High-speed 1T 8051-based Flash MCU, 1 Kbytes SRAM, 16 Kbytes Flash, 128 bytes independent EEPROM, 20-channel low-power high sensitivity Touch Key module, 12-bit ADC, 5 Timer/Counters, MDU, UART, SSI, Check Sum module

1 General Description

The SC92F84H3/84H9/84H2 (hereinafter referred to as the SC92F84HX) is a kind of enhanced high-speed 1T 8051-based industrial Flash Microcontroller unit (MCU) with integrated TouchKey function, in which the instruction system is completely compatible with standard 8051 product series.

The SC92F84HX has a 20-channel high sensitivity capacitor TouchKey circuit with low power consumption. The TouchKey circuit can be operated under STOP Mode.

The SC92F84HX is integrated with 16 Kbytes Flash ROM, 1K bytes SRAM, 128 bytes EEPROM, up to 26 General-purpose I/Os (GPIO), 8 IO external interrupters, five 16-bit timers, 21-channel 12-bit high-precision ADC, internal ±1% high-precision 12/6/2MHz high-frequency oscillator and ±4% precision low-frequency 32K oscillator、 external 32.768kHz crystal oscillator interface, one separate UART, one three-in-one serial communication interface which can be used as UART/SPI/TWI (SSI). To improve the reliability and simplify the circuit design, the SC92F84HX is also built in with 4-level optional LVR voltage, 2.4V ADC reference voltage and other high-reliability circuits.

The SC92F84HX features excellent anti-interference performance, which make it possible to be widely applied to industrial control system and consumer applications, such as intelligent household appliances, intelligent household furniture, Internet of Things, wireless communication, gaming peripherals.

2 Features

Operating Voltage: 2.4V ~ 5.5V

Operating Temperature: -40 ~ 85℃

EMS

- **ESD**
  - HBM：MIL-STD-883J Class 3A
  - MM：JEDEC EIA/JESD22-A115 Class C
  - CDM：ANSI/ESDA/JEDEC JS-002-2018 Class C3
- **EFT**
  - EN61000-4-4 Level 4

Core: 1T 8051

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

IAP: Can be code option into 0K, 0.5K, 1K or 16K

EEPROM: 128 bytes EEPROM can be rewritten for 100, 000 times. Written data have more than 10 years of storage life.

SRAM: Internal 256 bytes + external 768 bytes

System Clock (f_{SYS}):

- Built-in high-frequency 24MHz oscillator (f_{HRC}):
  - As the system clock source, f_{SYS} can be set to 12/6/2MHz@2.4~5.5V by programmer selection.
  - Frequency Error: Suitable for 2.9V ~ 5.5V and -40 ~ 85℃ application environment with no more than ±1% of frequency error

Built-in Low-Frequency crystal oscillator interface

- External 32.768kHz crystal oscillator interface connect available
- Clock source of Base Timer (BTM), which can wake up the SC92F84HX from STOP mode

Built-in Low-Frequency 32kHz Oscillator (LRC):

- Clock source of Base Timer (BTM), which can wake up the SC92F84HX from STOP mode
- Clock source of Watchdog (WDT)
- Frequency Error: Suitable for 4.0V ~ 5.5V and -20 ~ 85℃ application environment with 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:

- 2-wire JTAG programming and emulation interface

Interruption (INT):

- 12 interrupt sources: Timer0~4, INT0, INT1, ADC, UART, SSI, Base Timer and TK
- 2 external interrupt vectors shared by 8 interrupt ports, all of which can be defined in rising-edge, falling-edge or dual-edge trigger mode.
- Two-level interrupt priority capability

Digital Peripheral:

- Up to 26 bidirectional independently controllable I/O interfaces, able to configure pull-high resistor independently
- P0/P2 ports with 4-level drive capability
- All I/Os equipped with sink current drive capability (100mA@0.8V)
- 11-bit WDT with optional clock division ratio
- 5 Timer/Counters: Timer0, Timer1, Timer2, Timer3 and Timer4
  - ■ Timer2, Timer3 and Timer4 can be set to capture mode
  - ■ Timer2 Timer3 and Timer4 can each provide two regular PWM channels
  - ■ The clock input pins Tn (n=2, 3) and the falling edge capture pins TnEX of Timer2 and Timer3 can be mapped to other ports.
- 1 independent UART communication interface (Switchable I/O port)
- 1 three-in-one serial communication interface (SSI) (Switchable I/O port)
- 16*16-bit hardware multiplier and divider unit (MDU)
- Built-in Check Sum module

Analog Peripheral:

- 20-channel high sensitivity TouchKey (TK) circuit:
  - ■ Applicable to TouchKey sensor, proximity induction and other TouchKey applications featuring high requirements on sensitivity
  - ■ Features very strong anti-interference which is able to pass 10V dynamic CS test
  - ■ Supports low-power mode
  - ■ Comprehensive development support: Highly flexible touch control software library, intelligent debugging software.
  - ■ TK can quickly wake up from STOP mode
- 21-channel 12-bit ADC
  - ■ Built-in 2.4V reference voltage
  - ■ 2 options for ADC reference voltage: V_{DD} and internal 2.4V
  - ■ Internal one-channel ADC, where VDD can be measured directly
  - ■ ADC conversion completion interrupt

Power Saving Mode:

- IDLE Mode: can be woken up by any interrupt.
- STOP Mode: can be woken up by INT0, INT1, Base Timer and TK

Naming Rules for 92 Series Products

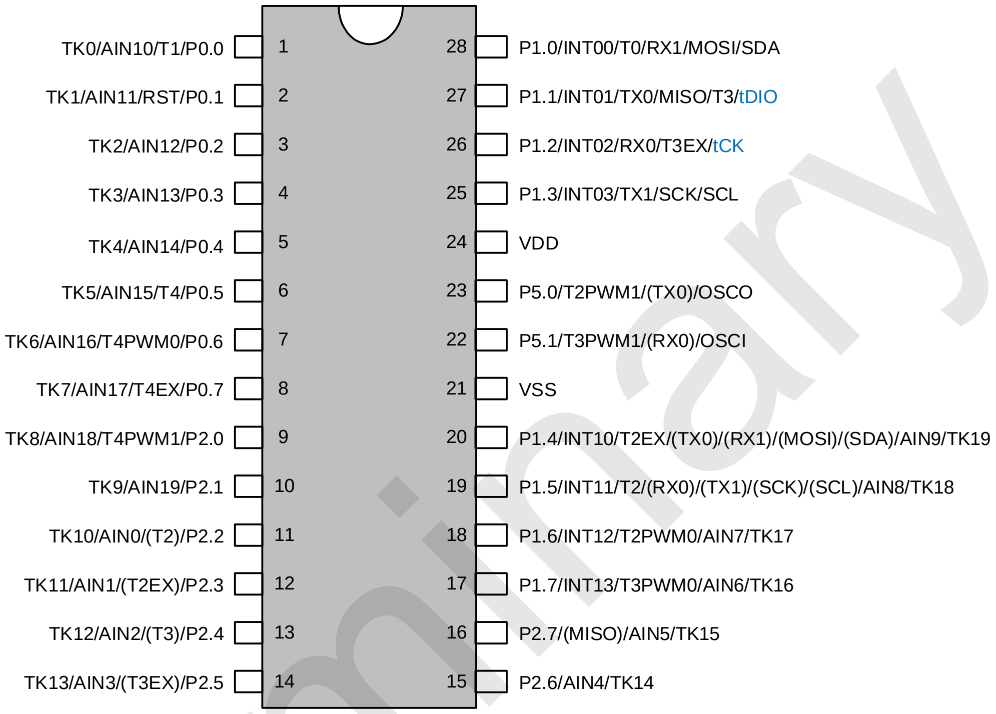
Name	SC	92	F	8	4	H	3	X	M	28	U
S/R	①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩	⑪

S/R	**Meaning**
①	SinOne Chip abbreviation
②	Name of product series
③	Product Type (F: Flash MCU)
④	Serial Number: 7: GP Series, 8: TK series
⑤	ROM Size: 1 for 2K, 2 for 4K, 3 for 8K, 4 for 16K and 5 for 32K…
⑥	Subseries Number.: 0 ~ 9, A ~ Z
⑦	Number of Pins: 0: 8pin, 1: 16pin, 2: 20pin, 3: 28pin, 5: 32pin, 6: 44pin, 7: 48pin, 8: 64pin, 9: 24pin
⑧	Version Number: (default, B, C, D)
⑨	Package Type: (D: DIP; M: SOP; X: TSSOP; F: QFP; P: LQFP; Q: QFN; K: SKDIP)
⑩	Number of Pins.
⑪	Packaging Mode: (U: Tube; R: Tray; T: Reel)

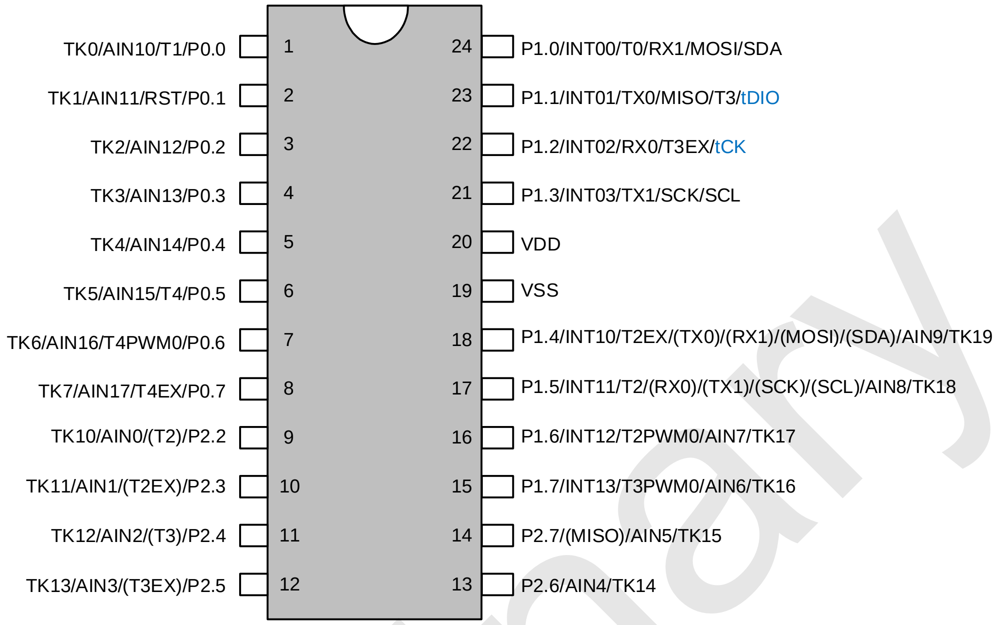
Contents

3 Pin Description

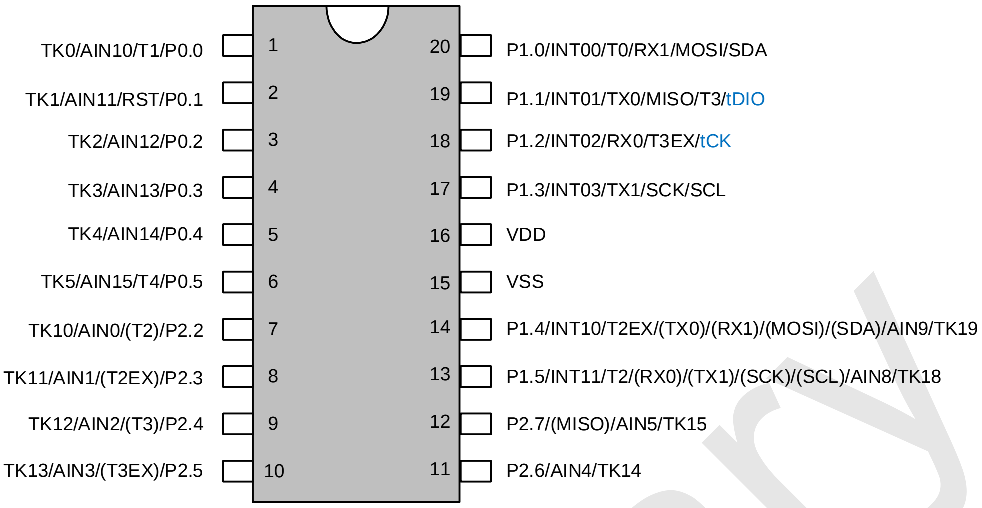
3.1 Pin Configuration



28PIN Pin Diagram
Suitable for SOP28、TSSOP28 package



24PIN Pin Diagram
Suitable for SOP24、TSSOP24 package



20PIN Pin Diagram
Suitable for SOP20、TSSOP20 package

3.2 Pin Definition

Pin number			I/O	TK	ADC	TIM	UART	SPI	TWI	Program	INT	OSC
20	24	28										
1	1	1	P0.0	TK0	AIN10	T1	-	-	-	-	-	-
2	2	2	P0.1	TK1	AIN11	-	-	-	-	RST	-	-
3	3	3	P0.2	TK2	AIN12	-	-	-	-	-	-	-
4	4	4	P0.3	TK3	AIN13	-	-	-	-	-	-	-
5	5	5	P0.4	TK4	AIN14	-	-	-	-	-	-	-
6	6	6	P0.5	TK5	AIN15	T4	-	-	-	-	-	-
-	7	7	P0.6	TK6	AIN16	T4PWM0	-	-	-	-	-	-
-	8	8	P0.7	TK7	AIN17	T4EX	-	-	-	-	-	-
-	-	9	P2.0	TK8	AIN18	T4PWM1	-	-	-	-	-	-
-	-	10	P2.1	TK9	AIN19		-	-	-	-	-	-
7	9	11	P2.2	TK10	AIN0	(T2)	-	-	-	-	-	-
8	10	12	P2.3	TK11	AIN1	(T2EX)	-	-	-	-	-	-
9	11	13	P2.4	TK12	AIN2	(T3)	-	-	-	-	-	-
10	12	14	P2.5	TK13	AIN3	(T3EX)	-	-	-	-	-	-
11	13	15	P2.6	TK14	AIN4	-	-	-	-	-	-	-
12	14	16	P2.7	TK15	AIN5	-	-	(MISO)	-	-	-	-
-	15	17	P1.7	TK16	AIN6	T3PWM0	-	-	-	-	INT13	-
-	16	18	P1.6	TK17	AIN7	T2PWM0	-	-	-	-	INT12	-
13	17	19	P1.5	TK18	AIN8	T2	(RX0)(TX1)	(SCK)	(SCL)	-	INT11	-
14	18	20	P1.4	TK19	AIN9	T2EX	(TX0)(RX1)	(MOSI)	(SDA)	-	INT10	-
15	19	21	VSS	-	-	-	-	-	-	-	-	-
-	-	22	P5.1	-	-	T3PWM1	(RX0)	-	-	-	-	OSCI
-	-	23	P5.0	-	-	T2PWM1	(TX0)	-	-	-	-	OSCO
16	20	24	VDD	-	-	-	-	-	-	-	-	-
17	21	25	P1.3	-	-	-	TX1	SCK	SCL	-	INT03	-
18	22	26	P1.2	-	-	T3EX	RX0	-	-	tCK	INT02	-
19	23	27	P1.1	-	-	T3	TX0	MISO	-	tDIO	INT01	-
20	24	28	P1.0	-	-	T0	RX1	MOSI	SDA	-	INT00	-

UART0 Mapping control

Signal	GPIO- default mapping UART0OS[1:0]=00	GPIO-A group mapping UART0OS[1:0]=01	GPIO-B group mapping UART0OS[1:0]=10
RX0	P1.2	P5.1	P1.5
TX0	P1.1	P5.0	P1.4

SSI Mapping control

Signal			GPIO-default mapping SSIOS=0	GPIO-A group mapping SSIOS=1
RX1	MOSI	SDA	P1.0	P1.4
-	MISO	-	P1.1	P2.7
TX1	SCK	SCL	P1.3	P1.5

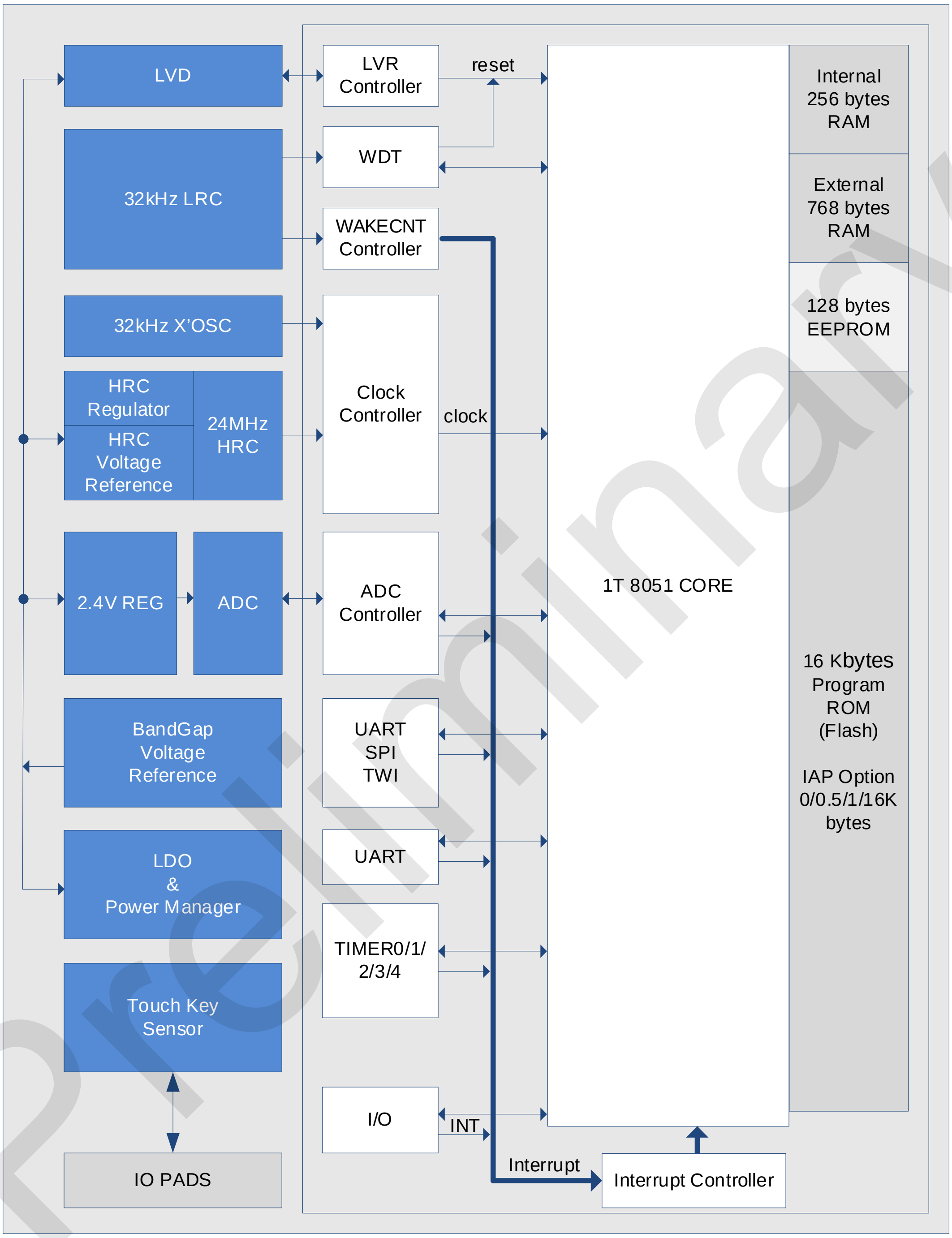
TIM2 Mapping control

Signal	GPIO- default mapping T2OS=0	GPIO-A group mapping T2OS=1
T2	P1.5	P2.2
T2EX	P1.4	P2.3

TIM3 Mapping control

Signal	GPIO default mapping T3OS=0	GPIO-A group mapping T3OS=1
T3	P1.1	P2.4
T3EX	P1.2	P2.5

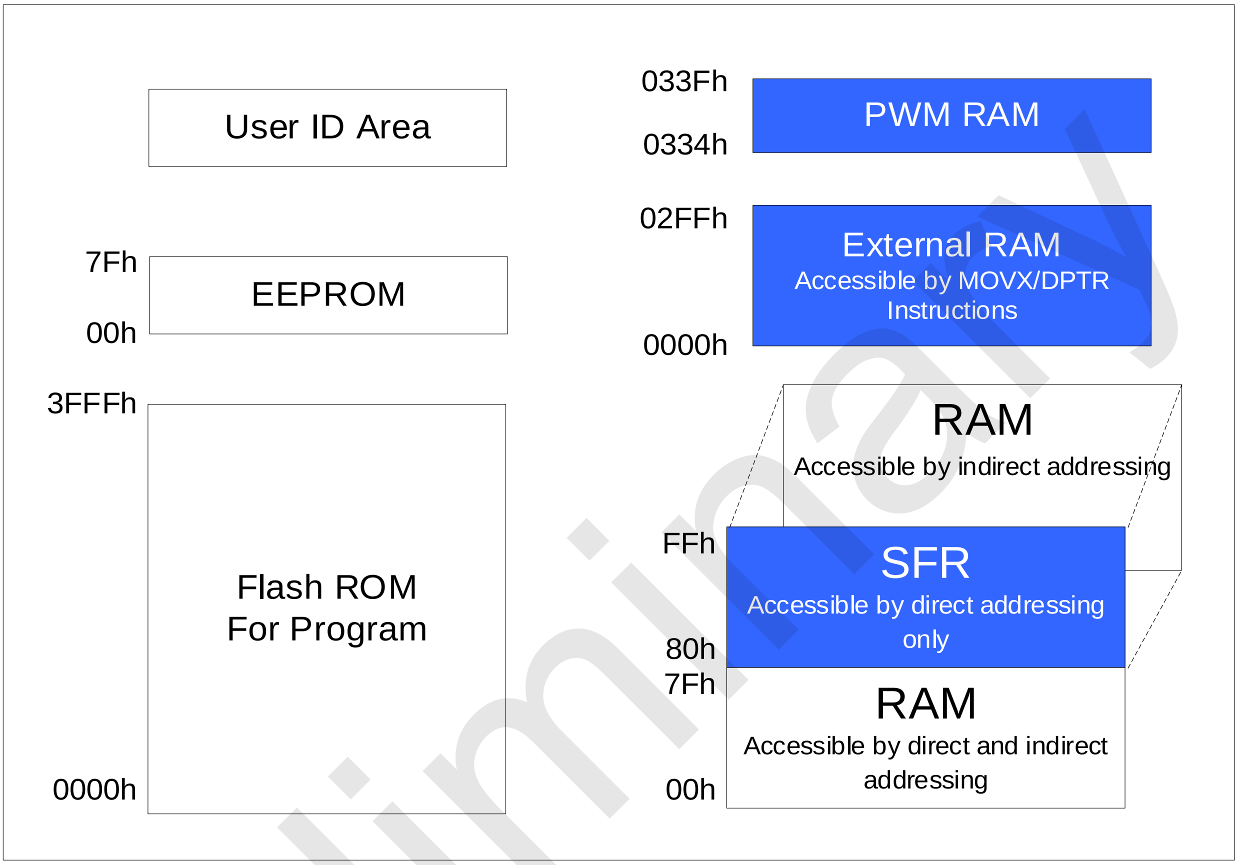
4 Inner BLOCK Diagram



The SC92F84HX BLOCK DIAGRAM

5 Flash ROM and SRAM Structure

The structures of the SC92F84HX's Flash ROM and SRAM are shown as follows:


User ID Area
7Fh
EEPROM
00h
3FFFh
Flash ROM
For Program
0000h
033Fh
PWM RAM
0334h
02FFh
External RAM
Accessible by MOVX/DPTR Instructions
0000h
RAM
Accessible by indirect addressing
FFh
SFR
Accessible by direct addressing only
80h
7Fh
RAM
Accessible by direct and indirect addressing
00h

Flash ROM and SRAM Structure Diagram

5.1 Flash ROM

The SC92F84HX provides 16 Kbytes 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 program and erase by specialized ICP programming device provided by SinOne.

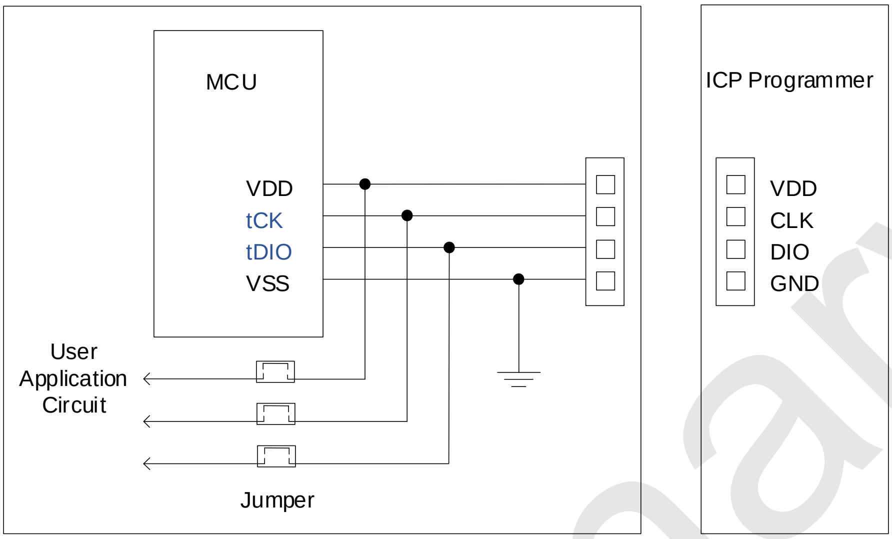
The MOVC instruction is non-addressable within 256 bytes (address of 0000H ~ 00FFH). That is to say, it is unable to read the contents of the 256 bytes region by program, so as to realize the encryption function of chip program. For more details, refer to "SinOne SC92F Series MCU Application Guide".

EEPROM is an data memory separated from 16K bytes ROM with the address of 00H ~ 7FH, which can be accessed by single-byte reading and writing operations in the program; for more details, refer to 19 EEPROM and IAP Operations.

User ID area: The user ID is written when leave the factory, and the user can only perform reading operations in this area. For more details, refer to 19 EEPROM and IAP Operations.

The SC92F84HX 16 Kbytes Flash ROM provide Empty Check, Program, Verify and Erase function other than Read function. This Flash ROM and EEPROM usually needs no Erase operation before writing. Directly writing data can realize coverage of new data.

The SC92F84HX 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

tDIO, tCLK are the programming and simulation signal lines of dual-line JTAG, user can configure the mode of these ports through Code Option when programming.

JTAG Dedicated Mode:

tDIO, tCLK are dedicated ports for programming and simulation, other functions reused with it are not available. This mode is typically used during the online debugging period, convenient for user's simulation debugging; When JTAG dedicated mode is activated, chip is able to enter programming or simulation mode directly without powering on or off again.

Normal Mode (JTAG dedicated ports invalid):

JTAG function is invalid, while other functions reused with it are available. This mode prevents programming port from occupying the MCU pin, convenient user to maximize MCU resources.

Attention: After successfully setting JTAG dedicated port into invalid, the chip must be powered off completely and then powered on again before entering the burning or simulation mode, however, in this way it will affect the programming and simulation in powering mode. SinOne suggests user select the invalid configuration of JTAG dedicated port during mass production programming stage, and JTAG dedicated mode during developing and debugging stage.

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

Bit Number	7	6	5	4	3	2	1	0
Bit Mnemonic	VREFS	-	-	DISJTG	IAPS[1: 0]		-	-

R/W	R/W	-	-	R/W	R/W	R/W	-	-
POR	n	x	x	n	n	n	x	x

Bit Number	Bit Mnemonic	Description
4	**DISJTG**	IO/JTAG port switch bit 0: JTAG mode enable, P1.2\P1.3 are only used as tCLK\tDIO. Recommended Settings during developing and debugging stage. 1:Normal mode, P1.2\P1.3 can be used as normal I/O port, JTAG function is invalid. Recommended setting for mass production burning stage.

5.2 Customer Option Memory (User Programming Setting)

A separate Flash data memory is embedded inside the SC92F84HX, which called Code Option area, to save the user's presets. These presets will be written into IC when programming and loaded into SFR as default values during reset.

Option-related SFR Operating Instructions:

Reading and writing operations to 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:

Symbol	Address	Description	7	6	5	4	3	2	1	0
OP_HRCR	83H@FFH	System Clock Change Register	OP_HRCR[7: 0]							
OP_CTM0	C1H@FFH	Customer Option Register 0	ENWDT	ENXTL	SCLKS[1: 0]		DISRST	DISLVR	LVRS[1: 0]	
OP_CTM1	C2H@FFH	Customer Option Register 1	VREFS	-	-	DISJTG	IAPS[1: 0]		-	-

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

Bit Number	7	6	5	4	3	2	1	0
Bit	OP_HRCR[7: 0]							

Mnemonic								
R/W	R/W							
POR	n	n	n	n	n	n	n	n

Bit Number	Bit Mnemonic	Description
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 versa.

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

Bit Number	7	6	5	4	3	2	1	0
Bit Mnemonic	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 Mnemonic	Description
7	**ENWDT**	Watchdog (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 32.768kHz crystal oscillator selection bit 0: External 32.768kHz crystal Interface disable, P5.0 and P5.1 valid 1: External 32.768kHz crystal Interface enable, P5.0 and P5.1 invalid

5 ~ 4	**SCLKS[1: 0]**	System clock frequency selection bits 00: Reserved; 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 P0.1 as External Reset input pin 1: configure P0.1 as GPIO
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: 3.7 V reset 01: 2.9V reset 00: 2.3 V reset

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

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

Bit Number	Bit Mnemonic	Description
7	**VREFS**	Reference voltage selection bit (Initial values are configured by the user and loaded from Code Options) 0: Configure ADC VREF as V_{DD} 1: Configure ADC VREF as internally correct 2.4V
4	**DISJTG**	IO/JTAG port switch bit 0: JTAG mode enable, P1.2\P1.3 are only used as tCLK\tDIO. Recommended Settings during developing and debugging stage. 1:Normal mode, P1.2\P1.3 can be used as normal I/O port, JTAG function is invalid. Recommended setting for mass production burning stage.
3 ~ 2	**IAPS[1: 0]**	EEPROM and IAP Area Selection Bits 00: Code memory prohibits IAP operations, only EEPROM data memory is used for data storage 01: last 0.5k code memory allows IAP operation (3E00H ~ 3FFFH) 10: Last 1k code memory allows IAP operation (3C00H ~ 3FFFH) 11: All code memory allows IAP operation (0000H ~ 3FFFH)
6 ~ 4, 1 ~ 0	**-**	Reserved

5.2.1 Customer-Option-related Registers Operation Instructions

Option-related SFRs reading and writing operations are controlled by both OPINX and OPREG registers, with their respective position of Option SFR depending on OPINX and its value written to option-related SFR depending on register OPREG:

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

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

For example: To configure ENWDT to 1, specific operation method is shown below:

C program example:

```
        OPINX = 0xC1;                //Write OP_CTM0 address into OPINX register
        OPREG = 0x01;                //Set ENWDT to 1
```

Assembler program example:

```
        MOV OPINX, #C1H             ;Write OP_HRCR address into OPINX register
        MOV OPREG, #80H              ;Set ENWDT to 1
```

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

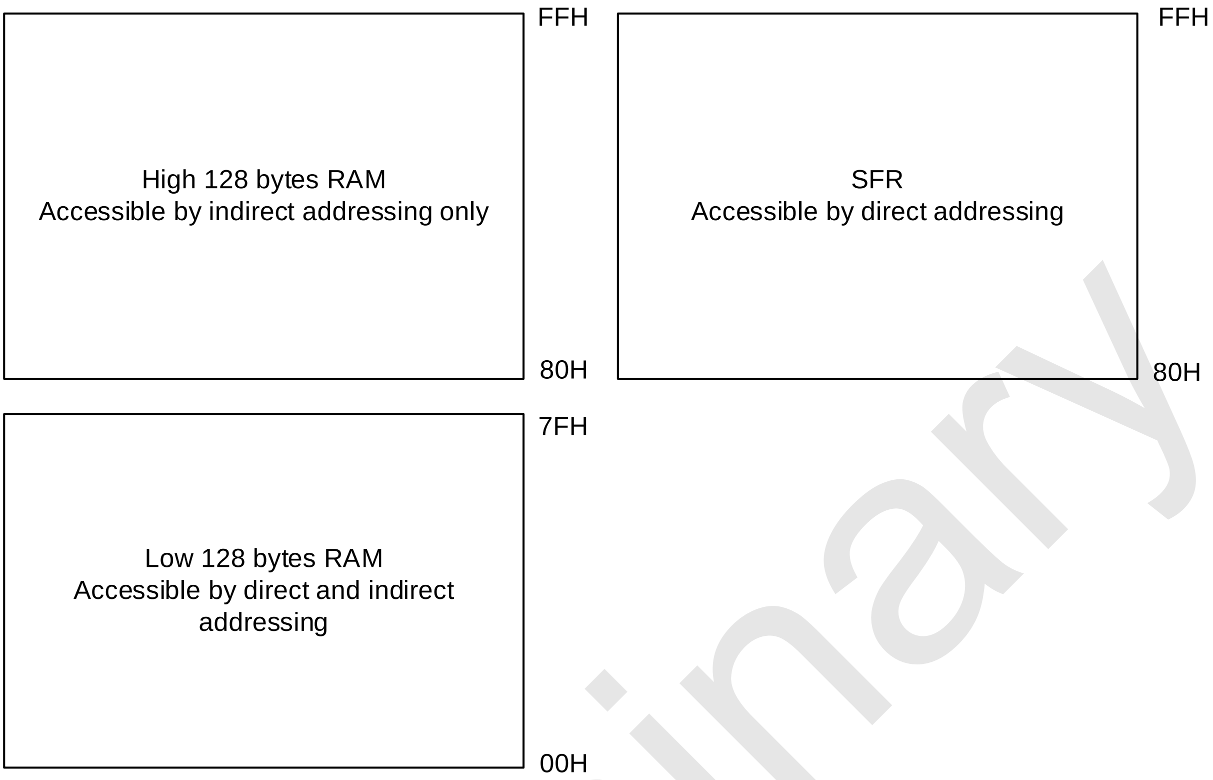
The SC92F84HX Microcontroller Unit, which integrates SRAM of 1 Kbytes, is divided into internal 256 bytes RAM and external 768 bytes RAM. The address of Internal RAM ranging from 00H to FFH, in which high 128 bytes (address of from 80H to FFH) only addressed indirectly and low 128 bytes (address of from 00H to 7FH) addressed both directly and indirectly).

The address of SFRs is also ranging from 80H to FFH. But the difference between SFR and internal high 128 bytes SRAM is that the former is addressed directly but the latter addressed indirectly only.

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

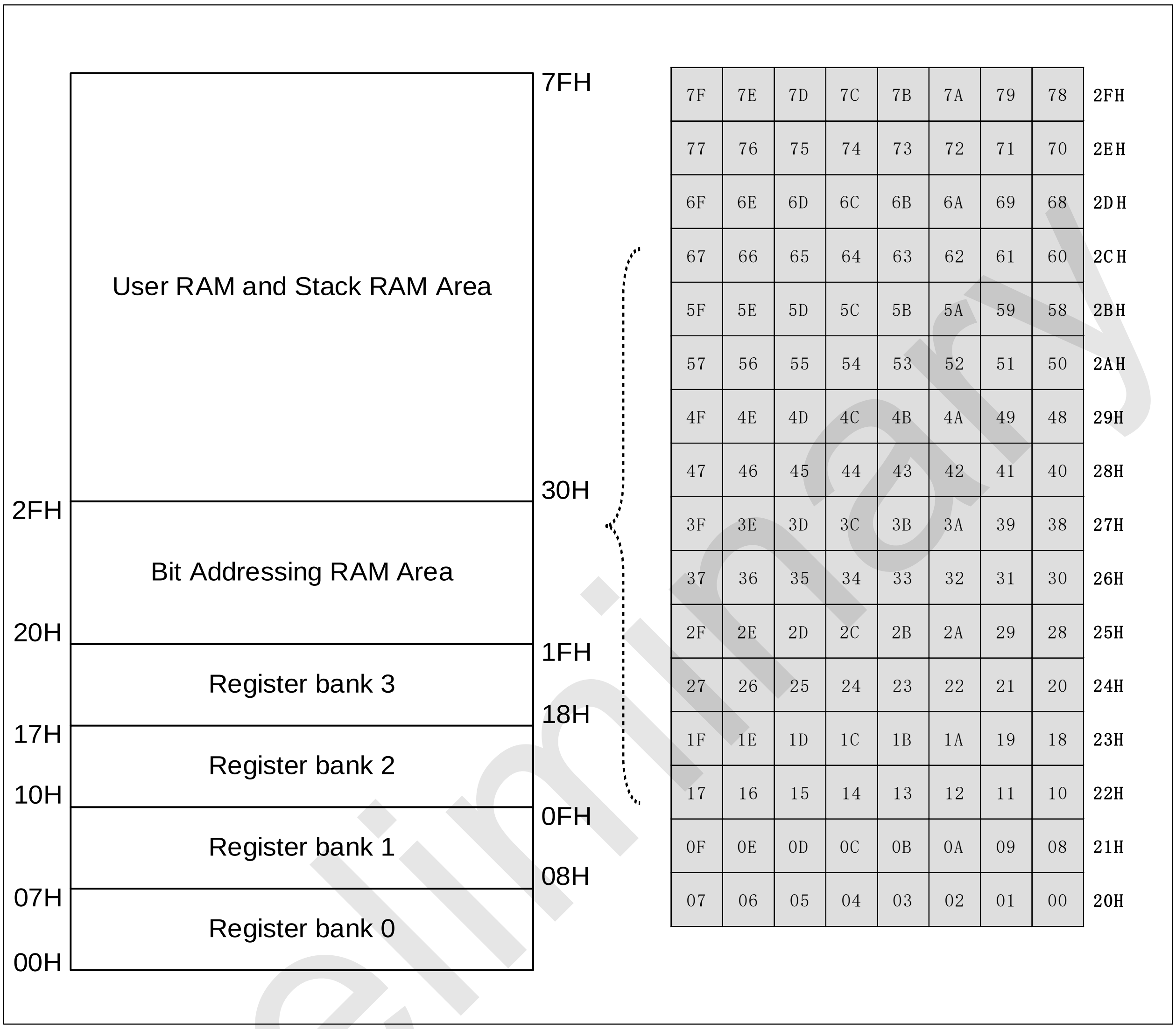
5.3.1 Internal 256 bytes SRAM

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 bits RS1 and RS0 of PSW register. Using Register bank 0 ~ 3 can accelerate arithmetic speed;② Bit addressing area , 20H ~2FH; user can use it as normal RAM or bitwise addressing RAM; for the latter, the bit address is from 00H to 7FH (bitwise addressing is different from normal SRAM byte-oriented addressing), which can be distinguished by instructions in program; ③ User RAM and stack area, the 8-bit stack pointer will point to stack area after the SC92F84HX reset; in general, users can set initial value in initializer, which is recommended to configure in the unit interval from E0H to FFH.


FFH
High 128 bytes RAM
Accessible by indirect addressing only
80H
FFH
SFR
Accessible by direct addressing
80H
7FH
Low 128 bytes RAM
Accessible by direct and indirect addressing
00H

Internal 256 bytes RAM Structure Diagram

Internal low 128 bytes RAM structure is shown below:



7F	7E	7D	7C	7B	7A	79	78	2FH
77	76	75	74	73	72	71	70	2EH
6F	6E	6D	6C	6B	6A	69	68	2DH
67	66	65	64	63	62	61	60	2CH
5F	5E	5D	5C	5B	5A	59	58	2BH
57	56	55	54	53	52	51	50	2AH
4F	4E	4D	4C	4B	4A	49	48	29H
47	46	45	44	43	42	41	40	28H
3F	3E	3D	3C	3B	3A	39	38	27H
37	36	35	34	33	32	31	30	26H
2F	2E	2D	2C	2B	2A	29	28	25H
27	26	25	24	23	22	21	20	24H
1F	1E	1D	1C	1B	1A	19	18	23H
17	16	15	14	13	12	11	10	22H
0F	0E	0D	0C	0B	0A	09	08	21H
07	06	05	04	03	02	01	00	20H

SRAM Structure Diagram

5.3.2 External 768 bytes SRAM

The external 768 bytes RAM (SRAM) can be accessed by instruction ‘’MOVX @DPTR’’ or instruction MOVX A ,@Ri or MOVX @Ri,A together whit EXADH register. EXADH register stores high address of external SRAM; Ri register stores low 8-bit address of external SRAM.

EXADH (F7H) External SRAM Operating Address High (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Mnemonic	-	-	-	-	-	EXADH [1:0]		

POR	x	x	x	x	x	0	0	0

Bit Number	Bit Mnemonic	Description
1 ~ 0	**EXADH [1: 0]**	External SRAM Address High position
7 ~ 2	-	Reserved

5.3.3 Additional PWM SRAM

The RAM addresses from 0334H to 033FH serve as additional PWM duty cycle adjustment registers and are readable and writable

6 Special Function Register (SFR)

6.1 SFR Mapping

The SC92F84HX 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 shall use direct addressing for addressing.

The name and address of the SC92F84HX 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	EXADH
E8h	-	EXA0	EXA1	EXA2	EXA3	EXBL	EXBH	OPERCON
E0h	ACC	-	-	-	-	-	-	-
D8h	P5	P5CON	P5PH	-	-	-	-	-
D0h	PSW	-	-	-	-	-	-	-
C8h	TXCON	TXMOD	RCAPXL	RCAPXH	TLX	THX	TXINX	WDTCON
C0h	-	-	-	-	-	-	-	-
B8h	IP	IP1	INT0F	INT0R	INT1F	INT1R	-	-
B0h	-	-	-	-	-	ADCCFG2	-	-
A8h	IE	IE1	ADCCFG3	ADCCFG0	ADCCFG1	ADCCON	ADCVL	ADCVH
A0h	P2	P2CON	P2PH	-	-	-	-	-
98h	SCON	SBUF	P0CON	P0PH	-	SSCON0	SSCON1	SSDAT
90h	P1	P1CON	P1PH	-	-	SSCON2	-	IOHCON
88h	TCON	TMOD	TL0	TL1	TH0	TH1	TMCON	OTCON

80h	P0	SP	DPL	DPH	-	-	-	PCON
	Bit Addressable	Not Bit Addressable						

Notes:

1. **Hollow space of SFR refers to the fact that there is no such register RAM, it is not recommended for user to use.**
2. **The address of SFR for system configuration is F1H ~ FFH, user use it may result in system exceptions. User are not allowed to conduct clearing or other operations to these registers during the system initialization process.**

6.2 SFR Instructions

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

Mnemonic	Add	Description	7	6	5	4	3	2	1	0	POR
P0	80H	P0 Data Register	P07	P06	P05	P04	P03	P02	P01	P00	00000000b
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	-	-	-	RST	-	STOP	IDL	0xxx0x00b
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 bits	TL0[7: 0]								00000000b
TL1	8BH	Timer1 Low 8 bits	TL1[7: 0]								00000000b
TH0	8CH	Timer0 High 8 bits	TH0[7: 0]								00000000b
TH1	8DH	Timer1 High 8 bits	TH1[7: 0]								00000000b
TMCON	8EH	Timer Frequency Control Register	-	-	-	T3OS	-	T2OS	T1FD	T0FD	xxx0x000b

OTCON	8FH	Output Control Register	SSMOD[1: 0]		UART0OS[1:0]		-	-	SSIOS	-	0000xx0xb
P1	90H	P1 Data Register	P17	P16	P15	P14	P13	P12	P11	P10	00000000b
P1CON	91H	P1 I/O 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	SSCON2[7: 0]								00000000b
IOHCON	97H	IOH Setup Register	P2H[1: 0]		P2L[1: 0]		P0H[1: 0]		P0L[1: 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 I/O Control Register	P0C7	P0C6	P0C5	P0C4	P0C3	P0C2	P0C1	P0C0	00000000b
P0PH	9BH	P0 Pull-up Resistor Control Register	P0H7	P0H6	P0H5	P0H4	P0H3	P0H2	P0H1	P0H0	00000000b
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 I/O 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	ET4	ET3	-	ETK	-	EBTM	-	ESSI	00x0x0x0b
ADCCFG3	AAH	ADC Configuration Register 3	-	-	-	-	EAIN19	EAIN18	EAIN17	EAIN16	xxxx0000b
ADCCFG0	ABH	ADC Configuration Register 0	EAIN7	EAIN6	EAIN5	EAIN4	EAIN3	EAIN2	EAIN1	EAIN0	00000000b

ADCCFG1	ACH	ADC Configuration Register 1	EAIN15	EAIN14	EAIN13	EAIN12	EAIN11	EAIN10	EAIN9	EAIN8	00000000b
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
ADCCFG2	B5H	ADC Configuration Register 2	-	-	-	-	LOWSP	ADCCK[2:0]			xxxx0000b
IP	B8H	Interrupt Priority Control Register	-	IPADC	IPT2	IPUART	IPT1	IPINT1	IPT0	IPINT0	x0000000b
IP1	B9H	Interrupt Priority Control Register 1	IPT4	IPT3	-	IPTK	-	IPBTM	-	IPSSI	00x0x0x0b
INT0F	BAH	INT0 Falling Edge Interrupt Control Register	-	-	-	-	INT0F3	INT0F2	INT0F1	INT0F0	xxxx0000b
INT0R	BBH	INT0 Rising Edge Interrupt Control Register	-	-	-	-	INT0R3	INT0R2	INT0R1	INT0R0	xxxx0000b
INT1F	BCH	INT1 Falling Edge Interrupt Control Register	-	-	-	-	INT1F3	INT1F2	INT1F1	INT1F0	xxxx0000b
INT1R	BDH	INT1 Rising Edge Interrupt Control Register	-	-	-	-	INT1R3	INT1R2	INT1R1	INT1R0	xxxx0000b
TXCON	C8H	Timer2/3/4 Control Register	TFX	EXFX	RCLK	TCLK	EXENX	TRX	C/TX	CP/RLX	00000000b
TXMOD	C9H	Timer2/3/4 Operating Mode Register	TXFD	-	EPWMN1	EPWMN0	INVN1	IVNN0	TXOE	DCEN	0x000000b
RCAPXL	CAH	Timer2/3/4 Reload Low 8 bits	RCAPXL[7: 0]								00000000b
RCAPXH	CBH	Timer2/3/4 Reload High 8 bits	RCAPXH[7: 0]								00000000b
TLX	CCH	Timer2/3/4 Low 8 bits	TLX[7: 0]								00000000b
THX	CDH	Timer2/3/4 High 8 bits	THX[7: 0]								00000000b
TXINX	CEH	Timer Control Register Pointer	-	-	-	-	-	TXINX[2:0]			xxxxx010b
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
P5	D8H	P5 Data Register	-	-	-	-	-	-	P51	P50	xxxxxx00b
P5CON	D9H	P5 I/O Control Register	-	-	-	-	-	-	P5C1	P5C0	xxxxxx00b
P5PH	DAH	P5 Pull-up Resistor Control Register	-	-	-	-	-	-	P5H1	P5H0	xxxxxx00b
ACC	E0H	Accumulator	ACC[7: 0]								00000000b
EXA0	E9H	Extended Accumulator 0	EXA[7: 0]								00000000b
EXA1	EAH	Extended Accumulator 1	EXA[15: 8]								00000000b
EXA2	EBH	Extended Accumulator 2	EXA[23: 16]								00000000b
EXA3	ECH	Extended Accumulator 3	EXA[31: 24]								00000000b
EXBL	EDH	Extended B Register L	EXB [7: 0]								00000000b
EXBH	EEH	Extended B Register H	EXB [15: 8]								00000000b
OPERCON	EFH	Arithmetic Control Register	OPERS	MD	-	-	-	-	-	CHKSUMS	00xxxxx0b
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
EXADH	F7H	External SRAM Operating Address High	-	-	-	-	-	-	EXADH [1: 0]		xxxxxx00b

BTMCON	FBH	Low Frequency Timer Control Register	ENBTM	BTMIF			BTMFS[3:0]				00xx0000b
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 PWM2~4 Duty Cycle Adjustment Register(R/W)

地址	7	6	5	4	3	2	1	0	POR
0334H	PDT20[15:8]								00000000b
0335H	PDT20[7:0]								00000000b
0336H	PDT21[15:8]								00000000b
0337H	PDT21[7:0]								00000000b
0338H	PDT30[15:8]								00000000b
0339H	PDT30[7:0]								00000000b
033AH	PDT31[15:8]								00000000b
203BH	PDT31[7:0]								00000000b
033CH	PDT40[15:8]								00000000b
033DH	PDT40[7:0]								00000000b
033EH	PDT41[15:8]								00000000b
033FH	PDT41[7:0]								00000000b

6.2.2 C51 Core SFRs

Program Counter (PC)

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 number~~s~~ 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 quotient 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 Mnemonic	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

<table>
<tr><th>Bit Number</th><th>Bit Mnemonic</th><th>Description</th></tr>
<tr><td>7</td><td><b>CY</b></td><td>Carry Flag bit
1: The top digit of add operation has carry bit or the top digit of subtraction operation has the borrow digit
0: The top digit of add operation has no carry bit or the top digit of subtraction operation has no borrow digit</td></tr>
<tr><td>6</td><td><b>AC</b></td><td>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</td></tr>
<tr><td>5</td><td><b>F0</b></td><td>User flag bit</td></tr>
<tr><td>4 ~ 3</td><td><b>RS1、RS0</b></td><td>Register banks selection bits
<table>
<tr><td>RS1</td><td>RS0</td><td>Current Selected Register banks 0 ~ 3</td></tr>
<tr><td>0</td><td>0</td><td>Group 0 (00H ~ 07H)</td></tr>
<tr><td>0</td><td>1</td><td>Group 1 (08H ~ 0FH)</td></tr>
</table>
</td></tr>
</table>

		1 0 Group 2 (10H ~ 17H) 1 1 Group 3 (18H ~ 1FH)
2	**OV**	Overflow flag bit
1	**F1**	F1 flag bit User customized flag
0	**P**	Parity flag bit. 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)

Data Pointer DPTR (82H, 83H)

The Data pointer DPTR is a 16-bit dedicated register, which is composed of Low byte DPL (82H) and High byte 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 System Clock

7.1 Power Circuit

The SC92F84HX 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. The user can search for specific configuration contents in 17 Analog-to-digital converter (ADC).

7.2 Power-on Reset

After the SC92F84HX 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

The SC92F84HX will always be in reset mode. There will not be a valid clock until the voltage supplied to the SC92F84HX 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 would be completed.

7.2.2 Loading Information Stage

There is a preheating counter inside the SC92F84HX. 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 of 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.

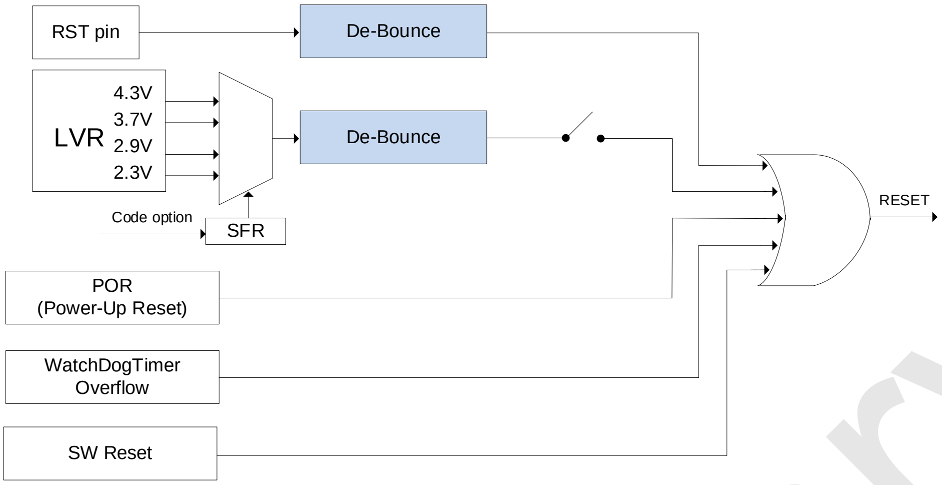
7.2.3 Normal Operating Stage

After the loading information stage has been completed, the SC92F84HX 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 Modes

The SC92F84HX has 5 kinds of reset modes, ①~④ are hardware resets: ① External RST reset ②Low-voltage reset (LVR) ③Power-on reset (POR) ④Watchdog (WDT) reset ⑤Software reset.

The reset section circuit diagram of SC92F84HX is as follows:



SC92F84HX Reset Circuit Diagram

7.3.1 External Reset

External reset is to supply a certain width reset pulse signal to the SC92F84HX from the RST pin to realize the SC92F84HX reset.

User can configure P1.1 pin as RST (reset pin) in Customer Option via PC program software before programming.

7.3.2 Low-voltage Reset (LVR)

The SC92F84HX provides a low-voltage reset circuit. There are 4-level LVR voltage options: 4.3V, 3.7V, 2.9V and 2.3V. The default is the Option value written by user. A reset will be set when the VDD voltage is lower than the threshold voltage for low-voltage reset and the duration is greater than T_{LVR}. T_{LVR} is approximately 30μs and T_{LVR} is the debounce time for LVR.

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

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	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 Mnemonic	Description

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

7.3.3 Power-on Reset (POR)

The SC92F84HX 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)

The SC92F84HX has a WDT, the clock source of which is the internal 32 kHz oscillator. User 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 Mnemonic	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 Mnemonic	Description
7	**ENWDT**	WDT control bit (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 Mnemonic	-	-	-	CLRWDT	-	WDTCKS[2: 0]		
R/W	-	-	-	R/W	-	R/W		
POR	x	x	x	0	x	0	0	0

Bit Number	Bit Mnemonic	Description
4	**CLRWDT**	Clear WDT (Only valid when set to 1) 1: WDT counter restart, cleared by system hardware
2 ~ 0	**WDTCKS [2: 0]**	WDT clock selection bits

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 ~ 5, 3	-	Reserved	

7.3.5 Software Reset

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

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

Bit Number	Bit Mnemonic	Description
3	**RST**	Software Reset Control Bit: 0: The program runs normally; 1: The CPU resets immediately after this bit is written with "1".

7.3.6 Register Reset Value

During reset, most registers are set to their initial values and the WDT remains disable. The register of PORT is FFh. 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.

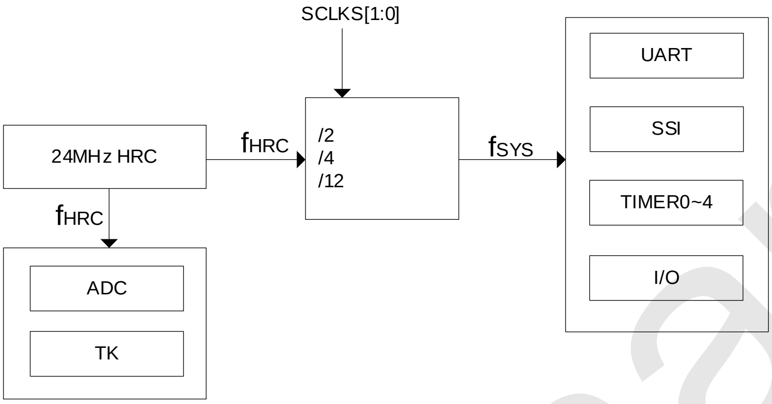
7.4 High-speed RC Oscillator

The SC92F84HX has a built-in adjustable high-precision HRC. HRC is precisely calibrated to 24 MHz@5V/25℃ when delivery. The user can set system clock as 12/6/2MHz by programmer Code Option. 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 (-40℃ ~ 85℃), the deviation is within ±1%.

In order to enhance the reliability of the system, the SC92F84HX has a system clock monitoring circuit built in.

When user chooses the crystal oscillation as system clock source 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 TouchKey circuit is f_{HRC} = 24MHz



The SC92F84HX Internal Clock Relationship

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

Bit Number	7	6	5	4	3	2	1	0
Bit Mnemonic	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 Mnemonic	Description
6	ENXTL	External 32.768kHz crystal oscillator selection bit 0: external 32.768kHz crystal Interface disable, P5.0 and P5.1 valid 1: external 32.768kHz crystal Interface enable, P5.0 and P5.1 invalid
5 ~ 4	SCLKS[1: 0]	System clock frequency selection bits: 00: reserved;

		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;

The SC92F84HX has a special function: the user can modify SFR value to adjust frequency of HRC within certain scope. User can realize this operation by configuring OP_HRCR register. For configuration method of this register, refer to5.2.1 Customer-Option-related Registers Operation Instructions.

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

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

Bit Number	Bit Mnemonic	Description
7 ~ 0	OP_HRCR[7: 0]	**HRC frequency change register** User can change high-frequency oscillator frequency f_{HRC} by modifying the value of this register, and then change the IC system clock frequency f_{SYS}: 1. Initial value OP_HRCR[s] after OP_HRCR[7: 0] power-on is a fixed value, which guarantee f_{HRC} is 24MHz, there may be difference in OP_HRCR[s] of each IC 2. When initial value is OP_HRCR[s], IC system clock frequency f_{SYS} can set specifically as 12/6/2MHz by Option. For each change of 1 for OP_HRCR [7: 0], the change of f_{SYS} frequency is about 0.23%. The relationship between OP_HRCR [7: 0] and output frequency f_{SYS} is shown as follows:

OP_HRCR [7:0]	f_{SYS} actual output frequency (taking 12M as an example)

OP_HRCR [s]-n	12000* (1-0.23%*n)kHz
...	....
OP_HRCR [s]-2	12000* (1-0.23%*2) = 11944.8kHz
OP_HRCR [s]-1	12000* (1-0.23%*1) = 11972.4kHz
OP_HRCR [s]	12000kHz
OP_HRCR [s]+1	12000* (1+0.23%*1) = 12027.6kHz
OP_HRCR [s]+2	12000* (1+0.23%*2) = 12055.2kHz
...	...
OP_HRCR [s]+n	12000* (1+0.23%*n)kHz

Notes:

1. **The value of OP_HRCR[7:0] after each power-on of the IC is the value of high-frequency oscillator frequency f_{HRC} closest to 24MHz; the user can modify the value of HRC after each power-on by means of EEPROM to make IC system clock frequency f_{SYS} work at the frequency the user needs.**

2. **To guarantee IC operating reliably, the maximum operating frequency of IC shall not exceed 10% of 12MHz, which is 13.2MHz;**

3. **The user shall confirm the change of HRC frequency will not influence other functions.**

7.5 Low-speed RC Oscillator

The SC92F84HX is equipped with a built-in 32kHz RC oscillation circuit and 32.768kHz crystal oscillator Interface, 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 (FBH) Low-Frequency Timer Control Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Mnemonic	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 Mnemonic	Description
7	**ENBTM**	Low-frequency Base Timer start control bit 0: Base Timer and it's clock source disable 1: Base Timer and it's clock source enable
6	**BTMIF**	Base Timer interrupt application flag bit When CPU receives Base Timer interrupt, this flag will be cleared automatically by hardware.
3 ~ 0	**BTMFS [3: 0]**	Low-frequency clock interrupt frequency selection bits 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: an interrupt is generated for every 8.0s 1010: an interrupt is generated for every 16.0s 1011: an interrupt is generated for every 32.0s

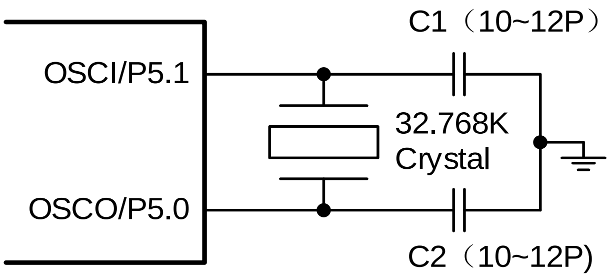
		1100: an interrupt is generated for every 64.0s 1101: an interrupt is generated for every 128.0s 1110: an interrupt is generated for every 256.0s 1111: reserved
5 ~ 4	-	Reserved

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

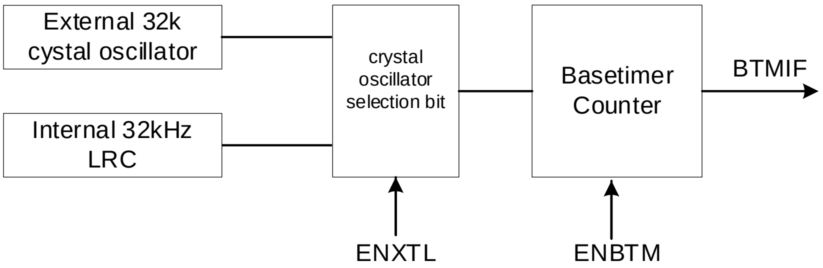
Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	ENWDT	ENXTL	SCLKS[1: 0]		DISRST	DISLVR	LVRS[1: 0]	
R/W	R/W	R/W	R/W		R/W	R/W	R/W	
POR	n	n	n		n	n	n	

Bit Number	Bit Mnemonic	Description
6	**ENXTL**	External 32.768kHz crystal oscillator selection bit 0: External 32.768kHz crystal Interface disable, P5.0 and P5.1 valid 1: External 32.768kHz crystal Interface enable, P5.0 and P5.1 invalid

The external connection circuit diagram for using an external 32.768kHz oscillator on P5.0/P5.1 as the BaseTimer is as follows:



The relationship diagram for selecting internal and external oscillators for the Base Timer is as follows:


External 32k cystal oscillator
Internal 32kHz LRC
crystal oscillator selection bit
Basetimer Counter
BTMIF
ENXTL
ENBTM

7.6 Power Saving Modes

The SC92F84HX provides a SFR PCON, the 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-frequency 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-frequency clock interrupt, TK interrupt, and external reset input.

When PCON.0 = 1, the program 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 Mnemonic	SMOD	-	-	-	RST	-	STOP	IDL
R/W	W	-	-	-	W	-	W	W
POR	0	x	x	x	n	x	0	0

Bit Number	Bit Mnemonic	Description
1	**STOP**	STOP mode control bit 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 bit 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;           //Set to 1 for PCON bit1 STOP bit, configure MCU to enter STOP mode
_nop_ ();               //At least 8 _nop_ () required
_nop_ ();
_nop_ ();
_nop_ ();
_nop_ ();
_nop_ ();
_nop_ ();
……
```

Assembly program example:

```
ORL PCON, #02H          ; Set to 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
……
```

8 CPU and Function System

8.1 CPU

CPU used by the SC92F84HX 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 the SC92F84HX 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 register 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 number 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 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 (Transmit the bit operation register with address of 20H into carry bit C)

9 Interrupt

The SC92F84HX provides 12 interrupt sources: Timer0~4, INT0, INT1, ADC, UART, SSI, Base Timer and TK. These 12 interrupt sources are equipped with 2-level interrupt priority capability and each interrupt source can be individually configured in high priority or low priority. As for 2 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 the SC92F84HX interrupt source, interrupt vector and related control bit are shown below:

Interrupt Source	**Interrupt condition**	**Interrupt Flag**	**Interrupt Enable Control**	**Interrupt Priority Control**	**Interrupt Vector**	**Query Priority**	**Interrupt Number (C51)**	**Flag Clear Mode**	**Capability of Waking 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	Timer1 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	Timer2 overflow	TFX	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
BTM	Base timer overflow	BTMIF	EBTM	IPBTM	004BH	10	9	H/W Auto	Yes
TK	TouchKey counter	TKIF	ETK	IPTK	005B	12	11	H/W Auto	Yes

	overflow								
Timer3	Timer3 overflow	TFX	ET3	IPT3	006BH	14	13	Must be cleared by user	No
Timer4	Timer4 overflow	TFX	ET4	IPT4	0073H	15	14	Must be cleared by user	No

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 generates when Timer2/3/4 overflows and the interrupt flag TFX is set to “1”. Once Timer2/3/4 interrupt generates, the hardware would not automatically clear TFX bit, which must be cleared by the user’s software.

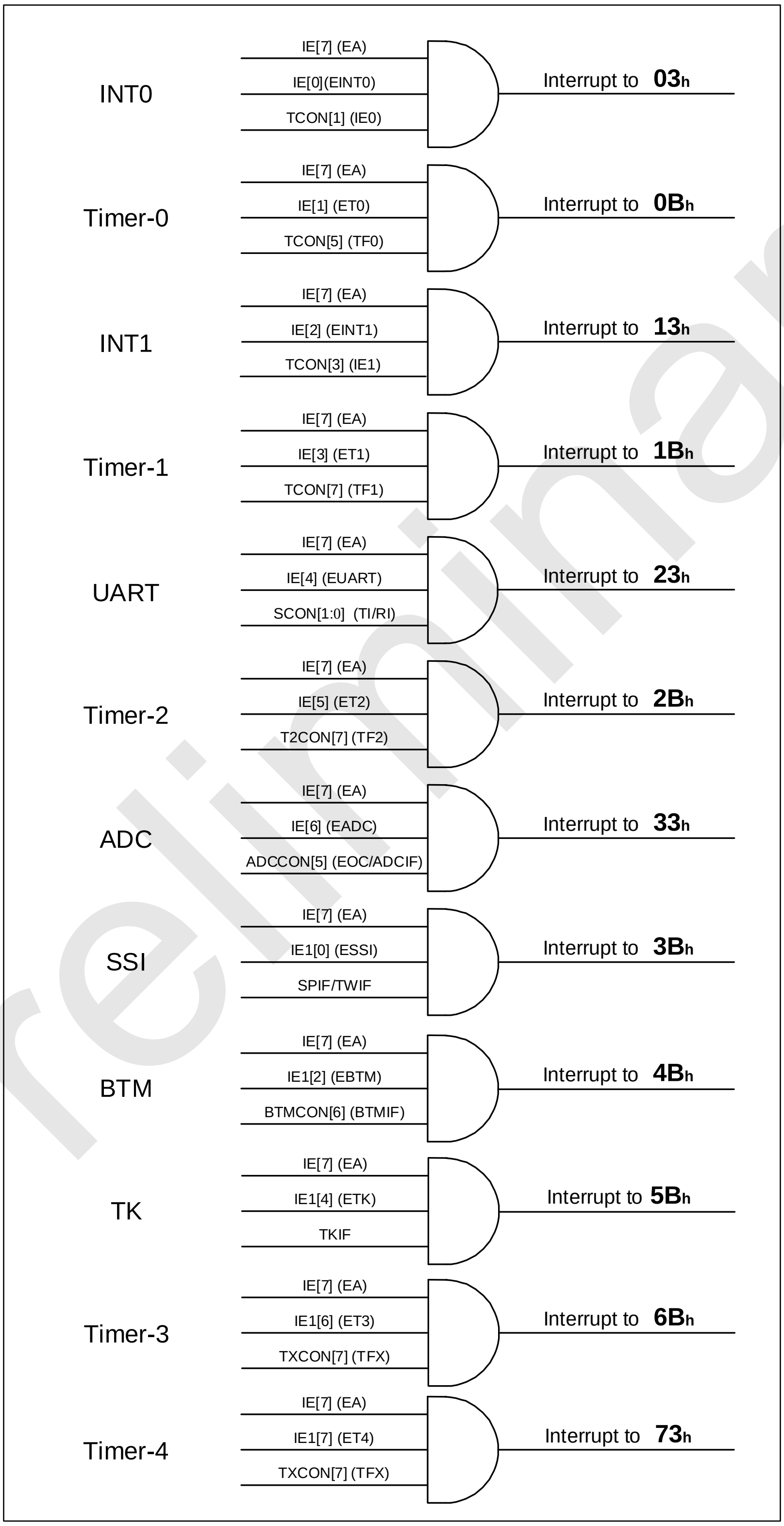
ADC Interrupt: After ADC conversion is completed, ADC interrupt generates, whose interrupt flag is the ADC conversion completion flag EOC/ADCIF (ADCCON.5). When user starts ADCS conversion, EOC will be reset automatically by hardware. Once conversion completes, EOC would be set to “1” automatically by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

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 generates. When the microcontroller unit serves SSI interrupt, the interrupt flag SPIF/TWIF must be cleared by software.

External Interrupt INT0 ~ 1: When any external interrupt pin meets the interrupt conditions, external interrupt generates. There are 4 external interrupt sources for INT0 and 4 external interrupt sources for INT1, which can be set in rising edge, falling edge or dual edge interrupt trigger mode by setting SFRs (INTxF and INTxR). User can set the priority level of each interrupt through IP register. Besides, external interrupt INT0 ~ 1 can also wake up STOP mode of microcontroller unit.

9.2 Interrupt Structure Diagram

The SC92F84HX interrupt structure is shown in the figure below:



The SC92F84HX Interrupt Structure and Vector

9.3 Interrupt Priority

The SC92F84HX 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 cannot 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 versa

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

Interrupt query sequence: As for the sequence of that the SC92F84HX microcontroller 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 the interrupt with bigger query number.

9.4 Interrupt Processing Flow

When any interrupt generates 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 interrupt.

9.5 Interrupt-related Registers

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

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Bit Mnemonic	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 Mnemonic	Description
7	**EA**	Global interrupt enable control bit 0: Disable all interrupts 1: Enable all interrupts
6	**EADC**	ADC interrupt enable control bit 0: Disable ADC interrupts 1: Interrupt is allowed upon completing ADC conversion
5	**ET2**	Timer2 interrupt enable control bit 0: Disable Timer2 interrupt 1: Enable Timer2 interrupt
4	**EUART**	UART interrupt enable control bit 0: Disable UART interrupt 1: Enable UART interrupt
3	**ET1**	Timer1 interrupt enable control bit 0: Disable Timer1 interrupt 1: Enable Timer1 interrupt
2	**EINT1**	External interrupt 1 enable control 0: Disable INT1 interrupt 1: Enable INT1 interrupt

1	**ET0**	Timer0 interrupt enable control bit 0: Disable Timer0 interrupt 1: Enable Timer0 interrupt
0	**EINT0**	External interrupt 0 enable control bit 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 Mnemonic	-	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 Mnemonic	Description
6	**IPADC**	ADC interrupt priority selection bit 0: ADC interrupt priority is low 1: ADC interrupt priority is high
5	**IPT2**	Timer2 interrupt priority selection bit 0: Timer2 interrupt priority is low 1: Timer2 interrupt priority is high
4	**IPUART**	UART interrupt priority selection bit 0: UART interrupt priority is low 1: UART interrupt priority is high

3	**IPT1**	Timer1 interrupt priority selection bit 0: Timer1 interrupt priority is low 1: Timer1 interrupt priority is high
2	**IPINT1**	INT1 interrupt priority selection bit 0: INT1 interrupt priority is low 1: INT1 interrupt priority is high
1	**IPT0**	Timer 0 interrupt priority selection bit 0: Timer0 interrupt priority is low 1: Timer0 interrupt priority is high
0	**IPINT0**	INT0 interrupt priority selection bit 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**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	ET4	ET3	-	ETK	-	EBTM	-	ESSI
R/W	R/W	R/W	-	R/W	-	R/W	-	R/W
POR	0	0	x	0	x	0	x	0

Bit Number	Bit Mnemonic	Description
7	**ET4**	Timer4 interrupt enable control bit 0: Disable Timer4 interrupt

		1: Enable Timer4 interrupt
6	**ET3**	Timer3 interrupt enable control bit 0: Disable Timer3 interrupt 1: Enable Timer3 interrupt
4	**ETK**	TouchKey interrupt enabling control bit 0: Disable TouchKey interrupt 1: Enable TouchKey interrupt
2	**EBTM**	Base Timer interrupt enabling control bit 0: Disable Base Timer interrupt 1: Enable Base Timer interrupt
0	**ESSI**	Three-in-on serial interrupt enabling control 0: Disable serial port interrupt 1: Enable serial port interrupt
5,3,1	**-**	Reserved

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

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	IPT4	IPT3	-	IPTK	-	IPBTM	-	IPSSI
R/W	R/W	R/W	-	R/W	-	R/W	-	R/W
POR	0	0	x	0	x	0	x	0

Bit Number	Bit Mnemonic	Description
7	**IPT4**	Timer 4 interrupt priority selection bit

		0: Timer4 interrupt priority is low 1: Timer4 interrupt priority is high
6	**IPT3**	Timer 3 interrupt priority selection bit 0: Timer3 interrupt priority is low 1: Timer3 interrupt priority is high
4	**IPTK**	TouchKey interrupt priority selection bit 0: TouchKey interrupt priority is low 1: TouchKey interrupt priority is high
2	**IPBTM**	Base Timer interrupt priority selection bit 0: Base Timer interrupt priority is low 1: Base Timer interrupt priority is high
0	**IPSSI**	Three-in-on serial interrupt priority selection bit 0: SSI interrupt priority is low 1: SSI interrupt priority is high
5,3,1	**-**	Reserved

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

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	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 Mnemonic	Description
3	**IE1**	INT1 overflow interrupt request flag bit When INT1 overflow occurs, interrupt generates, hardware set IE1 to "1"; when the application is interrupted, upon C PU responds, the hardware resets it to "0"
1	**IE0**	INT0 overflow interrupt request flag bit When INT1 overflow occurs, interrupt generates, hardware set IE0 to "1"; when 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 Mnemonic	-	-	-	-	INT0F3	INT0F2	INT0F1	INT0F0
R/W	-	-	-	-	R/W	R/W	R/W	R/W
POR	x	x	x	x	0	0	0	0

Bit Number	Bit Mnemonic	Description
3 ~ 0	**INT0Fn** **(n=0 ~ 3)**	INT0 falling edge interrupt control bit 0: INT0n falling edge interrupt off 1: INT0n falling edge interrupt enabling
7 ~ 4	**-**	Reserved

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

Bit Number	7	6	5	4	3	2	1	0
Bit Mnemonic	-	-	-	-	INT0R3	INT0R2	INT0R1	INTR0
R/W	-	-	-	-	R/W	R/W	R/W	R/W
POR	x	x	x	x	0	0	0	0

Bit Number	Bit Mnemonic	Description
3 ~ 0	**INT0Rn** **(n=0 ~ 3)**	INT0 rising edge interrupt control bit 0: INT0n rising edge interrupt off 1: INT0n rising edge interrupt enabling
7 ~ 4	**-**	Reserved

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

Bit Number	7	6	5	4	3	2	1	0
Bit Mnemonic	-	-	-	-	INT1F3	INT1F2	INT1F1	INT1F0
R/W	-	-	-	-	R/W	R/W	R/W	R/W
POR	x	x	x	x	0	0	0	0

Bit Number	Bit Mnemonic	Description
3 ~ 0	**INT1Fn** **(n=0 ~ 3)**	INT1 falling edge interrupt control bit 0: INT1n falling edge interrupt off 1: INT1n falling edge interrupt enabling

7 ~ 4	**-**	Reserved

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

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	-	-	-	-	INT1R3	INT1R2	INT1R1	INT1R0
R/W	-	-	-	-	R/W	R/W	R/W	R/W
POR	x	x	x	x	0	0	0	0

Bit Number	Bit Mnemonic	Description
3 ~ 0	**INT1Rn** **(n=0 ~ 3)**	INT1 rising edge interrupt control bit 0: INT1n rising edge interrupt off 1: INT1n rising edge interrupt enabling
7 ~ 4	**-**	Reserved

10 Timer/Counter T0 and T1

The SC92F84HX has five 16-bit Timer/Counters, Among them, Timer0/1 have independent register groups, while Timer2~4 share a register group. This section mainly introduces the functions of Timer0~1, and details about Timer2~4 can be found in the next section. They all can operate in two modes: counter mode and timer mode. The operating modes selected by bit C/Tx in the SFR TMOD. T0 and T1 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 P1.0/T0 and P0.0/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 Registers

Mnemonic	Add	Description	7	6	5	4	3	2	1	0	POR
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 byte	TL0[7: 0]								00000000b
TL1	8BH	Timer1 Low byte	TL1[7: 0]								00000000b
TH0	8CH	Timer0 High byte	TH0[7: 0]								00000000b
TH1	8DH	Timer1 High byte	TH1[7: 0]								00000000b
TMCON	8EH	Timer Frequency Control Register	-	-	-	T30S	-	T2OS	T1FD	T0FD	xxx0x000b

Register instructions are shown below:

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

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	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 Mnemonic	Description
7	**TF1**	Timer1 overflow flag bit Set by hardware on Timer/Counter overflow. Cleared by hardware when the processor vectors to interrupt routine.
6	**TR1**	Timer1 run control bit Set/cleared by software to turn Timer/Counter on/off.
5	**TF0**	Timer0 overflow flag bit Set by hardware on Timer/Counter overflow. Cleared by hardware when the processor vectors to interrupt routine.
4	**TR0**	Timer0 run control bit Set/cleared by software to turn Timer/Counter on/off.
2, 0	**-**	Reserved

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

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	-	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			

<table>
<tr><th>Bit Number</th><th>Bit Mnemonic</th><th>Description</th></tr>
<tr><td>6</td><td><b>C/T1</b></td><td>Timer or Counter selector 1
0: Cleared for Timer operation (input from internal system clock fsys).
1: Set for Counter operation (input from external pin T1/P0.0).</td></tr>
<tr><td>5 ~ 4</td><td><b>M11, M01</b></td><td>Timer1 operating mode
<table>
<tr><td>Mode</td><td>M11</td><td>M01</td><td>Operation</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>13-bit TIMER/Counter, TL1 high 3 bits invalid</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>16-bit Timer/Counter, TL1 and TL3 are valid</td></tr>
<tr><td>2</td><td>1</td><td>0</td><td>8-bit Auto-Reload Mode.
TH1 holds a value which is reloaded into 8-bit Timer/CounterTL1 each time it overflows.</td></tr>
<tr><td>3</td><td>1</td><td>1</td><td>Timer/Counter 1 is stopped</td></tr>
</table></td></tr>
<tr><td>2</td><td><b>C/T0</b></td><td>Timer or Counter selector 0
0: Cleared for Timer operation (input from internal system clock fsys).
1: Set for Counter operation (input from external pin T0/P1.0).</td></tr>
</table>

<table>
<tr><td>1 ~ 0</td><td><b>M10, M00</b></td><td>Timer0 operating mode
<table>
<tr><td>Mode</td><td>M10</td><td>M00</td><td>Operation</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>13-bit TIMER/Counter, TL0 high 3 bits invalid</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>16-bit Timer/Counter, TL0 and TH0 are valiad</td></tr>
<tr><td>2</td><td>1</td><td>0</td><td>8-bit Auto-Reload Mode.
TH0 holds a value which is reloaded into 8-bit Timer/Counter TL0 each time it overflows.</td></tr>
<tr><td>3</td><td>1</td><td>1</td><td>Split Timer Mode.
TL0 is an 8-bit Timer/Counter controlled by the standard Timer0 control bits. TH0 is only an 8-bit timer controlled by Timer1 control bits</td></tr>
</table></td></tr>
<tr><td>7, 3</td><td><b>-</b></td><td>Reserved</td></tr>
</table>

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, and it’s 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 Mnemonic	-	-	-	T30S	-	T2OS	T1FD	T0FD
R/W	-	-	-	R/W	-	R/W	R/W	R/W
POR	x	x	x	0	x	0	0	0

Bit Number	Bit Mnemonic	Description
1	**T1FD**	T1 input frequency selection control bit 0: T1 clock source is $f_{SYS}/12$ 1: T1 clock source is f_{SYS}
0	**T0FD**	T0 input frequency selection control bit 0: T0 clock source is $f_{SYS}/12$ 1: T0 clock source is f_{SYS}

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

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	EA	EADC	ET2	-	ET1	EINT1	ET0	EINT0
R/W	R/W	R/W	R/W	-	R/W	R/W	R/W	R/W
POR	0	0	0	x	0	0	0	0

Bit Number	Bit Mnemonic	Description
3	**ET1**	Timer1 interrupt enable control bit 0：Disable Timer1 interrupt 1：Enable Timer1 interrupt
1	**ET0**	Timer0 interrupt enable control bit 0：Disable Timer0 interrupt 1：Enable Timer0 interrupt

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

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

Bit Number	Bit Mnemonic	Description
3	**IPT1**	Timer1 interrupt priority selection bit 0: Configure Timer1 interrupt priority as “low” 1: Configure Timer1 interrupt priority as ”high”
1	**IPT0**	Timer0 interrupt priority selection bit 0: Configure Timer0 interrupt priority as “low” 1: Configure Timer0 interrupt priority as “high”

10.2 T0 Operating Modes

Timer0 can be configured in one of four operating modes by setting the bit pairs (M10, M00) in the TMOD register.

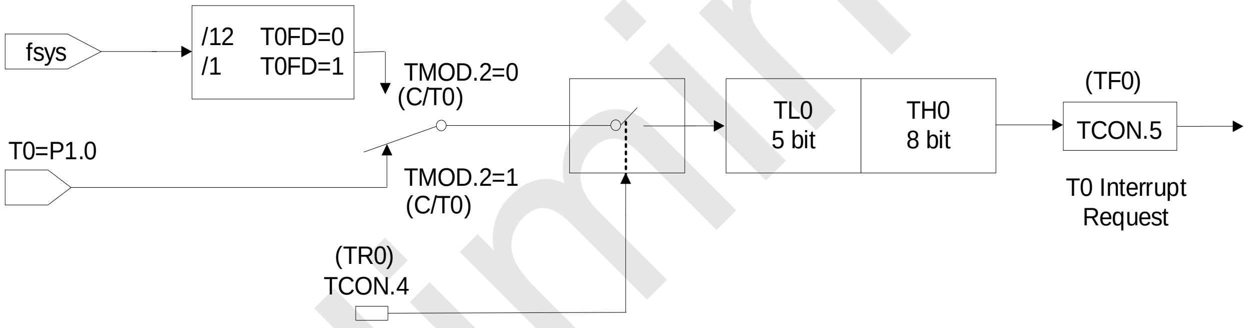
Operating mode 0: 13-bit Timer/Counter

TH0 register is to store the high 8 bits (TH0.7 ~ TH0.0) of 13-bit Timer/Counter 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 values, they shall be omitted upon reading. When 13-bit Timer/Counter overflows with count increment, the system will set timer overflow flag TF0 to 1. An interrupt will be generated if the timer0 interrupt is enabled.

C/T0 bit selects the clock input source of Timer/Counter. If C/T0=1, the level fluctuation from high to low of Counter 0 input pin T0 (P1.0) will make Counter 0 data register add 1. If C/T0=0, the frequency division of system clock is the clock source of Timer0.

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.

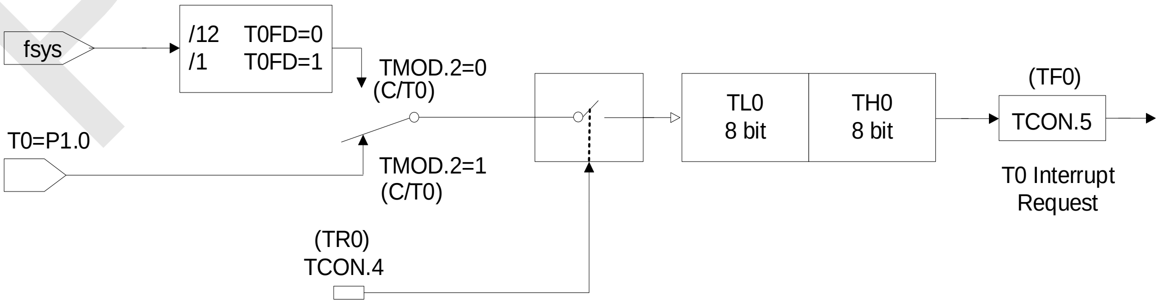
When configured as a timer, the SFR T0FD is used to select fractional frequency ratio of clock source.



Operating mode 0: 13-bit Timer/Counter

Operating Mode 1: 16 Counter/Timer

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



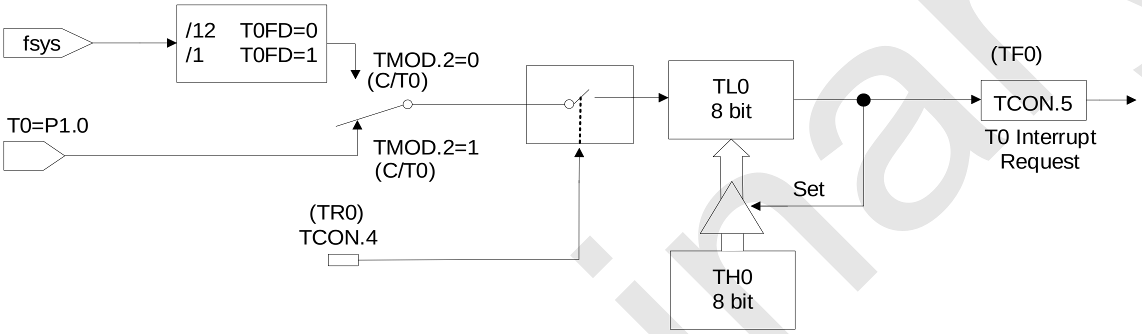
Operating mode 1: 16-bit Timer/Counter

Operating Mode 2: 8 Automatic Reload Counter/Timer

In operating mode 2, Timer0 is 8-bit automatic reload Timer/Counter. 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 to 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 Timer/Counter in operating mode 2 shall be the same as that in mode 0 and mode 1.

When configured as a timer, the SFR TMCON bit 0 (T0FD) is used 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 Timer0)

In operating mode 3, Timer0 is used as two independent 8-bit Timer/Counters, respectively controlled by TL0 and TH0. TL0 is controlled by control bit (in TCON) and status bit (in TMOD) of Timer0 (TR0), C/T0, TF0. Timer0 is selected as Timer or Counter by TMOD bit 2 (C/T0).

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 to 1. When overflow occurs and interrupt is discovered, set TF1 to 1 and proceed the interrupt as T1 interrupt.

When T0 is configured 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 Modes

Timer1 can be configured in one of three operating modes by setting the bit pairs (M11, M01) in the TMOD register.

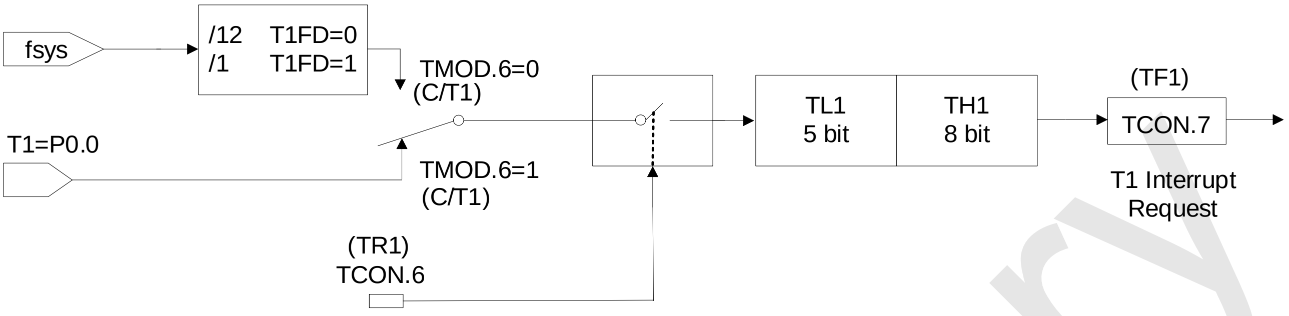
Operating mode 0: 13-bit Timer/Counter

TH1 register is to store high 8-bit (TH1.7 ~ TH1.0) of 13-bit Timer/Counter and TL1 is to store low 5-bit (TL1.4 ~ TL1.0). The high 3-bit 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. An interrupt will be generated if the timer1 interrupt is enabled. C/T1 bit selects the clock input source of Timer/Counter.

If C/T1=1, the level fluctuation from high to low of timer1 input pin T1 (P0.0) will make timer1 data register add 1. If C/T1=0, the frequency division of system clock is the clock source of timer1.

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.

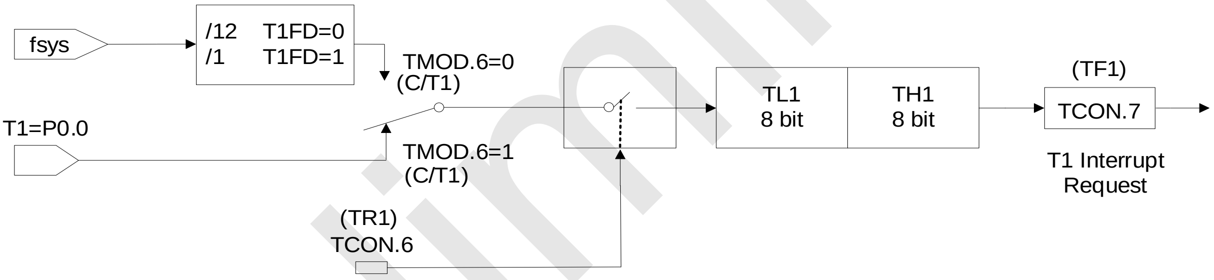
When configured as timer, the SFR T1FD is used to select fractional frequency ratio of clock source.



Operating mode 0: 13-bit Timer/Counter

Operating Mode 1: 16 Counter/Timer

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



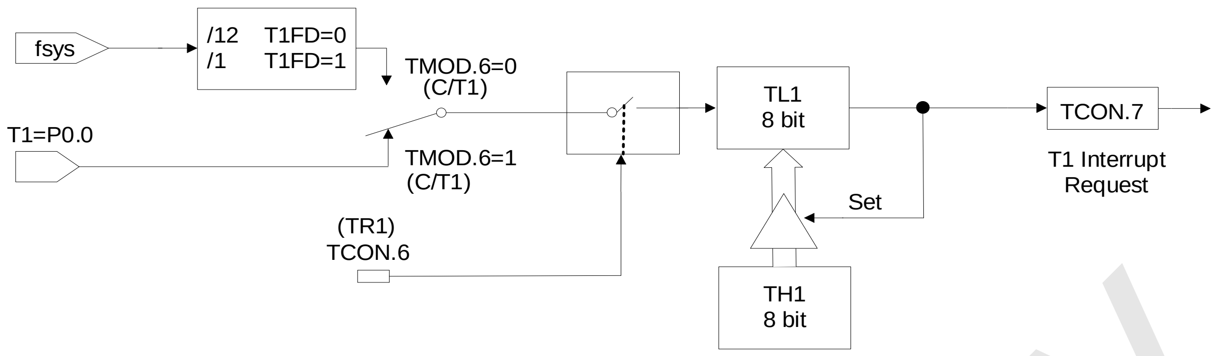
Operating mode 0: 16-bit Timer/Counter

Operating Mode 2: 8 Automatic Reload Counter/Timer

In operating mode 2, Timer1 is 8-bit automatic reload Timer/Counter. TL1 is 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 interrupt, setting TF1 to 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 Timer/Counter in operating mode 2 shall be the same as that of mode 0 and mode 1.

When configured as timer, the SFR TMCON bit 4 (T1FD) is used 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 Timer/Counter T2/3/4

The Timer2/3/4 inside the SC92F84HX microcontroller are three independent timers, with Timer2 having 4 operating modes, Timer3 and Timer4 having 3 operating modes.

The control registers for Timer2/3/4 share the same set of addresses (C8H-CDH). Users can direct the TimerX register group (TXCON / TXMOD / RCAPXL / RCAPXH / TLX / THX) to Timer2/3/4 through TXINX[2:0], thereby achieving the function of configuring three independent timers with a single set of registers.

Note: Only after the TXINX[2: 0] configuration is successful, the Timer X register group will point to the Timer 2/3/4 specified by the user. At this time, operating the TimeX register group is an effective operation for the corresponding Timer.

11.1 T2-related Registers

Mnemonic	Add	Description	7	6	5	4	3	2	1	0	POR
TXINX	CEH	Timer 2/3/4 control register pointer	-	-	-	-	-	TXINX[2:0]			xxxxx010b
TXCON	C8H	Timer2/3/4 Control Register	TFX	EXFX	RCLKX	TCLKX	EXENX	TRX	C/TX	CP/RLX	00000000b
TXMOD	C9H	Timer2/3/4 Operating Mode Register	TXFD	-	EPWMN1	EPWMN0	INVN1	INVN0	TXOE	DCXEN	0x000000b
RCAPXL	CAH	Timer2/3/4 Reload/Capture Low Byte	RCAPXL[7: 0]								00000000b
RCAPXH	CBH	Timer2/3/4 Reload/Capture High Byte	RCAPXH[7: 0]								00000000b
TLX	CCH	Timer2/3/4 Low Byte	TLX[7: 0]								00000000b
THX	CDH	Timer2/3/4 High Byte	THX[7: 0]								00000000b
TMCON	8EH	Timer Frequency Control Register	-	-	-	T3OS	-	T2OS	T1FD	T0FD	xxx0x000b

Register instructions are shown below:

TXINX (CEH) Timer 2/3/4 Control Register Pointer (read/write)

Bit number	7	6	5	4	3	2	1	0

Bit Mnemonic	-	-	-	-	-	TXINX[2: 0]		
R/W	-	-	-	-	-	R/W	R/W	R/W
POR	x	x	x	x	x	0	1	0

Bit number	Bit Mnemonic	Description
2~0	**TXINX[2: 0]**	Timer 2/3/4 control register pointer 010: Timer X register set: TXCON / TXMOD / RCAPXL / RCAPXH / TLX / THX points to T2 011: Timer X register set points to T3 100: Timer X register set points to T4 Other: reserved
7~3	**-**	Reserved

TMCON (8EH) Timer Frequency Control Register (R/W)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	-	-	-	T3OS	-	T2OS	T1FD	T0FD
R/W	-	-	-	R/W	-	R/W	R/W	R/W
POR	x	x	x	0	x	0	0	0

<table>
<tr><th>位编号</th><th>位符号</th><th>说明</th></tr>
<tr><td>4</td><td><b>T3OS</b></td><td><table>
<tr><td>Signal</td><td>GPIO-default singal port T3OS=0</td><td>GPIO-A group mapping T3OS=1</td></tr>
<tr><td>T3</td><td>P1.1</td><td>P2.4</td></tr>
<tr><td>T3EX</td><td>P1.2</td><td>P2.5</td></tr>
</table></td></tr>
<tr><td>2</td><td><b>T2OS</b></td><td><table>
<tr><td>Signal</td><td>GPIO-default signal port T2OS=0</td><td>GPIO-A group mapping T2OS=1</td></tr>
<tr><td>T2</td><td>P1.5</td><td>P2.2</td></tr>
</table></td></tr>
</table>

		T2EX	P1.4	P2.3
1	**T1FD**	T1 input frequency selection control 0: T1 frequency is derived from fsys/12 1: T1 frequency is derived from fsys		
0	**T0FD**	T0 input frequency selection control 0: T0 frequency is derived from fsys/12 1: T0 frequency is derived from fsys		

11.2 Timer 2

Timer 2 inside the SC92F84HX MCU has two operating modes: counting mode and timing mode. There is a control bit C/TX in the special function register TXCON to select whether T2 is a timer or a counter. They are essentially an addition counter, but the source of the count is different. The source of the timer is the system clock or its divided clock, but the source of the counter is the input pulse of the external pin. TRX is the switch control of T2/T3/T4 counting in the timer/counter mode. Only when TRX=1, T2 will be opened for counting.

In counter mode, for every pulse on the T2 pin, the count value of T2 increases by 1 respectively.

In timer mode, the count source of T2 can be selected as fsys/12 or fsys through the special function register TXMOD.7 (TXFD).

Timer/counter T2 has 4 operating modes:

① Mode 0: 16-bit capture mode

② Mode 1: 16-bit auto-reload timer mode

③ Mode 2: Baud rate generator mode

④ Mode 3: Programmable clock output mod

TXINX[2: 0] = 010, the Timer X register group points to Timer 2, the explanation of each register is as follows:

TXCON (C8H) Timer 2 Control Register (read/write) (TXINX[2: 0] = 010)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	TFX	EXFX	RCLKX	TCLKX	EXENX	TRX	C/TX	CP/RLX
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 Mnemonic	Description
7	**TFX**	Timer 2 overflow flag 0: No overflow (must be cleared by software) 1: Overflow (if RCLK = 0 and TCLK = 0, set by hardware 1)
6	**EXFX**	Flag bit detected by external event input (falling edge) of T2EX pin 0: No external event input (must be cleared by software) 1: External input detected (if EXENX = 1, set by hardware)
5	**RCLKX**	UART0 receive clock control bit 0: Timer 1 generates the receive baud rate 1: Timer 2 generates the receive baud rate
4	**TCLKX**	UART0 transmit clock control bit 0: Timer 1 generates transmission baud rate 1: Timer 2 generates transmission baud rate
3	**EXENX**	T2EX pin is used as a reload/capture trigger enable/disable control: 0: Ignore events on T2EX pin 1: When Timer 2 is not used as the UART0 clock, a falling edge on the T2EX pin is detected, and a capture or reload is generated
2	**TRX**	Timer 2 start/stop control bit 0: stop timer 2/stop PWM2 counter 1: Start timer 2/start PWM2 counter
1	**C/TX**	Timer 2 Timer/counter mode selection positioning 2 0: Timer mode, T2 pin is used as I/O port 1: Counter mode
0	**CP/RLX**	Capture/reload mode selection positioning 0: 16-bit timer/counter with reload function 1: 16-bit timer/counter with capture function, TXEX is timer 2 external

		capture signal input port

TXMOD (C9H) Timer 2 Operating Mode Register (read/write) (TXINX[2: 0] = 010)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	TXFD	-	EPWM21	EPWM20	INV21	INV20	TXOE	DCXEN
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 Mnemonic	Description
7	**TXFD**	T2 input frequency selection control 0: T2 frequency is derived from fsys/12 1: T2 frequency is derived from fsys
1	**TXOE**	Timer 2 output enable bit 0: Set T2 as clock input or I/O port 1: Set T2 as the clock output
0	**DCXEN**	Count down enable bit 0: Timer 2 is prohibited as an up/down counter, Timer 2 is only used as an up counter 1: Allow Timer 2 as an up/down counter
6	**-**	Reserved

IE (A8H) Interrupt Enable Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Bit Mnemonic	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 Mnemonic	Description
5	**ET2**	Timer 2 interrupt enable control 0: Disable Timer 2 interrupt 1: Enable Timer 2 interrupt

IP (B8H) Interrupt Priority Control Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	-	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 Mnemonic	Description
5	**IPT2**	Timer 2 interrupt priority 0: Set the interrupt priority of Timer 2 to "Low" 1: Set the interrupt priority of Timer 2 to "High"

11.3 Timer 3

Timer 3 inside the SC92F84HX MCU as a timer is essentially an addition counter. The clock source of the timer is the system clock or its divided clock. TRX is the switch control of T3 counting. Only when TRX=1, T3 will be opened to count.

In timer mode, the count source of T3 can be selected as fsys/12 or fsys through the special function register TXMOD.7 (TXFD).

TXINX[2: 0] = 011, the Timer X register group points to Timer 3, the explanation of each register is as follows:

TXCON (C8H) Timer 3 Control Register (read/write) (TXINX[2: 0] = 011)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	TFX	EXFX	-	-	EXENX	TRX	C/TX	CP/RLX
R/W	R/W	R/W	-	-	R/W	R/W	R/W	R/W
POR	0	0	x	x	0	0	0	0

Bit number	Bit Mnemonic	Description
7	**TFX**	Timer 3 control register (read/write) Timer 3 overflow flag 0: No overflow (must be cleared by software) 1: Overflow (set by hardware 1)
6	**EXFX**	Flag bit detected by external event input (falling edge) of T3EX pin 0: No external event input (must be cleared by software) 1: External input detected (if EXENX = 1, set by hardware)
3	**EXENX**	T3EX pin is used as a reload/capture trigger enable/disable control: 0: Ignore events on T3EX pin 1: A falling edge on the T3EX pin is detected, and a capture or reload is generated
2	**TRX**	Timer 3 start/stop control bit 0: stop timer 3/stop PWM3 counter 1: Start timer 3/start PWM3 counter

1	**C/TX**	Timer 3 Timer/counter mode selection positioning 2 0: Timer mode, T3 pin is used as I/O port 1: Counter mode
0	**CP/RLX**	Capture/reload mode selection positioning 0: 16-bit timer/counter with reload function 1: 16-bit timer/counter with capture function, TXEX is timer 3 external capture signal input port
5~4	**-**	Reserved

TXMOD (C9H) Timer 3 Operating Mode Register (read/write) (TXINX[2: 0] = 011)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	TXFD	-	EPWM31	EPWM30	INV31	INV30	TXOE	DCXEN
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 Mnemonic	Description
7	**TXFD**	T3 input frequency selection control 0: T3 frequency is derived from fsys/12 1: T3 frequency is derived from fsys
1	**TXOE**	Timer 3 output enable bit 0: Set T3 as clock input or I/O port 1: Set T3 as the clock output
0	**DCXEN**	Count down enable bit

		0: Timer 3 is prohibited as an up/down counter, Timer 3 is only used as an up counter 1: Allow Timer 3 as an up/down counter, T3EX is used to select the counting direction
6	**-**	Reserved

IE1 (A9H) Interrupt Enable Register 1 (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	ET4	ET3	-	ETK	-	EBTM	-	EUSCI0
R/W	R/W	R/W	-	R/W	-	R/W	-	R/W
POR	0	0	X	0	X	0	X	0

Bit number	Bit Mnemonic	Description
6	**ET3**	Timer 3 interrupt enable control 0: Disable Timer 3 interrupt 1: Enable Timer 3 interrupt
5,3,1	**-**	Reserved

IP1 (B9H) Interrupt Priority Control Register 1 (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	IPT4	IPT3	-	IPTK	-	IPBTM	-	IPUSCI0
R/W	R/W	R/W	-	R/W	-	R/W	-	R/W

POR	0	0	X	0	X	0	X	0

Bit number	Bit Mnemonic	Description
6	**IPT3**	Timer 3 interrupt priority selection 0: Timer 3 interrupt priority is low 1: Timer 3 interrupt priority is high
5,3,1	**-**	Reserved

11.4 Timer 4

Timer 4 inside the SC9284HX MCU as a timer is essentially an addition counter. The clock source of the timer is the system clock or its divided clock. TRX is the switch control of T4 count. Only when TRX=1, T4 will be turned on and counted.

In timer mode, the count source of T4 can be selected as fsys/12 or fsys through the special function register TXMOD.7 (TXFD).

TXINX[2: 0] = 100, Timer X register group points to Timer 4, the explanation of each register is as follows:

TXCON (C8H) Timer 4 Control Register (read/write) (TXINX[2: 0] = 100)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	TFX	EXFX	-	-	EXENX	TRX	C/TX	CP/RLX
R/W	R/W	R/W	-	-	R/W	R/W	R/W	R/W
POR	0	0	x	x	0	0	0	0

Bit number	Bit Mnemonic	Description
7	**TFX**	Timer 4 overflow flag 0: No overflow (must be cleared by software)

		1: Overflow (set by hardware 1)
6	**EXFX**	Flag bit detected by external event input (falling edge) of T4EX pin 0: No external event input (must be cleared by software) 1: External input detected (if EXENX = 1, set by hardware)
3	**EXENX**	T4EX pin is used as a reload/capture trigger enable/disable control: 0: Ignore events on T4EX pin 1: A falling edge on the T4EX pin is detected, and a capture or reload is generated
2	**TRX**	Timer 4 start/stop control bit 0: stop timer 4/stop PWM4 counter 1: Start timer 4/start PWM4 counter
1	**C/TX**	Timer 4 Timer/counter mode selection positioning 2 0: Timer mode, T4 pin is used as I/O port 1: Counter mode
0	**CP/RLX**	Capture/reload mode selection positioning 0: 16-bit timer/counter with reload function 1: 16-bit timer/counter with capture function, TXEX is timer 4 external capture signal input port
5~4	**-**	Reserved

TXMOD (C9H) Timer 4 Operating Mode Register (read/write) (TXINX[2: 0] = 100)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	TXFD	-	EPWM41	EPWM40	INV41	INV40	TXOE	DCXEN
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 Mnemonic	Description
7	**TXFD**	T4 input frequency selection control 0: T4 frequency is derived from fsys/12 1: T4 frequency is derived from fsys
1	**TXOE**	Timer 4 output enable bit 0: Set T4 as clock input or I/O port 1: Set T4 as the clock output
0	**DCXEN**	Count down enable bit 0: Timer 4 is prohibited as an up/down counter, Timer 4 is only used as an up counter 1: Allow Timer 4 as an up/down counter, T4EX is used to select the counting direction
6	**-**	Reserved

IE1 (A9H) Interrupt Enable Register 1 (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	ET4	ET3	-	ETK	-	EBTM	-	EUSCI0
R/W	R/W	R/W	-	R/W	-	R/W	-	R/W
POR	0	0	x	0	x	0	x	0

Bit number	Bit Mnemonic	Description

7	**ET4**	Timer 4 interrupt enable control 0: Disable Timer 4 interrupt 1: Enable Timer 4 interrupt
5,3,1	**-**	Reserved

IP1 (B9H) Interrupt Priority Control Register 1 (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	IPT4	IPT3	-	IPTK	-	IPBTM	-	IPUSCI0
R/W	R/W	R/W	-	R/W	-	R/W	-	R/W
POR	0	0	x	0	x	0	x	0

Bit number	Bit Mnemonic	Description
6	**IPT4**	Timer 4 interrupt priority selection 0: Timer 4 interrupt priority is low 1: Timer 4 interrupt priority is high
5,3,1	**-**	Reserved

11.5 Timer 2/3/4 Operating Modes

Timer Timer2/3/4's operating modes as follows:

① Mode 0: 16-bit capture

② Mode 1: 16-bit auto-reload timer

③ Mode 2: Baud rate generator, only Timer 2 support this mode

④ Mode 3: Programmable clock output

⑤ Mode 4: PWM output mode

The preceding working modes and configuration modes are listed as follows:

C/TX	TXOE	DCXEN	TRX	CP/RLX	EXENX	Mode	
X	0	X	1	1	1	Mode 0	16-bit capture
X	0	0	1	0	0	Mode 1	16-bit auto-reload timer/counter, normally auto-reload
X	0	0	1	0	1		16-bit auto-reload timer/counter, with TnEX trigger reload
X	0	1	1	0	X		16-bit auto-reload timer/counter, increase or decrease reload
X	0	X	1	X	X	Mode 2	UART0 Baud rate generator, only for Timer 2
0	1	X	1	X	X	Mode 3	Programmable clock output
X	X	X	0	X	1	X	Timer stops, TnEX(n=2~4) channel is still allowed

11.5.1 Timer 2/3/4 Operating Modes

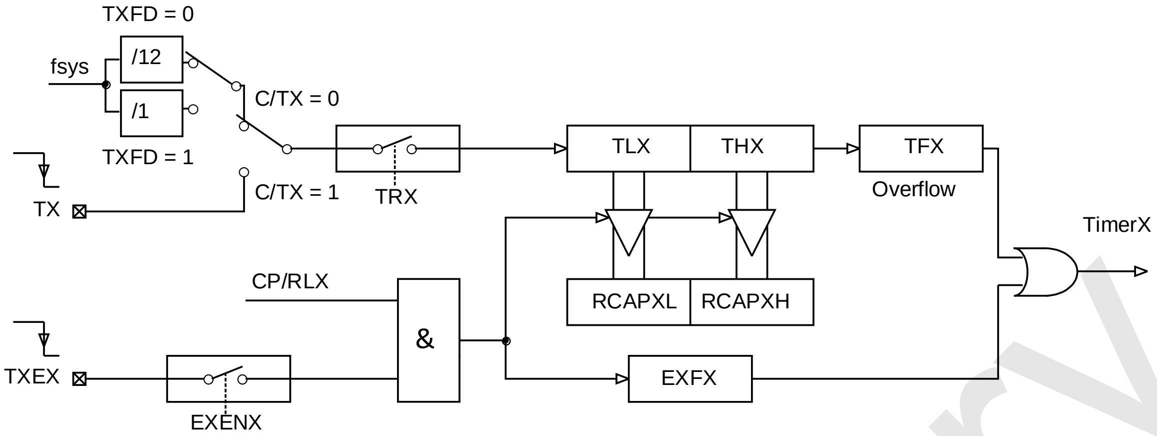
Operating Mode 0: 16-bit Capture

CP/RLX = 1, set Timer n(n=2~4) to 16-bit capture

In the capture mode, the EXENX bit of TXCON has two options:

If EXENX = 0, Timer n acts as a 16-bit timer or counter. If ETn is enabled, Timer n can set TFX overflow to generate an interrupt.

If EXENX = 1, Timer n performs the same operation, but the falling edge on external input TnEX can also cause the current values in THX and TLX to be captured in RCAPXH and RCAPXL, respectively. In addition, the falling edge on TnEX also Can cause EXFX in TXCON to be set. If ETn is enabled, the EXFX bit also generates an interrupt like TFX.



Mode 0: 16-bit capture

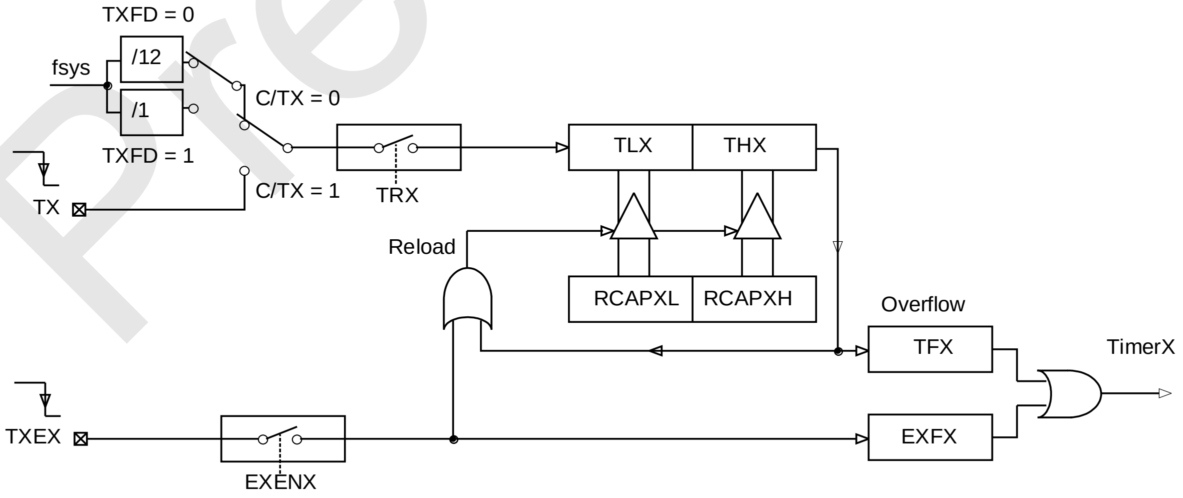
Operating Mode 1: 16-bit Auto-Reload Timer

In 16-bit auto-reload mode, Timer n(n=2~4) can be selected to count up or count down. This function is selected by the DCEN bit (down counting allowed) in TnMOD. After the system is reset, the reset value of the DCEN bit is 0, and the timer n counts up by default. When DCEN is set to 1, Timer n counts up or down depending on the level on the TnEX pin.

When DCEN = 0, two options are selected through the EXENX bit in TXCON.

If EXENX = 0, Timer n increments to 0xFFFFH, sets the TFX bit after overflow, and the timer automatically loads the 16-bit values of registers RCAPXH and RCAPXL written in user software into the THX and TLX registers.

If EXENX = 1, an overflow or a falling edge on TnEX can trigger a 16-bit reload and set the EXFX bit. If ETn is enabled, both TFX and EXFX bits can generate an interrupt.



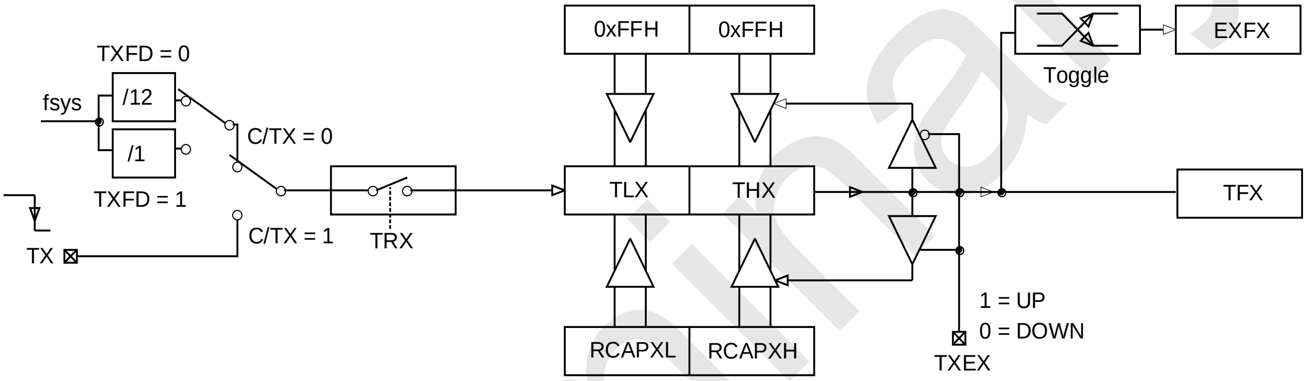
Mode 1: 16-bit auto-reload DCEN = 0

Setting the DCEN bit allows Timer n to count up or down. When DCEN = 1, the TnEX pin controls the direction of the count, and EXENX control is invalid.

Setting TnEX to 1 causes Timer n to count up. The timer overflows to 0xFFFFH, and then sets the TFX bit. Overflow can also cause the 16-bit values on RCAPXH and RCAPXL to be reloaded into the timer register, respectively.

Setting TnEX to 0 causes Timer n to count down. When the values of THX and TLX are equal to the values of RCAPXH and RCAPXL, the timer overflows. The TFX bit is set and 0xFFFFH is reloaded into the timer register.

Regardless of whether Timer n overflows or not, the EXFX bit is used as the 17th bit of the result. In this operating mode, EXFX is not used as an interrupt flag.



Mode 1: 16-bit auto-reload DCEN = 1

Operating Mode 2: Baud Rate Generator,only for Timer 2

Set TCLK and/or RCLK in the TXCON register to select Timer 2 as the baud rate generator. The baud rate of the receiver and transmitter can be different. If Timer 2 acts as a receiver or transmitter, then Timer 1 acts as another baud rate generator

Set TCLK and/or RCLK in the TXCON register to make Timer 2 enter the baud rate generator mode, which is similar to the automatic reload mode

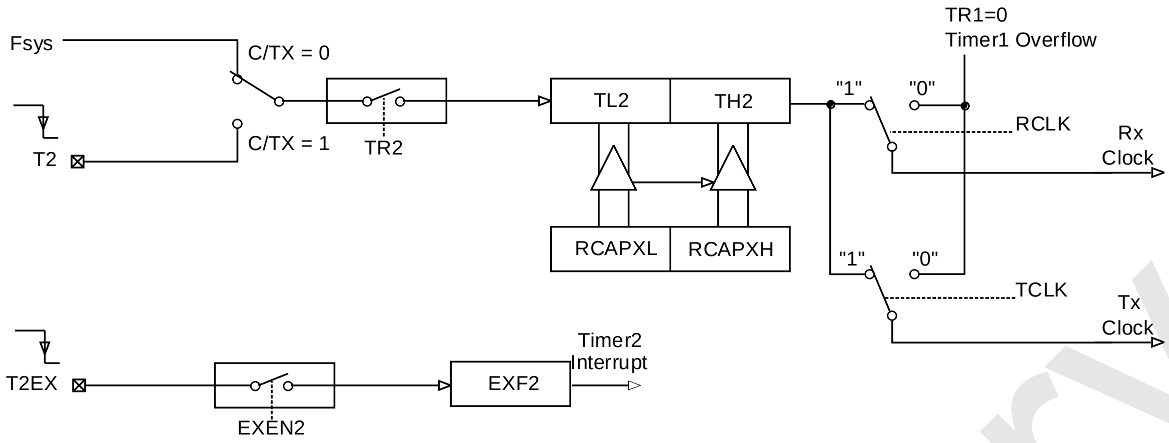
The overflow of Timer 2 will reload the values in the RCAPXH and RCAPXL registers to the Timer 2 count, but no interrupt will be generated

If EXENX is set to 1, the falling edge on the T2EX pin will set up EXFX, but it will not cause a heavy load. So when Timer 2 is used as a baud rate transmitter, T2EX can be used as an additional external interrupt

The baud rate in UART0 mode 1 and 3 is determined by the overflow rate of timer 2 according to the following equation:

$$\mathrm{BaudRate} = \frac{\mathrm{fsys}}{[\mathrm{RCAPXH,RCAPXL}]}$$; (Note: [RCAPXH, RCAPXL] must be bigger than 0x0010**)**

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



Mode 2: Baud rate generator

Operating Mode 3: Programmable Clock Output

In this way, Timer n(n=2~4) can be programmed to output a 50% duty cycle clock cycle: when $C/\overline{Tn}$ = 0; TnOE = 1, timer n is enabled as a clock generator

In this way, Tn outputs a clock with a 50% duty cycle

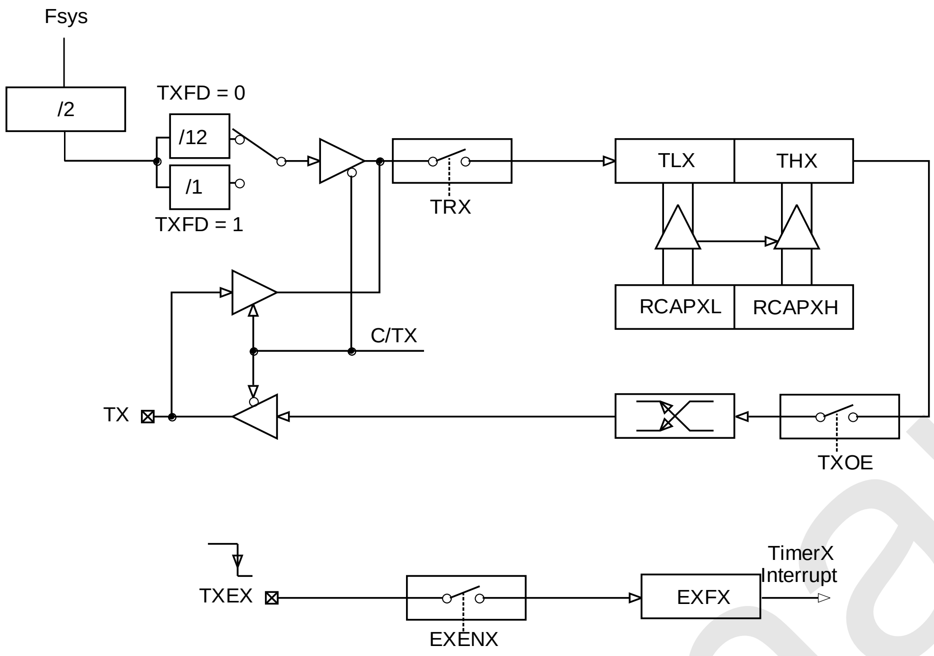
$$\text{Colck Out Frequency} = \frac{\text{fn}}{(65536-[\text{RCAPXH,RCAPXL}])\times 4};$$

Among them, fn is the timer n clock frequency:

$$\text{fn} = \frac{\text{fsys}}{12}; \quad \text{TXFD} = 0$$

$$\text{fn} = \text{fsys}; \quad \text{TXFD} = 1$$

Timer n overflow does not generate an interrupt, and the Tn port is used as a clock output.



Mode 3: Programmable clock output

Note:

1. Both TFX and EXFX can cause the interrupt request of Timer n(n=2~4), both have the same vector address;

2. When the event occurs or at any other time, TFX and EXFX can be set to 1 by software, and only software and hardware reset can clear it to 0;

3. When EA = 1 and ETn = 1, setting TFX or EXFX to 1 can cause Timer n to interrupt;

4. When Timer 2 is used as a baud rate generator, writing THX/TLX or RCAPXH/RCAPXL will affect the accuracy of the baud rate and cause communication errors.

12 PWM2/3/4

SC92F84HX provides 6 channels divided into 3 groups: PWM2(T2PWM0/T2PWM1), PWM3(T3PWM0/T3PWM1), PWM4(T4PWM0/T4PWM1).

Note: These three sets of PWM period registers are shared with the TLX and THX of Timer2, Timer3, and Timer4 respectively. Therefore, once users use the PWM2, PWM3, and PWM4 resources, they cannot change the timing/count value of Timer2, Timer3, and Timer4. Otherwise it will lead to abnormal OUTPUT of PWM cycle!

12.1 PWM2/3/4 related Registers

TXINX (CEH) Timer 2/3/4 Control Register Pointer (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	-	-	-	-	-	TXINX[2: 0]		
R/W	-	-	-	-	-	R/W	R/W	R/W
POR	x	x	x	x	x	0	1	0

Bit number	Bit Mnemonic	Description
2~0	**TXINX[2: 0]**	Timer 2/3/4 control register pointer 010: Timer X register set: TXCON / TXMOD / RCAPXL / RCAPXH / TLX / THX points to PWM2 011: Timer X register set points to PWM3 100: Timer X register set points to PWM4 Other: reserved
7~3	**-**	Reserved

TXCON (C8H) Timer n Control Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	TFX	EXFX	RCLKX	TCLKX	EXENX	TRX	C/TX	CP/RLX

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 Mnemonic	Description
2	**TRX**	Timer n start/stop control bit 0: stop timer n/stop PWMn counter 1: Start timer n/start PWMn counter

When EPWMn0 or EPWMn1 is set to 1, the Timer can start PWM mode, Tn and TnEX (n= 2~4) are invalid, and PWMxy (x= 2~4, y=0~1) can output PWM waveform.

TXMOD (C9H) Timer n(n=2,3,4) Operating Mode Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	TXFD	-	EPWMn1	EPWMn0	INVn1	INVn0	TXOE	DCXEN
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 Mnemonic	Description
7	**TXFD**	Tn(n=2,3,4) input frequency selection control 0: Tn frequency is derived from fsys/12 1: Tn frequency is derived from fsys
5	**ENPWMn1** **n=2,3,4**	PWMn1 Waveform output select 0: PWMn1 output is disabled 1: I/O where PWMn1 resides serves as the output port of the PWM waveform

4	**ENPWMn0** **n=2,3,4**	PWMn0 Waveform output select 0: PWMn0 output is disabled 1: I/O where PWMn0 resides serves as the output port of the PWM waveform
3	**INVn1** **n=2,3,4**	PWMn1 waveform output reverse control 1: PWMn1 waveform output is reversed 0: PWMn1 waveform output is not reversed
2	**INVn0** **n=2,3,4**	PWMn0 waveform output reverse control 1: PWMn0 waveform output is reversed 0: PWMn0 waveform output is not reversed

The THX and TLX counter starts counting from 0, and when the count value matches the value of the duty cycle setting item PDTxy [15: 0], the PWM output waveform switches between high and low levels, The THX and TLX counter then continues counting upward reload value PWMPDX, then THX and TLX counter is cleared and generate count overflow events and a PWM cycle ends. If the timer interrupt is enabled, a timer interrupt will be generated at this time.

The calculation formula of PWM period **T_{PWM}** output by Timer is as follows:

$$\mathrm{Tpwm} = \frac{\mathrm{PWMPDX}[15:0] + 1}{\mathrm{fsys}}$$

Duty calculation formula:

$$\mathrm{duty} = \frac{\mathrm{PDTxy}\ [15:0]}{\mathrm{PWMPDX}[15:0] + 1}$$

The PWM cycle is set through the following registers:

RCAPXH (CBH)

PWMn period register high 8 bits(R/W)

Note: The PWM2/3/4 cycle register is multiplexed with Timer2, Timer3, and Timer4. Therefore, once users use the PWM2, PWM3, and PWM4 resources, they cannot change the timer/counter value of Timer2, Timer3, and Timer4. Otherwise, the PWM cycle output will be abnormal!

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	PWMPDHX[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

RCAPXL (CAH)

PWMn period register low 8 bits (R/W) (TXINX[2:0] = 010)

Note: The PWM2/3/4 cycle register is multiplexed with Timer2, Timer3, and Timer4. Therefore, once users use the PWM2, PWM3, and PWM4 resources, they cannot change the timer/counter value of Timer2, Timer3, and Timer4. Otherwise, the PWM cycle output will be abnormal!

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	PWMPDLX[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 Mnemonic	Description
7~0	**PWMPDX[15:0]**	PWMn cycle set This value represents the output waveform of PWMn (cycle-1); that is to say PWMn cycle of output is(PWMPDX[15:0] + 1)* PW clock

The PWM duty is set through the following registers:

PWM2~4 Duty cycle adjustment register (R/W)

ADD	7	6	5	4	3	2	1	0	POR
2034H	PDT20[15:8]								00000000b
2035H	PDT20[7:0]								00000000b
2036H	PDT21[15:8]								00000000b
2037H	PDT21[7:0]								00000000b
2038H	PDT30[15:8]								00000000b
2039H	PDT30[7:0]								00000000b
203AH	PDT31[15:8]								00000000b
203BH	PDT31[7:0]								00000000b
203CH	PDT40[15:8]								00000000b
203DH	PDT40[7:0]								00000000b
203EH	PDT41[15:8]								00000000b
203FH	PDT41[7:0]								00000000b

Bit number	Bit Mnemonic	Description
7~0	**PDTxy[15:0]**	PWMxy waveform duty cycle length setting

	（x=2~4，y=0~1）	The high level width of the PWMxy waveform is： (PDTxy[15:0] + 1) PWM clock

12.2 PWM2/3/4 Duty Variation Characteristics

The duty can be changed by changing the high level setting register PDTxy (x=2~4, y=0~1) when PWM2/3/4 output waveform. However, it should be noted that if you change the PDTxy value, the duty will not change immediately, but wait until the end of this cycle and change in the next cycle.

12.3 PWM2/3/4 Cycle Variation Characteristics

If you need to change the period when the PWM2/3/4 outputs the waveform, you can set the TLX and THX values of the register groups by changing the period. Change the value of the cycle register, the PWM output cycle changes as follows:

Define the current period meter value as Tn, when writing the period register, the value recorded by the timer is Tm, and the period meter value to be updated is Tx, then:

Tm ≤ Tx: the period changes in real time according to Tx;

Tm > Tx: At this point, the cycle change will be divided into two stages. In the first stage, after writing to the cycle register, the cycle counter accumulates from the current count until overflow is cleared. In the second phase, the period changes with respect to Tx.

13 Multiplier-Divider Unit (MDU)

The SC92F84HX provides a 16-bit multiplier-divider, which is composed of extended accumulator EXA0 ~ EXA3, extended B register EXB and operation control register OPERCON. It can replace the software 16-bit*16-bit multiply operation and 32-bit /16-bit division operation.

Mnemonic	Add	Description	7	6	5	4	3	2	1	0	POR
EXA0	E9H	Extended Accumulator 0	EXA [7: 0]								00000000b
EXA1	EAH	Extended Accumulator 1	EXA [15: 8]								00000000b
EXA2	EBH	Extended Accumulator 2	EXA [23: 16]								00000000b
EXA3	ECH	Extended Accumulator 3	EXA [31: 24]								00000000b
EXBL	EDH	Extended B Register L	EXB [7: 0]								00000000b
EXBH	EEH	Extended B Register H	EXB [15: 8]								00000000b

OPERCON (EFH) Arithmetic Control Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Mnemonic	OPERS	MD	-	-	-	-	-	CHKSUMS
R/W	R/W	R/W	-	-	-	-	-	R/W
POR	0	0	x	x	X	x	x	0

Bit Number	Bit Mnemonic	Description
7	**OPERS**	Multiplier and divider operation trigger control bit (Operator Start) Set to start a new multiply-divide operation, this bit is only the trigger signal calculated with multiplier, when this bit is zero, the calculation is

<table>
<tr><td></td><td></td><td>completed. This bit is only valid for writing 1.</td></tr>
<tr><td>6</td><td><b>MD</b></td><td>Multiplier and divider selection bit
0: Multiply operation, writing of multiplicand and multiplier and reading of product are shown below:
<table>
<tr><th>Byte
Operations</th><th>Byte 3</th><th>Byte 2</th><th>Byte 1</th><th>Byte 0</th></tr>
<tr><td>multiplicand 16bits</td><td>-</td><td>-</td><td>EXA1</td><td>EXA0</td></tr>
<tr><td>multiplier 16bits</td><td>-</td><td>-</td><td>EXBH</td><td>EXBL</td></tr>
<tr><td>product 32bits</td><td>EXA3</td><td>EXA2</td><td>EXA1</td><td>EXA0</td></tr>
</table>
1: Division operation: writing of dividend and divisor and reading of quotient and remainder are shown below:
<table>
<tr><th>Byte
Operations</th><th>Byte 3</th><th>Byte 2</th><th>Byte 1</th><th>Byte 0</th></tr>
<tr><td>dividend 32bits</td><td>EXA3</td><td>EXA2</td><td>EXA1</td><td>EXA0</td></tr>
<tr><td>divisor 16bits</td><td>-</td><td>-</td><td>EXBH</td><td>EXBL</td></tr>
<tr><td>quotient 32bits</td><td>EXA3</td><td>EXA2</td><td>EXA1</td><td>EXA0</td></tr>
<tr><td>remainder 16bits</td><td>-</td><td>-</td><td>EXBH</td><td>EXBL</td></tr>
</table>
</td></tr>
</table>

Note:

1. **During the operation process, it is forbidden to read or write EXA and EXB data registers.**
2. **The time for operation conversion of multiplier is 16/f$_{SYS}$.**

14 General-purpose I/O (GPIO)

The SC92F84HX offers up to 26 bidirectional 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-up resistor controlled by PxPHy. Such 26 IOs are shared with other functions. Under input or output state, what I/O port read from the value of port data register is the actual state value of the port.

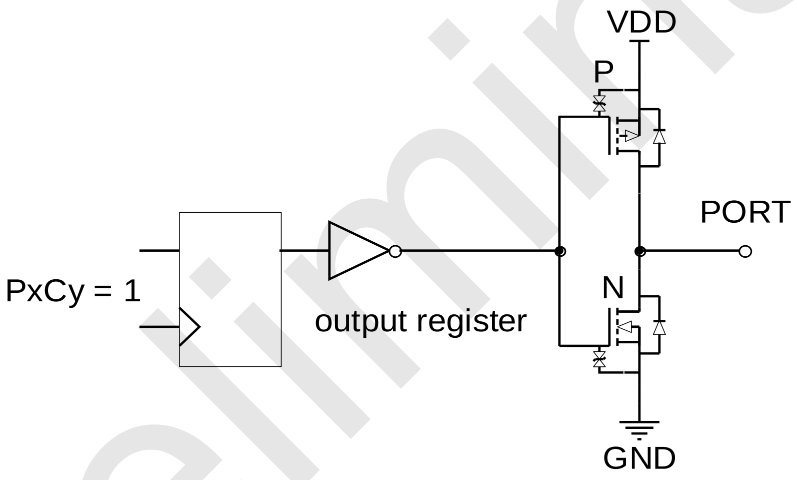
Note: Unused IO port or IO port with no package pin shall be configured as strong push-pull output mode.

14.1 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 100mA

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

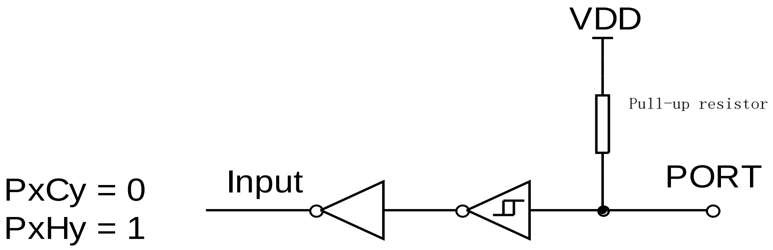


Strong Push-pull Output Mode

Pull-up Input Mode

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

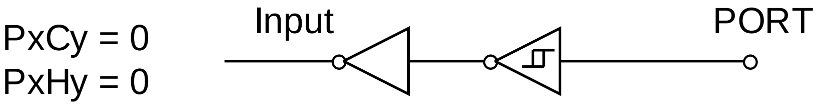
The port structure diagram for pull-up input mode is shown below:



Pull-up Input Mode

High Impedance Input Mode. (Input only)

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



High- impedance Input Mode

14.2 I/O Port-related Registers

P0CON (9AH) P0 I/O Control Register (Read/Write)

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	P0C7	P0C6	P0C5	P0C4	P0C3	P0C2	P0C1	P0C0
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

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

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	P0H7	P0H6	P0H5	P0H4	P0H3	P0H2	P0H1	P0H0
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

P1CON (91H) P1 I/O Control Register (Read/Write)

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	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 Mnemonic	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 I/O Control Register (Read/Write)

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	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	P2H7	P2H6	P2H5	P2H4	P2H3	P2H2	P2H1	P2H0

Mnemonic								
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

P5CON (D9H) P5 I/O Control Register (Read/Write)

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	-	-	-	-	-	-	P5C1	P5C0
R/W	-	-	-	-	-	-	R/W	R/W
POR	x	x	X	x	x	x	0	0

P5PH (DAH) P5 Pull-up Resistor Control Register (Read/Write)

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	-	-	-	-	-	-	P5H1	P5H0
R/W	-	-	-	-	-	-	R/W	R/W
POR	x	x	X	x	x	x	0	0

Bit Number	Bit Mnemonic	Description
7 ~ 0	**PxCy** **(x=0 ~ 2, 5, y=0 ~ 7)**	Px port input and output control bit 0: Pxy as input mode (initial value) 1: Pxy as strong push-pull output mode
7 ~ 0	**PxHy**	Px port pull-up resistance configuration, only valid when PxCy=0:

	(x=0 ~ 2, 5, y=0 ~ 7)	0: Pxy as high-impedance input mode (initial value), the pull-up resistor is turned off. 1: Pxy pull-up resistance is turned on.

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

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
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

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

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	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 Mnemonic	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

P5 (D8H) P5 Data Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Mnemonic	-	-	-	-	-	-	P5.1	P5.0
R/W	-	-	-	-	-	-	R/W	R/W
POR	x	x	x	x	x	x	0	0

IOHCON (97H) IOH Configuration Register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Mnemonic	P2H[1: 0]		P2L[1: 0]		P0H[1: 0]		P0L[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

Bit Number	Bit Mnemonic	Description

7 ~ 6	**P2H[1: 0]**	P2 high 4-bit IOH configuration bits
		00: Set P2 high 4-bit IOH level 0 (Maximum value);
		01: Set P2 high 4-bit IOH level 1;
		10: Set P2 high 4-bit IOH level 2;
		11: Set P2 high 4-bit IOH level 3 (Minimum value);

5 ~ 4	**P2L[1: 0]**	P2 low 4-bit IOH configuration bits
		00: Set P2 low 4-bit IOH level 0 (Maximum value);
		01: Set P2 low 4-bit IOH level 1;
		10: Set P2 low 4-bit IOH level 2;

		11: Set P2 low 4-bit IOH level 3 (Minimum value);

3 ~ 2	**P0H[1: 0]**	P0 high 4-bit IOH configuration bits
		00: Set P0 high 4-bit IOH level 0 (Maximum value);
		01: Set P0 high 4-bit IOH level 1;
		10: Set P0 high 4-bit IOH level 2;
		11: Set P0 high 4-bit IOH level 3 (Minimum value);

1 ~ 0	**P0L[1: 0]**	P0 low 4-bit IOH configuration bits
		00: Set P0 low 4-bit IOH level 0 (Maximum value);
		01: Set P0 low 4-bit IOH level 1;
		10: Set P0 low 4-bit IOH level 2;
		11: Set P0 low 4-bit IOH level 3 (Minimum value);

15 Serial Interface 0 (UART0)

The SC92F84HX supports a full-duplex serial port. It is convenient for connecting other device or equipment, for example, WiFi module or other drive chips with UART communication interface. UART0 functions and features are shown below:

1. Three kinds of communication mode: Mode 0, Mode 1 and Mode 3;
2. Configure Timer1 or Timer2 as baud rate generator;
3. Completion of transmission and reception can generate interrupt RI/TI, and such interrupt flag needs to be cleared up by software.

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

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	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 Mnemonic	Description
7 ~ 6	**SM0 ~ 1**	Serial communication mode control bits 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 bits will be received or transmitted firstly; 01: Mode 1, 10-bit full-duplex asynchronous communication composing of 1 starting bit, 8 data bits 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 bits 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 2 and 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 2 and 3, 9th bit of receiving data
2	**RB8**	Only valid for mode 2 and 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 Mnemonic	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 Mnemonic	Description
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 Mnemonic	SMOD	-	-	-	RST	-	STOP	IDL
R/W	W	-	-	-	W	-	W	W
POR	0	x	x	x	0	x	0	0

Bit Number	Bit Mnemonic	Description
7	**SMOD**	Baud rate multiplying power configuration bit(When SM0~1 = 01 or SM0~1 = 11): 0: Serial port operates under clock of 1/1 system clock 1: Serial port operates under clock of 1/16 system clock Baud rate multiplying power configuration bit(When SM0~1 = 00 or SM0~1 = 11): 0: Serial port operates under clock of 1/12 system clock 1: Serial port operates under clock of 1/4 system clock

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

Bit Number	7	6	5	4	3	2	1	0
Bit Mnemonic	SSMOD[1: 0]		UART0OS[1:0]		-	-	SSIOS	-
R/W	R/W	R/W	R/W	R/W	-	-	R/W	-
POR	0	0	0	0	x	x	0	x

Bit Number	Bit Mnem	Description

<table>
<tr><th></th><th>onic</th><th></th></tr>
<tr><td>5~4</td><td><b>UART0OS[1:0]</b></td><td>
<table>
<tr><th>Signal</th><th>GPIO-default port
UART0OS[1:0]=00</th><th>GPIO-A group mapping
UART0OS[1:0]=01</th><th>GPIO-B group mapping
UART0OS[1:0]=10</th></tr>
<tr><td>RX0</td><td>P1.2</td><td>P5.1</td><td>P1.5</td></tr>
<tr><td>TX0</td><td>P1.1</td><td>P5.0</td><td>P1.4</td></tr>
</table>
</td></tr>
</table>

15.1 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 to 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 modes 1 and 3, the serial port clock source can be programmed as either 1 or 16 of the system clock, determined by SMOD(pcon.7) bits. When SMOD is 0, the serial port runs at 1/fsys. When SMOD is 1, the serial port runs at 16/fsys. After the serial port clock source is determined, set the baud rate overflow rate by Timer 1 or Timer 2.

- When TCLK(TXCON. 4) and RCLK(TXCON. 5) bits are both 0, then timer 1 is baud rate generator mode, and the baud rate overflow rate of UART0 is set by [TH1,TL1]. The formula is as follows, note: When timer 1 acts as a baud rate generator, timer 1 must stop counting, i.e. TR1=0:
  - SMOD = 0： $\mathrm{BaudRate} = \frac{\mathrm{fsys}}{[\mathrm{TH1,TL1}]}$; (Note: [TH1, TL1] must be bigger than 0x0010)
  - SMOD = 1： $\mathrm{BaudRate} = \frac{1}{16} * \frac{\mathrm{fsys}}{[\mathrm{TH1,TL1}]}$;
- When TCLK(TXCON. 4) or RCLK(TXCON. 5) is 1, then timer 2 is in baud rate generator mode, and the baud rate overflow rate of UART0 is set by [RCAP2H, RCAP2L], the formula is as follows:
  - SMOD = 0： $\mathrm{BaudRate} = \frac{\mathrm{fsys}}{[\mathrm{RCAP2H,RCAP2L}]}$; (Note: [RCAP2H, RCAP2L] must be bigger than 0x0010)
  - SMOD = 1： $\mathrm{BaudRate} = \frac{1}{16} * \frac{\mathrm{fsys}}{[\mathrm{RCAP2H,RCAP2L}]}$;

16 SPI/TWI/UART Serial Interface (SSI)

The SC92F84HX integrates SPI/TWI/UART serial interface circuits (SSI), which is convenient for connecting MCU to devices or equipment with different interfaces. The user can configure SSI in 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 Mnemonic	SSMOD[1: 0]		UART0OS[1:0]		-	-	SSIOS	-
R/W	R/W	R/W	R/W	R/W	-	-	R/W	-
POR	0	0	0	0	x	x	0	x

<table>
<tr><th>Bit Number</th><th>Bit Mnemonic</th><th>Description</th></tr>
<tr><td>7 ~ 6</td><td><b>SSMOD[1: 0]</b></td><td>SSI communication mode control bits
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;</td></tr>
<tr><td>1</td><td><b>SSIOS</b></td><td>
<table>
<tr><th colspan="3">Signal</th><th>GPIO-default port
SSIOS=0</th><th>GPIO-A group mapping
SSIOS=1</th></tr>
<tr><td>RX1</td><td>MOSI</td><td>SDA</td><td>P1.0</td><td>P1.4</td></tr>
<tr><td>-</td><td>MISO</td><td>-</td><td>P1.1</td><td>P2.7</td></tr>
<tr><td>TX1</td><td>SCK</td><td>SCL</td><td>P1.3</td><td>P1.5</td></tr>
</table>
</td></tr>
</table>

16.1 Serial Peripheral Interface (SPI)

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

16.1.1 SPI Operation-related Registers

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

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	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 Mnemonic	Description
7	**SPEN**	SPI Enable Control Bit 0: Disable SPI 1: Enable SPI
5	**MSTR**	SPI Master/Slave Selection Bit 0: SPI as slave equipment 1: SPI as master equipment
4	**CPOL**	Clock Polarity Control Bit 0: SCK is at low level under idle state 1: SCK is at high level under idle state
3	**CPHA**	Clock Phase Control Bit 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 Bits 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 Mnemonic	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 Mnemonic	Description
7	**SPIF**	SPI Data Transmit Flag Bit 0: Must be cleared by software 1: Data transmission completed and flag is set to 1 by hardware
6	**WCOL**	Write-in Conflict Flag Bit 0: Cleared by software, indicating write-in conflict is processed 1: Set to 1 by hardware, indicating one conflict is detected

3	**TXE**	Transmit Buffer Empty Flag Bit 0: Transmitting buffer not empty 1: Transmitting buffer empty, must be cleared by software
2	**DORD**	Transfer Direction Configuration Bit 0: Transmit MSB first 1: Transmit LSB first
0	**TBIE**	Transmitting Buffer Interrupt Enable Bit 0: An interrupt will not be generated when TXE=1 1: An interrupt will be generated when TXE=1
5 ~ 4, 1	-	Reserved

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

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	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 Mnemonic	Description
7 ~ 0	**SPD[7: 0]**	SPI Data Cache Register Data written to SSDAT will be sent to the transmitting shift register. Upon reading SSDAT, data from the receive shift register is received.

16.1.2 Signal Description

Master-Out/Slave-In (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-In and Slave-Out (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-impedance 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.

16.1.3 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 is synchronically and serially moved in or out. Serial clock line (SCK) makes data movement and sampling on two serial data lines (MOSI and MISO) keep 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. The transmit shift register and the receive shift register use the same special function address. Conducting write operations to SPI data register(SSDAT) will write data to the transmit shift register, and conducting read operations to SSDAT will obtain the data from the receive shift register.

The SPI interface of some devices will lead to SS pin (Slave Select, active-low). When communicating with the SC92F84HX SPI, the SS pin from other devices on SPI bus shall be connected based on different communication modes. The following table lists the connection modes of the SS pin from other devices on SPI bus under different communication modes of the SC92F84HX SPI:

SC92F84HX SPI	Other Devices on SPI Bus	Mode	SS of Slave Device (Slave Device Select Pins)
Master Mode	Slave Mode	One Master One Slave	Pull low
		One Master Multiple Slaves	The SC92F84HX leads to multiple I/Os, which respectively connect to the SS pin of slave device. Before data transmission, the SS pin of slave device must be pulled low
Slave Mode	Master Mode	One Master One Slave	Pull high

Master Mode

- **Mode Startup:**

  Start of all data transmission on SPI bus is controlled by SPI master device. When MSTR bit in SSCON0 register is set to 1, SPI operates in master mode, and only one 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 the transmit shift buffer. If any data already exists in the transmit shift register, one WCOL signal will be generated from master SPI to indicate writing is too fast. However, data in the transmit shift register will not be influenced and transmitting will not be interrupted as well. Besides, if the transmit shift register is empty, the master device will move the data in the transmit shift register to MOSI line serially according to SPI clock frequency on SCK. After transmission, SPIF bit in SSCON1 register will be set to 1. If SPI interrupt is allowed, when SPIF bit is set to 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 the transmit shift register to the receive shift register of master device via MISO line so as to realize full-duplex operations. Therefore, setting SPIF flag bit to 1 indicates that transmission is completed and data has been received. Data received from slave device is stored in the receive shift register of master device in accordance with MSB first or LSB first transmission direction. When one byte of data is completely moved to the receive register, the processor can obtain such data by reading SSDAT register.

Slave Mode

- **Mode Startup:**

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

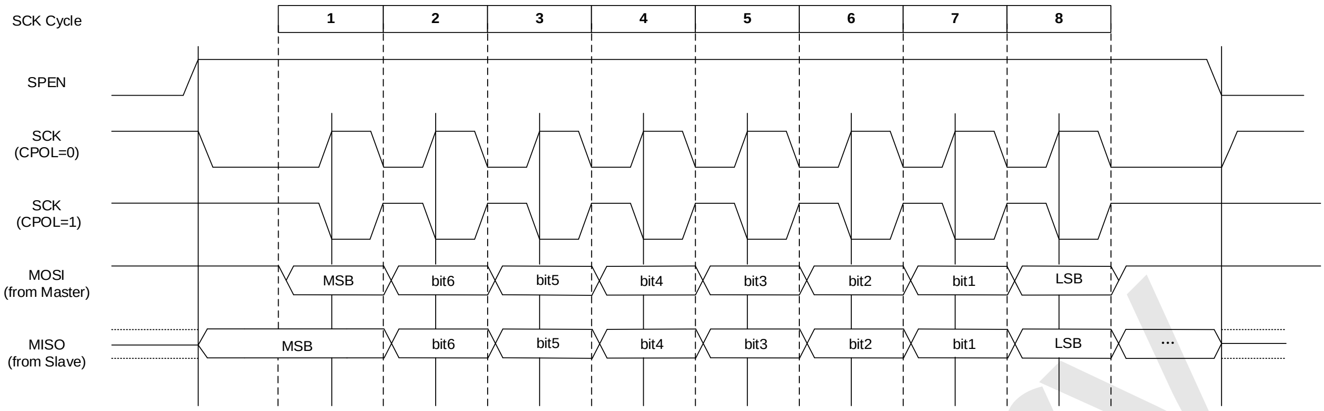
- **Transmitting and Receiving:**

  In slave mode, according to SCK signal controlled by master device, data is moved in via MOSI pin and out via MISO pin. A 1-bit counter records the number of SCK edge. When the receive shift register moves in 8-bit data (one byte) and the transmit shift register moves out 8-bit data (one byte), SPIF flag is set to 1. Data can be obtained by reading SSDAT register. If SPI interrupt is allowed, when setting SPIF to 1, an interrupt will be generated as well. At this time, the receive shift register keeps original data and set SPIF bit to 1, thus SPI slave device will not receive any data until SPIF is cleared to 0. SPI slave device must write the data to be transmitted before master device starts a new data transmission to the transmit 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 set to 1. That is to say, if data is already included in the transmit shift register, WCOL bit of SPI slave device is set to 1, indicating conflict of SSDAT writing. But the data of shift register will not be influenced and transmission will not be interrupted.

16.1.4 Transfer Form

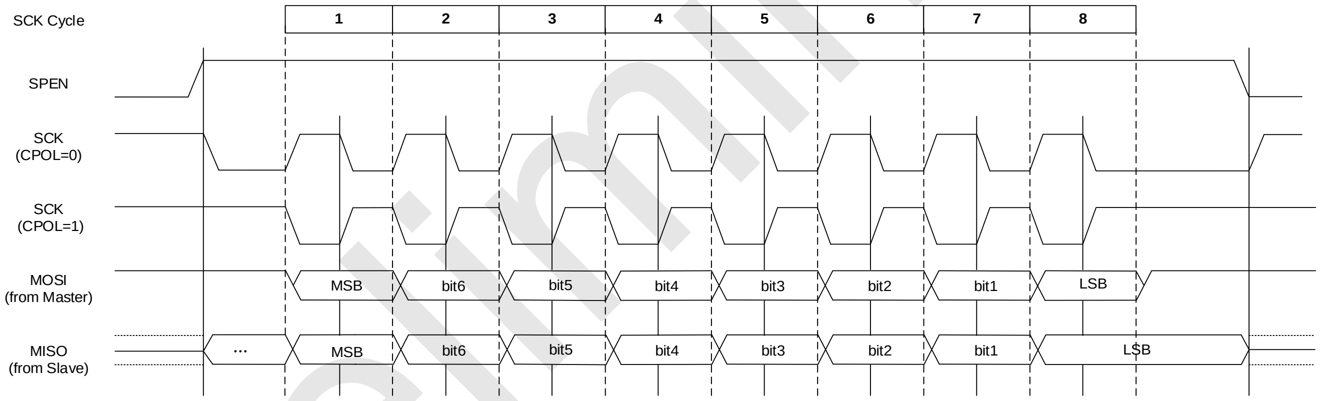
By setting CPOL bit and CPHA bit of SSCON0 register by software, the user can select four combinations of SPI clock polarity and clock phase. CPOL bit defines the polarity of clock, meaning the level status when idle, which has little influence on SPI transmission format. CPHA bit defines the phase of 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 get the data ready 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 the signal of start transmitting and start capturing data at the second edge of SCK. 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

16.1.5 Error Detection

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

16.2 Two-Wire Interface (TWI)

SSMOD[1: 0] = 10, SSI is configured as TWI interface. The SC92F84HX can only be used as slave device in TWI communication.

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

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	TWEN	TWIF	-	GCA	AA	STATE[2: 0]		
R/W	R/W	R/W	-	R	R/W	R/W	R/W	R/W
POR	0	0	x	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7	**TWEN**	TWI Enable Control Bit 0: Disable TWI 1: Enable TWI
6	**TWIF**	TWI Interrupt Flag Bit 0: cleared by software 1: Under the following conditions, interrupt flag bit will be set by hardware ① First frame of address matched successfully ②Successfully receiving or transmitting 8-bit data ③Restart ④Slave device receives stopping signal
4	**GCA**	General Address Response Flag Bit 0: Non-response general address 1: When GC = 1 and the general address matches, this bit will set to 1 by hardware and cleared to 0 automatically
3	**AA**	Receiving Enable Bit 0: Information sent by receiving master not allowed 1: Information sent by receiving master allowed

2 ~ 0	**STATE[2: 0]**	Device status flag Bits 000: slave device is in idle state, wait for TWEN to be set to 1, and detect TWI startup signal. When slave device receives stopping conditions, it will skip to this state 001: Slave device is receiving first frame of address and read and write bits (8th bit for read and write bit, 1 for reading, 0 for writing). After receiving initial conditions, slave device will skip to this state. 010: State of slave device receiving data 011: 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, setting AA to 0 and it will enter this state, waiting for restarting signal or stopping signal.
5	-	Reserved

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

Bit Number	7	6	5	4	3	2	1	0
Bit Mnemonic	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

Bit Number	Bit Mnemonic	Description
7 ~ 1	**TWA[6: 0]**	TWI Address Register
0	**GC**	TWI General Address Enable Bit

		0: Prohibits responding general address 1: Allow responding general address

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

Bit Number	7	6	5	4	3	2	1	0
Bit Mnemonic	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

Bit Number	Bit Mnemonic	Description
7 ~ 0	**TWDAT[7: 0]**	TWI Data Cache Register

16.2.1 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 a bidirectional signal line, and shall be in high level when idling, which is pulled up by pull-up resistance on SDA line.

16.2.2 Operating Modes

TWI communication of the SC92F84HX has only slave device mode:

- **Mode Startup:**

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

  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 read and write 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 read and write 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 read and write bits:

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

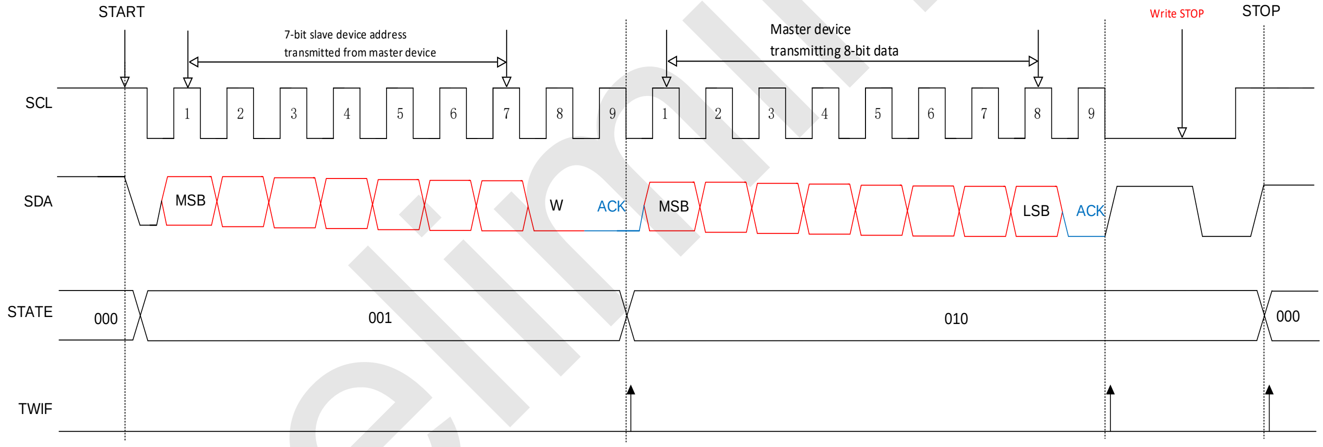
If the read and write bit received from the first frame is writing (0), the slave device enters into the receiving state of slave device (STATE [2: 0] = 010), and wait for data sent from receiving master device. Master device will release the bus for transmitting every 8 bits and then wait for the response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

1. If the response signal from slave device is in low level, there are three modes of master communication:

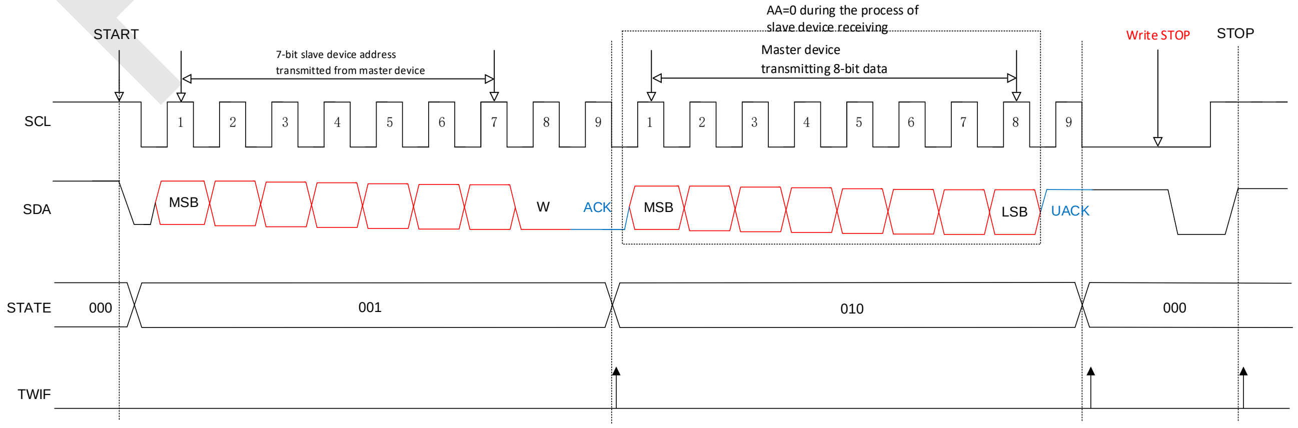
   1) Continue to send data;

   2) Resend start signal, then the slave device enters into the state of receiving first frame address (STATE[2: 0] = 001);

   3) Send stopping signal, indicating this transmission is ended, slave device returns to idle state and wait for next start signal from master device.



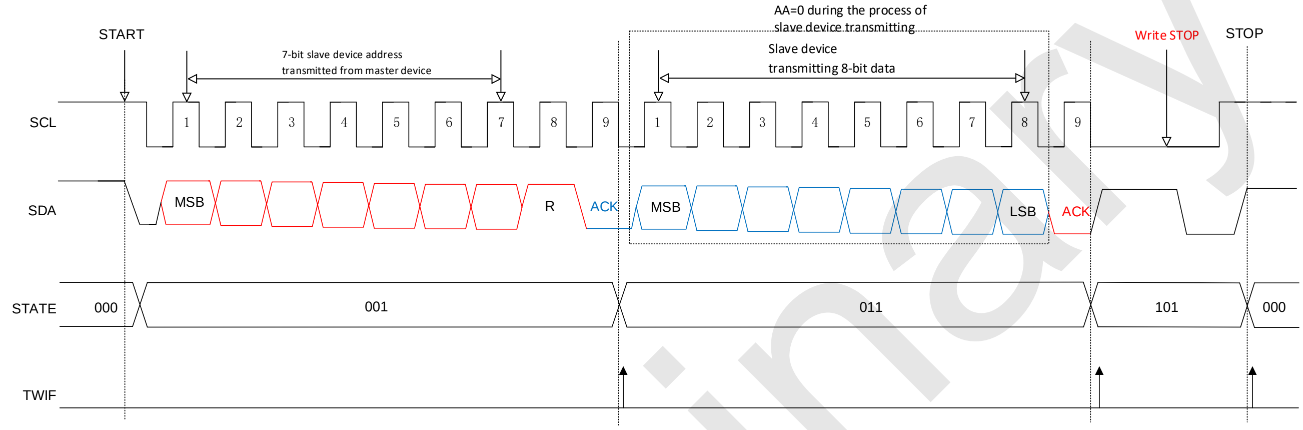
2. If the response state of slave device is in high level (during the receiving process, the value of AA in slave device register is rewritten to 0), it indicates that after transmitting current bytes, the slave device will stop this transmission automatically and return to idle state (STATE[2: 0] = 000), without receiving data sent from master device any more.



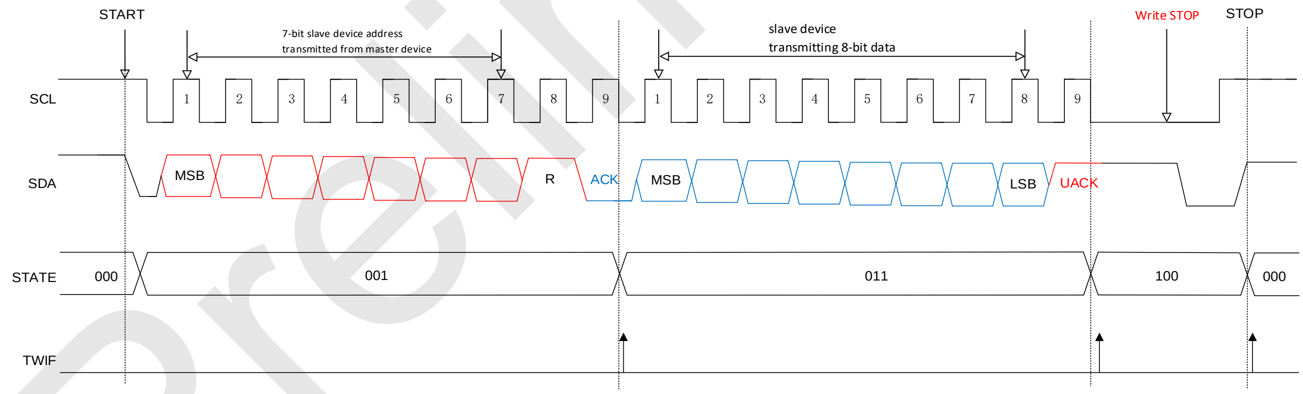
- **Non-general address response, master device transmitting mode:**

If the read and write 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 the value of AA in slave device register is rewritten to 0, the slave device will automatically end the transmission and release the bus after transmitting current bytes, and wait for stop signal or restart signal of the master device(STATE[2: 0] = 101).



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



- **Response to General Address:**

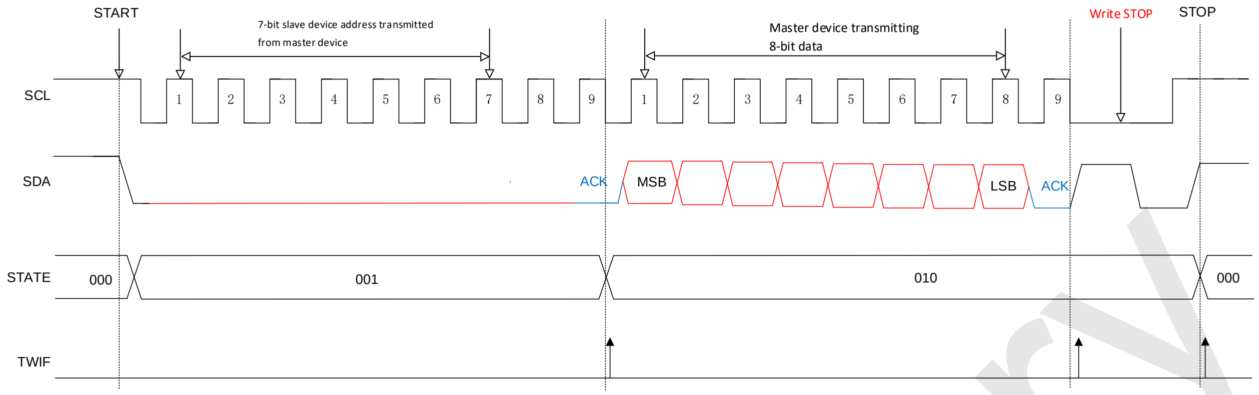
When GC=1, general address is allowed to be used. When the slave device enters into the state of receiving first frame address (STATE[2: 0] = 001), the address bit data received in first frame data will be 0x00, at this time, all slave device will respond the master device. The read and write bit sent 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;

3) Transmit the stop signal and end this communication.


START
7-bit slave device address transmitted from master device
Master device transmitting 8-bit data
Write STOP
STOP
SCL
1
2
3
4
5
6
7
8
9
1
2
3
4
5
6
7
8
9
SDA
ACK
MSB
LSB
ACK
STATE
000
001
010
000
TWIF

2. If there is no response from slave device, SDA will be in idle state.

Note: When using general address under the mode of one master and multiple slaves, the read and write bit sent by master device can not be read (1) status, or else, all the other devices on the bus will also transmit response except for equipment transmitting data.

16.2.3 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 to be set. The interrupt flag bit will be set to 1 when the slave device receives every 8-bit data. The interrupt flag bit shall be cleared by the 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.

16.3 Serial Interface 1 (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	SM0	-	SM2	REN	TB8	RB8	TI	RI

Mnemonic								
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 Mnemonic	Description
7	**SM0**	Serial Communication Mode Control Bit 0: Mode 1, 10-bit full-duplex asynchronous communication composing of 1 starting bit, 8 data bits and 1 stopping bit, with communication baud rate changeable; 1: Mode 3, 11-bit full-duplex asynchronous communication, composing of 1 starting bit, 8 data bits 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: Configure RI for receiving each complete data frame to generate interrupt request; 1: When receiving a complete data frame and only when RB8=1, will RI be configured 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**	Transmit Interrupt Flag Bit
0	**RI**	Receive Interrupt Flag Bit
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 Mnemonic	BAUDL [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 Mnemonic	BAUDH [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 Mnemonic	Description
7 ~ 0	**BAUD1 [15: 0]**	Serial Port Baud Rate Control Bit $\mathrm{BaudRate} = \frac{\mathrm{fsys}}{\mathrm{BAUD1H, BAUD1L}}$ **Note: [BAUD1H, BAUD1L] must be larger than 0x0010**

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

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	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 Mnemonic	Description
7 ~ 0	**SBUF[7: 0]**	Serial Data Buffer SBUF contains two registers: one for transmit shift register and one for receiving latch, data writing to SBUF will be sent to shift register and initiate transmitting process, reading SBUF will return the contents of receiving latch.

17 Analog-to-Digital Converter (ADC)

The SC92F84HX has a 12-bit high-precision successive approximation ADC with 21-channel, the external 20 ADC channel is multiplexing with other IO ports. Cooperating with the internal 2.4V reference voltage, one internal channel connected to 1/4 V_{DD} can be used for measuring V_{DD} voltage.

There are 2 options for ADC reference voltage:

① VDD pin (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: The clock source of ADC circuit is fixed as f_{HRC} = 24MHz, which will not change with the switch of internal and external system clocks.

17.1 ADC-related Registers

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

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	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 Mnemonic	Description
7	**ADCEN**	ADC Power Control Bit 0: Disable ADC module power 1: Enable ADC module power
6	**ADCS**	ADC Start Trigger Control Bit (ADC Start) Write “1” for this bit, an ADC conversion started, 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 and need the user cleared up by software. ADC conversion completion flag EOC: when the user sets up ADCS for conversions, this bit will be cleared 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 ADC interrupt is enabled, this bit must be cleared by the user with software after ADC interrupt generated.
4 ~ 0	**ADCIS[4: 0]**	ADC Input Selection Bits 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: Select AIN7 as ADC input 01000: Select AIN8 as ADC input 01001: Select AIN9 as ADC input 01010: Select AIN10 as ADC input 01011: Select AIN11 as ADC input 01100: Select AIN12 as ADC input 01101: Select AIN13 as ADC input 01110: Select AIN14 as ADC input 01111: Select AIN15 as ADC input 10000: Select AIN16 as ADC input 10001: Select AIN17 as ADC input 10010: Select AIN18 as ADC input 10011: Select AIN19 as ADC input

		10100 ~ 11110: Reserved 11111: ADC input is 1/4 V_{DD}, used for measuring power voltage

ADCCFG2 (B5H) ADC Configuration Register 2 (Read/Write)

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	-	-	-	-	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 Mnemonic	Description
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 controls ADC sampling clock frequency, conversion clock frequency of ADC is controlled by ADCCK[2: 0], independent of the influence of LOWSP bit The whole process from sampling to conversion of 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;
		000: Configure ADC clock frequency f_{ADC} as f_{HRC}/16;
		000: Configure ADC clock frequency f_{ADC} as f_{HRC}/12;

		000: Configure ADC clock frequency f_{ADC} as f_{HRC}/8;
		000: Configure ADC clock frequency f_{ADC} as f_{HRC}/6;
		000: Configure ADC clock frequency f_{ADC} as f_{HRC}/4;
		000: Configure ADC clock frequency f_{ADC} as f_{HRC}/3;
		Note: The clock source of the ADC circuit is fixed as fHRC = 24MHz and will not change with the switch 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 Mnemonic	EAIN7	EAIN6	EAIN5	EAIN4	EAIN3	EAIN2	EAIN1	EAIN0
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

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

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	EAIN15	EAIN14	EAIN13	EAIN12	EAIN11	EAIN10	EAIN9	EAIN8
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

ADCCFG3 (AAH) ADC Configuration Register 1 (Read/Write)

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	-	-	-	-	EAIN19	EAIN18	EAIN17	EAIN16
R/W	-	-	-	-	R/W	R/W	R/W	R/W
POR	x	x	x	x	0	0	0	0

Bit Number	Bit Mnemonic	Description
0	**EAINx** **(x=0 ~ 19)**	ADC Port Configuration Register 0: The AINx corresponding port cannot be used as an ADC input channel 1: The AINx corresponding port can be used as an ADC input channel, when ADCIS[4:0] selects AINx as the ADC input channel, the pull-up resistor on the AINx corresponding port will be automatically removed.

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

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

Bit Number	Bit Mnemonic	Description
7	**VREFS**	Reference Voltage Selection Bit (Default values are configured by the user and loaded from Code Option) 0: Configure ADC VREF as V_{DD}

		1: Configure ADC VREF as internal 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 Mnemonic	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 Mnemonic	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 Mnemonic	Description
11 ~ 4	**ADCV[11: 4]**	ADC conversion value high byte values
3 ~ 0	**ADCV[3: 0]**	ADC conversion value low 4-bit values

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

Bit Number	7	6	5	4	3	2	1	0
Bit Mnemonic	EA	EADC	ET2	-	ET1	EINT1	ET0	EINT0

R/W	R/W	R/W	R/W	-	R/W	R/W	R/W	R/W
POR	0	0	0	x	0	0	0	0

Bit Number	Bit Mnemonic	Description
6	**EADC**	ADC Interrupt Enable Control Bit 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 Mnemonic	-	IPADC	IPT2	-	IPT1	IPINT1	IPT0	IPINT0
R/W	-	R/W	R/W	-	R/W	R/W	R/W	R/W
POR	x	0	0	x	0	0	0	0

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

17.2 ADC Conversion Steps

Operating steps for the 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 Enable ADCS, and start conversion;

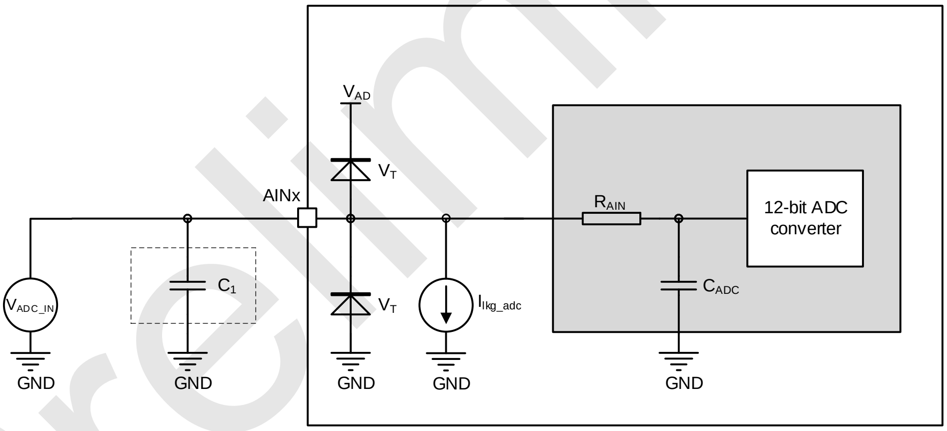
⑥ Wait for EOC/ADCIF=1, if ADC interrupt is enabled, ADC interrupt will be generated and the user shall clear EOC/ADCIF flag to 0 by software;

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

17.3 ADC Structure Diagram



Note:

1. **C1 is an external 0.01μF capacitor. Users are advised to add this capacitor to improve the performance of the ADC.**
2. **For detailed electrical parameters related to the ADC, please refer to Section 21.5 ADC Characteristics.**

18 High Sensitivity TouchKey Module

The SC92F84HX is equipped with a built-in high sensitivity capsense TouchKey circuit with 20-channel, its features are shown below:

1. Can be applied to TouchKey application featuring high sensitivity requirements including TouchKey sensor, proximity sensing, etc
2. Can pass 10V dynamic CS testing.
3. Realize 20-channel TouchKey sensors and derivative functions
4. High-flexibility development software library is provided for support with low development difficulty
5. Automatic debugging software is provided for support and featuring intelligent development
6. TouchKey module can operate in low-consumption mode under MCU STOP mode, when adopting single TouchKey to wake up STOP, the overall power consumption can be as low as 8uA.

Note: The clock source of the high sensitivity TouchKey circuit is fixed as f_{HRC} = 24MHz, and it will not change with the conversion of internal and external system clock.

18.1 Power Saving Modes of TouchKey Module

The SC92F84HX allows TouchKey scan function in STOP Mode, which can reduce the overall power consumption of MCU and satisfy the TouchKey application with low power consumption requirements.

There is two power consumption modes of the SC92F84HX TouchKey circuit:

1. Normal Run Mode
2. Low Consumption Run Mode

The definitions of these two power consumption modes are shown as follows:

Description	Normal Run Mode	Low Power Consumption Run Mode
CPU	RUN (Normal mode)	Stop (STOP Mode)
TouchKey circuit	RUN	RUN

19 EEPROM and IAP Operations

There are two options for the SC92F84HX IAP operating scope:

EEPROM and IAP operating modes are shown below:

1. Internal highest address 128 bytes EEPROM can be used as data storage;
2. The whole 16K bytes of IC ROM and 128 bytes of EEPROM can be used for IAP operations, which is mainly used for remote program updating.

As Code Option, the user shall select IAP operating space 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 Mnemonic	VREFS	XTLHF	-	DISJTG	IAPS[1: 0]		-	-
R/W	R/W	R/W	-	n	R/W	R/W	-	-
POR	n	n	x	R/W	n	n	x	x

Bits	Name	Description
3 ~ 2	**IAPS[1: 0]**	EEPROM and IAP Area Selection Bits 00: Code memory prohibits IAP operations, only EEPROM data memory is used for data storage 01: last 0.5k code memory allows IAP operation (3E00H ~ 3FFFH) 10: Last 1k code memory allows IAP operation (3C00H ~ 3FFFH) 11: All code memory allows IAP operation (0000H ~ 3FFFH)

19.1 EEPROM / IAP Operating-related Registers

Description for EEPROM / IAP operating-related registers:

Mnemonic	Add	Description	7	6	5	4	3	2	1	0	POR

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 Mnemonic	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 Mnemonic	Description
7 ~ 0	**IAPKEY[7: 0]**	Enable EEPROM/IAP function and operation time limit configuration, Written values must be non-zero: 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 Mnemonic	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 Mnemonic	Description
7 ~ 0	**IAPADR[7: 0]**	EEPROM/IAP writing address low byte

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

Bit Number	7	6	5	4	3	2	1	0
Bit Mnemonic	-	-	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 Mnemonic	Description
5 ~ 0	**IAPADR[13: 8]**	EEPROM/IAP writing address high 6-bits
7 ~ 6	-	Reserved

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

Bit Number	7	6	5	4	3	2	1	0

Bit Mnemonic	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 Mnemonic	Description
7 ~ 0	**IAPADER[7: 0]**	IAP Extended Address: 0x00: MOVC and IAP programming for Code 0x01: Perform reading operation on user’s ID area, but can’t perform writing operation 0x02: MOVC and IAP programming for EEPROM Other: Reserved

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

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	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 Mnemonic	Description

7 ~ 0	**IAPDAT**	Data written by IAP

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

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	-	-	-	-	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 Mnemonic	Description
3 ~ 2	**PAYTIMES[1: 0]**	Upon EEPROM/IAP writing operation, CPU Hold Time length configuration 00: Configure CPU HOLD TIME 4mS@12/6/2 MHz 01: Configure CPU HOLD TIME 2mS@12/6/2 MHz 10: Configure CPU HOLD TIME 1mS@12/6/2 MHz 11: Reserved **Notes: The 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.** Recommended Selection: 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!**

19.2 EEPROM / IAP Operating Procedures:

Writing procedure of the SC92F84HX EEPROM/IAP are shown below:

① Write 0x00 into IAPADE[7: 0]: select Code memory 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] (configue CPU Hold time by setting CMD[1: 0] to 1 or 0, CPU is Hold up and start up EEPROM/IAP writing);

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

Notes:

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

2. **When IAPADE= 0x01 or 0x02, MOVC instruction and writing operation are conducted in EEPROM or IFB data memory. If any interrupt occurs and there are also MOVC operations in this interrupt, it may result in error of MOVC operations and thus abnormal operation of program. To avoid such situation, the user shall disable global interrupt control bit (EA=0) before conducting IAPADE=0x01 or 0x02 operations. After operation completed, configure IAPADE =0x00 and enable global interrupt control bit (EA=1).**

19.2.1 128 bytes Independent EEPROM Operating Demo program

```
#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 data memory

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, 1ms@12/6/2 MHz;
_nop_ ();                       //Wait (at least 8 _nop_ ())
_nop_ ();
_nop_ ();
_nop_ ();
_nop_ ();
_nop_ ();
_nop_ ();
_nop_ ();
_nop_ ();
IAPADE = 0x00;                  //Return to ROM data memory
EA = 1;                         //Enable master interrupt
```

C Demo Program of EEPROM Read Operation

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

19.2.2 8K bytes Code memory IAP Operating Demo program

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

C Demo Program of IAP Write Operation:

```
IAPADE = 0x00;                        //Select Code memory
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, 1ms@12/6/2MHz;
_nop_ ();                             //Wait (at least 8 _nop_ ())
_nop_ ();
_nop_ ();
_nop_ ();
_nop_ ();
_nop_ ();
_nop_ ();
_nop_ ();
```

C Demo Program of IAP Read Operation:

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

Note: IAP operation in 16K bytes Code memory has certain risks, the user shall implement corresponding safety measures in software. Incorrect operation may result in the user program to be rewritten! Unless such function is required by the user (such as used for remote program update, etc.), it is not recommended to used by the user.

20 CheckSum Module

The SC92F84HX is equipped with a check sum module, which is used for generating 16-bit check sum of code memory in real time. The user can compare such check sum with theoretical value to monitor whether the contents in code memory are correct.

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

20.1 CheckSum-Related Registers

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

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	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 Mnemonic	Description
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 Mnemonic	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 Mnemonic	Description
7 ~ 0	**CHKSUMH [7: 0]**	CheckSum Result Register High Bit

OPERCON (EFH) Arithmetic Control Register (Read/Write)

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

Bit Number	Bit Mnemonic	Description
0	**CHKSUMS**	CheckSum Operation Starts Trigger **C**ontrol Bit (Start) Write “1” for this bit, start to conduct Check sum calculation. This bit is valid for only writing 1.

21 Electrical Characteristics

21.1 Absolute Maximum Ratings

Symbol	Parameter	Min Value	Max Value	Unit
VDD/VSS	DC supply voltage	-0.3	6	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	℃
I_{VDD}	Current value flowing through VDD	-	150	mA
I_{VSS}	Current value flowing through VSS	-	150	mA

21.2 Recommended Operating Conditions

Symbol	Parameter	Min Value	Max Value	Unit	System Clock Frequency
V_{DD}	Operating voltage	2.4	5.5	V	≤12MHz
T_A	Ambient temperature	-40	85	℃	-

21.3 DC Characteristics

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

Symbol	Parameter	Min Value	Typical Value	Max Value	Unit	Testing Conditions

Current						
I_{op1}	Operating current	-	6.5	-	mA	f_{SYS} =12MHz
I_{op2}	Operating current	-	5.1	-	mA	f_{SYS} =6MHz
I_{op3}	Operating current	-	4.2	-	mA	f_{SYS} =2MHz
I_{pd1}	Standby Current (Power Down Mode)	-	0.7	1.0	μA	
I_{IDL1}	Standby Current (IDLE Mode)	-	3.8	-	mA	
I_{BTM}	Base Timer Operating Current	-	4	6	μA	BTMFS[3: 0]= 1000 One interrupt occurs for every 4.0 seconds
I_{WDT}	WDT Current	-	2.5	3.5	μA	WDTCKS[2: 0]= 000 WDT overflows every 500ms
I_{TK1}	TouchKey operating current (High Sensitivity Mode)	-	1.3	1.7	mA	
I/O 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	-	50	-	mA	V_{Pin}=0.4V
I_{OL2}	Output low current	-	100	-	mA	V_{Pin}=0.8V
I_{OH1}	Output high current P1/P5	-	20	-	mA	V_{Pin}=4.3V
I_{OH2}	Output high current P1/P5	-	10	-	mA	V_{Pin}=4.7V
I_{OH3}	Output high current P0/P2	-	20	-	mA	V_{Pin}=4.3V Pxyz = 0, I_{OH} level 0
	Output high current P0/P2	-	10	-	mA	V_{Pin}=4.3V Pxyz = 1, I_{OH} level 1
	Output high current P0/P2	-	5.5	-	mA	V_{Pin}=4.3V Pxyz = 2, I_{OH} level 2
	Output high current P0/P20	-	1.8	-	mA	V_{Pin}=4.3V Pxyz = 3, I_{OH} level 3
I_{OH4}	Output high current P0/P2	-	9.7	-	mA	V_{Pin}=4.7V Pxyz = 0, I_{OH} level 0
	Output high current P0/P2	-	4.5	-	mA	V_{Pin}=4.7V Pxyz = 1, I_{OH} level 1
	Output high current P0/P2	-	2.4	-	mA	V_{Pin}=4.7V Pxyz = 2, I_{OH} level 2

	Output high current P0/P2	-	0.8	-	mA	V_{Pin}=4.7V Pxyz = 3, I_{OH} level 3
I_{lkg1}	Input leakage current	-1	-	1	μA	IO is in high-impedance input mode V_{IN}=V_{DD} or V_{SS}
R_{PH1}	Pull-up resistance	16.5	33	49.5	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	Min Value	Typical Value	Max Value	Unit	Conditions
Current						
I_{op4}	Operating current	-	5.6	-	mA	f_{SYS} =12MHz
I_{op5}	Operating current	-	4.5	-	mA	f_{SYS} =6MHz
I_{op6}	Operating current	-	3.7	-	mA	f_{SYS} =2MHz
I_{pd2}	Standby Current (Power Down Mode)	-	0.6	1.0	uA	
I_{IDL2}	Standby Current (IDLE Mode)	-	3.2	-	mA	
I_{TK2}	TouchKey operating current (High Sensitivity Mode)	-	1.2	1.5	mA	

I/O 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	-	70	-	mA	V_{Pin} =0.8V
I_{OL4}	Output low current	-	40	-	mA	V_{Pin}=0.4V
I_{OH5}	Output high current P1/P5	-	7	-	mA	V_{Pin}=3.0V
I_{OH6}	Output high current P0/P2	-	7	-	mA	V_{Pin}=3.0V Pxyz = 0, I_{OH} level 0
	Output high current P0/P2	-	4	-	mA	V_{Pin}=3.0V Pxyz = 0, I_{OH} level 1
	Output high current P0/P2	-	1.8	-	mA	V_{Pin}=3.0V Pxyz = 0, I_{OH} level 2
	Output high current P0/P2	-	0.6	-	mA	V_{Pin}=3.0V Pxyz = 0, I_{OH} level 3
I_{lkg2}	Input leakage current	-1	-	1	μA	IO is in high-impedance input mode $V_{IN}=V_{DD}$ or V_{SS}
R_{PH2}	Pull-up resistance	28	56	84	kΩ	

Internal calibrated 2.4V as ADC reference voltage						
V_{DD24}	Internal reference 2.4V voltage output	2.38	2.40	2.42	V	TA=-40 ~ 85℃

21.4 AC Characteristics

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

Symbol	Parameter	Min	Typical	Max	Unit	Conditions
T_{OSC}	External 32.768kHz oscillator start-up time	-	-	1	s	External 32.768 kHz crystal oscillator
T_{POR}	Power On Reset time	-	12	18	ms	
T_{PDW}	Power Down Mode waking-up time	-	65	130	μs	
T_{Reset}	Reset Pulse Width	18	-	-	μs	Valid for Low level
T_{LVR}	LVR buffeting time	-	30	-	μs	
f_{HRC}	RC oscillation stability	23.76	24	24.24	MHz	V_{DD}=2.9 ~ 5.5V T_A=-40 ~ 85 ℃

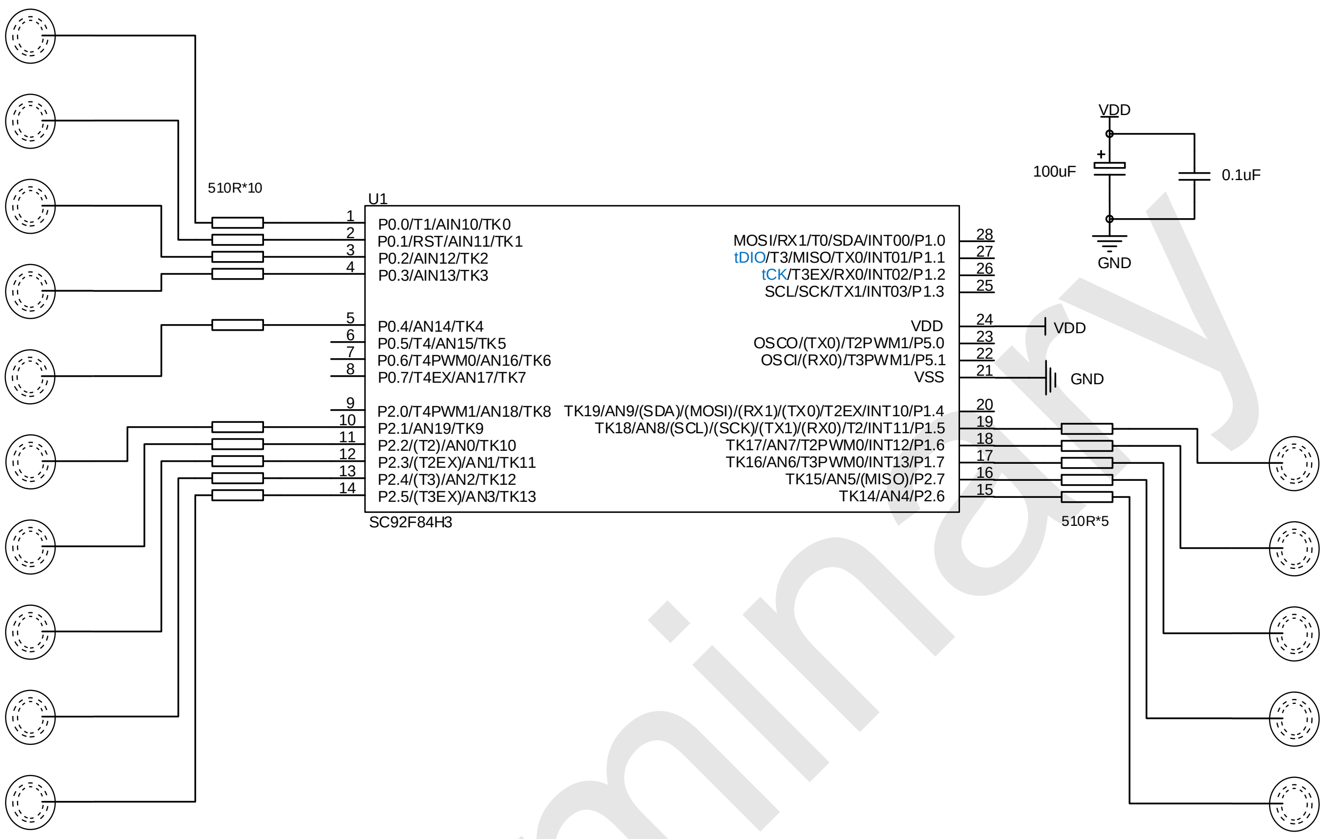
21.5 ADC Characteristics

(T_A= 25℃, unless otherwise specified)

Symbol	Parameter	Min	Typical	Max	Unit	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_{lkg_ADC}	ADC Input leakage current	-1	-	1	μA	V_{IN}= V_{AINx}
I_{ADC1}	ADC conversion current 1	-	2.7	3.2	mA	ADC Module on V_{DD}=5V
I_{ADC2}	ADC conversion current 2	-	2.1	2.5	mA	ADC module on V_{DD}=3.3V
DNL	Differential nonlinear error	-	±2	-	LSB	V_{DD}=5V V_{REF}=5V
INL	Integral nonlinear error	-	±2	-	LSB	
E_Z	Offset error	-	±5	-	LSB	
E_F	Full scale error	-	±8	-	LSB	
E_{AD}	Total absolute error	-	±8	-	LSB	
T_{ADC1}	ADC conversion time 1	-	10	-	μs	ADC Clock = 2MHz ADC sampling period =6
T_{ADC2}	ADC conversion time 2	-	20	-	μs	ADC Clock = 1MHz ADC sampling period =6

22 Application Circuit



The SC92F83A3 TouchKey Application Circuit Diagram

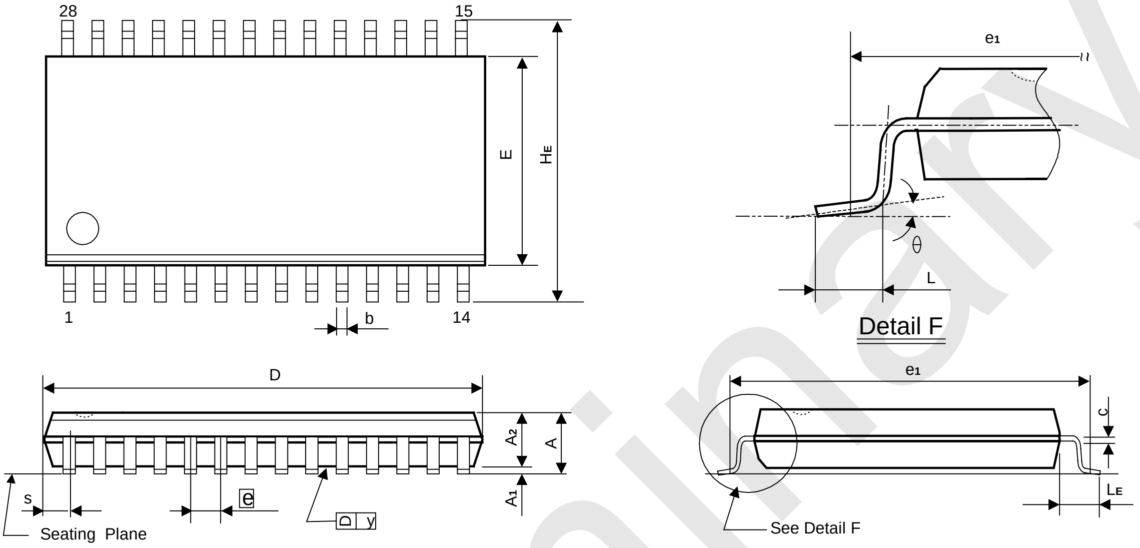
23 Ordering Information

PRODUCT NO	PKG	PACKING
SC92F84H3M28U	SOP28	TUBE
SC92F84H3X28U	TSSOP28	TUBE
SC92F84H9M24U	SOP24	TUBE
SC92F84H9X24U	TSSOP24	TUBE
SC92F84H2M20U	SOP20	TUBE
SC92F84H2X20U	TSSOP20	TUBE

24 Packageing Information

SC92F84H3M28U

SOP28L (300mil) Dimension Unit: mm

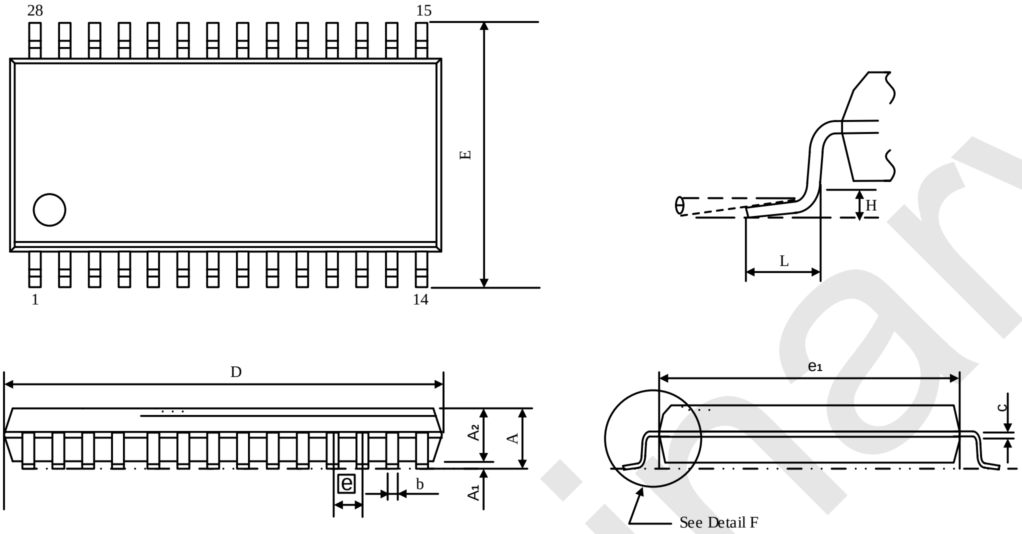


Symbol	mm (milimetre)		
	Min	Normal	Max
A	2.40	2.56	2.65
A**1**	0.100	0.200	0.300
A**2**	2.240	2.340	2.440
b	0.39	---	0.48
C	0.254(BSC)		
D	17.80	18.00	18.20
E	7.30	7.50	7.70

HE	10.100	10.300	10.500
e	1.270(BSC)		
L	0.7	0.85	1.0
LE	1.3	1.4	1.5
θ	0°	-	8°

SC92F84H3X28U

TSSOP28 Dimension Unit: mm

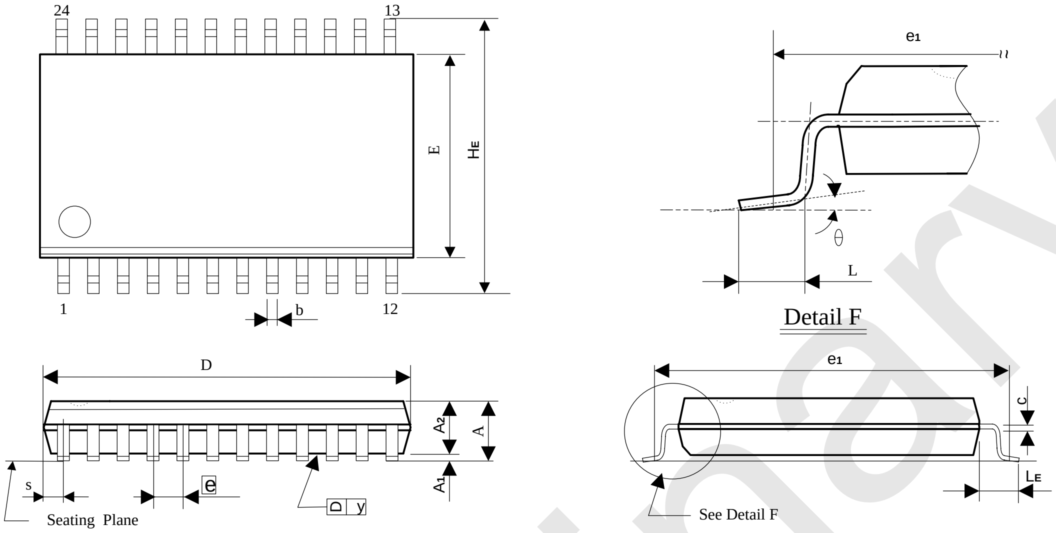


Symbol	mm (milimetre)		
	Min	Normal	Max
A	-	-	1.200
A1	0.050	-	0.150
A2	0.800	-	1.050
b	0.190	-	0.300
c	0.090	-	0.200
D	9.600	-	9.800
E	6.250	-	6.550
e1	4.300	-	4.500

e	0.65(BSC)		
L	-	-	1.0
θ	0°	-	8°
H	0.05	-	0.25

SC92F84H9M24U

SOP24L (300mil) Dimension Unit: mm

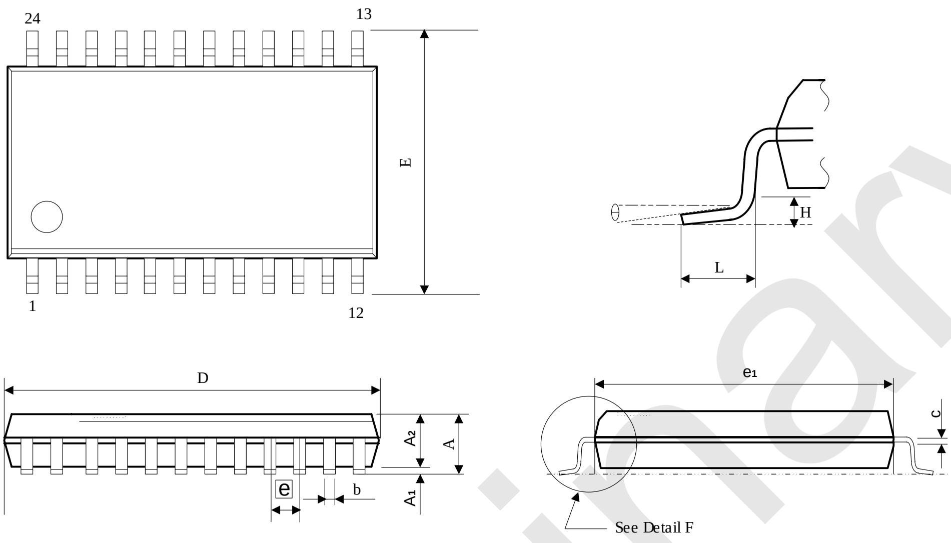


Symbol	mm (milimetre)		
	Min	Normal	Max
A	2.40	2.56	2.65
A1	0.100	0.200	0.300
A2	2.240	2.340	2.440
b	0.39	--	0.48
C	0.254(BSC)		
D	15.240(BSC)		
E	7.374	7.450	7.574
HE	10.100	10.300	10.500

e	1.27(BSC)		
L	0.7	0.85	1.0
LE	1.3	1.4	1.5
θ	0°	-	8°

SC92F84H9X24U

TSSOP24L Dimension Unit: mm

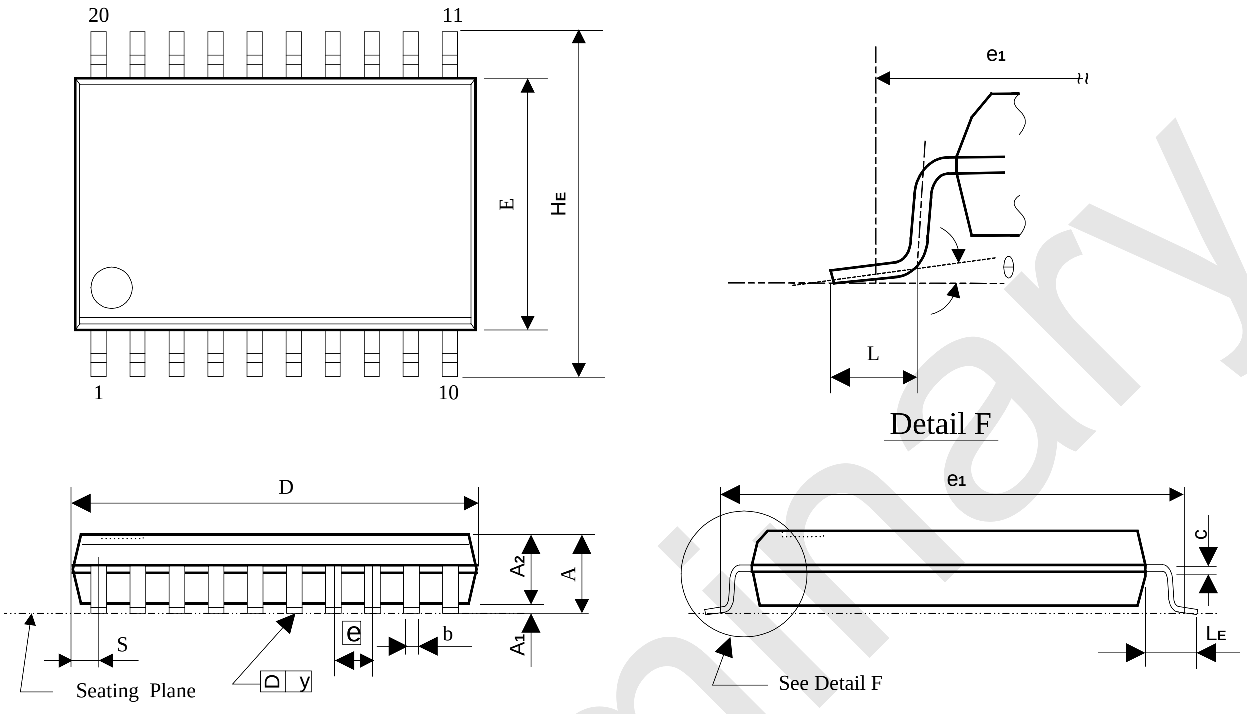


Symbol	mm (milimetre)		
	Min	Normal	Max
A	-	-	1.200
A**1**	0.050	-	0.150
A**2**	0.800	-	1.000
b	0.190	-	0.300
c	0.090	-	0.200
D	7.700	-	7.900
E	6.250	-	6.550
e1	4.300	-	4.500

e	0.65(BSC)		
L	0.450	-	0.750
θ	0°	-	8°
H	-	0.25	-

SC92F84H2M20U

SOP20L (300mil) Dimension   Unit: mm

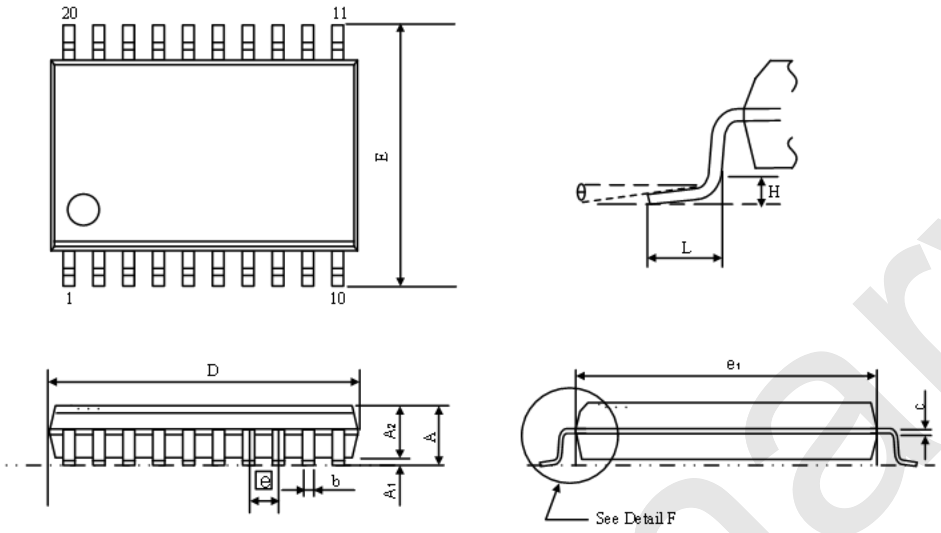


Symbol	mm (milimetre)		
	Min	Normal	Max
A	2.40	2.56	2.65
A1	0.100	0.200	0.300
A2	2.240	2.340	2.440
b	0.35	--	0.47
c	0.25	--	0.31
D	12.60	12.80	13.00
E	7.30	7.50	7.70

HE	10.100	10.300	10.500
e	1.27(BSC)		
L	0.700	0.850	1.000
LE	1.30	1.40	1.50
θ	0°	-	8°

SC92F84H2X20U

TSSOP20L Dimension Unit: mm



Symbol	mm (milimetre)		
	Min	Normal	Max
A	-	-	1.200
A**1**	0.050	-	0.150
A**2**	0.800	-	1.050
b	0.190	-	0.300
c	0.090	-	0.200
D	6.400	-	6.600
E	6.20	-	6.60
e1	4.300	-	4.500
e	0.65(BSC)		

L	-	-	1.00
θ	0°	-	8°
H	0.05		0.15

25 Revision History

Version	Notes	Date
V0.1	Initial Release.	December 2024

Important Notice

Shenzhen SinOne Microelectronics Co., Ltd. (hereinafter referred to as SinOne) reserves the right to change, correct, enhance, modify and improve SinOne products, documents or services at any time without prior notice. SinOne considers the information provided to be accurate and reliable. The information in this document will be used in December 2024. In the actual production design, please refer to the latest data manual of each product and other relevant materials.