High-speed 1T 8051-based Flash MCU, 384 bytes SRAM, 8 Kbytes Flash, 128 bytes independent EEPROM, 16-channel dual mode touch circuit, 4-channel 8-bit PWM, 3 Timer/Counters, SSI, CheckSum module

1 General Description

The SC92F8372/8371/8370 (hereinafter referred to as SC92F837X) is an enhanced ultra-high speed 1T 8051 core industrial level integrated touch button function Flash micro controller, command system is fully compatible with the traditional 8051 product series.

SC92F837X has a 16-channel, low-power, dual-mode capacitance touch circuit that can be selected to run in STOP Mode. SC92F837X is integrated with 8 K Bytes Flash ROM, 384 Bytes SRAM, 128 Bytes EEPROM, up to 18 GP I/O, 7 IO external interrupts, 3 16-bit timers, 4 total cycles, duty cycle adjustable PWM, internal 1% high precision high frequency 12/6/2MHz oscillator and 4% precision low frequency 128K oscillator, a UART/SPI/IIC three choice a random SSI and other resources.In order to improve the reliability and simplify the customer circuit, SC92F837X is also integrated with 4 levels of optional voltage LVR and other high reliability circuits. SC92F837X has excellent anti-interference performance and excellent touch button performance, which is very suitable for various applications of touch button and master control, such as the size of smart home appliances and smart home, Internet of things, wireless communication, game consoles and other industrial control and consumer applications.

2 features

Operating Voltage: 2.4V ~ 5.5V

Operating Temperature: -40 ~ 85℃

Package:

SC92F8372 (SOP20/TSSOP20)
SC92F8371 (SOP16)
SC92F8370 (SOP8)

Core: Ultra-speed 1T 8051

Flash ROM: 8 Kbytes Flash ROM (MOVC prohibited addressing 0000H~00FFH) can be rewritten 10,000 times

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

EEPROM: 128 bytes EEPROM can be rewritten 100,000 times. The data written-in has more than 10-year preservation life.

SRAM: Internal 256 bytes + external 128 bytes

System Clock (f_{SYS}):

- Built-in 24 MHz high-speed RC oscillator (f_{HRC}):
  - as a system clock source, f_{SYS} can choose set by programmer for 12/6/2 MHz
  - Frequency Error: Suitable for 2.9V ~ 5.5V and -20 ~ 85℃ application environment, no more than ±1%

Built-in 128 kHz LRC Low-speed Oscillator:

- Available to act as clock source of Base Timer (BTM), which can wake up STOP
- Available to act as clock source of WDT
- Frequency Error: 4.0V ~ 5.5V and -20 ~ 85℃ application environment, no more than ±4% of frequency error after register correction

Low-voltage Reset (LVR):

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

Flash Programming and Emulation:

- 2-wire JTAG programming and emulation interface

Interruption (INT):

- Totally 9 interrupt sources, including Timer0, Timer1, Timer2, INT0, INT2, PWM, SSI, Base Timer, TK
- Two external interrupt vectors shared by 7 external interrupt I/Os, which can be defined in

rising-edge, falling-edge or double-edge trigger mode.

- Two-level interrupt priority capability

Digital Peripheral:

- Up to 18 two-way independently controllable I/O interfaces, able to configure pull-high resistor independently
- P0, P2 4-level control drive capability
- All IOs equipped with sink current drive capability (50mA)
- 11-bit WDT, optional clock division ratio
- 3 standard 80C51 timers: Timer0, Timer1 and Timer 2
- 4 channels common cycle, duty cycle adjustable PWM
- 1 UART/SPI/IIC communication interface (SSI)

Analog Peripheral:

- 16-channel dual mode touch circuit with low power consumption can be configured for high sensitivity or high reliability mode:

  - High sensitivity model to adapt to touch keys from a distance, close to the induction of sensitivity to demand higher touch application
  - High reliability model has very strong anti-interference, can pass the test of 10 v dynamic CS
  - Supporting 16-channels touch keys and derivative functions
  - Development software library support high flexibility and low development difficult
  - Automatic debugging software support, intelligent development
  - When scanning with a single TouchKey in stop mode, the overall power consumption of the chip can be as low as 11uA

Power Saving Mode:

- IDLE Mode: can be woken up by any interrupt.
- STOP Mode: can be woken up by INT0, INT2, Base Timer and TK.

Naming Rules for 92 Series Products

Name	SC	92	F	8	3	7	1	X	M	20	U
S/R	①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩	⑪

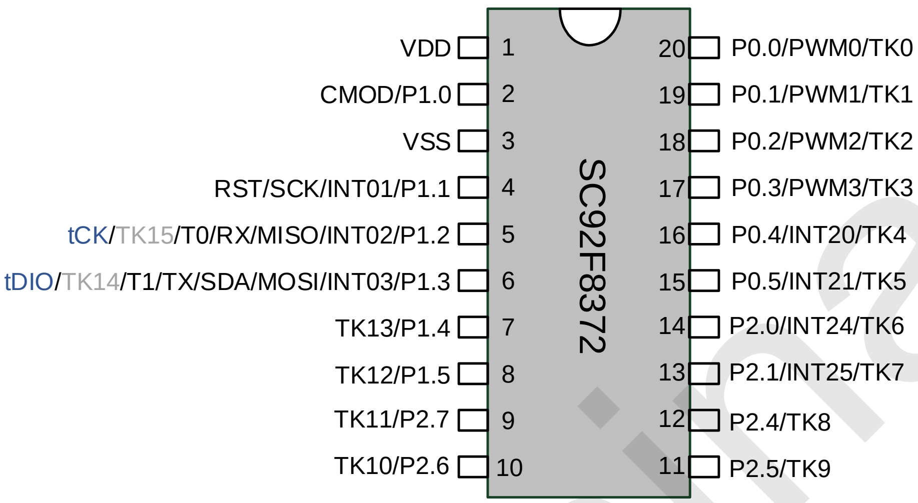
S/R	**Meaning**
①	SinOne Chip abbreviation
②	Name of product series
③	Product Type (F: Flash MCU)
④	Serial Number: 7: GP Series, 8: TK series
⑤	ROM Size: 1 for 2K, 2 for 4K, 3 for 8K, 4 for 16K and 5 for 32K…
⑥	Subseries Number.: 0 ~ 9, A ~ Z
⑦	Number of Pins: 0: 8pin, 1: 16pin, 2: 20pin, 3: 28pin, 5: 32pin, 6: 44pin, 7: 48pin, 8: 64pin, 9: 100pin
⑧	Version Number: (default, B, C, D)
⑨	Package Type: (D: DIP; M: SOP; X: TSSOP; F: QFP; P: LQFP; Q: QFN; K: SKDIP)
⑩	Number of Pins.
⑪	Packaging Mode: (U: Tube; R: Tray; T: Reel)

Contents

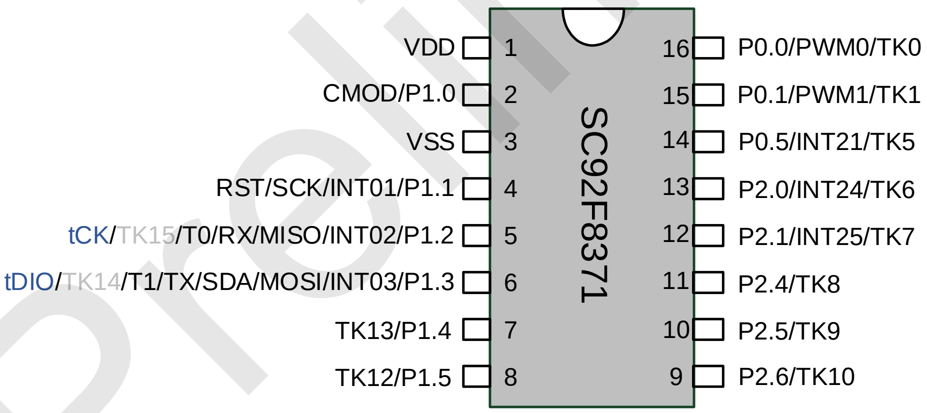
3 Pin Description

3.1 Pin Configuration

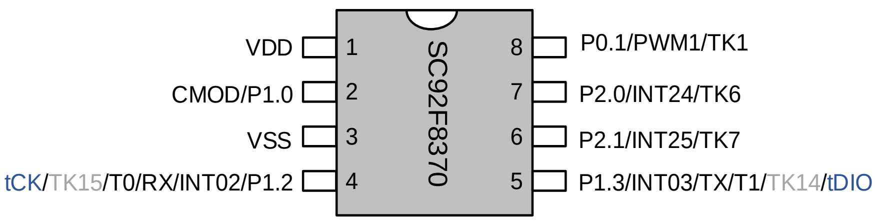
Note: In consideration of multiplexing of TK14/TK15 and TK debugging communication ports of the SC92F837X, if it is required to use the TK debugging function, please avoid using TK14/TK15!



The SC92F8372 Pin Diagram



The SC92F8371 Pin Diagram



The SC92F8270 Pin Diagram

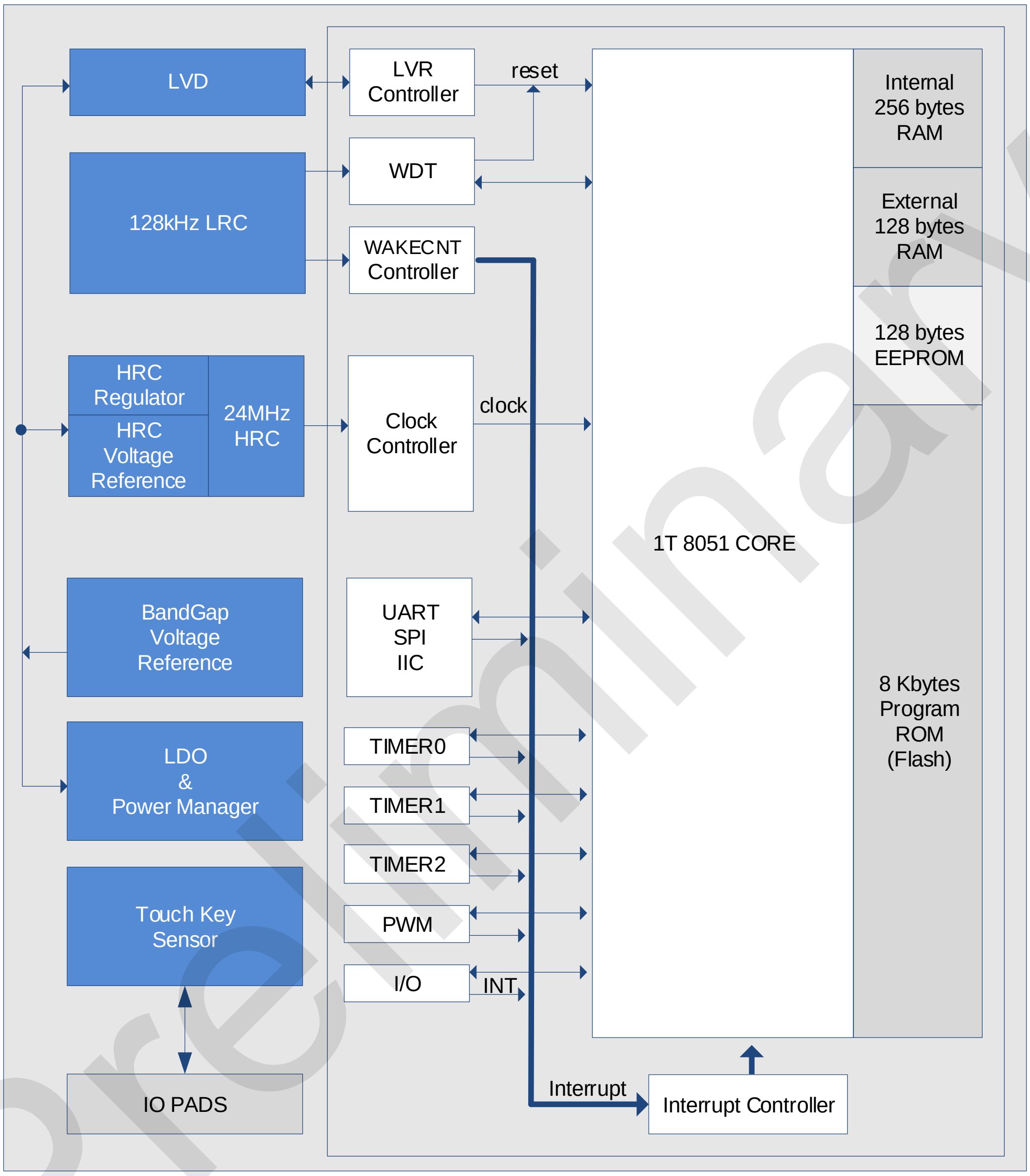
3.2 Pin Definition

Pin number			Pin Name	Type	Description
20PIN	16pin	8pin			
1	1	1	VDD	Power	Power
2	2	2	P1.0/CMOD	I/O	P1.0: GPIO P1.0 CMOD: Touch Key Touch external capacitance
3	3	3	VSS	Power	Ground
4	4	-	P1.1/INT01/SCK/RST	I/O	P1.1: GPIO P1.1 INT01: Input 1 for external interrupt 0 SCK: SPI and TWI RST: Reset pin
5	5	4	P1.2/INT02/MISO/RX/T0/TK15/tCK	I/O	P1.2: GPIO P1.2 INT02: Input 2 for external interrupt 0 MISO: SPI master input slave output RX: UART receive T0: Counter 0 external input TK15: TK channel 15, **if you need to use TK debugging function, please try to avoid using this TK channel!** tCK: Burning and simulation port clock line
6	6	5	P1.3/INT03/MOSI/SDA/TX/T1/TK14/tDIO	I/O	P1.3: GPIO P1.3 INT03: Input 3 for external interrupt 0 MOSI: SPI main output from the input SDA: TWI SDA

					TX: UART send T1: Counter 1 external input TK14: TK channel 14. **If you need to use TK debugging function, please try to avoid using this TK channel!** tDIO: Data line of burning and simulation port
7	**7**	**-**	**P1.4/TK13**	I/O	P1.4: GPIO P1.4 TK13: Channel 13 for TK
8	**8**	**-**	**P1.5/TK12**	I/O	P1.5: GPIO P1.5 TK12: Channel 12 for TK
9	**-**	**-**	**P2.7/TK11**	I/O	P2.7: GPIO P2.7 TK11: Channel 11 for TK
10	**9**	**-**	**P2.6/TK10**	I/O	P2.6: GPIO P2.6 TK10: Channel 10 for TK
11	**10**	**-**	**P2.5/TK9**	I/O	P2.5: GPIO P2.5 TK9: Channel 9 for TK
12	**11**	**-**	**P2.4/TK8**	I/O	P2.4: GPIO P2.4 TK8: Channel 8 for TK
13	**12**	**6**	**P2.1/INT25/TK7**	I/O	P2.1: GPIO P2.1 INT25: Input 5 for external interrupt 2 TK7: Channel 7 for TK
14	**13**	**7**	**P2.0/INT24/TK6**	I/O	P2.0: GPIO P2.0 INT24: Input 4 for external interrupt 2 TK6: Channel 6 for TK
15	**14**	**-**	**P0.5/INT21/TK5**	I/O	P0.5: GPIO P0.5

					INT21: Input 1 for external interrupt 2 TK5: Channel 5 for TK
16			**P0.4/INT20/TK4**	I/O	P0.4: GPIO P0.4 INT20: Input 0for external interrupt 2 TK4: Channel 4 for TK
17			**P0.3/PWM3/TK3**	I/O	P0.3: GPIO P0.3 PWM3: PWM3 outlet TK3: Channel 3 for TK
18			**P0.2/PWM2/TK2**	I/O	P0.2: GPIO P0.2 PWM2: PWM2 outlet TK2: Channel 2 for TK
19	**15**	**8**	**P0.1/PWM1/TK1**	I/O	P0.1: GPIO P0.1 PWM1: PWM1 outlet TK1: Channel 1 of TK
20	**16**	**-**	**P0.0/PWM0/TK0**	I/O	P0.0: GPIO P0.0 PWM0: PWM0 outlet TK0: Channel 0 for TK

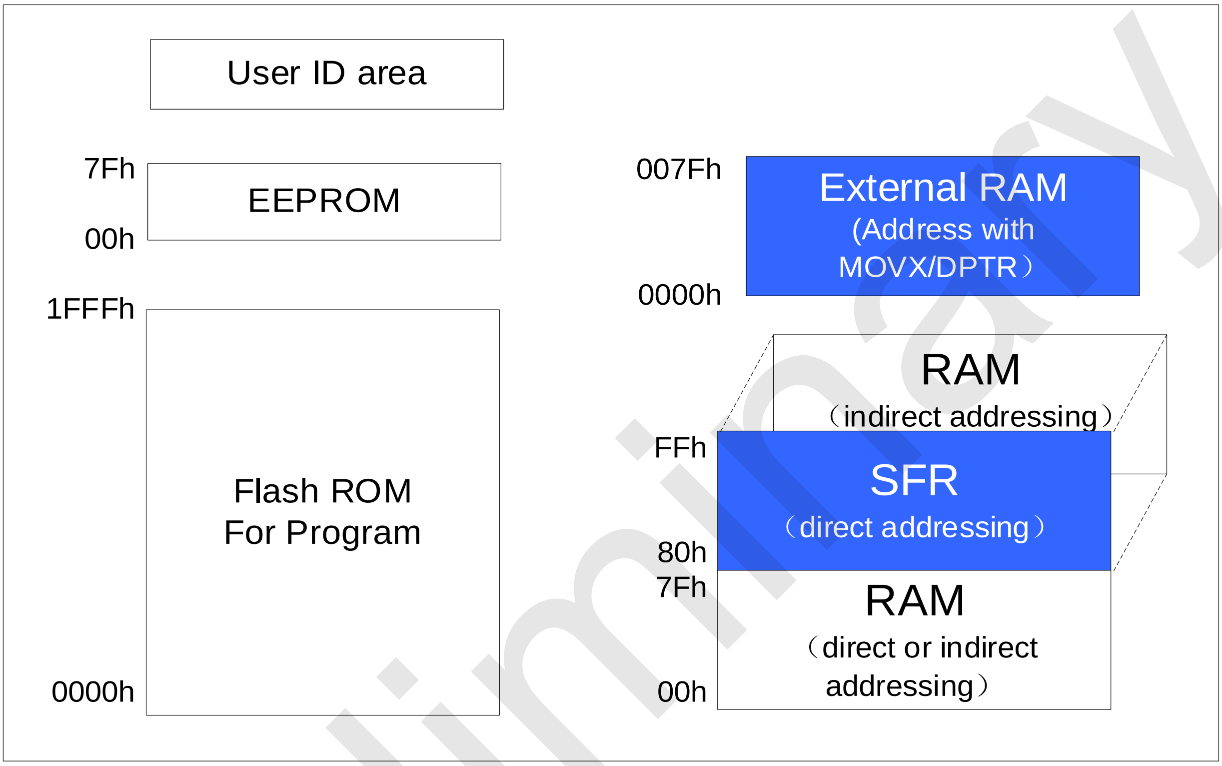
4 Inner BLOCK Diagram



SC92F837X BLOCK DIAGRAM

5 Flash ROM and SRAM Structure

The structures of the SC92F837X's Flash ROM and SRAM are shown as follows:



Flash ROM and SRAM Structure Diagram

5.1 flash rom

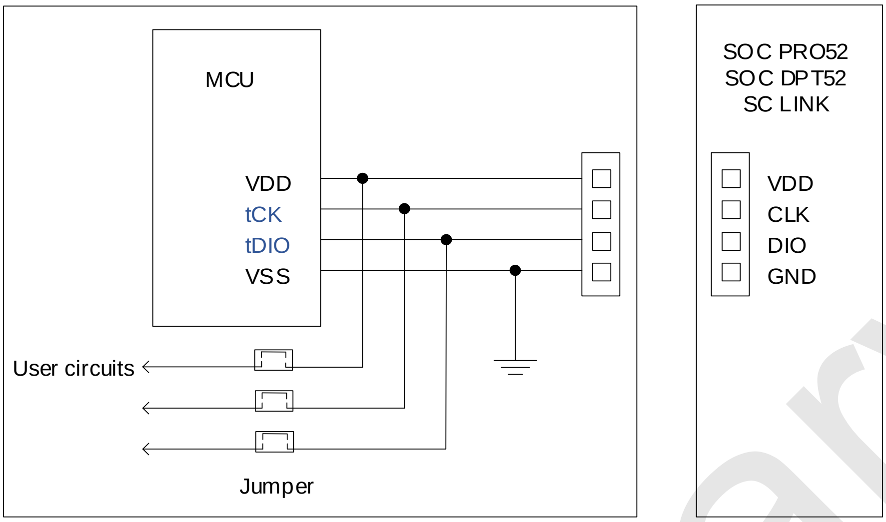
The SC92F837X provides 8 Kbytes of Flash ROM with the ROM address of 0000H ~ 1FFFH. These 8 Kbytes of Flash ROM can be rewritten 10,000 times, which is able to program and erase by specialized ICP programming device (SOC PRO52/DPT52/SC LINK) provided by SinOne. The MOVC instruction is non-addressable within 256 bytes (address of 0000H ~ 00FFH).

EEPROM is a data memory separated from 8K bytes ROM with the address of 00H ~ 7FH, which can be accessed by single-byte reading and writing operations in the program; for more details, refer to 16 EEPROM and IAP Operations.

Note: The EEPROM can be rewritten 100,000 times. Users cannot exceed the limit value, otherwise there will be an exception!

The SC92F837X 8 Kbytes Flash ROM provide Empty Check, Program, Verify and Erase function other than Read function. This Flash ROM and EEPROM usually needs no Erase operation before writing. Directly writing data can realize coverage of new data.

The SC92F837X Flash ROM can be programmed by tDIO, tCK, VDD and VSS, with its specific connection shown as follows:



ICP Mode Flash Writer Programming Connection Diagram

5.2 Customer Option Memory (User Programming Setting)

A separate Flash data memory is embedded inside the SC92F837X, called Code Option area, to save the user's presets. These presets will be written into IC when programming and loaded into SFR as default values during reset.

Option-related SFR Operating Instructions:

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

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

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]**	Internal high-frequency RC frequency adjustment Central value 10000000b corresponds to HRC central frequency, the larger the value is, the faster the frequency will be, vice versa.

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

Bit Number	7	6	5	4	3	2	1	0
Bit Mnemonic	ENWDT	-	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**	Watchdog (WDT) control bit (This bit is transferred by the system to the value set by the user Code Option) 0: WDT invalid

		1: WDT valid (WDT stops counting during IAP execution)
5 ~ 4	**SCLKS[1: 0]**	System clock frequency selection bits 00: Reserved; 01: System clock frequency is HRC frequency divided by 2; 10: System clock frequency is HRC frequency divided by 4; 11: System clock frequency is HRC frequency divided by 12;
3	**DISRST**	IO/RST selection bit 0: configure P1.1 as External Reset input pin 1: configure P1.1 as GPIO
2	**DISLVR**	LVR control bit 0: LVR valid 1: LVR invalid
1 ~ 0	**LVRS [1: 0]**	LVR voltage selection bits 11: 4.3V reset 10: 3.7 V reset 01: 2.9V reset 00: 2.3 V reset
6	**-**	Reserved

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

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

Bit Number	Bit Mnemonic	Description
3 ~ 2	**IAPS[1: 0]**	EEPROM and IAP Area Selection Bits 00: Code memory prohibits IAP operations 01: last 0.5k code memory allows IAP operation 10: Last 1k code memory allows IAP operation (0C00H~0FFFH) 11: All code memory allows IAP operation (0000H~0FFFH)
7~4,1~0	-	Reserved

5.2.1 Customer-Option-related Registers Operation Instructions

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

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

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

For Example: To configure OP_HRCR as 0x01, specific operation method is shown below:

C program example:

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

Assembler program example:

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

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

5.3 SRAM

SC92F837X MCU integrates SRAM with 384 Bytes, which is divided into internal 256 Bytes RAM and external 128 Bytes RAM. The internal RAM ranges from 00H to FFH, with high 128 Bytes (address 80H~FFH) addressing only indirectly and low 128 Bytes (address 00H~7FH) addressing either directly or indirectly.

The address of the special function register SFR is also 80H~FFH. The difference between the SFR and the internal high 128 Bytes SRAM is that the SFR register is directly addressed, while the internal high 128 Bytes SRAM can only be indirectly addressed.

The external RAM address is 0000H~07FH, but it needs to be addressed via the MOVX directive.

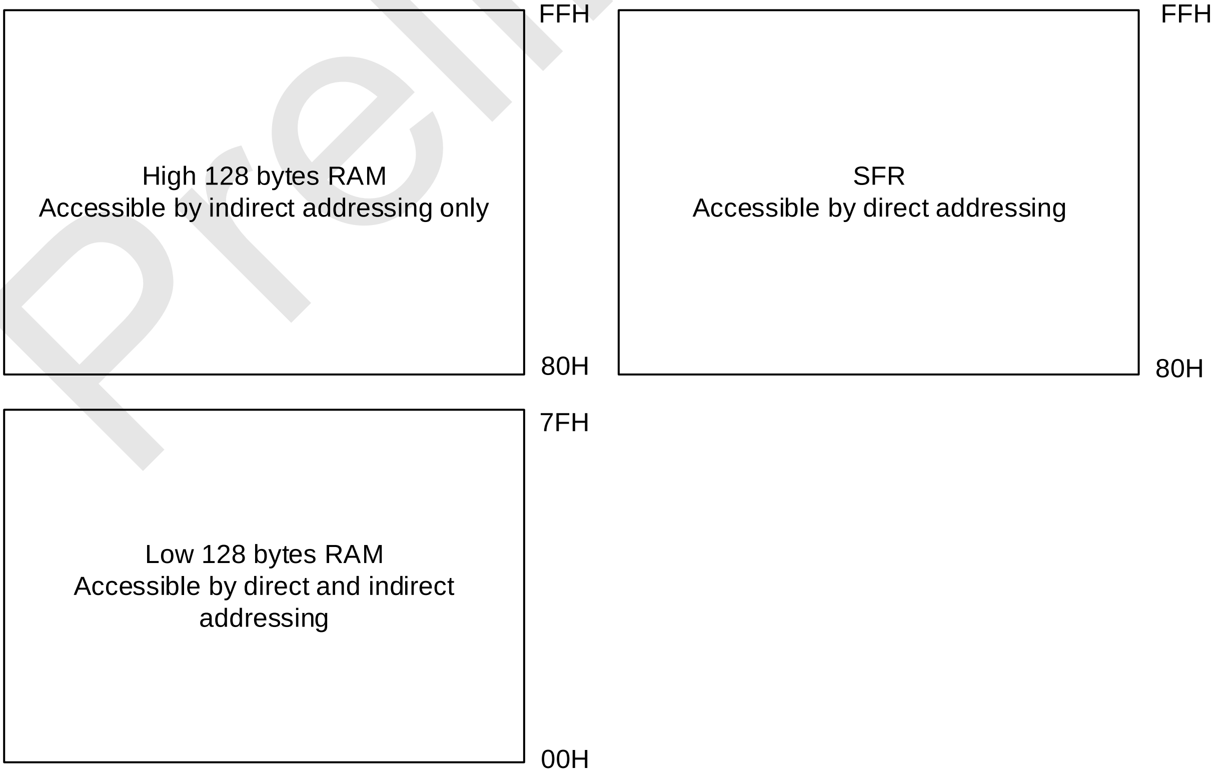
5.3.1 Internal 256 bytes SRAM

Internal low 128 bytes SRAM area is divided into three parts:

1. Register bank 0 ~ 3, address from 00H to 1FH. The RS1 and RS0 of PSW register can select the currently active SFR. Using Register bank 0 ~ 3 can accelerate arithmetic speed;

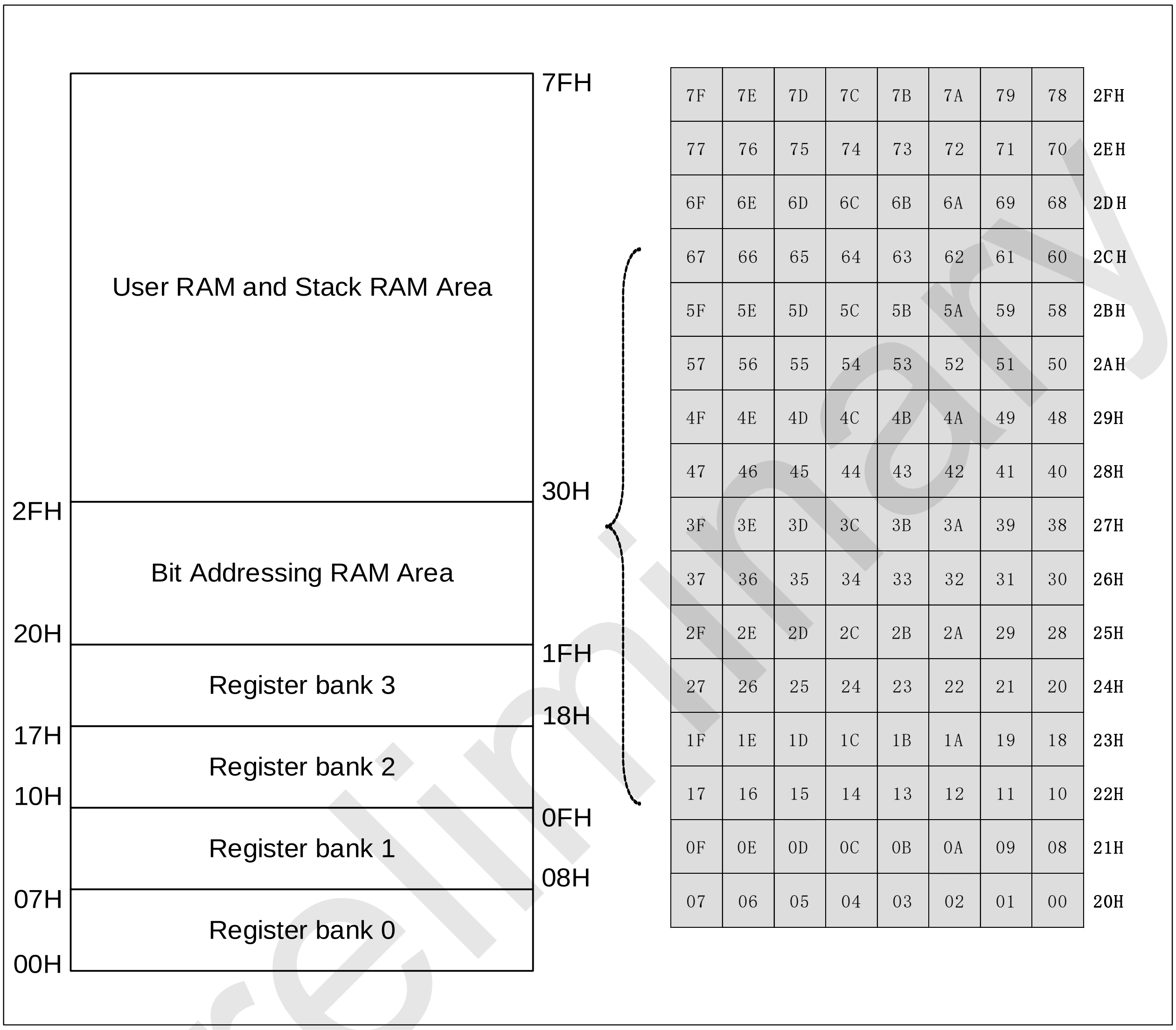
2. Bit addressing area, 20H ~2FH; user can use it as normal RAM or bitwise addressing RAM; for the latter, the bit address is from 00H to 7FH (bitwise addressing is different from normal SRAM byte-oriented addressing), which can be distinguished by instructions in program;

3. User RAM and stack area, the 8-bit stack pointer will point to stack area after the SC92F837X reset; in general, users can set initial value in initializer, it is recommended to configure in the unit interval from E0H to FFH.



Internal 256 bytes RAM Structure Diagram

Internal low 128 bytes RAM structure is shown below:



SRAM Structure Diagram

5.3.2 External 128 bytes SRAM

External 128 Bytes SRAM supports traditional access to external RAM methods. Use MOVX A, @Ri or MOVX @Ri, A to access external 128 Bytes RAM; External 128 Bytes RAM can also be accessed using MOVX A, @DPTR or MOVX @DPTR, A.

6 Special Function Register (SFR)

6.1 SFR Mapping

The SC92F837X provides some registers equipped with special functions, called SFR. The address of such SFRs is from 80H to FFH, some are bit-addressable, and others are not. It is very convenient for these bit addressable registers to change the value of single bit, of which the address is end up with figure “0” or “8”. All SFR shall use direct addressing for addressing.

The name and address of the SC92F837X special function registers are shown in the table below:

	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
F8h	-	-	-	-	CHKSUML	CHKSUMH	OPINX	OPREG
F0h	B	IAPKEY	IAPADL	IAPADH	IAPADE	IAPDAT	IAPCTL	-
E8h	-	-	-	-	-	-	-	OPERCON
E0h	ACC	-	-	-		-	-	-
D8h	-	-	-	-	-	-	-	-
D0h	PSW	PWMCFG	PWMCON	PWMPRD	PWMDTY3	PWMDTY0	PWMDTY1	PWMDTY2
C8h	T2CON	-	RCAP2L	RCAP2H	TL2	TH2	BTMCON	WDTCON
C0h	-	-	-	-	-	-	INT2F	INT2R
B8h	IP	IP1	INT0F	INT0R				
B0h	-	-	-	-	-	-	-	-
A8h	IE	IE1			-	-		
A0h	P2	P2CON	P2PH	-	-	-	-	-
98h	-	-	P0CON	P0PH	-	SSCON0	SSCON1	SSDAT
90h	P1	P1CON	P1PH	-	-	SSCON2	-	IOHCON
88h	TCON	TMOD	TL0	TL1	TH0	TH1	TMCON	OTCON

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

Notes:

1. **The hollow spaces in the table above means that there is no such register RAM, which is not recommended for users.**
2. **The address of SFR for system configuration is F1H ~ FFH, user use it may result in system exceptions. Users are not allowed to conduct clearing or other operations to these registers during the system initialization process.**

6.2 SFR Instructions

For details on each SFR, see the following table:

Mnemonic	Add	Description	7	6	5	4	3	2	1	0	POR
P0	80H	P0 Data Register	-	-	P05	P04	P03	P02	P01	P00	xx000000b
SP	81H	Stack Pointer	SP[7:0]								00000111b
DPL	82H	Data Pointer Low byte	DPL[7:0]								00000000b
DPH	83H	Data Pointer High byte	DPH[7:0]								00000000b
PCON	87H	Power Management Control Register	SMOD	-	-	-	-	-	STOP	IDL	0xxxxx00b
TCON	88H	Timer Control Register	TF1	TR1	TF0	TR0	-	-	IE0	-	0000xx0xb
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]		-	-	-	-	-	-	00xxxxxxb
P1	90H	P1 data register	-	-	P15	P14	P13	P12	P11	P10	xx000000b
P1CON	91H	P1 Input/output control register	-	-	P1C5	P1C4	P1C3	P1C2	P1C1	P1C0	xx000000b
P1PH	92H	P1 pull up resistance control register	-	-	P1H5	P1H4	P1H3	P1H2	P1H1	P1H0	xx000000b
SSCON2	95H	SSI control register 2	SSCON2[7:0]								00000000b
IOHCON	97H	IOH sets registers	P2H[1:0]		P2L[1:0]		P0H[1:0]		P0L[1:0]		00000000b
P0CON	9AH	P0 input/output control register	-	-	P0C5	P0C4	P0C3	P0C2	P0C1	P0C0	xx000000b
P0PH	9BH	Port P0 pull-up resistance control register	-	-	P0H5	P0H4	P0H3	P0H2	P0H1	P0H0	xx000000b
SSCON0	9DH	SSI control register 0	SSCON0[7:0]								00000000b
SSCON1	9EH	SSI control register 1	SSCON1[7:0]								00000000b
SSDAT	9FH	SSI data register	SSD[7:0]								00000000b
P2	A0H	P2 port data register	P27	P26	P25	P24	-	-	P21	P20	0000x00b
P2CON	A1H	P2 port Input/output control register	P2C7	P2C6	P2C5	P2C4	-	-	P2C1	P2C0	0000x00b
P2PH	A2H	P2 port pull up resistance control register	P2H7	P2H6	P2H5	P2H4	-	-	P2H1	P2H0	0000x00b
IE	A8H	Interrupt enable register	EA	-	ET2	-	ET1	-	ET0	EINT0	0x0x0x00b
IE1	A9H	Interrupt enable register 1	-	-	-	ETK	EINT2	EBTM	EPWM	ESSI	xxx00000b

IP	B8H	Interrupt priority control register	-	-	IPT2	-	IPT1	-	IPT0	IPINT0	xx0x0x00b
IP1	B9H	Interrupt priority control register 1	-	-	-	IPTK	IPINT2	IPBTM	IPPWM	IPSSI	xxx00000b
INT0F	BAH	INT0 drop edge interrupt control register	-	-	-	-	INT0F3	INT0F2	INT0F1	-	xxxx000xb
INT0R	BBH	INT0 upedge interrupt control register	-	-	-	-	INT0R3	INT0R2	INT0R1	-	xxxx000xb
INT2F	C6H	INT2 drop edge interrupt control register	-	-	INT2F5	INT2F4	-	-	INT2F1	INT2F0	xx00xx00b
INT2R	C7H	INT2 ascending edge interrupt control register	-	-	INT2R5	INT2R4	-	-	INT2R1	INT2R0	xx00xx00b
T2CON	C8H	Timer 2 controls registers	TF2	-	-	-	-	TR2	-	-	0xxxx0xxb
RCAP2L	CAH	Timer 2 overload 8 bits low	RCAP2L[7:0]								00000000b
RCAP2H	CBH	Timer 2 overload 8 bits high	RCAP2H[7:0]								00000000b
TL2	CCH	Timer 2 is 8 bits low	TL2[7:0]								00000000b
TH2	CDH	Timer 2 is 8 bits high	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 state word register	CY	AC	F0	RS1	RS0	OV	F1	P	00000000b
PWMCFG	D1H	PWM setting register	PWMCKS[2:0]			-	INV3	INV2	INV1	INV0	000x0000b
PWMCON	D2H	PWM control register	ENPWM	PWMIF	-	-	ENPWM3	ENPWM2	ENPWM1	ENPWM0	00xx0000b
PWMPRD	D3H	PWM cycle setting register	PWMPRD[7:0]								00000000b

PWMDTY3	D4H	PWM3 duty ratio setting register	PDT3[7:0]								00000000b
PWMDTY0	D5H	PWM0 duty ratio setting register	PDT0[7:0]								00000000b
PWMDTY1	D6H	PWM1 duty ratio setting register	PDT1[7:0]								00000000b
PWMDTY2	D7H	PWM2 duty ratio setting register	PDT2[7:0]								00000000b
ACC	E0H	accumulator	ACC[7:0]								00000000b
OPERCON	EFH	Operation control register	-	-	-	-	-	-	-	CHKSUMS	xxxxxxx0b
B	F0H	Register B	B[7:0]								00000000b
IAPKEY	F1H	IAP protects registers	IAPKEY[7:0]								00000000b
IAPADL	F2H	IAP writes to the low 8-bit register	IAPADR[7:0]								00000000b
IAPADH	F3H	IAP writes to the high 5-bit register	-	-	-	IAPADR[12:8]					xxx00000b
IAPADE	F4H	IAP writes to the extended address register	IAPADER[7:0]								00000000b
IAPDAT	F5H	IAP data register	IAPDAT[7:0]								00000000b
IAPCTL	F6H	IAP control register	-	-	-	-	PAYTIMES[1:0]		CMD[1:0]		xxxx0000b
CHKSUML	FCH	Check Sum results register low	CHKSUML[7:0]								00000000b
CHKSUMH	FDH	Check Sum result register high	CHKSUMH[7:0]								00000000b
OPINX	FEH	Option a pointer	OPINX[7:0]								00000000b
OPREG	FFH	The Option register	OPREG[7:0]								nnnnnnnnb

6.2.1 C51 Core SFRs

Program Counter (PC)

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

Accumulator ACC (E0H)

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

B Register (F0H)

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

Stack Pointer SP (81H)

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

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

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

Bit Number	Bit Mnemonic	Description

<table>
<tr><td>7</td><td>CY</td><td>Carry Flag bit

1: The top digit of add operation has carry bit or the top digit of subtraction operation has the borrow digit

0: The top digit of add operation has no carry bit or the top digit of subtraction operation has no borrow digit</td></tr>
<tr><td>6</td><td>AC</td><td>Carry-bit auxiliary flag bit (adjustable upon BCD code add and subtraction operations)

1: There is carry bit in bit 3 upon add operation and borrow bit in bit 3 upon subtraction operation

0: No borrow bit and carry bit</td></tr>
<tr><td>5</td><td>F0</td><td>User flag bit</td></tr>
<tr><td>4 ~ 3</td><td>RS1、RS0</td><td>Register banks selection bits
<table><tr><td>RS1</td><td>RS0</td><td>Current Selected Register banks 0 ~ 3</td></tr><tr><td>0</td><td>0</td><td>Group 0 (00H ~ 07H)</td></tr><tr><td>0</td><td>1</td><td>Group 1 (08H ~ 0FH)</td></tr><tr><td>1</td><td>0</td><td>Group 2 (10H ~ 17H)</td></tr><tr><td>1</td><td>1</td><td>Group 3 (18H ~ 1FH)</td></tr></table></td></tr>
<tr><td>2</td><td>OV</td><td>Overflow flag bit</td></tr>
<tr><td>1</td><td>F1</td><td>F1 flag bit

User customized flag</td></tr>
<tr><td>0</td><td>P</td><td>Parity flag bit. This flag bit is the parity value of the number of 1 in accumulator ACC.

1: Odd number of number of 1 in ACC

0: Even number of number of 1 in ACC (including 0)</td></tr>
</table>

Data Pointer DPTR (82H, 83H)

The data pointer DPTR is a 16-bit special register consisting of a low 8-bit DPL (82H) and a high 8-bit DPH (83H). DPTR is the only register in the traditional 8051 kernel micro controller that can directly carry out 16-bit operation, and can also carry out operation on DPL and DPH by byte respectively.

7 Power, Reset and System Clock

7.1 Power Circuit

The SC92F837X Power includes circuits such as BG, LDO, POR and LVR, which are able to reliably work within the scope of 2.4V ~ 5.5V.

7.2 Power-on Reset

After the SC92F837X 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 SC92F837X will always be in reset mode. There will not be a valid clock until the voltage supplied to the SC92F837X is higher than certain voltage. The duration of reset stage is related to rising speed of external power. Once the external supply voltage is up to built-in POR voltage, the reset stage would be completed.

7.2.2 Loading Information Stage

There is a preheating counter inside the SC92F837X. During the reset stage, this preheating counter is always reset as zero. After the voltage is higher than POR voltage, internal RC oscillator starts to oscillate and this preheating counter starts to count. When internal preheating counter counts up to certain number, one byte data will be read from IFB of Flash ROM (including Code Option) for every certain number of HRC clock, which is saved to internal system registers. After the preheating is completed, such reset signal will end.

7.2.3 Normal Operating Stage

After the loading information stage has been completed, the SC92F837X starts to read instruction code from Flash and enters normal operating stage. At this time, LVR voltage is the set value of Code Option written by user.

7.3 Reset Modes

The SC92F837X has 4 kinds of reset modes: ① External RST reset ②Low-voltage reset (LVR) ③Power-on reset (POR) ④Watchdog (WDT) reset.

7.3.1 External Reset

External reset is to supply a certain width reset pulse signal to the SC92F837X from the RST pin to realize the SC92F837X reset.

RST/SCK/INT01/P1.1 has a reset function. Before burning the program, users can modify it to a non-reset pin by burning the host computer software to configure the Customer Option.

7.3.2 Low-voltage Reset (LVR)

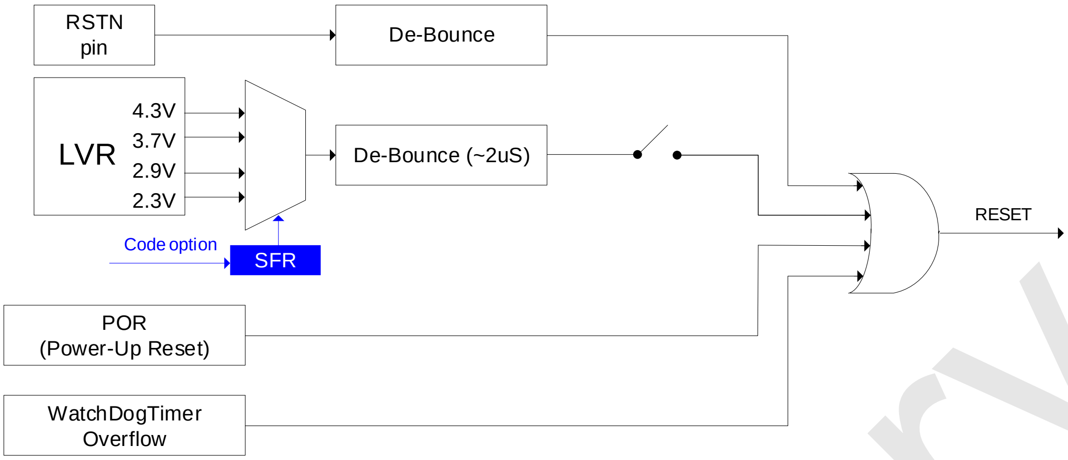
The SC92F837X provides a low-voltage reset circuit. There are 4-level LVR voltage options: 4.3V, 3.7V, 2.9V and 2.3V. The default is the Option value written by user.

OP_CTM0(C1H@FFH) Customer 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 control bit 0: LVR valid 1: LVR invalid
1 ~ 0	**LVRS [1: 0]**	LVR voltage selection bits 11: 4.3 V reset 10: 3.7 V reset 01: 2.9 V reset 00: 2.3 V reset

The Circuit Diagram of the SC92F837X Resetting Part is shown below:



The SC92F837X Reset Diagram

7.3.3 Power-on Reset (POR)

The SC92F837X provides a power-on reset circuit. When power voltage V_{DD} is up to POR reset voltage, the system will be reset automatically.

7.3.4 Watchdog Reset (WDT)

The SC92F837X has a WDT, the clock source of which is the internal 128 kHz oscillator. User can select whether to enable Watchdog Reset function by programmer Code Option.

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

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

<table>
<tr><th>Bit Number</th><th>Bit Mnemonic</th><th>Description</th></tr>
<tr><td>4</td><td>CLRWDT</td><td>Clear WDT (Only valid when set to 1)
1: WDT counter restart, cleared by system hardware</td></tr>
<tr><td>2 ~ 0</td><td>WDTCKS [2: 0]</td><td>WDT clock selection bits
<table>
<tr><td>WDTCKS[2: 0]</td><td>WDT overflow time</td></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>
</table></td></tr>
</table>

		111	3.94ms	
7 ~ 5, 3	-	Reserved		

7.3.5 Register Reset Value

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

The initial value of power-on reset in SFRs is shown in the table below:

Mnemonic	**Reset value**	**Mnemonic**	**Reset value**
ACC	00000000b	P1PH	xx000000b
B	00000000b	P2	0000xx00b
PSW	00000000b	P2CON	0000xx00b
SP	00000111b	P2PH	0000xx00b
DPL	00000000b	PWMCFG	00x00000b
DPH	00000000b	PWMCON	00xx0000b
PCON	0xxxxx00b	PWMDTY3	00000000b
BTMCON	00xx0000b	PWMDTY0	00000000b
IAPADE	00000000b	PWMDTY1	00000000b
IAPADH	xxx00000b	PWMDTY2	00000000b
IAPADL	00000000b	PWMPRD	00000000b
IAPCTL	xxxx0000b	RCAP2H	00000000b

IAPDAT	00000000b	RCAP2L	00000000b
IAPKEY	00000000b	SSDAT	00000000b
IE	0x0x0x00b	SSCON0	00000000b
IE1	xxx00000b	SSCON1	00000000b
INT0R	xxxx000xb	SSCON2	0000xx0xb
INT2R	xx00xx00b	TCON	0000xx0xb
INT0F	xxxx000xb	TMCON	xxxxx000b
INT2F	xx00xx00b	TMOD	x000x000b
IP	xx0x0x00b	TH0	00000000b
IP1	xxx00000b	TL0	00000000b
OPINX	00000000b	TH1	00000000b
OPREG	nnnnnnnnb	TL1	00000000b
OTCON	00xxxxxxb	T2CON	0xxxx0xxb
IOHCON	00000000b	TH2	00000000b
P0	xx000000b	TL2	00000000b
P0CON	xx000000b	WDTCON	xxx0x000b
P0PH	xx000000b	CHKSUML	00000000b
P1	xx000000b	CHKSUMH	00000000b
P1CON	xx000000b	OPERCON	xxxxxxx0b

7.4 High-speed RC Oscillator

The SC92F837X has a built-in adjustable high-precision HRC. HRC is precisely calibrated to 24 MHz@5V/25℃ when delivery. The user can set system clock as 12/6/2MHz by programmer Code Option. The calibration process is to filter the influence of processing deviation on precision. There will be certain drifting of this HRC depending on operating temperature and voltage. As for voltage drifting (2.9V ~ 5.5V) and temperature drifting (-20℃ ~ 85℃), the deviation is within ±1%.

OP_CTM0 (C1H@FFH) Customer 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
5 ~ 4	SCLKS[1: 0]	System clock frequency selection bits: 00: reserved; 01: system clock frequency is HRC frequency divided by 2; 10: system clock frequency is HRC frequency divided by 4; 11: system clock frequency is HRC frequency divided by 12;

The SC92F837X has a special function: the user can modify SFR value to adjust frequency of HRC within certain scope.

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

Bit Number	7	6	5	4	3	2	1	0
Bit 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** OP_HRCR[7:0] the value of HRC after power on ensures that HRC can work at 12/6/2MHz accurately (according to the user's Code Option). The initial value of this value may vary from IC to IC. Users can change the HRC operating frequency by modifying the value of this register. The initial value is OP_HRCR [s]. At this time, IC works at 12/6/2mhz. For every 1 change in OP_HRCR [7:0], HRC frequency changes by about 0.23%@12MHz. The relationship between OP_HRCR [7: 0] and output frequency f_{SYS} is shown as follows:

OP_HRCR [7:0]	HRC actual output frequency (taking 12M as an example)
OP_HRCR [s]-n	12000*(1-0.23%*n)kHz
...	....
OP_HRCR [s]-2	12000*(1-0.23%*2) = 11944.8kHz
OP_HRCR [s]-1	12000*(1-0.23%*1) = 11972.4kHz
OP_HRCR [s]	12000kHz
OP_HRCR [s]+1	12000*(1+0.23%*1) = 12027.6kHz

<table>
<tr>
<td></td>
<td></td>
<td>
<table>
<tr><td>OP_HRCR [s]+2</td><td>12000*(1+0.23%*2) = 12055.2kHz</td></tr>
<tr><td>...</td><td>...</td></tr>
<tr><td>OP_HRCR [s]+n</td><td>12000*(1+0.23%*n)kHz</td></tr>
</table>
Notes:

The value of OP_HRCR[7:0] after each power on of IC was that HRC operated at the HRC closest to 12/6/2mhz. The user can use EEPROM to correct the HRC value after each power to make the HRC work at the frequency required by the user;
To guarantee IC operating reliably, the maximum operating frequency of IC shall not exceed 10% of 12MHz, which is 13.2MHz;
The user shall confirm the change of HRC frequency will not influence other functions.

</td>
</tr>
</table>

7.5 Low-speed RC Oscillator and Low-speed Clock Timer

SC92F837X has a RC oscillation circuit with 128kHz built in as the clock source of the low-frequency clock Base Timer and WDT. Starting the Base Timer or enabling WDT can start the 128kHz low frequency oscillator.

Low frequency clock Base Timer can wake the CPU from STOP mode and generate interrupts.

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

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

Bit Number	Bit Mnemonic	Description

7	**ENBTM**	Low-frequency Base Timer start control bit 0: Base Timer not start 1: Base Timer start
6	**BTMIF**	Base Timer interrupt application flag bit When CPU receives Base Timer interrupt, this flag will be cleared automatically by hardware.
3 ~ 0	**BTMFS [3: 0]**	Low-frequency clock interrupt frequency selection bits 0000: an interrupt is generated for every 15.625ms 0001: an interrupt is generated for every 31.25ms 0010: an interrupt is generated for every 62.5ms 0011: an interrupt is generated for every 125ms 0100: an interrupt is generated for every 0.25s 0101: an interrupt is generated for every 0.5s 0110: an interrupt is generated for every 1.0s 0111: an interrupt is generated for every 2.0s 1000: an interrupt is generated for every 4.0s 1001~1111：reserved
5 ~ 4	-	Reserved

7.6 STOP Mode and IDLE Mode

The SC92F837X provides a SFR PCON, the user can configure bit 0 and bit 1 of this register to control MCU to enter different operating modes.

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

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

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

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

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

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

For example, configure MCU to enter STOP mode:

C program example:

```
#include"intrins.h"
    PCON |= 0x02;          //Set to 1 for PCON bit1 STOP bit, configure MCU to enter STOP mode
    _nop_ ();              //At least 8 _nop_ () required
    _nop_ ();
    _nop_ ();
    _nop_ ();
    _nop_ ();
```

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

Assembly program example:

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

8 CPU and Function System

8.1 CPU

CPU used by the SC92F837X is the high-speed 1T standard 8051 core, whose instructions are completely compatible with traditional 8051 core microcontroller unit.

8.2 Addressing Mode

The addressing mode of the SC92F837X 1T 8051 CPU instructions includes: ①Immediate Addressing ② Direct Addressing ③ Indirect Address ④ Register Addressing ⑤ Relative Addressing ⑥ Indexed Addressing ⑦ Bit Addressing

8.2.1 Immediate Addressing

Immediate addressing is also called immediate operand addressing, which is the operand given to participate in operation in instruction, the instruction is illustrated as follows:

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

8.2.2 Direct Addressing

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

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

8.2.3 Indirect Addressing

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

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

8.2.4 Register Addressing

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

INC R0 refers to (R0) +1→R0

8.2.5 Relative Addressing

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

JC ＄+50H

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

8.2.6 Indexed Addressing

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

MOVC A, @A+DPTR

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

8.2.7 Bits Addressing

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

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

9 Interrupt

The SC92F837X provides 9 interrupt sources: Timer0, Timer1, Timer2, INT0, INT2, PWM, SSI, Base Timer, TK. These 9 interrupt sources are equipped with 2-level interrupt priority-capability and each interrupt source can be individually configured in high priority or low priority. As for two external interrupts, the triggering condition of each interrupt source can be set as rising edge, falling edge or dual-edge trigger. Each interrupt is equipped with independent priority setting bit, interrupt flag, interrupt vector and enable bit. Global interrupt enable bit EA can enable or disable all interrupts.

9.1 Interrupt Source and Vector

Lists for the SC92F837X interrupt source, interrupt vector and related control bit are shown below:

Interrupt Source	Interrupt condition	Interrupt Flag	Interrupt Enable Control	Interrupt Priority Control	Interrupt Vector	Query Priority	Interrupt Number (C51)	Flag Clear Mode	Capability of Waking up STOP
INT0	Compliant with External interrupt 0 conditions	IE0	EINT0	IPINT0	0003H	1 (high)	0	H/W Auto	Yes
Timer0	Timer0 overflow	TF0	ET0	IPT0	000BH	2	1	H/W Auto	No
Timer1	Timer1 overflow	TF1	ET1	IPT1	001BH	3	3	H/W Auto	No
Timer2	Timer2 overflow	TF2	ET2	IPT2	002BH	4	5	Must be cleared by user	No
SSI	Receiving or transmitting completed	SPIF/TWIF	ESSI	IPSPI	003BH	5	7	Must be cleared by user	No
PWM	PWM overflow	PWMIF	EPWM	IPPWM	0043H	6	8	H/W Auto	No
BTM	Base timer overflow	BTMIF	EBTM	IPBTM	004BH	7	9	H/W Auto	Yes
INT2	External interrupt 2 conditions compliant	-	EINT2	IPINT2	0053H	8	10	-	Yes
TK	TouchKey counter overflow	TKIF	ETK	IPTK	005B	9	11	H/W Auto	Yes

Under the circumstance where the master interrupt control bit EA=1 and the respective interrupt control bit have been enable, the interrupt occurrence is shown below:

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

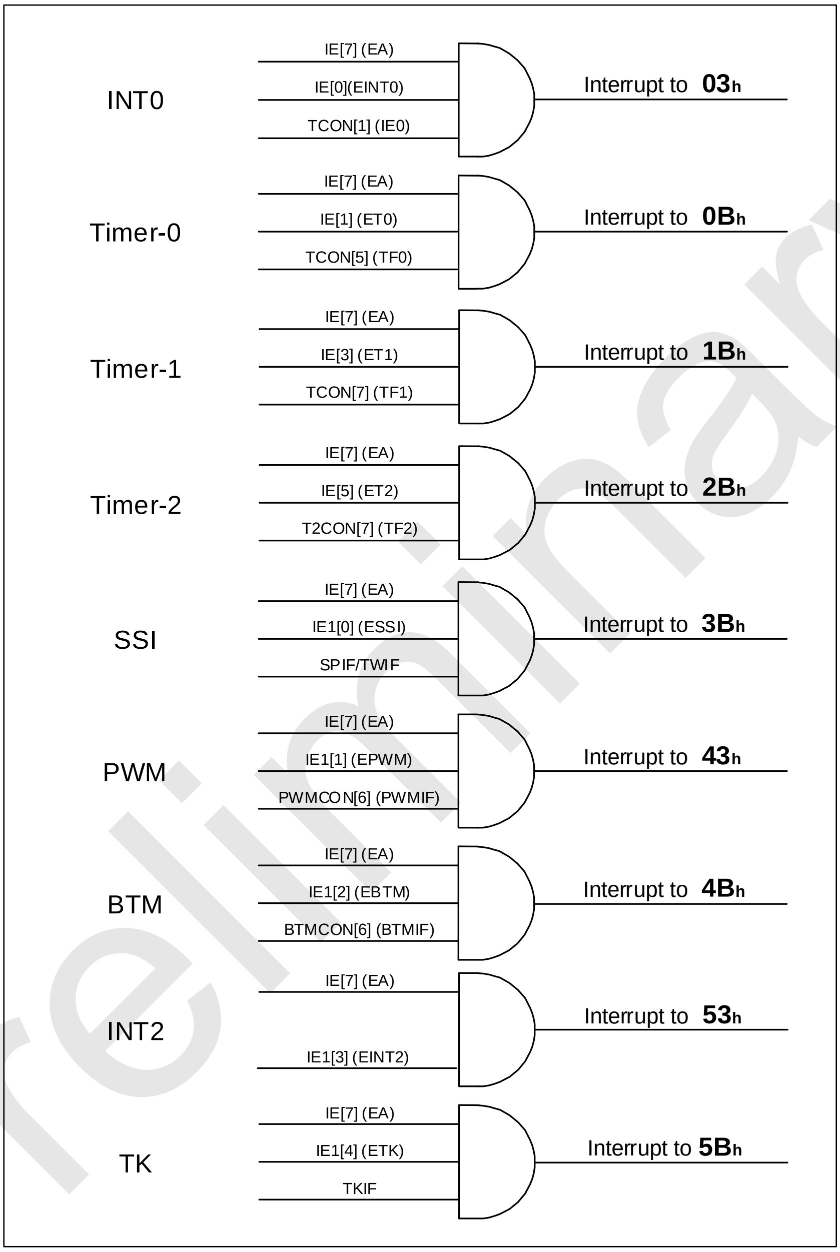
SSI Interrupt: When SSI completes receiving or transmitting a frame of data, SPIF/TWIF bit will be set to "1" automatically by hardware, and SSI interrupt generates. When the microcontroller unit serves SSI interrupt, the interrupt flag SPIF/TWIF must be cleared by software.

PWM Interrupt: When PWM counter overflows (beyond PWMPD), the flag will be set as 1 automatically by hardware. Meanwhile, if the PWM interrupt control bit IE1[1] (EPWM) is set as 1, PWM interrupt will occurs. Once PWM interrupt occurs, the hardware would not clear the interrupt flag automatically, which shall be cleared by user's software.

External Interrupt INT0, INT2: When any external interrupt pin meets the interrupt conditions, external interrupt generates. The external interrupt INT0 would set up interrupt flag IE0 which will be automatically cleared by hardware rather than user. INT0 have three external interrupt sources and INT2 have four external interrupt sources, which can be set in rising edge, falling edge or dual edge interrupt trigger mode by setting SFRs (INTxF and INTxR). User can set the priority level of each interrupt through IP register. Besides, external interrupt INT0, INT2 can also wake up STOP mode of microcontroller unit.

9.2 Interrupt Structure Diagram

The SC92F837X interrupt structure is shown in the figure below:



The SC92F837X Interrupt Structure and Vector

9.3 Interrupt Priority

The SC92F837X micro controller unit has two-level interrupt priority capability. The interrupt requests of these interrupt sources can be programmed as high-priority interrupt or low-priority interrupt, which is to realize the nesting of two levels of interrupt service programs. One interrupt can be interrupted by a higher priority interrupt request when being responded to, which can not be interrupted by another interrupt request at the same priority level, until such response to the first-come interrupt ends up with the instruction “RETI”. Exist the interrupt service routine and return to main program, the system would execute one more instruction before responding to new interrupt request.

That is to say:

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

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

Interrupt query sequence: As for the sequence of that the SC92F837X microcontroller unit responds to the same priority interrupts which occur in the meantime, the priority sequence of interrupt response shall be the same as the interrupt query number in C51, which is to preferentially respond to the interrupt with smaller query number then the interrupt with bigger query number.

9.4 Interrupt Processing Flow

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

① Complete execution of instruction being currently executed;

② Push the PC value into stack for site protection;

③ Load Interrupt vector address into program counter (PC);

④ Carry out corresponding interrupt service program;

⑤ End Interrupt service program ends and execute RETI;

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

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

9.5 Interrupt-related Registers

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

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

Bit Number	Bit Mnemonic	Description
7	EA	Global interrupt enable control bit 0: Disable all interrupts 1: Enable all interrupts
5	ET2	Timer2 interrupt enable control bit 0: Disable Timer2 interrupt 1: Enable Timer2 interrupt
3	ET1	Timer1 interrupt enable control bit 0: Disable Timer1 interrupt 1: Enable Timer1 interrupt
1	ET0	Timer0 interrupt enable control bit 0: Disable Timer0 interrupt 1: Enable Timer0 interrupt
0	EINT0	External interrupt0 enable control 0: Disable INT0 interrupt 1: Enable INT0 interrupt

6, 4, 2	-	Reserved

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

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

Bit Number	Bit Mnemonic	Description
5	**IPT2**	Timer2 interrupt priority selection bit 0: Timer2 interrupt priority is low 1: Timer2 interrupt priority is high
3	**IPT1**	Timer1 interrupt priority selection bit 0: Timer1 interrupt priority is low 1: Timer1 interrupt priority is high
1	**IPT0**	Timer 0 interrupt priority selection bit 0: Timer0 interrupt priority is low 1: Timer0 interrupt priority is high
0	**IPINT0**	INT0 interrupt priority selection bit 0: INT0 interrupt priority is low 1: INT0 interrupt priority is high
7~6, 4, 2	**-**	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	**ETK**	TouchKey interrupt enabling control bit 0: Disable TouchKey interrupt 1: Enable TouchKey interrupt
3	**EINT2**	External interrupt 2 enabling control bit 0: Disable External interrupt 2 1: Enable External interrupt 2
2	**EBTM**	Base Timer interrupt enabling control bit 0: Disable Base Timer interrupt 1: Enable Base Timer interrupt
1	**EPWM**	PWM interrupt enabling control bit 0: Disable PWM interrupt 1: Enable interrupt upon PWM counting overflows (counting to PWMPRD)
0	**ESSI**	Three-in-on serial interrupt enabling control 0: Disable serial port interrupt 1: Enable 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**	TouchKey interrupt priority selection bit 0: TouchKey interrupt priority is low 1: TouchKey interrupt priority is high
3	**IPINT2**	INT2 interrupt priority selection bit 0: INT2 interrupt priority is low 1: INT2 interrupt priority is high
2	**IPBTM**	Base Timer interrupt priority selection bit 0: Base Timer interrupt priority is low 1: Base Timer interrupt priority is high
1	**IPPWM**	PWM interrupt priority selection bit 0: PWM interrupt priority is low 1: PWM interrupt priority is high
0	**IPSSI**	Three-in-on serial interrupt priority selection bit 0: SSI interrupt priority is low 1: SSI interrupt priority is high
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	-	-	IE0	-
R/W	R/W	R/W	R/W	R/W	-	-	R/W	-
POR	0	0	0	0	x	x	0	x

Bit Number	Bit Mnemonic	Description
1	**IE0**	INT0 overflow interrupt request flag bit When INT1 overflow occurs, interrupt generates, hardware set IE0 to ”1”; when the application is interrupted, upon CPU responds, the hardware resets it to ”0”
3~2, 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 bit 0: INT0n falling edge interrupt off 1: INT0n falling edge interrupt enabling
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 bit 0: INT0n rising edge interrupt off 1: INT0n rising edge interrupt enabling
7 ~ 4,0	**-**	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	-	-	INT2F1	INT2F0
R/W	-	-	R/W	R/W	-	-	R/W	R/W

POR	x	x	0	0	x	x	0	0

Bit Number	Bit Mnemonic	Description
5 ~ 1	**INT2Fn** **(n=0,1,4,5)**	INT2 falling edge interrupt control bit 0: INT2n falling edge interrupt off 1: INT2n falling edge interrupt enabling
7~6,3~2,0	**-**	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	-	-	INT2R1	INT2R0
R/W	-	-	R/W	R/W	-	-	R/W	R/W
POR	x	x	0	0	x	x	0	0

Bit Number	Bit Mnemonic	Description
5~1	**INT2Rn** **(n=0,1,4,5)**	INT2 rising edge interrupt control bit 0: INT2n rising edge interrupt off 1: INT2n rising edge interrupt enabling
7~6,3~2,0	**-**	Reserved

10 Timer/Counter T0 and T1

The SC92F837X has two 16-bit Timer/Counters, Timer0 (T0) and Time1 (T1), with two operating modes: counter mode and timer mode. The operating modes selected by bit C/Tx in the SFR TMOD. T0 and T1 are essentially adding counters with different counting source. The source of timer generated from system clock or frequency division clock, but the source of counters is the input pulse to external pin. Only when TRx = 1, will T0 and T1 be enabled on for counting.

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

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

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

① Mode 0: 13-bit Timer/Counter mode

② Mode 1: 16-bit Timer/Counter mode

③ Mode 2: 8-bit automatic reload mode

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

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

10.1 T0 and T1-related Registers

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

Register instructions are shown below:

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

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

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

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

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

	T1	T0

<table>
<tr><th>Bit Number</th><th>Bit Mnemonic</th><th>Description</th></tr>
<tr><td>6</td><td>C/T1</td><td>Timer or Counter selector 1
0: Cleared for Timer operation (input from internal system clock fsys).
1: Set for Counter operation (input from external pin T1/P1.3).</td></tr>
<tr><td>5 ~ 4</td><td>M11, M01</td><td>Timer1 operating mode
<table>
<tr><td>Mode</td><td>M11</td><td>M01</td><td>Operation</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>13-bit TIMER/Counter, TL1 high 3 bits invalid</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>16-bit Timer/Counter</td></tr>
<tr><td>2</td><td>1</td><td>0</td><td>8-bit Auto-Reload Mode.
TH1 holds a value which is reloaded into 8-bit Timer/CounterTL1 each time it overflows.</td></tr>
<tr><td>3</td><td>1</td><td>1</td><td>Timer/Counter 1 is stopped</td></tr>
</table></td></tr>
<tr><td>2</td><td>C/T0</td><td>Timer or Counter selector 0
0: Cleared for Timer operation (input from internal system clock fsys).
1: Set for Counter operation (input from external pin T1/P1.2).</td></tr>
<tr><td>1 ~ 0</td><td>M10, M00</td><td>Timer0 operating mode
<table>
<tr><td>Mode</td><td>M10</td><td>M00</td><td>Operation</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>13-bit TIMER/Counter, TL0 high 3 bits invalid</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>16-bit Timer/Counter</td></tr>
</table></td></tr>
</table>

		2	1	0	8-bit Auto-Reload Mode. TH0 holds a value which is reloaded into 8-bit Timer/Counter TL0 each time it overflows.
		3	1	1	Split Timer Mode. TL0 is an 8-bit Timer/Counter controlled by the standard Timer0 control bits. TH0 is only an 8-bit timer controlled by Timer1 control bits
7, 3	**-**	Reserved			

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

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

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

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	-	-	-	-	-	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 bit 0: T1 clock source is $f_{SYS}/12$ 1: T1 clock source is f_{SYS}
0	**T0FD**	T0 input frequency selection control bit

		0: T0 clock source is $f_{SYS}/12$ 1: T0 clock source is f_{SYS}

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

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

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

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

Bit Number	7	6	5	4	3	2	1	0
Bit Mnemonic	-	-	IPT2	-	IPT1	-	IPT0	IPINT0
R/W	-	-	R/W	-	R/W	-	R/W	R/W

POR	x	x	0	x	0	x	0	0

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

10.2 T0 Operating Modes

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

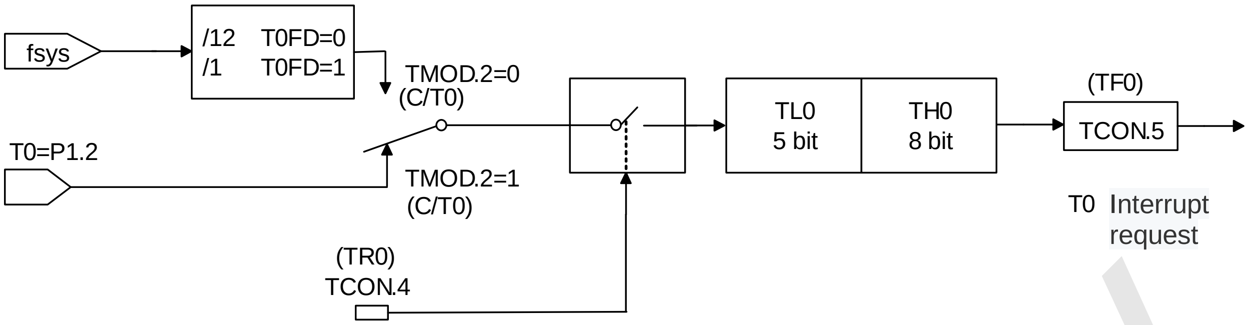
Operating Mode 0: 13-bit Timer/Counter

TH0 register is to store the high 8 bits (TH0.7 ~ TH0.0) of 13-bit Timer/Counter and TL0 is to store the low 5 bits (TL0.4 ~ TL0.0). The high three bits of TL0 (TL0.7 ~ TL0.5) are filled with uncertain values, they shall be omitted upon reading. When 13-bit Timer/Counter overflows with count increment, the system will set timer overflow flag TF0 to 1. An interrupt will be generated if the timer0 interrupt is enabled.

C/T0 bit selects the clock input source of Timer/Counter. If C/T0=1, the level fluctuation from high to low of Counter 0 input pin T0 (P1.2) will make Counter 0 data register add 1. If C/T0=0, the frequency division of system clock is the clock source of Timer0.

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

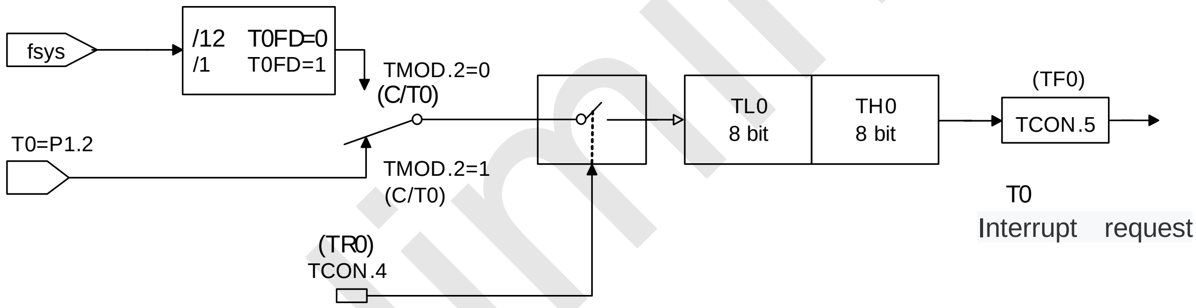
When configured as a timer, the SFR T0FD is used to select fractional frequency ratio of clock source.



Operating mode 0: 13-bit Timer/Counter

Operating Mode 1: 16-bit Counter/Timer

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



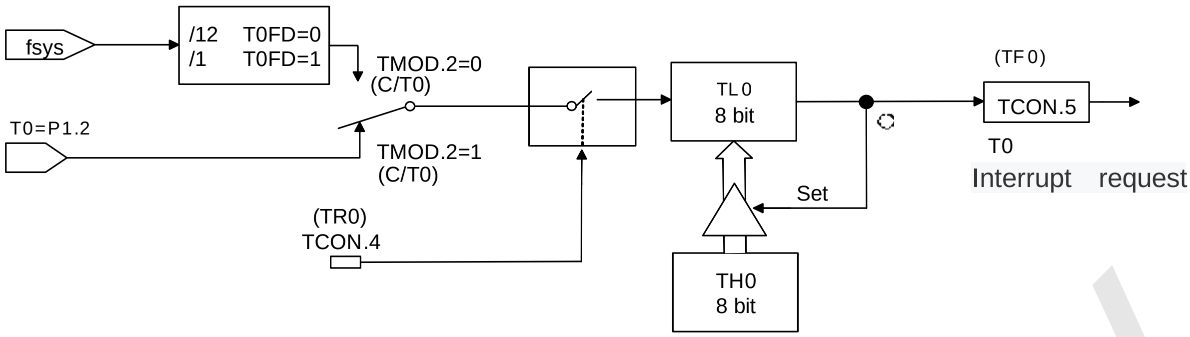
Operating mode 0: 16-bit Timer/Counter

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

In operating mode 2, Timer0 is 8-bit automatic reload Timer/Counter. TL0 is to store counting value and TH0 is to store the reload value. When the counter in TL0 overflows and turn to 0x00, the overflow flag of Timer TF0 will be set to 1, and the data in register TH0 will be reloaded into register TL0. If the timer interrupt enabled, setting TF0 to 1 will generate an interrupt, but the reloaded value in TH0 will remain the same. Before starting the Timer to count correctly, TL0 shall be initialized to the required value.

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

When configured as a timer, the SFR TMCON bit 0 (T0FD) is used to select fractional frequency ratio of system clock f_{SYS}.



Operating Mode 2: 8bit Automatic Reload Counter/Timer

Operating Mode 3: Two 8-bit Counter/Timer (only for Timer0)

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

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

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

10.3 T1 Operating Modes

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

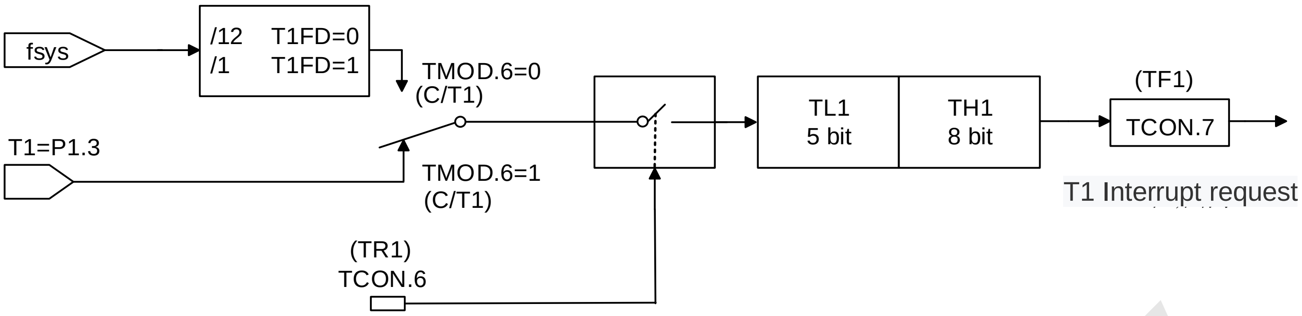
Operating Mode 0: 13-bit Timer/Counter

TH1 register is to store high 8-bit (TH1.7 ~ TH1.0) of 13-bit Timer/Counter and TL1 is to store low 5-bit (TL1.4 ~ TL1.0). The high 3-bit of TL1 (TL1.7 ~ TL1.5) are uncertain values, they shall be omitted upon reading. When 13-bit Timer/Counter overflow with count increment, the system will set timer overflow flag TF1 as1. An interrupt will be generated if the timer1 interrupt is enabled. C/T1 bit selects the clock input source of Timer/Counter.

If C/T1=1, the level fluctuation from high to low of timer1 input pin T1 (P0.3) will make timer1 data register add 1. If C/T1=0, the frequency division of system clock is the clock source of timer1.

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

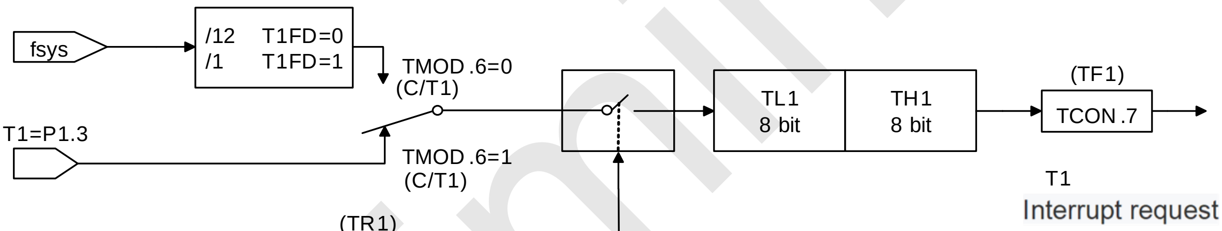
When configured as timer, the SFR T1FD is used to select fractional frequency ratio of clock source.



Operating mode 0: 13-bit Timer/Counter

Operating Mode 1: 16-bit Counter/Timer

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



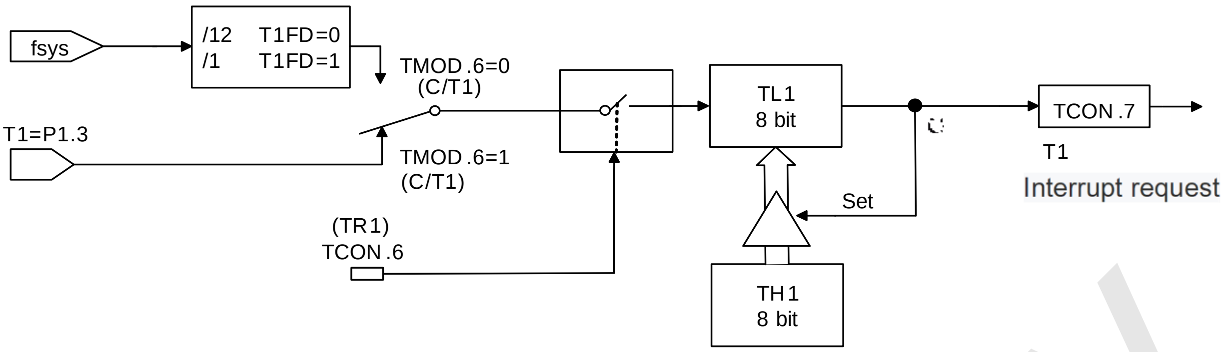
Operating mode 0: 16-bit Timer/Counter

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

In operating mode 2, Timer1 is 8-bit automatic reload Timer/Counter. TL1 is to store counting value and TH1 is to store the reload value. When the counter in TL1 overflows 0x00, the overflow flag of Timer TF1 will be set to 1, and the value of register TH1 will be reloaded into register TL1. If enable the timer interrupt, setting TF1 to 1 will generate an interrupt, but the reloaded value in TH1 will remain unchanged. Before allowing Timer to correctly count, TL1 shall be initialized to the required value.

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

When configured as timer, the SFR TMCON bit 4 (T1FD) is used to select the ratio of clock source of timer to fractional frequency of system clock f_{SYS}.



Operating Mode 2: 8bit Automatic Reload Counter/Timer

11 Timer/Counter T2

Timer2 is adding counters in nature, differing in counting source. The clock source of T2 comes from system clock or frequency division clock. TR2 is the counting switch of Timer/Counter T2. Only when TR2 = 1, will T2 be enabled for counting.

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 1 operating modes:

① Mode 1: 16-bit automatic reload timer mode

11.1 T2-related Registers

Mnemonic	Add	Description	7	6	5	4	3	2	1	0	POR
T2CON	C8H	Timer2 Control Register	TF2	-	-	-	-	TR2	-	-	0xxxx0xxb
RCAP2L	CAH	Timer2 Reload Low Byte	RCAP2L[7:0]								00000000b
RCAP2H	CBH	Timer2 Reload 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	-	-	-	-	TR2	-	-
R/W	R/W	-	-	-	-	R/W	-	-
POR	0	x	x	x	x	0	x	x

Bit Number	Bit Mnemonic	Description
7	**TF2**	Timer2 overflow flag bit 0: No overflow (must be cleared by software) 1: Overflow
2	**TR2**	Timer2 start/stop control bit 0: Stop Timer2 1: Start Timer2
6~3,1~0	-	**Fixed writing 0**

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	-	ET2	-	ET1	-	ET0	EINT0
R/W	R/W	-	R/W	-	R/W	-	R/W	R/W
POR	0	x	0	x	0	x	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)

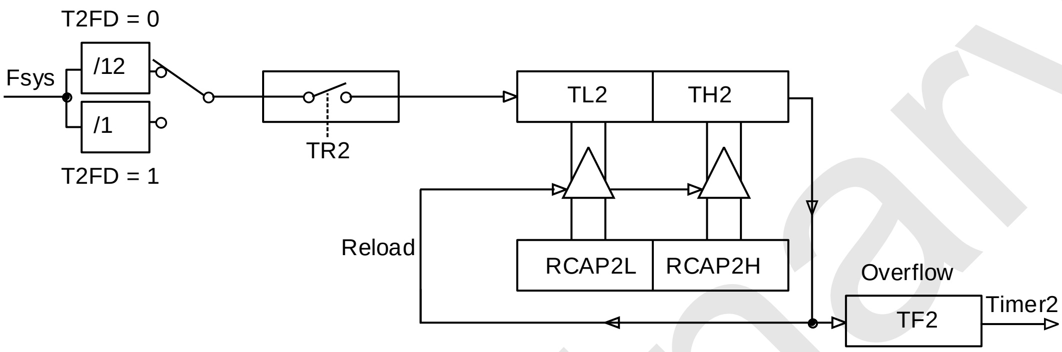
Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	-	-	IPT2	-	IPT1	-	IPT0	IPINT0
R/W	-	-	R/W	-	R/W	-	R/W	R/W
POR	x	x	0	x	0	x	0	0

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

Operating Mode 1: 16-bit Automatic Reload Timer

In the 16-bit automatic reload mode, Timer 2 increments to 0xFFFFH and sets up the TF2 bit after overflow. At the same time, timer automatically loads the 16-bit values of registers RCAP2H and RCAP2L written by user software into registers TH2 and TL2.



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

Note:

1. **TF2 can be set by software, only software and hardware reset can clear TF2;**
2. **When EA = 1 and ET2 = 1, setting up TF2to 1 can arouse interrupt of Timer2;**

12 PWM

The SC92F837X provides a stand-alone counter that supports four PWM outputs: PWM0 ~ PWM3.

The SC92F837X PWM has the following functions:

① 8-bit PWM precision;

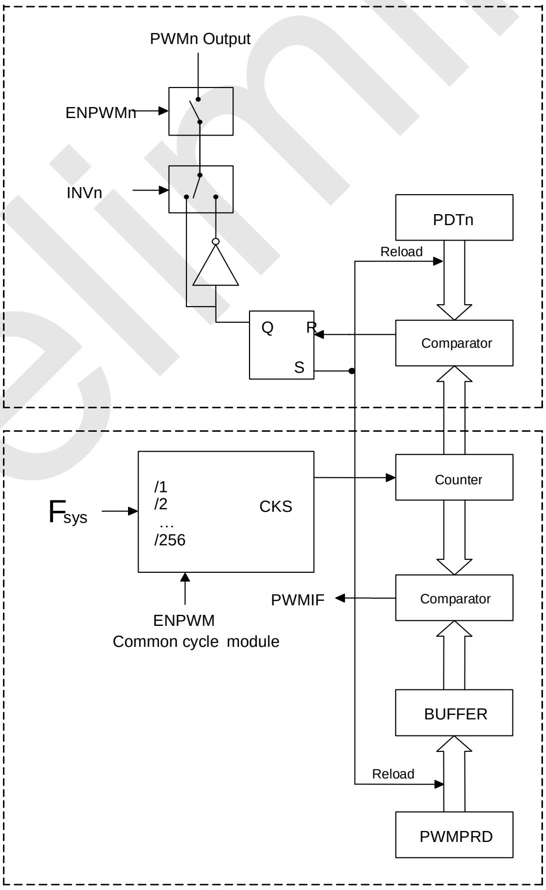
② PWM0, PWM1, PWM2 and PWM3 have the same period, but the duty ratio can be set separately;

③ Output can be configured in forward or reverse direction;

④ Provide 1 PWM overflow interrupt.

The PWM of SC92F837X can support the adjustment of cycle and duty cycle, registers PWMCON and PWMCFG control PWM0~3 related Settings, PWMPRD sets the common period of PWM, PWMDTY0~3 controls the duty cycle of PWM0~3 respectively.

12.1 PWM block Diagram



The SC92F837X PWM block Diagram

12.2 PWM-related Registers

symbol	address	Description	7	6	5	4	3	2	1	0	Reset
PWMCFG	D1H	PWM setting register	PWMCKS[2:0]			-	INV3	INV2	INV1	INV0	000x0000b
PWMCON	D2H	PWM control register	ENPWM	PWMIF	-	-	ENPWM3	ENPWM2	ENPWM1	ENPWM0	00xx0000b
PWMPRD	D3H	PWM cycle setting register	PWMPRD[7:0]								00000000b
PWMDTY3	D4H	PWM3 duty ratio setting register	PDT3[7:0]								00000000b
PWMDTY0	D5H	PWM0 duty ratio setting register	PDT0[7:0]								00000000b
PWMDTY1	D6H	PWM1 duty ratio setting register	PDT1[7:0]								00000000b
PWMDTY2	D7H	PWM2 duty ratio setting register	PDT2[7:0]								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

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

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

Bit Number	Bit Mnemonic	Description
7	**ENPWM**	PWM module switch control bit (Enable PWM) 1: Clock is allowed to enter the PWM unit, PWM is in working state. 0: PWM unit stops working, PWM counter is cleared. **PWMn is still attached to the outlet. To use other functions that are multiplexed with the outlet of PWMn, ENPWMn should be set to 0.**
6	**PWMIF**	PWM interrupt request flag bit (PWM Interrupt Flag) When the PWM counter overflows (that is to say, when the count exceeds PWMPD), this bit is automatically set to 1 by the hardware. If IE1[1] (EPWM) is also set to 1 at this time, PWM interrupt is generated.
3	**ENPWM3**	PWM3 function switch 0: PWM3 does not output to IO 1: PWM3 output to IO
2	**ENPWM2**	PWM2 function switch 0: PWM2 does not output to IO

		1: PWM2 output to IO
1	**ENPWM1**	PWM1 function switch 0: PWM1 does not output to IO 1: PWM1 output to IO
0	**ENPWM0**	PWM0 function switch 0: PWM0 does not output to IO 1: PWM0 output to IO
5~2	**-**	Reserved

PWMPRD (D3H) PWM Cycle Setting register (Read/write)

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Symbol	**PWMPRD[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	**PWMPRD[7:0]**	PWM Shared cycle setting; This value represents the output waveform of PWM0/1/2/3 (period - 1); In other words, the period value of PWM output is (PWMPRD[7:0] + 1)* PWM clock;

PWMPRD[7:0] is a four-channel PWM Shared cycle setting controller. Whenever the PWM counter counts to the pre-set value of PWMPRD[7:0], the next PWM CLK comes and the counter hops to 00h, that is to say, PWM0/1/2/3 cycles are (PWMPRD[7:0] + 1)*PWM clock.

PWMCFG (D1H) PWM Set register (read/write)

Bit Number	7	6	5	4	3	2	1	0
Bit Mnemonic	PWMCKS[2:0]			-	INV3	INV2	INV1	INV0
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
7~5	**PWMCKS[2:0]**	PWM ClocK Source Selector 000: f_{SYS} 001: f_{SYS} / 2 010: f_{SYS} / 4 011: f_{SYS} / 8 100: f_{SYS} / 32 101: f_{SYS} / 64 110: f_{SYS} / 128 111: f_{SYS} / 256
3	**INV3**	PWM3 output reverse control 1: Reverse the output of PWM3 0: The output of PWM3 is not reversed
2	**INV2**	PWM2 output reverse control 1: Reverse the output of PWM2 0: The output of PWM2 is not reversed
1	**INV1**	PWM1 output reverse control

		1: Reverse the output of PWM1 0: The output of PWM1 is not reversed
0	**INV0**	PWM0 output reverse control 1: Reverse the output of PWM0 0: The output of PWM0 is not reversed
4~2	**-**	Reserved

The counting time of the PWM counter can be controlled by PWMCFG[3:0]. Different number of system clocks can be selected to count one unit (pre-scalar selector), that is, to select the frequency division ratio of the clock source of the PWM counter divided by the system clock f_{SYS}. PWM0/1/2/3 can also be used to select whether the PWM output is reversed by INV0~INV3 in PWMCFG.

PWMDTY0 (D5H) PWM0 duty cycle Setting register (read/write)

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	**PDT0[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	**PDT0 [7:0]**	PWM0 duty cycle length setting; The high-level width of the PWM0 is (PDT0[7:0]) PWM clock

PWMDTY1 (D6H) PWM1 duty cycle setting register (read/write)

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	**PDT1[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	**PDT1[7:0]**	PWM1 duty cycle length setting; The high-level width of the PWM1 is (PDT1[7:0]) PWM clock

PWMDTY2 (D7H) PWM2 duty cycle Setting register (read/write)

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	**PDT2[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	**PDT2 [7:0]**	PWM2 duty cycle length setting; The high-level width of the PWM2 is (PDT2[7:0]) PWM clock

PWMDTY3 (D4H) PWM0 duty cycle Setting register (read/write)

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	**PDT3[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	**PDT3 [7:0]**	PWM3 duty cycle length setting; The high-level width of the PWM3 is (PDT3[7:0]) PWM clock

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
1	**EPWM**	PWM interrupt enabled control 0: Turn off PWM interrupt 1: Interrupt is allowed when PWM counter overflows

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
1	**IPPWM**	PWM interrupt priority selection 0: Set the interrupt priority of PWM to "low" 1: Set the interrupt priority of PWM to "high"

Note:

1.ENPWM potential energy controls whether the PWM module works.

2.ENPWMx potential can choose PWMx port as GPIO or PWMx output.

3.EPWM(IE1.1) potential control PWM is allowed to produce interruption.

4. If ENPWM is set to 1, the PWM module is opened, but ENPWMx=0, the PWM output is turned off and serves as the GPIO port. At this time, the PWM module can be used as an 8-bit Timer. At this time, EPWM(IE1.1) is set to 1, and PWM still produces interruption.

5. Two PWM cycles are Shared, and the PWM interrupt generated during overflow is the same interrupt vector.

12.3 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

When PWMn outputs waveform, if it is required to change the duty cycle, users can change the value of high level configuration registers PDTn. Note that changing the value of PDTn will change the duty cycle immediately.

② Period Change features

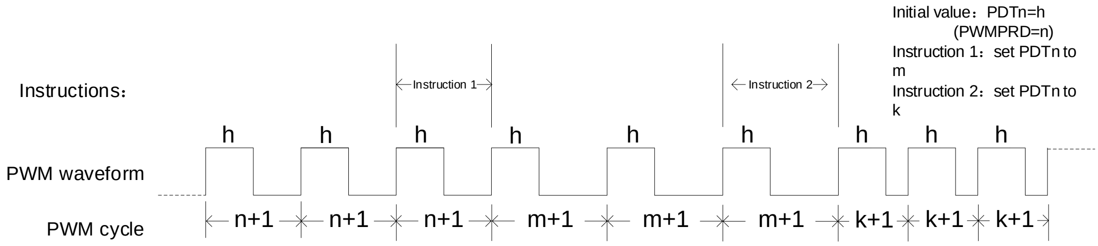


Diagram for Period Change Features

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 change the period immediately.

③ Relationship between Period and Duty cycle

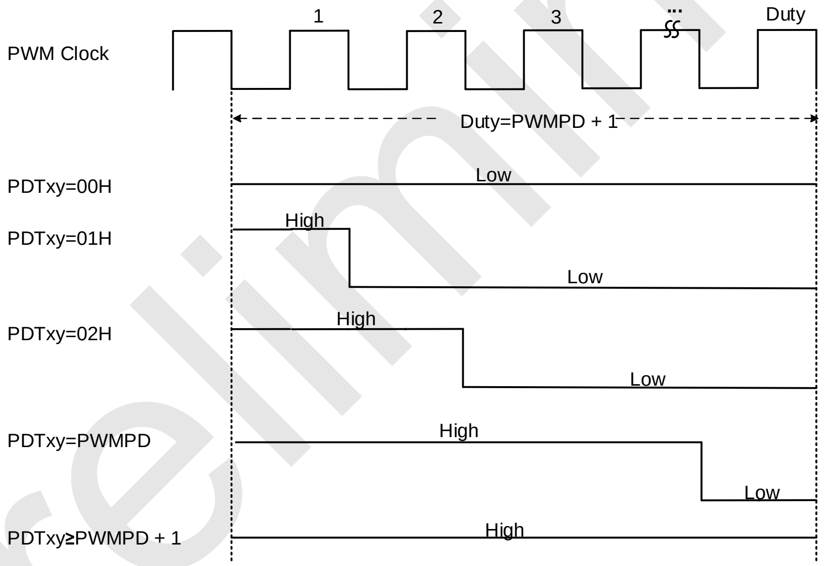


Diagram for Relationship between Period and Duty cycle

The relationship between period and duty cycle is shown in the figure above. The precondition of this result is the PWM0/1/2/3 output reverse control INV0/1/2/3 is initialized to 0; if it is required to get the contrary result, set INV0/1/2/3 to 1.

13 General-purpose I/O (GPIO)

The SC92F837X offers up to 18 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 18 IOs are shared with other functions. Under output state, I/O port reads the value in the port data register.

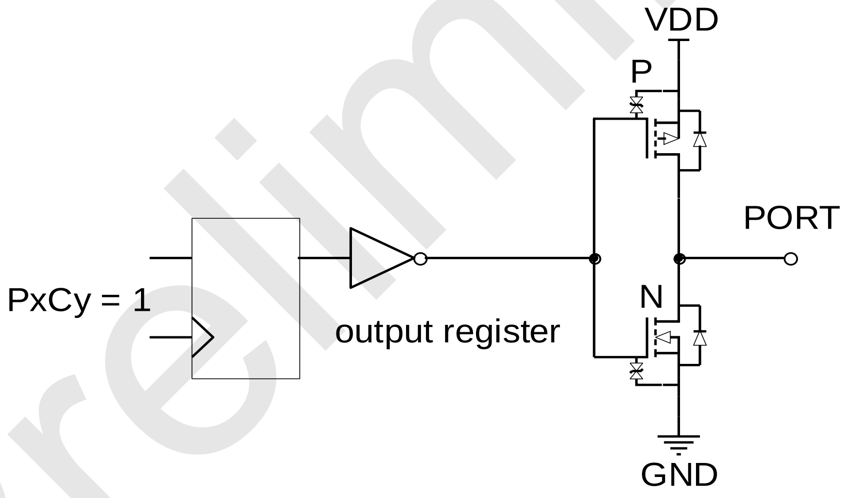
Note: Unused IO port or IO port with no package pin shall be configured as strong push-pull output mode.

13.1 GPIO Structure Diagram

Strong Push-pull Output Mode

In strong push-pull output mode, it is able to provide continuous high current drive: high output for the current larger than 21mA and low output for the current larger than50mA

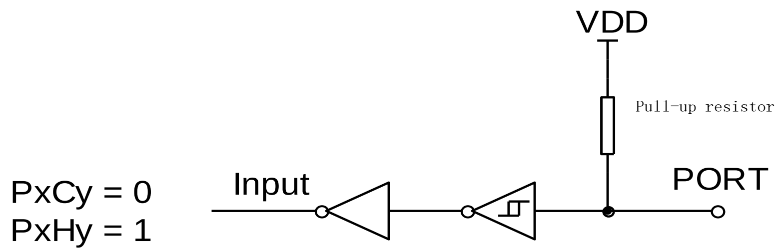
The port structure diagram for strong push-pull output mode is shown below:



Strong Push-pull Output Mode

Pull-up Input Mode

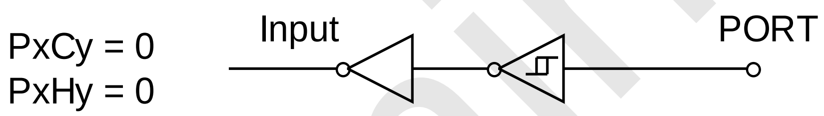
In pull-up input mode, a pull-up resistor is connected on the input port, only when the level on the input port is pulled down, low level signal can be detected.



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

13.2 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	-	-	P0C5	P0C4	P0C3	P0C2	P0C1	P0C0
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
POR	x	x	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	-	-	P0H5	P0H4	P0H3	P0H2	P0H1	P0H0
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
POR	x	x	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	-	-	P1C5	P1C4	P1C3	P1C2	P1C1	P1C0
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
POR	x	x	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	-	-	P1H5	P1H4	P1H3	P1H2	P1H1	P1H0
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
POR	x	x	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	-	-	P2C1	P2C0

R/W	R/W	R/W	R/W	R/W	-	-	R/W	R/W
POR	0	0	0	0	x	x	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	-	-	P2H1	P2H0
R/W	R/W	R/W	R/W	R/W	-	-	R/W	R/W
POR	0	0	0	0	x	x	0	0

Bit Number	Bit Mnemonic	Description
7 ~ 0	**PxCy** **（x=0~2, y=0~7）**	Px port input and output control bit 0: Pxy as input mode (initial value) 1: Pxy as strong push-pull output mode
7 ~ 0	**PxHy** **（x=0~2, y=0~7）**	Px port pull-up resistance configuration, only valid when PxCy=0: 0: Pxy as high-impedance input mode (initial value), 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	-	-	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0

R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
POR	x	x	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	-	-	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
POR	x	x	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	-	-	P2.1	P2.0
R/W	R/W	R/W	R/W	R/W	-	-	R/W	R/W
POR	0	0	0	0	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);

14 SPI/TWI/UART Serial Interface (SSI)

The SC92F837X integrates SPI/TWI/UART serial interface circuits (SSI), which is convenient for connecting MCU to devices or equipment with different interfaces. The user can configure SSI in any communication mode among SPI, TWI and UART by configuring SSMOD[1: 0] bit of register OTCON. Its features are shown below:

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

Specific configuration modes are shown below:

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

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

14.1 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.

14.1.1 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: Transmission interrupt not enable 1: Transmission interrupt enable, when SPIF=1, TBIE=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.

14.1.2 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.

14.1.3 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 SC92F837X 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 SC92F837X SPI:

SC92F837X 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 SC92F837X 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.

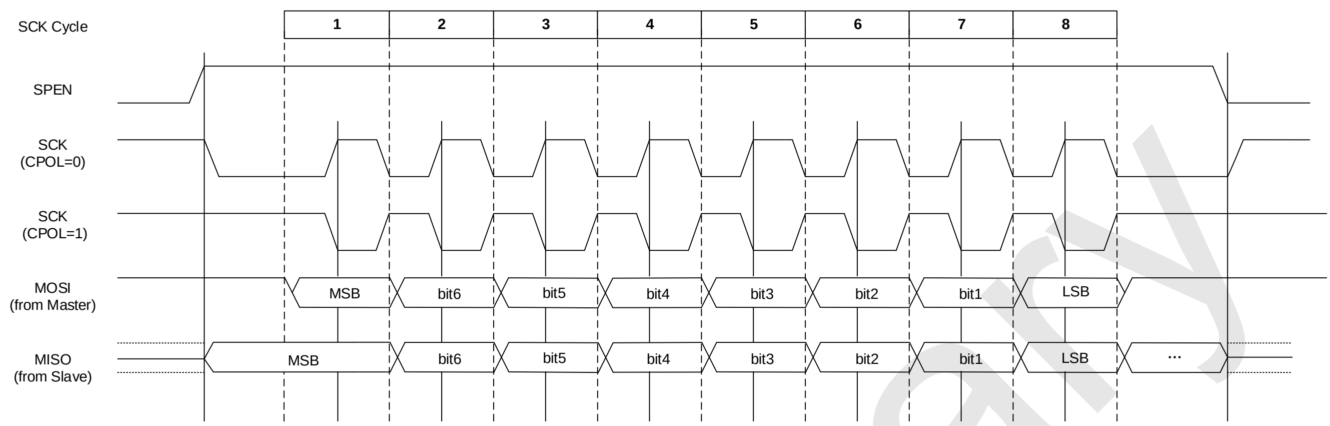
- **Transmitting and Receiving:**

  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.

14.1.4 Transfer Form

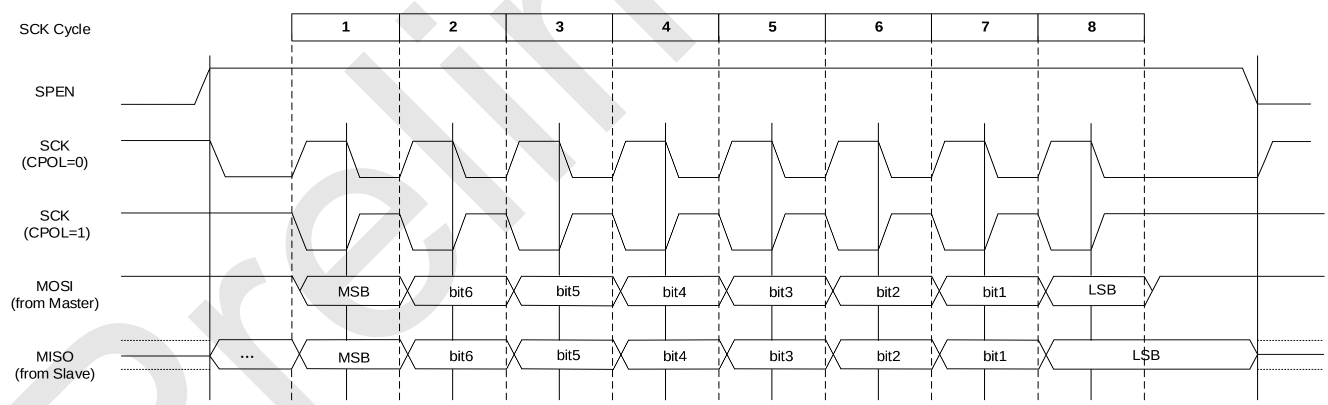
By setting CPOL bit and CPHA bit of SSCON0 register by software, the user can select four combinations of SPI clock polarity and clock phase. CPOL bit defines the polarity of clock, meaning the level status when idle, which has little influence on SPI transmission format. CPHA bit defines the phase of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

14.1.5 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.

14.2 TWI

14.2 Two-Wire Interface (TWI)

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

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

		1: Information sent by receiving master allowed
2 ~ 0	**STATE[2: 0]**	Device status flag Bits 000: slave device is in idle state, wait for TWEN to be set to 1, and detect TWI startup signal. When slave device receives stopping conditions, it will skip to this state 001: Slave device is receiving first frame of address and read and write bits (8th bit for read and write bit, 1 for reading, 0 for writing). After receiving initial conditions, slave device will skip to this state. 010: State of slave device receiving data 011: State of slave device transmitting data 100: In the state of transmitting data of slave device, when the master device returns to UACK (high level for acknowledge bit), skip to this state, wait for restarting signal or stopping signal. 101: When the slave device is in transmitting state, setting AA to 0 and it will enter this state, waiting for restarting signal or stopping signal.
5	-	Reserved

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

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

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

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

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

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

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

14.2.1 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.

14.2.2 Operating Modes

TWI communication of the SC92F837X 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:**

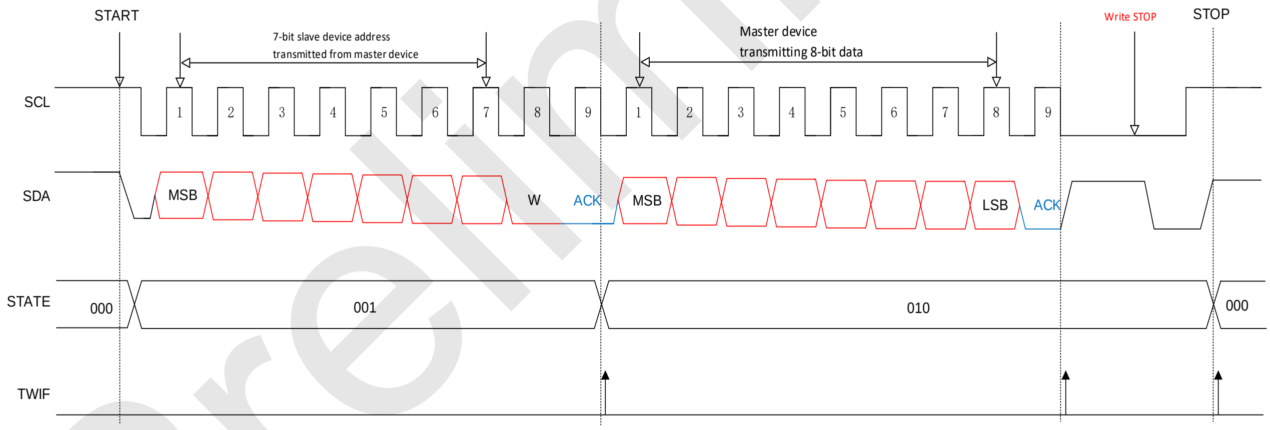
If the read and write bit received from the first frame is writing (0), the slave device enters into the receiving state of slave device (STATE [2: 0] = 010), and wait for data sent from receiving master device. Master device will release the bus for transmitting every 8 bits and then wait for the response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

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

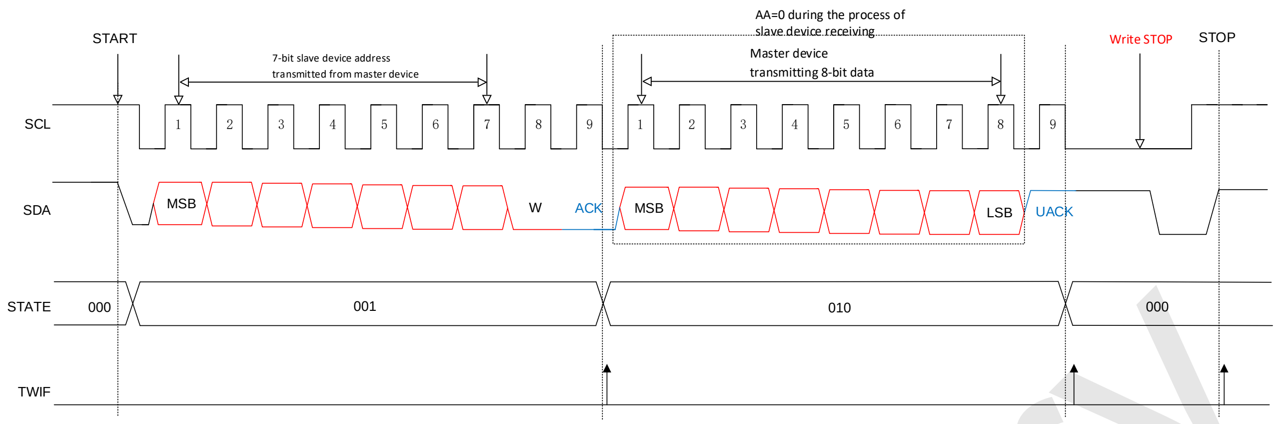
    1) Continue to send data;

    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.



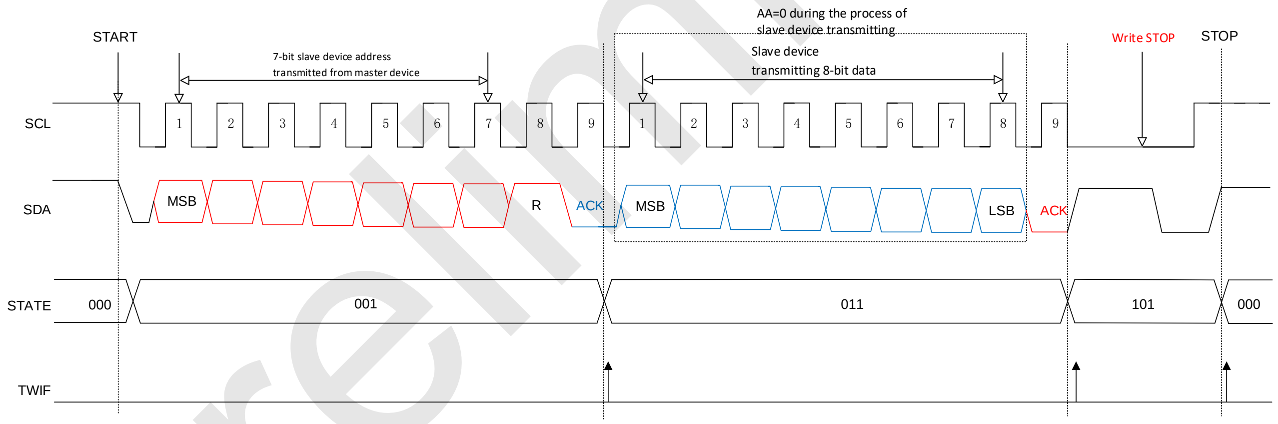
2. If the response state of slave device is in high level (during the receiving process, the value of AA in slave device register is rewritten to 0), it indicates that after transmitting current bytes, the slave device will stop this transmission automatically and return to idle state (STATE[2: 0] =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



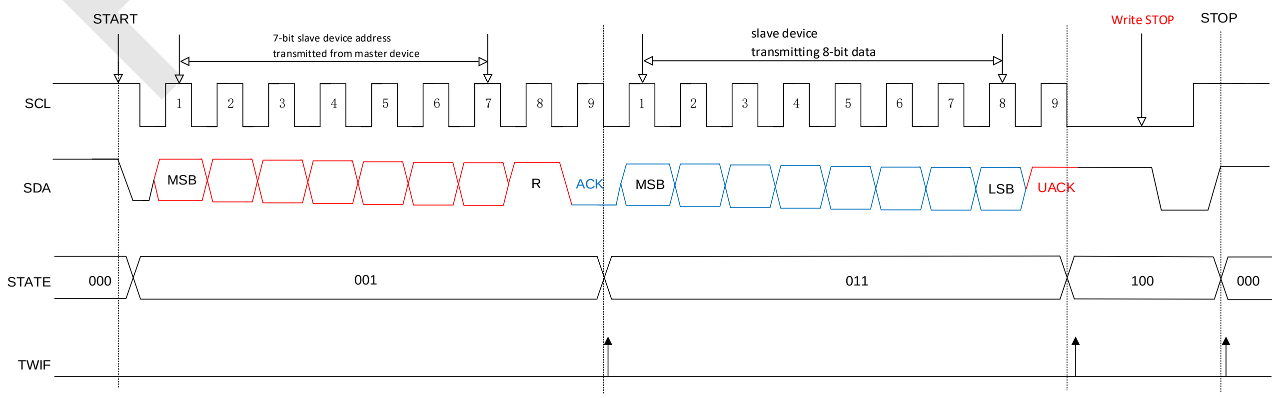
- **Non-general address response, master device transmitting mode:**

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:**

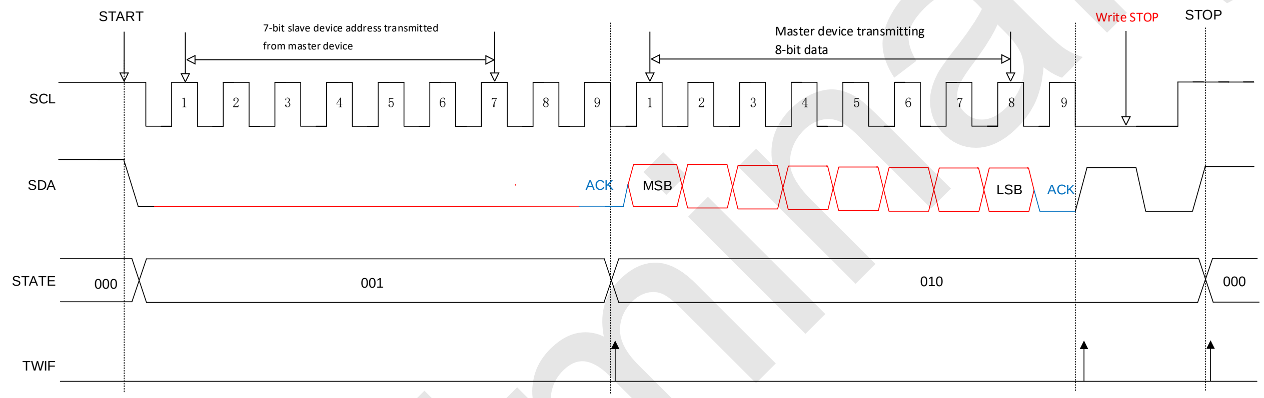
  When GC=1, general address is allowed to be used. When the slave device enters into the state of receiving first frame address (STATE[2: 0] = 001), the address bit data received in first frame data will be 0x00, at this time, all slave device will respond the master device. The read and write bit sent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

14.2.3 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.

14.3 Serial Interface　(UART)

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	BAUDL [7: 0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

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

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

Bit Number	Bit Mnemonic	Description
7 ~ 0	**BAUD [15: 0]**	Serial Port Baud Rate Control Bit $\mathrm{BaudRate} = \frac{\mathrm{fsys}}{\mathrm{BAUD1H, BAUD1L}}$ **Note: [BAUD1H, BAUD1L] must be larger than 0x0010**

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

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

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

15 Double mode touch circuit

SC92F837X has a 16-channel dual mode capacitive touch circuit, which can be configured for high sensitivity mode or high reliability mode. Its features are as follows:

1. The high-sensitivity mode can be used for touch applications requiring high sensitivity, such as spaced key touch control and proximity induction

2. Highly reliable mode has strong anti-interference ability and can pass 10V dynamic CS test

3. It can realize 16-channel touch keys and derivative functions

4. High flexibility development software library support, low development difficulty

5. Automatic debugging software support and intelligent development

6. The touch module can work in low-power mode in MCU STOP mode, and the overall power consumption of the chip can be as low as 11uA when a single touch button is awakened.

15.1 The power consumption mode of the touch circuit

SC92F837X allows touch scanning to be enabled in STOP Mode: this approach can reduce the overall power consumption of the MCU to meet the needs of touch applications with low power consumption.

The touch circuit of SC92F837X can be understood by the user as having two power consumption modes:

1. Normal operation mode

2. Low power operation mode

The two modes of power consumption are defined as follows:

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

15.2 Touch mode

The SC92F837X dual-mode touch circuit provides two touch modes:

1. High sensitivity mode

2. Highly reliable mode

By using the touch button library file provided by Saiyuan (which can be downloaded from Saiyuan official website), users can choose the touch mode and quickly and simply realize the required touch function.

Users can select the most suitable touch mode for the current application through the information in the following table:

instructions	High sensitivity mode	High reliability model
instructions	① High anti-interference capability, can pass 3V dynamic CS ② Ultra high sensitivity	① Super anti - interference ability, through 10V dynamic CS ② Lower power consumption
characteristics	① Normal touch button application ② Spaced touch button application ③ Proximity induction application ④ Touch applications requiring high sensitivity	① Applications requiring strong anti-interference performance ② Applications with 10V dynamic CS requirements
Applicable application	Select the high sensitivity mode by loading the high sensitivity touch library in the project	Select high reliability mode by loading high reliability touch library on the project
How to get into mode	"SOC MCU SC92f8XXX_highsensitive_lib_tn_vx.x.x.l IB library Files User Manual Vx.x"	"SOC MCU SC92f8XXx_highreliability_lib_tn_vx.x.x.l Ib library File User Manual Vx.x"
documentation	"SC92F8X7X_HighSensitive_Lib_Tn_Vx.x.x.LIB"	"SC92F8X7X_HighReliability_Lib_Tn_Vx.x.x.LIB "
The corresponding library file	① T1 library is used for spring type applications ② T2 library is suitable for spaced type applications, and the number of keys is at least 3 or more	Only applicable to spring type applications
Matters needing attention	It is generally recommended to use this high sensitivity mode for a better user experience.	There are only two cases where a highly reliable mode is recommended: ① Need to go through 10V dynamic CS ② A lower power consumption current is required, and the current cannot be full in high sensitivity mode

16 EEPROM and IAP Operations

There are two options for the SC92F837X IAP operating scope:

EEPROM and IAP operating modes are shown below:

1. Internal highest address 128 bytes EEPROM can be used as data storage;
2. The whole 32K bytes of IC ROM and 128 bytes of EEPROM can be used for IAP operations, which is mainly used for remote program updating.

As Code Option, the user shall select IAP operating space before it is written to IC by programmer:

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

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

Bits	Name	Description
3 ~ 2	**IAPS[1: 0]**	EEPROM and IAP Area Selection Bits 00: Code memory prohibits IAP operations, only EEPROM data memory is used for data storage 01: last 0.5k code memory allows IAP operation 10: Last 1k code memory allows IAP operation (0C00H~0FFFH) 11: All code memory allows IAP operation (0000H~0FFFH)
7~4,1~0	**-**	Reserved

16.1 EEPROM / IAP Operating-related Registers

Description for EEPROM / IAP operating-related registers:

Mnemonic	Add	Description	7	6	5	4	3	2	1	0	POR
IAPKEY	F1H	IAP Protection Register	IAPKEY[7:0]								00000000b
IAPADL	F2H	IAP Write Address Low Register	IAPADR[7:0]								00000000b
IAPADH	F3H	IAP Write Address High Register	-	-	-	IAPADR[12:8]					xxx00000b
IAPADE	F4H	IAP Write Extended Address Register	IAPADER[7:0]								00000000b
IAPDAT	F5H	IAP Data Register	IAPDAT[7:0]								00000000b
IAPCTL	F6H	IAP Control Register	-	-	-	-	PAYTIMES [1:0]		CMD[1:0]		xxxx0000b

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

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

Bit Number	Bit Mnemonic	Description
7 ~ 0	**IAPKEY[7: 0]**	Enable EEPROM/IAP function and operation time limit configuration, Written values must be non-zero: ① Enable IAP function; ② If no IAP writing command is received after n system clocks, IAP function will be reclosed.

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

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

Bit Number	Bit Mnemonic	Description
7 ~ 0	**IAPADR[7: 0]**	EEPROM/IAP the lower 8 bits of a write address

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

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	-	-	-	IAPADR[12:8]				
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
4~0	**IAPADR[12:8]**	EEPROM/IAP writing address high 5-bit
7~5	**-**	Reserved

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

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

Bit Number	Bit Mnemonic	Description
7 ~ 0	**IAPADER[7: 0]**	IAP Extended Address: 0x00: MOVC and IAP programming for Code 0x01: Read operation for user ID area, write operation is not allow. 0x02: MOVC and IAP programming for EEPROM **Note: EEPROM can be erased 100,000 times. User erasures should not exceed EEPROM rated burn times, otherwise an exception may occur!** 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 EEPROM / IAP

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

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

Bit Number	Bit Mnemonic	Description
3 ~ 2	**PAYTIMES[1: 0]**	Upon EEPROM/IAP writing operation, CPU Hold Time length configuration 00: Configure CPU HOLD TIME 4mS@12/6/2MHz 01: Configure CPU HOLD TIME 2mS@12/6/2MHz 10: Configure CPU HOLD TIME 1mS@12/6/2MHz 11: Reserved **Note: The CPU Hold is for PC pointer, other functional module continues to work; interrupt flag was saved, and interrupt will be generated after completing Hold, but only the last**

		time of interrupt can be saved. Recommended Selection: 2.7V ~ 5.5 V for V_{DD}, 10 is available 2.4V ~ 5.5V for V_{DD}, 01 or 00 is available
1 ~ 0	**CMD[1: 0]**	EEPROM / IAP writing operating command 10: Write Others: Reserved **Note: The statement of EEPROM/IAP write operation shall be followed by at least 8 NOP instructions to guarantee subsequent instruction can be implemented normally after finishing IAP operation!**

16.2 EEPROM / IAP OPerating procedures:

Writing procedure of the SC92F837X EEPROM/IAP are shown below:

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

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

③ Write address into {IAPADR[12: 8], IAPADR[7: 0]} (target address of EEPROM/IAP operation ready);

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

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

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

Notes:

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

16.2.1 128 bytes Independent EEPROM Operating Demo program

```
#include ”intrins.h”

unsigned char EE_Add;

unsigned char EE_Data;

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

C Demo Program of EEPROM Write Operation:

```
EA = 0;                         // Disable global Interrupt
IAPADE = 0x02;                  //Select EEPROM data memory
IAPDAT = EE_Data;               //Transmit data to EEPROM data register
IAPADH = 0x00;                  //High-bit address default write 0x00
IAPADL = EE_Add;                //Write EEPROM target address low bit
IAPKEY = 0xF0;                  //This value can be adjusted as required; it shall guarantee that
                                // The time interval between this instruction implemented and writing IAPCTL value shall be less than 240 (0xf0) system clocks, or else, IAP function is closed;
                                //Pay special attention to enabling interrupt;
IAPCTL = 0x0A;                  //Implement EEPROM write operation, 1ms@12M/6M/2M;
_nop_ ();                       //Wait (at least 8 _nop_ ())
_nop_ ();
_nop_ ();
_nop_ ();
_nop_ ();
_nop_ ();
_nop_ ();
_nop_ ();
_nop_ ();
IAPADE = 0x00;                  //Return to ROM data memory
EA = 1;                         //Enable master interrupt
```

C Demo Program of EEPROM Read Operation

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

16.2.2 8K bytes Code memory IAP Operating Demo program

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

C Demo Program of IAP Write Operation:

```
IAPADE = 0x00;                //Select Code memory
IAPDAT = IAP_Data;            //Transmit data to IAP data register
IAPADH = (unsigned char) ( (IAP_Add >> 8));   //Write IAP target address high bit
IAPADL = (unsigned char)IAP_Add;              //Write IAP target address low bit
IAPKEY = 0xF0; //This value can be adjusted as required; it shall guarantee this //instruction is implemented to assigned IAPTL value;
               // Time interval shall be less than 240 (0xf0) system clocks, or else, IAP function is closed;
               //Pay special attention upon starting interrupt
IAPCTL = 0x0A;                //Implement EEPROM write operation, 1ms@12M/6M/2M;
_nop_ ();                     //Wait (at least 8 _nop_ ())
_nop_ ();
_nop_ ();
_nop_ ();
_nop_ ();
_nop_ ();
_nop_ ();
_nop_ ();
```

C Demo Program of IAP Read Operation:

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

Note: IAP operation in 32K bytes Code memory has certain risks, the user shall implement corresponding safety measures in software. Incorrect operation may result in the user program to be rewritten! Unless such function is required by the user (such as used for remote program update, etc.), it is not recommended to used by the user.

17 CheckSum Module

The SC92F837X is equipped with a check sum module, which is used for generating 16-bit check sum of code memory in real time. The user can compare such check sum with theoretical value to monitor whether the contents in code memory are correct.

Note: Check sum is the sum of data in the whole code memory, which is the data of 0000H~1FFDH address unit. If there are residual values from the user's last operations in address unit, it may result in inconsistency of check sum with theoretical value. Therefore, it is recommended that the user shall erase the whole Code memory or write 0 before programming code so as to guarantee the consistency between check sum and theoretical value.

17.1 CheckSum-Related Registers

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

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

Bit Number	Bit Mnemonic	Description
7 ~ 0	**CHKSUML [7: 0]**	CheckSum Result Register Low Bit

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

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

Bit Number	Bit Mnemonic	Description
7 ~ 0	**CHKSUMH [7: 0]**	CheckSum Result Register High Bit

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

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

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

18 Electrical Characteristics

18.1 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	Ambient temperature	-40	85	℃
T_{STG}	Storage temperature	-55	125	℃

18.2 Recommended Operating Conditions

Symbol	Parameter	Min Value	Max Value	Unit	System Clock Frequency
V_{DD1}	Operating voltage	2.4	5.5	V	-
T_A	Ambient temperature	-40	85	℃	-

18.3 DC Characteristics

VDD = 5V, TA = +25℃, unless otherwise specified

Symbol	Parameter	Min Value	Typical Value	Max Value	Unit	Testing Conditions
Current						
I_{op1}	Operating current	-	7.0	-	mA	f_{SYS} =12MHz
I_{op2}	Operating current	-	6.0	-	mA	f_{SYS} =6MHz

I_{op3}	Operating current	-	5.0	-	mA	f_{SYS} =2MHz
I_{pd1}	Standby Current (Power Down Mode)	-	0.7	1.0	μA	
I_{IDL1}	Standby Current (IDLE Mode)	-	7.5	-	mA	
I_{BTM}	Base Timer Operating Current	-	5.6	7.0	μA	BTMFS[3: 0]= 1000 One interrupt occurs for every 4.0 seconds
I_{WDT}	WDT Current	-	4.3	5.1	μA	WDTCKS[2: 0]= 000 WDT overflows every 500ms
I_{TK1}	TouchKey operating current (High Sensitivity Mode)	-	1.4	1.7	mA	
I_{TK2}	TouchKey operating current (High reliability model)	-	0.3	1.0	mA	
IO Port Features						
V_{IH1}	Input high voltage	$0.7V_{DD}$	-	V_{DD}+0.3	V	
V_{IL1}	Input low voltage	-0.3	-	$0.3V_{DD}$	V	
V_{IH2}	Input high voltage	$0.8V_{DD}$	-	V_{DD}	V	Schmidt trigger input: RST/tCK/SCK
V_{IL2}	Input low voltage	-0.2	-	$0.2V_{DD}$	V	
I_{OL1}	Output low current	-	25	-	mA	V_{Pin}=0.4V
I_{OL2}	Output low current	-	50	-	mA	V_{Pin}=0.8V

I_{OH1}	Output high current P1	-	21	-	mA	V_{Pin}=4.3V
I_{OH2}	Output high current P1	-	11	-	mA	V_{Pin}=4.7V
I_{OH3}	Output high current P0/P2	-	21	-	mA	V_{Pin}=4.3V Pxyz=0, I_{OH} level 0
	Output high current P0/P2	-	17	-	mA	V_{Pin}=4.3V Pxyz=1, I_{OH} level 1
	Output high current P0/P2	-	11	-	mA	V_{Pin}=4.3V Pxyz=2, I_{OH} level 2
	Output high currentP0/P2	-	5	-	mA	V_{Pin}=4.3V Pxyz=3, I_{OH} level 3
I_{OH4}	Output high current P0/P2	-	11	-	mA	V_{Pin}=4.7V Pxyz=0, I_{OH} level 0
	Output high current P0/P2	-	8	-	mA	V_{Pin}=4.7V Pxyz=1, I_{OH} level 1
	Output high current P0/P2	-	6	-	mA	V_{Pin}=4.7V Pxyz=2, I_{OH} level 2
	Output high current P0/P2	-	3	-	mA	V_{Pin}=4.7V Pxyz=3, I_{OH} level 3
R_{PH1}	Pull-up resistance	-	32	-	kΩ	

VDD = 3.3V, TA = +25℃, unless otherwise specified

Symbol	Parameter	Min Value	Typical Value	Max Value	Unit	Testing Conditions
Current						
I_{op4}	Operating current	-	5.1	-	mA	f_{SYS} =12MHz
I_{op5}	Operating current	-	4.2	-	mA	f_{SYS} =6MHz
I_{op6}	Operating current	-	3.5	-	mA	f_{SYS} =2MHz
I_{pd2}	Standby Current (Power Down Mode)	-	0.6	1	uA	
I_{IDL2}	Standby Current (IDLE Mode)	-	5.2	-	mA	
I_{TK3}	TouchKey operating current (High Sensitivity Mode)	-	1.2	1.5	mA	
I_{TK4}	TouchKey operating current (High reliability model)	-	0.3	1.0	mA	
I/O Port Features						
V_{IH3}	Input high voltage	0.7VDD	-	VDD+0.3	V	
V_{IL3}	Input low voltage	-0.3	-	0.3VDD	V	
V_{IH4}	Input high voltage	0.8VDD	-	VDD	V	Schmidt trigger input: RST/tCK/SCK
V_{IL4}	Input low voltage	-0.2	-	0.2VDD	V	
I_{OL3}	Output low current	-	24	-	mA	V_{Pin}=0.4V
I_{OL4}	Output low current	-	44	-	mA	V_{Pin}=0.8V

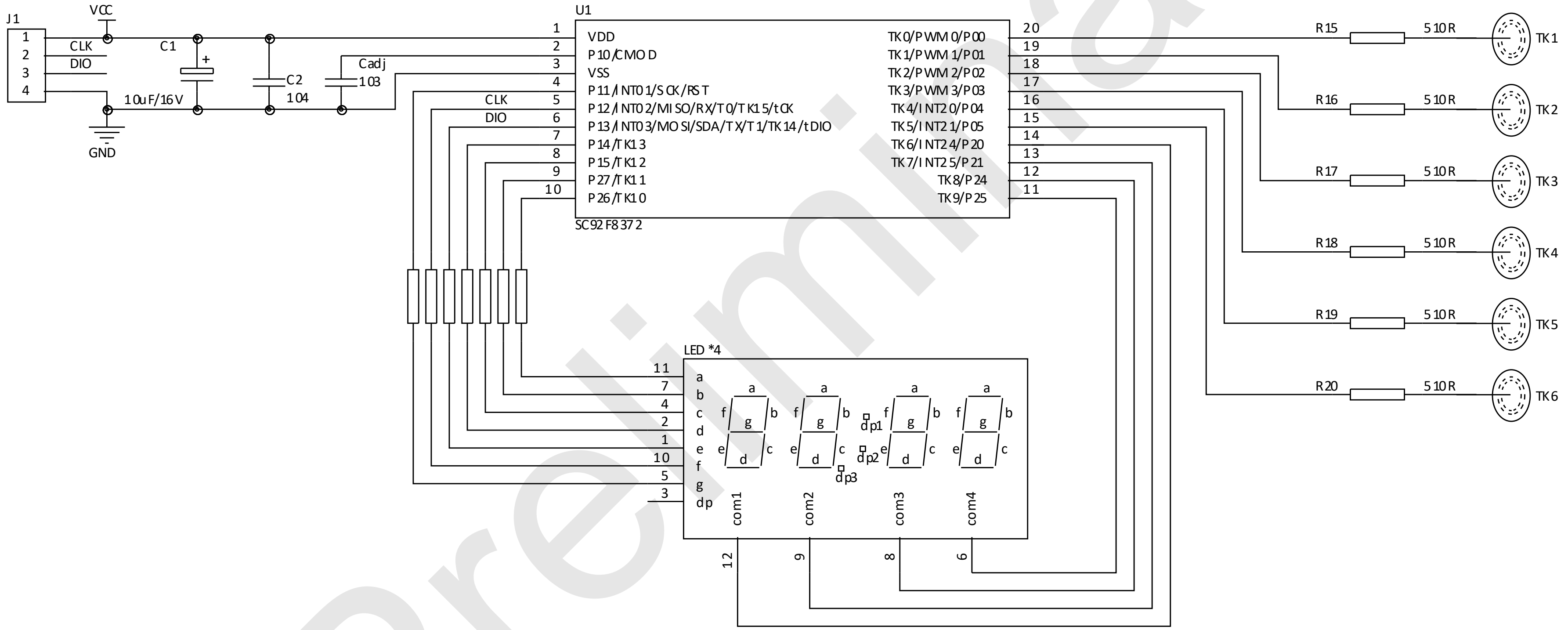
I_{OH5}	Output high current	-	6	-	mA	V_{Pin}=3.0V
R_{PH2}	Pull-up resistance	-	54	-	kΩ	

18.4 AC Characteristics

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

Symbol	Parameter	Min	Typical	Max	Unit	Conditions
T_{POR}	Power On Reset time	-	1	1.5	ms	
T_{PDW}	Power Down Mode waking-up time	-	1	1.5	ms	
T_{Reset}	Reset Pulse Width	18	-	-	μs	Valid for Low level
f_{HRC}	RC oscillation stability	23.76	24	24.24	MHz	V_{DD}=2.9~5.5V T_A=-20~85 ℃

19 Application Circuit



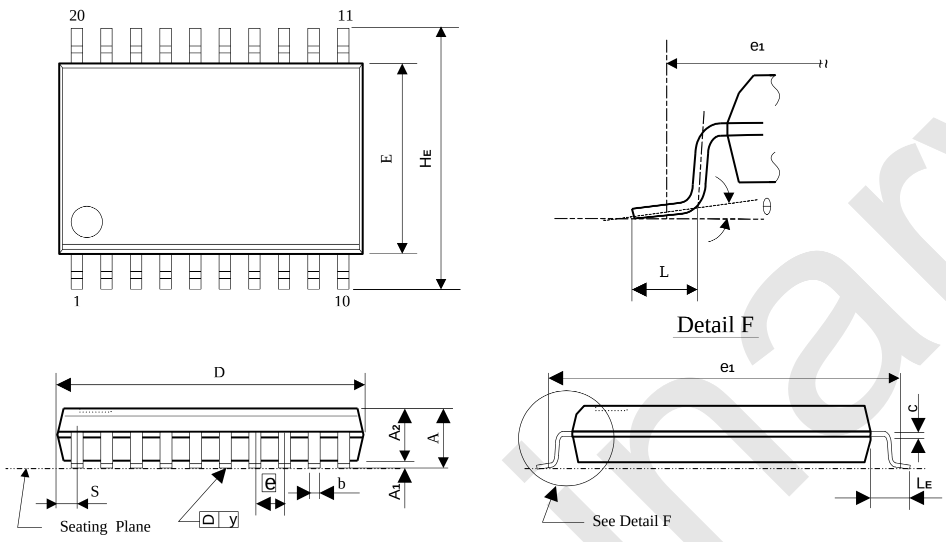
20 Ordering Information

PRODUCT NO	PKG	PACKING
SC92F8372M20U	SOP20L	TUBE
SC92F8372X20U	TSSOP20L	TUBE
SC92F8371M16U	SOP16L	TUBE
SC92F8370M08U	SOP8L	TUBE

21 Packageing Information

SC92F8372M20U

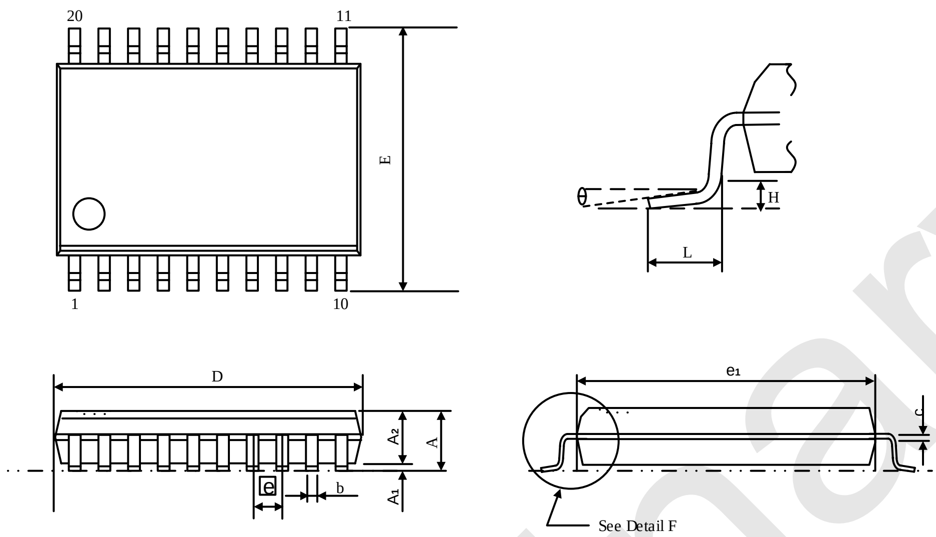
SOP20L(300mil) Dimension Unit: mm



Symbol	mm (milimetre)		
	Min	Normal	Max
A	2.40	2.56	2.65
A1	0.100	0.200	0.300
A2	2.240	2.340	2.440
b	0.35	--	0.47
C	0.25		
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°

SC92F8372X20U

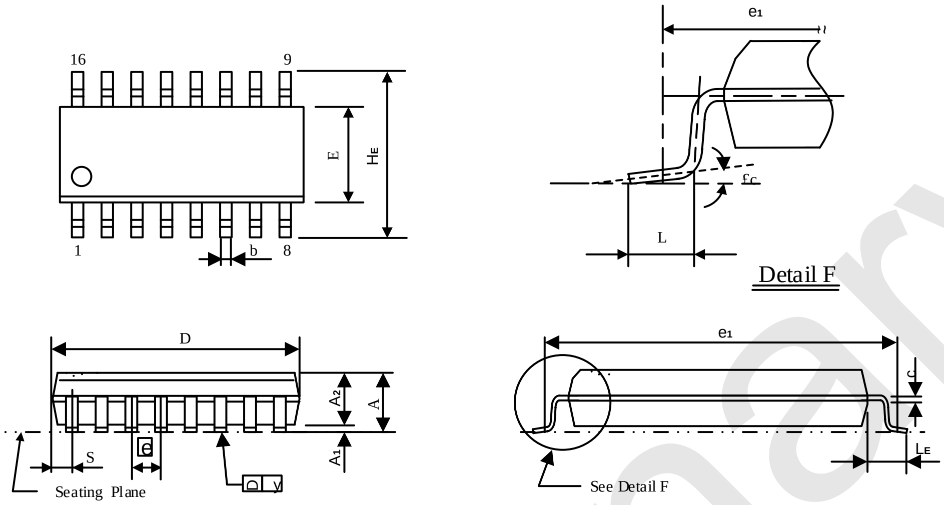
TSSOP20L Dimension Unit: mm



Symbol	mm (milimetre)		
	Min	Normal	Max
A	-	-	1.200
A1	0.050	-	0.150
A2	0.800	-	1.050
b	0.190	-	0.300
c	0.090	-	0.200
D	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

SC92F8371M16U

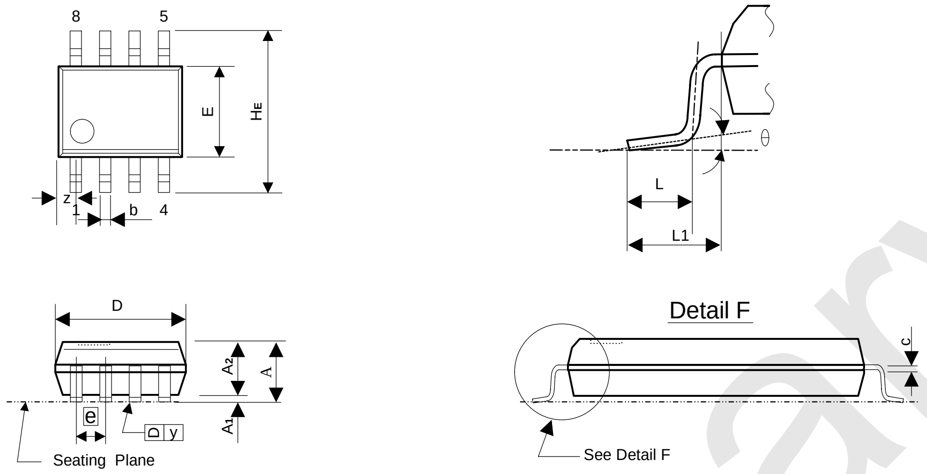
SOP16L(150mil) Dimension Unit: mm



Symbol	mm (milimetre)		
	Min	Normal	Max
A	1.500	1.625	1.750
A**1**	0.050	0.1375	0.225
A**2**	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°

SC92F8370M08U

SOP8L(150mil) Dimension Unit: mm



Symbol	mm (milimetre)		
	Min	Normal	Max
A	1.500	1.600	1.700
A1	0.100	0.150	0.200
A2	-	1.450	-
b	0.356	0.406	0.456
C	0.143	0.203	0.263
D	4.700	4.900	5.100
E	3.810	3.860	3.910
HE	5.900	6.000	6.100
e	1.270(BSC)		
L	0.650	0.660	0.670
L1	0.950	1.050	1.150
θ	0°	-	10°

22 Revision History

Version	Notes	Date
V0.1	Initial Release.	August 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 August 2022. In the actual production design, please refer to the latest data manual of each product and other relevant materials.