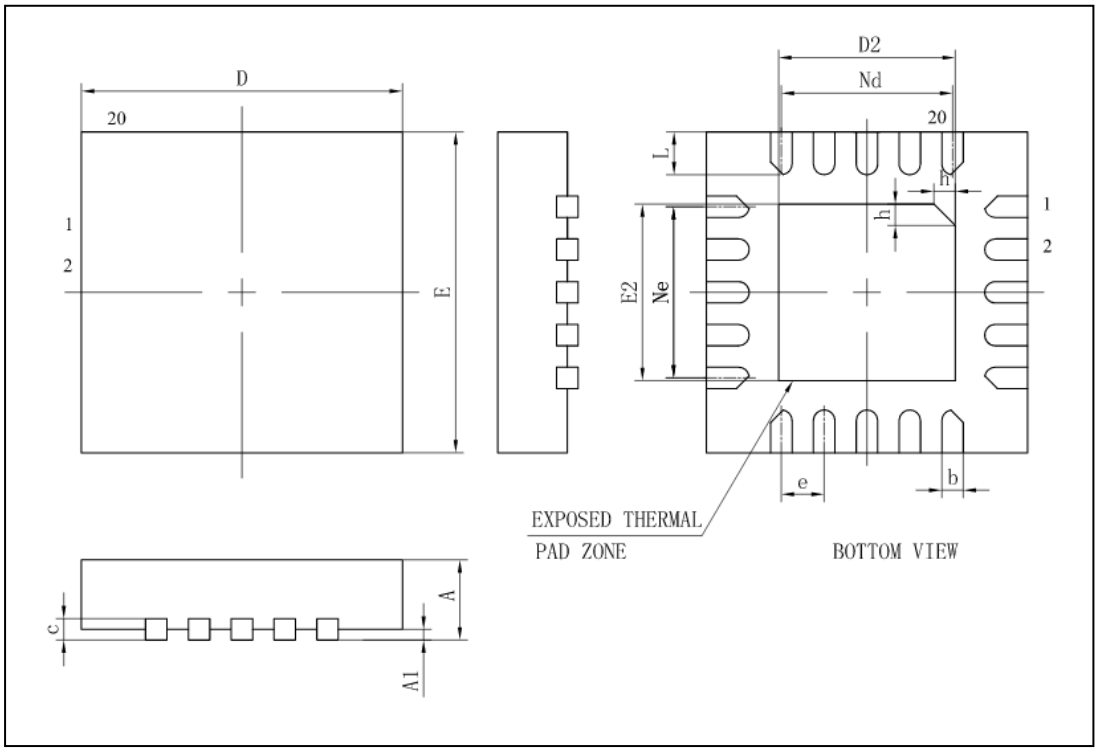
25. PACKAGE

25.1 TSSOP20



Symbol	Millimeter		
	Min	Nom	Max
A	-	-	1.20
A1	0.05	-	0.15
A2	0.80	1.00	1.05
A3	0.39	0.44	0.49
b	0.20	-	0.28
b1	0.19	0.22	0.25
c	0.13	-	0.17
c1	0.12	0.13	0.14
D	6.40	6.50	6.60
E1	4.30	4.40	4.50
E	6.20	6.40	6.60
e	0.65BSC		
L	0.45	0.60	0.75
L1	1.00REF		
θ	0	-	8°

25.2 QFN20



Symbol	Millimeter		
	Min	Nom	Max
A	0.70	0.75	0.80
A1	-	0.02	0.05
b	0.15	0.20	0.25
c	0.18	0.20	0.25
D	2.90	3.00	3.10
D2	1.55	1.65	1.75
e	0.40BSC		
Ne	1.60BSC		
Nd	1.60BSC		
E	2.90	3.00	3.10
E2	1.55	1.65	1.75
L	0.35	0.40	0.45
h	0.20	0.25	0.30

26. Revision History

Revision	Date	Modified content
V1.0	May 2019	Initial Version
V1.1	May 2019	Modify some register descriptions
V1.2	May 2019	Modify partial register format
V1.3	Jun 2019	Modify functional feature description
V1.31	Jul 2019	Add a section on interrupt flag clear operation
V1.32	Feb 2020	Modify functional feature description
V1.33	May 2021	Modify the description of the IIC register I2CMTP
V1.34	Aug 2021	Modify MLOCK reset value
V1.35	Nov 2021	Modify the contents of the flash memory operation section
V1.36	Apr 2022	Remove ADC clock frequency example, adjust I2C description, remove FLASH operation time related description, adjust port multiplexing function description, add BUZZER notes, change some register bit description, optimize some text expression
V1.3.7	Apr 2022	1) Correct 25.1 Packaging Dimensions 2) Optimize the content of 23.8 and 23.9 3) Optimize formatting