

Super High-speed 1T 8051 Core Flash MCU, 8 Kbytes SRAM, 64 Kbytes Flash, 0~4 Kbytes LDRAM, 6 Kbytes Independent EEPROM, 12-bit ADC, 14-channel 16-bit PWM, 5 Timers, MDU, UART, 6-channel USCI, CRC Check Module, 1 Analog Comparator

---

## 1 General Description

SC95F7617B/7616B/7615B/7613B/7612B (hereinafter referred to as the SC95F761XB) is a series of enhanced 1T 8051 core industry-standard Flash Microcontroller unit (MCU), the instruction set is compatible with the standard 8051 series.

The SC95F761XB has a Super-high-speed 1T 8051 CPU core with an operating frequency of up to 32 MHz.

The SC95F761XB integrates a hardware multiplier and divider hardware CRC and dual DPTR data pointers to accelerate data operations and movement speed. The hardware multiplier and divider and hardware CRC does not occupy CPU cycles, and the operation is implemented by hardware, and the speed is faster than the multiplication and division speed realized by software; dual DPTR data pointers can be used to accelerate data storage and movement.

The SC95F761XB 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 has good ESD performance and EFT anti-interference ability. Using the industry-leading eFlash process, the Flash can be written more than 100,000 times, and can be stored for 100 years at room temperature.

The SC95F761XB 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 6mA@32M at 5V.

The SC95F761XB series is also integrated with super rich hardware resources: 64 Kbytes Flash ROM, SRAM: internal 256 bytes+ external 8 Kbytes+ PWM&LCD 80 bytes, 6 Kbytes EEPROM,; up to 46 GP I/O (partially gradable control), 16 IO can be externally interrupted, 5 16-bit timers, 14-channel 16-bit PWM:8-channel multi-function dead zone complementary PWM,6 channel PWM output of timer,1 UART,6 USCI(UART/SPI/IIC), Built-in LCD/LED hardware driver, internal  $\pm 2\%$  high-precision high-frequency 32/16/8/4 MHz oscillator and  $\pm 4\%$  precision low-frequency 32 kHz oscillator, external 32.768 kHz resources such as crystal oscillators. 1 analog comparator, 17 channels 12-bit high precision ADC.

The SC95F761XB 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 SC95F761XB has very excellent anti-jamming performance. It is very suitable for various applications about main control system, such as Intelligent home appliances and Intelligent House System, Internet of things, wireless communication, game consoles and other industrial controls, and Consumer application areas.

## 2 Features

### Operating Conditions

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

### EMS

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

### Package

- 20 PIN: SOP20 / TSSOP20
- 28 PIN: SOP28 / TSSOP28
- 32 PIN: LQFP32 (7X7)
- 44 PIN: LQFP44 (10X10)
- 48 PIN: LQFP48 (7X7)

### 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)

### Flash ROM

- Flash ROM64 Kbytes Flash ROM
- Can be rewritten 100,000 times
- APROM area allowed IAP operation in Flash can be set to 0K/1K/2K/All APROM by Code Option.

### LDROM

- BootLoader code memory

- LDRAM area can be set to 0K/1K/2K/4K by Code Option

**EEPROM**

- Independent 6 Kbytes EEPROM
- Can be rewritten 100,000 times, has more than 100-year preservation life in the ambient temperature of 25°C

**SRAM**

- 256 bytes on-chip direct access RAM
- 8 Kbytes on-chip Indirect access RAM
- 80 bytes PWM&LCD RAM

**Flash Programming and Emulation**

- 2-wire JTAG programming and emulation interface

**System clock ( $f_{sys}$ )**

- Built-in high frequency 32 MHz oscillator ( $f_{HRC}$ )
  - can be selected and set by the programmer as: 32/16/8/4 MHz @ 2.0~5.5V
  - Frequency Error: Within  $\pm 1\%$  @ -40 ~ 85°C @ 2.0 ~ 5.5V
  - Frequency Error: Within  $\pm 2\%$  @ -40 ~ 105°C @ 2.0 ~ 5.5V
  - The system clock can be automatically calibrated by 32.768 kHz external crystal oscillator, after calibration HRC accuracy can be infinitely close to the accuracy of external 32.768 kHz 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 kHz oscillator (LRC):**

- used as the clock source for Base Timer and WDT and wake up STOP
- Frequency Error: After the register correction, within  $\pm 4\%$  @ -20 ~ 85°C @ 4.0 ~ 5.5V

**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)**

- Timer 0~Timer 4, INT0~2, ADC, PWM, UART, USCI0~5, Base Timer, CMP 19 interrupt sources

- External interrupt contains 3 interrupt vectors, 16 interrupt ports. All can set up rising edge, falling edge, dual edge interrupt.
- Two-level interrupt priority capability

#### **Digital Peripheral**

- GPIO: Up to 46 bidirectional independently controllable I/O ports
  - Independent setting of pull-up resistors
  - P0~P3L(P3.0/1/2/3) port source drive capacity is controlled by four levels
  - All IO ports have large sink current drive capability (50mA)
- Built-in WDT, optional clock frequency division ratio
- 5 Timers: Timer0~4, Timer1, Time2, Timer3 and Timer4
  - Time2, Timer3 and Timer4 have Capture function
  - Time2, Timer3 and Timer4 each can provide two conventional PWM
- 6-channel 16-bit conventional PWM
  - Time2, Timer3 and Timer4 each can provide two conventional PWM
- 8-channel 16-bit multi-function PWM
  - Public cycle and the duty cycle can be set separately
  - Complementary PWM waveforms with dead zones can be output
- One independent UART communication port UART0
- Six UART/SPI/TWI communication interfaces (USCI)
  - When USCI0 is set to SPI0, the driving capability of the pins corresponding to its signal port will be enhanced
- Built-in CRC check module
- Integrated with 16 \* 16-bit hardware Multiplier-Divide Unit (MDU)

#### **LCD/LED driver**

- Choose one of two LCD/LED, share registers and IO ports
- 8 X 24, 6 X 26, 5 X 27, or 4X 28 segments LED driver
- LED segment port source drive capability is selectable in four levels
- 8 X 24, 6 X 26, 5 X 27, or 4X 28 segments LCD driver

#### **Analog Peripheral**

- 17-channel 12-bit  $\pm 2$ LSB ADC
  - Build-in reference voltage of 2.048V, 1.024V and 2.4V
  - The ADC reference voltages is optional: VDD, internal 2.048V, internal 1.024V and 2.4V
  - 1 internal channel can measure the voltage of the power supply
  - ADC conversion complete interruption can be set
- 1 Analog Comparator
  - 4-channel input and 1-channel reference voltage input
  - 16-level optional comparison 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 CMP.

## Naming Rules for 95 Series Products

|             |    |    |   |   |   |   |   |   |   |    |   |
|-------------|----|----|---|---|---|---|---|---|---|----|---|
| <b>Name</b> | SC | 95 | F | 7 | 6 | 1 | 7 | X | P | 48 | R |
| <b>S/R</b>  | ①  | ②  | ③ | ④ | ⑤ | ⑥ | ⑦ | ⑧ | ⑨ | ⑩  | ⑪ |

| <b>S/R</b> | <b>Meaning</b>   |
|------------|--|
| 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          | Version:(default, B, C, D)   |
| ⑨          | Package Type: (D: DIP; M: SOP; X: TSSOP; F: QFP; P: LQFP; Q: QFN; K: SKDIP)                              |
| ⑩          | Number of Pins.  |
| ⑪          | Packaging Mode: (U: Tube; R: Tray; T: Reel)  |

## Ordering Information

| PRODUCT ID     | PACKAGE        | PACK |
|----------------|----------------|------|
| SC95F7612BM20U | SOP20          | TUBE |
| SC95F7612BX20U | TSSOP20        | TUBE |
| SC95F7613BM28U | SOP28          | TUBE |
| SC95F7613BX28U | TSSOP28        | TUBE |
| SC95F7615BP32R | LQFP32(7X7)    | TRAY |
| SC95F7616BP44R | LQFP44(10 X10) | TRAY |
| SC95F7617BP48R | LQFP48(7X7)    | TRAY |

## Contents

|   |           |
|---|-----------|
| <b>1 GENERAL DESCRIPTION .....</b>  | <b>1</b>  |
| <b>2 FEATURES.....</b>  | <b>2</b>  |
| <b>NAMING RULES FOR 95 SERIES PRODUCTS.....</b>                           | <b>6</b>  |
| <b>ORDERING INFORMATION .....</b>   | <b>7</b>  |
| <b>CONTENTS .....</b>   | <b>8</b>  |
| <b>3 PIN DESCRIPTION .....</b>  | <b>13</b> |
| <b>3.1 Pin Configuration.....</b>   | <b>13</b> |
| 3.1.1 48/44 Pin Definition.....   | 17        |
| <b>4 INNER BLOCK DIAGRAM.....</b>   | <b>24</b> |
| <b>5 FLASH ROM AND SRAM .....</b>   | <b>25</b> |
| <b>5.1 APROM and LDROM .....</b>  | <b>25</b> |
| 5.1.1 Flash ROM Sectors .....   | 26        |
| <b>5.2 6K bytes independent EEPROM.....</b>                               | <b>26</b> |
| <b>5.3 96 bits Unique ID Area.....</b>                                    | <b>27</b> |
| 5.3.1 Unique ID Read Operating Demo Program In C Language .....           | 28        |
| <b>5.4 User ID Area.....</b>  | <b>28</b> |
| <b>5.5 Programming.....</b>   | <b>29</b> |
| <b>5.6 In Application Programming (IAP) .....</b>                         | <b>30</b> |
| 5.6.1 IAP Operation Related Register .....                                | 32        |
| 5.6.2 IAP Operation process.....  | 35        |
| 5.6.3 IAP Operating Demo Program In C Language .....                      | 35        |
| 5.6.4 IAP Operation notes .....   | 36        |
| <b>5.7 BootLoader .....</b>   | <b>37</b> |
| 5.7.1 BootLoader Mode operation related registers .....                   | 37        |
| <b>5.8 Encryption .....</b>   | <b>42</b> |
| <b>5.9 Code Option Area (User Programming Settings) .....</b>             | <b>44</b> |
| 5.9.1 Customer-Option-related Registers Operation Instructions .....      | 48        |
| <b>5.10 SRAM.....</b>   | <b>49</b> |
| 5.10.1 Internal 256 Bytes SRAM .....                                      | 49        |
| 5.10.2 External 8 Kbytes SRAM .....                                       | 51        |
| 5.10.3 External 80 bytes PWM&LCD/LED SRAM .....                           | 51        |
| <b>6 SPECIAL FUNCTION REGISTER (SFR).....</b>                             | <b>52</b> |
| <b>6.1 SFR Mapping .....</b>  | <b>52</b> |
| <b>6.2 SFR Instructions .....</b>   | <b>53</b> |
| 6.2.1 SFR.....  | 53        |
| 6.2.2 PWM0 Duty Cycle Adjustment Register(R/W) .....                      | 60        |
| 6.2.3 PWM2~4 Duty Cycle Adjustment Register(R/W).....                     | 60        |
| 6.2.4 LCD/LED display RAM configuration (R/W) .....                       | 61        |
| 6.2.5 Introduction of Common Special Function Registers of 8051 Core..... | 62        |



|  |           |
|--|-----------|
| Program Counter PC .....   | 62        |
| Accumulator ACC (E0H) .....  | 62        |
| B Register (F0H) .....   | 62        |
| Stack Pointer SP (81H) .....   | 63        |
| Data Pointers DPTR0 (82H, 83H), DPTR1 (84H, 85H) And Its Selection Register DPS (86H) .....  | 64        |
| <b>7 POWER, RESET AND SYSTEM CLOCK .....</b>   | <b>67</b> |
| <b>7.1 Power Circuit .....</b>   | <b>67</b> |
| <b>7.2 Power-on Reset .....</b>  | <b>67</b> |
| 7.2.1 Reset Stage .....  | 67        |
| 7.2.2 Loading Information Stage .....  | 67        |
| 7.2.3 Normal Operation Stage .....   | 67        |
| <b>7.3 Reset Modes .....</b>   | <b>67</b> |
| 7.3.1 External Reset .....   | 68        |
| 7.3.2 Low-voltage Reset LVR .....  | 68        |
| 7.3.3 Power-on Reset (POR) .....   | 69        |
| 7.3.4 Watchdog Reset (WDT) .....   | 69        |
| 7.3.5 Software Reset .....   | 71        |
| 7.3.6 Register Reset Value .....   | 72        |
| <b>7.4 High- frequency System Clock Circuit .....</b>  | <b>72</b> |
| <b>7.5 Low- frequency RC Oscillator and Low- frequency Clock Timer .....</b>   | <b>75</b> |
| <b>7.6 Power Saving Modes .....</b>  | <b>78</b> |
| <b>8 CPU AND INSTRUCTION SET .....</b>   | <b>80</b> |
| <b>8.1 CPU .....</b>   | <b>80</b> |
| <b>8.2 Addressing Mode .....</b>   | <b>80</b> |
| 8.2.1 Immediate Addressing .....   | 80        |
| 8.2.2 Direct Addressing .....  | 80        |
| 8.2.3 Indirect Addressing .....  | 80        |
| 8.2.4 Register Addressing .....  | 81        |
| 8.2.5 Relative Addressing .....  | 81        |
| 8.2.6 Indexed Addressing .....   | 81        |
| 8.2.7 Bits Addressing .....  | 81        |
| <b>9 INTERRUPTS .....</b>  | <b>81</b> |
| <b>9.1 Interrupt Source and Vector .....</b>   | <b>82</b> |
| <b>9.2 Interrupt Structure Diagram .....</b>   | <b>85</b> |
| <b>9.3 Interrupt Priority .....</b>  | <b>86</b> |
| The SC95F761XB micro controller unit has two-level interrupt priority capability. The interrupt requests of these interrupt sources can be programmed as high-priority interrupt or low-priority interrupt, which is to realize the nesting of two levels of interrupt service programs. One interrupt can be interrupted by a higher priority interrupt request when being responded to, which can not be interrupted by another interrupt request at the same priority level, until such response to the first-come interrupt ends up with the instruction “RET”. Exist the interrupt service routine and return to main program, the system would execute one more instruction before responding to new interrupt request. .... | 86        |
| That is to say: .....  | 86        |
| ①A lower priority interrupt can be interrupted by a higher priority interrupt request, but not vice verse; .....   | 86        |

|   |            |
|---|------------|
| ②Any kind of interrupt being responded to can not be interrupted by another interrupt request at the same priority level..... | 86         |
| 9.4 Interrupt Processing Flow.....  | 86         |
| 9.5 Interrupt-related SFR Registers .....   | 86         |
| <b>10 TIMER/COUNTER T0 AND T1 .....</b>   | <b>99</b>  |
| 10.1 T0 and T1-related Registers.....   | 99         |
| 10.2 T0 Operating Modes .....   | 104        |
| 10.3 T1 Operating Mode .....  | 106        |
| <b>11 TIMER/COUNTER T2/T3/T4.....</b>   | <b>108</b> |
| 11.1 T2/3/4-related Registers .....   | 108        |
| 11.2 Timer 2 .....  | 109        |
| 11.3 Timer 3 .....  | 113        |
| 11.4 Timer 4 .....  | 117        |
| 11.5 Timer 2/3/4 Operating Modes.....   | 120        |
| 11.5.1 Timer 2/3/4 Operating Modes.....   | 121        |
| <b>12 PWM2/3/4 .....</b>  | <b>126</b> |
| 12.1 PWM2/3/4 related Registers .....   | 126        |
| 12.2 PWM2/3/4 Duty Variation Characteristics.....   | 130        |
| 12.3 PWM2/3/4 Cycle Variation Characteristics.....  | 130        |
| <b>13 PWM0 .....</b>  | <b>131</b> |
| 13.1 PWM0 Structure Diagram.....  | 131        |
| 13.2 PWM0 General Configuration Register.....   | 132        |
| 13.2.1 PWM0 General Configuration Register .....  | 132        |
| 13.2.2 PWM0 Fault Detection Function Setting.....   | 138        |
| 13.3 PWM0 Independent Mode .....  | 141        |
| 13.3.1 PWM0 Independent Mode Block Diagram .....  | 141        |
| 13.3.2 PWM0 Independent Mode Duty Cycle Configuration .....   | 142        |
| 13.4 PWM0 Complementary Model .....   | 143        |
| 13.4.1 PWM0 Block Diagram of Complementary Mode .....   | 143        |
| 13.4.2 PWM Complementary Mode Duty Cycle Configuration .....  | 143        |
| 13.4.3 PWM Complementary Mode Dead Time Setting.....  | 145        |
| 13.4.4 PWM0 Dead Zone Output Waveform.....  | 145        |
| 13.5 PWM0 Waveforms and Directions.....   | 146        |
| <b>14 GENERAL-PURPOSE I/O (GPIO).....</b>   | <b>148</b> |
| 14.1 GPIO Structure Diagram.....  | 148        |
| 14.2 I/O Port-related Registers.....  | 150        |
| <b>15 LCD/LED DISPLAY DRIVER .....</b>  | <b>160</b> |
| 15.1 LCD/LED Display Drive-related Registers .....  | 160        |
| 15.2 LCD/LED Display RAM Configuration.....   | 164        |

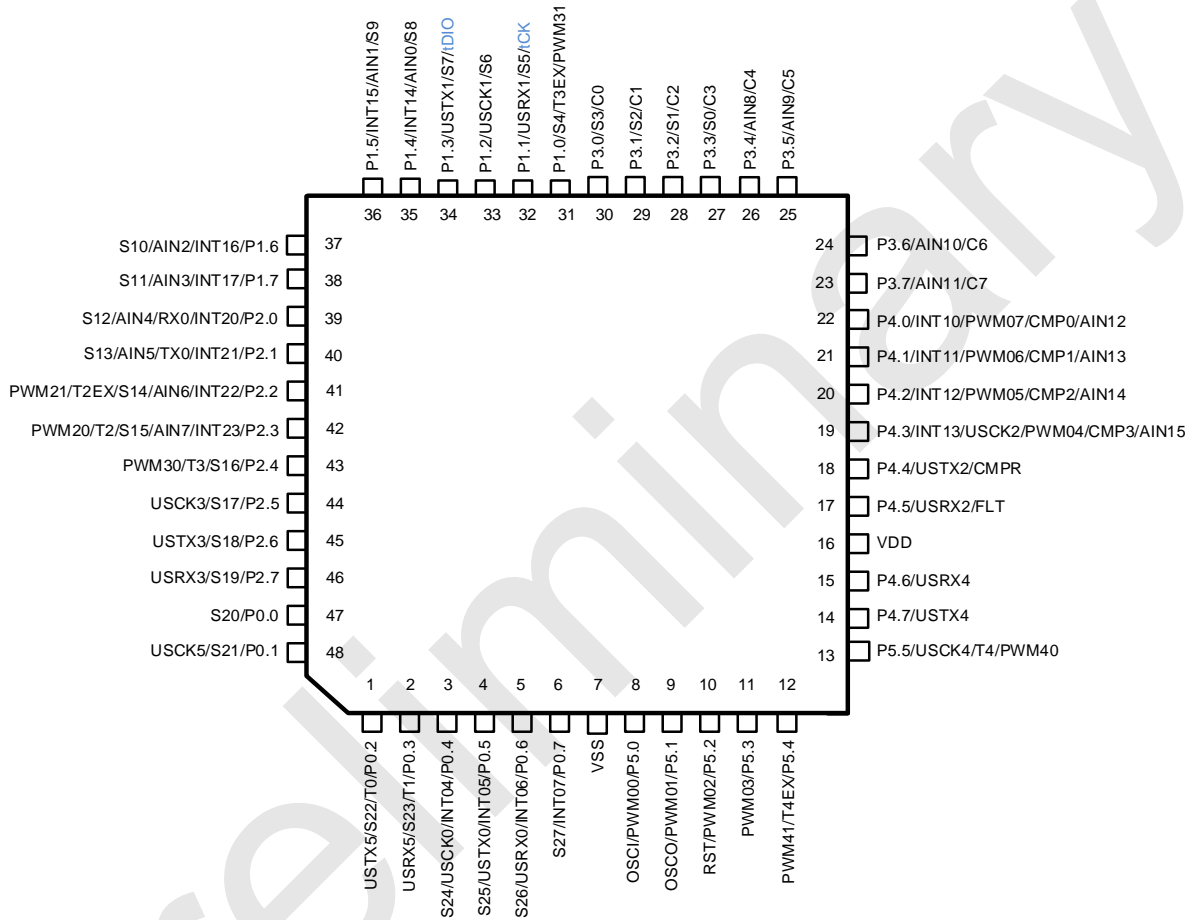
|  |            |
|--|------------|
| <b>15.3 LCD Waveform</b> .....                               | <b>166</b> |
| 15.3.1 1/3Bias LCD Waveform .....                            | 166        |
| 15.3.2 1/4Bias LCD Waveform .....                            | 168        |
| <b>15.4 LED Waveform</b> .....                               | <b>170</b> |
| <b>15.5 LCD/LED Example</b> .....                            | <b>170</b> |
| 15.5.1 LCD Configuration Demo Program.....                   | 170        |
| 15.5.2 LED Configuration Demo Program.....                   | 171        |
| <b>16 SERIAL INTERFACE (UART0)</b> .....                     | <b>172</b> |
| 16.1 Baud Rate of Serial Communication.....                  | 174        |
| <b>17 SPI/TWI/UART SERIAL INTERFACE (USCI)</b> .....         | <b>176</b> |
| <b>17.1 SPI</b> .....  | <b>180</b> |
| 17.1.1 SPI Operation Related Registers .....                 | 181        |
| 17.1.2 Signal Description.....                               | 185        |
| 17.1.3 Operating Modes .....                                 | 186        |
| 17.1.4 Transfer Form .....                                   | 187        |
| 17.1.5 Error Detection.....                                  | 188        |
| <b>17.2 TWI</b> .....  | <b>188</b> |
| 17.2.1 Signal Description.....                               | 194        |
| 17.2.2 Slave Operating Mode.....                             | 195        |
| 17.2.3 Slave Mode Operation Steps.....                       | 197        |
| 17.2.4 Master operating Mode.....                            | 198        |
| 17.2.5 Master Mode Operation Steps.....                      | 199        |
| <b>17.3 Serial Interface (UART)</b> .....                    | <b>201</b> |
| <b>18 HIGH-SPEED ANALOG-TO-DIGITAL CONVERTER (ADC)</b> ..... | <b>205</b> |
| 18.1 ADC-related Registers.....                              | 205        |
| 18.2 ADC Conversion Steps.....                               | 211        |
| <b>19 ANALOG COMPARATOR</b> .....                            | <b>213</b> |
| 19.1 Block Diagram of Analog Comparator.....                 | 213        |
| <b>20 CRC MODULE</b> .....                                   | <b>217</b> |
| 20.1 CRC Check Operation Related Registers.....              | 218        |
| <b>21 MULTIPLIER-DIVIDER UNIT (MDU)</b> .....                | <b>224</b> |
| <b>22 ELECTRICAL CHARACTERISTICS</b> .....                   | <b>227</b> |
| 22.1 Absolute Maximum Ratings.....                           | 227        |
| 22.2 Recommended Operating Conditions .....                  | 227        |
| 22.3 Flash ROM Characteristics .....                         | 228        |
| 22.4 DC Characteristics .....                                | 228        |
| 22.5 AC Characteristics .....                                | 233        |
| 22.6 ADC Characteristics .....                               | 234        |
| 22.7 Analog Comparator Characteristics .....                 | 236        |
| <b>23 PACKAGE INFORMATION</b> .....                          | <b>237</b> |
| <b>24 REVISION HISTORY</b> .....                             | <b>248</b> |

**IMPORTANT NOTICE .....249**

Preliminary

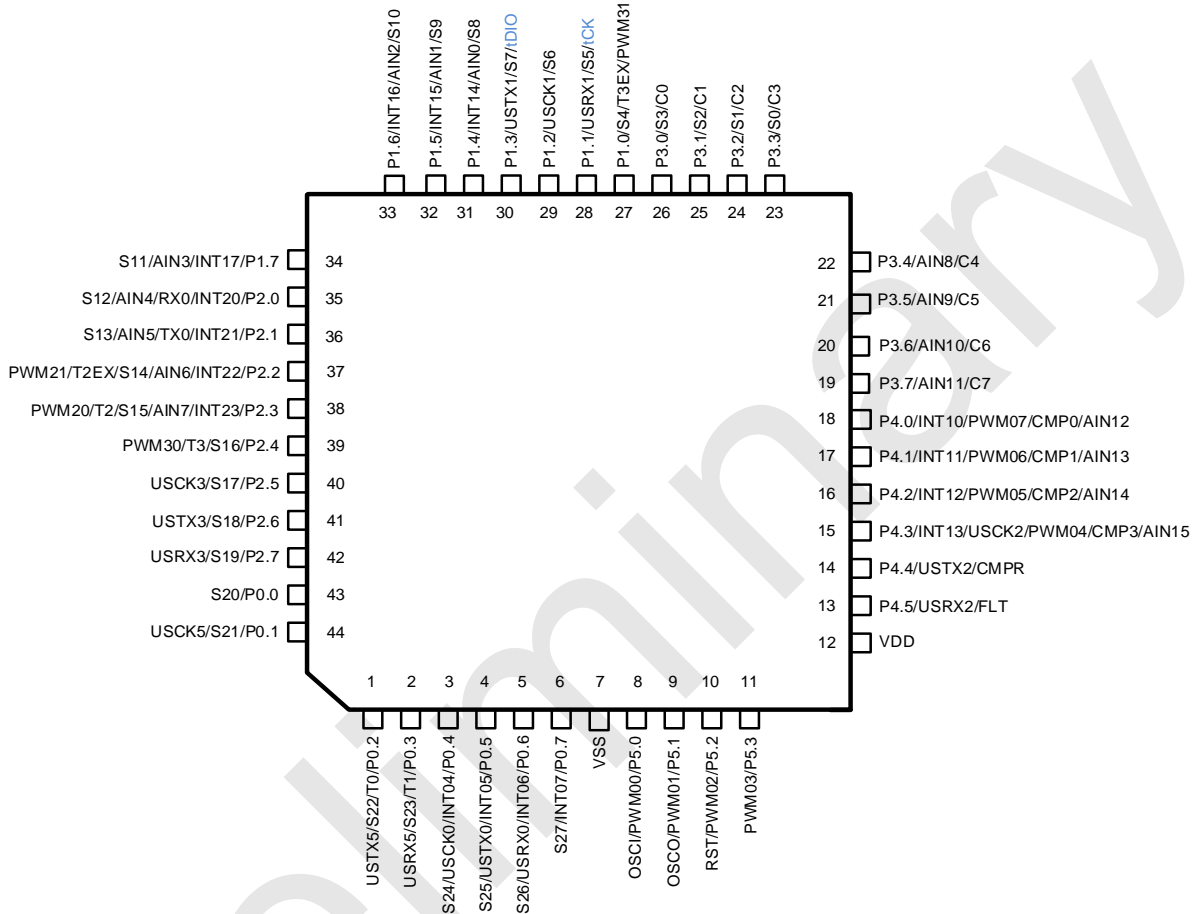
### 3 Pin Description

#### 3.1 Pin Configuration



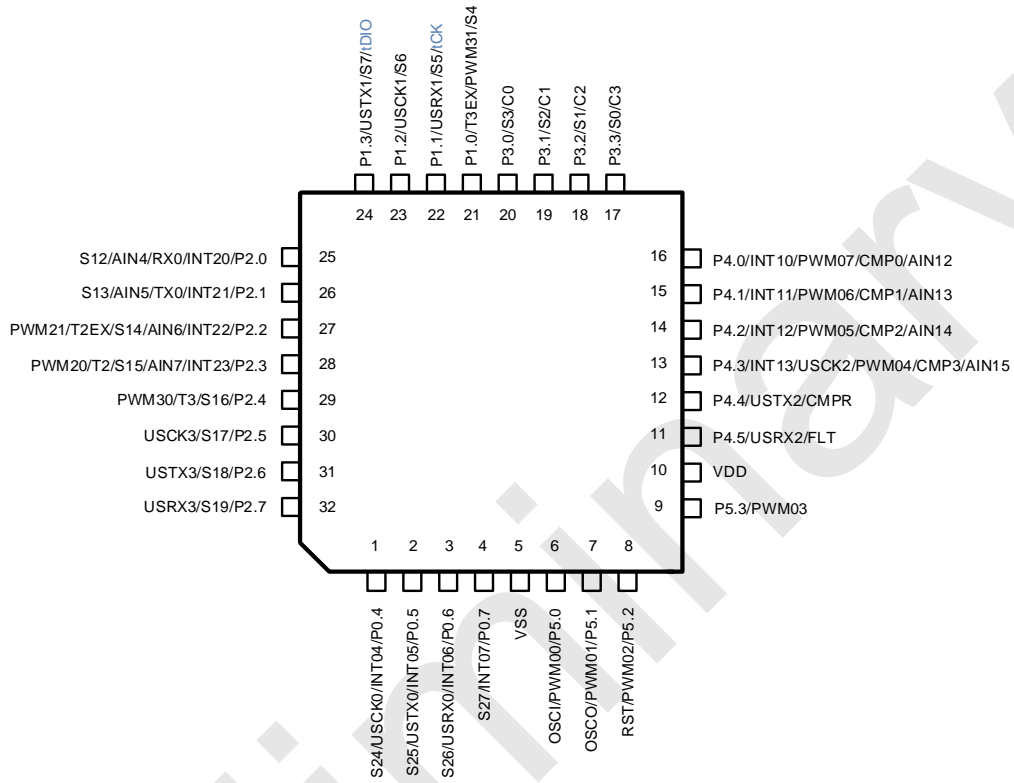
SC95F7617B Pin Diagram

Suitable for LQFP48 package



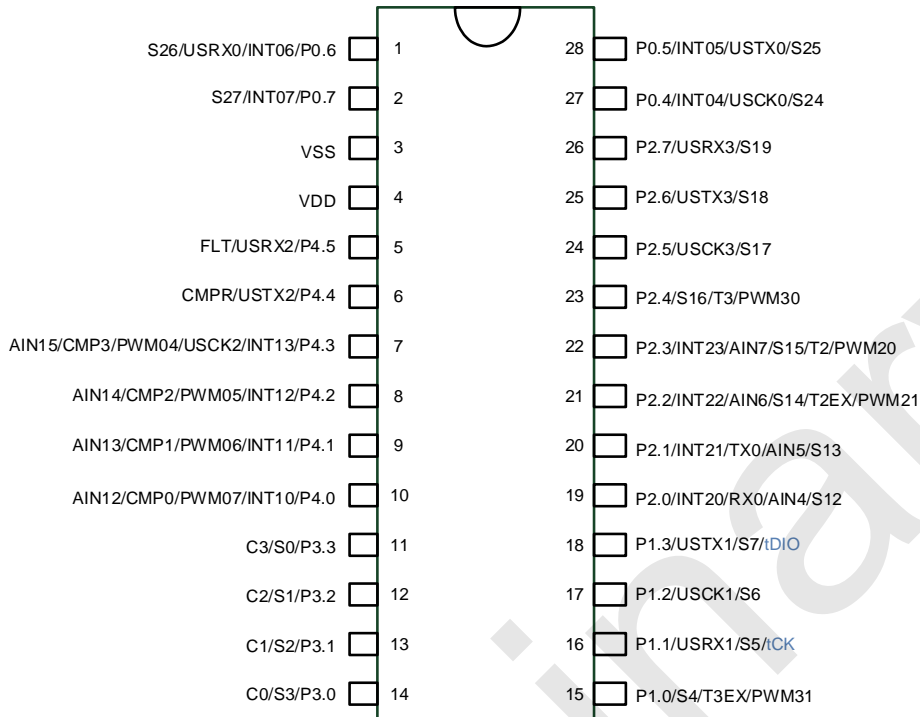
SC95F7616B Pin Diagram

Suitable for LQFP44 package



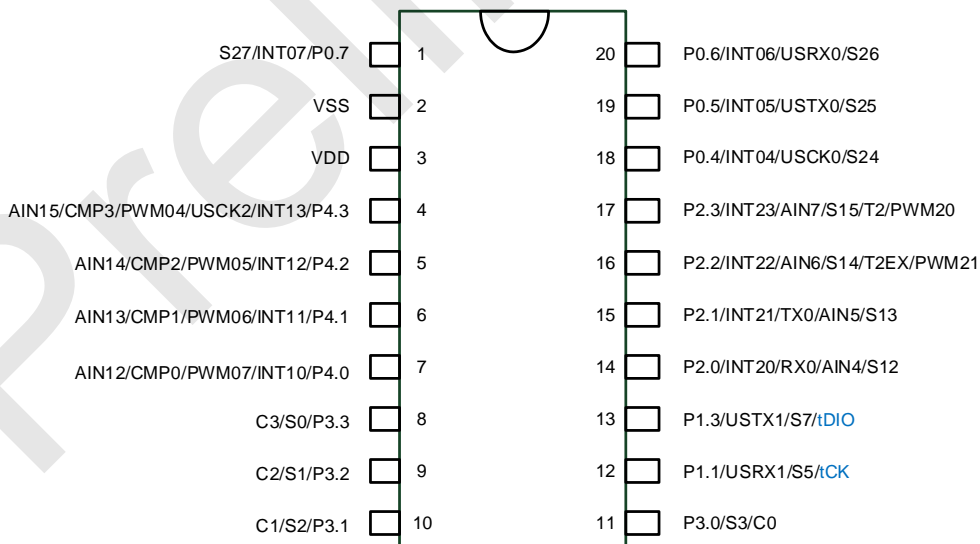
SC95F7615B Pin Diagram

Suitable for LQFP32 package



SC95F7613B Pin Diagram

Suitable for SOP28 and TSSOP28 package



SC95F7612B Pin Diagram

Suitable for SOP20 and TSSOP20 package



**3.1.1 48/44 Pin Definition**

| Pin number |    |    |    |    | Pin Name                     | Type | Description  |
|------------|----|----|----|----|------------------------------|------|--|
| 48         | 44 | 32 | 28 | 20 |                              |      |  |
| 1          | 1  | -  | -  | -  | <b>P0.2/T0/S22/USTX5</b>     | I/O  | P0.2: GPIO P0.2<br>T0: Timer/Counter 0 External Input<br>S22: LCD/LED SEG22<br>USTX5: USCI5 MOSI/SDA/TX    |
| 2          | 2  | -  | -  | -  | <b>P0.3/T1/S23/USRX5</b>     | I/O  | P0.3: GPIO P0.3<br>T1: Timer/Counter 1 External Input<br>S23: LCD/LED SEG23<br>USRX5: USCI5 MISO/RX        |
| 3          | 3  | 1  | 27 | 18 | <b>P0.4/INT04/ USCK0/S24</b> | I/O  | P0.4: GPIO P0.4<br>INT04: Input 4 of external interrupt<br>0USCK0: USCI0 SCK<br>S24: LCD/LED SEG24         |
| 4          | 4  | 2  | 28 | 19 | <b>P0.5/INT05/ USTX0/S25</b> | I/O  | P0.5: GPIO P0.5<br>INT05: Input 5 of external interrupt<br>0USTX0: USCI0 MOSI/SDA/TX<br>S25: LCD/LED SEG25 |
| 5          | 5  | 3  | 1  | 20 | <b>P0.6/INT06/USRX0/S26</b>  | I/O  | P0.6: GPIO P0.6<br>INT06: Input 6 of external interrupt 0<br>USRX0: USCI0 MISO/RX<br>S26: LCD/LED SEG26    |
| 6          | 6  | 4  | 2  | 1  | <b>P0.7/INT07/S27</b>        | I/O  | P0.7: GPIO P0.7<br>INT07: Input 7 of external interrupt 0  |

|    |    |   |   |   |                     |       |   |
|----|----|---|---|---|---------------------|-------|---|
|    |    |   |   |   |                     |       | S27: LCD/LED SEG27  |
| 7  | 7  | 5 | 3 | 2 | VSS                 | Power | Ground  |
| 8  | 8  | 6 | - | - | P5.0/PWM00/OSCI     | I/O   | P5.0: GPIO P5.0<br>PWM00: PWM00 Output<br>OSCI: 32K oscillator output                           |
| 9  | 9  | 7 | - | - | P5.1/PWM01/OSCO     | I/O   | P5.1: GPIO P5.1<br>PWM01: PWM01 Output<br>OSCO: 32K oscillator output                           |
| 10 | 10 | 8 | - | - | P5.2/PWM02/RST      | I/O   | P5.2: GPIO P5.2<br>PWM02: PWM02 Output<br>RST: Reset pin  |
| 11 | 11 | 9 | - | - | P5.3/PWM03          | I/O   | P5.3: GPIO P5.3<br>PWM03: PWM03 Output  |
| 12 | -  | - | - | - | P5.4/T4EX/PWM41     | I/O   | P5.4: GPIO P5.4<br>T4EX: External capture for Timer4<br>PWM41: PWM41 Output                     |
| 13 | -  | - | - | - | P5.5/USCK4/T4/PWM40 | I/O   | P5.5: GPIO P5.5<br>USCK4: USCI4 SCK<br>T4: Timer/Counter4 external input<br>PWM40: PWM40 Output |
| 14 | -  | - | - | - | P4.7/USTX4          | I/O   | P4.7: GPIO P4.7<br>USTX4: USCI4 MOSI/SDA/TX   |
| 15 | -  | - | - | - | P4.6/USRX4          | I/O   | P4.6: GPIO P4.6   |

|    |    |    |   |   |                                   |       |   |
|----|----|----|---|---|-----------------------------------|-------|---|
|    |    |    |   |   |                                   |       | USRX4: USCI4 MISO/RX  |
| 16 | 12 | 10 | 4 | 3 | VDD                               | Power | Power   |
| 17 | 13 | 11 | 5 | - | P4.5/USRX2/FLT                    | I/O   | P4.5: GPIO P4.5<br>USRX2: MISO/RX of USCI2<br>FLT: PWM1 fault detection input pin   |
| 18 | 14 | 12 | 6 | - | P4.4/USTX2/CMPR                   | I/O   | P4.4: GPIO P4.4<br>USTX2: MOSI/SDA/TX of USCI2<br>CMPR: The reference voltage of comparator input.  |
| 19 | 15 | 13 | 7 | 4 | P4.3/INT13/USCK2/PWM04/CMP3/AIN15 | I/O   | P4.3: GPIO P4.3<br>INT13: Input 3 of external interrupt 1<br>USCK2: SCK of USCI2<br>PWM04: PWM04 Output<br>CMP3: Analog comparator input channel 3<br>AIN15: ADC input channel 15 |
| 20 | 16 | 14 | 8 | 5 | P4.2/INT12/PWM05/CMP2/AIN14       | I/O   | P4.2: GPIO P4.2<br>INT12: Input 2 of external interrupt 1<br>PWM05: PWM05 Output<br>CMP2: Analog comparator input channel 2<br>AIN14: ADC input channel 14                        |
| 21 | 17 | 15 | 9 | 6 | P4.1/INT11/PWM06/CMP1/AIN13       | I/O   | P4.1: GPIO P4.1<br>INT11: Input 1 of external interrupt 1<br>PWM06: PWM06 Output<br>CMP1: Analog comparator input channel 1   |

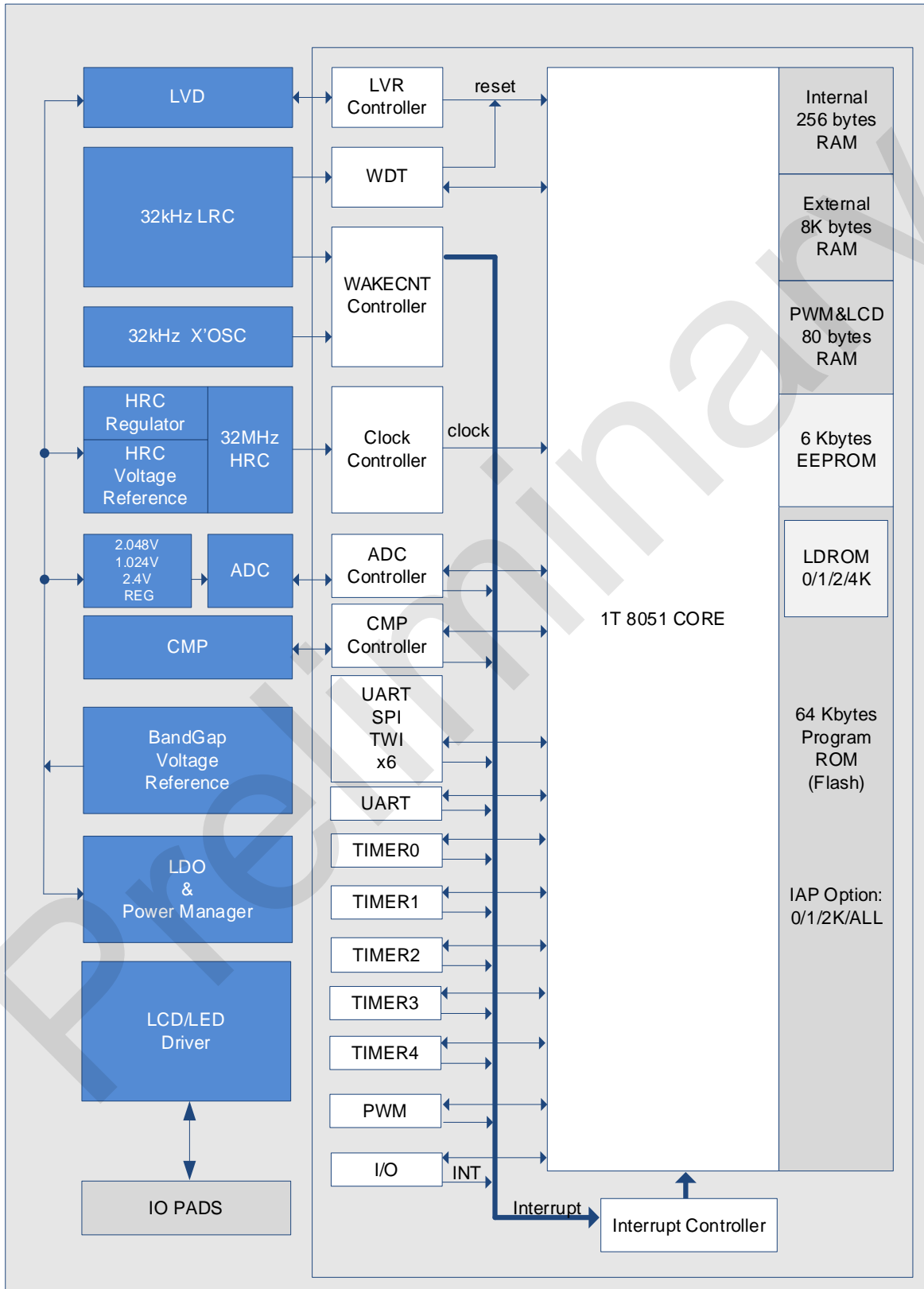
|    |    |    |    |   |                                 |     |  |
|----|----|----|----|---|---------------------------------|-----|--|
|    |    |    |    |   |                                 |     | AIN13: ADC input channel 13  |
| 22 | 18 | 16 | 10 | 7 | P4.0/INT10/PWM07/CMP0/<br>AIN12 | I/O | P4.0: GPIO P4.0<br>INT10: Input 0 of external interrupt 1<br>PWM07: PWM07 Output<br>CMP0: Analog comparator input channel 0<br>AIN12: ADC input channel 12 |
| 23 | 19 | -  | -  | - | P3.7/AIN11/C7                   | I/O | P3.7: GPIO P3.7<br>AIN11: ADC Input Channel 11<br>C7: LCD/LED common output 7  |
| 24 | 20 | -  | -  | - | P3.6/AIN10/C6                   | I/O | P3.6: GPIO P3.6<br>AIN10: ADC input channel 10<br>C6: LCD/LED common output 6  |
| 25 | 21 | -  | -  | - | P3.5/AIN9/C5                    | I/O | P3.5: GPIO P3.5<br>AIN9: ADC input channel 9<br>C5: LCD/LED common output 5  |
| 26 | 22 | -  | -  | - | P3.4/AIN8/C4                    | I/O | P3.4: GPIO P3.4<br>AIN9: ADC input channel 8<br>C4: LCD/LED common output 4  |
| 27 | 23 | 17 | 11 | 8 | P3.3/S0/C3                      | I/O | P3.3: GPIO P3.3<br>S0: LCD/LED SEG 0<br>C3: LCD/LED common output 3  |
| 28 | 24 | 18 | 11 | 8 | P3.2/S1/C2                      | I/O | P3.2: GPIO P3.2<br>S1: LCD/LED SEG 1<br>C2: LCD/LED common output 2  |

|    |    |    |    |    |                    |     |   |
|----|----|----|----|----|--------------------|-----|---|
| 29 | 25 | 19 | 13 | 10 | P3.1/S2/C1         | I/O | P3.1: GPIO P3.1<br>S2: LCD/LED SEG 2<br>C1: LCD/LED common output 1   |
| 30 | 26 | 20 | 14 | 11 | P3.0/S3/C0         | I/O | P3.0: GPIO P3.0<br>S3: LCD/LED SEG 3<br>C0: LCD/LED common output 0   |
| 31 | 27 | 21 | 15 | -  | P1.0/S4/T3EX/PWM31 | I/O | P1.0: GPIO P1.0<br>S4: LCD/LED SEG 4<br>T3EX: External capture for Timer3<br>PWM31: PWM31 Output                |
| 32 | 28 | 22 | 16 | 12 | P1.1/USRX1/S5/tCK  | I/O | P1.1: GPIO P1.1<br>USRX1: MISO/RX of USCI1<br>S5: LCD/LED SEG 5<br>tCK: Programming and Emulation Clock Pin     |
| 33 | 29 | 23 | 17 | -  | P1.2/USCK1/S6      | I/O | P1.2: GPIO P1.2<br>USCK1: SCK of USCI1<br>S6: LCD/LED SEG 6   |
| 34 | 30 | 24 | 18 | 13 | P1.3/USTX1/S7/tDIO | I/O | P1.3: GPIO P1.3<br>USTX1: MOSI/SDA/TX of USCI1<br>S7: LCD/LED SEG 7<br>tDIO: Programming and Emulation Data Pin |
| 35 | 31 | -  | -  | -  | P1.4/INT14/AIN0/S8 | I/O | P1.4: GPIO P1.4<br>INT14: Input 4 of external interrupt 1<br>AIN0: ADC input channel 0<br>S8: LCD/LED SEG 8     |

|    |    |    |    |    |                                       |     |  |
|----|----|----|----|----|---------------------------------------|-----|--|
| 36 | 32 | -  | -  | -  | <b>P1.5/INT15/AIN1/S9</b>             | I/O | P1.5: GPIO P1.5<br>INT15: Input 5 of external interrupt 1<br>AIN1: ADC input channel 1<br>S9: LCD/LED SEG 9  |
| 37 | 33 | -  | -  | -  | <b>P1.6/INT16/AIN2/S10</b>            | I/O | P1.6: GPIO P1.6<br>INT16: Input 6 of external interrupt 1<br>AIN2: ADC Input Channel 2<br>S10: LCD/LED SEG10   |
| 38 | 34 | -  | -  | -  | <b>P1.7/INT17/AIN3/S11</b>            | I/O | P1.7: GPIO P1.7<br>INT17: Input 7 of external interrupt 1<br>AIN3: ADC Input Channel 3<br>S11: LCD/LED SEG11   |
| 39 | 35 | 25 | 19 | 14 | <b>P2.0/INT20/RX0/AIN4/S12</b>        | I/O | P2.0: GPIO P2.0<br>INT20: Input 0 of external interrupt 2<br>RX0: UART0 Receiver<br>AIN4: ADC Input Channel 4<br>S12: LCD/LED SEG12                                      |
| 40 | 36 | 26 | 20 | 15 | <b>P2.1/INT21/TX0/AIN5/S13</b>        | I/O | P2.1: GPIO P2.1<br>INT21: Input 1 of external interrupt 2<br>TX0: UART0 Transmitter<br>AIN5: ADC Input Channel 5<br>S13: LCD/LED SEG13                                   |
| 41 | 37 | 27 | 21 | 16 | <b>P2.2/INT22/AIN6/S14/T2EX/PWM21</b> | I/O | P2.2: GPIO P2.2<br>INT22: Input 2 of external interrupt 2<br>AIN6: ADC Input Channel 6<br>S14: LCD/LED SEG14<br>T2EX: External capture for Timer2<br>PWM21: PWM21 Output |
| 42 | 38 | 28 | 22 | 17 | <b>P2.3/INT23/AIN7/S15/T2/PWM20</b>   | I/O | P2.3: GPIO P2.3<br>INT23: Input 3 of external interrupt 2  |

|    |    |    |    |   |                          |     |   |
|----|----|----|----|---|--------------------------|-----|---|
|    |    |    |    |   |                          |     | AIN7: ADC Input Channel 7<br>S15: LCD/LED SEG15<br>T2: Timer/Counter2 external input<br>PWM20: PWM20 Output |
| 43 | 39 | 29 | 23 | - | <b>P2.4/S16/T3/PWM30</b> | I/O | P2.4: GPIO P2.4<br>S16: LCD/LED SEG 16<br>T3: Timer/Counter3 external input<br>PWM30: PWM30 Output          |
| 44 | 40 | 30 | 24 | - | <b>P2.5/S17/USCK3</b>    | I/O | P2.5: GPIO P2.5<br>S17: LCD/LED SEG 17<br>USCK3:USCI3 SCK   |
| 45 | 41 | 31 | 25 | - | <b>P2.6/S18/USTX3</b>    | I/O | P2.6: GPIO P2.6<br>S18: LCD/LED SEG 18<br>USTX3: USCI3 MOSI/SDA/TX  |
| 46 | 42 | 32 | 26 | - | <b>P2.7/S19/USRX3</b>    | I/O | P2.7: GPIO P2.7<br>S19: LCD/LED SEG 19<br>USRX3: USCI3 MISO/RX  |
| 47 | 43 | -  | -  | - | <b>P0.0/S20</b>          | I/O | P0.0: GPIO P0.0<br>S20: LCD/LED SEG 20  |
| 48 | 44 | -  | -  | - | <b>P0.1/S21/USCK5</b>    | I/O | P0.1: GPIO P0.1<br>S21: LCD/LED SEG 21<br>USCK5: USCI5 SCK  |

## 4 Inner Block Diagram

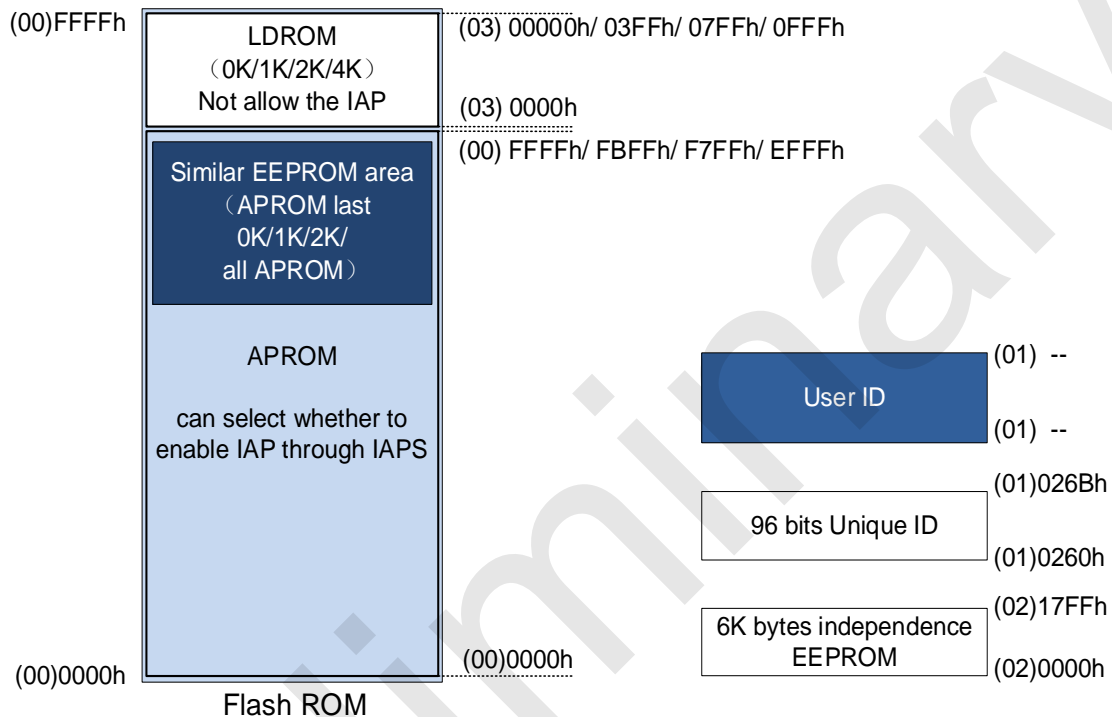


SC95F761XB BLOCK DIAGRAM



## 5 Flash ROM and SRAM

The Flash ROM of SC95F761XB is divided into five regions: APROM/LDROM/EEPROM/User ID/Unique ID, as shown in the following figure:



### 5.1 APROM and LDROM

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

- The extended address of the APROM area is "00". The maximum size of the area is 64 Kbytes. It supports IAP (In Application Programming) and APROM area allowed IAP operation in Flash can be set to 0K/1K/2K/All APROM by Code Option.
- The extended address of LDROM area is "03", area size 0~4 Kbytes optional. IAP on LDROM is not allowed.

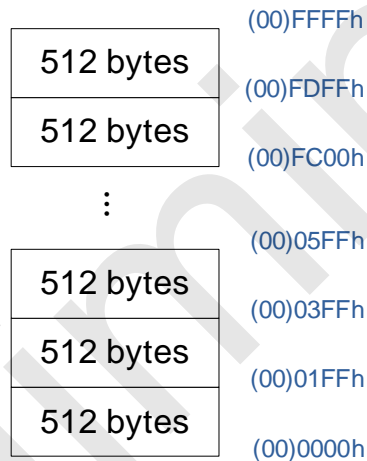
The SC95F761XB has 64 Kbytes of Flash ROM, the address is (00)0000H ~ (00)FFFFH, "00" in brackets is the extended address, which is set by the IAPADE register. Flash ROM can be programmed and erased by SC LINK PRO provided by SinOne. The characteristics of this 64 Kbytes Flash ROM are as follows:

- Divided into 128 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
- 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
- Secure Encryption: Optionally enable APROM (64 Kbytes Flash ROM) and LDROM secure encryption
- Support IAP (In Application Programming).

### 5.1.1 Flash ROM Sectors

The SC95F761XB has 64 Kbytes of Flash ROM divided into 128 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.

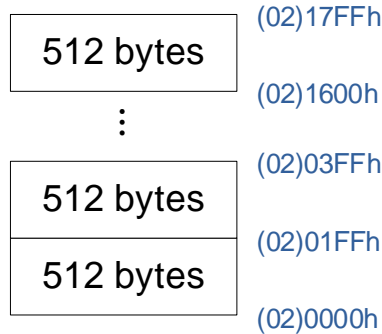


SC95F761XB 64 Kbytes Flash ROM Sectors

### 5.2 6K bytes independent EEPROM

The SC95F761XB has 6 Kbytes of independent EEPROM, the address is (02)000H ~ 17FFH, "02" in brackets is the extended address, which is set by the IAPADE register. Independent EEPROM can be rewritten 100,000 times and the data written-in has more than 100-year preservation life in the ambient temperature of 25°C . EEPROM supports blank checking, programming, verification, erasing and reading functions.

EEPROM divided into 12 sectors,512 bytes per sector



#### SC95F761XB EEPROM Sectors

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

### 5.3 96 bits Unique ID Area

The SC95F761XB 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        | <b>IAPADER[7: 0]</b> | <p>IAP Extended address:</p> <p>0x00: Both MOVC and IAP are for APROM</p> <p>0x01: For the unique ID area, <b>read and write operations are not allowed, otherwise it may cause an exception!</b></p> <p>0x02: Both MOVC and IAP are for EEPROM</p> <p>0x03: This parameter takes effect only when the LDROM program is operating. Programs running in the LDROM region are allowed to perform MOVC operations on the LDROM program region. <b>Note: LDROM operation permission is only for MOVC operation, prohibit IAP operation on LDROM, otherwise it will cause unpredictable exceptions!</b></p> <p>Other: reserved</p> |

### 5.3.1 Unique ID Read Operating Demo Program In C Language

```

#include "intrins.h"

unsigned char UniqueID [12]; //store UniqueID

unsigned char code * POINT =0x0260;

unsigned char i;

EA = 0;           // Disable the global interrupt

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

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

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

EA = 1;         // Enable global interrupt

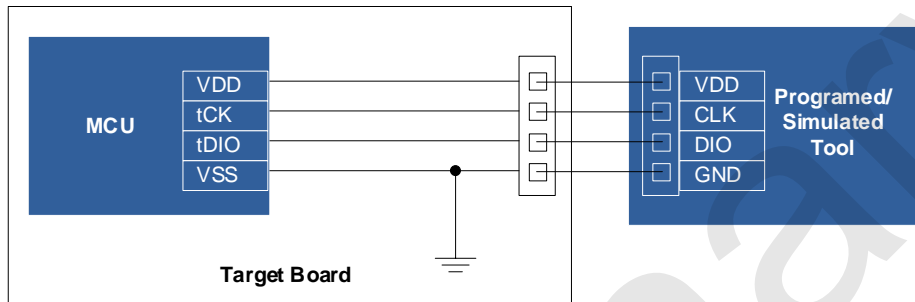
```

## 5.4 User ID Area

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

## 5.5 Programming

The SC95F761XB's Flash ROM can be programmed through tDIO, tCK, VDD, VSS, the specific connection relationship is as follows:



ICP mode Flash Writer programming connection diagram

tDIO,tCK is a 2-wire JTAG programming and emulation signal line. Users can configure the mode of these two ports through the Code Option when programming: JTAG Specific Mode and Normal Mode (JTAG specific port is invalid).

### 5.5.1 JTAG Specific Mode

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

### 5.5.2 Normal Mode (JTAG specific port is invalid)

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

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

**Code Option register:**

**OP\_CTM1 (C2H@FFH) Code Option register1 (read/write)**

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

|              |             |     |       |        |            |     |             |   |
|--------------|-------------|-----|-------|--------|------------|-----|-------------|---|
| Bit Mnemonic | VREFS[1: 0] |     | OP_BL | DISJTG | IAPS[1: 0] |     | LDSIZE[1:0] |   |
| R/W          | R/W         | R/W | R/W   | R/W    | R/W        | R/W | R           |   |
| POR          | n           | n   | n     | n      | n          | n   | n           | n |

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

## 5.6 In Application Programming (IAP)

Application Programming (IAP) operations can be carried out in the APROM of SC95F761XB (0K, 1K, 2K, or all APROM ranges are optional) and 6K bytes EEPROM. Users can implement remote program updates through IAP operations. You can also obtain Unique ID field or User ID field information via IAP reads. Before IAPS write data, you must erase the Sector to which the target address belongs. The length of a Sector is 512 bytes. Flash ROM is divided into 128 sectors from (00)0000H to (00)FFFFH. The "00" in brackets is the expanded address set by the IAPADE register:

|           |                                 |
|-----------|---------------------------------|
| 512 bytes | (00) FFFFh/ FBFFh/ F7FFh/ EFFFh |
| 512 bytes | (00) FDFFh/ F9FFh/ F5FFh/ EDFHh |
| ⋮         |                                 |
| 512 bytes | (00)005FFh                      |
| 512 bytes | (00)003FFh                      |
| 512 bytes | (00)001FFh                      |
| 512 bytes | (00)0000h                       |

SC95F761XB 64 Kbytes Flash ROM Sectors

**NOTE:**

1. During the IAP erase/write process, the CPU holds the program counter, and after the IAP erase/write is complete, the program counter continues to execute subsequent instructions.
2. IAP operation in APROM area has certain risks, users need to take corresponding security measures in the software, if improper operation may cause user program rewriting! This feature is not recommended unless it is required by the user (for example, for remote application updates).
3. The EEPROM erasure count is 100,000. Do not exceed the rated EEPROM erasure count; otherwise, an exception may occur.

The user can select the IAP region range of APROM through Customer Option during programming, or set the IAP region of APROM through IAPS control bit in the program. The relevant registers are as follows: **OP\_CTM1 (C2H@FFH) Code Option Register 1(Read/write)**

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

| Bit Number | Bit Mnemonic      | Description  |
|------------|-------------------|--|
| 3~2        | <b>IAPS[1: 0]</b> | <p>IAP spatial range selection</p> <p>00: Full Flash ROM not 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:</p> <p>1. The above setting items are invalid in BootLoader mode. The BootLoader program can perform IAP operation on the entire Flash ROM area.</p> <p>2. LDROM does not allow IAP operation under any circumstances.</p> |

**5.6.1 IAP Operation Related Register**

| Symbol | Address | Description                            | 7             | 6 | 5          | 4   | 3 | 2 | 1         | 0 | POR       |
|--------|---------|--|---------------|---|------------|-----|---|---|-----------|---|-----------|
| IAPKEY | F1H     | Data protection register               | IAPKEY[7: 0]  |   |            |     |   |   |           |   | 0000000b  |
| IAPADL | F2H     | IAP write address low register         | IAPADR[7: 0]  |   |            |     |   |   |           |   | 0000000b  |
| IAPADH | F3H     | IAP write address high register        | IAPADR[15: 8] |   |            |     |   |   |           |   | 0000000b  |
| IAPADE | F4H     | IAP write to extended address register | IAPADER[7: 0] |   |            |     |   |   |           |   | 0000000b  |
| IAPDAT | F5H     | IAP data register                      | IAPDAT[7: 0]  |   |            |     |   |   |           |   | 0000000b  |
| IAPCTL | F6H     | IAP control register                   | BTL<br>D      | - | SERAS<br>E | PRG | - | - | CMD[1: 0] |   | 0x00xx00b |

**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 | <b>IAPKEY[7: 0]</b> | <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        | <b>IAPADR[7: 0]</b> | 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        | <b>IAPADR[15: 8]</b> | 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        | <b>IAPADER[7: 0]</b> | <p>IAP extended address:</p> <p>0x00: Both MOVC and IAP are for code</p> <p>0x01: The Unique ID area is read but cannot be written</p> <p>0x02: Both MOVC and IAP are for independent EEPROM</p> <p>0x03: MOVC is performed in the LDROM region (Note: only MOVC can be used, not IAP, this item is only valid for LDROM operation, APROM operation this item is not valid!)</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       | R/W | Description         |
|------------|--------------------|-----|---------------------|
| 7~0        | <b>IAPDAT[7:0]</b> | R/W | Data written by IAP |

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

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

**5.6.2 IAP Operation process**

Please refer to the "SC95FXX6X\_SC95FXX1XB IAP Operation Library Documentation Pack" for the IAP operation process of SC95F761XB.

**5.6.3 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: read data:**

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

IAPADE = 0X00;    //The extended address is 0x00, selectFlash ROM

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

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

**IAP operation: erase and write data:**

Please refer to "SC95FXX6X\_SC95FXX1XB IAP Operation Library Kit" if you want to implement IAP erasable operation.

**5.6.4 IAP Operation notes**

1. The user must erase the target sector before writing.
2. IAP operation in APROM area has certain risks, users need to take corresponding security measures in the software, if improper operation may cause user program rewriting! This feature is not recommended unless required (for remote application updates, for example);
3. When programming IC, if "APROM zone prevents IAP operation" is selected through Code Option, then IAPADE [7:0]=0x00 (APROM zone is selected), IAP cannot be operated, that is, data cannot be written, data can only be read by MOVC instruction;
4. When IAPADE is not 0x00, the MOVC and write target is non-APROM region. At this time, if there is an interrupt and MOVC operation occurs in the interrupt, the result of MOVC will be wrong and the program will run abnormally. To avoid this situation, if IAPADE is not 0x00 during IAP operation, it is important to turn off total interrupt (EA=0) before operation, and set IAPADE = 0x00 after operation before turning on total interrupt (EA=1);
5. During the IAP wipe/write process, the CPU holds the program counter, and the program counter only continues to execute subsequent instructions after the IAP wipe/write is complete;
6. The EEPROM erasure count is 100,000. Do not exceed the rated EEPROM erasure count; otherwise, an exception may occur!

## 5.7 BootLoader

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

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

The LDROM has four address ranges:

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

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

### 5.7.1 BootLoader Mode operation related registers

#### OP\_CTM1 (C2H@FFH) Code Option Register1 (Read/Write)

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

| Bit Number | Bit Mnemonic | Description   |
|------------|--------------|---|
| 5          | <b>OP_BL</b> | Program run area control bit<br>0: After the chip is reset, it enters APROM |

|   |                     | <p>1: After the chip is reset, it enters LDROM</p> <p>① The MOVC and IAP restrictions for APROM are as follows:</p> <table border="1" data-bbox="624 461 1436 797"> <thead> <tr> <th>Operation</th> <th>Is it operable?</th> </tr> </thead> <tbody> <tr> <td>LDROM MOVC</td> <td>x</td> </tr> <tr> <td>APROM MOVC</td> <td>√</td> </tr> <tr> <td>LDROM IAP</td> <td>x</td> </tr> <tr> <td>APROM IAP</td> <td>√</td> </tr> </tbody> </table> <p>② The MOVC and IAP restrictions for LDROM are as follows:</p> <table border="1" data-bbox="624 896 1436 1234"> <thead> <tr> <th>Operation</th> <th>Is it operable?</th> </tr> </thead> <tbody> <tr> <td>LDROM MOVC</td> <td>√</td> </tr> <tr> <td>APROM MOVC</td> <td>√</td> </tr> <tr> <td>LDROM IAP</td> <td>x</td> </tr> <tr> <td>ALL APROM IAP, not restricted by IAPRANGE</td> <td>√</td> </tr> </tbody> </table> | Operation | Is it operable? | LDROM MOVC | x | APROM MOVC | √ | LDROM IAP | x | APROM IAP | √ | Operation | Is it operable? | LDROM MOVC | √ | APROM MOVC | √ | LDROM IAP | x | ALL APROM IAP, not restricted by IAPRANGE | √ |
|---|---------------------|---|-----------|-----------------|------------|---|------------|---|-----------|---|-----------|---|-----------|-----------------|------------|---|------------|---|-----------|---|---|---|
| Operation                                 | Is it operable?     |   |           |                 |            |   |            |   |           |   |           |   |           |                 |            |   |            |   |           |   |   |   |
| LDROM MOVC                                | x                   |   |           |                 |            |   |            |   |           |   |           |   |           |                 |            |   |            |   |           |   |   |   |
| APROM MOVC                                | √                   |   |           |                 |            |   |            |   |           |   |           |   |           |                 |            |   |            |   |           |   |   |   |
| LDROM IAP                                 | x                   |   |           |                 |            |   |            |   |           |   |           |   |           |                 |            |   |            |   |           |   |   |   |
| APROM IAP                                 | √                   |   |           |                 |            |   |            |   |           |   |           |   |           |                 |            |   |            |   |           |   |   |   |
| Operation                                 | Is it operable?     |   |           |                 |            |   |            |   |           |   |           |   |           |                 |            |   |            |   |           |   |   |   |
| LDROM MOVC                                | √                   |   |           |                 |            |   |            |   |           |   |           |   |           |                 |            |   |            |   |           |   |   |   |
| APROM MOVC                                | √                   |   |           |                 |            |   |            |   |           |   |           |   |           |                 |            |   |            |   |           |   |   |   |
| LDROM IAP                                 | x                   |   |           |                 |            |   |            |   |           |   |           |   |           |                 |            |   |            |   |           |   |   |   |
| ALL APROM IAP, not restricted by IAPRANGE | √                   |   |           |                 |            |   |            |   |           |   |           |   |           |                 |            |   |            |   |           |   |   |   |
| <b>1~0</b>                                | <b>LDSIZE [1:0]</b> | <p>LDROM space range selection</p> <p>00: None LDROM, the APROM address is 0000H to FFFFH</p> <p>01: The last 1K APROM area of the Flash ROM is LDROM, and the APROM address is 0000H-FBFFh</p> <p>10: The last 2K APROM area of the Flash ROM is LDROM, and the APROM address is 0000H-F7FFh</p> <p>11: The last 4K APROM area of the Flash ROM is LDROM, and the APROM address is 0000H-EFFFh</p> <p><b>NOTE: LDROM not allow IAP operation in anyways</b></p>  |           |                 |            |   |            |   |           |   |           |   |           |                 |            |   |            |   |           |   |   |   |

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

|                   |          |          |          |          |          |          |          |          |
|-------------------|----------|----------|----------|----------|----------|----------|----------|----------|
| <b>Bit number</b> | <b>7</b> | <b>6</b> | <b>5</b> | <b>4</b> | <b>3</b> | <b>2</b> | <b>1</b> | <b>0</b> |
|-------------------|----------|----------|----------|----------|----------|----------|----------|----------|

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

**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        | <b>IAPADR[7: 0]</b> | 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        | <b>IAPADR[15: 8]</b> | 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        | <b>IAPADER[7: 0]</b> | IAP extended address:<br>0x00: Both MOVC and IAP are for Flash ROM<br>0x01: The Unique ID area is read but cannot be written<br>0x02: Both MOVC and IAP are for independent EEPROM |



|  |  |   |
|--|--|---|
|  |  | 0x03: MOV <sub>C</sub> is performed in the LDROM region ( <b>Note: only MOV<sub>C</sub> can be used, not IAP, this item is only valid for LDROM operation, APROM operation this item is not valid!</b> )<br><br>Other: reserved |
|--|--|---|

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

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

| Bit number | Bit Mnemonic       | Description         |
|------------|--------------------|---------------------|
| 7~0        | <b>IAPDAT[7:0]</b> | Data written by IAP |

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

| Bit number   | 7                | 6 | 5      | 4   | 3 | 2 | 1         | 0   |
|--------------|------------------|---|--------|-----|---|---|-----------|-----|
| Bit Mnemonic | BTL <sub>D</sub> | - | SERASE | PRG | - | - | CMD[1: 0] |     |
| R/W          | R/W              | - | R/W    | R/W | - | - | R/W       | R/W |
| POR          | 0                | x | 0      | 0   | x | x | 0         | 0   |

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

|   |             |  |
|---|-------------|--|
| 7 | <b>BTLD</b> | BootLoader control bit<br><br>0: The program starts to run from the main program area (main program) after Reset;<br><br>1: The program starts to run from the BootLoader area after Reset |
|---|-------------|--|

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

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

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

**Bootloader Notes:**

1. The user must erase the target sector before writing LDR0M;
2. For the specific operation method, please refer to the description document " SinOne hardware BootLoader Function Implementation Application Guide" provided by SinOne.

## 5.8 Encryption

Users can choose whether to encrypt the SC95F761XB'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 (64 Kbytes Flash ROM) and LDROM will never be read from outside. It is recommended to enable the encryption function during mass production;
3. The only way to release security encryption is to re-programming
4. The encryption has no effect on iap read and write operation

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

### 5.9 Code Option Area (User Programming Settings)

| Symbol  | OPINX Address | Instructions                 | bit7         | bit6  | bit5       | bit4   | bit3      | bit2   | bit1        | bit0 |
|---------|---------------|------------------------------|--------------|-------|------------|--------|-----------|--------|-------------|------|
| OP_CTM0 | C1H@FFH       | Code Option register 0       | ENWDT        | ENXTL | SCLKS[1:0] |        | DISRST    | DISLVR | LVRS[1:0]   |      |
| OP_CTM1 | C2H@FFH       | Code Option register 1       | VREFS[1:0]   |       | OPBL       | DISJTG | IAPS[1:0] |        | LDSIZE[1:0] |      |
| OP_HRCR | 83H@FFH       | System clock change register | OP_HRCR[7:0] |       |            |        |           |        |             |      |

| IFB Address  | Symbol      | Write read | Instructions   |
|--------------|-------------|------------|--|
| OP_CTM0[7]   | ENWDT       | R/W        | <b>WDT Switch</b><br>0: WDT invalid<br>1: WDT valid(WDT stops counting during IAP execution)   |
| OP_CTM0[6]   | ENXTL       | R/W        | <b>External 32K crystal selector switch</b><br>0: External 32K crystal Interface disable, P5.0 and P5.1 valid<br>1: External 32K crystal Interface enable, P5.0 and P5.1 invalid   |
| OP_CTM0[5~4] | SCLKS [1:0] | R/W        | <b>System clock frequency selection bits</b><br>00: System clock frequency is HRC frequency divided by 1;<br>01: System clock frequency is HRC frequency divided by 2;<br>10: System clock frequency is HRC frequency divided by 4;<br>11: System clock frequency is HRC frequency divided by 8; |

| OP_CTM0[3]   | DISRST          | Read only | <b>IO/RST Selection bit</b><br>0: configure P5.2 as External Reset input pin<br>1: configure P5.2 as GPIO  |           |                 |             |   |             |   |
|--------------|-----------------|-----------|--|-----------|-----------------|-------------|---|-------------|---|
| OP_CTM0[2]   | DISLVR          | R/W       | <b>LVR control bit</b><br>0: LVR valid<br>1: LVR invalid   |           |                 |             |   |             |   |
| OP_CTM0[1~0] | LVRS [1:0]      | R/W       | <b>LVR voltage selection control</b><br>11: 4.3V reset<br>10: 3.7V reset<br>01: 2.9V reset<br>00: 1.9V reset   |           |                 |             |   |             |   |
| OP_CTM1[7~6] | VREFS [1:0]     | R/W       | <b>Reference voltage selection</b><br>00: Configure ADC VREF as VDD;<br>01: Configure ADC VREF as internal 2.048V<br>10: Configure ADC VREF as internal 1.024V<br>11: Configure ADC VREF as internal 2.4V  |           |                 |             |   |             |   |
| OP_CTM1[5]   | OP_BL           | R/W       | <b>Program run area control bit</b><br>0: After the chip is reset, it enters APROM<br>1: After the chip is reset, it enters LDROM<br><br>1. The MOV C and IAP restrictions for APROM are as follows: <table border="1" data-bbox="608 1756 1445 2038"> <thead> <tr> <th>Operation</th> <th>Is it operable?</th> </tr> </thead> <tbody> <tr> <td>LDROM MOV C</td> <td>x</td> </tr> <tr> <td>APROM MOV C</td> <td>√</td> </tr> </tbody> </table> | Operation | Is it operable? | LDROM MOV C | x | APROM MOV C | √ |
| Operation    | Is it operable? |           |  |           |                 |             |   |             |   |
| LDROM MOV C  | x               |           |  |           |                 |             |   |             |   |
| APROM MOV C  | √               |           |  |           |                 |             |   |             |   |

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

|                      |   |     | <p>01: The last 1K APROM area of the Flash ROM is LDROM, and the APROM address is 0000H-FBFFh</p> <p>10: The last 2K APROM area of the Flash ROM is LDROM, and the APROM address is 0000H-F7FFh</p> <p>11: The last 4K APROM area of the Flash ROM is LDROM, and the APROM address is 0000H-EFFFh</p> <p><b>NOTE: LDROM not allow IAP operation in anyways</b></p>   |                      |   |               |                          |     |      |               |                                      |               |                                      |             |          |               |                                      |               |                                      |
|----------------------|---|-----|--|----------------------|---|---------------|--------------------------|-----|------|---------------|--------------------------------------|---------------|--------------------------------------|-------------|----------|---------------|--------------------------------------|---------------|--------------------------------------|
| OP_HRCR              | OP_HRCR[7:0]  | R/W | <p><b>HRC frequency change register</b></p> <p>User can change the high-frequency oscillator frequency <math>f_{HRC}</math> by modifying the value of this register, and then change the system clock frequency <math>f_{sys}</math>:</p> <ol style="list-style-type: none"> <li>The initial value of OP_HRCR[7: 0] after power-on OP_HRCR[s] is a fixed value to ensure that <math>f_{HRC}</math> is 32MHz, OP_HRCR[s] of each IC may be different</li> <li>When the initial value is OP_HRCR[s], the system clock frequency <math>f_{sys}</math> 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 <math>f_{sys}</math> frequency changes by about 0.18%</li> </ol> <p>The relationship between OP_HRCR [7: 0] and <math>f_{sys}</math> output frequency is as follows:</p> <table border="1"> <thead> <tr> <th>OP_HRCR [7: 0] Value</th> <th><math>f_{sys}</math> actual output frequency (32M as an example)</th> </tr> </thead> <tbody> <tr> <td>OP_HRCR [s]-n</td> <td><math>32000*(1-0.18\%*n)</math>kHz</td> </tr> <tr> <td>...</td> <td>....</td> </tr> <tr> <td>OP_HRCR [s]-2</td> <td><math>32000*(1-0.18\%*2) = 31\ 884.8</math>kHz</td> </tr> <tr> <td>OP_HRCR [s]-1</td> <td><math>32000*(1-0.18\%*1) = 31\ 942.4</math>kHz</td> </tr> <tr> <td>OP_HRCR [s]</td> <td>32000kHz</td> </tr> <tr> <td>OP_HRCR [s]+1</td> <td><math>32000*(1+0.18\%*1) = 32\ 057.6</math>kHz</td> </tr> <tr> <td>OP_HRCR [s]+2</td> <td><math>32000*(1+0.18\%*2) = 32\ 115.2</math>kHz</td> </tr> </tbody> </table> | OP_HRCR [7: 0] Value | $f_{sys}$ actual output frequency (32M as an example) | OP_HRCR [s]-n | $32000*(1-0.18\%*n)$ kHz | ... | .... | OP_HRCR [s]-2 | $32000*(1-0.18\%*2) = 31\ 884.8$ kHz | OP_HRCR [s]-1 | $32000*(1-0.18\%*1) = 31\ 942.4$ kHz | OP_HRCR [s] | 32000kHz | OP_HRCR [s]+1 | $32000*(1+0.18\%*1) = 32\ 057.6$ kHz | OP_HRCR [s]+2 | $32000*(1+0.18\%*2) = 32\ 115.2$ kHz |
| OP_HRCR [7: 0] Value | $f_{sys}$ actual output frequency (32M as an example) |     |  |                      |   |               |                          |     |      |               |                                      |               |                                      |             |          |               |                                      |               |                                      |
| OP_HRCR [s]-n        | $32000*(1-0.18\%*n)$ kHz                              |     |  |                      |   |               |                          |     |      |               |                                      |               |                                      |             |          |               |                                      |               |                                      |
| ...                  | ....  |     |  |                      |   |               |                          |     |      |               |                                      |               |                                      |             |          |               |                                      |               |                                      |
| OP_HRCR [s]-2        | $32000*(1-0.18\%*2) = 31\ 884.8$ kHz                  |     |  |                      |   |               |                          |     |      |               |                                      |               |                                      |             |          |               |                                      |               |                                      |
| OP_HRCR [s]-1        | $32000*(1-0.18\%*1) = 31\ 942.4$ kHz                  |     |  |                      |   |               |                          |     |      |               |                                      |               |                                      |             |          |               |                                      |               |                                      |
| OP_HRCR [s]          | 32000kHz  |     |  |                      |   |               |                          |     |      |               |                                      |               |                                      |             |          |               |                                      |               |                                      |
| OP_HRCR [s]+1        | $32000*(1+0.18\%*1) = 32\ 057.6$ kHz                  |     |  |                      |   |               |                          |     |      |               |                                      |               |                                      |             |          |               |                                      |               |                                      |
| OP_HRCR [s]+2        | $32000*(1+0.18\%*2) = 32\ 115.2$ kHz                  |     |  |                      |   |               |                          |     |      |               |                                      |               |                                      |             |          |               |                                      |               |                                      |

|  |  |  |               |                      |
|--|--|--|---------------|----------------------|
|  |  |  | ...           | ...                  |
|  |  |  | OP_HRCR [s]+n | 32000*(1+0.18%*n)kHz |
|  |  | <p>Note:</p> <ol style="list-style-type: none"> <li>1. The OP_HRCR[7:0] value of IC after each power-on is the value of the high-frequency oscillator frequency fHRC closest to 32MHz; The user corrects the HRC value after each power-on so that the IC's system clock frequency fSYS works at the user's desired frequency;</li> <li>2. In order to ensure the reliable operation of IC, the highest operating frequency of IC should not exceed 10% of 32MHz, i.e. 35.2MHz;</li> <li>3. Confirm that the change of HRC frequency will not affect other functions.</li> </ol> |               |                      |

### 5.9.1 Customer-Option-related Registers Operation Instructions

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

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

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

For example: To set ENWDT (OP\_CTM0.7) to 1, the specific operation method is as follows:

C language example:

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

Assembly language example:



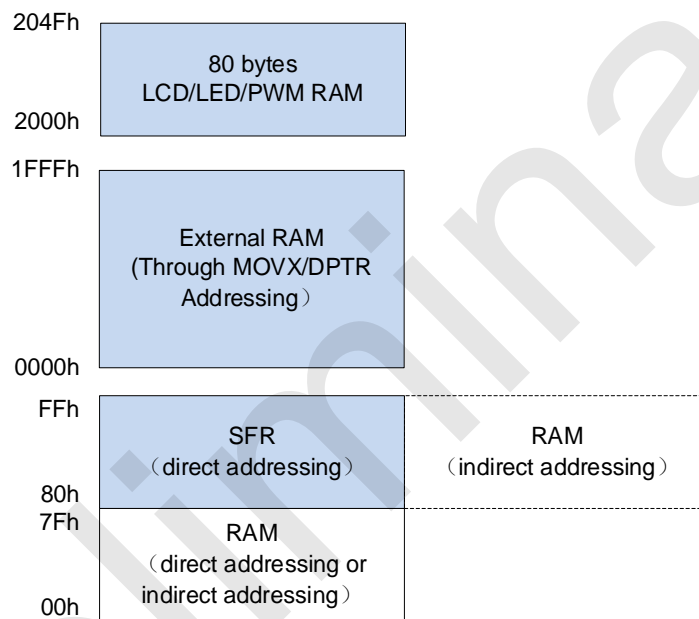
MOV OPINX,#C1H ; Write the address of OP\_CTM0 to the OPINX register

ORL OPREG,#80H ; Set 1 for OP\_CTM0.7

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

## 5.10 SRAM

The SRAM structure of the SC95F761XB is as follows:



The SRAM of the SC95F761XB MCU is divided into internal 256 bytes RAM, external 8192 bytes RAM and 80 bytes PWM&LCD 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.

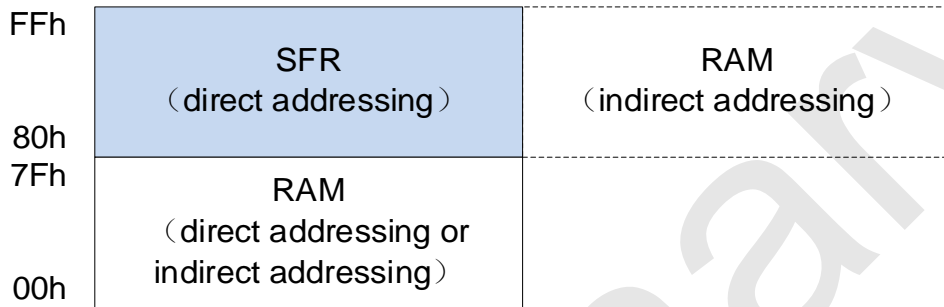
The address of the external RAM is 0000H~1FFFFH, but it needs to be addressed by the MOVX instruction.

### 5.10.1 Internal 256 Bytes SRAM

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

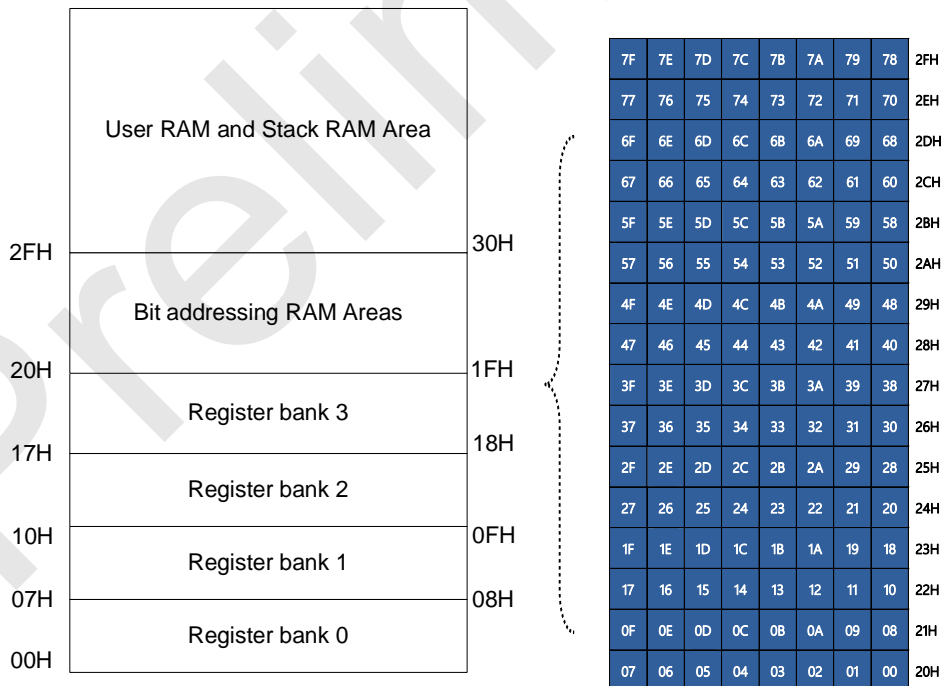
① Operating register group 0~3, address 00H~1FH, the combination of RS0 and RS1 in the program status word register PSW determines the operating register currently used, using operating register group 0~ 3 can speed up the operation;

- 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, (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 SC95F761XB is reset, the 8-bit stack pointer points to the stack area. Users generally set the initial value during initialization. It is recommended to set the initial value between E0H ~ FFH..



Internal 256 bytes RAM structure diagram

The internal low 128 bytes RAM structure is as follows:



SRAM structure diagram

### 5.10.2 External 8 Kbytes SRAM

External 8192 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 8192 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 [4: 0] |     |     |     |     |
| R/W          | - | - | - | R/W          | R/W | R/W | R/W | R/W |
| POR          | x | x | x | 0            | 0   | 0   | 0   | 0   |

| Bit number | Bit Mnemonic        | Description                                 |
|------------|---------------------|---|
| 4~0        | <b>EXADH [4: 0]</b> | High-bit of external SRAM operation address |
| 7~5        | -                   | reserved                                    |

### 5.10.3 External 80 bytes PWM&LCD/LED SRAM

The 2000H~204FH of the RAM address are used as PWM&LCD/LED SRAM of 80 bytes, among them:

1. The PWM duty cycle adjustment register occupies 2034H~204FH and can **be read and written.**;
2. LCD/LED display RAM occupies 2000H~201BH and can **be read and written.**

## 6 Special Function Register (SFR)

### 6.1 SFR Mapping

The SC95F761XB 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 SC95F761XB 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    | -       | USXINX  | -       | -       | -       |
| D0h | PSW   | PWMCFG | PWMCON0 | PWMCON1 | PWMPDL  | PWMPDH  | PWMDFR  | PWMFLT  |
| C8h | TXCON | TXMOD  | RCAPXL  | RCAPXH  | TLX     | THX     | TXINX   | WDTCON  |
| C0h | P4    | P4CON  | P4PH    | -       | USXCON0 | USXCON1 | USXCON2 | USXCON3 |
| B8h | IP    | IP1    | IP2     | INT0R   | INT1F   | INT1R   | INT2F   | INT2R   |
| B0h | P3    | P3CON  | P3PH    | P3VO    | INT0F   | ADCCFG2 | CMPCFG  | CMPCON  |
| A8h | IE    | IE1    | IE2     | ADCCFG0 | ADCCFG1 | ADCCON  | ADCVL   | ADCVH   |
| A0h | P2    | P2CON  | P2PH    | P2VO    | US1CON0 | US1CON1 | US1CON2 | US1CON3 |
| 98h | SCON  | SBUF   | P0CON   | P0PH    | P0VO    | US0CON1 | US0CON2 | US0CON3 |

|     |                 |                     |      |        |      |         |         |         |
|-----|-----------------|---------------------|------|--------|------|---------|---------|---------|
| 90h | P1              | P1CON               | P1PH | DDRCON | P1VO | 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: The empty part of the SFR register are not recommended for users.

## 6.2 SFR Instructions

### 6.2.1 SFR

SFR specific explanations are as follows:

| 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]   |     |     |     |     |     |      |     | 0000111b |
| DPL      | 82H | DPTR data pointer low             | DPL[7: 0]  |     |     |     |     |     |      |     | 0000000b |
| DPH      | 83H | DPTR data pointer high            | DPH[7: 0]  |     |     |     |     |     |      |     | 0000000b |
| DPL1     | 84H | DPTR1 data pointer low            | DPL1[7: 0] |     |     |     |     |     |      |     | 0000000b |
| DPH1     | 85H | DPTR1 data pointer high           | DPH1[7: 0] |     |     |     |     |     |      |     | 0000000b |
| DPS      | 86H | DPTR selection register           | ID1        | ID0 | TSL | AU1 | AU0 | -   | -    | SEL | 0000xx0b |
| PCON     | 87H | Power management control register | SMOD       | -   | -   | -   | RST | -   | STOP | IDL | 0xx0x00b |
| TCON     | 88H | Timer control register            | TF1        | TR1 | TF0 | TR0 | IE1 | -   | IE0  | -   | 0000x0xb |

|         |     |   |               |             |            |             |            |       |           |           |           |
|---------|-----|---|---------------|-------------|------------|-------------|------------|-------|-----------|-----------|-----------|
| 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          | USMDX[1: 0]   | -           | -          | -           | -          | T1FD  | T0FD      | 00xxxx00b |           |
| OTCON   | 8FH | Output control register                   | USMD1[1: 0]   | USMD0[1: 0] |            | VOIRS[1: 0] |            | SCS   | BIAS      | 00000000b |           |
| 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 |
| DDRCON  | 93H | Display drive control register            | DDR0N         | DMOD        | DUTY[1: 0] |             | VLCD[3: 0] |       |           | 00000000b |           |
| P1VO    | 94H | P1 port display driver output register    | P17VO         | P16VO       | P15VO      | P14VO       | P13VO      | P12VO | P11VO     | P10VO     | 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                    | -             | -           | P3L[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   | 0000000b  |
| P0PH    | 9BH | P0 port pull-up resistor control register | P0H7          | P0H6  | P0H5  | P0H4  | P0H3  | P0H2  | P0H1  | P0H0   | 0000000b  |
| P0VO    | 9CH | P0 port display driver output register    | P07VO         | P06VO | P05VO | P04VO | P03VO | P02VO | P01VO | P00VO  | 0000000b  |
| US0CON1 | 9DH | USCI0 control register 1                  | US0CON1[7: 0] |       |       |       |       |       |       |        | 0000000b  |
| US0CON2 | 9EH | USCI0 control register 2                  | 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  |
| P2VO    | A3H | P2 port display driver output register    | P27VO         | P26VO | P25VO | P24VO | P23VO | P22VO | P21VO | P20VO  | 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   | ECMP  | -     | EINT2 | EBTM  | EPWM  | EUSCI0 | 000x0000b |

|         |     |  |             |        |           |             |             |        |             |        |           |
|---------|-----|--|-------------|--------|-----------|-------------|-------------|--------|-------------|--------|-----------|
| IE2     | AAH | Interrupt enable register 2                  | -           | -      | -         | EUSCI5      | EUSCI4      | EUSCI3 | EUSCI2      | EUSCI1 | xxx00000b |
| ADCCFG0 | ABH | ADC setting register 0                       | EAIN7       | EAIN6  | EAIN5     | EAIN4       | EAIN3       | EAIN2  | EAIN1       | EAIN0  | 00000000b |
| ADCCFG1 | ACH | ADC setting register 1                       | EAIN15      | EAIN14 | EAIN13    | EAIN12      | EAIN11      | EAIN10 | EAIN9       | EAIN8  | 00000000b |
| ADCCON  | ADH | ADC control register                         | ADCEN       | ADCS   | EOC/ADCIF | ADCIS[4: 0] |             |        |             |        | 00000000b |
| ADCVL   | AEH | ADC Conversion Value Register                | ADCV[3: 0]  |        |           |             | -           | -      | -           | -      | 1111xxxxb |
| ADCVH   | AFH | ADC Conversion Value Register                | ADCV[11: 4] |        |           |             |             |        |             |        | 11111111b |
| P3      | B0H | P3 port data register                        | P37         | P36    | P35       | P34         | P33         | P32    | P31         | P30    | 00000000b |
| P3CON   | B1H | P3 port input/output control register        | P3C7        | P3C6   | P3C5      | P3C4        | P3C3        | P3C2   | P3C1        | P3C0   | 00000000b |
| P3PH    | B2H | P3 port pull-up resistor control register    | P3H7        | P3H6   | P3H5      | P3H4        | P3H3        | P3H2   | P3H1        | P3H0   | 00000000b |
| P3VO    | B3H | P3 port display driver output register       | P37VO       | P36VO  | P35VO     | P34VO       | P33VO       | P32VO  | P31VO       | P30VO  | 00000000b |
| INT0F   | B4H | INT0 falling edge interrupt control register | INT0F7      | INT0F6 | INT0F5    | INT0F4      | -           | -      | -           | -      | 0000xxxxb |
| ADCCFG2 | B5H | ADC setting register 2                       | -           | -      | -         | LOWSP[2: 0] |             |        | -           | -      | xxx000xxb |
| CMPCFG  | B6H | Analog comparator setting register           | -           | -      | -         | CMPP        | CMPIM[1: 0] |        | CMPIS[1: 0] |        | xxx00000b |
| CMPCON  | B7H | Analog Comparator Control Register           | CMPEN       | CMPIF  | CMPSTA    | -           | CMPRF[3: 0] |        |             |        | 000x0000b |
| IP      | B8H | Interrupt priority control register          | -           | IPADC  | IPT2      | IPUART      | IPT1        | IPINT1 | IPT0        | IPINT0 | x0000000b |
| IP1     | B9H | Interrupt priority control register 1        | IPT4        | IPT3   | IPCMP     | -           | IPINT2      | IPBTM  | IPPWM       | IPSSIO | 000x0000b |



|         |     |  |               |        |        |         |         |         |         |         |           |
|---------|-----|--|---------------|--------|--------|---------|---------|---------|---------|---------|-----------|
| IP2     | BAH | Interrupt priority control register 2        | -             | -      | -      | IPUSCI5 | IPUSCI4 | IPUSCI3 | IPUSCI2 | IPUSCI1 | xxx00000b |
| INT0R   | BBH | INT0 rising edge interrupt control register  | INT0R7        | INT0R6 | INT0R5 | INT0R4  | -       | -       | -       | -       | 0000xxxxb |
| INT1F   | BCH | INT1 falling edge interrupt control register | INT1F7        | INT1F6 | INT1F5 | INT1F4  | INT1F3  | INT1F2  | INT1F1  | INT1F0  | 00000000b |
| INT1R   | BDH | INT1 rising edge interrupt control register  | INT1R7        | INT1R6 | INT1R5 | INT1R4  | INT1R3  | INT1R2  | INT1R1  | INT1R0  | 00000000b |
| INT2F   | BEH | INT2 falling edge interrupt control register | -             | -      | -      | -       | INT2F3  | INT2F2  | INT2F1  | INT2F0  | xxxx0000b |
| INT2R   | BFH | INT2 rising edge interrupt control register  | -             | -      | -      | -       | INT2R3  | INT2R2  | INT2R1  | INT2R0  | xxxx0000b |
| P4      | C0H | P4 port data register                        | P47           | P46    | P45    | P44     | P43     | P42     | P41     | P40     | 00000000b |
| P4CON   | C1H | P4 port input/output control register        | P4C7          | P4C6   | P4C5   | P4C4    | P4C3    | P4C2    | P4C1    | P4C0    | 00000000b |
| P4PH    | C2H | P4 port pull-up resistor control register    | P4H7          | P4H6   | P4H5   | P4H4    | P4H3    | P4H2    | P4H1    | P4H0    | 00000000b |
| USXCON0 | C4H | USCI2/3/4/5 control register 0               | USXCON0[7: 0] |        |        |         |         |         |         |         | 00000000b |
| USXCON1 | C5H | USCI2/3/4/5 control register 1               | USXCON1[7: 0] |        |        |         |         |         |         |         | 00000000b |
| USXCON2 | C6H | USCI2/3/4/5 control register 2               | USXCON2[7: 0] |        |        |         |         |         |         |         | 00000000b |
| USXCON3 | C7H | USCI2/3/4/5 control register 3               | USXCON3[7: 0] |        |        |         |         |         |         |         | 00000000b |
| TXCON   | C8H | Timer 2/3/4 control register                 | TFX           | EXFX   | RCLKX  | TCLKX   | EXENX   | TRX     | C/TX    | CP/RLX  | 00000000b |
| TXMOD   | C9H | Timer 2/3/4 operating mode register          | TXFD          | -      | EPWMN1 | EPWMN0  | INVN1   | INVN0   | TXOE    | DCXEN   | 0x000000b |

|         |     |   |              |         |             |        |            |              |              |        |           |
|---------|-----|---|--------------|---------|-------------|--------|------------|--------------|--------------|--------|-----------|
| RCAPXL  | CAH | Timer 2/3/4 reload low 8 bits             | RCAPXL[7: 0] |         |             |        |            |              |              |        | 0000000b  |
| RCAPXH  | CBH | Timer 2/3/4 reload high 8 bits            | RCAPXH[7: 0] |         |             |        |            |              |              |        | 0000000b  |
| TLX     | CCH | Timer 2/3/4 low 8 bits                    | TLX[7: 0]    |         |             |        |            |              |              |        | 0000000b  |
| THX     | CDH | Timer 2/3/4 high 8 bits                   | THX[7: 0]    |         |             |        |            |              |              |        | 0000000b  |
| TXINX   | CEH | Timer control register pointer            | -            | -       | -           | -      | -          | TXINX[2: 0]  |              |        | xxxx010b  |
| WDTCON  | CFH | WDT control register                      | -            | -       | -           | CLRWDT | -          | WDTCKS[2: 0] |              |        | xxx0x000b |
| PSW     | D0H | Program status word register              | CY           | AC      | F0          | RS1    | RS0        | OV           | F1           | P      | 0000000b  |
| PWMCFG  | D1H | PWM0 setting register                     | INV7         | INV6    | INV5        | INV4   | INV3       | INV2         | INV1         | INV0   | 0000000b  |
| PWMCON0 | D2H | PWM0 control register0                    | ENPWM        | PWMIF   | PWMCK[1: 0] |        | -          | -            | PWMMD[1:0]   |        | 0000xx00b |
| PWMCON1 | D3H | PWM0 control register 1                   | ENPWM7       | ENPWM6  | ENPWM5      | ENPWM4 | ENPWM3     | ENPWM2       | ENPWM1       | ENPWM0 | x0000000b |
| PWMPDL  | D4H | PWM0 period register low 8 bits           | PWMPDL[7:0]  |         |             |        |            |              |              |        | 0000000b  |
| PWMPDH  | D5H | PWM0 period register high 8 bits          | PWMPDH[7:0]  |         |             |        |            |              |              |        | 0000000b  |
| PWMDFR  | D6H | PWM0 dead time setting register           | PDF1[3: 0]   |         |             |        | PDR1[3: 0] |              |              |        | 0000000b  |
| PWMFLT  | D7H | PWM0 fault detection setting register     | FLTEN1       | FLTSTA1 | FLTMD1      | FLTLV1 | -          | -            | FLTDT1[1: 0] |        | 0000xx00b |
| P5      | D8H | P5 port data register                     | -            | -       | P55         | P54    | P53        | P52          | P51          | P50    | xx000000b |
| P5CON   | D9H | P5 port input/output control register     | -            | -       | P5C5        | P5C4   | P5C3       | P5C2         | P5C1         | P5C0   | xx000000b |
| P5PH    | DAH | P5 port pull-up resistor control register | -            | -       | P5H5        | P5H4   | P5H3       | P5H2         | P5H1         | P5H0   | xx000000b |

|         |     |   |               |    |        |              |   |   |            |          |           |
|---------|-----|---|---------------|----|--------|--------------|---|---|------------|----------|-----------|
| USXINX  | DCH | USCI2/3/4/5 Control register pointer                | -             | -  | -      | -            | - | - | USCIX[2:0] | xxxx000b |           |
| ACC     | E0H | accumulator   | ACC[7: 0]     |    |        |              |   |   |            | 0000000b |           |
| EXA0    | E9H | Extended Accumulator 0                              | EXA[7: 0]     |    |        |              |   |   |            | 0000000b |           |
| EXA1    | EAH | Extended Accumulator 1                              | EXA[15: 8]    |    |        |              |   |   |            | 0000000b |           |
| EXA2    | EBH | Extended Accumulator 2                              | EXA[23: 16]   |    |        |              |   |   |            | 0000000b |           |
| EXA3    | ECH | Extended Accumulator 3                              | EXA[31: 24]   |    |        |              |   |   |            | 0000000b |           |
| EXBL    | EDH | Extended B register L                               | EXB [7: 0]    |    |        |              |   |   |            | 0000000b |           |
| EXBH    | EEH | Extended B register H                               | EXB [15: 8]   |    |        |              |   |   |            | 0000000b |           |
| OPERCON | EFH | Arithmetic control register                         | OPERS         | MD | -      | -            | - | - | CRCRST     | CRCSTA   | 00xxx00b  |
| B       | F0H | B register  | B[7: 0]       |    |        |              |   |   |            | 0000000b |           |
| IAPKEY  | F1H | IAP protection register                             | IAPKEY[7: 0]  |    |        |              |   |   |            | 0000000b |           |
| IAPADL  | F2H | IAP write address low register                      | IAPADR[7: 0]  |    |        |              |   |   |            | 0000000b |           |
| IAPADH  | F3H | IAP write address high register                     | IAPADR[15: 8] |    |        |              |   |   |            | 0000000b |           |
| IAPADE  | F4H | IAP write to extended address register              | IAPADER[7: 0] |    |        |              |   |   |            | 0000000b |           |
| IAPDAT  | F5H | IAP data register                                   | IAPDAT[7: 0]  |    |        |              |   |   |            | 0000000b |           |
| IAPCTL  | F6H | IAP control register                                | BTLD          | -  | SERASE | PRG          | - | - | CMD[1: 0]  |          | 0x00xx00b |
| EXADH   | F7H | High-bit address of external SRAM operation address | -             | -  | -      | EXADH [4: 0] |   |   |            | xxx0000b |           |

|        |     |                                      |              |       |   |   |             |           |
|--------|-----|--------------------------------------|--------------|-------|---|---|-------------|-----------|
| BTMCON | FBH | Low frequency timer control register | ENBTM        | BTMIF | - | - | BTMFS[3: 0] | 00xx0000b |
| CRCINX | FCH | CRC pointer                          | CRCINX[7: 0] |       |   |   |             | 00000000b |
| CRCREG | FDH | CRC register                         | CRCREG[7: 0] |       |   |   |             | nnnnnnnbn |
| OPINX  | FEH | Option pointer                       | OPINX[7: 0]  |       |   |   |             | 00000000b |
| OPREG  | FFH | Option register                      | OPREG[7: 0]  |       |   |   |             | nnnnnnnbn |

### 6.2.2 PWM0 Duty Cycle Adjustment Register(R/W)

| ADD   | 7           | 6 | 5 | 4 | 3 | 2 | 1 | 0 | POR       |
|-------|-------------|---|---|---|---|---|---|---|-----------|
| 2040H | PDT00[15:8] |   |   |   |   |   |   |   | 00000000b |
| 2041H | PDT00[7:0]  |   |   |   |   |   |   |   | 00000000b |
| 2042H | PDT01[15:8] |   |   |   |   |   |   |   | 00000000b |
| 2043H | PDT01[7:0]  |   |   |   |   |   |   |   | 00000000b |
| 2044H | PDT02[15:8] |   |   |   |   |   |   |   | 00000000b |
| 2045H | PDT02[7:0]  |   |   |   |   |   |   |   | 00000000b |
| 2046H | PDT03[15:8] |   |   |   |   |   |   |   | 00000000b |
| 2047H | PDT03[7:0]  |   |   |   |   |   |   |   | 00000000b |
| 2048H | PDT04[15:8] |   |   |   |   |   |   |   | 00000000b |
| 2049H | PDT04[7:0]  |   |   |   |   |   |   |   | 00000000b |
| 204AH | PDT05[15:8] |   |   |   |   |   |   |   | 00000000b |
| 204BH | PDT05[7:0]  |   |   |   |   |   |   |   | 00000000b |
| 204CH | PDT06[15:8] |   |   |   |   |   |   |   | 00000000b |
| 204DH | PDT06[7:0]  |   |   |   |   |   |   |   | 00000000b |
| 204EH | PDT07[15:8] |   |   |   |   |   |   |   | 00000000b |
| 204FH | PDT07[7:0]  |   |   |   |   |   |   |   | 00000000b |

### 6.2.3 PWM2~4 Duty Cycle Adjustment Register(R/W)

| ADD   | 7           | 6 | 5 | 4 | 3 | 2 | 1 | 0 | POR      |
|-------|-------------|---|---|---|---|---|---|---|----------|
| 2034H | PDT20[15:8] |   |   |   |   |   |   |   | 0000000b |
| 2035H | PDT20[7:0]  |   |   |   |   |   |   |   | 0000000b |
| 2036H | PDT21[15:8] |   |   |   |   |   |   |   | 0000000b |
| 2037H | PDT21[7:0]  |   |   |   |   |   |   |   | 0000000b |
| 2038H | PDT30[15:8] |   |   |   |   |   |   |   | 0000000b |
| 2039H | PDT30[7:0]  |   |   |   |   |   |   |   | 0000000b |
| 203AH | PDT31[15:8] |   |   |   |   |   |   |   | 0000000b |
| 203BH | PDT31[7:0]  |   |   |   |   |   |   |   | 0000000b |
| 203CH | PDT40[15:8] |   |   |   |   |   |   |   | 0000000b |
| 203DH | PDT40[7:0]  |   |   |   |   |   |   |   | 0000000b |
| 203EH | PDT41[15:8] |   |   |   |   |   |   |   | 0000000b |
| 203FH | PDT41[7:0]  |   |   |   |   |   |   |   | 0000000b |

#### 6.2.4 LCD/LED display RAM configuration (R/W)

| ADD   | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|       | COM7  | COM6  | COM5  | COM4  | COM3  | COM2  | COM1  | COM0  |
| 2000H | SEG0  | SEG0  | SEG0  | SEG0  | SEG0  | SEG0  | SEG0  | SEG0  |
| 2001H | SEG1  | SEG1  | SEG1  | SEG1  | SEG1  | SEG1  | SEG1  | SEG1  |
| 2002H | SEG2  | SEG2  | SEG2  | SEG2  | SEG2  | SEG2  | SEG2  | SEG2  |
| 2003H | SEG3  | SEG3  | SEG3  | SEG3  | SEG3  | SEG3  | SEG3  | SEG3  |
| 2004H | SEG4  | SEG4  | SEG4  | SEG4  | SEG4  | SEG4  | SEG4  | SEG4  |
| 2005H | SEG5  | SEG5  | SEG5  | SEG5  | SEG5  | SEG5  | SEG5  | SEG5  |
| 2006H | SEG6  | SEG6  | SEG6  | SEG6  | SEG6  | SEG6  | SEG6  | SEG6  |
| 2007H | SEG7  | SEG7  | SEG7  | SEG7  | SEG7  | SEG7  | SEG7  | SEG7  |
| 2008H | SEG8  | SEG8  | SEG8  | SEG8  | SEG8  | SEG8  | SEG8  | SEG8  |
| 2009H | SEG9  | SEG9  | SEG9  | SEG9  | SEG9  | SEG9  | SEG9  | SEG9  |
| 200AH | SEG10 | SEG10 | SEG10 | SEG10 | SEG10 | SEG10 | SEG10 | SEG10 |

| ADD   | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|       | COM7  | COM6  | COM5  | COM4  | COM3  | COM2  | COM1  | COM0  |
| 200BH | SEG11 | SEG11 | SEG11 | SEG11 | SEG11 | SEG11 | SEG11 | SEG11 |
| 200CH | SEG12 | SEG12 | SEG12 | SEG12 | SEG12 | SEG12 | SEG12 | SEG12 |
| 200DH | SEG13 | SEG13 | SEG13 | SEG13 | SEG13 | SEG13 | SEG13 | SEG13 |
| 200EH | SEG14 | SEG14 | SEG14 | SEG14 | SEG14 | SEG14 | SEG14 | SEG14 |
| 200FH | SEG15 | SEG15 | SEG15 | SEG15 | SEG15 | SEG15 | SEG15 | SEG15 |
| 2010H | SEG16 | SEG16 | SEG16 | SEG16 | SEG16 | SEG16 | SEG16 | SEG16 |
| 2011H | SEG17 | SEG17 | SEG17 | SEG17 | SEG17 | SEG17 | SEG17 | SEG17 |
| 2012H | SEG18 | SEG18 | SEG18 | SEG18 | SEG18 | SEG18 | SEG18 | SEG18 |
| 2013H | SEG19 | SEG19 | SEG19 | SEG19 | SEG19 | SEG19 | SEG19 | SEG19 |
| 2014H | SEG20 | SEG20 | SEG20 | SEG20 | SEG20 | SEG20 | SEG20 | SEG20 |
| 2015H | SEG21 | SEG21 | SEG21 | SEG21 | SEG21 | SEG21 | SEG21 | SEG21 |
| 2016H | SEG22 | SEG22 | SEG22 | SEG22 | SEG22 | SEG22 | SEG22 | SEG22 |
| 2017H | SEG23 | SEG23 | SEG23 | SEG23 | SEG23 | SEG23 | SEG23 | SEG23 |
| 2018H | SEG24 | SEG24 | SEG24 | SEG24 | SEG24 | SEG24 | SEG24 | SEG24 |
| 2019H | SEG25 | SEG25 | SEG25 | SEG25 | SEG25 | SEG25 | SEG25 | SEG25 |
| 201AH | SEG26 | SEG26 | SEG26 | SEG26 | SEG26 | SEG26 | SEG26 | SEG26 |
| 201BH | SEG27 | SEG27 | SEG27 | SEG27 | SEG27 | SEG27 | SEG27 | SEG27 |

## 6.2.5 Introduction of Common Special Function Registers of 8051 Core

### Program Counter PC

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

### Accumulator ACC (E0H)

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

### B Register (F0H)

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

### Stack Pointer SP (81H)

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

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

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

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

|     |                |   |     |   |
|-----|----------------|---|-----|---|
| 4~3 | <b>RS1,RS0</b> | 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   | <b>OV</b>      | Overflow flag   |     |   |
| 1   | <b>F1</b>      | F1 flag<br>User-defined flag  |     |   |
| 0   | <b>P</b>       | Parity flag. This flag bit is the parity value of the number of 1s in the accumulator ACC.<br><br>1: The number of 1s in ACC is odd<br><br>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 SC95F761XB 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          | <b>ID1</b>   | DPTR1 plus or minus control bit<br><br>0: When AU1=1, whenever the MOVC/MOVX @DPTR is executed, the current DPTR1 will automatically increase by 1.<br><br>1: When AU1=1, whenever MOVC/MOVX @DPTR is executed, the current DPTR1 will automatically decrease by 1. |
| 6          | <b>ID0</b>   | DPTR plus or minus control bit<br><br>0: When AU0=1, whenever MOVC/MOVX @DPTR is executed, the current DPTR0 will automatically increase by 1.<br><br>1: When AU0=1, whenever MOVC/MOVX @DPTR is executed, the current DPTR0 will automatically decrease by 1.      |
| 5          | <b>TSL</b>   | SEL flip control bit<br><br>0: Whenever MOVC/MOVX @DPTR is executed, DPS.0 (SEL) does not flip<br><br>1: Whenever MOVC/MOVX @DPTR is executed, DPS.0 (SEL) flips once   |
| 4          | <b>AU1</b>   | DPTR1 automatic plus and minus control bit<br><br>0: None<br><br>1: Whenever MOVC/MOVX @DPTR is executed, the current DPTR1 will increase or decrease by 1 (depending on ID1)   |
| 3          | <b>AU0</b>   | DPTR automatic plus and minus control bit<br><br>0: None<br><br>1: Whenever MOVC/MOVX @DPTR is executed, the current DPTR0 will increase or decrease by 1 (depending on ID0)  |

|     |            |   |
|-----|------------|---|
| 0   | <b>SEL</b> | DPTR0, DPTR1 selection bits<br><br>0: MOVC/MOVX @DPTR object is DPTR0<br><br>1: MOVC/MOVX @DPTR object is DPTR1 |
| 2~1 | -          | reserved  |

Preliminary

## 7 Power, Reset And System Clock

### 7.1 Power Circuit

The SC95F761XB 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 2.048V, 1.024V and 2.4V voltage that can be used as an internal reference voltage for the ADC. Users can find the specific settings in the [18 analog-to-digital converter \(ADC\)](#).

### 7.2 Power-on Reset

After the SC95F761XB 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 SC95F761XB will always be reset until the voltage supplied to SC95F761XB is higher than a certain voltage, and the internal Clock starts to be effective. The duration of reset stage is related to rising speed of external power. Once the external supply voltage is up to built-in POR voltage, the reset stage would be completed.

#### 7.2.2 Loading Information Stage

There is a warm-up counter inside The SC95F761XB. During the reset stage, the warm-up counter is cleared to 0 until the voltage exceeds the POR voltage, the internal RC oscillator starts to oscillate, and the warm-up counter starts counting. When the internal warm-up counter counts to a certain number, every certain number of HRC clocks will read a byte of data from the IFB (including Code Option) in the Flash ROM and store it in the internal system register. This reset signal will not end until the warm-up is completed.

#### 7.2.3 Normal Operation Stage

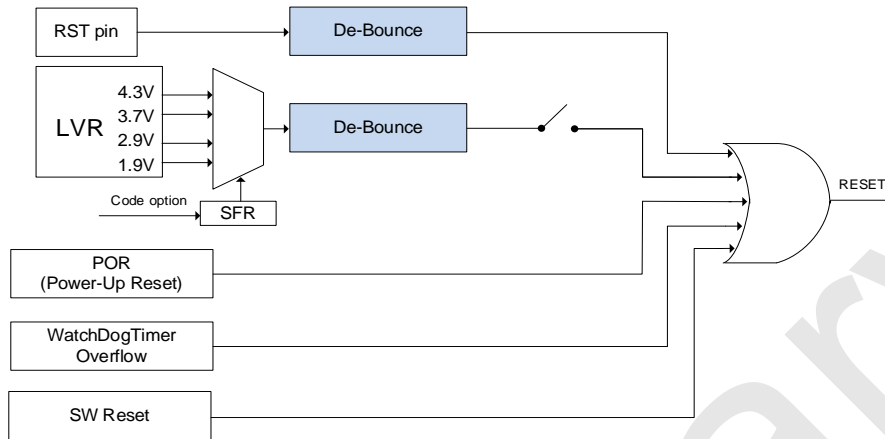
After finishing the Loading Information stage, the SC95F761XB starts to read the instruction code from Flash and enters the normal operation stage. The LVR voltage is the set value of Code Option written by the user.

### 7.3 Reset Modes

The SC95F761XB has 5 reset methods, the first four are hardware reset:

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 SC95F761XB is as follows:

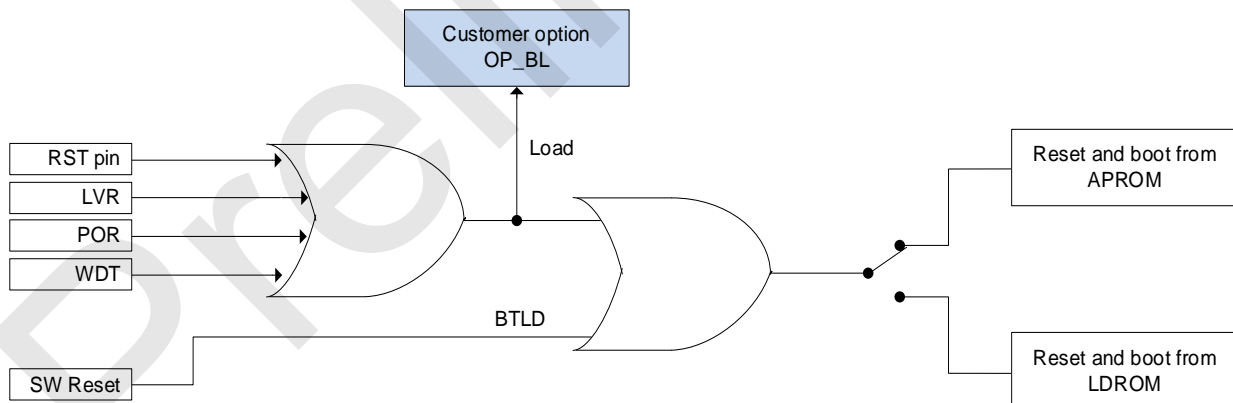


SC95F761XB Reset circuit diagram

After reset the boot area:

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

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



SC95F761XB's boot area switch after reset

### 7.3.1 External Reset

External reset is a reset pulse signal of a certain width given to SC95F761XB from external RST pin to realize the reset of SC95F761XB. The user can configure the P5.2/RST pin as RST (reset pin) by Code Option.

### 7.3.2 Low-voltage Reset LVR

The SC95F761XB provides a low-voltage reset circuit. There are 4-level LVR voltage options: 4.3V, 3.7V, 2.9V, 1.9V. The default value is the Option value written by the user. A reset occurs when the VDD voltage is less than the threshold voltage for low-voltage reset and the duration is greater than  $T_{LVR}$ . Among them,  $T_{LVR}$  is the buffering time of LVR, about 30 $\mu$ s.

**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         |   | read only | R/W    | R/W        |   |
| POR          | n     | n     | n           |   | n         | n      | n          |   |

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

**7.3.3 Power-on Reset (POR)**

The SC95F761XB 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 SC95F761XB has a WDT, the clock source of which is the internal 32 kHz LRC. The user can choose whether to enable the watchdog reset function by Code Option.

**OP\_CTM0 (C1H@FFH) Code Option Register 0 (read/write)**

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

| Bit number | Bit Mnemonic | Description  |
|------------|--------------|--|
| 7          | <b>ENWDT</b> | WDT control bit (This bit is transferred by the system to the value set by the user Code Option)<br><br>1: WDT valid<br>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          | <b>CLRWDT</b> | Clear WDT (Only valid when set to 1) |

|              |                      | 1: WDT counter restart, cleared by system hardware  |              |                   |     |       |     |       |     |       |     |        |     |        |     |         |     |        |     |        |
|--------------|----------------------|---|--------------|-------------------|-----|-------|-----|-------|-----|-------|-----|--------|-----|--------|-----|---------|-----|--------|-----|--------|
| 2~0          | <b>WDTCKS [2: 0]</b> | Watchdog clock selection<br><table border="1"> <thead> <tr> <th>WDTCKS[2: 0]</th> <th>WDT overflow time</th> </tr> </thead> <tbody> <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> </tbody> </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 | SMOD       | - | - | - | RST        | - | STOP       | IDL        |
| R/W          | write only | - | - | - | Write only | - | Write only | Write only |

|     |   |   |   |   |   |   |   |   |
|-----|---|---|---|---|---|---|---|---|
| POR | 0 | x | x | x | n | x | 0 | 0 |
|-----|---|---|---|---|---|---|---|---|

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

### 7.3.6 Register Reset Value

When The SC95F761XB is in reset state, most registers will return to their initial state. The watchdog (WDT) is turned off. The initial value of the program counter PC is 0000h, and the initial value of the stack pointer SP is 07h. The "hot restart" Reset (such as WDT, LVR, software reset, etc.) will not affect the SRAM, and the SRAM value is always the value before the reset. The loss of SRAM content will occur when the power supply voltage is so low that the RAM cannot be saved.

The initial values of the power-on reset of the SFR register see [6.2.1 SFR](#).

## 7.4 High- frequency System Clock Circuit

The SC95F761XB has a built-in high-precision high-frequency oscillator (HRC) with adjustable oscillation frequency. The HRC is accurately adjusted to 32 MHz@5V/25°C at the factory. Users can set the system clock to 32/16/ 8/4 MHz through the Code Option when programming. This HRC will have a certain drift under the influence of the ambient temperature and operating voltage:

- 2.0V ~ 5.5V,-40 ~ 85°C application environment, frequency error does not exceed ±1%
- -40 ~ 105°C application environment, does not exceed ±2%

The HRC can be automatically calibrated by connecting an external 32.768 kHz crystal oscillator. After calibration, HRC precision can be infinitely close to the precision of external 32.768kHz crystal oscillator.

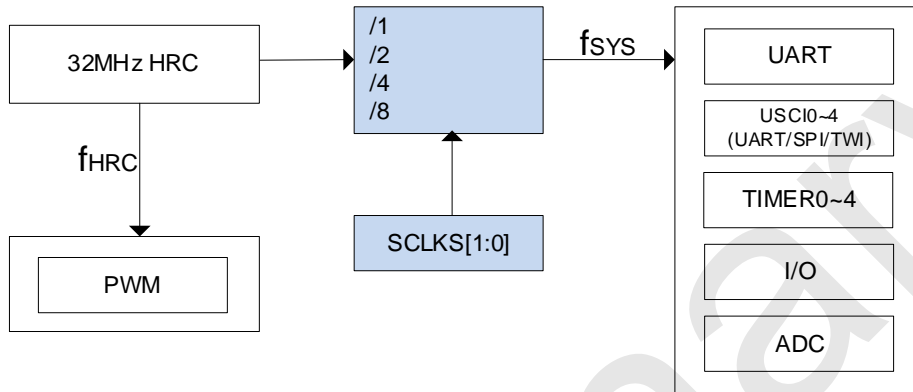
Users only need to connect an external 32.768 kHz crystal oscillator, and set the external 32.768 kHz crystal oscillator function through the Code Opiton when programming. The HRC self-calibration function is automatically turned on after the IC is powered on.

#### Note:

1. The clock source of the PWM is fixed at  $f_{HRC} = 32 \text{ MHz}$ .



2. After the external 32.768kHz crystal oscillator function is enabled, the HRC self-calibration function will start synchronously. At this point, if the external 32.768kHz crystal oscillator is not connected or abnormally connected, the HRC calibration value will have unexpected deviation, resulting in abnormal HRC frequency value.



SC95F761XB Internal clock relationship

**OP\_CTM0 (C1H@FFH) Code Option Register 0 (read/write)**

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

| Bit number | Bit Mnemonic       | Description   |
|------------|--------------------|---|
| 5~4        | <b>SCLKS[1: 0]</b> | System clock frequency selection bits<br>00: System clock frequency is HRC frequency divided by 1;<br>01: System clock frequency is HRC frequency divided by 2; |

|  |  |  |
|--|--|--|
|  |  | 10: System clock frequency is HRC frequency divided by 4;<br>11: System clock frequency is HRC frequency divided by 8; |
|--|--|--|

**Note:** The SC95F761XB 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 32.768 kHz crystal oscillator. Therefore, if the user uses the 32.768 kHz external crystal oscillator function, the HRC frequency will always be corrected to 32 MHz. 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        | <b>OP_HRCR[7: 0]</b> | <p><b>HRC frequency change register</b></p> <p>The user can change the high-frequency oscillator frequency <math>f_{HRC}</math> by modifying the value of this register, and then change the system clock frequency <math>f_{sys}</math> of the IC:</p> <p>The initial value of OP_HRCR[7: 0] after power-on OP_HRCR[s] is a fixed value to ensure that <math>f_{HRC}</math> is 32 MHz, OP_HRCR[s] of each IC may be different</p> <p>When the initial value is OP_HRCR[s], the system clock frequency <math>f_{sys}</math> of the IC can be set to an accurate 32/16/8/4 MHz through the Option item. When OP_HRCR [7: 0] changes by 1, the <math>f_{sys}</math> frequency changes by about 0.18%</p> <p>The relationship between OP_HRCR [7: 0] and HRC output frequency is as follows:</p> |

|  |  |                      |  |
|--|--|----------------------|--|
|  |  | OP_HRCR [7: 0] value | fsys actual output frequency (32M as an example) |
|  |  | OP_HRCR [s]-n        | $32000 * (1 - 0.18\% * n)$ kHz                   |
|  |  | ...                  | ....   |
|  |  | OP_HRCR [s]-2        | $32000 * (1 - 0.18\% * 2) = 31\ 884.8$ kHz       |
|  |  | OP_HRCR [s]-1        | $32000 * (1 - 0.18\% * 1) = 31\ 942.4$ kHz       |
|  |  | OP_HRCR [s]          | 32000 kHz  |
|  |  | OP_HRCR [s]+1        | $32000 * (1 + 0.18\% * 1) = 32\ 057.6$ kHz       |
|  |  | OP_HRCR [s]+2        | $32000 * (1 + 0.18\% * 2) = 32\ 115.2$ kHz       |
|  |  | ...                  | ...  |
|  |  | OP_HRCR [s]+n        | $32000 * (1 + 0.18\% * n)$ kHz                   |
|  | <p>Note:</p> <ol style="list-style-type: none"> <li>1. After each power-on of the IC, the value of OP_HRCR[7: 0] is the value of the HRC closest to 32/16/8/4 MHz; With the help of EEPROM, the user can correct the value of HRC after each power-on to allow HRC of the IC to work at Frequency required by users;</li> <li>2. In order to ensure the reliable operation of the IC, the maximum operating frequency of the IC should not exceed 10% of 32 MHz, that is 35.2 MHz;</li> <li>3. Please confirm that the change of HRC frequency will not affect other functions.</li> </ol> |                      |  |

## 7.5 Low- frequency RC Oscillator and Low- frequency Clock Timer

The SC95F761XB built-in a 32 kHz RC and 32.768 kHz crystal oscillator circuit, can be used as the Base Timer clock. source.SC95F761XB, which 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          | <b>ENBTM</b>        | Low frequency Base Timer start control<br>0: Base Timer and its clock source do not start<br>1: Base Timer and its clock source start   |
| 6          | <b>BTMIF</b>        | Base Timer interrupt application flag<br>When the CPU accepts the Base Timer interrupt, this flag will be automatically cleared by hardware.  |
| 3~0        | <b>BTMFS [3: 0]</b> | Low frequency clock interrupt frequency selection<br>0000: An interrupt is generated every 15.625ms<br>0001: An interrupt is generated every 31.25ms<br>0010: An interrupt is generated every 62.5ms<br>0011: An interrupt is generated every 125ms<br>0100: An interrupt is generated every 0.25 seconds<br>0101: An interrupt is generated every 0.5 seconds<br>0110: An interrupt is generated every 1.0 seconds<br>0111: An interrupt is generated every 2.0 seconds<br>1000: An interrupt is generated every 4.0ms |

|     |   |  |
|-----|---|--|
|     |   | 1001: An interrupt is generated every 8.0 seconds<br>1010: An interrupt is generated every 16.0 seconds<br>1011: An interrupt is generated every 32.0 seconds<br>1100~1111: reserved |
| 5~4 | - | reserved   |

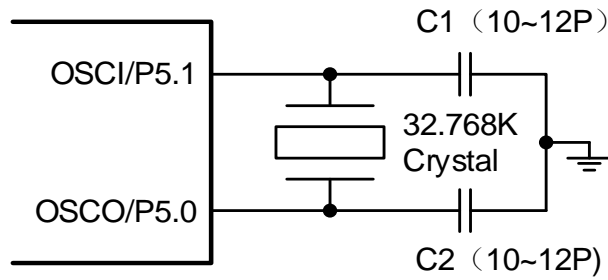
**OP\_CTM0 (C1H@FFH) Code Option Register 0 (read/write)**

| 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    | read only | R/W        |   |
| POR          | n     | n     | n           |   | n      | n         | n          |   |

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

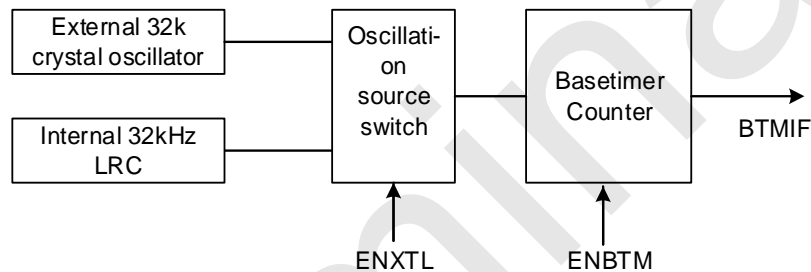
Note: HRC can be automatically calibrated by connecting a 32.768 kHz crystal oscillator. Therefore, if the user uses the 32.768 kHz external crystal oscillator function, the HRC frequency will always be corrected to 32 MHz. At this time, adjusting OP\_HRCR cannot change the HRC frequency.

The connection circuit used by P5.0/P5.1 external 32.768kHz 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 SC95F761XB 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 SC95F761XB through external interrupts INT0~2, Base Timer and CMP interrupt. Also STOP mode can be awakened by an 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 | SMOD | - | - | - | RST | - | STOP | IDL |

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

| Bit number | Bit Mnemonic | Description  |
|------------|--------------|--|
| 1          | <b>STOP</b>  | STOP mode bit. Setting this bit activates STOP mode operation. |
| 0          | <b>IDL</b>   | IDLE mode bit. Setting this bit activates IDLE mode operation. |

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

For example: set MCU to enter STOP mode:

Example in C Language

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

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

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

```
_nop_();
```

```
_nop_();
```

```
_nop_();
```

```
_nop_();
```

```
_nop_();
```

```
_nop_();
```

```
_nop_();
```

```
.....
```

Assembly Language:

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

NOP ; At least 8 NOPs are requiredNOP

NOP

NOP

NOP

NOP

NOP

NOP

.....

## 8 CPU and Instruction Set

### 8.1 CPU

The SC95F761XB 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 SC95F761XB are: ① Immediate Addressing ② Direct Addressing ③ Indirect Addressing ④ Register Addressing ⑤ Relative Addressing ⑥ Indexed Addressing ⑦ Bit Addressing.

#### 8.2.1 Immediate Addressing

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

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

#### 8.2.2 Direct Addressing

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

Examples are as follows:

ANL 50H, #91H

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

#### 8.2.3 Indirect Addressing



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

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

### 8.2.4 Register Addressing

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

```
INC R0 Refers to(R0)+1→R0
```

### 8.2.5 Relative Addressing

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

```
JC $ +50H
```

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

### 8.2.6 Indexed Addressing

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

```
MOVC A, @A+DPTR
```

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

### 8.2.7 Bits Addressing

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

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

## 9 Interrupts

SC95F761XB provides 19 interrupt sources: TIMER 0~4, INT0~2, ADC, PWM, UART, USCIO~5, BASE TIMER, CMP. The 19 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 SC95F761XB interrupt sources, interrupt vectors, and related control bits are as follows:

| Interrupt Source | Interrupt condition                     | Interrupt Flag | Interrupt Enable Control | Interrupt Priority Control | Interrupt Vector | Query Priority | Interrupt Number (C51) | Flag Clear Mode | Capability of Waking up STOP |
|------------------|---|----------------|--------------------------|----------------------------|------------------|----------------|------------------------|-----------------|------------------------------|
| INT0             | External interrupt 0 conditions are met | IE0            | EINT0                    | IPINT0                     | 0003H            | 1 (HIGH)       | 0                      | H/W Auto        | YES                          |
| Timer 0          | Timer 0 overflow                        | TF0            | ET0                      | IPT0                       | 000BH            | 2              | 1                      | H/W Auto        | NO                           |
| INT1             | External interrupt 1 conditions are met | IE1            | EINT1                    | IPINT1                     | 0013H            | 3              | 2                      | H/W Auto        | YES                          |
| Timer 1          | Timer 1 overflow                        | TF1            | ET1                      | IPT1                       | 001BH            | 4              | 3                      | H/W Auto        | NO                           |
| UART             | Receive or send completed               | RI/TI          | EUART                    | IPUART                     | 0023H            | 5              | 4                      | Must user Clear | NO                           |
| Timer 2          | Timer 2 overflow                        | TFX            | ET2                      | IPT2                       | 002BH            | 6              | 5                      | Must user Clear | NO                           |
| ADC              | ADC conversion completed                | ADCIF          | EADC                     | IPADC                      | 0033H            | 7              | 6                      | Must user Clear | NO                           |
| USCIO            | Receive or send completed               | SPIF0/TWIF0    | EUSCIO                   | 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 |
| CMP     | Comparator interrupt condition met      | CMPIF       | ECMP   | IPCMP  | 0063H | 13 | 12 | Must user Clear | YES |
| Timer 3 | Timer 3 overflow                        | TFX         | ET3    | IPT3   | 006BH | 14 | 13 | Must user Clear | NO  |
| Timer 4 | Timer 4 overflow                        | TFX         | ET4    | IPT4   | 0073H | 15 | 14 | Must user Clear | NO  |
| USCI1   | Receive or send completed               | SPIF1/TWIF1 | EUSCI1 | IPSPI1 | 007BH | 16 | 15 | Must user Clear | NO  |
| USCI2   | Receive or send completed               | SPIF2/TWIF2 | EUSCI2 | IPSPI2 | 0083H | 17 | 16 | Must user Clear | NO  |
| USCI3   | Receive or send completed               | SPIF3/TWIF3 | EUSCI3 | IPSPI3 | 008BH | 18 | 17 | Must user Clear | NO  |
| USCI4   | Receive or send completed               | SPIF4/TWIF4 | EUSCI4 | IPSPI4 | 0093H | 19 | 18 | Must user Clear | NO  |
| USCI5   | Receive or send completed               | SPIF5/TWIF5 | EUSCI5 | IPSPI5 | 009BH | 20 | 19 | 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 Timer 0 or Timer 1 overflows and the interrupt flag TF0 or TF1 is set to “1”. When the microcontroller unit responds to the timer interrupt, the interrupt flag TF0 or TF1 is reset automatically by hardware. Interrupt generates when Timer 2 overflows and the interrupt flag TF2 is set to “1”. Once Timer 2 interrupt generates, the hardware would not automatically clear TF2 bit, which must be cleared by the user’s software.

**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.

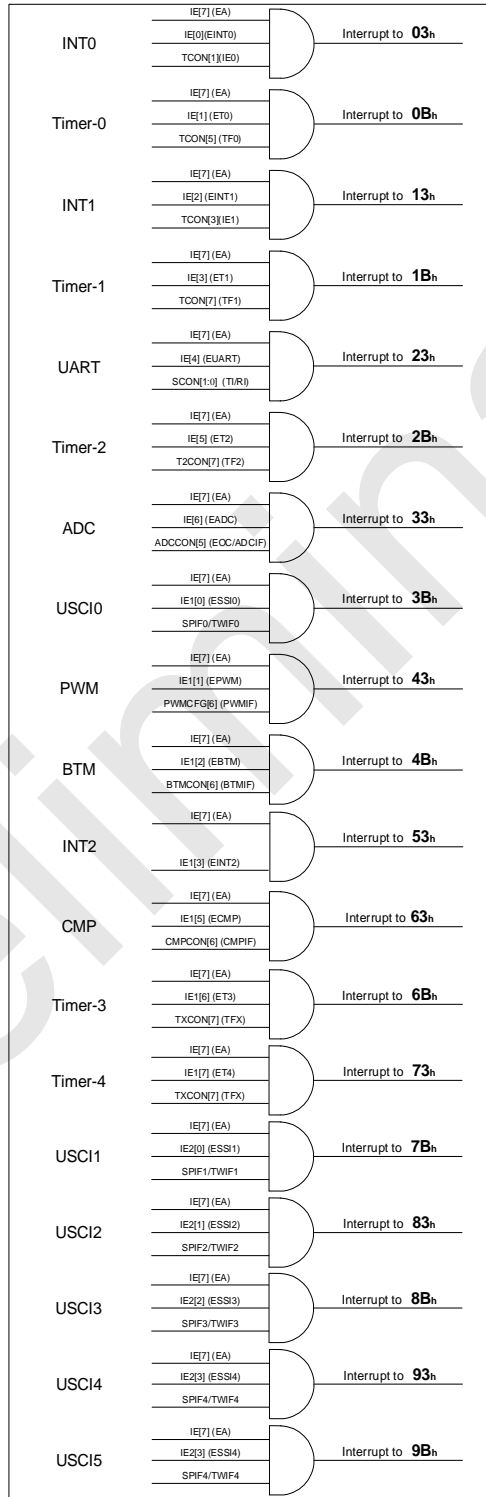
**SSI Interrupt:** When SSI completes receiving or transmitting a frame of data, SPIF/TWIF bit will be set to “1” automatically by hardware, and SSI interrupt generates. When the microcontroller unit serves SSI interrupt, the interrupt flag SPIF/TWIF must be cleared by software.

**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 occurs. Once PWM interrupt occurs, the hardware would not clear the interrupt flag automatically, which shall be cleared by user’s software.

**External Interrupt INT0 ~ 2:** When any external interrupt pin meets the interrupt conditions, external interrupt generates. The external interrupt INT0 and INT1 would set up interrupt flag IE0 and IE1 respectively, which will be automatically cleared by hardware rather than user. User can set the priority level of each interrupt through IP register. Besides, external interrupt INT0 ~ 2 can also wake up STOP mode of microcontroller unit.

## 9.2 Interrupt Structure Diagram

The interrupt structure of SC95F761XB is shown below:



SC95F761XB Interrupt structure and vector

### 9.3 Interrupt Priority

The SC95F761XB micro controller unit has two-level interrupt priority capability. The interrupt requests of these interrupt sources can be programmed as high-priority interrupt or low-priority interrupt, which is to realize the nesting of two levels of interrupt service programs. One interrupt can be interrupted by a higher priority interrupt request when being responded to, which can not be interrupted by another interrupt request at the same priority level, until such response to the first-come interrupt ends up with the instruction “RETI”. Exist the interrupt service routine and return to main program, the system would execute one more instruction before responding to new interrupt request.

That is to say:

- ① A lower priority interrupt can be interrupted by a higher priority interrupt request, but not vice versa;
- ② Any kind of interrupt being responded to can not be interrupted by another interrupt request at the same priority level.

Interrupt query sequence: As for the sequence of that the SC95F761XB microcontroller unit responds to the same priority interrupts which occur in the meantime, the priority sequence of interrupt response shall be the same as the interrupt query number in C51, which is to preferentially respond to the interrupt with smaller query number than the interrupt with bigger query number.

### 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          | <b>EA</b>    | Interrupt enable total control<br>0: Close all interrupts<br>1: Enable all interrupts   |
| 6          | <b>EADC</b>  | ADC interrupt enable control<br>0: Disable ADC interrupt<br>1: Allow the ADC to generate an interrupt when the conversion is complete |
| 5          | <b>ET2</b>   | Timer 2 interrupt enable control<br>0: Disable Timer 2 interrupt<br>1: Enable Timer 2 interrupt                                       |
| 4          | <b>EUART</b> | UART interrupt enable control<br>0: Disable UART interrupt<br>1: Allow UART interrupt   |
| 3          | <b>ET1</b>   | Timer 1 interrupt enable control<br>0: Disable Timer<br>1 interrupt1: Enable Timer 1 interrupt  |
| 2          | <b>EINT1</b> | External interrupt 1 enable control<br>0: close INT1 interrupt<br>1: Enable INT1 interrupt  |

|   |              |   |
|---|--------------|---|
| 1 | <b>ET0</b>   | Timer 0 interrupt enable control<br><br>0: Disable TIMER0 interrupt<br><br>1: Enable TIMER0 interrupt |
| 0 | <b>EINT0</b> | External interrupt 0 enable control<br><br>0: close INT0 interrupt<br><br>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          | <b>IPADC</b>  | ADC interrupt priority selection<br><br>0: ADC interrupt priority is low<br><br>1: ADC interrupt priority is high             |
| 5          | <b>IPT2</b>   | Timer 2 interrupt priority selection<br><br>0: Timer 2 interrupt priority is low<br><br>1: Timer 2 interrupt priority is high |
| 4          | <b>IPUART</b> | UART interrupt priority selection   |



|   |               |   |
|---|---------------|---|
|   |               | 0: UART interrupt priority is low<br>1: UART interrupt priority is high   |
| 3 | <b>IPT1</b>   | Timer 1 interrupt priority selection<br>0: Timer 1 interrupt priority is low<br>1: Timer 1 interrupt priority is high |
| 2 | <b>IPINT1</b> | INT1 counter interrupt priority selection<br>0: INT1 interrupt priority is low<br>1: INT1 interrupt priority is high  |
| 1 | <b>IPT0</b>   | Timer 0 interrupt priority selection<br>0: Timer 0 interrupt priority is low<br>1: Timer 0 interrupt priority is high |
| 0 | <b>IPINT0</b> | INT0 counter interrupt priority selection<br>0: INT0 interrupt priority is low<br>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 | ECMP | - | EINT2 | EBTM | EPWM | EUSCIO |
| R/W          | R/W | R/W | R/W  | - | R/W   | R/W  | R/W  | R/W    |
| POR          | 0   | 0   | 0    | x | 0     | 0    | 0    | 0      |

| Bit number | Bit Mnemonic  | Description   |
|------------|---------------|---|
| 7          | <b>ET4</b>    | Timer 4 interrupt enable control<br>0: Disable Timer 4 interrupt<br>1: Enable Timer 4 interrupt                                     |
| 6          | <b>ET3</b>    | Timer 3 interrupt enable control<br>0: Disable Timer 3 interrupt<br>1: Enable Timer 3 interrupt                                     |
| 5          | <b>ECMP</b>   | Analog comparator interrupt enable control<br>0: Disable the analog comparator interrupt<br>1: Open the analog comparator interrupt |
| 3          | <b>EINT2</b>  | External interrupt 2 enable control<br>0: close INT2 interrupt<br>1: Open INT2 interrupt  |
| 2          | <b>EBTM</b>   | Base Timer interrupt enable control<br>0: Disable Base Timer interrupt<br>1: Enable Base Timer interrupt                            |
| 1          | <b>EPWM</b>   | PWM interrupt enable control<br>0: Disable PWM interrupt<br>1: Enable interrupt when PWM count overflows                            |
| 0          | <b>EUSCIO</b> | Three-in-one serial port USCIO interrupt enable control<br>0: Disable serial port interrupt<br>1: Allow serial port interrupt       |
| 4          | -             | Reserved  |

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

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

| Bit number | Bit Mnemonic  | Description   |
|------------|---------------|---|
| 7          | <b>IPT4</b>   | Timer 4 interrupt priority selection<br>0: Timer 4 interrupt priority is low<br>1: Timer 4 interrupt priority is high                               |
| 6          | <b>IPT3</b>   | Timer 3 interrupt priority selection<br>0: Timer 3 interrupt priority is low<br>1: Timer 3 interrupt priority is high                               |
| 5          | <b>IPCMP</b>  | Analog comparator interrupt priority selection<br>0: Analog comparator interrupt priority is low<br>1: Analog comparator interrupt priority is high |
| 3          | <b>IPINT2</b> | INT2 counter interrupt priority selection<br>0: INT2 interrupt priority is low<br>1: INT2 interrupt priority is high                                |
| 2          | <b>IPBTM</b>  | Base Timer interrupt priority selection<br>0: Base Timer interrupt priority is low  |

|   |                |  |
|---|----------------|--|
|   |                | 1: Base Timer interrupt priority is high   |
| 1 | <b>IPPWM</b>   | PWM interrupt enable selection<br>0: PWM interrupt priority is low<br>1: PWM interrupt priority is high                                  |
| 0 | <b>IPUSCIO</b> | Three-in-one serial port USCIO interrupt priority selection<br>0: USCIO interrupt priority is low<br>1: USCIO interrupt priority is high |
| 4 | -              | Reserved   |

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

| Bit number                   | 7 | 6 | 5 | 4              | 3              | 2              | 1              | 0              |
|------------------------------|---|---|---|----------------|----------------|----------------|----------------|----------------|
| Symbol                       | - | - | - | EUSCI5         | EUSCI4         | EUSCI3         | EUSCI2         | EUSCI1         |
| Read/<br>Write               | - | - | - | Read/<br>Write | Read/<br>Write | Read/<br>Write | Read/<br>Write | Read/<br>Write |
| Initial<br>power-on<br>value | x | x | x | 0              | 0              | 0              | 0              | 0              |

| Bit number | Bit Mnemonic  | Description  |
|------------|---------------|--|
| 4          | <b>EUSCI5</b> | Three-in-one serial port USCIO5 interrupt enable control<br>0: Disable serial port interrupt<br>1: Allow serial port interrupt |
| 3          | <b>EUSCI4</b> | Three-in-one serial port USCIO4 interrupt enable control   |

|     |               |   |
|-----|---------------|---|
|     |               | 0: Disable serial port interrupt<br>1: Allow serial port interrupt  |
| 2   | <b>EUSCI3</b> | Three-in-one serial port USCI3 interrupt enable control<br>0: Disable serial port interrupt<br>1: Allow serial port interrupt |
| 1   | <b>EUSCI2</b> | Three-in-one serial port USCI2 interrupt enable control<br>0: Disable serial port interrupt<br>1: Allow serial port interrupt |
| 0   | <b>EUSCI1</b> | Three-in-one serial port USCI1 interrupt enable control<br>0: Disable serial port interrupt<br>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 | - | - | - | IPUSCI5 | IPUSCI4 | IPUSCI3 | IPUSCI 2 | IPUSCI1 |
| R/W          | - | - | - | R/W     | R/W     | R/W     | R/W      | R/W     |
| POR          | x | x | x | 0       | 0       | 0       | 0        | 0       |

| Bit number | Bit Mnemonic   | Description   |
|------------|----------------|---|
|            | <b>IPUSCI5</b> | Three-in-one serial port USCI5 interrupt priority selection |

|     |                |  |
|-----|----------------|--|
|     |                | 0: USC15 interrupt priority is low<br>1: USC15 interrupt priority is high  |
|     | <b>IPUSCI4</b> | Three-in-one serial port USC14 interrupt priority selection<br>0: USC14 interrupt priority is low<br>1: USC14 interrupt priority is high |
|     | <b>IPUSCI3</b> | Three-in-one serial port USC13 interrupt priority selection<br>0: USC13 interrupt priority is low<br>1: USC13 interrupt priority is high |
| 1   | <b>IPUSCI2</b> | Three-in-one serial port USC12 interrupt priority selection<br>0: USC12 interrupt priority is low<br>1: USC12 interrupt priority is high |
| 0   | <b>IPUSCI1</b> | Three-in-one serial port USC11 interrupt priority selection<br>0: USC11 interrupt priority is low<br>1: USC11 interrupt priority is high |
| 7~5 | -              | Reserved   |

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

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

| Bit number | Bit Mnemonic | Description  |
|------------|--------------|--|
| 3          | <b>IE1</b>   | 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          | <b>IE0</b>   | 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 | INT0F7 | INT0F6 | INT0F5 | INT0F4 | - | - | - | - |
| R/W          | R/W    | R/W    | R/W    | R/W    | - | - | - | - |
| POR          | 0      | 0      | 0      | 0      | x | x | x | x |

| Bit number | Bit Mnemonic                    | Description  |
|------------|---------------------------------|--|
| 7~4        | <b>INT0Fn</b><br><b>(n=7~4)</b> | INT0 falling edge interrupt control<br>0: INT0n falling edge interrupt close<br>1: INT0n falling edge interrupt enable |
| 3~0        | -                               | Reserved   |

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

| Bit number   | 7      | 6      | 5      | 4      | 3 | 2 | 1 | 0 |
|--------------|--------|--------|--------|--------|---|---|---|---|
| Bit Mnemonic | INT0R7 | INT0R6 | INT0R5 | INT0R4 | - | - | - | - |
| R/W          | R/W    | R/W    | R/W    | R/W    | - | - | - | - |
| POR          | 0      | 0      | 0      | 0      | x | x | x | x |

| Bit number | Bit Mnemonic                    | Description   |
|------------|---------------------------------|---|
| 7~4        | <b>INT0Rn</b><br><b>(n=7~4)</b> | INT0 rising edge interrupt control<br>0: INT0n rising edge interrupt close<br>1: INT0n rising edge interrupt enable |
| 3~0        | -                               | Reserved  |

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

| Bit number   | 7      | 6      | 5      | 4      | 3      | 2      | 1      | 0      |
|--------------|--------|--------|--------|--------|--------|--------|--------|--------|
| Bit Mnemonic | INT1F7 | INT1F6 | INT1F5 | INT1F4 | INT1F3 | INT1F2 | INT1F1 | INT1F0 |
| 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        | <b>INT1Fn</b><br><b>(n=7~0)</b> | INT1 falling edge interrupt control<br>0: INT1n falling edge interrupt close |



|  |  |  |
|--|--|--|
|  |  | 1: INT1n falling edge interrupt enable |
|--|--|--|

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

| Bit number   | 7      | 6      | 5      | 4      | 3      | 2      | 1      | 0      |
|--------------|--------|--------|--------|--------|--------|--------|--------|--------|
| Bit Mnemonic | INT1R7 | INT1R6 | INT1R5 | INT1R4 | INT1R3 | INT1R2 | INT1R1 | INT1R0 |
| 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        | <b>INT1Rn</b><br><b>(n=7~0)</b> | INT1 rising edge interrupt control<br>0: INT1n rising edge interrupt off<br>1: INT1n rising edge interrupt enable |

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

| Bit number   | 7 | 6 | 5 | 4 | 3      | 2      | 1      | 0      |
|--------------|---|---|---|---|--------|--------|--------|--------|
| Bit Mnemonic | - | - | - | - | INT2F3 | INT2F2 | INT2F1 | INT2F0 |
| 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        | <b>INT2Fn</b><br><b>(n=3~0)</b> | INT2 falling edge interrupt control<br>0: INT2n falling edge interrupt close<br>1: INT2n falling edge interrupt enable |
| 7~4        | -                               | Reserved   |

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

| Bit number   | 7 | 6 | 5 | 4 | 3      | 2      | 1      | 0      |
|--------------|---|---|---|---|--------|--------|--------|--------|
| Bit Mnemonic | - | - | - | - | INT2R3 | INT2R2 | INT2R1 | INT2R0 |
| 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        | <b>INT2Rn</b><br><b>(n=3~0)</b> | INT2 rising edge interrupt control<br>0: INT2n rising edge interrupt close<br>1: INT2n rising edge interrupt enable |
| 7~4        | -                               | Reserved  |

## 10 Timer/Counter T0 and T1

Timer 0 and Timer 1 inside the SC95F761XB 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 P0.2/T0 and P0.3/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 | USMDX[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          | <b>TF1</b>   | 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          | <b>TR1</b>   | 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          | <b>TF0</b>   | 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          | <b>TR0</b>   | 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          | <b>C/T1</b>    | TMOD[6] control timer 1<br>0: Timer, T1 count comes from fsys frequency division<br>1: Counter, T1 count comes from external pin T1/P0.3   |
| 5~4        | <b>M11,M01</b> | Timer/Counter 1 mode selection<br>00: 13-bit timer/counter, the upper 3 bits of TL1 are invalid<br>01: 16-bit timer/counter, TL1 and TH1 all are valid<br>10: 8-bit auto-reload timer, automatically reload the value stored in TH1 into TL1 when overflow<br>11: Timer/Counter 1 is invalid (stop counting) |
| 2          | <b>C/T0</b>    | TMOD[2] control timer 0<br>0: Timer, T0 count comes from fsys frequency division<br>1: Counter, T0 count comes from external pin T0/P0.2   |
| 1~0        | <b>M10,M00</b> | Timer/Counter 0 mode selection<br>00: 13-bit timer/counter, the upper 3 bits of TL0 are invalid<br>01: 16-bit timer/counter, TL0 and TH0 all are valid<br>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 | USMDX[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          | <b>T1FD</b>  | T1 input frequency selection control<br>0: T1 frequency is derived from fsys/12<br>1: T1 frequency is derived from fsys |
| 0          | <b>T0FD</b>  | T0 input frequency selection control<br>0: T0 frequency is derived from fsys/12<br>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          | <b>ET1</b>   | Timer 1 interrupt enable control<br>0: Disable Timer 1 interrupt<br>1: Enable Timer 1 interrupt |
| 1          | <b>ET0</b>   | Timer 0 interrupt enable control<br>0: Disable Timer 0 interrupt<br>1: Enable Timer 0 interrupt |

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

| Bit number   | 7 | 6     | 5    | 4      | 3    | 2      | 1    | 0      |
|--------------|---|-------|------|--------|------|--------|------|--------|
| Bit Mnemonic | - | IPADC | IPT2 | 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 | <b>IPT1</b> | Timer 1 interrupt priority<br><br>0: Set the interrupt priority of Timer 1 to "Low"<br><br>1: Set the interrupt priority of Timer 1 to "High" |
| 1 | <b>IPT0</b> | Timer 0 interrupt priority<br><br>0: Set the interrupt priority of Timer 0 to "Low"<br><br>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.

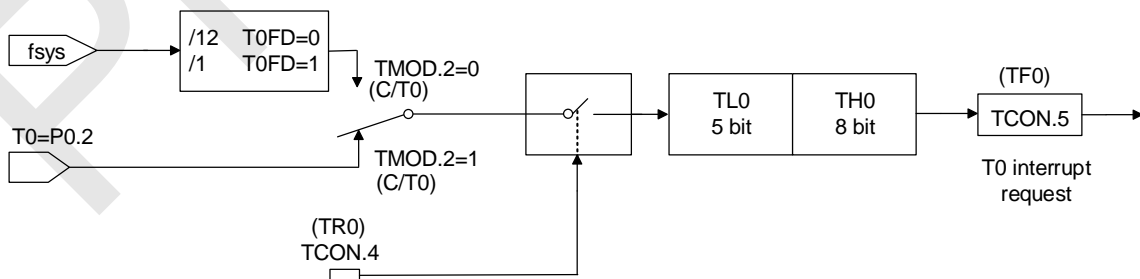
### Operating Mode 0: 13-bit Counter/Timer

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.

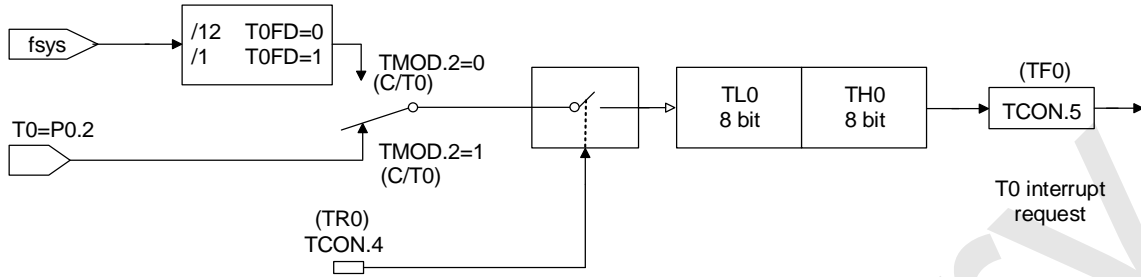


Timer/counter operating mode 0: 13-bit timer/counter

### 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.



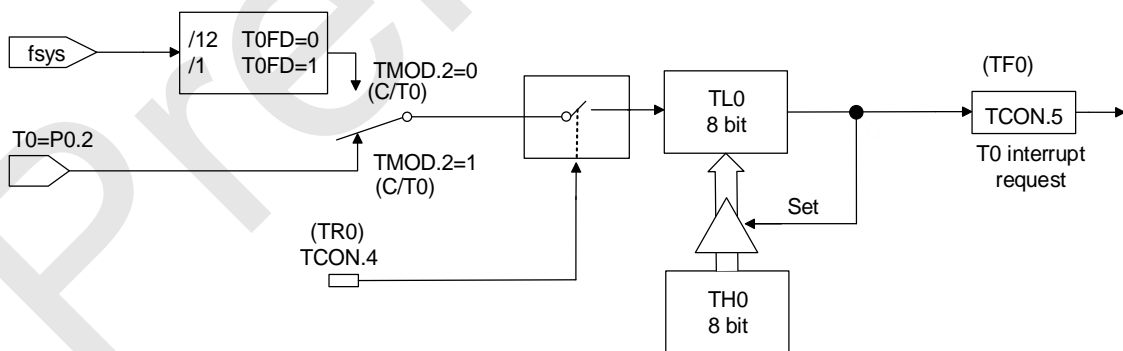
Timer/Counter Operating Mode 1: 16-bit Timer/Counter

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

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

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

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



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

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

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

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

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

### 10.3 T1 Operating Mode

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

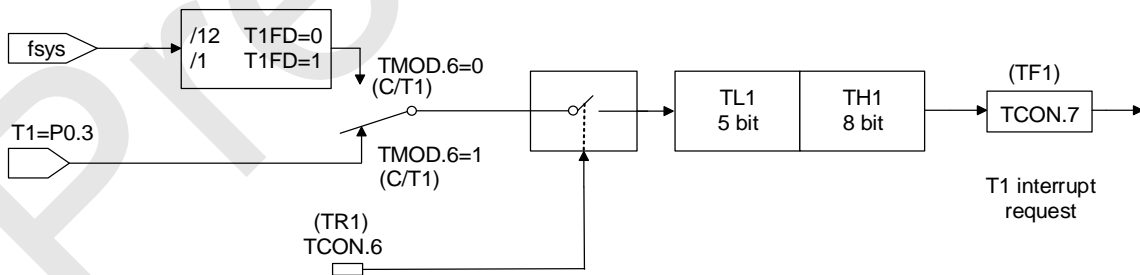
#### Operating mode 0: 13-bit Timer/Counter

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

If C/T1=1, the level of timer 1 input pin T1 (P0.3) changes from high to low, which will increase the timer 1 data register by 1. If C/T1=0, select the frequency division of the system clock as the clock source of timer 1.

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

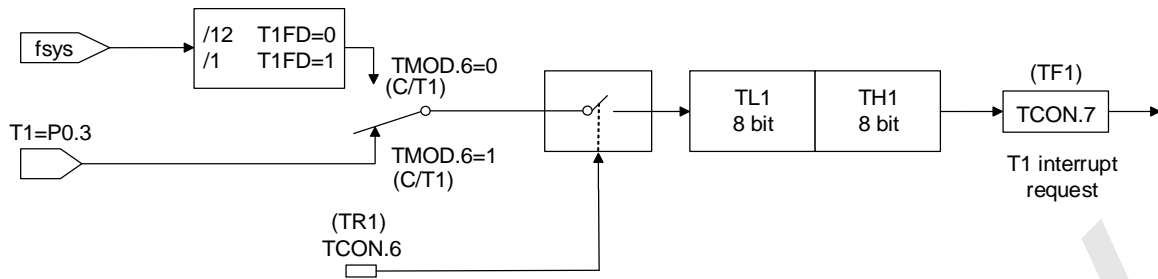
When applied as a timer, T1FD can be configured to select the frequency division ratio of the clock source.



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

#### Operating mode 1: 16-bit Counter/Timer

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



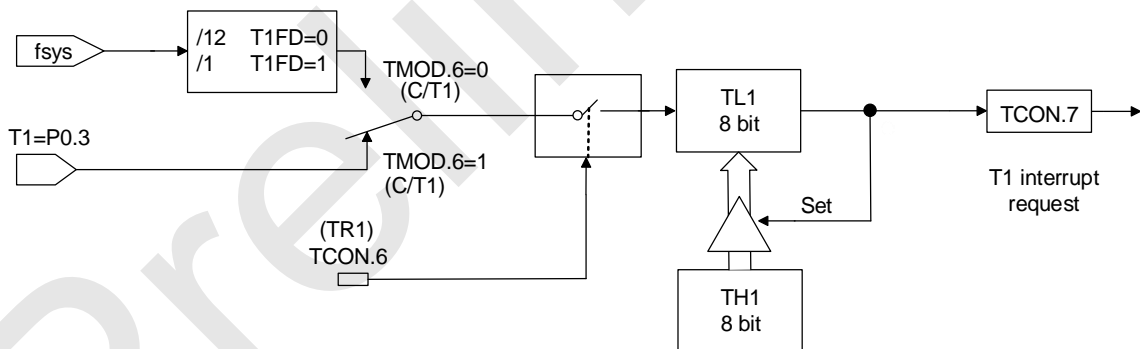
Timer/counter operating mode 1: 16-bit timer/counter

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

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

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

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



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

## 11 Timer/Counter T2/T3/T4

Timer 2/3/4 inside The SC95F761XB MCU are three independent Timers, among which Timer 2 has 4 operating modes, Timer 3 and Timer 4 have 1 operating mode.

The control registers of Timer 2/3/4 share the same set of addresses (C8H-CDH), users can point the Timer X register set (TXCON / TXMOD / RCAPXL / RCAPXH / TLX / THX) to Timer 2/3/4 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 Timer X register group will point to the Timer 2/3/4 specified by the user. At this time, operating the TimeX register group is an effective operation for the corresponding Timer.**

### 11.1 T2/3/4-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<br>X  | TCL<br>KX  | EXEN<br>X | TRX         | C/TX     | CP/RL<br>X | 00000000b |
| TXMOD  | C9H     | Timer 2/3/4 operating mode register  | TXF<br>D     | -    | EPW<br>MN1 | EPW<br>MN0 | INVN1     | INVN<br>0   | TXO<br>E | DCXE<br>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     | USMDX[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        | <b>TXINX[2: 0]</b> | Timer 2/3/4 control register pointer<br><br>010: Timer X register set: TXCON / TXMOD / RCAPXL / RCAPXH / TLX / THX points to T2<br><br>011: Timer X register set points to T3<br><br>100: Timer X register set points to T4<br><br>Other: reserved |
| 7~3        | -                  | Reserved   |

## 11.2 Timer 2

Timer 2 inside the SC95F761XB 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  $f_{sys}/12$  or  $f_{sys}$  through the special function register TXMOD.7 (TXFD).

Timer/counter T2 has 4 operating modes:

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

- ③ Mode 2: Baud rate generator mode
- ④ Mode 3: Programmable clock output mod

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

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

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

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

|   |               |  |
|---|---------------|--|
|   |               | 1: Timer 2 generates transmission baud rate  |
| 3 | <b>EXENX</b>  | T2EX pin is used as a reload/capture trigger enable/disable control:<br><br>0: Ignore events on T2EX pin<br><br>1: When Timer 2 is not used as the UART0 clock, a falling edge on the T2EX pin is detected, and a capture or reload is generated |
| 2 | <b>TRX</b>    | Timer 2 start/stop control bit<br><br>0: stop timer 2/stop PWM2 counter<br><br>1: Start timer 2/start PWM2 counter   |
| 1 | <b>C/TX</b>   | Timer 2 Timer/counter mode selection positioning 2<br><br>0: Timer mode, T2 pin is used as I/O port<br><br>1: Counter mode   |
| 0 | <b>CP/RLX</b> | Capture/reload mode selection positioning<br><br>0: 16-bit timer/counter with reload function<br><br>1: 16-bit timer/counter with capture function, TXEX is timer 2 external capture signal input port   |

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

| Bit number   | 7    | 6 | 5      | 4      | 3     | 2     | 1    | 0     |
|--------------|------|---|--------|--------|-------|-------|------|-------|
| Bit Mnemonic | TXFD | - | EPWM21 | EPWM20 | INV21 | INV20 | TXOE | DCXEN |
| R/W          | R/W  | - | R/W    | R/W    | R/W   | R/W   | R/W  | R/W   |
| POR          | 0    | x | 0      | 0      | 0     | 0     | 0    | 0     |

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

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

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

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

| Bit number | Bit Mnemonic | Description   |
|------------|--------------|---|
| 5          | <b>ET2</b>   | Timer 2 interrupt enable control<br>0: Disable Timer 2 interrupt<br>1: Enable Timer 2 interrupt |



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

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

| Bit number | Bit Mnemonic | Description   |
|------------|--------------|---|
| 5          | <b>IPT2</b>  | Timer 2 interrupt priority<br>0: Set the interrupt priority of Timer 2 to "Low"<br>1: Set the interrupt priority of Timer 2 to "High" |

### 11.3 Timer 3

Timer 3 inside the SC95F761XB MCU as a timer is essentially an addition counter. The clock source of the timer is the system clock or its divided clock. TRX is the switch control of T3 counting. Only when TRX=1, T3 will be opened to count.

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

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

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

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

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

| Bit number | Bit Mnemonic  | Description  |
|------------|---------------|--|
| 7          | <b>TFX</b>    | Timer 3 control register (read/write) Timer 3 overflow flag<br>0: No overflow (must be cleared by software)<br>1: Overflow (set by hardware 1)   |
| 6          | <b>EXFX</b>   | Flag bit detected by external event input (falling edge) of T3EX pin<br>0: No external event input (must be cleared by software)<br>1: External input detected (if EXENX = 1, set by hardware) |
| 3          | <b>EXENX</b>  | T3EX pin is used as a reload/capture trigger enable/disable control:<br>0: Ignore events on T3EX pin<br>1: A falling edge on the T3EX pin is detected, and a capture or reload is generated    |
| 2          | <b>TRX</b>    | Timer 3 start/stop control bit<br>0: stop timer 3/stop PWM3 counter<br>1: Start timer 3/start PWM3 counter   |
| 1          | <b>C/TX</b>   | Timer 3 Timer/counter mode selection positioning 2<br>0: Timer mode, T3 pin is used as I/O port<br>1: Counter mode   |
| 0          | <b>CP/RLX</b> | Capture/reload mode selection positioning<br>0: 16-bit timer/counter with reload function<br>1: 16-bit timer/counter with capture function, TXEX is timer 3 external capture signal input port |
| 5~4        | -             | Reserved   |

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

| Bit number   | 7    | 6 | 5      | 4      | 3     | 2     | 1    | 0     |
|--------------|------|---|--------|--------|-------|-------|------|-------|
| Bit Mnemonic | TXFD | - | EPWM31 | EPWM30 | INV31 | INV30 | TXOE | DCXEN |
| R/W          | R/W  | - | R/W    | R/W    | R/W   | R/W   | R/W  | R/W   |
| POR          | 0    | x | 0      | 0      | 0     | 0     | 0    | 0     |

| Bit number | Bit Mnemonic | Description   |
|------------|--------------|---|
| 7          | <b>TXFD</b>  | T3 input frequency selection control<br>0: T3 frequency is derived from fsys/12<br>1: T3 frequency is derived from fsys   |
| 1          | <b>TXOE</b>  | Timer 3 output enable bit<br>0: Set T3 as clock input or I/O port<br>1: Set T3 as the clock output  |
| 0          | <b>DCXEN</b> | Count down enable bit<br>0: Timer 3 is prohibited as an up/down counter, Timer 3 is only used as an up counter<br>1: Allow Timer 3 as an up/down counter, T3EX is used to select the counting direction |
| 6          | -            | Reserved  |

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

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

| Bit number | Bit Mnemonic | Description   |
|------------|--------------|---|
| 6          | <b>ET3</b>   | Timer 3 interrupt enable control<br>0: Disable Timer 3 interrupt<br>1: Enable Timer 3 interrupt |
| 4          | -            | Reserved  |

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

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

| Bit number | Bit Mnemonic | Description                          |
|------------|--------------|--------------------------------------|
| 6          | <b>IPT3</b>  | Timer 3 interrupt priority selection |

|   |   |   |
|---|---|---|
|   |   | 0: Timer 3 interrupt priority is low<br>1: Timer 3 interrupt priority is high |
| 4 | - | Reserved  |

## 11.4 Timer 4

Timer 4 inside the SC95F761XB MCU as a timer is essentially an addition counter. The clock source of the timer is the system clock or its divided clock. TRX is the switch control of T4 count. Only when TRX=1, T4 will be turned on and counted.

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

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

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

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

| Bit number | Bit Mnemonic | Description  |
|------------|--------------|--|
| 7          | <b>TFX</b>   | Timer 4 overflow flag<br>0: No overflow (must be cleared by software)<br>1: Overflow (set by hardware 1)   |
| 6          | <b>EXFX</b>  | Flag bit detected by external event input (falling edge) of T4EX pin<br>0: No external event input (must be cleared by software)<br>1: External input detected (if EXENX = 1, set by hardware) |

|     |               |  |
|-----|---------------|--|
| 3   | <b>EXENX</b>  | T4EX pin is used as a reload/capture trigger enable/disable control:<br><br>0: Ignore events on T4EX pin<br><br>1: A falling edge on the T4EX pin is detected, and a capture or reload is generated    |
| 2   | <b>TRX</b>    | Timer 4 start/stop control bit<br><br>0: stop timer 4/stop PWM4 counter<br><br>1: Start timer 4/start PWM4 counter   |
| 1   | <b>C/TX</b>   | Timer 4 Timer/counter mode selection positioning 2<br><br>0: Timer mode, T4 pin is used as I/O port<br><br>1: Counter mode   |
| 0   | <b>CP/RLX</b> | Capture/reload mode selection positioning<br><br>0: 16-bit timer/counter with reload function<br><br>1: 16-bit timer/counter with capture function, TXEX is timer 4 external capture signal input port |
| 5~4 | -             | Reserved   |

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

| Bit number   | 7    | 6 | 5      | 4      | 3     | 2     | 1    | 0     |
|--------------|------|---|--------|--------|-------|-------|------|-------|
| Bit Mnemonic | TXFD | - | EPWM41 | EPWM40 | INV41 | INV40 | TXOE | DCXEN |
| R/W          | R/W  | - | R/W    | R/W    | R/W   | R/W   | R/W  | R/W   |
| POR          | 0    | x | 0      | 0      | 0     | 0     | 0    | 0     |

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

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

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

| Bit number | Bit Mnemonic | Description                      |
|------------|--------------|----------------------------------|
| 7          | <b>ET4</b>   | Timer 4 interrupt enable control |

|   |   |   |
|---|---|---|
|   |   | 0: Disable Timer 4 interrupt<br>1: Enable Timer 4 interrupt |
| 4 | - | Reserved  |

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

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

| Bit number | Bit Mnemonic | Description   |
|------------|--------------|---|
| 6          | <b>IPT4</b>  | Timer 4 interrupt priority selection<br>0: Timer 4 interrupt priority is low<br>1: Timer 4 interrupt priority is high |
| 4          | -            | Reserved  |

## 11.5 Timer 2/3/4 Operating Modes

Timer 2/3/4's operating modes as follows:

- ① Mode 0: 16-bit capture
- ② Mode 1: 16-bit auto-reload timer
- ③ Mode 2: Baud rate generator, only Timer 2 support this mode
- ④ Mode 3: Programmable clock output



⑤ Mode 4: PWM output mode

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

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

### 11.5.1 Timer 2/3/4 Operating Modes

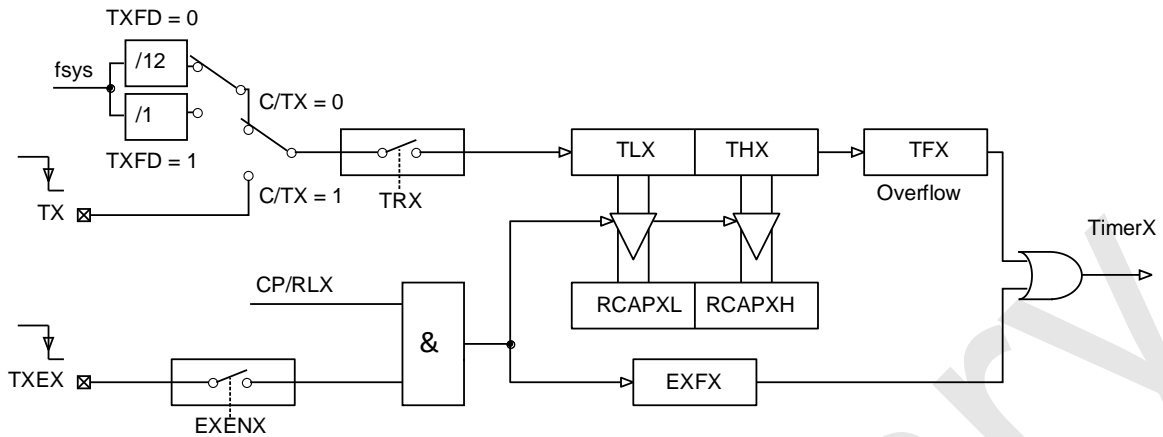
#### Operating Mode 0: 16-bit Capture

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

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

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

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



Mode 0: 16-bit capture

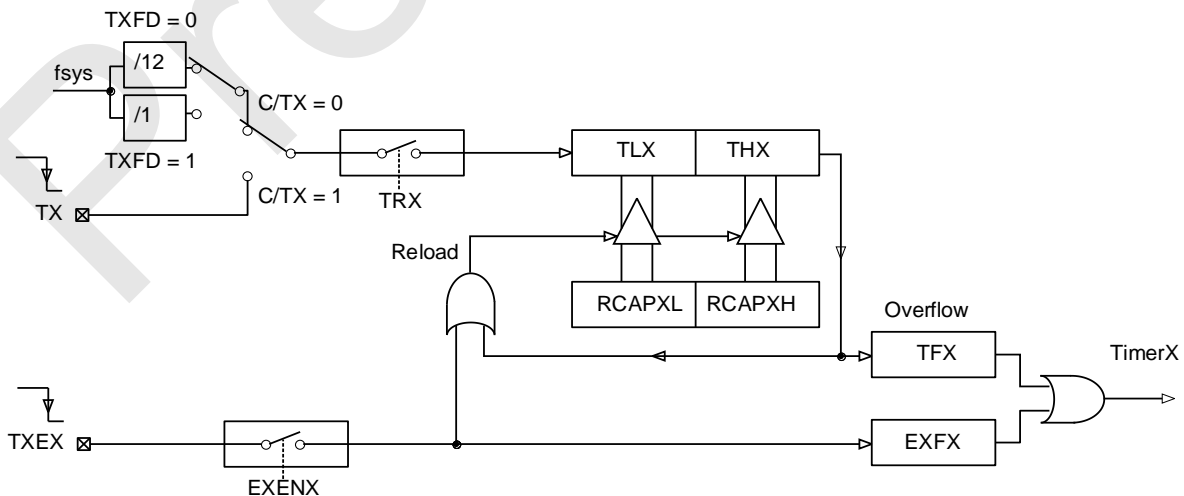
### Operating Mode 1: 16-bit Auto-Reload Timer

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

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

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

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



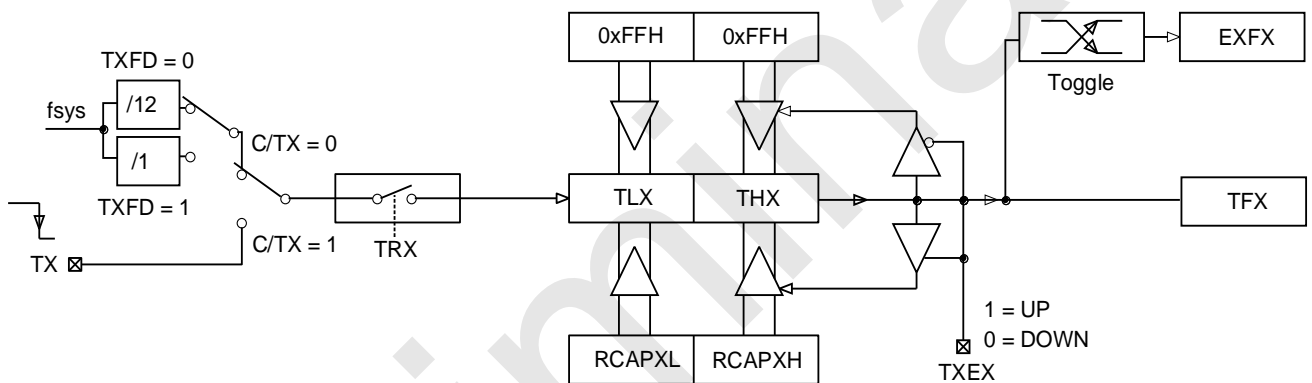
Mode 1: 16-bit auto-reload DCEN = 0

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

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

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

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



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

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

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

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

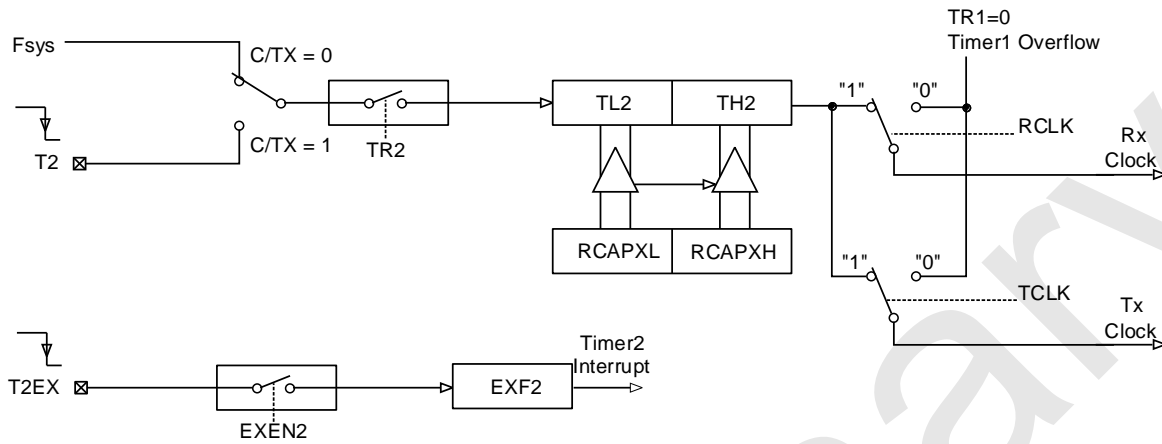
The overflow of Timer 2 will reload the values in the RCAPXH and RCAPXL registers to the Timer 2 count, but no interrupt will be generated

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

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

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

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



Mode 2: Baud rate generator

### Operating Mode 3: Programmable Clock Output

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

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

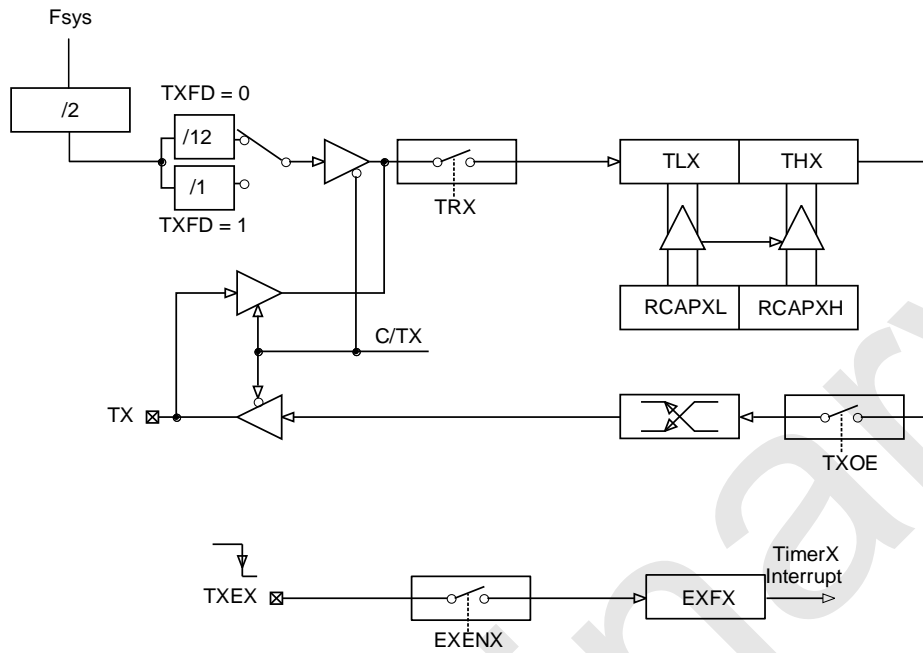
$$\text{Clock Out Frequency} = \frac{f_n}{(65536 - [RCAPXH, RCAPXL]) \times 4};$$

Among them,  $f_n$  is the timer n clock frequency:

$$f_n = \frac{f_{sys}}{12}; \quad TXFD = 0$$

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

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



Mode 3: Programmable clock output

**Note:**

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

## 12 PWM2/3/4

The SC95F761XB provides up to 14 PWM, which divided into two categories:

1. Multi-function PWM: 8 channels for only one group PWM0, output signal port: PWM00~07
2. Conventional PWM: 6 channels divided into 3 groups: PWM2, PWM3, PWM4. **Note: These three sets of PWM period registers are shared with the TLX and THX of Timer2, Timer3, and Timer4 respectively. Therefore, once users use the PWM2, PWM3, and PWM4 resources, they cannot change the timing/count value of Timer2, Timer3, and Timer4. Otherwise it will lead to abnormal OUTPUT of PWM cycle!**

### 12.1 PWM2/3/4 related Registers

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

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

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

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

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

| Bit number | Bit Mnemonic | Description  |
|------------|--------------|--|
| 2          | <b>TRX</b>   | Timer n start/stop control bit<br>0: stop timer n/stop PWMn counter<br>1: Start timer n/start PWMn counter |

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

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

| Bit number   | 7    | 6 | 5      | 4      | 3     | 2     | 1    | 0     |
|--------------|------|---|--------|--------|-------|-------|------|-------|
| Bit Mnemonic | TXFD | - | EPWM21 | EPWM20 | INV21 | INV20 | TXOE | DCXEN |
| R/W          | R/W  | - | R/W    | R/W    | R/W   | R/W   | R/W  | R/W   |
| POR          | 0    | x | 0      | 0      | 0     | 0     | 0    | 0     |

| Bit number | Bit Mnemonic   | Description   |
|------------|----------------|---|
| 5          | <b>ENPWMn1</b> | PWMn1 Waveform output select<br>0: PWMn1 output is disabled |

|   |                |   |
|---|----------------|---|
|   |                | 1: I/O where PWMn1 resides serves as the output port of the PWM waveform  |
| 4 | <b>ENPWMn0</b> | PWMn0 Waveform output select<br><br>0: PWMn0 output is disabled<br><br>1: I/O where PWMn0 resides serves as the output port of the PWM waveform |
| 3 | <b>INVn1</b>   | PWMn1 waveform output reverse control<br><br>1: PWMn1 waveform output is reversed<br><br>0: PWMn1 waveform output is not reversed               |
| 2 | <b>INVn0</b>   | PWMn0 waveform output reverse control<br><br>1: PWMn0 waveform output is reversed<br><br>0: PWMn0 waveform output is not reversed               |

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

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

$$T_{pwm} = \frac{PWMPDX[15:0] + 1}{f_{sys}}$$

Duty calculation formula:

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

The PWM cycle is set through the following registers:

#### RCAPXH (CBH)

#### PWMn period register high 8 bits(R/W)

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

|                   |   |   |   |   |   |   |   |   |
|-------------------|---|---|---|---|---|---|---|---|
| <b>Bit number</b> | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------------------|---|---|---|---|---|---|---|---|



|              |              |     |     |     |     |     |     |     |
|--------------|--------------|-----|-----|-----|-----|-----|-----|-----|
| Bit Mnemonic | PWMPDHX[7:0] |     |     |     |     |     |     |     |
| R/W          | R/W          | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| POR          | 0            | 0   | 0   | 0   | 0   | 0   | 0   | 0   |

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

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

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

| Bit number | Bit Mnemonic        | Description  |
|------------|---------------------|--|
| 7~0        | <b>PWMPDX[15:0]</b> | PWMn cycle set<br>This value represents the output waveform of PWMn (cycle-1); that is to say PWMn cycle of output is(PWMPDX[15:0] + 1) * PW clock |

The PWM duty is set through the following registers:

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

| ADD   | 7           | 6 | 5 | 4 | 3 | 2 | 1 | 0 | POR      |
|-------|-------------|---|---|---|---|---|---|---|----------|
| 2034H | PDT20[15:8] |   |   |   |   |   |   |   | 0000000b |
| 2035H | PDT20[7:0]  |   |   |   |   |   |   |   | 0000000b |
| 2036H | PDT21[15:8] |   |   |   |   |   |   |   | 0000000b |
| 2037H | PDT21[7:0]  |   |   |   |   |   |   |   | 0000000b |
| 2038H | PDT30[15:8] |   |   |   |   |   |   |   | 0000000b |
| 2039H | PDT30[7:0]  |   |   |   |   |   |   |   | 0000000b |
| 203AH | PDT31[15:8] |   |   |   |   |   |   |   | 0000000b |
| 203BH | PDT31[7:0]  |   |   |   |   |   |   |   | 0000000b |
| 203CH | PDT40[15:8] |   |   |   |   |   |   |   | 0000000b |

|       |             |           |
|-------|-------------|-----------|
| 203DH | PDT40[7:0]  | 00000000b |
| 203EH | PDT41[15:8] | 00000000b |
| 203FH | PDT41[7:0]  | 00000000b |

| Bit number | Bit Mnemonic                         | Description  |
|------------|--------------------------------------|--|
| 7~0        | <b>PDTxy[15:0]</b><br>(x=2~4, y=0~1) | PWMxy waveform duty cycle length setting<br>The high level width of the PWMxy waveform is:<br>(PDTxy[15:0] + 1 ) PWM clock |

## 12.2 PWM2/3/4 Duty Variation Characteristics

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

## 12.3 PWM2/3/4 Cycle Variation Characteristics

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

Define the current period meter value as  $T_n$ , when writing the period register, the value recorded by the timer is  $T_m$ , and the period meter value to be updated is  $T_x$ , then:

$T_m \leq T_x$ : the period changes in real time according to  $T_x$ ;

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

## 13 PWM0

The SC95F761XB provides up to 14 PWM, which divided into two categories:

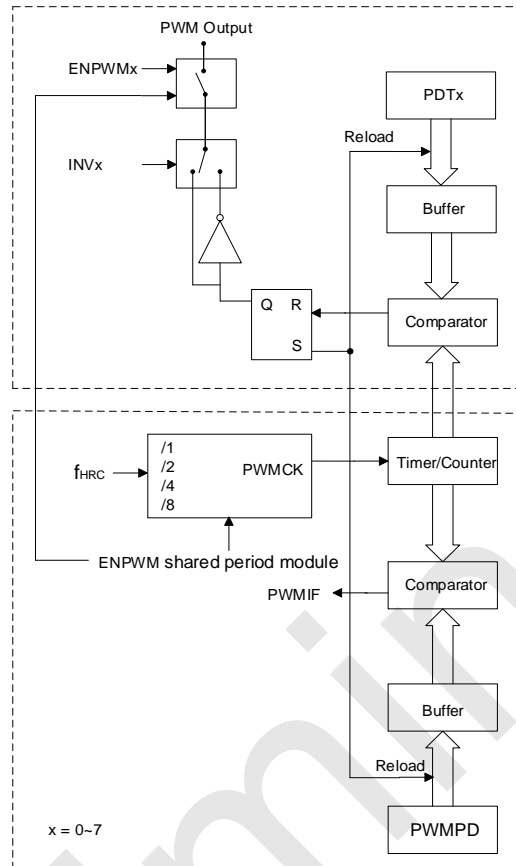
3. Multi-function PWM: 8 channels for only one group PWM0, output signal port: PWM00~07
4. Conventional PWM: 6 channels divided into 3 groups: PWM2, PWM3, PWM4. **Note: These three sets of PWM period registers are shared with the TLX and THX of Timer2, Timer3, and Timer4 respectively. Therefore, once users use the PWM2, PWM3, and PWM4 resources, they cannot change the timing/count value of Timer2, Timer3, and Timer4. Otherwise it will lead to abnormal OUTPUT of PWM cycle!**

The functions of the PWM0 of the SC95F761XB 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:
  - a) In independent mode, the 8 PWM cycles are the same, but the duty cycle of each PWM output waveform can be set separately;
  - b) 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 SC95F761XB can support the adjustment of period and duty cycle. The registers PWMCFG, PWMCON0 and PWMCON1 control the state and period of PWM. The opening of each PWM and the output waveform duty cycle can be adjusted separately.

### 13.1 PWM0 Structure Diagram



SC95F761XB PWM Structure diagram

## 13.2 PWM0 General Configuration Register

### 13.2.1 PWM0 General Configuration Register

The user can set the PWM output mode of SC95F761XB 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 SC95F761XB 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 PDTxy [15: 0], the PWM output waveform switches between high and low levels, The PWM counter then continues counting upward until it matches the value of the period setting PWMPD[15:0] +1 (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.

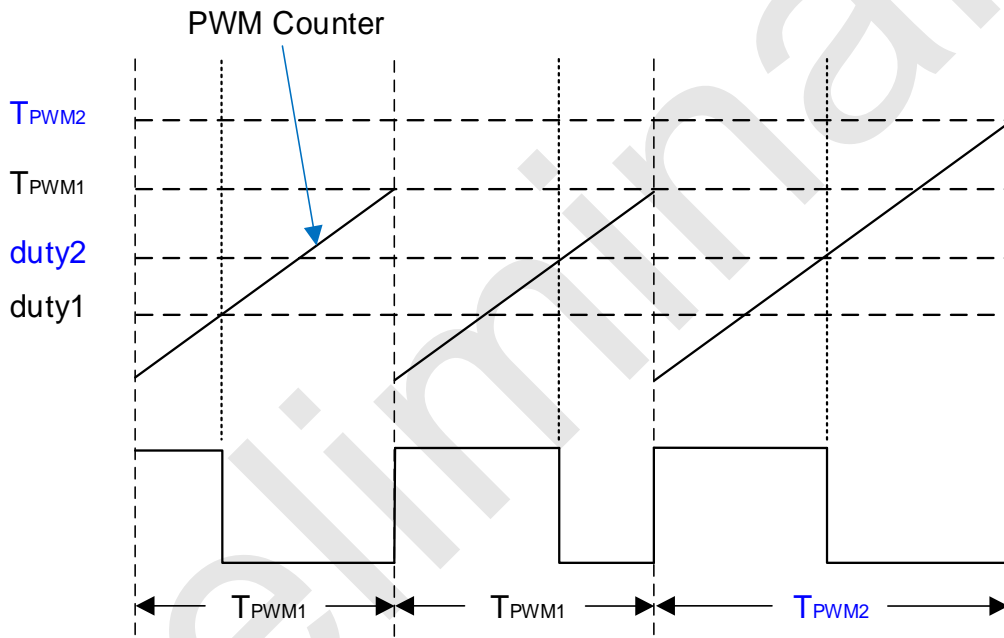
Calculation formula of edge-aligned period  $T_{PWM}$ :

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

Edge-aligned **duty** calculation formula:

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

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 PDTxy [15: 0], the PWM output waveform switches between high and low levels. Then the PWM counter continues to count up. When the count matches the value of PWMPD[15:0] +1 (that is, the midpoint of the PWM cycle), it automatically starts to count down. When the count value matches the value of PDTxy [15: 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.

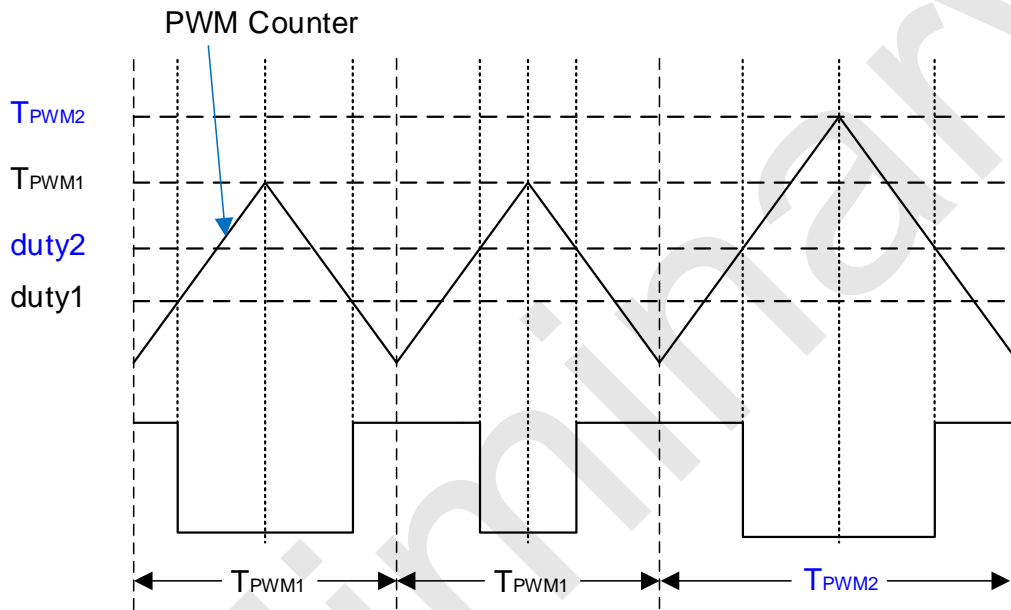
Calculation formula of center-aligned period  $T_{PWM}$ :

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

Center-aligned **duty** calculation formula:

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

The center aligned waveform is as follows:



Center-aligned PWM

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

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

| Bit number   | 7     | 6     | 5          | 4   | 3 | 2 | 1          | 0   |
|--------------|-------|-------|------------|-----|---|---|------------|-----|
| Bit Mnemonic | ENPWM | PWMIF | PWMCK[1:0] |     | - | - | PWMMD[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          | <b>ENPWM</b>       | <p>PWM module switch control (Enable PWM)</p> <p>1: Allow Clock to enter the PWM unit, the PWM is in the operating state, and the state of the PWM output port is controlled by the register ENPWMx (x=0~7)</p> <p>0: The PWM unit stops operating, the PWM counter is cleared, and all PWM output ports are set to the GPIO state</p>  |
| 6          | <b>PWMIF</b>       | <p>PWM interrupt request flag</p> <p>When the PWM counter overflows (that is, when the count exceeds PWMPD), this bit is automatically set to 1 by the hardware. If IE1[1] (EPWM) is also set to 1, the PWM interrupt is generated at this time. After the PWM interrupt occurs, the hardware will not automatically clear this bit. This bit must be cleared by the user's software.</p> |
| 5~4        | <b>PWMCK[1: 0]</b> | <p>PWM Clock Source Selector (PWM Clock Source Selector)</p> <p>00: <math>f_{HRC}</math></p> <p>01: <math>f_{HRC} / 2</math></p> <p>10: <math>f_{HRC} / 4</math></p> <p>11: <math>f_{HRC} / 8</math></p> <p>The frequency of the PWM clock source is fixed at <math>f_{HRC} = 32 \text{ MHz}</math></p>   |
| 1~0        | <b>PWMMD[1: 0]</b> | <p>PWM operating mode setting</p> <p>0x: Independent mode</p> <p>1x: complementary mode</p> <p>x0: edge alignment mode</p> <p>x1: center alignment mode</p>   |
| 3~2        | -                  | Reserved  |

**PWMCFG (D1H) PWM Set 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   |
|------------|---|---|
| 7~0        | <b>INV<math>x</math>(<math>x=0\sim7</math>)</b> | PWM $x$ waveform output reverse control<br>1: PWM $x$ waveform output is reversed<br>0: PWM $x$ waveform output is not reversed |

**PWMCON1 (D3H) PWM Control Register (read/write)**

| Bit number   | 7      | 6      | 5      | 4      | 3      | 2      | 1      | 0      |
|--------------|--------|--------|--------|--------|--------|--------|--------|--------|
| Bit Mnemonic | ENPWM7 | ENPWM6 | ENPWM5 | ENPWM4 | ENPWM3 | ENPWM2 | ENPWM1 | ENPWM0 |
| R/W          | R/W    | R/W    | R/W    | R/W    | R/W    | R/W    | R/W    | R/W    |
| POR          | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      |

| Bit number | Bit Mnemonic   | Description   |
|------------|--|---|
| 7~0        | <b>ENPWM<math>x</math></b><br><b>(<math>x=0\sim7</math>)</b> | PWM $x$ Waveform output select<br>0: PWM $x$ output is disabled<br>1: I/O where PWM $x$ resides serves as the output port of the PWM waveform |



**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) period register low 8 bits(R/W)**

| 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)PWMn period register high 8 bits (R/W)**

| 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        | <b>PWMPD[15:0]</b> | PWM cycle set<br>This value represents the output waveform of PWMn (cycle-1); that is to say PWMn cycle of output is(PWMPDX[15:0] + 1) * PW clock |

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

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

| Bit number | Bit Mnemonic | Description  |
|------------|--------------|--|
| 1          | <b>EPWM</b>  | PWM interrupt enable control<br><br>0: Disable PWM interrupt<br><br>1: Enable interrupt when PWM counter overflows |
| 4          | -            | Reserved   |

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

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

| Bit number | Bit Mnemonic | Description   |
|------------|--------------|---|
| 1          | <b>IPPWM</b> | PWM interrupt priority selection<br><br>0: Set the PWM interrupt priority to "low"<br><br>1: Set the PWM interrupt priority to "High" |
| 4          | -            | Reserved  |

**13.2.2 PWM0 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, and the PWM output stops. The PWM counter still keeps counting and the PWM interrupt is not affected. 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 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   |

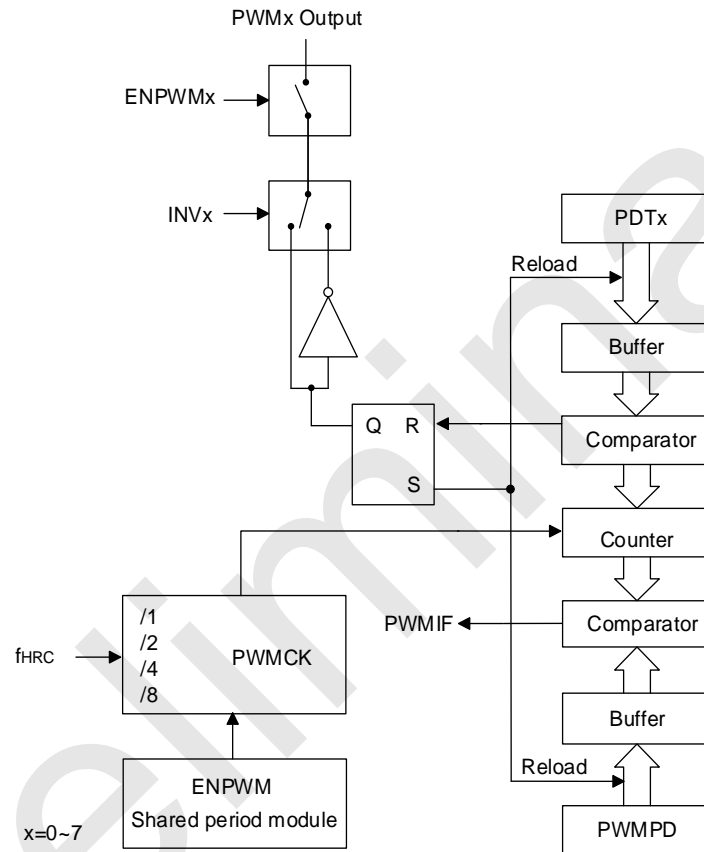
| Bit number | Bit Mnemonic   | Description  |
|------------|----------------|--|
| 7          | <b>FLTEN1</b>  | PWM fault detection function control bit<br>0: The fault detection function is turned off<br>1: The fault detection function is turned on  |
| 6          | <b>FLTSTA1</b> | PWM fault detection status flag<br>0: PWM is in normal output state;<br>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          | <b>FLTMD1</b>  | PWM fault detection mode setting bit<br>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<br>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 count to zero   |
| 4   | <b>FLTLV1</b>       | PWM fault detection level selection bit<br><br>0: Low level of fault detection is effective<br><br>1: High level of fault detection is effective  |
| 1~0 | <b>FLTDT1[1: 0]</b> | PWM fault detection input signal filtering time setting<br><br>00: filtering time is 0<br><br>01: filtering time is 1us<br><br>10: filter time is 4us<br><br>11: The filtering time is 16us |
| 3~2 | -                   | Reserved  |

### 13.3 PWM0 Independent Mode

In independent mode (PWMMD.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 PWM0 Independent Mode Block Diagram



SC95F761XB PWM Independent mode block diagram

### 13.3.2 PWM0 Independent Mode Duty Cycle Configuration

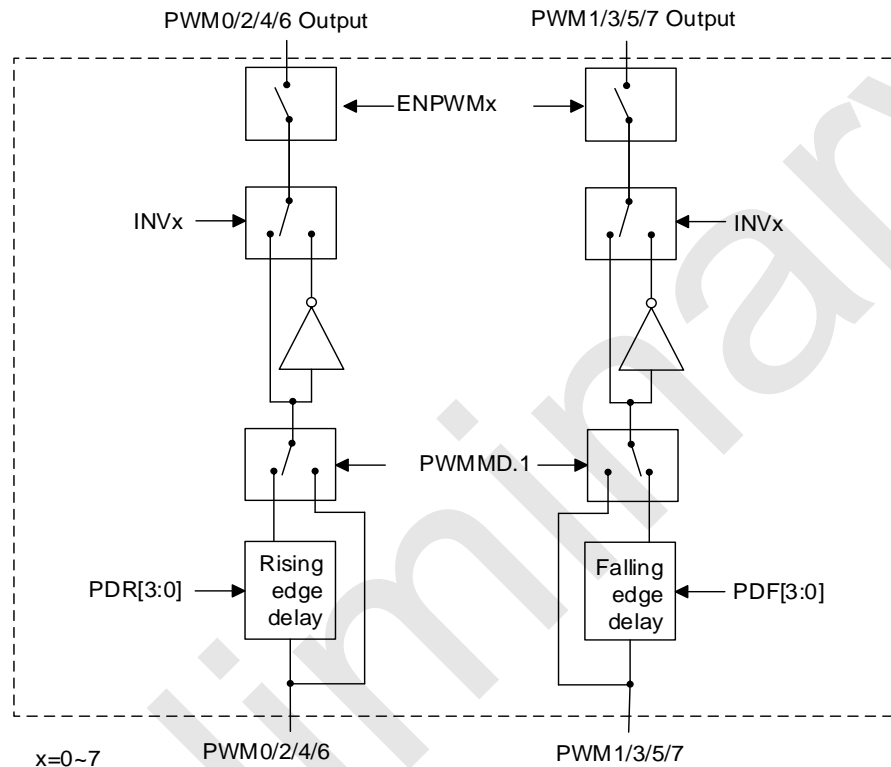
#### PWM0 Duty Cycle Adjustment Register PDT0x (Read/Write)

|       |             |
|-------|-------------|
| 2040H | PDT00[15:8] |
| 2041H | PDT00[7:0]  |
| 2042H | PDT01[15:8] |
| 2043H | PDT01[7:0]  |
| 2044H | PDT02[15:8] |
| 2045H | PDT02[7:0]  |
| 2046H | PDT03[15:8] |
| 2047H | PDT03[7:0]  |
| 2048H | PDT04[15:8] |
| 2049H | PDT04[7:0]  |
| 204AH | PDT05[15:8] |
| 204BH | PDT05[7:0]  |
| 204CH | PDT06[15:8] |
| 204DH | PDT06[7:0]  |
| 204EH | PDT07[15:8] |
| 204FH | PDT07[7:0]  |

| Bit number | Bit Mnemonic                           | Description  |
|------------|--|--|
| 7~0        | <b>PDT0x [15: 0]</b><br><b>(x=0~7)</b> | PWM0x waveform duty cycle length setting<br>The high-level width of the PWM0x waveform is (PDT0x [15: 0]) PWM clocks |

## 13.4 PWM0 Complementary Model

### 13.4.1 PWM0 Block Diagram of Complementary Mode



SC95F761XB PWM block diagram of complementary mode

### 13.4.2 PWM Complementary Mode Duty Cycle Configuration

In complementary mode ( $PWMMD[1:0] = 1x$ ), PWM00/PWM01, PWM02/PWM03, PWM04/PWM05 and PWM06/PWM07 are divided into four groups, respectively through  $PDT00[15:0]$ ,  $PDT02[15:0]$ ,  $PDT04[15:0]$  and  $PDT06[15:0]$  adjust the duty ratio;

**The registers  $PDT01[15:0]$ ,  $PDT03[15:0]$ ,  $PDT05[15:0]$  and  $PDT07[15:0]$  are invalid in the complementary mode.**

#### PWM0 Duty Cycle Adjustment Register $PDT0x$ (Read/Write)

|       |               |
|-------|---------------|
| 2040H | $PDT00[15:8]$ |
| 2041H | $PDT00[7:0]$  |

|       |             |
|-------|-------------|
| 2042H | PDT01[15:8] |
| 2043H | PDT01[7:0]  |
| 2044H | PDT02[15:8] |
| 2045H | PDT02[7:0]  |
| 2046H | PDT03[15:8] |
| 2047H | PDT03[7:0]  |
| 2048H | PDT04[15:8] |
| 2049H | PDT04[7:0]  |
| 204AH | PDT05[15:8] |
| 204BH | PDT05[7:0]  |
| 204CH | PDT06[15:8] |
| 204DH | PDT06[7:0]  |
| 204EH | PDT07[15:8] |
| 204FH | PDT07[7:0]  |

| Bit number | Bit Mnemonic                           | Description  |
|------------|--|--|
| 7~0        | <b>PDT0x [15: 0]</b><br><b>(x=0~7)</b> | PWM0x and PWM0y, y=x+1 port PWM waveform duty cycle length setting<br><br>The high-level width of the PWM waveform on the Px and Py pins is (PDT0x [15: 0]) PWM clocks |

**Note:**

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



### 13.4.3 PWM Complementary Mode Dead Time Setting

When the PWM0 of the SC95F761XB 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) PWM0 Dead Time Setting Register (read/write)

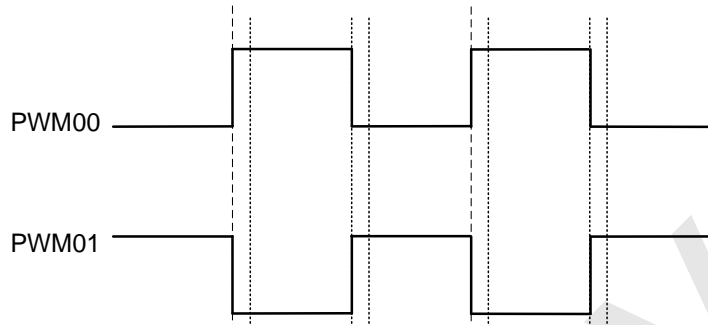
| Bit number   | 7         | 6   | 5   | 4   | 3         | 2   | 1   | 0   |
|--------------|-----------|-----|-----|-----|-----------|-----|-----|-----|
| Bit Mnemonic | PDF[3: 0] |     |     |     | PDR[3: 0] |     |     |     |
| R/W          | R/W       | R/W | R/W | R/W | R/W       | R/W | R/W |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

| Bit number | Bit Mnemonic     | Description  |
|------------|------------------|--|
| 7~4        | <b>PDF[3: 0]</b> | Complementary mode:<br>PWM falling edge dead time= $4 * PDF[3: 0] / f_{HRC}$ |
| 3~0        | <b>PDR[3: 0]</b> | Complementary mode:<br>PWM rising edge dead time= $4 * PDR[3: 0] / f_{HRC}$  |

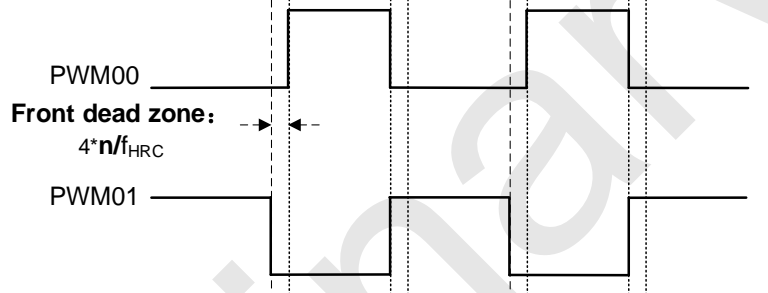
### 13.4.4 PWM0 Dead Zone Output Waveform

The following figure is based on the PWM00 and PWM01 in the complementary mode of the dead time adjustment waveform, in order to facilitate the distinction, PWM01 has reversed (INV1=1).

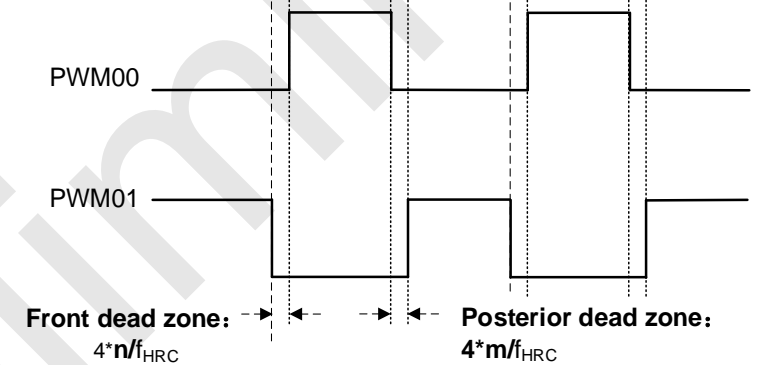
1. Dead zone not output:  
 PWMPD.1 = X  
 PDF = 0  
 PDR = 0



2. Setting PWM0 rising edge dead zone:  
 PWMPD.1 = 1  
 PDF = 0  
 PDR = n



3. Setting PWM1 falling edge dead zone:  
 PWMPD.1 = 1  
 PDF = m  
 PDR = n  
 Note: PWM1 has reversed at this time, then PDF corresponds to control the Rising edge dead zone delay time of PWM1 output waveform



PWM0 dead zone output waveform

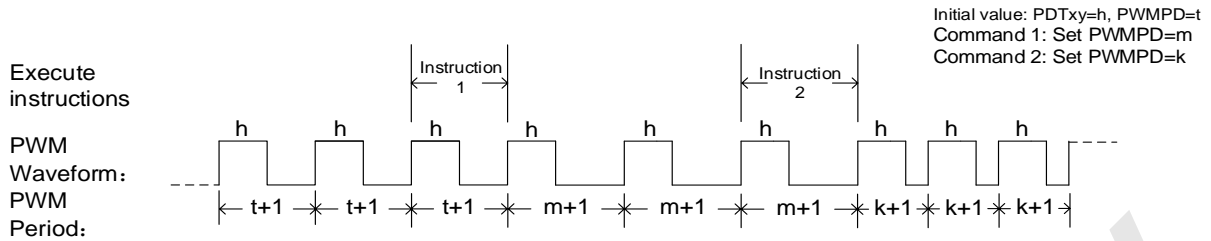
### 13.5 PWM0 Waveforms and Directions

The effect of changing SFR parameters on the PWM0 waveform is as follows:

① Duty cycle change characteristics

When the PWM0n 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 (PDT0x). But need to pay attention: change the value of PDT0x, the duty ratio will not change immediately, but wait for the PWM counter changing when it counts to 0 or up until it matches the value of the period setting item PWMPD[15:0] +1.

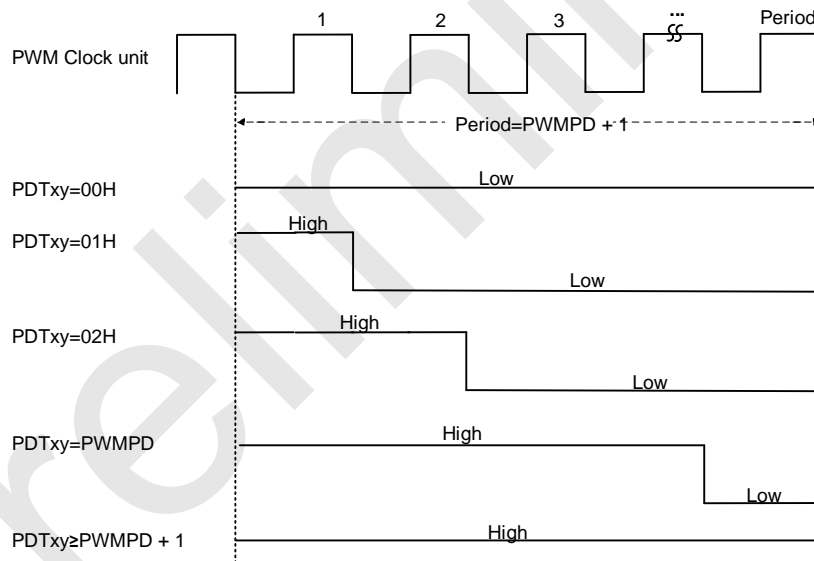
② Periodic change characteristics



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 PWM counter changing when it counts to 0 or up until it matches the value of the period setting item PWMPD[15:0] +1, 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, x=0~7) is initially 0. If you want to get the opposite result, you can set INVx to 1.

## 14 General-purpose I/O (GPIO)

The SC95F761XB provides up to 46 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 46 IOs are multiplexed with other functions. Among them, P3 can be set to output 1/4VDD or 1/3VDD voltage, which can be used as a COM driver for LCD display. 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.**

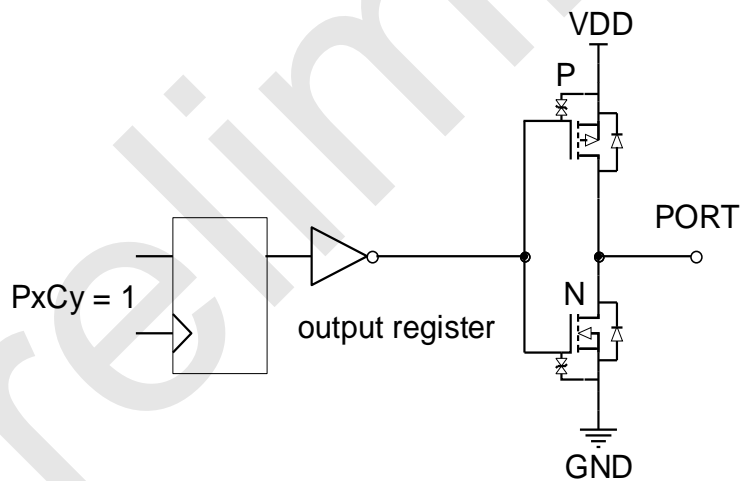
### 14.1 GPIO Structure Diagram

#### Strong Push-pull Output Mode

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

- Other than P04, P05, and P06, the I/O driver capability is an output greater than 10mA is high, and an output greater than 50mA is low.
- P04/P05/P06 drive can be achieved an output greater than 20mA is high, and an output greater than 50mA is low.

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

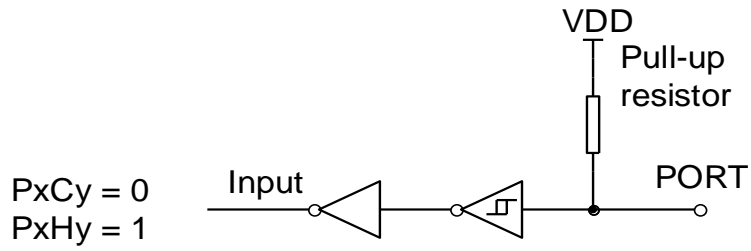


Strong push-pull output mode

#### 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.

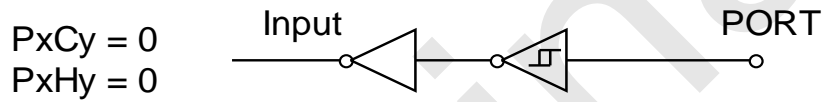
The schematic diagram of the port structure with pull-up input mode is as follows:



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)**

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

**P3CON (B1H) P3 Port Input/output Control Register (read/write)**

| Bit number   | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
|--------------|------|------|------|------|------|------|------|------|
| Bit Mnemonic | P3C7 | P3C6 | P3C5 | P3C4 | P3C3 | P3C2 | P3C1 | P3C0 |
| 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    |

**P3PH (B2H) P3 Port Pull-up Resistor Control Register (read/write)**

| Bit number   | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
|--------------|------|------|------|------|------|------|------|------|
| Bit Mnemonic | P3H7 | P3H6 | P3H5 | P3H4 | P3H3 | P3H2 | P3H1 | P3H0 |
| 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    |

**P4CON (C1H) P4 Port Input/output Control Register (read/write)**

| Bit number   | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
|--------------|------|------|------|------|------|------|------|------|
| Bit Mnemonic | P4C7 | P4C6 | P4C5 | P4C4 | P4C3 | P4C2 | P4C1 | P4C0 |
| 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    |

**P4PH (C2H) P4 Port Pull-up Resistor Control Register (read/write)**



| Bit number   | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
|--------------|------|------|------|------|------|------|------|------|
| Bit Mnemonic | P4H7 | P4H6 | P4H5 | P4H4 | P4H3 | P4H2 | P4H1 | P4H0 |
| 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 | - | - | P5C5 | P5C4 | P5C3 | P5C2 | P5C1 | P5C0 |
| R/W          | - | - | R/W  | R/W  | R/W  | R/W  | R/W  | R/W  |
| POR          | x | x | 0    | 0    | 0    | 0    | 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 | - | - | P5H5 | P5H4 | P5H3 | P5H2 | P5H1 | P5H0 |
| 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 |
|------------|--------------|-------------|
|------------|--------------|-------------|

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

**P3 (B0H) P3 Port Data Register (read/write)**

| Bit number   | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
|--------------|------|------|------|------|------|------|------|------|
| Bit Mnemonic | P3.7 | P3.6 | P3.5 | P3.4 | P3.3 | P3.2 | P3.1 | P3.0 |
| 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    |

**P4 (C0H) P4 Port Data Register (read/write)**

| Bit number   | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
|--------------|------|------|------|------|------|------|------|------|
| Bit Mnemonic | P4.7 | P4.6 | P4.5 | P4.4 | P4.3 | P4.2 | P4.1 | P4.0 |
| 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 | - | - | P5.5 | P5.4 | P5.3 | P5.2 | P5.1 | P5.0 |
| 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                 |
|------------|------------------------|-----------------------------|
| 7~0        | <b>P0.x</b><br>(x=0~7) | P0 port latch register data |
| 7~0        | <b>P1.x</b><br>(x=0~7) | P1 port latch register data |
| 7~0        | <b>P2.x</b><br>(x=0~7) | P2 port latch register data |
| 7~0        | <b>P3.x</b><br>(x=0~7) | P3 port latch register data |
| 7~0        | <b>P4.x</b><br>(x=0~7) | P4 port latch register data |
| 5~0        | <b>P5.x</b><br>(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        | <b>P1H[1: 0]</b> | P1 high four IOH settings<br>00: Set P1 high four IOH level 0 (maximum);<br>01: Set P1 high four IOH level 1;<br>10: Set P1 high four IOH level 2;<br>11: Set P1 high four IOH level 3 (minimum); |
| 5~4        | <b>P1L[1: 0]</b> | P1 low four IOH settings<br>00: Set P1 low four IOH level 0 (maximum);<br>01: Set P1 low four IOH level 1;<br>10: Set P1 low four IOH level 2;<br>11: Set P1 low four IOH level 3 (minimum);      |
| 3~2        | <b>P0H[1: 0]</b> | P0 high four IOH settings<br>00: Set P0 high four IOH level 0 (maximum);<br>01: Set P0 high four IOH level 1;<br>10: Set P0 high four IOH level 2;<br>11: Set P0 high four IOH level 3 (minimum)  |
| 1~0        | <b>P0L[1: 0]</b> | P0 low four IOH settings  |

|  |  |  |
|--|--|--|
|  |  | 00: Set P0 low four IOH level 0 (maximum);<br>01: Set P0 low four IOH level 1;<br>10: Set P0 low four IOH level 2;<br>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 | - | - | P3L[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        | <b>P3L[1: 0]</b> | P3 low four IOH settings<br>00: set P3 low four IOH level 0 (maximum);<br>01: Set P3 low four IOH level 1;<br>10: Set P3 low four IOH level 2;<br>11: Set P3 low four IOH level 3 (minimum);      |
| 3~2        | <b>P2H[1: 0]</b> | P2 high four IOH settings<br>00: Set P2 high four IOH level 0 (maximum);<br>01: Set P2 high four IOH level 1;<br>10: Set P2 high four IOH level 2;<br>11: Set P2 high four IOH level 3 (minimum); |

|     |                  |  |
|-----|------------------|--|
| 1~0 | <b>P2L[1: 0]</b> | P2 low four IOH settings<br>00: Set P2 low four IOH level 0 (maximum);<br>01: Set P2 low four IOH level 1;<br>10: Set P2 low four IOH level 2;<br>11: Set P2 low four IOH level 3 (minimum); |
| 7~6 | -                | Reserved   |

## 15 LCD/LED Display Driver

The SC95F761XB integrates hardware LCD/LED display drive circuit inside, which can facilitate users to realize LCD and LED display drive. Its main features are as follows:

1. Choose one of LCD and LED display driver;
2. LCD and LED display drivers share related IO ports and registers.

**The LCD display driver functions are as follows:**

1. 4 display drive modes are available: 8 X 24, 6 X 26, 5 X 27, or 4X 28 segments;
2. 2 kinds of offset methods are available: 1/4 Bias and 1/3 Bias;
3. 4 levels of COM port drive capability are optional;
4. The display drive circuit can choose the built-in 32 kHz LRC or external 32.768 kHz oscillator as the clock source, the frame frequency is about 64Hz.

**Note: If the LCD driver frame frequency is not 64Hz, you are advised to contact sinone technical support for solutions.**

**The LED display driver functions are as follows:**

1. 4 display drive modes are available: 8 X 24, 6 X 26, 5 X 27, or 4X 28 segments;
2. SEG port drive capability is optional in 4 levels;
3. The display drive circuit can select the built-in 32 kHz LRC or external 32.768kHz oscillator as the clock source, and the frame frequency is about 64Hz.

### 15.1 LCD/LED Display Drive-related Registers

**DDRCON (93H) Display Drive Control Register (read/write)**

| Bit number   | 7     | 6    | 5          | 4   | 3          | 2   | 1   | 0   |
|--------------|-------|------|------------|-----|------------|-----|-----|-----|
| Bit Mnemonic | DDRON | DMOD | DUTY[1: 0] |     | VLCD[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          | <b>DDRON</b>      | <b>LCD/LED display drive enable control</b><br>0: The display driver scan is turned off<br>1: The display driver scan is turned on   |
| 6          | <b>DMOD</b>       | <b>LCD/LED display drive mode</b><br>0: LCD mode;<br>1: LED mode   |
| 5~4        | <b>DUTY[1: 0]</b> | <b>LCD/LED display duty control</b><br>00: 1/8 duty cycle, S4~S27 are segments, C0~C7 are common;<br>01: 1/6 duty cycle, S2~S27 are segments, C2~C7 are common;<br>10: 1/5 duty cycle, S1~S27 are segments, C3~C7 are common;<br>11: 1/4 duty cycle, S0~S27 is segment, C4~C7 is common, or S4~S27 is segment, C0~C3 is common |
| 3~0        | <b>VLCD[3: 0]</b> | <b>LCD voltage regulation</b><br>$VLCD = V_{DD} * (17 + VLCD[3: 0]) / 32$  |

**P0VO (9CH) P0 Port Display Driver Output Register (read/write)**

| Bit number   | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|--------------|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit Mnemonic | P07VO | P06VO | P05VO | P04VO | P03VO | P02VO | P01VO | P00VO |
| 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        | <b>P0nVO</b> | <b>Open P0n port display driver output</b><br>0: Disable the display driver output function of P0n port<br>1: Enable the display driver output function of P0n port |

**P1VO (94H) P1 Port Display Driver Output Register (read/write)**

| Bit number   | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|--------------|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit Mnemonic | P17VO | P16VO | P15VO | P14VO | P13VO | P12VO | P11VO | P10VO |
| 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        | <b>P1nVO</b> | <b>Open P1n port display driver output</b><br>0: Disable the display driver output function of P1n port<br>1: Enable the display driver output function of P1n port |

**P2VO (A3H) P2 Port Display Driver Output Register (read/write)**

| Bit number   | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|--------------|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit Mnemonic | P27VO | P26VO | P25VO | P24VO | P23VO | P22VO | P21VO | P20VO |
| 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        | <b>P2nVO</b> | <b>Open P2n port display driver output</b><br>0: Disable the display driver output function of P2n port<br>1: Enable the display driver output function of P2n port |

**P3VO (B3H) P3 Port Display Driver Output Register (read/write)**

| Bit number   | 7     | 6     | 5     | 4     | 3     | 2     | 1     | 0     |
|--------------|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit Mnemonic | P37VO | P36VO | P35VO | P34VO | P33VO | P32VO | P31VO | P30VO |
| 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        | <b>P3nVO</b> | <b>Open P3n port display driver output</b><br>0: Disable the display driver output function of P3n port<br>1: Enable the display driver output function of the P3n port |

**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   |
|------------|--------------------|---|
| 3~2        | <b>VOIRS[1: 0]</b> | <p><b>LCD voltage output port voltage divider resistance selection</b></p> <p>00: Set the total resistance of the internal voltage divider to be 100kΩ</p> <p>01: Set the total resistance of the internal voltage divider resistor to 200kΩ</p> <p>10: Set the total resistance of the internal voltage divider to 400kΩ</p> <p>11: Set the total resistance of the internal voltage divider to 800kΩ</p> <p>Each time Common is switched, the first 1/16 time is fixed to select a 100k resistor, and the last 15/16 time is switched to the resistance value selected by VORIS</p> |
| 1          | <b>SCS</b>         | <p><b>LCD/LED Segment/Common multiplex pin selection</b></p> <p>0: When set to 1/4 duty cycle, S0~S27 are segments and C4~C7 are common</p> <p>1: When set to 1/4 duty cycle, S4~S27 are segments and C0~C3 are common</p>  |
| 0          | <b>BIAS</b>        | <p><b>LCD display drive bias voltage setting:</b></p> <p>0: 1/4 bias voltage;</p> <p>1: 1/3 bias voltage</p>  |

## 15.2 LCD/LED Display RAM Configuration

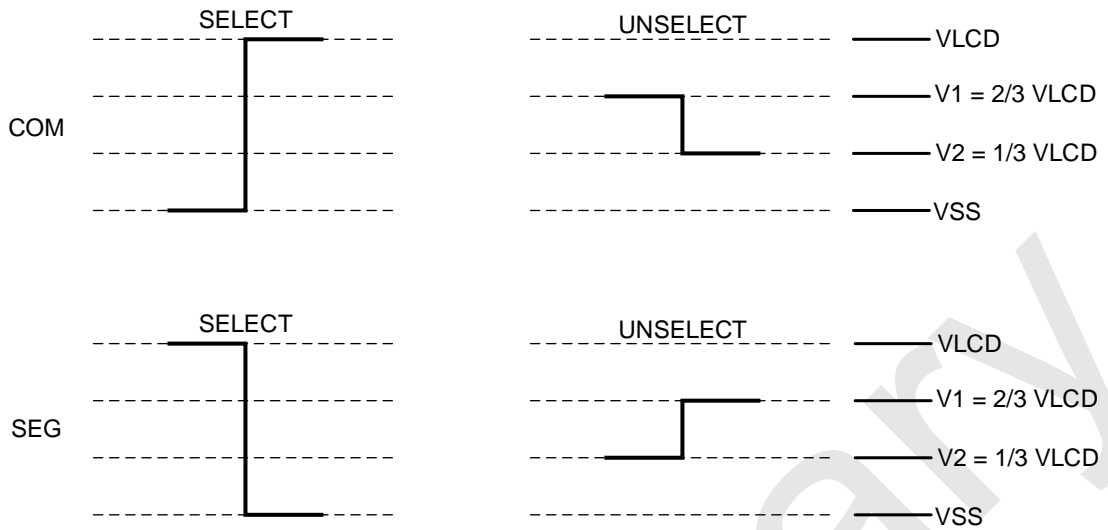
|         |   |   |   |   |   |   |   |   |
|---------|---|---|---|---|---|---|---|---|
| Address | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|---|---|---|---|---|---|---|---|

|       | COM7  | COM6  | COM5  | COM4  | COM3  | COM2  | COM1  | COM0  |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 2000H | SEG0  | SEG0  | SEG0  | SEG0  | SEG0  | SEG0  | SEG0  | SEG0  |
| 2001H | SEG1  | SEG1  | SEG1  | SEG1  | SEG1  | SEG1  | SEG1  | SEG1  |
| 2002H | SEG2  | SEG2  | SEG2  | SEG2  | SEG2  | SEG2  | SEG2  | SEG2  |
| 2003H | SEG3  | SEG3  | SEG3  | SEG3  | SEG3  | SEG3  | SEG3  | SEG3  |
| 2004H | SEG4  | SEG4  | SEG4  | SEG4  | SEG4  | SEG4  | SEG4  | SEG4  |
| 2005H | SEG5  | SEG5  | SEG5  | SEG5  | SEG5  | SEG5  | SEG5  | SEG5  |
| 2006H | SEG6  | SEG6  | SEG6  | SEG6  | SEG6  | SEG6  | SEG6  | SEG6  |
| 2007H | SEG7  | SEG7  | SEG7  | SEG7  | SEG7  | SEG7  | SEG7  | SEG7  |
| 2008H | SEG8  | SEG8  | SEG8  | SEG8  | SEG8  | SEG8  | SEG8  | SEG8  |
| 2009H | SEG9  | SEG9  | SEG9  | SEG9  | SEG9  | SEG9  | SEG9  | SEG9  |
| 200AH | SEG10 | SEG10 | SEG10 | SEG10 | SEG10 | SEG10 | SEG10 | SEG10 |
| 200BH | SEG11 | SEG11 | SEG11 | SEG11 | SEG11 | SEG11 | SEG11 | SEG11 |
| 200CH | SEG12 | SEG12 | SEG12 | SEG12 | SEG12 | SEG12 | SEG12 | SEG12 |
| 200DH | SEG13 | SEG13 | SEG13 | SEG13 | SEG13 | SEG13 | SEG13 | SEG13 |
| 200EH | SEG14 | SEG14 | SEG14 | SEG14 | SEG14 | SEG14 | SEG14 | SEG14 |
| 200FH | SEG15 | SEG15 | SEG15 | SEG15 | SEG15 | SEG15 | SEG15 | SEG15 |
| 2010H | SEG16 | SEG16 | SEG16 | SEG16 | SEG16 | SEG16 | SEG16 | SEG16 |

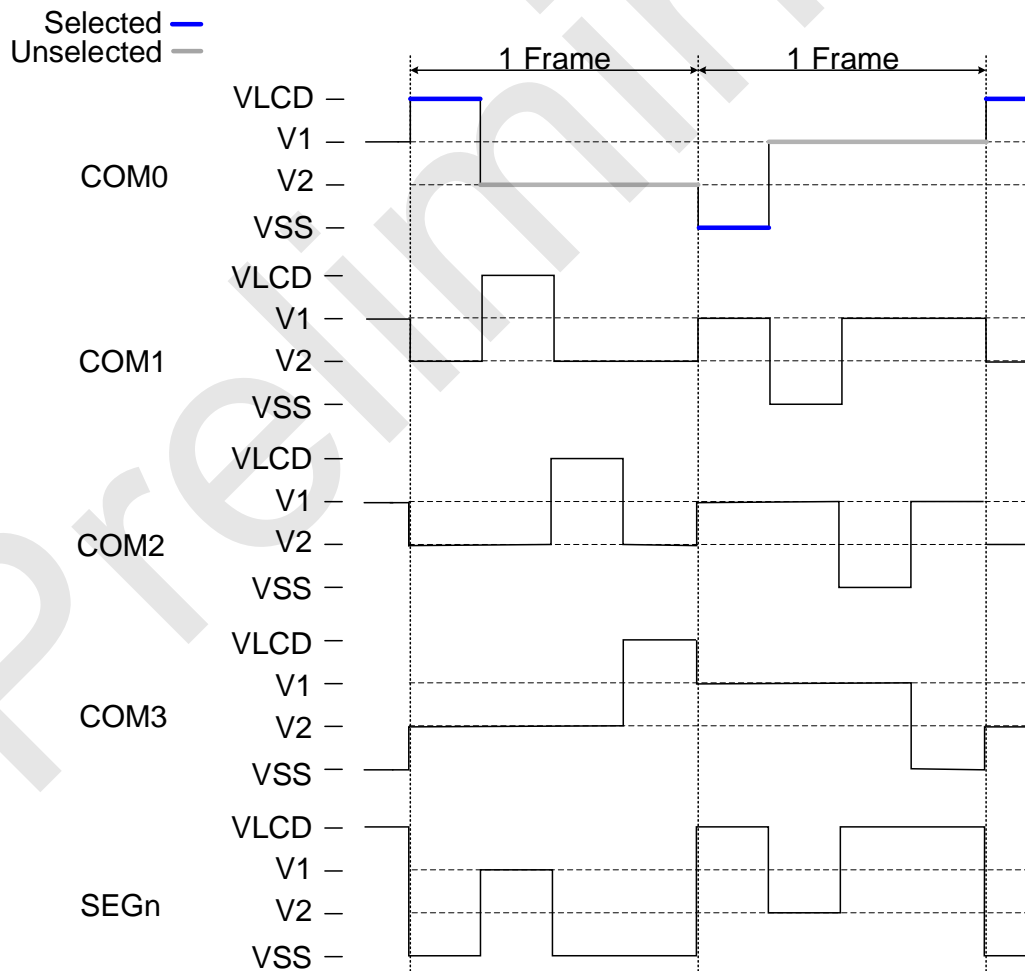
|       |       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 2011H | SEG17 | SEG17 | SEG17 | SEG17 | SEG17 | SEG17 | SEG17 | SEG17 |
| 2012H | SEG18 | SEG18 | SEG18 | SEG18 | SEG18 | SEG18 | SEG18 | SEG18 |
| 2013H | SEG19 | SEG19 | SEG19 | SEG19 | SEG19 | SEG19 | SEG19 | SEG19 |
| 2014H | SEG20 | SEG20 | SEG20 | SEG20 | SEG20 | SEG20 | SEG20 | SEG20 |
| 2015H | SEG21 | SEG21 | SEG21 | SEG21 | SEG21 | SEG21 | SEG21 | SEG21 |
| 2016H | SEG22 | SEG22 | SEG22 | SEG22 | SEG22 | SEG22 | SEG22 | SEG22 |
| 2017H | SEG23 | SEG23 | SEG23 | SEG23 | SEG23 | SEG23 | SEG23 | SEG23 |
| 2018H | SEG24 | SEG24 | SEG24 | SEG24 | SEG24 | SEG24 | SEG24 | SEG24 |
| 2019H | SEG25 | SEG25 | SEG25 | SEG25 | SEG25 | SEG25 | SEG25 | SEG25 |
| 201AH | SEG26 | SEG26 | SEG26 | SEG26 | SEG26 | SEG26 | SEG26 | SEG26 |
| 201BH | SEG27 | SEG27 | SEG27 | SEG27 | SEG27 | SEG27 | SEG27 | SEG27 |

## 15.3 LCD Waveform

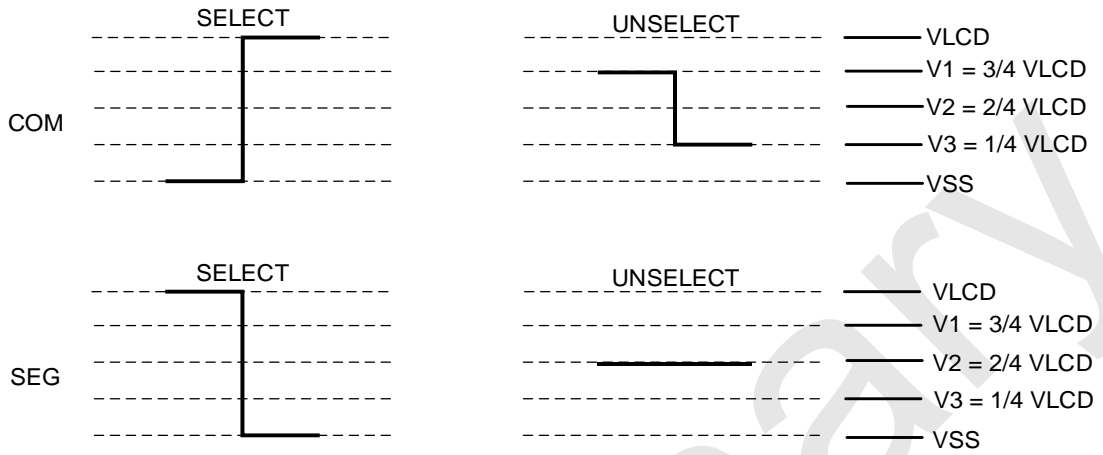
### 15.3.1 1/3Bias LCD Waveform



1/3 Bias LCD Gated and non-gated voltages

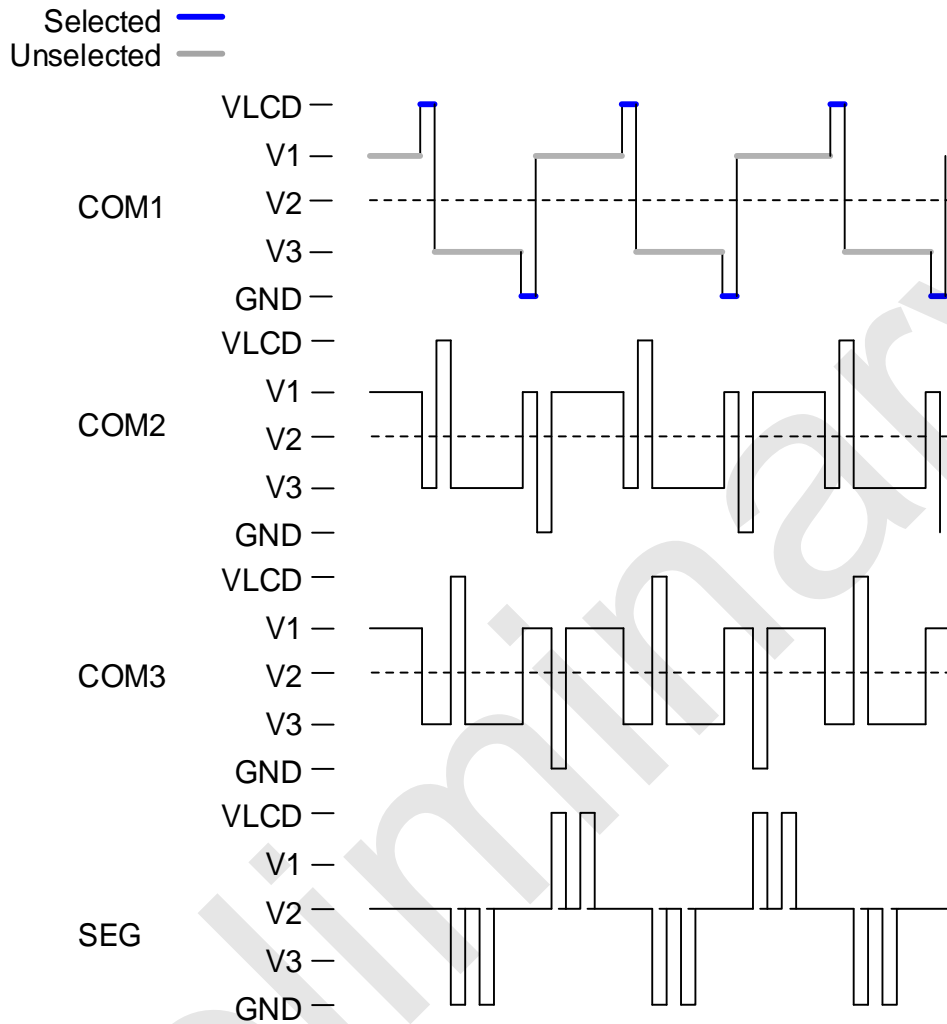


1/3 Bias Waveforms of COM and SEG in LCD applications

**15.3.2 1/4Bias LCD Waveform**


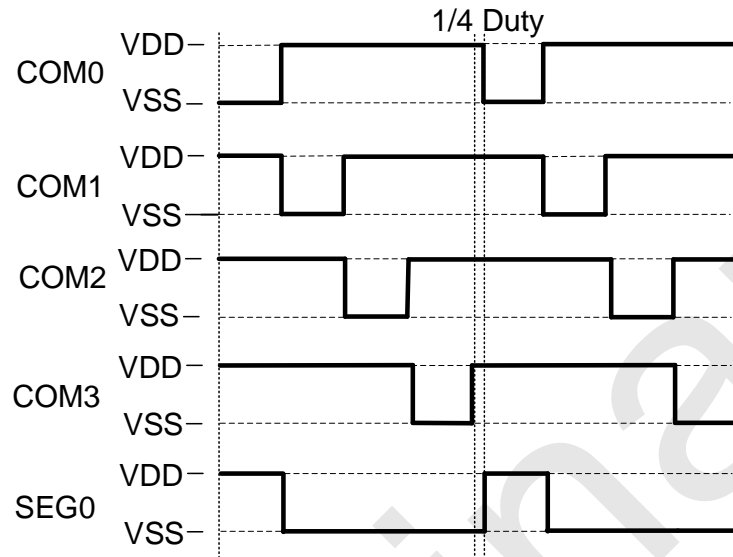
1/4 Bias LCD Gated and non-gated voltages





1/4 Bias Waveforms of COM and SEG in LCD applications

## 15.4 LED Waveform



Waveforms of COM and SEG in LED applications

## 15.5 LCD/LED Example

### 15.5.1 LCD Configuration Demo Program

```
unsigned char xdata LCDRAM[30] _at_ 0x2000;
```

```
unsigned char lcd_addr;
```

```
unsigned char lcd_data;
```

```
DDRCN |= 0x00; //0: LCD mode 1: LED mode
```

```
DDRCN |= 0x30; //1/4 Duty cycle
```

```
DDRCN |= 0x07; // VLCD=VDD*3/4
```

```
DDRCN |= 0x80; // Display driver scan on
```

```
P0VO = 0xFF; // Open the display driver output function of P0 port
```

```
P1VO = 0xFF; // Open the display driver output function of P1 port
```

```
P2VO = 0xFF; // Open the display driver output function of P2 port
```

```
P3VO = 0xFF;          // Open the display driver output function of P3 port

OTCON = 0x06; // Set the total resistance of the internal voltage divider resistor to 200KΩ

                //1/4 Bias voltage; S4~S27 is segment,C0~C3 is common

LCDRAM[lcd_addr] = lcd_data; //Write the value to be displayed to the LCD RAM
```

### 15.5.2 LED Configuration Demo Program

```
unsigned char xdata LEDRAM[30] _at_ 0x2000;

unsigned char led_addr;

unsigned char led_data;

DDRCON |= 0x4F;      //0: LCD mode1 : LED mode

                    //LED mode; 1/8 Duty cycle

                    //S4~S27 is segment,C0~C7 is common;

DDRCON |= 0x80;      // Display driver scan on

IOHCON0 = 0xC0;      // Set P1 high four IOH level 3 (minimum), other pins IOH level 0

                    (maximum)

IOHCON1 = 0x00;

P0VO = 0xFF;         // Open the display driver output function of P0 port

P1VO = 0xFF;         // Open the display driver output function of P1 port

P2VO = 0xFF;         // Open the display driver output function of P2 port

P3VO = 0xFF;         // Open the display driver output function of P3 port

OTCON = 0x00;

LCDRAM[led_addr] = led_data; //Write the value to be displayed to the LED RAM
```

## 16 SERIAL INTERFACE (UART0)

The SC95F761XB 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        | <b>SM0~1</b> | Serial communication mode control bit<br>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;<br>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;<br>10: reserved;<br>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          | <b>SM2</b>   | 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><b>The baud rate override setting bit is only valid in mode 0 (SM0~1 = 00):</b></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 | <b>REN</b> | <p>Receive enable control bit</p> <p>0: data reception is not allowed;</p> <p>1: Allow receiving data.</p>   |
| 3 | <b>TB8</b> | Only valid for mode 3, which is the 9th bit of the transmitted data  |
| 2 | <b>RB8</b> | Only valid for mode 3, the 9th bit of the received data  |
| 1 | <b>TI</b>  | Transmit interrupt flag  |
| 0 | <b>RI</b>  | 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 | <b>SBUF[7: 0]</b> | <b>Serial data buffer register</b><br><br>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. |
|-----|-------------------|--|

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

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

| Bit number | Bit Mnemonic | Description   |
|------------|--------------|---|
| 7          | <b>SMOD</b>  | When SM0~1 = 01 (UART0 mode 1) or SM0~1 = 11 (UART0 mode 3), the baud rate is set to bit:<br>0: the serial port runs at $f_{sys}/1$<br>1: indicates that the serial port runs at $f_{sys}/16$ |

## 16.1 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 modes 1 and 3, the serial port clock source can be programmed as either 1 or 16 of the system clock, determined by SMOD(pcon.7) bits. When SMOD is 0, the serial port runs at  $1/f_{sys}$ . When SMOD is 1, the serial port runs at  $16/f_{sys}$ . After the serial port clock source is determined, set the baud rate overflow rate by Timer 1 or Timer 2: When TCLK(TXCON. 4) and RCLK(TXCON. 5) bits are both 0, then timer 1 is baud rate generator mode, and the baud rate overflow rate of UART0 is set by [TH1,TL1]. The formula is as follows, note: When timer 1 acts as a baud rate generator, timer 1 must stop counting, i.e. TR1=0:

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

- SMOD = 1:  $\text{BaudRate} = \frac{1}{16} * \frac{f_{\text{sys}}}{[\text{TH1}, \text{TL1}]}$ ;

- When TCLK(TXCON. 4) or RCLK(TXCON. 5) is 1, then timer 2 is in baud rate generator mode, and the baud rate overflow rate of UART0 is set by [RCAP2H, RCAP2L], the formula is as follows:

- SMOD = 0:  $\text{BaudRate} = \frac{f_{\text{sys}}}{[\text{RCAP2H}, \text{RCAP2L}]}$ ; Note: [RCAP2H, RCAP2L] must be bigger than 0x0010)

- SMOD = 1:  $\text{BaudRate} = \frac{1}{16} * \frac{f_{\text{sys}}}{[\text{RCAP2H}, \text{RCAP2L}]}$ ;

## 17 SPI/TWI/UART Serial Interface (USCI)

| Symbol  | Address | Description              | 7             | 6 | 5 | 4 | 3 | 2 | 1 | 0 | POR           |
|---------|---------|--------------------------|---------------|---|---|---|---|---|---|---|---------------|
| US0CON0 | 95H     | USCI0 control register 0 | US0CON0[7: 0] |   |   |   |   |   |   |   | 0000000<br>0b |
| US0CON1 | 9DH     | USCI0 control register 1 | US0CON1[7: 0] |   |   |   |   |   |   |   | 0000000<br>0b |
| US0CON2 | 9EH     | USCI0 control register 2 | US0CON2[7: 0] |   |   |   |   |   |   |   | 0000000<br>0b |
| US0CON3 | 9FH     | USCI0 control register 3 | US0CON3[7: 0] |   |   |   |   |   |   |   | 0000000<br>0b |
| US1CON0 | A4H     | USCI1 control register 0 | US1CON0[7: 0] |   |   |   |   |   |   |   | 0000000<br>0b |
| US1CON1 | A5H     | USCI1 control register 1 | US1CON1[7: 0] |   |   |   |   |   |   |   | 0000000<br>0b |
| US1CON2 | A6H     | USCI1 control register 2 | US1CON2[7: 0] |   |   |   |   |   |   |   | 0000000<br>0b |
| US1CON3 | A7H     | USCI1 control register 3 | US1CON3[7: 0] |   |   |   |   |   |   |   | 0000000<br>0b |
| USXCON0 | C4H     | USCI2 control register 0 | US2CON0[7: 0] |   |   |   |   |   |   |   | 0000000<br>0b |
| USXCON1 | C5H     | USCI2 control register 1 | US2CON1[7: 0] |   |   |   |   |   |   |   | 0000000<br>0b |
| USXCON2 | C6H     | USCI2 control register 2 | US2CON2[7: 0] |   |   |   |   |   |   |   | 0000000<br>0b |
| USXCON3 | C7H     | USCI2 control register 3 | US2CON3[7: 0] |   |   |   |   |   |   |   | 0000000<br>0b |



|        |     |                                      |            |            |   |            |   |            |      |           |           |
|--------|-----|--------------------------------------|------------|------------|---|------------|---|------------|------|-----------|-----------|
| USXINX | DCH | USCI2/3/4/5 Control register pointer | -          | -          | - | -          | - | USCIX[2:0] |      | xxxxx000b |           |
| TMCON  | 8EH | Timer frequency control register     | USMDX[1:0] |            | - | -          | - | -          | T1FD | T0FD      | 00xxxx00b |
| OTCON  | 8FH | Output control register              | USMD1[1:0] | USMD0[1:0] |   | VOIRS[1:0] |   | SCS        | BIAS | 00000000b |           |

The SC95F761XB internally integrates 6 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 USCIO~5 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 USMDX[1: 0] bits of TMCON. Its characteristics are as follows:

1. When USCIO is set to SPI0, the output drive capability is enhanced, which is consistent with common I/O features in other modes
2. SPI mode can be configured as one of master mode or slave mode, with 8-bit or 16-bit transmission mode
3. TWI mode communication can be configured as master mode or slave mode
4. UART modes can work in: Mode 0 (8-bit half-duplex synchronous communication), Mode 1 (10-bit full-duplex asynchronous communication) and Mode 3 (11-bit full-duplex asynchronous communication)
5. USCIO/1/2/3/4 are independent USCI interfaces. Note the following: The USCIX[2:0] control registers share the same set of addresses (C4H-C7H). Users can point the USCIX register group (USXCON0~3) to USCIO/1/2/3/4 through USXINX[2:0], thus realizing the function of configuring three independent USCI interfaces for one set of registers

**Note: Only after USXINX[2:0] has been successfully configured will the USCIX register group point to USCIO/1/2/3/4 specified by the user, in which case set the USCIX register group is valid for the corresponding USCI interface.**

The specific configuration method is as follows:

#### USXINX (DCH) USCIO/1/2/3/4 Control register pointer (R/W)

| Bit number   | 7 | 6 | 5 | 4 | 3 | 2          | 1   | 0   |
|--------------|---|---|---|---|---|------------|-----|-----|
| Bit Mnemonic | - | - | - | - | - | USCIX[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 | <b>USCIX[2:0]</b> | USCIX2/3/4/5 Control register pointer<br>010: USCIX register (USXCON0~3 及 USMDX) point to USCIX2<br>011: USCIX register point to USCIX3<br>100: USCIX register point to USCIX4<br>101: USCIX register point to USCIX5<br>Other:Reserved |
| 5~3 | -                 | Reserved  |

**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        | <b>USMD1[1: 0]</b> | <b>USCIX1 Communication mode control bit</b><br>00: USCIX1 close<br>01: USCIX1 Set to SPI communication mode;<br>10: USCIX1 Set to TWI communication mode;<br>11: USCIX1 Set to UART communication mode; |
| 5~4        | <b>USMD0[1: 0]</b> | <b>USCIX0 Communication mode control bit</b><br>00: Reserved;<br>01: USCIX0 Set to SPI communication mode;<br>10: USCIX0 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 | USMDX[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 | Condition          | Description   |
|------------|--------------|--------------------|---|
| 7~6        | USMDX[1: 0]  | <b>USCIX = 010</b> | <b>USC12 Communication mode control bit</b><br>00: USC12 close<br>01: USC12 set to SPI communication mode;<br>10: USC12 set to TWI communication mode;<br>11: USC12 set to UART communication mode; |
|            |              | <b>USCIX = 011</b> | <b>USC13 Communication mode control bit</b><br>00: USC13 close<br>01: USC13 set to SPI communication mode;<br>10: USC13 set to TWI communication mode;<br>11: USC13 set to UART communication mode; |
|            |              | <b>USCIX = 100</b> | <b>USC14 Communication mode control bit</b><br>00: USC14 close<br>01: USC14 set to SPI communication mode;  |

|  |  |                    |   |
|--|--|--------------------|---|
|  |  |                    | 10: USCI4 set to TWI communication mode;<br>11: USCI4 set to UART communication mode;   |
|  |  | <b>USCIX = 101</b> | <b>USCI5 Communication mode control bit</b><br>00: USCI5 close<br>01: USCI5 set to SPI communication mode;<br>10: USCI5 set to TWI communication mode;<br>11: USCI5 set to UART communication mode; |

Note:

A USCI interface can be set to different communication modes by USMD, each of which has a corresponding operation register group. 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.

For example:

- Set USMD1 = 01, USCI0 is SPI communication interface, in this mode set US0CON0 (95H) = 0x80H;
- Then set USMD1 = 11, USCI0 is the UART communication interface, set US0CON0 (95H) = 0x0FH;
- Then set USMD1 = 01, USCI0 is set back to SPI communication interface, read US0CON0 (95H) in this mode, should be 0x80H;
- Then set USMD1 = 11, USCI0 is set back to the UART communication interface, and US0CON0 (95H) is read in this mode, which should be 0x0FH.

## 17.1 SPI

USMDn[1: 0] = 01, n=0~5, select one of three serial interface USCI is configured as SPI interface:

- USTXn as MOSI signal
- USRXn as MISO signal
- USCKn as CLK signal

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 SPI interface of USCI0 has 16-bit 8-level FIFO cache and is independent of sending and receiving, that is, users can achieve:

- Write up to 8 bits or less 16-bit data sequentially to SPI send caches (US0CON2, US0CON3). When the SPI sends data, the first data written is also sent first. When the data written into the FIFO by the user is sent, the null flag TXE of the sending buffer is set to 1. If the DATA in THE FIFO is full, the write conflict flag bit WCOL is set, and the user cannot write data to the FIFO until the data in the FIFO is sent out and the FIFO is not satisfied. The interrupt flag SPIF is raised when all data in the FIFO is sent.

- The SPI receive cache (US0CON2, US0CON3) continuously reads 8 or less 16-bit received data, and the first received data is also read first.
- In addition, when USC10 is set as SPI0, the driving ability of the pins corresponding to its signal port will be enhanced.

| Contrast<br>BIT | USC11~5 SPI  | USC10 SPI   |
|-----------------|--|---|
| TXE             | The send buffer is empty, set 1                                    | Set to 1 when data written to the FIFO is sent                      |
| WCOL            | When a frame is being sent, rewrites are set to 1 and cannot write | If the FIFO is filled, set it to 1, and can't write the FIFO        |
| SPIF            | Send complete, interrupt flag set up                               | The interrupt flag is not raised until all data in the FIFO is sent |

### 17.1.1 SPI Operation Related Registers

**US0CON0 (95H) SPI0 control register (read/write)**

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

**USXCON0 (C4H) SPI2~5 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          | <b>SPEN</b>  | <b>SPI Enable control</b><br>0: close SPI |

|     |                  |  |
|-----|------------------|--|
|     |                  | 1: open SPI  |
| 5   | <b>MSTR</b>      | <b>SPI master-slave selection</b><br>0: SPI is slave device<br>1: SPI is master device   |
| 4   | <b>CPOL</b>      | <b>Clock polarity control bit</b><br>0: SCK is low in idle state<br>1: SCK is high in idle state   |
| 3   | <b>CPHA</b>      | <b>Clock phase control bit</b><br>0: Collect data on the first edge of the SCK cycle<br>1: Collect data on the second edge of the SCK cycle                    |
| 2~0 | <b>SPR[2: 0]</b> | <b>SPI Clock rate selection bit</b><br>000: fsys<br>001: fsys/2<br>010: fsys/4<br>011: fsys/8<br>100: fsys/16<br>101: fsys/32<br>110: fsys/64<br>111: fsys/128 |
| 1   | -                | Reserved   |

US0CON1 (9DH) SPI0 Status Register (read/write)

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

USXCON1 (C5H) SPI2~5 Status Register (read/write)

| 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          | <b>SPIF</b>  | <b>SPI data transmission flag</b><br>0: Cleared by software<br>1: Indicates that data transmission has been completed, set by hardware  |
| 6          | <b>WCOL</b>  | <b>Write conflict flag</b><br>0: Cleared by software, indicating that the write conflict has been processed<br>1: Set by hardware to indicate that a conflict is detected<br><br>Transmission direction selection bit |
| 3          | <b>TXE</b>   | <b>Send cache null flag</b><br>0: The send cache is not empty<br>1: The send cache is empty and must be cleared by the software. At this time, the user can write data to FIFO.                                       |
| 2          | <b>DORD</b>  | <b>Transmission direction selection bit</b><br>0: MSB first sent  |

|     |             |   |
|-----|-------------|---|
|     |             | 1: LSB first sent   |
| 1   | <b>SPMD</b> | <b>SPI transmission mode selection:</b><br><br>0: 8-bit mode<br><br>1: 16-bit mode  |
|     | <b>TBIE</b> | <b>Interrupt enable bit when the send cache is empty:</b><br><br>0: When TXE=1, interrupts are not allowed<br><br>1: When TXE=1, SPI interrupt is generated |
| 5~4 | -           | Reserved  |

**US0CON2 (9EH) SPI0 Data register low byte (read/write)**

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

**USXCON2 (C6H) SPI2~5 Data register low byte (read/write)**

| Bit number   | 7          | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|--------------|------------|-----|-----|-----|-----|-----|-----|-----|
| Bit Mnemonic | SPDL[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        | <b>SPDL[7: 0]</b> | <b>SPI data buffer register low byte (8/16 bit mode)</b><br><br>Low byte of data written to data register SPD<br><br>Read data low byte of data register SPD |



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

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

USXCON3 (C7H) SPI2~5 Data register high byte (read/write)

| Bit number   | 7          | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
|--------------|------------|-----|-----|-----|-----|-----|-----|-----|
| Bit Mnemonic | SPDH[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        | <b>SPDH[7: 0]</b> | <p><b>SPI data buffer register high byte (only for 16-bit mode)</b></p> <p>High byte of data written to data register SPD</p> <p>Read data high byte of data register SPD</p> <p>Note: When SPI is set to 16-bit mode, you must write the high byte first, then write the low byte, and start sending immediately after the low byte is written</p> |

### 17.1.2 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.

### 17.1.3 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 SC95F761XB, 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 SC95F761XB:

| SC95F761XB 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 SC95F761XB 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   |

#### 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 set 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.

#### **Slave mode**

- **Mode Startup:**

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

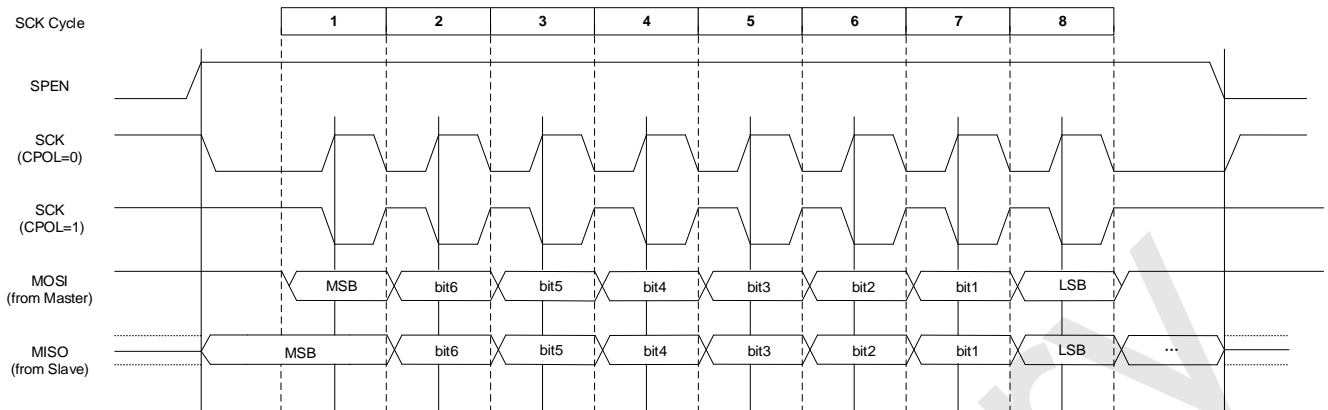
- **Transmitting and Receiving :**

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.

#### **17.1.4 Transfer Form**

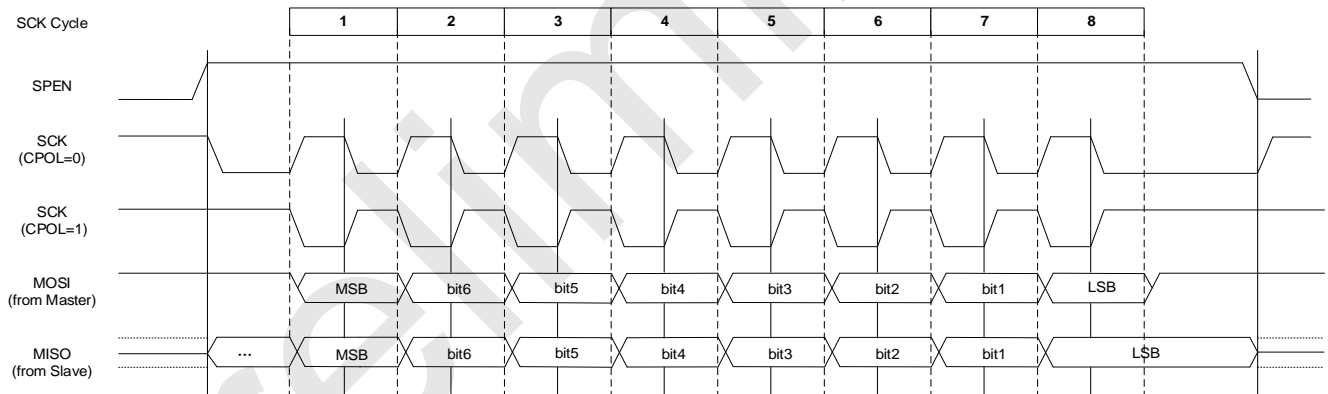
By software setting the CPOL bit and CPHA bit of the SPI control register USnCON0, the user can select four combinations of SPI clock polarity and phase. The CPOL bit defines the polarity of the clock, that is, the level state when idle, and it has little effect on the SPI transmission format. The CPHA 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 sending signal, and the second edge of SCK starts to capture data, So the user must write SPD inside two edges of the first 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

### 17.1.5 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.

### 17.2 TWI

USMDn[1: 0] = 10, n=0~5 One of three serial interface USCI is configured as TWI interface:

- USTXn as SDA signal
- USCKn as CLK signal

The SC95F761XB 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)**

**USXCON0 (C4H) TWI2~5 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          | <b>TWEN</b>  | TWI enable control<br>0: Disable TWI<br>1: Enable TWI  |
| 6          | <b>TWIF</b>  | TWI interrupt flag<br>1. Cleared by software<br>2. Under the following conditions, the interrupt flag bit is set by hardware:<br><b>1) Master mode:</b> <ul style="list-style-type: none"> <li>① Send start signal</li> <li>② After sending the address frame</li> <li>③ Receive or send the data frame</li> </ul> <b>2) Slave mode:</b> |

|     |                    |  |
|-----|--------------------|--|
|     |                    | <ul style="list-style-type: none"> <li>① The first frame address matches successfully</li> <li>② Successfully receive or send 8-bit data</li> <li>③ Receive repeated start condition</li> <li>④ The slave receives a stop signal</li> </ul>  |
| 5   | <b>MSTR</b>        | <p>Master-slave flag</p> <p>0: Slave mode</p> <p>1: Master mode</p> <p>Description:</p> <p>1. When the TWI interface sends a start condition to the bus, it will automatically switch to the master 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   | <b>GCA</b>         | <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 cleared</p>   |
| 3   | <b>AA</b>          | <p><b>Answer enable bit</b></p> <p><b>0: No response, return UACK</b> (the response bit is high)</p> <p><b>1: After receiving a matching address or data, a response ACK is returned</b></p>   |
| 2~0 | <b>STATE[2: 0]</b> | <p>State machine status flag</p> <p><b>Slave mode:</b></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><b>Master mode:</b></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)**
**USXCON1 (C5H) TWI2~5 Control Register 1 (read/write)**

| Bit number   | 7             | 6       | 5   | 4   | 3          | 2   | 1   | 0   |
|--------------|---------------|---------|-----|-----|------------|-----|-----|-----|
| Bit Mnemonic | TXnE/<br>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          | <b>TXnE/RXnE</b> | <p>Send/receive complete flag</p> <p>In the following situations, TXnE/RXnE is set to 1</p> <p>Master mode:</p> <ol style="list-style-type: none"> <li>① The Master sends an address frame (write) and receives an ACK from the slave</li> <li>② The master sends the data and receives the slave ACK</li> <li>③ The Master receives the data, and the Master returns an ACK to the slave</li> </ol> <p>Slave mode:</p> <ol style="list-style-type: none"> <li>① The slave receives the address frame (read), and it matches the slave address (TWA)</li> <li>② The slave receives the data, and the slave returns an ACK to the master</li> <li>③ The slave sends the data and receives the master ACK (AA=1)</li> </ol> <p>Reading and writing to TWIDAT will clear this flag.</p> |
| 6          | <b>STRETCH</b>   | <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          | <b>STA</b>       | <p>Start bit</p> <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          | <b>STO</b>       | <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 | <b>TWCK[3: 0]</b> | <p>TWI communication rate setting in Master mode:</p> <p>0000: <math>f_{SYS}/1024</math></p> <p>0001: <math>f_{SYS}/512</math></p> <p>0010: <math>f_{SYS}/256</math></p> <p>0011: <math>f_{SYS}/128</math></p> <p>0100: <math>f_{SYS}/64</math></p> <p>0101: <math>f_{SYS}/32</math></p> <p>0110: <math>f_{SYS}/16</math></p> <p>Others: Reserved</p> <p><b>Note:</b></p> <ol style="list-style-type: none"> <li>1. The setting is invalid in slave mode. The maximum clock frequency is 400 kHz;</li> <li>2. The clock source of TWI follow the System clock <math>f_{SYS}</math></li> </ol> |
| 7   | -                 | Reserved  |

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

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

**USXCON2 (C6H) TWI2~5 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        | <b>TWA[6: 0]</b> | TWI address register<br><br>TWA[6: 0] cannot be written as all 0s, 00H is dedicated to general address addressing. Invalid setting in Master mode |
| 0          | <b>GC</b>        | TWI general address enable<br><br>0: Forbid to respond to general address 00H<br>1: Allow response to general address 00H                         |

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

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

**USXCON3 (C7H) TWI2~5 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        | <b>TWDAT[7: 0]</b> | TWI Data buffer register |

## 17.2.1 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.

## 17.2.2 Slave Operating Mode

- **Mode Start:**

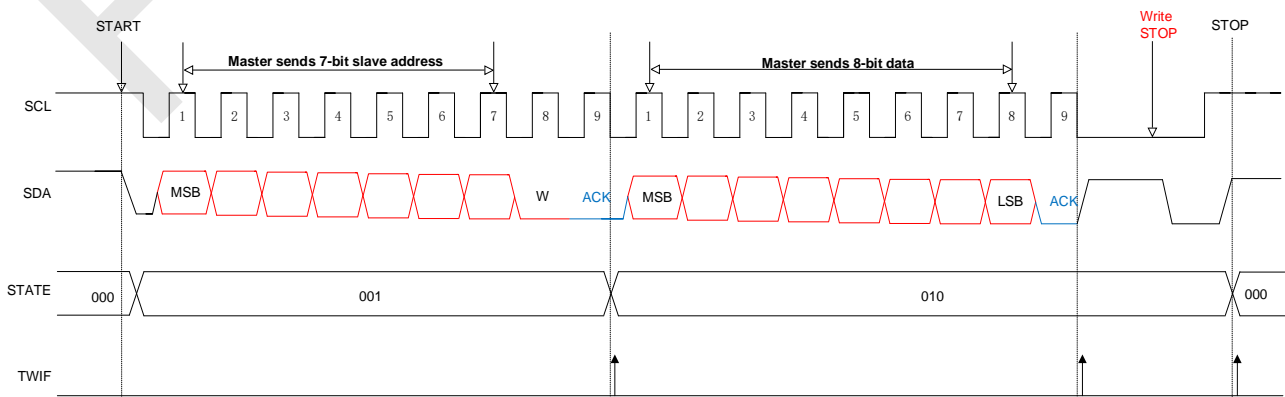
When the TWI enable flag is turned on (TWEN = 1) and the start signal sent by the Master is received at the same time, the mode is started.

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:

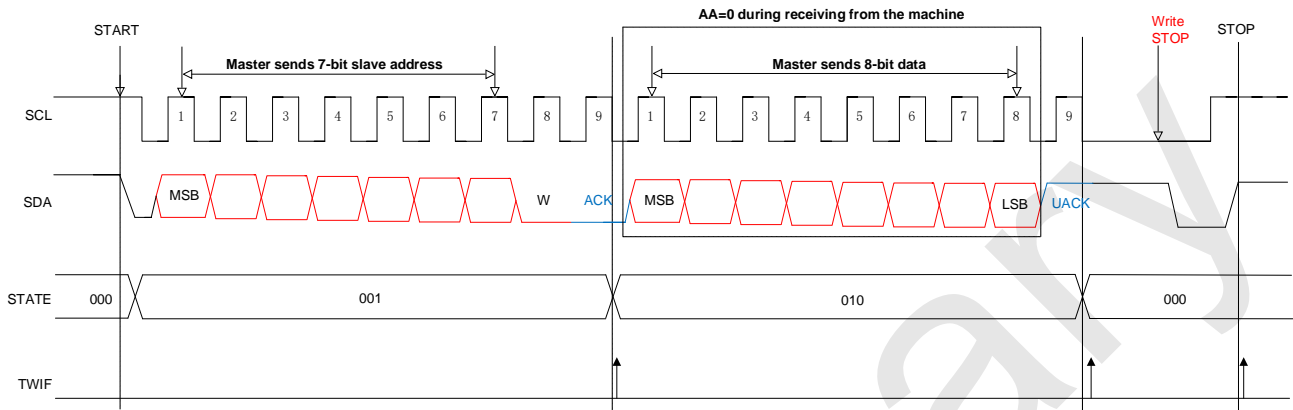
- **Non-general Address Response, Slave Device Receiving Mode:**

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.



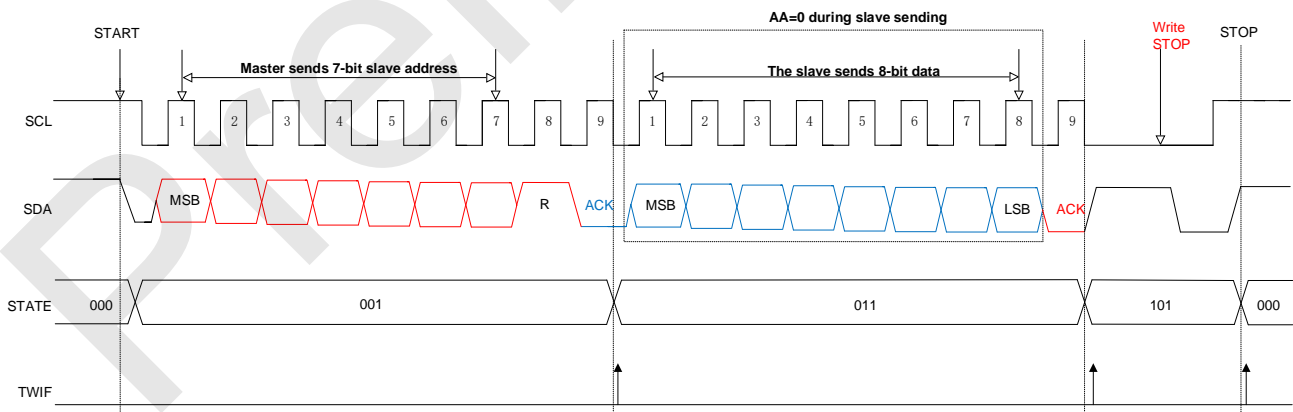
- If the slave responds to a high level (during the receiving process, the AA value in the slave register is rewritten to 0), it means that after the current byte is transmitted, the slave will actively end the transmission and return to the idle state ( STATE[2: 0] = 000), no longer receive data from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



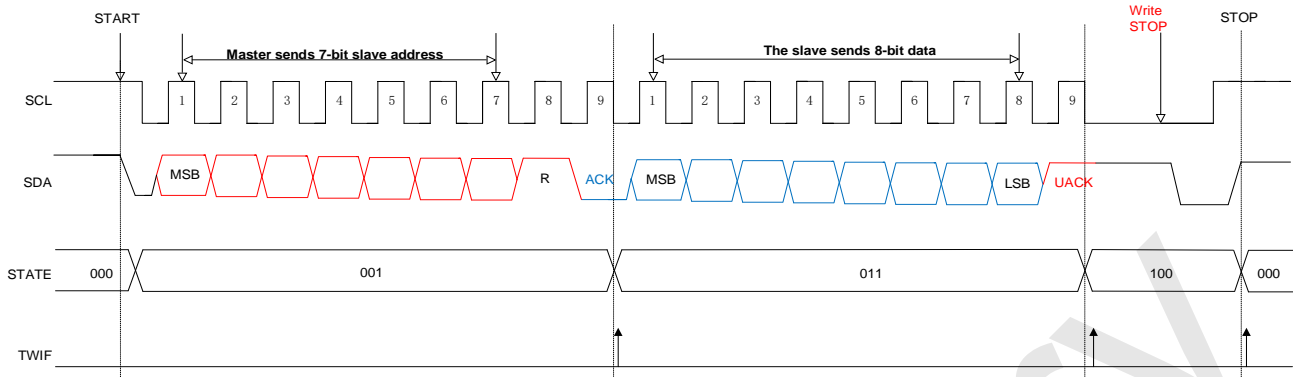
● **Non-general Address Response, Slave Device Transmitting Mode:**

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).



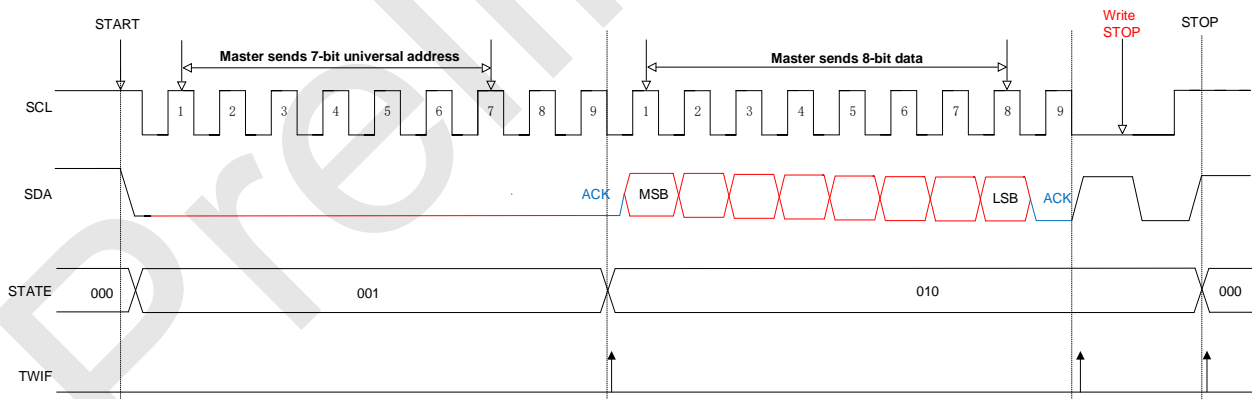
- If the master responds to a high level, the slave STATE[2: 0] = 100, waiting for the master's stop signal or restart signal.



● **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.

### 17.2.3 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.

### 17.2.4 Master operating Mode

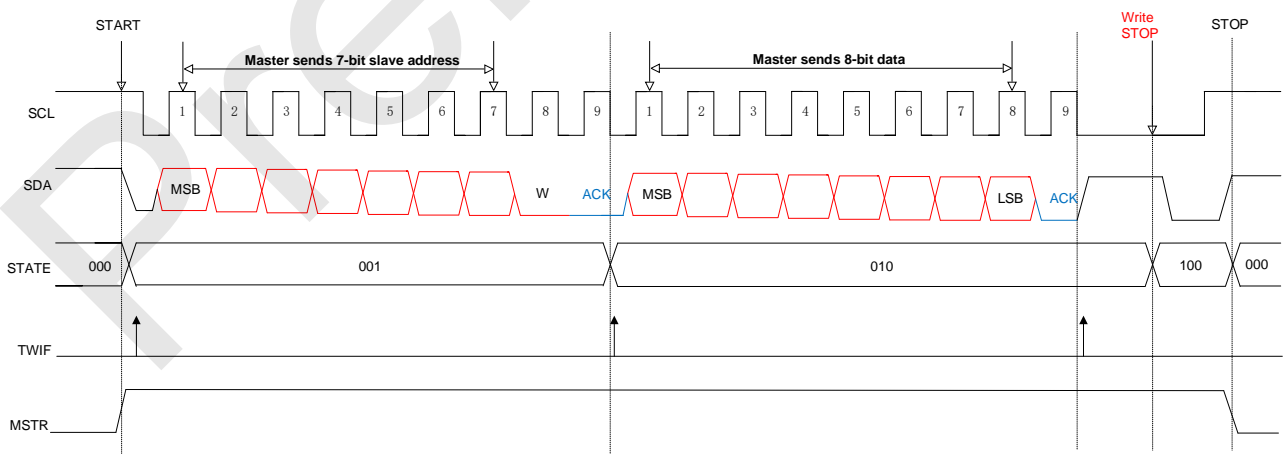
- **Mode startup:**

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.

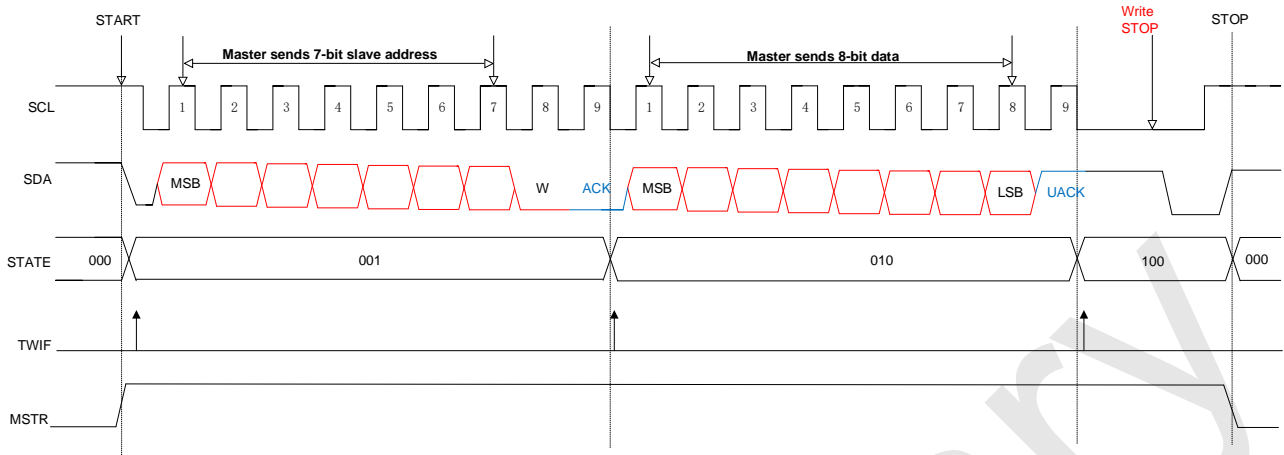
- **TWI Master sending mode:**

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:

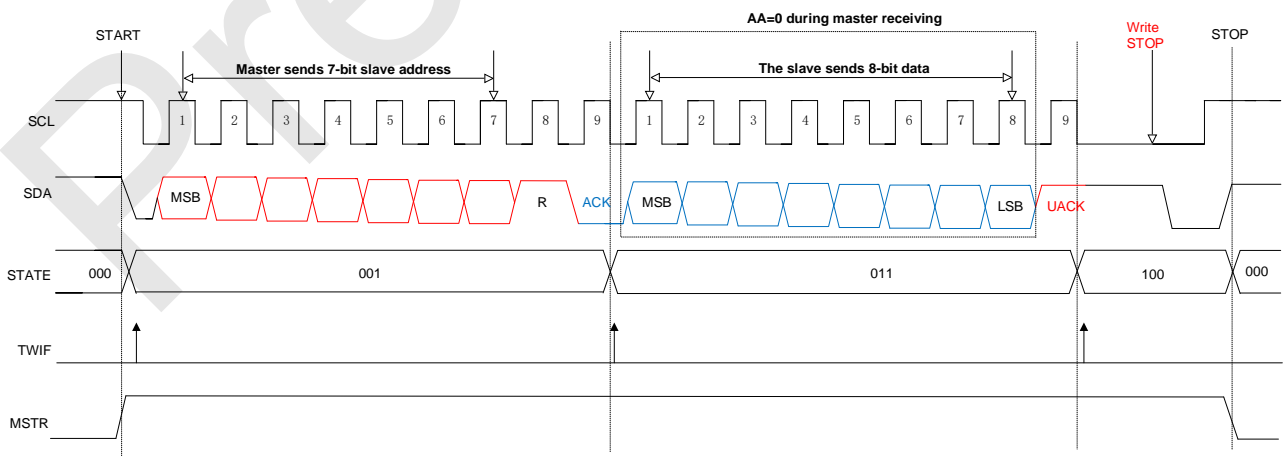


● **TWI Master Receiving Mode:**

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;
2. 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:



**17.2.5 Master Mode Operation Steps**

1. Configure USMDn[1: 0] and select TWI mode;

2. Configure the TWIn control register USnCON0: TWEN = 1, enable TWI
3. Configure the TWIn control register USnCON1: configure the TWI communication rate (TWCK[3: 0]), set the start bit STA to "1"
4. Configure TWIn address register USnCON3: write "slave address plus read and write bits" into TWDAT, and send out an address frame on the bus
5. 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;
6. 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.
7. 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!



### 17.3 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, the clearing mode is "Write 1 clear".
4. Can realize full duplex communication

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)**

**USXCON0 (C4H) Serial Port 3~6 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/ Write 1 clear | R/ Write 1 clear |
| POR          | 0   | 0   | 0   | 0   | 0   | 0   | 0                | 0                |

| Bit number | Bit Mnemonic | Description  |
|------------|--------------|--|
| 7~6        | <b>SM0~1</b> | Serial communication mode control bit<br><br>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 |

|   |            |  |
|---|------------|--|
|   |            | <p>transmit shift clock. 8 bits are sent and received per frame, and the lower bits are received or sent first;</p> <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 | <b>SM2</b> | <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><b>Baud rate multiplier setting bit, only valid in mode 0 (SM0~1 = 00):</b></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 | <b>REN</b> | <p>Receive permission control bit</p> <p>0: It is not allowed to receive data;</p> <p>1: Allow to receive data.</p>  |
| 3 | <b>TB8</b> | Only valid for mode 3, which is the 9th bit of the transmitted data  |
| 2 | <b>RB8</b> | Only valid for mode 3, which is the 9th bit of the received data   |
| 1 | <b>TI</b>  | Send interrupt flag  |
| 0 | <b>RI</b>  | 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)**

**USXCON1 (C5H) Serial Port 3~6 Baud Rate Control Register Low Bit (read/write)**

| Bit number             | 7              | 6              | 5              | 4              | 3              | 2              | 1              | 0              |
|------------------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| Symbol                 | BAUD1L [7: 0]  |                |                |                |                |                |                |                |
| Read/<br>Write         | Read/<br>Write | Read/<br>Write | Read/<br>Write | Read/<br>Write | Read/<br>Write | Read/<br>Write | Read/<br>Write | Read/<br>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)**

**USXCON2 (C6H) Serial Port 3~6 Baud Rate Control Register High Bit (read/write)**

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

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

|  |  |   |
|--|--|---|
|  |  | 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)**

**USXCON3 (C7H) Serial 3~6 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        | <b>SBUF1[7: 0]</b> | <b>Serial Data Buffer Register</b><br>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. |

## 18 High-speed Analog-to-Digital Converter (ADC)

The SC95F761XB integrates 17 channels of 12-bit high-precision 1M high-speed ADC, and the external 16 channels of ADC and other functions of the IO port are multiplexed. The internal channel can be connected to 1/4 VDD, and the internal 2.048V, 1.024V or 2.4V reference voltage is used for Measure the VDD voltage. 1 MHz super-high-speed sampling clock, the total time from sampling to completion of conversion is as low as 2 $\mu$ s

There are 4 choices for the ADC reference voltage of SC95F761XB:

- ① VDD pin (that is directly the internal VDD);
- ② The reference voltage output by the internal Regulator is accurately 2.048V.
- ③ The reference voltage output by the internal Regulator is exactly 1.024V.
- ④ The reference voltage output by the internal Regulator is exactly 2.4V.

### 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          | <b>ADCEN</b> | Power on ADC<br>0: Disable the ADC module power<br>1: Enable the ADC module power  |
| 6          | <b>ADCS</b>  | ADC start trigger control (ADC Start)<br><br>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. |

|     |                    |  |
|-----|--------------------|--|
|     |                    | <b>Note: After writing "1" to the ADCS, do not write to the ADCCON register until the interrupt flag EOC/ADCIF is set.</b>   |
| 5   | <b>EOC /ADCIF</b>  | <p>Conversion complete/ADC Interrupt Flag (End Of Conversion / ADC Interrupt Flag)</p> <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 | <b>ADCIS[4: 0]</b> | <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: select AIN2 as ADC input</p> <p>00011: select AIN3 as ADC input</p> <p>00100: select AIN4 as ADC input</p> <p>00101: select AIN5 as ADC input</p> <p>00110: select AIN6 as ADC input</p> <p>00111: select AIN7 as ADC input</p> <p>01000: select AIN8 as ADC input</p> <p>01001: select AIN9 as ADC input</p> <p>01010: select AIN10 as ADC input</p> <p>01011: select AIN11 as ADC input</p> <p>01100: select AIN12 as ADC input</p> <p>01101: select AIN13 as ADC input</p> <p>01110: select AIN14 as ADC input</p> <p>01111: select AIN15 as ADC input</p> |

|  |  |  |
|--|--|--|
|  |  | 10000~11110: reserved<br><br>11111: ADC input is 1/4 V <sub>DD</sub> , which can be used to measure power supply voltage |
|--|--|--|

**ADCCFG2 (B5H) ADC Set Register 2 (read/write)**

| Bit number             | 7 | 6 | 5 | 4              | 3              | 2              | 1 | 0 |
|------------------------|---|---|---|----------------|----------------|----------------|---|---|
| Symbol                 | - | - | - | LOWSP[2: 0]    |                |                | - | - |
| Read/<br>Write         | - | - | - | Read/<br>Write | Read/<br>Write | Read/<br>Write | - | - |
| Initial power-on value | x | x | x | 0              | 0              | 0              | x | x |

| Bit number | Bit Mnemonic       | Description  |
|------------|--------------------|--|
| 4~2        | <b>LOWSP[2: 0]</b> | <p>ADC sampling period selection:</p> <p>100: The sampling time is 3 system clocks, (about 100ns @f<sub>sys</sub> = 32 MHz)</p> <p>101: The sampling time is about 6 system clocks, (about 200ns @f<sub>sys</sub> = 32 MHz)</p> <p>110: The sampling time is about 16 system clocks, (about 500ns @ f<sub>sys</sub> = 32 MHz)</p> <p>111: The sampling time is about 32 system clocks, (about 1000ns @ f<sub>sys</sub> = 32 MHz)</p> <p>Other: Reserved</p> <p>Description: The total time from ADC sampling to completion of conversion T<sub>ADC</sub> = sampling time + conversion time</p> <p>The ADC conversion time 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 | EAIN15 | EAIN14 | EAIN13 | EAIN12 | EAIN11 | EAIN10 | EAIN9 | EAIN8 |
| R/W          | R/W    | R/W    | R/W    | R/W    | R/W    | R/W    | R/W   | R/W   |
| POR          | 0      | 0      | 0      | 0      | 0      | 0      | 0     | 0     |

| Bit number | Bit Mnemonic                    | Description   |
|------------|---------------------------------|---|
| 0          | <b>EAINx</b><br><b>(x=0~15)</b> | <b>ADC port setting register</b><br>0: Set AINx as IO port<br>1: Set AINx as ADC input and automatically remove the pull-up resistor. |



**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] |     | OP_BL | DISJTG | IAPS[1: 0] |     | LDSIZE[1:0] |   |
| R/W          | R/W         | R/W | R/W   | R/W    | R/W        | R/W | Read only   |   |
| POR          | n           | n   | n     | n      | n          | n   | n           | n |

| Bit number | Bit Mnemonic | Description  |
|------------|--------------|--|
| 7~6        | VREFS[1: 0]  | <b>Reference voltage selection (the initial value is transferred from Customer Option, the user can modify the setting)</b><br>00: Set VREF of ADC to VDD;<br>01: Set the VREF of ADC to the internal accurate 2.048V;<br>10: Set the VREF of ADC to the internal accurate 1.024V;<br>11: Set the VREF of ADC to the internal accurate 2.4V; |

**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       | <b>ADCV[11: 4]</b> | The high 8-bit value of ADC conversion value |
| 3~0        | <b>ADCV[3: 0]</b>  | 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          | <b>EADC</b>  | ADC interrupt enable control<br>0: Do not allow EOC/ADCIF to generate interrupts<br>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          | <b>IPADC</b> | ADC interrupt priority selection<br>0: Set the interrupt priority of ADC to "low"<br>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. .

Preliminary

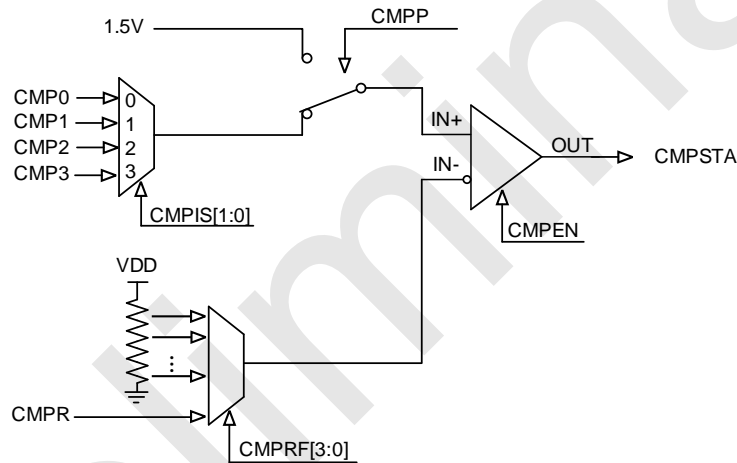
## 19 Analog Comparator

The SC95F761XB has a built-in analog comparator(CMP), CMP interrupt can wake up STOP mode. It can be used for alarm circuit, power supply voltage monitoring circuit, zero-crossing detection circuit, etc.

This comparator has four analog signal positive input terminals: CMP0~3, which can be switched through CMPIS [1: 0]. The negative input voltage can be switched to one of the external voltage on the CMPP pin or the internal 16-level comparison voltage through CMPRF[3: 0].

CMPIM[1: 0] can conveniently set the interrupt mode of the comparator. When the interrupt condition set by CMPIM[1: 0] occurs, the comparator interrupt flag CMPIF will be set to 1, and the interrupt flag needs to be cleared by software.

### 19.1 Block Diagram of Analog Comparator



Block Diagram of Analog Comparator

#### CMPCON (B7H) Analog Comparator Control Register (read/write)

| Bit number   | 7     | 6     | 5      | 4 | 3           | 2   | 1   | 0   |
|--------------|-------|-------|--------|---|-------------|-----|-----|-----|
| Bit Mnemonic | CMPEN | CMPIF | CMPSTA | - | CMPRF[3: 0] |     |     |     |
| R/W          | R/W   | R/W   | R/W    | - | R/W         | R/W | R/W | R/W |
| POR          | 0     | 0     | 0      | x | 0           | 0   | 0   | 0   |

| Bit number | Bit Mnemonic       | Description   |
|------------|--------------------|---|
| 7          | <b>CMPEN</b>       | <b>Analog comparator enable control bit</b><br>0: Disable the analog comparator;<br>1: Enable analog comparator   |
| 6          | <b>CMPIF</b>       | <b>Analog comparator interrupt flag</b><br>0: Comparator interrupt is not triggered;<br>1: When the comparator meets the interrupt trigger condition, this bit will be automatically set to 1 by the hardware. If IE1[5] (ECMP) is also set to 1 at this time, the comparator interrupt is generated. After the comparator interrupt occurs, the hardware will not automatically clear this bit, the user's software must be responsible for clearing this bit.   |
| 5          | <b>CMPSTA</b>      | <b>Analog comparator output status</b><br>0: The voltage at the positive terminal of the comparator is less than the voltage at the negative terminal<br>1: The voltage at the positive terminal of the comparator is greater than the voltage at the negative terminal   |
| 3~0        | <b>CMPRF[3: 0]</b> | <b>Analog comparator negative terminal comparison voltage selection:</b><br>0000: Choose CMPR as the comparison voltage of the analog comparator;<br>0001: Choose $1/16V_{DD}$ as the comparison voltage of the analog comparator;<br>0010: Choose $2/16V_{DD}$ as the comparison voltage of the analog comparator;<br>0011: Choose $3/16V_{DD}$ as the comparison voltage of the analog comparator;<br>0100: Choose $4/16V_{DD}$ as the comparison voltage of the analog comparator;<br>0101: Choose $5/16V_{DD}$ as the comparison voltage of the analog comparator;<br>0110: Choose $6/16V_{DD}$ as the comparison voltage of the analog comparator; |

|   |   |   |
|---|---|---|
|   |   | <p>0111: Choose <math>7/16V_{DD}</math> as the comparison voltage of the analog comparator;</p> <p>1000: Choose <math>8/16V_{DD}</math> as the comparison voltage of the analog comparator;</p> <p>1001: Choose <math>9/16V_{DD}</math> as the comparison voltage of the analog comparator;</p> <p>1010: Choose <math>10/16V_{DD}</math> as the comparison voltage of the analog comparator;</p> <p>1011: Choose <math>11/16V_{DD}</math> as the comparison voltage of the analog comparator;</p> <p>1100: Choose <math>12/16V_{DD}</math> as the comparison voltage of the analog comparator;</p> <p>1101: Choose <math>13/16V_{DD}</math> as the comparison voltage of the analog comparator;</p> <p>1110: Choose <math>14/16V_{DD}</math> as the comparison voltage of the analog comparator;</p> <p>1111: Choose <math>15/16V_{DD}</math> as the comparison voltage of the analog comparator;</p> |
| 4 | - | Reserved  |

**CMPCFG (B6H) Analog Comparator Setting Register (read/write)**

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

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

|     |                    |  |
|-----|--------------------|--|
| 4   | <b>CMPP</b>        | <p><b>Analog comparator positive input options:</b></p> <p>0: The positive input of the analog comparator is CMP0~3, which is set by CMPIS[1:0]</p> <p>1: The positive input of the analog comparator is the internal 1.5V reference voltage</p>   |
| 3~2 | <b>CMPIM[1: 0]</b> | <p><b>Analog comparator interrupt mode selection:</b></p> <p>00: no interrupt</p> <p>01: Rising edge interrupt: An interrupt will be generated after IN+ is less than IN- to greater than IN-;</p> <p>10: Falling edge interrupt: An interrupt will be generated after IN+ is greater than IN- to less than IN-;</p> <p>11: Double-edge interrupt: IN+ from less than IN- to greater than IN-, or IN+ from greater than IN- to less than IN- will generate an interrupt;</p>   |
| 1~0 | <b>CMPIS[1: 0]</b> | <p><b>Analog comparator positive terminal input channel selection:<br/>When CMPP is 1, this bit is invalid</b></p> <p>00: Choose CMP0 as the input of the positive terminal of the analog comparator;</p> <p>01: Select CMP1 as the input of the positive terminal of the analog comparator;</p> <p>10: Select CMP2 as the input of the positive terminal of the analog comparator;</p> <p>11: Choose CMP3 as the input of the positive terminal of the analog comparator;</p> |
| 7~5 | -                  | Reserved   |



## 20 CRC Module

The SC95F761XB 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@32 MHz.

### Hardware CRC mode 2: CRC calculation processing for APROM:

It can be used to generate the 32-bit CRC value of APROM (ie 64 Kbytes 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".

It takes about 16.5ms@32 MHz to calculate CRC for 64 Kbytes APROM.

### The hardware CRC parameter model of SC95F761XB:

|                        |   |
|------------------------|---|
| 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)<br>Write "1" to this bit to reset CRCDR to all 1s  |
| 0          | CRCSTA       | CRC hardware calculation start bit<br>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<sub>n</sub> (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<sub>n</sub> (n = 0~3) bits is as follows:

| Bit number | Bit Mnemonic     | Description   |
|------------|------------------|---|
| Bit31~0    | Qx<br>(x = 0~31) | Hardware CRC mode 1: CRC operation processing on specified data: <ol style="list-style-type: none"> <li>1. You must write CRCRST first, reset CRCDR to all 1s</li> <li>2. When CRCREG is written, the hardware automatically calculates the CRC result and continues to store it in CRCDR</li> <li>3. When needed, read the CRC calculation result instantly</li> </ol> |

|  |  |  |
|--|--|--|
|  |  | <p>Hardware CRC mode 2: CRC calculation processing on APROM:</p> <ol style="list-style-type: none"> <li>1. Started by CRCSTA, at this time the CPU automatically enters IDLE</li> <li>2. Automatically reset CRCDR to all 1:</li> </ol> <p><b>The hardware CRC calculation range does not include the IAP area.</b> The calculation range of CRC is divided into four types according to the value of IAPS[1: 0]:</p> <ol style="list-style-type: none"> <li>① IAPS[1:0]=00(Flash ROM last 0K available for IAP): 0000H ~ before last 0K</li> <li>② IAPS[1:0]=01(Flash ROM last 1K available for IAP): 0000H ~ before last 1K</li> <li>③ IAPS[1:0]=10(Flash ROM last 1K available for IAP): 0000H ~ before last 2K</li> <li>④ IAPS[1:0]=11(Flash ROM all available for IAP): 0000H ~ All Flash ROM</li> </ol> <ol style="list-style-type: none"> <li>3. After the end, the CPU automatically exits IDLE, you can read the CRC calculation result</li> </ol> <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 CRCREG 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_();  
    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_();
_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 Multiplier-Divider Unit (MDU)

The SC95F761XB provides a 16-bit hardware multiplier and divider, which consists of extended accumulators EXA0~EXA3, extended B register EXB and operation control register OPERCON. It can replace software for 16-bit × 16-bit multiplication and 32-bit/16-bit division.

The SC95F761XB 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]   |    |   |   |   |   |            |            | 0000000b  |
| EXA1    | EAH     | Extended Accumulator 1     | EXA [15: 8]  |    |   |   |   |   |            |            | 0000000b  |
| EXA2    | EBH     | Extended Accumulator 2     | EXA [23: 16] |    |   |   |   |   |            |            | 0000000b  |
| EXA3    | ECH     | Extended Accumulator 3     | EXA [31: 24] |    |   |   |   |   |            |            | 0000000b  |
| EXBL    | EDH     | Extended B register L      | EXB [7: 0]   |    |   |   |   |   |            |            | 0000000b  |
| EXBH    | EEH     | Extended B register H      | EXB [15: 8]  |    |   |   |   |   |            |            | 0000000b  |
| OPERCON | EFH     | Operation control register | OPE<br>RS    | MD | - | - | - | - | CRCR<br>ST | CRCS<br>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                  | <b>OPERS</b> | <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                  | <b>MD</b>    | <p>Multiplication and division</p> <p>0: Multiplication operation. The multiplicand and multiplier are written and the product is read as follows:</p> <table border="1" data-bbox="609 1214 1452 1691"> <thead> <tr> <th>Byte<br/>Operand</th> <th>Byte 3</th> <th>Byte 2</th> <th>Byte 1</th> <th>Byte 0</th> </tr> </thead> <tbody> <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> </tbody> </table> <p>1: Divide operation, write the dividend and divisor, read the quotient and remainder as follows:</p> <table border="1" data-bbox="609 1879 1452 2033"> <thead> <tr> <th>Byte<br/>Operand</th> <th>Byte 3</th> <th>Byte 2</th> <th>Byte 1</th> <th>Byte 0</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table> | Byte<br>Operand | Byte 3 | Byte 2 | Byte 1 | Byte 0 | multiplicand 16bit | - | - | EXA1 | EXA0 | multiplier 16bit | - | - | EXBH | EXBL | multiplier 32bit | EXA3 | EXA2 | EXA1 | EXA0 | Byte<br>Operand | Byte 3 | Byte 2 | Byte 1 | Byte 0 |  |  |  |  |  |
| Byte<br>Operand    | Byte 3       | Byte 2  | Byte 1          | Byte 0 |        |        |        |                    |   |   |      |      |                  |   |   |      |      |                  |      |      |      |      |                 |        |        |        |        |  |  |  |  |  |
| multiplicand 16bit | -            | -   | EXA1            | EXA0   |        |        |        |                    |   |   |      |      |                  |   |   |      |      |                  |      |      |      |      |                 |        |        |        |        |  |  |  |  |  |
| multiplier 16bit   | -            | -   | EXBH            | EXBL   |        |        |        |                    |   |   |      |      |                  |   |   |      |      |                  |      |      |      |      |                 |        |        |        |        |  |  |  |  |  |
| multiplier 32bit   | EXA3         | EXA2  | EXA1            | EXA0   |        |        |        |                    |   |   |      |      |                  |   |   |      |      |                  |      |      |      |      |                 |        |        |        |        |  |  |  |  |  |
| Byte<br>Operand    | 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/fsys.

## 22 Electrical Characteristics

### 22.1 Absolute Maximum Ratings

| Symbol             | Parameter                         | Min Value | Max Value            | UNIT |
|--------------------|-----------------------------------|-----------|----------------------|------|
| VDD/VSS            | DC supply voltage                 | -0.3      | 5.5                  | V    |
| Voltage ON any Pin | Input/output voltage of any pin   | -0.3      | V <sub>DD</sub> +0.3 | V    |
| T <sub>A</sub>     | Operating temperature             | -40       | 105                  | °C   |
| T <sub>STG</sub>   | Storage temperature               | -55       | 125                  | °C   |
| I <sub>VDD</sub>   | Current value flowing through VDD | -         | 200                  | mA   |
| I <sub>VSS</sub>   | Current value flowing through VSS | -         | 200                  | mA   |

### 22.2 Recommended Operating Conditions

| Symbol          | Parameter             | Min Value | Max Value | UNIT | System Clock frequency |
|-----------------|-----------------------|-----------|-----------|------|------------------------|
| V <sub>DD</sub> | Operating Voltage     | 2.0       | 5.5       | V    | 32 MHz                 |
| T <sub>A</sub>  | Operating temperature | -40       | 105       | °C   |                        |

## 22.3 Flash ROM Characteristics

| Symbol        | Parameter           | Min Value | Typical Values | Max Value | UNIT          | Condition                   |
|---------------|---------------------|-----------|----------------|-----------|---------------|-----------------------------|
| $N_{END}$     | Wipe the number     | 100,000   | -              | -         | Cycle<br>s    |                             |
| $T_{DR}$      | Data Retention Time | 100       | -              | -         | Years         | $T_A = +25^{\circ}\text{C}$ |
| $T_{S-Erase}$ | Sector Erase Time   | -         | 5              | -         | ms            | $T_A = +25^{\circ}\text{C}$ |
| $T_{Erase}$   | All Chip Erase Time | 30        | -              | 40        | ms            | $T_A = +25^{\circ}\text{C}$ |
| $T_{Write}$   | Byte Program Time   | -         | 150            | -         | $\mu\text{s}$ | $T_A = +25^{\circ}\text{C}$ |

## 22.4 DC Characteristics

( $V_{DD} = 5\text{V}, T_A = +25^{\circ}\text{C}$ , Unless otherwise specified)

| Symbol         | Parameter         | Minimum | Typical value | Maximum | Unit          | Test Conditions         |
|----------------|-------------------|---------|---------------|---------|---------------|-------------------------|
| <b>Current</b> |                   |         |               |         |               |                         |
| $I_{op1}$      | Operating current | -       | 6             | -       | mA            | $f_{sys}=32\text{ MHz}$ |
| $I_{op2}$      | Operating current | -       | 3.8           | -       | mA            | $f_{sys}=16\text{ MHz}$ |
| $I_{op3}$      | Operating current | -       | 2.4           | -       | mA            | $f_{sys}=8\text{ MHz}$  |
| $I_{op4}$      | Operating current | -       | 1.7           | -       | mA            | $f_{sys}=4\text{ MHz}$  |
| $I_{pd1}$      | Stand-by current  | -       | 2.2           | -       | $\mu\text{A}$ |                         |

|                         |                                 |                    |     |                      |    |  |
|-------------------------|---------------------------------|--------------------|-----|----------------------|----|--|
|                         | (Power Down Mode)               |                    |     |                      |    |  |
| I <sub>IDL1</sub>       | Stand-by current<br>(IDLE Mode) | -                  | 2.2 | -                    | mA | f <sub>sys</sub> =32 MHz   |
| I <sub>BTM</sub>        | Base Timer Operating current    | -                  | 1.3 | 3                    | μA | BTMFS[3: 0]=<br>1000<br>Generate an interrupt every 4.0 seconds  |
| I <sub>WDT</sub>        | WDT current                     | -                  | 1.3 | 3                    | μA | WDTCKS[2: 0]=<br>000<br>WDT overflow time 500ms  |
| IO port characteristics |                                 |                    |     |                      |    |  |
| V <sub>IH1</sub>        | Input high voltage              | 0.7V <sub>DD</sub> | -   | V <sub>DD</sub> +0.3 | V  | GPIO   |
| V <sub>IL1</sub>        | Input low voltage               | -0.3               | -   | 0.3V <sub>DD</sub>   | V  |  |
| V <sub>IH2</sub>        | Input high voltage              | 0.8V <sub>DD</sub> | -   | V <sub>DD</sub>      | V  | Schmitt trigger input: RST   |
| V <sub>IL2</sub>        | Input low voltage               | -0.2               | -   | 0.2V <sub>DD</sub>   | V  | tCK / tDIO<br>UART0 input RX0<br>USCI signal input port<br>INT0~2<br>PWM fault detection FLT<br>Timer clock input port |

|                      |   |   |     |   |    |  |
|----------------------|---|---|-----|---|----|--|
|                      |   |   |     |   |    | Timer capture port   |
| I <sub>OL1</sub>     | Output low current  | - | 27  | - | mA | V <sub>Pin</sub> =0.4V   |
| I <sub>OL2</sub>     | Output low current  | - | 50  | - | mA | V <sub>Pin</sub> =0.8V   |
| I <sub>OHSP10A</sub> | SPI0 Signal port:<br>USCK0 (P04)<br>USTX0 (P05)<br>USRX0 (P06)<br><br>Output high current @<br>V <sub>Pin</sub> =4.3V                 | - | 21  | - | mA | Only applicable to SPI0 data transmission  |
| I <sub>OH1</sub>     | Drive capability fixed I/O group:<br>P3H <sup>1</sup><br>P4<br>P5<br><br>Output high current @<br>V <sub>Pin</sub> =4.3V              | - | 10  | - | mA | Pxyz=0,I <sub>OH</sub> level 0<br><br>Pxyz=1,I <sub>OH</sub> level 1<br><br>Pxyz=2,I <sub>OH</sub> level 2<br><br>Pxyz=3,I <sub>OH</sub> level 3 |
| I <sub>OH2</sub>     | Drive capability can be graded IO group:<br>P0<br>P1<br>P2<br>P3L <sup>1</sup><br><br>Output high current @<br>V <sub>Pin</sub> =4.3V | - | 10  | - | mA | Pxyz=0,I <sub>OH</sub> level 0   |
|                      |   | - | 7   | - | mA | Pxyz=1,I <sub>OH</sub> level 1   |
|                      |   | - | 5   | - | mA | Pxyz=2,I <sub>OH</sub> level 2   |
|                      |   | - | 2.5 | - | mA | Pxyz=3,I <sub>OH</sub> level 3   |
| I <sub>OHSP10B</sub> | SPI0 Signal port:<br>USCK0 (P04)<br>USTX0 (P05)<br>USRX0 (P06)<br><br>Output high current @<br>V <sub>Pin</sub> =4.7V                 | - | 8   | - | mA | Only applicable to SPI0 data transmission  |

|           |   |   |    |   |            |  |
|-----------|---|---|----|---|------------|--|
| $I_{OH3}$ | Drive capability fixed I/O group:<br>P3H <sup>1</sup><br>P4<br>P5<br>Output high current @<br>$V_{Pin}=4.7V$                  | - | 4  | - | mA         | Pxyz=0,IOH level 0<br><br>Pxyz=1,IOH level 1<br><br>Pxyz=2,IOH level 2<br><br>Pxyz=3,IOH level 3 |
| $I_{OH4}$ | Drive capability can be graded IO group:<br>P0<br>P1<br>P2<br>P3L <sup>1</sup><br><br>Output high current @<br>$V_{Pin}=4.3V$ | - | 4  | - | mA         | Pxyz=0, $I_{OH}$ level 0   |
|           |   | - | 3  | - | mA         | Pxyz=1, $I_{OH}$ level 1   |
|           |   | - | 2  | - | mA         | Pxyz=2, $I_{OH}$ level 2   |
|           |   | - | 1  | - | mA         | Pxyz=3, $I_{OH}$ level 3   |
| $R_{PH1}$ | Pull-up resistor  | - | 30 | - | k $\Omega$ |  |

Note:

1. H and L:

- H: the port is four bits higher.
- L: The port is four bits lower.

( $V_{DD} = 3.3V, T_A = +25^\circ C$ , Unless otherwise specified)

| Symbol    | Parameters        | Min Value | Typical value | Max Value | Unit | Test condition |
|-----------|-------------------|-----------|---------------|-----------|------|----------------|
| Current   |                   |           |               |           |      |                |
| $I_{op5}$ | Operating current | -         | 6             | -         | mA   | fsys=32 MHz    |
| $I_{op6}$ | Operating current | -         | 3.8           | -         | mA   | fsys=16 MHz    |
| $I_{op7}$ | Operating current | -         | 2.4           | -         | mA   | fsys=8 MHz     |

|                         |   |             |     |              |         |  |
|-------------------------|---|-------------|-----|--------------|---------|--|
| $I_{op8}$               | Operating current                               | -           | 1.7 | -            | mA      | $f_{sys}=4$ MHz  |
| $I_{pd2}$               | Stand-by current(Power Down Mode)               | -           | 2.2 | -            | $\mu$ A |  |
| $I_{IDL2}$              | Stand-by current (IDLE Mode)                    | -           | 2.2 | -            | mA      | $f_{sys}=32$ MHz   |
| IO port characteristics |   |             |     |              |         |  |
| $V_{IH3}$               | Input high voltage                              | $0.7V_{DD}$ | -   | $V_{DD}+0.3$ | V       |  |
| $V_{IL3}$               | Input low voltage                               | -0.3        | -   | $0.3V_{DD}$  | V       |  |
| $V_{IH4}$               | Input high voltage                              | $0.8V_{DD}$ | -   | $V_{DD}$     | V       | Schmitt trigger input: RST   |
| $V_{IL4}$               | Input low voltage                               | -0.2        | -   | $0.2V_{DD}$  | V       | tCK / tDIO<br>UART0 input RX0<br>USCI signal input port<br>INT0~2<br>PWM fault detection FLT<br>Timer clock input port<br>Timer capture port |
| $I_{OL3}$               | Output low current                              | -           | 22  | -            | mA      | $V_{Pin}=0.4V$   |
| $I_{OL4}$               | Output low current                              | -           | 35  | -            | mA      | $V_{Pin}=0.8V$   |
| $I_{OHSP10C}$           | SPI0 Signal port:<br>USCK0 (P04)<br>USTX0 (P05) | -           | 7   | -            | mA      | Only applicable to SPI0 data transmission  |



|           |   |   |     |   |            |  |
|-----------|---|---|-----|---|------------|--|
|           | USRX0 (P06)<br><br>Output high current @<br>$V_{Pin}=3.0V$  |   |     |   |            |  |
| $I_{OH5}$ | Drive capability fixed I/O group:<br>P3H <sup>1</sup><br>P4<br>P5<br><br>Output high current @<br>$V_{Pin}=3.0V$              | - | 3   | - | mA         | Pxyz=0,IOH level 0<br><br>Pxyz=1,IOH level 1<br><br>Pxyz=2,IOH level 2<br><br>Pxyz=3,IOH level 3 |
| $I_{OH6}$ | Drive capability can be graded IO group:<br>P0<br>P1<br>P2<br>P3L <sup>1</sup><br><br>Output high current @<br>$V_{Pin}=3.0V$ | - | 3   | - | mA         | Pxyz=0,IOH level 0   |
|           |   | - | 2   | - | mA         | Pxyz=1,IOH level 1   |
|           |   | - | 1.5 | - | mA         | Pxyz=2,IOH level 2   |
|           |   | - | 0.8 | - | mA         | Pxyz=3,IOH level 3   |
| $R_{PH2}$ | Pull-up resistor  | - | 55  | - | k $\Omega$ |  |

Note:

1. H and L:

- H: the port is four bits higher.
- L: The port is four bits lower.

## 22.5 AC Characteristics

( $V_{DD} = 2.0V \sim 5.5V, T_A = 25^\circ C$ , Unless otherwise indicated)

| Symbol | Parameters | Min Value | Typical Value | Max Value | Unit | Test condition |
|--------|------------|-----------|---------------|-----------|------|----------------|
|--------|------------|-----------|---------------|-----------|------|----------------|

|                    |                                       |       |    |       |     |   |
|--------------------|---------------------------------------|-------|----|-------|-----|---|
| T <sub>Osc</sub>   | External 32k oscillator start-up time | -     | -  | 1     | s   | External 32k crystal oscillator                         |
| T <sub>POR</sub>   | Power On Reset time                   | -     | 15 | -     | ms  |   |
| T <sub>PDW</sub>   | Power Down mode wake-up time          | -     | 65 | 130   | μs  |   |
| T <sub>Reset</sub> | Reset pulse width                     | 18    | -  | -     | μs  | Low level valid   |
| T <sub>LVR</sub>   | LVR buffeting time                    | -     | 30 | -     | μs  |   |
| f <sub>HRC1</sub>  | RC oscillation stability              | 31.68 | 32 | 32.32 | MHz | V <sub>DD</sub> =2.0~5.5V<br>T <sub>A</sub> =-40~85 °C  |
| f <sub>HRC2</sub>  | RC oscillation stability              | 31.36 | 32 | 32.64 | MHz | V <sub>DD</sub> =2.0~5.5V<br>T <sub>A</sub> =-40~105 °C |

## 23.6 ADC Characteristics

(T<sub>A</sub> = 25°C, Unless otherwise indicated)

| Symbol            | Parameters                | Min Value | Typical Value | Max Value | Unit | Condition   |
|-------------------|---------------------------|-----------|---------------|-----------|------|---|
| V <sub>AD1</sub>  | Supply voltage 1          | 2.7       | 5.0           | 5.5       | V    | V <sub>ref</sub> = 2.048V   |
| V <sub>AD2</sub>  | Supply voltage 2          | 2.4       | 5.0           | 5.5       | V    | V <sub>ref</sub> = 1.024V<br>or<br>V <sub>ref</sub> = V <sub>DD</sub> |
| V <sub>AD3</sub>  | Power supply voltage 3    | 2.7       | 5.0           | 5.5       | V    | V <sub>ref</sub> = 2.4V   |
| V <sub>REF1</sub> | Internal reference 2.048V | 2.033     | 2.048         | 2.063     | V    | V <sub>DD</sub> = 2.7~5.5V  |
| V <sub>REF2</sub> | Internal reference 1.024V | 1.004     | 1.024         | 1.044     | V    | V <sub>DD</sub> = 2.0~5.5V  |

|                   |                            |      |      |                 |     |   |
|-------------------|----------------------------|------|------|-----------------|-----|---|
| V <sub>REF3</sub> | Internal reference 2.4V    | 2.37 | 2.40 | 2.45            | V   | V <sub>DD</sub> = 2.7~5.5V                    |
| N <sub>R</sub>    | Precision                  | -    | 12   | -               | bit | GND ≤ V <sub>AIN</sub> ≤ V <sub>DD</sub>      |
| V <sub>AIN</sub>  | ADC input voltage          | GND  | -    | V <sub>DD</sub> | V   |   |
| R <sub>AIN</sub>  | ADC input resistance       | 1    | -    |                 | MΩ  | V <sub>IN</sub> =5V                           |
| I <sub>ADC1</sub> | ADC conversion current 1   | -    | -    | 2               | mA  | ADC module open<br>V <sub>DD</sub> =5V        |
| I <sub>ADC2</sub> | ADC conversion current 2   | -    | -    | 1.8             | mA  | ADC module open<br>V <sub>DD</sub> =3.3V      |
| DNL               | Differential Non-Linearity | -    | -    | ±3              | LSB | V <sub>DD</sub> =5V<br>V <sub>REF</sub> =5V   |
| INL               | Integral Non-Linearity     | -    | -    | ±3              | LSB |   |
| E <sub>Z</sub>    | Offset error               | -    | ±3   | -               | LSB |   |
| E <sub>F</sub>    | Full scale error           | -    | ±1   | -               | LSB |   |
| E <sub>AD</sub>   | Absolute Accuracy          | -    | ±3   | -               | LSB |   |
| T <sub>ADC1</sub> | ADC conversion time 1      | -    | 1.1  | 1.4             | μs  | f <sub>sys</sub> =32 MHz<br>LOWSP[2: 0] = 100 |
| T <sub>ADC2</sub> | ADC conversion time 2      | -    | 1.2  | 1.5             | μs  | f <sub>sys</sub> =32 MHz<br>LOWSP[2: 0] = 101 |
| T <sub>ADC3</sub> | ADC conversion time 3      | -    | 1.5  | 1.9             | μs  | f <sub>sys</sub> =32 MHz<br>LOWSP[2: 0] = 110 |

|            |                       |   |     |     |         |   |
|------------|-----------------------|---|-----|-----|---------|---|
| $T_{ADC4}$ | ADC conversion time 4 | - | 2.0 | 2.6 | $\mu s$ | $f_{sys}=32\text{ MHz}$<br>$LOWSP[2:0]=111$ |
|------------|-----------------------|---|-----|-----|---------|---|

## 22.7 Analog Comparator Characteristics

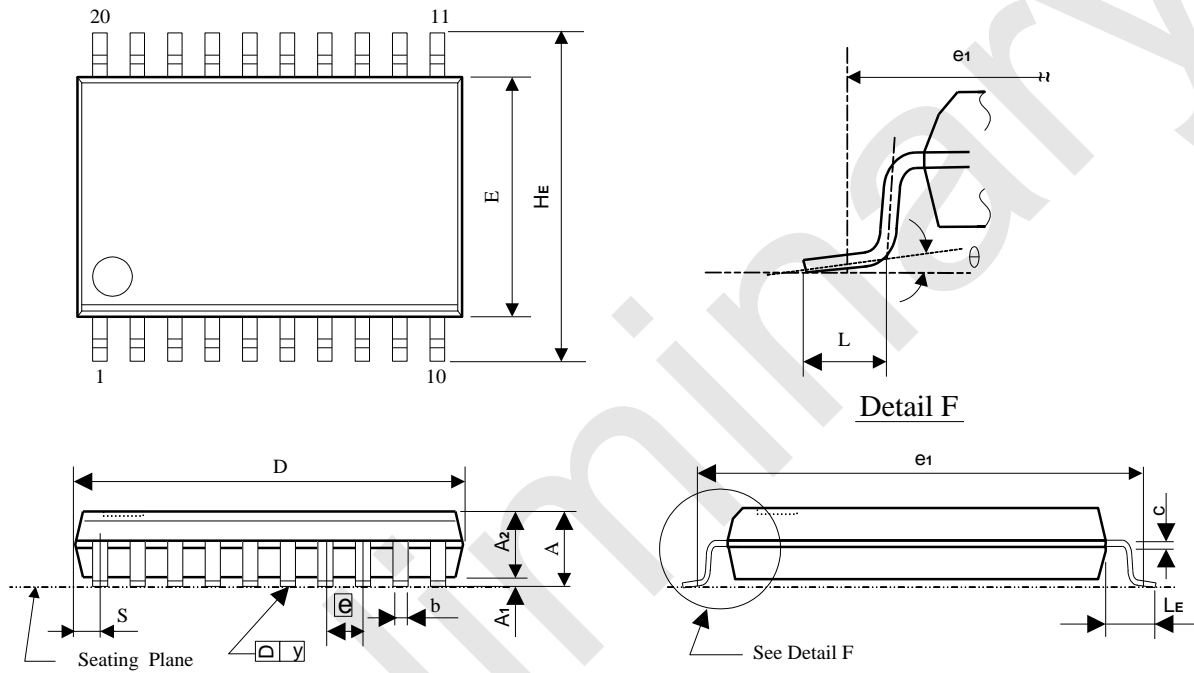
( $V_{DD}=5V, T_A=25^\circ C$ , Unless otherwise indicated)

| Symbol    | Parameters                         | Min Value | Typical Value | Max Value | Unit    | Test condition |
|-----------|------------------------------------|-----------|---------------|-----------|---------|----------------|
| $V_{CM}$  | Input voltage range                | 0         | -             | $V_{DD}$  | V       |                |
| $V_{OS}$  | Comparator offset error            | -         | 10            | 30        | mV      |                |
| $V_{HYS}$ | Schmitt trigger voltage hysteresis | -         | 40            | -         | mV      |                |
| $I_{CMP}$ | Analog comparator consumption      | -         | -             | 100       | $\mu A$ | $V_{DD}=5V$    |
| $T_{CMP}$ | Response time                      | -         | -             | 2         | $\mu s$ |                |

## 23 Package Information

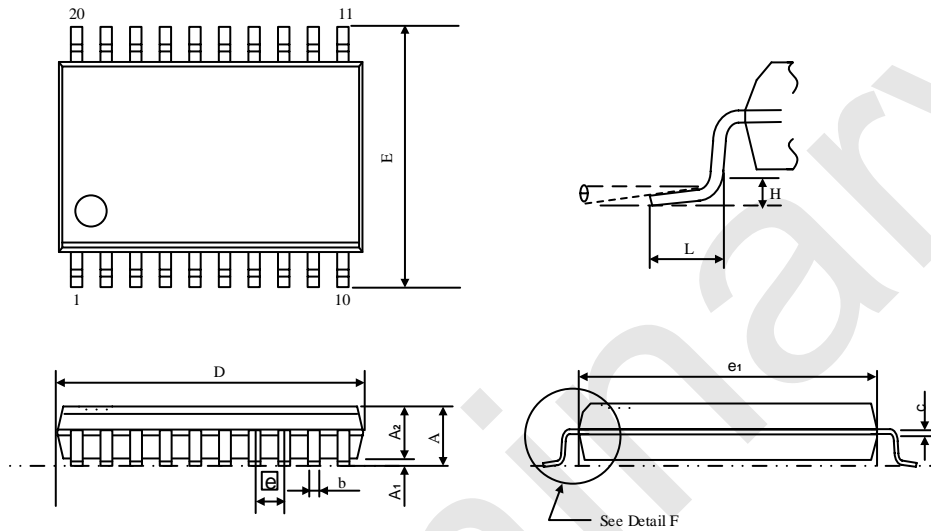
SC95F7612BM20U

SOP20L(300mil) Overall Dimensions Unit: mm

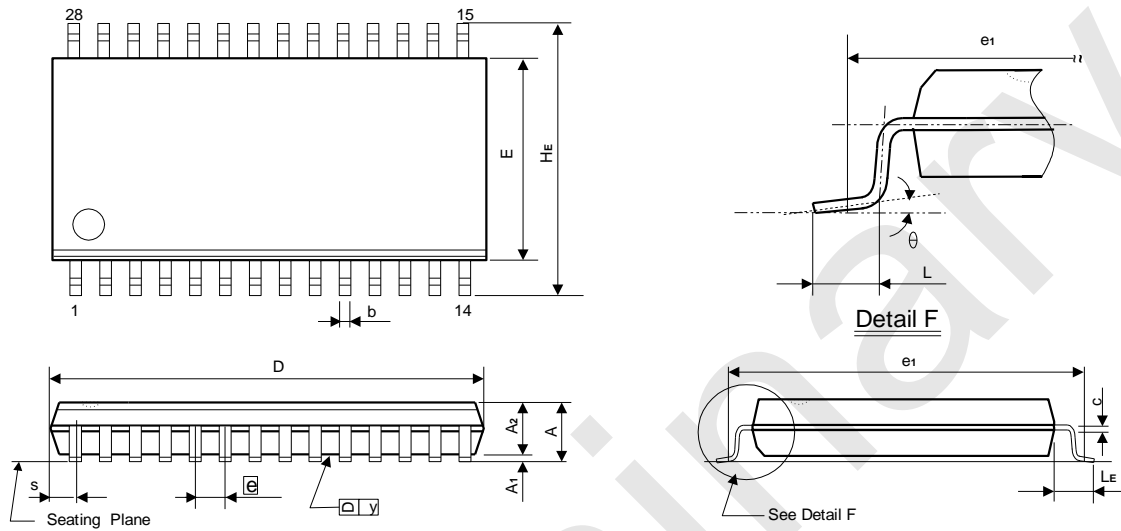


| Symbol | mm        |               |           |
|--------|-----------|---------------|-----------|
|        | Min Value | Typical Value | Max Value |
| A      | 2.40      | 2.56          | 2.65      |
| A1     | 0.100     | 0.200         | 0.300     |
| A2     | 2.240     | 2.340         | 2.440 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

|          |           |       |       |
|----------|-----------|-------|-------|
| e        | 1.27(BSC) |       |       |
| L        | 0.700     | 0.850 | 1.000 |
| LE       | 1.30      | 1.40  | 1.50  |
| $\theta$ | 0°        | -     | 8°    |

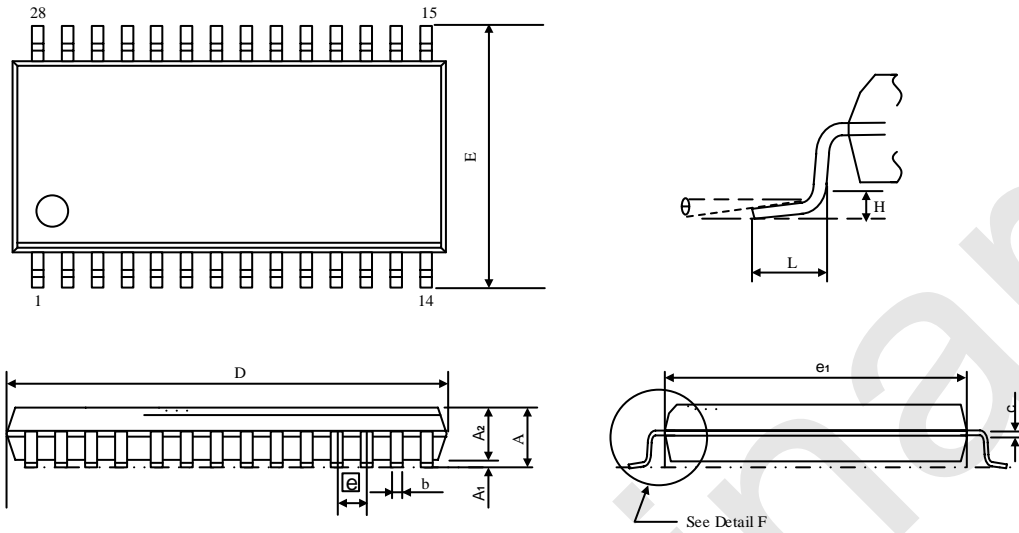
**SC95F7612BX20U**
**TSSOP20L Overall Dimensions Unit: mm**


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

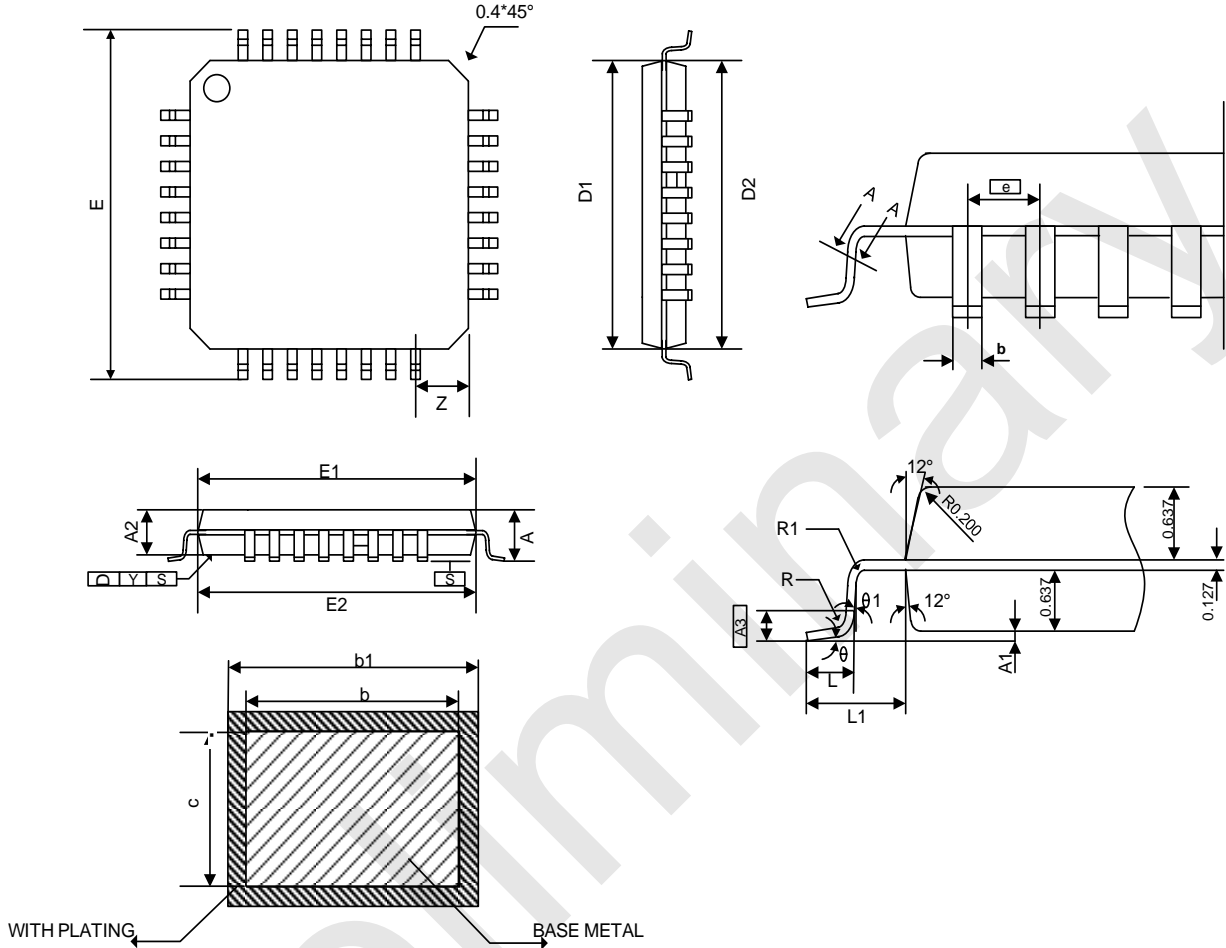
**SC95F7613BM28U**
**SOP28L(300mil) Overall Dimensions Unit: mm**


| Symbol   | mm         |               |           |
|----------|------------|---------------|-----------|
|          | Min Value  | Typical Value | Max Value |
| A        | 2.40       | 2.56          | 2.65      |
| A1       | 0.100      | 0.200         | 0.300     |
| A2       | 2.240      | 2.340         | 2.440     |
| b        | 0.39       | ---           | 0.48      |
| C        | 0.254(BSC) |               |           |
| D        | 17.80      | 18.00         | 18.20     |
| E        | 7.30       | 7.50          | 7.70      |
| HE       | 10.100     | 10.300        | 10.500    |
| e        | 1.270(BSC) |               |           |
| L        | 0.7        | 0.85          | 1.0       |
| LE       | 1.3        | 1.4           | 1.5       |
| $\theta$ | 0°         | -             | 8°        |



**SC95F7613BX28U**
**TSSOP28L Overall Dimensions Unit:mm**


| Symbol    | mm        |               |           |
|-----------|-----------|---------------|-----------|
|           | Min Value | Typical Value | Max Value |
| A         | -         | -             | 1.200     |
| A1        | 0.050     | -             | 0.150     |
| A2        | 0.800     | -             | 1.050     |
| b         | 0.190     | -             | 0.300     |
| c         | 0.090     | -             | 0.200     |
| D         | 9.600     | -             | 9.800     |
| E         | 6.250     | -             | 6.550     |
| e1        | 4.300     | -             | 4.500     |
| $\bar{e}$ | 0.65(BSC) |               |           |
| L         | -         | -             | 1.0       |
| $\theta$  | 0°        | -             | 8°        |
| H         | 0.05      | -             | 0.25      |

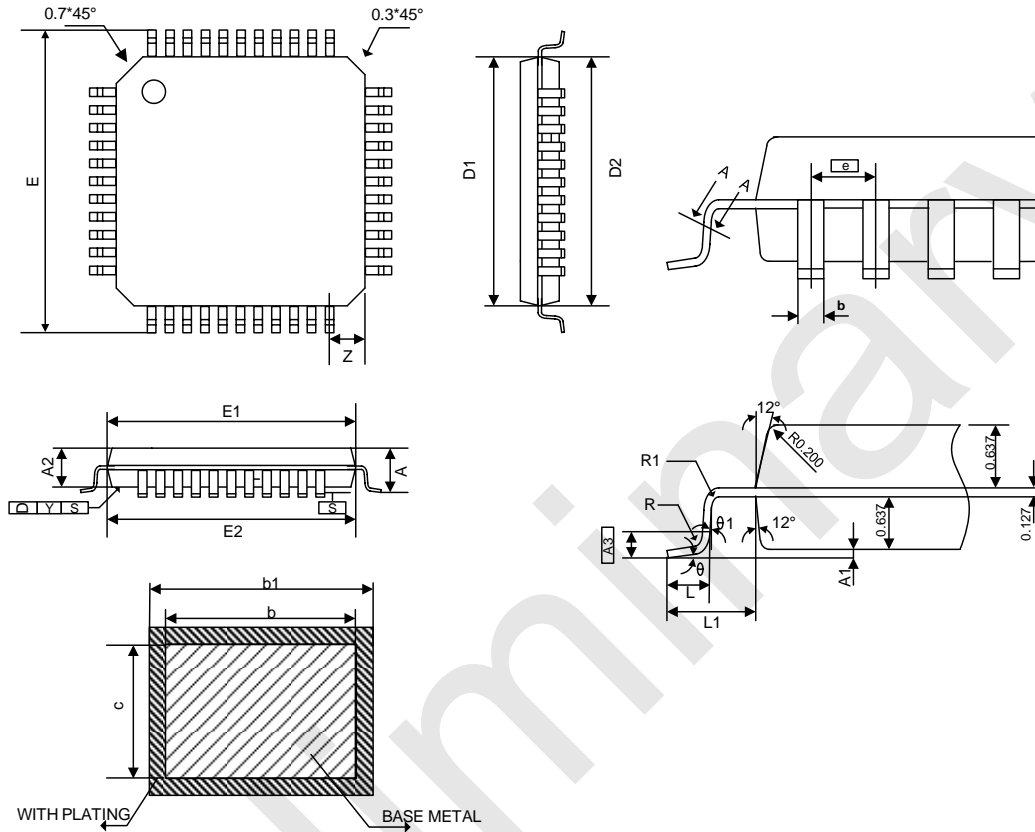
**SC95F7615BP32R**
**LQFP32 (7X7) Overall Dimensions Unit: mm**


| Symbol | mm        |               |           |
|--------|-----------|---------------|-----------|
|        | Min Value | Typical Value | Max Value |
| A      | 1.45      | 1.55          | 1.65      |
| A1     | 0.01      | --            | 0.21      |
| A2     | 1.30      | 1.4           | 1.5       |
| A3     | --        | 0.254         | --        |
| b      | 0.30      | 0.35          | 0.41      |
| b1     | 0.31      | 0.37          | 0.43      |
| c      | 0.12      | 0.13          | 0.14      |

|            |      |      |      |
|------------|------|------|------|
| D1         | 6.85 | 6.95 | 7.05 |
| D2         | 6.90 | 7.00 | 7.10 |
| E          | 8.80 | 9.00 | 9.20 |
| E1         | 6.85 | 6.95 | 7.05 |
| E2         | 6.90 | 7.00 | 7.10 |
| $\bar{e}$  | --   | 0.8  | --   |
| L          | 0.43 | --   | 0.75 |
| L1         | 0.90 | 1.0  | 1.10 |
| R          | 0.1  | --   | 0.25 |
| R1         | 0.1  | --   | --   |
| $\theta$   | 0°   | --   | 10°  |
| $\theta_1$ | 0°   | --   | --   |
| y          | --   | --   | 0.1  |
| Z          | --   | 0.70 | --   |

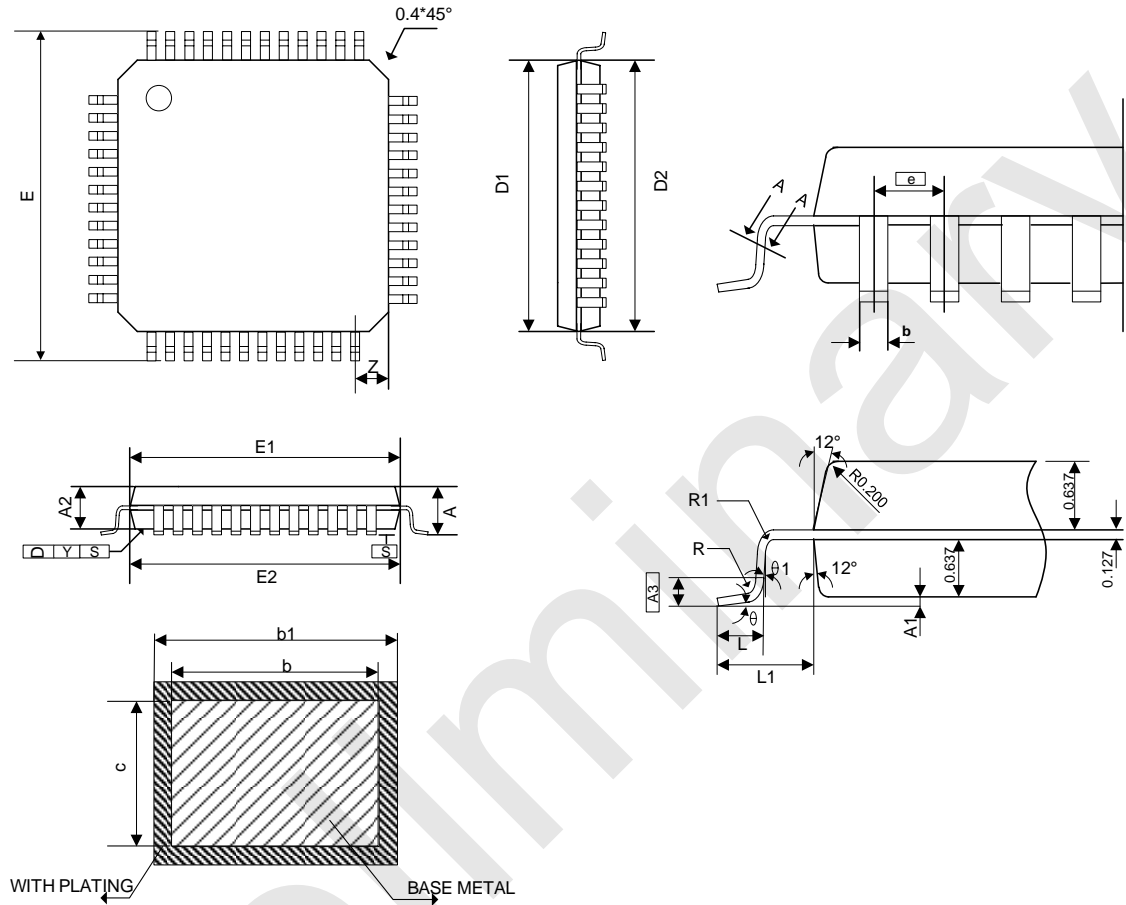
**SC95F7616BP44R**
**LQFP44 (10X10) Overall Dimensions**

Unit: mm



| Symbol | mm        |               |           |
|--------|-----------|---------------|-----------|
|        | Min Value | Typical Value | Max Value |
| A      | 1.45      | 1.55          | 1.65      |
| A1     | 0.015     | --            | 0.21      |
| A2     | 1.3       | 1.4           | 1.5       |
| A3     | --        | 0.254         | --        |
| b      | 0.25      | 0.30          | 0.36      |
| b1     | 0.26      | 0.32          | 0.38      |
| c      | 0.12      | 0.13          | 0.14      |
| D1     | 9.85      | 9.95          | 10.05     |

|            |       |       |       |
|------------|-------|-------|-------|
| D2         | 9.90  | 10.00 | 10.10 |
| E          | 11.80 | 12.00 | 12.20 |
| E1         | 9.85  | 9.95  | 10.05 |
| E2         | 9.90  | 10.00 | 10.10 |
| $e$        | --    | 0.8   | --    |
| L          | 0.42  | --    | 0.75  |
| L1         | 0.95  | 1.0   | 1.15  |
| R          | 0.08  | --    | 0.25  |
| R1         | 0.08  | --    | --    |
| $\theta$   | 0°    | --    | 10°   |
| $\theta_1$ | 0°    | --    | --    |
| y          | --    | --    | 0.1   |
| Z          | --    | 1.0   | --    |

**SC95F7617BP48R**
**LQFP48 (7X7) Overall Dimensions Unit: mm**


| Symbol | mm        |               |           |
|--------|-----------|---------------|-----------|
|        | Min Value | Typical Value | Max Value |
| A      | 1.45      | 1.55          | 1.65      |
| A1     | 0.01      | --            | 0.21      |
| A2     | 1.3       | 1.4           | 1.5       |
| A3     | --        | 0.254         | --        |
| b      | 0.15      | 0.20          | 0.25      |
| b1     | 0.16      | 0.22          | 0.28      |
| c      | 0.12      | --            | 0.17      |

|            |      |      |      |
|------------|------|------|------|
| D1         | 6.85 | 6.95 | 7.05 |
| D2         | 6.90 | 7.00 | 7.10 |
| E          | 8.8  | 9.00 | 9.20 |
| E1         | 6.85 | 6.95 | 7.05 |
| E2         | 6.9  | 7.00 | 7.10 |
| $\bar{e}$  | --   | 0.5  | --   |
| L          | 0.43 | --   | 0.75 |
| L1         | 0.90 | 1.0  | 1.10 |
| R          | 0.1  | --   | 0.25 |
| R1         | 0.1  | --   | --   |
| $\theta$   | 0°   | --   | 10°  |
| $\theta_1$ | 0°   | --   | --   |
| y          | --   | --   | 0.1  |
| Z          | --   | 0.75 | --   |

## 24 Revision History

| Revision | Changes          | Date       |
|----------|------------------|------------|
| V0.1     | Initial Release. | March 2022 |

Preliminary



## Important Notice

Shenzhen SinOne Microelectronics Co., Ltd. (hereinafter referred to as SinOne) reserves the right to change, correct, enhance, modify and improve SinOne products, documents or services at any time without prior notice. Cytron considers the information provided to be accurate and reliable. The information in this document will be used in March 2022. In the actual production design, please refer to the latest data manual of each product and other relevant materials.

Preliminary