

Super High-speed 1T 8051 Core Flash MCU, 32 Kbytes Flash, 4 Kbytes SRAM, 1 Kbytes Independent LDROM, 23-channel dual-mode TouchKey module, 12-bit High-speed 1M ADC, 16-bit PWM, 5 Timers, MDU, UART, 3×USCI, Smart Card, CRC Check Module

1 General Description

SC95F8523/8522/8521 (hereinafter referred to as the SC95F852X) is a series of enhanced 1T 8051 core industry-standard Flash Microcontroller unit (MCU) with integrated TouchKey function, the instruction set is compatible with the standard 80C51 series.

The SC95F852X has a Super-high-speed 1T 8051 CPU core with an operating frequency of up to 32MHz.

The SC95F852X integrates a hardware multiplier and divider and dual DPTR data pointers to accelerate data operations and movement speed. The hardware multiplier and divider does not occupy CPU cycles, and the operation is implemented by hardware, and the speed is dozens of times faster than the multiplication and division speed realized by software; dual DPTR data pointers can be used to accelerate data storage and movement.

The SC95F852X has high performance and reliability, with a wide operating voltage of 2.0V~5.5V, a super-wide operating temperature of -40°C to 105°C, and powerful 6KV ESD and 4KV EFT capabilities. Using the industry-leading eFlash process, the Flash can be written more than 100,000 times stored for 100 years at room temperature.

The SC95F852X has a built-in low power consumption WDT Watchdog Timer. It has a 4-level selectable voltage LVR low voltage reset function and a system clock monitoring function. It has low power consumption capability in operation and power-down modes. Under normal operating mode: about 4.5mA@32M at 5V.

The SC95F852X series is also integrated with super rich hardware resources: multiple built-in 23-channel dual-mode (high sensitivity/high reliability) touch circuits, 32K bytes Flash ROM, 4K bytes SRAM, 1K bytes independent LDROM; up to 26 GP I/O, 13 IO can be externally interrupted, 5 16-bit timer, 11 channels 12-bit high-precision 1M high-speed ADC, 8-channel 16-bit dead zone complementary PWM, internal high-frequency 32/16/8/4MHz oscillator and low-frequency 32.768kHz oscillator, external 32.768kHz resources such as crystal oscillators, UART, SPI, TWI and other communication interface resources. The SC95F852X also integrates 11 channels 12-bit high-precision 1M high-speed ADC with 1.024V/2.048V reference voltage function. 1 UART, 3 USCI (UART/TWI/SPI). So many functions are integrated in the SC95F852X, which can reduce the number of peripheral components of the system and save board space and system cost.

The SC95F852X is very convenient for development and debugging, with ISP (In-System Programming), ICP (In-Circuit Programming) and IAP (In-Application Programming) . Allow the chip to debug and upgrade the program memory directly on the circuit board when the chip is online or powered.

The SC95F852X has very excellent anti-jamming performance, high reliability, large resources, multiple interfaces, low power consumption, high efficiency, etc. It is very suitable for Intelligent House System, industrial control, Internet of Things (IoT), medical, wearable devices, Consumer goods and other application areas.

Super High-speed 1T 8051 Core Flash MCU, 32 Kbytes Flash, 4 Kbytes SRAM, 1 Kbytes Independent LDRM, 23-channel dual-mode TouchKey module, 12-bit High-speed 1M ADC, 16-bit PWM, 5 Timers, MDU, UART, 3×USCI, Smart Card, CRC Check Module

2 Features

Operating Conditions

- Voltage Range: 2.0V~5.5V
- Temperature Range: -40°C ~ +105°C

ESD&EFT

- ESD 6KV
- EFT 4KV

Package

- 28 PIN: SOP28 / TSSOP28
- 20 PIN: SOP28 / TSSOP28
- 16 PIN: SOP16

CPU

- Super-high-speed 1T 8051 core
- The instruction set compatible with 8051
- The execution speed is about twice that of other 1T 8051
- Double data pointers (DPTRs)

Storage

- 32K bytes Flash ROM
 - Divided into 64 sectors
 - 512 bytes per sector
 - Can be rewritten 100,000 times
 - The data written-in has more than 100-year preservation life in the ambient temperature of 25°C
- IAP (In Application Programming): The area allowed IAP operation in Flash can be set to 0K, 1K, 2K or 32K by Code Option.
- BootLoader: Built-in 1K bytes LDRM
 - Boot from APROM or LDRM can be selected through the Code Option Settings item

- ◆ Unique ID: 96 bit unique ID memory
- SRAM:
 - Internal 256 bytes of in-chip direct access RAM
 - 4 Kbytes of external RAM (XRAM)
 - 16 bytes Indirect access RAM in the chip read and write PWM duty cycle SFR: 1040H~104FH 16 bytes in total

Flash Programming and Emulation

- 2-wire JTAG programming and emulation interface

Clock Source

- Built-in high frequency 32MHz oscillator (HRC)
 - The system clock frequency (fsys) of the IC can be selected and set by the programmer as: 32/16/8/4MHz
 - Frequency Error:
 - ◆ Within $\pm 1\%$ @ -20 ~ 85°C
 - ◆ Within $\pm 2\%$ @ -40 ~ 105°C
 - The system clock can be automatically calibrated by 32.768kHz external crystal oscillator, after calibration HRC accuracy can be infinitely close to the accuracy of external 32.768kHz crystal oscillator.
- Built-in low-frequency crystal oscillator circuit: 32.768k oscillator can be connected externally as a Base Timer clock source.
- Built-in low-frequency 32.768kHz oscillator (LRC): used as the clock source for Base Timer and WDT.

Low-voltage Reset (LVR)

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

Interrupts (INT)

- Up to 16 interrupt sources including Timer0~Timer4, INT0~2, ADC, PWM, UART, USCI0~2, Base Timer, TK
- External interrupt INT0~2 contains 3 interrupt vectors, 13 intermediate fractures. All can set up rising edge, falling edge, double edge interrupt.
- Two-level interrupt priority capability

Digital Peripheral

- GPIO
 - Up to 26 bidirectional independently controllable I/O ports, independent setting of pull-up resistors

Each source drive capability of I/O has 4 levels of control

- All I/Os have large sink current drive capability (50mA)
- Built-in WDT, optional clock frequency division ratio
- 5 Timers: Timer0~4
- 8-channel 16-bit PWM
 - Can be set to independent mode or complementary mode: 8 channels of PWM share cycle and the duty cycle can be set separately in the independent mode
 - In complementary mode, four sets of complementary PWM waveforms with dead zones can be output simultaneously
 - The output waveform can be reversed and can be set to center-aligned or edge-aligned
 - Fault detection mechanism
- One independent UART communication port (UART0)
- Three UART/SPI/TWI communication interfaces (USCI)
 - USCI0 has 7816 mode
 - USCI0's SPI interface has a 16-bit 8-level read/write FIFO cache in the main mode
- Integrated with 16 * 16-bit hardware Multiplier-Divide Unit (MDU)
- Built-in CRC check module

Analog Peripheral

- 23-channel dual-mode TouchKey (TK) circuit.
 - High-sensitivity mode applicable to TouchKey sensor, proximity induction and other TouchKey applications featuring high requirements on sensitivity
 - High reliability mode features very strong anti-interference which is able to pass 10V dynamic CS test
 - Support low power consumption mode.
 - Complete development support: High-flexible touch software library, intelligent debugging software .
- 11-channel 12-bit high-speed ADC
 - 1MHz sampling clock
 - The ADC reference voltages is optional:
 - ◆ Internal 2.048V
 - ◆ Internal 1.024V
 - ◆ VDD

- The internal ADC can directly measure $1/4V_{DD}$ voltage

Power Saving Mode

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

Naming Rules for 95 Series Products

Name	SC	95	F	8	5	2	3	X	M	28	U
S/R	①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩	⑪

S/R	Meaning
①	SinOne Chip abbreviation
②	Name of product series
③	Product Type (F: Flash MCU)
④	Serial Number: 7: GP Series, 8: TK series
⑤	ROM Size: 1 for 2K, 2 for 4K, 3 for 8K, 4 for 16K, 5 for 32K and 6 for 64K
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	Package Type: (D: DIP; M: SOP; X: TSSOP; F: QFP; P: LQFP; Q: QFN; K: SKDIP)
⑨	Number of Pins.
⑩	Number of Pins.
⑪	Packaging Mode: (U: Tube; R: Tray; T: Reel)

Ordering Information

PRODUCT ID	PACKAGE	PACK
SC95F8523M28U	SOP28	TUBE
SC95F8523X28U	TSSOP28	TUBE
SC95F8522M20U	SOP20	TUBE
SC95F8522X20U	TSSOP20	TUBE
SC95F8521M16U	SOP16	TUBE

Contents

1 GENERAL DESCRIPTION	1
2 FEATURES.....	2
NAMING RULES FOR 95 SERIES PRODUCTS.....	6
ORDERING INFORMATION	7
CONTENTS	8
3 PIN DESCRIPTION	12
3.1 Pin Configuration	12
3.2 28/20/16 Pin Configuration	14
4 INNER BLOCK DIAGRAM	18
5 MEMORY ORGANIZATION	19
5.1 Flash ROM.....	19
5.1.1 Flash ROM Sectors	19
5.2 In Application Programming (IAP)	20
5.2.1 IAP Operation Related Registers	20
5.2.2 IAP Operating Demo Program In C Language	26
5.3 BootLoader.....	29
5.3.1 BootLoader Mode operation related registers.....	29
5.4 Encryption.....	34
5.5 Unique ID (Unique ID) Area	34
5.5.1 Unique ID Read Operating Demo Program In C Language	35
5.6 Programming	36
5.7 Code Option Area (User Programming Settings)	38
5.7.1 Option-related Registers Operation Instructions.....	42
5.8 SRAM	43
5.8.1 Internal 256 Bytes SRAM.....	43
5.8.2 External 4096 Bytes SRAM	45
5.8.3 PWM 16 Bytes SRAM	46
6 SPECIAL FUNCTION REGISTER (SFR)	47
6.1 SFR Mapping.....	47
6.2 Register Summary	48
7 POWER, RESET AND SYSTEM CLOCK	55
7.1 Power Circuit.....	55
7.2 Power-on Reset	55
7.2.1 Reset Stage	55
7.2.2 Loading Information Stage	55
7.2.3 Normal Operation Stage	55
7.3 Reset Modes	55

7.3.1 External Reset.....	57
7.3.2 Low-voltage Reset LVR.....	57
7.3.3 Power-on Reset (POR).....	57
7.3.4 Watchdog Reset (WDT).....	58
7.3.5 Software Reset.....	59
7.3.6 Register Reset Value.....	60
7.4 High-speed RC Oscillator	60
7.5 Low-speed RC Oscillator and Low-speed Clock Timer	63
7.6 Power Saving Modes	66
8 CPU AND INSTRUCTION SET	68
8.1 CPU.....	68
8.2 Addressing Mode	68
8.2.1 Immediate Addressing	68
8.2.2 Direct Addressing	68
8.2.3 Indirect Addressing.....	68
8.2.4 Register Addressing.....	68
8.2.5 Relative Addressing.....	68
8.2.6 Indexed Addressing	69
8.2.7 Bits Addressing.....	69
8.3 Introduction of Common Special Function Registers of 8051 Core.....	69
Program Counter PC	69
Accumulator ACC (E0H)	69
B Register (F0H)	69
Stack Pointer SP (81H).....	70
Data Pointers DPTR0 (82H, 83H), DPTR1 (84H, 85H) And Its Selection Register DPS (86H)	71
9 INTERRUPTS	74
9.1 Interrupt Source and Vector	74
9.2 Interrupt Structure Diagram.....	77
9.3 Interrupt Priority	78
9.4 Interrupt Processing Flow	78
9.5 Interrupt-related SFR Registers.....	78
10 TIMER/COUNTER T0 AND T1	90
10.1 T0 and T1-related Registers	90
10.2 T0 Operating Modes.....	95
10.3 T1 Operating Mode	97
11 TIMER/COUNTER T2/T3/T4.....	99
11.1 T2/3/4-related Registers	99
11.2 Timer2	100
11.3 Timer3	104
11.4 Timer4	108
11.5 T2/3/4 Operating Modes.....	111
11.5.1 Operating Mode 0: 16-bit Capture.....	112
11.5.2 Operating Mode 1: 16-bit Auto-Reload Timer	113

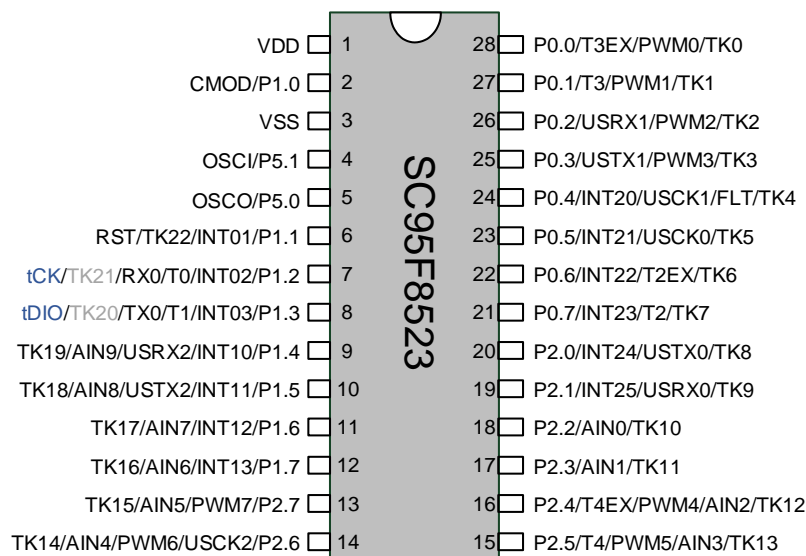
11.5.3 Operating Mode 2: Baud Rate Generator	114
11.5.4 Operating Mode 3: Programmable Clock Output	115
12 MULTIPLIER-DIVIDER UNIT (MDU)	117
13 PWM	120
13.1 PWM Structure Diagram.....	121
13.2 PWM General Configuration Register	122
13.2.1 PWM General Configuration Register	122
13.2.2 PWM Fault Detection Function Setting	129
13.3 PWM Independent Mode.....	131
13.3.1 PWM Independent Mode Block Diagram	131
13.3.2 PWM Independent Mode Duty Cycle Configuration	132
13.4 PWM Complementary Model	134
13.4.1 PWM Block Diagram of Complementary Mode.....	134
13.4.2 PWM Complementary Mode Duty Cycle Configuration.....	134
13.4.3 PWM Complementary Mode Dead Time Setting	136
13.4.4 PWM Dead Zone Output Waveform	136
13.5 PWM Waveforms and Directions	137
14 GENERAL-PURPOSE I/O (GPIO).....	139
14.1 GPIO Structure Diagram.....	139
14.2 I/O Port-related Registers	141
15 UART0	149
15.1 Baud Rate of Serial Communication	151
16 SPI/TWI/UART SERIAL INTERFACE (USCI0/1/2)	152
16.1 SPI.....	155
16.1.1 SPI0	155
16.1.2 SPI1/2.....	162
16.2 TWI.....	168
16.2.1 Signal Description	174
16.2.2 Slave Operating Mode	174
16.2.3 Slave Mode Operation Steps	177
16.2.4 Master Mode	177
16.2.5 Master Mode Operation Steps	179
16.3 Serial Interface (UART).....	180
17 SMART CARD (SC)	184
17.1 Functional Description	184
17.2 SC Control Register.....	184
17.3 Operating Mode.....	188
17.3.1 Smartcard Description	188
17.3.2 Smart card data transmission	190
17.3.3 Initialization character TS	190
17.3.4 Error signals and character duplicates	191
17.3.5 Extended protection time	192
18 ANALOG-TO-DIGITAL CONVERTER (ADC)	193

18.1 ADC-Related Registers.....	193
18.2 ADC Conversion Steps.....	199
19 DUAL MODE TOUCH CIRCUIT	200
19.1 Power Consumption Mode of Touch circuit	200
19.2 Touch Mode	200
20 CRC MODULE.....	202
20.1 CRC Check Operation Related Registers.....	203
21 ELECTRICAL CHARACTERISTICS	209
21.1 Absolute Maximum Ratings	209
21.2 Recommended Operating Conditions	209
21.3 Flash ROM Characteristics	210
21.4 LVR Characteristics.....	210
21.5 DC Characteristics	211
21.6 AC Characteristics	215
21.7 ADC Characteristics.....	215
22 APPLICATION CIRCUIT	218
23 PACKAGE INFORMATION.....	219
24 REVISION HISTORY	224

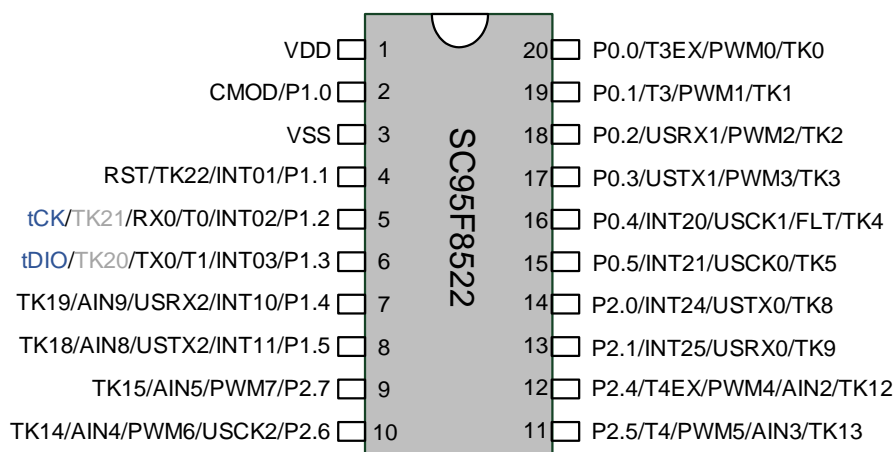
3 Pin Description

3.1 Pin Configuration

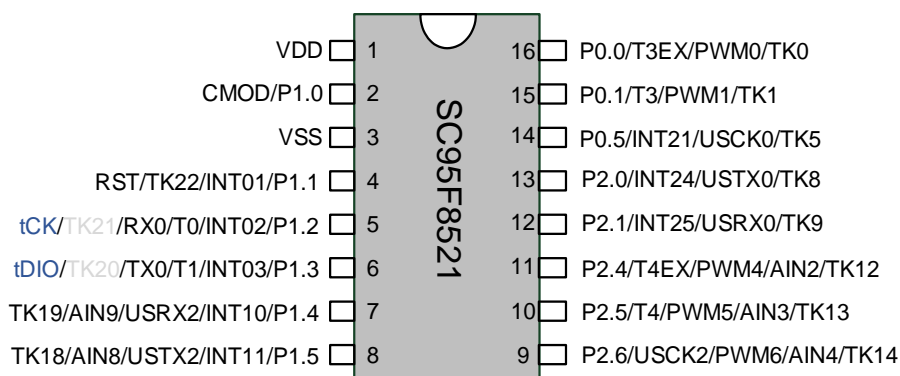
Note: In consideration of multiplexing of TK20/TK21 and TK debugging communication ports of the SC95F852X, if it is required to use the TK debugging function, please avoid using TK20/TK21!



SC95F8523 Pin Diagram



SC95F8522Pin Diagram



SC95F8521Pin Diagram

3.2 28/20/16 Pin Configuration

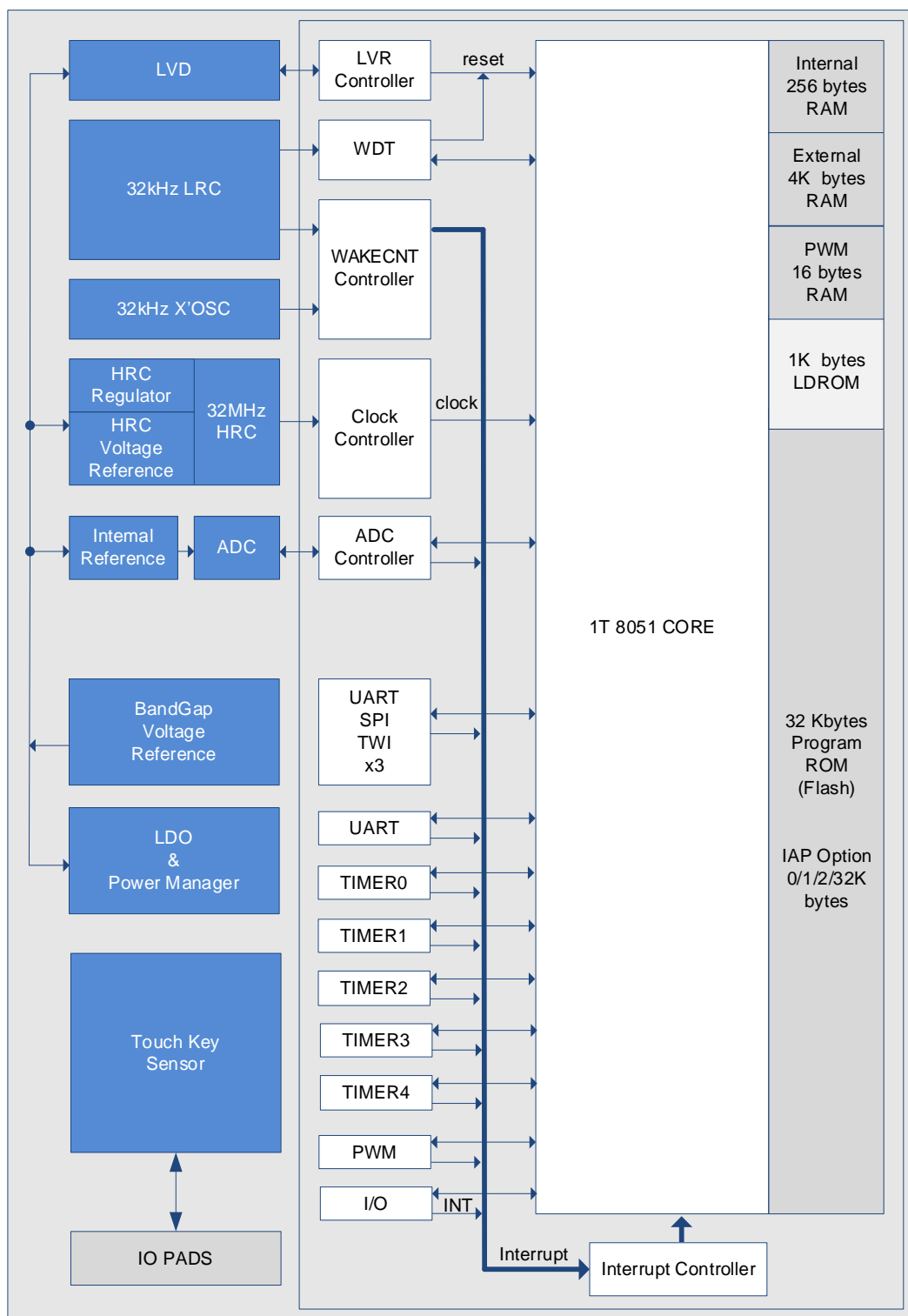
Pin number			Pin Name	Type	Description
28PIN	20PIN	16PIN			
1	1	1	VDD	Power	Power
2	2	2	P1.0/CMOD	I/O	P1.0: GPIO P1.0 CMOD: Touch Key External Capacitance
3	3	3	VSS	Power	Ground
4	-	-	P5.1/OSCI	I/O	P5.1: GPIO P5.1 OSCI: External crystal oscillator input
5	-	-	P5.0/OSCO	I/O	P5.0: GPIO P5.0 OSCO: External crystal oscillator output
6	4	4	P1.1/INT01/TK22/RST	I/O	P1.1: GPIO P 1.1 INT01: Input 1 of external interrupt 0 TK22: TK Channel 22 RST: Reset Pin
7	5	5	P1.2/INT02/T0/RX0/TK21/tCK	I/O	P1.2: GPIO P1.2 INT02: Input 2 of external interrupt 0 T0: Timer/Counter 0 External Input RX0: UART0 Receiver TK21: TK channel 21, if it is required to use the TK debugging function, please avoid using TK21 tCK: Programming and Emulation Clock Pin
8	6	6	P1.3/INT03/T1/TX0/TK20/tDIO	I/O	P1.3: GPIO P1.3 INT03: Input 3 of external interrupt 0 T1: Timer/Counter 1 External Input TX0: UART 0 Transmitter TK20: TK channel 20, if it is required to use the TK debugging function, please avoid using TK20 tDIO: Programming and Emulation Data Pin
9	7	7	P1.4/INT10/USRX2/AIN9/TK19	I/O	P1.4: GPIO P1.4

					INT10: Input 0 of external interrupt 1 USRX2: USCI2 MISO/RX AIN9: ADC Input Channel 9 TK19: TK Channel 19
10	8	8	P1.5/INT11/USTX2/AIN8/TK18	I/O	P1.5: GPIO P1. 5 INT11: Input 1 of external interrupt 1 USTX2: USCI2 MOSI/SDA/TX AIN8: ADC Input Channel 8 TK18: TK Channel 18
11	-	-	P1.6/INT12/AIN7/TK17	I/O	P1.6: GPIO P1.6 INT12: Input 2 of external interrupt 1 AIN7: ADC Input Channel 7 TK17: TK Channel 17
12	-	-	P1.7/INT13/AIN6/TK16	I/O	P1.7: GPIO P1.7 INT13: Input 3 of external interrupt 1 AIN6: ADC Input Channel 6 TK16: TK Channel 16
13	9	-	P2.7/PWM7/AIN5/TK15	I/O	P2.7: GPIO P2.7 PWM7: PWM7 Output AIN5: ADC Input Channel 5 TK15: TK Channel 15
14	10	9	P2.6/USCK2/PWM6/AIN4/TK14	I/O	P2.6: GPIO P2.6 USCK2: USCI2 SCK PWM6: PWM6 Output AIN4: ADC Input Channel 4 TK14: TK Channel 14
15	11	10	P2.5/T4/PWM5/AIN3/TK13	I/O	P2.5: GPIO P2.5 T4: Timer 4 external input PWM5: PWM5 Output AIN3: ADC Input Channel 3 TK13: TK Channel 13
16	12	11	P2.4/T4EX/PWM4/AIN2/TK12	I/O	P2.4: GPIO P2.4 T4EX: Timer 4 External capture signal input PWM4: PWM4 Output AIN2: ADC Input Channel 2

					TK12: TK Channel 12
17	-	-	P2.3/AIN1/TK11	I/O	P2.3: GPIO P 2.3 AIN1: ADC Input Channel 1 TK11: TK Channel 11
18	-	-	P2.2/AIN0/TK10	I/O	P2.2: GPIO P2.2 AIN0: ADC Input Channel 0 TK10: TK Channel 10
19	13	12	P2.1/INT25/MISO/RX1/TK9	I/O	P2.1: GPIO P2.1 INT25: Input 5 of external interrupt 2 USRX0: USCI0 MISO/RX TK9: TK Channel 9
20	14	13	P2.0/INT24/USTX0/TK8	I/O	P2.0: GPIO P2.0 INT24: Input 4 of external interrupt 2 USTX0: USCI0 MOSI/SDA/TX/SC_DAT TK8: TK Channel 8
21	-	-	P0.7/INT23/T2/TK7	I/O	P0.7: GPIO P 0.7 INT23: Input 3 of external interrupt 2 T2: Timer/Counter 2 External input TK7: TK Channel 7
22	-	-	P0.6/INT22/T2EX/TK6	I/O	P0.6: GPIO P0.6 INT22: Input 2 of external interrupt 2 T2EX: External Signal Input Capture for Timer2 TK6: TK Channel 6
23	15	14	P0.5/INT21/USCK0/TK5	I/O	P0.5: GPIO P0.5 INT21: Input 1 of external interrupt 2 USCK0: USCI0 SCK/SC_CLK TK5: TK Channel 5
24	16	-	P0.4/INT20/USCK1/FLT/TK4	I/O	P0.4: GPIO P0.4 INT20: Input 0 of external interrupt 2 USCK1: USCI1 SCK FLT: PWM fault detection input pin TK4: TK Channel 4
25	17	-	P0.3/USTX1/PWM3/TK3	I/O	P0.3: GPIO P0.3

					USTX1: USCI1 MOSI/SDA/TX PWM3: PWM3 output TK3: TK Channel 3
26	18	-	P0.2/USRX1/PWM2/TK2	I/O	P0.2: GPIO P0.2 USRX1: USCI1 MISO/RX PWM2: PWM2 output TK2: TK Channel 2
27	19	15	P0.1/T3/PWM1/TK1	I/O	P0.1: GPIO P0.1 T3: Timer 3 external input PWM1: PWM1 output TK1: TK Channel 1
28	20	16	P0.0/T3EX/PWM0/TK0	I/O	P0.0: GPIO P0.0 T3EX: Timer 3 External capture signal input PWM0: PWM0 output TK0: TK Channel 0

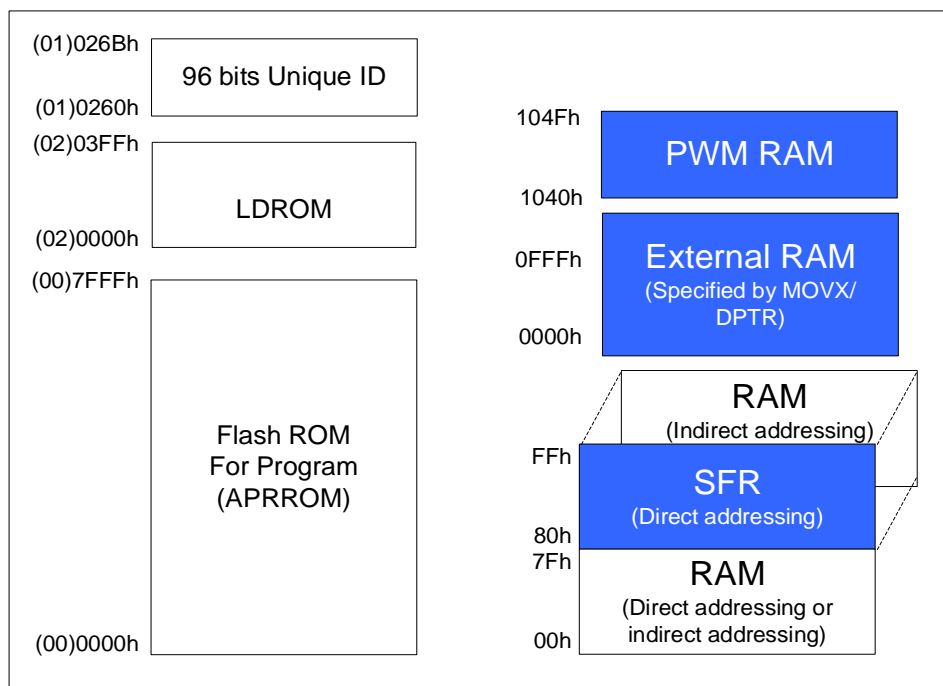
4 Inner Block Diagram



SC95F852X BLOCK DIAGRAM

5 Memory Organization

The structure of the SC95F852X's Flash ROM and SRAM are shown as follows:



Flash ROM and SRAM structure diagram

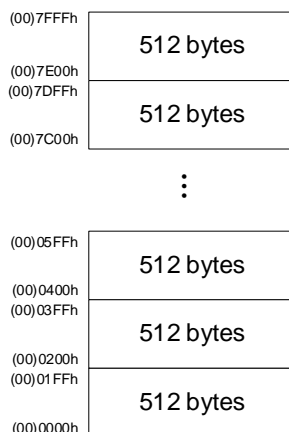
5.1 Flash ROM

The SC95F852X has 32.768K bytes of Flash ROM, the address is (00)0000H ~ (00)7FFFH, "00" in brackets is the extended address, which is set by the IAPADE register. Flash ROM can be programmed and erased by SC LINK provided by SinOne. The characteristics of this 32.768K bytes Flash ROM are as follows:

- ① Divided into 64 sectors
- ② 512 bytes per sector
- ③ Can be rewritten 100,000 times
- ④ The data written-in has more than 100-year preservation life in the ambient temperature of 25°C
- ⑤ In ICP mode, BLANK, PROGRAM, VERIFY, ERASE and READ functions are supported. The READ function is only valid for ICs with no security encryption function enabled
- ⑥ Secure Encryption: Optionally enable APROM (32.768K bytes Flash ROM) and LDROM secure encryption
- ⑦ Support IAP (In Application Programming).

5.1.1 Flash ROM Sectors

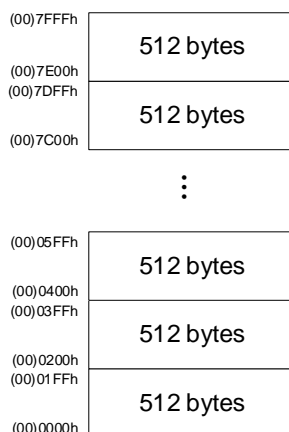
The SC95F852X has 32.768K bytes of Flash ROM divided into 64 sectors, each sector is 512 bytes, the sector to which the target address belongs will be forcibly erased by the programmer during writing, and then write data; When the user writes, must erasing it before writing.



SC95F852X 32K bytes Flash ROM Sectors

5.2 In Application Programming (IAP)

The 32.768K bytes Flash ROM can be set to 0K, 1K, 2K, or 32K through the Code Option Settings. Flash ROM is divided into 64 sectors from (00)0000H to (00)7FFFH. The "00" in brackets is the expanded address set by the IAPADE register:



SC95F852X 32K bytes Flash ROM Sectors

Note: During the IAP operation, the program counter will hold until the operation is completed.

5.2.1 IAP Operation Related Registers

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

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

Bit Number	Bit Mnemonic	Description
3~2	IAPS[1: 0]	<p>IAP spatial range selection</p> <p>00: Last 0K Flash ROM allows IAP operation</p> <p>01: Last 1K Flash ROM allows IAP operation</p> <p>10: Last 2K Flash ROM allows IAP operation</p> <p>11: Full Flash ROM allows IAP operation</p> <p>Note: The above setting items are invalid in BootLoader mode. The BootLoader program can perform IAP operation on the entire Flash ROM area.</p>

IAP Operation Related Register Description:

Symbol	Address	Description	7	6	5	4	3	2	1	0	POR
IAPKEY	F1H	Data protection register	IAPKEY[7: 0]								00000000b
IAPADL	F2H	IAP write address low register	IAPADR[7: 0]								00000000b
IAPADH	F3H	IAP write address high register	IAPADR[15: 8]								00000000b

IAPADE	F4H	IAP write to extended address register	IAPADER[7: 0]							00000000b
IAPDAT	F5H	IAP data register	IAPDAT[7: 0]							00000000b
IAPCTL	F6H	IAP control register	-	ERASE	SERASE	PRG	-	BTLD	CMD[1: 0]	x000x000b

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

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

Bit Number	Bit Mnemonic	Description
7~0	IAPKEY[7: 0]	<p>Open IAP function and operation time limit setting</p> <p>Write a value n greater than or equal to 0x40, which represents:</p> <p>① Enable the IAP function;</p> <p>② If no IAP write command is received after n system clocks, the IAP function is turned off again.</p>

IAPADL (F2H) IAP Write Address Low Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	IAPADR[7: 0]							

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7~0	IAPADR[7: 0]	IAP writes the low 8 bits of the address

IAPADH (F3H) IAP Write Address High Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	IAPADR[15: 8]							
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[15: 8]	IAP writes the high 8 bits of the address

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

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	IAPADER[7: 0]							

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7~0	IAPADER[7: 0]	<p>IAP extended address:</p> <p>0x00: Both MOVC and IAP are for Flash ROM</p> <p>0x01: The read operation is performed on the Unique ID area, and the erase/write operation is not allowed, otherwise it may cause an exception!</p> <p>Other: reserved</p>

IAPDAT (F5H) IAP Data Register (read/write)

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

Bit number	Bit Mnemonic	Description
7~0	IAPDAT	Data written by IAP

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

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

Bit number	Bit Mnemonic	Description
5	SERASE	<p>Sector Erase(Sector Erase)control bit</p> <p>0: No operation</p> <p>1: Set CMD[1: 0]=10 after setting "1", then enter the Flash ROM sector erase operation, the specified sector of Flash ROM will be erased</p>
4	PRG	<p>Program control bits</p> <p>0: No operation</p> <p>1: Set CMD[1: 0]=10 after setting "1", then enter the Flash ROM write operation, and the data in the IADPDA register will be written to the specified Flash ROM address</p>
2	BTLD	<p>BootLoader control bit</p> <p>0: The program starts to run from the main program area (main program) after Reset;</p> <p>1: The program starts to run from the BootLoader area after Reset</p>
1~0	CMD[1: 0]	<p>IAP Command enable control bit</p> <p>10: Perform write or sector erase operation commands</p> <p>Others: reserved</p> <p>Note:</p> <ol style="list-style-type: none"> After SERASE / PRG set to "1", CMD[1: 0]=10 must be configured, the corresponding operation will start to execute

		<p>2. Only one IAP operation can be performed at a time, so ERASE / SERASE / PRG can only have one bit set 1 at the same time</p> <p>3. Be sure to add at least 8 NOP instructions after the IAP operation statement to ensure that the subsequent instructions can be executed normally after the IAP operation is completed</p>
7,3	-	reserve

5.2.2 IAP Operating Demo Program In C Language

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

```
#include "intrins.h"

unsigned int IAP_Add;

unsigned char IAP_Data;

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

IAP Operation: Sector erase:

```
EA = 0;           //Close global interrupt

IAPADE = 0x00;    //Expand address is 0x00, select Flash ROM

IAPADH = (unsigned char)((IAP_Add >> 8)); // Write the high-bit value of the IAP target address

IAPADL = (unsigned char)IAP_Add;          // Write the low-bit value of the IAP target address

IAPKEY = 0xF0;

IAPCTL = 0x20;    // Set sector erase bit

IAPCTL |= 0x02;   // Block erase

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

_nop_();

_nop_();

_nop_();

_nop_();

_nop_();

_nop_();
```

```
_nop();  
  
_nop();  
  
EA = 1;           // Open global interrupt
```

IAP operation: Write data:

```
EA = 0;           // Close global interrupt  
  
IAPADE = 0X00;    // Extended address is 0x00, choose Flash ROM  
  
IAPDAT = IAP_Data; // Send data to IAP data register  
  
IAPADH = (unsigned char)((IAP_Add >> 8)); // Write the high-bit value of the IAP target address  
  
IAPADL = (unsigned char)IAP_Add;          // Write the low-bit value of the IAP target address  
  
IAPKEY = 0xF0; // This value can be adjusted according to the actual situation; it is  
               // necessary to ensure that after this instruction is executed and before  
               // the IAPCTL is assigned,  
  
               // The time interval needs to be less than 240 (0xF0) system clocks, otherwise  
               // the IAP is disabled;  
  
               // Pay special attention when opening interrupt  
  
IAPCTL = 0X10;    // Set the IAP write operation bit.  
  
IAPCTL |= 0X02;   // Execute write instruction  
  
_nop();           // Wait (at least 8 _nop_() required)  
  
_nop();  
  
_nop();  
  
_nop();  
  
_nop();  
  
_nop();  
  
_nop();  
  
_nop();  
  
EA = 1;           // Open global interrupt
```

IAP operation: read data:

```
EA = 0;           // Close global interrupt  
  
IAPADE = 0X00;    //The extended address is 0x00, selectFlash ROM
```

```
IAP_Data = *( POINT+IAP_Add);    //Read the value of IAP_Add to IAP_Data
```

```
EA = 1;                          // Open global interrupt
```

IAP Operation notes: The user must erase the target sector before writing.

5.3 BootLoader

The LDROM is used to store the bootLoader code. The LDROM range is (02)0000H ~ (02)03FFH, the "02" In brackets is the extended address, which is set by the IAPADE register. LDROM supports blank checking (BLANK), programming (PROGRAM), verifying (VERIFY), erasing (ERASE) and reading (READ) functions in ICP mode. The user must erase the target sector before writing LDROM. One sector is 512 bytes, and the LDROM is divided into 2 sectors:

(02)03FFh	512 bytes
(02)0200h (02)01FFh	512 bytes
(02)0000h	

SC95F852X 1K bytes LDROM Sector

5.3.1 BootLoader Mode operation related registers

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

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

Bit number	Bit Mnemonic	Description
1~0	OP_BL [1: 0]	10: After the chip is reset, it enters the LDROM Other: After the chip is reset, it enters the APROM

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

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

Bit Mnemonic	IAPKEY[7: 0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description
7~0	IAPKEY[7: 0]	<p>Open IAP and operation time limit setting</p> <p>Write a value n greater than or equal to 0x40, which represents:</p> <ul style="list-style-type: none"> ① Enable the IAP; ② If no IAP write command is received after n system clocks, the IAP is turned off again.

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

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

Bit number	Bit Mnemonic	Description
7~0	IAPADR[7: 0]	IAP writes the low 8 bits of the address

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

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	IAPADR[15: 8]							
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[15: 8]	IAP writes the upper 8 bits of the address

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

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

Bit number	Bit Mnemonic	Description
7~0	IAPADER[7: 0]	<p>IAP extended address:</p> <p>0x00: Both MOVc and IAP are for Flash ROM</p> <p>0x01: The read operation is performed on the Unique ID area, and the erase/write operation is not allowed, otherwise it may cause an exception!</p> <p>Other: reserved</p>

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

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

Bit number	Bit Mnemonic	Description
7~0	IAPDAT	Data written by IAP

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

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

Bit number	Bit Mnemonic	Description
6	ERASE	All Erase control bit 0: No operation 1: Set CMD[1: 0]=10 after setting "1", then enter the Flash ROM erase operation, 32K Flash ROM will erase all

5	SERASE	Sector Erase control bit 0: No operation 1: Set CMD[1: 0]=10 after setting "1", then enter the Flash ROM sector erase operation, the specified sector of the Flash ROM will be erased
4	PRG	Program control bit 0: No operation 1: Set CMD[1: 0]=10 after setting "1", then enter the Flash ROM write operation, and the data in the IADPDA register will be written to the specified Flash ROM address
2	BTLD	BootLoader control bit 0: The program starts to run from the APROM after Reset; 1: The program starts to run from the LDROM area after Reset
1~0	CMD[1: 0]	IAP command enable control bit 10: execute write or sector erase operation command Others: reserved Note: <ol style="list-style-type: none"> After ERASE / SERASE / PRG is set to "1", CMD[1: 0]=10 must be configured, and then the corresponding operation will start Only one IAP operation can be performed at a time, so the ERASE / SERASE / PRG can only have one bit set 1 at the same time. Be sure to add at least 8 NOP instructions after the IAP operation statement to ensure that the subsequent instructions can be executed normally after the IAP operation is completed

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

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

Bit Mnemonic	-	-	-	-	RST	-	STOP	IDL
R/W	-	-	-	-	write only	-	write only	write only
POR	x	x	x	x	n	x	0	0

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

Bootloader Notes:

1. The user must erase the target sector before writing LDROM;
2. For the specific operation method, please refer to the description document "SinOne SC95F Series BootLoader Function Implementation Application Guide" provided by SinOne.

5.4 Encryption

Users can choose whether to encrypt the SC95F852X's ROM through the settings on the computer program:

1. If the encryption function is disabled, users can read the last data written in APROM and LDROM by SC LINK;
2. If the encryption function is enabled, the data written in APROM (32 Kbytes Flash ROM) and LDROM will never be read from outside. It is recommended to enable the encryption function during mass production;
3. The only way to release security encryption is to re-programming
4. The encryption has no effect on iap read and write operation
5. For the specific operation method, please refer to the chapter of Secure Encryption and Reading in the "SinOne Development and Mass Production Tool User Manual".

5.5 Unique ID (Unique ID) Area

The SC95F852X provides an independent Unique ID area. A 96-bit unique code can be pre-programmed before leaving the factory to ensure the uniqueness of the chip. The only way for the user to obtain the serial number is to read the relative address (01)0260H~(01)026BH through the IAP instruction. The Unique ID range is (01)0260H ~ (01)026BH, the "01" in brackets indicates the extended address which is set by the IAPADE register. The specific operation method is as follows:

IAPADE (F4H) IAP Write to extended address register (Read/Write)

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

Bit number	Bit Mnemonic	Description
7~0	IAPADER[7: 0]	IAP Extended address: 0x00: Both MOVC and IAP are for Flash ROM 0x01: For the unique ID area, read and write operations are not allowed, otherwise it may cause an exception! Other: reserved

5.5.1 Unique ID Read Operating Demo Program In C Language

```
#include "intrins.h"

unsigned char UniqueID [12]; //store UniqueID

unsigned char code * POINT =0x0260;

unsigned char i;

EA = 0;           // Disable the global interrupt

IAPADE = 0X01;    // Expand address 0x01, select Unique ID area
```

```

for(i=0;i<12;i++)
{
    UniqueID [i]= *( POINT+i);    // Read the value of UniqueID
}

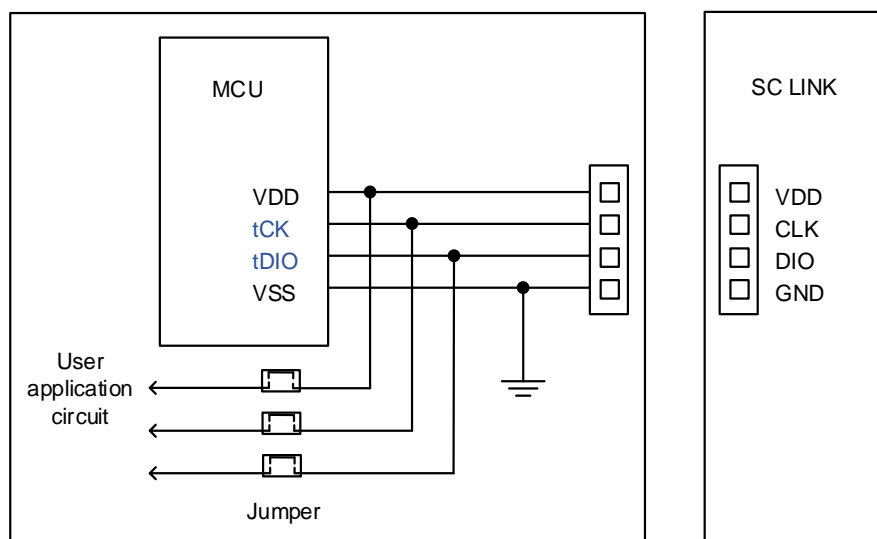
IAPADE = 0X00;                    // Expand address 0x00, return to Code area

EA = 1;                          // Enable global interrupt

```

5.6 Programming

The SC95F852X's Flash ROM and 1K bytes LDROM can be programmed through tDIO, tCK, VDD, VSS, the specific connection relationship is as follows:



ICP mode Flash Writer programming connection diagram

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

JTAG Specific Mode:

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

Normal Mode (JTAG specific port is invalid):

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

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

JTAG mode configuration register:

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

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

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

5.7 Code Option Area (User Programming Settings)

There is a separate Flash area inside the SC95F852X to save the customer's initial settings. This area is called Customer Option area. The user writes this part of the code inside the IC when programming the IC. When the IC is reset and initialized, it will transfer this setting to SFR as the initial setting.

Option related SFR operation instructions:

The read and write operations of Option-related SFRs are controlled by OPINX and OPREG registers. The specific location of each Option SFR is determined by OPINX, as shown in the following table:

Symbol	Address	Instructions	7	6	5	4	3	2	1	0
OP_HRCR	83H@FFH	System clock change register	OP_HRCR[7: 0]							
OP_CTM0	C1H@FFH	Customer Option register 0	ENWDT	ENXTL	SCLKS[1: 0]		DISRST	DISLVR	LVRS[1: 0]	
OP_CTM1	C2H@FFH	Customer Option register 1	VREFS[1: 0]		-	DISJTG	IAPS[1: 0]		OP_BL	

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

		<p>User can change the high-frequency oscillator frequency f_{HRC} by modifying the value of this register, and then change the system clock frequency f_{sys}:</p> <ol style="list-style-type: none">1. The initial value of OP_HRCR[7: 0] after power-on OP_HRCR[s] is a fixed value to ensure that f_{HRC} is 32MHz, OP_HRCR[s] of each IC may be different2. When the initial value is OP_HRCR[s], the system clock frequency f_{sys} of the IC can be set to an accurate 32/16/8/4MHz through the Option item. When OP_HRCR [7: 0] changes by 1, the f_{sys} frequency changes by about 0.18% <p>The relationship between OP_HRCR [7: 0] and f_{sys} output frequency is as follows:</p> <table><tr><td>OP_HRCR [7: 0] Value</td><td>f_{sys} actual output frequency (32M as an example)</td></tr><tr><td>OP_HRCR [s]-n</td><td>$32000 \times (1 - 0.18\% \times n)$ kHz</td></tr><tr><td>...</td><td>....</td></tr><tr><td>OP_HRCR [s]-2</td><td>$32000 \times (1 - 0.18\% \times 2) = 31\,884.8$ kHz</td></tr><tr><td>OP_HRCR [s]-1</td><td>$32000 \times (1 - 0.18\% \times 1) = 31\,942.4$ kHz</td></tr><tr><td>OP_HRCR [s]</td><td>32000 kHz</td></tr><tr><td>OP_HRCR [s]+1</td><td>$32000 \times (1 + 0.18\% \times 1) = 32\,057.6$ kHz</td></tr><tr><td>OP_HRCR [s]+2</td><td>$32000 \times (1 + 0.18\% \times 2) = 32\,115.2$ kHz</td></tr><tr><td>...</td><td>...</td></tr><tr><td>OP_HRCR [s]+n</td><td>$32000 \times (1 + 0.18\% \times n)$ kHz</td></tr></table> <p>Note:</p> <ol style="list-style-type: none">1. The value of OP_HRCR[7: 0] after each power-on of the IC is the value of the high-frequency oscillator frequency f_{HRC} closest to 32MHz; the user corrects the value of HRC after each power-on to	OP_HRCR [7: 0] Value	f_{sys} actual output frequency (32M as an example)	OP_HRCR [s]-n	$32000 \times (1 - 0.18\% \times n)$ kHz	OP_HRCR [s]-2	$32000 \times (1 - 0.18\% \times 2) = 31\,884.8$ kHz	OP_HRCR [s]-1	$32000 \times (1 - 0.18\% \times 1) = 31\,942.4$ kHz	OP_HRCR [s]	32000 kHz	OP_HRCR [s]+1	$32000 \times (1 + 0.18\% \times 1) = 32\,057.6$ kHz	OP_HRCR [s]+2	$32000 \times (1 + 0.18\% \times 2) = 32\,115.2$ kHz	OP_HRCR [s]+n	$32000 \times (1 + 0.18\% \times n)$ kHz
OP_HRCR [7: 0] Value	f_{sys} actual output frequency (32M as an example)																					
OP_HRCR [s]-n	$32000 \times (1 - 0.18\% \times n)$ kHz																					
...																					
OP_HRCR [s]-2	$32000 \times (1 - 0.18\% \times 2) = 31\,884.8$ kHz																					
OP_HRCR [s]-1	$32000 \times (1 - 0.18\% \times 1) = 31\,942.4$ kHz																					
OP_HRCR [s]	32000 kHz																					
OP_HRCR [s]+1	$32000 \times (1 + 0.18\% \times 1) = 32\,057.6$ kHz																					
OP_HRCR [s]+2	$32000 \times (1 + 0.18\% \times 2) = 32\,115.2$ kHz																					
...	...																					
OP_HRCR [s]+n	$32000 \times (1 + 0.18\% \times n)$ kHz																					

		<p>allow the system clock frequency fsys of the IC to work at the user Required frequency</p> <p>2. In order to ensure the reliable operation of the IC, the maximum operating frequency of the IC should not exceed 10% of 32MHz, that is 35.2MHz;</p> <p>3. Please confirm that the change of HRC frequency will not affect other functions.</p>
--	--	--

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

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

Bit number	Bit Mnemonic	Description
7	ENWDT	<p>Watchdog (WDT) control bit (This bit is transferred by the system to the value set by the user Code Option)</p> <p>0: WDT invalid</p> <p>1: WDT valid (WDT stops counting during IAP execution)</p>
6	ENXTL	<p>External High-Frequency crystal oscillator selection bit</p> <p>0: External 32.768kHz crystal Interface disable, P5.0 and P5.1 valid</p> <p>1: External 32.768kHz crystal Interface enable, P5.0 and P5.1 invalid</p>
5~4	SCLKS[1: 0]	<p>System clock frequency selection bits</p> <p>00: System clock frequency is HRC frequency divided by 1;</p> <p>01: System clock frequency is HRC frequency divided by 2;</p>

		10: System clock frequency is HRC frequency divided by 4; 11: System clock frequency is HRC frequency divided by 8;
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 control 11: 4.3V reset 10: 3.7V reset 01: 2.9V reset 00: 1.9V reset

OP_CTM1 (C2H@FFH) Code Option Register 1(read/write)

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

Bit number	Bit Mnemonic	Description
------------	--------------	-------------

7~6	VREFS[1: 0]	Reference voltage selection (initial value are configured by the user and loaded from Code Options) 00: Configure ADC VREF as VDD; 01: Configure ADC VREF as internal 2.048V; 10: Configure ADC VREF as internal 1.024V; 11: reserved
4	DISJTG	IO/JTAG Port switching control 0: JTAG mode is enabled, P1.2 and P1.3 can only be used as tCK/tDIO . 1: Normal mode (Normal), JTAG function is invalid.
3~2	IAPS[1: 0]	IAP spatial range selection 00: The last 0K Flash ROM allows IAP operation 01: Last 1K Flash ROM allows IAP operation 10: Last 2K Flash ROM allows IAP operation 11: All Flash ROM allows IAP operation
1~0	OP_BL [1:0]	10: Enter LDROM after chip reset

5.7.1 Option-related Registers Operation Instructions

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

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

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

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

C language example:

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

Assembly language example:

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

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

5.8 SRAM

The SRAM of the SC95F852X MCU is divided into internal 256 bytes RAM, external 4096 bytes RAM and 16 bytes PWM RAM. The address range of the internal RAM is 00H~FFH, where the high 128 bytes (address 80H~FFH) can only be indirectly addressed, and the low 128 bytes (address 00H~7FH) can be directly or indirectly addressed.

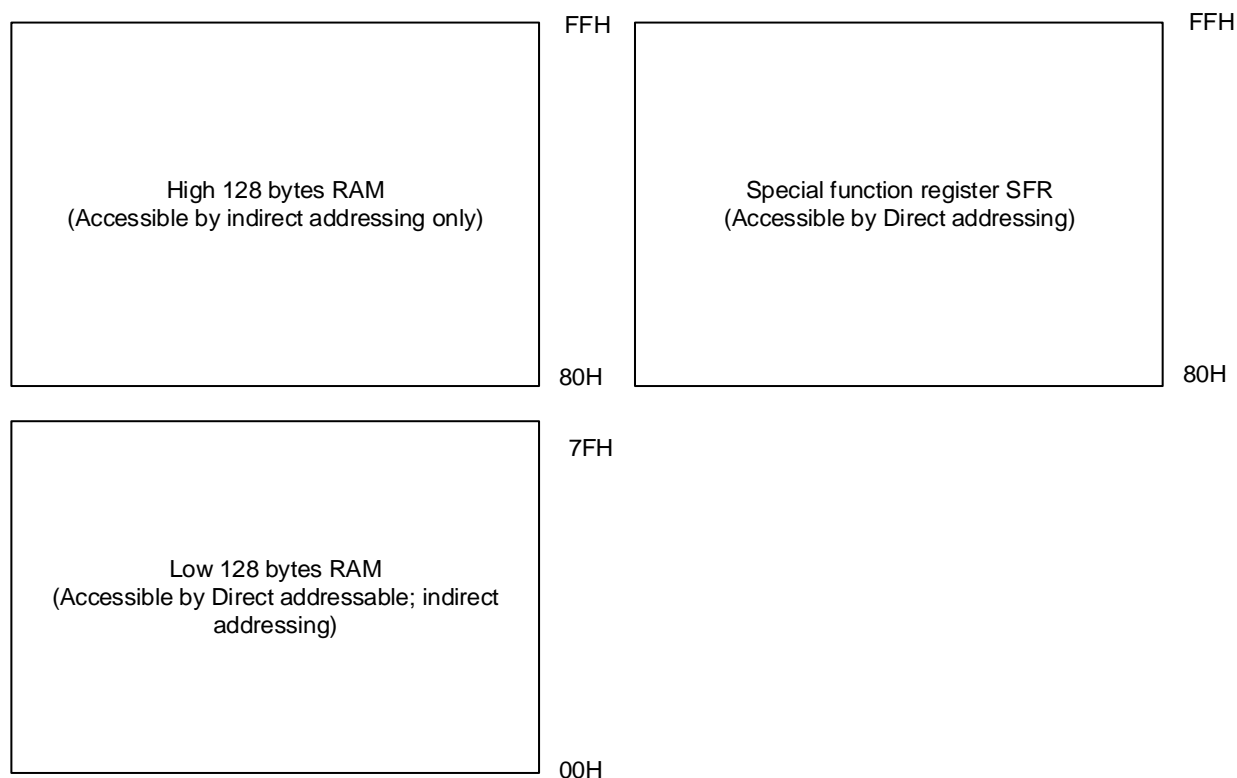
The address of the special function register SFR is also 80H~FFH. But the difference between SFR and internal high 128 bytes SRAM is: SFR register is directly addressed, while internal high 128 bytes SRAM can only be indirectly addressed.

The address of the external RAM is 0000H~0FFFFH, but it needs to be addressed by the MOVX instruction.

5.8.1 Internal 256 Bytes SRAM

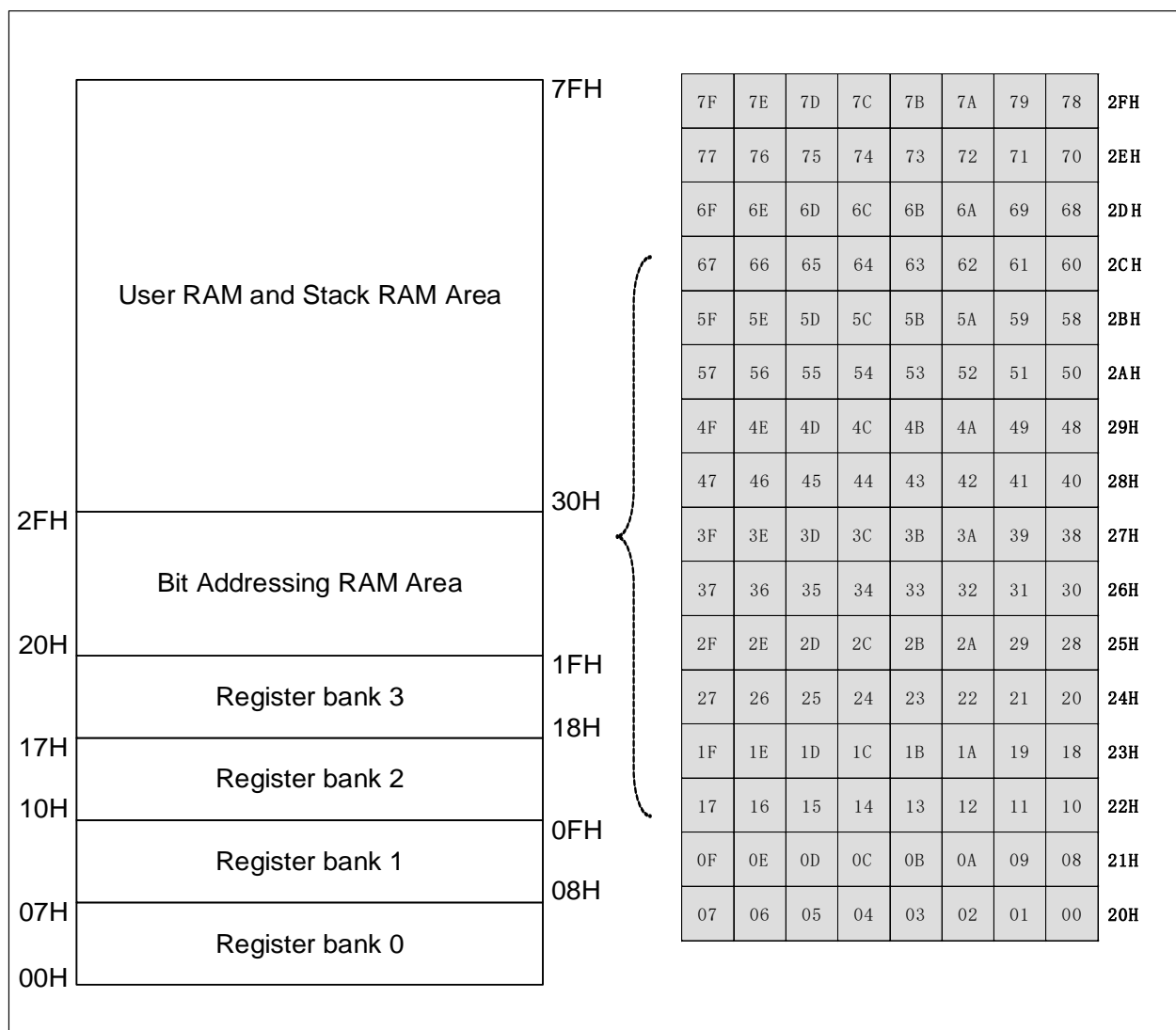
The internal low 128 bytes SRAM area can be divided into three parts:

- ① Operating register group 0~3, address 00H~1FH, the combination of RS0 and RS1 in the program status word register PSW determines the operating register currently used, using operating register group 0~ 3 can speed up the operation;
- ② bit addressing area 20H~2FH, this area can be used as ordinary RAM or bit-wise addressing RAM; when addressing by bit, the bit address is 00H~7FH, (this The address is programmed bit by bit, which is different from the general SRAM coded by byte), which can be distinguished by instructions in the program;
- ③ User RAM and stack area, after the SC95F852X is reset, the 8-bit stack pointer points to the stack area, the user needs to set the initial value.



Internal 256 bytes RAM structure diagram

The internal low 128 bytes RAM structure is as follows:



SRAM structure diagram

5.8.2 External 4096 Bytes SRAM

External 4096 bytes RAM can be accessed through MOVX @DPTR, A; you can also use MOVX A, @Ri or MOVX @Ri, A with EXADH register to access external 4096 bytes RAM: EXADH register stores the high address of external SRAM, Ri register stores the low 8 bits of the external SRAM.

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

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

POR	x	x	x	x	0	0	0	0
-----	---	---	---	---	---	---	---	---

Bit number	Bit Mnemonic	Description
3~0	EXADH [3: 0]	High-bit of external SRAM operation address
7~4	-	reserved

5.8.3 PWM 16 Bytes SRAM

The 1040H~104FH of the RAM address are used as PWM SRAM of 16 bytes. The PWM duty cycle adjustment register occupies 1040H~104FH and can **be read and written**. For specific operation methods, refer to [13.3.2 PWM Independent Mode Duty Cycle Configuration](#) or [13.4.2 PWM Complementary Mode Duty Cycle Configuration](#);

6 Special Function Register (SFR)

6.1 SFR Mapping

The SC95F852X provides some registers equipped with special functions, called SFR. The addresses of these registers are located at 80H~FFH, some are bit-addressable, and others are not. It is very convenient for these bit addressable registers to change the value of single bit, of which the address is end up with figure "0" or "8". All SFR shall use direct addressing for addressing.

The SC95F852X SFR Map is as follows:

	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
F8h	-	-	-	BTMCON	CRCINX	CRCREG	OPINX	OPREG
F0h	B	IAPKEY	IAPADL	IAPADH	IAPADE	IAPDAT	IAPCTL	EXADH
E8h	-	EXA0	EXA1	EXA2	EXA3	EXBL	EXBH	OPERCON
E0h	ACC	-	-	-	-	-	-	-
D8h	P5	P5CON	P5PH	-	-	-	-	-
D0h	PSW	PWMCFG	PWMCON0	PWMCON1	PWMPDL	PWMPDH	PWMDFR	PWMFLT
C8h	TXCON	TXMOD	RCAPXL	RCAPXH	TLX	THX	TXINX	WDTCON
C0h	-	-	-	-	US2CON0	US2CON1	US2CON2	US2CON3
B8h	IP	IP1	IP2	INT0R	INT1F	INT1R	INT2F	INT2R
B0h	-	-	-	-	INT0F	ADCCFG2	US0CON4	US0CON5
A8h	IE	IE1	IE2	ADCCFG0	ADCCFG1	ADCCON	ADCVL	ADCVH
A0h	P2	P2CON	P2PH	-	US1CON0	US1CON1	US1CON2	US1CON3
98h	SCON	SBUF	P0CON	P0PH	-	US0CON1	US0CON2	US0CON3

90h	P1	P1CON	P1PH	-	-	US0CON0	IOHCON0	IOHCON1
88h	TCON	TMOD	TL0	TL1	TH0	TH1	TMCON	OTCON
80h	P0	SP	DPL	DPH	DPL1	DPH1	DPS	PCON
	Bit addressable	Non-bit addressable						

Note:

1. The empty part of the SFR register are not recommended for users.
2. The FEH to FFH in SFR are special function registers used in system configuration. Using these registers may cause system exceptions. During system initialization, users cannot clear these registers or perform other operations.

6.2 Register Summary

Mnemonic	Add	Description	7	6	5	4	3	2	1	0	POR
P0	80H	P0 port data register	P07	P06	P05	P04	P03	P02	P01	P00	0000000b
SP	81H	Stack pointer	SP[7: 0]								00000111b
DPL	82H	DPTR data pointer low	DPL[7: 0]								00000000b
DPH	83H	DPTR data pointer high	DPH[7: 0]								00000000b
DPL1	84H	DPTR1 data pointer low	DPL1[7: 0]								00000000b
DPH1	85H	DPTR1 data pointer high	DPH1[7: 0]								00000000b
DPS	86H	DPTR selection register	ID1	ID0	TSL	AU1	AU0	-	-	SEL	00000xx0b
PCON	87H	Power management control register	-	-	-	-	RST	-	STOP	IDL	xxxx0x00b
TCON	88H	Timer control register	TF1	TR1	TF0	TR0	IE1	-	IE0	-	00000x0xb

TMOD	89H	Timer operating mode register	-	C/T1	M11	M01	-	C/T0	M10	M00	x000x000b
TL0	8AH	Low 8 bits of timer 0	TL0[7: 0]								00000000b
TL1	8BH	Low 8 bits of timer 1	TL1[7: 0]								00000000b
TH0	8CH	Timer 0 high 8 bits	TH0[7: 0]								00000000b
TH1	8DH	Timer 1 high 8 bits	TH1[7: 0]								00000000b
TMCON	8EH	Timer frequency control register	USMD2[1: 0]		-	-	-	-	T1FD	T0FD	00xxx00b
OTCON	8FH	Output control register	USMD1[1:0]		USMD0[1:0]		-	-	-	-	0000xxxxb
P1	90H	P1 port data register	P17	P16	P15	P14	P13	P12	P11	P10	00000000b
P1CON	91H	P1 port input/output control register	P1C7	P1C6	P1C5	P1C4	P1C3	P1C2	P1C1	P1C0	00000000b
P1PH	92H	P1 port pull-up resistor control register	P1H7	P1H6	P1H5	P1H4	P1H3	P1H2	P1H1	P1H0	00000000b
US0CON0	95H	USC10 control register 0	US0CON0[7: 0]								00000000b
IOHCON0	96H	IOH setting register 0	P1H[1: 0]		P1L[1: 0]		P0H[1: 0]		P0L[1: 0]		00000000b
IOHCON1	97H	IOH setting register 1	-	-	P5L[1: 0]		P2H[1: 0]		P2L[1: 0]		xx000000b
SCON	98H	Serial control register	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	00000000b
SBUF	99H	Serial data buffer register	SBUF[7: 0]								00000000b
P0CON	9AH	P0 port input/output control register	P0C7	P0C6	P0C5	P0C4	P0C3	P0C2	P0C1	P0C0	00000000b
P0PH	9BH	P0 port pull-up resistor control register	P0H7	P0H6	P0H5	P0H4	P0H3	P0H2	P0H1	P0H0	00000000b
ETUCK0	9DH	ETU rate frequency division register 0 @USMD0= 00	ETUCK[7:0]								01110100b
US0CON1	9DH	USC10 control register 1 @USMD0= 01/10/11	US0CON1[7: 0]								00000000b

ETUCK1	9EH	ETU rate frequency division register 1 @USMD0= 00	SCCK[4:0]				ETUCK [11:8]				0000001b
US0CON2	9EH	USCI0 control register 2 @USMD0= 01/10/11	US0CON2[7: 0]								0000000b
US0CON3	9FH	USCI0 control register 3	US0CON3[7: 0]								0000000b
P2	A0H	P2 port data register	P27	P26	P25	P24	P23	P22	P21	P20	0000000b
P2CON	A1H	P2 port input/output control register	P2C7	P2C6	P2C5	P2C4	P2C3	P2C2	P2C1	P2C0	0000000b
P2PH	A2H	P2 port pull-up resistor control register	P2H7	P2H6	P2H5	P2H4	P2H3	P2H2	P2H1	P2H0	0000000b
US1CON0	A4H	USCI1 control register 0	US1CON0[7: 0]								0000000b
US1CON1	A5H	USCI1 control register 1	US1CON1[7: 0]								0000000b
US1CON2	A6H	USCI1 control register 2	US1CON2[7: 0]								0000000b
US1CON3	A7H	USCI1 control register 3	US1CON3[7: 0]								0000000b
IE	A8H	Interrupt enable register	EA	EADC	ET2	EUART	ET1	EINT1	ET0	EINT0	0000000b
IE1	A9H	Interrupt enable register 1	ET4	ET3	-	ETK	EINT2	EBTM	EPWM	ESSI0	00x00000b
IE2	AAH	Interrupt enable register 2	-	-	-	-	-	-	ESSI2	ESSI1	xxxxxx00b
ADCCFG0	ABH	ADC setting register 0	EAIN7	EAIN6	EAIN5	EAIN4	EAIN3	EAIN2	EAIN1	EAIN0	0000000b
ADCCFG1	ACH	ADC setting register 1	-	-	-	-	-	-	EAIN9	EAIN8	xxxxxx00b
ADCCON	ADH	ADC control register	ADCE N	ADCS	EOC/ ADCIF	ADCIS[4: 0]				0000000b	
ADCVL	AEH	ADC result register	ADCV[3: 0]				-	-	-	-	1111xxxxb
ADCVH	AFH	ADC result register	ADCV[11: 4]								11111111b
INT0F	B4H	INT0 falling edge interrupt control register	-	-	-	-	INT0F3	INT0F2	INT0F1	-	xxxx000xb

ADCCFG2	B5H	ADC setting register 2	-	-	-	LOWSP[2: 0]			-	-	xxx000xb
US0CON4	B6H	EGT Extended protection time register	EGT [7:0]								0000000b
US0CON5	B7H	SCSTA0 SC status register @SES=0	SES	TC	RC	WTRT	TBUSY	RBUSY	WTER	FER	0000000b
US0CON5	B7H	SCSTA1 SC R/W error information register @SES=1	SES	-	-	-	TPER	RPER	-	ROVF	0xxx00x0b
IP	B8H	Interrupt priority control register	-	IPADC	IPT2	IPUART	IPT1	IPINT1	IPT0	IPINT0	x000000b
IP1	B9H	Interrupt priority control register 1	IPT4	IPT3	-	IPTK	IPINT2	IPBTM	IPPWM	IPSSI0	00x0000b
IP2	BAH	Interrupt priority control register 2	-	-	-	-	-	-	IPSSI2	IPSSI1	xxxxxx00b
INT0R	BBH	INT0 rising edge interrupt control register	-	-	-	-	INT0R3	INT0R2	INT0R1	-	xxxx000xb
INT1F	BCH	INT1 falling edge interrupt control register	-	-	-	-	INT1F3	INT1F2	INT1F1	INT1F0	xxxx0000b
INT1R	BDH	INT1 rising edge interrupt control register	-	-	-	-	INT1R3	INT1R2	INT1R1	INT1R0	xxxx0000b
INT2F	BEH	INT2 falling edge interrupt control register	-	-	INT2F5	INT2F4	INT2F3	INT2F2	INT2F1	INT2F0	xx000000b
INT2R	BFH	INT2 rising edge interrupt control register	-	-	INT2R5	INT2R4	INT2R3	INT2R2	INT2R1	INT2R0	xx000000b
US2CON0	C4H	USCI2 control register 0	US2CON0[7: 0]								0000000b
US2CON1	C5H	USCI2 control register 1	US2CON1[7: 0]								0000000b
US2CON2	C6H	USCI2 control register 2	US2CON2[7: 0]								0000000b
US2CON3	C7H	USCI2 control register 3	US2CON3[7: 0]								0000000b
TXCON	C8H	Timer 2/3/4 control register	TFX	EXFX	RCLKX	TCLKX	EXENX	TRX	C/TX	CP/RLX	0000000b
TXMOD	C9H	Timer 2/3/4 operating mode register	TXFD	-	-	-	-	-	TXOE	DCXEN	0xxxxx00b

RCAPXL	CAH	Timer 2/3/4 reload low 8 bits	RCAPXL[7: 0]								00000000b
RCAPXH	CBH	Timer 2/3/4 reload high 8 bits	RCAPXH[7: 0]								00000000b
TLX	CCH	Timer 2/3/4 low 8 bits	TLX[7: 0]								00000000b
THX	CDH	Timer 2/3/4 high 8 bits	THX[7: 0]								00000000b
TXINX	CEH	Timer control register pointer	-	-	-	-	-	TXINX[2: 0]			xxxxx010b
WDTCON	CFH	WDT control register	-	-	-	CLRWDT	-	WDTCKS[2: 0]			xxx0x000b
PSW	D0H	Program status word register	CY	AC	F0	RS1	RS0	OV	F1	P	00000000b
PWMCFG	D1H	PWM setting register	INV7	INV6	INV5	INV4	INV3	INV2	INV1	INV0	x0000000b
PWMCON0	D2H	PWM control register 0	ENPWM	PWMIF	PWMCK[1:0]		-	-	PWMMD[1:0]		00000000b
PWMCON1	D3H	PWM control register 1	ENPWM7	ENPWM6	ENPWM5	ENPWM4	ENPWM3	ENPWM2	ENPWM1	ENPWM0	x0000000b
PWMPDL	D4H	PWM cycle register low 8 bits	PWMPDL[7:0]								00000000b
PWMPDH	D5H	PWM cycle register high 8 bits	PWMPDH[7:0]								00000000b
PWMDFR	D6H	PWM dead time setting register	PDF1[3:0]				PDR1[3:0]				00000000b
PWMFLT	D7H	PWM fault detection setting register	FLTEN1	FLTSTA1	FLTMDD1	FLTLV1	-	-	FLTDT1[1:0]		0000xx00b
P5	D8H	P5 port data register	-	-	-	-	-	-	P51	P50	xxxxxx00b
P5CON	D9H	P5 port input/output control register	-	-	-	-	-	-	P5C1	P5C0	xxxxxx00b
P5PH	DAH	P5 port pull-up resistor control register	-	-	-	-	-	-	P5H1	P5H0	xxxxxx00b
ACC	E0H	accumulator	ACC[7: 0]								00000000b
EXA0	E9H	Extended Accumulator 0	EXA[7: 0]								00000000b

EXA1	EAH	Extended Accumulator 1	EXA[15: 8]								00000000b
EXA2	EBH	Extended Accumulator 2	EXA[23: 16]								00000000b
EXA3	ECH	Extended Accumulator 3	EXA[31: 24]								00000000b
EXBL	EDH	Extended B register L	EXB [7: 0]								00000000b
EXBH	EEH	Extended B register H	EXB [15: 8]								00000000b
OPERCON	EFH	Arithmetic control register	OPER S	MD	-	-	-	-	CRCRST	CRCSTA	00xxx00b
B	F0H	B register	B[7: 0]								00000000b
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[15: 8]								00000000b
IAPADE	F4H	IAP write to extended address register	IAPADER[7: 0]								00000000b
IAPDAT	F5H	IAP data register	IAPDAT[7: 0]								00000000b
IAPCTL	F6H	IAP control register	-	ERASE	SERASE	PRG	-	BTLD	CMD[1: 0]		x000x000b
EXADH	F7H	High-bit address of external SRAM operation address	-	-	-	-	EXADH [3: 0]			xxxx0000b	
BTMCON	FBH	Low frequency timer control register	ENBT M	BTMIF	-	-	BTMFS[3: 0]			00xx0000b	
CRCINX	FCH	CRC pointer	CRCINX[7: 0]								00000000b
CRCREG	FDH	CRC register	CRCREG[7: 0]								nnnnnnnnb
OPINX	FEH	Customer Option pointer	OPINX[7: 0]								00000000b
OPREG	FFH	Customer Option register	OPREG[7: 0]								nnnnnnnnb

PWM duty cycle adjustment register (R/W)

Add	7	6	5	4	3	2	1	0	POR
1040H	PDT0[15:8]								00000000b
1041H	PDT0[7:0]								00000000b
1042H	PDT1[15:8]								00000000b
1043H	PDT1[7:0]								00000000b
1044H	PDT2[15:8]								00000000b
1045H	PDT2[7:0]								00000000b
1046H	PDT3[15:8]								00000000b
1047H	PDT3[7:0]								00000000b
1048H	PDT4[15:8]								00000000b
1049H	PDT4[7:0]								00000000b
104AH	PDT5[15:8]								00000000b
104BH	PDT5[7:0]								00000000b
104CH	PDT6[15:8]								00000000b
104DH	PDT6[7:0]								00000000b
104EH	PDT7[15:8]								00000000b
104FH	PDT7[7:0]								00000000b

7 Power, Reset And System Clock

7.1 Power Circuit

The SC95F852X power supply system includes BG, LDO, POR, LVR and other circuits, which can achieve reliable operation in the range of 2.0~5.5V. In addition, the IC has a built-in, accurate 2.048/1.024V voltage that can be used as an internal reference voltage for the ADC. Users can find the specific settings in the [18Analog-to-Digital Converter\(ADC\)](#).

7.2 Power-on Reset

After the SC95F852X 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 SC95F852X will always be in the reset mode, There will not be a valid clock until the voltage supplied to the SC95F852X 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

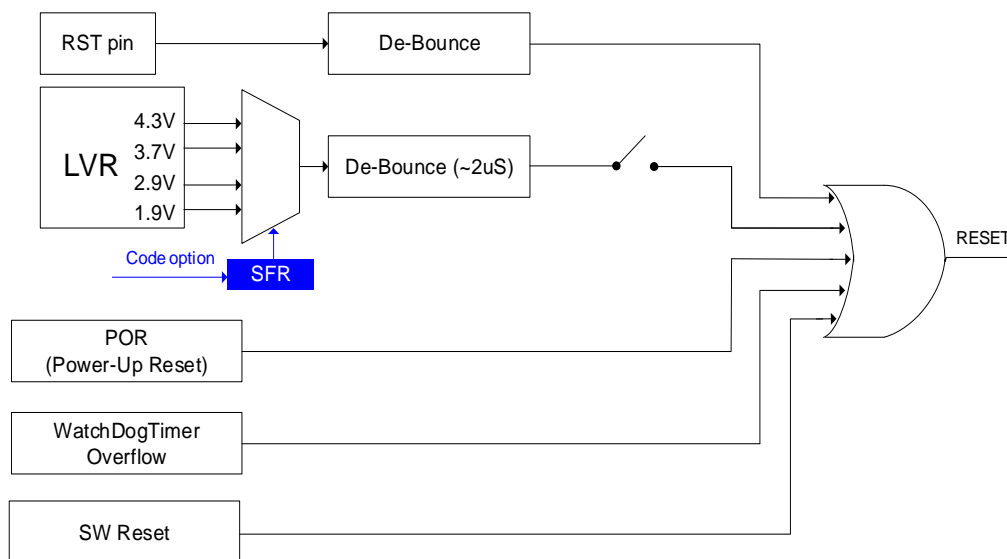
There is a warm-up counter inside The SC95F852X. During the reset stage, the warm-up counter is cleared to 0 until the voltage exceeds the POR voltage, the internal RC oscillator starts to oscillate, and the warm-up counter starts counting. When the internal warm-up counter counts to a certain number, every certain number of HRC clocks will read a byte of data from the IFB (including Customer Option) in the Flash ROM and store it in the internal system register. This reset signal will not end until the warm-up is completed.

7.2.3 Normal Operation Stage

After finishing the Loading Information stage, The SC95F852X starts to read the instruction code from Flash and enters the normal operation stage. The LVR voltage is the set value of Customer Option written by user.

7.3 Reset Modes

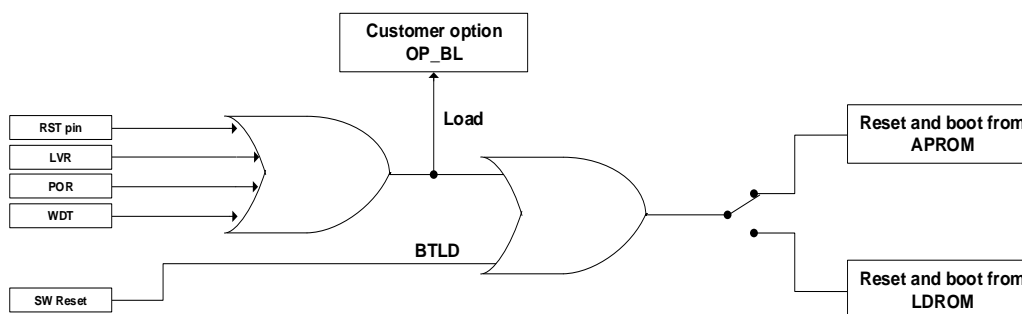
The SC95F852X has 5 reset methods: ① External reset ② Low-voltage reset (LVR) ③ Power-on reset (POR) ④ Watchdog (WDT) reset ⑤ Software reset. The circuit diagram of the reset part of the SC95F852X is as follows:



SC95F852X Reset circuit diagram

Reset after the launch of the area:

1. the external Reset, Low-voltage Reset, Power-on Reset (POR), Watchdog (WDT) Reset, these four modes following a hardware reset chip from the start of the user OP_BL set area (APROM/LDROM) started.
2. Software reset after the chip according to BTLD (IAPCTL. 2) set the start area (APROM/LDROM) start.



SC95F852X Reset start zone switchover diagram

7.3.1 External Reset

External reset is a reset pulse signal of a certain width given to SC95F852X from external RST pin to realize the reset of SC95F852X. The user can configure the P1.1/RST pin as RST (reset pin) by Customer Option.

7.3.2 Low-voltage Reset LVR

The SC95F852X provides a low-voltage reset circuit. There are 4-level LVR voltage options: 4.3V, 3.7V, 2.9V, 1.9V. The default value is the Option value written by the user.

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

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

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

7.3.3 Power-on Reset (POR)

The SC95F852X has a power-on reset circuit inside. When the power supply voltage VDD reaches the POR reset voltage, the system automatically resets.

7.3.4 Watchdog Reset (WDT)

The SC95F852X has a WDT, the clock source of which is the internal 32.768kHz LRC. The user can choose whether to enable the watchdog reset function by Customer Option.

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

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

Bit number	Bit Mnemonic	Description
7	ENWDT	WDT control bit (This bit is transferred by the system to the value set by the user Code Option) 1: WDT valid 0: WDT invalid

WDTCON (CFH) WDT Control Register (read/write)

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

Bit number	Bit Mnemonic	Description																		
4	CLRWDT	Clear WDT (Only valid when set to 1) 1: WDT counter restart, cleared by system hardware																		
2~0	WDTCKS [2: 0]	Watchdog clock selection <table><tr><th>WDTCKS[2: 0]</th><th>WDT overflow time</th></tr><tr><td>000</td><td>500ms</td></tr><tr><td>001</td><td>250ms</td></tr><tr><td>010</td><td>125ms</td></tr><tr><td>011</td><td>62.5ms</td></tr><tr><td>100</td><td>31.5ms</td></tr><tr><td>101</td><td>15.75ms</td></tr><tr><td>110</td><td>7.88ms</td></tr><tr><td>111</td><td>3.94ms</td></tr></table>	WDTCKS[2: 0]	WDT overflow time	000	500ms	001	250ms	010	125ms	011	62.5ms	100	31.5ms	101	15.75ms	110	7.88ms	111	3.94ms
WDTCKS[2: 0]	WDT overflow time																			
000	500ms																			
001	250ms																			
010	125ms																			
011	62.5ms																			
100	31.5ms																			
101	15.75ms																			
110	7.88ms																			
111	3.94ms																			
7~5,3	-	Reserved																		

7.3.5 Software Reset

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

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	-	-	-	-	RST	-	STOP	IDL

R/W	-	-	-	-	Write only	-	Write only	Write only
POR	x	x	x	x	n	x	0	0

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

7.3.6 Register Reset Value

When The SC95F852X is in reset state, most registers will return to their initial state. The watchdog (WDT) is turned off. The initial value of the program counter PC is 0000h, and the initial value of the stack pointer SP is 07h. The "hot restart" Reset (such as WDT, LVR, software reset, etc.) will not affect the SRAM, and the SRAM value is always the value before the reset. The loss of SRAM content will occur when the power supply voltage is so low that the RAM cannot be saved.

The initial values of the power-on reset of the SFR register [6.2 Register Summary](#).

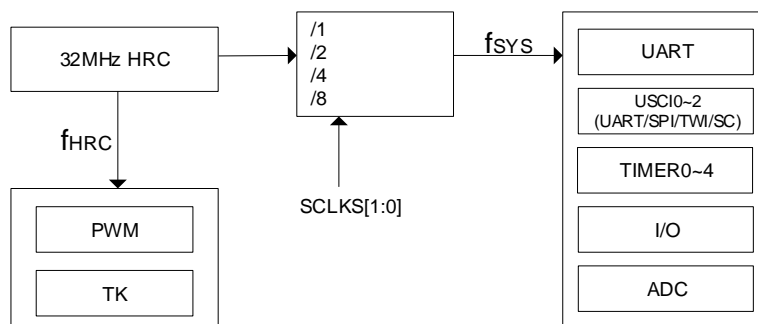
7.4 High-speed RC Oscillator

The SC95F852X has a built-in high-precision high-precision oscillator (HRC) with adjustable oscillation frequency. The HRC is accurately adjusted to 32MHz@5V/25°C at the factory. Users can set the system clock to 32/16/8/4MHz through the Customer Option when programming. This HRC will drift to a certain extent within the full voltage range (2.0V~5.5V) due to the operating ambient temperature:

- -20 ~ 85°C application environment, frequency error does not exceed $\pm 1\%$
- -40 ~ 105°C application environment, frequency error does not exceed $\pm 2\%$

The HRC can be automatically calibrated by connecting an external 32.768kHz crystal oscillator. Users only need to connect an external 32.768kHz crystal oscillator, Users can set the external 32.768kHz crystal oscillator function through the Customer Option when programming. The HRC self-calibration function is automatically turned on after the IC is powered on.

Note: The clock source of the PWM and TK circuits is fixed at $f_{HRC} = 32\text{MHz}$.



SC95F852X Internal clock relationship

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

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

Bit number	Bit Mnemonic	Description
5~4	SCLKS[1: 0]	System clock frequency selection bits 00: System clock frequency is HRC frequency divided by 1; 01: System clock frequency is HRC frequency divided by 2; 10: System clock frequency is HRC frequency divided by 4; 11: System clock frequency is HRC frequency divided by 8;

The SC95F852X has a special function: the user can modify the value of SFR to adjust the HRC frequency within a certain range. The user can achieve this by configuring the OP_HRCR register. Note: HRC can be automatically

calibrated by connecting a 32.768kHz crystal oscillator. Therefore, if the user uses the 32.768kHz external crystal oscillator function, the HRC frequency will always be corrected to 32MHz. At this time, adjusting OP_HRCR cannot change the HRC frequency.

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]	<p>HRC frequency change register</p> <p>The user can change the high-frequency oscillator frequency f_{HRC} by modifying the value of this register, and then change the system clock frequency f_{sys} of the IC:</p> <ol style="list-style-type: none">1. The initial value of OP_HRCR[7: 0] after power-on OP_HRCR[s] is a fixed value to ensure that f_{HRC} is 32MHz, OP_HRCR[s] of each IC may be different2. When the initial value is OP_HRCR[s], the system clock frequency f_{sys} of the IC can be set to an accurate 32/16/8/4MHz through the Option item. When OP_HRCR [7: 0] changes by 1, the f_{sys} frequency changes by about 0.18% <p>The relationship between OP_HRCR [7: 0] and f_{sys} output frequency is as follows:</p> <table><tr><td>OP_HRCR [7: 0] value</td><td>f_{sys} actual output frequency (32M as an example)</td></tr><tr><td>OP_HRCR [s]-n</td><td>$32000 \times (1 - 0.18\% \times n) \text{kHz}$</td></tr></table>	OP_HRCR [7: 0] value	f_{sys} actual output frequency (32M as an example)	OP_HRCR [s]-n	$32000 \times (1 - 0.18\% \times n) \text{kHz}$
OP_HRCR [7: 0] value	f_{sys} actual output frequency (32M as an example)					
OP_HRCR [s]-n	$32000 \times (1 - 0.18\% \times n) \text{kHz}$					

	
		OP_HRCR [s]-2	$32000*(1-0.18\%*2) = 31\,884.8\text{kHz}$
		OP_HRCR [s]-1	$32000*(1-0.18\%*1) = 31\,942.4\text{kHz}$
		OP_HRCR [s]	32000kHz
		OP_HRCR [s]+1	$32000*(1+0.18\%*1) = 32\,057.6\text{kHz}$
		OP_HRCR [s]+2	$32000*(1+0.18\%*2) = 32\,115.2\text{kHz}$
	
		OP_HRCR [s]+n	$32000*(1+0.18\%*n)\text{kHz}$
<p>Note:</p> <ol style="list-style-type: none">1. After each power-on of the IC, the value of OP_HRCR[7: 0] is the value of the high-frequency oscillator frequency f_{HRC} closest to 32MHz; the user can correct the value of HRC after each power-on to allow the system clock frequency f_{sys} of the IC to work at Frequency required by users;2. In order to ensure the reliable operation of the IC, the maximum operating frequency of the IC should not exceed 10% of 32MHz, that is 35.2MHz;3. Please confirm that the change of HRC frequency will not affect other functions.			

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

The SC95F852X has a built-in RC and 32.768kHz crystal oscillator circuit with a frequency of 32kHz, which can be used as the clock source of the Base Timer. The oscillator is directly connected to a Base Timer, which can wake the CPU from STOP mode and generate an interrupt.

BTMCON (FBH) Low-frequency Timer Control Register (read/write)

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

Bit Mnemonic	ENBTM	BTMIF	-	-	BTMFS[3: 0]			
R/W	R/W	R/W	-	-	R/W			
POR	0	0	x	x	0	0	0	0

Bit number	Bit Mnemonic	Description
7	ENBTM	Low frequency Base Timer start control 0: Base Timer and its clock source do not start 1: Base Timer and its clock source start
6	BTMIF	Base Timer interrupt application flag When the CPU accepts the Base Timer interrupt, this flag will be automatically cleared by hardware.
3~0	BTMFS [3: 0]	Low frequency clock interrupt frequency selection 0000: An interrupt is generated every 15.625ms 0001: An interrupt is generated every 31.25ms 0010: An interrupt is generated every 62.5ms 0011: An interrupt is generated every 125ms 0100: An interrupt is generated every 0.25 seconds 0101: An interrupt is generated every 0.5 seconds 0110: An interrupt is generated every 1.0 seconds 0111: An interrupt is generated every 2.0 seconds 1000: An interrupt is generated every 4.0ms 1001: An interrupt is generated every 8.0 seconds 1010: An interrupt is generated every 16.0 seconds 1011: An interrupt is generated every 32.0 seconds

		1100~1111: reserved
5~4	-	reserved

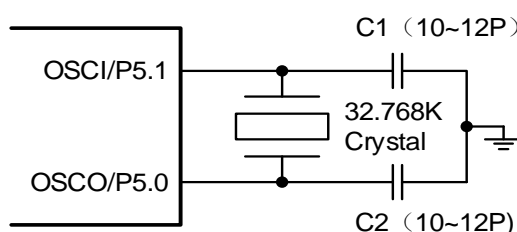
OP_CTM0 (C1H@FFH) Customer Option Register 0 (read/write)

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

Bit number	Bit Mnemonic	Description
6	ENXTL	<p>External 32.768kHz crystal selector switch</p> <p>0: The external 32.768kHz crystal is off, P5.0 and P5.1 are valid, and the internal LRC is valid;</p> <p>1: The external 32.768kHz crystal is turned on, P5.0 and P5.1 are invalid, and the internal LRC is invalid.</p>

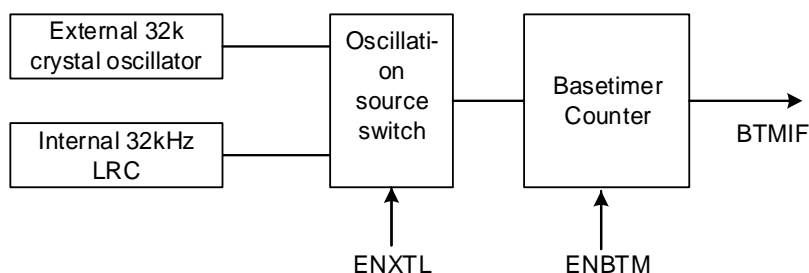
Note: HRC can be automatically calibrated by connecting a 32.768kHz crystal oscillator. Therefore, if the user uses the 32.768kHz external crystal oscillator function, the HRC frequency will always be corrected to 32MHz. At this time, adjusting OP_HRCR cannot change the HRC frequency.

The connection circuit used by P5.0/P5.1 external 32k oscillator as BaseTimer is as follows:



32.768kHz external crystal connection diagram

The internal and external oscillation selection relationship of Base Timer is as follows:



Base Timer Structure diagram

7.6 Power Saving Modes

The SC95F852X supports two different software selectable power-reducing modes: IDLE and STOP. These modes are accessed through the PCON register.

Setting the PCON.1 bit enters STOP mode. STOP mode stops the internal high-frequency oscillator in order to minimize power consumption. In STOP mode, users can wake up the SC95F852X through external interrupts INT0~INT2, low-frequency clock interrupt and WDT, or STOP through external reset.

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

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

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

Bit number	Bit Mnemonic	Description
------------	--------------	-------------

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

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

For example: set MCU to enter STOP mode:

Example in C Language

```
#include"intrins.h"
```

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

```
_nop_();        // At least 8 _nop_() are required
```

```
_nop_();
```

```
_nop_();
```

```
_nop_();
```

```
_nop_();
```

```
_nop_();
```

```
_nop_();
```

```
_nop_();
```

```
.....
```

Assembly Language:

```
ORL PCON,#02H    ; PCON bit1 STOP bit write 1, configure the MCU to enter STOP mode
```

```
NOP              ; At least 8 NOPs are required
```

```
NOP
```

```
NOP
```

```
NOP
```

```
NOP
```

```
NOP
```

```
NOP
```

```
.....
```

8 CPU and Instruction Set

8.1 CPU

The SC95F852X is built around an enhanced super-high-speed 1T 8051 core, and its instructions are fully compatible with classic 8051 core.

8.2 Addressing Mode

The addressing modes of 1T 8051 CPU instructions of the SC95F852X : ① Immediate 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 data. It directly gives the operands participating in the operation in the instruction operand. Examples of instructions are as follows:

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

8.2.2 Direct Addressing

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

Examples are as follows:

ANL 50H, #91H

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

8.2.3 Indirect Addressing

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

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

8.2.4 Register Addressing

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

INC R0 Refers to (R0)+1→R0

8.2.5 Relative Addressing

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

JC \$ +50H

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

8.2.6 Indexed Addressing

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

MOVC A, @A+DPTR

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

8.2.7 Bits Addressing

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

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

8.3 Introduction of Common Special Function Registers of 8051 Core

Program Counter PC

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

Accumulator ACC (E0H)

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

B Register (F0H)

The B register must be used with the accumulator A in multiplication and division operations. The multiplication instruction MUL A, B multiplies the 8-bit unsigned number in accumulator A and register B. The low-bit byte of the

resulting 16-bit product is placed in A, and the high-bit byte is placed in B. The division instruction DIV A, B divides A by B, the integer quotient is placed in A, and the remainder is placed in B. Register B can also be used as a general temporary storage register.

Stack Pointer SP (81H)

The stack pointer is an 8-bit special register that indicates the location of the top of the stack in general-purpose RAM. After the one-chip computer is reset, the initial value of SP is 07H, that is, the stack will increase upward from 08H. 08H~1FH is operating register group 1~3.

PSW (D0H) Program Status Word Register (read/write)

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

Bit number	Bit Mnemonic	Description
7	CY	Flag 1: When there is a carry in the highest bit of addition, or a borrow in the highest bit of subtraction 0: When there is no carry in the highest bit of addition, or there is no borrow in the highest bit of subtraction
6	AC	Carry auxiliary flag (can be easily adjusted during the addition and subtraction of BCD code) 1: When the addition operation has a carry in bit3, or the subtraction operation has a borrow in bit3 0: No borrowing, carry
5	F0	User flag

4~3	RS1、RS0	Operating register group selection bits:		
		RS1	RS0	Operating register set currently in use 0~3
		0	0	TEAM 0 (00H~07H)
		0	1	TEAM 1 (08H~0FH)
		1	0	TEAM 2 (10H~17H)
		1	1	TEAM 3 (18H~1FH)
2	OV	Overflow flag		
1	F1	F1 sign User-defined sign		
0	P	Parity flag. This flag bit is the parity value of the number of 1s in the accumulator ACC. 1: The number of 1s in ACC is odd 0: The number of 1s in ACC is even (including 0)		

Data Pointers DPTR0 (82H, 83H), DPTR1 (84H, 85H) And Its Selection Register DPS (86H)

The SC95F852X has two data pointers DPTR0 and DPTR1. Data pointers DPTR0/DPTR1 are 16-bit special registers, which are composed of low 8-bit DPL/DPL1 and high 8-bit DPH/DPH1. DPTR0/DPTR1 is a register that can directly perform 16-bit operations, and can also operate on DPL and DPH in bytes respectively. The selection and operating status of the data pointer DPTR0/DPTR1 are set by the data pointer selection register DPS.

DPS(86H)Data Pointer Selection Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	ID1	ID0	TSL	AU1	AU0	-	-	SEL

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

Bit number	Bit Mnemonic	Description
7	ID1	<p>DPTR1 plus or minus control bit</p> <p>0: When AU1=1, whenever the MOVC/MOVX @DPTR is executed, the current DPTR1 will automatically increase by 1.</p> <p>1: When AU1=1, whenever MOVC/MOVX @DPTR is executed, the current DPTR1 will automatically decrease by 1.</p>
6	ID0	<p>DPTR plus or minus control bit</p> <p>0: When AU0=1, whenever MOVC/MOVX @DPTR is executed, the current DPTR0 will automatically increase by 1.</p> <p>1: When AU0=1, whenever MOVC/MOVX @DPTR is executed, the current DPTR0 is automatically decremented by 1.</p>
5	TSL	<p>SEL flip control bit</p> <p>0: Whenever MOVC/MOVX @DPTR is executed, DPS.0 (SEL) does not flip</p> <p>1: Whenever MOVC/MOVX @DPTR is executed, DPS.0 (SEL) flips once</p>
4	AU1	<p>DPTR1 automatic plus and minus control bit</p> <p>0: None</p> <p>1: Whenever MOVC/MOVX @DPTR is executed, the current DPTR1 will increase or decrease by 1 (depending on ID1)</p>
3	AU0	<p>DPTR automatic plus and minus control bit</p> <p>0: None</p> <p>1: Whenever MOVC/MOVX @DPTR is executed, the current DPTR0 will increase or decrease by 1 (depending on ID0)</p>

0	SEL	DPTR0, DPTR1 selection bits 0: MOVX/MOVB @DPTR object is DPTR0 1: MOVX/MOVB @DPTR object is DPTR1
2~1	-	reserved

9 Interrupts

SC95F852X provides 16 interrupt sources: TIMER 0~4, INT0~2, ADC, PWM, UART, USCI0~2, BASE TIMER, TK. The 16 interrupt sources are divided into two interrupt priorities and can be set to either high or low priority separately. Three external interrupts can be set as up, down or both trigger conditions for each interrupt source respectively. Each interrupt has its own priority setting bit, interrupt flag, interrupt vector and enable bit respectively. The total enable bit EA can open or close all interrupts.

9.1 Interrupt Source and Vector

The list of the SC95F852X interrupt sources, interrupt vectors, and related control bits are as follows:

Interrupt Source	Interrupt condition	Interrupt Flag	Interrupt Enable Control	Interrupt Priority Control	Interrupt Vector	Query Priority	Interrupt Number (C51)	Flag Clear Mode	Capability of Waking up STOP
INT0	External interrupt 0 conditions are met	IE0	EINT0	IPINT0	0003H	1 (HIGH)	0	H/W Auto	YES
Timer0	Timer0 overflow	TF0	ET0	IPT0	000BH	2	1	H/W Auto	NO
INT1	External interrupt 1 conditions are met	IE1	EINT1	IPINT1	0013H	3	2	H/W Auto	YES
Timer1	Timer1 overflow	TF1	ET1	IPT1	001BH	4	3	H/W Auto	NO
UART	Receive or send completed	RI/TI	EUART	IPUART	0023H	5	4	Must user Clear	NO
Timer2	Timer2 overflow	TFX	ET2	IPT2	002BH	6	5	Must user Clear	NO
ADC	ADC conversion completed	ADCIF	EADC	IPADC	0033H	7	6	Must user Clear	NO

USCI0	Receive or send completed	SPIF0/TWIF0	ESSI0	IPSPI	003BH	8	7	Must user Clear	NO
PWM	PWM overflow	PWMIF	EPWM	IPPWM	0043H	9	8	Must user Clear	NO
BTM	Base timer overflow	BTMIF	EBTM	IPBTM	004BH	10	9	H/W Auto	YES
INT2	External interrupt 2 conditions are met	-	EINT2	IPINT2	0053H	11	10	-	YES
TK	Touch Key counter overflowed	TKIF	ETK	IPTK	005BH	12	11	H/W Auto	NO
Timer3	Timer3 overflow	TFX	ET3	IPT3	006BH	14	13	Must user Clear	NO
Timer4	Timer4 overflow	TFX	ET4	IPT4	0073H	15	14	Must user Clear	NO
USCI1	Receive or send completed	SPIF1/TWIF1	ESSI1	IPSPI1	007BH	16	15	Must user Clear	NO
USCI2	Receive or send completed	SPIF2/TWIF2	ESSI2	IPSPI2	0083H	17	16	Must user Clear	NO

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

Timer Interrupt: Interrupt generates when Timer0 or Timer1 overflows and the interrupt flag TF0 or TF1 is set to “1”. When the microcontroller unit responds to the timer interrupt, the interrupt flag TF0 or TF1 is reset automatically by hardware. Interrupt generates when Timer2 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.

UART Interrupt: When UART0 completes receiving or transmitting a frame of data, bit RI or TI will be set to “1” automatically by hardware, and UART interrupt occurs. Once UART interrupt occurs, the hardware would not automatically clear up RI/TI bit, which shall be cleared by user’s software.

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

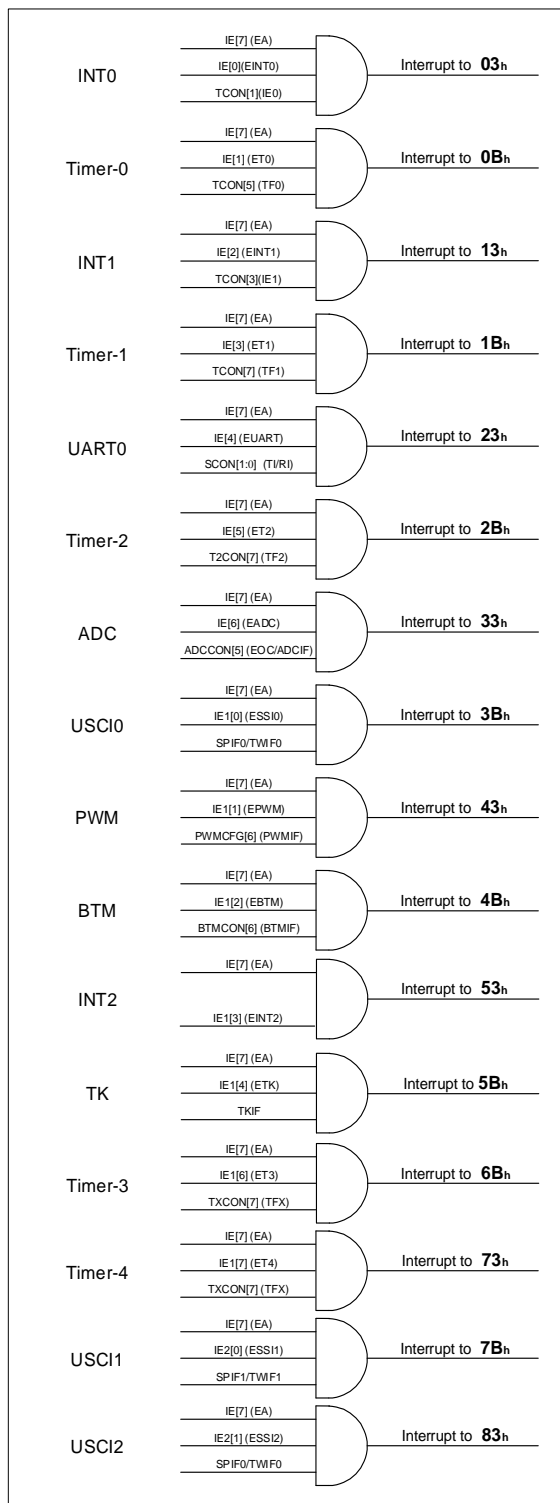
SSI Interrupt: When SSI completes receiving or transmitting a frame of data, SPIF/TWIF bit will be set to “1” automatically by hardware, and SSI interrupt generates. When the microcontroller unit serves SSI interrupt, the interrupt flag SPIF/TWIF must be cleared by software.

PWM Interrupt: When PWM counter overflows (beyond PWMPD), 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 occur. Once PWM interrupt occurs, the hardware would not clear the interrupt flag automatically, which shall be cleared by user’s software.

External Interrupt INT0 ~ 2: When any external interrupt pin meets the interrupt conditions, external interrupt generates. INT0 and INT1 will generate interrupt flags IE0/IE1, the user does not need to handle, the hardware will automatically clear. INT0 has three external interrupt sources, INT1 has four external interrupt sources, INT2 has six external interrupt sources, users can according to the need to set the upper edge, lower edge or double edge interrupt, can be achieved by setting SFR (INTxF and INTxR). The IP register allows the user to set the priority level of each interrupt. The external interrupt INT0~2 can also wake up the STOP of the SCM.

9.2 Interrupt Structure Diagram

The interrupt structure of SC95F852X is shown below:



SC95F852X Interrupt structure and vector

9.3 Interrupt Priority

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

9.4 Interrupt Processing Flow

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

- ① The currently executing instruction is finished;
- ② The PC value is pushed into the stack to protect the scene;
- ③ The interrupt vector address is loaded into the program counter PC;
- ④ Execute the corresponding interrupt service program;
- ⑤ The interrupt service routine ends and RETI;
- ⑥ Unstack the PC value and return to the program before the interruption.

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

9.5 Interrupt-related SFR Registers

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

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

Bit number	Bit Mnemonic	Description
------------	--------------	-------------

7	EA	Interrupt enable total control 0: Close all interrupts 1: Enable all interrupts
6	EADC	ADC interrupt enable control 0: Disable ADC interrupt 1: Allow the ADC to generate an interrupt when the conversion is complete
5	ET2	Timer2 interrupt enable control 0: Disable Timer2 interrupt 1: Enable Timer2 interrupt
4	EUART	UART interrupt enable control 0: Disable UART interrupt 1: Allow UART interrupt
3	ET1	Timer1 interrupt enable control 0: Disable Timer 1 interrupt1: Enable Timer1 interrupt
2	EINT1	External interrupt 1 enable control 0: close INT1 interrupt 1: Enable INT1 interrupt
1	ET0	Timer0 interrupt enable control 0: Disable TIMER0 interrupt 1: Enable TIMER0 interrupt
0	EINT0	External interrupt 0 enable control 0: close INT0 interrupt

		1: Enable INT0 interrupt
--	--	--------------------------

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

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

Bit number	Bit Mnemonic	Description
6	IPADC	ADC interrupt priority selection 0: ADC interrupt priority is low 1: ADC interrupt priority is high
5	IPT2	Timer2 interrupt priority selection 0: Timer2 interrupt priority is low 1: Timer2 interrupt priority is high
4	IPUART	UART interrupt priority selection 0: UART interrupt priority is low 1: UART interrupt priority is high
3	IPT1	Timer1 interrupt priority selection 0: Timer1 interrupt priority is low 1: Timer1 interrupt priority is high

2	IPINT1	INT1 counter interrupt priority selection 0: INT1 interrupt priority is low 1: INT1 interrupt priority is high
1	IPT0	Timer0 interrupt priority selection 0: Timer0 interrupt priority is low 1: Timer0 interrupt priority is high
0	IPINT0	INT0 counter interrupt priority selection 0: INT0 interrupt priority is low 1: INT0 interrupt priority is high
7	-	Reserved

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

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	ET4	ET3	-	ETK	EINT2	EBTM	EPWM	ESSIO
R/W	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
POR	0	0	x	0	0	0	0	0

Bit number	Bit Mnemonic	Description
7	ET4	Timer4 interrupt enable control 0: Disable Timer4 interrupt 1: Enable Timer4 interrupt

6	ET3	Timer3 interrupt enable control 0: Disable Timer3 interrupt 1: Enable Timer3 interrupt
4	TK	Touch Key interrupts enable control 0: Turn off Touch Key interrupt 1: Open Touch Key interrupt
3	EINT2	External interrupt 2 enable control 0: close INT2 interrupt 1: Open INT2 interrupt
2	EBTM	Base Timer interrupt enable control 0: Disable Base Timer interrupt 1: Enable Base Timer interrupt
1	EPWM	PWM interrupt enable control 0: Disable PWM interrupt 1: Enable interrupt when PWM count overflows
0	ESSIO	Three-in-one serial port USCIO interrupt enable control 0: Disable serial port interrupt 1: Allow serial port interrupt
5	-	Reserved

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

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

Bit Mnemonic	IPT4	IPT3	-	IPTK	IPINT2	IPBTM	IPPWM	IPSSI0
R/W	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
POR	0	0	x	0	0	0	0	0

Bit number	Bit Mnemonic	Description
7	IPT4	Timer4 interrupt priority selection 0: Timer4 interrupt priority is low 1: Timer4 interrupt priority is high
6	IPT3	Timer3 interrupt priority selection 0: Timer3 interrupt priority is low 1: Timer3 interrupt priority is high
4	IPTK	Touch Key interrupts priority selection 0: Touch Key interrupt priority is low 1: Touch Key interrupt priority is high
3	IPINT2	INT2 counter interrupt priority selection 0: INT2 interrupt priority is low 1: INT2 interrupt priority is high
2	IPBTM	Base Timer interrupt priority selection 0: Base Timer interrupt priority is low 1: Base Timer interrupt priority is high
1	IPPWM	PWM interrupt enable selection 0: PWM interrupt priority is low

		1: PWM interrupt priority is high
0	IPSSI0	Three-in-one serial port USC10 interrupt priority selection 0: USC10 interrupt priority is low 1: USC10 interrupt priority is high
0	-	Reserved

IE2 (AAH) Interrupt Enable Register 2 (read/write)

Bit number	7	6	5	4	3	2	1	0
Symbol	-	-	-	-	-	-	ESSI2	ESSI1
Read/ Write	-	-	-	-	-	-	Read/ Write	Read/ Write
Initial power-on value	x	x	x	x	x	x	0	0

Bit number	Bit Mnemonic	Description
1	ESSI2	Three-in-one serial port USC12 interrupt enable control 0: Disable serial port interrupt 1: Allow serial port interrupt
0	ESSI1	Three-in-one serial port USC11 interrupt enable control 0: Disable serial port interrupt 1: Allow serial port interrupt
7~2	-	Reserved

IP2 (BAH) Interrupt Priority Control Register 2 (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	-	-	-	-	-	-	IPSSI2	IPSSI1
R/W	-	-	-	-	-	-	R/W	R/W
POR	x	x	x	x	x	x	0	0

Bit number	Bit Mnemonic	Description
1	IPSSI2	Three-in-one serial port USC12 interrupt priority selection 0: USC12 interrupt priority is low 1: USC12 interrupt priority is high
0	IPSSI1	Three-in-one serial port USC11 interrupt priority selection 0: USC11 interrupt priority is low 1: USC11 interrupt priority is high
7~2	-	Reserved

TCON (88H) Timer Control Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	TF1	TR1	TF0	TR0	IE1	-	IE0	-
R/W	R/W	R/W	R/W	R/W	R/W	-	R/W	-

POR	0	0	0	0	0	x	0	x
-----	---	---	---	---	---	---	---	---

Bit number	Bit Mnemonic	Description
3	IE1	INT1 overflow interrupt request flag. INT1 generates an overflow. When an interrupt occurs, the hardware sets IE1 to "1" and applies for an interrupt. When the CPU responds, the hardware clears "0".
1	IE0	INT0 overflow interrupt request flag. INT0 generates an overflow. When an interrupt occurs, the hardware sets IE0 to "1" and applies for an interrupt. When the CPU responds, the hardware clears "0".
2,0	-	Reserved

INT0F (B4H) 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=3~1)	INT0 falling edge interrupt control 0: INT0n falling edge interrupt close 1: INT0n falling edge interrupt enable
7~4, 0	-	Reserved

INT0R (BBH) INT0 Rising Edge Interrupt Control Register (read/write)

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

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

INT1F (BCH) INT1 Falling Edge Interrupt Control Register (read/write)

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

Bit number	Bit Mnemonic	Description
------------	--------------	-------------

3~0	INT1Fn (n=3~0)	INT1 falling edge interrupt control 0: INT1n falling edge interrupt close 1: INT1n falling edge interrupt enable
7~4	-	Reserved

INT1R (BDH) INT1 Rising Edge Interrupt Control Register (read/write)

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

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

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

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	-	-	INT2F5	INT2F5	INT2F3	INT2F2	INT2F1	INT2F0

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

Bit number	Bit Mnemonic	Description
5~0	INT2Fn (n=5~0)	INT2 falling edge interrupt control 0: INT2n falling edge interrupt close 1: INT2n falling edge interrupt enable
7~6	-	Reserved

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

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	-	-	INT2R5	INT2R4	INT2R3	INT2R2	INT2R1	INT2R0
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
POR	x	x	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description
5~0	INT2Rn (n=5~0)	INT2 rising edge interrupt control 0: INT2n rising edge interrupt close 1: INT2n rising edge interrupt enable
7~6	-	Reserved

10 Timer/Counter T0 and T1

Timer0 and Timer1 inside the SC95F852X MCU are two 16-bit timers/counters. They have two operating modes: counting mode and timing mode. There is a control bit C/Tx in the special function register TMOD to select whether T0 and T1 are timers or counters. They are essentially an addition counter, but the source of the count is different. The source of the timer is the system clock or its divided clock, but the source of the counter is the input pulse of the external pin. Only when TRx=1, T0 and T1 will be opened to count.

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

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

There are 4 operating modes for timer/counter T0, and 3 operating modes for timer/counter T1 (mode 3 does not exist):

- ① Mode 0: 13-bit timer/counter mode
- ② Mode 1: 16-bit timer/counter mode
- ③ Mode 2: 8-bit auto-reload mode
- ④ Mode 3: Two 8-bit timer/counter modes

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

10.1 T0 and T1-related Registers

Symbol	Address	Description	7	6	5	4	3	2	1	0	POR
TCON	88H	Timer control register	TF1	TR1	TF0	TR0	IE1	-	IE0	-	00000x0xb
TMOD	89H	Timer operating mode register	-	C/T1	M11	M01	-	C/T0	M10	M00	x000x000b
TL0	8AH	Low 8 bits of timer 0	TL0[7: 0]								00000000b
TL1	8BH	Low 8 bits of timer 1	TL1[7: 0]								00000000b
TH0	8CH	Timer 0 high 8 bits	TH0[7: 0]								00000000b
TH1	8DH	Timer 1 high 8 bits	TH1[7: 0]								00000000b
TMCON	8EH	Timer frequency control register	USMD2[1: 0]	-	-	-	-	-	T1FD	T0FD	00xxxx00b

The explanation of each register is as follows:

TCON (88H) Timer Control Register (read/write)

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

Bit number	Bit Mnemonic	Description
7	TF1	T1 overflow interrupt request flag. T1 generates an overflow. When an interrupt occurs, the hardware sets TF1 to "1" and applies for an interrupt. When the CPU responds, the hardware clears "0".
6	TR1	Operation control bit of timer T1. This bit is set and cleared by software. When TR1=1, T1 is allowed to start counting. When TR1=0, T1 counting is prohibited.
5	TF0	T0 overflow interrupt request flag. T0 overflows. When an interrupt occurs, the hardware sets TF0 to "1" and applies for an interrupt. When the CPU responds, the hardware clears "0".
4	TR0	Operation control bit of timer T0. This bit is set and cleared by software. When TR0=1, T0 is allowed to start counting. When TR0=0, T0 counting is prohibited.
2,0	-	Reserved

TMOD (89H) Timer Operating Mode Register (read/write)

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

Bit Mnemonic	-	C/T1	M11	M01	-	C/T0	M10	M00
R/W	-	R/W	R/W	R/W	-	R/W	R/W	R/W
POR	x	0	0	0	x	0	0	0
	T1				T0			

Bit number	Bit Mnemonic	Description
6	C/T1	TMOD[6] control timer 1 0: Timer, T1 count comes from fsys frequency division 1: Counter, T1 count comes from external pin T1/P1.3
5~4	M11,M01	Timer/Counter 1 mode selection 00: 13-bit timer/counter, the upper 3 bits of TL1 are invalid 01: 16-bit timer/counter, TL1 and TH1 all 10: 8-bit auto-reload timer, automatically reload the value stored in TH1 into TL1 when overflow 11: Timer/Counter 1 is invalid (stop counting)
2	C/T0	TMOD[2] control timer 0 0: Timer, T0 count comes from fsys frequency division 1: Counter, T0 count comes from external pin T0/P1.2
1~0	M10,M00	Timer/Counter 0 mode selection 00: 13-bit timer/counter, the upper 3 bits of TL0 are invalid 01: 16-bit timer/counter, TL0 and TH0 all 10: 8-bit auto-reload timer, automatically reload the value stored in TH0 into TL0 when overflow

		11: Timer 0 is now a dual 8-bit timer/counter. TL0 is an 8-bit timer/counter controlled by the control bits of standard timer 0; TH0 is only an 8-bit timer controlled by the control bits of timer 1.
7,3	-	Reserved

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

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

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

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	USMD2[1: 0]		-	-	-	-	T1FD	T0FD
R/W	R/W	R/W	-	-	-	-	R/W	R/W
POR	0	0	x	x	x	x	0	0

Bit number	Bit Mnemonic	Description
1	T1FD	T1 input frequency selection control 0: T1 frequency is derived from fsys/12 1: T1 frequency is derived from fsys
0	T0FD	T0 input frequency selection control 0: T0 frequency is derived from fsys/12 1: T0 frequency is derived from fsys

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

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

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

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

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

Bit number	Bit Mnemonic	Description
------------	--------------	-------------

3	IPT1	Timer1 interrupt priority 0: Set the interrupt priority of Timer 1 to "Low" 1: Set the interrupt priority of Timer 1 to "High"
1	IPT0	Timer0 interrupt priority 0: Set the interrupt priority of Timer 0 to "Low" 1: Set the interrupt priority of Timer 0 to "High"

10.2 T0 Operating Modes

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

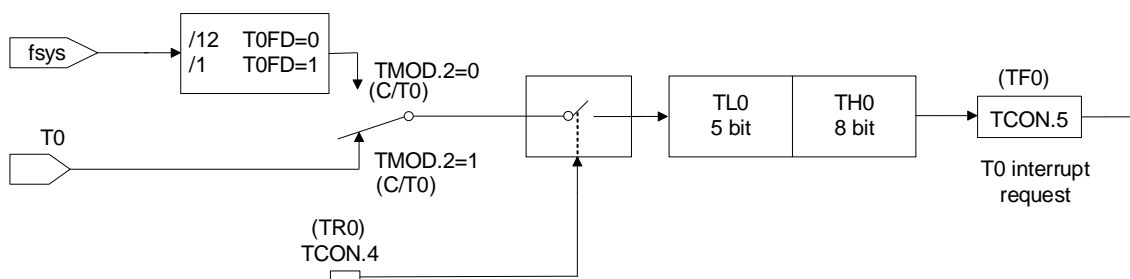
Operating Mode 0: 13-bit Counter/Timer

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

C/T0 bit selects the clock input source of the counter/timer. If C/T0=1, the level change of the timer 0 input pin T0 (P0.2) from high to low will increase the timer 0 data register by 1. If C/T0=0, select the frequency division of the system clock as the clock source of timer 0.

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

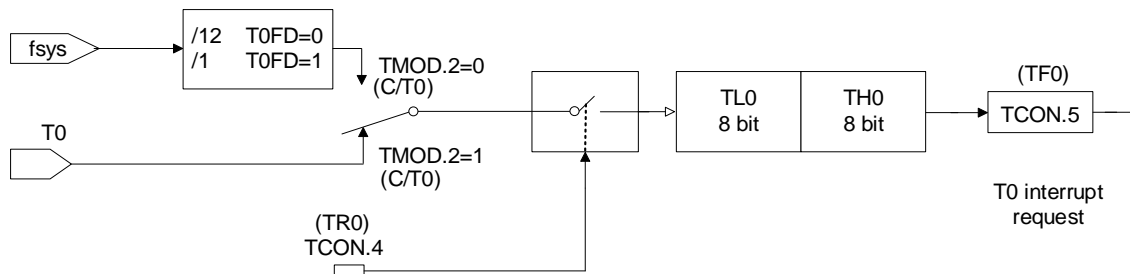
When applied as a timer, TOFD can be configured to select the frequency division ratio of the clock source.



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

Operating Mode 1: 16-bit Counter/Timer

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



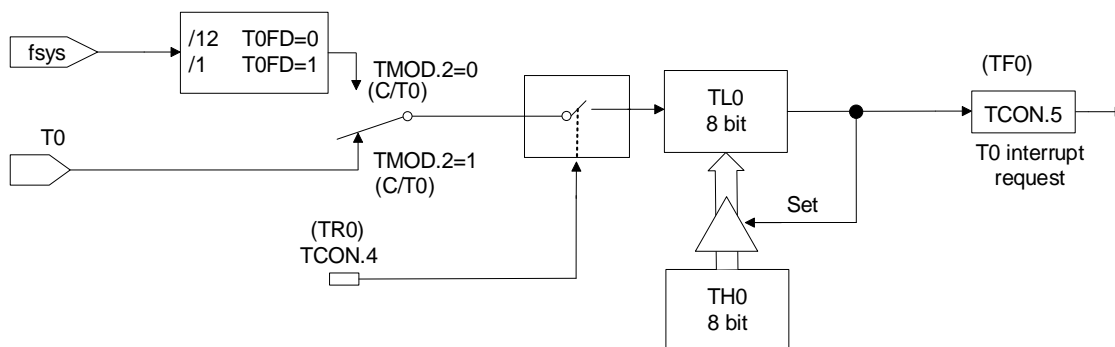
Timer/Counter Operating Mode 0: 16-bit Timer/Counter

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

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

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

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



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

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

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

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

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

10.3 T1 Operating Mode

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

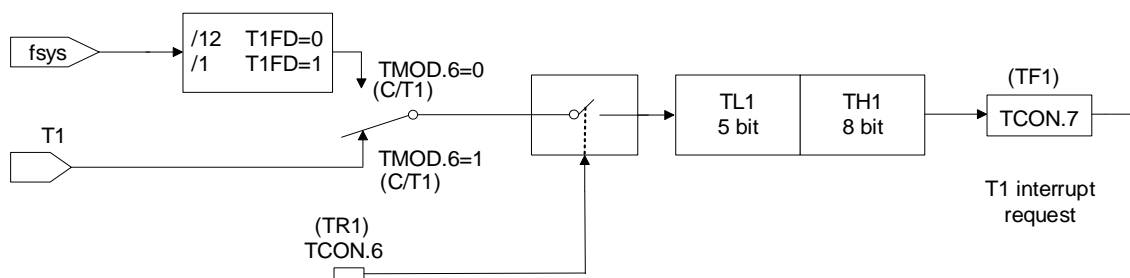
Operating mode 0: 13-bit Timer/Counter

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

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

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

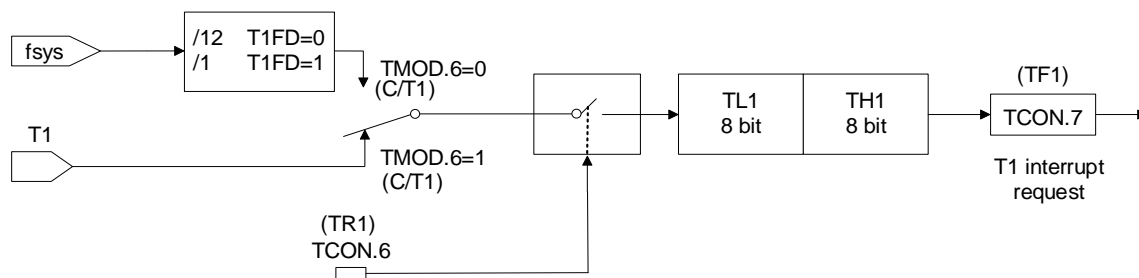
When applied as a timer, T1FD can be configured to select the frequency division ratio of the clock source.



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

Operating mode 1: 16-bit Counter/Timer

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



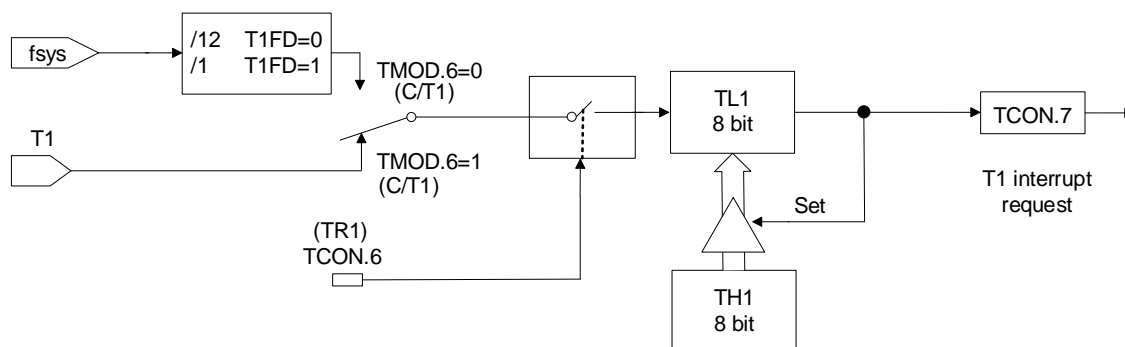
Timer/counter operating mode 0: 16-bit timer/counter

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

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

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

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



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

11 Timer/Counter T2/T3/T4

Timer2/3/4 inside The SC95F852X MCU are three independent Timers, among which Timer2 has 4 operating modes, Timer3 and Timer4 have 3 operating mode.

The control registers of Timer2/3/4 share the same set of addresses (C8H-CDH), users can point the TimerX register set (TXCON / TXMOD / RCAPXL / RCAPXH / TLX / THX) to Timer2/3/4 through TXINX[2: 0] In order to realize the function of three independent Timers configured by a group of registers.

Note: Only after the TXINX[2: 0] configuration is successful, the TimerX register group will point to the Timer2/3/4 specified by the user. At this time, operating the TimeX register group is an effective operation for the corresponding Timer.

11.1 T2/3/4-related Registers

Symbol	Address	Description	7	6	5	4	3	2	1	0	POR
TXINX	CEH	Timer 2/3/4 control register pointer	-	-	-	-	-	TXINX[2: 0]			xxxxx010b
TXCON	C8H	Timer 2/3/4 control register	TFX	EXFX	RCLK X	TCL KX	EXEN X	TRX	C/TX	CP/RL X	00000000b
TXMOD	C9H	Timer 2/3/4 operating mode register	TXF D	-	-	-	-	-	TXO E	DCXE N	0xxxxx00b
RCAPXL	CAH	Timer 2/3/4 reload low 8 bits	RCAPXL[7: 0]								00000000b
RCAPXH	CBH	Timer 2/3/4 reload high 8 bits	RCAPXH[7: 0]								00000000b
TLX	CCH	Timer 2/3/4 low 8 bits	TLX[7: 0]								00000000b
THX	CDH	Timer 2/3/4 high 8 bits	THX[7: 0]								00000000b
TMCON	8EH	Timer frequency control register	USMD2[1: 0]	-	-	-	-	-	T1FD	T0FD	00xxxx00b

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

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

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

11.2 Timer2

Timer2 inside the SC95F852X MCU has two operating modes: counting mode and timing mode. There is a control bit C/TX in the special function register TXCON to select whether T2 is a timer or a counter. They are essentially an addition counter, but the source of the count is different. The source of the timer is the system clock or its divided clock, but the source of the counter is the input pulse of the external pin. TRX is the switch control of T2/T3/T4 counting in the timer/counter mode. Only when TRX=1, T2 will be opened for counting.

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

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

Timer/counter T2 has 4 operating modes:

- ① Mode 0: 16-bit capture mode
- ② Mode 1: 16-bit auto-reload timer mode

③ Mode 2: Baud rate generator mode

④ Mode 3: Programmable clock output mod

TXINX[2: 0] = 010, the TimerX register group points to Timer2, the explanation of each register is as follows:

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

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

Bit number	Bit Mnemonic	Description
7	TFX	Timer 2 overflow flag 0: No overflow (must be cleared by software) 1: Overflow (if RCLKX = 0 and TCLKX = 0, set by hardware 1)
6	EXFX	Flag bit detected by external event input (falling edge) of T2EX pin 0: No external event input (must be cleared by software) 1: External input detected (if EXENX = 1, set by hardware)
5	RCLKX	UART0 receive clock control bit 0: Timer 1 generates the receive baud rate 1: Timer 2 generates the receive baud rate
4	TCLKX	UART0 transmit clock control bit 0: Timer 1 generates transmission baud rate

		1: Timer 2 generates transmission baud rate
3	EXENX	T2EX pin is used as a reload/capture trigger enable/disable control: 0: Ignore events on T2EX pin 1: When Timer 2 is not used as the UART0 clock, a falling edge on the T2 pin is detected, and a capture or reload is generated
2	TRX	Timer 2 start/stop control bit 0: stop timer 2 1: Start timer 2
1	C/TX	Timer 2 Timer/counter mode selection positioning 2 0: Timer mode, T2 pin is used as I/O port 1: Counter mode
0	CP/RLX	Capture/reload mode selection positioning 0: 16-bit timer/counter with reload function 1: 16-bit timer/counter with capture function, TXEX is timer 2 external capture signal input port

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

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	TXFD	-	-	-	-	-	TXOE	DCXEN
R/W	R/W	-	-	-	-	-	R/W	R/W
POR	0	x	x	x	x	x	0	0

Bit number	Bit Mnemonic	Description
------------	--------------	-------------

7	TXFD	T2 input frequency selection control 0: T2 frequency is derived from fsys/12 1: T2 frequency is derived from fsys
1	TXOE	Timer 2 output enable bit 0: Set T2 as clock input or I/O port 1: Set T2 as the clock output
0	DCXEN	Count down enable bit 0: Timer 2 is prohibited as an up/down counter, Timer 2 is only used as an up counter 1: Allow Timer 2 as an up/down counter
6~2	-	Reserved

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

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

Bit number	Bit Mnemonic	Description
5	ET2	Timer2 interrupt enable control 0: Disable Timer2 interrupt 1: Enable Timer2 interrupt

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

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

Bit number	Bit Mnemonic	Description
5	IPT2	Timer2 interrupt priority 0: Set the interrupt priority of Timer2 to "Low" 1: Set the interrupt priority of Timer2 to "High"

11.3 Timer3

Timer3 in SC95F852X MCU has two working modes: counting mode and timing mode. The special function register TXCON has a control bit C/TX to choose whether T3 is a timer or a counter. They are essentially an addition counter, just from different sources. The source of the timer is the system clock or its divider clock, but the source of the counter is the input pulse of the external pin. TRX is the switch control of T3 counting in timer/counter mode. Only when TRX=1, T3 will be turned on to count.

In counter mode, the count value of T3 increases by 1 for each pulse on T3 pin.

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

Timer/counter T3 has 3 operating modes:

Mode 0: 16-bit 16 capture mode

Mode 1: 16-bit auto-reload timer mode

Mode 2: programmable clock output mode

TXINX[2: 0] = 011, the TimerX register group points to Timer3, the explanation of each register is as follows:

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

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

Bit number	Bit Mnemonic	Description
7	TFX	Timer 3 control register (read/write) Timer 3 overflow flag 0: No overflow (must be cleared by software) 1: Overflow (set by hardware 1)
6	EXFX	Flag bit that the T3 pin external event input (falling edge) is detected 0: No external event input (must be cleared by software) 1: External input detected (if EXENX = 1, set by hardware to 1)
3	EXENX	The external event input (falling edge) on the T3EX pin is used as a reload/capture trigger to allow/disable control: 0: Ignore events on T3EX pins 1: The T3EX is always connected to a pull-up resistor, which detects a falling edge on the T3EX pin, resulting in a capture or overload
2	TRX	Timer 3 start/stop control bit 0: stop timer 3 1: Start timer 3
1	C/TX	Timer 3 Timer/counter mode selected bit 0: Timer mode, T3 pin used as I/O port

		1: counter mode
0	CP/RLX	Capture/overload mode select bit 0: 16 bit timer/counter with overload function 1: 16 bit timer/counter with capture function, TXEX for timer 3 external capture signal input port
5~4	-	reserved

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

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	TXFD	-	-	-	-	-	TXOE	DCXEN
R/W	R/W	-	-	-	-	-	R/W	R/W
POR	0	x	x	x	x	x	0	0

Bit number	Bit Mnemonic	Description
7	TXFD	T3 input frequency selection control 0: T3 frequency is derived from fsys/12 1: T3 frequency is derived from fsys
1	TXOE	Timer 3 outputs the allowed bit 0: Set T3 as the clock input or I/O port 1: Set T3 as the clock output
0	DCXEN	Decrement count allows bits 0: Disallow timer 3 as an increment/decrement counter, Timer 3 is used only as an increment counter

		1: Allow timer 3 as an increment/decrement counter
6~2	-	Reserved

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

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	ET4	ET3	-	ETK	EINT2	EBTM	EPWM	ESSIO
R/W	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
POR	0	0	x	0	0	0	0	0

Bit number	Bit Mnemonic	Description
6	ET3	Timer3 interrupt enable control 0: Disable Timer3 interrupt 1: Enable Timer3 interrupt

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

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	IPT4	IPT3	-	IPTK	IPINT2	IPBTM	IPPWM	IPSSIO
R/W	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
POR	0	0	x	0	0	0	0	0

Bit number	Bit Mnemonic	Description
6	IPT3	Timer3 interrupt priority selection 0: Timer3 interrupt priority is low 1: Timer3 interrupt priority is high

11.4 Timer4

Timer4 in SC95F852X microcontroller has two working modes: counting mode and timing mode. The special function register TXCON has a control bit C/TX to choose whether T4 is a timer or a counter. They are essentially an addition counter, just from different sources. The source of the timer is the system clock or its divider clock, but the source of the counter is the input pulse of the external pin. TRX is the on-off control of T4 counting in timer/counter mode. T4 will only be counted when TRX=1.

In counter mode, the T4 count increases by 1 for each pulse on the T4 pin.

In timer mode, the special function register TXMOD.7(TXFD) can be used to select whether the count source of T4 is $f_{SYS}/12$ or f_{SYS} .

Timer/counter T4 has 3 operating modes:

- ① Mode 0:16 bit capture mode
- ② Mode 1:16 bit automatic overload timer mode
- ③ Mode 3: Programmable clock output mode

TXINX[2: 0] = 100, TimerX register group points to Timer4, the explanation of each register is as follows:

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

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

Bit number	Bit Mnemonic	Description
7	TFX	Timer 4 overflow flag 0: No overflow (must be cleared by software) 1: Overflow (set by hardware 1)
6	EXFX	The flag bit that the external event input (falling edge) of the T4 pin is detected 0: No external event input (must be cleared by software) 1: External input detected (if EXENX = 1, set by hardware to 1)
3	EXENX	The external event input (falling edge) on the T4EX pin is used as a reload/capture trigger to allow/disable control: 0: Ignore events on T4EX pins 1: The T4EX is always connected to a pull-up resistor, which detects a falling edge on the T4EX pin, resulting in a capture or overload
2	TRX	Timer 4 start/stop control bit 0: stop timer 4 1: Start timer 4
1	C/TX	Timer 4 Timer/counter mode selected bit 0: Timer mode, T4 pin used as I/O port 1: counter mode
0	CP/RLX	Fixed write 0
5~4	-	Reversed

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

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

Bit Mnemonic	TXFD	-	-	-	-	-	TXOE	DCXEN
R/W	R/W	-	-	-	-	-	R/W	R/W
POR	0	x	x	x	x	x	0	0

Bit number	Bit Mnemonic	Description
7	TXFD	T4 input frequency selection control 0: T4 frequency is derived from fsys/12 1: T4 frequency is derived from fsys
1	TXOE	Timer 4 outputs the allowed bit 0: Set T4 as the clock input or I/O port 1: Set T4 as the clock output
0	DCXEN	Decrement count allows bits 0: Disallow timer 4 as an increment/decrement counter and timer 4 as an increment counter only, Timer 4 is used only as an increment counter 1: Allow timer 4 as an increment/decrement counter
6~2	-	Reversed

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

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	ET4	ET3	-	ETK	EINT2	EBTM	EPWM	ESSIO
R/W	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W

POR	0	0	x	0	0	0	0	0
-----	---	---	---	---	---	---	---	---

Bit number	Bit Mnemonic	Description
7	ET4	Timer4 interrupt enable control 0: Disable Timer4 interrupt 1: Enable Timer4 interrupt

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

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	IPT4	IPT3	-	IPTK	IPINT2	IPBTM	IPPWM	IPSSIO
R/W	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
POR	0	0	x	0	0	0	0	0

Bit number	Bit Mnemonic	Description
7	IPT4	Timer4 interrupt priority selection 0: Timer4 interrupt priority is low 1: Timer4 interrupt priority is high

11.5 T2/3/4 Operating Modes

Timer T2 has four operating modes:

- ① Mode 0: 16-bit capture

- ② Mode 1: 16-bit auto-reload timer
- ③ Mode 2: Baud rate generator
- ④ Mode 3: Programmable clock output

The operating mode and configuration mode of timer 2 are as follows:

C/TX	TXOE	DCXEN	TRX	CP/RLX	RCLKX	TCLKX	Mode	
X	0	X	1	1	0	0	0	16-bit capture
X	0	0	1	0	0	0	1	16-bit auto-reload timer
X	0	1	1	0	0	0		
X	0	X	1	X	1	X	2	Baud rate generator (only Timer2)
					X	1		
0	1	X	1	X	0	0	3	Only for programmable clock
					1	X	3	Programmable clock output with baud rate generator
					X	1		
X	X	X	0	X	X	X	X	Timer stops, TnEX(n=2/3/4) channel is still allowed
1	1	X	1	X	X	X		Not recommended

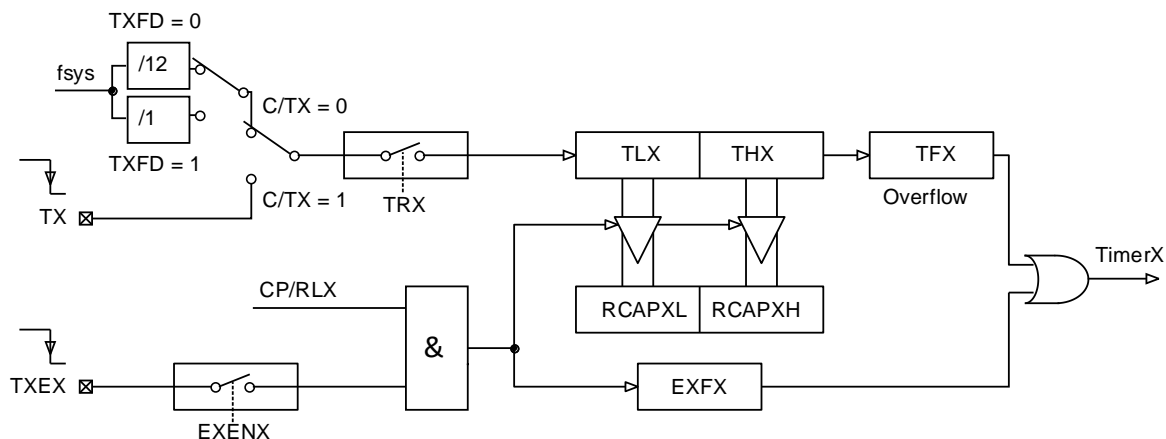
11.5.1 Operating Mode 0: 16-bit Capture

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

If EXENX = 0, Timer T_n (n=2/3/4) acts as a 16-bit timer or counter. If ET_n is enabled, T_n can set TFX overflow to generate an interrupt.

If EXENX = 1, Timer T_n (n=2/3/4) performs the same operation, but the falling edge on external input TnEX can also cause the current values in THX and TLX to be captured in RCAPXH and RCAPXL, respectively. In addition,

the falling edge on TnEX also Can cause EXFX in TXCON to be set. If ETn is enabled, the EXFX bit also generates an interrupt like TFX.



Mode 0: 16-bit capture

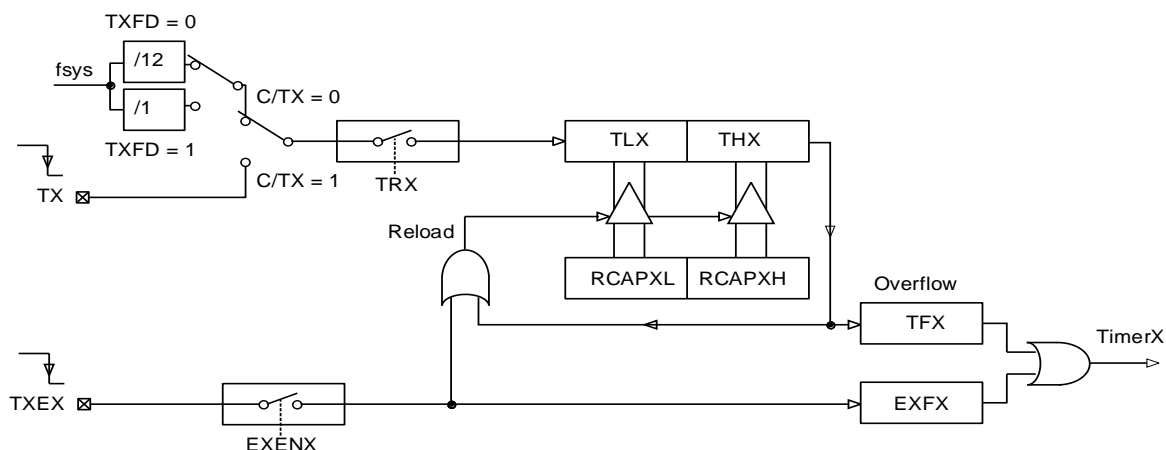
11.5.2 Operating Mode 1: 16-bit Auto-Reload Timer

In 16-bit auto-reload mode, Timer T_n (n=2/3/4) can be selected to count up or count down. This function is selected by the DCEN bit (down counting allowed) in TXMOD. After the system is reset, the reset value of the DCEN bit is 0, and the T_n counts up by default. When DCEN is set to 1, T_n counts up or down depending on the level on the TnEX pin.

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

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

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



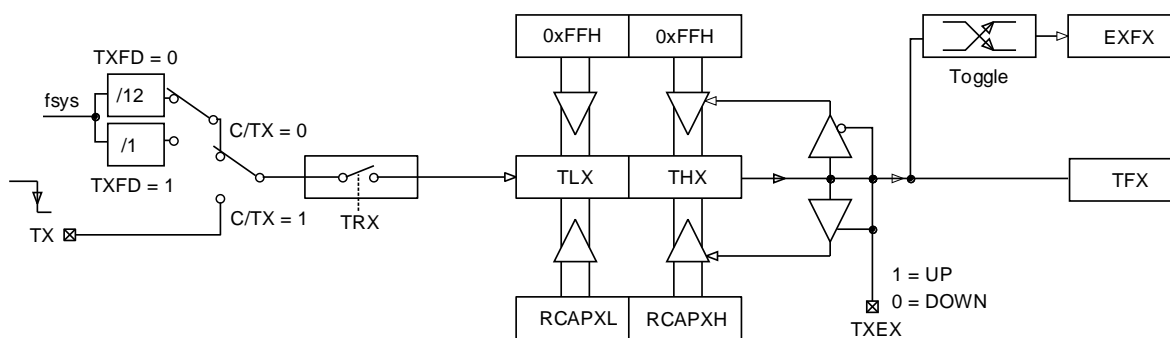
Mode 1: 16-bit auto-reload DCEN = 0

Setting the DCEN bit allows T_n (n=2/3/4) to count up or down. When DCEN = 1, the T_nEX pin controls the direction of the count, and EXENX control is invalid.

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

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

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



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

11.5.3 Operating Mode 2: Baud Rate Generator

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

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

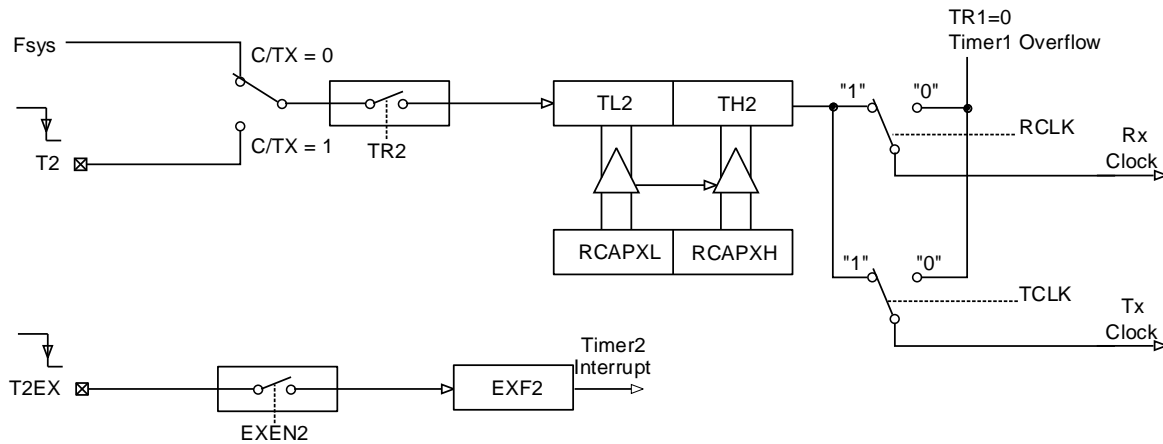
The overflow of Timer 2 will reload the values in the RCAPXH and RCAPXL registers to the Timer 2 count, but no interrupt will be generated.

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

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

$$\text{BaudRate} = \frac{f_{\text{sys}}}{[\text{RCAPXH}, \text{RCAPXL}]}; \text{ (Note: } [\text{RCAPXH}, \text{RCAPXL}] \text{ must be bigger than } 0x0010 \text{)}$$

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



Mode 2: Baud rate generator

11.5.4 Operating Mode 3: Programmable Clock Output

In this way, T2(P0.7)/ T3(P0.1)/ T4(P2.5) can be programmed to output a 50% duty cycle clock cycle: when C/TX = 0; TXOE = 1, Tn (n=2/3/4) is enabled as a clock generator

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

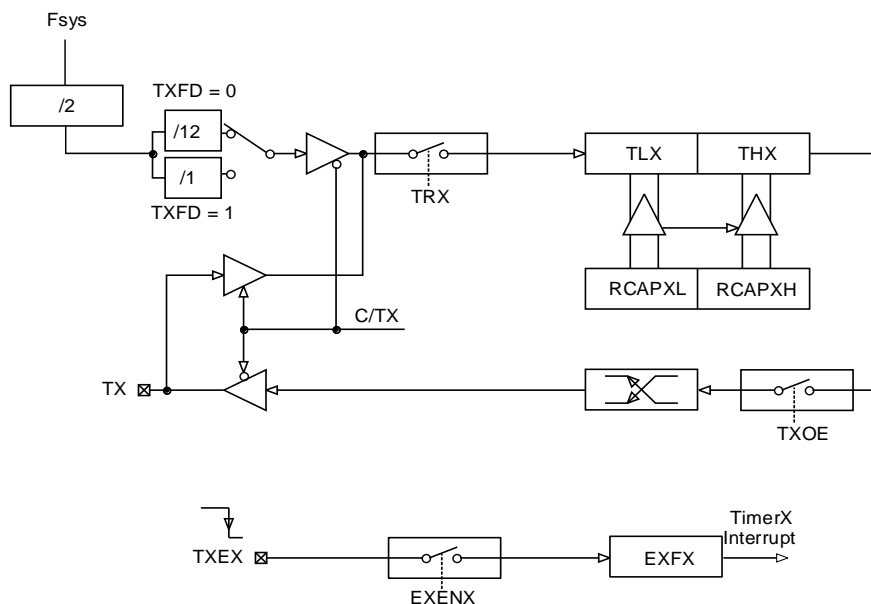
$$\text{Colck Out Frequency} = \frac{f_{tn}}{(65536 - [RCAPXH, RCAPXL]) \times 4};$$

Among them, f_{tn} is the Tn (n=2/3/4) clock frequency:

$$f_{tn} = \frac{f_{sys}}{12}, \quad TXFD = 0$$

$$f_{tn} = f_{sys}; \quad TXFD = 1$$

Tn overflow does not generate an interrupt, and the T2(P0.7)/ T3(P0.1)/ T4(P2.5) port is used as a clock output.



Mode 3: Programmable clock output

Note:

1. Both TFX and EXFX can cause the interrupt request of Tn (n=2/3/4) , both have the same vector address;
2. When the event occurs or at any other time, TFX and EXFX can be set to 1 by software, and only software and hardware reset can clear it to 0;
3. When EA = 1 and ETn = 1, setting TFX or EXFX to 1 can cause Tn to interrupt;
4. When Timer 2 is used as a baud rate generator, writing THX/TLX or RCAPXH/RCAPXL during UART0 communication will affect the accuracy of the baud rate and cause communication errors.

12 Multiplier-Divider Unit (MDU)

The SC95F852X provides a 16-bit multiplier and divider, which consists of extended accumulators EXA0~EXA3, extended B register EXB and operation control register OPERCON. Can replace software for 16 x 16 bit multiplication and 32 /16 bit division.

The SC95F852X hardware multiplier and divider does not occupy CPU cycles, and the operation is implemented by hardware. The speed is dozens of times faster than the software implementation of multiplication and division. It can replace software for 16-bit × 16-bit multiplication and 32-bit/16-bit division and increase program running efficiency.

Symbol	Address	Description	7	6	5	4	3	2	1	0	POR
EXA0	E9H	Extended Accumulator 0	EXA [7: 0]								00000000b
EXA1	EAH	Extended Accumulator 1	EXA [15: 8]								00000000b
EXA2	EBH	Extended Accumulator 2	EXA [23: 16]								00000000b
EXA3	ECH	Extended Accumulator 3	EXA [31: 24]								00000000b
EXBL	EDH	Extended B register L	EXB [7: 0]								00000000b
EXBH	EEH	Extended B register H	EXB [15: 8]								00000000b
OPERCON	EFH	Operation control register	OPE RS	MD	-	-	-	-	CRCR ST	CRCS TA	00xxxx00b

OPERCON (EFH) Operation control register (read/write)

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

Bit Mnemonic	OPERS	MD	-	-	-	-	CRCRST	CRCSTA
R/W	R/W	R/W	-	-	-	-	R/W	R/W
POR	0	0	x	x	x	x	0	0

Bit number	Bit Mnemonic	Description																									
7	OPERS	<p>Multiplier-divider operation start trigger control (Operater Start)</p> <p>Write "1" to this bit to start a multiplication and division calculation, that is, this bit is just the trigger signal for the multiplication and division to start calculation. When the bit is zero, it means that the calculation has been completed. This bit can only be written to 1 valid.</p>																									
6	MD	<p>Multiplication and division</p> <p>0: Multiplication operation. The multiplicand and multiplier are written and the product is read as follows:</p> <table><tr><td><div>Byte Operand</div></td><td>Byte 3</td><td>Byte 2</td><td>Byte 1</td><td>Byte 0</td></tr><tr><td>multiplicand 16bit</td><td>-</td><td>-</td><td>EXA1</td><td>EXA0</td></tr><tr><td>multiplier 16bit</td><td>-</td><td>-</td><td>EXBH</td><td>EXBL</td></tr><tr><td>multiplier 32bit</td><td>EXA3</td><td>EXA2</td><td>EXA1</td><td>EXA0</td></tr></table> <p>1: Divide operation, write the dividend and divisor, read the quotient and remainder as follows:</p> <table><tr><td><div>Byte Operand</div></td><td>Byte 3</td><td>Byte 2</td><td>Byte 1</td><td>Byte 0</td></tr></table>	<div>Byte Operand</div>	Byte 3	Byte 2	Byte 1	Byte 0	multiplicand 16bit	-	-	EXA1	EXA0	multiplier 16bit	-	-	EXBH	EXBL	multiplier 32bit	EXA3	EXA2	EXA1	EXA0	<div>Byte Operand</div>	Byte 3	Byte 2	Byte 1	Byte 0
<div>Byte Operand</div>	Byte 3	Byte 2	Byte 1	Byte 0																							
multiplicand 16bit	-	-	EXA1	EXA0																							
multiplier 16bit	-	-	EXBH	EXBL																							
multiplier 32bit	EXA3	EXA2	EXA1	EXA0																							
<div>Byte Operand</div>	Byte 3	Byte 2	Byte 1	Byte 0																							

		dividend 32bit	EXA3	EXA2	EXA1	EXA0
		divisor 16bit	-	-	EXBH	EXBL
		quotient 32bit	EXA3	EXA2	EXA1	EXA0
		remainder 16bit	-	-	EXBH	EXBL

Note:

1. It is forbidden to perform read or write operations on the EXA and EXB data registers during the calculation operation.
2. The time required for the operation conversion of the multiplier-divider is $16/f_{sys}$.

13 PWM

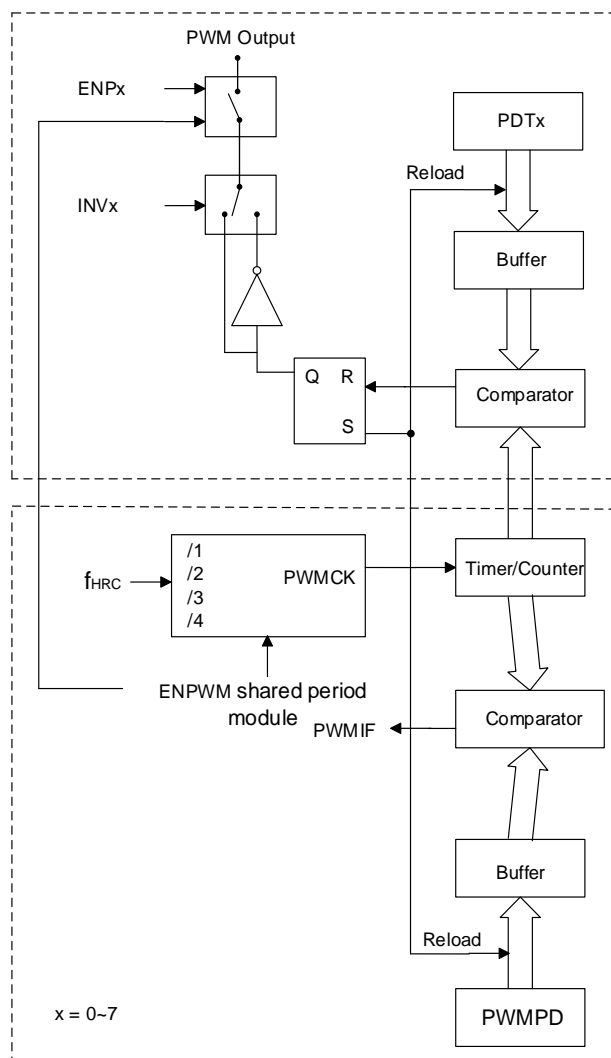
The SC95F852X provides 16-bit PWM with up to 8 shared cycles and individually adjustable duty cycle.

The functions of the PWM of the SC95F852X are as follows:

- ① 16-bit PWM accuracy;
- ② The output waveform can be reversed;
- ③ Type: Can be set to center-aligned or edge-aligned;
- ④ Mode: can be set to independent mode or complementary mode:
 - 1) In independent mode, the 8 PWM cycles are the same, but the duty cycle of each PWM output waveform can be set separately;
 - 2) In complementary mode, four sets of complementary PWM waveforms with dead zones can be output simultaneously;
- ⑤ Provide one PWM overflow interrupt;
- ⑥ Support fault detection mechanism.

The PWM of the SC95F852X can support the adjustment of period and duty cycle. The registers PWMCFG and PWMCON0 and PWEMCON1 control the state and period of PWM. The opening of each PWM and the output waveform duty cycle can be adjusted separately.

13.1 PWM Structure Diagram



SC95F852X PWM Structure diagram

13.2 PWM General Configuration Register

13.2.1 PWM General Configuration Register

The user can set the PWM output mode of SC95F852X to independent mode or complementary mode by configuring PWMMD[1: 0]. In independent mode, the 8 PWM cycles are the same, but the duty cycle of each PWM output waveform can be set separately. In complementary mode, four complementary PWM waveforms with dead zones can be output simultaneously.

The PWM type of SC95F852X is divided into edge-aligned type and center-aligned type:

Edge-aligned:

The PWM counter starts counting from 0, and when the count value matches the value of the duty cycle setting item PDTx [15: 0], the PWM output waveform switches between high and low levels, and then the PWM counter continues to count up until it matches the period setting item PWMPD[15: 0]. The value of +1 matches (the end of a PWM period), the PWM counter is cleared, if the PWM interrupt is enabled, a PWM interrupt will be generated at this time.

The output PWM waveform is aligned on the left edge.

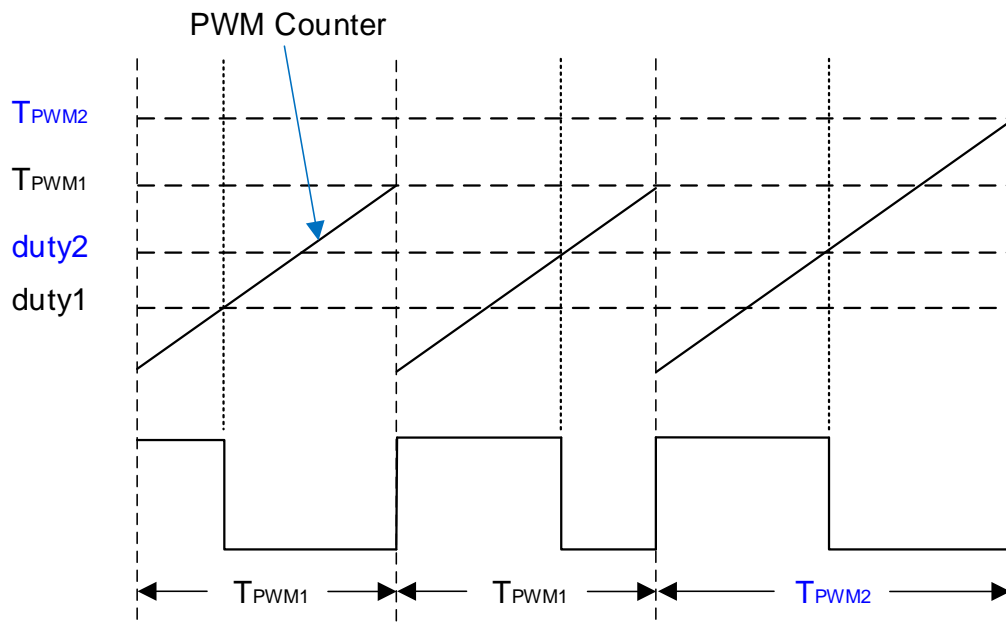
Calculation formula of edge-aligned period T_{PWM} :

$$T_{pwm} = \frac{PWMPD[15: 0] + 1}{PWM \text{ Clock frequency}}$$

Edge-aligned duty calculation formula:

$$duty = \frac{PDT_{xy} [15: 0]}{PWMPD[15: 0] + 1}$$

The edge-aligned waveform is as follows:



Edge-aligned PWM

Center-aligned type:

The PWM counter starts counting from 0. When the count value matches the value of the duty cycle setting item PDTx [15: 0], the PWM output waveform switches between high and low levels. Then the PWM counter continues to count up. When the count value matches the period setting item PWMPD [15: 0] When the value of +1 matches (that is, the midpoint of the PWM cycle), it automatically starts to count down. When the count value matches the value of PDTxy [11: 0] again, the PWM output waveform switches high and low again, and then The PWM counter continues to count down until it overflows (the end of a PWM period). If the PWM interrupt is enabled, a PWM interrupt will be generated at this time.

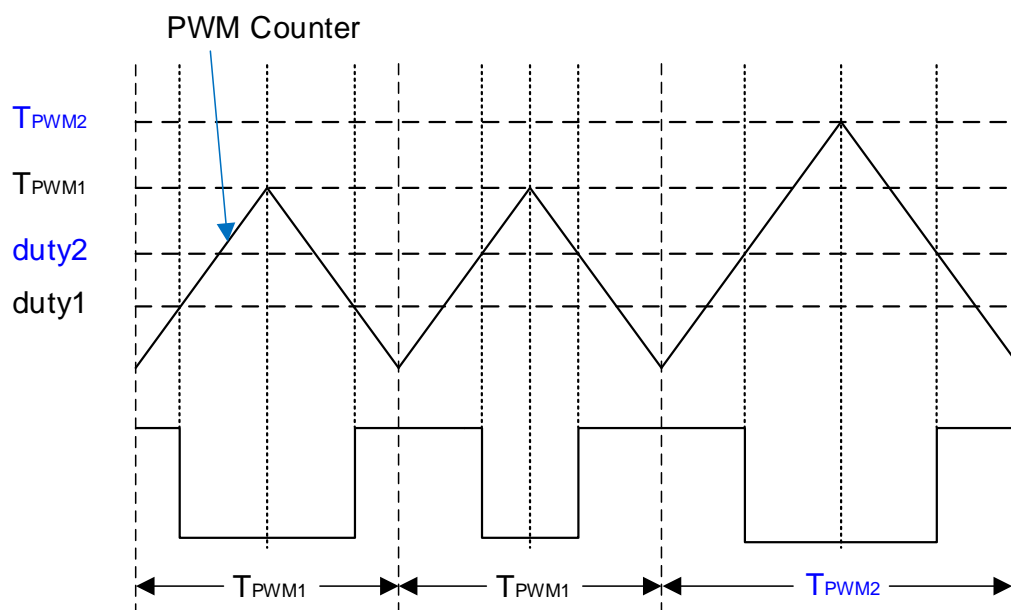
Calculation formula of center-aligned period T_{PWM} :

$$T_{pwm} = 2 * \frac{PWMPD[15: 0] + 1}{PWM \text{ Clock frequency}}$$

Center-aligned duty duty calculation formula:

$$duty = \frac{PDTx [15: 0]}{PWMPD[15: 0] + 1}$$

The center aligned waveform is as follows:



Center-aligned PWM

The above modes and types can be set through the PWMCON0 register:

PWMCON0 (D2H) PWM Control Register Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	ENPWM	PWMIF	PWMCK[1:0]		-	-	PWMM D[1:0]	ENPWM
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	ENPWM	PWM Module Switch control (Enable PWM) 1: Clock is allowed to enter the PWM unit, PWM is in working state, PWM output state is controlled by register ENPWMx (x=0~7)

		0: THE PWM unit stops working, the PWM counter is cleared to zero, and all PWM outputs are set to GPIO state
6	PWMIF	<p>PWM Interrupt Flag</p> <p>This bit is automatically set to 1 by the hardware when the PWM counter overflows (that is, when the count exceeds the PWMPD). If IE1[1] (EPWM) is also set to 1, the PWM interrupts. After the PWM interrupt occurs, the hardware does not automatically clear this bit, this bit must be cleared by the user's software.</p>
5~4	PWMCK[1:0]	<p>PWM Clock Source Selector</p> <p>00: fHRC</p> <p>01: fHRC / 2</p> <p>10: fHRC / 4</p> <p>11: fHRC / 8</p> <p>Note: the clock source frequency of PWM is fixed as fHRC = 32MH</p>
1~0	PWMMD[1:0]	<p>Set the PWM working mode</p> <p>0x: independent mode</p> <p>1x: complementary mode</p> <p>X0: edge alignment mode</p> <p>X1: Center alignment mod</p>
3~2	-	Reversed

PWMCFG (D1H) PWM Setting Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	INV7	INV6	INV5	INV4	INV3	INV2	INV1	INV0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

POR	0	0	0	0	0	0	0	0
-----	---	---	---	---	---	---	---	---

Bit number	Bit Mnemonic	Description
6	INVx (x=0~7)	PWMx waveform output reverse control 1: The PWMx waveform output is reversed 0: indicates that the PWMx waveform output is not reversed

PWMCON1 (D3H) PWM Control Register 1 (读/写)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	EPWM7	EPWM6	EPWM5	EPWM4	EPWM3	EPWM2	EPWM1	EPWM0
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	ENPWMx (x=0~7)	PWMx waveform output selection 0: PWMx output is turned off and used as GPIO port ¹ 1: When ENPWM is 1, the I/O where PWMx resides serves as the waveform output port

Note:

1. If ENPWM is set to 1, the PWM module is turned on, but ENPWMx=0, and the PWM output is turned off as a GPIO port. In this case, the PWM module can be used as a 16-bit Timer. When EPWM(IE1.1) is set to 1, the PWM will still interrupt

PWMPDL (D4H) PWM cycle register low 8 bits (read/write)

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

PWMPDH (D5H) PWM cycle register high 8 bits (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	PWMPDH[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	PWMPD[15:0]	Period setting for PWM sharing This value represents the PWM output waveform (period - 1); In other words, the cycle value of PWM output is (PWMPD[15:0] + 1) * PWM clock;

IE1 (A9H) INTERRUPT ENABLE REGISTER 1 (读/写)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	ET4	ET3	-	ETK	EINT2	EBTM	EPWM	ESSIO

R/W	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
POR	0	0	X	0	0	0	0	0

Bit number	Bit Mnemonic	Description
1	EPWM	PWM interrupt enable control 0: disables PWM interruption 1: allows interruption when PWM counter overflows

IP1 (B9H) INTERRUPT PRIORITY CONTROL REGISTER 1 (读/写)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	IPT4	IPT3	-	IPTK	IPINT2	IPBTM	IPPWM	IPSSI0
R/W	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
POR	0	0	X	0	0	0	0	0

Bit number	Bit Mnemonic	Description
1	IPPWM	PWM interrupt priority selection 0: Set interrupt priority of PWM to "low" 1: Set the INTERRUPT priority of PWM to "high"

13.2.2 PWM Fault Detection Function Setting

The fault detection function is often applied to the protection of motor systems. When the fault detection function is enabled, FLTEN1 (PWMFLT.7) is set to 1, and the fault detection signal input pin (FLT) becomes effective. When the signal of the FLT pin meets the fault condition, the flag bit FLTSTA1 is set by hardware, the PWM counter stops counting, and the PWM output stops. The fault detection mode is divided into latch mode and immediate mode; in immediate mode, when the fault signal on the FLT pin meets the disabling condition, the flag FLTSTA1 is cleared by hardware, and the PWM counter resumes counting until the PWM counter returns to zero. Output; In the latch mode, when the fault signal on the FLT pin meets the disabling condition, the status of the FLTSTA1 flag remains unchanged, and the user can clear it through software. Once the FLTSTA1 status is cleared, the PWM counter resumes counting until the PWM counter returns The PWM resumes output after zero. The fault detection mode is divided into latch mode and immediate mode. The specific configuration methods are as follows:

PWMFLT (D7H) PWM Fault Detection Setting Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	FLTEN1	FLTSTA1	FLTMD1	FLTLV1	-	-	FLTDT1[1: 0]	
R/W	R/W	R/W	R/W	R/W	-	-	R/W	R/W
POR	0	0	0	0	x	x	0	0

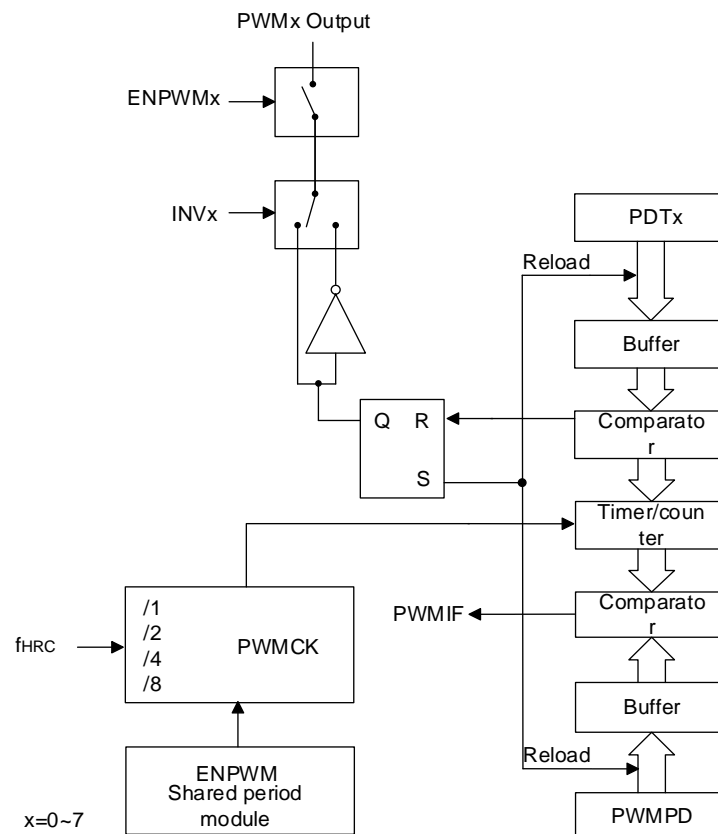
Bit number	Bit Mnemonic	Description
7	FLTEN1	PWM fault detection function control bit 0: failure detection function is off 1: The fault detection function is turned on
6	FLTSTA1	PWM fault detection status flag 0: PWM is in normal output state; 1: Fault detection is valid, the PWM output is in a high-impedance state, if in latch mode, this bit can be cleared by software
5	FLTMD1	PWM fault detection mode setting bit

		<p>0: Latch mode, when the fault input is valid, FLTSTA1 is set to "1", the PWM stops outputting, and the FLTSTA1 state remains unchanged when the fault input is invalid</p> <p>1: Immediate mode: When the fault input is valid, FLTSTA1 is set to "1" and the PWM stops outputting. When the fault input is invalid, the state of FLTSTA1 is cleared immediately, and the PWM waveform will resume output when the PWM time base counter returns to zero</p>
4	FLTLV1	<p>PWM fault detection level selection bit</p> <p>0: Low level of fault detection is effective</p> <p>1: High level of fault detection is effective</p>
1~0	FLTDT1[1: 0]	<p>PWM fault detection input signal filtering time setting</p> <p>00: filtering time is 0</p> <p>01: filtering time is 1us</p> <p>10: filter time is 4us</p> <p>11: The filtering time is 16us</p>
3~2	-	Reserved

13.3 PWM Independent Mode

In independent mode ($PWMMOD.1 = 0$), the duty cycle of 8 PWM channels can be set independently. The user configures the PWM output status and period, and then configures the duty cycle register of the corresponding PWM channel to output the PWM waveform at a fixed duty cycle.

13.3.1 PWM Independent Mode Block Diagram



SC95F852X PWM Independent mode block diagram

13.3.2 PWM Independent Mode Duty Cycle Configuration

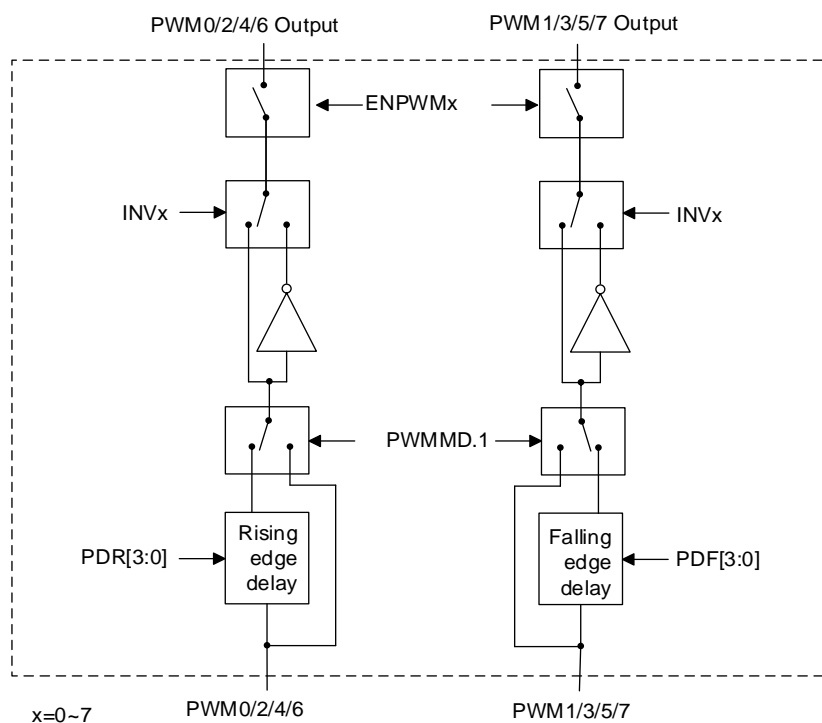
PWM Duty Cycle Adjustment Register PDTx (Read/Write)

Add	7	6	5	4	3	2	1	0	POR
1040H	PDT0[15:8]								00000000b
1041H	PDT0[7:0]								00000000b
1042H	PDT1[15:8]								00000000b
1043H	PDT1[7:0]								00000000b
1044H	PDT2[15:8]								00000000b
1045H	PDT2[7:0]								00000000b
1046H	PDT3[15:8]								00000000b
1047H	PDT3[7:0]								00000000b
1048H	PDT4[15:8]								00000000b
1049H	PDT4[7:0]								00000000b
104AH	PDT5[15:8]								00000000b
104BH	PDT5[7:0]								00000000b
104CH	PDT6[15:8]								00000000b
104DH	PDT6[7:0]								00000000b
104EH	PDT7[15:8]								00000000b
104FH	PDT7[7:0]								00000000b

Bit number	Bit Mnemonic	Description
3~0	PDTx [15:8] (x=0~7)	PWMx waveform duty cycle length setting The high level width of the PWMx waveform is (PDTx [15:0]) PWM clock

13.4 PWM Complementary Model

13.4.1 PWM Block Diagram of Complementary Mode



SC95F852X PWM block diagram of complementary mode

13.4.2 PWM Complementary Mode Duty Cycle Configuration

In complementary mode ($PWMMD[1:0] = 1x$), PWM0/PWM1, PWM2/PWM3, PWM4/PWM5 and PWM6/PWM7 are divided into four groups. Duty cycle was adjusted by PDT0[15:0], PDT2[15:0], PDT4[15:0] and PDT6[15:0] respectively.

In complementary mode, registers PDT0[15:0], PDT2[15:0], PDT4[15:0] and PDT5[15:0] are invalid.

PWM Duty Cycle Adjustment Register PDTx (Read/Write)

Add	7	6	5	4	3	2	1	0	POR
1040H	PDT0[15:8]								00000000b
1041H	PDT0[7:0]								00000000b

1042H	PDT1[15:8]	00000000b
1043H	PDT1[7:0]	00000000b
1044H	PDT2[15:8]	00000000b
1045H	PDT2[7:0]	00000000b
1046H	PDT3[15:8]	00000000b
1047H	PDT3[7:0]	00000000b
1048H	PDT4[15:8]	00000000b
1049H	PDT4[7:0]	00000000b
104AH	PDT5[15:8]	00000000b
104BH	PDT5[7:0]	00000000b
104CH	PDT6[15:8]	00000000b
104DH	PDT6[7:0]	00000000b
104EH	PDT7[15:8]	00000000b
104FH	PDT7[7:0]	00000000b

Bit number	Bit Mnemonic	Description
7~0	PDTx [7:0] (x=0,2,4,6)	PWMx and PWMy, y = x + 1 mouth PWM waveform duty cycle length setting Px and Py pin on the high level width of the PWM waveform is (PDTx [15:0]) PWM clock

13.4.3 PWM Complementary Mode Dead Time Setting

When the PWM of the SC95F852X works in complementary mode, the dead zone control module can prevent the effective time zones of the two PWM signals of complementary outputs from overlapping each other, so as to ensure that a pair of complementary power switch tubes driven by PWM signals will not be turned on at the same time. .

PWMDFR (D6H) PWM Dead Time Setting Register (read/write)

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

Bit number	Bit Mnemonic	Description
7~4	PDF[3: 0]	Complementary mode: PWM falling edge dead time= $4 \times \text{PDF}[3: 0] / f_{\text{HRC}}$
3~0	PDR[3: 0]	Complementary mode: PWM rising edge dead time= $4 \times \text{PDR}[3: 0] / f_{\text{HRC}}$

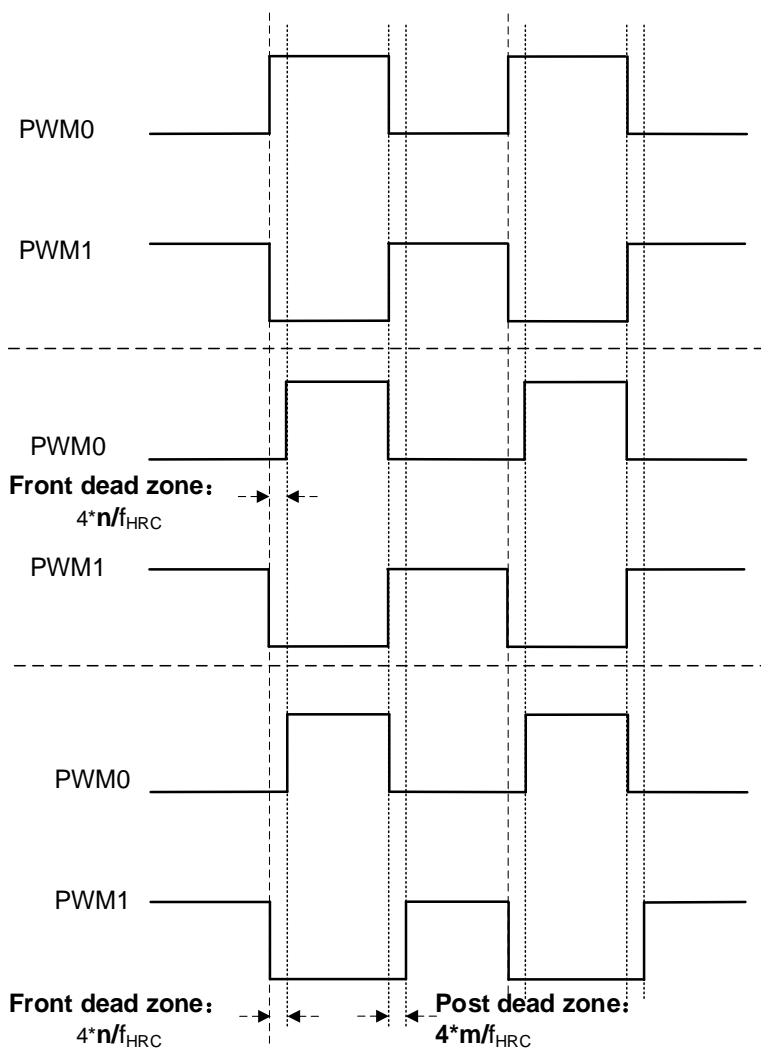
13.4.4 PWM Dead Zone Output Waveform

The following figure is based on the PWM0 and PWM1 in the complementary mode of the dead time adjustment waveform, in order to facilitate the distinction, PWM1 has reversed (INV1=1).

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

2.Set PWM0 rising edge
dead zone:
PWMMOD.1 = 1
PDF = 0
PDR = n

3.Set the falling edge dead
zone of PWM1:
PWMMOD.1 = 1
PDF = m
PDR = n
Note: PWM1 has been
reversed at this time, the
corresponding control of PDF
is actually the rising edge
dead zone delay time of
PWM1 output port waveform



PWM dead zone output waveform

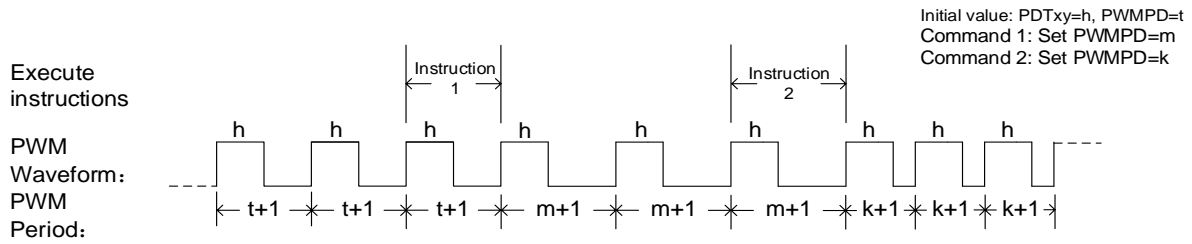
13.5 PWM Waveforms and Directions

The effect of changing SFR parameters on the PWM waveform is as follows:

① Duty cycle change characteristics

When the PWMn outputs a waveform, if the duty cycle needs to be changed, it can be achieved by changing the value of the high-level setting register (PDTx). But need to pay attention: change the value of PDTx, the duty ratio will not change immediately, but wait for the end of this cycle and change it in the next cycle.

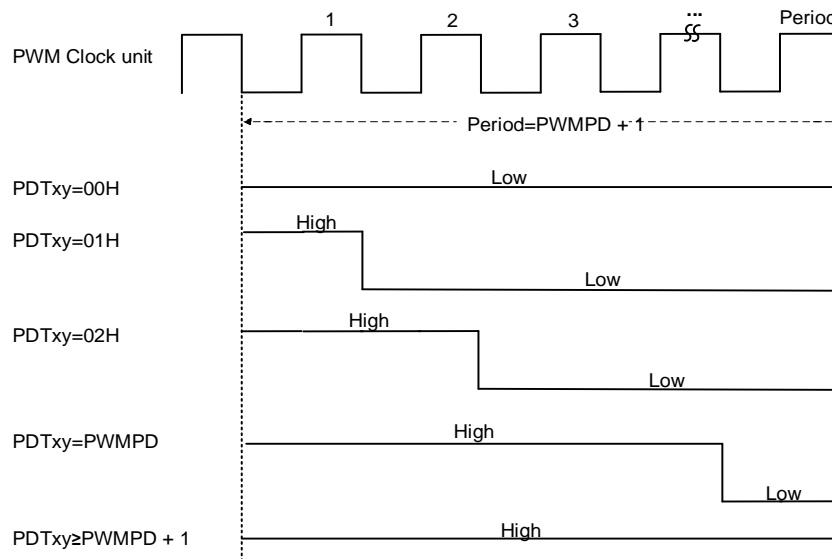
② Periodic change characteristics



Periodic change characteristic diagram

When the PWM outputs a waveform, if the period needs to be changed, it can be achieved by changing the value of the period setting register PWMPD. Change the value of PWMPD, the cycle will not change immediately, but wait for the end of the cycle, and change in the next cycle, refer to the figure above.

③ Relationship between period and duty cycle



Relationship between cycle and duty cycle

The relationship between period and duty cycle is shown in the figure above. The premise of this result is that the PWM output inverse control (INVx) is initially 0. If you want to get the opposite result, you can set INVx to 1.

14 General-purpose I/O (GPIO)

The SC95F852X provides up to 26 bidirectional GPIO ports that can be controlled. The input and output control registers are used to control the input and output status of each port. When the port is used as an input, each I/O port has an internal pull-up resistor controlled by PxPHY. The 26 I/Os are multiplexed with other functions. When the I/O port is in the input or output state, the actual state value of the port is read from the port data register.

Note: The unused and unlead IO ports should be set to strong push-pull output mode.

14.1 GPIO Structure Diagram

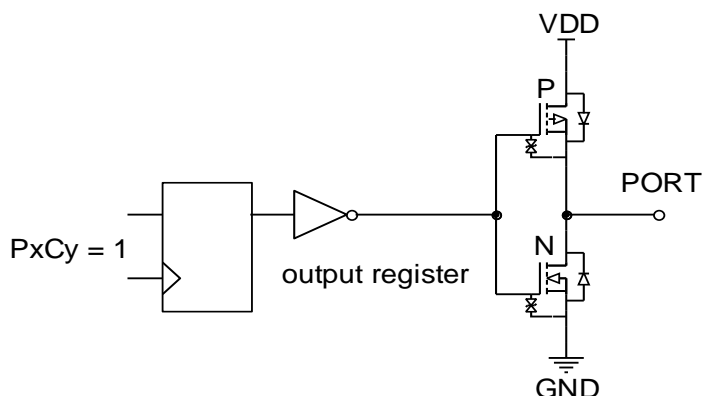
Strong Push-pull Output Mode

In the strong push-pull output mode, it can provide continuous high-current drive:

Except for P05/P20, the I/o driver has high output if the output is greater than 10mA, and low output if the output is greater than 50mA.

The driving capacity of P05/P20 is: the output greater than 20mA is high, and the output greater than 50mA is low.

The schematic diagram of the port structure of the strong push-pull output mode is as follows:

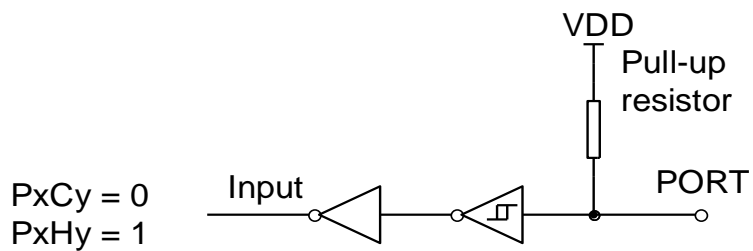


Strong push-pull output mode

Pull-up Input Mode

In the pull-up input mode, a pull-up resistor is constantly connected to the input port. Only when the input port is pulled low, the low-level signal is detected.

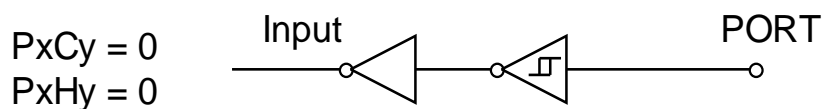
The schematic diagram of the port structure with pull-up input mode is as follows:



Input mode with pull-up resistor

High Impedance Input Mode (Input only)

The schematic diagram of the port structure of the high impedance input mode is as follows:



High impedance input mode

14.2 I/O Port-related Registers

P0CON (9AH) P0 Port Input/Output Control Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	P0C7	P0C6	P0C5	P0C4	P0C3	P0C2	P0C1	P0C0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

P0PH (9BH) P0 Port pull-up Resistor Control Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	P0H7	P0H6	P0H5	P0H4	P0H3	P0H2	P0H1	P0H0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

P1CON (91H) P1 Port Input/Output Control Register (read/write)

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

P1PH (92H) P1 Port Pull-up Resistor Control Register (read/write)

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

P2CON (A1H) P2 Port Input/output Control Register (read/write)

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

P2PH (A2H) P2 Port Pull-up Resistor Control Register (read/write)

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

P5CON (D9H) P5 Port Input/output Control Register (read/write)

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

P5PH (DAH) P5 Port Pull-up Resistor Control Register (read/write)

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

Bit number	Bit Mnemonic	Description
7~0	PxCy (x=0~5, y=0~7)	Px port input and output control: 0: Pxy is the input mode (initial value at power-on) 1: Pxy is a strong push-pull output mode
7~0	PxHy (x=0~5, y=0~7)	The Px port pull-up resistor setting is only valid when PxCy=0: 0: Pxy is the high-impedance input mode (initial value at power-up), and the pull-up resistor is turned off; 1: Pxy pull-up resistor is on

P0 (80H) P0 Port Data Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

P1 (90H) P1 Port Data Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

P2 (A0H) P2 Port Data Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

P5 (D8H) P5 Port Data Register (read/write)

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

Bit number	Bit Mnemonic	Description
7~0	P0.x (x=0~7)	P0 port latch register data
7~0	P1.x (x=0~7)	P1 port latch register data
7~0	P2.x (x=0~7)	P2 port latch register data
5~0	P5.x (x=0~5)	P5 port latch register data

IOHCON0 (96H) IOH Setting Register 0 (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	P1H[1: 0]		P1L[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	P1H[1: 0]	P1 high four IOH settings 00: Set P1 high four IOH level 0 (maximum); 01: Set P1 high four IOH level 1; 10: Set P1 high four IOH level 2; 11: Set P1 high four IOH level 3 (minimum);
5~4	P1L[1: 0]	P1 low four IOH settings 00: Set P1 low four IOH level 0 (maximum); 01: Set P1 low four IOH level 1; 10: Set P1 low four IOH level 2; 11: Set P1 low four IOH level 3 (minimum);
3~2	P0H[1: 0]	P0 high four IOH settings 00: Set P0 high four IOH level 0 (maximum); 01: Set P0 high four IOH level 1; 10: Set P0 high four IOH level 2; 11: Set P0 high four IOH level 3 (minimum);
1~0	P0L[1: 0]	P0 low four IOH settings 00: Set P0 low four IOH level 0 (maximum); 01: Set P0 low four IOH level 1; 10: Set P0 low four IOH level 2; 11: Set P0 low four IOH level 3 (minimum);

IOHCON1 (97H) IOH Setting Register 1 (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	-	-	P5L[1: 0]		P2H[1: 0]		P2L[1: 0]	
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
POR	x	x	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description
5~4	P5L[1: 0]	P5 low four IOH settings 00: set P5 low four IOH level 0 (maximum); 01: Set P5 low four IOH level 1; 10: Set P5 low four IOH level 2; 11: Set P5 low four IOH level 3 (minimum);
3~2	P2H[1: 0]	P2 high four IOH settings 00: Set P2 high four IOH level 0 (maximum); 01: Set P2 high four IOH level 1; 10: Set P2 high four IOH level 2; 11: Set P2 high four IOH level 3 (minimum);
1~0	P2L[1: 0]	P2 low four IOH settings 00: Set P2 low four IOH level 0 (maximum); 01: Set P2 low four IOH level 1; 10: Set P2 low four IOH level 2; 11: Set P2 low four IOH level 3 (minimum);
7~6	-	Reserved

15 UART0

The SC95F852X supports a full-duplex serial port, which can be conveniently used for connection with other devices or equipment, such as Wifi module circuit or other UART communication interface driver chip. The functions and features of UART0 are as follows:

1. Three communication modes are available: Mode 0, Mode 1 and Mode 3;
2. Can choose Timer 1 or Timer 2 as the baud rate generator;
3. Interrupt RI/TI can be generated after transmission and reception are completed, and the interrupt flag needs to be cleared by software.

SCON (98H) Serial Control Register (read/write)

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

Bit number	Bit Mnemonic	Description
7~6	SM0~1	Serial communication mode control bit 00: Mode 0, 8-bit half-duplex synchronous communication mode, serial data is sent and received on the RX pin. The TX pin is used as the transmit shift clock. 8 bits are sent and received per frame, the low bit is received or sent first; 01: Mode 1, 10-bit full-duplex asynchronous communication, consisting of 1 start bit, 8 data bits and 1 stop bit, the communication baud rate is variable; 10: reserved; 11: Mode 3, 11-bit full-duplex asynchronous communication, consisting of 1 start bit, 8 data bits, a programmable 9th bit, and 1 stop bit. The communication baud rate is variable.
5	SM2	Serial communication mode control bit 2, this control bit is only valid for mode 3

		<p>0: set RI to generate an interrupt request every time a complete data frame is received;</p> <p>1: When a complete data frame is received, RI will be set to generate an interrupt request only when RB8=1.</p> <p>The baud rate override setting bit is only valid in mode 0 (SM0~1 = 00):</p> <p>0: The serial port runs at 1/12 of the system clock</p> <p>1: The serial port runs at 1/4 of the system clock</p>
4	REN	<p>Receive enable control bit</p> <p>0: data reception is not allowed;</p> <p>1: Allow receiving data.</p>
3	TB8	Only valid for mode 3, which is the 9th bit of the transmitted data
2	RB8	Only valid for mode 3, the 9th bit of the received data
1	TI	Transmit interrupt flag
0	RI	Receive interrupt flag

SBUF (99H) Serial Data Buffer 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 register SBUF contains two registers: a transmit shift register and a receive latch. The data written to SBUF will be sent to the transmit shift register and start the transmission process. Reading SBUF will return the contents of the receive latch.
-----	-------------------	--

15.1 Baud Rate of Serial Communication

In mode 0, the baud rate can be programmed to 1/12 or 1/4 of the system clock:

1. SM2=0, the serial port runs at 1/12 of the system clock;
2. SM2=1, the serial port runs at 1/4 of the system clock.

In Mode 1 and Mode 3, the baud rate can be selected from Timer 1 or Timer 2 overflow rate.

Set the TCLKX (TXCON.4) and RCLKX (TXCON.5) bits to 1 to select Timer 2 as the baud clock source for TX and RX (see the timer section for details). Regardless of whether TCLKX or RCLKX is logic 1, Timer 2 is a baud rate generator. If TCLK and RCLK are logic 0, Timer 1 serves as the baud clock source for Tx and Rx.

The baud rate formulas for Mode 1 and Mode 3 are shown below, where [TH1, TL1] is the 16-bit counter register of Timer 1, and [RCAPXH, RCAPXL] is the 16-bit reload register of Timer 2.

1. Use Timer 1 as the baud rate generator. Timer 1 must stop counting, that is, TR1=0:

$$\text{BaudRate} = \frac{f_{\text{sys}}}{[\text{TH1}, \text{TL1}]}; \text{ (Note: [TH1, TL1] must be bigger than 0x0010)}$$

2. Use Timer 2 as the baud rate generator:

$$\text{BaudRate} = \frac{f_{\text{sys}}}{[\text{RCAPXH}, \text{RCAPXL}]}; \text{ (Note: [RCAPXH, RCAPXL] must be bigger than 0x0010)}$$

16 SPI/TWI/UART Serial Interface (USCI0/1/2)

Symbol	Address	Description	7	6	5	4	3	2	1	0	POR
US0CON0	95H	USCI0 control register 0	US0CON0[7: 0]								00000000 0b
US0CON1	9DH	USCI0 control register 1	US0CON1[7: 0]								00000000 0b
US0CON2	9EH	USCI0 control register 2	US0CON2[7: 0]								00000000 0b
US0CON3	9FH	USCI0 control register 3	US0CON3[7: 0]								00000000 0b
US1CON0	A4H	USCI1 control register 0	US1CON0[7: 0]								00000000 0b
US1CON1	A5H	USCI1 control register 1	US1CON1[7: 0]								00000000 0b
US1CON2	A6H	USCI1 control register 2	US1CON2[7: 0]								00000000 0b
US1CON3	A7H	USCI1 control register 3	US1CON3[7: 0]								00000000 0b
US2CON0	C4H	USCI2 control register 0	US2CON0[7: 0]								00000000 0b
US2CON1	C5H	USCI2 control register 1	US2CON1[7: 0]								00000000 0b
US2CON2	C6H	USCI2 control register 2	US2CON2[7: 0]								00000000 0b
US2CON3	C7H	USCI2 control register 3	US2CON3[7: 0]								00000000 0b

The SC95F852X internally integrates three three-select one universal serial circuits interface (referred to as USCI), which can facilitate the connection between MCU and devices or equipment with different interfaces. The user

can configure the USCI interface to any one of SPI, TWI and UART through the USMD1[1: 0], USMD0[1: 0] bits of the configuration register OTCON, or the USMD2[1: 0] bits of TMCON. Its characteristics are as follows:

1. SPI mode can be configured as one of master mode or slave mode, with 8-bit or 16-bit transmission mode
2. TWI mode communication can be configured as master mode or slave mode
3. There are three UART modes:
 - ① Mode 0: 8-bit half-duplex synchronous communication
 - ② Mode 1: 10-bit full-duplex asynchronous communication
 - ③ Mode 3: 11-bit full-duplex asynchronous communication

The specific configuration method is as follows:

OTCON (8FH) Output Control Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	USMD1[1: 0]		USMD0[1: 0]		VOIRS[1: 0]		SCS	BIAS
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	USMD1[1: 0]	USCI1 Communication mode control bit 00: USCI1 close 01: USCI1 Set to SPI communication mode; 10: USCI1 Set to TWI communication mode; 11: USCI1 Set to UART communication mode;
5~4	USMD0[1: 0]	USCI0 Communication mode control bit 00: USCI0 Set to SC communication mode;

		01: USC10 Set to SPI communication mode; 10: USC10 Set to TWI communication mode; 11: USC10 Set to UART communication mode;
--	--	---

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

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	USMD2[1: 0]		-	-	-	-	T1FD	T0FD
R/W	R/W	R/W	-	-	-	-	R/W	R/W
POR	0	0	x	x	x	x	0	0

Bit number	Bit Mnemonic	Description
7~6	USMD2[1: 0]	USC12 Communication mode control bit 00: USC12 close 01: USC12 set to SPI communication mode; 10: USC12 set to TWI communication mode; 11: USC12 set to UART communication mode;

Note:

A USCI interface can be set to different communication modes via the USMD, each of which has a corresponding operation register set. The control register groups in different communication modes share the same mapped address, but the operations among the groups are independent. Setting the control register in one communication mode does not affect the values in the register groups in other communication modes.

Such as:

- At the same time, US0CON0 (95H) = 0x80H is set in SPI communication mode.
- Anyway, USMD1 =11, USC10 being the UART communication interface, is set to US0CON0 (95H) = 0x0FH.

- Anyway, USMD1 is set to 01, USCIO is set back to the SPI communication interface, and US0CON0 (95H) is read in this mode, which reads 0x80H.
- Anyway, USMD1 is set to 11, and USCIO is set back to the UART communication interface. US0CON0 (95H) is read in this mode, which reads 0x0FH.

16.1 SPI

Serial Peripheral Device Interface (SPI for short) is a high-speed serial communication interface that allows the MCU to perform full-duplex, synchronous serial communication with peripheral devices (including other MCUs).

The three SPI interfaces of SC95F852X, SPI0/1/2, can be set to master mode or slave mode. SPI1 and SPI2 are the conventional SPI communication interfaces. SPI0 adds an 8-level transceiver FIFO in the main mode based on the conventional SPI communication interface.

16.1.1 SPI0

USMD0[1:0] = 01, one of three serial interfaces USCIO is configured as SPI interface, namely SPI0:

- USTX0 as MOSI signal
- USRX0 as MISO signal
- USCK0 as CLK signal

SPI0 can be set as the main mode or the slave mode. In the main mode, SPI0 has 16-bit 8-level FIFO cache for receiving and sending. Users can achieve in the main mode of SPI0:

- The SPI0 data cache SPD continuously writes 8 or less bits of 16-bit data. The first data written is also sent into the sending FIFO first, and then sent out in turn. When THE FIFO launched the last data, send status flag bit TXE set up, on behalf of send FIFO is empty; If the sending FIFO is full, the writing conflict flag bit WCOL is set, and the data written to SPD is invalid until the data in THE FIFO is sent out and the FIFO is not satisfied, the data of SPD can be sent to FIFO again. When all the data in FIFO is sent, SPIF is set as the signal of data transmission completion in SPI0.
- A queue reads eight or less 16-bit data continuously from the data cache SPD, allowing the first data to be received to be read first.

Note the following two points when using the grade 8 FIFO of SPI0:

1. SPI0 only has 16-bit grade 8 FIFO cache in host mode, not in slave mode;
2. When the enable control bit of SPI0 SPEN=0, grade 8 FIFO will be cleared to zero. At this time, it is invalid for users to write data into FIFO. Therefore, sending data to the FIFO is placed after SPEN position 1.

The FIFO function of SPI0 compares with the normal SPI interface as follows:

<div>Contrast</div> <div>BIT</div>	SPI0	SPI1~2
------------------------------------	------	--------

TXE	FIFO out the last data, send FIFO is empty, set 1	no such flag bit
WCOL	If the FIFO is full and WCOL is set to 1, the data written to the SPD cannot be sent into the FIFO	When a frame is being sent, data is written to the buffer SPD, WCOL is set to 1, and the data written to the SPD is invalid
SPIF	FIFO launched the last data has been sent, by the hardware to set this interrupt flag	After sending, the hardware sets the interrupt flag bit

16.1.1.1 SPI0 Operation-dependent register

US0CON0 (95H) USCIO control register 0 (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: close the SPI 1: open the SPI
5	MSTR	SPI master/slave select bit 0: SPI is the slave device 1: SPI is the master device
4	CPOL	Clock polarity control bit 0: SCK is low in idle state 1: SCK is high in idle state
3	CPHA	Clock phase control bit 0: Data is collected at the first edge of the SCK cycle 1: Data is collected at the second edge of the SCK cycle
2~0	SPR[2:0]	SPI clock rate selection bit 000: f_{sys} 001: $f_{sys}/2$ 010: $f_{sys}/4$ 011: $f_{sys}/8$ 100: $f_{sys}/16$ 101: $f_{sys}/32$ 110: $f_{sys}/64$ 111: $f_{sys}/128$
1	-	Reversed

US0CON1 (9DH) SPI0 状态寄存器(读/写)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	SPIF	WCOL	-	-	TXE	DORD	SPMD	TBIE
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	SPIF	SPI data transmission flag 0: Cleared by software 1: Indicates that data transmission has been completed, set by hardware
6	WCOL	Write conflict flag 0: Cleared by software, indicating that the write conflict has been processed 1: When the FIFO is full, WCOL is set to 1 by the hardware
3	TXE	Send status flag bit 0: Send FIFO not empty 1: Send FIFO is empty, TXE set, must be cleared by the software.
2	DORD	Transmission direction selection bit 0: MSB first sent 1: LSB first sent
1	SPMD	SPI transmission mode selection: 0: 8-bit mode 1: 16-bit mode
0	TBIE	Send cache interrupt to allow control bit 0: Send interrupts are not allowed 1: Allow sending interrupts: WHEN the interrupt flag SPIF=1, TBIE=1 will generate an SPI interrupt
5~4	-	Reversed

SPDL
US0CON2 (9EH) SPI0 Data register low byte (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 buffer register low byte (8/16 bit mode) Low byte of data written to data register SPD Read data low byte of data register SPD

SPDH
US0CON3 (9FH) SPI0 Data register high byte (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	SPD[15:8]							
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[15:8]	SPI data buffer register high byte (only for 16-bit mode) High byte of data written to data register SPD Read data high byte of data register SPD Note: 1. When the SPI is set to 16-bit mode, high bytes must be written first, then low bytes. After the low bytes are written, data is immediately sent to the FIFO. 2. When SPI0 is used as the host, it continuously writes 8 or less 16-bit transmitting data to SPD, and the data written first is also sent first. Continuously read 8 or less from the data cache 16-bit received data, the first received data is also the first to be read;

16.1.1.2 SPI0 Signal Description

Master-Out/Slave-In (MOSI):

This signal connects the master device and a slave device. Data is serially transmitted from the master device to the slave device through MOSI, the master device outputs, and the slave device inputs.

Master-In and Slave-Out (MISO):

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

SPI Serial Clock (SCK):

The SCK signal is used to control the synchronous movement of input and output data on the MOSI and MISO lines. A byte is transmitted on the wire every 8 clock cycles. If the slave is not selected, the SCK signal is ignored by the slave.

16.1.1.3 SPI0 Operating Modes

SPI can be configured as one of master mode or slave mode. The configuration and initialization of the SPI module are completed by setting the SPI control register USnCON0 (n=0~2) and the SPI status register USnCON1. After the configuration is completed, the data transfer is completed by setting the SPI data registers USnCON2, USnCON3 (hereinafter referred to as SPD).

During SPI communication, data is shifted in and out serially synchronously. The serial clock line (SCK) keeps the movement and sampling of data on the two serial data lines (MOSI and MISO) synchronized. If the slave is not selected, it cannot participate in activities on the SPI bus.

When the SPI master device transmits data to the slave device through the MOSI line, the slave device sends the data to the master device as a response via the MISO line, which realizes the synchronous full-duplex transmission of data sending and receiving under the same clock. The sending shift register and the receiving shift register use the same special function address. Writing to the SPI data register SPD will write to the sending shift register, and reading the SPD will get the data of the receiving shift register.

The SPI interface of some devices will lead to the SS pin (slave device selection pin, active low). When communicating with the SPI of the SC95F852X, the connection mode of the SS pin of other devices on the SPI bus needs to be connected according to different communication modes. The following table lists the connection modes of the SS pin of other devices on the SPI bus in different SPI communication modes of the SC95F852X:

SC95F852X SPI	Other devices on the SPI bus	Mode	Slave SS (Slave selection pin)
Master mode	Slave mode	One master and one slave	Pull down
		One master and multiple slaves	The SC95F852X leads to multiple I/Os, which are connected to the SS pin of the slave. Before data transmission, the SS pin of the slave device must be set low
Slave mode	Master mode	One master and one slave	Pull up

SPI0 Master Mode

- **Mode Startup:**

The SPI master device controls the start of all data transfers on the SPI bus. When the MSTR bit in the SPI control register USnCON0 is set to 1, the SPI runs in the master mode and only one master device can start the transfer.

- **Transmitting:**

In SPI master mode, perform the following operations on SPD: write a byte of data to SPDL in 8-bit mode or write the high byte to SPDH first, and then write the low byte to SPDL in 16-bit mode, the data will be written to the transmit shift buffer. If there is already a data in the transmit shift register, the main SPI generates a WCOL signal to indicate that the write is too fast. But the data in the transmission shift register will not be affected, and the transmission will not be interrupted. In addition, if the transmission shift register is empty, the master device immediately shifts the data in the transmission shift register to the MOSI line in accordance with the SPI clock frequency on SCK. When the transfer is complete, the SPIF bit in the SPI status register USnCON1 is set to 1. If the SPI interrupt is enabled, an interrupt will also be generated when the SPIF bit is set.

- **Receiving:**

When the master device transmits data to the slave device through the MOSI line, the corresponding slave device also transmits the contents of its transmitting shift register to the receiving shift register of the master device through the MISO line, realizing full-duplex operation. Therefore, the SPIF flag position 1 means that the transmission is complete and the data is received. The data received by the slave device is stored in the receive shift register of the master device according to the MSB or LSB first transmission direction. When a byte of data is completely moved into the receive register, the processor can obtain the data by reading the SPD.

SPI0 Slave mode

- **Mode Startup:**

When the MSTR bit in the SPI control register USnCON0 register is cleared to 0, SPI runs in slave mode.

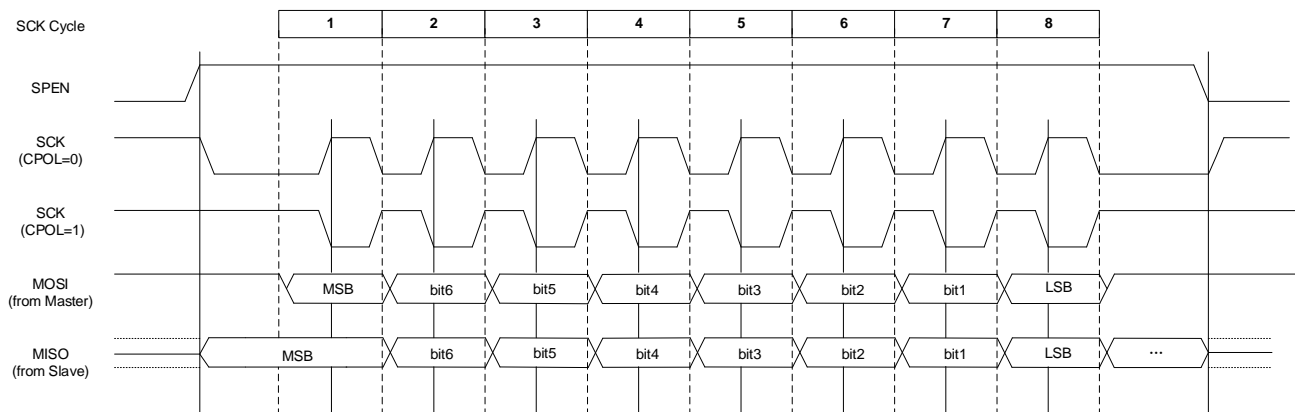
● Transmitting and Receiving :

In slave mode, according to the SCK signal controlled by the master device, data is shifted in through the MOSI pin, and the MISO pin is shifted out. A bit counter records the number of edges of SCK. When the receiving shift register shifts in 8-bit data (one byte) and the sending shift register shifts out 8-bit data (one byte), the SPIF flag bit is set to 1. The data can be obtained by reading the SPD register. If the SPI interrupt is enabled, an interrupt will also be generated when SPIF is set. At this time, the receiving shift register keeps the original data and the SPIF bit is 1, so that the SPI slave device will not receive any data until SPIF is cleared. The SPI slave device must write the data to be transmitted into the transmit shift register before the master device starts a new data transmission. If no data is written before starting to send, the slave device will transmit the "0x00" byte to the master device. If the SPD write operation occurs during the transfer, the WCOL flag of the SPI slave device is set to 1, that is, if the transfer shift register already contains data, the WCOL bit of the SPI slave device is set to 1, indicating that the write SPD conflicts. But the data of the shift register is not affected, and the transmission will not be interrupted.

transfer Form

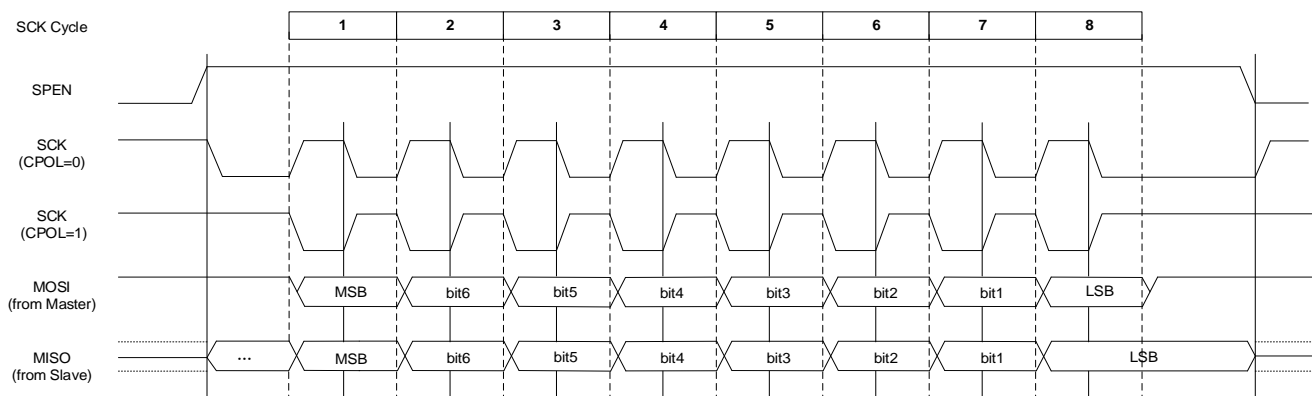
By software setting the CPOL bit and CPHA bit of the SPI control register USnCON0, the user can select four combinations of SPI clock polarity and phase. The CPOL bit defines the polarity of the clock, that is, the level state when idle, and it has little effect on the SPI transmission format. The CPHA bit defines the phase of the clock, that is, defines the clock edge that allows data sampling and shifting. In the two devices of master-slave communication, the setting of the clock polarity phase should be the same.

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



CPHA = 0 Data transfer diagram

When CPHA = 1, the master device outputs data to the MOSI line on the first edge of SCK, the slave device uses the first edge of SCK as the start signal, and the second edge of SCK starts to capture data, so the user must The operation of writing SPD is completed within two edges of one SCK. This form of data transmission is the preferred form of communication between a master device and a slave device.



CPHA = 1 Data transfer diagram

Error Detection

Writing to SPD during the data transmission sequence will cause a write conflict, and the WCOL bit in the SPI status register USnCON1 is set to 1. WCOL bit 1 will not cause interruption, and transmission will not be aborted. The WCOL bit needs to be cleared by software.

16.1.2 SPI1/2

USMDn[1:0] = 01 (n=1,2) serial interface USCI is configured as SPI interface, that is, spi1/2:

- Static USTXn acts as an MOSI signal
- Lent USRXn acts as a MISO signal
- Queue USCKn acts as a CLK signal

Spi1/2 can be set to master mode or slave mode.

16.1.2.1 SPI1/2 Operation-dependent register

US1CON0 (A4H) SPI1 control register (read/write)

US2CON0 (C4H) SPI2 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: close the SPI 1: open the SPI
5	MSTR	SPI master/slave select bit 0: SPI is the slave device 1: SPI is the master device
4	CPOL	Clock polarity control bit 0: SCK is low in idle state 1: SCK is high in idle state
3	CPHA	Clock phase control bit 0: Data is collected at the first edge of the SCK cycle 1: Data is collected at the second edge of the SCK cycle
2~0	SPR[2:0]	SPI clock rate selection bit 000: f _{sys} 001: f _{sys} /2 010: f _{sys} /4 011: f _{sys} /8 100: f _{sys} /16 101: f _{sys} /32 110: f _{sys} /64 111: f _{sys} /128
1	-	Reversed

US1CON1 (A5H) SPI1 Status Register (read/write)

US2CON1 (C5H) SPI2 Status Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	SPIF	WCOL	-	-	-	DORD	SPMD	TBIE
R/W	R/W	R/W	-	-	-	R/W	R/W	R/W
POR	0	0	x	x	x	0	0	0

Bit number	Bit Mnemonic	Description
7	SPIF	SPI data transmission flag 0: Cleared by software 1: Indicates that data transmission has been completed, set by hardware
6	WCOL	Write conflict flag 0: Cleared by software, indicating that the write conflict has been processed 1: When the FIFO is full, WCOL is set to 1 by the hardware
2	DORD	Transmission direction selection bit 0: MSB first sent 1: LSB first sent
1	SPMD	SPI transmission mode selection: 0: 8-bit mode 1: 16-bit mode
0	TBIE	Send cache interrupt to allow control bit 0: Send interrupts are not allowed 1: Allow sending interrupts: WHEN the interrupt flag SPIF=1, TBIE=1 will generate an SPI interrupt
5~4	-	Reversed

SPDL

US1CON2 (A6H) SPI1 Data register low byte (read/write)

US2CON2 (C6H) SPI2 Data register low byte (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 buffer register low byte (8/16 bit mode) Low byte of data written to data register SPD Read data low byte of data register SPD

SPDH

US1CON3 (A7H) SPI1 Data register high byte (read/write)

US2CON3 (C7H) SPI2 Data register high byte (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	SPD[15:8]							
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[15:8]	SPI data buffer register high byte (only for 16-bit mode) High byte of data written to data register SPD Read data high byte of data register SPD Note: 1. When the SPI is set to 16-bit mode, high bytes must be written first, then low bytes. After the low bytes are written, data is immediately sent to the FIFO. 2. When SPI0 is used as the host, it continuously writes 8 or less 16-bit transmitting data to SPD, and the data written first is also sent first. Continuously read 8 or less from the data cache 16-bit received data, the first received data is also the first to be read;

16.1.2.2 SPI1/2 Signal Description

Master-Out/Slave-In (MOSI):

This signal connects the master device and a slave device. Data is serially transmitted from the master device to the slave device through MOSI, the master device outputs, and the slave device inputs.

Master-In and Slave-Out (MISO):

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

SPI Serial Clock (SCK):

The SCK signal is used to control the synchronous movement of input and output data on the MOSI and MISO lines. A byte is transmitted on the wire every 8 clock cycles. If the slave is not selected, the SCK signal is ignored by the slave.

16.1.2.3 SPI1/2 Operating Modes

SPI can be configured as one of master mode or slave mode. The configuration and initialization of the SPI module are completed by setting the SPI control register USnCON0 (n=0~2) and the SPI status register USnCON1. After the configuration is completed, the data transfer is completed by setting the SPI data registers USnCON2, USnCON3 (hereinafter referred to as SPD).

During SPI communication, data is shifted in and out serially synchronously. The serial clock line (SCK) keeps the movement and sampling of data on the two serial data lines (MOSI and MISO) synchronized. If the slave is not selected, it cannot participate in activities on the SPI bus.

When the SPI master device transmits data to the slave device through the MOSI line, the slave device sends the data to the master device as a response via the MISO line, which realizes the synchronous full-duplex transmission of data sending and receiving under the same clock. The sending shift register and the receiving shift register use the same special function address. Writing to the SPI data register SPD will write to the sending shift register, and reading the SPD will get the data of the receiving shift register.

The SPI interface of some devices will lead to the SS pin (slave device selection pin, active low). When communicating with the SPI of the SC95F852X, the connection mode of the SS pin of other devices on the SPI bus needs to be connected according to different communication modes. The following table lists the connection modes of the SS pin of other devices on the SPI bus in different SPI communication modes of the SC95F852X:

SC95F852X SPI	Other devices on the SPI bus	Mode	Slave SS (Slave selection pin)
Master mode	Slave mode	One master and one slave	Pull down
		One master and multiple slaves	The SC95F852X leads to multiple I/Os, which are connected to the SS pin of the slave. Before data transmission, the SS pin of the slave device must be set low
Slave mode	Master mode	One master and one slave	Pull up

SPI1/2 Master Mode

- **Mode Startup:**

The SPI master device controls the start of all data transfers on the SPI bus. When the MSTR bit in the SPI control register USnCON0 is set to 1, the SPI runs in the master mode and only one master device can start the transfer.

- **Transmitting:**

In SPI master mode, perform the following operations on SPD: write a byte of data to SPDL in 8-bit mode or write the high byte to SPDH first, and then write the low byte to SPDL in 16-bit mode, the data will be written to the transmit shift buffer. If there is already a data in the transmit shift register, the main SPI generates a WCOL signal to indicate that the write is too fast. But the data in the transmission shift register will not be affected, and the transmission will not be interrupted. In addition, if the transmission shift register is empty, the master device immediately shifts the data in the transmission shift register to the MOSI line in accordance with the SPI clock frequency on SCK. When the transfer is complete, the SPIF bit in the SPI status register USnCON1 is set to 1. If the SPI interrupt is enabled, an interrupt will also be generated when the SPIF bit is set.

- **Receiving:**

When the master device transmits data to the slave device through the MOSI line, the corresponding slave device also transmits the contents of its transmitting shift register to the receiving shift register of the master device through the MISO line, realizing full-duplex operation. Therefore, the SPIF flag position 1 means that the transmission is complete and the data is received. The data received by the slave device is stored in the receive shift register of the master device according to the MSB or LSB first transmission direction. When a byte of data is completely moved into the receive register, the processor can obtain the data by reading the SPD.

SPI1/2 Slave mode

- **Mode Startup:**

When the MSTR bit in the SPI control register USnCON0 register is cleared to 0, SPI runs in slave mode.

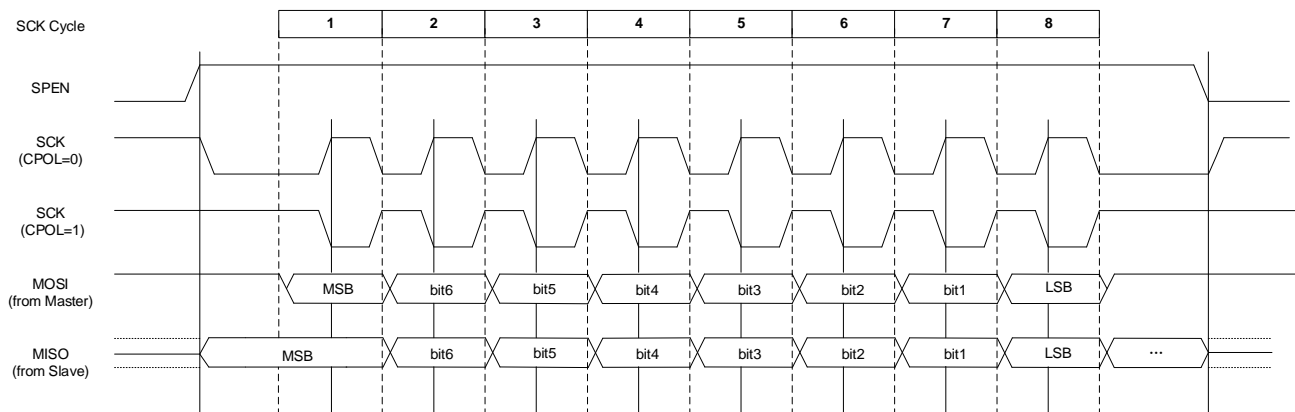
● Transmitting and Receiving :

In slave mode, according to the SCK signal controlled by the master device, data is shifted in through the MOSI pin, and the MISO pin is shifted out. A bit counter records the number of edges of SCK. When the receiving shift register shifts in 8-bit data (one byte) and the sending shift register shifts out 8-bit data (one byte), the SPIF flag bit is set to 1. The data can be obtained by reading the SPD register. If the SPI interrupt is enabled, an interrupt will also be generated when SPIF is set. At this time, the receiving shift register keeps the original data and the SPIF bit is 1, so that the SPI slave device will not receive any data until SPIF is cleared. The SPI slave device must write the data to be transmitted into the transmit shift register before the master device starts a new data transmission. If no data is written before starting to send, the slave device will transmit the "0x00" byte to the master device. If the SPD write operation occurs during the transfer, the WCOL flag of the SPI slave device is set to 1, that is, if the transfer shift register already contains data, the WCOL bit of the SPI slave device is set to 1, indicating that the write SPD conflicts. But the data of the shift register is not affected, and the transmission will not be interrupted.

transfer Form

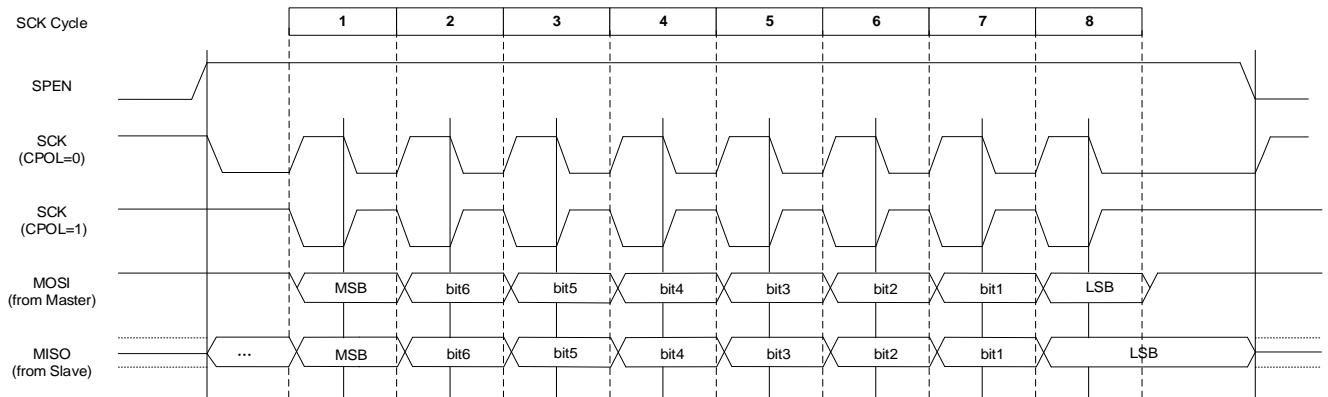
By software setting the CPOL bit and CPHA bit of the SPI control register USnCON0, the user can select four combinations of SPI clock polarity and phase. The CPOL bit defines the polarity of the clock, that is, the level state when idle, and it has little effect on the SPI transmission format. The CPHA bit defines the phase of the clock, that is, defines the clock edge that allows data sampling and shifting. In the two devices of master-slave communication, the setting of the clock polarity phase should be the same.

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



CPHA = 0 Data transfer diagram

When CPHA = 1, the master device outputs data to the MOSI line on the first edge of SCK, the slave device uses the first edge of SCK as the start signal, and the second edge of SCK starts to capture data, so the user must The operation of writing SPD is completed within two edges of one SCK. This form of data transmission is the preferred form of communication between a master device and a slave device.



CPHA = 1 Data transfer diagram

Error Detection

Writing to SPD during the data transmission sequence will cause a write conflict, and the WCOL bit in the SPI status register USnCON1 is set to 1. WCOL bit 1 will not cause interruption, and transmission will not be aborted. The WCOL bit needs to be cleared by software.

16.2 TWI

USMDn[1: 0] = 10, n=0~2 One of three serial interface USCI is configured as TWI interface:

- USTXn as SDA signal
- USCKn as CLK signal

The SC95F852X can be set as master or slave mode according to application requirements during TWI communication.

US0CON0 (95H) TWI0 Control Register 0 (read/write)

US1CON0 (A4H) TWI1 Control Register 0 (read/write)

US2CON0 (C4H) TWI2 Control Register 0 (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	TWEN	TWIF	MSTR	GCA	AA	STATE[2: 0]		
R/W	R/W	R/W	Read	Read	R/W	Read	Read	Read
POR	0	0	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description
7	TWEN	TWI enable control 0: Disable TWI 1: Enable TWI
6	TWIF	TWI interrupt flag 1. Cleared by software 2. Under the following conditions, the interrupt flag bit is set by hardware: 1) Master mode:

		<p>① Send start signal</p> <p>② After sending the address frame</p> <p>③ Receive or send the data frame</p> <p>2) Slave mode:</p> <p>① The first frame address matches successfully</p> <p>② Successfully receive or send 8-bit data</p> <p>③ Receive repeated start condition</p> <p>④ The slave receives a stop signal</p>
5	MSTR	<p>Master-slave flag</p> <p>0: Slave mode</p> <p>1: Main mode</p> <p>Description:</p> <p>1. When the TWI interface sends a start condition to the bus, it will automatically switch to the main mode, and the hardware will set this bit at the same time;</p> <p>2. When a stop condition is detected on the bus, the hardware clears this bit.</p>
4	GCA	<p>General address response flag</p> <p>0: Non-response general address</p> <p>1: When GC is set to 1 and the general address matches at the same time, this bit is set to 1 by hardware and automatically clear</p>
3	AA	<p>Answer enable bit</p> <p>0: No response, return UACK (the response bit is high)</p> <p>1: After receiving a matching address or data, a response ACK is returned</p>
2~0	STATE[2: 0]	<p>State machine status flag</p> <p>Slave mode:</p>

		<p>000: The slave is in the idle state, waiting for TWEN to be set to 1, and detecting the TWI start signal. When the slave receives the stop condition, the jump will go to this state</p> <p>001: The slave is receiving the first frame address and read/write bit (the 8th bit is the read/write bit, 1 is read, and 0 is write). The slave will jump to this state after receiving the start condition</p> <p>010: Slave receiving data status</p> <p>011: slave sending data status</p> <p>100: In the state of sending data from the slave, when the master returns to UACK, it jumps to this state and waits for a restart signal or a stop signal.</p> <p>101: When the slave is in the sending state, writing 0 to AA will enter this state, waiting for a restart signal or a stop signal.</p> <p>110: If the address of the slave does not match the address sent by the master, it will jump to this state and wait for a new start condition or stop condition.</p> <p>Master mode:</p> <p>000: The state machine is idle</p> <p>001: The Master sends the start condition or the Master is sending the slave device address</p> <p>010: Master sends data</p> <p>011: Master receives data</p> <p>100: The master sends a stop condition or receives a UACK signal from the slave</p>
--	--	---

US0CON1 (9DH) TWI0 Control Register 1 (read/write)

US1CON1 (A5H) TWI1 Control Register 1 (read/write)

US2CON1 (C5H) TWI2 Control Register 1 (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	TXnE/ RXnE	STRETCH	STA	STO	TWCK[3: 0]			

R/W	Read Only	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	TXnE/RXnE	<p>Send/receive complete flag</p> <p>In the following situations, TXnE/RXnE is set to 1</p> <p>Master mode:</p> <ul style="list-style-type: none"> ① The Master sends an address frame (write) and receives an ACK from the slave ② The master sends the data and receives the slave ACK ③ The Master receives the data, and the Master returns ACK from the slave <p>Slave mode:</p> <ul style="list-style-type: none"> ① The slave receives the address frame (read), and it matches the slave address (TWA) ② The slave receives the data, and the slave returns an ACK to the master ③ The slave sends the data and receives the master ACK (AA=1) <p>Reading and writing to TWIDAT will clear this flag.</p>
6	STRETCH	<p>Allow clock extension (slave mode)</p> <p>0: disable clock extension</p> <p>1: Allow clock extension, the Master needs to support the clock extension function</p> <p>Description: After the data transmission is completed, and ACK is 0, clock stretching occurs at this time</p>
5	STA	Start bit

		<p>Set "1" to generate start condition, TWI will switch to Master mode</p> <p>Software can set or clear this bit, or it can be cleared by hardware when the start condition is issued.</p>
4	STO	<p>Master mode stop bit</p> <p>Set to "1" in the Master mode, a stop condition will be generated after the current byte is transmitted or the start condition is sent</p> <p>Software can set or clear this bit, or it can be cleared by hardware when a stop condition is detected.</p>
3~0	TWCK[3: 0]	<p>TWI communication rate setting in Master mode:</p> <p>0000: $f_{SYS} / 1024$</p> <p>0001: $f_{SYS} / 512$</p> <p>0010: $f_{SYS} / 256$</p> <p>0011: $f_{SYS} / 128$</p> <p>0100: $f_{SYS} / 64$</p> <p>0101: $f_{SYS} / 32$</p> <p>0110: $f_{SYS} / 16$</p> <p>Others: Reserved</p> <p>Note:</p> <p>1. The setting is invalid in slave mode. The maximum clock frequency is 400kHz;</p>
7	-	Reserved

US0CON2 (9EH) TWI0 Address Register (read/write)

US1CON2 (A6H) TWI1 Address Register (read/write)

US2CON2 (C6H) TWI2 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 TWA[6: 0] cannot be written as all 0s, 00H is dedicated to general address addressing. Invalid setting in Master mode
0	GC	TWI general address enable 0: Forbid to respond to general address 00H 1: Allow response to general address 00H

US0CON3 (9FH) TWI0 Data Buffer Register (read/write)

US1CON3 (A7H) TWI1 Data Buffer Register (read/write)

US2CON3 (C7H) TWI2 Data Buffer 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 buffer register

16.2.1 Signal Description

TWI Clock Signal Line(SCL)

The clock signal is sent by the master and connected to all slaves. One byte of data is transmitted every 9 clock cycles. The first 8 cycles are used for data transmission, and the last clock is used as the receiver's response clock. It should be high when it is idle, pulled up by the pull-up resistor on the SCL line.

TWI Data Signal Line(SDA)

SDA is a bidirectional signal line, which should be high when it is idle, and is pulled high by the pull-up resistor on the SDA line.

16.2.2 Slave Operating Mode

- **Mode Start:**

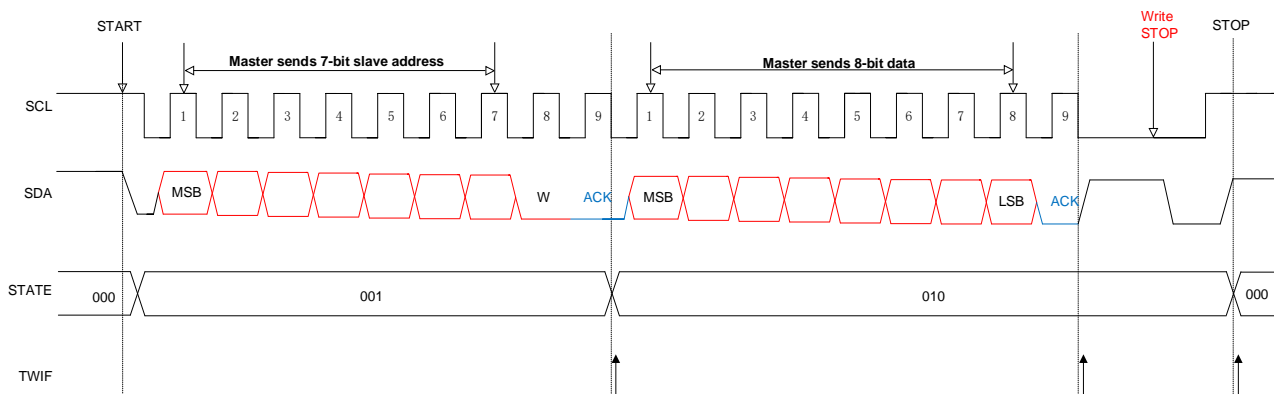
When the TWI enable flag is turned on (TWEN = 1) and the start signal sent by the Master is received at the same time, the mode is started.

The slave enters the state of receiving the first frame address (STATE[2: 0] = 001) from the idle mode (STATE[2: 0] = 000), and waits for the first frame of data from the master. The first frame of data is sent by the Master, including 7-bit address bits and 1 bit for reading and writing. All slaves on the TWI bus will receive the first frame of data from the Master. The Master releases the SDA signal line after sending the first frame of data. If the address sent by the Master is the same as the value in a slave's own address register, it means that the slave is selected. The selected slave will judge the 8th bit on the bus, that is, the data read and write bit (=1, read command) ;=0, write command), then occupy the SDA signal line, give the Master a low-level response signal in the 9th clock cycle of SCL, and then release the bus. After the slave is selected, it will enter different states according to the different read and write bits:

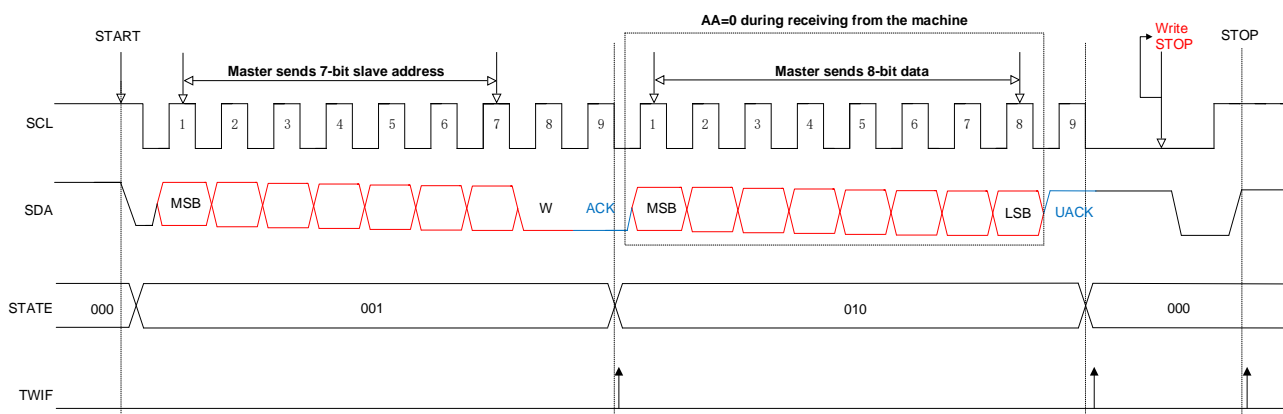
- **Non-general Address Response, Slave Device Receiving Mode:**

If the read/write bit received in the first frame is write (0), the slave enters the slave receiving state (STATE[2: 0] = 010) and waits for the data sent by the Master. The master must release the bus every time it sends 8 bits and wait for the response signal from the slave in the 9th cycle.

1. If the response signal of the slave is low, the communication of the master can be in the following three ways:
 - 1) Continue to send data;
 - 2) Resend the start signal (start), at this time the slave re-enters the state of receiving the first frame address (STATE[2: 0] = 001);
 - 3) Send a stop signal to indicate the end of this transmission, and the slave returns to the idle state, waiting for the next start signal from the Master.



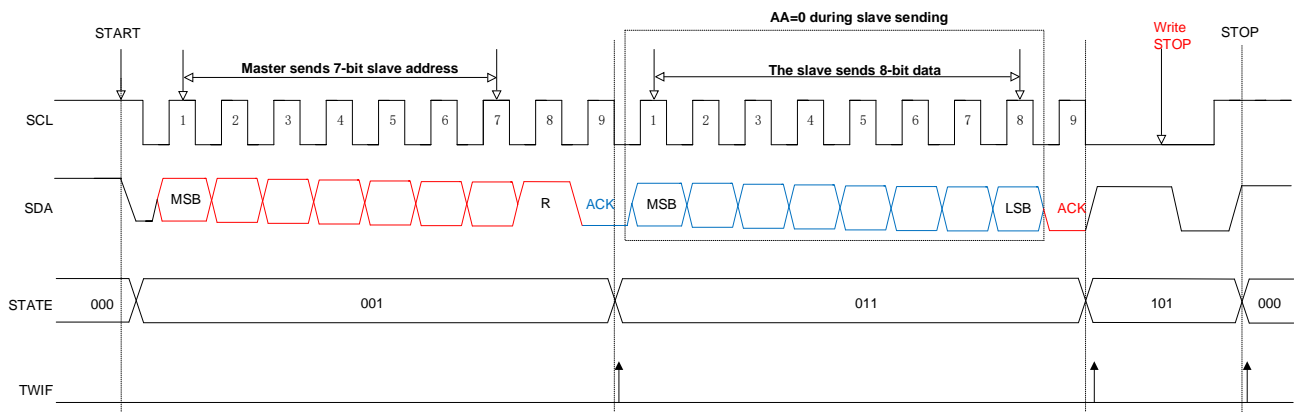
- If the slave responds to a high level (during the receiving process, the AA value in the slave register is rewritten to 0), it means that after the current byte is transmitted, the slave will actively end the transmission and return to the idle state (STATE[2: 0] = 000), no longer receive data from the Master.



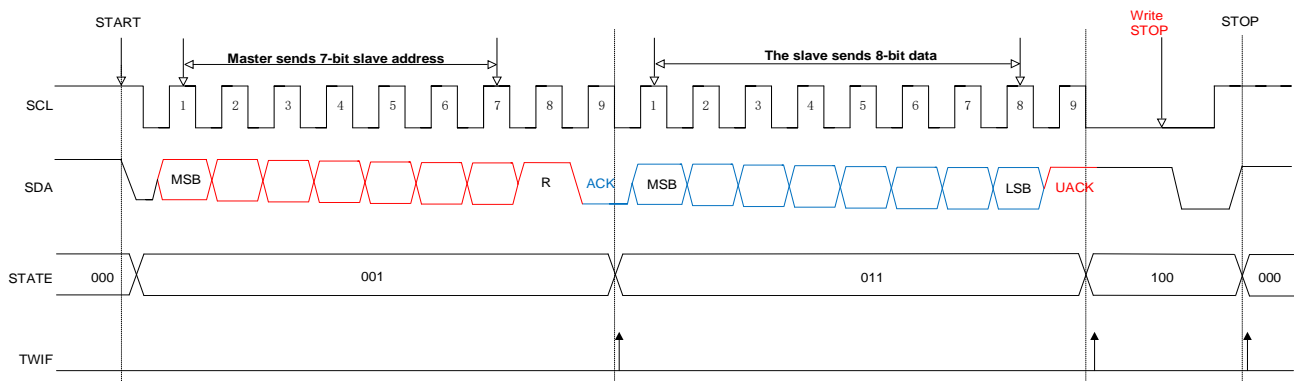
● Non-general Address Response, Slave Device Transmitting Mode:

If the read/write bit received in the first frame is read (1), the slave will occupy the bus and send data to the Master. Every time 8 bits of data are sent, the slave releases the bus and waits for the response from the master:

- If the master responds with a low level, the slave continues to send data. In the process of sending, if the AA value in the slave register is rewritten to 0, the slave will actively end the transmission and release the bus after the current byte is transmitted, and wait for the stop signal or restart signal of the master (STATE[2: 0] = 101).



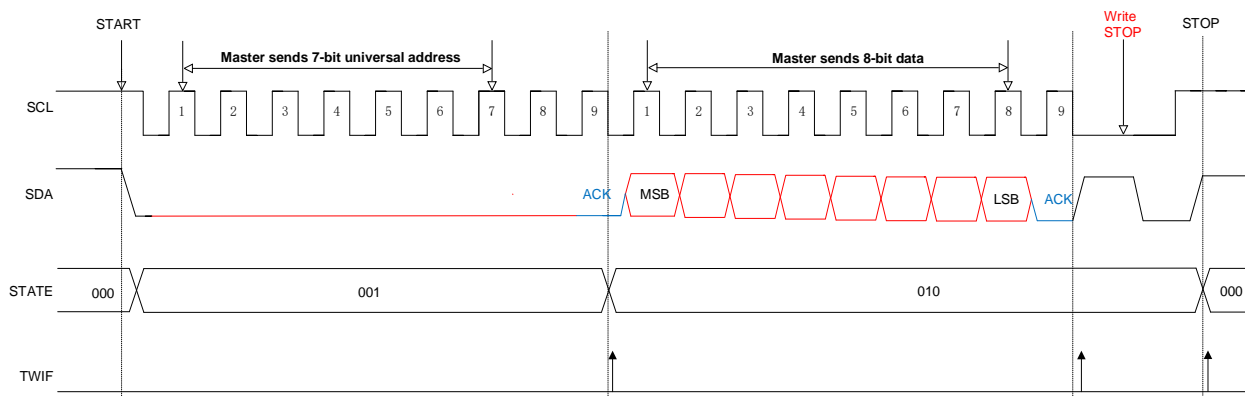
2. If the master responds to a high level, the slave STATE[2: 0] = 100, waiting for the master's stop signal or restart signal.



● General Address Response:

When GC=1, the general address is allowed to be used at this time. The slave enters the state of receiving the first frame address (STATE[2: 0] = 001), the address bit data in the first frame of data received is 0x00, and all slaves respond to the master at this time. The read and write bits sent by the master must be write (0), and all slaves enter the state of receiving data (STATE[2: 0] = 010) after receiving. The Master releases the SDA line every time 8 data is sent, and reads the status on the SDA line:

1. If there is a response from the slave, the communication of the master can be in the following three ways:
 - 1) Continue to send data;
 - 2) Restart;
 - 3) Send a stop signal to end this communication.



2. If no slave responds, SDA is idle.

Note: When using the universal address in the one-master multiple-slave mode, the read and write bits sent by the Master cannot be in the read (1) state, otherwise, all devices on the bus will respond except for the device sending the data.

16.2.3 Slave Mode Operation Steps

1. Configure USMDn[1: 0] and select TWI mode;
2. Configure the TWIn control registers USnCON0 and USnCON1;
3. Configure the TWI address register USnCON2;
4. If the slave receives data, it waits for the interrupt flag bit TWIF in USnCON0 to be set. Every time the slave receives 8 bits of data, TWIF will be set to 1. The interrupt flag bit TWIF needs to be manually cleared;
5. If the slave sends data, write the data to be sent into TWDAT, and TWI will automatically send the data. Every 8 bits are sent, the interrupt flag bit TWIF will be set.

16.2.4 Master Mode

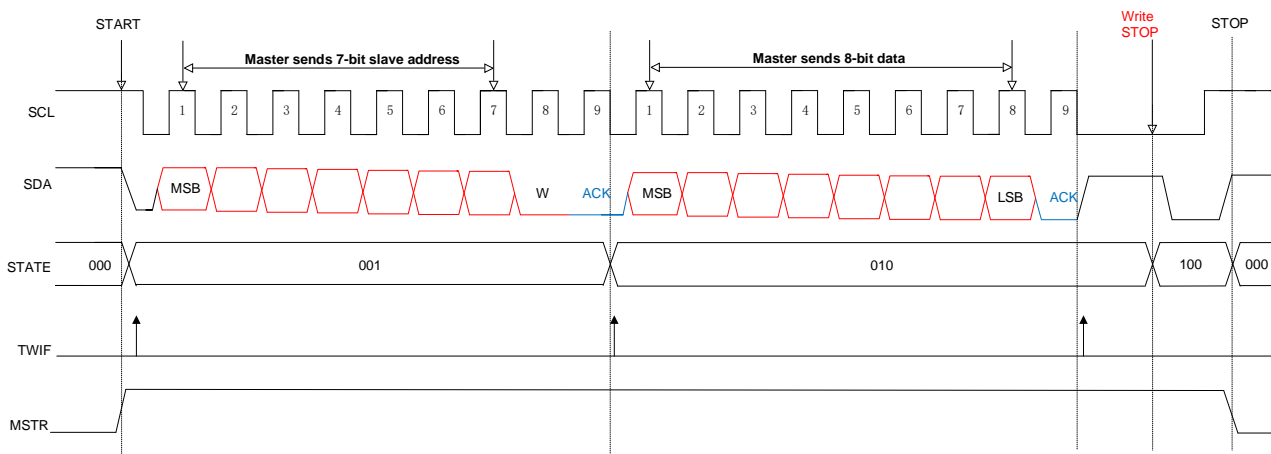
● Mode startup:

When the TWI interface sends an initial condition to the bus, it will automatically switch to the main mode, and the hardware will set the MSTR bit to 1. The Master state bit STATE[2: 0] switches from 000 to 001, and the interrupt condition TWIF is set to 1.

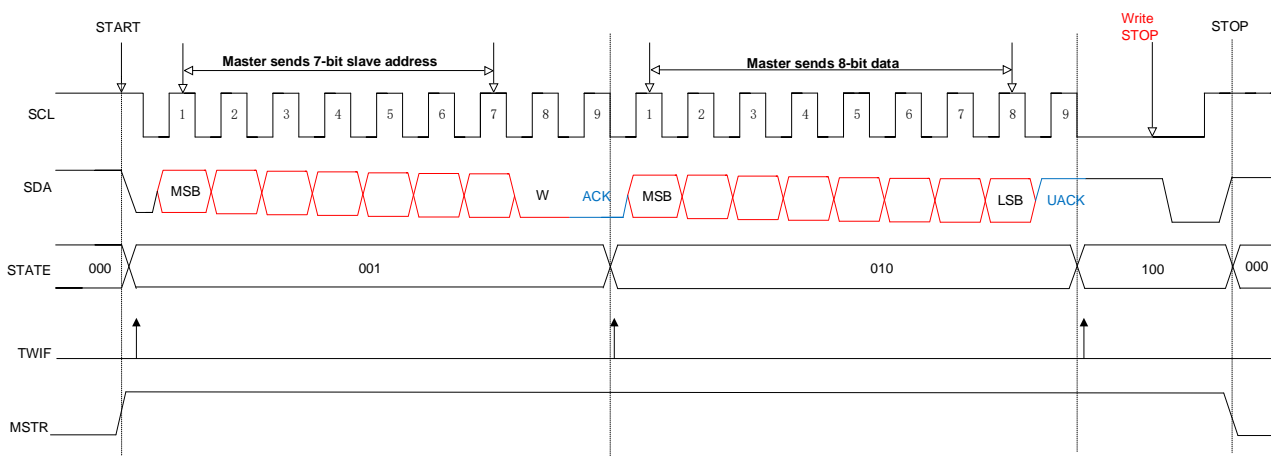
● TWI Master sending mode:

In the master sending mode, the first frame of data sent by the master includes 7 address bits (selected slave address) and 1 read/write bit (=0, write command). All slaves on the TWI bus will receive the master The first frame of data. The Master releases the SDA signal line after sending the first frame of data. The selected slave sends a response signal to the master in the 9th clock cycle of SCL, and then releases the bus and enters the slave receiving state to wait for the data sent by the master. The master must release the bus every time it sends 8 bits and wait for the response signal from the slave in the 9th cycle.

1. If the slave responds low, the master can continue to send data. You can also resend the start signal:



2. If the slave responds to a high level, it means that after the current byte has been transmitted, the slave will actively end this transmission and will no longer receive the data sent by the master. The master STATE[2: 0] will switch from the sending data state 010 to 100:



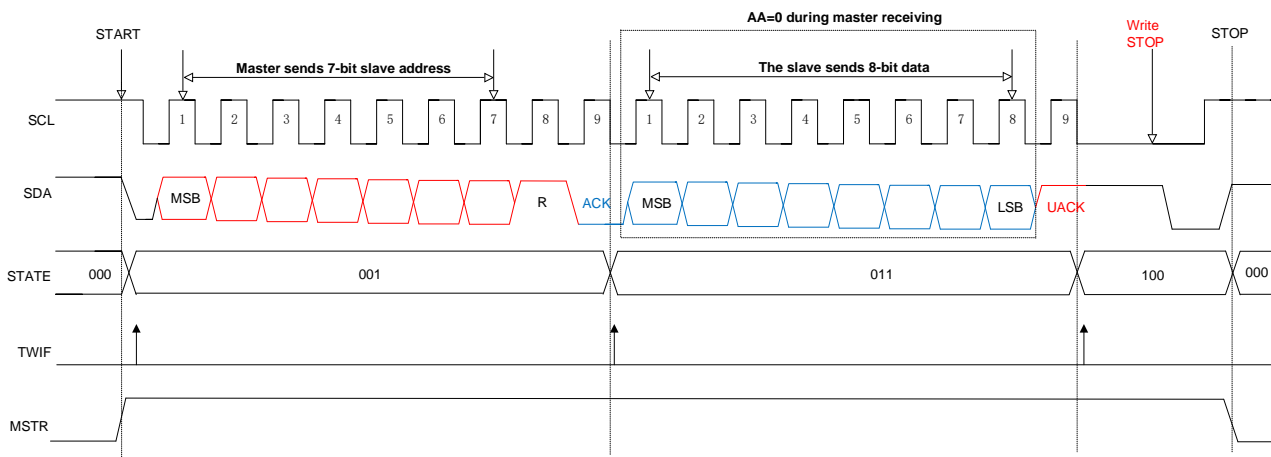
● TWI Master Receiving Mode:

In the master sending mode, the first frame of data sent by the master includes a 7-bit address bit (selected slave address) and a 1-bit read and write bit (=1, read command). All slaves on the TWI bus will receive The first frame of data to the Master. The Master releases the SDA signal line after sending the first frame of data. The selected slave sends an acknowledge signal to the master in the 9th clock cycle of SCL, and then will occupy the bus and send data to the master. Every time 8 bits of data are sent, the slave releases the bus and waits for the response from the master. The Master receives the response signal ACK after the slave address is successfully matched, and starts to receive the slave data (STATE=011):

1. If the Master response bit is enabled (AA=1), every time a BYTE data is received, the Master responds with the response signal ACK, and TWIF is set;

- Before receiving the last byte of data, if the response enable bit is turned off (AA=0), the Master will reply UACK after receiving the last byte of data, and then the Master can send a stop signal.

In the Master receiving mode, the way to actively release the bus is as follows:



16.2.5 Master Mode Operation Steps

- Configure USMDn[1: 0] and select TWI mode;
- Configure the TWIn control register USnCON0: TWEN = 1, enable TWI
- Configure the TWIn control register USnCON1: configure the TWI communication rate (TWCK[3: 0]), set the start bit STA to "1"
- Configure TWIn address register USnCON3: write "slave address plus read and write bits" into TWDAT, and send out an address frame on the bus
- If the Master receives data, it waits for the interrupt flag bit TWIF in USnCON0 to be set 1. When the Master receives 8 bits of data, the interrupt flag bit will be set. The interrupt flag bit needs to be manually cleared;
- If the Master sends data, write the data to be sent into TWDAT, and TWI will automatically send the data. Every 8 bits are sent, the interrupt flag bit TWIF will be set 1.
- After the data is sent and received, the Master can send a stop condition (STO=1), and the Master state switches to 000. Or send a repeated start signal to start a new round of data transmission.

The TWIF of the Master will not be set after the Master generates a stop!

16.3 Serial Interface (UART)

USMDn[1: 0] = 11, n=0~2 one of three serial interface USCI is configured as UART interface. It can be easily used to connect with other devices or equipment, such as Wifi module circuit or other UART communication interface driver chip. Its functions and characteristics are as follows:

1. Three communication modes are available: mode 0, mode 1 and mode 3;
2. Independent baud rate generator;
3. The interrupt RI/TI can be generated after sending and receiving, and the interrupt flag needs to be cleared by software.

When USCI is configured as UART interface: :

- USTXn as TX signal
- USRXn as RX signal

US0CON0 (95H) Serial Port 1 Control Register (read/write)

US1CON0 (A4H) Serial Port 2 Control Register (read/write)

US2CON0 (C4H) Serial Port 3 Control Register (read/write)

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

Bit number	Bit Mnemonic	Description
7~6	SM0~1	Serial port 1 control register (read/write) serial communication mode control bit 00: Mode 0, 8-bit half-duplex synchronous communication mode, serial data is sent and received on the RX pin. The TX pin is used as the transmit shift clock. 8 bits are sent and received per frame, and the lower bits are received or sent first;

		<p>01: Mode 1, 10-bit full-duplex asynchronous communication, composed of 1 start bit, 8 data bits and 1 stop bit, and the communication baud rate is variable;</p> <p>10: Reserved;</p> <p>11: Mode 3, 11-bit full-duplex asynchronous communication, composed of 1 start bit, 8 data bits, a programmable 9th bit and 1 stop bit, and the communication baud rate is variable.</p>
5	SM2	<p>Serial communication mode control bit 2, this control bit is only valid for mode 3</p> <p>0: Set RI every time a complete data frame is received to generate an interrupt request;</p> <p>1: When a complete data frame is received, RI will be set to generate an interrupt request only when RB8=1.</p> <p>Baud rate multiplier setting bit, only valid in mode 0 (SM0~1 = 00):</p> <p>0: The serial port runs at 1/12 of the system clock</p> <p>1: The serial port runs at 1/4 of the system clock</p>
4	REN	<p>Receive permission control bit</p> <p>0: It is not allowed to receive data;</p> <p>1: Allow to receive data.</p>
3	TB8	Only valid for mode 3, which is the 9th bit of the transmitted data
2	RB8	Only valid for mode 3, which is the 9th bit of the received data
1	TI	Send interrupt flag
0	RI	Receive interrupt flag

US0CON1 (9DH) Serial Port 1 Baud Rate Control Register Low Bit (read/write)

US1CON1 (A5H) Serial Port 2 Baud Rate Control Register Low Bit (read/write)

US2CON1 (C5H) Serial Port 3 Baud Rate Control Register Low Bit (read/write)

Bit number	7	6	5	4	3	2	1	0
Symbol	BAUD1L [7: 0]							
Read/ Write	Read/ Write	Read/ Write	Read/ Write	Read/ Write	Read/ Write	Read/ Write	Read/ Write	Read/ Write
Initial power-on value	0	0	0	0	0	0	0	0

US0CON2 (9EH) Serial Port 1 Baud Rate Control Register High Bit (read/write)

US1CON2 (A6H) Serial Port 2 Baud Rate Control Register High Bit (read/write)

US2CON2 (C6H) Serial Port 3 Baud Rate Control Register High Bit (read/write)

Bit number	7	6	5	4	3	2	1	0
Symbol	BAUD1H [7: 0]							
Read/ Write	Read/ Write	Read/ Write	Read/ Write	Read/ Write	Read/ Write	Read/ Write	Read/ Write	Read/ Write
Initial power-on value	0	0	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description
7~0	BAUD1 [15: 0]	USCI Serial port baud rate control $\text{BaudRate} = \frac{f_{\text{sys}}}{[\text{BAUD1H}, \text{BAUD1L}]}$

Note: [BAUD1H,BAUD1L] must be greater than 0x0010

US0CON3 (9FH) Serial 1 Data Buffer Register (read/write)

US1CON3 (A7H) Serial 2 Data Buffer Register (read/write)

US2CON3 (C7H) Serial 3 Data Buffer Register (read/write)

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

Bit number	Bit Mnemonic	Description
7~0	SBUF1[7: 0]	Serial Data Buffer Register SBUF1 contains two registers: a sending shift register and a receiving latch. The data written in SBUF1 will be sent to the sending shift register and the sending process will be started. Reading SBUF1 will return the contents of the receiving latch.

17 Smart Card (SC)

USMD0[1:0] = 00, Optional Serial port USCIO is configured as the SC interface:

- USTX0 as SC_DAT signal
- USCK0 as SC_CLK signal

SC95F852X Smart Card controller is based on the ISO/IEC 7816-3 standard. Software-controlled GPIO pins for smart card reset function and smart card insertion detection function.

17.1 Functional Description

The interface features of the SC95F852X smart card are as follows:

1. Iso-7816-3 T = 0
2. You can select the forward/reverse convention function
3. The clock source frequency (fSC) is adjustable
4. The ETU is adjustable
5. The extended protection time is adjustable
6. Programmable parity or parity bit generation and detection
7. The stop bit and Error Signal length are configurable

17.2 SC Control Register

SC95F852X smart card interface registers are as follows:

SCCON SC control register (Read/Write)

US0CON0 (95H) @ USMD0[1:0] = 00

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	SCEN	CKEN	TREN	ERS [1:0]		CONS	TRER	PCS
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	SCEN	SC Interface Settings 0: the channel serves as common I/O 1: the channel serves as port 7816
6	CKEN	SC clock output enable bit 0: disables clock output 1: enables clock output

5	TREN	SC Send and receive enable bit 0: Indicates that receiving is enabled but sending is disabled 1: indicates that sending is enabled but receiving is disabled. After sending a frame, the interface releases SC_DAT and starts to detect the Error Signal in the stop bit
4~3	ERS[1:0]	Stop and Error Signal length select bits 00: Stop and Error Signal contain 2 ETU 01: Stop and Error Signal contain 2 ETU 10: Stop and Error Signal contain 1.5 ETU 11: Stop and Error Signal contain 1 ETU
2	CONS	Encoding mode control bit 0: positive convention, LSB transmission, positive logic level 1: reverse convention, MSB transmission, reverse logic level
1	TRER	Data sending and receiving verification error retransmission control bit 0: check error, directly set interrupt flag bit 1: a low level reply will be sent if there is an error in receiving data verification. After sending data, the low level reply will be received and the data will be resend
0	PCS	Parity selection bit 0: no check 1: even check

ETUCK0 ETU Rate Division Frequency Register 0 (Read/Write)

US0CON1 (9DH) @ USMD0[1:0] = 00

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

ETUCK1 ETU Rate Division Frequency Register 1 (Read/Write)

US0CON2 (9EH) @ USMD0[1:0] = 00

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	SCCK[4:0]				ETUCK [11:8]			
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	1

Bit number	Bit Mnemonic	Description
7~4	SCCK[4:0]	SC Clock setting: SC Clock cycle $T_{SC} = (SCCK[4:0] + 1) * 2 / f_{sys}$
ETUCK0[7:0] ETUCK1[3:0]	ETUCK [11:0]	ETU setting An ETU is $(ETUCK [11:0] + 1)$ SC Clock, i.e. ETU cycle $T_{ETU} = T_{SC} * (ETUCK [11:0] + 1)$ Note: The frequency range of smart card communication is between 1MHz and 5MHz, so ETUCK [11:0] must be greater than 0x004

SCDATA SC data register(Read/Write)
US0CON3 (9FH) @ USMD0[1:0] = 00

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	SCD[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	SCD[7:0]	Stores data sent and received by the SC interface

EGT SCExtended protection time register (Read/Write)
US0CON4(B6H) @ USMD0[1:0] = 00

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	EGT [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	EGT [7:0]	SC extended protection time Expand the protection time to EGT [7:0] ETUS, namely: Actual extended protection time in SC communication $T_{EGT} = T_{ETU}^*$ EGT [7:0]

SCSTA0 SC Status register (Read/Write)
US0CON5 (B7H) @ SES=0

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	SES	TC	RC	WTRT	TBUSY	RBUSY	WTER	FER
R/W	R/W	R/ Write the 1 to clear the 0	R/ Write the 1 to clear the 0	R	R	R	R/ Write the 1 to clear the 0	R/ Write the 1 to clear the 0
POR	0	0	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description
7	SES	Status/error register switch control bit 0: US0CON5 (B7H) is the read/write status register 1: US0CON5 (B7H) is the read/write error register
6	TC	After sending, the hardware is set to 1 and the software must be cleared
5	RC	After receiving, hardware is set to 1 and software is required to clear
4	WTRT	Waiting for data to resend flag bits 0: no data retransmission waiting event occurs 1: data is being retransmitted
3	TBUSY	Data sending busy flag bit 0: Data transmission is idle 1: Data is being sent. The hardware sets the start bit of data to 1. The hardware clears the end bit of data and the end bit of data
2	RBUSY	Data receive busy flag bit 0: Data receiving is idle 1: Data is being received. When the start bit is received, the hardware sets the value to 1. When the stop bit is received, the hardware clears the value

1	WTER	Wait timeout flag bit 0: no wait timeout occurs 1: The hardware is set to 1 when the wait time exceeds 40,000 clock cycles between two consecutive characters in receive mode or when the reset response exceeds 40000
0	FER	Received frame error flag bit 0: The frame format is correct 1: the frame format is incorrect. No valid stop bit is received

SCSTA1 SC Read/write error register (read/write)
US0CON5 (B7H) @ SES=1

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	SES	-	-	-	TPER	RPER	-	ROVF
R/W	R/W	-	-	-	R/ Write the 1 to clear the 0	R/ Write the 1 to clear the 0	-	R/ Write the 1 to clear the 0
POR	0	x	x	x	0	0	x	0

Bit number	Bit Mnemonic	Description
7	SES	Register switch control bit 0: US0CON5 (B7H) is the read/write status register 1: US0CON5 (B7H) is the read/write error register
3	TPER	Send data parity error flag bit 0: The sent data verification is normal 1: The sent data verification is error
2	RPER	Received data parity error flag bit 0: The received data verification is normal 1: The received data verification is error
0	ROVF	Receive overflow flag bit 0: no overflow error occurs 1: The receiving data buffer is not read and receives new data
6~4,1	-	Reversed

17.3 Operating Mode

17.3.1 Smartcard Description

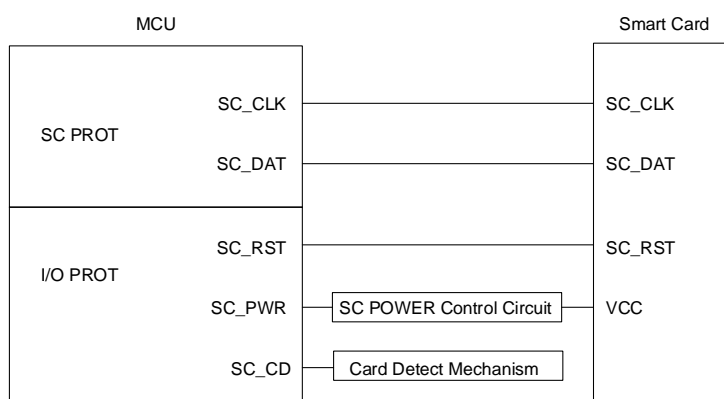
The smart card interface controller supports activation, cold reset, warm reset and release sequences. Activation, cold reset, warm reset and release sequences are described below.

SC interface connection

1. SC_CLK: SC Interface clock pin (MCU output)
2. SC_DAT: SC Interface data pin (bidirectional)

The following signal ports can be customized using GPIO:

1. SC_RST: the reset pin of smart card, controlled by GPIO of MCU, with low effective reset signal
2. SC_PWR: The power switch control pin of the smart card is connected to GPIO of MCU. The high output of SC_PWR indicates that the smart card VCC is powered on
3. SC_CD: Smart card detection pin, connected to GPIO of MCU, to detect whether the card is inserted

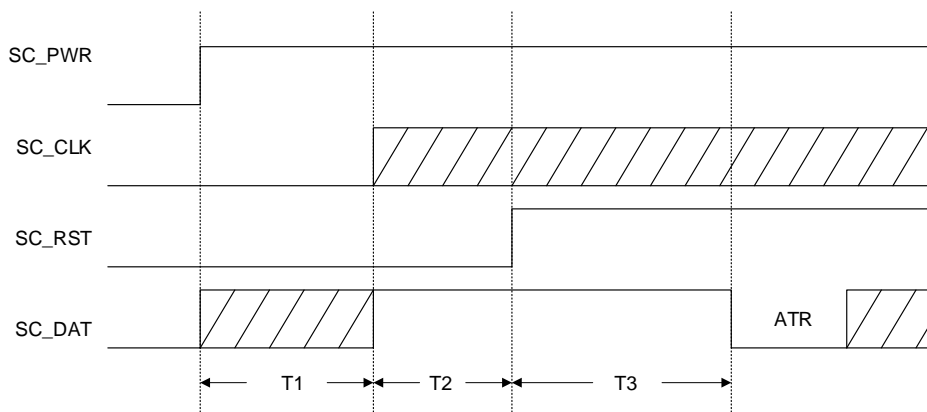


SCInterface connection diagram

Activation and cold reset

The timing requirements for activation and cold reset are as follows:

1. Set the output low level of SC_RST pin
2. The smart card power supply is enabled (set the SC_PWR output high) and the T1 cycle starts. Before the T1 cycle ends, set the SC port to receive mode and pull the SC_DAT port to a stable high level
3. Set CKEN (SCcon.6) to 1 to enable the SC_CLK output clock and enter the T2 cycle
4. After a period of time, set the SC_RST pin output height to end the T2 cycle and enter the T3 cycle
5. After a period of time, the smart card responds to the ATR signal, and the T3 cycle ends. The user can obtain the ATR data read by the SC interface through SCDATA



SC Activate and cold reset timing

period	time series
T1	The smartcard VCC is powered on and the SC_CLK output is enabled
T2	The stage before the SC_CLK signal is output to the SC_RST reset signal rising edge
T3	The SC_RST rises to the stage where the smart card responds to the ATR signal

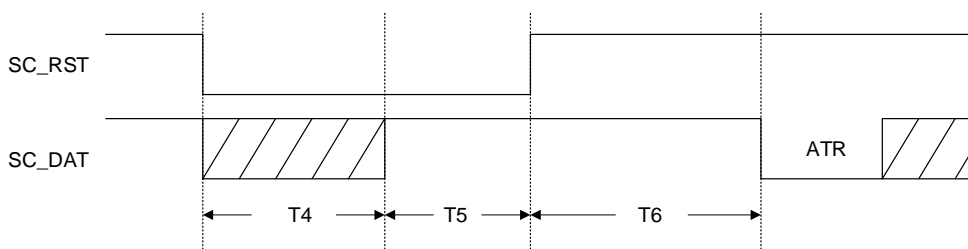
Recommended timing relation (unit: SC Clock)

T1	T2	T3
83.5	491	$400 \leq T3 \leq 40000$
133	537	
165	569	
165	42060	

Warm reset

The timing requirements of warm reset are as follows:

1. Pull down SC_RST to enter the T4 cycle. Before the T4 cycle ends, pull the SC_DAT port to a stable high level
2. Enter the T5 cycle. After a period of time, set the SC_RST output high level to end the T5 cycle
3. Enter the T6 period and wait for the smartcard to answer
4. After a period of time, the smart card responds to the ATR signal, and the T6 cycle ends. The user can obtain ATR data read by the SC interface through SCDATA



SC warm reset time series

period	time series
T4	The SC_RST reset signal is enabled to the SC_DAT setting phase
T5	SC_DAT is steadily pulled to the output stage of SC_RST rising edge
T6	SC_RST reset signal ends when the smart card responds to the ATR signal

Recommended sequence relation (unit: SC Clock)

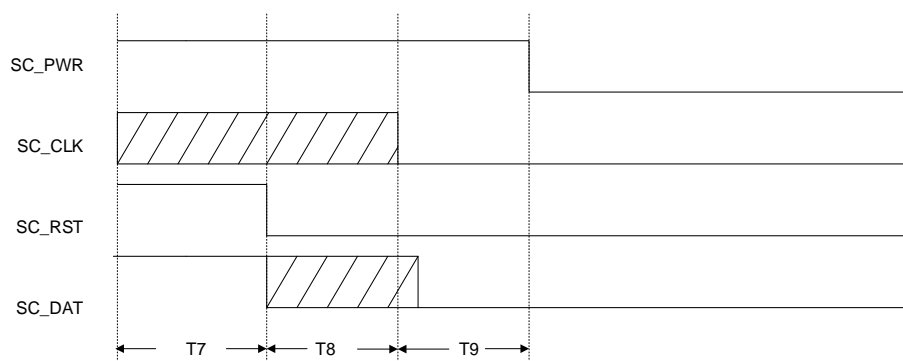
T4	T5	T6
81	483	$400 \leq T6 \leq 40000$

129	531	
161	563	
161	42106	

Release

After communication, regardless of whether the result is normal or not, the SC interface must perform the following steps to release the smart card contact:

1. In T7, SC_DAT is idle. When SC_RST is lowered, the signal output is reset and the contact release sequence starts
2. Before the T8 cycle ends, set CKEN (SCcon.6) to 0 to stop SC_CLK signal output
3. After the SC_CLK signal output stops during the T9 cycle, pull down the SC_DAT
4. After waiting for a period of time, power off the smart card (set SC_PWR output low here).
5. Complete the release sequence.



SC release sequence

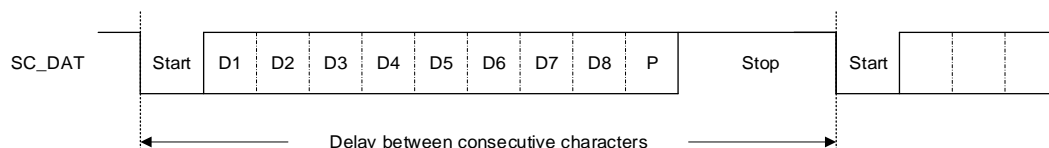
period	time series
T7	SC_DAT is idle, and the smart card detects the reset signal
T8	The SC_RST reset signal is enabled to the SC_CLK output stop phase
T9	The output of SC_CLK and SC_DAT is stopped until the VCC of the smartcard is disabled

Recommended timing relation (unit: SC Clock)

T7	T8	T9
97	83	87
145	131	135
177	163	167

17.3.2 Smart card data transmission

The data format of the smart card interface consists of ten consecutive bits as follows:



SC data character

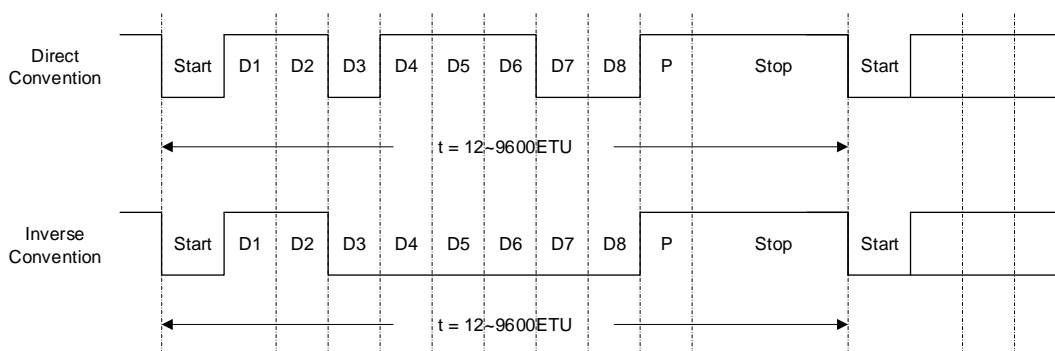
17.3.3 Initialization character TS

According to 7816-3, there are two possible modes for the initialization character TS in the smart card ATR information:

If the TS mode is 1101_1100, it is a forward convention. Decoded by forward convention, the bytes transmitted are equal to 0x3B.

If the TS mode is 1100_0000, it is the reverse convention. Decoded by reverse convention, the bytes transmitted are equal to 0x3F.

The user can set CONS (SCCON.2) to '0' or '1' and change the convention of the operation after receiving the TS from the ATR.

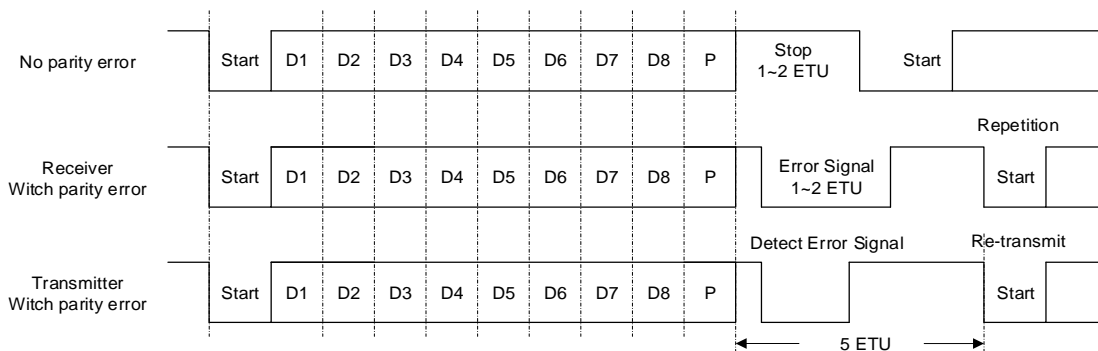


Initialization character

17.3.4 Error signals and character duplicates

The SC interface of SC95F852X complies with the error signal and character repetition rules in ISO7816-3 T= 0 mode:

- During normal communication, Stop bit is released as high level, and the duration of Stop bit provided by SC is set by ERS[1:0].
- In receiving mode, users can set TRER (SCCON.1) to select the processing method for a parity bit exception in SC receiving mode.
 - While TRER (SCCON.1)=0, SC will generate a transmission error interrupt to the CPU;
 - TRER (SCCON. 1)=1, the low level consistent with ETU cycle setting set by ERS[1:0] is sent at stop bit time when the checksum Error of the received data is detected, that is, Error signal, and the current sender shall retransmit this character.
- In sending mode, if the receiver is detected to lower SC_DAT at the stop bit time and the duration is consistent with the Error Signal set by ERS[1:0], SC retransmits this character without adding the extended protection time TEGT. The period from the Stop bit to the Start of the retransmission character is fixed to five ETUs, and the maximum number of multiple transmission times is three.

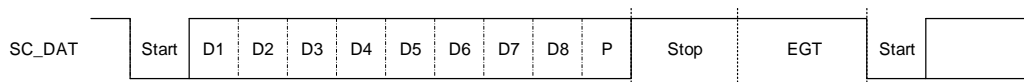


SC error signal

17.3.5 Extended protection time

Extended protection duration The EGT is valid only when the SC interface sends packets. You can set the extended protection duration using EGT [7:0].

In the SC interface transmission state, there is an extended protection period (TEGT) before the Start of character transmission. TEGT is invalid during character retransmission.



SC Extended protection time

18 Analog-to-Digital Converter (ADC)

The SC95F852X is built with a 12-bit 11-channel high-precision 1M high-speed successive approximation ADC. The external 10-channel ADC and other functions of the IO port are multiplexed. The internal one can be connected to 1/4 VDD and used for measuring VDD voltage with internal 2.048V or 1.024V reference voltage.

The reference voltage of the ADC can be selected from two options:

- ① VDD pins (that is, directly internal VDD);
- ② The internal Regulator output reference voltage 2.048V or 1.024V precision.

18.1 ADC-Related Registers

ADCCON (ADH) ADC Control Register (read/write)

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

Bit number	Bit Mnemonic	Description
7	ADCEN	Power on ADC 0: Disable the ADC module power 1: Enable the ADC module power
6	ADCS	ADC start trigger control (ADC Start) Write "1" to this bit to start ADC conversion, that is, this bit is only the trigger signal of ADC conversion. This bit can only be written with 1 to be valid.
5	EOC /ADCIF	Conversion complete/ADC Interrupt Flag (End Of Conversion / ADC Interrupt Flag)

		<p>0: Conversion has not been completed</p> <p>1: ADC conversion is complete. Need user software to clear</p> <p>ADC conversion complete flag EOC: when the user sets ADCS to start conversion, this bit will be automatically cleared to 0 by the hardware; when the conversion is completed, this bit will be automatically set to 1 by the hardware;</p> <p>ADC interrupt request flag ADCIF:</p> <p>This bit is also used as an interrupt request flag for ADC interrupt. If the user enables the ADC interrupt, the user must clear this bit by software after the ADC interrupt occurs.</p>
4~0	ADCIS[4: 0]	<p>ADC Input Selector (ADC Input Selector)</p> <p>00000: select AIN0 as ADC input</p> <p>00001: select AIN1 as ADC input</p> <p>00010: Use AIN2 as ADC input</p> <p>00011: Choose AIN3 as ADC input</p> <p>00100: Choose AIN4 as ADC input</p> <p>00101: Choose AIN5 as ADC input</p> <p>00110: select AIN6 as ADC input</p> <p>00111: Choose AIN7 as ADC input</p> <p>01000: select AIN8 as ADC input</p> <p>01001: select AIN9 as ADC input</p> <p>01010~11110: reserved</p> <p>11111: ADC input is 1/4 V_{DD}, which can be used to measure power supply voltage</p>

ADCCFG2 (B5H) ADC Set Register 2 (read/write)

Bit number	7	6	5	4	3	2	1	0
Symbol	-	-	-	LOWSP[2: 0]			-	-
Read/	-	-	-	Read/	Read/	Read/	-	-

Write				Write	Write	Write		
Initial power-on value	x	x	x	0	0	0	x	x

Bit number	Bit Mnemonic	Description
4~2	LOWSP[2: 0]	<p>ADC sampling periods selection:</p> <p>100: The sampling time is 3 system clocks, (about 100ns @fsys = 32MHz)</p> <p>101: The sampling time is about 6 system clocks, (about 200ns @fsys = 32MHz)</p> <p>110: The sampling time is about 16 system clocks, (about 500ns @ fsys = 32MHz)</p> <p>111: The sampling time is about 32 system clocks, (about 1000ns @ fsys = 32MHz)</p> <p>Other: Reserved</p> <p>Description: The total time from ADC sampling to completion of conversion TADC = sampling time + conversion time</p> <p>The ADC conversion time of the SC95F852X is fixed at 950ns.</p>
7~5, 1~0	-	Reserved

ADCCFG0 (ABH) ADC Set Register 0 (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	EAIN7	EAIN6	EAIN5	EAIN4	EAIN3	EAIN2	EAIN1	EAIN0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

POR	0	0	0	0	0	0	0	0
-----	---	---	---	---	---	---	---	---

ADCCFG1 (ACH) ADC Set Register 1 (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	-	-	-	-	-	-	EAIN9	EAIN8
R/W	-	-	-	-	-	-	R/W	R/W
POR	x	x	x	x	x	x	0	0

Bit number	Bit Mnemonic	Description
0~7	EAINx (x=0~9)	ADC port setting register 0: Set AINx as IO port 1: Set AINx as ADC input and automatically remove the pull-up resistor.

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

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

Bit number	Bit Mnemonic	Description
7~6	VREFS[1: 0]	Reference voltage selection (the initial value is transferred from Customer Option, the user can modify the setting) 00: Set VREF of ADC to VDD; 01: Set the VREF of ADC to the internal accurate 2.048V; 10: Set the VREF of ADC to the internal accurate 1.024V; 11: Reserved

ADCVL (AEH) ADC Conversion Value Register (low bit) (read/write)

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

ADCVH (AFH) ADC Conversion Value Register (high bit) (read/write)

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

Bit number	Bit Mnemonic	Description
11~4	ADCV[11: 4]	The high 8-bit value of ADC conversion value
3~0	ADCV[3: 0]	Low 4 bits of ADC conversion value

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

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

Bit number	Bit Mnemonic	Description
6	EADC	ADC interrupt enable control 0: Do not allow EOC/ADCIF to generate interrupts 1: Enable EOC/ADCIF to generate interrupt

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

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	-	IPADC	IPT2	IPUART	IPT1	IPINT1	IPT0	IPINT0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W

POR	x	0	0	0	0	0	0	0
-----	---	---	---	---	---	---	---	---

Bit number	Bit Mnemonic	Description
6	IPADC	ADC interrupt priority selection 0: Set the interrupt priority of ADC to "low" 1: Set the interrupt priority of ADC to "High"

18.2 ADC Conversion Steps

The actual operation steps required for the user to perform ADC conversion are as follows:

- ① Set the ADC input pin; (set the bit corresponding to AINx as ADC input, usually the ADC pin will be fixed in advance);
- ② Set ADC reference voltage Vref, set the frequency used for ADC conversion;
- ③ Enable the ADC module power supply;
- ④ Select ADC input channel; (set ADCIS bit, select ADC input channel);
- ⑤ Start ADCS and start conversion;
- ⑥ Wait for EOC/ADCIF=1. If the ADC interrupt is enabled, the ADC interrupt will be generated. The user needs to clear the EOC/ADCIF flag by software;
- ⑦ Get 12-bit data from ADCVH and ADCVL, first high bit and then low bit, one conversion is completed;
- ⑧ If you do not change the input channel, repeat steps 5~7 for the next conversion.

Note: Before setting IE[6] (EADC), the user is better to clear EOC/ADCIF with software, and also clear the EOC/ADCIF when the ADC interrupt service routine is executed to avoid continuous ADC interrupts. .

19 Dual Mode Touch Circuit

The SC95F852X has a 23-channel dual mode capacitive touch circuit, which can be configured in high sensitivity mode or high reliability mode. Its features are as follows:

1. High sensitivity mode can be adapted to touch applications requiring high sensitivity, such as spacer button touch control and proximity induction
2. High reliability mode has a strong anti-interference ability, can pass 10V dynamic CS test
3. It can realize 23 touch control keys and derivative functions
4. High flexibility to develop software library support, low development difficulty
5. Automated debugging software support, intelligent development
6. The touch module can work in the low-power mode under the MCU Stop mode

19.1 Power Consumption Mode of Touch circuit

The SC95F852X allows touch scanning to be enabled in STOP Mode: this approach can reduce the overall power consumption of the MCU for touch applications with low power requirements.

Users can understand that the touch circuit of SC95F852X has two power consumption modes:

1. Normal operation mode
2. Low power operation mode

The two power consumption modes are defined as follows:

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

19.2 Touch Mode

The SC95F852X dual-mode touch circuit provides two touch modes:

1. High sensitivity mode
2. High reliability mode

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

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 mode
characteristics	<ol style="list-style-type: none"> ① High anti-jamming ability, can pass 3V dynamic CS ② High sensitivity 	<ol style="list-style-type: none"> ① Super anti-jamming ability, can pass 10V dynamic CS ② Low power consumption
Applicable application	<ol style="list-style-type: none"> ① Common touch button application ② Space touch button application ③ Proximity induction applications 	<ol style="list-style-type: none"> ① requires the application of super anti-interference ② there are 10V dynamic CS requirements

	④ touch control applications with higher sensitivity requirements	for applications
How to get into mode	Select the high sensitivity mode by loading the high sensitivity touch library into the project project	Select the high sensitivity mode by loading the high sensitivity touch library into the project project
documentat ion	SinOne SC95F Series TouchKey MCU Application Guide Related chapters: <ul style="list-style-type: none"> ● 2 SC95F8XXX _HIGHSENSITIVE_LIB_T1 Library ● 3 SC95F8XXX _HIGHSENSITIVE_LIB_T2 Library 	SinOne SC95F Series TouchKey MCU Application Guide Related chapters: <ul style="list-style-type: none"> ● 4 SC95F8XXX _HIGHRELIABILITY_LIB_T1 Library
The correspondi ng library file	SC95F8XXX _HighSensitive_Lib_Tn_Vx.x.x.LIB	SC95F8XXX _HighReliability_Lib_Tn_Vx.x.x.LIB
Matters needing attention	① The T1 library is used for spring type applications ② The T2 library is applied to the application of spacer type, and the number of keys is at least 3 or more	For spring type applications only
Choose instructions	It is generally recommended to use this high sensitivity mode for a better use experience.	There are only two situations where high reliability mode is recommended: ① Need to go through 10V dynamic CS ② Low power consumption current is required, and the current cannot be full in high sensitivity mode

20 CRC Module

The SC95F852X has a built-in hardware CRC module. During the CRC execution calculation, the CPU keeps the program counter. After the CRC calculation is completed, the program counter continues to execute the following instructions.

The module has two calculation modes:

Hardware CRC mode 1: CRC operation processing on specified data:

Write the data needed for CRC calculation to the CRC data register CRCREG. When the CRC calculation result needs to be read, read it out from CRCDRn (n = 0~3).

CRC calculation for a single byte requires 8 system clocks, namely 0.25μs@32MHz.

Hardware CRC mode 2: CRC calculation processing for APROM:

It can be used to generate the 32-bit CRC value of APROM (ie 64K bytes Flash ROM) in real time. This value is compared with the theoretical value to monitor whether the content of the program area is correct. The theoretical value of CRC does not need to be calculated by the user. The burning software will automatically complete the calculation according to the loaded code and Code area setting items and write the 4 bytes CRC32 calculation result into the CRC result storage area through the programmer during burning. The specific operation For the method, please refer to "User Manual of SinOne Development Mass Production Tool".

Note: Mode two is invalid when hardware CRC is started in LDRM.

The hardware CRC parameter model of SC95F852X:

CRC algorithm name	CRC-32/MPEG-2
Polynomial formula	$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$
Data width	32bit
Initial value	0xFFFFFFFF
XOR value	0x00000000
Input value inversion	false
Output value inversion	false
LSB/MSB	MSB

CRC Precautions for use:

1. CRCDRn write data and read data are not the same;

2. The CRC value calculated by the hardware is the 32-bit CRC check value of the data in the entire program area (note that the IAP area is not included here!). If there is a residual value after the user's last operation in the address unit, it will cause the CRC value to be inconsistent with the theoretical value. Therefore, it is recommended that the user erase the entire Flash ROM before programming the code to ensure that the CRC value is consistent with the theoretical value;
3. The hardware CRC calculation range does not include the IAP area;
4. Be sure to add at least 8 NOP instructions after the CRC start operation statement to ensure that the CRC calculation is completed;
5. When performing CRC calculation, it is necessary to disable the global interrupt EA, and then reopen the global interrupt after 8 NOPs.

20.1 CRC Check Operation Related Registers

OPERCON (EFH) Operation Control Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	OPERS	MD	-	-	-	-	CRCRST	CRCSTA
R/W	R/W	R/W	-	-	-	-	R/W	R/W
POR	0	0	x	x	x	x	0	0

Bit number	Bit Mnemonic	Description
1	CRCRST	CRCDR register reset (Q31~Q0) Write "1" to this bit to reset CRCDR to all 1s
0	CRCSTA	CRC hardware calculation start bit Write "1" to this bit to start a check sum calculation. This bit can only be written with 1 to be valid.

The read and write operations of the CRC data register CRCDR_n (n = 0~3) are controlled by the two registers CRCINX and CRCREG. The specific position of each CRCRN is determined by CRCINX, as shown in the following table:

Symbol	Address	Description		POR
CRCINX	FCH	CRC pointer	CRCINX[7: 0]	00000000b
CRCREG	FDH	CRC register	CRCREG[7: 0]	nnnnnnnnb

Symbol	Address	Description	7	6	5	4	3	2	1	0
CRCDR3	03H@FDH	CRC Data register 3	Q31	Q30	Q29	Q28	Q27	Q26	Q25	Q24
CRCDR2	02H@FDH	CRC Data register 2	Q23	Q22	Q21	Q20	Q19	Q18	Q17	Q16
CRCDR1	01H@FDH	CRC Data register 1	Q15	Q14	Q13	Q12	Q11	Q10	Q9	Q8
CRCDR0	00H@FDH	CRC Data register 0	Q7	Q6	Q5	Q4	Q3	Q2	Q1	Q0

The related description of CRCDR_n (n = 0~3) bits is as follows:

Bit number	Bit Mnemonic	Description
Bit31~0	Qx (x = 0~31)	<p>Hardware CRC mode 1: CRC operations are performed on specified data:</p> <ol style="list-style-type: none"> 1. Write the CRCRST and reset the CRCDR to all 1s 2. When CRCREG is written, the hardware automatically calculates the CRC result and continues to store it in the CRCDR 3. Read CRC calculation results immediately when necessary <p>Hardware CRC mode 2: CRC operation on APROM:</p> <ol style="list-style-type: none"> 4. When the CRCSTA starts, the CPU enters IDLE automatically

		<p>5. Automatically reset CRCDR to all 1:</p> <p>The hardware CRC calculation scope does not include the IAP area. The calculation range of CRC is divided into four types according to the value of IAPS[1:0] :</p> <p>① IAPS[1:0]=00 (Flash ROM last 0K can be IAP) : 0000H ~ before last 0K</p> <p>② IAPS[1:0]=01 (Flash ROM last 1K can IAP) : 0000H ~ before last 1K</p> <p>③ IAPS[1:0]=10 (Flash ROM last 2K IAP) : 0000H ~ before last 2K</p> <p>④ IAPS[1:0]=11 (Flash ROM all IAP) : all Flash ROM</p> <p>6. The CPU automatically exits IDLE and the CRC calculation result can be read</p> <p>Note: Write data and read data are not the same data.</p>
--	--	--

When operating CRC-related SFR, the CRCINX register stores the address of the relevant CRCTION register, and the CRCREG register stores the corresponding value.

Before reading CRCREG, you need to set CRCINX and then read it. After each reading, CRCINX automatically adds 1 (0~3 cycles).

Hardware CRC mode 1 example: calculate CRC according to the data provided by the user

```
#include "intrins.h"

xdata unsigned int i;

xdata unsigned long int CRC_Result = 0x00; // Verification result

unsigned char a[16] = {0x00,0x01,0x02,0x03,0x04,0x05,
0x06,0x07,0x08,0x09,0x0A,
0x0B,0x0C,0x0D,0x0E,0x0F}; // The value to be verified

typedef struct
{
    char a3; // Highest address
    char a2; // Second highest address
    char a1; // Second lowest address
```

```
char a0; // Lowest address

}Value_Typedef;

typedef union
{
    Value_Typedef reg;
    unsigned long int result; // Final Results
}Result_Typedef;

Result_Typedef CRC_Result;

    EA = 0;                // Disable the global interrupt
    OPERCON |= 0x02;       // Start software inspection
    _nop();                // At least 8 NOP instructions
    _nop();
    _nop();
    _nop();
    _nop();
    _nop();
    _nop();
    _nop();
    _nop();
    _nop();
    _nop();
    _nop();
    for(i=0; i<16; i++)
    {
        CRCREG = a[i];    // Calculated value
        _nop();           // At least 8 NOP instructions
        _nop();
        _nop();
        _nop();
```

```
_nop();  
_nop();  
_nop();  
_nop();  
_nop();  
_nop();  
}  
  
CRCINX = 0x00;  
  
CRC_Result.reg.a0 = CRCREG;  
CRC_Result.reg.a1 = CRCREG;  
CRC_Result.reg.a2 = CRCREG;  
CRC_Result.reg.a3 = CRCREG;  
  
temp = CRC_Result.result; // Get results  
  
EA = 1; // Enable global interrupt
```

Hardware CRC mode 2 routines: generate APROM CRC in real time

```
#include "intrins.h"  
  
typedef struct  
{  
    char a3; // Highest address  
    char a2; // Second highest address  
    char a1; // Second lowest address  
    char a0; // Lowest address  
}Value_Typedef;  
  
typedef union  
{  
    Value_Typedef reg;  
    unsigned long int result; //Final Results  
}Result_Typedef;
```

```
unsigned long int temp = 0x00;

Result_Typedef CRC_Result;

    EA = 0;                // Disable global interrupts
    OPERCON |= 0x01;       // Start hardware verification
    _nop();                // At least 8 NOP instructions
    _nop();
    _nop();
    _nop();
    _nop();
    _nop();
    _nop();
    _nop();
    _nop();

    CRCINX = 0x00;

    CRC_Result.reg.a0 = CRCREG;
    CRC_Result.reg.a1 = CRCREG;
    CRC_Result.reg.a2 = CRCREG;
    CRC_Result.reg.a3 = CRCREG;

    temp = CRC_Result.result;    // Get results

    EA = 1;                    // Enable global interrupt
```

Note: It is prohibited to write values other than the CRC register address to the CRCINX register! Otherwise it will cause abnormal system operation!

21 Electrical Characteristics

21.1 Absolute Maximum Ratings

Symbol	Parameter	Min Value	Max Value	UNIT	Condition
VDD/VSS	DC supply voltage	-0.3	5.5	V	
Voltage ON any Pin	Input/output voltage of any pin	-0.3	V _{DD} +0.3	V	
T _A	Operating temperature	-40	105	°C	
T _{STG}	Storage temperature	-55	125	°C	
I _{VDD}	Current value flowing through VDD	-	200	mA	T _A = +25°C
I _{VSS}	Current value flowing through VSS	-	200	mA	T _A = +25°C

21.2 Recommended Operating Conditions

Symbol	Parameter	Min Value	Max Value	UNIT	System Clock frequency
V _{DD}	Operating Voltage	2.0	5.5	V	32MHz
T _A	Operating temperature	-40	105	°C	

21.3 Flash ROM Characteristics

Symbol	Parameter	Min Value	Typical Values	Max Value	UNIT	Condition
N_{END}	Wipe the number	100,000	-	-	Cycle s	
T_{DR}	Data Retention Time	100	-	-	Years	$T_A = +25^{\circ}\text{C}$
$T_{S-Erase}$	Sector Erase Time	-	5	-	ms	$T_A = +25^{\circ}\text{C}$
$T_{All-Erase}$	32K Flash ROM Erase Time	-	25	-	ms	$T_A = +25^{\circ}\text{C}$
T_{Write}	Byte Program Time	-	30	-	μs	$T_A = +25^{\circ}\text{C}$

21.4 LVR Characteristics

($V_{DD} = 5\text{V}$, $T_A = +25^{\circ}\text{C}$, Unless otherwise indicated)

Symbol	Parameter	Min Value	Typical	Max Value	Unit	Conditions
V_{LVR1}	LVR Threshold voltage 4.3V	4.22	4.30	4.38	V	$\text{LVRS}[1:0] = 11$
V_{LVR2}	LVR Threshold voltage 3.7V	3.62	3.70	3.77	V	$\text{LVRS}[1:0] = 10$
V_{LVR3}	LVR Threshold voltage 2.9V	2.84	2.90	2.95	V	$\text{LVRS}[1:0] = 01$
V_{LVR4}	LVR Threshold voltage 1.9V	1.86	1.90	1.93	V	$\text{LVRS}[1:0] = 00$

21.5 DC Characteristics

($V_{DD} = 5V, T_A = +25^{\circ}C$, Unless otherwise specified)

Symbol	Parameter	Minimum	Typical value	Maximum	Unit	Test Conditions
Current						
I_{op1}	Operating current	-	4.5	-	mA	$f_{sys}=32MHz$
I_{op2}	Operating current	-	3	-	mA	$f_{sys}=16MHz$
I_{op3}	Operating current	-	2	-	mA	$f_{sys}=8MHz$
I_{op4}	Operating current	-	1.6	-	mA	$f_{sys}=4MHz$
I_{pd1}	Stand-by current (Power Down Mode)	-	4	8	μA	
I_{IDL1}	Stand-by current (IDLE Mode)	-	2	-	mA	$f_{sys}=32MHz$
I_{BTM}	Base Timer Operating current	-	6	10	μA	BTMFS[3: 0]= 1000 Generate an interrupt every 4.0 seconds
I_{WDT}	WDT current	-	6	10	μA	WDTCKS[2: 0]= 000 WDT overflow time 500ms
I_{TK1}	TK operating current	-	0.4	0.6	mA	

	(Highly reliable)					
I_{TK2}	TK operating current (High sensitivity)	-	0.8	1.2	mA	
IO port characteristics						
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	Schmitt trigger input: RST/tCK/SCK
V_{IL2}	Input low voltage	-0.2	-	$0.2V_{DD}$	V	
I_{OL1}	Output low current	-	29	-	mA	$V_{Pin}=0.4V$
I_{OL2}	Output low current	-	50	-	mA	$V_{Pin}=0.8V$
I_{OH1}	Output high current P05/P20	-	20	-	mA	$V_{Pin}=4.3V$ $P_{xyz}=0, I_{OH}$ level 0
	Output high current (except P05/P20)	-	10	-	mA	$V_{Pin}=4.3V$ $P_{xyz}=0, I_{OH}$ level 0
	Output high current	-	7	-	mA	$V_{Pin}=4.3V$ $P_{xyz}=1, I_{OH}$ level 1
	Output high current	-	5	-	mA	$V_{Pin}=4.3V$ $P_{xyz}=2, I_{OH}$ level 2
	Output high current	-	2.5	-	mA	$V_{Pin}=4.3V$ $P_{xyz}=3, I_{OH}$ level 3
I_{OH2}	Output high current P05/P20	-	8	-	mA	$V_{Pin}=4.7V$

						Pxyz=0,I _{OH} level 0
	Output high current (except P05/P20)	-	4	-	mA	V _{PIN} =4.7V Pxyz=0,I _{OH} level 0
	Output high current	-	3	-	mA	V _{PIN} =4.7V Pxyz=1,I _{OH} level 1
	Output high current	-	2	-	mA	V _{PIN} =4.7V Pxyz=2,I _{OH} level 2
	Output high current	-	1	-	mA	V _{PIN} =4.7V Pxyz=3,I _{OH} level 3
R _{PH1}	Pull-up resistor	-	30	-	kΩ	
ADC Internal reference for reference voltage 2.048V						
V _{REF1}	Internal reference 2.048V voltage output	2.028	2.048	2.068	V	
ADC Internal reference for reference voltage 1.024V						
V _{REF2}	Internal reference 1.024V voltage output	1.014	1.024	1.034	V	

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

Symbol	Parameters	Min Value	Typical value	Max Value	Unit	Test condition
Current						
I _{op5}	Operating current	-	4.5	-	mA	f _{SYS} =32MHz

I _{op6}	Operating current	-	3	-	mA	f _{sys} =16MHz
I _{op7}	Operating current	-	2	-	mA	f _{sys} =8MHz
I _{op8}	Operating current		1.5	-	mA	f _{sys} =4MHz
I _{pd2}	Stand-by current(Power Down Mode)	-	4	8	uA	
I _{IDL2}	Stand-by current (IDLE Mode)	-	2.5	-	mA	f _{sys} =32MHz
I _{TK3}	TK operating current (Highly reliable)	-	0.3	0.5	mA	
I _{TK4}	TK operating current (High sensitivity)	-	0.7	1.0	mA	
IO port characteristics						
V _{IH3}	Input high voltage	0.7V _{DD}	-	V _{DD} +0.3	V	
V _{IL3}	Input low voltage	-0.3	-	0.3V _{DD}	V	
V _{IH4}	Input high voltage	0.8V _{DD}	-	V _{DD}	V	Schmitt trigger input: RST/tCK/SCK
V _{IL4}	Input low voltage	-0.2	-	0.2V _{DD}	V	
I _{OL3}	Output low current	-	20	-	mA	V _{Pin} =0.4V
I _{OL4}	Output low current	-	35	-	mA	V _{Pin} =0.8V
I _{OH3}	Output high current (except P05/P20)	-	3	-	mA	V _{Pin} =3.0V
I _{OH4}	Output high current P05/P20	-	6	-	mA	V _{Pin} =3.0V

R _{PH2}	Pull-up resistor	-	55	-	kΩ	
V _{REF3}	Internal reference 2.048V voltage output	2.028	2.048	2.068	V	
V _{REF4}	Internal reference 1.024V voltage output	1.014	1.024	1.034	V	

21.6 AC Characteristics

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

Symbol	Parameters	Min Value	Typical Value	Max Value	Unit	Test condition
T _{Osc}	External 32.768kHz oscillator start-up time	-	-	1	s	External 32.768kHz crystal oscillator
T _{POR}	Power On Reset time	-	15	-	ms	
T _{PDW}	Power Down mode wake-up time	-	65	130	μs	
T _{Reset}	Reset pulse width	18	-	-	μs	Low level valid
f _{HRC1}	RC oscillation stability	31.68	32	32.32	MHz	T _A =-20~85°C
f _{HRC2}	RC oscillation stability	31.36	32	32.64	MHz	T _A =-40~105 °C

21.7 ADC Characteristics

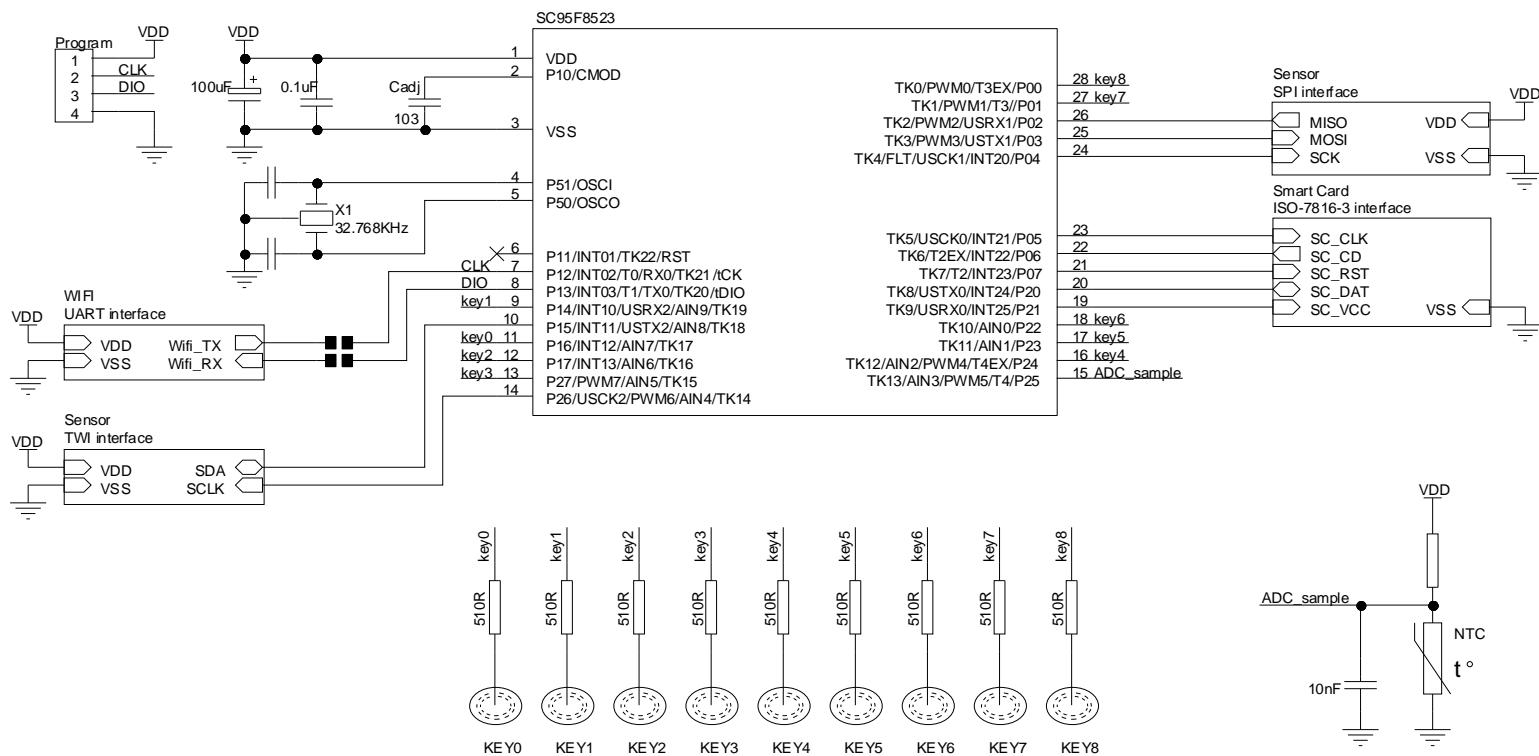
(T_A = 25°C, Unless otherwise indicated)

Symbol	Parameters	Min Value	Typical Value	Max Value	Unit	Condition
--------	------------	-----------	---------------	-----------	------	-----------

V _{AD1}	Supply voltage 1	2.7	5.0	5.5	V	V _{ref} = 2.048V
V _{AD2}	Supply voltage 2	2.4	5.0	5.5	V	V _{ref} = 1.024V or V _{ref} = V _{DD}
N _R	Precision	-	12	-	bit	GND ≤ V _{AIN} ≤ V _{DD}
V _{AIN}	ADC input voltage	GND	-	V _{DD}	V	
R _{AIN}	ADC input resistance	1	-		MΩ	V _{IN} =5V
I _{ADC1}	ADC conversion current 1	-	-	2	mA	ADC module open V _{DD} =5V
I _{ADC2}	ADC conversion current 2	-	-	1.8	mA	ADC module open V _{DD} =3.3V
DNL	Differential Non-Linearity	-	-	±3	LSB	V _{DD} =5V V _{REF} =5V
INL	Integral Non-Linearity	-	-	±3	LSB	
E _Z	Offset error	-	±3	-	LSB	
E _F	Full scale error	-	±1	-	LSB	
E _{AD}	Absolute Accuracy	-	±3	-	LSB	
T _{ADC1}	ADC conversion time 1	-	1.1	1.4	μs	f _{sys} =32MHz LOWSP[2: 0] = 100
T _{ADC2}	ADC conversion time 2	-	1.2	1.5	μs	f _{sys} =32MHz LOWSP[2: 0] = 101

T _{ADC3}	ADC conversion time 3	-	1.5	1.9	μs	f _{sys} =32MHz LOWSP[2: 0] = 110
T _{ADC4}	ADC conversion time 4	-	2.0	2.6	μs	f _{sys} =32MHz LOWSP[2: 0] = 111

22 Application Circuit

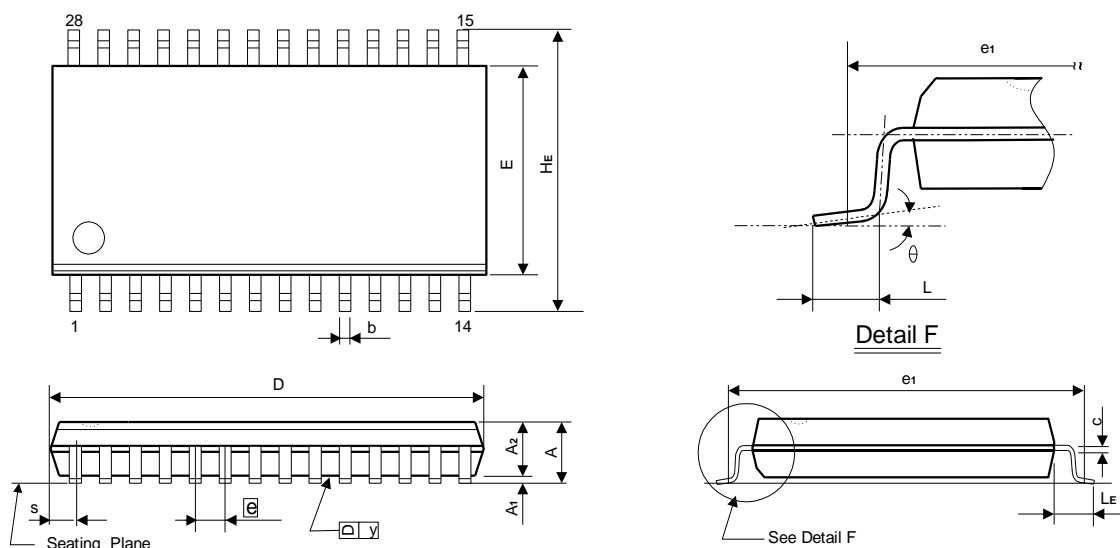


23 Package Information

SC95F8523M28U

SOP28L(300mil) Overall Dimensions

Unit: mm

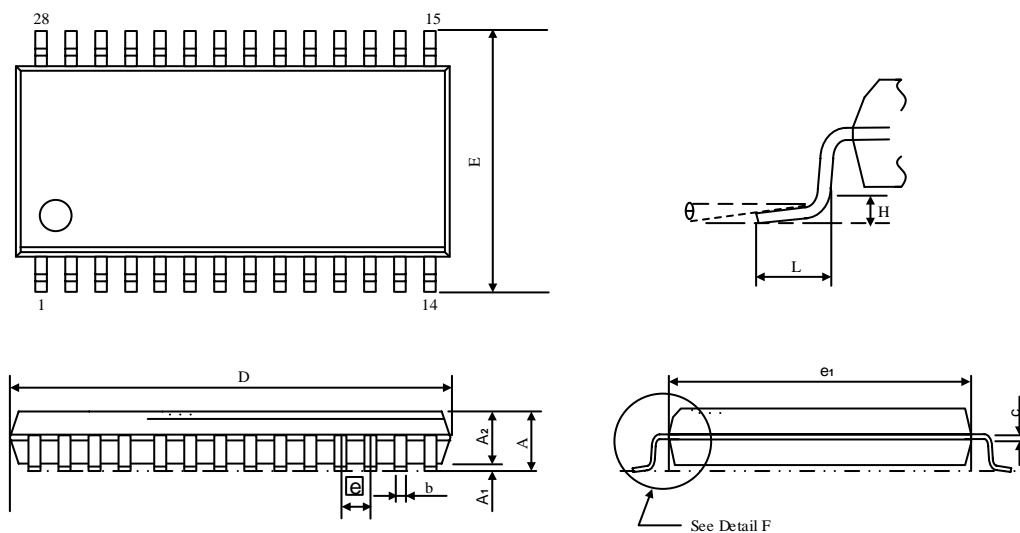


Symbol	mm		
	Min Value	Typical Value	Min Value
A	2.40	2.56	2.65
A1	0.100	0.200	0.300
A2	2.240	2.340	2.440
b	0.39	---	0.48
C	0.254(BSC)		
D	17.80	18.00	18.20
E	7.30	7.50	7.70
HE	10.100	10.300	10.500
e	1.270(BSC)		
L	0.7	0.85	1.0
LE	1.3	1.4	1.5
θ	0°	-	8°

SC95F8523X28U

TSSOP28L Overall Dimensions

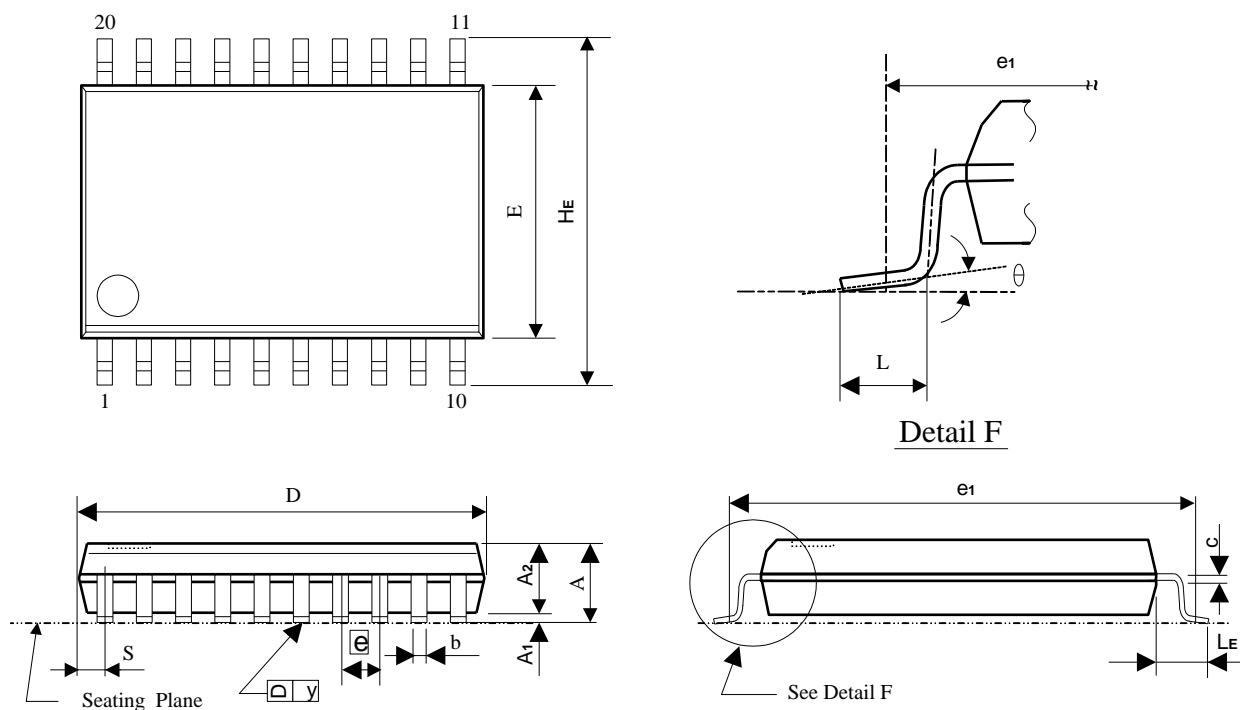
Unit: mm



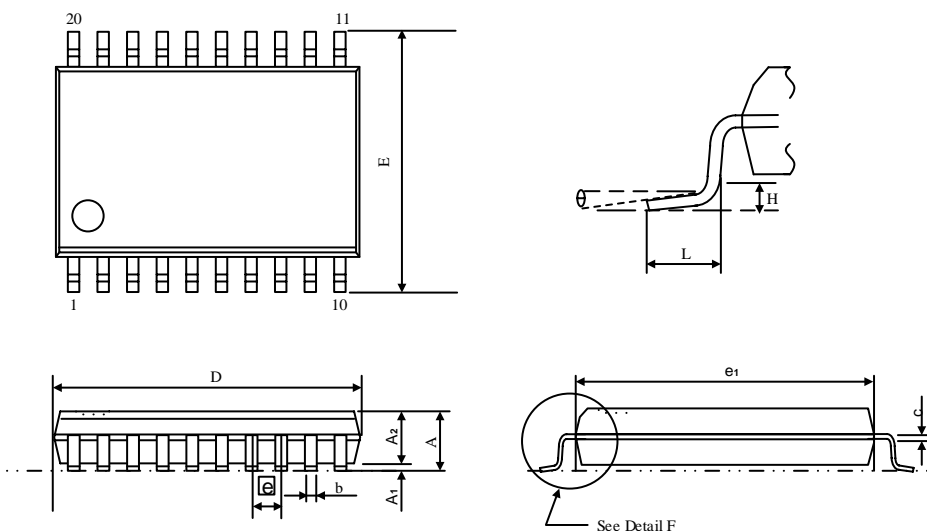
Symbol	mm		
	Min Value	Typical Value	Min Value
A	-	-	1.200
A1	0.050	-	0.150
A2	0.800	-	1.050
b	0.190	-	0.300
c	0.090	-	0.200
D	9.600	-	9.800
E	6.250	-	6.550
e1	4.300	-	4.500
\bar{e}	0.65(BSC)		
L	-	-	1.0
θ	0°	-	8°
H	0.05	-	0.25

SC95F8522M20U
SOP20L Overall Dimensions

Unit: mm



Symbol	mm		
	Min Value	Typical Value	Min Value
A	2.40	2.56	2.65
A1	0.100	0.200	0.300
A2	2.240	2.340	2.440
b	0.35	--	0.47
c	0.25	--	0.31
D	12.60	12.80	13.00
E	7.30	7.50	7.70
HE	10.100	10.300	10.500
e	1.27(BSC)		
L	0.700	0.850	1.000
LE	1.30	1.40	1.50
θ	0°	-	8°

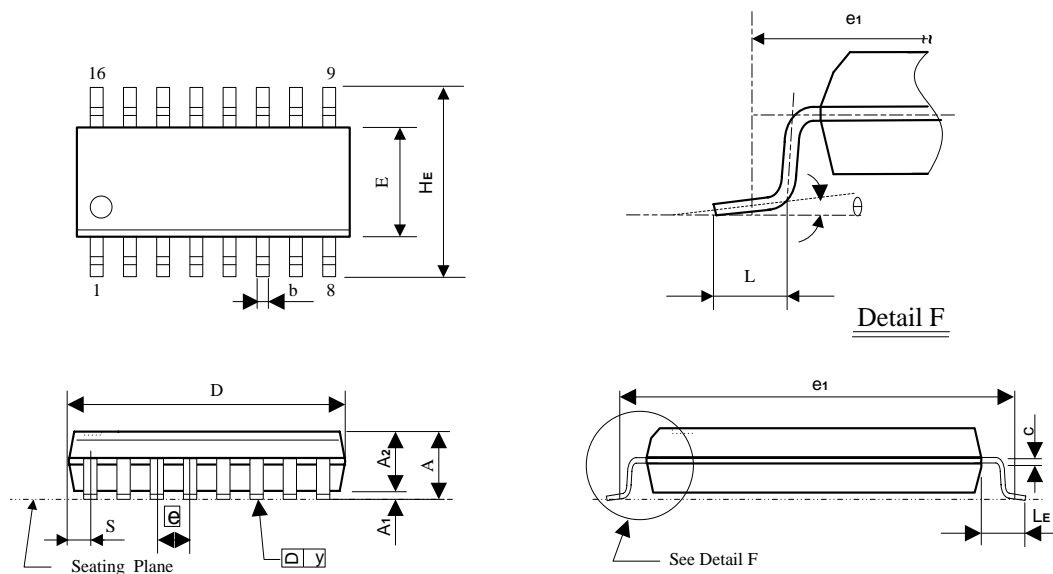
SC95F8522X20U
TSSOP20L Overall Dimensions
Unit: mm


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

SC95F8521M16U

SOP16L(150mil) Overall Dimensions

Unit: mm



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

24 Revision History

Revision	Changes	Date
V1.0	Initial Release.	July 2021