Super High-speed 1T 8051 Core Flash MCU, 2 Kbytes SRAM, 16 Kbytes Flash, 0~4 Kbytes LDROM, 1 Kbytes Independent EEPROM, 23-channel low-power and high sensitivity TouchKey circuits ,11-channel 12-bit ADC, 6-channel 10-bit PWM, 3 Timers, MDU, UART, SSI, Check Sum Module

1 General Description

SC92F8483/8482/8481/8480 (hereinafter referred to as the SC92F848X) is a series of based on 1T 8051 core, embedded Flash, instruction set is fully compatible with standard 80C51 enhanced microcontroller. Its core adopts enhanced pipeline architecture, and its processing capacity is 12 times faster than that of standard 80C51 at the same clock frequency. It can cooperate with Keil C debugging and development software, and supports C language and assembly language.

SC92F848X series has high performance and reliability, with a wide operating voltage of 2.0V~5.5V, ultra-wide operating temperature of -40℃~105℃, and has good ESD performance and EFT anti-interference ability. Using the industry leading eFlash process, Flash write more than 100,000 times, and can be stored for 100 years at room temperature.

SC92F848X has a built-in 23-channel touch circuit with low-power consumption and high sensitivity. The touch circuit can optionally run in STOP Mode. SC92F848X offers a wealth of peripherals, includes 256 bytes SRAM, 1792 bytes external RAM(XRAM), up to 26 general-purpose I/O, 13 externally interruptable I/O, 3 16-bit timer/counter Timer0/1/2, a watchdog timer (WDT), a BaseTimer for timed wake up, One UART, one UART/SPI/IIC triplex SSI, six 10-bit PWM output channels, MDU, Check Sum module, and 11 channels 12-bit ADC. There are 12 interrupt sources with two priorities.

SC92F848X has a built-in high precision (±2%) oscillator running up to 32MHz of system clock frequency over a wide voltage range of 2.0V to 5.5V. Built in 32kHz low frequency internal oscillator.

SC92F848X has five reset modes, including external RST pin reset, level 4 LVR reset, power-on reset, watchdog reset and software reset, which provide a guarantee for high reliability system design.

SC92F848X's 1T 8051 high-performance core, low-power consumption performance and rich well-designed peripheral modules make it suitable for industrial control and consumer applications such as IoT, smart home appliances, chargers, power supplies, model airplanes, wireless communications, game consoles and so on.

2 Features

Operating Conditions

- operating voltage: 2.0V~5.5V
- operating temperature: -40~105℃

EMS

- **ESD**

HBM：MIL-STD-883J Class 3A

MM：JEDEC EIA/JESD22-A115 Class C

- **EFT**

EN61000-4-4 Level 4

Core

- Fully static 8-bit 1T 8051 core CMOS microcontroller
- Instruction set fully compatible with MCS-51
- Level 2 Priority interrupt configuration

Flash ROM

- 16 Kbytes Flash ROM

The APROM area can be set to the range of IAP operation to 0K/1K/2K/ all APROM through the Code Option setting item

- LDROM

Can used to store the user's BootLoader code.

The LDROM can be set to 0K/1K/2K/4K by Code Option setting

- EEPROM

1K bytes independent EEPROM

write more than 100,000 times, and can be stored for 100 years at room temperature.

SRAM

- 256 bytes on-chip direct access RAM
- Additional 1792 bytes of In-chip Indirect Access RAM(XRAM)

System clock (f_{SYS})

- Built-in high frequency 32 MHz oscillator (HRC)

The system clock for IC operation can be set to 32/16/8/2.66mhz @2.0~5.5V by programmer choice

In the full voltage range (2.0V~5.5V), -40 ~ 105℃ application environment, the frequency error is not more than ±2%

Built-in low-frequency 32 kHz oscillator (LRC):

- used as the clock source for Base Timer and WDT and wake up STOP
- Frequency error: across (4.0~5.5V) and (-20 ~ 85℃) application environment, after the register correction frequency error is not more than ±4%

Low-voltage Reset (LVR)

- 4 options of reset voltage: 4.3、3.7V、2.9V、1.9V
- the default value can be selected by the Code Option

Flash program and simulation

- 2 - line JTAG write, simulation interface, support live simulation

Interrupts (INT)

- Timer0~2，INT0~2，ADC，PWM，UART，SSI，Base Timer，TK 12 interrupt sources
- External interrupt contains 3 interrupt vectors, 13 interrupt ports. All can set up rising edge, falling edge, dual edge interrupt.
- Two-level interrupt priority capability

Digital Peripheral

- GPIO: Up to 26 bidirectional independently controllable I/O ports

Independent setting of pull-up resistors

P0 and P2 port source drive capacity is controlled by four levels

All IO ports have large sink current drive capability (50mA)

- Built-in WDT, optional clock frequency division ratio
- Three standard 80C51 timers, Timer0, Timer1, and Timer2

Timer2 provides Capture

- 6-channel 10-bit PWM

Shared cycle, duty cycle can be adjusted separately

Can output three groups of complementary PWM waveform with dead zone at the same time

- One independent UART communication port
- 1 UART/SPI/IIC communication interfaces (SSI)
- Integrated with 16 * 16-bit hardware Multiplier-Divide Unit (MDU)
- Check Sum module

Analog Peripheral

- 23-channel high sensitivity TouchKey circuit.
  - Supports low-power mode
  - Applicable to TouchKey sensor, proximity induction and other TouchKey applications featuring high requirements on sensitivity
  - Can pass 10V dynamic CS test
  - Support low-power consumption mode
  - It can realize 23 high sensitivity space touch keys and derivative functions
  - Complete development support: High-flexible touch software library, intelligent software of debugging
  - High flexibility development software library support, low development difficulty
  - Automatic debugging software support, intelligent development
- 11-channel 12-bit±2LSB ADC
  - Built-in 1.024V, 2.4V and 2.048V reference voltages
  - The ADC has four reference voltages to choose from: VDD, 1.024V, 2.4V, and 2.048V
  - 1 internal channel can measure the 1/4 voltage of the power supply
  - ADC conversion can be set to complete the interrupt

Power Saving Mode

- IDLE Mode: can be woken up by any interrupt
- STOP Mode: can be woken up by INT0~2, Base Timer and TK.

Naming Rules for 92F Series Products

Name	SC	92	F	8	4	8	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, 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: 100pin
⑧	Version:(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)

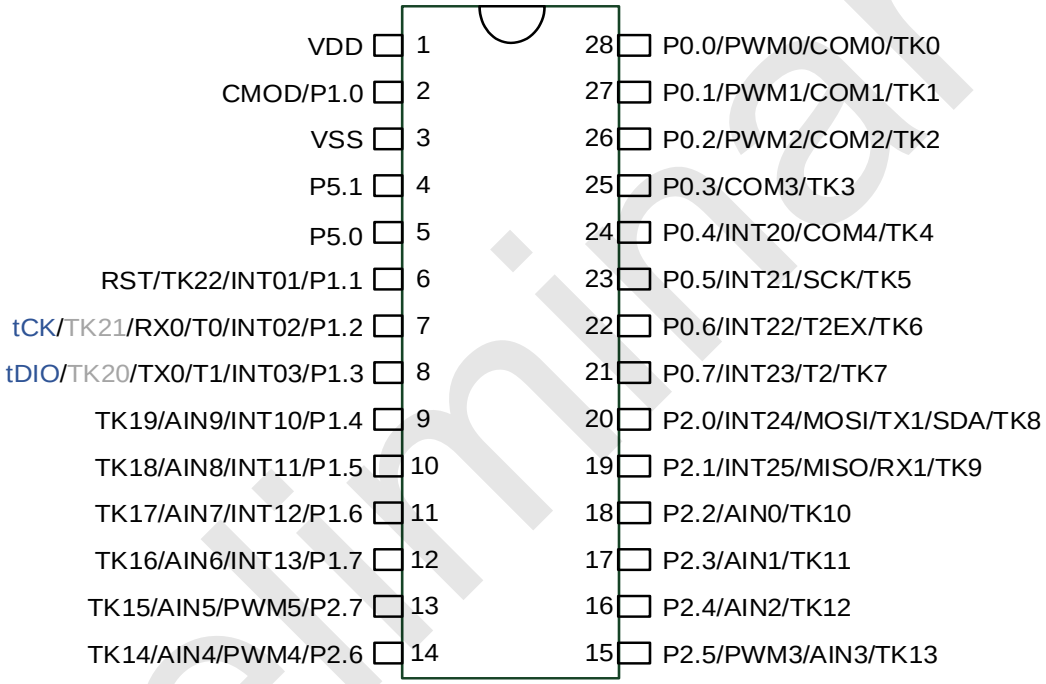
Contents

3 Pin Description

3.1 Pin Configuration

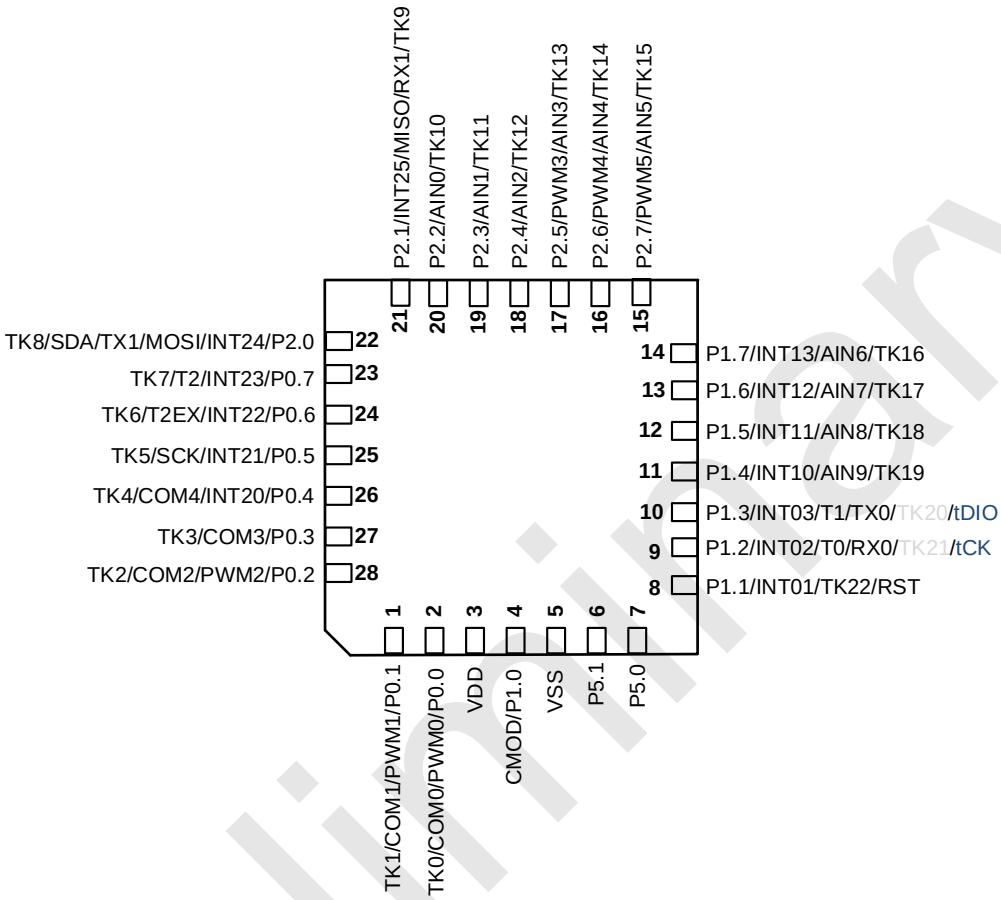
Note:

1. **In consideration of multiplexing of TK20/TK21 and TK debugging communication ports of the SC92F848X, if it is required to use the TK debugging function, please avoid using TK20/TK21!**
2. **SC92F848 series 8PIN package：SSI only supports UART function**



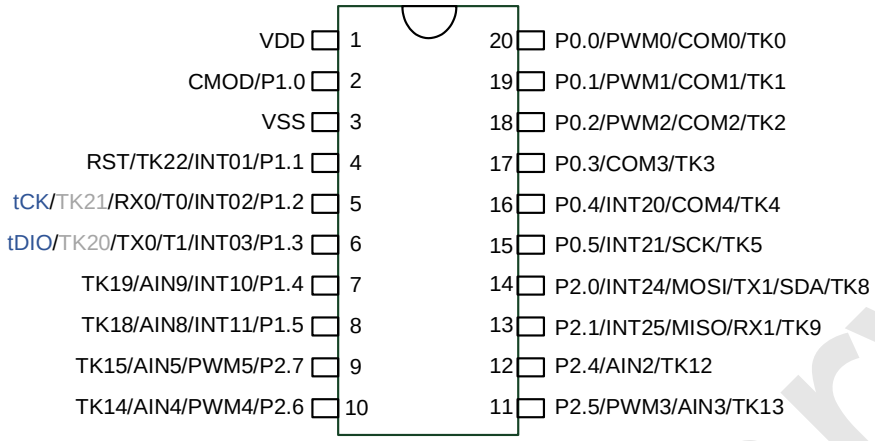
SC92F8483 Pin Diagram

Suitable for SOP28 & TSSOP28 package



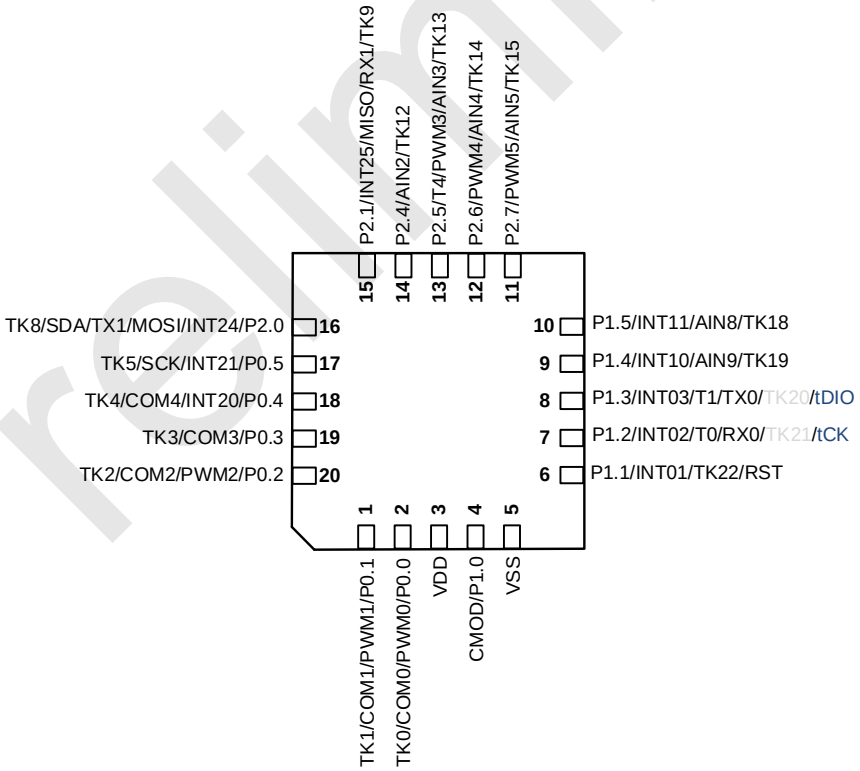
SC92F8483 Pin Diagram

Suitable for QFN28(4*4) package



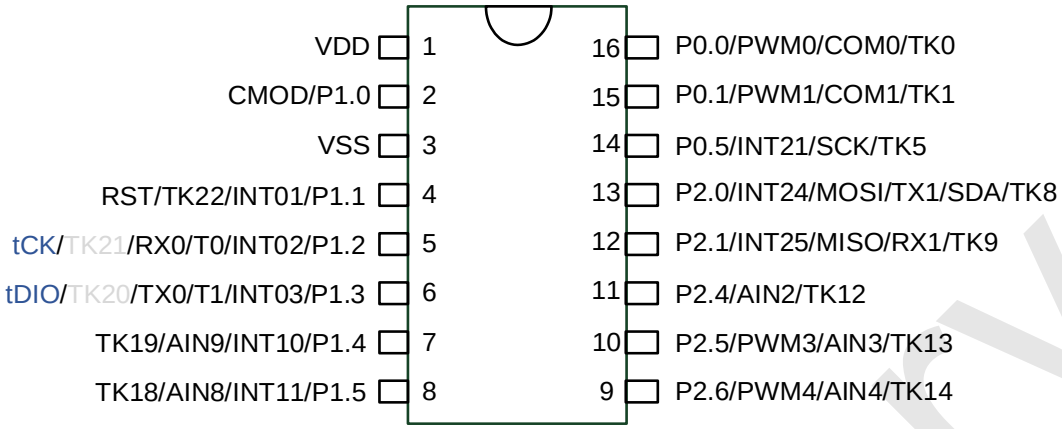
SC92F8482 Pin Diagram

Suitable for SOP20 & TSSOP20 package



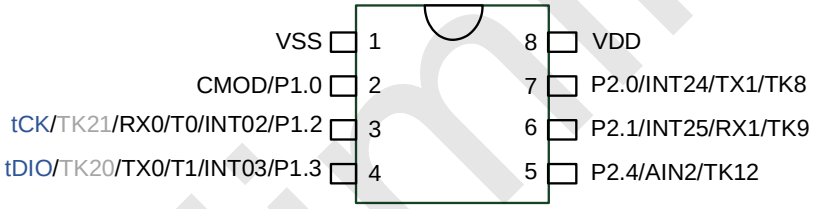
SC92F8482 Pin Diagram

Suitable for QFN20(3*3) package



SC92F8481 Pin Diagram

Suitable for SOP16 package



SC92F8480 Pin Diagram

Suitable for SOP8 package

3.2 Pin Definition

Package						Pin Name	Type	Description
SOP28& TSSOP28	SOP20& TSSOP20	SOP16	SOP 8	QFN 28	QFN 20			
1	1	1	8	3	3	**VDD**	Power	Power
2	2	2	2	4	4	**P1.0/CMOD**	I/O	P1.0: GPIO P1.0 CMOD: Touch Key external capacitance
3	3	3	1	5	5	**VSS**	Power	Ground
4	-	-	-	6	-	**P5.1**	I/O	P5.1: GPIO P5.1
5	-	-	-	7	-	**P5.0**	I/O	P5.0: GPIO P5.0
6	4	4	-	8	6	**P1.1/INT01/TK22/RST**	I/O	P1.1: GPIO P1.1 INT01: Input 1 of external interrupt 0 TK22: TK Channel 22

								RST: Reset pin
7	**5**	**5**	**3**	**9**	**7**	**P1.2/INT02/T0/RX0/TK21/tCK**	I/O	P1.2: GPIO P1.2 INT02: Input 2 of external interrupt 0 T0: Timer/Counter0 external input RX0: UART0 Receiver TK21: TK Channel 21, **if it is required to use the TK debugging function, please avoid using TK25!** tCK: Programming and Emulation Clock Pin
8	**6**	**6**	**4**	**10**	**8**	**P1.3/INT03/T1/TX0/TK20/tDIO**	I/O	P1.3: GPIO P1.3 INT03: Input 3 of external interrupt 0 T1: Timer/Counter1 external input TX0: UART0 Transmitter TK20: TK Channel 20, **if it is required to use the TK debugging function, please avoid using TK24!** tDIO: Programming and Emulation Data Pin
9	**7**	**7**	**-**	**11**	**9**	**P1.4/INT10/AIN9/TK19**	Power	P1.4: GPIO P1.4

								INT10: Input 0 of external interrupt 1 AIN9: ADC input channel 9 TK19: TK Channel 19
10	**8**	**8**	**-**	**12**	**10**	**P1.5/INT11/AIN8/TK18**	I/O	P1.5: GPIO P1.5 INT11: Input 1 of external interrupt 1 AIN8: ADC input channel 8 TK18: TK Channel 18
11	**-**	**-**	**-**	**13**	**-**	**P1.6/INT12/AIN7/TK17**	I/O	P1.6: GPIO P1.6 INT12: Input 2 of external interrupt 1 AIN7: ADC input channel 7 TK17: TK Channel 17
12	**-**	**-**	**-**	**14**	**-**	**P1.7/INT13/AIN6/TK16**	I/O	P1.7: GPIO P1.7 INT13: Input 3 of external interrupt 1 AIN6: ADC input channel 6 TK16: TK Channel 16

13	**9**	**-**	**-**	**15**	**11**	**P2.7/PWM5/AIN5/TK15**	I/O	P2.7: GPIO P2.7 PWM5: PWM5 Output AIN5: ADC input channel 5 TK15: TK Channel 15
14	**10**	**9**	**-**	**16**	**12**	**P2.6/PWM4/AIN4/TK14**	I/O	P2.6: GPIO P2.6 PWM4: PWM4 Output AIN4: ADC input channel 4 TK14: TK Channel 14
15	**11**	**10**	**-**	**17**	**13**	**P2.5/PWM3/AIN3/TK13**	I/O	P2.5: GPIO P2.5 PWM3: PWM3 Output AIN3: ADC input channel 3 TK13: TK Channel 13
16	**12**	**11**	**5**	**18**	**14**	**P2.4/AIN2/TK12**	I/O	P2.4: GPIO P2.4 AIN2: ADC input channel 2 TK12: TK Channel 12

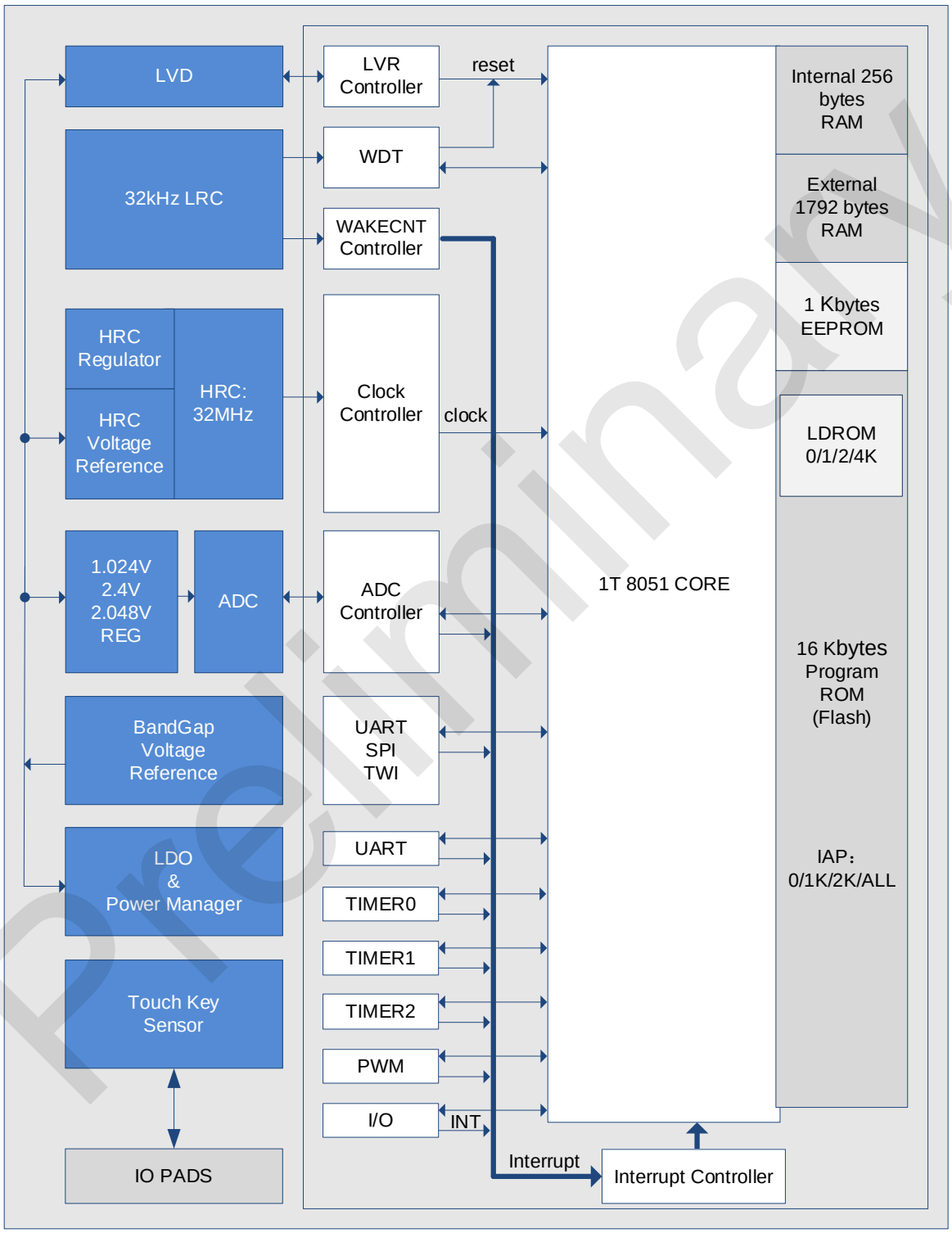
17	**-**	**-**	**-**	**19**	**-**	**P2.3/AIN1/TK11**	I/O	P2.3: GPIO P2.3 AIN1: ADC input channel 1 TK11: TK Channel 11
18	**-**	**-**	**-**	**20**	**-**	**P2.2/AIN0/TK10**	I/O	P2.2: GPIO P2.2 AIN0: ADC input channel 0 TK10: TK Channel 10
19	**13**	**12**	**6**	**21**	**15**	**P2.1/INT25/MISO/RX1/TK9**	I/O	P2.1: GPIO P2.1 INT25: Input 5 of external interrupt 2 MISO: Master-In and Slave-Out of SPI RX1: UART1 Receiver TK9: TK Channel 9
20	**14**	**13**	**7**	**22**	**16**	**P2.0/INT24/MOSI/TX1/SDA/TK8**	I/O	P2.0: GPIO P2.0 INT24: Input 4 of external interrupt 2 MOSI: Master-Out and Slave-In of SPI TX1: UART1 Transmitter

								SDA: SDA of TWI TK8: TK Channel 8
21	**-**	**-**	**-**	**23**	**-**	**P0.7/INT23/T2/TK7**	I/O	P0.7: GPIO P0.7 INT23: Input 3 of external interrupt 2 T2: Timer/Counter2 external input TK7: TK Channel 7
22	**-**	**-**	**-**	**24**	**-**	**P0.6/INT22/T2EX/TK6**	I/O	P0.6: GPIO P0.6 INT22: Input 2 of external interrupt 2 T2EX: External capture for Timer2 TK6: TK Channel 6
23	**15**	**14**	**-**	**25**	**17**	**P0.5/INT21/SCK/TK5**	I/O	P0.5: GPIO P0.5 INT21: Input 1 of external interrupt 2 SCK: SCK of SPI and TWI TK5: TK Channel 5
24	**16**	**-**	**-**	**26**	**18**	**P0.4/INT20/COM4/TK4**	I/O	P0.4: GPIO P0.4

								INT20: Input 0 of external interrupt 2 COM4: LCD common output 4 TK4: TK Channel 4
25	**17**	**-**	**-**	**27**	**19**	**P0.3/COM3/TK3**	I/O	P0.3: GPIO P0.3 COM3: LCD common output 3 TK3: TK Channel 3
26	**18**	**-**	**-**	**28**	**20**	**P0.2/PWM2/COM2/TK2**	I/O	P0.2: GPIO P0.2 PWM2: PWM2 Output COM2: LCD common output 2 TK2: TK Channel 2
27	**19**	**15**	**-**	**1**	**1**	**P0.1/PWM1/COM1/TK1**	I/O	P0.1: GPIO P0.1 PWM1: PWM1 Output COM1: LCD common output 1 TK1: TK Channel 1
28	**20**	**16**	**-**	**2**	**2**	**P0.0/PWM0/COM0/TK0**	I/O	P0.0: GPIO P0.0

								PWM0: PWM0 Output COM0: LCD common output 0 TK0: TK Channel 0

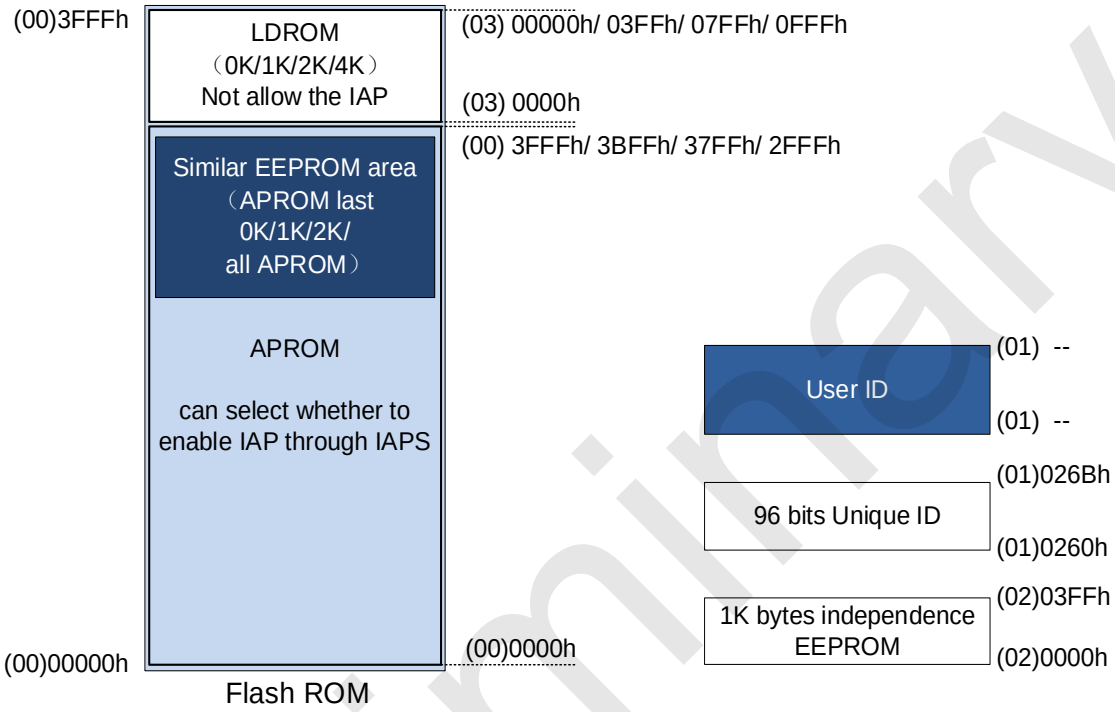
4 Inner Block Diagram



SC92F848X BLOCK DIAGRAM

5 Flash ROM and SRAM

The Flash ROM of SC92F848X is divided into five regions: APROM/LDROM/EEPROM/User ID/Unique ID, as shown in the following figure:



5.1 APROM and LDROM

APROM and LDROM are two independent pieces of hardware that divide ROM by LDSIZE[1:0]. They are distinguished by the extended address "00" and "03" set by IAPADE register. They can be programmed and erased by SC LINK PRO.

- The extended address of the APROM area is “00”. The choose of the area is 12~16 Kbytes. It supports IAP (In Application Programming) and APROM area allowed IAP operation in Flash can be set to 0K/1K/2K/All APROM by Code Option.
- The extended address of LDROM area is “03”, area size 0~4 Kbytes optional. IAP on LDROM is not allowed.

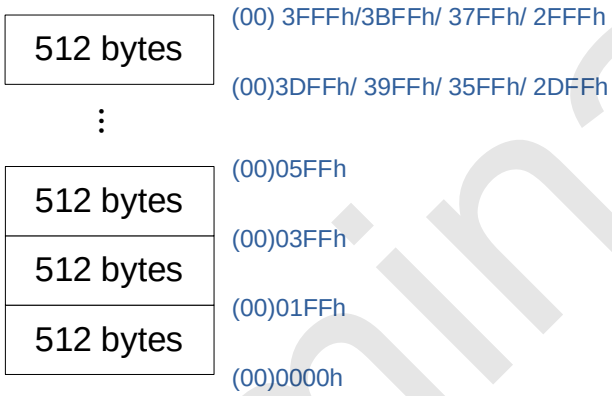
The SC92F848X has a 16 Kbytes Flash ROM. The address ranges from (00)0000H to (00)3FFFH. 00 in brackets is the extended address set by the IAPADE register. It can be programmed and erased by SC LINK PRO provided by Sinone. The 16 Kbytes Flash ROM features are as follows:

- Divided into 32 sectors, each 512 bytes;
- Can be repeatedly written for 100,000 times;
- Data can be stored for more than 100 years at 25 ° C;

- The ICP mode supports BLANK, PROGRAM, VERIFY, ERASE, and READ functions. The READ function is only valid for IC that do not have security encryption enabled;
- Secure encryption: Users can choose whether to enable APROM (16 Kbytes Flash ROM) and LDROM secure encryption;
- Support for IAP (In Application Programming).

5.1.1 Flash ROM Sector

The SC92F848X Flash ROM with 16 Kbytes was divided into 32 sectors, each sector 512 bytes. The sector to which the target address belongs was forcibly erased by the writer before data was written. When user write data, you must erase data before writing data.

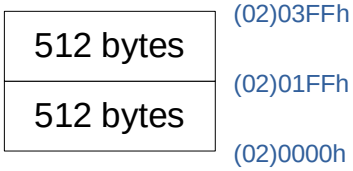


SC92F848X APROM Sector

5.2 1K bytes independent EEPROM

The SC92F848X has 1 Kbytes of independent EEPROM, the address is (02)000H ~ 03FFH, "02" in brackets is the extended address, which is set by the IAPADE register. Independent EEPROM can be rewritten 100,000 times and the data written-in has more than 100-year preservation life in the ambient temperature of 25℃ . EEPROM supports blank checking, programming, verification, erasing and reading functions.

EEPROM divided into 2 sectors,512 bytes per sector



SC92F848X EEPROM Sectors

Notes: EEPROM can be rewritten 100,000 times. User should not exceed the rated burn times of EEPROM, otherwise there will be exceptions!

5.3 96 bits Unique ID Area

96 bits Unique ID area. The address range is (01) 0260H to 026BH, where (01) is the extended address set by the IAPADE register. Stores the IC Unique ID. Users can read the IC Unique ID, but cannot write the Unique ID.

The SC92F848X provides an independent Unique ID area. A 96-bit unique code can be pre-programmed before leaving the factory to ensure the uniqueness of the chip. The only way for the user to obtain the serial number is to read the relative address (01)0260H~(01)026BH through the IAP instruction. The Unique ID range is (01)0260H ~ (01)026BH, the "01" in brackets indicates the extended address which is set by the IAPADE register. The specific operation method is as follows:

IAPADE (F4H) IAP Write to 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: Both MOVC and IAP are for APROM 0x01: For the unique ID area, **read but write operations are not allowed, otherwise it may cause an exception!** 0x02: Both MOVC and IAP are for EEPROM 0x03: Only valid for LDROM program operation. Programs in the LDROM region are now allowed to perform MOVC operations on the LDROM program region. **Note: LDROM operation permission is only for MOVC operation, IAP operation on LDROM is not valid, otherwise it will cause unpredictable exceptions!** Other: reserved

5.3.1 Unique ID Read Operating Demo Program In C Language

```
#include ”intrins.h”

unsigned char UniqueID [12];//store UniqueID
```

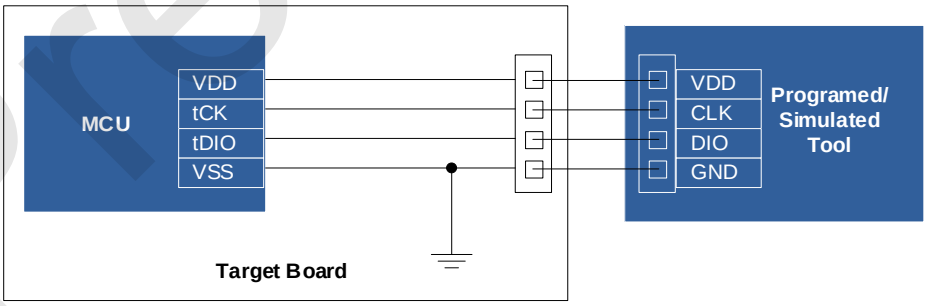
```
unsigned char code * POINT =0x0260;
unsigned char i;

EA = 0;                 // Disable the global interrupt
IAPADE = 0X01;                  // Expand address 0x01, select Unique ID area
for(i=0;i<12;i++)
{
    UniqueID [i]= *( POINT+i);      // Read the value of UniqueID
}
IAPADE = 0X00;                  // Expand address 0x00, return to Code area
EA = 1;                 // Enable global interrupt
```

5.4 User ID Area

User ID area, whose extended address is (01), is written for user in the factory. Users can read the User ID area, but cannot write the User ID area.

5.5 Programming

The SC92F848X's APROM, LDROM, EEPROM can be programmed through tDIO, tCK, VDD, VSS, the specific connection relationship is as follows:



ICP mode Flash Writer programming connection diagram

tDIO,tCK is a 2-wire JTAG programming and emulation signal line. Users can configure the mode of these two ports through the Code Option when programming:

5.5.1 JTAG Specific Mode

tDIO,tCK are specific port for programming and emulation, and other functions multiplexed with it are not available. This mode is generally used in the online debugging stage, which is convenient for users to simulate and debug. After the JTAG special mode takes effect, the chip can directly enter the programming or emulation mode without powering on and off again.

5.5.2 Normal Mode (JTAG specific port is invalid)

The JTAG function is not available, and other functions multiplexed with it can be used normally. This mode can prevent the programming port from occupying the MCU pins, which is convenient for users to maximize the use of MCU resources.

Note: When the invalid configuration setting of the JTAG dedicated port is successful, the chip must be completely powered off and then on again to enter the programming or emulation mode, which will affect the programming and emulation in the live mode. SinOne recommends that users select the invalid configuration of the JTAG dedicated port during mass production and programming, and select the JTAG mode during the development and debugging phase.

Code Option register:

OP_CTM1 (C2H@FFH) Code Option register1 (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	VREFS[1: 0]		OP_BL	DISJTG	IAPS[1: 0]		LDSIZE[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
POR	n	n	n	n	n	n	n	n

Bit number	Bit Mnemonic	Description
4	**DISJTG**	IO/JTAG port switching control 0 : JTAG mode is enabled, P1.2 and P1.3 can only be used as tCK/tDIO. Recommended settings during R&D and commissioning 1 : Normal mode (Normal), JTAG function is invalid. The recommended setting for the mass production burning stage.

5.6 In Application Programming (IAP)

Application Programming (IAP) operations can be carried out in the APROM of SC92F848X (0K, 1K, 2K, or all APROM ranges are optional) and 6K bytes EEPROM. Users can implement remote program updates through

IAP operations. You can also obtain Unique ID field or User ID field information via IAP reads. Before IAPS write data, you must erase the Sector to which the target address belongs. The length of a Sector is 512 bytes.

NOTE:

1. During the IAP erase/write process, the CPU holds the program counter, and after the IAP erase/write is complete, the program counter continues to execute subsequent instructions.

2. IAP operation in APROM area has certain risks, users need to take corresponding security measures in the software, if improper operation may cause user program rewriting! This feature is not recommended unless it is required by the user (for example, for remote application updates).

3. The EEPROM erasure count is 100,000. Do not exceed the rated EEPROM erasure count; otherwise, an exception may occur.

The user can select the IAP region range of APROM through Customer Option during programming, or set the IAP region of APROM through IAPS control bit in the program. The relevant registers are as follows:

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

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

Bit Number	Bit Mnemonic	Description
3~2	**IAPS[1: 0]**	IAP spatial range selection 00: Full Flash ROM not allows IAP operation 01: Last 1K Flash ROM allows IAP operation 10: Last 2K Flash ROM allows IAP operation 11: Full Flash ROM allows IAP operation Note:

		1.The above setting items are invalid in BootLoader mode. The BootLoader program can perform IAP operation on the entire Flash ROM area. 2. LDROM does not allow IAP operation under any circumstances.

5.6.1 IAP Operation Related Register

Symbol	Address	Description	7	6	5	4	3	2	1	0	POR
IAPKEY	F1H	Data 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 to extended address register	IAPADER[7: 0]								00000000b
IAPDAT	F5H	IAP data register	IAPDAT[7: 0]								00000000b
IAPCTL	F6H	IAP control register	BTLD	-	SERASE	PRG	-	-	CMD[1: 0]		0x00xx00b

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]**	Open IAP function and operation time limit setting Write a non-zero value n, representing: 1.Enable the IAP function; 2.If no IAP write command is received after n system clocks, the IAP function is turned off again.

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]**	IAP writes the low 8 bits of the address

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]**	IAP writes the high 6 bits of the address

IAPADE (F4H) IAP Write to 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: Both MOVC and IAP are for APROM 0x01: The Unique ID area is read but cannot be written 0x02: Both MOVC and IAP are for independent EEPROM 0x03: Only valid for LDROM program operation. Programs in the LDROM region are now allowed to perform MOVC operations on the LDROM program region. **Note: LDROM operation permission is only for MOVC operation, IAP operation on LDROM is not valid, otherwise it will cause unpredictable exceptions!**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	R/W	Description
7~0	**IAPDAT[7:0]**	R/W	Data written by IAP

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

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

Bit number	Bit Mnemonic	Description
7	**BTLD**	BootLoader control bit 0: software reset after the program from the main area (main program), began to run; 1: the software reset after the program from the BootLoader area began to run
5	**SERASE**	Sector Erase control bit 0: no operation

		1: place a "1" and then configure CMD (1-0) = 10, then enter the Flash ROM Sector Erase operation, Flash ROM specified Sector will be erased
4	**PRG**	Program (the Program) control bit 0: no operation 1: place a "1" and then configure CMD (1-0) = 10, Flash ROM to write operation, IAPDAT register data will be written to Flash ROM address specified
1~0	**CMD[1:0]**	IAP command can control bit 10: write or sector erase operation command Other：reservations NOTE: 1. After SERASE/PRG is set to 1, you must set CMD[1:0]=10 to start the corresponding operation 2. Only one IAP action can be performed at a time, so SERASE/PRG can only have one position 1 at a time 3. The IAP operation statement must be followed by at least 8 NOP instructions to ensure the normal execution of subsequent instructions after the IAP operation is completed

5.6.2 IAP Operation process

The IAP writing process for SC92F848X is as follows:

1. Write IAPADER[7:0]:
   - =0x00, perform IAP operation on APROM region
   - =0x01, read the Unique ID field, **Note: this can only be read, not rewritten!**
   - =0x02, reads and writes the EEPROM area
   - =0x03, read the LDROM area .

   Note: MOVC only, not IAP, this is only valid for LDROM program operation, APROM program operation this is not valid!
2. Write IAPDAT[7:0], ready for IAP to write data;
3. Write IAPADR [13:0], prepare the target address of IAP operation;
4. Write IAPKEY[7:0] a non-zero value n, turn on IAP protection, and IAP operation will be closed if no write command is received within n system clocks;
5. After the IAP writes, the CPU continues.

5.6.3 Notes for IAP operation

1. When programming IC, if "APROM zone prevents IAP operation" is selected through Code Option, then IAPADE[7:0]=0x00 (APROM zone is selected), IAP cannot be operated, that is, data cannot be written, data can only be read by MOVC instruction.
2. When IAPADE not 0 x00 MOVC and writing is not aimed at APROM area, at this time if there is interruption, and interrupt with MOVC operation, can cause MOVC result error, cause the program to run. To avoid this situation, if IAPADE is not 0x00 during IAP operation, it is important to turn off total interrupt (EA=0) before operation, and set IAPADE = 0x00 after operation before turning on total interrupt (EA=1).
3. IAP wipe/writing process, the CPU program counter, IAP wipe/write, after the completion of the program counter to continue after the instruction;
4. APROM area of IAP operation has certain risk, requires the user to do the corresponding safety measures in the software, if the improper operation may cause the user program has changed! This feature is not recommended unless it is required by the user (for example, for remote application updates).
5. EEPROM to wipe the number for 100000 times, user wipe block don't exceed the EEPROM rating of the burning times, otherwise will be abnormal!

5.6.4 IAP Operating Demo Program In C Language

The header files shared by the following routines are as follows:

```
#include ”intrins.h”

unsigned int IAP_Add;

unsigned char IAP_Data;

unsigned char code * POINT =0x00;
```

IAP operation: sector erase:

```
EA = 0; / / close the interrupt

IAPADE = 0 x00; / / expand address 0 x00, select Flash ROM

IAPADH = (unsigned char) ((IAP_Add > > 8)); / / write IAP target address

IAPADL = high value (unsigned char) IAP_Add; / / write IAP target address

IAPKEY = 0 xf0 low value;

IAPCTL = 0 x20; / / set up the sector erase bit

IAPCTL | = 0 x02; / / execution block wipe

_nop_ (); / / wait (need at least eight _nop_ ())

_nop_ ();

_nop_ ();

_nop_ ();

_nop_ ();
```

```
_nop_ ();
_nop_ ();
 _nop_ ();
_nop_ ();
EA = 1; // Enable total interrupt
```

IAP operation: write data:

```
EA = 0; / / close the interrupt
IAPADE = 0 x00; / / expand address 0 x00, select Flash ROM
IAPDAT = IAP_Data; / / send data to the IAP data register
IAPADH = (unsigned char) ((IAP_Add > > 8)); / / write IAP target address
IAPADL = high value (unsigned char) IAP_Add; / / write IAP target address
IAPKEY = 0 xf0 low value; // This value can be adjusted according to actual situation; After the need to ensure this instruction execution to IAPCTL before assigning, / / the time interval should be less than 240 (0 xf0) a system clock, otherwise the IAP functions closed; Be especially careful when / / open the interrupt
IAPCTL = 0 x10; // Set the IAP write operation bit.
IAPCTL | = 0 x02; / / execution _nop_ written instructions (); / / wait (need at least eight _nop_ ())
_nop_ ();
_nop_ ();
_nop_ ();
_nop_ ();
_nop_ ();
_nop_ ();
_nop_ ();
EA = 1; // Enable total interrupt
```

IAP operation: read data:

```
EA = 0;                 // Close global interrupt
IAPADE = 0X00;                  //The extended address is 0x00, selectFlash ROM
IAP_Data = *( POINT+IAP_Add);   //Read the value of IAP_Add toIAP_Data
EA = 1;                 // Open global interrupt
```

5.7 BootLoader

The LDROM is used to store the bootLoader code. LDROM supports blank checking (BLANK), programming (PROGRAM), verifying (VERIFY), erasing (ERASE) and reading (READ) functions in ICP mode.

Users can realize ISP (In System Programing) function through LDROM: when ISP is executed, IC runs the boot code In LDROM area. When the boot code is executed, IC receives new program code through serial port, and then programs the received code into user code area through IAP command.

The LDROM has four address ranges:

- (03)0000H~(03)0000H（without LDROM）
- (03)0000H~(03)03FFH（1K）
- (03)0000H~(03)07FFH（2K）
- (03)0000H~(03)0FFFH（4K）

Where: "03" in the brackets above indicates the extended address, which is set by LDSIZE [1:0].

5.7.1 BootLoader Mode operation related registers

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

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	VREFS[1: 0]		OP_BL	DISJTG	IAPS[1: 0]		LDSIZE [1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
POR	n	n	n	n	n	n	n	n

Bit Number	Bit Mnemonic	Description
5	**OP_BL**	Program run area control bit 0: After the chip is reset, it enters APROM

<table>
<tr><td></td><td></td><td>1: After the chip is reset, it enters LDROM

① The MOVC and IAP restrictions for APROM are as follows:
<table><tr><td>Operation</td><td>Is it operable?</td></tr><tr><td>LDROM MOVC</td><td>x</td></tr><tr><td>APROM MOVC</td><td>Only the first 256 bytes cannot be MOVC</td></tr><tr><td>LDROM IAP</td><td>×</td></tr><tr><td>APROM IAP</td><td>√</td></tr></table>
② The MOVC and IAP restrictions for LDROM are as follows:
<table><tr><td>Operation</td><td>Is it operable?</td></tr><tr><td>LDROM MOVC</td><td>√</td></tr><tr><td>APROM MOVC</td><td>√</td></tr><tr><td>LDROM IAP</td><td>×</td></tr><tr><td>ALL APROM IAP, not restricted by IAPRANGE</td><td>√</td></tr></table></td></tr>
<tr><td>1~0</td><td>LDSIZE [1:0]</td><td>LDROM space range selection

00: None LDROM, the APROM address is 0000H~3FFFH

01: The last 1K APROM area of the Flash ROM is LDROM, and the APROM address is 0000H~3BFFH

10: The last 2K APROM area of the Flash ROM is LDROM, and the APROM address is 0000H~37FFH

11: The last 4K APROM area of the Flash ROM is LDROM, and the APROM address is 0000H~2FFFH

NOTE: LDROM not allow IAP operation in anyways</td></tr>
</table>

IAPKEY (F1H) Data 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]**	Open IAP and operation time limit setting Write a value n greater than or equal to 0x40, which represents: 1. Enable the IAP; 2. If no IAP write command is received after n system clocks, the IAP is turned off again.

IAPADL (F2H) IAP Write Low Address 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]**	IAP writes the low 8 bits of the address

IAPADH (F3H) IAP Write High Address 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]**	IAP writes the high 6 bits of the address

IAPADE (F4H) IAP Write to 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: Both MOVC and IAP are for Flash ROM 0x01: The Unique ID area is read but cannot be written 0x02: Both MOVC and IAP are for independent EEPROM

		0x03: Only valid for LDROM program operation. Programs in the LDROM region are now allowed to perform MOVC operations on the LDROM program region. **Note: LDROM operation permission is only for MOVC operation, IAP operation on LDROM is not valid, otherwise it will cause unpredictable exceptions!** 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	BTLD	-	SERASE	PRG	-	-	CMD[1: 0]	
R/W	R/W	-	R/W	R/W	-	-	R/W	R/W
POR	0	x	0	0	x	x	0	0

Bit number	Bit Mnemonic	Description
7	**BTLD**	BootLoader control bit 0: software reset after the program from the main area (main program), began to run; 1: the software reset after the program from the BootLoader area began to run
5	**SERASE**	Sector Erase control bit 0: no operation 1: place a "1" and then configure CMD (1-0) = 10, then enter the Flash ROM Sector Erase operation, Flash ROM specified Sector will be erased
4	**PRG**	Program (the Program) control bit 0: no operation 1: place a "1" and then configure CMD (1-0) = 10, Flash ROM to write operation, IAPDAT register data will be written to Flash a ROM address specified
1~0	**CMD[1:0]**	IAP command can control bit 10: write or sector erase operation command Other：reservations NOTE: 4. After SERASE/PRG is set to 1, you must set CMD[1:0]=10 to start the corresponding operation 5. Only one IAP action can be performed at a time, so SERASE/PRG can only have one position 1 at a time The IAP operation statement must be followed by at least 8 NOP instructions to ensure the normal execution of subsequent instructions after the IAP operation is completed

PCON (87h) Power Management Control Register (write only, *not readable*)

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

Bit number	Bit Mnemonic	Description
3	**RST**	Software reset control bit: Write status: 0: The program runs normally; 1: The CPU resets immediately after this bit is written to "1"

Bootloader Notes:

1. The user must erase the target sector before writing LDROM;
2. For the specific operation method, please refer to the description document " SinOne hardware BootLoader Function Implementation Application Guide" provided by SinOne.

5.8 Encryption

Users can choose whether to encrypt the SC92F848X's ROM through the settings on the computer program:

1. If the encryption function is disabled, users can read the last data written in APROM and LDROM by SC LINK;
2. If the encryption function is enabled, the data written in APROM (16 Kbytes Flash ROM) and LDROM will never be read from outside. It is recommended to enable the encryption function during mass production;
3. The only way to release security encryption is to re-programming
4. The encryption has no effect on iap read and write operation

For the specific operation method, please refer to the chapter of Secure Encryption and Reading in the "SOC LINK Series Programmer & Simulator User Manual".

5.9 Code Option Area (User Programming Settings)

SC92F848X has a separate Flash area for storing the initial power-on Settings of the customer, which is called the Code Option area. This part of the setting data is written into the IC when the user programs the IC, and this setting data will be called into the SFR as the initial setting when the IC is reset and initialized.

Option related SFR operation instructions:

OPINX and OPREG registers control the read and write operations of option-related SFR. OPINX determines the specific positions of each Option SFR, as shown in the following table:

Symbol	OPINX Address	Instructions	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
OP_CTM0	0xC1	Code Option register 0	ENWDT	-	SCLKS[1: 0]		DISRST	DISLVR	LVRS[1: 0]	
OP_CTM1	0xC2	Code Option register 1	VREFS[1: 0]		OP_BL	DISJTG	IAPS[1: 0]		LDSIZE[1:0]	
OP_HRCR	0x83	System clock change register	OP_HRCR[7: 0]							

IFB Address	Symbol	Write read	Instructions
OP_CTM0[7]	**ENWDT**	**R/W**	**WDT Switch** 0：WDT invalid 1：WDT valid
OP_CTM0[5~4]	**SCLKS [1:0]**	**R/W**	**System clock frequency selection bits** 00: System clock frequency is HRC frequency divided by 1; 01: System clock frequency is HRC frequency divided by 2; 10: System clock frequency is HRC frequency divided by 4; 11: System clock frequency is HRC frequency divided by12;

<table>
<tr><td>OP_CTM0[3]</td><td>DISRST</td><td>Read only</td><td>IO/RST Selection bit
0: configure P1.1 as External Reset input pin
1: configure P1.1 as GPIO</td></tr>
<tr><td>OP_CTM0[2]</td><td>DISLVR</td><td>R/W</td><td>LVR control bit
0: LVR valid
1: LVR invalid</td></tr>
<tr><td>OP_CTM0[1~0]</td><td>LVRS [1:0]</td><td>R/W</td><td>LVR voltage selection control
11: 4.3V reset
10: 3.7V reset
01: 2.9V reset
00: 1.9V reset</td></tr>
<tr><td>OP_CTM1[7~6]</td><td>VREFS [1:0]</td><td>R/W</td><td>Reference voltage selection
00: Configure ADC VREF as VDD;
01: Configure ADC VREF as internal 1.024V
10: Configure ADC VREF as internal 2.4V
11: Configure ADC VREF as internal 2.048</td></tr>
<tr><td>OP_CTM1[5]</td><td>OP_BL</td><td>R/W</td><td>Program run area control bit
0: After the chip is reset, it enters APROM
1: After the chip is reset, it enters LDROM

1. The MOVC and IAP restrictions for APROM are as follows:
<table><tr><td>Operation</td><td>Is it operable?</td></tr><tr><td>LDROM MOVC</td><td>x</td></tr><tr><td>APROM MOVC</td><td>Only the first 256 bytes cannot be MOVC</td></tr></table></td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td>LDROM IAP</td><td>×</td></tr>
<tr><td>APROM IAP</td><td>√</td></tr>
</table>
2. The MOVC and IAP restrictions for LDROM are as follows:
<table>
<tr><td>Operation</td><td>Is it operable?</td></tr>
<tr><td>LDROM MOVC</td><td>√</td></tr>
<tr><td>APROM MOVC</td><td>√</td></tr>
<tr><td>LDROM IAP</td><td>×</td></tr>
<tr><td>ALL APROM IAP, not restricted by IAPRANGE</td><td>√</td></tr>
</table>
</td></tr>
<tr><td>OP_CTM1[4]</td><td>DISJTG</td><td>R/W</td><td>IO/JTAG Port switching control
0: JTAG mode is enabled, P1.2 and P1.3 can only be used as tCK/tDIO.
1: Normal mode (Normal), JTAG function is invalid.</td></tr>
<tr><td>OP_CTM1[3~2]</td><td>IAPS[1:0]</td><td>R/W</td><td>IAP spatial range selection
00: All Flash ROM not allows IAP operation
01: Last 1K Flash ROM allows IAP operation
10: Last 2K Flash ROM allows IAP operation
11: All Flash ROM allows IAP operation
Note：
1. The preceding Settings are invalid in BootLoader mode, and the BootLoader program can perform IAP operations on the entire APROM region
2. LDROM does not allows IAP operation</td></tr>
<tr><td>OP_CTM1[1:0]</td><td>LDSIZE[1:0]</td><td>Read only</td><td>LDROM space range selection
00: None LDROM, the APROM address is 0000H to 3FFFH</td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td>01: The last 1K APROM area of the Flash ROM is LDROM, and the APROM address is 0000H-3BFFh

10: The last 2K APROM area of the Flash ROM is LDROM, and the APROM address is 0000H-37FFh

11: The last 4K APROM area of the Flash ROM is LDROM, and the APROM address is 0000H-2FFFh

NOTE: LDROM not allow IAP operation in anyways</td></tr>
<tr><td>OP_HRCR</td><td>OP_HRCR[7:0]</td><td>R/W</td><td>HRC frequency change register

User can change the high-frequency oscillator frequency f_{HRC} by modifying the value of this register, and then change the system clock frequency fsys:

1. The initial value of OP_HRCR[7: 0] after power-on OP_HRCR[s] is a fixed value to ensure that f_{HRC} is 32MHz, OP_HRCR[s] of each IC may be different

2. When the initial value is OP_HRCR[s], the system clock frequency fsys of the IC can be set to an accurate 32/16/8/2.66MHz through the Option item. When OP_HRCR [7: 0] changes by 1, the fsys frequency changes by about 0.18%

The relationship between OP_HRCR [7: 0] and fsys output frequency is as follows:
<table>
<tr><td>OP_HRCR [7: 0] Value</td><td>fsys actual output frequency (32M as an example)</td></tr>
<tr><td>OP_HRCR [s]-n</td><td>32000*(1-0.18%*n)kHz</td></tr>
<tr><td>...</td><td>....</td></tr>
<tr><td>OP_HRCR [s]-2</td><td>32000*(1-0.18%*2) = 31 884.8kHz</td></tr>
<tr><td>OP_HRCR [s]-1</td><td>32000*(1-0.18%*1) = 31 942.4kHz</td></tr>
<tr><td>OP_HRCR [s]</td><td>32000kHz</td></tr>
<tr><td>OP_HRCR [s]+1</td><td>32000*(1+0.18%*1) = 32 057.6kHz</td></tr>
<tr><td>OP_HRCR [s]+2</td><td>32000*(1+0.18%*2) = 32 115.2kHz</td></tr>
</table></td></tr>
</table>

			...	...
			OP_HRCR [s]+n	32000*(1+0.18%*n)kHz

5.9.1 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	Instructions		POR
OPINX	FEH	Option pointer	OPINX[7: 0]	00000000b
OPREG	FFH	Option register	OPREG[7: 0]	nnnnnnnnb

The OPINX register stores the address of the related OPTION register when operating the Option related SFR, and the OPREG register stores the corresponding value.

For example: To set ENWDT (OP_CTM0.7) to 1, the specific operation method is as follows:

C language example:

```
OPINX = 0xC1;          // Write the address of OP_CTM0 to the OPINX register
OPREG |= 0x80;         // Set 1 for OP_CTM0.7
```

Assembly language example:

```
MOV OPINX,#C1H         ; Write the address of OP_CTM0 to the OPINX register
ORL OPREG,#80H         ; Set 1 for OP_CTM0.7
```

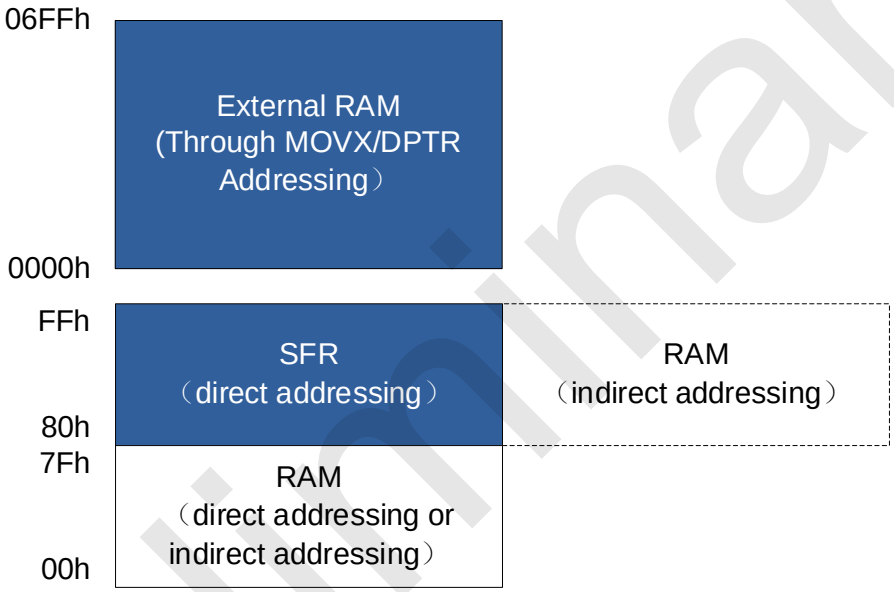
Note: It is forbidden to write any value beyond SFR address of Customer Option region into OPINX register! Or else, it may cause abnormal system operation.

5.10 SRAM

The SRAM of SC92F848X microcontroller, It is divided into internal 256 bytes RAM, external 1792 bytes RAM. The address range of the internal RAM is 00H to FFH, where high 128 bytes (address 80H to FFH) can only be addressed indirectly, and low 128 bytes (address 00H to 7FH) can be addressed directly or indirectly. Special function register address is 80 h ~ FFH SFR.

However, the SFR differs from the internal high 128 bytes SRAM in that the SFR registers are addressed directly, while the internal high 128 bytes SRAM can only be addressed indirectly.

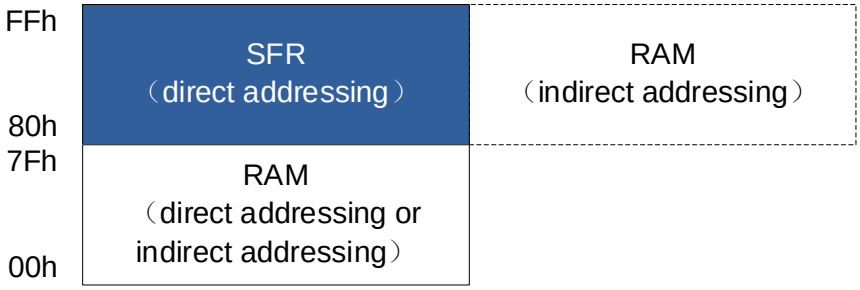
External RAM address is 0000 H ~ 06FFH, but need to be addressed through the MOVX instructions.



SRAM structure of the SC92F848X

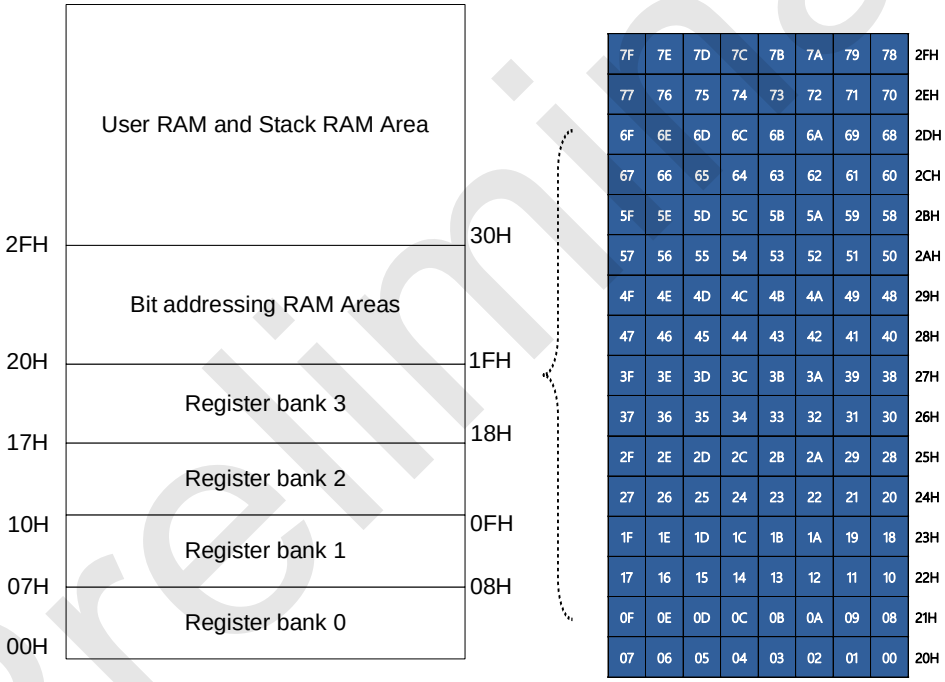
5.10.1 Internal 256 Bytes SRAM

The internal low 128 bytes SRAM area can be divided into three parts: ① Working register groups 0~3, address 00H~1FH, the combination of RS0 and RS1 in the program state word register PSW determines the current working register group, using working register groups 0~3 can speed up the operation speed; ② Bit addressable area 20H~2FH, this area can be used as normal RAM or as bit-addressable RAM; When addressing by bit, the bit address is 00H~7FH, (this address is encoded by bit, different from the general SRAM encoded by byte address), the program can be distinguished by instruction; ③ User RAM and stack area, SC92F848X after reset, the 8-bit stack pointer points to the stack area, Users generally set the initial value during initialization. It is recommended to set the initial value between E0H and FFH..



Internal 256 bytes RAM structure diagram

The internal low 128 bytes RAM structure is as follows:



SRAM structure diagram

5.10.2 External 1792 bytes SRAM

External 1792 bytes RAM can be accessed through MOVX @DPTR, A; you can also use MOVX A, @Ri or MOVX @Ri, A with EXADH register to access external 1792 bytes RAM: EXADH register stores the high address of external SRAM, Ri register stores the low 8 bits of the external SRAM.

EXADH (F7H) External SRAM Operation Address High Bit (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	-	-	-	-	-	EXADH [2: 0]		
POR	x	x	x	x	x	0	0	0

Bit number	Bit Mnemonic	Description
2~0	**EXADH [2: 0]**	High-bit of external SRAM operation address
7~3	**-**	reserved

6 Special Function Register (SFR)

6.1 SFR Mapping

The SC92F848X provides some registers equipped with special functions, called SFR. The addresses of these registers are located at 80H~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 SC92F848X SFR Map is as follows:

	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	-	PWMDTYB	PWMDTY3	PWMDTY4	PWMDTY5
D0h	PSW	PWMCFG	PWMCON	PWMPRD	PWMDTYA	PWMDTY0	PWMDTY1	PWMDTY2
C8h	T2CON	T2MOD	RCAP2L	RCAP2H	TL2	TH2	BTMCON	WDTCON
C0h	-	-	-	-	-	-	INT2F	INT2R
B8h	IP	IP1	INT0F	INT0R	INT1F	INT1R	-	-
B0h	-	-	-	-	-	-	-	-
A8h	IE	IE1	ADCCFG2	ADCCFG0	ADCCFG1	ADCCON	ADCVL	ADCVH
A0h	P2	P2CON	P2PH	-	-	-	-	-
98h	SCON	SBUF	P0CON	P0PH	P0VO	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	Non-bit addressable						

Note:

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

6.2 SFR Instructions

6.2.1 SFR

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	-	-	-	-	-	T2FD	T1FD	T0FD	xxxxx000b
OTCON	8FH	Output Control Register	SSMOD[1: 0]		-	-	VOIRS[1: 0]		-	-	00xx00xxb
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
P0VO	9CH	P0 Port LCD Voltage Output Register	-	-	-	P04VO	P03VO	P02VO	P01VO	P00VO	xxx00000b
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	SSDAT[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	-	-	-	ETK	EINT2	EBTM	EPWM	ESSI	xxx00000b
ADCCFG2	AAH	ADC Configuration Register 2	-	-	-	-	LOWSP	ADCCK[2: 0]			xxxx0000b
ADCCFG0	ABH	ADC Configuration Register 0	EAIN7	EAIN6	EAIN5	EAIN4	EAIN3	EAIN2	EAIN1	EAIN0	00000000b
ADCCFG1	ACH	ADC Configuration Register 1	-	-	-	-	-	-	EAIN9	EAIN8	xxxxxx00b
ADCCON	ADH	ADC Control Register	ADCEN	ADCS	EOC/ ADCIF	ADCIS[4: 0]					00000000b
ADCVL	AEH	ADC Result Register	ADCV[3: 0]				-	-	-	-	1111xxxxb
ADCVH	AFH	ADC Result Register	ADCV[11: 4]								11111111b
IP	B8H	Interrupt Priority Control Register	-	IPADC	IPT2	IPUART	IPT1	IPINT1	IPT0	IPINT0	x0000000b
IP1	B9H	Interrupt Priority Control Register 1	-	-	-	IPTK	IPINT2	IPBTM	IPPWM	IPSSI	xxx00000b
INT0F	BAH	INT0 Falling Edge Interrupt Control Register	-	-	-	-	INT0F3	INT0F2	INT0F1	-	xxxx000xb
INT0R	BBH	INT0 Rising Edge Interrupt Control Register	-	-	-	-	INT0R3	INT0R2	INT0R1	-	xxxx000xb
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
INT2F	C6H	INT2 Falling Edge Interrupt Control Register	-	-	INT2F5	INT2F4	INT2F3	INT2F2	INT2F1	INT2F0	xx000000b
INT2R	C7H	INT2 Rising Edge Interrupt Control Register	-	-	INT2R5	INT2R4	INT2R3	INT2R2	INT2R1	INT2R0	xx000000b

T2CON	C8H	Timer2 Control Register	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2	00000000b
T2MOD	C9H	Timer2 Operating Mode Register	-	-	-	-	-	-	T2OE	DCEN	xxxxxx00b
RCAP2L	CAH	Timer2 Reload Low 8 bits	RCAP2L[7: 0]								00000000b
RCAP2H	CBH	Timer2 Reload High 8 bits	RCAP2H[7: 0]								00000000b
TL2	CCH	Timer2 Low 8 bits	TL2[7: 0]								00000000b
TH2	CDH	Timer2 High 8 bits	TH2[7: 0]								00000000b
BTMCON	CEH	Low-Frequency Timer Control Register	ENBTM	BTMIF	-	-	BTMFS[3: 0]				00xx0000b
WDTCON	CFH	WDT Control Register	-	-	-	CLRWDT	-	WDTCKS[2: 0]			xxx0x000b
PSW	D0H	Program Status Word Register	CY	AC	F0	RS1	RS0	OV	F1	P	00000000b
PWMCFG	D1H	PWM Setup Register	PWMCKS[1: 0]		INV5	INV4	INV3	INV2	INV1	INV0	00000000b
PWMCON	D2H	PWM Control Register	ENPWM	PWMIF	ENPWM5	ENPWM4	ENPWM3	ENPWM2	ENPWM1	ENPWM0	00000000b
PWMPRD	D3H	PWM Period Setting Register	PWMPRD[9: 2]								00000000b
PWMDTYA	D4H	PWM0 duty cycle setting register A	PWMPRD[1: 0]		PDT2[1: 0]		PDT1[1: 0]		PDT0[1: 0]		00000000b
PWMDTY0	D5H	PWM0 duty cycle setting register	PDT0[9: 2]								00000000b
PWMDTY1	D6H	PWM1 duty cycle setting register	PDT1[9: 2]								00000000b
PWMDTY2	D7H	PWM2 duty cycle setting register	PDT2[9: 2]								00000000b
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
PWMDTYB	DCH	PWM duty cycle setting register B	PWMMOD	-	PDT5[1: 0]		PDT4[1: 0]		PDT3[1: 0]		0x000000b
PWMDTY3	DDH	PWM3 duty cycle setting register / PWM Dead-time Configuration Register	PDT3[9: 2]								00000000b
PWMDTY4	DEH	PWM4 duty cycle setting register	PDT4[9: 2]								00000000b
PWMDTY5	DFH	PWM5 duty cycle setting register	PDT5[9: 2]								00000000b
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 0	EXB [7: 0]								00000000b
EXBH	EEH	Extended B Register 1	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	BTLD	-	SERASE	PRG	-	-	CMD[1: 0]	0x00xx00b
EXADH	F7H	External SRAM Operating Address High	-	-	-	-	-	EXADH [2: 0]		xxxxx000b
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

6.2.2 Introduction of Common Special Function Registers of 8051 Core

Program Counter PC

The program counter PC does not belong to the SFR register. The PC has 16 bits and is a register used to control the order of execution of instructions. After the MCU is powered on or reset, the PC value is 0000H, which means that the MCU program starts executing the program from the 0000H address.

Accumulator ACC (E0H)

The accumulator ACC is one of the most commonly used registers of the 8051 core single-chip microcomputer, and A is used as a mnemonic in the instruction set. Commonly used to store operands and results that participate in calculations or logical operations.

B Register (F0H)

The B register must be used with the accumulator A in multiplication and division operations. The multiplication instruction MUL A, B multiplies the 8-bit unsigned number in accumulator A and register B. The low-bit byte of the resulting 16-bit product is placed in A, and the high-bit byte is placed in B. The division instruction DIV A, B divides A by B, the integer quotient is placed in A, and the remainder is placed in B. Register B can also be used as a general temporary storage register.

Stack Pointer SP (81H)

The stack pointer is an 8-bit special register that indicates the location of the top of the stack in general-purpose RAM. After the Microcontroller unit (MCU) is reset, the initial value of SP is 07H, that is, the stack will increase upward from 08H. 08H~1FH is operating register group 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

Bit number	Bit Mnemonic	Description
7	**CY**	Flag 1: When there is a carry in the highest bit of addition, or a borrow in the highest bit of subtraction 0: When there is no carry in the highest bit of addition, or there is no borrow in the highest bit of subtraction
6	**AC**	Carry auxiliary flag (can be easily adjusted during the addition and subtraction of BCD code) 1: When the addition operation has a carry in bit3, or the subtraction operation has a borrow in bit3 0: No borrowing, carry
5	**F0**	User flag
4~3	**RS1,RS0**	Operating register group selection bits: RS1 / RS0 / Operating register set currently in use 0~3 0 / 0 / TEAM 0 (00H~07H) 0 / 1 / TEAM 1 (08H~0FH) 1 / 0 / TEAM 2 (10H~17H) 1 / 1 / TEAM 3 (18H~1FH)

2	**OV**	Overflow flag
1	**F1**	F1 flag User-defined flag
0	**P**	Parity flag. This flag bit is the parity value of the number of 1s in the accumulator ACC. 1: The number of 1s in ACC is odd 0: The number of 1s in ACC is even (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 SC92F848X power supply system includes BG, LDO, POR, LVR and other circuits, which can achieve reliable operation in the range of 2.0~5.5V. In addition, the IC has a built-in, accurate 1.024V, 2.4V and 2.048V voltage that can be used as an internal reference voltage for the ADC. Users can find the specific settings in the 18 analog-to-digital converter (ADC).

7.2 Power-on Reset

After the SC92F848X 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 SC92F848X will always be reset until the voltage supplied to SC92F848X is higher than a certain voltage, and the internal Clock starts to be effective. 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 warm-up counter inside The SC92F848X. During the reset stage, the warm-up counter is cleared to 0 until the voltage exceeds the POR voltage, the internal RC oscillator starts to oscillate, and the warm-up counter starts counting. When the internal warm-up counter counts to a certain number, every certain number of HRC clocks will read a byte of data from the IFB (including Code Option) in the Flash ROM and store it in the internal system register. This reset signal will not end until the warm-up is completed.

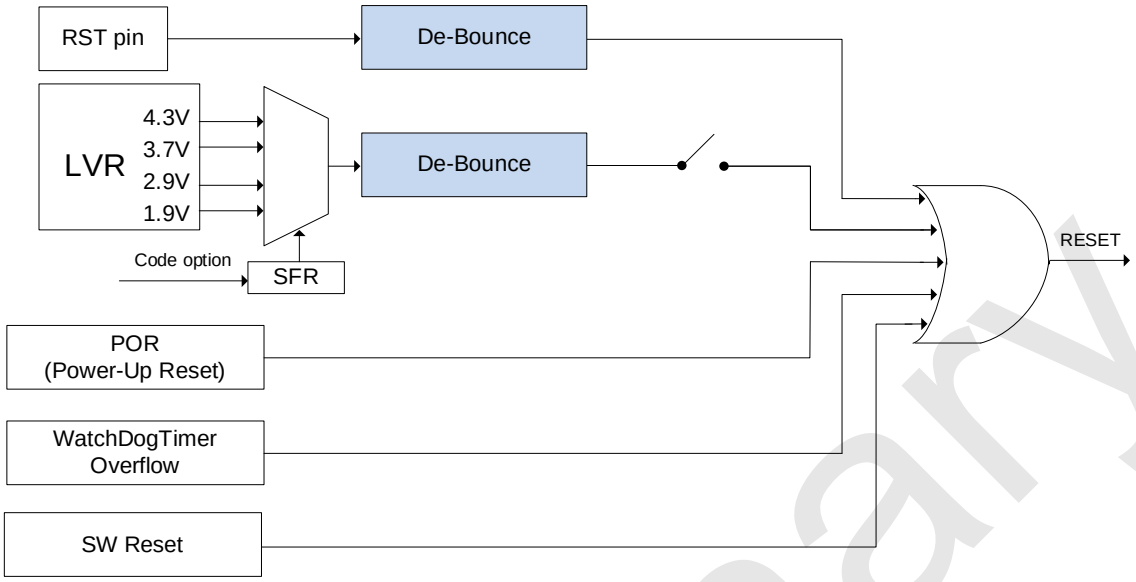
7.2.3 Normal Operation Stage

After finishing the Loading Information stage, The SC92F848X starts to read the instruction code from Flash and enters the normal operation stage. The LVR voltage is the set value of Code Option written by the user.

7.3 Reset Modes

1. The SC92F848X has 5 kinds of reset modes: ① External RST reset ②Low-voltage reset (LVR) ③Power-on reset (POR) ④Watchdog (WDT) reset⑤Software reset.

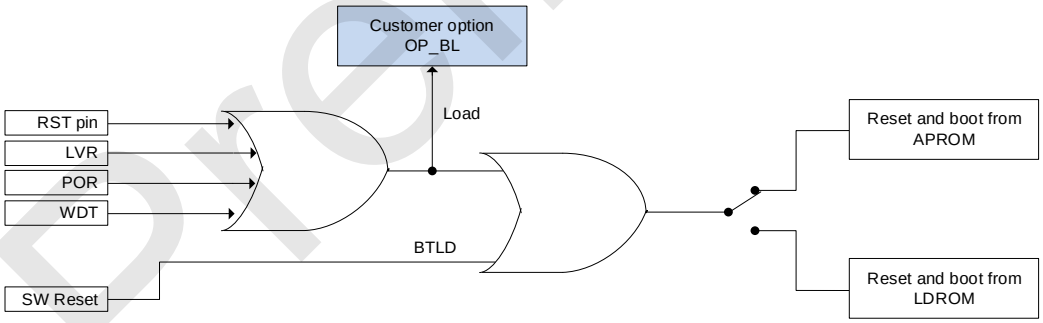
The circuit diagram of the reset part of the SC92F848X is as follows:



SC92F848X Reset circuit diagram

After reset the boot area：
After the external RST reset, low voltage reset LVR, power-on reset POR, watchdog WDT, the chip starts from the boot region (APROM/LDROM) set by the user OP_BL.

After the software is reset, the chip is started according to the boot region (APROM/LDROM) set by BTLD (IAPCTL.7).



SC92F848X's boot area switch after reset

7.3.1 External Reset

External reset is a reset pulse signal of a certain width given to SC92F848X from external RST pin to realize the reset of SC92F848X. The user can configure the P1.1 pin as RST (reset pin) by Code Option.

7.3.2 Low-voltage Reset LVR

The SC92F848X provides a low-voltage reset circuit. There are 4-level LVR voltage options: 4.3V, 3.7V, 2.9V, 1.9V. The default value is the Option value written by the user. A reset occurs when the VDD voltage is less than the threshold voltage for low-voltage reset and the duration is greater than T_{LVR}. Among them, T_{LVR} is the buffeting time of LVR, about 30μs.

OP_CTM0(C1H@FFH) Code Option Register 0 (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	ENWDT	-	SCLKS[1: 0]		DISRST	DISLVR	LVRS[1: 0]	
R/W	R/W	-	R/W		R/W	R/W	R/W	
POR	n	x	n		n	n	n	

Bit number	Bit Mnemonic	Description
2	**DISLVR**	LVR enable setting 0: LVR valid 1: LVR invalid
1~0	**LVRS [1: 0]**	LVR voltage threshold selection control 11: 4.3V 10: 3.7V 01: 2.9V 00: 1.9V

7.3.3 Power-on Reset (POR)

The SC92F848X has a power-on reset circuit inside. When the power supply voltage VDD reaches the POR reset voltage, the system automatically resets.

7.3.4 Watchdog Reset (WDT)

The SC92F848X has a WDT, the clock source of which is the internal 32kHz LRC. The user can choose whether to enable the watchdog reset function by Code Option.

OP_CTM0 (C1H@FFH) Code Option Register 0 (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	ENWDT	-	SCLKS[1: 0]		DISRST	DISLVR	LVRS[1: 0]	
R/W	R/W	-	R/W		R/W	R/W	R/W	
POR	n	x	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)

<table>
<tr><td></td><td></td><td>1: WDT counter restart, cleared by system hardware</td></tr>
<tr><td>2~0</td><td>WDTCKS [2: 0]</td><td>Watchdog clock selection
<table>
<tr><th>WDTCKS[2: 0]</th><th>WDT overflow time</th></tr>
<tr><td>000</td><td>500ms</td></tr>
<tr><td>001</td><td>250ms</td></tr>
<tr><td>010</td><td>125ms</td></tr>
<tr><td>011</td><td>62.5ms</td></tr>
<tr><td>100</td><td>31.5ms</td></tr>
<tr><td>101</td><td>15.75ms</td></tr>
<tr><td>110</td><td>7.88ms</td></tr>
<tr><td>111</td><td>3.94ms</td></tr>
</table></td></tr>
<tr><td>7~5,3</td><td>-</td><td>Reserved</td></tr>
</table>

7.3.5 Software Reset

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

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	SMOD	-	-	-	RST	-	STOP	IDL
R/W	write only	-	-	-	Write only	-	Write only	Write only

POR	0	x	x	x	n	x	0	0

Bit number	Bit Mnemonic	Description
3	**RST**	Software reset control bit: Write status: 0: The program runs normally; 1: The CPU resets immediately after this bit is written to "1"

7.3.6 Register Reset Value

When The SC92F848X is in reset state, most registers will return to their initial state. The watchdog (WDT) is turned off. The initial value of the program counter PC is 0000h, and the initial value of the stack pointer SP is 07h. The "hot restart" Reset (such as WDT, LVR, software reset, etc.) will not affect the SRAM, and the SRAM value is always the value before the reset. The loss of SRAM content will occur when the power supply voltage is so low that the RAM cannot be saved.

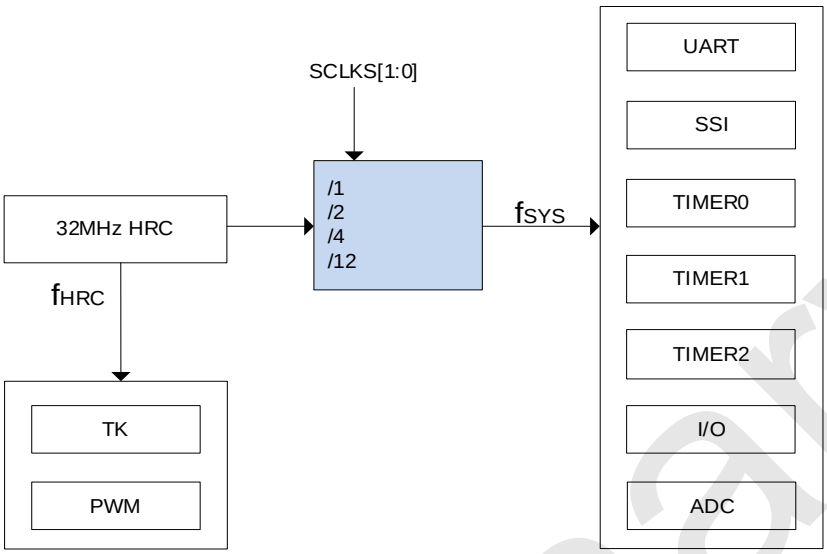
The initial values of the power-on reset of the SFR register see 6.2.1 SFR.

7.4 High- frequency System Clock Circuit

The SC92F848X has a built-in high-precision high-frequency oscillator (HRC) with adjustable oscillation frequency as system clock. The HRC is accurately adjusted to 32 MHz@5V/25℃ at the factory. Users can use the Code Option of the programmer to set the frequency division of the high-frequency clock source to /1 /2 /4 /12 as the system clock fSYS . The adjustment process is to filter out the effect of the deviation on the accuracy. This HRC will have a certain drift under the influence of the ambient temperature and operating voltage: -40 ~ 105℃ application environment, does not exceed ±2%.

Note:

The clock source of the PWM and TK circuits is fixed at f_{HRC} = 32 MHz, do not change with the internal and external system clocks.



SC92F848X Internal clock relationship

OP_CTM0 (C1H@FFH) Code Option Register 0 (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	ENWDT	-	SCLKS[1: 0]		DISRST	DISLVR	LVRS[1: 0]	
R/W	R/W	-	R/W		read only	R/W	R/W	
POR	n	x	n		n	n	n	

Bit number	Bit Mnemonic	Description
5~4	**SCLKS[1: 0]**	System clock frequency selection bits 00: System clock frequency is HRC frequency divided by 1; 01: System clock frequency is HRC frequency divided by 2; 10: System clock frequency is HRC frequency divided by 4;

		11: System clock frequency is HRC frequency divided by 12;

Note: The SC92F848X has a special function: the user can modify the value of SFR to adjust the HRC frequency within a certain range. The user can achieve this by configuring the OP_HRCR register.

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** The user can change the high-frequency oscillator frequency f_{HRC} by modifying the value of this register, and then change the system clock frequency fsys of the IC: The initial value of OP_HRCR[7: 0] after power-on OP_HRCR[s] is a fixed value to ensure that f_{HRC} is 32 MHz, OP_HRCR[s] of each IC may be different When the initial value is OP_HRCR[s], the system clock frequency fsys of the IC can be set to an accurate 32/16/8/2.66 MHz through the Option item. When OP_HRCR [7: 0] changes by 1, the fsys frequency changes by about 0.18% The relationship between OP_HRCR [7: 0] and Fsys output frequency is as follows:

OP_HRCR [7: 0] value	fsys actual output frequency (32M as an example)

		OP_HRCR [s]-n	32000*(1-0.18%*n) kHz
		...	....
		OP_HRCR [s]-2	32000*(1-0.18%*2) = 31 884.8 kHz
		OP_HRCR [s]-1	32000*(1-0.18%*1) = 31 942.4 kHz
		OP_HRCR [s]	32000 kHz
		OP_HRCR [s]+1	32000*(1+0.18%*1) = 32 057.6 kHz
		OP_HRCR [s]+2	32000*(1+0.18%*2) = 32 115.2 kHz
		...	...
		OP_HRCR [s]+n	32000*(1+0.18%*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 fHRC closest to 32/16/8/2.66MHz; the user can modify the value of HRC after each power-on by means of EEPROM to make IC system clock frequency fSYS work at the frequency the user needs.

2.To guarantee IC operating reliably, the maximum operating frequency of IC shall not exceed 10% of 32MHz, which is 35.2MHz;

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

7.5 Low- frequency Clock Timer

The SC92F848X built-in a 32kHz LRC oscillator, can be used as the Base Timer clock. source. SC92F848X, which The oscillator is directly connected to a Base Timer, which can wake the CPU from STOP mode and generate an interrupt.

BTMCON (CEH) 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 0: Base Timer and its clock source do not start 1: Base Timer and its clock source start
6	**BTMIF**	Base Timer interrupt application flag When the CPU accepts the Base Timer interrupt, this flag will be automatically cleared by hardware.
3~0	**BTMFS [3: 0]**	Low frequency clock interrupt frequency selection 0000: An interrupt is generated every 15.625ms 0001: An interrupt is generated every 31.25ms 0010: An interrupt is generated every 62.5ms 0011: An interrupt is generated every 125ms 0100: An interrupt is generated every 0.25 seconds 0101: An interrupt is generated every 0.5 seconds

		0110: An interrupt is generated every 1.0 seconds 0111: An interrupt is generated every 2.0 seconds 1000: An interrupt is generated every 4.0ms 1001: An interrupt is generated every 8.0 seconds 1010: An interrupt is generated every 16.0 seconds 1011: An interrupt is generated every 32.0 seconds 1100~1111: reserved
5~4	-	reserved

7.6 STOP Mode and IDLE Mode

The SC92F848X supports two different software selectable power-reducing modes: IDLE and STOP. These modes are accessed through the PCON register.

Write 1 to PCON. 1, the internal high frequency system clock will STOP, into the STOP mode, to achieve power saving function. In STOP mode, the user can wake up SC92F848X by external interrupt INT0~2, Base Timer interrupt and TK interrupt, or by external reset STOP.

Setting the PCON.0 bit enters IDLE mode. In IDLE mode the program stops running and all CPU states are saved before entering IDLE mode. IDLE mode can be woken up by any interrupt.

PCON (87H) Power Management Control Register (write only, *not readable*)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	SMOD	-	-	-	RST	-	STOP	IDL
R/W	Write only	-	-	-	Write only	-	Write only	Write only
POR	0	x	x	x	n	x	0	0

Bit number	Bit Mnemonic	Description

1	**STOP**	STOP mode bit. Setting this bit activates STOP mode operation.
0	**IDL**	IDLE mode bit. Setting this bit activates IDLE mode operation.

Notes: When Configuring 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: set MCU to enter STOP mode:

Example in C Language

```
#include”intrins.h”
PCON |= 0x02;        // PCON bit1 STOP bit write 1, configure the MCU to enter STOP mode
_nop_();             // At least 8 _nop_() are required
_nop_();
_nop_();
_nop_();
_nop_();
_nop_();
_nop_();
_nop_();
……
```

Assembly Language:

```
ORL PCON,#02H        ; PCON bit1 STOP bit write 1, configure the MCU to enter STOP mode
NOP                  ; At least 8 NOPs are requiredNOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
……
```

8 CPU and Instruction Set

8.1 CPU

The SC92F848X is built around an enhanced high-speed 1T 8051 core, and its instructions are fully compatible with classic 8051 core.

8.2 Addressing Mode

The addressing modes of 1T 8051 CPU instructions of the SC92F848X are: ①Immediate Addressing ② Direct Addressing ③ Indirect Addressing ④ Register Addressing ⑤ Relative Addressing ⑥ Indexed Addressing ⑦ Bit Addressing.

8.2.1 Immediate Addressing

Immediate addressing is also called immediate data addressing. It directly gives the operands participating in the operation in the instruction operand. Examples of instructions are as follows:

MOV A, #50H (This instruction moves the immediate value 50H to accumulator A)

8.2.2 Direct Addressing

In direct addressing mode, the instruction operand field gives the address of the operand to participate in the operation. The direct addressing mode can only be used to represent special function registers, internal data registers, and bit address spaces. The special function registers and bit address spaces can only be accessed by direct addressing.

Examples are as follows:

ANL 50H, #91H

(indicating that the number in the 50H unit is ANDed with the immediate 91H, and the result is stored in the 50H unit. 50H is direct address, representing a unit in the internal data register RAM.)

8.2.3 Indirect Addressing

Indirect addressing is indicated by adding the "@" symbol before R0 or R1. Assuming that the data in R1 is 40H, and the data in the internal data memory 40H unit is 55H, the instruction is

MOV A, @R1 (Move data 55H to accumulator A).

8.2.4 Register Addressing

When register addressing, operate on the selected operating registers R7~R0, accumulator A, general register B, address register and carry C. Registers R7~R0 are represented by the low three bits of the instruction code, and ACC, B, DPTR and carry bit C are implicitly contained in the instruction code. Therefore, register addressing also includes an implicit addressing method. The selection of the register operating area is determined by RS1 and RS0 in the program status word register PSW. The register specified by the instruction operand refers to the register in the current operating area.

INC R0 Refers to(R0)+1→R0

8.2.5 Relative Addressing

Relative addressing is to add the current value in the program counter PC to the number given by the second byte of the instruction, and the result is used as the branch address of the branch instruction. The branch address also becomes the branch destination address, the current value in the PC becomes the base address, and the number given by the second byte of the instruction becomes the offset. Since the destination address is relative to the base address in the PC, this addressing method becomes relative addressing. The offset is a signed number, and the range that can be expressed is -128~+127. This addressing method is mainly used for branch instructions.

JC $+50H

It means that if the carry bit C is 0, the content in the program counter PC does not change, that is, it does not transfer. If the carry bit C is 1, the current value as base address in the PC plus the offset 50H will be used as the destination address of the branch instruction.

8.2.6 Indexed Addressing

In the indexed addressing mode, the instruction operand specifies an index register that stores the index base address. In indexed addressing, the offset is added to the index base value, and the result is used as the address of the operand. The index registers are the program counter PC and the address register DPTR.

MOVC A, @A+DPTR

It indicates that the accumulator A is an offset register, and its content is added to the content of the address register DPTR. The result is used as the address of the operand, and the number in this unit is taken out and sent to the accumulator A.

8.2.7 Bits Addressing

Bit addressing refers to the addressing mode when performing bit operations on some internal data memory RAMs and special function registers that can perform bit operations. When performing bit operations, with the help of carry bit C as a bit operation accumulator, the instruction operand directly gives the address of the bit, and then performs bit operation on the bit according to the nature of the opcode. The bit address is exactly the same as the byte address encoding method in direct byte addressing, which is mainly distinguished by the nature of the operation instruction, and special attention should be paid when using it.

MOV C, 20H (The value of the bit manipulation register with address 20H is sent to carry bit C)

9 Interrupts

SC92F848X provides 12 interrupt sources: TIMER0,TIMER1, TIMER2, INT0~2, ADC, PWM, UART, SSI, BASE TIMER, TK. The 12 interrupt sources are divided into two interrupt priorities and can be set to either high or low priority separately.Three external interrupts can be set as up, down or both trigger conditions for each interrupt source respectively. Each interrupt has its own priority setting bit, interrupt flag, interrupt vector and enable bit respectively. The total enable bit EA can open or close all interrupts.

9.1 Interrupt Source and Vector

The list of the SC92F848X interrupt sources, interrupt vectors, and related control bits are as follows:

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	External interrupt 0 conditions are met	IE0	EINT0	IPINT0	0003H	1 (HIGH)	0	H/W Auto	YES
Timer 0	Timer 0 overflow	TF0	ET0	IPT0	000BH	2	1	H/W Auto	NO
INT1	External interrupt 1 conditions are met	IE1	EINT1	IPINT1	0013H	3	2	H/W Auto	YES
Timer 1	Timer 1 overflow	TF1	ET1	IPT1	001BH	4	3	H/W Auto	NO
UART	Receive or send completed	RI/TI	EUART	IPUART	0023H	5	4	Must user Clear	NO
Timer 2	Timer 2 overflow	TF2	ET2	IPT2	002BH	6	5	Must user Clear	NO
ADC	ADC conversion completed	ADCIF	EADC	IPADC	0033H	7	6	Must user Clear	NO

SSI	Receive or send completed	SPIF/TWIF	ESSI	IPSPI	003BH	8	7	Must user Clear	NO
PWM	PWM overflow	PWMIF	EPWM	IPPWM	0043H	9	8	H/W Auto	NO
BTM	Base timer overflow	BTMIF	EBTM	IPBTM	004BH	10	9	H/W Auto	YES
INT2	External interrupt 2 conditions are met	-	EINT2	IPINT2	0053H	11	10	-	YES
TK	Touch Key counter overflowed	TKIF	ETK	IPTK	005BH	12	11	H/W Auto	YES

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

Timer Interrupt: Interrupt generates when Timer 0 or Timer 1 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 Timer 2 overflows and the interrupt flag TF2 is set to “1”. Once Timer 2 interrupt generates, the hardware would not automatically clear TF2 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.

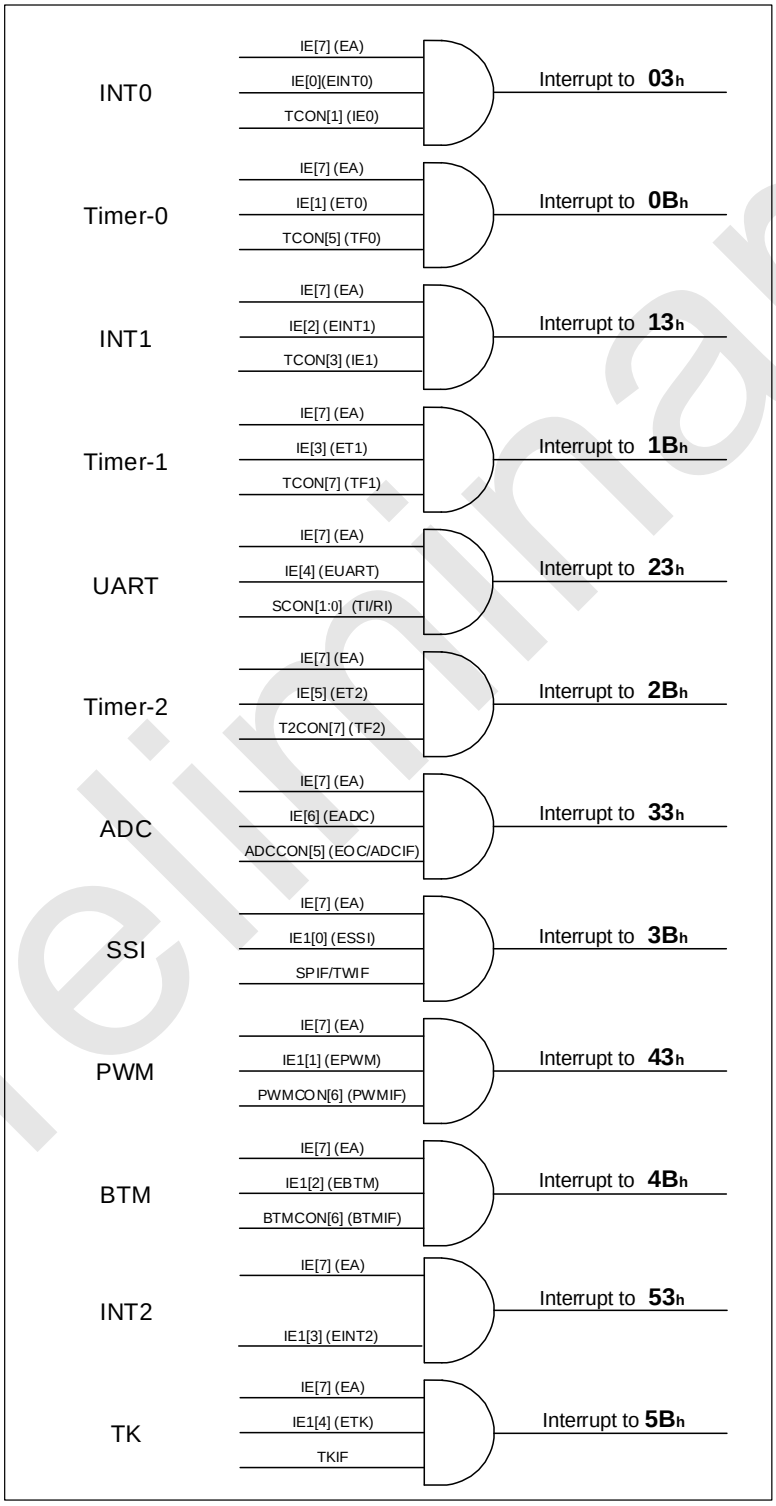
SSI Interrupt: When SSI completes receiving or transmitting a frame of data, SPIF/TWIF bit will be set to “1” automatically by hardware, and SSI interrupt generates. When the microcontroller unit serves SSI interrupt, the interrupt flag SPIF/TWIF must be cleared by software.

PWM Interrupt: When PWM counter overflows (beyond PWMPD),PWMIF(PWM Interrupt Flag) will be set as 1 automatically by hardware, PWM interrupt will occurs. Once PWM interrupt occurs, the hardware would clear PWMIF(the interrupt flag) automatically.

External Interrupt INT0 ~ 2: An external interrupt occurs when an external interrupt condition occurs at an external intermediate fracture. INT0 has three external interrupt sources, INT1 has four external interrupt sources, INT2 has six external interrupt sources, users can be as required to set the top edge, bottom edge, or double edge interrupt, by setting SFR (INTxF and INTxR) to achieve. The user can set the priority level of each interrupt through the IP register. External interrupt INT0~2 can also wake up the MCU STOP.

9.2 Interrupt Structure Diagram

The interrupt structure of SC92F848X is shown below:



SC92F848X Interrupt structure and vector

9.3 Interrupt Priority

Each interrupt source can be individually programmed to one of two priority levels by setting or clearing bits in the interrupt priority registers: IP, IP0, IP1. An interrupt service routine in progress can be interrupted by a higher priority interrupt. The highest priority interrupt cannot be interrupted by any other interrupt source. If two requests of different priority levels are pending at the end of an instruction, the request of higher priority level is serviced. If requests of the same priority level are pending at the end of an instruction, an internal polling sequence determines which request is serviced. The polling sequence is based on the vector address; an interrupt with a lower vector address has higher priority than an interrupt with a higher vector address.

9.4 Interrupt Processing Flow

When an interrupt is generated and responded by the CPU, the main program execution is interrupted and the following operations will be performed:

① The currently executing instruction is finished;

② The PC value is pushed into the stack to protect the scene;

③ The interrupt vector address is loaded into the program counter PC;

④ Execute the corresponding interrupt service program;

⑤ The interrupt service routine ends and RETI;

⑥ Unstack the PC value and return to the program before the interruption.

In this process, the system will not immediately execute other interrupts of the same priority, but will retain the interrupt request that has occurred, and after the current interrupt processing is completed, go to execute a new interrupt request.

9.5 Interrupt-related SFR 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**	Interrupt enable total control 0: Close all interrupts 1: Enable all interrupts
6	**EADC**	ADC interrupt enable control 0: Disable ADC interrupt 1: Allow the ADC to generate an interrupt when the conversion is complete
5	**ET2**	Timer 2 interrupt enable control 0: Disable Timer 2 interrupt 1: Enable Timer 2 interrupt
4	**EUART**	UART interrupt enable control 0: Disable UART interrupt 1: Allow UART interrupt
3	**ET1**	Timer 1 interrupt enable control 0: Disable Timer1 interrupt 1: Enable Timer 1 interrupt
2	**EINT1**	External interrupt 1 enable control 0: close INT1 interrupt 1: Enable INT1 interrupt
1	**ET0**	Timer 0 interrupt enable control 0: Disable TIMER0 interrupt 1: Enable TIMER0 interrupt
0	**EINT0**	External interrupt 0 enable control 0: close 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 0: ADC interrupt priority is low 1: ADC interrupt priority is high
5	**IPT2**	Timer 2 interrupt priority selection 0: Timer 2 interrupt priority is low 1: Timer 2 interrupt priority is high
4	**IPUART**	UART interrupt priority selection 0: UART interrupt priority is low 1: UART interrupt priority is high
3	**IPT1**	Timer 1 interrupt priority selection 0: Timer 1 interrupt priority is low 1: Timer 1 interrupt priority is high

2	**IPINT1**	INT1 counter interrupt priority selection 0: INT1 interrupt priority is low 1: INT1 interrupt priority is high
1	**IPT0**	Timer 0 interrupt priority selection 0: Timer 0 interrupt priority is low 1: Timer 0 interrupt priority is high
0	**IPINT0**	INT0 counter interrupt priority selection 0: INT0 interrupt priority is low 1: INT0 interrupt priority is high
7	**-**	Reserved

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

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	-	-	-	ETK	EINT2	EBTM	EPWM	ESSI
R/W	-	-	-	R/W	R/W	R/W	R/W	R/W
POR	x	x	x	0	0	0	0	0

Bit number	Bit Mnemonic	Description
4	**TK**	Touch Key interrupts enable control 0: Turn off Touch Key interrupt 1: Open Touch Key interrupt

3	**EINT2**	External interrupt 2 enable control 0: close INT2 interrupt 1: Open INT2 interrupt
2	**EBTM**	Base Timer interrupt enable control 0: Disable Base Timer interrupt 1: Enable Base Timer interrupt
1	**EPWM**	PWM interrupt enable control 0: Disable PWM interrupt 1: Enable interrupt when PWM count overflows(count to PWMPRD)
0	**ESSI**	Three-in-one serial port interrupt enable control 0: Disable serial port interrupt 1: Allow serial port interrupt
7~5	**-**	Reserved

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

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	-	-	-	IPTK	IPINT2	IPBTM	IPPWM	IPSSI
R/W	-	-	-	R/W	R/W	R/W	R/W	R/W
POR	x	x	x	0	0	0	0	0

Bit number	Bit Mnemonic	Description

4	**IPTK**	Touch Key interrupts priority selection 0: Touch Key interrupt priority is low 1: Touch Key interrupt priority is high
3	**IPINT2**	INT2 counter interrupt priority selection 0: INT2 interrupt priority is low 1: INT2 interrupt priority is high
2	**IPBTM**	Base Timer interrupt priority selection 0: Base Timer interrupt priority is low 1: Base Timer interrupt priority is high
1	**IPPWM**	PWM interrupt enable selection 0: PWM interrupt priority is low 1: PWM interrupt priority is high
0	**IPSSI**	Three-in-one serial port interrupt priority selection 0: SSI interrupt priority is low 1: SSI interrupt priority is high
7~5	**-**	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. INT1 generates an overflow. When an interrupt occurs, the hardware sets IE1 to "1" and applies for an interrupt. When the CPU responds, the hardware clears "0".
1	**IE0**	INT0 overflow interrupt request flag. INT0 generates an overflow. When an interrupt occurs, the hardware sets IE0 to "1" and applies for an interrupt. When the CPU responds, the hardware clears "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	-
R/W	-	-	-	-	R/W	R/W	R/W	-
POR	x	x	x	x	0	0	0	x

Bit number	Bit Mnemonic	Description
3~1	**INT0Fn** **(n=1~3)**	INT0 falling edge interrupt control 0: INT0n falling edge interrupt close 1: INT0n falling edge interrupt enable
7~4,0	**-**	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	-
R/W	-	-	-	-	R/W	R/W	R/W	-
POR	x	x	x	x	0	0	0	x

Bit number	Bit Mnemonic	Description
3~1	**INT0Rn** **(n=1~3)**	INT0 rising edge interrupt control 0: INT0n rising edge interrupt close 1: INT0n rising edge interrupt enable
7~4,0	**-**	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 0: INT1n falling edge interrupt close

		1: INT1n falling edge interrupt enable
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 0: INT1n rising edge interrupt off 1: INT1n rising edge interrupt enable
7~4	-	Reserved

INT2F (C6H) INT2 Falling Edge Interrupt Control Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	-	-	INT2F5	INT2F4	INT2F3	INT2F2	INT2F1	INT2F0
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	**INT2Fn** **(n=0~5)**	INT2 falling edge interrupt control 0: INT2n falling edge interrupt close 1: INT2n falling edge interrupt enable
7~6	**-**	Reserved

INT2R (C7H) INT2 Rising Edge Interrupt Control Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	-	-	INT2R5	INT2R4	INT2R3	INT2R2	INT2R1	INT2R0
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	**INT2Rn** **(n=0~5)**	INT2 rising edge interrupt control 0: INT2n rising edge interrupt close 1: INT2n rising edge interrupt enable
7~6	**-**	Reserved

10 Timer/Counter T0 and T1

Timer 0 and Timer 1 inside the SC92F848X MCU are two 16-bit timers/counters. They have two operating modes: counting mode and timing mode. There is a control bit C/Tx in the special function register TMOD to select whether T0 and T1 are timers or counters. 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. Only when TRx=1, T0 and T1 will be opened to count.

In counter mode, for each pulse on the P1.2/T0 and P1.3/T1 pins, the count value of T0 and T1 increases by 1, respectively.

In the timer mode, the count source of T0 and T1 can be selected as fsys/12 or fsys through the special function register TMCON (fsys is the divided system clock).

There are 4 operating modes for timer/counter T0, and 3 operating modes for timer/counter T1 (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 auto-reload mode

④ Mode 3: Two 8-bit timer/counter modes

In the 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

Symbol	Address	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	Low 8 bits of timer 0	TL0[7: 0]								00000000b
TL1	8BH	Low 8 bits of timer 1	TL1[7: 0]								00000000b
TH0	8CH	Timer 0 high 8 bits	TH0[7: 0]								00000000b
TH1	8DH	Timer 1 high 8 bits	TH1[7: 0]								00000000b

TMCON	8EH	Timer frequency control register	-	-	-	-	-	T2FD	T1FD	T0FD	xxxxx000b

The explanation of each register is as follows:

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**	T1 overflow interrupt request flag. T1 generates an overflow. When an interrupt occurs, the hardware sets TF1 to "1" and applies for an interrupt. When the CPU responds, the hardware clears "0".
6	**TR1**	Operation control bit of timer T1. This bit is set and cleared by software. When TR1=1, T1 is allowed to start counting. When TR1=0, T1 counting is prohibited.
5	**TF0**	T0 overflow interrupt request flag. T0 overflows. When an interrupt occurs, the hardware sets TF0 to "1" and applies for an interrupt. When the CPU responds, the hardware clears "0".
4	**TR0**	Operation control bit of timer T0. This bit is set and cleared by software. When TR0=1, T0 is allowed to start counting. When TR0=0, T0 counting is prohibited.
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			

Bit number	Bit Mnemonic	Description
6	**C/T1**	TMOD[6] control timer 1 0: Timer, T1 count comes from fsys frequency division 1: Counter, T1 count comes from external pin T1/P1.3
5~4	**M11,M01**	Timer/Counter 1 mode selection 00: 13-bit timer/counter, the upper 3 bits of TL1 are invalid 01: 16-bit timer/counter, TL1 and TH1 all are valid 10: 8-bit auto-reload timer, automatically reload the value stored in TH1 into TL1 when overflow 11: Timer/Counter 1 is invalid (stop counting)
2	**C/T0**	TMOD[2] control timer 0 0: Timer, T0 count comes from fsys frequency division 1: Counter, T0 count comes from external pin T0/P1.2
1~0	**M10,M00**	Timer/Counter 0 mode selection 00: 13-bit timer/counter, the upper 3 bits of TL0 are invalid

		01: 16-bit timer/counter, TL0 and TH0 all are valid 10: 8-bit auto-reload timer, automatically reload the value stored in TH0 into TL0 when overflow 11: Timer 0 is now a dual 8-bit timer/counter. TL0 is an 8-bit timer/counter controlled by the control bits of standard timer 0; TH0 is only an 8-bit timer controlled by the control bits of timer 1.
7,3	-	Reserved

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

The timer and counter Tx functions are selected by the control bits C/Tx of the special function register TMOD. M0x and M1x are used to select the Tx operating mode. TRx acts as the switch control of T0 and T1. Only when TRx=1, T0 and T1 are turned on.

TMCON (8EH) Timer Frequency Control Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	-	-	-	-	-	T2FD	T1FD	T0FD
R/W	-	-	-	-	-	R/W	R/W	R/W
POR	x	x	x	x	x	0	0	0

Bit number	Bit Mnemonic	Description
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

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**	Timer 1 interrupt enable control 0: Disable Timer 1 interrupt 1: Enable Timer 1 interrupt
1	**ET0**	Timer 0 interrupt enable control 0: Disable Timer 0 interrupt 1: Enable Timer 0 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**	Timer 1 interrupt priority 0: Set the interrupt priority of Timer 1 to "Low" 1: Set the interrupt priority of Timer 1 to "High"
1	**IPT0**	Timer 0 interrupt priority 0: Set the interrupt priority of Timer 0 to "Low" 1: Set the interrupt priority of Timer 0 to "High"

10.2 T0 Operating Modes

By setting M10 and M00 (TMOD[1], TMOD[0]) in the register TMOD, timer/counter 0 can realize 4 different operating modes.

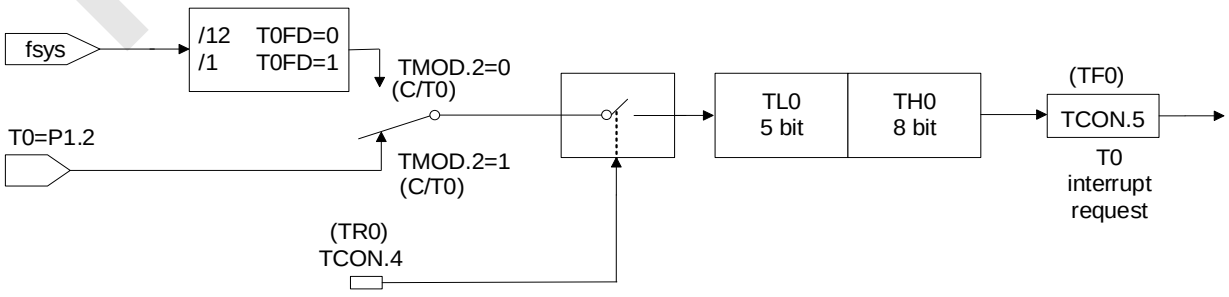
Operating Mode 0: 13-bit Counter/Timer

TH0 register stores the upper 8 bits (TH0.7~TH0.0) of the 13-bit counter/timer, and the TL0 stores the low 5 bits (TL0.4~TL0.0). The upper three bits of TL0 (TL0.7~TL0.5) are uncertain values and should be ignored when reading. When the 13-bit timer/counter overflows, the system will set the timer overflow flag TF0 to 1. If the timer 0 interrupt is enabled, an interrupt will be generated.

C/T0 bit selects the clock input source of the counter/timer. If C/T0=1, the level change of the timer 0 input pin T0 (P1.2) from high to low will increase the timer 0 data register by 1. If C/T0=0, select the frequency division of the system clock as the clock source of timer 0.

When TR0 is set to 1, the timer T0 is started. Setting TR0 does not forcibly reset the timer, meaning that if TR0 is set, the timer register will start counting from the value when TR0 was cleared last time. Therefore, before enabling the timer, the initial value of the timer register should be set.

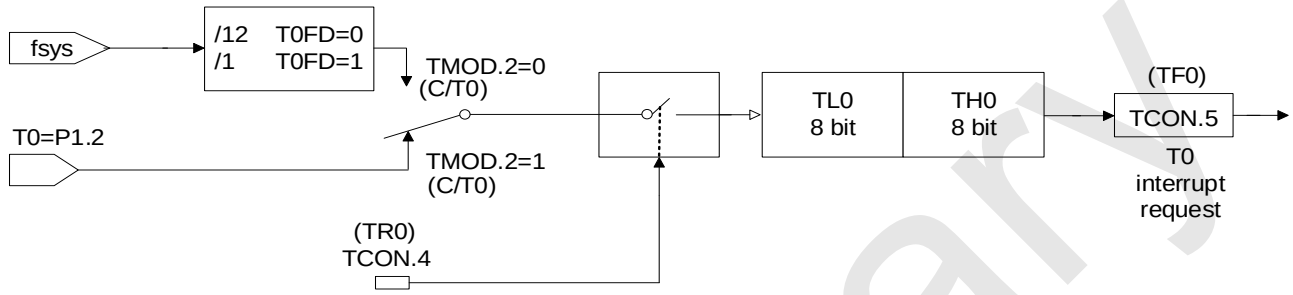
When applied as a timer, TOFD can be configured to select the frequency division ratio of the clock source.



Timer/counter operating mode 0: 13-bit timer/counter

Operating Mode 1: 16-bit Counter/Timer

Except for using a 16-bit (all 8-bit data of TL0 is valid) counters/timers, Mode 1 and Mode 0 operate in the same way. The way to open and configure the counter/timer is the same.



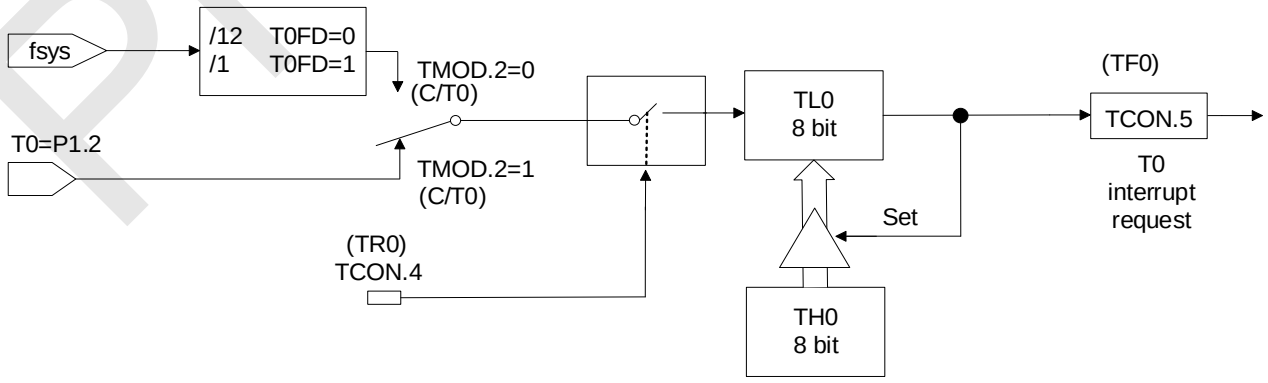
Timer/Counter Operating Mode 1: 16-bit Timer/Counter

Operating Mode 2: 8-bit Automatic Reload Counter/Timer

In operating mode 2, Timer 0 is an 8-bit auto-reload counter/timer. TL0 stores the count value, and TH0 stores the reload value. When the counter in TL0 overflows to 0x00, the timer overflow flag TF0 is set to 1, and the value of register TH0 is reloaded into register TL0. If the timer interrupt is enabled, an interrupt will be generated when TF0 is set to 1, but the reload value in TH0 will not change. Before allowing the timer to count correctly, TL0 must be initialized to the required value.

Except for the auto-reload function, the counter/timer in operating mode 2 is enabled and configured in the same way as in modes 0 and 1.

When used as a timer, the register TMCON.0 (T0FD) can be configured to select the ratio of the timer clock source divided by the system clock fsys.



Timer/counter operating mode 2: 8-bit timer/counter with automatic reload

Operating Mode 3: Two 8-bit Counters/Timers (Timer 0 Only)

In operating mode 3, Timer 0 is used as two independent 8-bit counters/timers, which are controlled by TL0 and TH0, respectively. TL0 is controlled by timer 0 control bits (in TCON) and status bits (in TMOD): TR0, C/T0, TF0. Timer 0 can select the timer mode or counter mode through T0 TMOD.2 (C/T0).

TH0 sets related control by timer 1 control TCON, but TH0 is only limited to timer mode and cannot be set to counter mode by TMOD.2 (C/T0). TH0 is enabled by the control of the timer control bit TR1, and TR1=1 needs to be set. When an overflow occurs and an interrupt is generated, TF1 will be set to 1, and the interrupt will be processed according to T1.

When T0 is set to operating mode 3, the TH0 timer occupies the interrupt resources of T1 and the registers in TCON, and the 16-bit counter of T1 will stop counting, which is equivalent to "TR1=0". When using the TH0 timer to work, set TR1=1.

10.3 T1 Operating Mode

By setting M11 and M01 (TMOD[5], TMOD[4]) in the register TMOD, timer/counter 1 can realize three different operating modes.

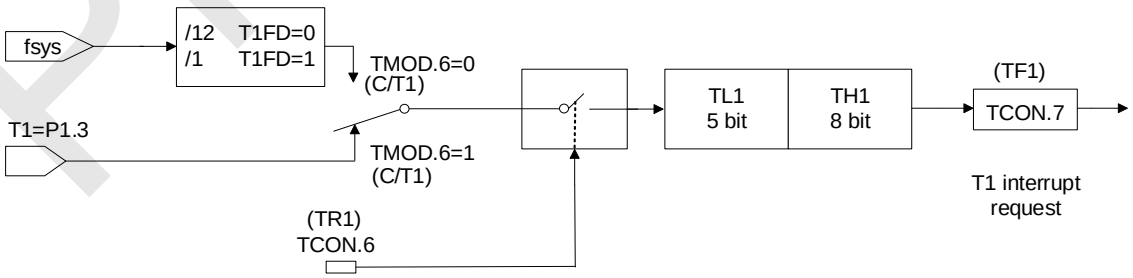
Operating mode 0: 13-bit Timer/Counter

The TH1 register stores the upper 8 bits (TH1.7~TH1.0) of the 13-bit counter/timer; the TL1 stores the low 5 bits (TL1.4~TL1.0). The upper three bits of TL1 (TL1.7~TL1.5) are uncertain values and should be ignored when reading. When the 13-bit timer counter increments and overflows, the system sets the timer overflow flag TF1 to 1. If Timer 1 interrupt is enabled, an interrupt will be generated. The C/T1 bit selects the clock source of the counter/timer.

If C/T1=1, the level of timer 1 input pin T1 (P1.3) changes from high to low, which will increase the timer 1 data register by 1. If C/T1=0, select the frequency division of the system clock as the clock source of timer 1.

Set TR1 to enable the timer. Setting TR1 does not forcibly reset the timer, meaning that if TR1 is set to 1, the timer register will start counting from the value when TR1 was cleared to 0 last time. Therefore, before enabling the timer, the initial value of the timer register should be set.

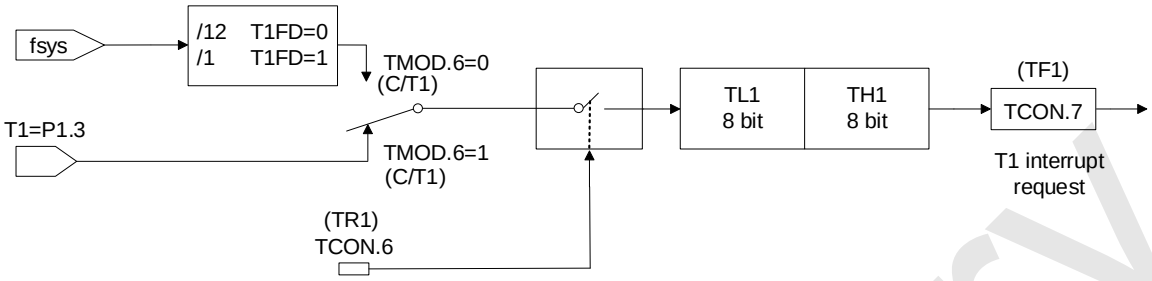
When applied as a timer, T1FD can be configured to select the frequency division ratio of the clock source.



Timer/counter operating mode 0: 13-bit timer/counter

Operating mode 1: 16-bit Counter/Timer

Except for using a 16-bit (all 8-bit data of TL1 is valid) counter/timer, Mode 1 and Mode 0 operate in the same way. The way to open and configure the counter/timer is the same.



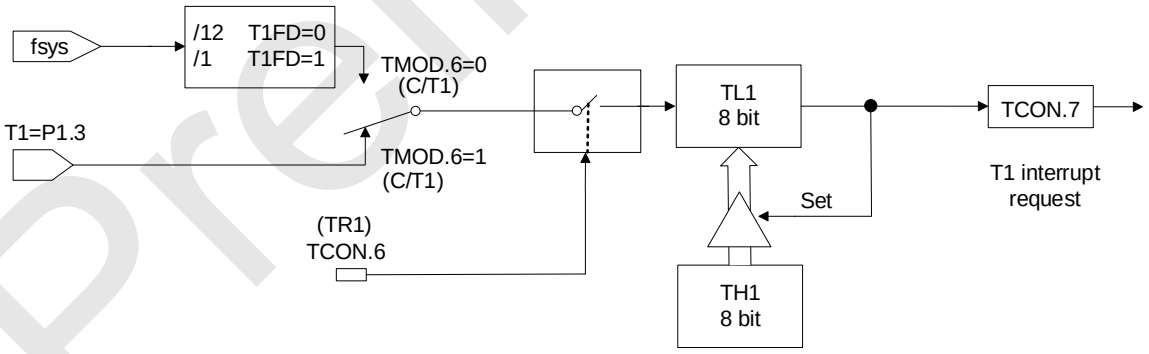
Timer/counter operating mode 1: 16-bit timer/counter

Operating mode 2: 8-bit Automatic Reload Counter/Timer

In operating mode 2, Timer 1 is an 8-bit auto-reload counter/timer. TL1 stores the count value, and TH1 stores the reload value. When the counter in TL1 overflows to 0x00, the timer overflow flag TF1 is set to 1, and the value of register TH1 is reloaded into register TL1. If the timer interrupt is enabled, an interrupt will be generated when TF1 is set to 1, but the reload value in TH1 will not change. Before allowing the timer to count correctly, TL1 must be initialized to the required value.

Except for the auto-reload function, the counter/timer in operating mode 2 is enabled and configured in the same way as modes 0 and 1.

When used as a timer, the register TMCON.1 (T1FD) can be configured to select the ratio of the timer clock source divided by the system clock fsys.



Timer/counter operating mode 2: 8-bit timer/counter with automatic reload

11 Timer/Counter T2

Timer2 inside the SC92F848X microcontroller unit has two operating modes, namely counter mode and timer mode. There is a control bit C/T2 in SFR T2CON to select Timer or Counter for T2. They are adding counters in nature, differing in counting source. The clock source of T2 comes from system clock or frequency division clock, but the source of counters is the input pulse to external pin. TR2 is the counting switch of Timer/Counter T2. Only when TR2 = 1, will T2 be enabled for counting.

In counter mode, each input pulse on T2(P0.7) pin will make the counting value of T2 increase by 1.

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

Timer/Counter T2 has 4 operating modes:

① Mode 0: 16-bit capture mode

② Mode 1: 16-bit automatic reload timer mode

③ Mode 2: Baud rate generator mode

④ Mode 3: Programmable clock output mode

11.1 T2-related Registers

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

Register instructions are shown below:

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

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

Bit Number	Bit Mnemonic	Description
7	**TF2**	Timer2 overflow flag bit 0: No overflow (must be cleared by software) 1: Overflow (if RCLK=0 and TCLK=0, set to 1 by hardware)
6	**EXF2**	T2EX pin external event input (falling edge) detected flag bit 0: No external event input (must be cleared by software) 1:When detecting external input (if EXEN2=1, SET to 1 by hardware)
5	**RCLK**	UART receiving clock control bit 0: Timer1 generates receiving baud rate 1: Timer2 generates receiving baud rate
4	**TCLK**	UART transmitting clock control bit 0: Timer1 generates transmitting baud rate 1: Timer2 generates transmitting baud rate
3	**EXEN2**	External event input (falling edge) on T2EX pin used as reload/capture trigger allowed/prohibited control bit 0: Omit event on T2EX pin 1: When the timer2 is not used as UART clock, a falling edge is detected on T2EX pin and a capture or reload will be generated.

2	**TR2**	Timer2 start/stop control bit 0: Stop Timer2 1: Start Timer2
1	**C/T2**	Timer2 Timer/Counter mode selection bit 2 0: Timer mode, used as I/O interface on T2 pin 1: Counter mode
0	**CP/RL2**	Capture/reload mode selection bit 0: 16-bit Timer/Counter with reload function 1: 16-bit Timer/Counter with capture function, T2EX as timer2 external capture signal input port

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

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

Bit Number	Bit Mnemonic	Description
1	**T2OE**	Timer2 output allow bit 0: Set T2 as clock input or I/O port 1: Set T2 as clock output
0	**DCEN**	Decreasing counting allow bit 0: Prohibits Timer2 as incremental/decreasing counter, Timer2 only used

		as incremental counter 1: Allow Timer2 as incremental/decreasing timer, T2EX is used to select the direction of counting
7 ~ 2	-	Reserved

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

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

Bit Number	Bit Mnemonic	Description
2	**T2FD**	T2 input frequency selection control bit 0: T2 clock source is $f_{SYS}/12$ 1: T2 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
5	**ET2**	Timer2 interrupt enable control bit 0: Disable TIMER2 interrupt 1: Enable TIMER2 interrupt

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

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit 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
5	**IPT2**	Timer2 interrupt priority selection bit 0: Configure Timer2 interrupt priority as “low” 1: Configure Timer2 interrupt priority as “high”

11.2 T2 Operating Modes

The operating mode and configuration mode of Timer2 are shown in the table below:

C/$\overline{T2}$	T2OE	DCEN	TR2	CP/$\overline{RL2}$	EXEN2	Mode	
X	0	X	1	1	1	0	16-bit capture
X	0	0	1	0	0	1	16-bit automatic reload timer, normal automatic reload

X	0	0	1	0	1		16-bit automatic reload timer, with T2EX trigger reload
X	0	1	1	0	X		16-bit automatic reload timer, with increasing or decreasing reload
X	0	X	1	X	X	2	UART0 Baud Rate Generator
0	1	X	1	X	X	3	Programmable clock output
X	X	X	0	X	1	X	Timer stops, but T2EX channel is also available

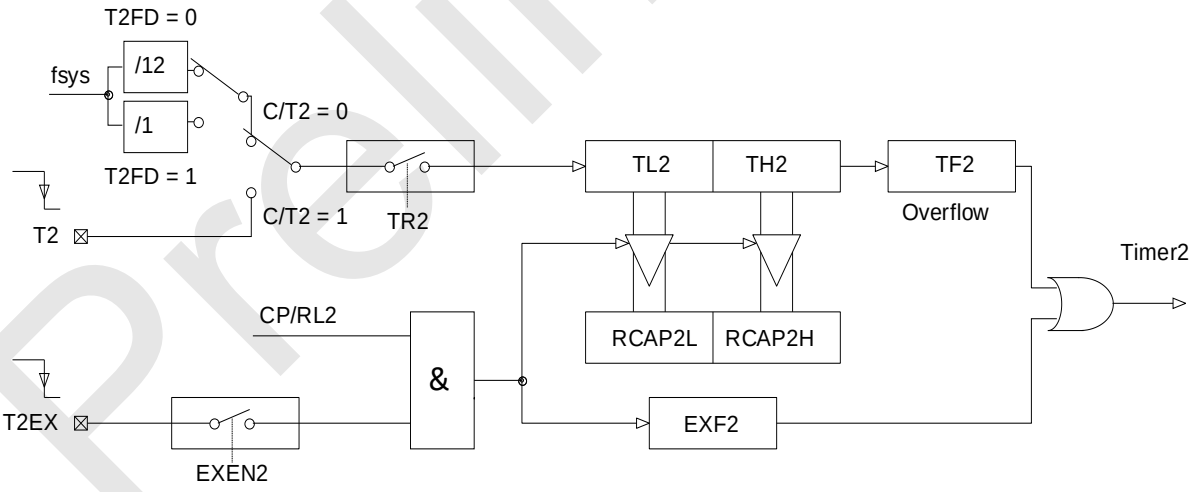
Operating Mode 0: 16-bit capture

Set CP/RL2 to 1 and set timer 2 to 16-bit capture mode

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

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

If EXEN2=1, conduct the same operations as above on Timer2, the falling edge signal on external input T2EX can make current value in TH2 and TL2 captured into RCAP2H and RCAP2L. Besides, the falling edge signal on T2EX can also cause EXF2 in T2CON to be set to 1. If ET2 is set to 1, bit EXTF2, like TF2, will also trigger an interrupt.



Operating Mode 0: 16-bit capture

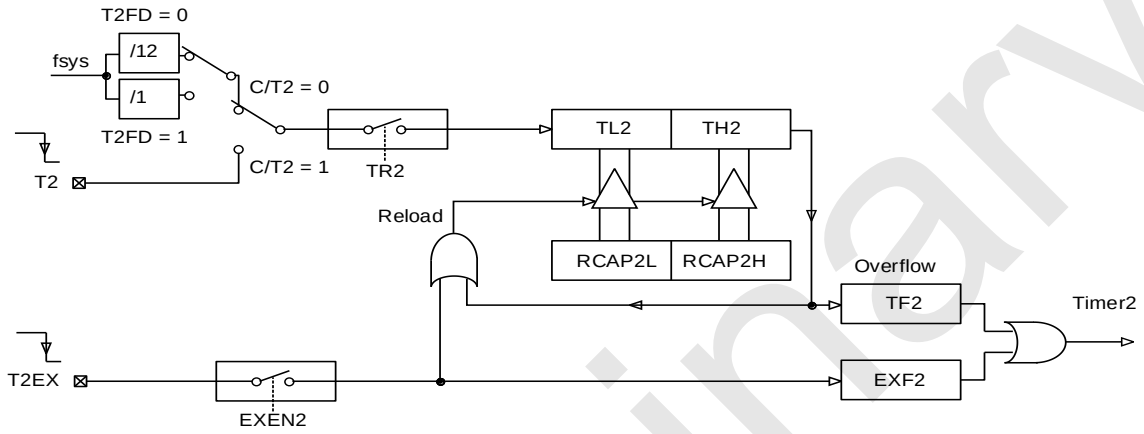
Operating Mode 1: 16-bit Automatic Reload Timer

In the 16-bit automatic reload mode, Timer2 can be selected to work in incrementing or decreasing counting mode. This function can be selected by DCEN bit in T2MOD (decreasing counting allowed). After system reset, the reset value of DCEN bit is 0 and Timer2 is defaulted as decreasing counting. When setting DCEN to 1, the incrementing or decreasing counting depends on the level of T2EX pin.

When DCEN = 0, There are two options for EXEN2 bit in T2CON:

1. EXEN=0, Timer2 will increase to 0xFFFFH and set TF2 bit after overflow. Meanwhile, the timer will load 16-bit value in registers RCAP2H and RCAP2L written by user software into registers TH2 and TL2 automatically.

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



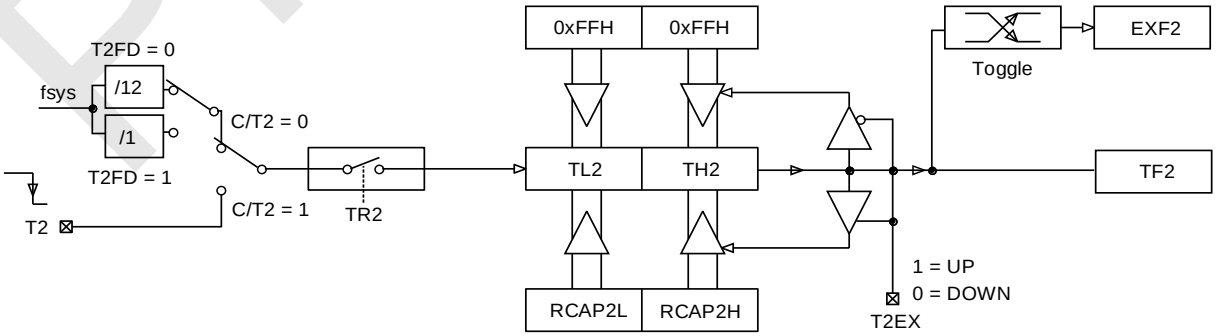
Operating Mode 1: 16-bit Automatic Reload Timer DCEN = 0

Configure CEN bit to allow Timer2 for incremental or decreasing counting. When DCEN=1, T2EX pin controls the count direction, and the control of EXEN2 becomes invalid.

Setting T2EX to 1 can conduct incremental count on Timer2. The Timer overflows when it increases to 0xFFFFH, then it sets TF2 bit. Besides, the overflow can also respectively cause 16-bit value in RCAP2H and RCAP2L to be reloaded into timer registers.

Setting T2EX to 0 can conduct decreasing count on Timer2. When the value in TH2 and TL2 is equal to that of RCAP2H and RCAP2L, the timer overflows. TF2 bit will be set up and 0xFFFFH reloaded into timer register.

No matter whether timer2 overflows or not, bit EXF2 will be used as the 17th bit of the results. Under such operating mode, EXF2 is no longer taken as interrupt flag.



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

Operating Mode 2: Baud Rate Generator

Configure TCLK and RCLK in T2CON register to select Timer2 as baud rate generator. The baud rate of receiver and transmitter can be different. If Timer2 is taken as either one between receiver and transmitter, Timer1 will be taken as another.

Configure TCLK and RCLK in T2CON register to make Timer2 in baud rate generator mode. Such mode is similar to automatic reload mode

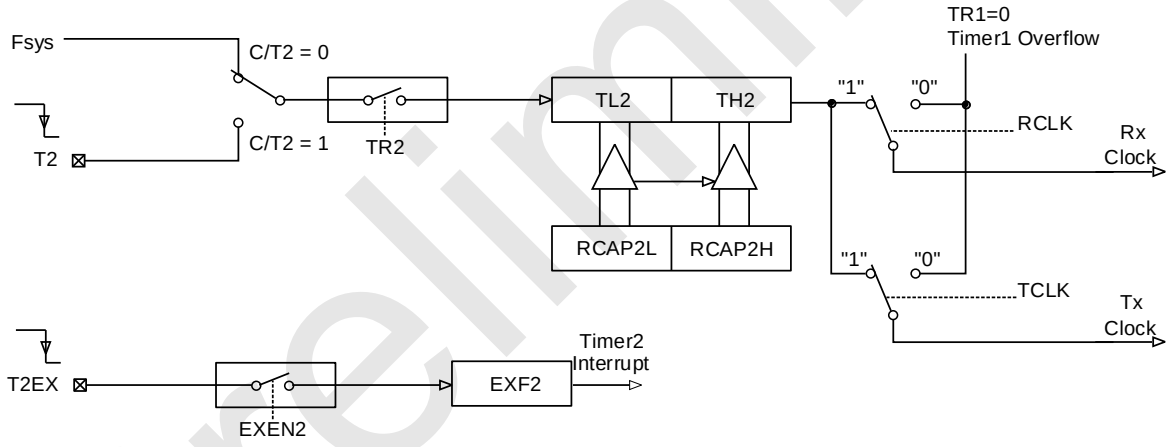
Overflow of Timer2 can make the value in registers RCAP2H and RCAP2L reloaded into the Timer2 and counting, but no interrupt will occur.

If EXEN2 is set to 1, the falling edge on T2EX pin will be set up EXF2 without a reloading. Therefore, when Timer2 is taken as baud rate generator, T2EX can be taken as an additional external interrupt

The baud rate of UART mode 1 and mode 3 depends on overflow rate of Timer2 and the following formula:

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

The schematic diagram of Timer2 as baud rate generator is shown as follows:



Mode 2: Baud Rate Generator

Operating Mode 3: Programmable Clock Output

In this mode, T2 can be programmed to output a 50% duty cycle clock: when C/$\overline{T2}$ = 0 and T2OE = 1, Timer2 is taken as clock generator

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

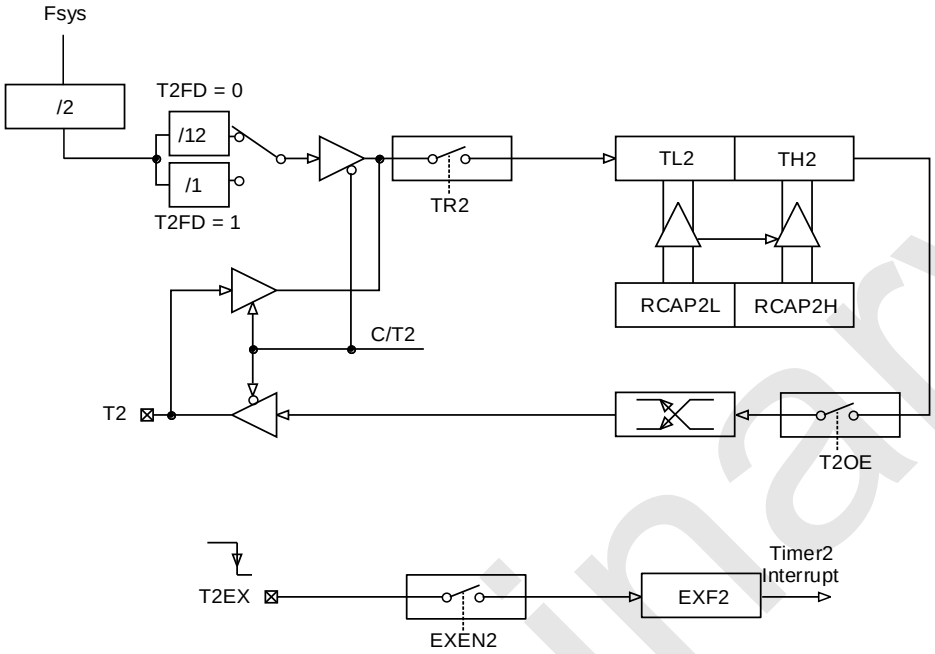
$$\text{Colck Out Frequency} = \frac{\text{fn2}}{(65536-[\text{RCAP2H,RCAP2L}])\times 4};$$

Including, fn2 is the clock frequency of Timer2

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

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

Overflow of Timer2 does not generate an interrupt, T2 pin is taken as clock output.



Operating Mode 3: Programmable Clock Output

Note:

1. Both TF2 and EXF2 can generate interrupt request of Timer2, both of which has the same interrupt vector;

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

3. When EA = 1 and ET2 = 1, setting up TF2 or EXF2 to 1 can arouse interrupt of Timer2;

4. When Timer2 is taken as baud rate generator, the value written in TH2/TL2 or RCAP2H/RCAP2L may influence the accuracy of baud rate and thus result in error of communication.

12 Multiplier-Divider Unit (MDU)

The SC92F848X 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)

<table>
<tr><td></td><td></td><td>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 completed. This bit is only valid for writing 1.</td></tr>
<tr><td>6</td><td>MD</td><td>Multiplier and divider selection bit

0: Multiply operation, writing of multiplicand and multiplier and reading of product are shown below:
<table>
<tr><td>Byte
Operations</td><td>Byte 3</td><td>Byte 2</td><td>Byte 1</td><td>Byte 0</td></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><td>Byte
Operations</td><td>Byte 3</td><td>Byte 2</td><td>Byte 1</td><td>Byte 0</td></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}.**

13 PWM

The SC92F848X provides an independent counter, which is able to support 6-channel PWM output: PWM0 ~ 5.

The SC92F848X PWM has the following functions:

① 10-bit precision;

② Output can be configured in forward or reverse direction;

③ Independent mode and complementary mode:

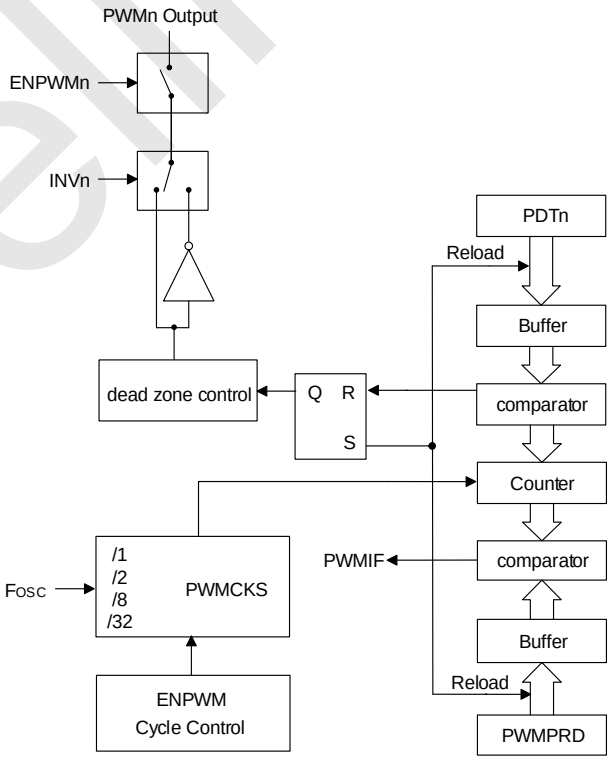
1) In independent mode, PWM0 ~ 5 shared the same clock cycle, but the duty cycle of each PWM channel can be configured separately;

2) In complementary mode, three pairs of complementary PWM waveform with dead zone can be output simultaneously;

④ Provide one PWM overflow interrupt.

The cycle and duty cycle of the SC92F848X PWM is adjustable. Registers PWMCFG, PWMCON controll PWM status and cycle as well as opening of each channel of PWM and duty cycle of output waveform can be adjusted separately.

13.1 PWM block Diagram



The SC92F848X PWM block Diagram

13.2 PWM-related Registers

Mnemonic	Add	Description	7	6	5	4	3	2	1	0	POR
PWMCFG	D1H	PWM Configuration Register	PWMCKS[1: 0]		INV5	INV4	INV3	INV2	INV1	INV0	00000000b
PWMCON	D2H	PWM Control Register	ENPWM	PWMIF	ENPWM5	ENPWM4	ENPWM3	ENPWM2	ENPWM1	ENPWM0	00000000b
PWMPRD	D3H	PWM Period Setting Register	PWMPRD[9: 2]								00000000b
PWMDTYA	D4H	PWM Duty Cycle Configuration Register A	PWMPRD[1: 0]		PDT2[1: 0]		PDT1[1: 0]		PDT0[1: 0]		00000000b
PWMDTY0	D5H	PWM0 Duty Cycle Configuration Register	PDT0[9: 2]								00000000b
PWMDTY1	D6H	PWM1 Duty Cycle Configuration Register	PDT1[9: 2]								00000000b
PWMDTY2	D7H	PWM2 Duty Cycle Configuration Register	PDT2[9: 2]								00000000b
PWMDTYB	DCH	PWM Duty Cycle Configuration Register B	PWMMOD	-	PDT5[1: 0]		PDT4[1: 0]		PDT3[1: 0]		0x000000b
PWMDTY3	DDH	PWM3 Duty Cycle Configuration Register /PWM Dead zone Configuration Register	PDT3[9: 2]								00000000b
PWMDTY4	DEH	PWM4 Duty Cycle Configuration Register	PDT4[9: 2]								00000000b
PWMDTY5	DFH	PWM5 Duty Cycle Configuration Register	PDT5[9: 2]								00000000b

IE1	A9H	Interrupt Enable register 1	-	-	-	ETK	EINT2	EBTM	EPWM	ESSI	xxx00000b
IP1	B9H	Interrupt Priority Control Register 1	-	-	-	IPTK	IPINT2	IPBTM	IPPWM	IPSSI	xxx00000b

13.3 PWM General Configuration Registers

The SC92F848X PWM working mode is divided into independent mode and complementary mode. The registers shared by these two modes are as follows:

The user can select PWM clock source from 4 options by configuring PWMCFG[7: 6]. INV0 ~ 5 is used to select if PWM0 ~ 5 output is in reverse direction. PWMPRD [9: 0] is the six-channel PWM shared period configuration controller. When PWM counter counts to the preset value of PWMPRD[9: 0], this counter will skip to 00_h when next PWM CLK comes. That is to say, PWM0-5 period is (PWMPRD [9: 0] + 1) *PWM clock. Users can set PWM0 ~ 5 shared period by configuring PWMPRD [7: 0] and PWMDTYA [7: 6].

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

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

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

Bit Number	Bit Mnemonic	Description
1	**EPWM**	PWM Interrupt Control Bit 0: Clear to disable the PWM interrupt 1: Set to enable the interrupt when PWM counter overflows

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

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

Bit Number	Bit Mnemonic	Description
1	**IPPWM**	PWM interrupt priority selection bit 0: Clear to configure PWM interrupt priority as “low” 1: Set to configure PWM interrupt priority as “ high”

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

Bit Number	7	6	5	4	3	2	1	0
Bit Mnemonic	ENPWM	PWMIF	ENPWM5	ENPWM4	ENPWM3	ENPWM2	ENPWM1	ENPWM0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7	**ENPWM**	PWM module switch control bit (Enable PWM) 1: Enable Clock to enter PWM unit and PWM starts to work 0: PWM unit stops operating and PWM counter resets to zero. PWMn

		still connects to output pin. If using other functions multiplexed with PWMn output pin, set ENPWMn to 0
6	**PWMIF**	PWM interrupt flag When PWM counter overflows (that is to say, the figure exceeds PWMPRD), this bit will be automatically set to 1 by hardware. If at this time IE1[1] (EPWM) is set to 1 as well, PWM interrupt generates. **Note: Six PWMs share the same period and the same PWM interrupt vector.**
5 ~ 0	**ENPWMx** **(x=0 ~ 5)**	PWMx functional switch control bit 0: PWMx do not output to IO 1: PWMx output to IO

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

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	PWMCKS[1: 0]		INV5	INV4	INV3	INV2	INV1	INV0
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	**PWMCKS[1: 0]**	PWM clock source selector 00: f_{HRC} 01: $f_{HRC}/2$ 10: $f_{HRC}/8$ 11: $f_{HRC}/32$

5 ~ 0	**INVx** **(x=0 ~ 5)**	PWMx output reverse control bit 0: PWMx output not invert 1: PWMx output reverse

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

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

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

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

Bit Number	Bit Mnemonic	Description
7 ~ 6	**PWMPRD[9: 0]**	PWM0 ~ PWM5 shared period configuration bit This figure represents period of PWM0 ~ PWM5 output waveform subtract 1; that is to say, period of PWM output is (PWMPRD [9: 0] + 1)* PWM clock;

PWMDTYB (DCH) PWM Duty cycle Configuration Register B (Read/Write)

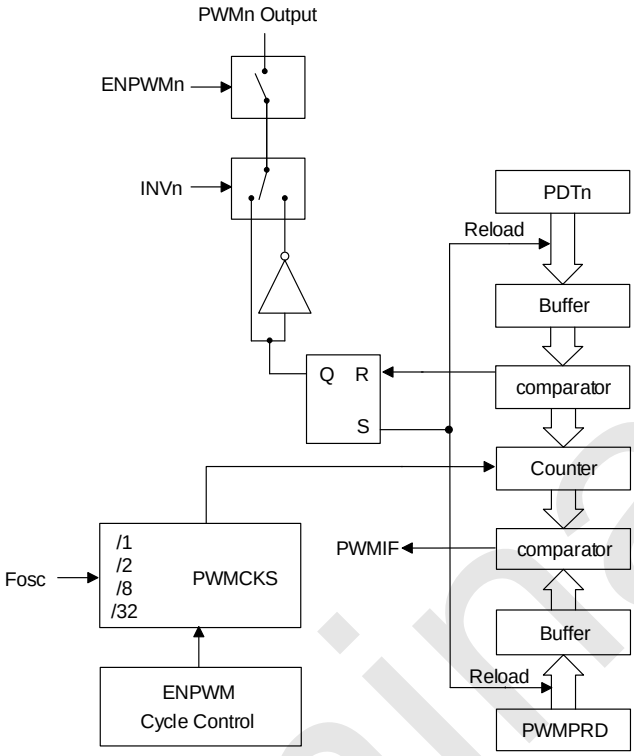
Bit Number	7	6	5	4	3	2	1	0
Bit Mnemonic	PWMMOD	-	PDT5[1: 0]		PDT4[1: 0]		PDT3[1: 0]	
R/W	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	x	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7	**PWMMOD**	PWM Mode Configuration Bit 0: Independent Mode: PWM0 ~ 5 duty configuration 1: Complementary Mode: PWM0/3, PWM1/4, PWM2/5. The output pulse width of the same group of PWM is the same, which is separately controlled by PDT0 ~ 2[9: 0] and dead zone is configured by register PDT3. **Note: When ENPWM = 1, PWM module is enabled. When ENPWMn = 0, PWM output is closed and used as GPIO. At this time, PWM module can be used as a 10-bit Timer, and PWM will still generate interrupt if EPWM (IE1.1) is set to 1.**

13.4 PWM Independent Mode

In independent mode (PWMMOD = 0), PWMDTY0 ~ 5, PWMDTYA and PWMDTYB can be used as duty cycle configuration registers of PWM0 ~ 5. The user shall configure PWM output status and period and corresponding duty cycle registers of PWM channel to output PWM waveform with fixed duty cycle output.

13.4.1 PWM Independent Mode Diagram



The SC92F848X PWM Independent Mode Block Diagram

13.4.2 PWM Independent Mode Duty cycle Configuration

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

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

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

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

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

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

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

PWMDTY3 (DDH) PWM3 Duty Cycle Configuration Register (Read/Write)

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

PWMDTY4 (DEH) PWM4 Duty Cycle Configuration Register (Read/Write)

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

PWMDTY5 (DFH) PWM5 Duty Cycle Configuration Register (Read/Write)

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

Bit Number	Bit Mnemonic	Description

7 ~ 0	**PDTx[9: 2]** **(x=0 ~ 5)**	Independent Mode: PWMx duty cycle length configuration of high 8 bits; High level width of PWMx is (PDTx[9: 0]) PWM clocks.

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

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

PWMDTYB (DCH) PWM Duty Cycle Configuration Register (Read/Write)

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	PWMMOD	-	PDT5[1: 0]		PDT4[1: 0]		PDT3[1: 0]	
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
5 ~ 0	**PDTx [1: 0]** **(x=0 ~ 5)**	PWMx duty cycle length configuration of low 2 bits; High level width of PWMx is (PDTx[9: 0]) PWM clocks.

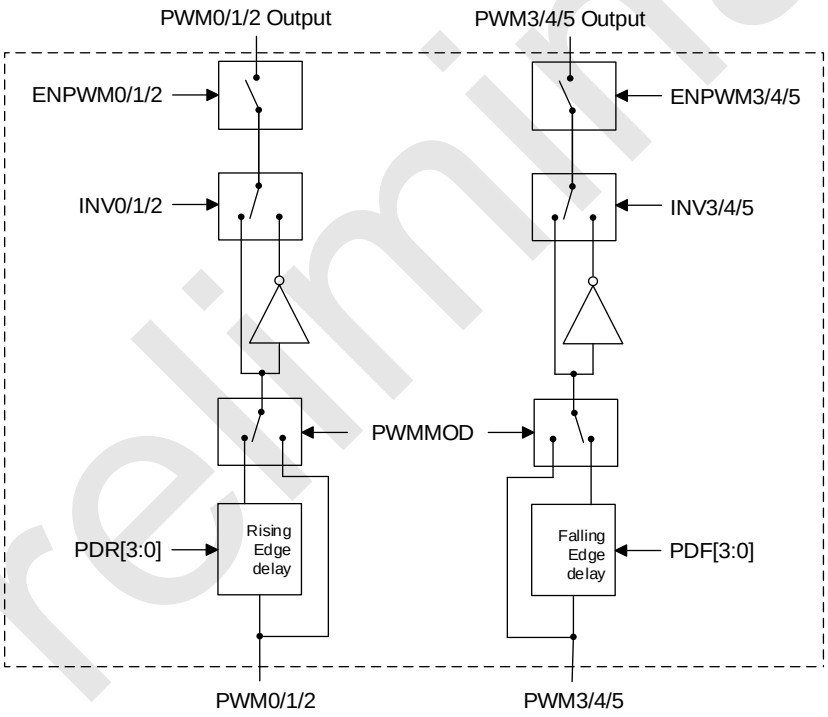
13.5 PWM Complementary Mode

When the SC92F848X PWM works in complementary mode, the dead-time control module can prevent the overlap of valid time zones between two complementary output channel PWM signals, so as to guarantee that a pair of complementary power switches tube driven by PWM signals will not work at the same time in practical applications.

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

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

13.5.1 PWM Complementary Mode Diagram



The SC92F848X PWM Complementary Mode Block Diagram

13.5.2 PWM duty cycle Configuration in complementary mode

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

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

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

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

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

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

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

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

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

Bit Number	Bit Mnemonic	Description
5 ~ 0	**PDTx [1: 0]** **(x=0 ~ 3)**	PWMx duty cycle length is configured as low 2 bits; High level width of PWMx is (PDTx[9: 0]) PWM clocks

13.5.3 PWM Dead Zone Time Configuration in Complementary Mode

PWMDTY3 (DDH) PWM Dead zone Configuration Register (Read/Write)

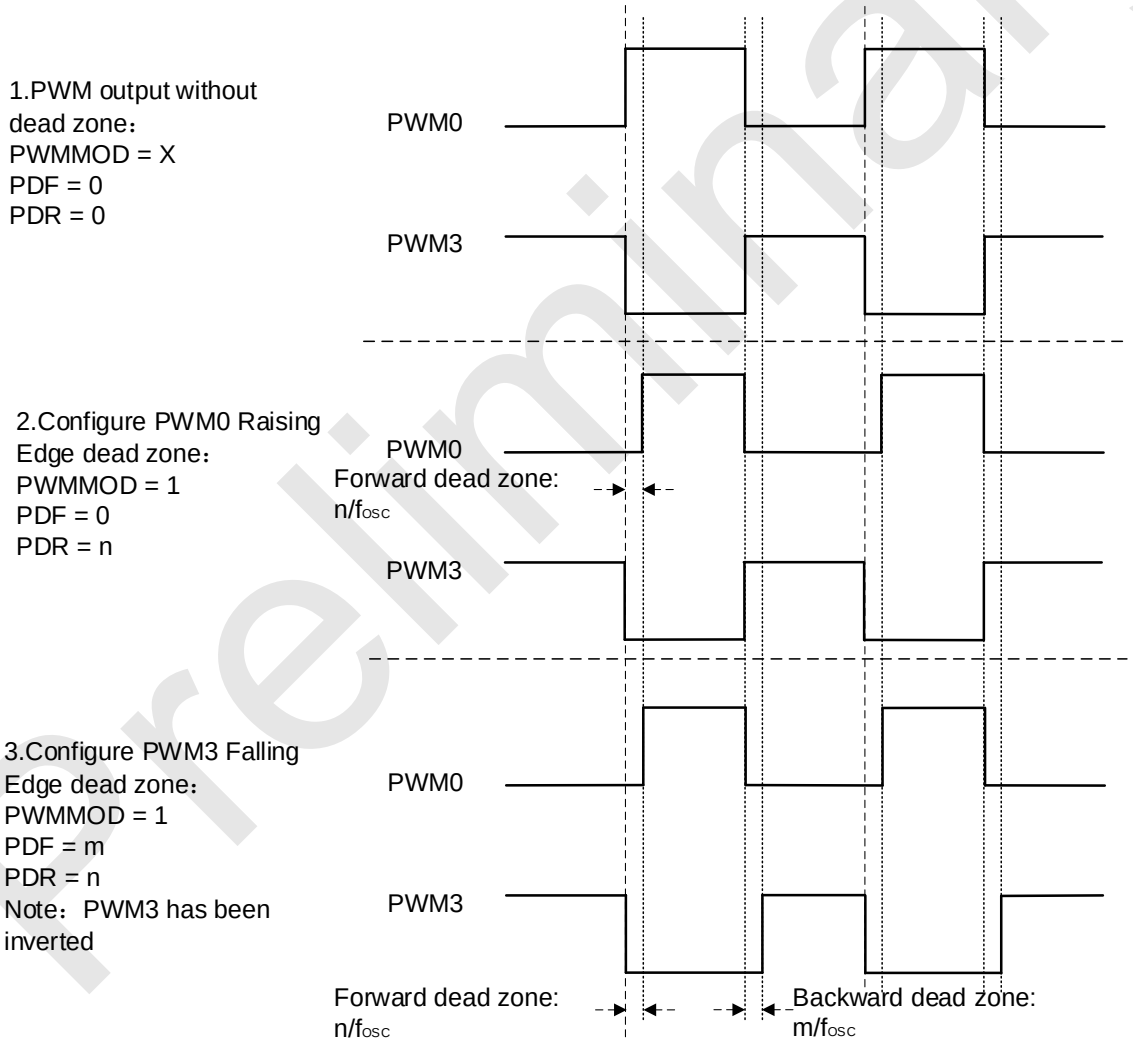
Bit Number	7	6	5	4	3	2	1	0
Bit Mnemonic	**PDF[3: 0]**				**PDR[3: 0]**			
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7 ~ 4	**PDF[3: 0]**	Complementary model:

		PWM3/4/5 falling edge dead zone = PDF [3: 0]/ f_{HRC}
3 ~ 0	**PDR[3: 0]**	Complementary model: PWM0/1/2 rising edge dead zone = PDR [3: 0]/ f_{HRC}

13.5.4 PWM Dead-time Output Waveform

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



Waveform of PWM Output with Dead Zone

13.6 PWM Waveforms and Directions

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

① Diagram for Duty Cycle Change features

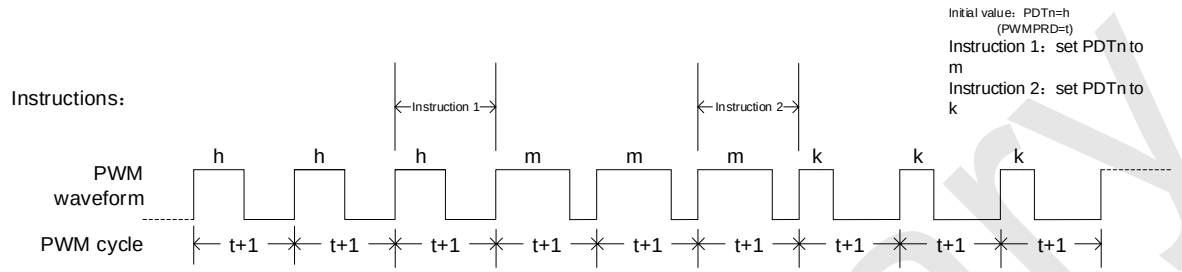


Diagram for Duty Cycle Change Features

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

② Period Change features

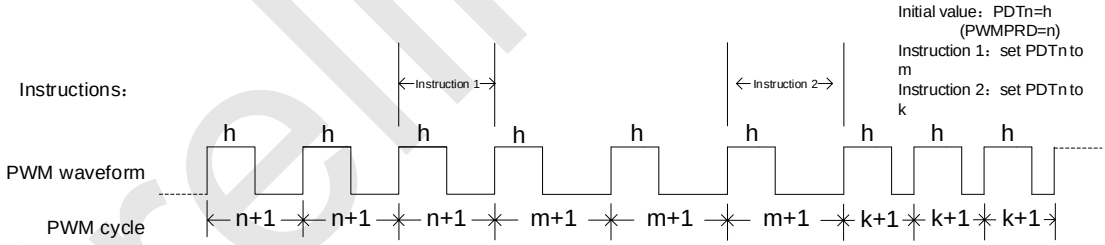


Diagram for Period Change Features

When PWMn outputs waveform, if it is required to change the period, the user can change the value of period configuration registers PWMPRD. Same as changing the duty cycle, change the value of PWMPRD will not change the period immediately. It is required to wait until the end of this period and change in the next period. Refer to the figure above.

③ Relationship between Period and Duty cycle

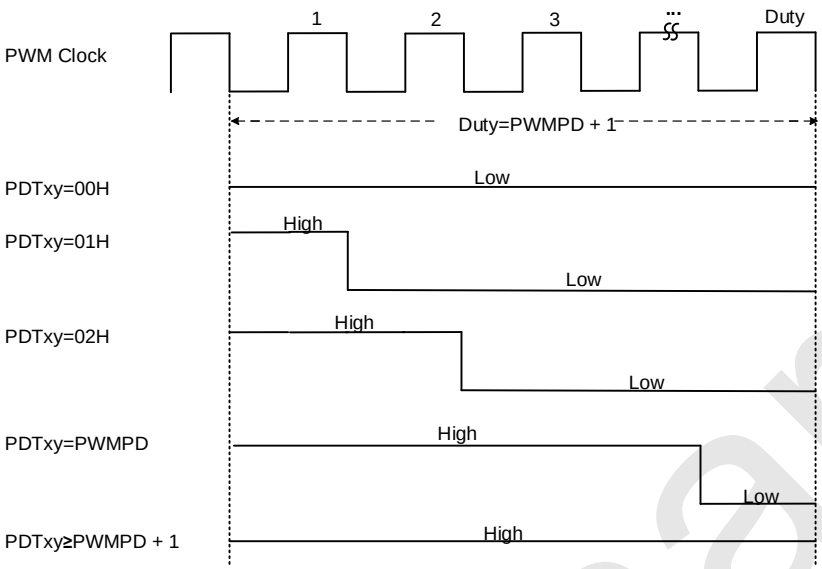


Diagram for Relationship between Period and Duty cycle

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

14 General-purpose I/O (GPIO)

The SC92F848X 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, including P0.0 ~ P0.4 can be used as LCD COM driver by configuring output voltage as 1/2 V_{DD}. 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.

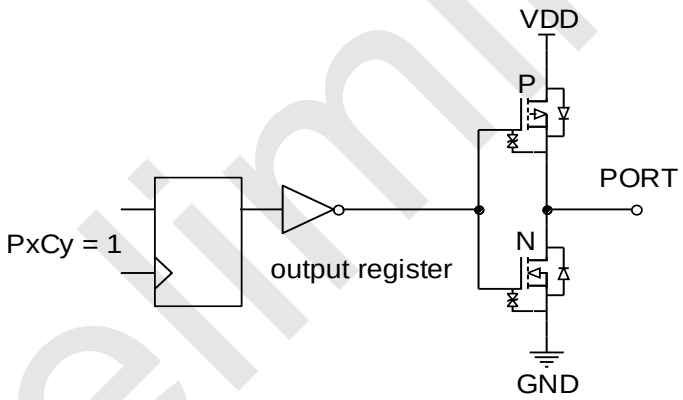
Note: Unused IO port or IO port with no package pin shall be configured as strong push-pull output mode.

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 10mA and low output for the current larger than 50mA

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

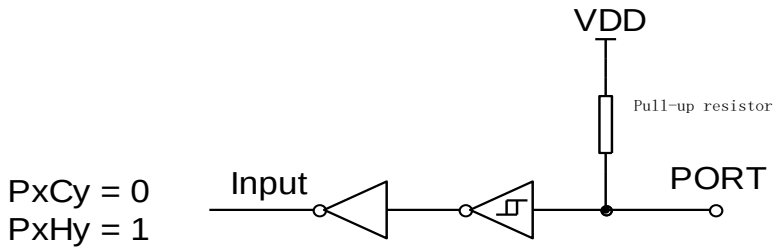


Strong Push-pull Output Mode

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.

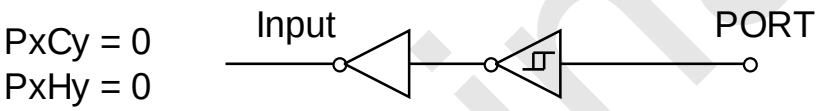
The port structure diagram for pull-up input mode is shown below:



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 Mnemonic	P2H7	P2H6	P2H5	P2H4	P2H3	P2H2	P2H1	P2H0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

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, y=0~7 ; x=5,y=0~1）**	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** **（ x=0~2, y=0~7 ; x=5,y=0~1）**	Px port pull-up resistance configuration, only valid when PxCy=0: 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 Software LCD Driver

The P0.0 ~ P0.4 of the SC92F848X can be used as the COM port of the software LCD. In addition to the normal IO functions, these IOs can also output 1/2VDD. The user can select the corresponding IO as the LCD driver COM according to the usage.

15.1 Software LCD Drives-related Registers

LCD Driver Related SFR Register Description:

P0VO (9CH) P0 port LCD voltage output register (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Mnemonic	-	-	-	P04VO	P03VO	P02VO	P01VO	P00VO
R/W	-	-	-	R/W	R/W	R/W	R/W	R/W
POR	x	x	x	0	0	0	0	0

P0yVO (y=0 ~ 4)	P0y selection output port
0	Ordinary IO port
1	The output voltage of P0y port is 1/2V_{DD}.

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

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

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

Bit Number	Bit Mnemonic	Description
3 ~ 2	**VOIRS[1: 0]**	Selection bits of voltage dividing resistance of LCD voltage output port (suitable driving according to LCD screen size) 00: Disable internal voltage divider resistor. (Energy saving) 01: Set the internal partial resistance to 12.5K 10: Set the internal partial resistance to 37.5K 11: Set the internal partial resistance to 87.5K

16 Serial Interface 0 (UART0)

The SC92F848X 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.

16.1 UART0 Related Register

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 3, 9th bit of receiving data
2	**RB8**	Only valid for mode 3, 9th bit of receiving data
1	**TI**	Transmission interrupt flag bit
0	**RI**	Reception interrupt flag bit

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

Bit Number	7	6	5	4	3	2	1	0
Bit 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	n	x	0	0

Bit Number	Bit Mnemonic	Description
7	**SMOD**	● SM0~1 = 01(UART0 mode 1) or SM0~1 = 11(UART0 mode 3), baud rate multiplying power configuration bit: ■ 0: Serial port operates under clock of 1 system clock ■ 1: Serial port operates under clock of 1/16 system clock ● SM0 ~ 1 = 00(UART0 mode 0), baud rate multiplying power configuration bit: ■ 0: Serial port operates under clock of 1/12 system clock ■ 1: Serial port operates under clock of 1/4 system clock

16.2 Baud Rate of Serial Communication

In mode 0, baud rate can be programmed as 1/12 or 1/4 of system clock and determined by SMOD (PCON.7) bit. When SMOD is set 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 mode 1 and mode 3, baud rate can be programmed as 1 or 1/16 of system clock and determined by SMOD (PCON.7) bit. When SMOD is set to 0, the serial port operates in 1 of system clock. When SMOD is set to 1, serial port operates in 1/16 of system clock. After the serial port clock source is determined, timer 1 or timer 2 sets the baud overflow rate:

- When TCLK (T2CON.4) and RCLK (T2CON.5) are set to 0, configure Timer1 as baud rate generator, 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： $\text{BaudRate} = \frac{\text{fsys}}{[\text{TH1,TL1}]}$; **(Note: [TH1, TL1] must be larger than 0x0010)**
  - SMOD = 1： $\text{BaudRate} = \frac{1}{16} * \frac{\text{fsys}}{[\text{TH1,TL1}]}$;
- When either TCLK(T2con. 4) or RCLK(T2Con. 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： $\text{BaudRate} = \frac{\text{fsys}}{[\text{RCAP2H,RCAP2L}]}$; **(Note: [TH1, TL1] must be larger than 0x0010)**
  - SMOD = 1： $\text{BaudRate} = \frac{1}{16} * \frac{\text{fsys}}{[\text{RCAP2H,RCAP2L}]}$;

17 SPI/TWI/UART Serial Interface (SSI)

The SC92F848X 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)

Special note:

SC92F848 series 8-pin package ： SSI supports only UART function.

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

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

Bit Number	Bit Mnemonic	Description
7 ~ 6	**SSMOD[1: 0]**	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;

17.1 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.

17.1.1 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: TXE=1, interrupt is not enable 1: TXE=1, it will generate SPI interrupt
5 ~ 4, 1	-	Reserved

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

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

17.1.2 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.

17.1.3 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 SC92F848X 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 SC92F848X SPI:

SC92F848X 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 SC92F848X 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:**

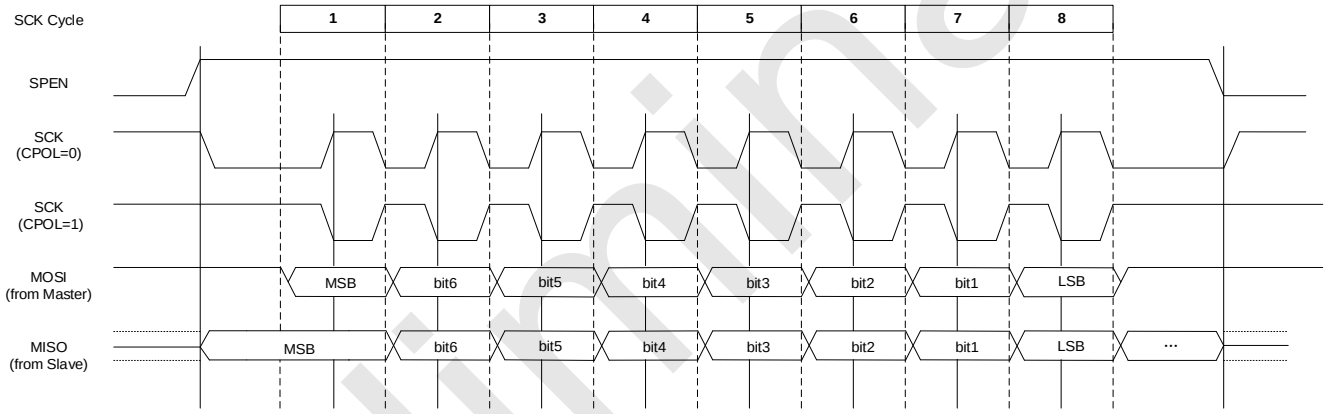
  In slave mode, according to SCK signal controlled by master device, data is moved in via MOSI pin and out via MISO pin. A 1-bit counter records the number of SCK edge. When the receive shift register moves in 8-bit data (one byte) and the transmit shift register moves out 8-bit data (one byte), SPIF 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.

17.1.4 Transfer Form

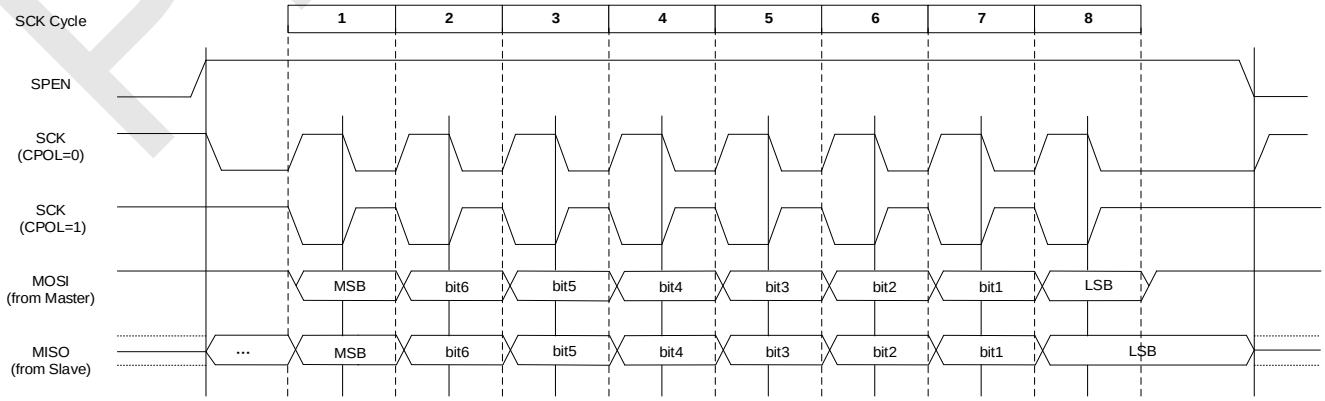
By setting CPOL bit and CPHA bit of SSCON0 register by software, the user can select four combinations of SPI clock polarity and clock phase. CPOL bit defines the polarity of clock, meaning the level status when idle, which has little influence on SPI transmission format. CPHA bit defines the phase of 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

17.1.5 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.

17.2 Two-Wire Interface (TWI)

SSMOD[1: 0] = 10, SSI is configured as TWI interface. The SC92F848X 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/W	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 (8^{th} 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

17.2.1 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.

17.2.2 Operating Modes

TWI communication of the SC92F848X 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:**

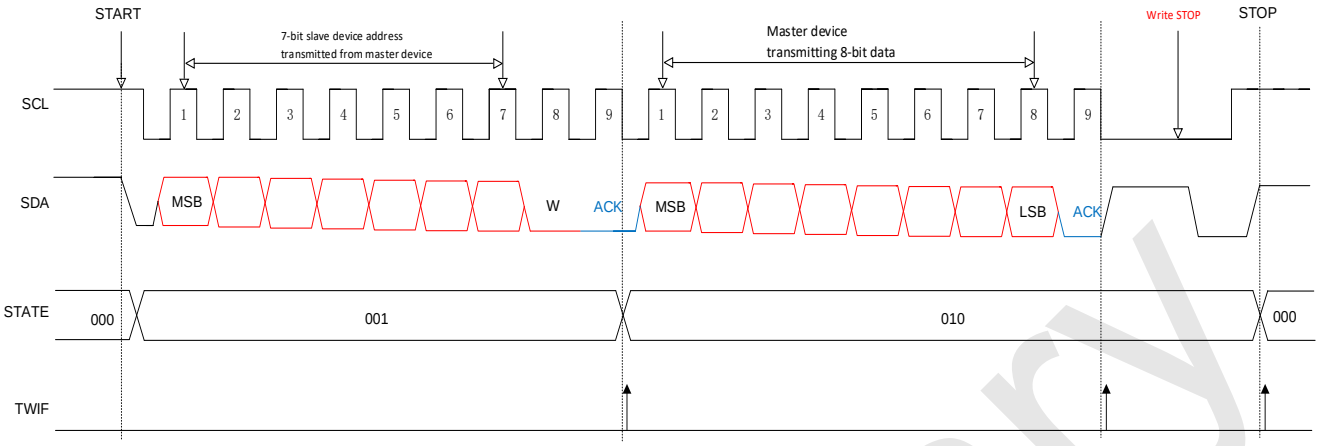
  If the read and write bit received from the first frame is writing (0), the slave device enters into the receiving state of slave device (STATE [2: 0] = 010), and wait for data sent from receiving master device. Master device will release the bus for transmitting every 8 bits and then wait for the response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

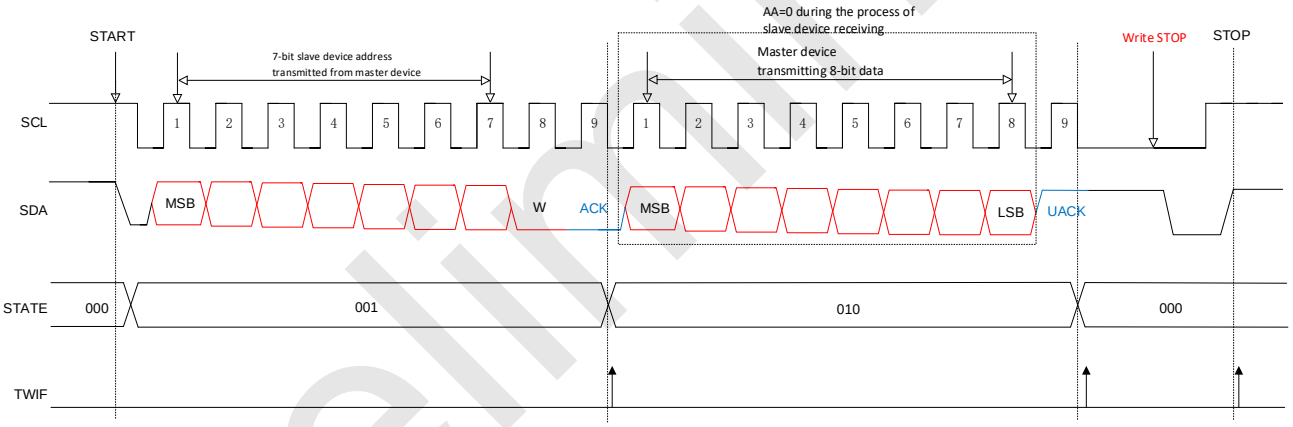
     1) Continue to send data;

     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.



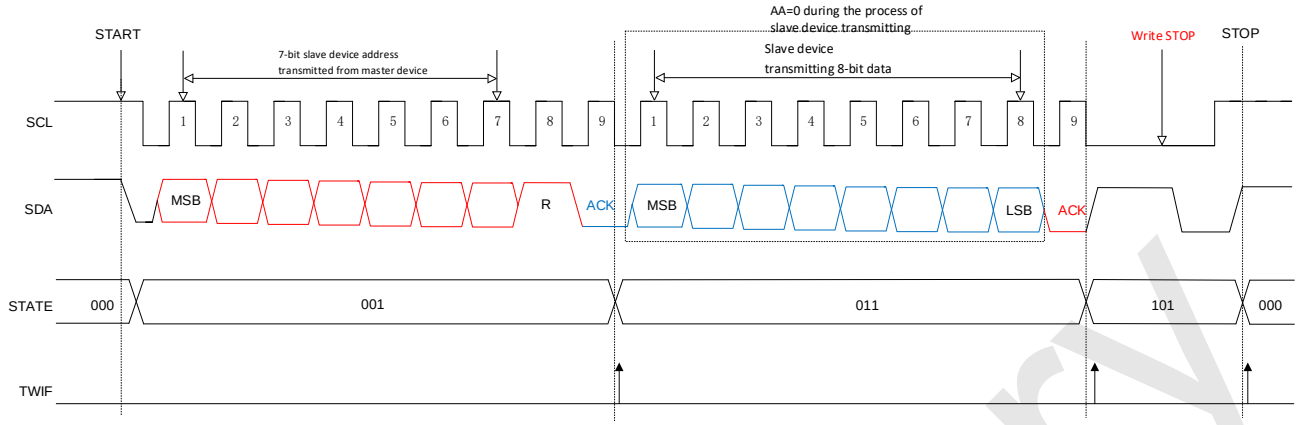
2. If the response state of slave device is in high level (during the receiving process, the value of AA in slave device register is rewritten to 0), it indicates that after transmitting current bytes, the slave device will stop this transmission automatically and return to idle state (STATE[2: 0] =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



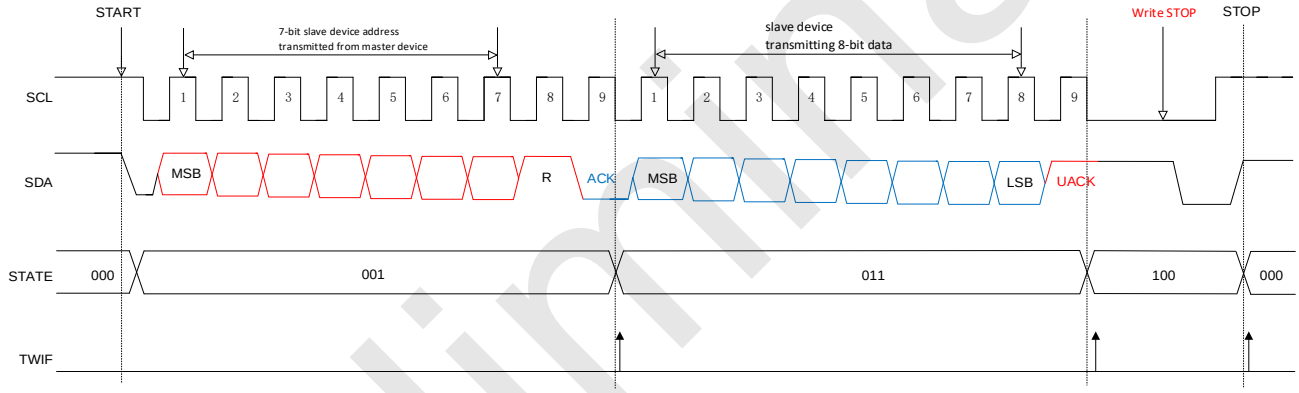
- **Non-general address response, master device transmitting mode:**

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:**

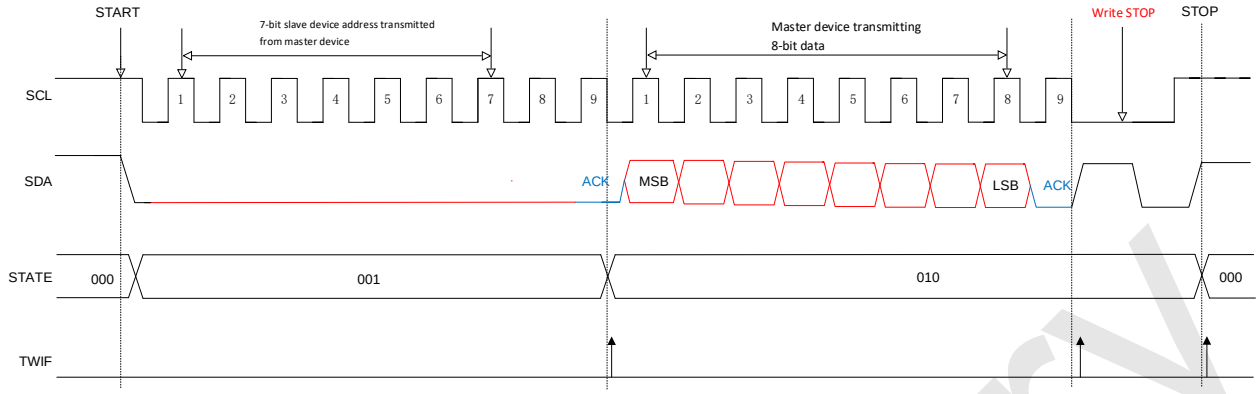
When GC=1, general address is allowed to be used. When the slave device enters into the state of receiving first frame address (STATE[2: 0] = 001), the address bit data received in first frame data will be 0x00, at this time, all slave device will respond the master device. The read and write bit sent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



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.

17.2.3 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.

17.3 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 Mnemonic	SM0	-	SM2	REN	TB8	RB8	TI	RI

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

Bit Number	Bit 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	BAUD1L [7: 0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

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

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

Bit Number	Bit Mnemonic	Description
7 ~ 0	**BAUD1 [15: 0]**	Serial Port Baud Rate Control Bit $\text{BaudRate} = \frac{\text{fsys}}{\text{BAUD1H, BAUD1L}}$ **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	SBUF[7: 0]							

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

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 SBUF1 will return the contents of receiving latch.

18 Analog-to-Digital Converter (ADC)

The SC92F848X has a 12-bit high-precision successive approximation ADC with 11-channel, the external 10 ADC channel is multiplexing with other IO ports. Cooperating with the internal 1.024V, 2.4V and 2.048V reference voltage, one internal channel connected to 1/4 V_{DD} can be used for measuring V_{DD} voltage.

There are 3 options for ADC reference voltage:

① VDD pin (internal V_{DD});

② Precise 1.024V reference output from internal Regulator

③ Precise 2.4V reference output from internal Regulator (at this time, MCU supply voltage VDD can not be lower than 2.9V).

④ Precise 2.048V reference output from internal Regulator

Note: The clock source of ADC follows f_{SYS}.

18.1 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 ~ 11110: Reserved 11111: ADC input is 1/4 V_{DD}, used for measuring power voltage

ADCCFG2 (AAH) 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 Periods Selector 0: Configure ADC sampling time as 6 ADC sampling clock periods 1: Configure ADC sampling time as 36 ADC sampling clock periods The total time of ADC from sampling to conversion T_{ADC} = Sampling time + conversion time . Sampling time is controlled by LOWSP, while conversion time is fixed as: 950ns@f_{HRC} = 32MHz The total time of ADC from sampling to conversion shall be calculated as follows: LOWSP=0: Tadc1= 6/Fadc + 950ns LOWSP=1: Tadc2= 36/Fadc + 950ns
2 ~ 0	**ADCCK[2: 0]**	ADC Sampling Clock Frequency Selector 000: Configure ADC sampling clock frequency Fadc = Fsys/16; 001: Configure ADC sampling clock frequency Fadc = Fsys/12; 010: Configure ADC sampling clock frequency Fadc = Fsys/8; 011: Configure ADC sampling clock frequency Fadc = Fsys/6; 100: Configure ADC sampling clock frequency Fadc = Fsys/4; 101: Configure ADC sampling clock frequency Fadc = Fsys/3; 110: Configure ADC sampling clock frequency Fadc = Fsys/2;

		111: Configure ADC sampling clock frequency Fadc = Fsys/1;
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	-	-	-	-	-	-	EAIN9	EAIN8
R/W	-	-	-	-	-	-	R/W	R/W
POR	x	x	x	x	x	x	0	0

Bit Number	Bit Mnemonic	Description
0~7	**EAINx** **(x=0 ~ 9)**	ADC Port Configuration Register 0: Configure AINx as IO PORT 1: Configure ANIx as ADC input and remove pull-up resistance automatically.

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

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

Bit Number	Bit Mnemonic	Description
7~6	**VREFS[1:0]**	Reference Voltage Selection Bit 00: Configure ADC VREF as V_{DD} 01: Configure ADC VREF as internal correct 1.024 V 10: Configure ADC VREF as internal correct 2.4 V 11: Configure ADC VREF as internal correct 2.048 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	1	1	1	1	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	1	1	1	1	1	1	1	1

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	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
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	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 Interruption Priority Selection Bit 0: Set the interrupt priority of ADC to be "low" 1: Set the interrupt priority of ADC to be "high"

18.2 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.

19 High Sensitivity TouchKey Circuits

The SC92F848X has a built-in 23-channel capacitive touch circuit with high sensitivity mode. Its features are as follows:

1. To adapt to touch keys from a distance, close to the induction of sensitivity to demand higher touch application
2. Dynamic CS testing can be realized through 10V
3. It can realize 23 way touch keys and derivative functions
4. High flexibility development software library support, low development difficulty
5. Automatic debugging software support, intelligent development
6. The touch module can work in low-power mode in THE MCU STOP mode, and the overall power consumption of the chip can be as low as 11uA when the single touch button wakes up

Note: high sensitive touch circuit clock source fixed fHRC = 32 MHZ, do not change with the switch of inside and outside the system clock.

19.1 Power Consumption of TouchKey Circuits

The SC92F848X allows touch scanning to be enabled in STOP Mode: this approach can reduce the overall power consumption of the MCU for touch applications with low-power requirements.

Users can understand that the touch circuit of SC92F848X has two power consumption modes:

1. Normal operation mode

2. Low-power operation mode

The two power consumption modes are defined as follows:

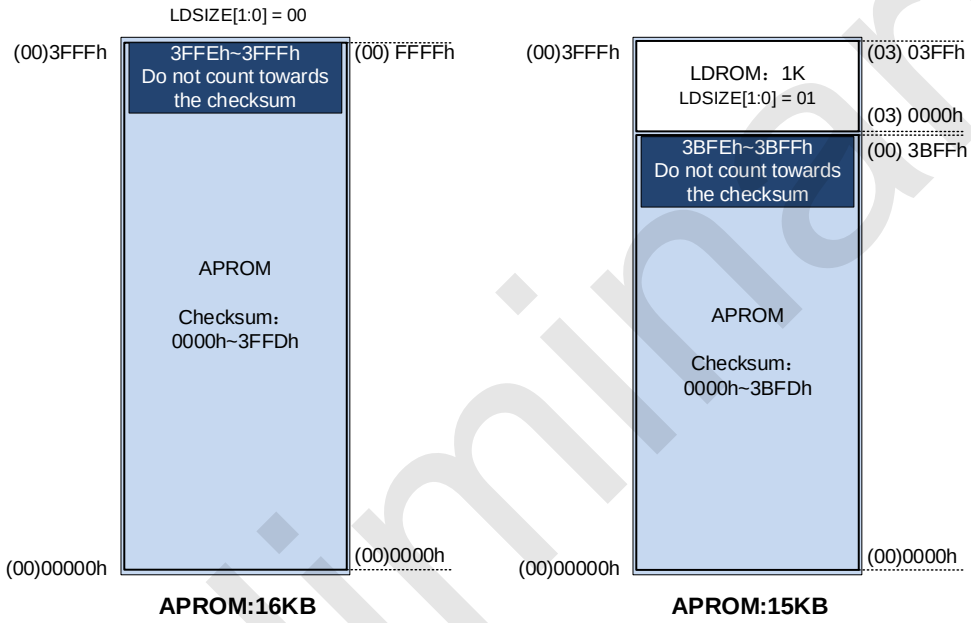
instructions	Normal operation mode	Low-power operation mode
CPU	RUN（Normal mode）	Stop（STOP Mode）
Touch the circuit	RUN	RUN

Note: Users can realize the required touch functions quickly and simply by using the touch button library file provided by SinOne (which can be downloaded from the official website of SinOne).

20 Check Sum Module

SC92F848X has a built-in Checksum module, which can be used to generate the 16-bit Checksum value of the program code in real time. Users can use this Checksum to compare with the theoretical value, and monitor the content of the program area is correct.

The Checksum module calculates the sum of the APROM address contents, but the last two bytes of APROM address are not included in the calculation result. The range of APROM changes with the LDSIZE[1:0] value. Therefore, the Checksum module has a different calculation range.



Note:

1. The Checksum value is the sum of the entire APROM minus the last two bytes.

2. If there are residual values in the address unit after the last operation, the Checksum value is inconsistent with the theoretical value. Therefore, it is recommended that users erase the whole APROM or write zeros before burning the code to ensure that the Checksum value is consistent with the theoretical value.

20.1 Check Sum Result Register

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	5.5	V
Voltage ON any Pin	Input/output voltage of any pin	-0.3	V_{DD}+0.3	V
T_A	Operating temperature	-40	105	℃
T_{STG}	Storage temperature	-55	125	℃
I_{VDD}	Current value flowing through VDD	-	200	mA
I_{VSS}	Current value flowing through VSS	-	200	mA

21.2 Recommended Operating Conditions

Symbol	Parameter	Min Value	Max Value	UNIT	Condition
V_{DD}	Operating Voltage	2.0	5.5	V	32MHz
T_A	Ambient temperature	-40	105	℃	-

21.3 Flash ROM Characteristics

Symbol	Parameter	Min Value	Typical Values	Max Value	UNIT	Condition
N_{END}	Wipe the number	100,000	-	-	Cycles	
T_{DR}	Data Retention Time	100	-	-	Years	TA = +25℃
$T_{S\text{-}Erase}$	Sector Erase Time	-	5	-	ms	TA = +25℃

T_{Write}	Byte Program Time	-	30	-	μs	TA = +25℃

21.4 DC Characteristics

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

Symbol	Parameter	Minimum	Typical value	Maximum	Unit	Test Conditions
Current						
I_{op1}	Operating current	-	3	-	mA	Fsys=32MHz
I_{op2}	Operating current	-	2	-	mA	Fsys=16MHz
I_{op3}	Operating current	-	1.7	-	mA	Fsys=8MHz
I_{op4}	Operating current	-	1.3	-	mA	Fsys=2.66MHz
I_{pd1}	Stand-by current (Power Down Mode)	-	2.5	6	mA	Fsys=32MHz
I_{IDL1}	Stand-by current (IDLE Mode)	-	1.6	-	mA	fsys=32 MHz
I_{BTM}	Base Timer Operating current	-	1.3	3	μA	BTMFS[3: 0]= 1000 Generate an interrupt every 4.0 seconds
I_{WDT}	WDT current	-	1.3	3	μA	WDTCKS[2: 0]= 000

						WDT overflow time 500ms
I_{TK1}	High sensitivity TK operating current	-	0.8	1.2	mA	
IO port characteristics						
V_{IH1}	Input high voltage	0.7VDD	-	VDD+0.3	V	GPIO
V_{IL1}	Input low voltage	-0.3	-	0.3VDD	V	
V_{IH2}	Input high voltage	0.8VDD	-	VDD	V	Schmitt trigger input: RST
V_{IL2}	Input low voltage	-0.2	-	0.2VDD	V	tCK / tDIO UART0 input RX0 SSI signal input port INT0~2 PWM fault detection FLT Timer clock input port Timer capture port
I_{OL1}	Output low current	-	27	-	mA	V_{Pin} = 0.4V
I_{OL2}	Output low current	-	50	-	mA	V_{Pin} = 0.8V
I_{OH1}	Output high current P1、P5	-	10	-	mA	V_{Pin} = 4.3V
I_{OH2}	Output high current P1、P5	-	4.5	-	mA	V_{Pin}=4.7V

I_{OH3}	Output high current P0、P2 @ Pxyz=0,IOH level 0	-	10	-	mA	V_{Pin} = 4.3V
	Output high current P0、P2 @ Pxyz=1,IOH level 1	-	7	-	mA	
	Output high current P0、P2 @ Pxyz=2,IOH level2	-	5	-	mA	
	Output high current P0、P2 @ Pxyz=3,IOH level3	-	2.5	-	mA	
I_{OH4}	Output high current P0、P2 @ Pxyz=0,IOH level0	-	4.5	-	mA	V_{Pin} = 4.7V
	Output high current P0、P2 @ Pxyz=1,IOH level1	-	3	-	mA	
	Output high current P0、P2 @ Pxyz=2,IOH level2	-	2	-	mA	
	Output high current P0、P2 @ Pxyz=3,IOH level3	-	1	-	mA	
R_{PH1}	Pull-up resistor	-	30	-	kΩ	

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

Symbol	Parameters	Min Value	Typical value	Max Value	Unit	Test condition
Current						
I_{op5}	Operating current	-	3	-	mA	Fsys=32MHz
I_{op6}	Operating current	-	2	-	mA	Fsys=16MHz
I_{op7}	Operating current	-	1.7	-	mA	Fsys=8MHz
I_{op8}	Operating current	-	1.3	-	mA	Fsys=2.66MHz
I_{pd2}	Stand-by current(Power Down Mode)	-	2.5	6	μA	
I_{IDL2}	Stand-by current (IDLE Mode)	-	1.6	-	mA	Fsys=32MHz
I_{TK2}	High sensitivity TK operating current	-	0.8	1.2	mA	
IO port characteristics						
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	Schmitt trigger input: RST
V_{IL4}	Input low voltage	-0.2	-	0.2V_{DD}	V	tCK / tDIO UART0 input RX0

						SSI signal input port INT0~2 PWM fault detection FLT Timer clock input port Timer capture port
I_{OL3}	Output low current	-	20	-	mA	V_{Pin}=0.4V
I_{OL4}	Output low current	-	35	-	mA	V_{Pin}=0.8V
I_{OL3}	Output low current	-	20	-	mA	V_{Pin}=0.4V
I_{OL4}	Output low current	-	35	-	mA	V_{Pin}=0.8V
I_{OH5}	Output high current	-	3	-	mA	V_{Pin}=3.0V
R_{PH2}	Pull-up resistor	-	52	-	kΩ	

21.5 AC Characteristics

(V_{DD} = 2.0V ~ 5.5V,T_A = 25℃, Unless otherwise indicated)

Symbol	Parameters	Min Value	Typical Value	Max Value	Unit	Test condition
T_{POR}	Power On Reset time	-	15	-	ms	
T_{PDW}	Power Down mode wake-up time	-	65	130	μs	
T_{Reset}	Reset pulse width	18	-	-	μs	Low level valid

T_{LVR}	LVR buffeting time	-	30	-	μs	
f_{HRC}	RC oscillation stability	31.36	32	32.64	MHz	V_{DD}=2.0~5.5V T_A=-40~105 ℃

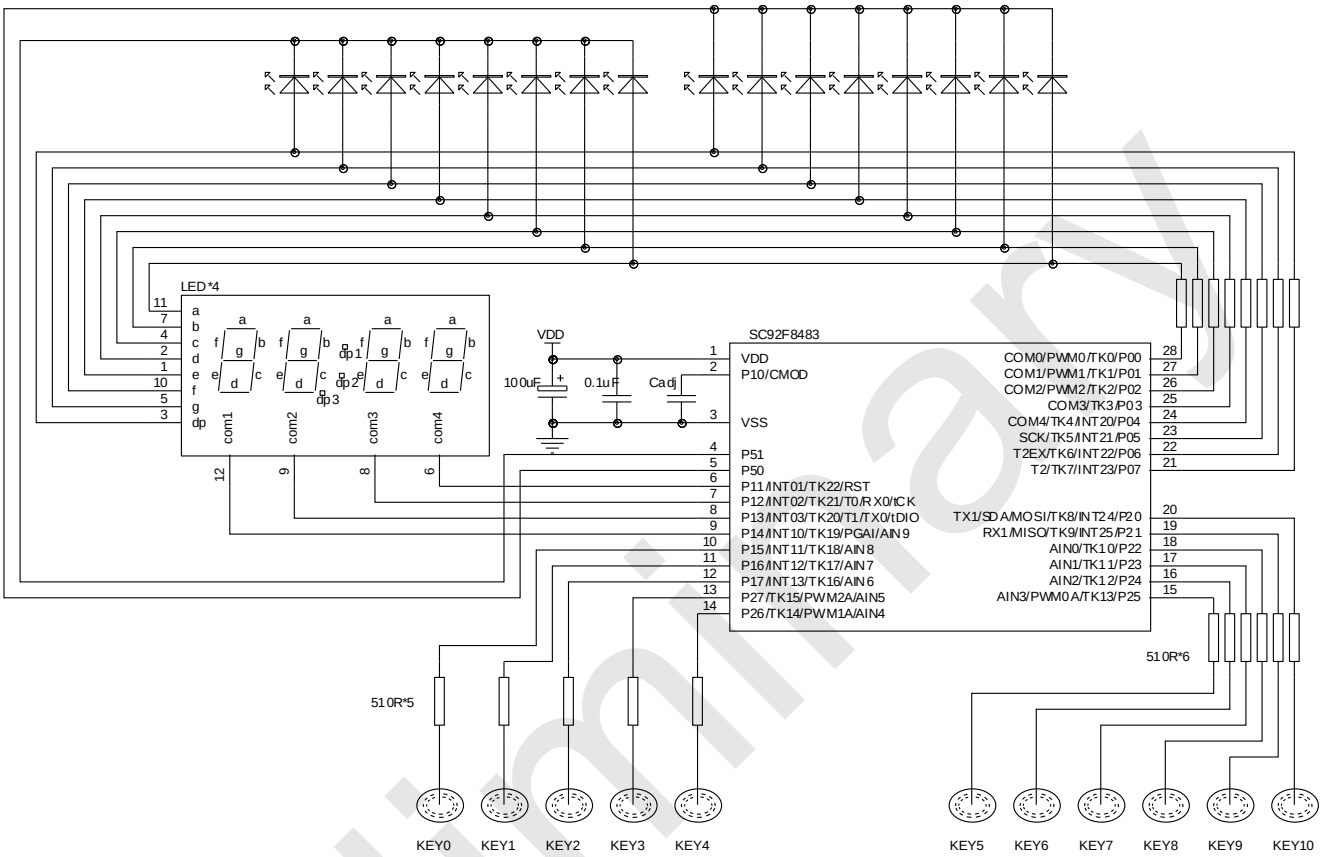
21.6 ADC Characteristics

(T_A = 25℃,Unless otherwise indicated)

Symbol	Parameters	Min Value	Typical Value	Max Value	Unit	Condition
V_{AD1}	Supply voltage 1	2.7	5.0	5.5	V	Vref = 2.4V
V_{AD2}	Supply voltage 2	2.0	5.0	5.5	V	Vref = 1.024V OR Vref = V_{DD}
V_{REF1}	Internal reference 2.4V	1.004	1.024	1.044	V	V_{DD} = 5V OR 3.3V
V_{REF2}	Internal reference 1.024V	2.38	2.40	2.42	V	V_{DD} = 5V OR 3.3V
V_{REF3}	Internal reference 2.048V	2.028	2.048	2.068	V	V_{DD} = 5V OR 3.3V
N_R	Precision	-	12	-	bit	GND≤V_{AIN}≤V_{DD}
V_{AIN}	ADC input voltage	GND	-	V_{DD}	V	
R_{AIN}	ADC input resistance	1	-	-	MΩ	V_{IN}=5V
I_{ADC1}	ADC conversion current 1	-	-	2	mA	ADC module open V_{DD}=5V
I_{ADC2}	ADC conversion current 2	-	-	1.8	mA	ADC module open

						V_{DD}=3.3V
DNL	Differential Non-Linearity	-	-	±3	LSB	V_{DD}=5V V_{REF}=5V
INL	Integral Non-Linearity	-	-	±3	LSB	
E_Z	Offset error	-	±3	-	LSB	
E_F	Full scale error	-	±1	-	LSB	
E_{AD}	Absolute Accuracy	-	±3	-	LSB	
T_{ADC}	ADC conversion time	-	1.2	1.6	μs	f_{HRC} = 32MHz

22 Application circuit



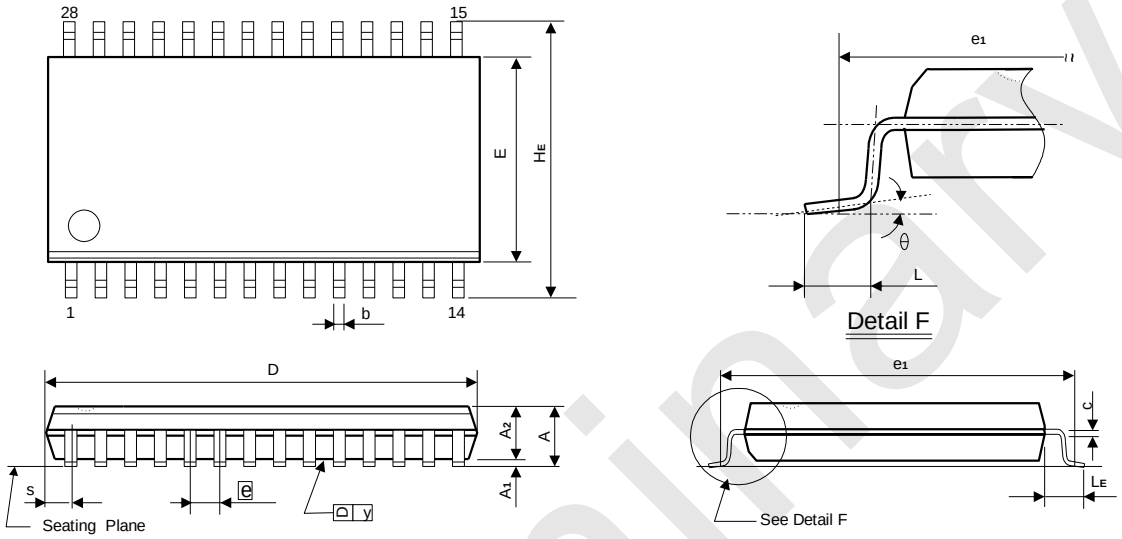
23 Ordering Information

PRODUCT ID	PACKAGE	PACK
SC92F8483M28U	SOP28	TUBE
SC92F8483X28U	TSSOP28	TUBE
SC92F8483Q28R	QFN28	TRAY
SC92F8482M20U	SOP20	TUBE
SC92F8482X20U	TSSOP20	TUBE
SC92F8482Q20R	QFN20	TRAY
SC92F8481M16U	SOP16	TUBE
SC92F8480M08U	SOP8	TUBE

24 Package Information

SC92F8483M28U

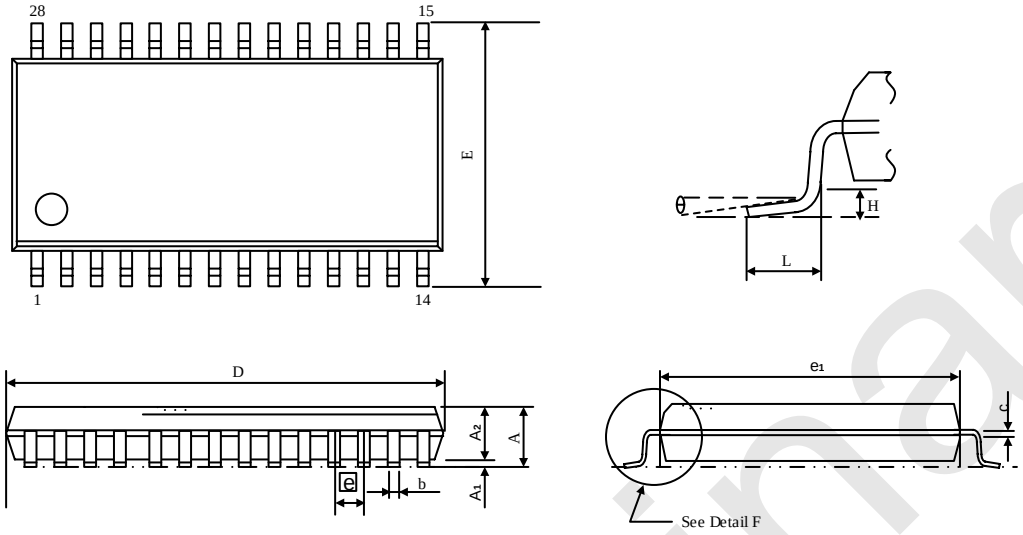
SOP28L(300mil) Overall Dimensions **Unit: mm**



Symbol	mm		
	Min Value	Typical Value	Max Value
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	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°

SC92F8483X28U

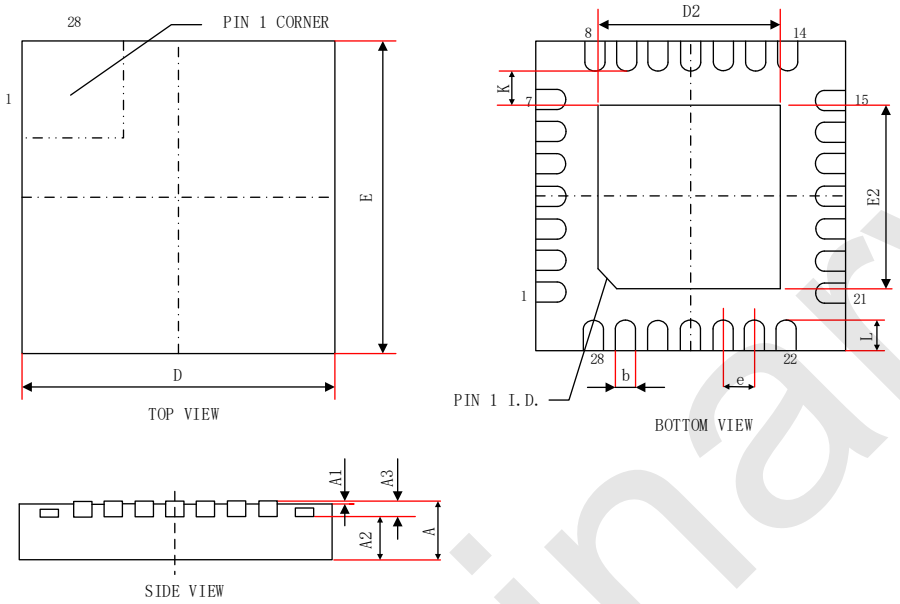
TSSOP28 Overall Dimensions Unit:mm



Symbol	mm		
	Min Value	Typical Value	Max Value
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

SC92F8483Q28R

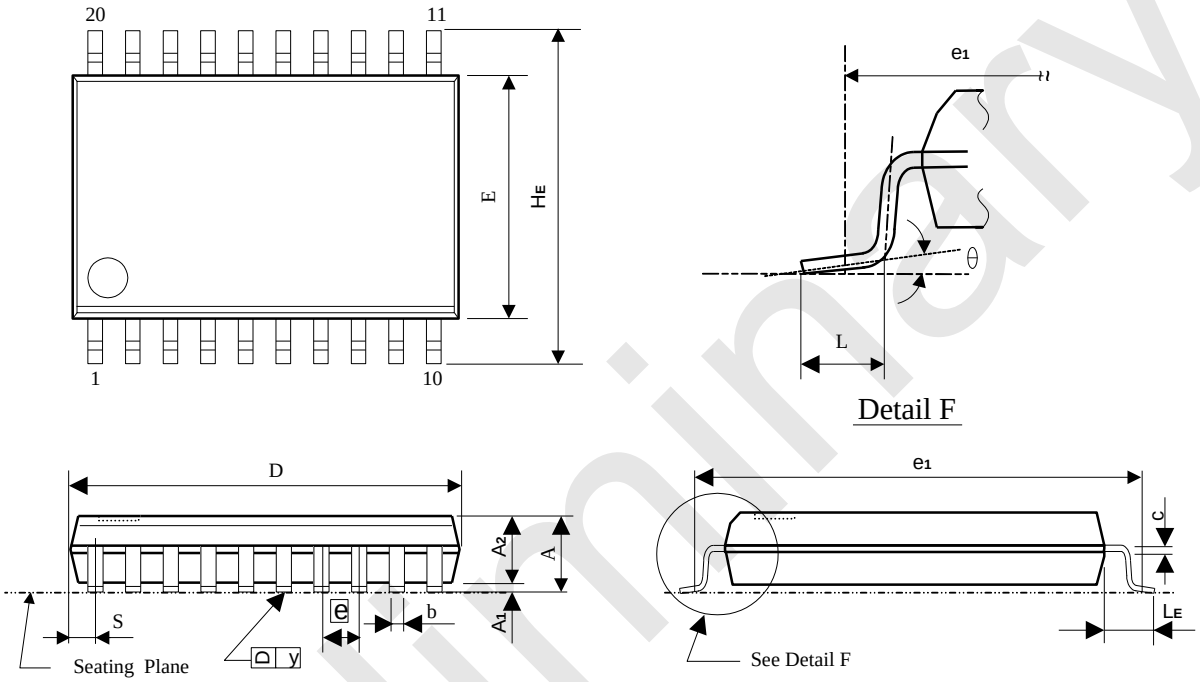
QFN28L(4X4) Overall Dimensions Unit:mm



Symbol	mm		
	Min Value	Typical Value	Max Value
A	0.7	0.75	0.8
A1	0	0.02	0.05
A2	-	0.55	-
A3	0.203 REF		
b	0.18	0.23	0.28
D	4 BSC		
E	4 BSC		
e	0.45 BSC		
D2	2.5	2.6	2.7
E2	2.5	2.6	2.7
L	0.25	0.35	0.45
K	0.35 REF		

SC92F8482M20U

SOP20L(300mil) Overall Dimensions **Unit: mm**

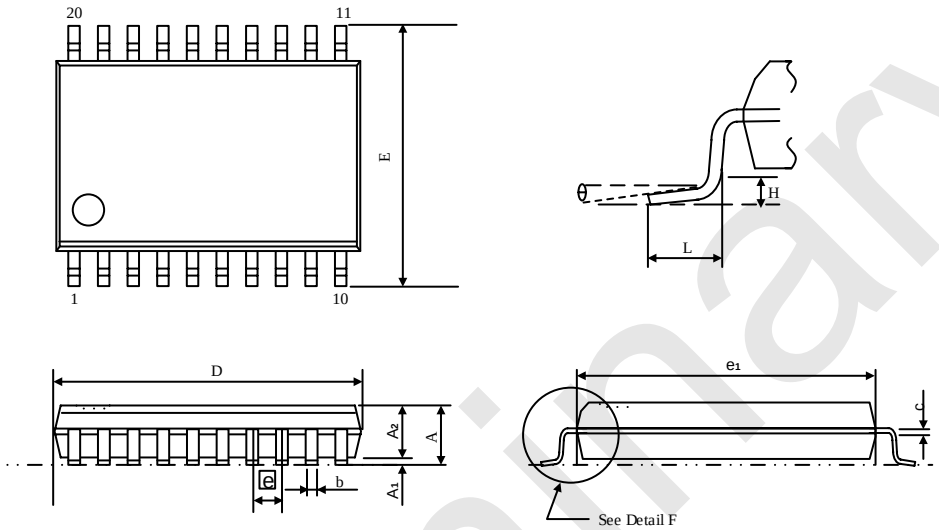


Symbol	mm		
	Min Value	Typical Value	Max Value
A	2.40	2.56	2.65
A1	0.100	0.200	0.300
A2	2.240	2.340	2.440
b	0.35	--	0.47
c	0.25	--	0.31
D	12.60	12.80	13.00
E	7.30	7.50	7.70
HE	10.100	10.300	10.500
e	1.27(BSC)		

L	0.700	0.850	1.000
LE	1.30	1.40	1.50
θ	0°	-	8°

SC92F8482X20U

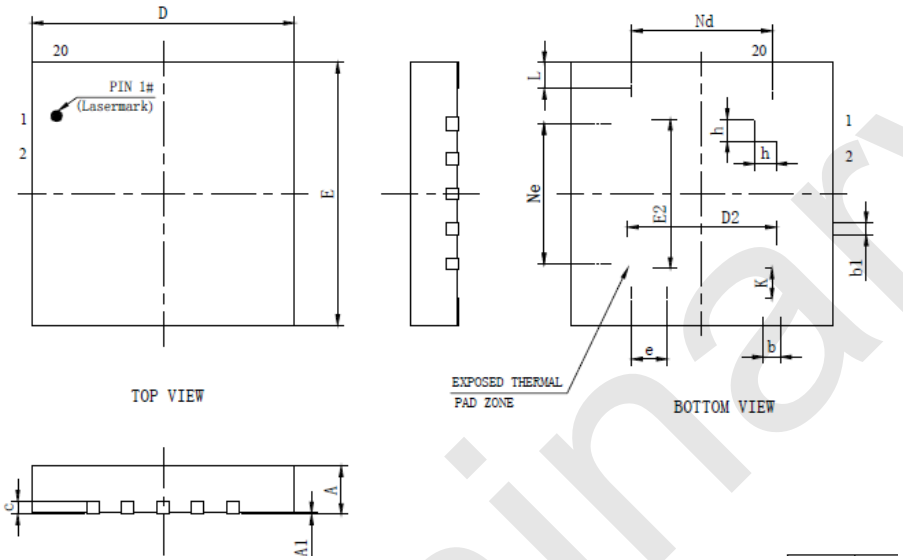
TSSOP20L Overall Dimensions Unit: mm



Symbol	mm		
	Min Value	Typical Value	Max Value
A	-	-	1.200
A1	0.050	-	0.150
A2	0.800	-	1.050
b	0.190	-	0.300
c	0.090	-	0.200
D	6.400	-	6.600
E	6.20	-	6.60
e1	4.300	-	4.500
e	0.65(BSC)		
L	-	-	1.00
θ	0°	-	8°
H	0.05	-	0.15

SC92F8482Q20R

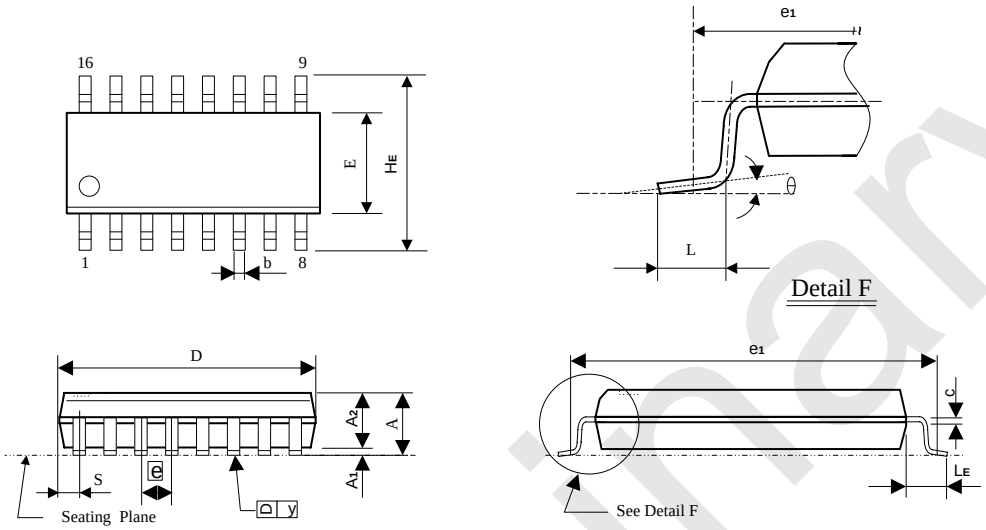
QFN20 L(3*3) Overall Dimensions Unit: mm



Symbol	mm		
	Min Value	Typical Value	Min Value
A	0.50	0.55	0.60
A1	0	0.02	0.05
b	0.15	0.20	0.25
b1	0.14REF		
c	0.15REF		
D	2.90	3.00	3.10
D2	1.60	1.70	1.80
e	0.40BSC		
Ne	1.60BSC		
Nd	1.60BSc		
E	2.90	3.00	3.10
E2	1.60	1.70	1.80
L	0.25	0.30	0.35
h	0.20	0.25	0.30
K	0.30	0.35	0.40

SC92F8481M16U

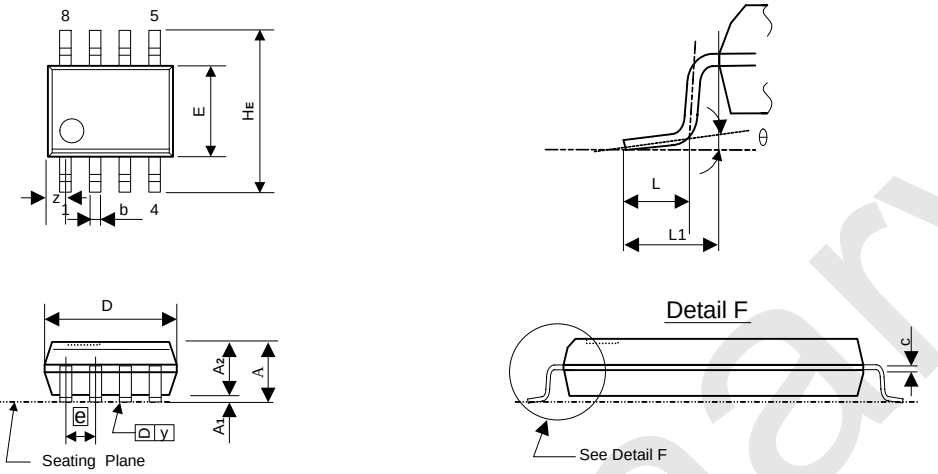
SOP16L(150mil) Overall Dimensions **Unit: mm**



Symbol	mm		
	Min Value	Typical Value	Min Value
A	1.500	1.625	1.750
A1	0.050	0.1375	0.225
A2	1.30	1.45	1.55
b	0.38	0.43	0.48
c	0.20	0.23	0.26
D	9.70	9.90	10.10
E	3.70	3.90	4.10
HE	5.80	6.00	6.20
e	1.27(BSC)		
L	0.50	0.65	0.80
LE	0.95	1.05	1.15
θ	0°	-	8°

SC92F8480M08U

SOP8L(150mil) Overall Dimensions **Unit: mm**



Symbol	mm		
	Min Value	Typical Value	Min Value
A	1.500	1.625	1.750
A**1**	0.100	0.1625	0.225
A**2**	1.30	1.425	1.55
b	0.39	0.435	0.48
c	0.20	0.23	0.26
D	4.70	4.90	5.10
E	3.70	3.90	4.10
HE	5.80	6.00	6.20
e	1.270(BSC)		
L	0.50	0.65	0.80
L1	0.95	1.05	1.15
θ	0°	-	8°

25 Revision History

Revision	Changes	Date
V0.1	Initial Release.	June 2022

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. Cytron considers the information provided to be accurate and reliable. The information in this document will be used in June 2022. In the actual production design, please refer to the latest data manual of each product and other relevant materials.