High-speed IT 8051-based Flash MCU, 512 bytes SRAM, 16 Kbytes Flash, 128 bytes independent EEPROM, eight 12-bit ADC, Seven 10-bit PWM Outputs, 3 Timer/Counters, UART, SSI, CheckSum module

1. GENERAL DESCRIPTION

SC92F8003 is integrated with 16 Kbytes Flash ROM,512 bytes SRAM, 128 bytes EEPROM, up to 18 GP I/Os, 16 IO external interrupters, three 16-bit timers, 8-channel 12-bit high-precision ADC, 7-channel independent 10-bit PWM, internal ±1% high-precision 16/8/4/1.33MHz high-frequency oscillator and 4% precision low-frequency 128K oscillator, UART, SSI. To improve the reliability and simplify the circuit design, SC92F8003 is also built in with 4-level optional LVR voltage, 2.4V ADC reference voltage ,WDT and other high-reliability circuit.

SC92F8003 is a High-density performance line 1T 8051-based industrial Flash microcontrollers(MCUs), in which the instruction system is completely compatible with standard 8051 product series.

SC92F8003 features excellent anti-interference performance, It is very suitable for industrial control and consumer applications such as IoT control, smart home appliances and smart homes, chargers, power supplies, model airplanes, walkie-talkies, wireless communications, and gaming consoles.

2. FEATURES

Operating Voltage: 2.4V ~ 5.5V

Operating Temperature: -40 ~ 85℃

Package:

SC92F8003Q20R（QFN20）

SC92F8003X20U（TSSOP20）

Core: high-speed 1T 8051

Flash ROM: 16 Kbytes Flash ROM (MOVC prohibited addressing 0000H~00FFH) can be rewritten for 10,000 times

IAP: Code option into 0K, 0.5K, 1K or 8K

EEPROM:Independent 128 bytes of EEPROM can be written repeatedly 100000 times. The data written-in has more than 10-year preservation life.

SRAM: Internal 256 bytes + external 256 bytes

System Clock (f_{SYS}):

- Built-in high-frequency 16MHz oscillator (f_{HRC}):
  - As the system clock source, f_{SYS} can be set to 16/8/4/1.33MHz by programmer selection.
  - Frequency Error: Suitable for 2.9V ~ 5.5V and -20 ~ 85℃ application environment, no more than ±1% of frequency deviation.
- Built-in high-frequency Crystal Oscillator Circuit
  - Enabled external 2-16 MHz Oscillator
  - As the system clock source, f_{SYS} can be set as /1, /2, /4 or /12 of the frequency of oscillator via code option
- Operating voltage at different system frequency
  - >12MHz @2.9~5.5V
  - ≤12MHz @2.4~5.5V

Built-in Low-frequency 128kHz LRC Oscillator:

- Clock soruce of Base Timer (BTM), which AS the Base Timer (BTM) clock soruce, it can wake up stop mode
- Clock soruce of watchdog (WDT)
- Frequency deviation: 4.0V ~ 5.5V and -20 ~ 85℃ application environment, no more than ±4% of frequency error

Low-voltage Reset (LVR):

- 4-level LVR voltage options: 4.3V, 3.7V, 2.9V, 2.3V
- The default is the Code Option value selected by the user.

Flash Programming and Emulation interface:

- 2-wire programming interface

INTERRUPT:

- Totally 11 interrupt sources, including Timer0, Timer1, Timer2, INT0,INT1,INT2, ADC, PWM, UART, SSI and Base Timer
- 3 external interrupt vectors shared by 16 external interrupt I/Os, which can be defined in rising-edge, falling-edge or dule-edge trigger mode.
- Two levels of interrupt priority can be set.

Digital Peripheral:

- Up to 18 two-way independently controllable I/O interfaces, able to configure where pull-up resistor can be configured independently

- All IOs equipped with sink current drive capability (70mA)
- 11-bit WDT, optional clock division ratio
- 3 standard 80C51 timers: Timer0, Timer1 and Timer 2
- Seven 10-bit PWM output channels with variable period and individual duty cycle, where PWM0~5 can be divided into three sets of output complementary PWM signals with dead zone
- 1 independent UART communication port (switchable IO port)
- 1 three-in-one serial communication interface (SSI) (switchable IO port)

Analog Peripheral:

- 8-channel 12-bit ±2LSB ADC
  - Built-in 2.4V reference voltage
  - 2 options for ADC reference voltage : V_{DD} and internal 2.4V
  - Internal one-channel ADC, which is able to directly measure V_{DD} where VDD can be measured directly
  - ADC conversion completion interrupt

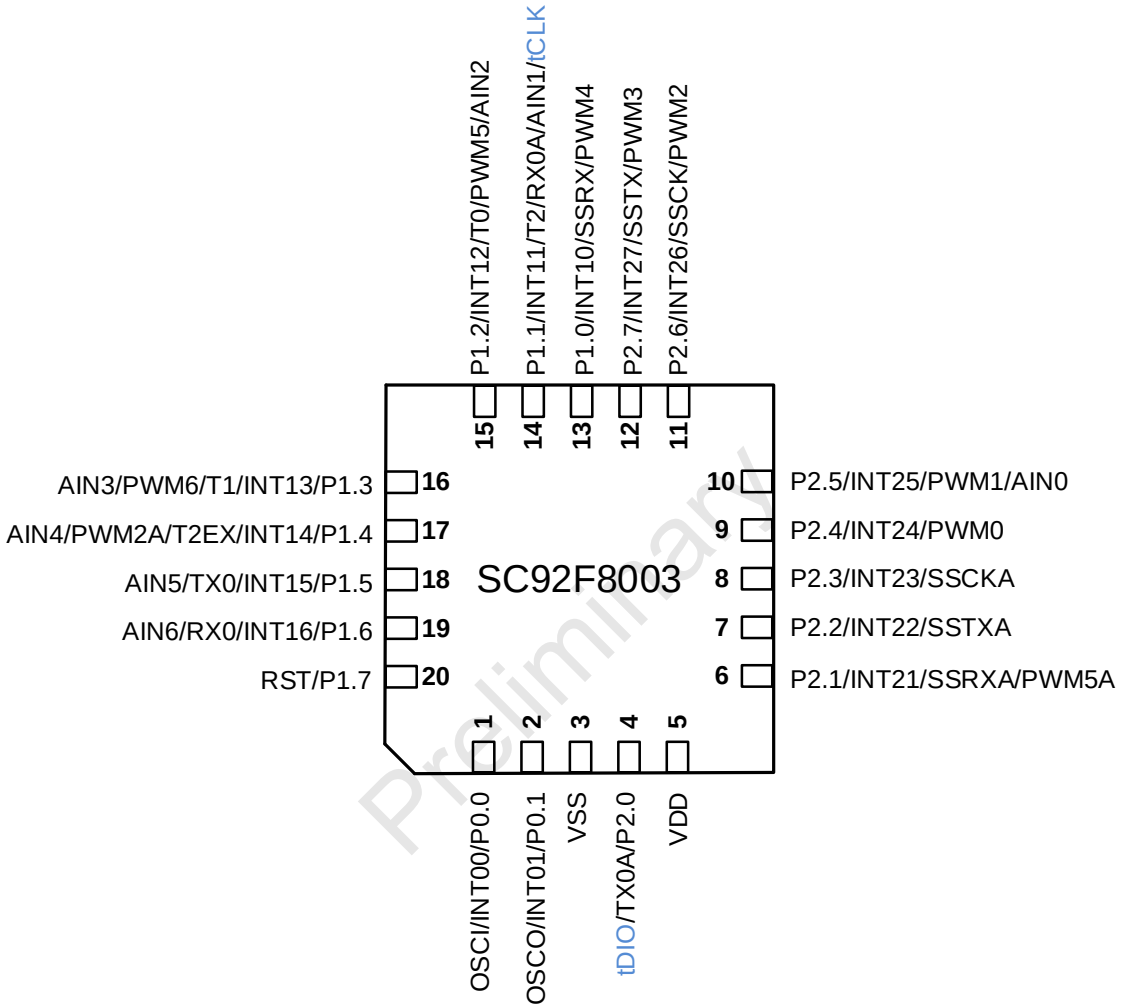
Power Saving Mode:

- IDLE Mode can be woken up by any interrupt.
- STOP Mode, able to be woken up by INT0,INT1,INT2,and Base Timer .

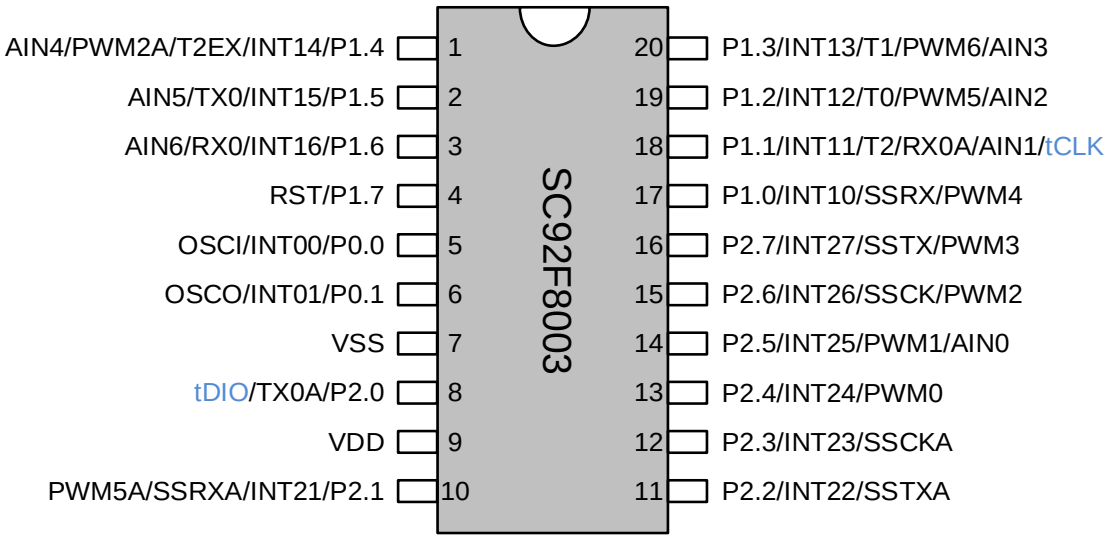
CONTENTS

3. PIN DESCRIPTION

3.1 PIN CONFIGURATION



SC92F8003Q20R Pin Deployment Diagram （QFN20）



SC92F8003X20U Pin Deployment Diagram （TSSOP20）

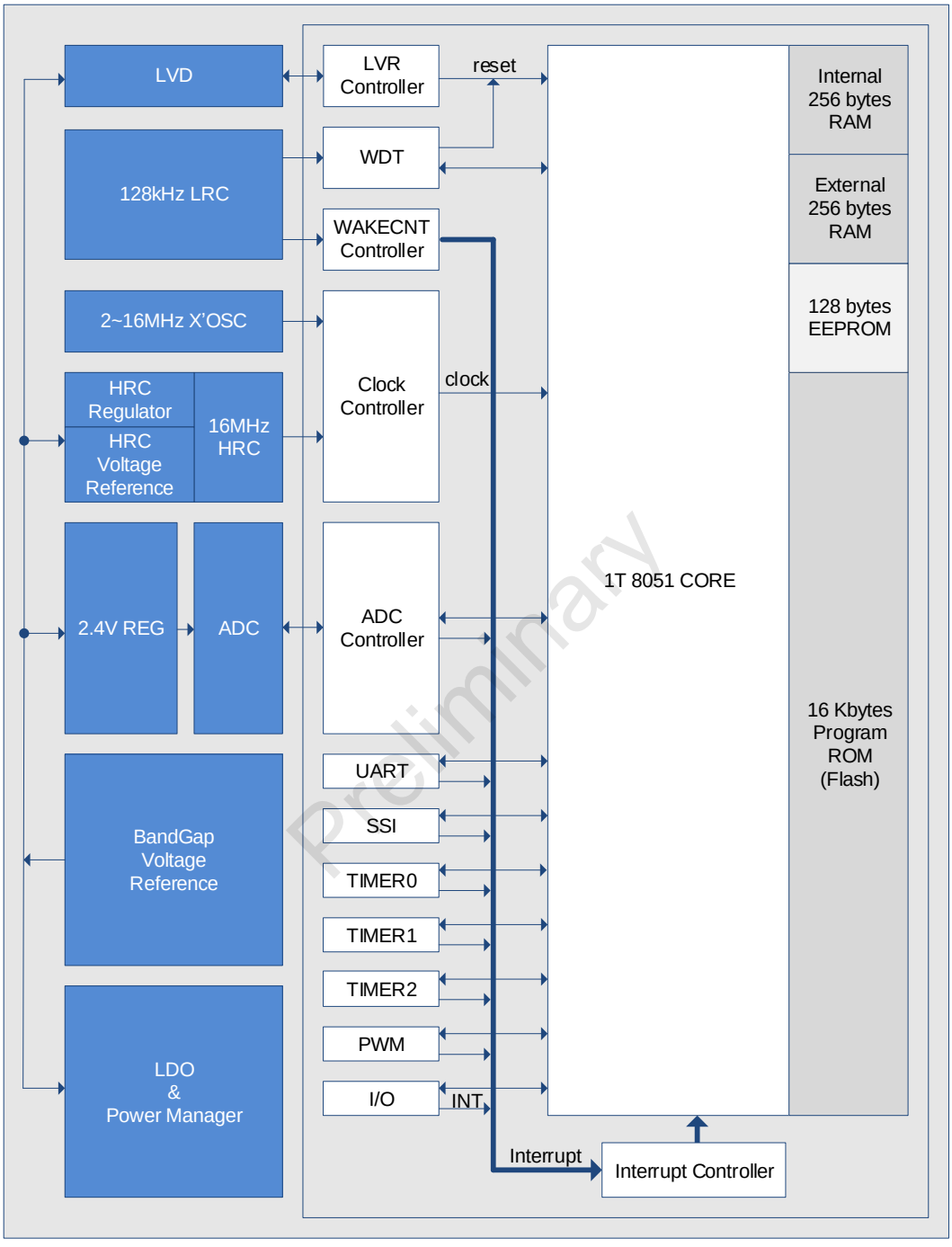
3.2 PIN DEFINITION

PIN Number		PIN Name	PIN Type	Function Description
TSSOP20	QFN20			
1	17	**P1.4/INT14/T2EX/PWM2A/AIN4**	I/O	P1.4: GPIO P1.4 INT14: External Interrupt 14 T2EX: External Capture Signal Input for Timer2 PWM2A: PWM2A output PWM2A:One of the PWM2 output ports AIN4: ADC Input Channel 4
2	18	**P1.5/INT15/TX0/AIN5**	I/O	P1.5: GPIO P1.5 INT15: External Interrupt 15 TX0: UART 0 Transmitter Output TX0: one of the UART 0 TX Transmitter Output AIN5: ADC Input Channel 5
3	19	**P1.6/INT16/RX0/AIN6**	I/O	P1.6: GPIO P1.6 INT16: External Interrupt 16 RX0: UART 0 Receiver Input AIN6: ADC Input Channel 6
4	20	**P1.7/RST**	I/O	P1.7: GPIO P1.7 RST: External Reset input
5	1	**P0.0/INT00/OSCI**	I/O	P0.0: GPIO P0.0 INT00: External Interrupt 00 OSCI: External crystal oscillator input pin
6	2	**P0.1/INT01/OSCO**	I/O	P0.1: GPIO P0.1 INT01: External Interrupt 01 OSCO: External crystal oscillator output pin
7	3	**VSS**	Ground	Ground
8	4	**P2.0/TX0A/tDIO**	I/O	P2.0: GPIO P2.0 TX0A: UART 0 Transmitter Output

				tDIO:Programming and Emulation Data Pin line
9	**5**	**VDD**	Power	Power
10	**6**	**P2.1/INT21/SSRXA/PWM5A**	I/O	P2.1: GPIO P2.1 INT21: External Interrupt 21 SSRXA: One of the SSRXs of the SSI communication interfaces: SPI MISO, UART RX in SSI communication interfaces PWM5A:One of the PWM5 output ports
11	**7**	**P2.2/INT22/SSTXA**	I/O	P2.2: GPIO P2.2 INT22: External Interrupt 22 SSTXA: One of the SSTXs of the SSI communication interfaces: SPI MISO, UART TX and TWI SDA of the SSI communication interfaces
12	**8**	**P2.3/INT23/SSCKA**	I/O	P2.3: GPIO P2.3 INT23: External Interrupt 23 SSCKA: One of the SSCKs of the SSI communication interfaces: SPI SCK and TWI SCL of the SSI communication Pin
13	**9**	**P2.4/INT24/PWM0**	I/O	P2.4: GPIO P2.4 INT24: External Interrupt 24 PWM0: PWM0 output interfaces
14	**10**	**P2.5/INT25/PWM1/AIN0**	I/O	P2.5: GPIO P2.5 INT25: External Interrupt 25 PWM1: PWM1 output interfaces AIN0: ADC Input Channel 0
15	**11**	**P2.6/INT26/SSCK/PWM2**	I/O	P2.6 : GPIO P2.6 INT26: External Interrupt 26 SSCK: One of the SSCKs of the SSI communication interfaces: SPI SCK and TWI SCL of the SSI communication interfaces PWM2:One of the PWM2 output interfaces
16	**12**	**P2.7/INT27/SSTX/PWM3**	I/O	P2.7: GPIO P2.7 INT27: External Interrupt 27 SSTX: One of the SSTXs of the SSI communication Pin: SPI MOSI , UART TX and TWI SDA of the SSI communication Pin PWM3: PWM3 output interfaces
17	**13**	**P1.0/INT10/SSRX/PWM4**	I/O	P1.0: GPIO P1.0 INT10: External Interrupt 10 SSRX: One of the SSRXs of the SSI communication Pin: SPI MOSI and UART RX of the SSI communication Pin PWM4: PWM4 output interfaces
18	**14**	**P1.1/INT11/T2/RX0A/AIN1/tCK**	I/O	P1.1: GPIO P1.1 INT11: External Interrupt 11 T2: Counter 2 external input or output interfaces RX0A: One of UART0 RX AIN1: ADC Input Channel 1

				TCK: Programming and Emulation Clock line
19	**15**	**P1.2/INT12/T0/PWM5/AIN2**	I/O	P1.2: GPIO P1.2 INT12: External Interrupt 12 T0: Timer/Counter 0 External input interfaces PWM5: PWM5 output PWM5:One of the PWM5 output ports AIN2: ADC Input Channel 2
20	**16**	**P1.3/INT13/T1/PWM6/AIN3**	I/O	P1.3: GPIO P1.3 INT13: External Interrupt 13 T1: Timer/Counter 1 External input interfaces PWM6: PWM6 output interfaces AIN3: ADC Input Channel 3

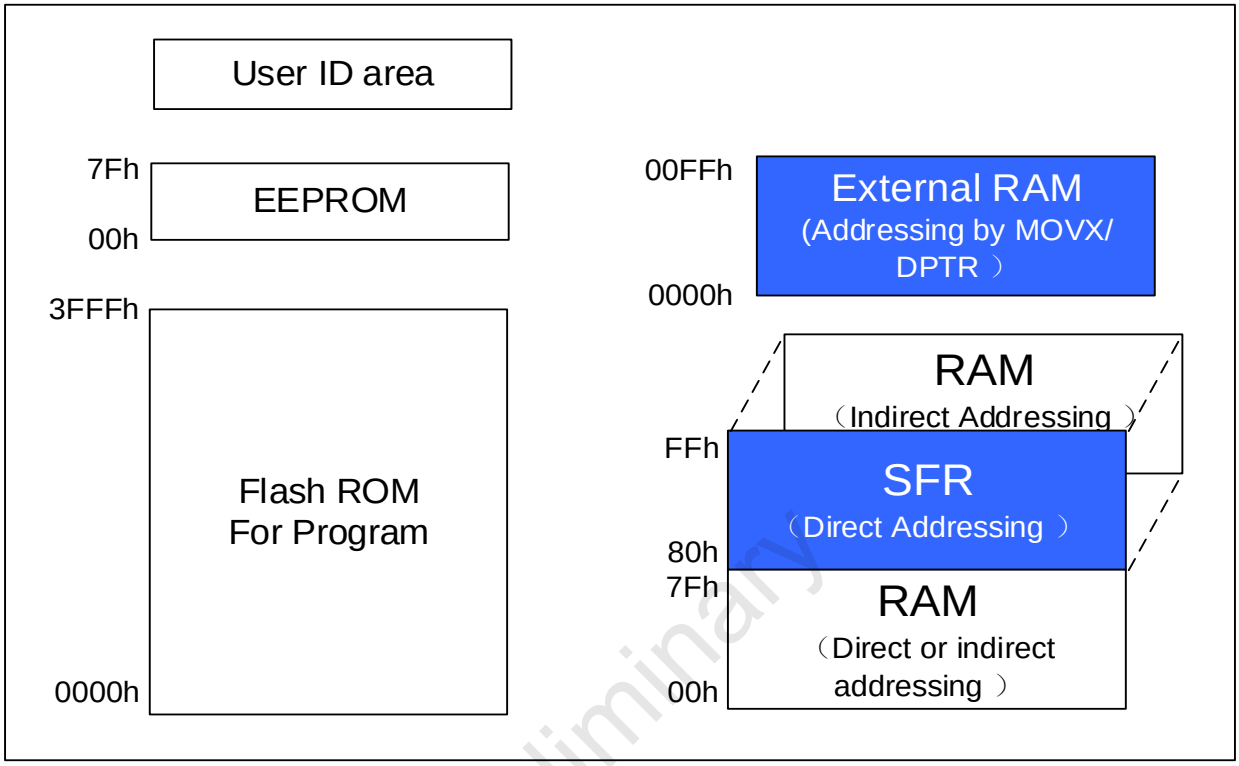
4. INNER BLOCK DIAGRAM



SC92F8003 BLOCK DIAGRAM

5. FLASH ROM AND SRAM STRUCTURE

The structures of SC92F8003's Flash ROM and SRAM are shown as follows:



Flash ROM and SRAM Structure Diagram

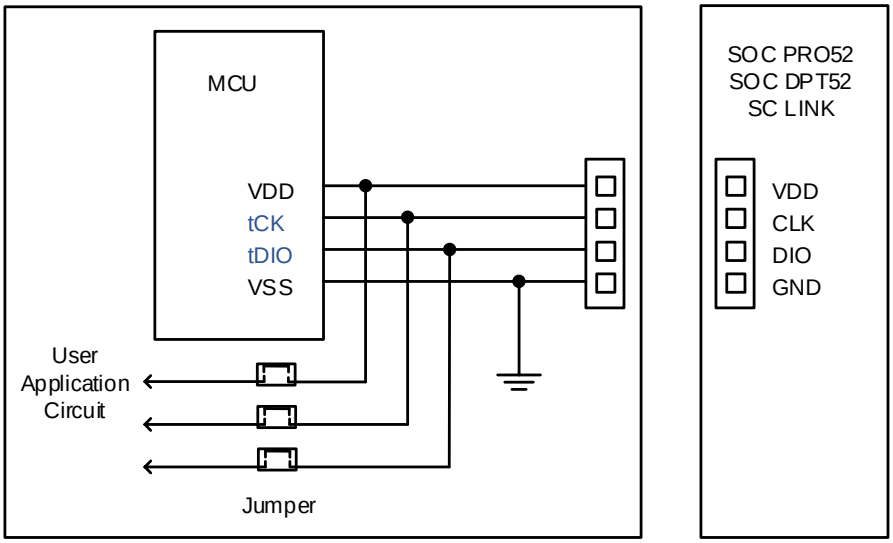
5.1 FLASH ROM

The SC92F8003 provides 16K Bytes of Flash ROM with the ROM address of 0000H ~ 3FFFH. These 16 Kbytes of Flash ROM can be rewritten 10,000 times, which is able to programme and erase by specialized ICP programming device (SOC PRO52/DPT52/SC LINK) provided by SinOne. MOVC instruction is non-addressable within 256 bytes (address of 0000H ~ 00FFH).

EEPROM is an area separated from 162 Kbytes ROM with the address of 00H ~ 7FH, which can be accessed by single-byte reading and writing operations in the programme; for more details, refer to 17 EEPROM and IAP operations .

User ID area: Write user ID upon shipment, users have read-only access, for more details, refer to 17 EEPROM and IAP operations. SC92F8003 16 Kbytes Flash ROM provides Blank Check, PROGRAM, VERIFY and ERASE function other than Read function. This Flash ROM and EEPROM usually needs no ERASE operation before writing.

SC92F8003 Flash ROM can be programmed by tDIO, tCK, VDD and VSS, with its specific connection shown as follows:



ICP Mode Flash Writer Programming Connection Diagram

5.2 CUSTOMER OPTION AREA (USER PROGRAMMING SETTING)

A seperate Flash area is built inside SC92F8003, called Code Option area,to save customers' presets. These presets will be written into IC when programming and loaded into SFR as initial values during reset.

Option-related SFR Operating Instructions:

Reading and writing operations of option-related SFR are controlled by both register OPINX and register OPREG, with its respective address of Option SFR depending on register OPINX, as shown below:

OPINX Address	OPREG										POR
	Symbol	Instruction	7	6	5	4	3	2	1	0	
83H@FFH	OP_HRCR	High frequency RC oscillation frequency adjustment (fine adjustment)	OP_HRCR[7:0]								nnnnnnnnb
C1H@FFH	OP_CTM0	Customer Option Register 0	ENWDT	ENXTL	SCLKS[1:0]		DISRST	DISLVR	LVRS[1:0]		nnnnnnnnb
C2H@FFH	OP_CTM1	Customer Option Register 1	VREFS	XTLHF	-	-	IAPS[1:0]		-	-	nnxxnnxxb

OP_HRCR (83H@FFH) System Clock Change Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbol	OP_HRCR[7:0]							
R/W	R/W							
POR	n	n	n	n	n	n	n	n

Bit Number	Bit Symbol	Instruction
7~0	**OP_HRCR[7:0]**	Internal high-frequency RC frequency adjustment Central value 10000000b corresponds to HRC central frequency, the larger the value is ,the faster the frequency will be, vice verse.

OP_CTM0 (C1H@FFH) Customer Option Register0 (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbol	ENWDT	ENXTL	SCLKS[1:0]		DISRST	DISLVR	LVRS[1:0]	
R/W	R/W	R/W	R/W		R/W	R/W	R/W	
POR	n	n	n		n	n	n	

Bit Number	Bit Symbol	Instruction
7	**ENWDT**	WDT control bit:（This bit is transferred by the system to the value set by the user Code Option） 0: WDT invalid 1: WDT valid (WDT stops counting during IAP execution)
6	**ENXTL**	External High frequency crystal oscillator selection bit: 0: External High frequency crystal Interface disable, P0.0 and P0.1 Valid 1: External High frequency crystal Interface enable, P0.0 and P0.1 invalid
5~4	**SCLKS[1:0]**	System clock frequency selection bits: 00: System clock frequency is HRC frequency divided by 1; 01: System clock frequency is HRC frequency divided by 2; 10: System clock frequency is HRC frequency divided by 4; 11: System clock frequency is HRC frequency divided by 12;
3	**DISRST**	IO/RST selection bit: 0: configure P1.7 as External Reset input pin 1: configure P1.7 as General Purpose I/O
2	**DISLVR**	LVR control bit: 0: LVR valid 1: LVR invalid
1~0	**LVRS [1:0]**	LVR voltage selection bits: 11: 4.3V reset 10: 3.7V reset 01: 2.9V reset 00: 2.3V reset

OP_CTM1 (C2H@FFH) Customer Option Register1 (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbol	VREFS	XTLHF	-	-	IAPS[1:0]		-	-
R/W	R/W	R/W	-	-	R/W	R/W	-	-
POR	n	n	x	x	n	n	x	x

Bit Number	Bit Symbol	Instruction
7	**VREFS**	**Reference voltage selection (initial value is loaded from Code Option, user can modify such configuration)** 0: Configure ADC VREF as V_{DD} 1: Configure ADC VREF as internally accurate 2.4V
6	**XTLHF**	External clock source frequency **mode register** Set to configure external clock source frequency ≥12M Clear to configure external clock source frequency <12M
3~2	**IAPS[1:0]**	IAP Spatial Range Selection 00: Code data memory prohibits IAP operations, only EEPROM data memory is used for data storage 01: last 0.5k code zone allows IAP operation (3E00H ~ 3FFFH) 10: Last 1k code zone allows IAP operation (3C00H ~ 3FFFH)

		11: All code zone allows IAP operation (0000H ~ 3FFFH)

5.2.1 OPTION-RELATED SFR INSTRUCTIONS

Option-related SFR can be accessed by both OPINX and OPREG registers addresses, with its respective position of Option SFR depending on OPINX and its value written to option-related SFR depending on register OPREG:

Symbol	Address	Instruction		POR
OPINX	FEH	Option Pointer	OPINX[7:0]	00000000b
OPREG	FFH	Option Register	OPREG[7:0]	nnnnnnnnb

When operating Option-related SFR, register OPINX stores the address of option-related registers and register OPREG stores corresponding value.

For example: To configure OP_HRCR as 0x01

C program example:

```
OPINX = 0x83;          //Write OP_HRCR address into OPINX register
OPREG = 0x01;          //Write OPREG register into 0x01 (the value to be written into OP_HRCR register)
```

assembler program example:

```
MOV OPINX,#83H         ;Write OP_HRCR address into OPINX register
MOV OPREG,#01H         ; Write 0x01 into OPREG register (the value to be written into OP_HRCR register)
```

Note: It is forbidden to write any value beyond SFR address of Customer Option region into OPINX register! Or else, it may cause abnormal system operation!

5.3 SRAM

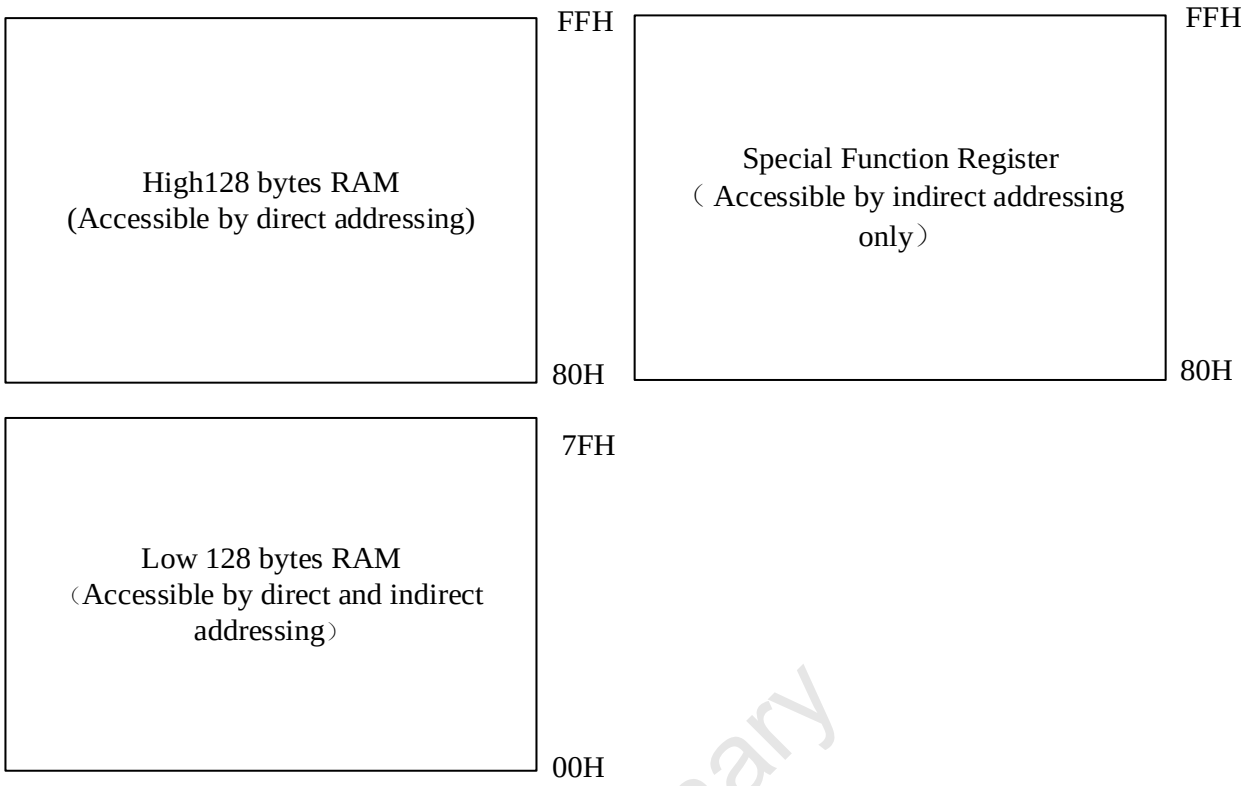
SC92F80033 Microcontroller Unit integrates SRAM of 512 bytes is divided into internal RAM and external RAM. The address of Internal RAM from 00H to FFH, including high 128 bytes (address of from80H to FFH) only addressed indirectly and low 128 bytes (address of from 00H to 7FH) addressed both directly and indirectly).

The address of special function register SFR is also from 80H to FFH. But the difference between SFR and internal high 128 bytes SRAM is that the former is Accessible by direct addressing but the latter Accessible by indirect addressing only.

The address of External RAM from 0000H to 02FFH, which is to be addressed by MOVX instruction.

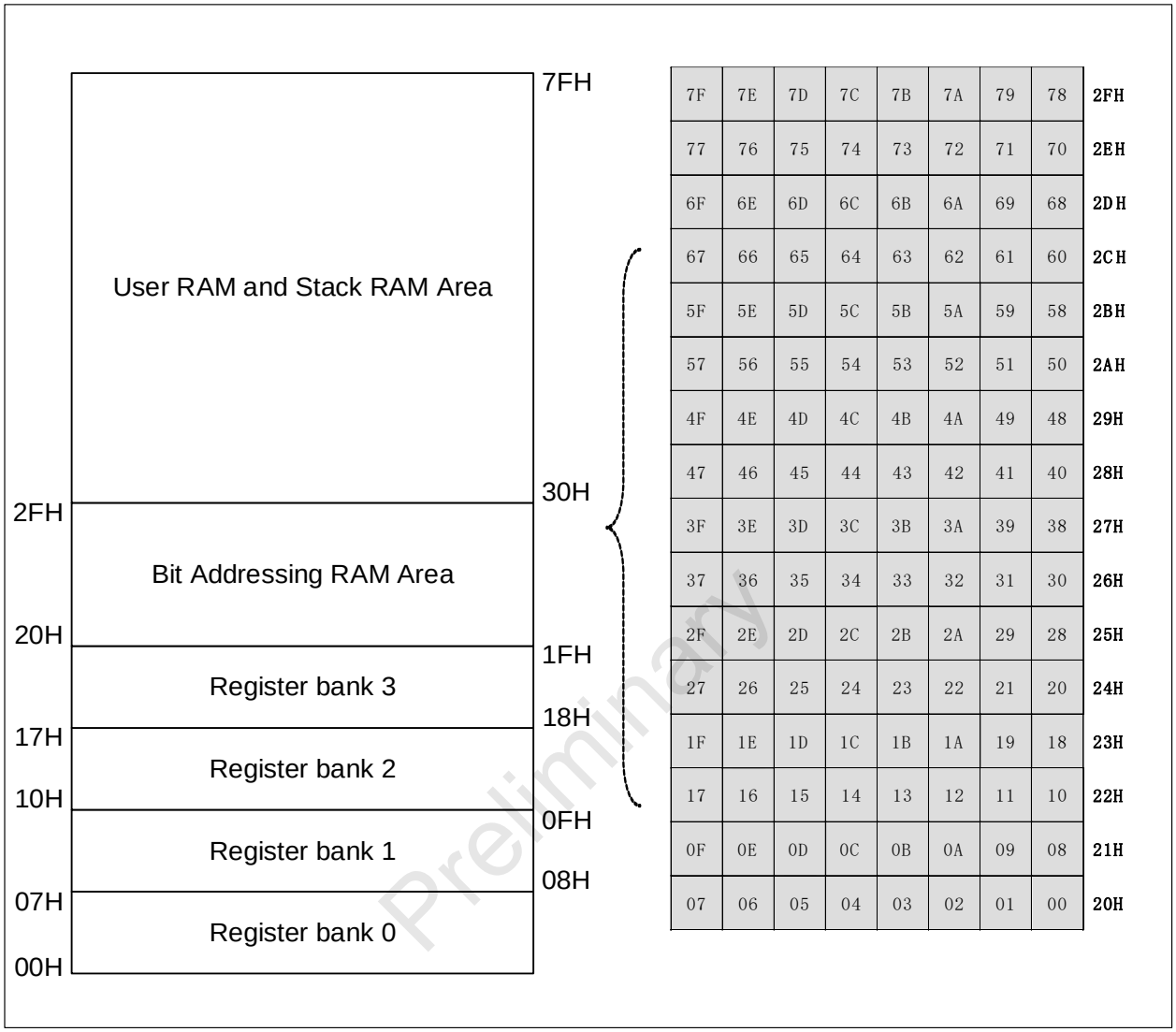
5.3.1 INTERNAL 256 BYTES SRAM

Internal low 128 bytes SRAM area is divided into three parts: ①Register bank 0~3, address from 00H to 1FH. the active bank is selected by the control bits RS1 and RS0 in the PSW register.Using Register bank 0~3 can accelerate arithmetic speed;② Bit addressing area , from 20H to 2FH; user can use it as normal RAM or bitswise addressing RAM; for the latter, the bit address is from 00H to 7FH, (bitswise addressing, different from normal SRAM byte-oriented addressing), which can be distinguished by instructions in programme; ③ User RAM and stack area;after resetting SC92F8003 , 8-bit stack pointer will point to stack area; in general, user can set initial value in initializer, which is recommended to configure in the unit interval from E0H to FFH.



Internal 256 bytes RAM Structure Diagram

Internal low 128 bytes RAM structure is shown below:



SRAM Structure Diagram

5.3.2 EXTERNAL 256 BYTES SRAM

External 256 bytes SRAM supports traditional access to external RAM methods. Use MOVX A, @Ri or MOVX @Ri, A to access external 256 bytes of RAM; you can also use MOVX A, @DPTR or MOVX @DPTR,A to access external 256 bytes of RAM.

6 SPECIAL FUNCTION REGISTER (SFR)

6.1 SFR MAPPING

The SC92F8003 provides some registers equipped with special functions, called SFR. The address of such SFRs is from 80H to FFH, some are bit-addressable, and others are not.It is very convenient for these bit addressable registers to change the value of single bit, of which the address is end up with figure “0” or “8”. All SFR special function registers shall use direct addressing for addressing.

The name and address of SC92F8003 special function registers are shown in the table below:

	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
F8h	-	-	-	-	CHKSUML	CHKSUMH	OPINX	OPREG
F0h	B	IAPKEY	IAPADL	IAPADH	IAPADE	IAPDAT	IAPCTL	-
E8h	-	-	-	-	-	-	-	OPERCON
E0h	ACC		-	-	-	-	-	-
D8h	-	-	PWMCON1	PWMDTYB	PWMDTY3	PWMDTY4	PWMDTY5	PWMDTY6
D0h	PSW	PWMCFG	PWMCON0	PWMPRD	PWMDTYA	PWMDTY0	PWMDTY1	PWMDTY2
C8h	T2CON	T2MOD	RCAP2L	RCAP2H	TL2	TH2	BTMCON	WDTCON
C0h	-	-	-	-	-	-	INT2F	INT2R
B8h	IP	IP1	INT0F	INT0R	INT1F	INT1R	-	-
B0h	-	-	-	-	-	-	-	-
A8h	IE	IE1	ADCCFG1	ADCCFG0	-	ADCCON	ADCVL	ADCVH
A0h	P2	P2CON	P2PH	-	-	-	-	-
98h	SCON	SBUF	P0CON	P0PH	-	SSCON0	SSCON1	SSDAT
90h	P1	P1CON	P1PH	-	-	SSCON2	-	-
88h	TCON	TMOD	TL0	TL1	TH0	TH1	TMCON	OTCON
80h	P0	SP	DPL	DPH	-	-	-	PCON
	Addressable addressing	Unaddressable						

Instructions:

1. Hollow space of SFR register refers to the fact that there is no such register RAM, it is not recommended for user to use.

6.2 SFR INSTRUCTIONS

For a description of each SFR, see the following table:

Bit Symbol	Address	Instruction	7	6	5	4	3	2	1	0	POR
P0	80H	P0 Data Register	-	-	-	-	-	-	P01	P00	xxxxxx00b
SP	81H	Stack Pointer	SP[7:0]								00000111b
DPL	82H	Data Pointer Low byte	DPL[7:0]								00000000b
DPH	83H	Data Pointer High byte	DPH[7:0]								00000000b
PCON	87H	Power Management Control Register	SMOD	-	-	-	-	-	STOP	IDL	0xxxxx00b
TCON	88H	Timer Control Register	TF1	TR1	TF0	TR0	IE1	-	IE0	-	00000x0xb
TMOD	89H	Timer Operating Mode Register	-	C/T1	M11	M01	-	C/T0	M10	M00	x000x000b
TL0	8AH	Timer 0 Low 8 bits	TL0[7:0]								00000000b
TL1	8BH	Timer 1 Low 8 bits	TL1[7:0]								00000000b
TH0	8CH	Timer 0 High 8 bits	TH0[7:0]								00000000b
TH1	8DH	Timer 1 High 8 bits	TH1[7:0]								00000000b
TMCON	8EH	Timer Frequency Control Register	-	-	-	-	-	T2FD	T1FD	T0FD	xxxxx000b
OTCON	8FH	Output Control Register	SSMOD[1:0]		SPOS[1:0]		-	-	-	-	0000xxxxb
P1	90H	P1 Data Register	P17	P16	P15	P14	P13	P12	P11	P10	00000000b
P1CON	91H	P1 Input/Output Control Register	P1C7	P1C6	P1C5	P1C4	P1C3	P1C2	P1C1	P1C0	00000000b
P1PH	92H	P1 Pull-up Resistor Control Register	P1H7	P1H6	P1H5	P1H4	P1H3	P1H2	P1H1	P1H0	00000000b
SSCON2	95H	SSI Control Register 2	SSC2[7:0]								00000000b
SCON	98H	Serial Port Control Register	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	00000000b
SBUF	99H	Serial Port Data Cache Register	SBUF[7:0]								00000000b
P0CON	9AH	P0 Input/Output Control Register	-	-	-	-	-	-	P0C1	P0C0	xxxxxx00b
P0PH	9BH	P0 Pull-up Resistor Control Register	-	-	-	-	-	-	P0H1	P0H0	xxxxxx00b
SSCON0	9DH	SSI Control Register 0	SSCON0[7:0]								00000000b
SSCON1	9EH	SSI Control Register 1	SSCON1[7:0]								00000000b
SSDAT	9FH	SSI Data Register	SSD[7:0]								00000000b
P2	A0H	P2 Data Register	P27	P26	P25	P24	P23	P22	P21	P20	00000000b
P2CON	A1H	P2 Input/Output Control Register	P2C7	P2C6	P2C5	P2C4	P2C3	P2C2	P2C1	P2C0	00000000b
P2PH	A2H	P2 Pull-up Resistor Control Register	P2H7	P2H6	P2H5	P2H4	P2H3	P2H2	P2H1	P2H0	00000000b
IE	A8H	Interrupt Enable Register	EA	EADC	ET2	EUART	ET1	EINT1	ET0	EINT0	00000000b
IE1	A9H	Interrupt Enable Register 1	-	-	-	-	EINT2	EBTM	EPWM	ESSI	xxxx0000b
ADCCFG1	AAH	ADC Configuration Register 2	-	-	-	-	LOWSP	ADCCK[2:0]			xxxx0000b
ADCCFG0	ABH	ADC Configuration Register 0	-	EAIN6	EAIN5	EAIN4	EAIN3	EAIN2	EAIN1	EAIN0	x0000000b
ADCCON	ADH	ADC Control Register	ADCEN	ADCS	EOC/ ADCIF	ADCIS[4:0]					00000000b
ADCVL	AEH	ADC Result Register	ADCV[3:0]				-	-	-	-	0000xxxxb
ADCVH	AFH	ADC Result Register	ADCV[11:4]								00000000b
IP	B8H	Interrupt Priority Control Register	-	IPADC	IPT2	IPUART	IPT1	IPINT1	IPT0	IPINT0	x0000000b
IP1	B9H	Interrupt Priority Control Register 1	-	-	-	-	IPINT2	IPBTM	IPPWM	IPSSI	xxxx0000b
INT0F	BAH	INT0 Falling Edge Interrupt Control Register	-	-	-	-	-	-	INT0F1	INT0F0	xxxxxx00b

INT0R	BBH	INT0 Rising Edge Interrupt Control Register	-	-	-	-	-	-	INT0R1	INT0R0	xxxxxx00b
INT1F	BCH	INT1 Falling Edge Interrupt Control Register	-	INT1F6	INT1F5	INT1F4	INT1F3	INT1F2	INT1F1	INT1F0	x0000000b
INT1R	BDH	INT1 Rising Edge Interrupt Control Register	-	INT1R6	INT1R5	INT1R4	INT1R3	INT1R2	INT1R1	INT1R0	x0000000b
INT2F	C6H	INT2 Falling Edge Interrupt Control Register	INT2F7	INT2F6	INT2F5	INT2F4	INT2F3	INT2F2	INT2F1	-	0000000xb
INT2R	C7H	INT2 Rising Edge Interrupt Control Register	INT2R7	INT2R6	INT2R5	INT2R4	INT2R3	INT2R2	INT2R1	-	0000000xb
T2CON	C8H	Timer 2 Control Register	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2	00000000b
T2MOD	C9H	Timer 2 Operating Mode Register	-	-	-	-	-	-	T2OE	DCEN	xxxxxx00b
RCAP2L	CAH	Timer 2 Reload Low 8 bits	RCAP2L[7:0]								00000000b
RCAP2H	CBH	Timer 2 Reload High 8 bits	RCAP2H[7:0]								00000000b
TL2	CCH	Timer 2 Low 8 bits	TL2[7:0]								00000000b
TH2	CDH	Timer 2 High 8 bits	TH2[7:0]								00000000b
BTMCON	CEH	Low-frequency Timer Control Register	ENBTM	BTMIF	-	-	BTMFS[3:0]				00xx0000b
WDTCON	CFH	WDT Control Register	-	-	-	CLRWDT	-	WDTCKS[2:0]			xxx0x000b
PSW	D0H	Program Status Word Register	CY	AC	F0	RS1	RS0	OV	F1	P	00000000b
PWMCFG	D1H	PWM Setup Register	-	INV6	INV5	INV4	INV3	INV2	INV1	INV0	x0000000b
PWMCON0	D2H	PWM Control Register 0	ENPWM	PWMIF	PWMCKS[1:0]		PWMOS5	PWMOS2	PWMPRD[1:0]		00000000b
PWMPRD	D3H	PWM Period Setting Register	PWMPRD[9:2]								00000000b
PWMDTYA	D4H	PWM0 duty cycle setting register A	PDT3[1:0]		PDT2[1:0]		PDT1[1:0]		PDT0[1:0]		00000000b
PWMDTY0	D5H	PWM0 duty cycle setting register	PDT0[9:2]								00000000b
PWMDTY1	D6H	PWM1 duty cycle setting register	PDT1[9:2]								00000000b
PWMDTY2	D7H	PWM2 duty cycle setting register	PDT2[9:2]								00000000b
PWMCON1	DAH	PWM Control Register 1	PWMMOD	ENPWM6	ENPWM5	ENPWM4	ENPWM3	ENPWM2	ENPWM1	ENPWM0	00000000b
PWMDTYB	DBH	PWM duty cycle setting register B	-	-	PDT6[1:0]		PDT5[1:0]		PDT4[1:0]		xx000000b
PWMDTY3	DCH	PWM3 duty cycle setting register / PWM Dead-time Configuration Register	PDT3[9:2]								00000000b
			PDF[3:0]				PDR[3:0]				
PWMDTY4	DDH	PWM4 duty cycle setting register	PDT4[9:2]								00000000b
PWMDTY5	DEH	PWM5 duty cycle setting register	PDT5[9:2]								00000000b
PWMDTY6	DFH	PWM6 duty cycle setting register	PDT6[9:2]								00000000b
ACC	E0H	Accumulator	ACC[7:0]								00000000b
OPERCON	EFH	Arithmetic Control Register	-	-	-	-	-	-	-	CHKSUMS	xxxxxxx0b
B	F0H	B Register	B[7:0]								00000000b
IAPKEY	F1H	IAP Protection Register	IAPKEY[7:0]								00000000b
IAPADL	F2H	IAP Address Low byte Register	IAPADR[7:0]								00000000b
IAPADH	F3H	IAP Address High byte Register	-	-	IAPADR[13:8]						xx000000b
IAPADE	F4H	IAP Extended Address Register	IAPADER[7:0]								00000000b
IAPDAT	F5H	IAP Data Register	IAPDAT[7:0]								00000000b

IAPCTL	F6H	IAP Control Register	-	-	-	-	PAYTIMES[1:0]	CMD[1:0]	xxxx0000b
CHKSUML	FCH	Check Sum Result Register Low	CHKSUML[7:0]						00000000b
CHKSUMH	FDH	Check Sum Result Register High	CHKSUMH[7:0]						00000000b
OPINX	FEH	Option Pointer	OPINX[7:0]						00000000b
OPREG	FFH	Option Register	OPREG[7:0]						nnnnnnnnb

6.2.1 INTRODUCTION TO 8051 CPU CORE COMMONLY-USED SPECIAL FUNCTION REGISTERS

Program Counter PC

Program counter PC does not belong to SFR .16-bit PC is the register used to control instruction execution sequence. After power-on or reset of microcontroller unit, PC value is 0000H, that is to say, the microcontroller unit is to execute program from 0000H.

Accumulator ACC (E0H)

Accumulator ACC is one of the commonly-used registers in 8051-based microcontroller unit, using A as mnemonic symbol in the instruction system. It is usually used to store operand and results for calculation or logical operations.

B Register (F0H)

B Register shall be used together with Accumulator A in multiplication and division operations. For example, instruction “MUL A, B” is used to multiply 8-bit unsigned numbers of Accumulator A and Register B, . As for the acquired 16-bit product, low byte is placed in A and high byte in B. As for DIV A, B is used to divide A by B, place integer in A and remainder in B. Register B can also be used as common temporary register.

Stack Pointer SP (81H)

Stack pointer is an 8-bit specialized register, it indicates the address of top stack in common RAM. After resetting of microcontroller unit, the initial value of SP is 07H, and the stack will increase from 08H. 08H ~ 1FH is address of register banks 1 ~ 3.

PSW (D0H) Program Status Word Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	CY	AC	F0	RS1	RS0	OV	F1	P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit Number	Bit Symbor	Instruction
7	**CY**	Carry Flag 1: the top digit of add operation has carry bit or the top digit of subtraction operation has borrow bit 0: The top digit of add operation has no carry bit or the top digit of subtraction operation has no borrow digit
6	**AC**	Carry-bit auxiliary flag bit (adjustable upon BCD code add and subtraction operations) 1: There is carry bit in bit 3 upon add operation and borrow bit in bit 3 upon subtraction operation 0: No borrow bit and carry bit
5	**F0**	User flag bit

4~3	**RS1,RS0**	Register banks selection bit: RS1 | RS0 | Currently used Register banks 0 ~ 3 0 | 0 | Group 0 (00H ~ 07H) 0 | 1 | Group 1 (08H ~ 0FH) 1 | 0 | Group 2 (10H ~ 17H) 1 | 1 | Group 3 (18H ~ 1FH)
2	**OV**	Overflow flag bit
1	**F1**	F1 flag User customized flag
0	**P**	Parity flag. This flag bit is the parity value of the number of 1 in accumulator ACC. 1: Odd number of number of 1 in ACC 0: Even number of number of 1 in ACC (including 0)

The Data Pointer DPTR (82H, 83H)

Data pointer DPTR is a 16-bit dedicated register, which is composed of low 8-bit DPL (82H) and high 8-bit DPH (83H). DPTR is the only register in the traditional 8051-based MCU that can directly conduct 16-bit operation, which can also conduct operations on DPL and DPH by byte.

7 POWER, RESET AND CLOCK

7.1 POWER CIRCUIT

SC92F8003 Power includes circuits such as BG, LDO, POR and LVR, which are able to reliably work within the scope of 2.4V~ 5.5V. Besides, a calibrated 2.4V voltage is built in the IC, which is used as ADC internal reference voltage. Users can search for specific configuration contents in 16 analog-digital conversion ADC.

7.2 POWER-ON RESET PROCESS

After SC92F8003 power-on, the processes carried out before execution of client software are as follows:

- Reset stage
- Loading information stage
- Normal operation stage

7.2.1 RESET STAGE

SC92F8003 will always be in reset mode, .There will not be a valid clock until the voltage supplied to SC92F8003 is higher than certain voltage. The duration of reset stage is related to rising speed of external power. Once the external supply voltage is up to built-in POR voltage, the reset stage ~~is~~ would be completed.

7.2.2 LOADING INFORMATION STAGE

There is a preheating counter inside SC92F8003. During the reset stage, this preheating counter is always reset as zero. After the voltage is higher than POR voltage, internal RC oscillator starts to oscillate and this preheating counter starts to count. When internal preheating counter counts up to certain number, one byte data will be read from IFB of Flash ROM (including Code Option) for every certain number of HRC clock, which is saved to internal system registers. After the preheating is completed, such reset signal will end.

7.2.3 NORMAL OPERATING STAGE

After the loading information stage has been completed, SC92F8003 starts to read instruction code from Flash

and enters normal operating stage. At this time, LVR voltage is the set value of Code Option written by user.

7.3 RESET MODE

SC92F8003 has 4 kinds of reset modes: ① External RST reset ②Low-voltage reset (LVR) ③Power-on reset (POR) ④Watchdog (WDT) reset.

7.3.1 EXTERNAL RST REST

External reset is to supply a certain width reset pulse signal to SC92F8003 from the RST pin to realize SC92F8003 reset.

RST/P1.7 has a reset function. Users can select the customer option to edit it to a non-reset pin by programming the Customer Option item before programming the program.

7.3.2 LOW-VOLTAGE RESET LVR

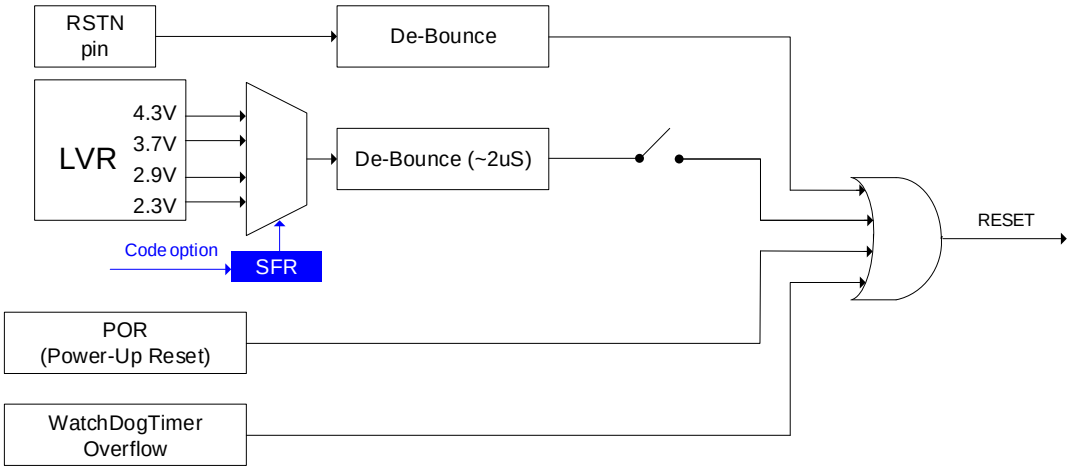
SC92F8003 provides a low-voltage reset circuit. There are 4-level LVR voltage options: 4.3V, 3.7V, 2.9V and 2.3V. The default value is the Option value written by user.

OP_CTM0(C1H@FFH) Customer Option Register 0 (Read/Write)

Bit Number	7	6	5	4	3	**2**	**1**	**0**
Bit Symbor	ENWDT	ENXTL	SCLKS[1:0]		DISRST	DISLVR	LVRS[1:0]	
R/W	R/W	R/W	R/W		R/W	R/W	R/W	
POR	n	n	n		n	n	n	

Bit Number	Bit Symbor	Instruction
2	**DISLVR**	LVR control 0: LVR valid 1: LVR invalid
1~0	**LVRS [1:0]**	LVR voltage selection bits: 11: 4.3V reset 10: 3.7 V reset 01: 2.9V reset 00: 2.3 V reset

Circuit Diagram of SC92F8003 Resetting Part is shown below:



SC92F8003 Reset Circuit Diagram

7.3.3 POWER-ON RESET POR

SC92F8003 provides a power-on reset circuit. When power voltage V_{DD} is up to POR reset voltage, the system will be reset automatically.

7.3.4 WATCHDOG RESET WDT

SC92F8003 has a WDT, the clock source of which is the internal 128 kHz oscillator. Users can select whether to enable Watchdog Reset function by programmer Code Option.

OP_CTM0 (C1H@FFH) Customer Option Register 0 (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	ENWDT	ENXTL	SCLKS[1:0]		DISRST	DISLVR	LVRS[1:0]	
R/W	R/W	R/W	R/W		R/W	R/W	R/W	
POR	n	n	n		n	n	n	

Bit Number	Bit Symbor	Instruction
7	**ENWDT**	WDT control（This bit is transferred by the system to the value set by the user Code Option） 1: WDT valid 0: WDT invalid

WDTCON (CFH) WDT Control Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	-	-	-	CLRWDT	-	WDTCKS[2:0]		
R/W	-	-	-	R/W	-	R/W		
POR	x	x	x	0	x	0	0	0

Bit Number	Bit Symbor	Instruction
4	**CLRWDT**	Clear WDT (Only valid when writing 1) 1: WDT counter restart cleared by system hardware
2~0	**WDTCKS [2:0]**	Watchdog clock selection
7~5,3	**-**	Reserved

WDTCKS[2:0]	WDT overflow time
000	500ms
001	250ms
010	125ms
011	62.5ms
100	31.5ms
101	15.75ms
110	7.88ms
111	3.94ms

7.3.5 RESET INITIAL STATE

During reset, most registers are set to their initial values, and the WDT remains disable. The initial value of program counter PC is 0000h, and the initial value of stack pointer SP is 07h. Reset of “Hot Start” (such as WDT, LVR, etc.) will not influence SRAM which always keep the value before resetting. The SRAM contents will be retained until the power voltage is too low to keep RAM alive.

The initial value of each SFR is shown in the table below: omitted here

SFR Name	POR	SFR Name	POR
ACC	00000000b	P2	00000000b
B	00000000b	P2CON	00000000b
PSW	00000000b	P2PH	00000000b
SP	00000111b	SSCON0	00000000b
DPL	00000000b	SSCON1	00000000b
DPH	00000000b	SSCON2	00000000b
PCON	0xxxxx00b	SSDAT	00000000b
ADCCFG0	x0000000b	PWMCFG	x0000000b
ADCCFG1	xxxx0000b	PWMCON0	00000000b
ADCCON	00000000b	PWMCON1	00000000b
ADCVH	00000000b	PWMDTYA	00000000b
ADCVL	0000xxxxb	PWMDTYB	xx000000b
BTMCON	00xx0000b	PWMDTY0	00000000b
WDTCON	xxx0x000b	PWMDTY1	00000000b
IAPADE	00000000b	PWMDTY2	00000000b
IAPADH	xx000000b	PWMDTY3	00000000b
IAPADL	00000000b	PWMDTY4	00000000b
IAPCTL	xxxx0000b	PWMDTY5	00000000b
IAPDAT	00000000b	PWMDTY6	00000000b
IAPKEY	00000000b	PWMPRD	00000000b
IE	00000000b	OPERCON	xxxxxxx0b
IE1	xxxx0000b	RCAP2H	00000000b
INT0R	xxxxxx00b	RCAP2L	00000000b
INT1R	x0000000b	CHKSUML	00000000b
INT2R	0000000xb	CHKSUMH	00000000b

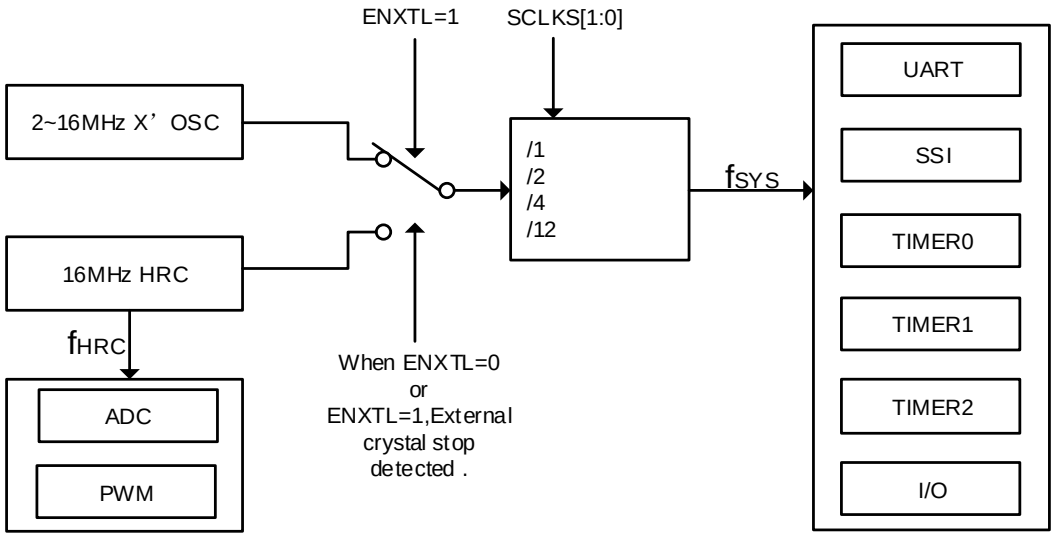
INT0F	xxxxxx00b	SBUF	00000000b
INT1F	x0000000b	SCON	00000000b
INT2F	0000000xb	TCON	00000x0xb
IP	x0000000b	TMCON	xxxxx000b
IP1	xxxx0000b	TMOD	x000x000b
OTCON	0000xxxxb	TH0	00000000b
OPINX	00000000b	TL0	00000000b
OPREG	nnnnnnnnb	TH1	00000000b
P0	xxxxxxx00b	TL1	00000000b
P0CON	xxxxxxx00b	T2CON	00000000b
P0PH	xxxxxxx00b	T2MOD	xxxxxx00b
P1	00000000b	TH2	00000000b
P1CON	00000000b	TL2	00000000b
P1PH	00000000b	-	-

7.4 HIGH-SPEED RC OSCILLATOR CIRCUIT

SC92F8003 has a built-in adjustable high-precision HRC with adjustable oscillation frequency and a crystal oscillation circuit. HRC is precisely calibrated to 16MHz@5V/25℃ when delivery. users can set system clock as 16/8/4/1.33MHz by programmer Code Option. The calibration process is to filter the influence of processing deviation on precision. There will be certain drifting of this HRC depending on operating temperature and voltage. As for voltage drifting (2.9V ~ 5.5V) and temperature drifting (-20℃ ~ 85℃), the deviation is within ±1%.

In order to enhance the reliability of the system, SC92F8003 has a system clock monitoring circuit built in. When the user chooses the system clock source as crystal oscillation and the crystal oscillation circuit stops oscillating, the system clock source will be automatically switched to the built-in HRC, and this state will be maintained until the next reset.

Note: The clock source of ADC and PWM is f_{HRC} = 16MHz, which is independent of switch of system clock.



SC92F8003 Internal Clock Relationship

OP_CTM0 (C1H@FFH) Customer Option Register 0 (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	ENWDT	ENXTL	SCLKS[1:0]		DISRST	DISLVR	LVRS[1:0]	
R/W	R/W	R/W	R/W		R/W	R/W	R/W	
POR	n	n	n		n	n	n	

Bit Number	Bit Symbor	Instruction
6	ENXTL	External High frequency crystal oscillator selection bit 0: External High frequency crystal Interface disable, P5.0 and P5.1 Valid 1: External High frequency crystal Interface enable, P5.0 and P5.1 invalid
5~4	**SCLKS[1:0]**	System clock frequency selection bits: 00: System clock frequency is HRC frequency divided by1; 01: System clock frequency is HRC frequency divided by 2; 10: System clock frequency is HRC frequency divided by 4; 11: System clock frequency is HRC frequency divided by 12;

OP_CTM1 (C2h@FFH) Customer Option Register 1 (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	VREFS	XTLHF	-	-	IAPS[1:0]		-	-
R/W	R/W	R/W	-	-	R/W	R/W	-	-
POR	n	n	x	x	n	n	x	x

Bit Number	Bit Symbor	Instruction
6	**XTLHF**	**External Crystal Control Mode Register** 0: External crystal oscillation frequency < 12M 1: External crystal oscillation frequency ≥ 12M

SC92F8003 has a special function: User can modify SFR value to adjust frequency of HRC within certain scope.

OP_HRCR (83h@FFH) System Clock Change Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	OP_HRCR[7:0]							
R/W	R/W							
POR	n	n	n	n	n	n	n	n

Bit Number	Bit Symbor	Instruction
7~0	**OP_HRCR[7:0]**	HRC frequency change register The value of OP_HRCR[7:0] after power-on ensures that HRC can work accurately at 16/8/4/1.33MHz (according to the user's Code Option). The initial value of this value may vary from IC to IC. The user can change the HRC operating frequency by modifying the value of this register. The initial value is OP_HRCR[s]. At this time, the IC operates at

16/8/4/1.33MHz. When the value of OP_HRCR [7:0] changes by 1, the IRC frequency changes by about 0.23% @16MHz based on the original frequency.
The relationship between OP_HRCR [7:0] and output frequency f_{SYS} is shown as follows:

OP_HRCR [7:0] value	HRC actual output frequency (taking 16M as an example)
OP_HRCR [s]-n	16000*(1-0.23%*n)kHz
...	....
OP_HRCR [s]-2	16000*(1-0.23%*2) = 15926.4kHz
OP_HRCR [s]-1	16000*(1-0.23%*1) = 15963.2kHz
OP_HRCR [s]	16000kHz
OP_HRCR [s]+1	16000*(1+0.23%*1) = 16036.8kHz
OP_HRCR [s]+2	16000*(1+0.23%*2) = 16073.6kHz
...	...
OP_HRCR [s]+n	16000*(1+0.23%*n)kHz

Note:
1. The value of OP_HRCR[7:0] after each power-on of the IC is HRC working at the HRC closest to 16/8/4/1.33MHz; the user can correct the value of HRC after each power-on by means of EEPROM to make HRC work. At the frequency the user needs;
2. To guarantee IC operating reliably, the maximum operating frequency of IC shall not exceed 18MHz;

3. User shall confirm the change of HRC frequency will not influence other functions.

7.5 LOW-SPEED RC OSCILLATOR AND LOW-SPEED CLOCK TIMER

SC92F8003 is equipped with a built-in 128kHz RC oscillation circuit, which can be set as clock source of low-frequency clock timer Base Timer and WDT.

Base Timer, a low-frequency clock timer which can wake up CPU from STOP mode and generate interrupt.

BTMCON (CEH) Low-frequency Timer Control Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	ENBTM	BTMIF	-	-	BTMFS[3:0]			
R/W	R/W	R/W	-	-	R/W			
POR	0	0	x	x	0	0	0	0

Bit Number	Bit Symbor	Instruction
7	**ENBTM**	Low-frequency Base Timer start control 0: Base Timer not start 1: Base Timer start
6	**BTMIF**	Base Timer interrupt application flag When CPU receives Base Timer interrupt, this flag will be cleared automatically by software.
3~0	**BTMFS [3:0]**	Low-frequency clock interrupt frequency selection 0000: An interrupt is generated for every 15.625ms 0001: An interrupt is generated for every 31.25ms 0010: An interrupt is generated for every 62.5ms

		0011: An interrupt is generated for every 125ms 0100: An interrupt is generated for every 0.25s 0101: An interrupt is generated for every 0.5s 0110: An interrupt is generated for every 1.0s 0111: An interrupt is generated for every 2.0s 1000: An interrupt is generated for every 4.0s 1001~1111:Reserved
5~4	-	Reserved

7.6 STOP MODE AND IDLE MODE

SC92F8003 provides a SFR PCON, user can configure bit 0 and bit 1 of this register to control MCU to enter different operating modes.

When PCON.1 = 1, internal high-speed system clock would stop and System enter STOP mode, to save power. The system can be woken up from STOP by external interrupt INT0 ~ INT2, low-speed clock interrupt, WDT, and external reset input.

When PCON.0 = 1, the programme would stop running and System enter IDLE mode. But the external equipment and clock will continue running, CPU will keep all states before entering IDLE mode. The system can be woken up from IDLE by any interrupt.

PCON (87H) Power Management Control Register (only for write, *unreadable*)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	SMOD	-	-	-	-	-	STOP	IDL
R/W	W	-	-	-	-	-	W	W
POR	0	x	x	x	x	x	0	0

Bit Number	Bit Symbor	Instruction
1	**STOP**	STOP mode control 0: Normal operating mode 1: Stop mode, high-frequency oscillator stops operating, low-frequency oscillator and WDT can select to work based on configuration
0	**IDL**	IDLE mode control 0: Normal operating mode 1: IDLE mode, the program stops operating, but external equipment and clock continue to operate and all CPU states are saved before entering IDLE mode.

Notes:

When Configure MCU to enter STOP OR IDLE mode, the instruction of configuring PCON register should be followed by 8 "NOP" instructions rather than other instructions. Or else, it will be unable to execute following instructions normally after wake-up!

For example: Configure MCU to enter STOP mode:

C program example:

```
#include"intrins.h"

PCON |= 0x02;        //write 1 for PCON bit1 STOP bit, configure MCU to enter STOP mode
```

```
_nop_();          // at least 8 _nop_() required
_nop_();
_nop_();
_nop_();
_nop_();
_nop_();
_nop_();
_nop_();
……
```

Assembly program example:

```
ORL PCON,#02H     ; write 1 for PCON bits1 STOP bit, configure MCU to enter STOP mode
NOP               ; at least 8 NOP required
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
……
```


8. CPU AND INSTRUCTION SYSTEM

8.1 CPU

CPU used by SC92F8003 is the high-speed 1T standard 8051 core, whose instructions are completely compatible with traditional 8051 core microcontroller unit.

8.2 ADDRESSING MODE

The addressing mode of SC92F8003 1T 8051 CPU instructions includes: ①Immediate addressing ② Direct Addressing ③ Indirect Address ④ Register Addressing ⑤ Relative Addressing ⑥ Indexed Addressing ⑦ Bit Addressing

8.2.1 IMMEDIATE ADDRESSING

Immediate addressing is also called immediate operand addressing, which is the operand given to participate in

operation in instruction, the instruction is illustrated as follows:

MOV A, #50H (This instruction is to move immediate operand 50H to Accumulator A)

8.2.2 DIRECT ADDRESSING

In direct addressing mode, the instruction operand field indicates the address to participate in operation operand. Direct addressing can only be used to address SFRs, internal data registers and bit address space. The SFRs and bit address space can only be accessed by direct addressing. For example:

ANL 50H, #91H (The instruction indicates the data in 50H unit. AND immediate operand 91H, and the results are stored in 50H unit. 50H refers to direct address, indicating one unit in internal data register RAM)

8.2.3 INDIRECT ADDRESSING

Indirect addressing is expressed as adding "@" before R0 or R1. Suppose the data in R1 is 40H and the data of internal data registers 40H unit is 55H, then the instruction will be

MOV A,@R1 (Move the data 55h to Accumulator A).

8.2.4 REGISTER ADDRESSING

Register addressing is to operate the data in the selected registers R7 ~ R0, Accumulator A, general-purpose register B, address registers and carry bit C. The registers R7-R0 is indicated by lower 3 bits of instruction code. ACC, B, DPTR and carry bit C are implied in the instruction code. Therefore, register addressing can also include an implied addressing mode. The selection of register operating area depends on RS1 and RS0 of PSW. The registers indicated by instruction operand refers to the registers in current operating area.

INC R0 refers to (R0) +1→R0

8.2.5 RELATIVE ADDRESSING

Relative addressing is to add current value in program counter PC and the data in the second byte of the instruction, whose result shall be taken as the jump address of jump instruction. The Jump address is the target jump address, the current value in PC is the base address and the data in the second byte of the instruction is the offset address. Because the target jump address is relative to base address in PC, such addressing mode is called relative addressing. The offset is signed number, which ranges from +127 to -128, such addressing mode is mainly applied to jump instruction.

JC $+50H

It indicates that if the carry bit C is 0, the contents in program counter PC remain the same, meaning no jump. On the contrary, if the carry bit C is 1, take the sum of the current value in PC and base address as well as offset 50H as the target jump address of this jump instruction.

8.2.6 INDEXED ADDRESSING

In indexed addressing mode, the instruction operand is to develop an indexed register to store indexed base address. Upon indexed addressing, the result by adding offset and indexed base address is taken as the address of operation operand. The indexed registers include program counter PC and address register DPTR.

MOVC A, @A+DPTR

It indicates Accumulator A is used as offset register . Take the sum of the value in A and that in the address register DPTR as the address of operand. Then take the figure in the address out and transmit it to Accumulator A.

8.2.7 BITS ADDRESSING

Bit addressing is a kind of addressing mode when conducting bit operation on internal data storage RAM and SFRs which are able to carry out bit operations. Upon bit operations, by taking carry bit C as bit operation accumulator, the instruction operand will give the address of this bit directly, then execute bit operation based on the nature of operation code.

```
MOV   C, 20H      (Send the bit operation register with address of 20H into carry bit C)
```

9 INTERRUPT

The SC92F8003 provides 11 interrupt sources: Timer0, Timer1, Timer2, INT0~2, UART,ADC, SSI,PWM, and Base Timer. Each interrupt source can be individually configured in high priority or low priority. As for three external interrupts, the triggering condition of each interrupt source can be set as rising edge, falling edge or dual-edge trigger. Each interrupt is equipped with independent priority setting bit, interrupt flag, interrupt vector and enable bit. Global interrupt enable bit EA can enable or disable all interrupts.

9.1 INTERRUPT SOURCE AND VECTOR

Lists for SC92F8003 interrupt source, interrupt vector and related control bit are shown below:

Interrupt Source	Interrupt conditon	Interrupt Flag	Interrupt Enable Control	Interrupt Priority Control	Interrupt Vector	Query Priority	（C51）Interrupt Number (C51)	Flag Clear Mode	capability of Wakeing up STOP
INT0	compliant with External interrupt 0 conditions	IE0	EINT0	IPINT0	0003H	1（high）	0	H/W Auto	Yes
Timer0	Timer0 overflow	TF0	ET0	IPT0	000BH	2	1	H/W Auto	No
INT1	compliant with External interrupt 1 conditions	IE1	EINT1	IPINT1	0013H	3	2	H/W Auto	Yes
Timer1	Timer 1 overflow	TF1	ET1	IPT1	001BH	4	3	H/W Auto	No
UART	Receiving or transmitting completed	RI/TI	EUART	IPUART	0023H	5	4	Must be cleared by user	No
Timer2	Timer 2 overflow	TF2	ET2	IPT2	002BH	6	5	Must be cleared by user	No
ADC	ADC conversion completed	ADCIF	EADC	IPADC	0033H	7	6	Must be cleared by user	No
SSI	Receiving or transmitting completed	SPIF/TWIF	ESSI	IPSPI	003BH	8	7	Must be cleared by user	No
PWM	PWM overflow	PWMIF	EPWM	IPPWM	0043H	9	8	H/W Auto	No
BTM	Base timer overflow	BTMIF	EBTM	IPBTM	004BH	10	9	H/W Auto	Yes
INT2	External interrupt 2 conditions compliant	-	EINT2	IPINT2	0053H	11	10	-	Yes

Under the circumstance where the master interrupt control bit EA and the respective interrupt control bit have been enable, the interrupt occurrence is shown below:

Timer Interrupt: Interrupt generates when Timer0 or Timer1 overflows and the interrupt flag TF0 or TF1 is set to “1”. When the microcontroller unit responds to the timer interrupt, the interrupt flag TF0 or TF1 is reset automatically by hardware. Interrupt occurs when Timer2 overflows and the interrupt flag TF2 is set to “1”. Once Timer2 interrupt occurs, the hardware would not automatically clear TF2 bit, which must be cleared by software.

UART interrupt: When the UART receives or transmits a frame of data, the RI or TI bit is automatically set to "1" by the hardware, and the UART interrupt is generated. After the UART interrupt occurs, the hardware does not automatically clear the RI/TI bit. This bit must be cleared by the user's software.

ADC Interrupt: After ADC conversion is completed, ADC interrupt occurs, whose interrupt flag is the ADC conversion completion flag EOC/ADCIF (ADCCON.5). When users starts ADCS conversion, EOC will be reset automatically by hardware. Once conversion completes, EOC would be set to “1” automatically by hardware. Users should clear the ADC interrupt flag by software when the interrupt service routine is executed after ADC interrupt occurs.

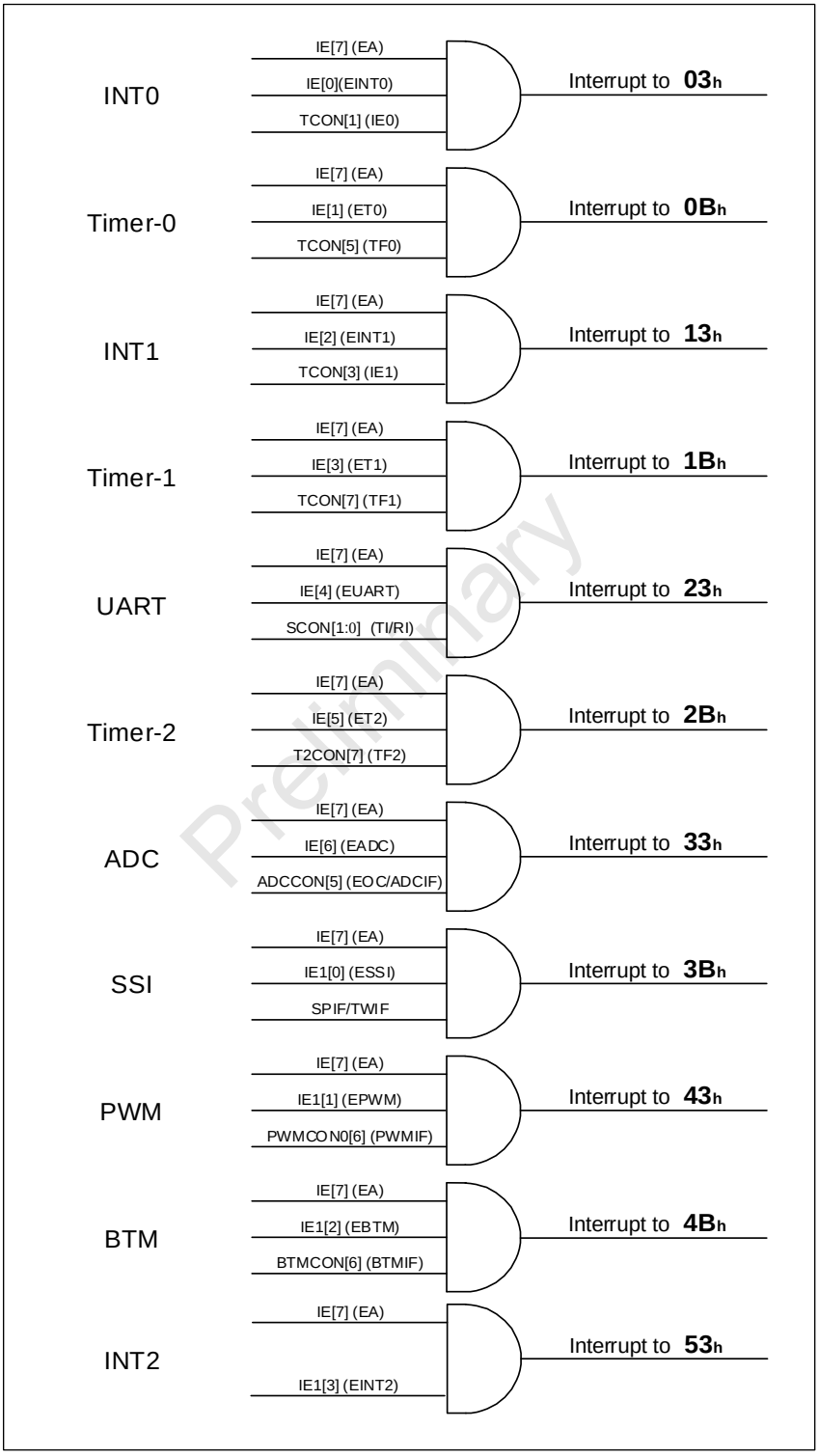
SSI Interrupt: When SSI completes receiving or transmitting a frame of data, SPIF/TWIF bit will be set to “1” automatically by hardware, and SSI interrupt occurs.When the microcontroller unit serves SSI interrupt, the interrupt flag SPIF/TWIF must be cleared by software.

PWM Interrupt: When PWM counter overflows (beyond PWMPD), The PWMIF bit will be set to 1 automatically by hardware, PWM interrupt occurs. When the microcontroller unit executes PWM interrupt, The interrupt sign PWMIF will be reset to “0” automatically by hardware.

External Interrupt INT0~2: When any external interrupt pin meets the interrupt conditions, external interrupt occurs. The external interrupt INT0 and INT1 would set up interrupt flag IE0 and IE1 respectively, which will be automatically cleared by hardware rather than user. Users can be set in rising edge, falling edge or dual edge interrupt trigger mode by setting SFRs (INTxF and INTxR). Users can set the priority level of each interrupt through IP register. Besides, external interrupt INT0~2 can also wake up STOP mode of microcontroller unit.

9.2 INTERRUPT STRUCTURE DIAGRAM

SC92F8003 interrupt structure is shown in the figure below:



SC92F8003 Interrupt Structure and Vector

9.3 INTERRUPT PRIORITY

SC92F8003 microcontroller unit has two-level interrupt priority capability. The interrupt requests of these interrupt sources can be programmed as high-priority interrupt or low-priority interrupt, which is to realize the nesting of two levels of interrupt service programs. One interrupt can be interrupted by a higher priority interrupt request when being responded to, which can not be interrupted by another interrupt request at the same priority level, until such response to the first-come interrupt ends up with the instruction “RETI”. Exist the interrupt service routine and return to main program, the system would execute one more instruction before responding to new interrupt request.

That is to say:

① A lower priority interrupt can be interrupted by a higher priority interrupt request, but not vice verse;

② Any kind of interrupt being responded to can not be interrupted by another interrupt request at the same priority level.

Interrupt query sequence: As for the sequence of that SC92F8003microcontroller unit responds to the same priority interrupts which occur in the meantime, the priority sequence of interrupt response shall be the same as the interrupt query number in C51, which is to preferentially respond to the interrupt with smaller query number then to the interrupt with bigger query number.

9.4 INTERRUPT PROCESSING FLOW

When any interrupt occurs and is responded by CPU, the operation of main program will be interrupted to carry out the following operations

① Complete execution of instruction being currently executed;

② Push the PC value into stack for site protection;

③ Load Interrupt vector address into program counter PC;

④ Carry out corresponding interrupt service program;

⑤ End Interrupt service program ends and execute RETI;

⑥ Pop PC value from stack and return to the program before responding to the interrupt.

During this process, the system will not immediately respond to other interrupts at the same priority level, but it will keep all interrupt requests having occurred and respond to new interrupt requests upon completing handling of the current interrput.

9.5 INTERRUPT-RELATED SFR REGISTERS

IE (A8H) Interrupt Enable Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	EA	EADC	ET2	EUART	ET1	EINT1	ET0	EINT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit Number	Bit Symbor	Instruction
7	**EA**	**Global Interrupt Enable** All interrupts are disabled when EA = 0. When EA = 1, each interrupt source is enabled/disabled by setting /clearing its own enable bit.
6	**EADC**	Enable ADC interrupt 0: disable ADC interrupt 1: Interrupt is allowed upon completing ADC conversion
5	**ET2**	Timer2 interrupt enabling control 0: disable Timer2 interrupt 1: enable Timer2 interrupt
4	**EUART**	UART interrupt enabling control 0: disable UART interrupt 1: enable UART interrupt
3	**ET1**	Timer1 interrupt enabling control 0: disable Timer1 interrupt 1: enable Timer1 interrupt
2	**EINT1**	External interrupt 1 enabling control 0: disable INT1 interrupt 1: enable INT1 interrupt
1	**ET0**	Timer0 interrupt enabling control 0: disable Timer0 interrupt 1: enable Timer0 interrupt
0	**EINT0**	External interrupt 0 enabling control 0: disable INT0 interrupt 1: enable INT0 interrupt

IP (B8H) Interrupt Priority Control Register (Read/Write)

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Symbor	-	IPADC	IPT2	IPUART	IPT1	IPINT1	IPT0	IPINT0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	x	0	0	0	0	0	0	0

Bit Number	Bit Symbor	Instruction
6	**IPADC**	ADC interrupt priority selection 0: ADC interrupt priority is low 1: ADC interrupt priority is high
5	**IPT2**	Timer2 interrupt priority selection 0: Timer2 interrupt priority is low 1: Timer2 interrupt priority is high
4	**IPUART**	UART interrupt priority selection 0: UART interrupt priority is low 1: UART interrupt priority is high
3	**IPT1**	Timer1 interrupt priority selection 0: Timer1 interrupt priority is low 1: Timer1 interrupt priority is high
2	**IPINT1**	INT1 interrupt priority selection 0: INT1 interrupt priority is low 1: INT1 interrupt priority is high
1	**IPT0**	Timer1 interrupt priority selection 0: Timer0 interrupt priority is low

		1: Timer0 interrupt priority is high
0	**IPINT0**	INT0 interrupt priority selection 0: INT0 interrupt priority is low 1: INT0 interrupt priority is high
7	**-**	Reserved

IE1 (A9H) Interrupt Enable Register 1 (Read/Write)

Bit Number	7	6	5	-	3	2	1	0
Bit Symbor	-	-	-	-	EINT2	EBTM	EPWM	ESSI
R/W	-	-	-	-	R/W	R/W	R/W	R/W
POR	x	x	x	x	0	0	0	0

Bit Number	Bit Symbor	Instruction
3	**EINT2**	External interrupt 2 enabling control 0: disable External interrupt 2 1: enable External interrupt 2
2	**EBTM**	Base Timer interrupt enabling control 0: disable Base Timer interrupt 1: enable Base Timer interrupt
1	**EPWM**	PWM interrupt enabling control 0: disable PWM interrupt 1: enable interrupt upon PWM counting overflows (number to PWMPRD)
0	**ESSI**	SSI interrupt enabling control 0: disable serial port interrupt 1: enable serial port interrupt
7~4	**-**	Reserved

IP1 (B9H) Interrupt Priority Control Register 1(Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	-	-	-	-	IPINT2	IPBTM	IPPWM	IPSSI
R/W	-	-	-	-	R/W	R/W	R/W	R/W
POR	x	x	x	x	0	0	0	0

Bit Number	Bit Symbor	Instruction
3	**IPINT2**	INT2 counter interrupt priority selection 0: INT2 interrupt priority is low 1: INT2 interrupt priority is high
2	**IPBTM**	Base Timer interrupt priority selection 0: Base Timer interrupt priority is low 1: Base Timer interrupt priority is high
1	**IPPWM**	PWM interrupt priority selection 0: PWM interrupt priority is low 1: : PWM interrupt priority is high
0	**IPSSI**	SSI interrupt priority selection 0: SSI interrupt priority is low 1: SSI interrupt priority is high
7~4	**-**	Reserved

TCON (88H) Timer Control Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	TF1	TR1	TF0	TR0	IE1	-	IE0	-
R/W	R/W	R/W	R/W	R/W	R/W	-	R/W	-
POR	0	0	0	0	0	x	0	x

Bit Number	Bit Symbor	Instruction
3	**IE1**	INT1 overflow interrupt request flag. INT1 overflow occurs, interrupt occurs, hardware places IE1 in ”1”, the application is interrupted, upon CPU responds, the hardware resets it to ”0”
1	**IE0**	INT0 overflow interrupt request flag. INT0 overflow occurs, interrupt occurs, hardware places IE0 in ”1”, the application is interrupted, upon CPU responds, the hardware resets it to ”0”
2,0	**-**	Reserved

INT0F (BAH) INT0 Falling Edge Interrupt Control Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	-	-	-	-	-	-	INT0F1	INT0F0
R/W	-	-	-	-	-	-	R/W	R/W
POR	x	x	x	x	x	x	0	0

Bit Number	Bit Symbor	Instruction
1~0	**INT0Fn (n=0~1)**	INT0 falling edge interrupt control 0: INT0n falling edge interrupt off 1: INT0n falling edge interrupt enabling
7~2	**-**	Reserved

INT0R (BBH) INT0 Rising Edge Interrupt Control Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	-	-	-	-	-	-	INT0R1	INT0R0
R/W	-	-	-	-	-	-	R/W	R/W
POR	x	x	x	x	x	x	0	0

Bit Number	Bit Symbor	Instruction
1~0	**INT0Rn (n=0~1)**	INT0 rising edge interrupt control 0: INT0n rising edge interrupt off 1: INT0n rising edge interrupt enabling
7~2	**-**	Reserved

INT1F (BCH) INT1 Falling Edge Interrupt Control Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	-	INT1F6	INT1F5	INT1F4	INT1F3	INT1F2	INT1F1	INT1F0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W

POR	x	0	0	0	0	0	0	0

Bit Number	Bit Symbor	Instruction
6~0	**INT1Fn (n=0~6)**	INT1 falling edge interrupt control 0: INT1n falling edge interrupt off 1: INT1n falling edge interrupt enabling
7	**-**	Reserved

INT1R (BDH) INT1 Rising Edge Interrupt Control Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	-	INT1R6	INT1R5	INT1R4	INT1R3	INT1R2	INT1R1	INT1R0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	x	0	0	0	0	0	0	0

WEI	Bit Symbor	Instruction
6~0	**INT1Rn (n=0~6)**	INT1 rising edge interrupt control 0: INT1n rising edge interrupt off 1: INT1n rising edge interrupt enabling
7	**-**	Reserved

INT2F (C6H) INT2 Falling Edge Interrupt Control Register (Read/Write)

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Symbor	INT2F7	INT2F6	INT2F5	INT2F4	INT2F3	INT2F2	INT2F1	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	-
POR	0	0	0	0	0	0	0	x

Bit Number	Bit Symbor	Instruction
7~1	**INT2Fn (n=1~7)**	INT2 falling edge interrupt control 0: INT2n falling edge interrupt off 1: INT2n falling edge interrupt enabling
0	**-**	Reserved

INT2R (C7H) INT2 Rising Edge Interrupt Control Register (Read/Write)

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Symbor	INT2R7	INT2R6	INT2R5	INT2R4	INT2R3	INT2R2	INT2R1	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	-
POR	0	0	0	0	0	0	0	x

Bit Number	Bit Symbor	Instruction
7~1	**INT2Rn (n=1~7)**	INT2 rising edge interrupt control 0: INT2n rising edge interrupt off 1: INT2n rising edge interrupt enabling
0	**-**	Reserved

10 TIMER/EVENT COUNTER TIMER0 AND TIMER1

T0 and T1 is the SC92F8003 microcontroller provides two 16-bit timers/counters, each of which can operate as either a timer or an event counter. The type of operation is selected by bit C/Tx in the SFR TMOD. They are essentially adding counters with different counting source. The source of timer generated from system clock or frequency division clock, but the source of counters is the input pulse to external pin. Only when TRx = 1, will T0 and T1 be enabled on for counting.

In counter mode, each input pulse on P1.2/T0 and P1.3/T1 pin will make the count value of T0 and T1 increase by 1 respectively.

In timer mode, users can select $f_{SYS}/12$ or f_{SYS} (f_{SYS} is the system clock after frequency division) as counting source of T0 and T1 by configuring SFR TMCON.

Timer/counter T0 has 4 operating modes, and timer/counter T1 has 3 operating modes (Mode 3 does not exist):

① Mode 0: 13-bit timer/counter mode

② Mode 1: 16-bit timer/counter mode

③ Mode 2: 8-bit automatic reload mode

④ Mode 3: Two 8-bit timers/counters mode

In above modes, modes 0, 1 and 2 of T0 and T1 are the same, and mode 3 is different.

10.1 T0 AND T1-RELATED SPECIAL FUNCTION REGISTERS

Symbol	Address	Instruction	7	6	5	4	3	2	1	0	Reset Value
TCON	88H	Timer Control Register	TF1	TR1	TF0	TR0	IE1	-	IE0	-	00000x0xb
TMOD	89H	Timer Operating Mode Register	-	C/T1	M11	M01	-	C/T0	M10	M00	x000x000b
TL0	8AH	Timer0 Low 8-bit	TL0[7:0]								00000000b
TL1	8BH	Timer1 Low 8-bit	TL1[7:0]								00000000b
TH0	8CH	Timer0 High 8-bit	TH0[7:0]								00000000b
TH1	8DH	Timer1 High 8-bit	TH1[7:0]								00000000b
TMCON	8EH	Timer Frequency Control Register	-	-	-	-	-	T2FD	T1FD	T0FD	xxxxx000b

Register instructions are shown below: omitted here

TCON (88H) Timer Control Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	TF1	TR1	TF0	TR0	IE1	-	IE0	-
R/W	R/W	R/W	R/W	R/W	R/W	-	R/W	-
POR	0	0	0	0	0	x	0	x

Bit Number	Bit Symbor	Instruction
7	**TF1**	**Timer 1 Overflow Flag** Set by hardware on Timer/Counter overflow. Cleared by hardware when the

		processor vectors to interrupt routine.
6	**TR1**	**Timer 1 Run Control** Set/cleared by software to turn Timer/Counter on/off.
5	**TF0**	**Timer 0 Overflow Flag** Set by hardware on Timer/Counter overflow. Cleared by hardware when the processor vectors to interrupt routine.
4	**TR0**	**Timer 0 Run Control** Set/cleared by software to turn Timer/Counter on/off.

TMOD (89H) Timer Operating Mode Register (Read/Write)

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Symbor	-	C/T1	M11	M01	-	C/T0	M10	M00
R/W	-	R/W	R/W	R/W	-	R/W	R/W	R/W
POR	x	0	0	0	x	0	0	0
	T1				T0			

Bit Number	Bit Symbor	Instruction
6	**C/T1**	**Timer or Counter Selector 1** Cleared for Timer operation (input from internal system clock). Set for Counter operation (input from T1 input pin).
5~4	**M11,M01**	**Timer 1 Operating Mode** **Mode / M11 / M01 / Operation** **0** / **0** / **0** / 13-bit TIMER/Counter, TL1 high 3 bits invalid **1** / **0** / **1** / 16-bit timer/counter **2** / **1** / **0** / 8-bit Auto-Reload Mode. TH1 holds a value which is reloaded into 8-bit Timer/Counter TL1 each time it overflows. **3** / **1** / **1** / Timer/Counter 1 is stopped
2	**C/T0**	**Timer or Counter Selector 0** Cleared for Timer operation (input from internal system clock). Set for Counter operation (input from T0 input pin).
1~0	**M10,M00**	**Timer 0 Operating Mode** **Mode / M10 / M00 / Operation** **0** / **0** / **0** / 13-bit TIMER/Counter, TL0 high 3 bits invalid **1** / **0** / **1** / 16-bit timer/counter **2** / **1** / **0** / 8-bit Auto-Reload Mode. TH0 holds a value which is reloaded into 8-bit Timer/Counter TL0 each time it overflows. **3** / **1** / **1** / Split Timer Mode. TL0 is an 8-bit Timer/Counter controlled by the standard Timer 0 control bits. TH0 is only an 8-bit timer controlled by Timer 1 control bits
7,3	**-**	Reserved

TMOD[0] ~ TMOD[2] of TMOD register is to set operating mode of T0; TMOD[4]~TMOD[6] is to set the operating mode of T1.

The function of timer and counter Tx is selected by the control bit C/Tx of SFR TMOD, both M0x and operating mode selected by M0x and M1x Only when TRx, the switch of T0 and T1, is set to 1, will T0 and T1 be enabled

TMCON (8EH) Timer Frequency Control Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	-	-	-	-	-	T2FD	T1FD	T0FD
R/W	-	-	-	-	-	R/W	R/W	R/W
POR	x	x	x	x	x	0	0	0

Bit Number	Bit Symbor	Instruction
1	**T1FD**	T1 input frequency selection control 0: T1 frequency is generated from $f_{SYS}/12$ 1: T1 frequency is generated from f_{SYS}
0	**T0FD**	T0 input frequency selection control 0: T0 frequency is generated from $f_{SYS}/12$ 1: T0 frequency is generated from f_{SYS}

IE (A8H) Interrupt Enable Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	EA	EADC	ET2	EUART	ET1	-	ET0	EINT0
R/W	R/W	R/W	R/W	R/W	R/W	-	R/W	R/W
POR	0	0	0	0	0	x	0	0

Bit Number	Bit Symbor	Instruction
3	**ET1**	**Timer 1 Interrupt Enable** Clear to disable the Timer 1 interrupt. Set to enable the Timer 1 interrupt when EA = 1.
1	**ET0**	**Timer 0 Interrupt Enable** Clear to disable the Timer 0 interrupt. Set to enable the Timer 0 interrupt when EA = 1.

IP (B8H) Interrupt Priority Control Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	-	IPADC	IPT2	IPUART	IPT1	IPINT1	IPT0	IPINT0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	x	0	0	0	0	0	0	0

Bit Number	Bit Symbor	Instruction
3	**IPT1**	Timer1 interrupt priority 0: Configure Timer1 interrupt priority as ”low ” 1: Configure Timer1 interrupt priority as ”high”
1	**IPT0**	Timer 0 interrupt priority 0: Configure Timer 0 interrupt priority as ”low ” 1: Configure Timer 0 interrupt priority as ”high”

10.2 T0 OPERATING MODE

Timer 0 can be configured in one of four operating modes by setting the bit pairs (M10, M00) in the TMOD

register.

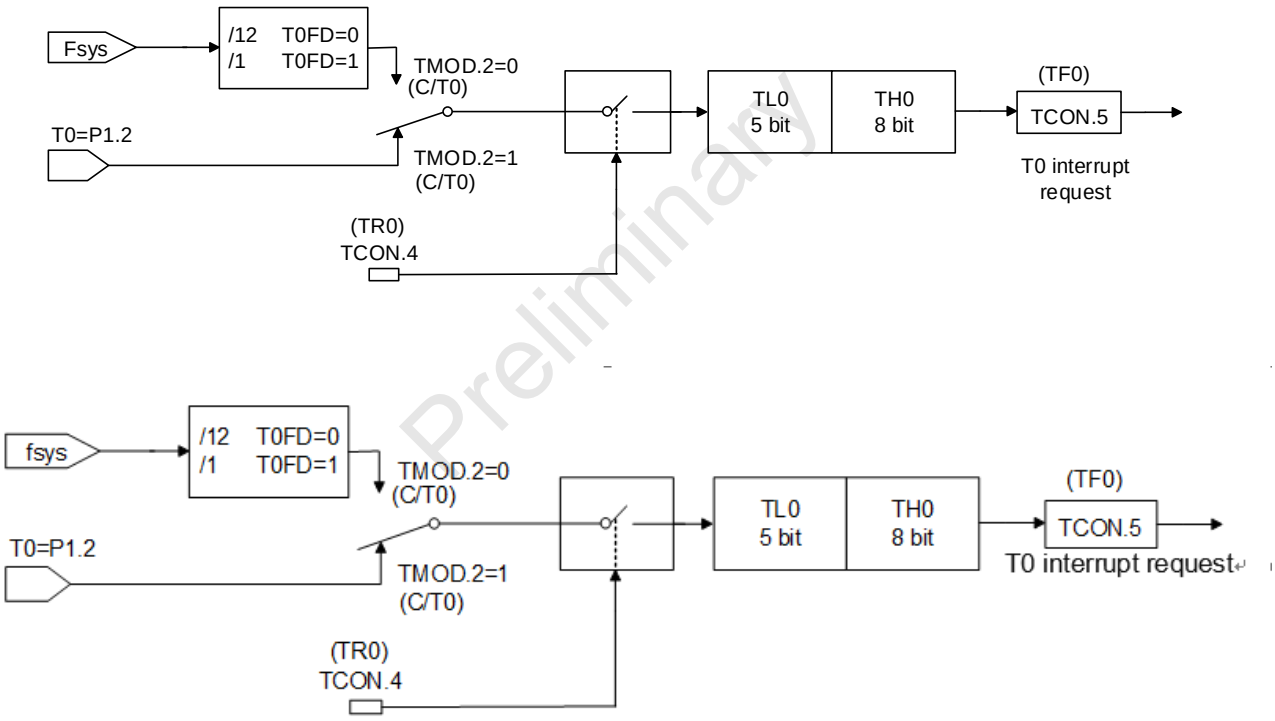
Operating mode 0: 13-bit counter/timer

TH0 register is to store the high 8 bits (TH0.7~TH0.0) of 13-bit counter/timer and TL0 is to store the low 5 bits (TL0.4~TL0.0). The high three bits of TL0 (TL0.7~TL0.5) are filled with uncertain numbers, they shall be omitted upon reading. When 13-bit timer/counter overflows with count increment, the system will set timer overflow flag TF0 as 1. If the timer 0 interrupt is enabled, it will generate an interrupt.

C/T0 bit selects the clock input source of counter/timer. If C/T0=1, the level fluctuation from high to low of Counter 0 input pin T0 (P1.2) will make Counter 0 data register add 1. If C/T0=0, the frequency division of system clock is the clock source of Timer 0.

When TR0 = 1, Timer 0 is enabled. Setting TR0 would not reset the timer forcibly. It means that the timer register will start to count from the value of last clearing of TR0. Therefore, before enable the timer, it is required to configure the initial value of timer register.

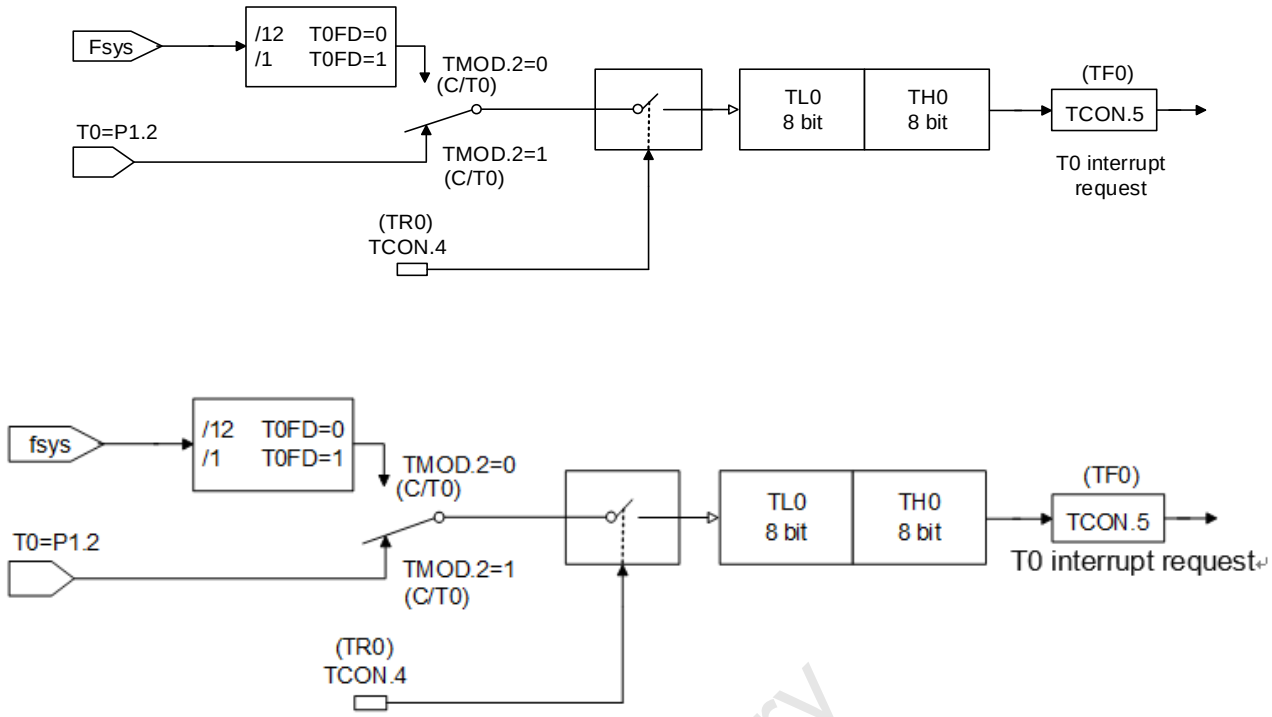
When configured as a timer, configure T0FD to select fractional frequency ratio of clock source.



Operating mode 0: 13-bit counter/timer

Operating Mode 1: 16 Counter/Timer

Except for using 16 bits of (valid for all 8 bits of TL0) counter/timer, in mode 1 and mode 0, the operating mode, opening and configuration method are the same.



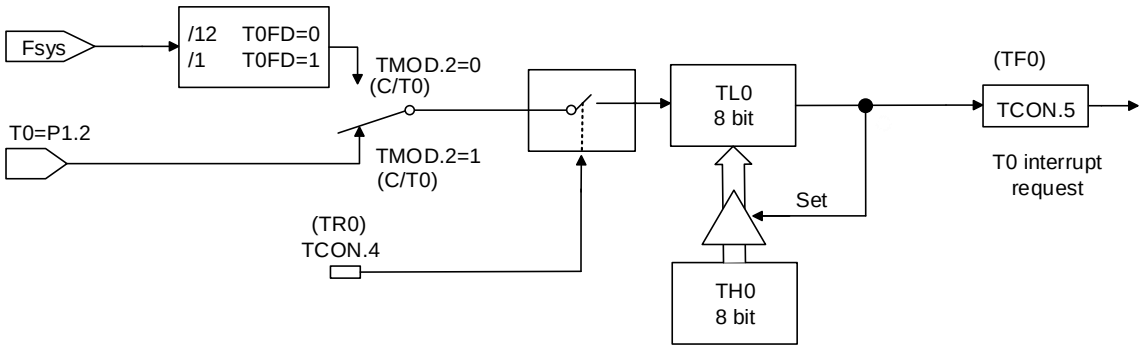
Operating mode 1: 16-bit counter/timer

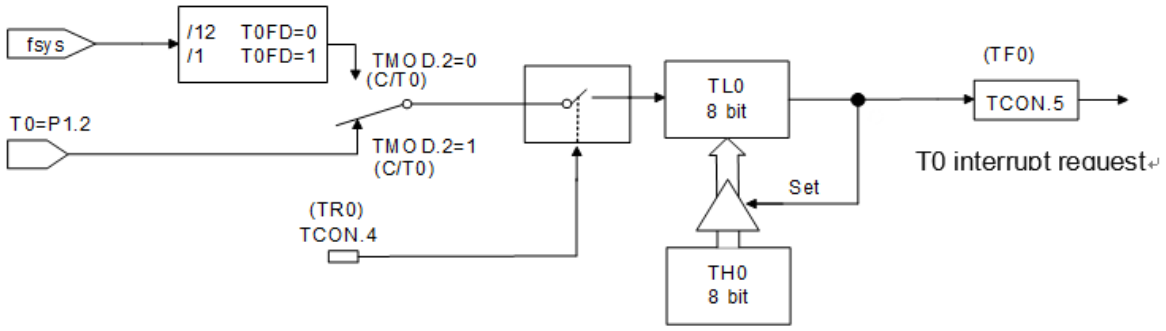
Operating Mode 2: 8 Automatic Reload Counter/Timer

In operating mode 2, Timer 0 is 8-bit automatic reload counter/timer. TL0 is to store counting value and TH0 is to store the reload value. When the counter in TL0 overflows and turn to 0x00, the overflow flag of Timer TF0 will be set to 1, and the data in register TH0 will be reloaded into register TL0. If the timer interrupt enabled, setting TF0 as 1 will generate an interrupt, but the reloaded value in TH0 will remain the same .Before starting the Timer to count correctly, TL0 shall be initialized to the required value.

Except for automatic reloaded function, the enabling and configuration mode of counter/timer in operating mode 2 shall be the same as that in mode 0 and mode 1.

When configured as a timer, it is advisable to configure register TMCON bit 0 (T0FD) to select fractional frequency ratio of system clock f_{SYS}.





Operating Mode 2: 8 Automatic Reload Counter/Timer

Operating Mode 3: Two 8-bit Counter/Timer (only for Timer 0)

In operating mode 3, Timer 0 is used as two independent 8-bit counters/timers, respectively controlled by TL0 and TH0. TL0 is controlled by control bit (in TCON) and status bit (in TMOD) of Timer 0, namely TR0, C/T0, TF0. Timer 0 is selected as Timer or Counter by TMOD bit 2(C/T0).

TH0 is controlled by configuring control bit (in TCON) of Timer 1, but TH0 is only limited to in Timer Mode, which is unable to configure as a Counter by TMOD.2(C/T0). TH0 is enabled by set the timer control bit TR1 as 1 . When overflow occurs and interrupt is discovered, set TF1 as 1 and proceed the interrupt as T1 interrupt.

When T0 is configured as in Operating Mode 3, TH0 Timer occupies T1 interrupt resources and TCON register and the 16-bit counter of T1 will stop counting, equivalently “TR1=0”. When adopting TH0 timer, it is required to configure TR1=1.

10.3 T1 OPERATING MODE

Timer 1 can be configured in one of four operating modes by setting the bit pairs (M11, M01) in the TMOD register.

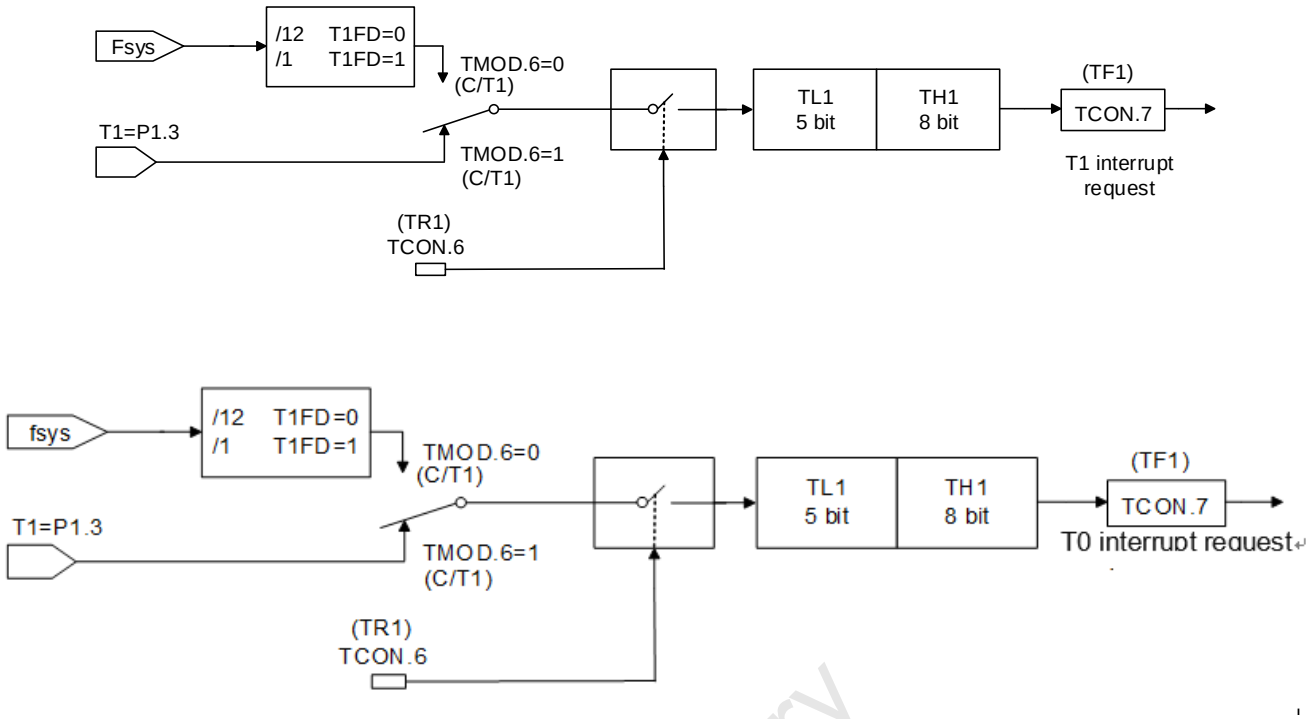
Operating mode 0: 13-bit counter/timer

TH1 register is to store high 8 digits (TH1.7~TH1.0) of 13-digit counter/timer and TL1 is to store low 5 digits (TL1.4~TL1.0). The high three digits of TL1 (TL1.7~TL1.5) are uncertain values, they shall be omitted upon reading. When 13-bit timer/counter overflow with count increment, the system will set timer overflow flag TF1 as1. If the timer 1 interrupt is allowed, it will generate an interrupt. C/T1 bit selects the clock input source of counter/timer.

If C/T1=1, the level fluctuation from high to low of timer 1 input pin T1 (P1.3) will make timer 1 data register add 1. If C/T1=0, the frequency division of system clock is the clock source of timer 1.

When TR1 is set to 1 and the timer is enabled. Setting TR1 does not force to reset timer counters, it means, if set TR1 to 1, the timer register will start to count from the value of last clearing of TR1. Therefore, before allowing timer, it is required to configure the initial value of timer register.

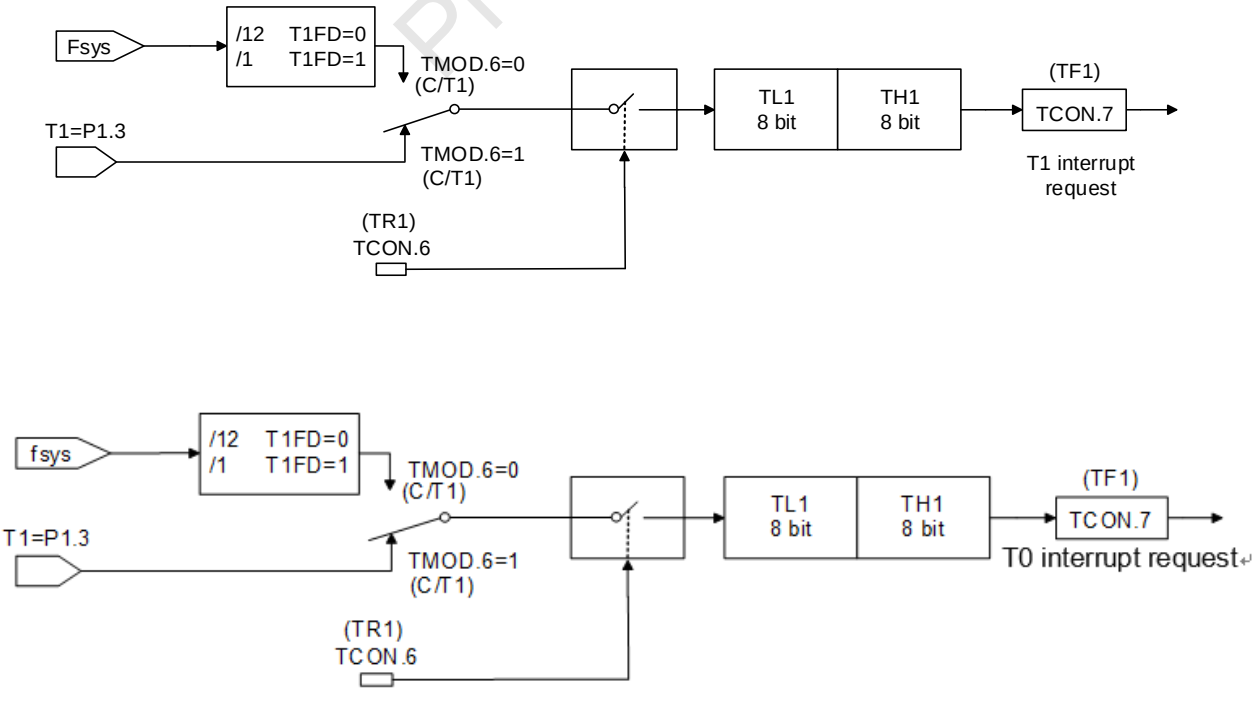
When configured as timer, configure T1FD to select fractional frequency ratio of clock source.



Operating mode 0: 13-bit counter/timer

Operating Mode 1:16 Counter/Timer

Except for using 16-bit (valid for 8-bit data of TL1) counter/timer, the operating mode of mode 1 and mode 0 is the same. And the opening and configuration mode of both are also the same.



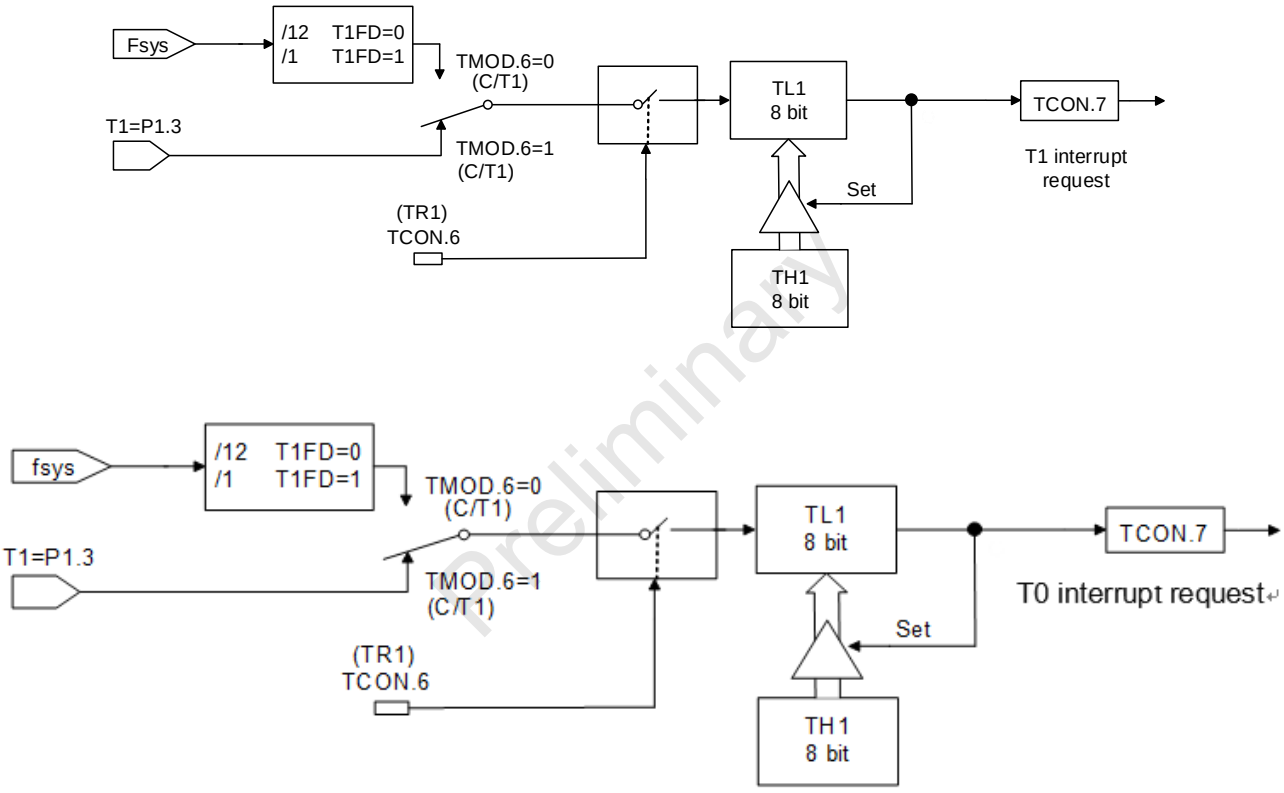
Operating mode 0: 16-bit counter/timer

Operating Mode 2:8 Automatic Reload Counter/Timer

In operating mode 2, Timer 1 is 8-digit automatic reload counter/timer. TL1is to store counting value and TH1 is to store the reload value. When the counter in TL1 overflows 0x00, the overflow flag of Timer TF1 will be set to 1, and the value of register TH1 will be reloaded into register TL1. If enable the timer interrupts, setting TF1 in 1 will generate an interrupt, but the reloaded value in TH1 will remain unchanged. Before allowing Timer to correctly count, TL1 shall be initialized to the required value.

Except for automatic reloaded function, the enabling and configuration mode of counter/timer in operating mode 2 shall be the same as that of mode 0 and mode 1.

When configured as timer, it is advisable to configure register TMCON.4 (T1FD) to select the ratio of clock source of timer to fractional frequency of system clock f_{SYS}.



Operating Mode 2: 8 Automatic Reload Counter/Timer

11 TIMER2

Timer 2 inside SC92F8003 microcontroller unit has two operating modes, namely counter mode and timer mode. There is a control bit C/T2 in special function register T2CON to select Timer or Counter for T2. They are adding counters in nature, differing in counting source. The source of timer comes from system clock or frequency division clock, but the source of counters is the input pulse to external pin. TR2 is the counting switch of timer/counter T2. Only when TR2 = 1, will T2 be enabled for counting.

In counter mode, each input pulse on T2 pin will make the counting value of T2 increase by 1.

In timer mode, users can select $f_{SYS}/12$ or f_{SYS} as counting source of T2 by configuring SFR TMCON.

Timer/counter T2 has 4 operating modes:

① Mode 0: 16-bit capture mode

② Mode 1: 16-bit automatic reload timer mode

③ Mode 2: Baud rate generator mode

④ Mode 3: Programmable clock output mode

11.1 T2-RELATED SPECIAL FUNCTION REGISTERS

Symbol	Address	Instruction	7	6	5	4	3	2	1	0	Reset Value
T2CON	C8H	Timer2 Control Register	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2	00000000b
T2MOD	C9H	Timer2 Operating Mode Register	-	-	-	-	-	-	T2OE	DCEN	xxxxxx00b
RCAP2L	CAH	Timer 2 Reload /Capture Low 8-bit	RCAP2L[7:0]								00000000b
RCAP2H	CBH	Timer 2 Reload /Capture High 8-bit	RCAP2H[7:0]								00000000b
TL2	CCH	Timer2 Low 8-bit	TL2[7:0]								00000000b
TH2	CDH	Timer2 High 8-bit	TH2[7:0]								00000000b
TMCON	8EH	Timer Frequency Control Register	-	-	-	-	-	T2FD	T1FD	T0FD	xxxxx000b

The explanation of each register is as follows:

T2CON (C8H) Timer2 Control Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit Number	Bit Symbor	Instruction
7	**TF2**	Timer 2 overflow flag bit 0: No overflow (must be cleared by software) 1: Overflow (if RCLK=0 and TCLK=0, set to 1 by hardware)
6	**EXF2**	T2 pin external event input (falling edge) detected flag bit 0: no external event input (must be cleared by software) 1: Detect external input (if EXEN2=1, SET to 1 by hardware)
5	**RCLK**	UART receiving clock control bit 0: Timer1 generates receiving baud rate 1: Timer2 generates receiving baud rate
4	**TCLK**	UART transmitting clock control bit 0: Timer1 generates transmitting baud rate 1: Timer2 generates transmitting baud rate
3	**EXEN2**	External event input (falling edge) on T2 pin used as reload/capture trigger allowed/prohibitsed control: 0: Omit event on T2 pin 1: When the timer 2 is not used as UART clock (T2EX always including pull-up resistance), a falling edge is detected on T2 pin and generate a capture or reload
2	**TR2**	Timer2 start/stop control bit 0: Stop Timer 2 1: Start Timer 2
1	**C/T2**	Timer 2 timer/counter mode selection bit 2 0: Timer mode, used as I/O interface on T2 pin 1: Counter mode

0	**CP/RL2**	Capture/reload mode selection bit 0: 16-bit timer/counter with reload function 1: 16-bit timer/counter with capture function, T2EX as timer 2 external capture signal input port

T2MOD (C9H) Timer2 Operating Mode Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	-	-	-	-	-	-	T2OE	DCEN
R/W	-	-	-	-	-	-	R/W	R/W
POR	x	x	x	x	x	x	0	0

Bit Number	Bit Symbor	Instruction
1	**T2OE**	Timer 2 output allow bit 0: Set T2 as clock input or I/O port 1: Set T2 as clock output
0	**DCEN**	Decreasing counting allow bit 0: Prohibits Timer 2 as incremental/decreasing counter, Timer 2 only used as incremental counter 1: Allow Timer 2 as incremental/decreasing timer
7~2	**-**	Reserved

TMCON (8EH) Timer Frequency Control Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	-	-	-	-	-	T2FD	T1FD	T0FD
R/W	-	-	-	-	-	R/W	R/W	R/W
POR	x	x	x	x	x	0	0	0

Bit Number	Bit Symbor	Instruction
2	**T2FD**	T2 input frequency selection control 0: T2 frequency comes from $f_{SYS}/12$ 1: T2 frequency comes from f_{SYS}

IE (A8H) Interrupt Enable Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	EA	EADC	ET2	EUART	ET1	-	ET0	EINT0
R/W	R/W	R/W	R/W	R/W	R/W	-	R/W	R/W
POR	0	0	0	0	0	x	0	0

Bit Number	Bit Symbor	Instruction
5	**ET2**	Timer2 interrupt enabling control 0: Disable TIMER2 interrupt 1: Enable TIMER2 interrupt

IP (B8H) Interrupt Priority Control Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	-	IPADC	IPT2	IPUART	IPT1	IPINT1	IPT0	IPINT0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	x	0	0	0	0	0	0	0

Bit Number	Bit Symbor	Instruction
5	**IPT2**	Timer2 interrupt priority 0: Configure Timer2 interrupt priority as ”low ” 1: Configure Timer2 interrupt priority as ”high”

11.2 T2 OPERATING MODE

The operating mode and configuration mode of Timer 2 are shown in the table below:

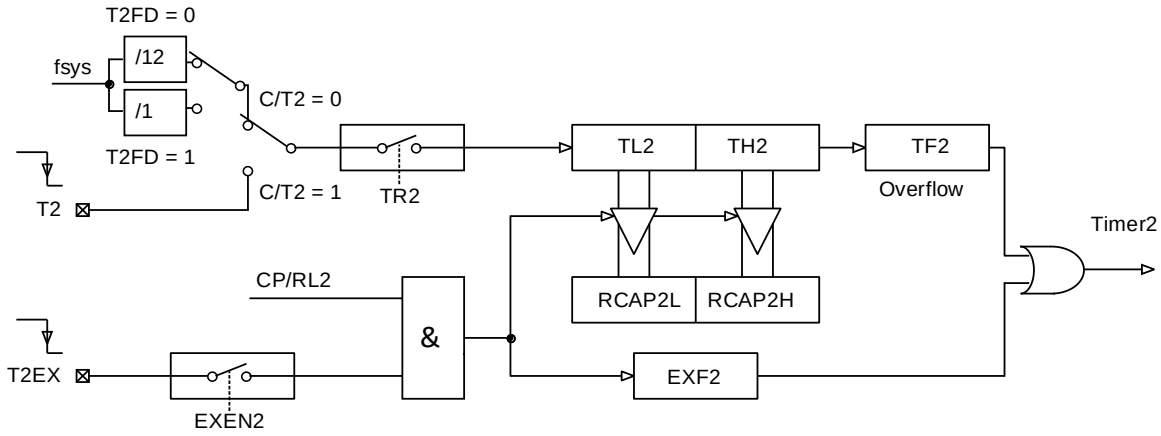
C/$\overline{T2}$	T2OE	DCEN	TR2	CP/$\overline{RL2}$	RCLK	TCLK	The way	
X	0	X	1	1	0	0	0	16-bit capture 16-bit capture 16-bit capture
X	0	0	1	0	0	0	1	16-bit automatic reload timer
X	0	1	1	0	0	0		
X	0	X	1	X	1	X	2	Baud Rate Generator
					X	1		
0	1	X	1	X	0	0	3	Only used for programmable clock
					1	X	3	Programmable clock output with baud rate generator
					X	1		
X	X	X	0	X	X	X	X	Timer 2 stops, but T2EX channel is also available
1	1	X	1	X	X	X		Not recommended

Operating Mode 0: 16-bit capture

In capture mode, there are two options for EXEN2 bit in T2CON.

If EXEN2 = 0, Timer 2 is taken as 16-bit timer or counter; if ET2 is set to 1,Timer2 will set up TF2 and generate an interrupt when Timer 2 overflows.

If EXEN2=1, conduct the same operations as above on Timer 2, the falling edge signal on external input T2EX can make current value in TH2 and TL2 captured into RCAP2H and RCAP2L. Besides, the falling edge signal on T2EX can also cause EXF2 in T2CON to be configured as 1. If ET2 is set to 1, bit EXTF2 will also trigger an interrupt.



Operating Mode 0: 16-bit capture

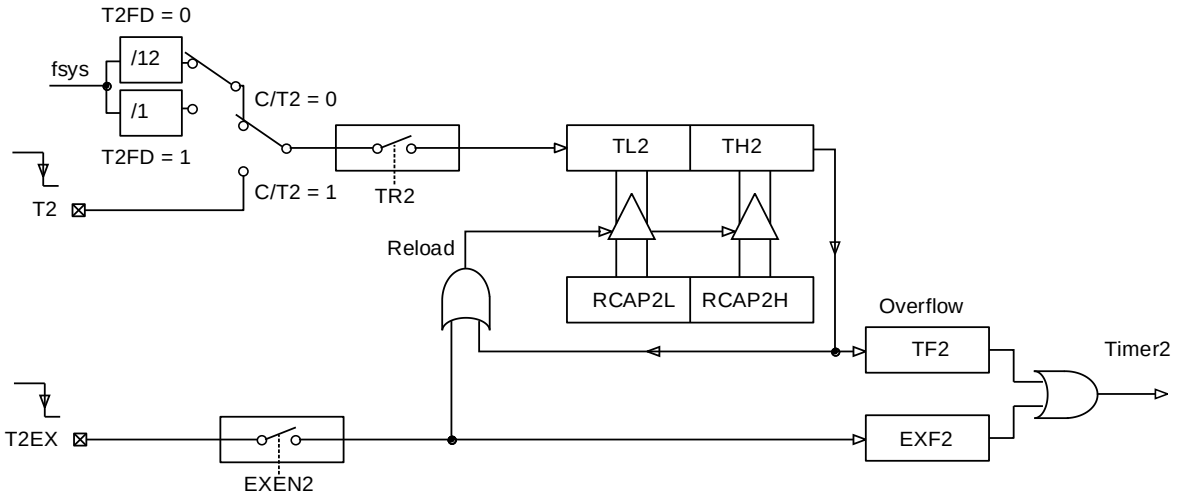
Operating Mode 1: 16-bit Automatic Reload Timer

In the 16-bit automatic reload mode, Timer 2 can be selected to work in incrementing or decreasing counting mode. This function can be selected by DCEN bit in T2MOD (decreasing counting allowed). After system reset, the reset value of DCEN bit is 0 and Timer 2 is defaulted as decreasing counting. When setting DCEN in 1, the incrementing or decreasing counting depends on the level of T2EX pin.

When DCEN = 0, The EXEN2 bit of T2CON has two options to configured.

If EXEN=0, Timer 2 will increase to 0xFFFFH and set TF2 bit after overflow . Meanwhile, the timer will load 16-bit value in registers RCAP2H and RCAP2L written by user software into registers TH2 and TL2 automatically.

If EXEN2=1, both the overflow and the falling edge signal on external input T2EX can trigger a 16-bit count value reloading and set EXF2 bit. If T2 interrupt is enabled (ET2=1), both TF2 and EXF2 bit can generate an interrupt.



Operating Mode 1: 16-bit Automatic Reload Timer DCEN = 0

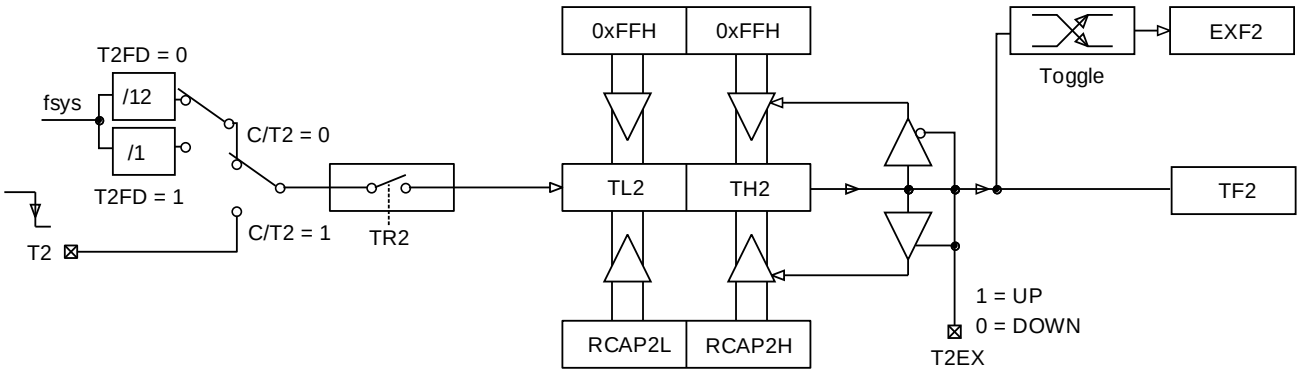
Configure CEN bit to allow Timer 2 for incremental or decreasing counting. When DCEN=1, T2EX pin controls the count direction, but the control of EXEN2 becomes invalid.

Setting T2EX as 1 can conduct incremental count on Timer 2. The Timer overflows when it increases to 0xFFFFH,

then it sets TF2 bit. Besides, the overflow can also respectively cause 16-bit value in RCAP2H and RCAP2L to be reloaded into timer registers.

Setting T2EX as 0 can conduct decreasing count on Timer 2. When the value in TH2 and TL2 is equal to that of RCAP2H and RCAP2L, the timer overflows. TF2 bit will be set up and 0xFFFFH reloaded into timer register.

No matter whether timer 2 overflows or not, bit EXF2 will be used as the 17th bit of the results. Under such operating mode, EXF2 is no longer taken as interrupt flag.



Operating Mode 1: 16-bit Automatic Reload Timer DCEN = 1

Operating Mode 2: Baud Rate Generator

Configure TCLK and RCLK in T2CON register to select Timer 2 as baud rate generator. The baud rate of receiver and transmitter can be different. If Timer 2 is taken as baud rate generator of receiver or transmitter, Timer 1 will be taken as baud rate generator of another.

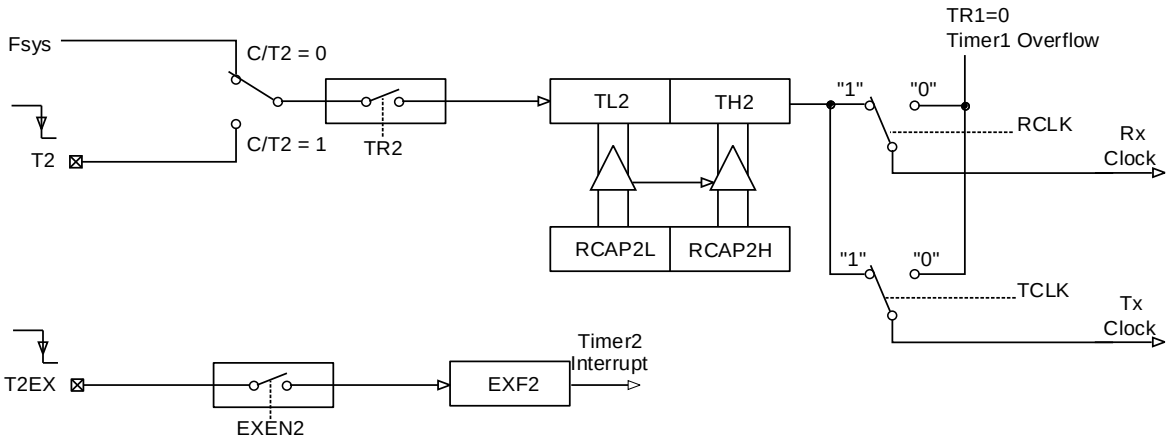
Configure TCLK and RCLK in T2CON register to make Timer 2 in baud rate generator mode. such mode is similar to automatic reload mode

Overflow of Timer 2 can make the value in registers RCAP2H and RCAP2L reloaded into the Timer 2 and counting, but no interrupt will occur.

The baud rate of UART mode 1 and mode 3 depends on overflow rate of Timer 2 based on the following formula:

$$\text{BaudRate} = \frac{\text{fsys}}{[\text{RCAP2H,RCAP2L}]}$$; (note: [RCAP2H, RCAP2L] must be larger than 0x0010)

The schematic diagram of Timer 2 as baud rate generator is shown as follows:



Mode 2: Baud Rate Generator

Operating Mode 3: Programmable Clock Output

In this mode, T2(P1.1) can be programmed to output a 50% duty ratio clock: when C/$\overline{T2}$ = 0 and T2OE = 1, Timer 2 is taken as clock generator

In this mode, duty ratio of T2 output clock is 50%

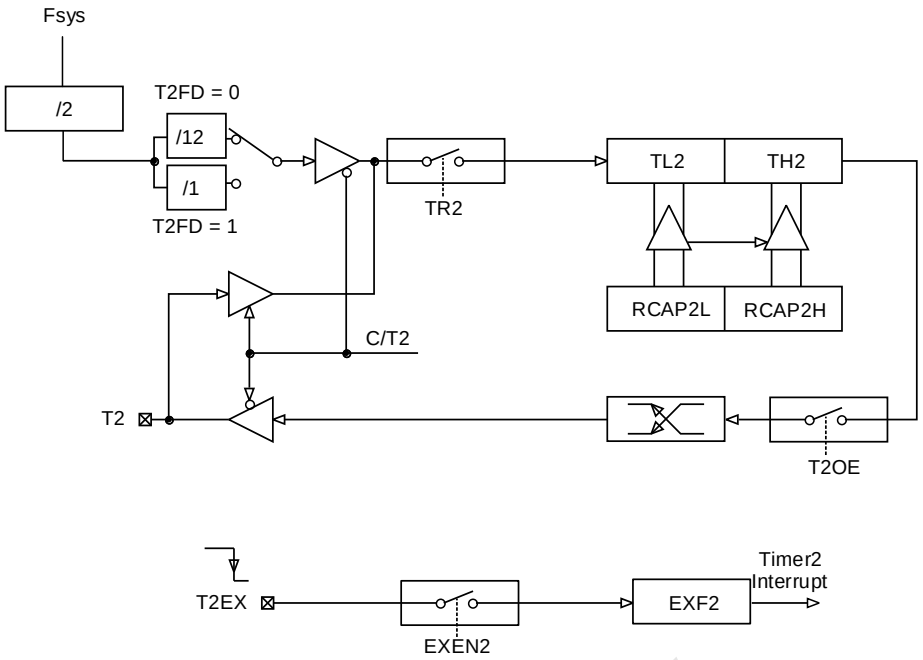
$$\text{Colck Out Frequency} = \frac{\text{fn2}}{(65536-[\text{RCAP2H,RCAP2L}])\times 4};$$

Including, fn2 is the clock frequency of Timer 2

$$\text{fn2} = \frac{\text{fsys}}{12}; \quad \text{T2FD = 0}$$

$$\text{fn2} = \text{fsys}; \quad \text{T2FD = 1}$$

Overflow of Timer 2 does not generate an interrupt.T2 pin is taken as clock output.



Operating Mode 3: Programmable Clock Output

Note:

1. Both TF2 and EXF2 can generate interrupt request of Timer 2, both of which has the same interrupt vector;

2. F2 and EXF2 can be set by software, only software and hardware reset can clear TF2 and EXF2;

3. When EA = 1 and ET2 = 1, setting up TF2 or EXF2 as 1 can arouse interrupt of Timer 2;

4. When Timer 2 is taken as baud rate generator, the value writted in TH2/TL2 or RCAP2H/RCAP2L may influence the accuracy of baud rate and thus result in error of communication.

12 PWM

The SC92F8003 provides a 10-bit PWM output with 7 shared cycles and individually adjustable duty cycles: PWM0~6.

SC92F8003 PWM has the following functions:

① 10-bit precision;

② PWMn (n=0~6) shared the same clock cycle, but the duty cycle can be configured separately;

③ Output can be configured in forward or reverse direction;

④ Independent mode and complementary mode:

1) In independent mode, PWM0~6 shared the same clock cycle, but the duty cycle of each PWM channel can be configured separately;

2) In complementary mode, three pairs(PWM0/3,PWM1/4,PWM2/5) of complementary PWM waveform with dead zone can be output simultaneously;

⑤ Provide one PWM overflow interrupt.

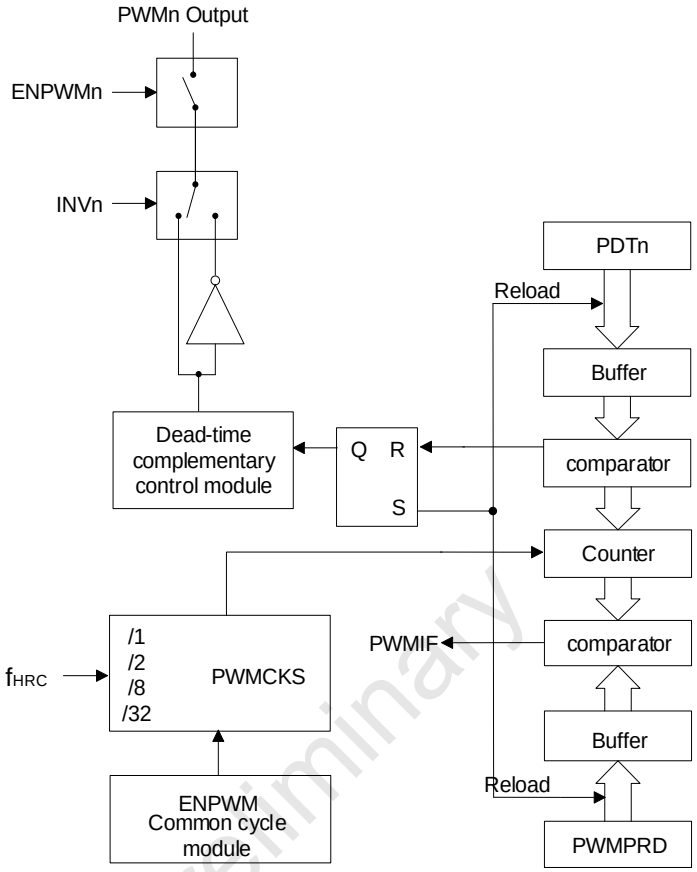
⑥ PWM2 output can be switched to P2.6 or P1.4 by PWMCON0[2];

⑦ PWM5 output can be switched to P1.2 or P2.1 by PWMCON0[3];

The PWM of SC92F8003 can support cycle and duty cycle adjustment. Register PWMCON0 and PWMCON1 control PWMn related settings, PWMCFG sets the polarity of PWM output waveform, PWMPRD sets the common period of 7 PWMs, and PDTn controls the duty cycle of PWMn.

Note: The clock source of the PWM circuit is fixed at fHRC = 16MHz and will not change with the switching of the internal and external system clocks.

12.1 PWM BLOCK DIAGRAM



SC92F8003 PWM block Diagram

12.2 PWM-RELATED SFR REGISTERS

Symbol	address	Instruction	7	6	5	4	3	2	1	0	Reset value
PWMCFG	D1H	PWM Configuration Register	-	INV6	INV5	INV4	INV3	INV2	INV1	INV0	x0000000b
PWMCON0	D2H	PWM Control Register 0	ENPWM	PWMIF	PWMCKS[1:0]		PWMOS5	PWMOS2	PWMPRD[1:0]		00000000b
PWMPRD	D3H	PWM Period Setting Register	PWMPRD[9:2]								00000000b
PWMDTYA	D4H	PWM Duty cycle Configuration Register A	PDT3[1:0]		PDT2[1:0]		PDT1[1:0]		PDT0[1:0]		00000000b
PWMDTY0	D5H	PWM0 Duty Ratio Configuration Register	PDT0[9:2]								00000000b
PWMDTY1	D6H	PWM1 Duty Ratio Configuration Register	PDT1[9:2]								00000000b
PWMDTY2	D7H	PWM2 Duty Ratio Configuration Register	PDT2[9:2]								00000000b

PWMCON1	DAH	PWM Control Register 1	PWMMOD	ENPWM6	ENPWM5	ENPWM4	ENPWM3	ENPWM2	ENPWM1	ENPWM0	00000000b
PWMDTYB	DBH	PWM Duty Ratio Configuration Register B	-	-	PDT6[1:0]		PDT5[1:0]		PDT4[1:0]		xx000000b
PWMDTY3	DCH	PWM3 Duty Ratio Configuration Register /PWM dead time configuration register	PDT3[9:2]								00000000b
			PDF[3:0]				PDR[3:0]				
PWMDTY4	DDH	PWM4 Duty Ratio Configuration Register	PDT4[9:2]								00000000b
PWMDTY5	DEH	PWM5 Duty Ratio Configuration Register	PDT5[9:2]								00000000b
PWMDTY6	DFH	PWM6 Duty Ratio Configuration Register	PDT6[9:2]								00000000b

12.3 PWM UNIVERSAL CONFIGURATION REGISTER

The SC92F8003 PWM woking mode is divided into independent mode and complementary mode. The registers shared by these two modes are as follows:

User can select PWM clock source from 4 options by configuring PWMCKS[1:0]. INV0~6 is used to select if PWM0~6 output is in reverse direction. PWMPRD [9:0] is the seven-channel PWM shared period setting controller. When PWM counter counts to the preset value of PWMPRD[9:0], this counter will skip to 00_h when next PWM CLK comes. That is to say, PWM0-6 period is (PWMPRD[9:0] + 1)*PWM clock.

Note: To ensure **write data correctly, writing operation to PWM period register must follow the sequence of first low 2 bits followed by high 8 bits.**

IE1 (A9H) Interrupt Enabled Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	-	-	-	-	EINT2	EBTM	EPWM	ESSI
R/W	-	-	-	-	R/W	R/W	R/W	R/W
POR	x	x	x	x	0	0	0	0

Bit Number	Bit Symbor	Instruction
1	**EPWM**	Enable PWM Interrupt Control clear to Disable PWM interrupt set to Enable Interrupt when PWM counter overflows

IP1 (B9H) Interrupt Priority Register 1 (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	-	-	-	-	IPINT2	IPBTM	IPPWM	IPSSI
R/W	-	-	-	-	R/W	R/W	R/W	R/W
POR	x	x	x	x	0	0	0	0

Bit Number	Bit Symbor	Instruction
1	**IPPWM**	PWM interrupt priority select clear to configure PWM interrupt priority as “Low” set to configure PWM interrupt priority as “High”

PWMCFG (D1H) PWM Configuration Register (Read/Write)

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Symbor	-	INV6	INV5	INV4	INV3	INV2	INV1	INV0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	x	0	0	0	0	0	0	0

Bit Number	Bit Symbor	Instruction
6~0	**INVx** **（x=0~6）**	PWMx output reverse control 0: PWMx output not invert 1: PWMx output reverse
7	**-**	Reserved

PWMCON0 (D2H) PWM Control Register (Read/Write)

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Symbor	ENPWM	PWMIF	PWMCKS[1:0]		PWMOS5	PWMOS2	PWMPRD[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
POR	0	0	0	0	0	0	0	0

Bit Number	Bit Symbor	Instruction
7	**ENPWM**	PWM module switch control (Enable PWM) **0: PWM unit stops operating and PWM counter resets to zero. PWMn still connects to output pin. If using other functions multiplexed with PWMn output pin, set ENPWMn to 0** 1: Enable Clock to enter PWM unit and PWM starts to work
6	**PWMIF**	PWM interrupt Flag When PWM counter overflows (that is to say, the figure exceeds PWMPRD), this bit will be automatically set to 1 by hardware. If at this time IE1[1] (EPWM) is set to 1 as well, PWM interrupt occurs. Note: Seven PWMs share the same period and the same PWM interrupt vector.
5~4	**PWMCKS[1:0]**	PWM ClocK source Selector 00: f_{HRC} 01: $f_{HRC}/2$ 10: $f_{HRC}/8$ 11: $f_{HRC}/32$ **Note: The clock source of the PWM circuit is fixed at fHRC = 16MHz and will not change with the switching of the internal and external system clocks.**
3	**PWMOS5**	PWM5 output port selection:

		0: PWM5 output to P1.2 1: PWM5 output to P2.1
2	**PWMOS2**	PWM2 output port selection: 0: PWM2 output to P2.6 1: PWM2 output to P1.4
1~0	**PWMPRD[1:0]**	PWM0 ~ PWM6 shared period configuration low 2 bit; This figure represents Period of PWM0 ~ PWM6 output waveform subtract 1; that is to say, period of PWM output is (PWMPRD[9:0] + 1)* PWM clock;

PWMPRD (D3H) PWM Period Configuration Register (Read/Write)

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Symbor	**PWMPRD[9:2]**							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit Number	Bit Symbor	Instruction
7~0	**PWMPRD[9:2]**	PWM0 ~ PWM6 shared period configuration high 8 bit; This figure represents Period of PWM0 ~ PWM6 output waveform subtract 1; that is to say, period of PWM output is (PWMPRD[9:0] + 1)* PWM clock;

PWMCON1 (DAH) PWM Configuration Register (Read/Write)

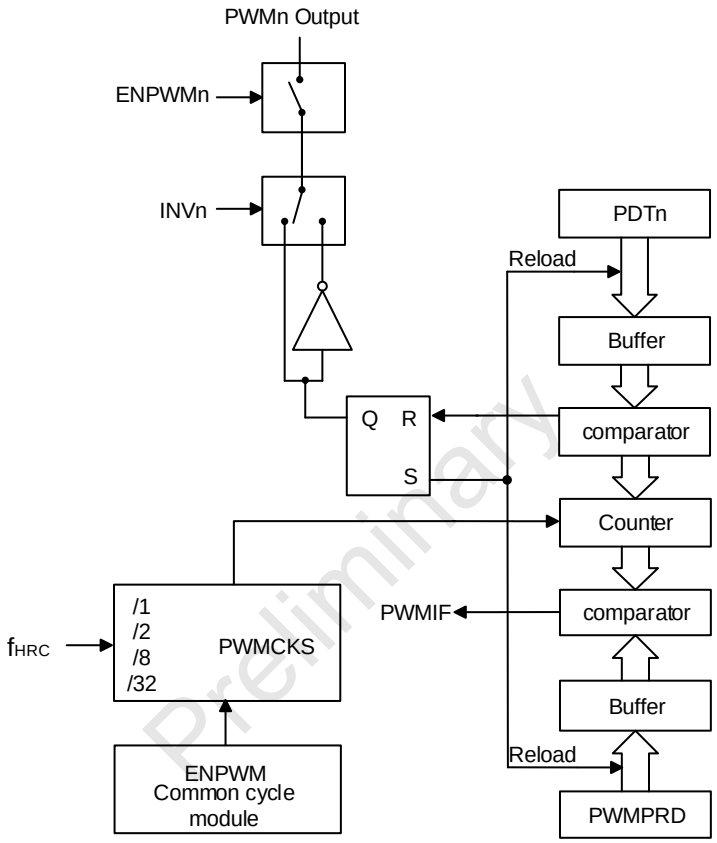
Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Symbor	PWMMOD	ENPWM6	ENPWM5	ENPWM4	ENPWM3	ENPWM2	ENPWM1	ENPWM0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit Number	Bit Symbor	Instruction
7	**PWMMOD**	PWM Mode Configuration: 0: Independent Mode: PWM0~6 , 7-channel PWM of Independent setting; 1: Complementary Mode: PWM0/3, PWM1/4, PWM2/5. the output pulse width of the same group of PWM is the same, which is controlled by PDT0~2[9:0] separately and dead zone time is configured by register PDT3. In the complementary mode, the duty of PWM6 is still set independently.
6~0	**ENPWMx** **（x=0~6）**	PWMx Functional Switch 0: PWMx do not output to IO 1: PWMx output to IO Note: If setting ENPWM as 1, PWM module is enabled. But if ENPWMn=0, PWM output is closed and used as GPIO At this time, PWM module can be used as a 10-bit Timer, and PWM will still generate interrupt if EPWM(IE1.1) is set to 1.

12.4 PWM INDEPENDENT MODE

In independent mode (PWMMOD = 0), PWMDTY0~6, PWMDTYA and PWMDTYB can be used as duty ratio configuration registers of PWM0~6. User shall configure PWM output status and period and corresponding duty ratio registers of PWM channel to output PWM waveform with fixed duty ratio output.

12.4.1 PWM INDEPENDENT MODE DIAGRAM



SC92F8003 PWM Independent Mode Block Diagram

12.4.2 PWM INDEPENDENT MODE DUTY RATIO CONFIGURATION

To guarantee write data correctly, writing operation to PWM duty ratio registers must follow the sequence of first low 2-bit followed by high 8-bit.

PWMDTY0(D5H) PWM0 Duty Ratio Configuration Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	PDT0[9:2]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

PWMDTY1(D6H) PWM1 Duty Ratio Configuration Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0

Bit Symbor	PDT1[9:2]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

PWMDTY2(D7H) PWM2 Duty Ratio Configuration Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	PDT2[9:2]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

PWMDTY3(DCH) PWM3 Duty Ratio Configuration Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	PDT3[9:2]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

PWMDTY4(DDH) PWM4 Duty Ratio Configuration Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	PDT4[9:2]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

PWMDTY5(DEH) PWM5 Duty Ratio Configuration Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	PDT5[9:2]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

PWMDTY6(DFH) PWM6 Duty Ratio Configuration Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	PDT6[9:2]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit Number	Bit Symbor	Instruction
7~0	**PDTx[9:2]** **（x=0~6）**	Independent Mode: PWMx duty ratio length configuration of high 8 bits; High level width of PWMx is (PDTx[9:0]) PWM clocks.

PWMDTYA(D4H) PWM Duty Ratio Configuration Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	PDT3[1:0]		PDT2[1:0]		PDT1[1:0]		PDT0[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

PWMDTYB(DBH) PWM Duty Ratio Configuration Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	-	-	PDT6[1:0]		PDT5[1:0]		PDT4[1:0]	
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
POR	x	x	0	0	0	0	0	0

Bit Number	Bit Symbor	Instruction
7~0	**PDTx [1:0]** **（x=0~6）**	PWMx duty ratio length configuration of low 2 bits; High level width of PWMx is (PDTx[9:0]) PWM clocks.

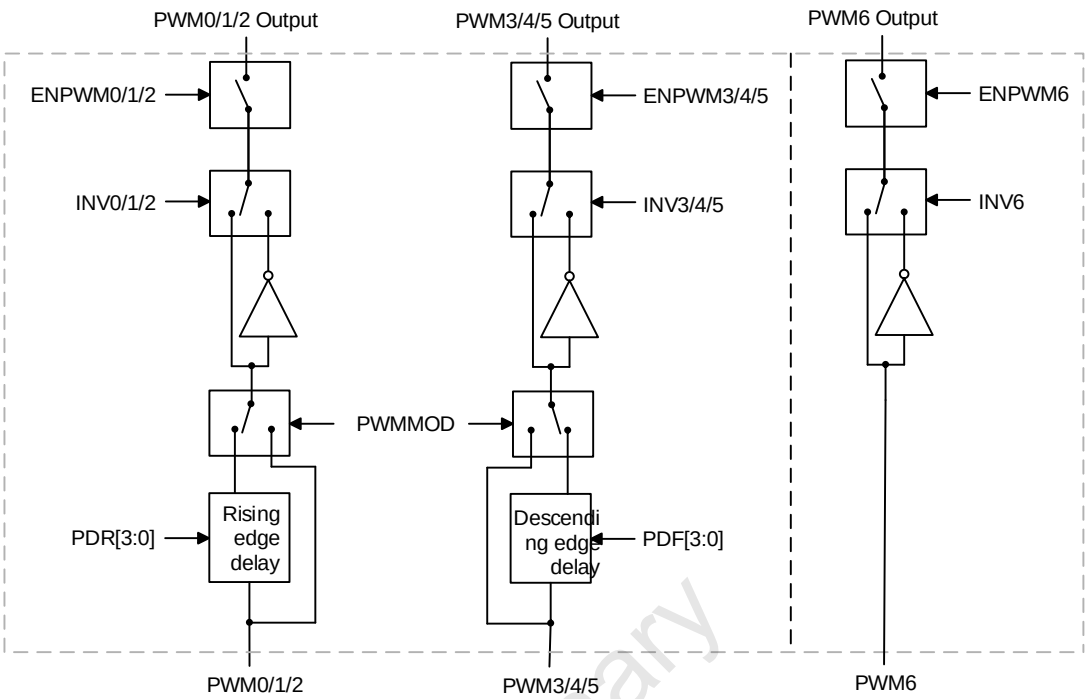
12.5 PWM COMPLEMENTARY MODE

When SC92F8003B PWM works in complementary mode, the dead-time control module can prevent the overlap of valid time zones between two complementary output channel PWM, so as to guarantee that a pair of complementary power switches tube driven by PWM signals will not work at the same time in practical applications.

In complementary mode (PWMMOD = 1), PWM6 is an independent PWM output and the duty ratio is to be adjusted by PDT6[9:0];,PWM0 and PWM3 become one group and the duty ratio is to be adjusted by PDT0[9:0]; PWM1 and PWM4 become one group and the duty ratio is to be adjusted by PDT1[9:0]; PWM2 and PWM5 become one group and the duty ratio is to be adjusted by PDT2[9:0].

In complementary mode, the registers PWMDTY4~5 is invalid, the register PWMDTY3 bit is redefined as PWM3/4/5 falling edge dead zone time control bit PDF[3:0] and PWM0/1/2 rising edge dead zone time control bit PDR[3:0].

12.5.1 PWM COMPLEMENTARY MODE DIAGRAM



SC92F8003 PWM Complementary Mode Block Diagram

12.5.2 PWM DUTY RATIO CONFIGURATION IN COMPLEMENTARY MODE

Note: To guarantee write data correctly, writing operation to PWM duty ratio registers must follow the sequence of first low 2 bits followed by high 8 bits.

PWMDTY0 (D5H) PWM0 Duty Ratio Configuration Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	PDT0[9:2]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

PWMDTY1 (D6H) PWM2 Duty Ratio Configuration Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	PDT1[9:2]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

PWMDTY2 (D7H) PWM2 Duty Ratio Configuration Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	PDT2[9:2]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

PWMDTY6(DFH) PWM6 Duty Ratio Configuration Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	PDT6[9:2]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit Number	Bit Symbor	Instruction
7~0	**PDT0[9:2]**	Complementary model: PWM0 、 PWM3 duty ratio length configuration of high 8 bits High level width of PWM0 and PWM3 is (PDT0x[9:0]) PWM clocks
7~0	**PDT1[9:2]**	Complementary model: PWM1 、 PWM4 duty ratio length configuration of high 8 bits High level width of PWM1 and PWM4 is (PDT1[9:0]) PWM clocks
7~0	**PDT2[9:2]**	Complementary model: PWM2 、 PWM5 duty ratio length configuration of high 8 bits High level width of PWM2 and PWM5 is (PDT2[9:0]) PWM clocks
7~0	**PDT6[9:2]**	Complementary model: PWM6 duty ratio length configuration of high 8 bits High level width of PWM6 is (PDT6[9:0]) PWM clocks

PWMDTYA (D4H) PWM Duty Ratio Configuration Register A (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	PDT3[1:0]		PDT2[1:0]		PDT1[1:0]		PDT0[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

PWMDTYB (DBH) PWM Duty Ratio Configuration Register B (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	-	-	PDT6[1:0]		PDT5[1:0]		PDT4[1:0]	
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
POR	x	x	0	0	0	0	0	0

Bit Number	Bit Symbor	Instruction
7~0	**PDT0[1:0]**	Complementary model: PWM0 、PWM3 duty ratio length configuration of low 2 bits High level width of PWM0 and PWM3 is (PDT0[9:0]) PWM clocks
7~0	**PDT1[1:0]**	Complementary model: PWM1 、PWM4 duty ratio length configuration of low 2 bits High level width of PWM1 and PWM4 is (PDT1[9:0]) PWM clocks
7~0	**PDT2[1:0]**	Complementary model: PWM2 、PWM5 duty ratio length configuration of low 2 bits High level width of PWM2 and PWM5 is (PDT2[9:0]) PWM clocks
7~0	**PDT6[1:0]**	Complementary model: PWM6 duty ratio length configuration of low 2 bits High level width of PWM6 is (PDT6[9:0]) PWM clocks

12.5.3 PWM DEAD ZONE TIME CONFIGURATION IN COMPLEMENTARY MODE

PWMDTY3 (DCH) PWM Dead Zone Time Configuration Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	PDF[3:0]				PDR[3:0]			
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit Number	Bit Symbor	Instruction
7~4	**PDF[3:0]**	Complementary model: PWM3/4/5 falling edge dead zone time = PDF [3:0]/f_{OSC}
3~0	**PDR[3:0]**	Complementary model: PWM0/1/2 rising edge dead zone time = PDR [3:0]/f_{OSC}

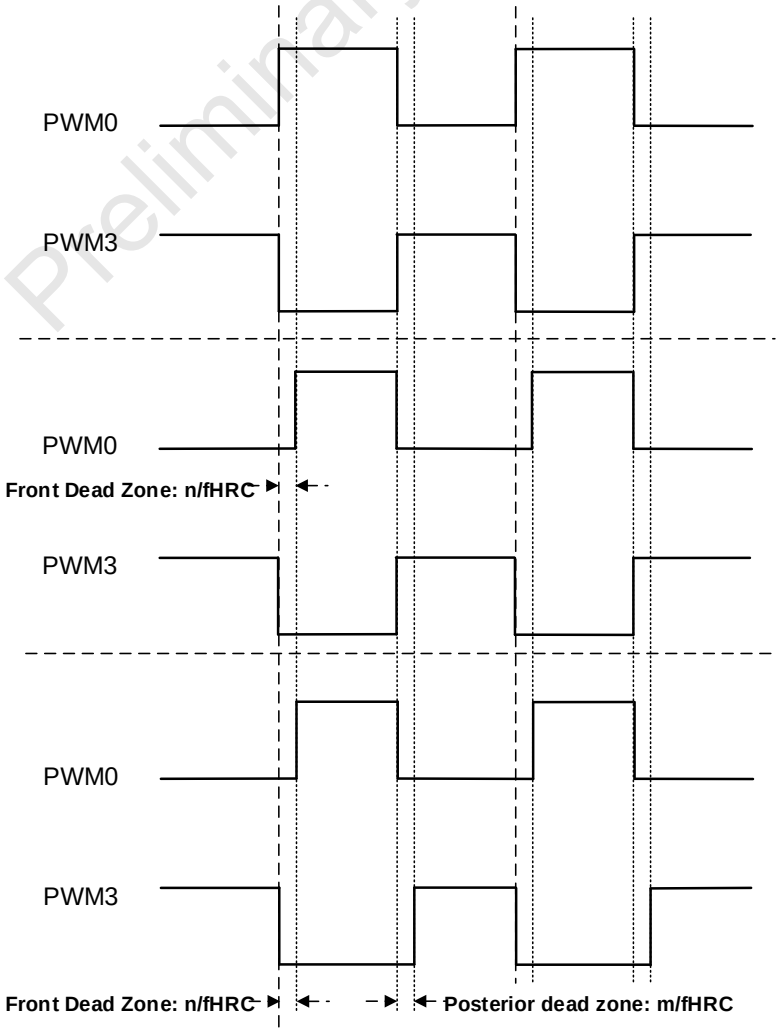
12.5.4 PWM DEAD-TIME OUTPUT WAVEFORM

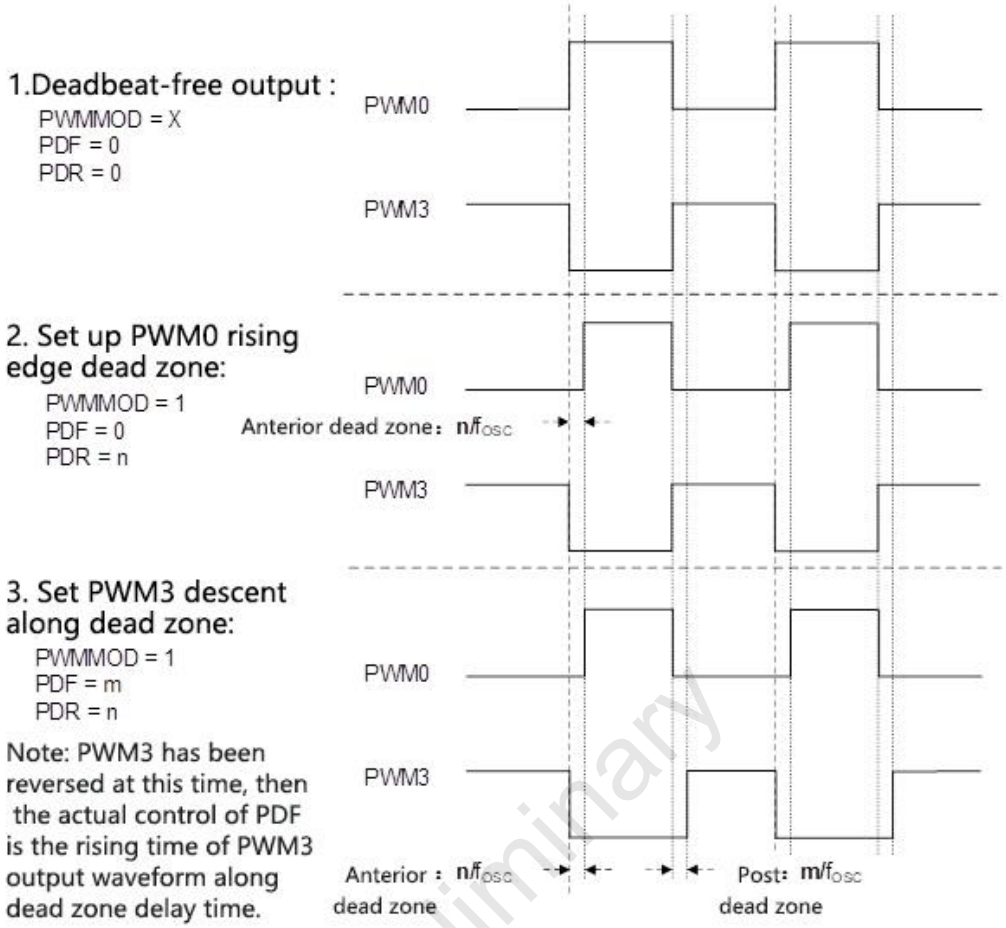
Below is the waveform diagram for PWM0 and PWM3 adjusted by dead time under complementary mode. To be better distinguish able, PWM3 is in reverse direction (INV3=1).

1. No dead-time output:
PWMMOD = X
PDF = 0
PDR = 0

2.Setting up PWM0 Rising Edge Dead Zone：
PWMMOD = 1
PDF = 0
PDR = n

3.Setting PWM3 descent along dead zone：
PWMMOD = 1
PDF = m
PDR = n
Note: PWM3 has been reversed at this time, then the PDF corresponding control is actually the rising time of PWM3 output waveform along dead zone delay time.



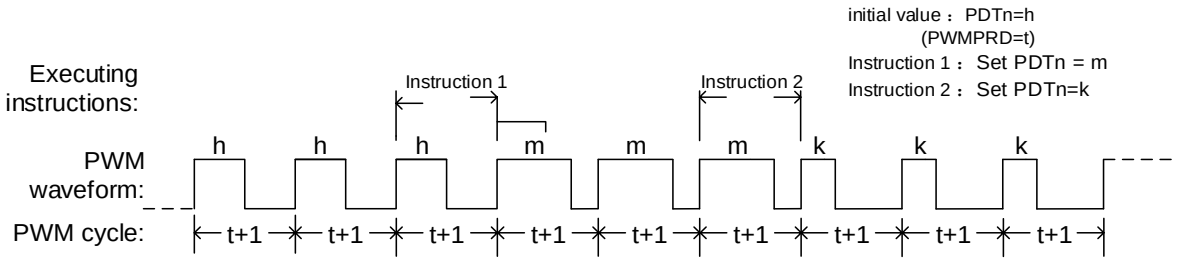


Waveform of PWM Output with Dead Zone

12.6 PWM WAVEFORM AND DIRECTIONS

The influence of changing various SFR parameters on PWM waveform is shown as follows:

① DIAGRAM FOR DUTY RATIO CHANGE FEATURES



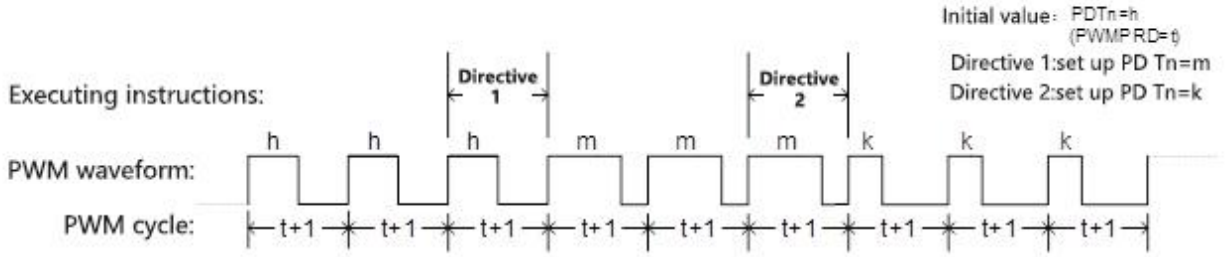
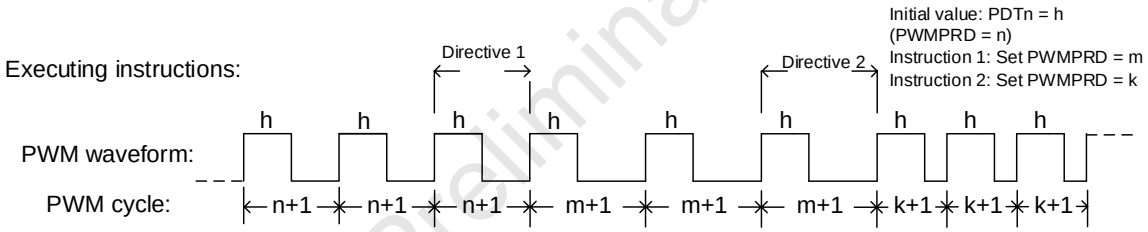


Diagram for Duty ratio Change Features

When PWMn outputs waveform, if it is required to change the duty ratio, users can change the value of high level configuration registers (PDTn). But note that changing the value of PDTn will not change the duty ratio immediately. It is required to wait for the end of this period and change in the next period . To guarantee write data correctly, writing operations to PWM period and DUTY registers must follow the sequence of first low 2 bits followed by high 8 bits. Related waveform output is shown in the figure above.

② PERIOD CHANGE FEATURES



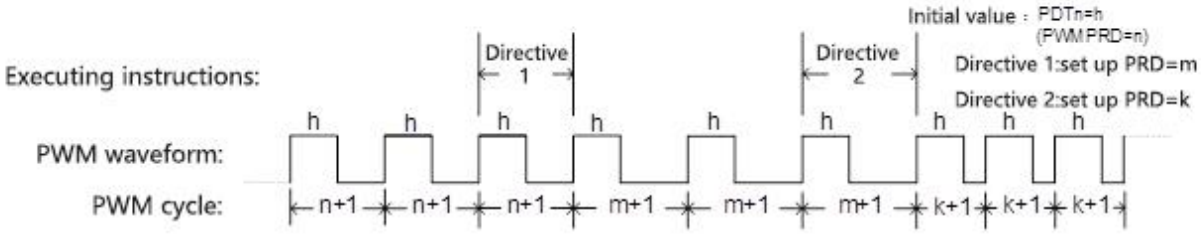


Diagram for Period Change Features

When PWMn outputs waveform, if it is required to change the period, users can change the value of period configuration registers PWMPRD. Same as changing the duty ratio, change the value of PWMPRD and the period will not be changed immediately. It is required to wait for the end of this period and change in the next period. see the figure above for reference.

③ RELATIONSHIP BETWEEN PERIOD AND DUTY RATIO

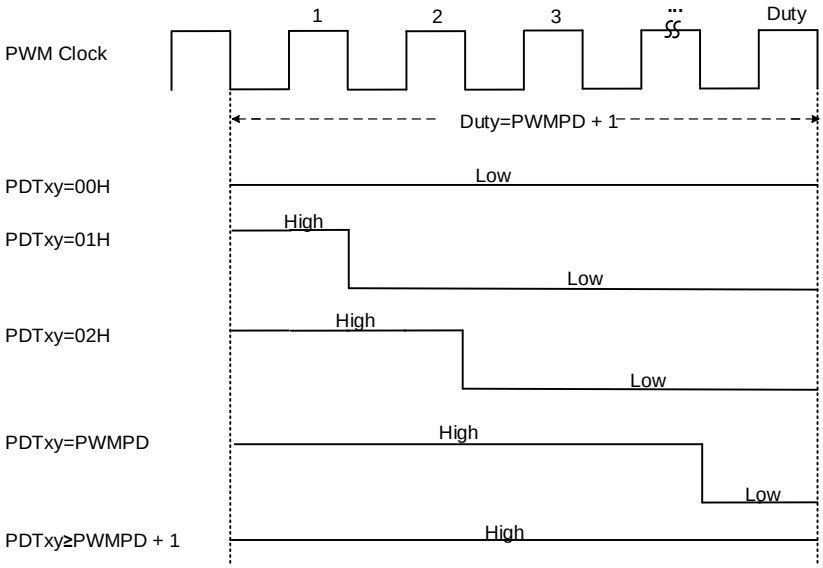


Diagram for Relationship between Period and Duty ratio

The relationship between period and duty ratio is shown in the figure above. The precondition of this result is the PWMn output reverse control (INVn) is initialized as 0; if it is required to get the contrary result, set INVn as 1.

13 GP I/O

SC92F8003 offers up to 18 two-way controllable GPIOs, input and output control registers are used to control the input and output state of various ports, when the port is used as input, each I/O port is equipped with internal pull-high resistor controlled by PxPHy. Such 18 IOs are shared with other functions.Under output state, I/O port read from the value of port data register is the actual state value of the port.

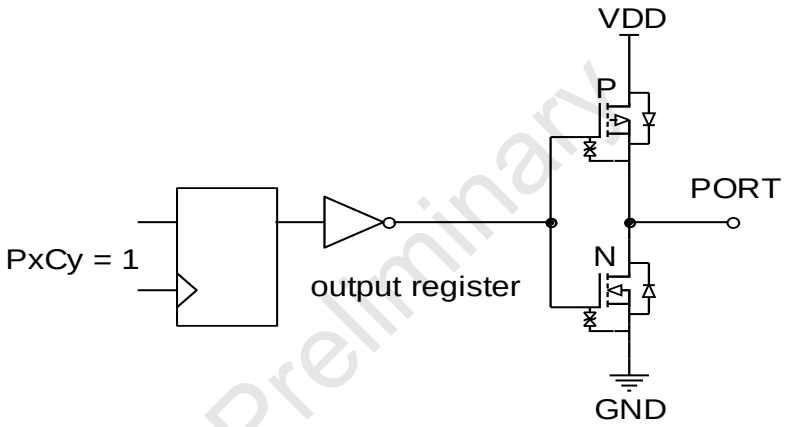
Note: Unused IO port or IO port with no package pin shall be configured as strong push-pull output mode.

13.1 GPIO STRUCTURE DIAGRAM

Strong Push-pull Output Mode

In strong push-pull output mode, it is able to provide continuous high current drive: high output for the current larger than 20mA and low output for the current larger than 70mA

The port structure diagram for strong push-pull output mode is shown below:

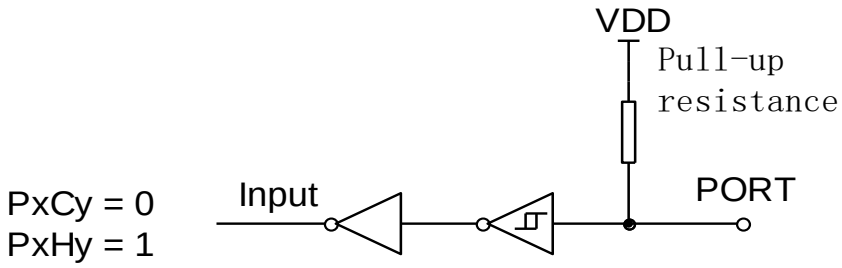


Strong Push-pull Output Mode

Pull-high Input Mode

In pull-high input mode, a pull-high resistor is connected on the input port, only when the level on the input port is pulled down, low level signal can be detected.

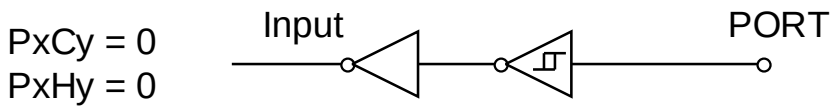
The port structure diagram forpull-high input mode is shown below:



Pull-up Input Mode

Input only Mode (Input only)

The port structure diagram for input only mode is shown below:



High- impedance Input Mode

13.2 I/O PORT-RELATED REGISTERS

P0CON (9AH) P0 Input/Output Control Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	-	-	-	-	-	-	P0C1	P0C0
R/W	-	-	-	-	-	-	R/W	R/W
POR	x	x	x	x	x	x	0	0

P0PH (9BH) P0 Pull-up Resistor Control Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	-	-	-	-	-	-	P0H1	P0H0
R/W	-	-	-	-	-	-	R/W	R/W
POR	x	x	x	x	x	x	0	0

P1CON (91H) P1 Input/Output Control Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	P1C7	P1C6	P1C5	P1C4	P1C3	P1C2	P1C1	P1C0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

P1PH (92H) P1 Pull-up Resistor Control Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	P1H7	P1H6	P1H5	P1H4	P1H3	P1H2	P1H1	P1H0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

P2CON (A1H) P2 Input/Output Control Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	P2C7	P2C6	P2C5	P2C4	P2C3	P2C2	P2C1	P2C0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

P2PH (A2H) P2 Pull-up Resistor Control Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	P2H7	P2H6	P2H5	P2H4	P2H3	P2H2	P2H1	P2H0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit Number	Bit Symbor	Instruction
7~0	**PxCy** **（x=0~2, y=0~7）**	Px port input and output control: 0: Pxy as input mode (initial value) 1: Pxy as strong push-pull output mode
7~0	**PxHy** **（x=0~2, y=0~7）**	Px port pull-up resistance configuration, only valid when PxCy=0: 0: Pxy as high-impedance input mode (initial value), pull-up resistance is switched off; 1: Pxy pull-up resistance on

P0 (80H) P0 Data Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	-	-	-	-	-	-	P0.1	P0.0
R/W	-	-	-	-	-	-	R/W	R/W
POR	x	x	x	x	x	x	0	0

P1 (90H) P1 Data Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

P2 (A0H) P2 Data Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

14 UART0

SC92F8003 supports a full-duplex serial port. It is convenient for connecting other device or equipment, for example, Wifi module circuit or other drive chips with UART communication interface. UART0 functions and features are shown below:

1. The serial communication port can be switched to different pins through SPOS[1:0];
2. Three kinds of communication mode : Mode 0, Mode 1 and Mode 3;
3. Configure Timer 1 or Timer 2 as baud rate generator;
4. Completion of transmission and reception can generate interrupt RI/TI, and such interrupt flag requires software clearup.

14.1 UART0

OTCON (8FH) Output Control Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	SSMOD[1:0]		SPOS[1:0]		-	-	-	-
R/W	R/W	R/W	R/W	R/W	-	-	-	-
POR	0	0	0	0	x	x	x	x

Bit Number	Bit Symbor	Instruction
5~4	**SPOS[1:0]**	**Serial communication port output pin selection** X0: UART0 shares pins with P1.5 and P1.6; X1: UART0 shares pins with P1.1 and P2.0;

SCON (98H) Serial Port Control Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit Number	Bit Symbor	Instruction
7~6	**SM0~1**	Serial Communication Mode Control Bit 00: Mode 0, 8-bit half-duplex synchronous communication mode. Serial data is received and transmitted on RX pin. TX pin is used to transmit shift clock. Receive and transmit 8 bits for each frame, and low bit will be received or transmitted firstly; 01: Mode 1, 10-bit full-duplex asynchronous communication composing of 1 starting bit, 8 data bit and 1 stopping bit, with communication baud rate changeable; 10: Reserved; 11: Mode 3, 11-bit full-duplex asynchronous communication, composing of 1 starting bit, 8 data bit and 1 programmable 9th bit and 1 stopping bit, with communication baud rate changeable.
5	**SM2**	Serial communication mode control bit 2, this control bit is only valid for mode 3 0: RI is set upon receiving a complete data frame to generate interrupt request; 1: When receiving a complete data frame, only when RB8=1, will RI be set to generate interrupt request.
4	**REN**	Receive allowing control bit 0: Receiving data not allowed; 1: Receiving data allowed.
3	**TB8**	Only valid for mode 3, 9th bit of receiving data
2	**RB8**	Only valid for mode 3, 9th bit of receiving data
1	**TI**	Transmission interrupt flag bit
0	**RI**	Reception interrupt flag bit

SBUF (99H) Serial Data Cache Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	SBUF[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit Number	Bit Symbor	Instruction
7~0	**SBUF[7:0]**	**Serial Port Data Cache Register** SBUF contains two registers: one for transmitting shift register and one for receiving latch. data written into SBUF will be transmitted to shift register and initiate transmitting process. reading SBUF will return the contents of receiving latch.

PCON (87H) Power Management Control Register (only readable, * unreadable*)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	SMOD	-	-	-	-	-	STOP	IDL
R/W	W	-	-	-	-	-	W	W
POR	0	x	x	x	x	x	0	0

Bit Number	Bit Symbor	Instruction
7	**SMOD**	**Baud rate multiplying power configuration bit, only valid for mode 0 (SM0~1 = 00)** 0: Serial port operates under clock of 1/12 system clock 1: Serial port operates under clock of 1/4 system clock

14.2 BAUD RATE OF SERIAL COMMUNICATION

In mode 0, baud rate can be programmed as 1/12 or 1/4 of system clock and determined by SMOD(PCON.7) bit . When SMOD is set as 0, the serial port operates in 1/12 of system clock. When SMOD is set to 1, serial port operates in 1/4 of system clock.

In mode 1 and mode 3, user can select overflow rate of Timer 1 or Timer 2 as baud rate by configuration.

Set TCLK(T2CON.4) and RCLK(T2CON.5) bit to configure Timer 2 as TX and RX clock source of baud rate (refer to section timer for details). No matter TCLK or RCLK is set as logic 1, Timer 2 can be in the mode of baud rate generator. If TCLK and RCLk are set as logic 0, Timer1 can be baud clock source of Tx and Rx.

Mode 1 and Mode 3 baud rate formula is shown below, including that [TH1,TL1] is the 16-bit counter registers of Timer 1, and [RCAP2H,RCAP2L] is the 16-bit reload registers of Timer 2.

1. Usinge Timer 1 as baud rate generator, Timer 1 must stop counting, TR1=0:

$$\text{BaudRate} = \frac{\text{fsys}}{[\text{TH1,TL1}]}$$(Note: [TH1,TL1] must be larger than 0x0010**)**

2. Use Timer 2 as baud rate generator:

$$\text{BaudRate} = \frac{\text{fsys}}{[\text{RCAP2H,RCAP2L}]}$$(Note: [RCAP2H,RCAP2L] must be larger than 0x0010**)**

15 SPI/TWI/UART THREE-CHOICE ONE SERIAL INTERFACE(SSI)

SC92F8003 integrates SPI/TWI/UART serial interface circuits (short for SSI), which is convenient for connecting MCU to devices or equipments with different interfaces. User can configure SSI as any communication mode among SPI, TWI and UART by configuring SSMOD[1:0] bit of register OTCON. Its features are shown below:

1. SPI mode can be configured as master mode or slave mode
2. TWI mode can only be used as slave in communication
3. UART mode can work in Mode 1 (10-bit full duplex asynchronous communication) and Mode 3 (11-bit full duplex asynchronous communication)

Specific configuration modes are shown below:

OTCON (8FH) Output Control Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	SSMOD[1:0]		SPOS[1:0]		-	-	-	-
R/W	R/W	R/W	R/W	R/W	-	-	-	-
POR	0	0	0	0	x	x	x	x

Bit Number	Bit Symbor	Instruction
7~6	**SSMOD[1:0]**	**SSI Communication Mode** 00: SSI OFF 01: SSI is set in SPI communication mode; 10: SSI is set in TWI communication mode; 11: SSI is set in UART communication mode;
5~4	**SPOS[1:0]**	**Serial communication port output pin selection** X0: UART0 shares pins with P1.5 and P1.6; X1: UART0 shares pins with P1.1 and P2.0; 0X: SSI shares pins with P1.0、P2.7 and P2.6; **1X: SSI shares pins with P2.1、P2.2 and P2.3;**

15.1 SPI

SSMOD[1:0] = 01, SSI is configured as SPI interface. Serial Peripheral Interface (short for SPI) is a kind of high-speed serial communication interface, allowing MCU and peripheral equipment (including other MCU) to conduct full duplex synchronous serial communication.

15.1.1 SPI OPERATION-RELATED REGISTERS

SSCON0 (9DH) SPI Control Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbor	SPEN	-	MSTR	CPOL	CPHA	SPR2	SPR1	SPR0
R/W	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	x	0	0	0	0	0	0

Bit Number	Bit Symbor	Instruction
7	**SPEN**	**SPI Enabling Control** 0: Disable SPI 1: Enable SPI
5	**MSTR**	**SPI Master/Slave Selection** 0: SPI as slave equipment 1: SPI as master equipment
4	**CPOL**	**Clock Polarity Control** 0: SCK is low level under idle state 1: SCK is high level under idle state
3	**CPHA**	**Clock Phase Control** 0: First edge collection data of SCK period 1: Second edge collection data of SCK period
2-0	**SPR[2:0]**	**SPI Clock Speed Selection** 000：f_{SYS} /4 001：f_{SYS} /8 010：f_{SYS} /16 011：f_{SYS} /32 100：f_{SYS} /64 101：f_{SYS} /128 110：f_{SYS} /256 111：f_{SYS} /512
6	**-**	Reserved

SSCON1 (9EH) SPI Status Register (Read/Write)

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Symbor	SPIF	WCOL	-	-	TXE	DORD	-	TBIE
R/W	R/W	R/W	-	-	R/W	R/W	-	R/W
POR	0	0	x	x	0	0	x	0

Bit Number	Bit Symbor	Instruction
7	**SPIF**	**SPI Data Transfer Flag** 0: must be cleared by software 1: Data transmission completed and flag is set to 1 by hardware
6	**WCOL**	**Write-in Conflict Flag** 0: clear to 1 by software, indicating write-in conflict is processed 1: flag is set to 1 by hardware, indicating one conflict is detected
3	**TXE**	**Send Buffer Empty Flag** 0: Transmitting buffer not empty 1: Transmitting buffer empty, clear to 0 by software
2	**DORD**	**Transfer Direction configuration** 0: Transmit MSB first 1: Transmit LSB first
0	**TBIE**	**Transmitting Buffer interrupt enable** 0: Transmission interrupt not enable 1: Transmission interrupt enable, when SPIF=1, TBIE=1 will generate SPI interrupt

5~4,1	-	Reserved

SSDAT (9FH) SPI Data Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbol	SPD[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit Number	Bit Symbol	Instruction
7~0	**SPD[7:0]**	**SPI Data Cache Register** Data written to SSDAT is placed in transmitting shift register. Up on reading SSDAT, data from receiving shift register is received

15.1.2 SIGNAL DESCRIPTION

Master Output and Slave Input (MOSI):

This signal connects master device with one slave device. Data is serially transmitted from master device to slave device via MOSI, featuring master device output and slave device input.

Master Input and Slave Output (MISO):

This signal connects slave device with master device. Data is serially transmitted from slave device to master device via MISO featuring slave device output and master device input. When SPI is configured as slave device and is not selected, the MISO pin of slave device is in high resistance state.

SPI Serial Clock (SCK):

SCK signal is used to control synchronous movement of input and output data on MOSI and MISO. Transmit one byte for every 8 clock periods. If no slave device is selected, SCK signal will be ignored from slave device.

15.1.3 OPERATING MODES

SPI can be configured as master mode or slave mode. The configuration and initialization of SPI module can be completed via setting SSCON0 register (SPI control register) and SSCON1 (SPI state register). After completing configuration, data is transmitted by setting SSCON0, SSCON1 and SSDAT (SPI data register).

During SPI communication period, data synchronic is moved in or out serially. Serial clock line (SCK) makes data movement and sampling on two lines of serial data line (MOSI and MISO) synchronous. If any slave device is not selected, it is unable to participate in activities on SPI line.

When SPI master device transmits data to slave device via MOSI, slave device sends data to master device via MISO as response, which realizes synchronous full duplex transmission of data transmitting and receiving at the same clock. Transmitting shift register and receiving shift register use the same special function address, writing operations to SPI data register SSDAT will be written to transmitting shift register and reading operations to SSDAT register will obtain the data of receiving shift register.

SPI port of some device will lead to SS pin (select slave device pin, low effective), when communicating with SC92F8003 SPI, SS pin from other device on SPI bus shall be connected based on different communication modes. The following table lists the connection modes of SS pin of other device on SPI bus under different communication modes of SC92F8003 SPI:

SC92F8003 SPI	Other Device on SPI Bus	Mode	SS of Slave Device (Select Pins from Device)
Master Mode	Slave Mode	One Master One Slave	Pull down
		One Master Multiple Slave	SC92F8003 leads to multiple I/Os, respectively connecting to SS pin of slave device. Before data transmission, SS pin of slave device must be placed in low position
Slave Mode	Master Mode	One Master One Slave	Pull up

Master Mode

- **Mode Startup:**

  Start all data transmission on SPI bus controlled by SPI master device. When MSTR position in SSCON0 register is 1, SPI operates in master mode, only master device can start the transmission.

- **Transmitting:**

  In SPI master mode, write one byte of data to SPI data register SSDAT, the data will write to transmitting shift cache. If any data already exists in transmitting shift register, one WCOL signal will be generated from master SPI to indicate writing is too fast. However, data in transmitting shift register will not be influenced and transmitting will not be interrupted as well. Besides, if the transmitting shift register is empty, the master device will move the data in transmitting shift register to MOSI line according to SPI clock frequency on SCK serially. After transmission, SPIF bit in SSCON1 register will be placed in 1. If SPI interrupt is allowed, when SPIF bit is placed in 1, an interrupt will be generated as well.

- **Receiving:**

  When master device transmits data to slave device via MOSI line, corresponding slave device will also transmit the contents in transmitting shift register to receiving shift register of master Device via MISO line so as to realize full duplex operations. Therefore, setting SPIF flag bit in 1 indicates transmission completion as well as data receiving completion. Data received from slave device is stored in receiving shift register of master device in accordance with MSB or LSB preferred transmission direction. When one byte of data is completely moved to receiving register, the processor can obtain such data by reading SSDAT register.

Slave Mode

- **Mode Startup:**

  When the MSTR bit in SSCON0 register is reset to 0, SPI operates in slave mode.

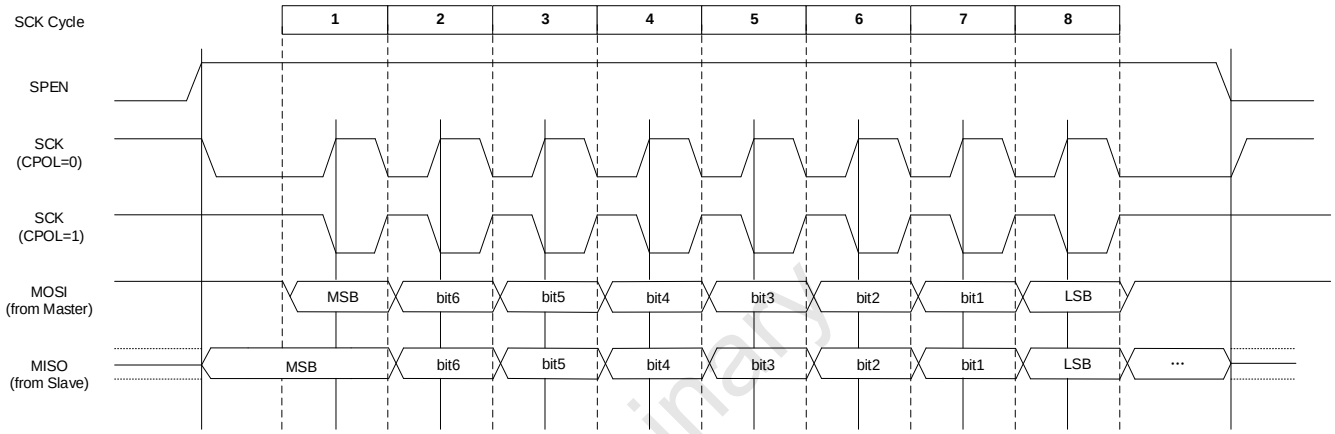
- **Transmitting and Receiving:**

  In slave mode, according to SCK signal controlled by master device, data is moved in via MOSI pin and out via MISO pin. 1-bit counter records the edge number of SCK. When the receiving shift register moves in 8-bit data (one byte) and transmitting shift register moves out 8-bit data (one byte), SPIF flag is placed in 1. Data can be obtained by reading SSDAT register. If SPI interrupt is allowed, when setting SPIF in 1, an interrupt will be generated as well. At this time, receiving shift register keeps original data and place SPIF bit in 1, thus SPI slave device will not receive any data until SPIF is reset to 0. SPI slave device must write the data to be transmitted before master device starts to new data transmission to the transmitting shift register. If no data is written before transmitting, slave device will transmit “0x00” bytes to master device. If SSDAT writing operation occurs during the process of transmission, the WCOL flag bit of SPI slave device is placed in 1, and data is already included in the transmitting shift register, WCOL bit of SPI slave device is placed in 1, indicating conflict of SSDAT writing. But the data of shift register will not be influenced and transmission will not be interrupted.

15.1.4 TRANSFER FORM

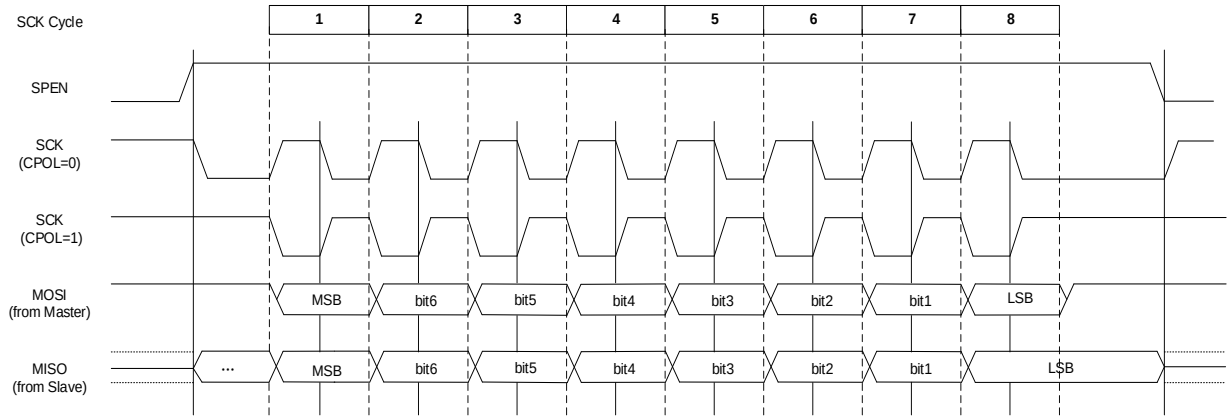
By setting CPOL bit and CPHA bit of SSCON0 register by software, user can select four kinds of combinations related to SPI clock polarity and phase. CPOL bit defines the polarity of clock, meaning the level status when idle, which has little influence on SPI transmission format. CPHA bit defines the phase of clock, meaning clock edge allowing data sampling shift. In two devices of master and slave communication, the configuration of clock polarity and phase shall be consistent.

When CPHA = 0, first edge of SCK captures data, and slave device must prepare for the data before the first edge of SCK.



CPHA = 0 Data transmission

When CPHA = 1, master device outputs data to MOSI line at the first edge of SCK, slave device takes the first edge of SCK as transmitting signal and second edge as capturing data. Therefore, user must complete SSDAT writing operation in two edges of first SCK. Such data transmission form is the preferred form of communication between one master device and one slave device.



CPHA = 1 Data transmission

15.1.5 ERROR DETECTION

Writing in SSDAT register may cause conflict during the period of transmitting data sequence, set WCOL bit in SSCON1 register as 1. Setting WCOL bit as 1 will not generate interrupt, transmitting will not be interrupted. WCOL bit shall be reset to 0 by software.

15.2 TWI

SSMOD[1:0] = 10, SSI is configured as TWI interface. SC92F8003 can only be used as slave device in TWI communication.

SSCON0 (9DH) TWI Control Register (Read/Write)

Bits	7	6	5	4	3	2	1	0
Name	TWEN	TWIF	-	GCA	AA	STATE[2:0]		
R/W	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
POR	0	0	x	0	0	0	0	0

Bits	Name	Instruction
7	**TWEN**	TWI control 0: Disable TWI 1: Enable TWI
6	**TWIF**	TWI interrupt flag bit 0: clear to 0 by software 1: Under the following conditions, interrupt flag is set to 1 by hardware ①First frame of address matched successfully ② Successfully receiving or transmitting 8-bit data ③ Restart ④ Slave device receives stopping signal
4	**GCA**	General Address Response Flag Bit 0: Non-response general address 1: When GC is placed in 1, and this bit is set to 1 by hardware and clear to 0 automatically
3	**AA**	Receiving enabling bit 0: Information sent by receiving master not allowed 1: Information sent by receiving master allowed
2~0	**STATE[2:0]**	State device state flag bit 000: Slave device is in idle state, wait for TWEN to be set to 1, and detect TWI startup signal. 001: Slave device is receiving first frame of address and reading and writing bits (8^{th} bit for reading and writing bit, 1 for reading, 0 for writing). 010: The master transmitting data and state of slave device receiving data 011: The master receiving data and state of slave device transmitting data 100: In the state of transmitting data of slave device, when the master device returns to UACK (high level for acknowledge bit), skip to this state, wait for restarting signal or stopping signal. 101: When the slave device is in transmitting state, write AA in 0, it will enter this state, and wait for restarting signal or stopping signal.
5	**-**	Reserved

SSCON1 (9EH) TWI Address Register (Read/Write)

Bits	7	6	5	4	3	2	1	0
Name	TWA[6:0]							GC
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bits	Name	Instruction
7~1	**TWA[6:0]**	TWI Address Register
0	**GC**	TWI General Address Enabling 0: Prohibits responding general address 1: Allow responding general address

SSDAT (9FH) TWI Data Cache Register (Read/Write)

Bits	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	TWDAT[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bits	Name	Instruction
7~0	**TWDAT[7:0]**	TWI Data Cache Register

15.2.1 SIGNAL DESCRIPTION

TWI Clock Signal Line (SCL)

This clock signal is sent from master device and connects all slave device. One byte of data is transmitted for every 9 clock periods. First 8 periods are used for data transmission and last one for receiver response clock.

TWI Data Signal Line (SDA)

SDA is two-way signal line, and shall be high level when idling, and can be pulled up by pull-up resistance on SDA line.

15.2.2 OPERATING MODES

TWI communication of SC92F8003 has only slave device mode:

- **Mode Startup:**

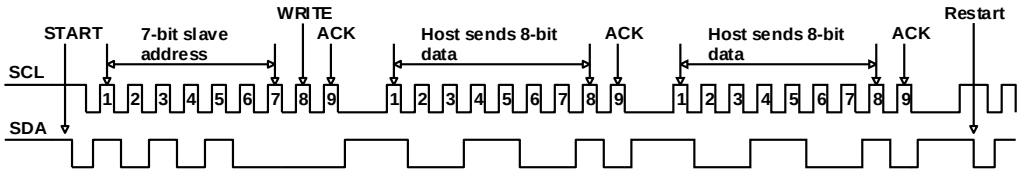
  When TWI enabling flag bit opens (TWEN = 1) and receives enabling signal sent from master device, this mode is initiated.

- **Non-general address response, slave device receiving mode:**

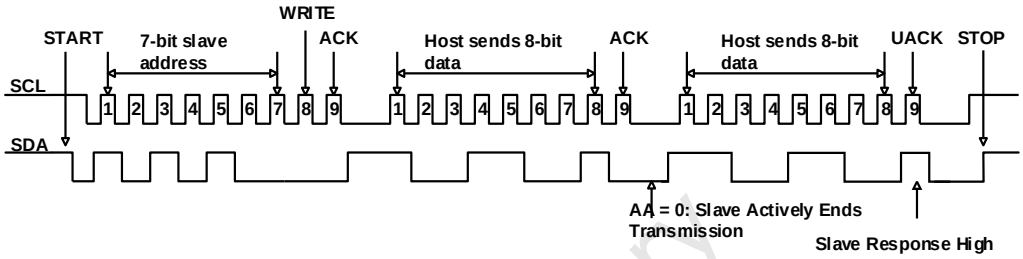
  The slave device enters first frame address (STATE[2:0] = 001) state from idle mode (STATE[2:0] = 000), and waits for first frame data from master device. First frame data is sent by master device, including 7-bit address bit and 1-bit reading and writing bit, all slave devices on TWI bus will receive first frame data of master device. After transmitting first frame data, master device will release SDA signal line. If the address sent by master device is the same as the value of address register of slave device, it indicates that the slave device has been selected and the selected slave device will judge to connect the 8th bit on the bus , which is the data reading and writing bit (=1, reading the command; =0, writing the command), then occupies SDA signal line, after transmitting a low-level response signal at the 9th clock period of SCL, release the bus. After the slave device is selected, enter into different status according to different reading and writing bits:

  ① If the reading and writing bit received from the first frame is writing (0), the slave device enters into the receiving state of slave device (STATE[2:0] = 010), and wait for data sent from receiving master device. Master device will release the bus for transmitting every 8 bits and then wait for the response signal of 9th period of slave device.

1. If the response signal from slave device is low level（ACK）,the master can Continue to send data,The master also can Resend start signal, then the slave device enters into the state of receiving first frame address (001); or to send stopping signal, indicating this transmission is ended , slave device returns to idle state and wait for next start signal from master device.

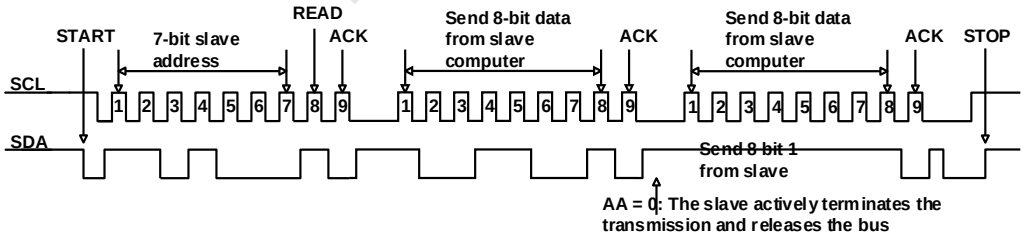


2. If the slave responds with a high level (AA = 0). Indicates that the slave machine no longer receives the data sent by the host, and actively terminates the transmission. the slave return to the idle.

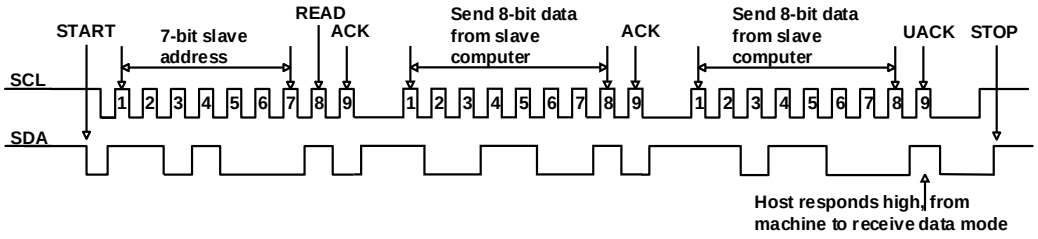


② If the reading and writing bit received from the first frame is reading (1), the slave device will occupy the bus and send data to master device. The slave device will release the bus for transmitting every 8-bit data and wait for the response from master device:

1. If the response from master device is low level, the slave device continues to send data. During the transmitting process, if AA value in slave device register is rewritten as 0, the slave device will automatically end the transmission and release the bus after transmitting current bytes, and wait for master device to stop signal or restart signal (STATE[2:0] = 101).



2. If the response from master device is high level, then the slave device state will be STATE[2:0] = 100 and wait for the master device to stop signal or restart signal.



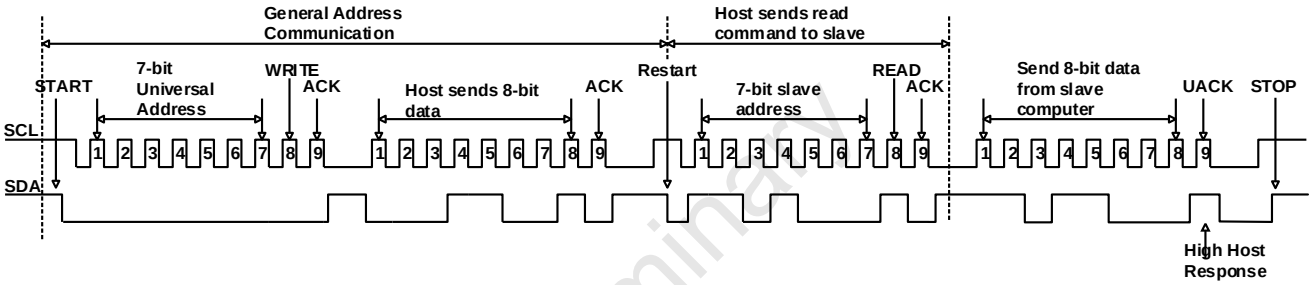
- **Response to General Address:**

  When GC=1, general address is allowed to be used. When the slave device enters into the state of receiving first frame address (STATE[2:0] = 001), the address bit data received in first frame data will be 0x00, at this time, all slave device will respond the master device. The reading and writing bit sent from master device must be write (0), all slave device will enter into the state of receiving data (STATE[2:0] = 010). The master device will release SDA line for transmitting every 8-bit data and read the state on SDA line:

  ① If any response from slave device occurs, there are three modes of master device communication, as shown below:

  1. Continue to transmit data;
  2. Restart, the slave device enters into the state of receiving first frame address (STATE[2:0] = 001);
  3. Transmitting stopping signal and end this communication.

  ② If the SDA is high level, SDA will be in idle state（STATE[2:0] = 000）.



Note: When using general address, the reading and writing bit sent by master device can not be Reading (1) state, or else, all the other devices on the bus will also send response except for equipment transmitting data.

15.2.3 OPERATING STEPS

The operating steps of TWI in SSI are shown below:

① Configure SSMOD[1:0] and select TWI mode;

② Configure SSCON0 TWI control register;

③ Configure SSCON1 TWI address register;

④ If the slave device receives data, wait for interrupt flag bit TWIF in SSCON0 be set to 1.The interrupt flag bit will be set to 1 when the slave device receives every 8-bit data. The interrupt flag bit shall be reset to zero by user manually;

⑤ If the slave device transmits data, write the data to be transmit into TWDAT, TWI will transmit the data automatically. Interrupt flag bit TWIF will be set to 1 for transmitting every 8 bits.

15.3 UART1

SSMOD[1:0] = 11, SSI is configured as UART interface.

SSCON0 (9DH) Serial Port 1 Control Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbol	SM0	-	SM2	REN	TB8	RB8	TI	RI
R/W	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	x	0	0	0	0	0	0

Bit Number	Bit Symbol	Instruction
7	**SM0**	Serial Communication Mode Control 0: Mode 1, 10-bit full duplex asynchronous communication composing of 1 starting bit, 8 data bit and one stopping bit, with communication baud rate changeable; 1: Mode 3, 11-bit full duplex asynchronous communication, composing of 1 starting bit, 8 data bit and one programmable 9th bit and one stopping bit, with communication baud rate changeable.
5	**SM2**	Serial communication mode control bit 2, this control bit is only valid for mode 3 0: Configure RI for receiving each complete data frame to generate interrupt request; 1: When receiving a complete data frame and only when RB8=1, configure RI to generate interrupt request.
4	**REN**	Receive allowing control 0: Receiving data not allowed; 1: Receiving data allowed.
3	**TB8**	Only valid for mode 3, 9th bit of receiving data
2	**RB8**	Only valid for mode 3, 9th bit of receiving data
1	**TI**	Transmit interrupt flag
0	**RI**	Receive interrupt flag
6	**-**	Reserved

SSCON1 (9EH) Serial Port 1 Baud Rate Control Register Low (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbol	BAUD1L [7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

SSCON2(95H) Serial Port 1 Baud Rate Control Register Low (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbol	BAUD1H [7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit Number	Bit Symbol	Instruction
7~0	**BAUD [15:0]**	**Serial Port Baud Rate Control** $\text{BaudRate} = \frac{\text{fsys}}{\text{BAUD1H, BAUD1L}}$

SSDAT (9FH) Serial Port Data Cache Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbol	SBUF1[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit Number	Bit Symbol	Instruction
7~0	**SBUF[7:0]**	**Serial Port Data Cache Register** SBUF contains two registers: one for transmitting shift register and one for receiving latch, data writing to SBUF will be sent to shift register and initiate transmitting process, reading SBUF1 will return the contents of receiving latch.

16 ANALOG-TO-DIGITAL CONVERTER(ADC)

SC92F8003 has a 12-bit high-precision successive approximation ADC with 8 channel. the external 7 ACD channel is multiplexing with other IO interfaces. Cooperating with the internal 2.4V reference voltage, One internal channel connected to 1/4 VDD can be used for measuring VDD voltage.

There are 2 options for ADC reference voltage:

① VDD pin (directly internal V_{DD});

② Precise 2.4V reference output from internal Regulator (at this time, MCU supply voltage VDD can not be lower than 2.9V).

Note: f_{ADC} is obtained directly from internal f_{HRC} fractional frequency. user shall note that ADC clock frequency f_{ADC} can not be faster than the frequency of system clock f_{SYS}, or else, it will cause abnormal result of ADC conversion!

16.1 ADC-RELATED REGISTERS

ADCCON (ADH) ADC Control Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbol	ADCEN	ADCS	EOC/ADCIF	ADCIS[4:0]				
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	n

Bit Number	Bit Symbol	Instruction
7	**ADCEN**	Start up ADC power 0: Disable ADC module power 1: Enable ADC module power
6	**ADCS**	ADC start trigger control (ADC Start) Write “1” for this bit, start to conduct ADC switch, this bit is the trigger signal only for ADC switch. This bit is valid only for writing “1”. Note: After writing “1” to ADCS, do not write to the ADCCON register until the interrupt flag EOC/ADCIF is set.
5	**EOC/ADCIF**	End Of Conversion / ADC Interrupt Flag 0: Conversion not completed

		1: ADC conversion completed. User software is used for clearing up ADC conversion completion flag EOC: When user sets up ADCS for conversions, this bit will be reset to 0 by hardware automatically; After completing conversion, this bit will be configured to 1 automatically by hardware; ADC interrupt request flag ADCIF: This bit is also used as interrupt request flag of ADC interrupt, if user enables ADC interrupt, user must use this software to clear up this bit after ADC interrupt.
4~0	**ADCIS[4:0]**	ADC Input Selector 00000: Select AIN0 as ADC input 00001: Select AIN1 as ADC input 00010: Select AIN2 as ADC input 00011: Select AIN3 as ADC input 00100: Select AIN4 as ADC input 00101: Select AIN5 as ADC input 00110: Select AIN6 as ADC input 00111~11110：Reserved 11111: ADC input as 1/4 V_{DD}, used for measuring power voltage

ADCCFG1 (AAH) ADC Configuration Register 1(Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	LOWSP	ADCCK[2:0]		
R/W	-	-	-	-	R/W	R/W	R/W	R/W
POR	x	x	x	x	0	0	0	0

Bit Number	Bit Symbol	Instruction
3	**LOWSP**	ADC Sampling Clock Frequency Selector 0: Configure ADC sampling time as 6 ADC sampling clock periods 1: Configure ADC sampling time as 36 ADC sampling clock periods LOWSP control is ADC sampling clock frequency, ADC conversion clock frequency is controlled by ADCCK[2:0], independent of the influence of LOWSP bit The whole process from sampling to conversion for ADC needs 6 or 36 ADC sampling clocks plus 14 ADC conversion clocks, therefore, in practical application, the total time of ADC from sampling to conversion shall be calculated as follows: LOWSP=0：$T_{ADC1}=(6+14)/f_{ADC}$； LOWSP=1：$T_{ADC2}=(36+14)/f_{ADC}$
2~0	**ADCCK[2:0]**	ADC Sampling Clock Frequency Selector 000: Configure ADC clock frequency f_{ADC} as $f_{HRC}/32$; 001: Configure ADC clock frequency f_{ADC} as $f_{HRC}/24$; 010: Configure ADC clock frequency f_{ADC} as $f_{HRC}/16$; 011: Configure ADC clock frequency f_{ADC} as $f_{HRC}/12$; 100: Configure ADC clock frequency f_{ADC} as $f_{HRC}/8$; 101: Configure ADC clock frequency f_{ADC} as $f_{HRC}/6$; 110: Configure ADC clock frequency f_{ADC} as $f_{HRC}/4$; 111: Configure ADC clock frequency f_{ADC} as $f_{HRC}/3$; **Note: The clock source of the ADC circuit is fixed at f_{HRC} = 24MHz and will not change with the switching of the internal and external system clocks.**
7~4	**-**	Reserved

ADCCFG0 (ABH) ADC Configuration Register 0(Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbol	-	EAIN6	EAIN5	EAIN4	EAIN3	EAIN2	EAIN1	EAIN0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	x	0	0	0	0	0	0	0

Bit Number	Bit Symbol	Instruction
0	**EAINx (x=0~6)**	**ADC Port Configuration Register** 0: Configure AINx as IO PORT 1: Configure ANIx as ADC input and remove pull-up resistance automatically.
7	**-**	Reserved

OP_CTM1 (C2H@FFH) Customer Option Register 1(Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbol	VREFS	XTLHF	-	-	IAPS[1:0]		-	-
R/W	R/W	R/W	-	-	R/W	R/W	-	-
POR	n	n	x	x	n	n	x	x

Bit Number	Bit Symbol	Instruction
7	**VREFS**	**Reference voltage selection (initial value is called in from Code Option, user can modify the setting)** 0: Configure ADC VREF as V_{DD} 1: Configure ADC VREF as internally correct 2.4 V

ADCVL (AEH) ADC Conversion Value Register (Low Bit) (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbol	ADCV[3:0]				-	-	-	-
R/W	R/W	R/W	R/W	R/W	-	-	-	-
POR	0	0	0	0	x	x	x	x

ADCVH (AFH) ADC Conversion Value Register (High Bit) (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbol	ADCV[11:4]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit Number	Bit Symbol	Instruction
11~4	**ADCV[11:4]**	ADC conversion value high 8-bit value
3~0	**ADCV[3:0]**	ADC conversion value low 4-bit value

IE (A8H) Interrupt Enable Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbol	EA	EADC	ET2	EUART	ET1	EINT1	ET0	EINT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit Number	Bit Symbol	Instruction
6	**EADC**	ADC interrupt enabling control 0: EOC/ADCIF interrupt not allowed 1: EOC/ADCIF interrupt allowed

IP (B8H) Interrupt Priority Control Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbol	-	IPADC	IPT2	IPART	IPT1	IPINT1	IPT0	IPINT0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	x	0	0	0	0	0	0	0

Bit Number	Bit Symbol	Instruction
6	**IPADC**	ADC Interruption Priority Selection 0: Set the interrupt priority of ADC to be "low" 1: Set the interrupt priority of ADC to be "high"

16.2 ADC CONVERSION STEPS

Operating steps for user to practically conduct ADC conversion are shown below:

① Configure ADC input pin; (configure corresponding bit of AINx as ADC input, in general, ADC pin will be prefixed);

② Configure ADC reference voltage Vref and ADC conversion frequency;

③ Enable ADC ;

④ Select ADC input channel; (Configure ADCIS bit and select ADC input channel);

⑤ Set ADCS, and start conversion;

⑥ Wait for EOC/ADCIF=1, if ADC interrupt is enabled, it will generates and user shall clear EOC/ADCIF flag to 0 by software;

⑦ Obtain 12-bit data from ADCVH, ADCVL from high bit to low bit, and complete a conversion

⑧ If no change in input channel, repeat Step 5 to Step 7 for next conversion.

Note: Before setting up IE[6](EADC), it is recommended for the user to use software to clean up

EOC/ADCIF first. After completing ADC interrupt service process, user shall eliminate EOC/ADCIF to avoid generating ADC interrupt constantly.

17 EEPROM AND IAP OPERATIONS

There are two options for SC92F8003 IAP operating scope:

① EEPROM and IAP operating mode is shown below:

② 128 bytesEEPROM can be used as data storage;

③ The area (range optional) of IC's Code and 128 bytes of EEPROM can be used for In Application Programming(IAP) operations, which is mainly used for remote programs updating.

As Code Option, user shall Select EEPROM and IAP operating mode before it is written to IC by programmer:

OP_CTM1 (C2H@FFH) Customer Option Register 1 (Read/Write)

Bit number	7	6	5	4	3	2	1	0
Bit symbol	VREFS	XTLHF	-	-	IAPS[1:0]		-	-
R/W	R/W	R/W	-	-	R/W	R/W	-	-
POR	n	n	x	x	n	n	x	x

Bits	Name	Instruction
3~2	**IAPS[1:0]**	IAP space scope selection 00: Code zone prohibits IAP operations, only EEPROM zone used for data storage 01: last 0.5k code zone allows IAP operation (3E00H ~ 3FFFH) 01: last 1k code zone allows IAP operation (3C00H ~ 3FFFH) 11: All code zone allows IAP operation (0000H ~ 3FFFH)

17.1EEPROM / IAP OPERATING-RELATED REGISTERS

Instruction for EEPROM / IAP operating-related registers:

Symbol	Address	Instruction	7	6	5	4	3	2	1	0	Reset value
IAPKEY	F1H	IAP Protection Register	IAPKEY[7:0]								00000000b
IAPADL	F2H	IAP Write Address Low Register	IAPADR[7:0]								00000000b
IAPADH	F3H	IAP Write Address High Register	-	-	IAPADR[13:8]						xx000000b
IAPADE	F4H	IAP Write Extended Address Register	IAPADER[7:0]								00000000b
IAPDAT	F5H	IAP Data Register	IAPDAT[7:0]								00000000b
IAPCTL	F6H	IAP Control Register	-	-	-	-	PAYTIMES [1:0]		CMD[1:0]		xxxx0000b

IAPKEY (F1H) IAP Protection Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbol	IAPKEY[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit Number	Bit symbol	Instruction
7~0	**IAPKEY[7:0]**	Enable EEPROM/IAP function and operation time limit configuration Write a non-zero value n, representing: ① Enable EEPROM / IAP function; ② If no writing command is received after n system clocks, EEPROM / IAP function will be reclosed.

IAPADL (F2H) IAP Write Address Low Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbol	IAPADR[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit Number	Bit symbol	Instruction
7~0	**IAPADR[7:0]**	EEPROM/IAP writing address low 8-bit

IAPADH (F3H) IAP Write Address High Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbol	-	-	IAPADR[13:8]					
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
POR	x	x	0	0	0	0	0	0

Bit Number	Bit symbol	Instruction
5~0	**IAPADR[13:8]**	EEPROM/IAP writing address high 6-bit
7~6	**-**	Reserved

IAPADE (F4H) IAP Write Extended Address Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbol	IAPADER[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit Number	Bit symbol	Instruction
7~0	**IAPADER[7:0]**	IAP extended address: 0x00: MOVC and IAP programming for Code 0x01: Conduct reading operation for user ID region, no writing operation is allowed 0x02: MOVC and write are performed for EEPROM Other: Reserved

IAPDAT (F5H) IAP Data Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbol	IAPDAT[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit Number	Bit symbol	Instruction
7~0	**IAPDAT**	EEPROM / IAP written data

IAPCTL (F6H) IAP Control Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	PAYTIMES[1:0]		CMD[1:0]	
R/W	-	-	-	-	R/W	R/W	R/W	R/W
POR	x	x	x	x	0	0	0	0

Bit Number	Bit symbol	Instruction
3~2	**PAYTIMES[1:0]**	Upon EEPROM/IAP writing operation, CPU Hold Time length configuration 00: Configure CPU HOLD TIME 6mS@16/8/4/1.33 MHz 01: Configure CPU HOLD TIME 3mS@16/8/4/1.33 MHz 10: Configure CPU HOLD TIME1.5mS@16/8/4/1.33 MHz 11：Reserved Instruction: CPU Hold is for PC pointer, other functional module continues to work; interrupt flag is saved, and interrupt is generated after completing Hold, but several times of interrupt can only be saved once. Selection suggestion: 2.7V~5.5 V for $V_{DD,}$ 10 is available 2.4V~5.5V for V_{DD}, 01 or 00 is available
1~0	**CMD[1:0]**	EEPROM / IAP writing operating command 10: write Other: Reserved **Note: the statement of EEPROM/IAP write operation shall be followed by at least 8 NOP instructions to guarantee subsequent instruction can be implemented normally after finishing IAP operation!**
7~4	**-**	Reserved

17.2 EEPROM / IAP OPERATING PROCEDURES:

Writing procedure of SC92F8003 EEPROM/IAP are shown below:

① Write 0x00 into IAPADE[7:0]: select Code and conduct IAP operation; Write 0x02 into IAPADE[7:0]: Select EEPROM and conduct EEPROM reading and writing operations;

② Write data into IAPDAT[7:0] (data for EEPROM / IAP writing ready);

③ Write address into {IAPADR[13:8], IAPADR[7:0]} (target address of EEPROM/IAP operation ready);

④ Write a nonzero value n into IAPKEY[7:0] (switch on protection of EEPROM / IAP, and EEPROM / IAP function will be switched off when there is no writing command within n system clocks);

⑤ Write CPU Hold time into IAPCTL[3:0] (set CPU Hold time by configuring CMD[1:0] as 1 or 0, CPU is Hold up and start up EEPROM/IAP writing);

⑥ EEPROM/IAP writing ends, CPU proceeds to subsequent operations.

Notes: When programming IC, if “Code Area Prohibits IAP Operations” is selected by Code Option, IAP is unavailable upon IAPADE[7:0]=0x00 (Select Code Area), meaning it is unable to write data, and such data can only be read by MOVC command.

17.2.1 128 BYTES INDEPENDENT EEPROM OPERATING DEMO PROGRAMME

```
#include ”intrins.h”

unsigned char EE_Add;

unsigned char EE_Data;

unsigned char code * POINT =0x0000;
```

C Demo Program of EEPROM Write Operation:

```
EA = 0;                 // Disable global Interrupt

IAPADE = 0x02;          //Select EEPROM Area

IAPDAT = EE_Data;       //Transmit data to EEPROM data register

IAPADH = 0x00;          //High-bit address default write 0x00

IAPADL = EE_Add;        //Write EEPROM target address low bit

IAPKEY = 0xF0;          //This value can be adjusted as required; it shall guarantee that

                        // The Time interval between this instruction implemented and writing
                    IAPCTL value shall be less than 240（0xf0） system clocks, or else, IAP
                    function is closed;

                        //Pay special attention to enabling interrupt

IAPCTL = 0x0A;          //Implement EEPROM write operation, 1.5ms@16/8/4/1.33 MHz;

_nop_();                //Wait (at least 8 _nop_())

_nop_();

_nop_();

_nop_();

_nop_();

_nop_();

_nop_();
```

```
_nop_();
IAPADE = 0x00;                  //Return to ROM Area
EA = 1;                         //enable master interrupt
```

C Demo Program of EEPROM Read Operation

```
EA = 0;                         //disable master interrupt
IAPADE =  0x02;                 //Select EEPROM Area
EE_Data = *( POINT +EE_Add);          //Read value in IAP_Add to IAP_Data
IAPADE =  0x00;                 //Return to ROM Area ,prevent MOVC operates to EEPROM
EA = 1;                         // enable master interrupt
```

17.2.2 16 KBYTES CODE ZONE IAP OPERATING DEMO PROGRAMME

```
#include ”intrins.h”
unsigned int IAP_Add;
unsigned char IAP_Data;
unsigned char code * POINT =0x0000;
```

Demo: IAP Writing Operations:

```
IAPADE = 0x00;                  //Select Code Area
IAPDAT = IAP_Data;              //Transmit data to IAP data register
IAPADH = (unsigned char)((IAP_Add >> 8));      //Write IAP target address high bit

IAPADL = (unsigned char)IAP_Add;               //Write IAP target address low bit

IAPKEY = 0xF0;
//This value can be adjusted as required; it shall guarantee this instruction is implemented to assigned IAPTL value;
                        // Time interval shall be less than 240（0xf0） system clocks, or else, IAP function is closed;
                        //Pay special attention upon starting interrupt
IAPCTL = 0x0A;                  //Implement EEPROM write operation, 1.5ms@16/8/4/1.33MHz;
_nop_();                        //Wait (at least 8 _nop_())
_nop_();
_nop_();
```

```
_nop_();
_nop_();
_nop_();
_nop_();
_nop_();
```

C Demo Program of IAP Read Operation:

```
IAPADE = 0x00;                    //Select Code Area
IAP_Data = *( POINT+IAP_Add);     //Read value in IAP_Add to IAP_Data
```

Note: IAP operation in 16 Kbytes Code Area has certain risks, user shall implement corresponding safety measures in software. **Misoperation may result in user programme to be rewritten!** Unless such function is required by the user (such as used for remote programme update, etc.), it is not recommended to use by user.

18 CHECKSUM CHECK

SC92F8003 is equipped with a check sum module, which is used for generating 16-bit check sum of programme code in real time. user can compare such check sum with theoretical value to monitor whether the contents in programme zone are correct.

Note: Check sum is the sum of data in the whole programe area**, which is the data of 0000H ~ 3FFDH address unit. If there are residual values from user's last operations in address unit, it may result in inconsistency of check sum with theoretical value. Therefore, it is recommended that user shall erase the whole code** area **or write 0 before programming code so as to guarantee the consistency between check sum and theoretical value.**

18.1 REGISTERES RELATED TO CHECK SUM CHECK OPERATIONS

CHKSUML (FCH) Check Sum CheckSum Result Register Low Bit (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbol	CHKSUML[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit Number	Bit Symbol	Instruction
7~0	**CHKSUML [7:0]**	CheckSum result register low bit

CHKSUMH (FDH) Check Sum Result Register High Bit (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbol	CHKSUMH[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit Number	Bit Symbol	Instruction
7~0	**CHKSUMH [7:0]**	CheckSum result register high bit

OPERCON (EFH) operation control register (read and write)

Bit Number	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	-	-	CHKSUMS
R/W	-	-	-	-	-	-	-	R/W
POR	x	x	x	x	x	x	x	0

Bit Number	Bit Symbol	Instruction
0	**CHKSUMS**	CheckSum operation starts trigger control (Start) Write “1” for this bit, start to conduct Check sum calculation. This bit is valid for only writing 1.

19 ELECTRICAL CHARACTERISTICS

19.1 ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Minimum Value	Maximum Value	UNIT
VDD/VSS	DC supply voltage	-0.3	5.5	V
Voltage ON any Pin	Input/output voltage of any pin	-0.3	V_{DD}+0.3	V
T_A	Ambient temperature	-40	85	℃
T_{STG}	Storage temperature	-55	125	℃

19.2 RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Minimum Value	Maximum Value	UNIT	System Clock Frequency
V_{DD1}	Operating voltage	2.9	5.5	V	>12MHz
V_{DD2}	Operating voltage	2.4	5.5	V	≤12MHz
T_A	Ambient temperature	-40	85	℃	

19.3 D.C. CHARACTERISTICS

(V_{DD} = 5V,T_A = +25℃, unless otherwise specified)

Symbol	Parameter	Minimum Value	Typical value	Maximum value	Unit	Testing conditions
Current						
I_{op1}	Operating current	-	7.8	-	mA	f_{SYS}=16MHz
I_{op2}	Operating current	-	5.8	-	mA	f_{SYS} =8MHz
I_{op3}	Operating current	-	4.9	-	mA	f_{SYS} =4MHz
I_{op4}	Operating current	-	4.2	-	mA	f_{SYS} =1.33MHz
I_{pd1}	Standby Current (Power Down Mode)	-	0.7	1.0	μA	
I_{IDL1}	Standby Current	-	5.4	-	mA	

	(IDLE Mode)					
I_{BTM}	Base Timer Operating Current	-	5.9	-	μA	BTMFS[3:0]= 1000 One interrupt occurs for every 4.0 seconds
I_{WDT}	WDT Current	-	4.2	5.0	μA	WDTCKS[2:0]= 000 WDT overflows every 500ms
IO Port Features						
V_{IH1}	Input high voltage	$0.7V_{DD}$	-	V_{DD}+0.3	V	
V_{IL1}	Input low voltage	-0.3	-	$0.3V_{DD}$	V	
V_{IH2}	Input high voltage	$0.8V_{DD}$	-	V_{DD}	V	Schmidt trigger input: RST/tCK/SCK
V_{IL2}	Input low voltage	-0.2	-	$0.2V_{DD}$	V	
I_{OL1}	Output low current	-	40	-	mA	V_{Pin}=0.4V
I_{OL2}	Output low current	-	70	-	mA	V_{Pin}=0.8V
I_{OH1}	Output high current	-	20	-	mA	V_{Pin}=4.3V
I_{OH2}	Output high current	-	10	-	mA	V_{Pin}=4.7V
R_{PH1}	Pull-up resistance	-	32	-	kΩ	
Internal calibrated 2.4V as ADC reference voltage						
V_{DD24}	Internal reference 2.4V voltage output	2.38	2.40	2.42	V	T_A=-40~85 ℃

(V_{DD} = 3.3V,T_A = +25℃, unless otherwise specified)

Symbol	Parameter	Minimum Value	Typical value	Maximum value	Unit	Testing conditions
Current						
I_{op5}	Operating current	-	5.7	-	mA	f_{SYS} =16MHz
I_{op6}	Operating current	-	4.4	-	mA	f_{SYS} =8MHz
I_{op7}	Operating current	-	3.8	-	mA	f_{SYS} =4MHz
I_{op8}	Operating current	-	3.3	-	mA	f_{SYS} =1.33MHz
I_{pd2}	Standby Current (Power Down Mode)	-	0.7	1	uA	
I_{IDL2}	Standby Current (IDLE Mode)	-	4.0	-	mA	
IO Port Features						
V_{IH3}	Input high voltage	$0.7V_{DD}$	-	V_{DD}+0.3	V	
V_{IL3}	Input low voltage	-0.3	-	$0.3V_{DD}$	V	
V_{IH4}	Input high voltage	$0.8V_{DD}$	-	V_{DD}	V	Schmidt trigger input: RST/tCK/SCK
V_{IL4}	Input low voltage	-0.2	-	$0.2V_{DD}$	V	
I_{OL3}	Output low current	-	30	-	mA	Vpin=0.4V
I_{OL4}	Output low current	-	51	-	mA	Vpin=0.8V
I_{OH3}	Output high current	-	6	-	mA	Vpin=3.0V
R_{PH2}	Pull-up resistance	-	55	-	kΩ	
Internal calibrated 2.4V as ADC reference voltage						
V_{DD24}	Internal reference 2.4V voltage output	2.38	2.40	2.42	V	T_A=-40~85 ℃

19.4 AC ELECTRICAL FEATURES

(V_{DD} = 2.4V ~ 5.5V, TA = 25℃, unless otherwise specified)

Symbol	Parameters	Minimum Value	Typical Value	Maximum Value	Unit	Testing Conditions
T_{OSC1}	External high frequency oscillator start time	-	9	-	ms	External 16 MHz crystal oscillator
T_{OSC2}	External high start time frequency oscillator	-	17	-	ms	External 8 MHz crystal oscillator
T_{OSC3}	External high frequency oscillator start time	-	33	-	ms	External 4 MHz crystal oscillator
T_{POR}	Power On Reset time	-	5	10	ms	
T_{PDW}	Power Down Mode waking-up Time	-	1	1.5	ms	
T_{Reset}	Reset Pulse Width	18	-	-	us	valid for Low level
f_{HRC}	RC oscillation stability	15.84	16	16.16	MHz	V_{DD}=2.9~5.5V T_A=-20~85 ℃

19.5 ADC ELECTRICAL FEATURES

(**T_A**= 25℃, unless otherwise specified)

Symbol	Parameter	Minimum Value	Typical Value	Maximum Value	Unit	Testing Conditions
V_{AD}	Supply Voltage	2.4	5.0	5.5	V	
N_R	Precision	-	12	-	bit	GND≤V_{AIN}≤V_{DD}
V_{AIN}	ADC Input Voltage	GND	-	V_{DD}	V	
R_{AIN}	ADC input resistance	1	-	-	MΩ	V_{IN}=5V
I_{ADC1}	ADC conversion current 1	-	-	2	mA	ADC Module on V_{DD}=5V
I_{ADC2}	ADC conversion current 2	-	-	1.8	mA	ADC module on V_{DD}=3.3V
DNL	Differential nonlinear error	-	±1	-	LSB	V_{DD}=5V VREF=5V
INL	Integral nonlinear error	-	±7	-	LSB	
E_Z	Offset error	-	±6	-	LSB	
E_F	Full scale error	-	0	-	LSB	
E_{AD}	Total absolute error	-	±7	-	LSB	V_{DD}=5V VREF=5V
T_{ADC1}	ADC conversion time 1	-	7.5	-	μs	ADC Clock = 2.67MHz Adc sampling period =6
T_{ADC2}	ADC 转换时间 2 ADC conversion time 2	-	15	-	μs	ADC Clock = 1.33MHz Adc sampling period =6

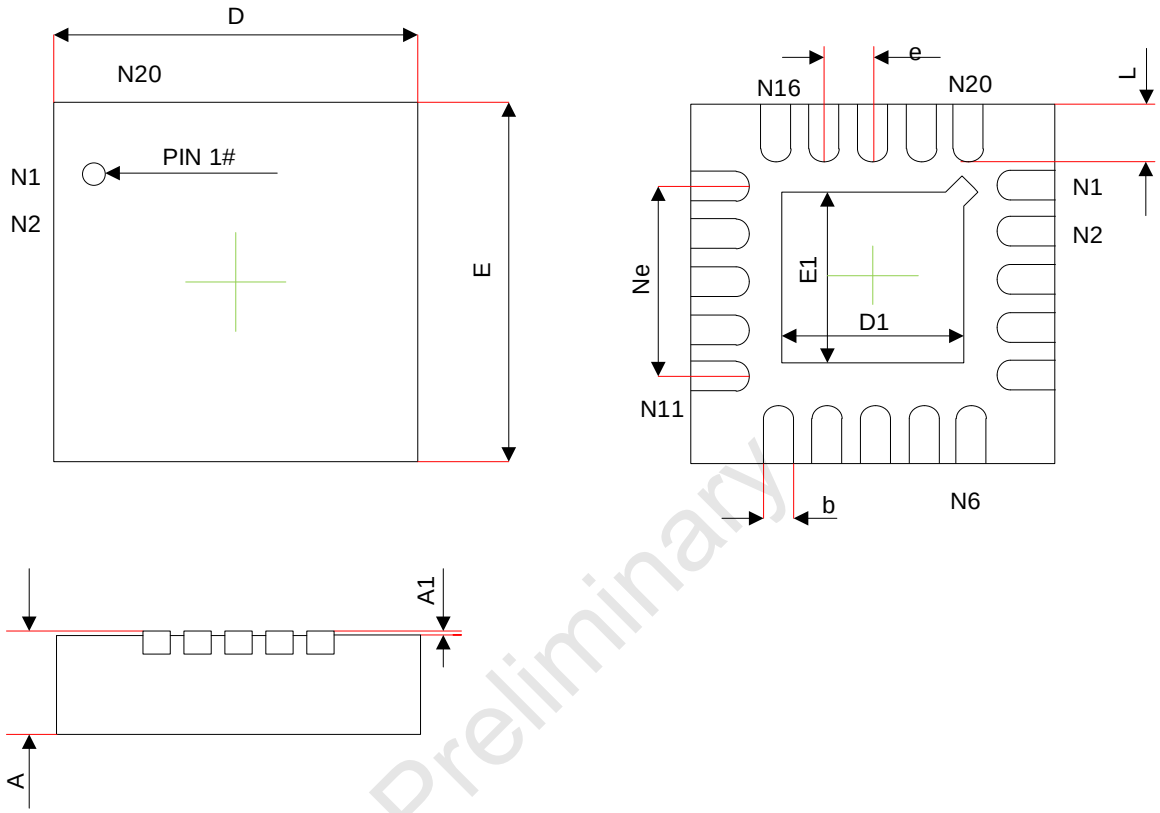
20 ORDERING INFORMATION

PRODUCT NUMBER	ENCAPSULATION	PACKING
SC92F8003Q20R	QFN20	TUBE LOADING
SC92F8003X20U	TSSOP20L	TUBE LOADING

21 PACKAGEING INFORMATION

SC92F8003Q20R

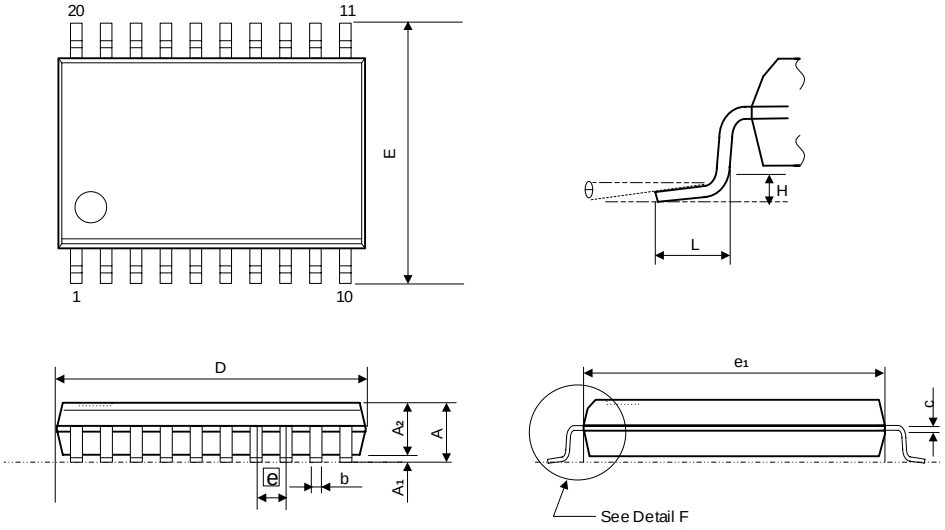
QFN20(3X3)Dimension **Unit: mm**



Symbol	mm(milimetre)		
	Min	Normal	Min
A	0.50	0.55	0.60
A1	0	0.02	0.05
b	0.15	0.20	0.25
D	2.90	3.00	3.10
D1	1.60	1.70	1.80
e	0.40BSC		
Ne	1.60BSC		
E	2.90	3.00	3.10
E1	1.60	1.70	1.80
L	0.25	0.30	0.35

SC92F8003X20U

TSSOP20L Dimension Unit: mm



Symbol	mm(milimetre)		
	Min	Normal	Min
A	-	-	1.200
A1	0.050	-	0.150
A2	0.800	-	1.050
b	0.190	-	0.300
c	0.090	-	0.200
D	6.400	-	6.600
E	6.200	-	6.600
e1	4.300	-	4.500
e	0.65(BSC)		
L	0.450	-	0.750
θ	0°	-	8°
H	0.25(TYP)		

22 REVISION HISTORY

	Notes	Date
V0.1	First edition	March 2018