# TM52eF1374G/75G

## *DATA SHEET*

## *Rev 0.90*

**(Please read the precautions on the second page before use)**

**tenx** reserves the right to change or discontinue the manual and online documentation to this product herein to improve reliability, function or design without further notice. **tenx** does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others. **tenx** products are not designed, intended, or authorized for use in life support appliances, devices, or systems. If Buyer purchases or uses **tenx** products for any such unintended or unauthorized application, Buyer shall indemnify and hold **tenx** and its officers, employees, subsidiaries, affiliates and distributors harmless against all claims, cost, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use even if such claim alleges that **tenx** was negligent regarding the design or manufacture of the part.

# PRECAUTIONS

1. Before entering Stop/Halt mode (PDOWN), it must be set to slow clock mode (SELFCK = 0).

# AMENDMENT HISTORY

| Version | Date | Description |
|---|---|---|
| V0.90 | Jan, 2023 | New release. |

# CONTENTS

# TM52 F13xx FAMILY

## Common Feature

| CPU | MTP/Flash Program memory | RAM bytes | Dual Clock | Operation Mode | Timer0<br>Timer1<br>Timer2 | UART | Real-time Timer3 | LVD | LVR |
|---|---|---|---|---|---|---|---|---|---|
| Fast 8051 (2T) | 4K~32K with IAP, ISP, ICP | 256<br>~<br>1024 | SXT<br>SRC<br>FXT<br>FRC | Fast<br>Slow<br>Idle<br>Stop<br>Halt | 8051 Standard | | 15-bit | 8 level | 8 level |

***Note:*** *IAP, ISP only for Flash type program memory*

## Family Members Features

| P/N | Program Memory | RAM Bytes | IO Pin | PWM | SAR ADC | Touch Key | LCD | LED | Interface |
|---|---|---|---|---|---|---|---|---|---|
| **TM52-eF1716**<br>**TM52-eF1732** | Flash<br>16KB<br>32KB | 1280 | 30 | 16-bit x3<br>8-bit x3 | 12-bit<br>16-ch | 20-ch | 8com | BiD 4Cx6S | SPI<br>UARTx2<br>I$^2$C |
| **TM52-eF1374G**<br>**TM52-eF1375G** | Flash<br>20KB<br>32KB | 1280 | 26 | 16-bit x3 | 12-bit<br>16-ch | 20-ch | 8com | BiD 4Cx6S<br>DMX 8x8 | SPI<br>UARTx2<br>I$^2$C |

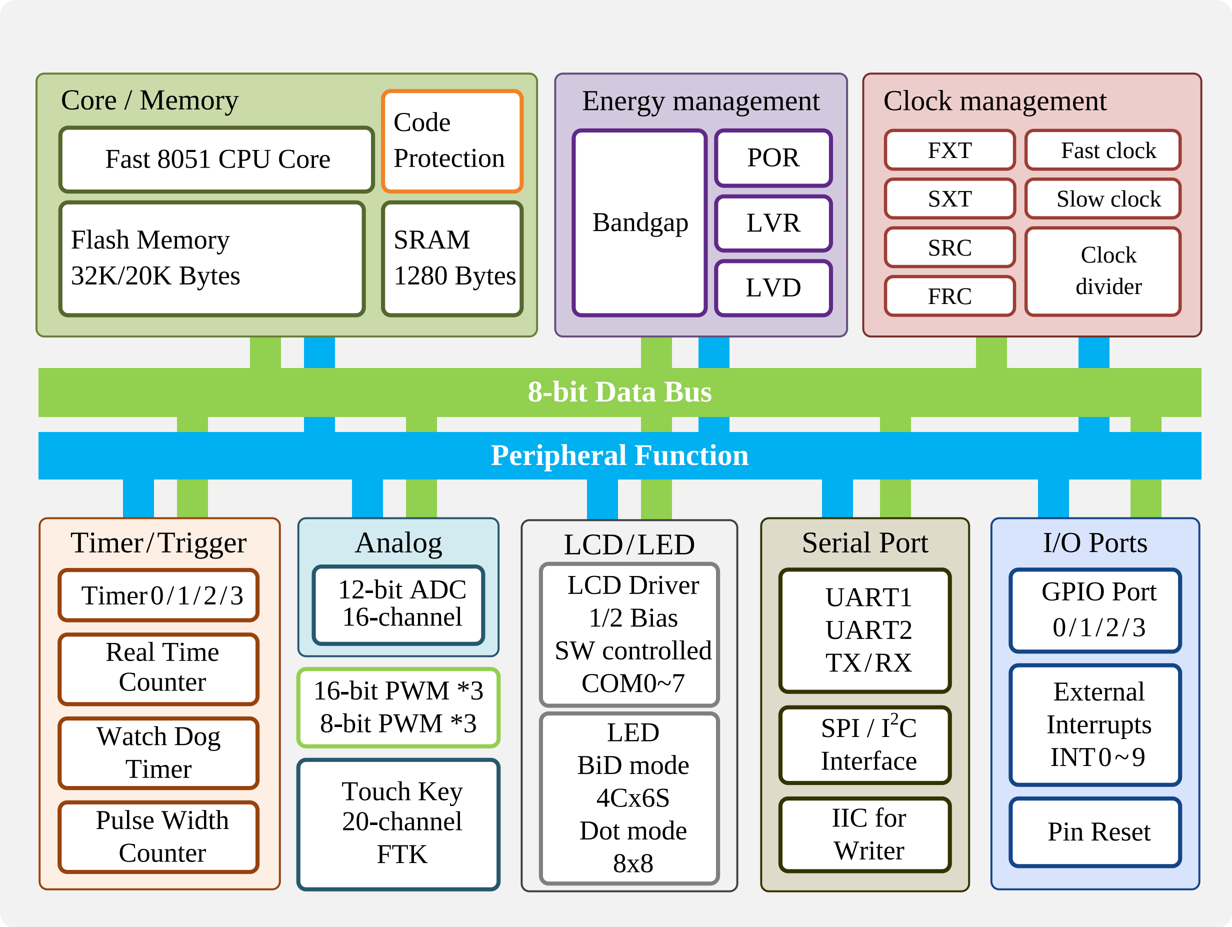
| P/N | Operation Voltage | Operation Current | | | | | Max. System Clock (Hz) | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Fast FRC | Slow SRC | Idle SRC | Stop | Halt | SXT | SRC | FXT | FRC |
| **TM52-eF1716**<br>**TM52-eF1732** | 2.5~5.5V | 3.5mA | 0.18mA | 0.15 mA | 7uA@5V<br>1.4uA@3V | 11uA@5V<br>4uA@3V | 32K | 80K | 16M | 14.7456M |
| **TM52-eF1374G**<br>**TM52-eF1375G** | 2.2~5.5V | 4mA | 0.22mA | 0.2mA | 10uA@5V<br>4uA@3V | 13uA@5V<br>6uA@3V | 32K | 80K | 18M | 18.432M |

# GENERAL DESCRIPTION

**TM52 $_{\text{series}}$ eF1374G/75G** are versions of a new, fast 8051 architecture for an 8-bit microcontroller single chip with an instruction set fully compatible with industry standard 8051, and retains most 8051 peripheral's functional block. Typically, the **TM52** executes instructions six times faster than the standard 8051 architecture.

The **TM52-eF1374G/75G** provides improved performance, lower cost and fast time-to-market by integrating features on the chip, including 32K Bytes Flash program memory, 1280 Bytes SRAM, Low Voltage Reset (LVR), Low Voltage Detector (LVD), dual clock power saving operation mode, 8051 standard UART and Timer0/1/2, real time clock Timer3, LCD/LED driver, 3 set 16-bit PWMs, 16 channels 12-bit A/D Convertor, 20 channels Touch Key, I$^2$C/SPI interface and Watch Dog Timer. It's a high reliability and low power consumption feature can be widely applied in consumer and home appliance products.

# BLOCK DIAGRAM



# FEATURES

1. **Standard 8051 Instruction set, fast machine cycle**
   - Executes instructions six times faster than the standard 8051.
2. **Flash Program Memory**
   - 32K Bytes (TM52eF1375G)
   - 20K Bytes (TM52eF1374G)
   - Support IAP “In Application Programming” (EEPROM like)
   - Code Protection Capability
   - 100K erase times at least
   - 10 years data retention at least
3. **Total 1280 Bytes SRAM (IRAM + XRAM)**
   - 256 Bytes IRAM in the 8051 internal data memory area
   - 1024 Bytes XRAM in the 8051 external data memory area (accessed by MOVX Instruction)
4. **Four System Clock type selections**
   - Fast clock from 1~18MHz Crystal (FXT)
   - Fast clock from Internal RC (FRC, 18.432 MHz)
   - Slow clock from 32768Hz Crystal (SXT)
   - Slow clock from Internal RC (SRC,80 KHz)
   - System Clock can be divided by 1/2/4/16 option
5. **8051 Standard Timer – Timer0/1/2**
   - 16-bit Timer0, also supports T0O clock output for Buzzer application
   - 16-bit Timer1
   - 16-bit Timer2, also supports T2O clock output for Buzzer application
6. **15-bit Timer3**
   - Clock source is Slow clock
   - Interrupt period can be clock divided by 32768/16384/8192/128 option
7. **UARTs**
   - UART1, 8051 standard UART
   - UART2, the second UART, supports only mode1 and mode3

**8. Three independent 16 bits PWMs with period-adjustment**

**9. SPI Interface**

- Master or Slave mode selectable
- Programmable transmit bit rate
- Serial clock phase and polarity options
- MSB-first or LSB-first selectable

**10. I$^2$C interface (Master / Slave)**

**11. 20-Channel Touch Key (FTK)**

**12. 12-bit ADC with 13 channels External Pin Input and 2 channels Internal Reference Voltage**

- Internal Reference Voltage:

  VBG 1.27V @$V_{CC}$=5V~3V, 25℃
- Internal Reference Voltage: 1/4$V_{CC}$

**13. LCD Driver**

- 1/8 duty
- Software controlled COM0~7
- 1/2 LCD Bias

**14. LED Controller/Driver**

- Bidirection matrix mode (BiD) : 4Cx6S, 10 pins up to 48 dots
- Dot matrix mode: 8*8, 9 pins up to 64 dots

**15. 14 Sources, 4-level priority Interrupt**

- Timer0/Timer1/Timer2/Timer3 Interrupt
- INT0~INT1 pin low level or falling edge Interrupt
- INT2~INT9 pin Falling-Edge Interrupt
- Port1 Pin Change Interrupt
- UART1/UART2 TX/RX Interrupt
- ADC/Touch Key Interrupt
- SPI Interrupt
- I$^2$C interrupt
- PWM0/PWM1/PWM2 interrupt

### 16. Pin Interrupt can Wake up CPU from Power-Down (Stop/Halt) mode

- INT0~INT9 Interrupt & Wake-up
- Each Port1 pin can be defined as Interrupt & Wake-up pin (by pin change)

***Note:*** Chip cannot enter Stop/Halt Mode if INTn pin is low and wakeup is enabled. (INTn=0 and EXn=1, n=0~9)

### 17. Max. 26 Programmable I/O pins

- CMOS Output
- Pseudo-Open-Drain, or Open-Drain Output
- Schmitt Trigger Input
- Pin Pull-up can be Enabled or Disabled
- All pin with High sink (60mA@$V_{CC}$=5V，$V_{OL}$=0.1$V_{CC}$)

### 18. Independent RC Oscillating Watch Dog Timer

- 400ms/200ms/100ms/50ms selectable WDT timeout options

### 19. Five types Reset

- Power on Reset
- Selectable External Pin Reset
- Selectable Watch Dog Reset
- Software Command Reset
- Selectable Low Voltage Reset

### 20. 8-level Low Voltage Detect

- 2.3V/2.54V/2.78V/3.04V/3.28V/3.54V/3.8V/4.04V

### 21. 8-level Low Voltage Reset

- 2.3V/2.54V/2.78V/3.04V/3.28V/3.54V/3.8V/4.04V

### 22. Five Power Operation Modes

- Fast/Slow/Idle/Halt /Stop mode

## 23. Integrated 16-bit Cyclic Redundancy Check function

## 24. Multiplication and division

- 8 bit Multiplier & Divider (standard 8051)
- 16 bits Multiplier & Divider
- 32 bits ÷ 16 bits hardware Divider

## 25. On-chip Debug/ICE interface

- Use P3.0/P3.1 pin or P0.0/P0.1 pin
- Share with ICP programming pin

## 26. Writer interface

- Use P3.0/P3.1

## 27. Operating Voltage and Current

- $V_{CC}$ =2.3V ~ 5.5V @$F_{SYSCLK}$=18.432MHz
- $I_{CC}$ =7μA @Stop mode, $V_{CC}$=5V
- $I_{CC}$ =1.4μA @Stop mode, $V_{CC}$=3V
- $I_{CC}$ =150μA @Idle mode, $V_{CC}$=5V

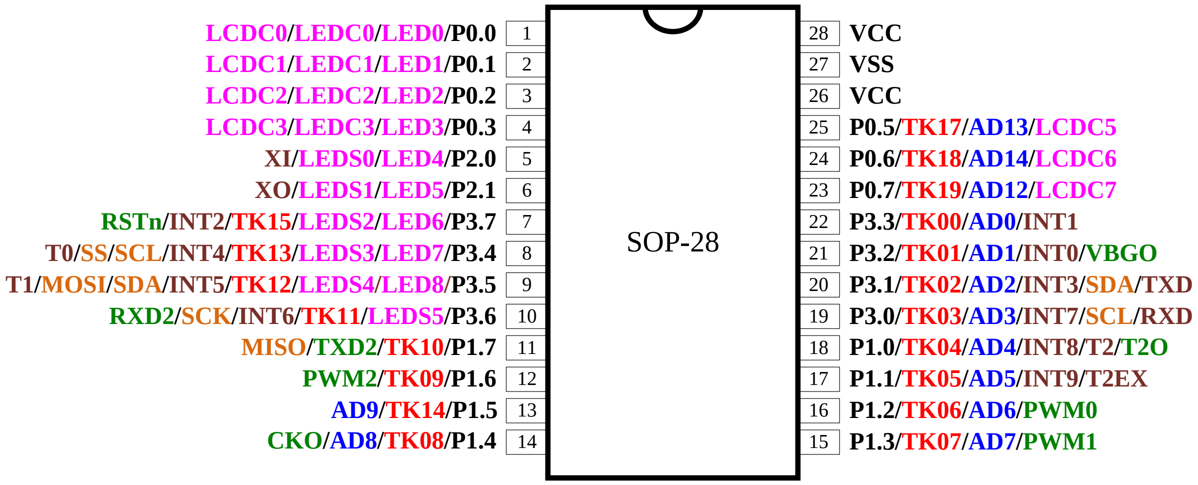
## 28. Operating Temperature Range

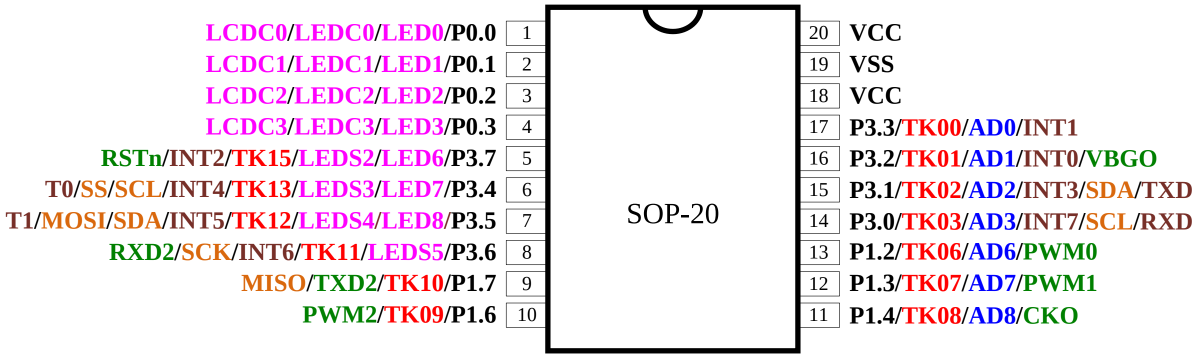
- –40℃ ~ +85℃

## 29. Package Types

- 28-pin SOP28 (300 mil)
- 20-pin SOP28 (300 mil)

# PIN ASSIGNMENT





# PIN DESCRIPTION

| Name | In/Out | Pin Description |
|---|---|---|
| P0.0~P0.7 | I/O | Bit-programmable I/O port for Schmitt-trigger input or CMOS push-pull output. Pull-up resistors are assignable by software. |
| P1.0~P1.7 | I/O | Bit-programmable I/O port for Schmitt-trigger input, CMOS push-pull output or "open-drain" output. Pull-up resistors are assignable by software. These pin's level change can interrupt/wake up CPU from Idle/Stop/Halt mode. |
| P2.0~P2.1 | I/O | Bit-programmable I/O port for Schmitt-trigger input, CMOS push-pull output or "open-drain" output. Pull-up resistors are assignable by software. |
| P3.0~P3.2 | I/O | Bit-programmable I/O port for Schmitt-trigger input, CMOS push-pull output or "**pseudo open drain**" output. Pull-up resistors are assignable by software. |
| P3.3~P3.7 | I/O | Bit-programmable I/O port for Schmitt-trigger input, CMOS push-pull output or "open-drain" output. Pull-up resistors are assignable by software. |
| INT0, INT1 | I | External low level or falling edge Interrupt input, Idle/Halt/Stop mode wake up input. |
| INT2~9 | I | External falling edge Interrupt input, Idle/Halt /Stop mode wake up input. |
| RXD | I/O | UART1 Mode0 transmit & receive data, Mode1/2/3 receive data |
| RXD2 | I/O | UART2 Mode1/3 receive data |
| TXD | I/O | UART1 Mode0 transmit clock, Mode1/2/3 transmit data. In One Wire UART mode, this pin transmits and receives serial data. |
| TXD2 | I/O | UART2 Mode1/3 transmit data. |
| T0, T1, T2 | I | Timer0, Timer1, Timer2 event count pin input. |
| T2EX | I | Timer2 external trigger input. |
| T0O | O | Timer0 overflow divided by 64 output |
| T2O | O | Timer2 overflow divided by 2 output |
| CKO | O | System Clock divided by 2 output |
| VBGO | O | Bandgap voltage output |
| PWM0~PWM2 | O | 16 bit PWM output |
| AD0~AD9<br>AD12~AD14 | I | ADC input |
| TK00~TK19 | I | Touch Key input |
| CLD | I | Touch Key charge collection capacitor connection pin |
| LCDC0~LCDC7 | O | LCD 1/2 bias output |
| LEDC0~LEDC3 | O | LED BiD matrix mode common output |
| LEDS0~LEDS5 | O | LED BiD matrix mode segment output |
| LED0~LED8 | O | LED Dot matrix mode output |
| MISO | I/O | SPI data input for master mode, data output for slave mode |
| MOSI | I/O | SPI data output for master mode, data input for slave mode |
| SS | I | SPI active low slave select input for slave mode |
| SCK | I/O | SPI clock output for master or clock input for slave mode |
| SCL | I/O | $I^2C$ SCL |
| SDA | I/O | $I^2C$ SDA |
| RSTn | I | External active low reset input, Pull-up resistor is fixed enable. |
| XI, XO | – | Crystal/Resonator oscillator connection for System clock (FXT or SXT) |
| VCC, VSS | P | Power input pin and ground |

## PIN SUMMERY

| Pin # | Pin Name | Type | Initial State | Input | | | Output | | | Alternative Function | | | | | | | | | | MISC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SOP-28 | | | | Pull-up Control | Wake up | Ext. Interrupt | CMOS P.P. | P.O.D. | O.D. | LCD | LED BiD matrix | LED Dot matrix | ADC | Touch Key | UART | PWM | Timer | SPI | $I^2C$ | |
| 1 | P0.0 | I/O | Hi-Z | ◉ | | | ● | | | ● | ● | ● | | | | | | | | |
| 2 | P0.1 | I/O | Hi-Z | ◉ | | | ● | | | ● | ● | ● | | | | | | | | |
| 3 | P0.2 | I/O | Hi-Z | ◉ | | | ● | | | ● | ● | ● | | | | | | | | |
| 4 | P0.3 | I/O | Hi-Z | ◉ | | | ● | | | ● | ● | ● | | | | | | | | CLD |
| 5 | P2.0 | I/O | Hi-Z | ○ | | | ● | | ● | ● | ● | ● | | | | | | | | Crystal |
| 6 | P2.1 | I/O | Hi-Z | ○ | | | ● | | ● | ● | ● | ● | | | | | | | | Crystal |
| 7 | P3.7 | I/O | PU | ○ | ● | ● | ● | | ● | | ● | ● | | ● | | | | | | Reset |
| 8 | P3.4 | I/O | Hi-Z | ○ | ● | ● | ● | | ● | | ● | ● | | ● | | | ● | ● | ● | |
| 9 | P3.5 | I/O | Hi-Z | ○ | ● | ● | ● | | ● | | ● | ● | | ● | | | ● | ● | ● | |
| 10 | P3.6 | I/O | Hi-Z | ○ | ● | ● | ● | | ● | | ● | | | ● | ● | | | ● | | |
| 11 | P1.7 | I/O | Hi-Z | ○ | ● | | ● | | ● | | | | | ● | ● | | | ● | | |
| 12 | P1.6 | I/O | Hi-Z | ○ | ● | | ● | | ● | | | | | ● | | ● | | | | |
| 13 | P1.5 | I/O | Hi-Z | ○ | ● | | ● | | ● | | | | ● | ● | | | | | | |
| 14 | P1.4 | I/O | Hi-Z | ○ | ● | | ● | | ● | | | | ● | ● | | | | | | CKO |
| 15 | P1.3 | I/O | Hi-Z | ○ | ● | | ● | | ● | | | | ● | ● | | ● | | | | |
| 16 | P1.2 | I/O | Hi-Z | ○ | ● | | ● | | ● | | | | ● | ● | | ● | | | | |
| 17 | P1.1 | I/O | Hi-Z | ○ | ● | ● | ● | | ● | | | | ● | ● | | | ● | | | |
| 18 | P1.0 | I/O | Hi-Z | ○ | ● | ● | ● | | ● | | | | ● | ● | | | ● | | | T2O |
| 19 | P3.0 | I/O | Hi-Z | ○ | ● | ● | ● | ● | | | | | ● | ● | ● | | | | ● | |
| 20 | P3.1 | I/O | Hi-Z | ○ | ● | ● | ● | ● | | | | | ● | ● | ● | | | | ● | |
| 21 | P3.2 | I/O | Hi-Z | ○ | ● | ● | ● | ● | | | | | ● | ● | | | | | | VBGO |
| 22 | P3.3 | I/O | Hi-Z | ○ | ● | ● | ● | | ● | | | | ● | ● | | | | | | |
| 23 | P0.7 | I/O | Hi-Z | ◉ | | | ● | | | ● | | | ● | ● | | | | | | |
| 24 | P0.6 | I/O | Hi-Z | ◉ | | | ● | | | ● | | | ● | ● | | | | | | |
| 25 | P0.5 | I/O | Hi-Z | ◉ | | | ● | | | ● | | | ● | ● | | | | | | |
| 26 | P0.4 | I/O | Hi-Z | ◉ | | | ● | | | ● | | | | ● | | | | | | |
| 27 | VSS | P | | | | | | | | | | | | | | | | | | |
| 28 | VCC | P | | | | | | | | | | | | | | | | | | |

Symbol:

P.P.: Push-Pull    O.D: Open Drain    P.O.D: Pseudo Open Drain    PU: Pull up

PS:

1. ○ Port1, Port2, Port3 these pins control Pull up resistor by operation modes
2. ◉ Port0, control Pull up resistor while PxOE.n=0 and Px.n=1

# FUNCTIONAL DESCRIPTION

## 1. CPU Core

In the 8051 architecture, the C programming language is used as a development platform. The TM52 device features a fast 8051 core in a highly integrated microcontroller, allowing designers to be able to achieve improved performance compared to a classic 8051 device. TM52 series microcontrollers provide a complete binary code with standard 8051 instruction set compatibility, ensuring an easy migration path to accelerate the development speed of system products. The CPU core includes an ALU, a program status word (PSW), an accumulator (ACC), a B register, a stack point (SP), DPTRs, a program counter, an instruction decoder, and core special function registers (SFRs).

### 1.1 Accumulator (ACC)

This register provides one of the operands for most ALU operations. Accumulators are generally referred to as A or Acc and sometimes referred to as Register A. In this document, the accumulator is represented as "A" or "ACC" including the instruction table. The accumulator, as its name suggests, is used as a general register to accumulate the intermediate results of a large number of instructions. The accumulator is the most important and frequently used register to complete arithmetic and logical operations. It holds the intermediate results of most arithmetic and logic operations and assists in data transportation.

| SFR **E0h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **ACC** | ACC.7 | ACC.6 | ACC.5 | ACC.4 | ACC.3 | ACC.2 | ACC.1 | ACC.0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

E0h.7~0 **ACC:** Accumulator

### 1.2 B Register (B)

The "B" register is very similar to the ACC and may hold a 1 Byte value. This register provides the second operand for multiply or divide instructions. Otherwise, it may be used as a scratch pad register. The B register is only used by two 8051 instructions, MUL and DIV. When A is to be multiplied or divided by another number, the other number is stored in B. For MUL and DIV instructions, it is necessary that the two operands are in A and B.

ex: DIV AB

When this instruction is executed, data inside A and B are divided, and the answer is stored in A.

| SFR **F0h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **B** | B.7 | B.6 | B.5 | B.4 | B.3 | B.2 | B.1 | B.0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

F0h.7~0 **B:** B register

## 1.3 Stack Pointer (SP)

The SP register contains the Stack Pointer. The Stack Pointer is used to load the program counter into memory during LCALL and ACALL instructions and is used to retrieve the program counter from memory in RET and RETI instructions. The stack may also be saved or loaded using PUSH and POP instructions, which also increment and decrement the Stack Pointer.

| SFR **81h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **SP** | SP | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

81h.7~0 **SP:** Stack Point

## 1.4 Dual Data Pointer (DPTRs)

TM52 device has two DPTRs, which share the same SFR address. Each DPTR is 16 bits in size and consists of two registers: the DPTR high byte (DPH) and the DPTR low byte (DPL). The DPTR is used for 16-bit-address external memory accesses, for offset code byte fetches, and for offset program jumps. Setting the DPSEL control bit allows the program code to switch between the two physical DPTRs.

| SFR **82h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **DPL** | DPL | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

82h.7~0 **DPL:** Data Point low byte

| SFR **83h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **DPH** | DPH | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

83h.7~0 **DPH:** Data Point high byte

| SFR **F8h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **AUX1** | CLRWDT | CLRTM3 | TKSOC | ADSOC | LVRPD | T2SEL | T1SEL | DPSEL |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

F8h.0 **DPSEL:** Active DPTR Select

## 1.5 Program Status Word (PSW)

This register contains status information resulting from CPU and ALU operations. The instructions that affect the PSW are listed below.

| Instruction | Flag | | | Instruction | Flag | | |
|---|---|---|---|---|---|---|---|
| | C | OV | AC | | C | OV | AC |
| ADD | X | X | X | CLR C | 0 | | |
| ADDC | X | X | X | CPL C | X | | |
| SUBB | X | X | X | ANL C, bit | X | | |
| MUL | 0 | X | | ANL C, /bit | X | | |
| DIV | 0 | X | | ORL C, bit | X | | |
| DA | X | | | ORL C, /bit | X | | |
| RRC | X | | | MOV C, bit | X | | |
| RLC | X | | | CJNE | X | | |
| SETB C | 1 | | | | | | |

A "0" means the flag is always cleared, a "1" means the flag is always set and an "X" means that the state of the flag depends on the result of the operation.

| SFR **D0h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PSW** | 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 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

D0h.7 **CY:** ALU carry flag

D0h.6 **AC:** ALU auxiliary carry flag

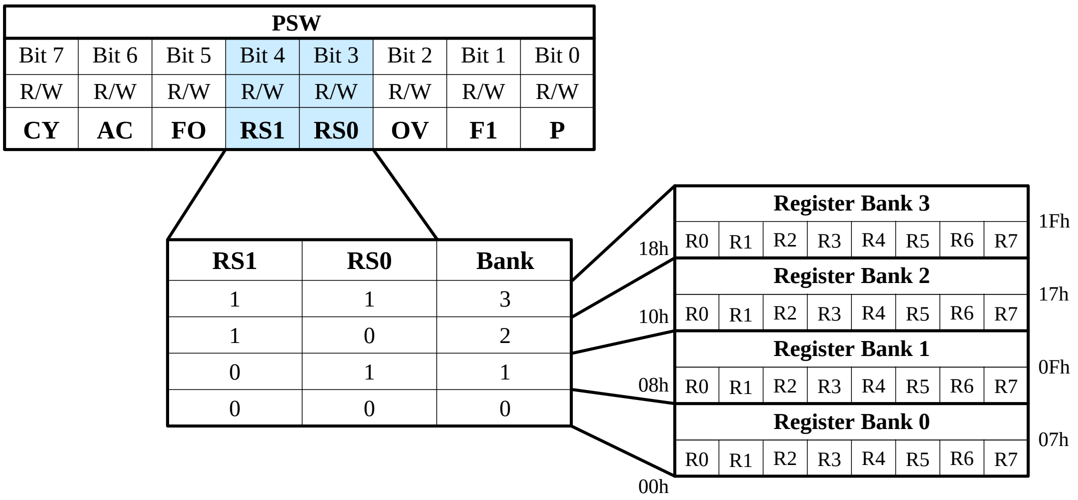
D0h.5 **F0:** General purpose user-definable flag

D0h.4~3 **RS1, RS0:** The contents of (RS1, RS0) enable the working register banks as:
00: Bank 0 (00h~07h)
01: Bank 1 (08h~0Fh)
10: Bank 2 (10h~17h)
11: Bank 3 (18h~1Fh)

D0h.2 **OV:** ALU overflow flag

D0h.1 **F1:** General purpose user-definable flag

D0h.0 **P:** Parity flag. Set/cleared by hardware each instruction cycle to indicate odd/even number of "one" bits in the accumulator.



| PSW | | | | | | | |
|---|---|---|---|---|---|---|---|
| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| **CY** | **AC** | **FO** | **RS1** | **RS0** | **OV** | **F1** | **P** |

| **RS1** | **RS0** | **Bank** |
|---|---|---|
| 1 | 1 | 3 |
| 1 | 0 | 2 |
| 0 | 1 | 1 |
| 0 | 0 | 0 |

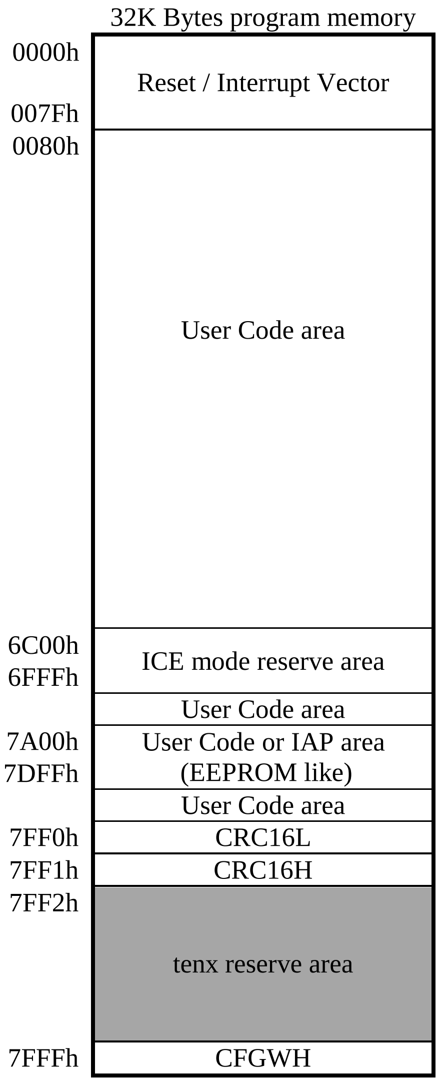
# 2. Memory

## 2.1 Program Memory

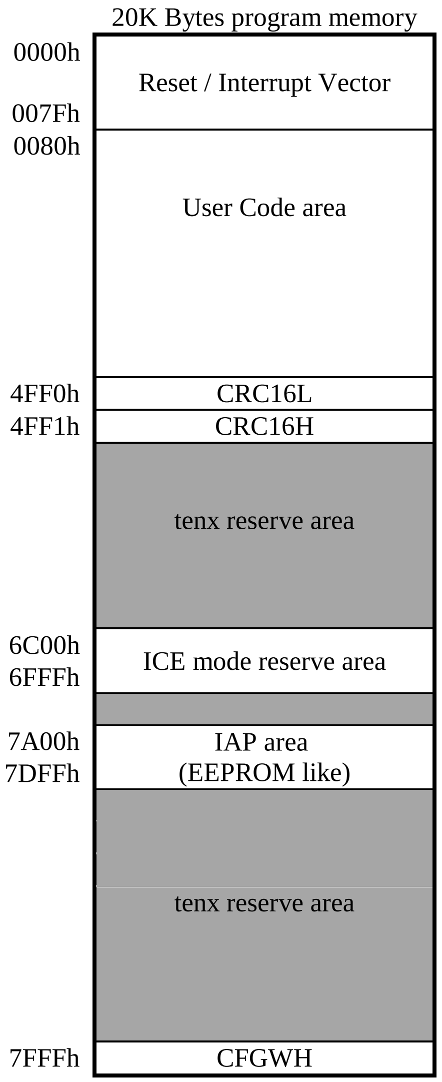
The Chip has a 32K Bytes Flash program memory for **TM52eF1374G/75G** which can support In Application Programming (IAP) function modes. The Flash write endurance is at least 100K cycles. The program memory address continuous space (0000h~7FFFh) is partitioned to several sectors for device operation.

### 2.1.1 Program Memory Functional Partition

The last bytes (7FFFh) of program memory is defined as chip Configuration Word (CFGW), which is loaded into the device control registers upon power on reset (POR). The 0000h~007Fh is occupied by Reset/Interrupt vectors as standard 8051 definition. For **TM52eF1374G/75G**, the address space 7A00h~7DFFh is defined as the IAP area. In the in-circuit emulation (ICE) mode, user also needs to reserve the address space 6C00h~6FFFh for ICE System communication.CRC16H/L is the reserved area of the checksum. Tenx can provide a CRC verification subroutine. The user can calculate the checksum by the CRC verification subroutine to compare with CRC16H/L and check the validity of the ROM code.



**TM52eF1375G**



**TM52eF1374G**

### 2.1.2 Flash ICP Mode

The Flash memory can be programmed by the tenx proprietary writer **(TWR writer)**, which needs at least four wires (VCC, VSS, P3.0 and P3.1) to connect to this chip. If user wants to program the Flash memory on the target circuit board (In Circuit Program, ICP), these pins must be reserved sufficient freedom to be connected to the Writer.

| Writer wire number | Pin connection |
|---|---|
| 4-Wire | VCC, VSS, P3.0, P3.1 |

### 2.1.3 Flash IAP Mode (EEPROM like)

The **eF1374G/75G** has “In Application Program” (IAP) capability, which allows software to read/write data from/to the Flash memory during CPU run time.

There are two pages (7A00h~7BFFh and 7C00h~7DFF) can be IAP write and erase. When using IAP to write, you need to erase first and then write bytes. After erasing, each address can only be written once

IAP erase operation will erase 512 bytes at a time from 7A00h~7BFFh or 7C00h~7DFF. When writing any value in address 7B2Dh, 512 bytes of 7A00h~7BFFh can be erased. Similarly, when writing any value in address 7D69h, 512 bytes of 7C00h~7DFFh can be erased.

Before IAP writing or erasing, there are two SFR, IAPWE and SWCMD, should be set as flowing table. After IAP writing or erasing, IAPWE and SWCMD should be cleared immediately.

Through the "MOVX @DPTR, A" instruction, IAP can be written and erased simply and
IAP reading can be done easily by "MOVC" instruction.

| SFR Setting | IAP Write | IAP page Erase<br>(Erase 512 bytes) | IAP Disable |
|---|---|---|---|
| Address 7A00h ~ 7BFFh | SWCMD = 65h<br>IAPWE = 4Ah | SWCMD = 65h<br>IAPWE = BAh | SWCMD = 0h<br>IAPWE = 0h |
| Address 7C00h ~ 7DFFh | SWCMD = 65h<br>IAPWE = 4Ch | SWCMD = 65h<br>IAPWE = BCh | SWCMD = 0h<br>IAPWE = 0h |

| Address | Byte Write | Page Erase |
|---|---|---|
| 0000h<br>~<br>79FFh | N | N |
| 7A00h<br>~<br>7BFFh | Y<br>Byte write | Y<br>Page Erase |
| 7C00h<br>~<br>7DFFh | Y<br>Byte write | Y<br>Page Erase |
| 7E00h<br>~<br>7FFFh | N | N |

### 2.1.4 IAP Mode Access Routines

**Flash IAP Write** is simply achieved by a “MOVX @DPTR, A” instruction while the DPTR contains the target Flash address from 7A00h to 7DFEh, and the ACC contains the data being written. The eF1374G/75G accepts IAP write commands only when IAPWE and SWCMD are set to appropriate values. Flash IAP writing one byte requires approximately 20 us and erasing one page requires approximately 2ms. While IAP writing or erasing the CPU stays in a waiting state, but all peripheral modules (Timers, LED, and others) continue running during the writing/erase time. The software must handle the pending interrupts after an IAP write. The **eF1374G/75G** has a build-in IAP Time-out function for escaping write fail state. Flash IAP writing needs higher $V_{CC}$ voltage, $V_{CC}$>2.5V.

Before IAP Write, the user should disable the LVR first.

How to erase page 7A00h~7BFFh

(1) Set the DPTR to 7B2Dh
(2) Set the SWCMD to 65h
(3) Set the IAPWE to BAh
(4) MOVX @DPTR, A (write any data to 7B2Dh to erase 7A00h~7BFFh)

```
  ; IAP example code
  ; need 2.5V < VCC < 5.5V
SETB    LVRPD                 ; Disable LVR
MOV     DPTR, #7B2Dh          ; DPTR=7B2Dh=target IAP address
MOV     SWCMD, #65h           ; IAP write enable
MOV     IAPWE, #BAh           ; IAP 7A00h~7BFFh erase enable
MOVX    @DPTR, A              ; write any data to 7B2Dh to erase 7A00h~7BFFh
                              ; 7A00h~7BFFh convert to ‘1’ after IAP erase
                              ; 2ms H/W writing time, CPU wait
MOV     IAPWE, #00h           ; IAP write disable, immediately after IAP write
CLR     LVRPD                 ; Enable LVR
```

How to erase page 7C00h~7DFFh

(1) Set the DPTR to 7D69h
(2) Set the SWCMD to 65h
(3) Set the IAPWE to BCh
(4) MOVX @DPTR, A (write any data to 7D69h to erase 7C00h~7DFFh)

```
  ; IAP example code
  ; need 2.5V < VCC < 5.5V
SETB    LVRPD                 ; Disable LVR
MOV     DPTR, #7D69h          ; DPTR=7D69h=target IAP address
MOV     SWCMD, #65h           ; IAP write enable
MOV     IAPWE, #0BCh          ; IAP 7C00h~7DFFh erase enable
MOVX    @DPTR, A              ; write any data to 7D69h to erase 7C00h~7DFFh
                              ; 7C00h~7DFFh convert to ‘1’ after IAP erase
                              ; 2ms H/W writing time, CPU wait
MOV     IAPWE, #00h           ; IAP write disable, immediately after IAP write
CLR     LVRPD                 ; Enable LVR
```

How to write a byte from 7A00h to 7BFFh

(1) Set the DPTR to 7A00h
(2) Set the SWCMD to 65h
(3) Set the IAPWE to 4Ah
(4) MOVX @DPTR, A (write data to 7A00h)

```
  ; IAP example code
  ; need 2.5V < VCC < 5.5V
SETB    LVRPD               ; Disable LVR
MOV     DPTR, #7A00h        ; DPTR=7A00h=target IAP address
MOV     A, #5Ah             ; A=5Ah=target IAP write data
MOV     SWCMD, #65h         ; IAP write enable
MOV     IAPWE, #4Ah         ; IAP write range 7A00h~7BFFh enable

MOVX    @DPTR, A            ; Flash[7A00h] =5Ah, after IAP write
                            ; 20us H/W writing time, CPU wait
MOV     IAPWE, #00h         ; IAP write disable, immediately after IAP write
CLR     A                   ; A=0
MOVC    A, @A+DPTR          ; A=5Ah
CLR     LVRPD               ; Enable LVR
```

How to write a byte from 7C00h to 7DFFh

(1) Set the DPTR to 7C00h
(2) Set the SWCMD to 65h
(3) Set the IAPWE to 4Ch
(4) MOVX @DPTR, A (write data to 7C00h)

```
  ; IAP example code
  ; need 2.5V < VCC < 5.5V
SETB    LVRPD               ; Disable LVR
MOV     DPTR, #7C00h        ; DPTR=7C00h=target IAP address
MOV     A, #5Ah             ; A=5Ah=target IAP write data
MOV     SWCMD, #65h         ; IAP write enable
MOV     IAPWE, #4Ch         ; IAP write range 7C00h~7DFFh enable

MOVX    @DPTR, A            ; Flash[7C00h] =5Ah, after IAP write
                            ; 20us H/W writing time, CPU wait
MOV     IAPWE, #00h         ; IAP write disable, immediately after IAP write
CLR     A                   ; A=0
MOVC    A, @A+DPTR          ; A=5Ah
CLR     LVRPD               ; Enable LVR
```

| Flash **7FFFh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **CFGWH** | PROTN | XRSTEN | LVR | | | – | MVCLOCKN | FRCPSC |

7FFFh.1 **MVCLOCKN:** If 0, the MOVC & MOVX cannot access address from 0000h to 01FFh.

| SFR **97h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **SWCMD** | IAPEN/SWRST | | | | | | | |
| | – | | | | | | WDTO | IAPEN |
| R/W | W | | | | | | R | R |
| Reset | – | | | | | | 0 | 0 |

97h.7~0 **IAPEN (W):**
Write 65h to enable IAP write/erase;
Write other value to disable IAP write/erase. It is recommended to clear it immediately after IAP access.

97h.0 **IAPEN (R):** Flag indicates Flash memory sectors can be accessed by IAP or not.

| SFR **C9h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **IAPWE** | IAPWE | | | | | | | |
| | IAPWE | IAPTO | – | | | | | |
| R/W | R | R | W | | | | | |
| Reset | 0 | 0 | – | | | | | |

C9h.7~0 **IAPWE (W):**
Write 4Ah to enable IAP one byte write to ROM[7A00~7BFF]
Write 4Ch to enable IAP one byte write to ROM[7C00~7DFF]
Write BAh to enable IAP ERASE 512 byte of ROM[7A00~7BFF]
Write BCh to enable IAP ERASE 512 byte of ROM[7C00~7DFF]
Write other value to disable IAP write/page erase

C9h.7 **IAPWE (R):**
0: IAP write/page erase disable
1: IAP write/page erase enable

C9h.6 **IAPTO (R):**
IAP Time-Out flag, Set by H/W when IAP Time-out occurs. Cleared by H/W when IAPWE=0.

| SFR **F7h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **AUX2** | WDTE | | PWRSAV | VBGOUT | DIV32 | IAPTE | | MULDIV16 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

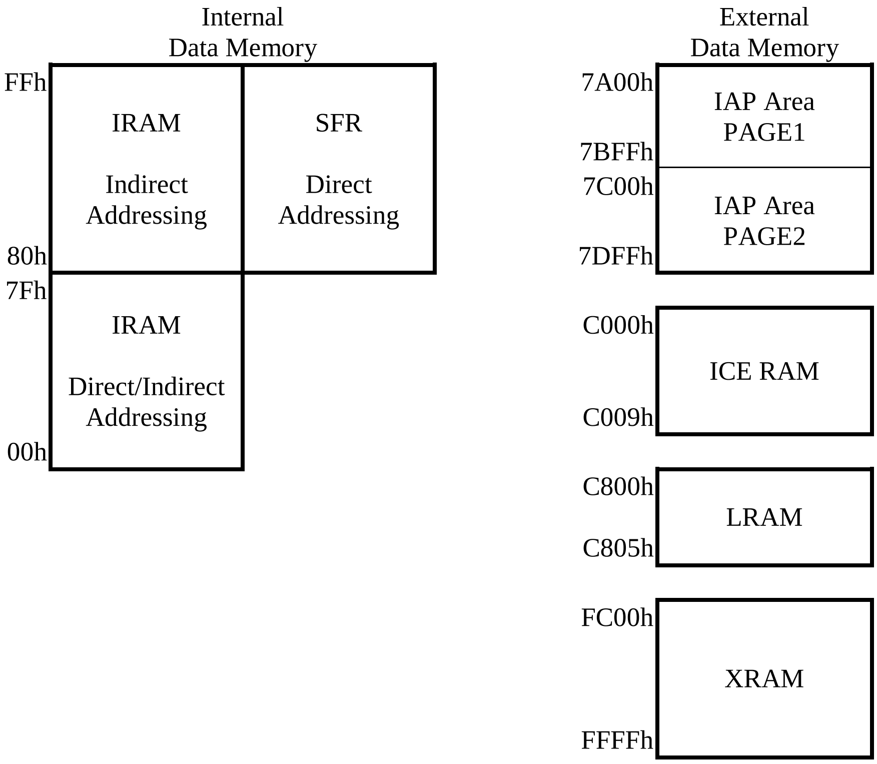
F7h.2~1 **IAPTE:** IAP write watchdog timer enable
00: Disable
01: wait 0.8mS trigger watchdog time-out flag, and escape the write fail state
10: wait 3.2mS trigger watchdog time-out flag, and escape the write fail state
11: wait 6.4mS trigger watchdog time-out flag, and escape the write fail state

| SFR **F8h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **AUX1** | CLRWDT | CLRTM3 | TKSOC | ADSOC | LVRPD | T2SEL | T1SEL | DPSEL |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

F8h.3 **LVRPD:** Low Voltage Reset function select
0: enable
1: disable

## 2.2 Data Memory

As the standard 8051, the Chip has both Internal and External Data Memory space. The Internal Data Memory space consists of 256 Bytes IRAM and SFRs, which are accessible through a rich instruction set. The External Data Memory space consists of 1024 Bytes XRAM, 6 Bytes LCD RAM, 10 Bytes ICE RAM, which can be only accessed by MOVX instruction.



### 2.2.1 IRAM

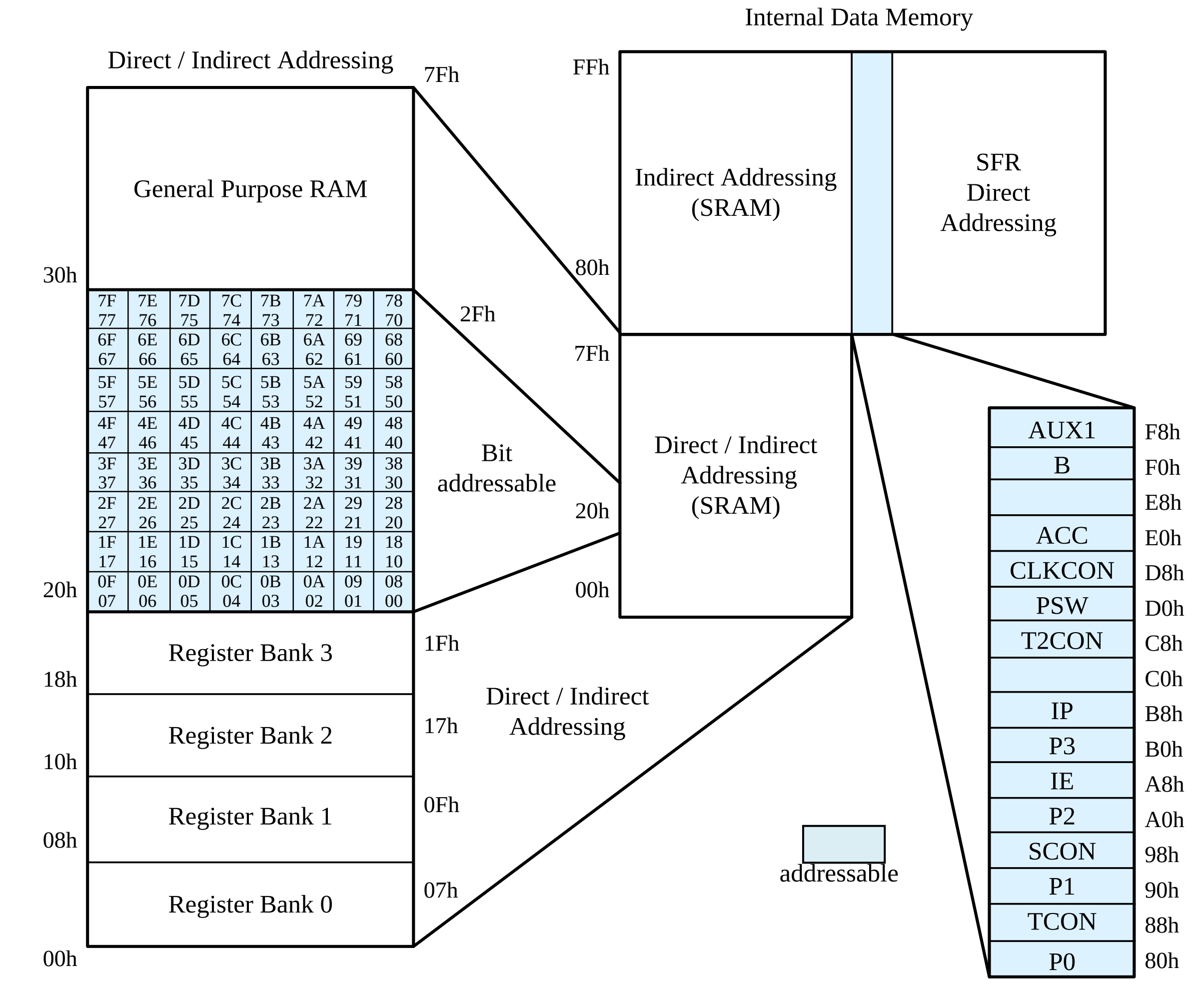
IRAM is located in the 8051 internal data memory space. The whole 256 Bytes IRAM are accessible using indirect addressing but only the lower 128 Bytes are accessible using direct addressing. There are four directly addressable register banks (switching by PSW), which occupy IRAM space from 00h to 1Fh. The address 20h to 2Fh 16 Bytes IRAM space is bit-addressable. IRAM can be used as scratch pad registers or program stack.

### 2.2.2 XRAM

XRAM is located in the 8051 external data memory space (address from FC00h to FFFFh). The 1024 Bytes XRAM can be only accessed by "MOVX" instruction.

### 2.2.3 SFRs

All peripheral functional modules such as I/O ports, Timers and UART operations for the chip are accessed via Special Function Registers (SFRs). These registers occupy upper 128 Bytes of direct Data Memory space locations in the range 80h to FFh. There are 14 bit-addressable SFRs (which means that eight individual bits inside a single byte are addressable), such as ACC, B register, PSW, TCON, SCON, and others. The remaining SFRs are only byte addressable. SFRs provide control and data exchange with the resources and peripherals of the Chip. The TM52 series of microcontrollers provides complete binary code with standard 8051 instruction set compatibility. Beside the standard 8051 SFRs, the Chip implements additional SFRs used to configure and access subsystems such as the ADC/LCD, which are unique to the Chip.



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

| | 8/0 | 9/1 | A/2 | B/3 | C/4 | D/5 | E/6 | F/7 |
|---|---|---|---|---|---|---|---|---|
| F8h | AUX1 | | | | | | | |
| F0h | B | CRCDL | CRCDH | CRCIN | | CFGBG | CFGWL | AUX2 |
| E8h | | SIADR | SICON | SIRCD1 | SITXDRCD2 | | | |
| E0h | ACC | MICON | MIDAT | | | | EXA | EXB |
| D8h | CLKCON | PWM0PRDH | PWM0PRDL | PWM1PRDH | PWM1PRDL | PWM2PRDH | PWM2PRDL | |
| D0h | PSW | PWM0DH | PWM0DL | PWM1DH | PWM1DL | PWM2DH | PWM2DL | |
| C8h | T2CON | IAPWE | RCP2L | RCP2H | TL2 | TH2 | EXA2 | EXA3 |
| C0h | | TKPINSEL0 | TKPINSEL1 | TKPINSEL2 | | ATKCH0 | ATKCH1 | ATKCH2 |
| B8h | IP | IPH | IP1 | IP1H | SPCON | SPSTA | SPDAT | LVDS |
| B0h | P3 | LEDCON | LEDCON2 | LEDCON3 | TKTMRL | TKCON2 | | |
| A8h | IE | INTE1 | ADCDL | ADCDH | | TKCON | CHSEL | P0ADIE |
| A0h | P2 | PWMCON | P1MODL | P1MODH | P3MODL | P3MODH | PINMOD | TKCHS |
| 98h | SCON | SBUF | | | | | PWMOE | PWMCLR |
| 90h | P1 | P0OE | P0LOE | P2MOD | OPTION | INTFLG | P1WKUP | SWCMD |
| 88h | TCON | TMOD | TL0 | TL1 | TH0 | TH1 | SCON2 | SBUF2 |
| 80h | P0 | SP | DPL | DPH | INTEX | INTEXF | INTPWM | PCON |

## 3. LVR and LVD setting

The Chip provides LVR and Low Voltage Detection (LVD) functions. There are 8-level LVR can be selected by CFGWH and 8-level LVD can be selected by SFR LVDS. The SFR PWRSAV/LVRPD bits also affect LVR function as tables below.

| **Operation Mode** | SFR | | CFGWH | LVR | LV Reset | Current consumption |
|---|---|---|---|---|---|---|
| | LVRPD | PWRSAV | LVRE | | | |
| Fast<br>Slow | 0 | X | 000 | ON | 2.3V | |
| | 0 | X | 001 | ON | 2.54V | |
| | 0 | X | 010 | ON | 2.78V | |
| | 0 | X | 011 | ON | 3.04V | |
| | 0 | X | 100 | ON | 3.28V | |
| | 0 | X | 101 | ON | 3.54V | |
| | 0 | X | 110 | ON | 3.8V | |
| | 0 | X | 111 | ON | 4.04V | |
| Idle<br>Halt<br>Stop | 0 | 0 | 000 | ON | 2.3V | Idle: 200uA<br>Halt: 68uA<br>Stop: 65uA |
| | 0 | 0 | 001 | ON | 2.54V | |
| | 0 | 0 | 010 | ON | 2.78V | |
| | 0 | 0 | 011 | ON | 3.04V | |
| | 0 | 0 | 100 | ON | 3.28V | |
| | 0 | 0 | 101 | ON | 3.54V | |
| | 0 | 0 | 110 | ON | 3.8V | |
| | 0 | 0 | 111 | ON | 4.04V | |
| Idle | 0 | 1 | XXX | ON | POR 2.2V | 183uA |
| Halt<br>Stop | 0 | 1 | XXX | OFF | - | Halt: 13uA<br>Stop: 10uA |
| Fast<br>Slow<br>Idle | 1 | X | XXX | ON | POR 2.2V | Idle: 183uA |
| HALT<br>Stop | 1 | X | XXX | OFF | - | Halt: 13uA<br>Stop: 10uA |

**LVR and LVD function**

| SFR **F7h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **AUX2** | WDTE | | PWRSAV | VBGOUT | DIV32 | IAPTE | | MULDIV16 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

F7h.5 **PWRSAV:** Power saving mode control
0: No power saving
1: Power saving, disable LVR in IDLE/HALT/STOP mode

| SFR **F8h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **AUX1** | CLRWDT | CLRTM3 | TKSOC | ADSOC | LVRPD | T2SEL | T1SEL | DPSEL |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

F8h.3 **LVRPD:** Low Voltage Reset function select
0: enable
1: disable

| SFR **BFh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **LVDS** | LVDIE | LVDO | – | – | LVDS | | | ENVPULL |
| R/W | R/W | R | – | – | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | – | – | 0 | 0 | 0 | 0 |

BFh.7 **LVDIE:** Low Voltage Detect interrupt enable
0: Disable
1: Enable (note: EXLVDIE must be 1 at the same time to generate LVD interrupt)

BFh.6 **LVDO:** Low Voltage Detect output

BFh.3~1 **LVDS:** Low Voltage Detect select
000: Set LVD at 2.3V
001: Set LVD at 2.54V
010: Set LVD at 2.78V
011: Set LVD at 3.04V
100: Set LVD at 3.28V
101: Set LVD at 3.54V
110: Set LVD at 3.8V
111: Set LVD at 4.04V

BFh. 0 **ENVPULL:** Power control, force VPULL enable, Must be set to 0
0: Disable
1: Don’t use, cannot be set to 1

| Flash **7FFFh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **CFGWH** | PROTN | XRSTEN | LVRE | | | – | MVCLOCKN | FRCPSC |

7FFFh.5~3 **LVRE:** Low Voltage Reset function select
000: Set LVR at 2.3V
001: Set LVR at 2.54V
010: Set LVR at 2.78V
011: Set LVR at 3.04V
100: Set LVR at 3.28V
101: Set LVR at 3.54V
110: Set LVR at 3.8V
111: Set LVR at 4.04V

# 4. Reset

The Chip has five types of reset methods. Resets can be caused by Power on Reset (POR), External Pin Reset (XRST), Software Command Reset (SWRST), Watchdog Timer Reset (WDTR), or Low Voltage Reset (LVR). The CFGWH controls the Reset functionality. The SFRs are returned to their default value after Reset.

## 4.1 Power on Reset

After Power on Reset, the device stays on Reset state for 40 ms as chip warm up time, then downloads the CFGW register from ROM‘s last six bytes. The Power on Reset needs VCC pin’s voltage first discharge to near VSS level, then rise beyond 2.2V.

## 4.2 External Pin Reset

External Pin Reset is active low. It needs to keep at least 2 SRC clock cycle long to be seen by the Chip. External Pin Reset can be disabled or enabled by CFGW.

## 4.3 Software Command Reset

Software Reset is activated by writing the SFR 97h with data 56h.

## 4.4 Watchdog Timer Reset

WDT overflow Reset is disabled or enabled by SFR F7h. The WDT uses SRC as its counting time base. It runs in Fast/Slow mode and runs or stops in Idle/Stop mode. WDT overflow speed can be defined by WDTPSC SFR. WDT is cleared by device Reset or CLRWDT SFR bit.

## 4.5 Low Voltage Reset

The Chip provides LVR and Low Voltage Detection (LVD) functions. There are 8-level LVR can be selected by CFGWH.

| Flash **7FFFh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **CFGWH** | PROTN | XRSTEN | LVRE | | | – | MVCLOCKN | FRCPSC |

7FFFh.6 **XRSTEN:** External Pin Reset control
0: Enable External Pin Reset
1: Disable External Pin Reset

7FFFh.5~3 **LVRE:** Low Voltage Reset function select
000: Set LVR at 2.3V
001: Set LVR at 2.54V
010: Set LVR at 2.78V
011: Set LVR at 3.04V
100: Set LVR at 3.28V
101: Set LVR at 3.54V
110: Set LVR at 3.8V
111: Set LVR at 4.04V

| SFR 94h | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **OPTION** | UART1W | – | WDTPSC | | ADCKS | | TM3PSC | |
| R/W | R/W | – | R/W | | R/W | | R/W | |
| Reset | 0 | – | 0 | 0 | 0 | 0 | 0 | 0 |

94h.5~4 **WDTPSC:** Watchdog Timer pre-scalar time select

00: 400ms WDT overflow rate
01: 200ms WDT overflow rate
10: 100ms WDT overflow rate
11: 50ms WDT overflow rate

| SFR 95h | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **INTFLG** | LVDIF | – | TKIF | ADIF | – | – | P1IF | TF3 |
| R/W | R/W | – | R/W | R/W | – | – | R/W | R/W |
| Reset | 0 | – | 0 | 0 | – | – | 0 | 0 |

95h.7 **LVDIF:** Low Voltage Detect interrupt flag

Set by H/W. S/W writes 7Fh to INTFLG to clear this flag.

| SFR 97h | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **SWCMD** | IAPEN/SWRST | | | | | | | |
| R/W | W | | | | | | R/W | R/W |
| Reset | – | | | | | | – | 0 |

97h.7~0 **SWRST:** Write 56h to generate S/W Reset

| SFR F7h | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **AUX2** | WDTE | | PWRSAV | VBGOUT | DIV32 | IAPTE | | MULDIV16 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

F7h.7~6 **WDTE:** Watchdog Timer Reset control

0x: Watchdog Timer Reset disable
10: Watchdog Timer Reset enable in Fast/Slow mode, disable in Idle/Halt/Stop mode
11: Watchdog Timer Reset always enable

| SFR F8h | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **AUX1** | CLRWDT | CLRTM3 | TKSOC | ADSOC | LVRPD | T2SEL | T1SEL | DPSEL |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

F8h.7 **CLRWDT:** Set to clear WDT, H/W auto clear it at next clock cycle

F8h.3 **LVRPD:** Low Voltage Reset function select

0: enable
1: disable

# 5. Clock Circuitry & Operation Mode

## 5.1 System Clock

The Chip is designed with dual-clock system. During runtime, user can directly switch the System clock from fast to slow or from slow to fast. It also can directly select a clock divider of 1, 2, 4 or 16. The Fast clock can be selected as FXT (Fast Crystal, 1~18 MHz) or FRC (Fast Internal RC, 18.432 MHz). The Slow clock can be selected as SXT (Slow Crystal, 32 KHz) or SRC (Slow Internal RC, 80 KHz). Fast mode and Slow mode are defined as the CPU running at Fast and Slow clock speeds.

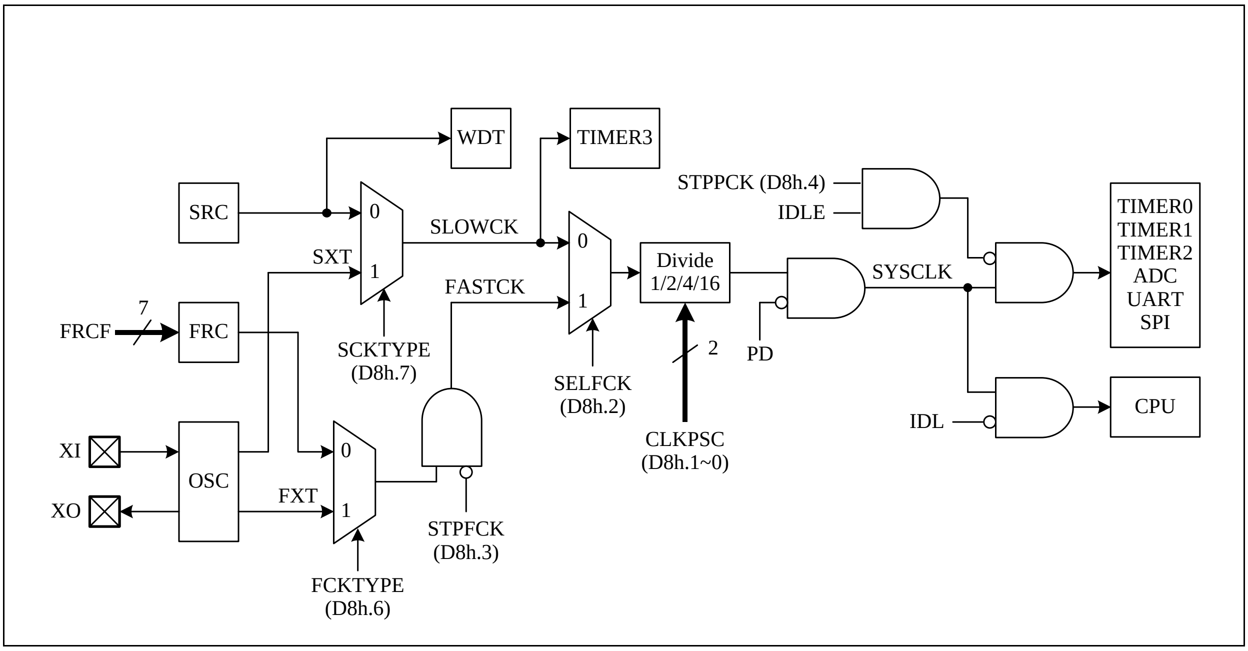
After Reset, the device is running at Slow mode with 80 KHz SRC. S/W should select the proper clock rate for chip operation safety. The higher $V_{CC}$ allows the chip to run at a higher System clock frequency. In a typical condition, a 18 MHz System clock rate requires $V_{CC}$> 2.3V.

The Chip has an external oscillators connected to the XI/XO pins. It relies on external circuitry for the clock signal and frequency stabilization, such as a stand-alone oscillator, quartz crystal, or ceramic resonator. In Fast mode, the fast oscillator can be used in the range from 1~18 MHz. In Slow mode, the slow oscillator can only use a clock frequency of 32.768 KHz.

The **CLKCON** SFR controls the System clock operating. H/W automatically blocks the S/W abnormally setting for this register. S/W can only change the Slow clock type in Fast mode and change the Fast clock type in Slow mode. Never to write both STPFCK=1 & SELFCK=1. It is recommended to write this SFR bit by bit.

**If user wants to switch Fsys from Slow clock to FXT, user should be following the step below**

1. Set FCKTYPE (D8h.6)
2. Wait 2ms until FXT oscillation stable (The actual waiting time depends on the application conditions)
3. Set SELFCK (D8h.2)



**Clock Structure**

The chip can also output the "System clock divided by 2" signal (CKO) to P1.4 pin. CKO pin’s output setting is controlled by TCOE SFR (***see section 7***).

| SYSCLK | CLKCON (D8h) | | | |
|---|---|---|---|---|
| | bit7<br>SCKTYPE | bit6<br>FCKTYPE | bit3<br>STPFCK | bit2<br>SELFCK |
| Fast FXT | 0/1 | 1 | 0 | 1 |
| Fast FRC | 0/1 | 0 | 0 | 1 |
| Slow SXT | 1 | 0/1 | 0/1 | 0 |
| Slow SRC | 0 | 0/1 | 0/1 | 0 |
| Fast type change | 0/1 | 0 ← → 1 | 0/1 | 0 |
| Slow type change | 0 ← → 1 | 0/1 | 0 | 1 |
| Stop FRC/FXT | 0/1 | 0/1 | 0 → 1 | 0 |
| Switch to FRC/FXT | 0/1 | 0/1 | 0 | 0 → 1 |
| Switch to SRC/SXT | 0/1 | 0/1 | 0 | 1 → 0 |

| SFR **F6h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **CFGWL** | – | FRCF | | | | | | |
| R/W | – | R/W | | | | | | |
| Reset | – | – | – | – | – | – | – | – |

F6h.6~0 **FRCF:** FRC frequency adjustment, automatically load the calibration value after power-on
00h= lowest frequency, 7Fh=highest frequency.

| SFR **D8h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **CLKCON** | SCKTYPE | FCKTYPE | STPSCK | STPPCK | STPFCK | SELFCK | CLKPSC | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Reset | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |

D8h.7 **SCKTYPE:** Slow clock type. This bit can be changed only in Fast mode (SELFCK=1).
0: SRC
1: SXT, P2.0 and P2.1 are crystal pins

D8h.6 **FCKTYPE:** Fast clock type. This bit can be changed only in Slow mode (SELFCK=0).
0: FRC
1: FXT, P2.0 and P2.1 are crystal pins, oscillator gain is high for FXT

D8h.5 **STPSCK:** Set 1 to stop Slow clock after PD=1 (Halt / Stop mode control)

D8h.4 **STPPCK:** Set 1 to stop UARTs/Timer0/Timer1/Timer2/ADC clock in Idle mode for current reducing. If set, only Timer3 and pin interrupts are alive in Idle Mode.

D8h.3 **STPFCK:** Set 1 to stop Fast clock for power saving in Slow/Idle mode. This bit can be changed only in Slow mode.

D8h.2 **SELFCK:** System clock source selection. This bit can be changed only when STPFCK=0.
0: Slow clock
1: Fast clock

D8h.1~0 **CLKPSC:** System clock prescaler. Effective after 16 clock cycles (Max.) delay.
00: System clock is Fast/Slow clock divided by 16
01: System clock is Fast/Slow clock divided by 4
10: System clock is Fast/Slow clock divided by 2
11: System clock is Fast/Slow clock divided by 1

## 5.2 Operation Modes

There are four operation modes for this device. **Fast Mode** is defined as the CPU running at Fast clock speed. **Slow Mode** is defined as the CPU running at Slow clock speed. When the System clock speed is lower, the power consumption is lower.

**Idle Mode** is entered by setting the IDL bit in PCON SFR. Both Fast and Slow clock can be set as the System clock source in Idle Mode, but Slow clock is better for power saving. In Idle mode, the CPU puts itself to sleep while the on-chip peripherals stay active. The "STPPCK" bit in CLKCON SFR can be set to furthermore reduce Idle mode current. If STPPCK is set, only Timer3 and pin interrupts are alive in Idle Mode, others peripherals such as Timer0/1/2, UARTs and ADC are stop. The slower System clock rate also helps current saving. It can be achieved by setup the CLKPSC SFR to divide System clock frequency. Idle mode is terminated by Reset or enabled Interrupts wake up.

**Halt Mode** is entered by setting the PD bit in PCON SFR and STPSCK is cleared. In Halt mode, all clocks stop except the Timer3 and WDT could be alive if they are enabled. Halt Mode is terminated by Reset, pin wake up or Timer3 interrupt.

**Stop Mode** is entered by setting the PD bit in PCON SFR. This mode is the so-called "Power Down" mode in standard 8051. In Stop mode, all clocks stop except the WDT could be alive if it is enabled. Stop Mode is terminated by Reset or pin wake up. Must be set to slow clock mode (SELFCK=0) before entering Stop mode (PDOWN).

***Note:*** *Chip cannot enter Stop/Halt Mode if INTn pin is low and wakeup is enable. (INTn=0 and EXn=1, n=0~9)*

| SFR **87h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PCON** | SMOD | – | – | – | GF1 | GF0 | PD | IDL |
| R/W | R/W | – | – | – | R/W | R/W | R/W | R/W |
| Reset | 0 | – | – | – | 0 | 0 | 0 | 0 |

87h.1 **PD:** Power down control bit, set 1 to enter STOP/Halt mode.

87h.0 **IDL:** Idle mode control bit, set 1 to enter IDLE mode.

| SFR **F7h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **AUX2** | WDTE | | PWRSAV | VBGOUT | DIV32 | IAPTE | | MULDIV16 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

F7h.4 **VBGOUT:** VBG voltage output to P3.2
0: Disable
1: Enable, The additional condition VBGEN=1 (AEh.1) should be set.

| SFR **D8h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **CLKCON** | SCKTYPE | FCKTYPE | STPSCK | STPPCK | STPFCK | SELFCK | CLKPSC | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Reset | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |

D8h.7 **SCKTYPE:** Slow clock type. This bit can be changed only in Fast mode (SELFCK=1).
0: SRC 1: SXT

D8h.6 **FCKTYPE:** Fast clock type. This bit can be changed only in Slow mode (SELFCK=0).
0: FRC 1: FXT

D8h.5 **STPSCK:** Set 1 to stop Slow clock after PD=1 (Halt / Stop mode control)

D8h.4 **STPPCK:** Set 1 to stop UART/Timer0/Timer1/Timer2/ADC clock in Idle mode for current reducing. If set, only Timer3 and pin interrupts are alive in Idle Mode.

D8h.3 **STPFCK:** Set 1 to stop Fast clock for power saving in Slow/Idle mode. This bit can be changed only in Slow mode.

D8h.2 **SELFCK:** System clock source selection. This bit can be changed only when STPFCK=0.
0: Slow clock 1: Fast clock

D8h.1~0 **CLKPSC:** System clock prescaler. Effective after 16 clock cycles (Max.) delay.
00: System clock is Fast/Slow clock divided by 16
01: System clock is Fast/Slow clock divided by 4
10: System clock is Fast/Slow clock divided by 2
11: System clock is Fast/Slow clock divided by 1

## 6. Interrupt & Wake-up

This Chip has a 14-source four-level priority interrupt structure. All enabled Interrupts can wake up CPU from Idle mode, but only the Pin Interrupts can wake up CPU from Stop mode. The Halt mode can be waked up by Time3 and Pin Interrupts. Each interrupt source has its own enable control bit. An interrupt event will set its individual Interrupt Flag, no matter whether its interrupt enable control bit is 0 or 1. The Interrupt vectors and flags are list below.

| No. | Vector | Flag | Description |
|---|---|---|---|
| 0 | 0003 | IE0 | INT0 external pin Interrupt (can wake up Halt/Stop mode) |
| 1 | 000B | TF0 | Timer0 Interrupt |
| 2 | 0013 | IE1 | INT1 external pin Interrupt (can wake up Halt/Stop mode) |
| 3 | 001B | TF1 | Timer1 Interrupt |
| 4 | 0023 | RI+TI | Serial Port (UART1) Interrupt |
| 5 | 002B | TF2+EXF2 | Timer2 Interrupt |
| 6 | 0033 | – | Reserved for ICE mode use |
| 7 | 003B | TF3 | Timer3 Interrupt (can wake up Halt mode) |
| 8 | 0043 | P1IF | Port1 external pin change Interrupt (can wake up Halt/Stop mode) |
| 9 | 004B | IE2~IE9<br>LVDIF | INT2~INT9 external pin Interrupt (can wake up Halt/Stop mode)<br>LVD interrupt |
| 10 | 0053 | ADIF+TKIF | ADC/Touch Key Interrupt |
| 11 | 005B | SPIF+WCOL+MODF | SPI Interrupt |
| 12 | 0063 | RI2+TI2 | Serial Port (UART2) Interrupt |
| 13 | 006B | MIIF<br>TXDF<br>RCD2F<br>RCD1F | I$^2$C interrupt Vector |
| 14 | 0073 | PWM0IF<br>PWM1IF<br>PWM2IF | PWM0~2 Interrupt Vector |

**Interrupt Vector & Flag**

### 6.1 Interrupt Enable and Priority Control

The IE and INTE1 SFRs decide whether the pending interrupt is serviced by CPU. The P1WKUP SFR controls the individual Port1 pin's wake-up and interrupt capability. The IP, IPH, IP1 and IP1H SFRs decide the interrupt priority. An interrupt will be serviced as long as an interrupt of equal or higher priority is not already being serviced. If an interrupt of equal or higher level priority is being serviced, the new interrupt will wait until it is finished before being serviced. If a lower priority level interrupt is being serviced, it will be stopped and the new interrupt serviced. When the new interrupt is finished, the lower priority level interrupt that was stopped will be completed.

### 6.2 Suggestions on interrupting subroutines

The period and duty cycle of PWM are 16-bit operations. When writing and reading the high and low bytes of PWMxDH, PWMxDL, PWMxPRDH and PWMxPRDL, interrupts should be avoided. If you are reading and writing these 16-bit SFRs in the meantime an interrupt occurs. And these SFRs are read and written in the interrupt. It is easy to cause read and write errors. For the 16-bit PWM period and duty to read and write, it is recommended to update the data only in the main program, or update the data only in the interrupt to avoid possible errors.

| SFR **84h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **INTEX** | EX9 | EX8 | EX7 | EX6 | EX5 | EX4 | EX3 | EX2 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

84h.7~0 **EX9~EX2:** External INT9~INT2 pin Interrupt enable and Stop/Halt mode wake up enable.

0: Disable INTx pin Interrupt and Stop/Halt mode wake up

1: Enable INTx pin Interrupt and Stop/Halt mode wake up, it can wake up CPU from Stop/Halt mode no matter EA is 0 or 1. (note: EXLVDIE must be 1 at the same time to generate INTx interrupt and wake up)

| SFR **96h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **P1WKUP** | P1WKUP | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

96h.7~0 **P1WKUP:** P1.7~P1.0 pin individual Wake-up / Interrupt enable control

0: Disable

1: Enable

| SFR **9Eh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWMOE** | PWM1IE | PWM0IE | – | – | – | PWM2OE | PWM1OE | PWM0OE |
| R/W | R/W | R/W | – | – | – | R/W | R/W | R/W |
| Reset | 0 | 0 | – | – | – | 0 | 0 | 0 |

9Eh.7 **PWM1IE:** PWM1 Interrupt Enable

0: disable

1: enable (note: PWMIE must be 1 at the same time to generate PWM interrupt)

9Eh.6 **PWM0IE:** PWM0 Interrupt Enable

0: disable

1: enable (note: PWMIE must be 1 at the same time to generate PWM interrupt)

| SFR **9Fh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWMCLR** | PWM2IE | – | – | – | – | PWM2CLR | PWM1CLR | PWM0CLR |
| R/W | R/W | – | – | – | – | R/W | R/W | R/W |
| Reset | 0 | – | – | – | – | 0 | 0 | 0 |

9Fh.7 **PWM2IE:** PWM2 Interrupt Enable

0: disable

1: enable (note: PWMIE must be 1 at the same time to generate PWM interrupt)

| SFR **A8h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **IE** | EA | – | ET2 | ES | ET1 | EX1 | ET0 | EX0 |
| R/W | R/W | – | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | – | 0 | 0 | 0 | 0 | 0 | 0 |

A8h.7 **EA:** Global interrupt enable control.
0: Disable all Interrupts.
1: Each interrupt is enabled or disabled by its individual interrupt control bit

A8h.5 **ET2:** Timer2 interrupt enable
0: Disable Timer2 interrupt
1: Enable Timer2 interrupt

A8h.4 **ES:** Serial Port (UART1) interrupt enable
0: Disable Serial Port (UART1) interrupt
1: Enable Serial Port (UART1) interrupt

A8h.3 **ET1:** Timer1 interrupt enable
0: Disable Timer1 interrupt
1: Enable Timer1 interrupt

A8h.2 **EX1:** External INT1 pin Interrupt enable and Stop/Halt mode wake up enable
0: Disable INT1 pin Interrupt and Stop/Halt mode wake up
1: Enable INT1 pin Interrupt and Stop/Halt mode wake up, it can wake up CPU from Stop/Halt mode no matter EA is 0 or 1.

A8h.1 **ET0:** Timer0 interrupt enable
0: Disable Timer0 interrupt
1: Enable Timer0 interrupt

A8h.0 **EX0:** External INT0 pin Interrupt enable and Stop/Halt mode wake up enable
0: Disable INT0 pin Interrupt and Stop/Halt mode wake up
1: Enable INT0 pin Interrupt and Stop/Halt mode wake up, it can wake up CPU from Stop/Halt mode no matter EA is 0 or 1.

| SFR **A9h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **INTE1** | PWMIE | I2CE | ES2 | SPIE | ADTKIE | EXLVDIE | P1IE | TM3IE |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

A9h.7 **PWMIE:** PWM0~PWM2 interrupt enable
0: Disable PWM0~PWM2 interrupt
1: Enable PWM0~PWM2 interrupt

A9h.6 **I2CE:** I$^2$C (master/slave) interrupt enable
0: Disable I$^2$C interrupt
1: Enable I$^2$C interrupt

A9h.5 **ES2:** Serial Port (UART2) interrupt enable
0: Disable Serial Port (UART2) interrupt
1: Enable Serial Port (UART2) interrupt

A9h.4 **SPIE:** SPI interrupt enable
0: Disable SPI interrupt
1: Enable SPI interrupt

A9h.3 **ADTKIE:** ADC/Touch Key interrupt enable
0: Disable ADC/Touch Key interrupt
1: Enable ADC/Touch Key interrupt

A9h.2 **EXLVDIE:** External INT2~INT9 and LVD interrupt enable and Stop/Halt mode wake up enable
0: Disable INT2~INT9 pin Interrupt and Stop/Halt mode wake up
Disable LVD interrupt
1: Enable INT2~INT9 pin Interrupt and Stop/Halt mode wake up, it can wake up CPU from Stop/Halt mode no matter EA is 0 or 1.
Enable LVD interrupt.

A9h.1 **P1IE:** Port1 pin change interrupt enable. This bit does not affect the Port1 pin's Stop/Halt mode wake up capability.
0: Disable Port1 pin change interrupt
1: Enable Port1 pin change interrupt

A9h.0 **TM3IE:** Timer3 interrupt enable and Halt mode wake up enable
0: Disable Timer3 interrupt t and Halt mode wake up
1: Enable Timer3 interrupt t and Halt mode wake up, it can wake up CPU from Halt mode no matter EA is 0 or 1.

| SFR **B9h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **IPH** | – | – | PT2H | PSH | PT1H | PX1H | PT0H | PX0H |
| R/W | – | – | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | – | – | 0 | 0 | 0 | 0 | 0 | 0 |

| SFR **B8h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **IP** | – | – | PT2 | PS | PT1 | PX1 | PT0 | PX0 |
| R/W | – | – | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | – | – | 0 | 0 | 0 | 0 | 0 | 0 |

B9h.5, B8h.5 **PT2H, PT2 :** Timer2 Interrupt Priority control. (PT2H, PT2) =
11: Level 3 (highest priority)
10: Level 2
01: Level 1
00: Level 0 (lowest priority)

B9h.4, B8h.4 **PSH, PS :** Serial Port (UART1) Interrupt Priority control. Definition as above.

B9h.3, B8h.3 **PT1H, PT1 :** Timer1 Interrupt Priority control. Definition as above.

B9h.2, B8h.2 **PX1H, PX1 :** External INT1 pin Interrupt Priority control. Definition as above.

B9h.1, B8h.1 **PT0H, PT0 :** Timer0 Interrupt Priority control. Definition as above.

B9h.0, B8h.0 **PX0H, PX0 :** External INT0 pin Interrupt Priority control. Definition as above.

| SFR **BBh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **IP1H** | PPWMH | PI2CH | PS2H | PSPIH | PADTKIH | PX2_9LVDH | PP1H | PT3H |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| SFR **BAh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **IP1** | PPWM | PI2C | PS2 | PSPI | PADTKI | PX2_9LVD | PP1 | PT3 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

BBh.7, BAh.7 **PPWMH, PPWM:** PWM0~PWM2 Interrupt Priority control. Definition as above.

BBh.6, BAh.6 **PI2CH, PI2C:** I2C (Master/Slave) Interrupt Priority control. Definition as above.

BBh.5, BAh.5 **PS2H, PS2:** Serial Port (UART2) Interrupt Priority control. Definition as above.

BBh.4, BAh.4 **PSPIH, PSPI:** SPI Interrupt Priority control. Definition as above.

BBh.3, BAh.3 **PADTKIH, PADTKI:** ADC/Touch Key Interrupt Priority control. Definition as above.

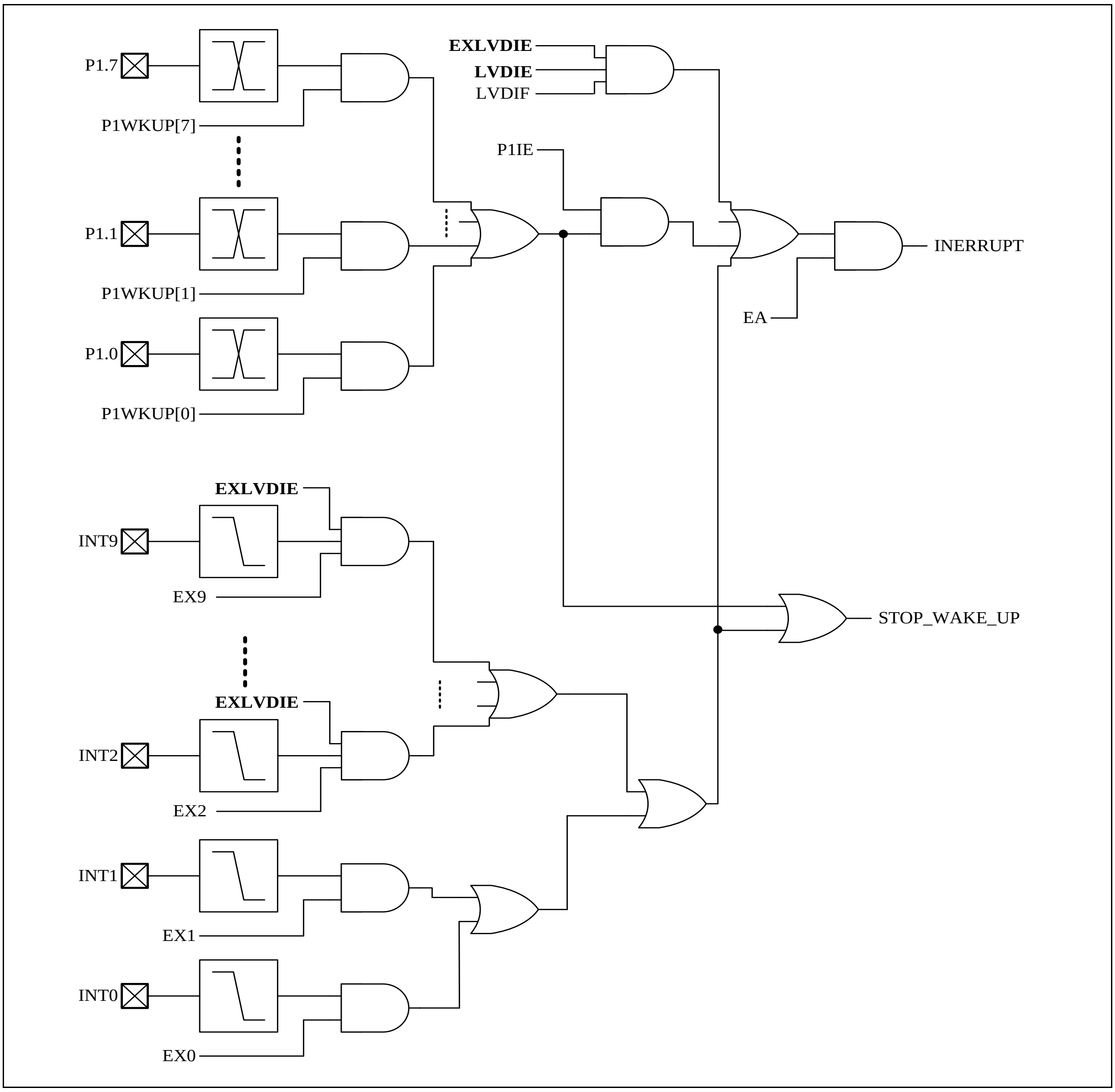
BBh.2, BAh.2 **PX2_9LVDH, PX2_9LVD:** External INT2~INT9 pin and LVD Interrupt Priority control. Definition as above.

BBh.1, BAh.1 **PP1H, PP1:** Port1 Pin Change Interrupt Priority control. Definition as above.

BBh.0, BAh.0 **PT3H, PT3:** Timer3 Interrupt Priority control. Definition as above.

## 6.3 Pin Interrupt and LVD interrupt

Pin Interrupts include INT0~INT9 and Port1 Change. INT0~INT9 and Port1 also have the Stop/Halt mode wake up capability. INT0 and INT1 are falling edge or low level triggered as the 8051 standard. INT2~INT9 is falling edge triggered and Port1 Change Interrupt is triggered by Port1 state change. LVD interrupt can be used to detect the $V_{CC}$ voltage level and generate an interrupt.



**Pin interrup/Wake up & LVD interrupt**

***Note:*** Chip cannot enter Stop/Halt Mode if INTn pin is low and wakeup is enabled. (INTn=0 and EXn=1, n=0~9)

| SFR **84h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **INTEX** | EX9 | EX8 | EX7 | EX6 | EX5 | EX4 | EX3 | EX2 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

84h.7~0 **EX9~EX2:** External INT9~INT2 pin Interrupt enable and Stop/Halt mode wake up enable.

0: Disable INTx pin Interrupt and Stop/Halt mode wake up

1: Enable INTx pin Interrupt and Stop/Halt mode wake up, it can wake up CPU from Stop/Halt mode no matter EA is 0 or 1. (note: EXLVDIE must be 1 at the same time to generate INTx interrupt wake up)

| SFR **85h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **INTEXF** | IE9 | IE8 | IE7 | IE6 | IE5 | IE4 | IE3 | IE2 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

85h.7~0 **IE9~2:** External Interrupt INT9~INT2 edge flag.

Set by H/W when an INTx pin falling edge is detected, no matter the EXx is 0 or 1.

S/W Write 0 to clear interrupt flag, no automatic clear after the interrupt service routine.

| SFR **88h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TCON** | TF1 | TR1 | TF0 | TR0 | IE1 | IT1 | IE0 | IT0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

88h.3 **IE1:** External Interrupt 1 (INT1 pin) edge flag.

Set by H/W when an INT1 pin falling edge is detected, no matter the EX1 is 0 or 1.

It is cleared automatically when the program performs the interrupt service routine.

88h.2 **IT1:** External Interrupt 1 control bit

0: Low level active (level triggered) for INT1 pin

1: Falling edge active (edge triggered) for INT1 pin

88h.1 **IE0:** External Interrupt 0 (INT0 pin) edge flag

Set by H/W when an INT0 pin falling edge is detected, no matter the EX0 is 0 or 1.

It is cleared automatically when the program performs the interrupt service routine.

88h.0 **IT0:** External Interrupt 0 control bit

0: Low level active (level triggered) for INT0 pin

1: Falling edge active (edge triggered) for INT0 pin

| SFR **95h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **INTFLG** | LVDIF | – | TKIF | ADIF | – | – | P1IF | TF3 |
| R/W | R | – | R/W | R/W | – | – | R/W | R/W |
| Reset | – | – | 0 | 0 | – | – | 0 | 0 |

95h.7 **LVDIF:** LVD interrupt flag

Set by H/W, S/W can write 7Fh to INTFLG to clear this bit.

95h.1 **P1IF:** Port1 pin change interrupt flag

Set by H/W when a Port1 pin state change is detected and its interrupt enable bit is set (P1WKUP). P1IE does not affect this flag's setting.

It is cleared automatically when the program performs the interrupt service routine.

S/W can write FDh to INTFLG to clear this bit. (***Note1***)

***Note1:*** *S/W can write 0 to clear a flag in the INTFLG, but writing 1 has no effect.*

| SFR **96h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **P1WKUP** | P1WKUP | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

96h.7~0 **P1WKUP:** P1.7~P1.0 pin individual Wake-up / Interrupt enable control
0: Disable
1: Enable

| SFR **A8h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **IE** | EA | – | ET2 | ES | ET1 | EX1 | ET0 | EX0 |
| R/W | R/W | – | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | – | 0 | 0 | 0 | 0 | 0 | 0 |

A8h.7 **EA:** Global interrupt enable control.
0: Disable all Interrupts.
1: Each interrupt is enabled or disabled by its individual interrupt control bit

A8h.2 **EX1:** External INT1 pin Interrupt enable and Stop/Halt mode wake up enable
0: Disable INT1 pin Interrupt and Stop/Halt mode wake up
1: Enable INT1 pin Interrupt and Stop/Halt mode wake up, it can wake up CPU from Stop/Halt mode no matter EA is 0 or 1.

A8h.0 **EX0:** External INT0 pin Interrupt enable and Stop/Halt mode wake up enable
0: Disable INT0 pin Interrupt and Stop/Halt mode wake up
1: Enable INT0 pin Interrupt and Stop/Halt mode wake up, it can wake up CPU from Stop/Halt mode no matter EA is 0 or 1.

| SFR **A9h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **INTE1** | PWMIE | I2CE | ES2 | SPIE | ADTKIE | EXLVDIE | P1IE | TM3IE |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

A9h.2 **EXLVDIE:** External INT2~INT9 and LVD interrupt enable and Stop/Halt mode wake up enable
0: Disable INT2~INT9 pin Interrupt and Stop/Halt mode wake up
Disable LVD interrupt
1: Enable INT2~INT9 pin Interrupt and Stop/Halt mode wake up, it can wake up CPU from Stop/Halt mode no matter EA is 0 or 1.
Enable LVD interrupt.

| SFR **BFh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **LVDS** | LVDIE | LVDO | – | – | LVDS | | | ENVPULL |
| R/W | R/W | R | – | – | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | – | – | 0 | 0 | 0 | 0 |

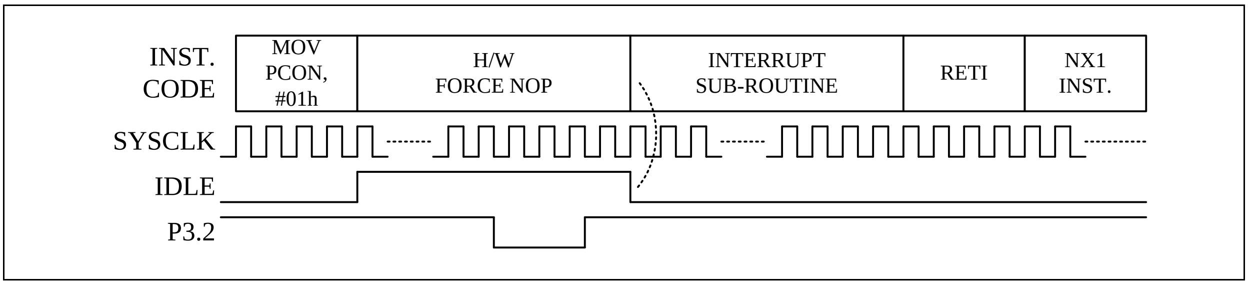
BFh.7 **LVDIE:** Low Voltage Detect interrupt enable
0: Disable
1: Enable (note: EXLVDIE must be 1 at the same time to generate LVD interrupt)

BFh.3~1 **LVDS:** Low Voltage Detect select
000: Set LVD at 2.3V
001: Set LVD at 2.54V
010: Set LVD at 2.78V
011: Set LVD at 3.04V
100: Set LVD at 3.28V
101: Set LVD at 3.54V
110: Set LVD at 3.8V
111: Set LVD at 4.04V

BFh. 0 **ENVPULL:** Power control, force VPULL enable, Must be set to 0
0: Disable
1: Don’t use, cannot be set to 1

## 6.4 Idle mode Wake up and Interrupt

Idle mode is waked up by enabled Interrupts, which means individual interrupt enable bit (ex: EX0) and EA bit must be both set to 1 to establish Idle mode wake up capability. All enabled Interrupts (Pins, Timers, ADC, TK, SPI and UARTs) can wake up CPU from Idle mode. Upon Idle wake-up, Interrupt service routine is entered immediately. “The first instruction behind IDL (PCON.0) setting” is executed after interrupt service routine return.



**EA=EX0=1, Idle mode wake-up and Interrupt by P3.2 (INT0)**

| SFR **87h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PCON** | SMOD | – | – | – | GF1 | GF0 | PD | IDL |
| R/W | R/W | – | – | – | R/W | R/W | R/W | R/W |
| Reset | 0 | – | – | – | 0 | 0 | 0 | 0 |

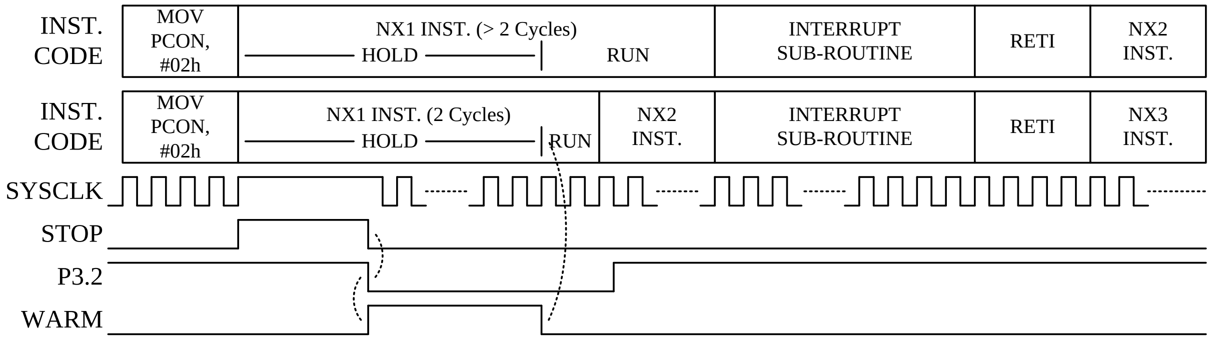
87h.1 **PD:** Power down control bit, set 1 to enter STOP/Halt mode.

87h.0 **IDL:** Idle mode control bit, set 1 to enter IDLE mode.

## 6.5 Stop/Halt mode Wake up and Interrupt

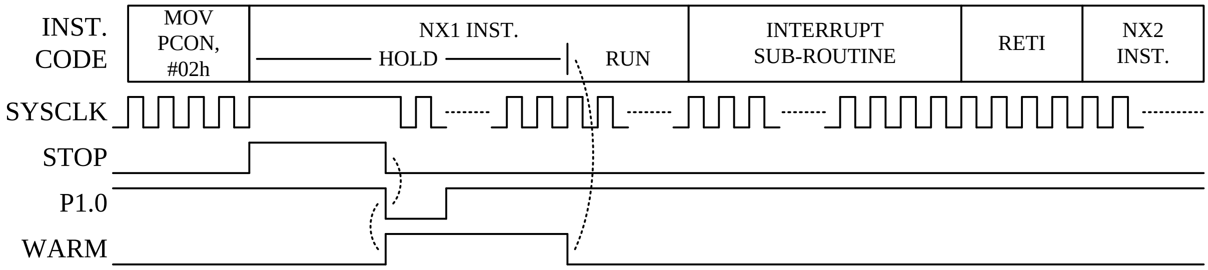
Stop/Halt mode wake up is simple, as long as the individual pin interrupt enable bit (ex: EX0) is set, the pin wake up capability is asserted. Set EX0/EX1/EXLVDIE can enable INT0/INT1/INT2 pins’ Stop/Halt mode wake up capability. Set P1WKUP bit 7~0 can enable P1.7~P1.0’s Stop/Halt mode wake up capability. Upon Stop/Halt wake up, “the first instruction behind PD setting (PCON.1)” is executed immediately before Interrupt service. Interrupt entry requires EA=1 (P1WKUP also needs P1IE=1) and trigger state of the pin staying sufficiently long to be observed by the System clock. This feature allows CPU to enter or not enter Interrupt sub-routine after Stop/Halt mode wake up.

***Note:*** *It is recommended to place the NX1/NX2 with NOP Instruction in figures below.*
***Note:*** *If the INTn pin is low and this wakeup function is enabled, the chip cannot enter stop/suspend mode. (INTn=0 and Exn=1, n=0~9)*



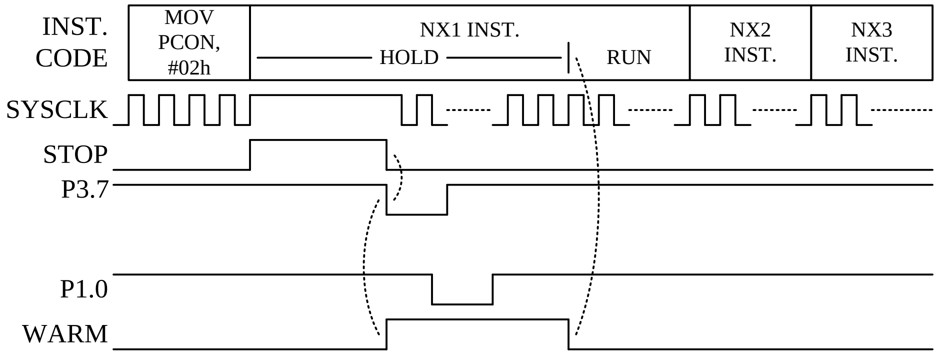
**EA=EX0=1**

**P3.2 (INT0) is sampled after warm-up, Stop mode wake-up and Interrupt.**



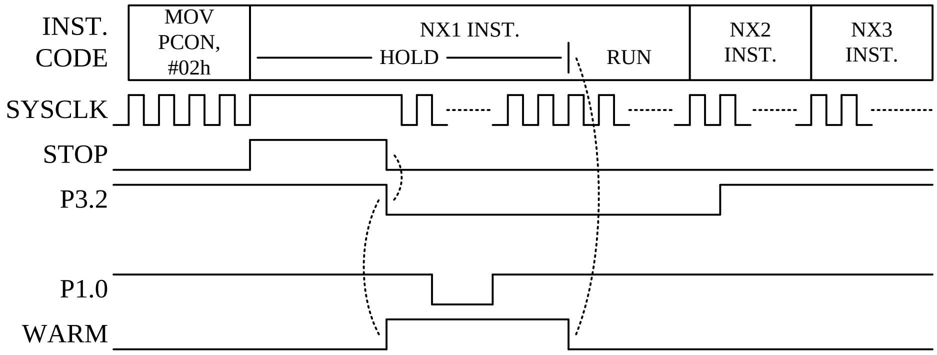
**EA=P1IE=P1WKUP=1**

**P1.0 change (not need clock sample), Stop mode wake-up and Interrupt.**



**EA=EX0=P1WKUP=1, P1IE=0**

**Stop mode wake-up but not Interrupt, P3.2/P3.7 pulse too narrow.**



**EX0=P1WKUP=1=P1IE=1, EA=0**

**Stop mode wake-up but not Interrupt.**

# 7. I/O Ports

The Chip has total 26 multi-function I/O pins. All I/O pins follow the standard 8051 “Read-Modify-Write” feature. The instructions that read the SFR rather than the Pin State are the ones that read a port or port bit value, possibly change it, and then rewrite it to the SFR (ex: ANL P1, A; INC P2; CPL P3.0).

## 7.1 Port1 & Port2 & Port 3

These pins can operate in four different modes as below.

| Mode | Port1, Port2, Port3 pin function | | Px.n SFR data | Pin State | Resistor Pull-up | Digital Input |
|---|---|---|---|---|---|---|
| | P3.0~P3.2 | Others | | | | |
| **Mode 0** | Pseudo Open Drain | Open Drain | 0 | Drive Low | N | N |
| | | | 1 | Pull-up | Y | Y |
| **Mode 1** | Pseudo Open Drain | Open Drain | 0 | Drive Low | N | N |
| | | | 1 | Hi-Z | N | Y |
| **Mode 2** | CMOS Output | | 0 | Drive Low | N | N |
| | | | 1 | Drive High | N | N |
| **Mode 3** | Analog input for ADC, digital input buffer is disabled | | X (don’t care) | – | N | N |

**Port1, Port2, Port3 I/O Pin Function Table**

If Port1, Port2 or Port3 pin is used for Schmitt-trigger input, S/W must set the I/O pin to Mode0 or Mode1 and set the corresponding Port Data SFR to 1 to disable the pin’s output driving circuitry.

Beside I/O port function, each Port1, Port2 and Port3 pin has one or more alternative functions, such as LED, ADC and Touch Key. Most of the functions are activated by setting the individual pin mode control SFR to Mode3. Port1/Port3 pins have standard 8051 auxiliary definition such as INT0/1, T0/1/2, or RXD/TXD. These pin functions need to set the pin mode SFR to Mode0 or Mode1 and keep the P1.n/P3.n SFR at 1.

| Pin Name | 8051 | Wake-up | CKO | ADC | TK | LED<br>BiD matrix | LED<br>Dot matrix | others |
|---|---|---|---|---|---|---|---|---|
| P1.7 | TXD2 | Y | | | TK10 | | | MISO |
| P1.6 | | Y | | | TK9 | | | PWM2 |
| P1.5 | | Y | | AD9 | TK14 | | | |
| P1.4 | | Y | CKO | AD8 | TK8 | | | |
| P1.3 | | Y | | AD7 | TK7 | | | PWM1 |
| P1.2 | | Y | | AD6 | TK6 | | | PWM0 |
| P1.1 | T2EX | Y | | AD5 | TK5 | | | |
| P1.0 | T2 | Y | T2O | AD4 | TK4 | | | |

**Port1 multi-function Table**

| Pin Name | 8051 | Wake-up | CKO | ADC | TK | LED<br>BiD matrix | LED<br>Dot matrix | others |
|---|---|---|---|---|---|---|---|---|
| P3.7 | INT2 | Y | | | TK15 | LEDS2 | LED6 | RSTn |
| P3.6 | RXD2 | Y | | | TK11 | LEDS5 | | SCK |
| P3.5 | T1 | Y | | | TK12 | LEDS4 | LED8 | MOSI |
| P3.4 | T0 | Y | T0O | | TK13 | LEDS3 | LED7 | SS |
| P3.3 | INT1 | Y | | AD0 | TK0 | | | |
| P3.2 | INT0 | Y | | AD1 | TK1 | | | VBGO |
| P3.1 | TXD | Y | | AD2 | TK2 | | | SDA |
| P3.0 | RXD | Y | | AD3 | TK3 | | | SCL |

**Port3 multi-function Table**

| Pin Name | 8051 | Wake-up | CKO | ADC | TK | LED<br>BiD matrix | LED<br>Dot matrix | others |
|---|---|---|---|---|---|---|---|---|
| P2.1 | | | | | | LEDS1 | LED5 | XO |
| P2.0 | | | | | | LEDS0 | LED4 | XI |

**P2 multi-function Table**

The necessary SFR setting for Port1/ Port2/Port3 pin’s alternative function is list below.

| Alternative Function | Mode | Px.n SFR data | Pin State | Other necessary SFR setting |
|---|---|---|---|---|
| T0, T1, T2, T2EX, INT0, INT1, INT2 | **0** | 1 | Input with Pull-up | |
| | **1** | 1 | Input | |
| RXD, TXD | **0** | 1 | Input with Pull-up / Pseudo Open Drain Output | |
| | **1** | 1 | Input / Pseudo Open Drain Output | |
| RXD2,TXD2 | **0** | 1 | Input with Pull-up / Open Drain Output | |
| | **1** | 1 | Input / Open Drain Output | |
| T0O, T2O, CKO | **0** | X | Clock Open Drain Output with Pull-up | PINMOD |
| | **1** | X | Clock Open Drain Output | |
| | **2** | X | Clock Output (CMOS Push-Pull) | |
| VBGO | **X** | X | Bandgap Voltage output | VBGOUT<br>VBGEN |
| LEDS0~ LEDS5<br>LEDC0~ LEDC3 | **X** | X | LED BiD matrix mode Output | LEDCON |
| LED0~ LED8 | **X** | X | LED Dot matrix mode Output | LEDCON3 |
| TK0~TK23 | **3** | X | Touch Key (Hi-Z) | TKCHS<br>ATKCH2<br>ATKCH1<br>ATKCH0 |
| AD0~AD14 | **3** | X | ADC Channel | |
| PWM0~PWM2 | **0** | X | PWM Open Drain Output with Pull-up | PWMOE |
| | **1** | X | PWM Open Drain Output | |
| | **2** | X | PWM Output (CMOS Push-Pull) | |
| XI, XO | **0** | 1 | Crystal oscillation | CLKCON |
| I$^2$C Master SCL | **0** | X | I$^2$C Clock Output (Open Drain Output, Pull-up) | |
| | **1** | X | I$^2$C Clock Output (CMOS Push-Pull) | |
| I$^2$C Slave SCL | **1** | 1 | I$^2$C Clock Input (Hi-Z) | |
| I$^2$C Master/Slaver SDA | **0** | 1 | I$^2$C DATA (Pull-up) | |
| SPI Master Mode MISO | **1** | 1 | SPI Data Input | SPCON |
| SPI Master Mode SCK, MOSI | **2** | X | SPI Clock/Data Output (CMOS Push-Pull) | |
| SPI Slave Mode MISO | **2** | X | SPI Data Output (CMOS Push-Pull) | |
| SPI Slave Mode SCK, MOSI | **1** | 1 | SPI Clock/Data Input | |
| SS | **1** | 1 | SPI Chip Selection | |

**Mode Setting for Port1, Port2, Port3 Alternative Function**

or tables above, a “**CMOS Output**” pin means it can sink and drive at least 4 mA current. It is not recommended to use such pin as input function.

An “**Open Drain**” pin means it can sink at least 4 mA current but only drive a small current (<20 μA). It can be used as input or output function and typically needs an external pull up resistor.

An 8051 standard pin is a “**Pseudo Open Drain**” pin. It can sink at least 4 mA current when output is at low level, and drives at least 4 mA current for 1~2 clock cycle when output transits from low to high, then keeps driving a small current (<20 μA) to maintain the pin at high level. It can be used as input or output function.

***Note2:*** *for the necessary SFR setting above, LCD/LED pin has the highest priority. Therefore, if a pin is not used for Segment (ex: pin is I/O, ADC, TK, and SPI…), S/W must disable the LCD/LED function.*

| SFR **90h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **P1** | 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 |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

90h.7~0 **P1:** Port1 data

| SFR **A0h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **P2** | 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 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

A0h.1~0 **P2.7~P2.0:** P2.7~P2.0 data

| SFR **B0h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **P3** | 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 |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

B0h.7~0 **P3:** Port1 data

| SFR **D8h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **CLKCON** | SCKTYPE | FCKTYPE | STPSCK | STPPCK | STPFCK | SELFCK | CLKPSC | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Reset | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |

D8h.7 **SCKTYPE:** Set 1 to enable P2.0 and P2.1 pin’s crystal oscillation mode

D8h.6 **FCKTYPE:** Set 1 to enable P2.0 and P2.1 pin’s crystal oscillation mode

| SFR **A2h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **P1MODL** | P1MOD3 | | P1MOD2 | | P1MOD1 | | P1MOD0 | |
| R/W | R/W | | R/W | | R/W | | R/W | |
| Reset | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

A2h.7~6 **P1MOD3:** P1.3 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, P1.3 is ADC input

A2h.5~4 **P1MOD2:** P1.2 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, P1.2 is ADC input

A2h.3~2 **P1MOD1:** P1.1 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, P1.1 is ADC input

A2h.1~0 **P1MOD0:** P1.0 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, P1.0 is ADC input

| SFR **A3h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **P1MODH** | P1MOD7 | | P1MOD6 | | P1MOD5 | | P1MOD4 | |
| R/W | R/W | | R/W | | R/W | | R/W | |
| Reset | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

A3h.7~6 **P1MOD7:** P1.7 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3,

A3h.5~4 **P1MOD6:** P1.6 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3,

A3h.3~2 **P1MOD5:** P1.5 pin control.
00: Mode0
01: Mode1
10: Mode2
11: Mode3, P1.5 is ADC input

A3h.1~0 **P1MOD4:** P1.4 pin control.
00: Mode0
01: Mode1
10: Mode2
11: Mode3, P1.4 is ADC input

| SFR **A4h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **P3MODL** | P3MOD3 | | P3MOD2 | | P3MOD1 | | P3MOD0 | |
| R/W | R/W | | R/W | | R/W | | R/W | |
| Reset | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

A4h.7~6 **P3MOD3:** P3.3 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, P3.3 is ADC input

A4h.5~4 **P3MOD2:** P3.2 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, P3.2 is ADC input

A4h.3~2 **P3MOD1:** P3.1 pin control.
00: Mode0
01: Mode1
10: Mode2
11: Mode3, P3.1 is ADC input

A4h.1~0 **P3MOD0:** P3.0 pin control.
00: Mode0
01: Mode1
10: Mode2
11: Mode3, P3.0 is ADC input

| SFR **A5h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **P3MODH** | P3MOD7 | | P3MOD6 | | P3MOD5 | | P3MOD4 | |
| R/W | R/W | | R/W | | R/W | | R/W | |
| Reset | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |

A5h.7~6 **P3MOD7:** P3.7 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

A5h.5~4 **P3MOD6:** P3.6 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

A5h.3~2 **P3MOD5:** P3.5 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

A5h.1~0 **P3MOD4:** P3.4 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

| SFR **93h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **P2MOD** | – | – | – | – | P2MOD1 | | P2MOD0 | |
| R/W | – | – | – | – | R/W | | R/W | |
| Reset | – | – | – | – | 0 | 1 | 0 | 1 |

93h.3~2 **P2MOD1:** P2.1 pin control
00: Mode0
01: Mode1
10: Mode2
11: not defined

93h.1~0 **P2MOD0:** P2.0 pin control
00: Mode0
01: Mode1
10: Mode2
11: not defined

| SFR **A6h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PINMOD** | – | I2CSEL | TCOE | T2OE | – | – | – | T0OE |
| R/W | – | R/W | R/W | R/W | – | – | – | R/W |
| Reset | – | 0 | 0 | 0 | – | – | – | 0 |

A6h.5 **TCOE:** System clock signal output (CKO) control
0: Disable "System clock divided by 2" output to P1.4 pin
1: Enable "System clock divided by 2" output to P1.4 pin

A6h.4 **T2OE:** Timer2 signal output (T2O) control
0: Disable "Timer2 overflow divided by 2" output to P1.0 pin
1: Enable "Timer2 overflow divided by 2" output to P1.0 pin

A6h.0 **T0OE:** Timer0 signal output (T0O) control
0: Disable "Timer0 overflow divided by 64" output to P3.4 pin
1: Enable "Timer0 overflow divided by 64" output to P3.4 pin

| SFR **B1h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **LEDCON** | LEDEN | | LEDPSC | | LEDHOLD | LEDBRIT | | |
| R/W | R/W | | R/W | | R/W | R/W | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

B1h.7~6 **LEDEN:** LED BiD matrix mode
00: LED BiD matrix mode disable
01: LED 1/8 duty (COM0~3, SEG0~3), the LED pins' state will be controlled automatically
10: LED 1/9 duty (COM0~3, SEG0~4), the LED pins' state will be controlled automatically
11: LED 1/10 duty (COM0~3, SEG0~5), the LED pins' state will be controlled automatically

| SFR **BCh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **SPCON** | SPEN | MSTR | CPOL | CPHA | SSDIS | LSBF | SPCR | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

BCh.7 **SPEN:** SPI enable
0: SPI disable
1: SPI enable

BCh.3 **SSDIS:** SS pin disable
0: Enable SS pin
1: Disable SS pin

| SFR **F7h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **AUX2** | WDTE | | PWRSAV | VBGOUT | DIV32 | IAPTE | | MULDIV16 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

F7h.4 **VBGOUT:** Bandgap voltage output control
0: Disable
1: Bandgap voltage output to P3.2 pin, The additional condition VBGEN=1 (AEh.1) should be set.

| SFR **9Eh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWMOE** | PWM1IE | PWM0IE | – | – | – | PWM2OE | PWM1OE | PWM0OE |
| R/W | R/W | R/W | – | – | – | R/W | R/W | R/W |
| Reset | 0 | 0 | – | – | – | 0 | 0 | 0 |

9Eh.2 **PWM2OE:** PWM2 control
0: PWM2 disable
1: PWM2 enable and signal output to P1.6

9Eh.1 **PWM1OE:** PWM1 control
0: PWM1 disable
1: PWM1 enable and signal output to P1.3

9Eh. 0 **PWM0OE:** PWM0 control
0: PWM0 disable
1: PWM0 enable and signal output to P1.2

### 7.2 Port0

These pins are shared with TK, ADC and LCD/LED. If a Port0 is defined as I/O pin, it can be used as CMOS push-pull output or Schmitt-trigger input. The pin's pull up function is enable while SFR bit P0OE.n=0 and P0.n=1.

| Port0<br>pin function | P0OE.n | P0.n<br>SFR data | Pin State | Resistor<br>Pull-up | Digital<br>Input |
|---|---|---|---|---|---|
| Input | 0 | 0 | Hi-Z | N | Y |
| | 0 | 1 | Pull-up | Y | Y |
| CMOS Output | 1 | 0 | Drive Low | N | N |
| | 1 | 1 | Drive High | N | N |

**Port0 Pin Function Table**

| Pin Name | Wake-up | ADC | TK | LCD | LED BiD | LED Dot |
|---|---|---|---|---|---|---|
| P0.7 | | AD12 | TK19 | LCDC7 | | |
| P0.6 | | AD14 | TK18 | LCDC6 | | |
| P0.5 | | AD13 | TK17 | LCDC5 | | |
| P0.4 | | | TK16 | LCDC4 | | |
| P0.3 | | | CLD | LCDC3 | LEDC3 | LED3 |
| P0.2 | | | | LCDC2 | LEDC2 | LED2 |
| P0.1 | | | | LCDC1 | LEDC1 | LED1 |
| P0.0 | | | | LCDC0 | LEDC0 | LED0 |

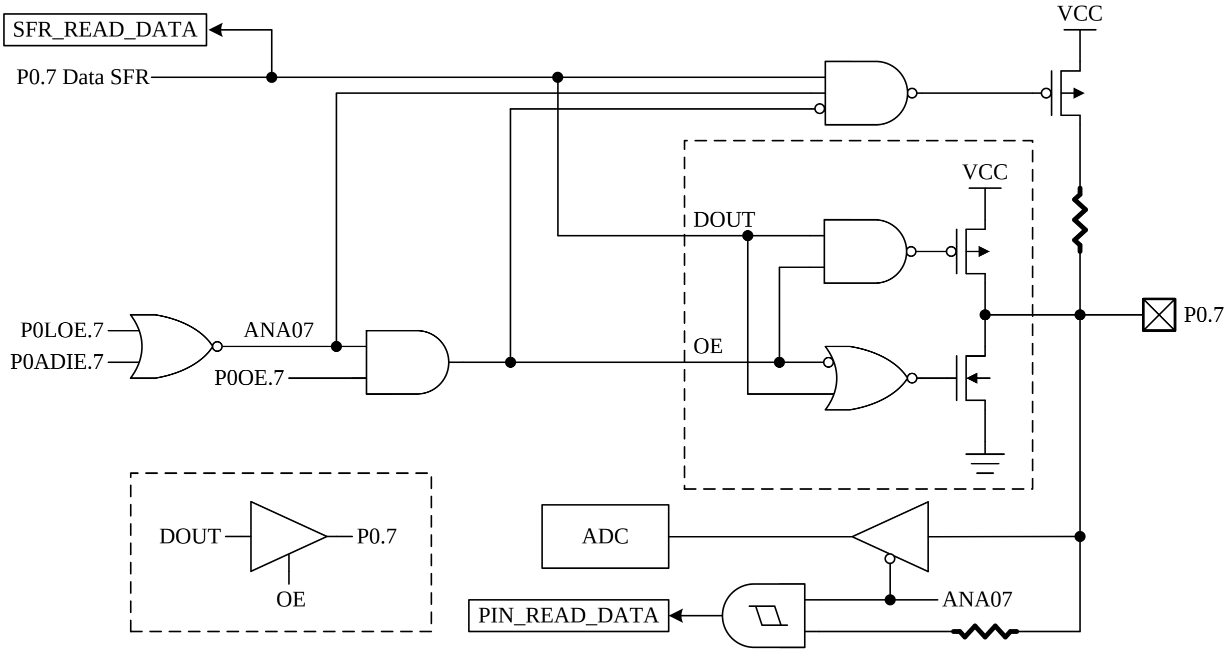
**Port0 multi-function Table**

The necessary SFR setting for Port0 pin's alternative function is list below.

| Alternative Function | PxOE.n | Px.n<br>SFR data | Pin State | other necessary<br>SFR setting |
|---|---|---|---|---|
| LEDC0~ LEDC3 | X | X | LED Bdi matrix mode Output | LEDCON |
| LED0~ LED3 | X | X | LED Dot matrix mode Output | LEDCON3 |
| LCDC0~ LCDC7 | X | X | 1/2 Bias Output | P0LOE |
| AD12~AD14 | X | X | ADC Channel | P0ADIE |
| CLD | 0 | 0 | Touch Key Capacitor Connection | |
| TK16~TK19 | **0** | 0 | Touch Key (CMOS output high) | TKCHS |

**Mode Setting for Port0 Alternative Function Table**

***Note:*** *P0LOE and P0ADIE have higher priority than P0OE.*



**P0.7 Pin Structure**

| SFR 80h | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **P0** | 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 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

80h.7~0 **P0:** Port0 data, also controls the P0.n pin’s pull-up function. If the P0.n SFR data is “1” and the corresponding P0OE.n = 0 (input mode), the pull-up is enabled.

| SFR 91h | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **P0OE** | P0OE | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

91h.7~0 **P0OE:** Port0 CMOS Push-Pull output enable control
0: Disable
1: Enable

| SFR 92h | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **P0LOE** | P0LOE | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

92h.7~0 **P0LOE:** Port0 LCD 1/2 bias output enable control
0: Disable
1: Enable

| SFR **AFh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **P0ADIE** | P0ADIE | | | – | – | – | – | – |
| R/W | R/W | | | – | – | – | – | – |
| Reset | 0 | 0 | 0 | – | – | – | – | – |

AFh.7~5 **P0ADIE:** ADC channel input Enable

000: P0.7~P0.5 are digital input
1xx: P0.7 is ADC input
x1x: P0.6 is ADC input
xx1: P0.5 is ADC input

| SFR **B1h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **LEDCON** | LEDEN | | LEDPSC | | LEDHOLD | LEDBRIT | | |
| R/W | R/W | | R/W | | R/W | R/W | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

B1h.7~6 **LEDEN:** LED BiD matrix mode Enable

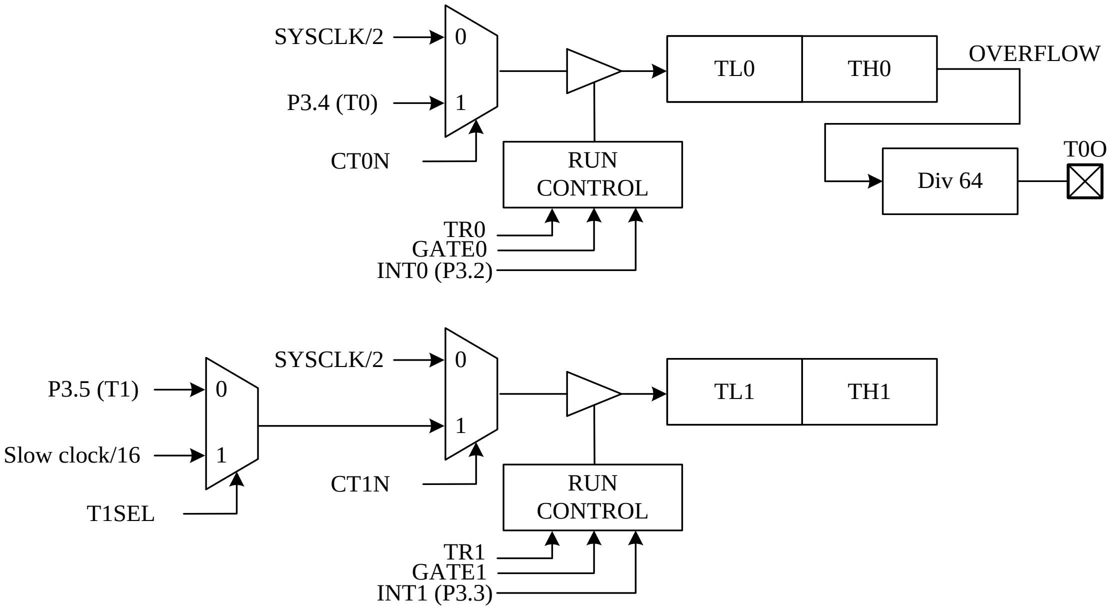
00: LED BiD matrix mode disable
01: LED 1/8 duty (COM0~3, SEG0~3), the LED pins’ state will be controlled automatically
10: LED 1/9 duty (COM0~3, SEG0~4) , the LED pins’ state will be controlled automatically
11: LED 1/10 duty (COM0~3, SEG0~5) , the LED pins’ state will be controlled automatically

# 8. Timers

Timer0, Timer1 and Timer2 are provided as standard 8051 compatible timer/counter. Compare to the traditional 12T 8051, the Chip's Timer0/1/2 use 2 System clock cycle as the time base unit. That is, in timer mode, these timers increase at every “2 System clock” rate; in counter mode, T0/T1/T2 pin input pulse must be wider than 2 System clock to be seen by this device. In addition to the standard 8051 timers function. The T0O pin can output the “Timer0 overflow divided by 64” signal, and the T2O pin can output the “Timer2 overflow divided by 2” signal. Timer3 is provided for a real-time clock count, when its time base is SXT.

## 8.1 Timer0 / Timer1

TCON and TMOD are used to set the mode of operation and to control the running and interrupt generation of the Timer0/1, with the timer/counter values stored in two pairs of 8-bit registers (TL0, TH0, and TL1, TH1).



**Timer0 and Timer1 Structure**

| SFR **88h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TCON** | TF1 | TR1 | TF0 | TR0 | IE1 | IT1 | IE0 | IT0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

88h.7 **TF1:** Timer1 overflow flag
Set by H/W when Timer/Counter 1 overflows
Cleared by H/W when CPU vectors into the interrupt service routine.

88h.6 **TR1:** Timer1 run control
0: Timer1 stops
1: Timer1 runs

88h.5 **TF0:** Timer0 overflow flag
Set by H/W when Timer/Counter 0 overflows
Cleared by H/W when CPU vectors into the interrupt service routine.

88h.4 **TR0:** Timer0 run control
0: Timer0 stops
1: Timer0 runs

| SFR **89h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TMOD** | GATE1 | CT1N | TMOD1 | | GATE0 | CT0N | TMOD0 | |
| R/W | R/W | R/W | R/W | | R/W | R/W | R/W | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

89h.7 **GATE1:** Timer1 gating control bit
0: Timer1 enable when TR1 bit is set
1: Timer1 enable only while the INT1 pin is high and TR1 bit is set

89h.6 **CT1N:** Timer1 Counter/Timer select bit
0: Timer mode, Timer1 data increases at 2 System clock cycle rate
1: Counter mode, Timer1 data increases at T1 pin's negative edge

89h.5~4 **TMOD1:** Timer1 mode select
00: 13-bit timer/counter
01: 16-bit timer/counter
10: 8-bit auto-reload timer/counter (TL1). Reloaded from TH1 at overflow.
11: Timer1 stops

89h.3 **GATE0:** Timer0 gating control bit
0: Timer0 enable when TR0 bit is set
1: Timer0 enable only while the INT0 pin is high and TR0 bit is set

89h.2 **CT0N:** Timer0 Counter/Timer select bit
0: Timer mode, Timer0 data increases at 2 System clock cycle rate
1: Counter mode, Timer0 data increases at T0 pin's negative edge

89h.1~0 **TMOD0:** Timer0 mode select
00: 13-bit timer/counter
01: 16-bit timer/counter
10: 8-bit auto-reload timer/counter (TL0). Reloaded from TH0 at overflow.
11: TL0 is an 8-bit timer/counter. TH0 is an 8-bit timer/counter using Timer1's TR1 and TF1 bits.

| SFR **8Ah** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TL0** | TL0 | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

8Ah.7~0 **TL0:** Timer0 data low byte

| SFR **8Bh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TL1** | TL1 | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

8Bh.7~0 **TL1:** Timer1 data low byte

| SFR **8Ch** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TH0** | TH0 | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

8Ch.7~0 **TH0:** Timer0 data high byte

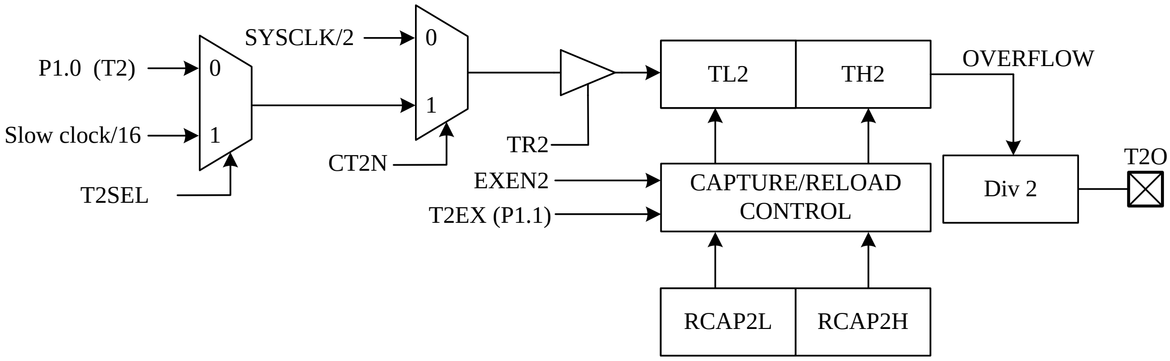
| SFR **8Dh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TH1** | TH1 | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

8Dh.7~0 **TH1:** Timer1 data high byte

***Note****: See also Chapter 6 for more information on Timer0/1 interrupt enable and priority.*
***Note****: See also Chapter 7 for details on T0O pin output settings.*

## 8.2 Timer2

Timer2 is controlled through the TCON2 register with the low and high bytes of Timer/Counter2 stored in TL2 and TH2 and the low and high bytes of the Timer2 reload/capture registers stored in RCAP2L and RCAP2H.



**Timer2 Structure**

| SFR **C8h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **T2CON** | TF2 | EXF2 | RCLK | TCLK | EXEN2 | TR2 | CT2N | CPRL2N |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

C8h.7 **TF2:** Timer2 overflow flag
Set by H/W when Timer/Counter 2 overflows unless RCLK=1 or TCLK=1. This bit must be cleared by S/W.

C8h.6 **EXF2:** T2EX interrupt pin falling edge flag
Set when a capture or a reload is caused by a negative transition on T2EX pin if EXEN2=1. This bit must be cleared by S/W.

C8h.5 **RCLK:** UART receive clock control bit
0: Use Timer1 overflow as receive clock for serial port in mode 1 or 3
1: Use Timer2 overflow as receive clock for serial port in mode 1 or 3

C8h.4 **TCLK:** UART transmit clock control bit
0: Use Timer1 overflow as transmit clock for serial port in mode 1 or 3
1: Use Timer2 overflow as transmit clock for serial port in mode 1 or 3

C8h.3 **EXEN2:** T2EX pin enable
0: T2EX pin disable
1: T2EX pin enable, it cause a capture or reload when a negative transition on T2EX pin is detected if RCLK=TCLK=0

C8h.2 **TR2:** Timer2 run control
0: Timer2 stops
1: Timer2 runs

C8h.1 **CT2N:** Timer2 Counter/Timer select bit
0: Timer mode, Timer2 data increases at 2 System clock cycle rate
1: Counter mode, Timer2 data increases at T2 pin’s negative edge

C8h.0 **CPRL2N:** Timer2 Capture/Reload control bit
0: Reload mode, auto-reload on Timer2 overflows or negative transitions on T2EX pin if EXEN2=1.
1: Capture mode, capture on negative transitions on T2EX pin if EXEN2=1.
If RCLK=1 or TCLK=1, CPRL2N is ignored and timer is forced to auto-reload on Timer2 overflow.

| SFR **CAh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **RCP2L** | RCP2L | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

CAh.7~0 **RCP2L:** Timer2 reload/capture data low byte

| SFR **CBh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **RCP2H** | RCP2H | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

CBh.7~0 **RCP2H:** Timer2 reload/capture data high byte

| SFR **CCh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TL2** | TL2 | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

CCh.7~0 **TL2:** Timer2 data low byte

| SFR **CDh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TH2** | TH2 | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

CDh.7~0 **TH2:** Timer2 data high byte

| SFR **F8h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **AUX1** | CLRWDT | CLRTM3 | TKSOC | ADSOC | LVRPD | T2SEL | T1SEL | DPSEL |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

F8h.2 **T2SEL:** Timer2 counter mode (CT2N=1) input select
0: P1.0 (T2) pin (8051standard)
1:Slow clock divide by 16 (SLOWCLK/16)

F8h.1 **T1SEL:** Timer1 counter mode (CT1N=1) input select
0: P3.5 (T1) pin (8051 standard)
1: Slow clock divide by 16 (SLOWCLK/16)

***Note****: See also Chapter 6 for more information on Timer2 interrupt enable and priority.*
***Note****: See also Chapter 7 for details on T2O pin output settings.*

## 8.3 Timer3

Timer3 works as a time-base counter, which generates interrupts periodically. It generates an interrupt flag (TF3) with the clock divided by 32768, 16384, 8192, or 128 depending on the TM3PSC SFR. The Timer3 clock source is Slow clock (SRC or SXT). This is ideal for real-time-clock (RTC) functionality when the clock source is SXT.

| SFR **94h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **OPTION** | UART1W | – | WDTPSC | | ADCKS | | TM3PSC | |
| R/W | R/W | – | R/W | | R/W | | R/W | |
| Reset | 0 | – | 0 | 0 | 0 | 0 | 0 | 0 |

94h.1~0 **TM3PSC:** Timer3 Interrupt rate
00: Timer3 Interrupt rate is 32768 Slow clock cycle
01: Timer3 Interrupt rate is 16384 Slow clock cycle
10: Timer3 Interrupt rate is 8192 Slow clock cycle
11: Timer3 Interrupt rate is 128 Slow clock cycle

| SFR **95h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **INTFLG** | LVDIF | – | TKIF | ADIF | – | – | P1IF | TF3 |
| R/W | R | – | R/W | R/W | – | – | R/W | R/W |
| Reset | – | – | 0 | 0 | – | – | 0 | 0 |

95h.0 **TF3:** Timer3 Interrupt Flag
Set by H/W when Timer3 reaches TM3PSC setting cycles. Cleared automatically when the program performs the interrupt service routine. S/W can write FEh to INTFLG to clear this bit. (***Note1***)

| SFR **F8h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **AUX1** | CLRWDT | CLRTM3 | TKSOC | ADSOC | LVRPD | T2SEL | T1SEL | DPSEL |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

F8h.6 **CLRTM3:** Set 1 to clear Timer3, H/W auto clear it at next clock cycle.

***Note:*** *also refer to Section 6 for more information about Timer3 Interrupt enable and priority.*

## 8.4 T0O and T2O Output Control

This device can generate various frequency waveform pin output (in CMOS or Open-Drain format) for Buzzer. The T0O and T2O waveform is divided by Timer0/Timer2 overflow signal. The T0O waveform is Timer0 overflow divided by 64, and T2O waveform is Timer2 overflow divided by 2. User can control their frequency by Timers auto reload speed. Set T0OE and T2OE SFRs can output these waveforms.

| SFR **A6h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PINMOD** | – | I2CSEL | TCOE | T2OE | – | – | – | T0OE |
| R/W | – | R/W | R/W | R/W | – | – | – | R/W |
| Reset | – | 0 | 0 | 0 | – | – | – | 0 |

A6h.4 **T2OE:** Timer2 signal output (T2O) control
0: Disable Timer2 overflow divided by 2 output to P1.0
1: Enable Timer2 overflow divided by 2 output to P1.0

A6h.0 **T0OE:** Timer0 signal output (T0O) control
0: Disable Timer0 overflow divided by 64 output to P3.4
1: Enable Timer0 overflow divided by 64 output to P3.4

## 9. UARTs

This Chip has two UARTs, UART1 and UART2.

The **UART1** uses SCON and SBUF SFRs. SCON is the control register, SBUF is the data register. Data is written to SBUF for transmission and SBUF is read to obtain received data. The received data and transmitted data registers are completely independent. In addition to standard 8051's full duplex mode, this chip also provides one wire mode. If the UART1W bit is set, both transmit and receive data use P3.1 pin.

The **UART2** uses SCON2 and SBUF2 SFRs. SCON2 is the control register, SBUF2 is the data register. Data is written to SBUF2 for transmission and SBUF2 is read to obtain received data. The received data and transmitted data registers are completely independent. The UART2 supports most of the functions of UART, but it does not support Mode0 and Mode2, it also does not support Timer2 and one wire UART mode. On other hand, the option of SMOD is not use for UART2. UART2 double baud rate is always enabled.

| SFR **87h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PCON** | SMOD | – | – | – | GF1 | GF0 | PD | IDL |
| R/W | R/W | – | – | – | R/W | R/W | R/W | R/W |
| Reset | 0 | – | – | – | 0 | 0 | 0 | 0 |

87h.7 **SMOD:** UART1 double baud rate control bit
0: Disable UART1 double baud rate
1: Enable UART1 double baud rate

| SFR **94h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **OPTION** | UART1W | – | WDTPSC | | ADCKS | | TM3PSC | |
| R/W | R/W | – | R/W | | R/W | | R/W | |
| Reset | 0 | – | 0 | 0 | 0 | 0 | 0 | 0 |

94h.7 **UART1W:** One wire UART1 mode enable, both TXD/RXD use P3.1 pin
0: Disable one wire UART1 mode
1: Enable one wire UART1 mode

| SFR **98h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **SCON** | 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 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

98h.7~6 **SM0,SM1:** UART1 serial port mode select bit 0,1

00: Mode0: 8 bit shift register, Baud Rate=$F_{SYSCLK}$/2

01: Mode1: 8 bit UART1, Baud Rate is variable

10: Mode2: 9 bit UART1, Baud Rate=$F_{SYSCLK}$/32 or/64

11: Mode3: 9 bit UART1, Baud Rate is variable

98h.5 **SM2:** UART1 Serial port mode select bit 2

SM2 enables multiprocessor communication over a single serial line and modifies the above as follows. In Modes 2 & 3, if SM2 is set then the received interrupt will not be generated if the received ninth data bit is 0. In Mode 1, the received interrupt will not be generated unless a valid stop bit is received. In Mode 0, SM2 should be 0.

98h.4 **REN:** UART1 reception enable

0: Disable reception

1: Enable reception

98h.3 **TB8:** UART1 Transmit Bit 8, the ninth bit to be transmitted in Mode 2 and 3

98h.2 **RB8:** UART1 Receive Bit 8, contains the ninth bit that was received in Mode 2 and 3 or the stop bit is Mode 1 if SM2=0

98h.1 **TI:** UART1 Transmit interrupt flag

Set by H/W at the end of the eighth bit in Mode 0, or at the beginning of the stop bit in other modes. Must be cleared by S/W.

98h.0 **RI:** UART1 Receive interrupt flag

Set by H/W at the end of the eighth bit in Mode 0, or at the sampling point of the stop bit in other modes. Must be cleared by S/W.

| SFR **99h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **SBUF** | SBUF | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | – | – | – | – | – | – | – | – |

99h.7~0 **SBUF:** UART1 transmit and receive data. Transmit data is written to this location and receive data is read from this location, but the paths are independent.

| SFR **8Eh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **SCON2** | SM | – | – | REN2 | TB82 | RB82 | TI2 | RI2 |
| R/W | R/W | – | – | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | – | – | 0 | 0 | 0 | 0 | 0 |

8Eh.7 **SM:** UART2 Serial port mode select bit
0: Mode1: 8 bit UART2, Baud Rate is variable
1: Mode3: 9 bit UART2, Baud Rate is variable
**(UART2 does not support Mode0/Mode2)**

8Eh.4 **REN2:** UART2 reception enable
0: Disable reception
1: Enable reception

8Eh.3 **TB82:** UART2 Transmit Bit 8, the ninth bit to be transmitted in Mode 3

8Eh.2 **RB82:** UART2 Receive Bit 8, contains the ninth bit that was received in Mode3

8Eh.1 **TI2:** UART2 Transmit interrupt flag
Set by H/W at the beginning of the stop bit in Mode 1 & 3. Must be cleared by S/W.

8Eh.0 **RI2:** UART2 Receive interrupt flag
Set by H/W at the sampling point of the stop bit in Mode 1 & 3. Must be cleared by S/W.

| SFR **8Fh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **SBUF2** | SBUF2 | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | – | – | – | – | – | – | – | – |

8Fh.7~0 **SBUF2:** UART2 transmit and receive data. Transmit data is written to this location and receive data is read from this location, but the paths are independent.

| SFR **A8h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **IE** | EA | – | ET2 | ES | ET1 | EX1 | ET0 | EX0 |
| R/W | R/W | – | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | – | 0 | 0 | 0 | 0 | 0 | 0 |

A8h.4 **ES:** Serial Port (UART1) interrupt enable
0: Disable Serial Port (UART1) interrupt
1: Enable Serial Port (UART1) interrupt

$F_{SYSCLK}$ denotes System clock frequency, the UART baud rate is calculated as below.

- **Mode 0: (UART2 invalid)**
  Baud Rate=$F_{SYSCLK}$/2
- **Mode 1, 3:** if using Timer1 auto reload mode
  Baud Rate= (SMOD + 1) x $F_{SYSCLK}$/ (32 x 2 x (256 – TH1))
- **Mode 1, 3:** if using Timer2 **(UART2 invalid)**
  Baud Rate=Timer2 overflow rate/16 = $F_{SYSCLK}$/ (32 x (65536 – RCP2H, RCP2L))
- **Mode 2: (UART2 invalid)**
  Baud Rate= (SMOD + 1) x $F_{SYSCLK}$/64

***Note:*** *also refer to Section 6 for more information about UART Interrupt enable and priority.*
***Note:*** *also refer to Section 8 for more information about how Timer2 controls UART clock.*

# 10. PWMs

## 10.1 16-bit PWM

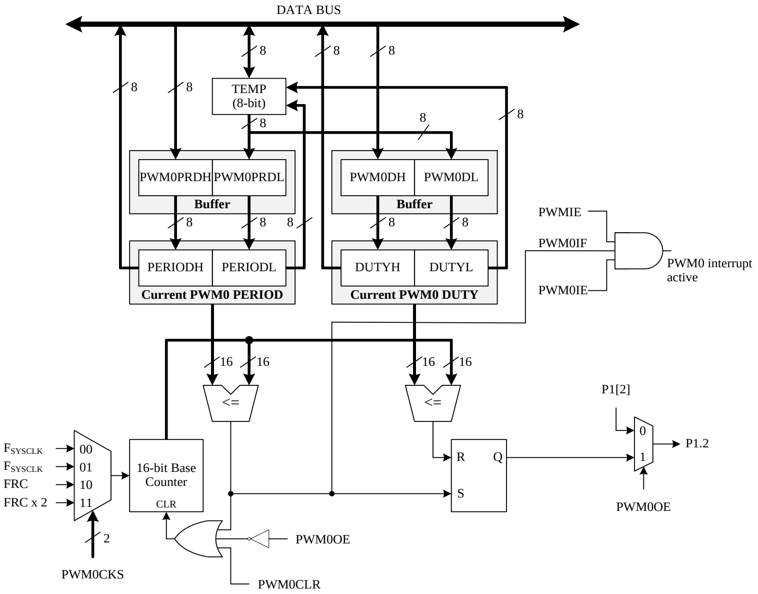
The Chip has three independent 16-bit PWM modules PWM0, PWM1 and PWM2. PWM0~2 have the same operation structure. The following takes PWM0 as an example for description. The PWM can generate varies frequency waveform with 65536 duty resolution on the basis of the PWM clock. The PWM clock can select FRC double frequency (FRC x 2), FRC or $F_{SYSCLK}$ as its clock source.

The pin mode SFR controls the PWM output waveform format. Mode1 makes the PWM open drain output and Mode2 makes the PWM CMOS push-pull output. (***see section 7***)

The 16-bit PWM0PRD, PWM0D registers all have a low byte and high byte structure. The high bytes can be directly accessed, but the low bytes can only be accessed via an internal 8-bit buffer, reading or writing to these register pairs must be carried out in a specific way. The important point to notes is that data transfer to and from the 8-bit buffer and its related low byte only takes place when write or read operation to its corresponding high bytes is executed. ***Briefly speaking, write low byte first and then high byte; read high byte first and then low byte***.

The PWM0OE bit is used to select the output to PWM0. If PWM0OE are cleared, the PWM0 will be cleared and stopped, otherwise the PWM0 is running. The PWM0CLR bit has the same function. When PWM0CLR bit is set, the PWM0 will be cleared and held, otherwise the PWM0 is running. The PWM0 structure is shown as follow. The PWM0 duty cycle can be changed by writing to PWM0DH and PWM0DL. The PWM0 output signal resets to a low level whenever the 16-bit base counter matches the 16-bit PWM0 duty register {PWM0DH, PWM0DL}. The PWM0 period can be set by writing the period value to the PWM0PRDH and PWM0PRDL registers. After writing the PWM0D or PWM0PRD register, the new values will immediately save to their own buffer. H/W will update these values at the end of current period or while PWM0 is cleared. PWM0~2 has a corresponding interrupt flag, and an interrupt flag is generated at the end of the period.

PWMxDH, PWMxDL, PWMxPRDH or PWMxPRDL is a 16-bit operation, and the program should avoid interrupts when writing and reading the high byte and low byte. If you are reading and writing these 16-bit SFRs in the meantime an interrupt occurs. And these SFRs are read and written in the interrupt. It is easy to cause read and write errors. For the 16-bit PWM period and duty to read and write, it is recommended to update the data only in the main program, or update the data only in the interrupt to avoid possible errors.



**PWM0 Structure**

| SFR **86h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **INTPWM** | – | – | – | – | – | PWM2IF | PWM1IF | PWM0IF |
| R/W | – | – | – | – | – | R/W | R/W | R/W |
| Reset | – | – | – | – | – | 0 | 0 | 0 |

86h.2 **PWM2IF:**
0: S/W write 0 to clear it
1: Set by H/W at the end of the period

86h.1 **PWM1IF:**
0: S/W write 0 to clear it
1: Set by H/W at the end of the period

86h.0 **PWM0IF:**
0: S/W write 0 to clear it
1: Set by H/W at the end of the period

| SFR **9Eh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWMOE** | PWM1IE | PWM0IE | – | – | – | PWM2OE | PWM1OE | PWM0OE |
| R/W | R/W | R/W | – | – | – | R/W | R/W | R/W |
| Reset | 0 | 0 | – | – | – | 0 | 0 | 0 |

9Eh.7 **PWM1IE:** PWM1 Interrupt Enable
0: disable
1: enable (note: PWMIE must be 1 at the same time to generate PWM interrupt)

9Eh.6 **PWM0IE:** PWM0 Interrupt Enable
0: disable
1: enable (note: PWMIE must be 1 at the same time to generate PWM interrupt)

9Eh.2 **PWM2OE:**
0: disable 1: PWM2 enable and signal output to P1.6 pin

9Eh.1 **PWM1OE:**
0: disable 1: PWM1 enable and signal output to P1.3 pin

9Eh.0 **PWM0OE:**
0: disable 1: PWM0 enable and signal output to P1.2 pin

| SFR **9Fh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWMCLR** | PWM2IE | – | – | – | – | PWM2CLR | PWM1CLR | PWM0CLR |
| R/W | R/W | – | – | – | – | R/W | R/W | R/W |
| Reset | 0 | – | – | – | – | 0 | 0 | 0 |

9Fh.7 **PWM2IE:** PWM2 Interrupt Enable
0: disable
1: enable (note: PWMIE must be 1 at the same time to generate PWM interrupt)

9Fh.2 **PWM2CLR:**
0: PWM2 is running 1: PWM2 is cleared and held

9Fh.1 **PWM1CLR:**
0: PWM1 is running 1: PWM1 is cleared and held

9Fh.0 **PWM0CLR:**
0: PWM0 is running 1: PWM0 is cleared and held

| SFR **A1h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWMCON** | – | – | PWM2CKS | | PWM1CKS | | PWM0CKS | |
| R/W | – | – | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | – | – | 1 | 0 | 1 | 0 | 1 | 0 |

A1h.5~4 **PWM2CKS:** PWM2 Clock source

00: $F_{SYSCLK}$
01: $F_{SYSCLK}$
10: FRC
11: FRC x 2

A1h.3~2 **PWM1CKS:** PWM1 Clock source

00: $F_{SYSCLK}$
01: $F_{SYSCLK}$
10: FRC
11: FRC x 2

A1h.1~0 **PWM0CKS:** PWM0 Clock source

00: $F_{SYSCLK}$
01: $F_{SYSCLK}$
10: FRC
11: FRC x 2

| SFR **A9h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **INTE1** | PWMIE | I2CE | ES2 | SPIE | ADTKIE | EXLVDIE | P1IE | TM3IE |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

A9h.6 **PWMIE:** PWM0~2 interrupt enable

0: Disable PWM0~2 interrupt
1: Enable PWM0~2 interrupt

| SFR **D1h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWM0DH** | PWM0DH | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

D1h.7~0 **PWM0DH:** PWM0 duty high byte
write sequence: PWM0DL then PWM0DH
read sequence: PWM0DH then PWM0DL

| SFR **D2h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWM0DL** | PWM0DL | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

D2h.7~0 **PWM0DL:** PWM0 duty low byte
write sequence: PWM0DL then PWM0DH
read sequence: PWM0DH then PWM0DL

| SFR **D3h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWM1DH** | PWM1DH | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

D3h.7~0 **PWM1DH:** PWM1 duty high byte
write sequence: PWM1DL then PWM1DH
read sequence: PWM1DH then PWM1DL

| SFR **D4h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWM1DL** | PWM1DL | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

D4h.7~0 **PWM1DL:** PWM1 duty low byte
write sequence: PWM1DL then PWM1DH
read sequence: PWM1DH then PWM1DL

| SFR **D5h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWM2DH** | PWM2DH | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

D5h.7~0 **PWM2DH:** PWM2 duty high byte
write sequence: PWM2DL then PWM2DH
read sequence: PWM2DH then PWM2DL

| SFR **D6h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWM2DL** | PWM2DL | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

D6h.7~0 **PWM2DL:** PWM2 duty low byte
write sequence: PWM2DL then PWM2DH
read sequence: PWM2DH then PWM2DL

| SFR **D9h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWM0PRDH** | PWM0PRDH | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

D9h.7~0 **PWM0PRDH:** PWM0 period high byte
write sequence: PWM0PRDL then PWM0PRDH
read sequence: PWM0PRDH then PWM0PRDL

| SFR **DAh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWM0PRDL** | PWM0PRDL | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

DAh.7~0 **PWM0PRDL:** PWM0 period low byte
write sequence: PWM0PRDL then PWM0PRDH
read sequence: PWM0PRDH then PWM0PRDL

| SFR **DBh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWM1PRDH** | PWM1PRDH | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

DBh.7~0 **PWM1PRDH:** PWM1 period high byte
write sequence: PWM1PRDL then PWM1PRDH
read sequence: PWM1PRDH then PWM1PRDL

| SFR **DCh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWM1PRDL** | PWM1PRDL | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

DCh.7~0 **PWM1PRDL:** PWM1 period low byte
write sequence: PWM1PRDL then PWM1PRDH
read sequence: PWM1PRDH then PWM1PRDL

| SFR **DDh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWM2PRDH** | PWM2PRDH | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

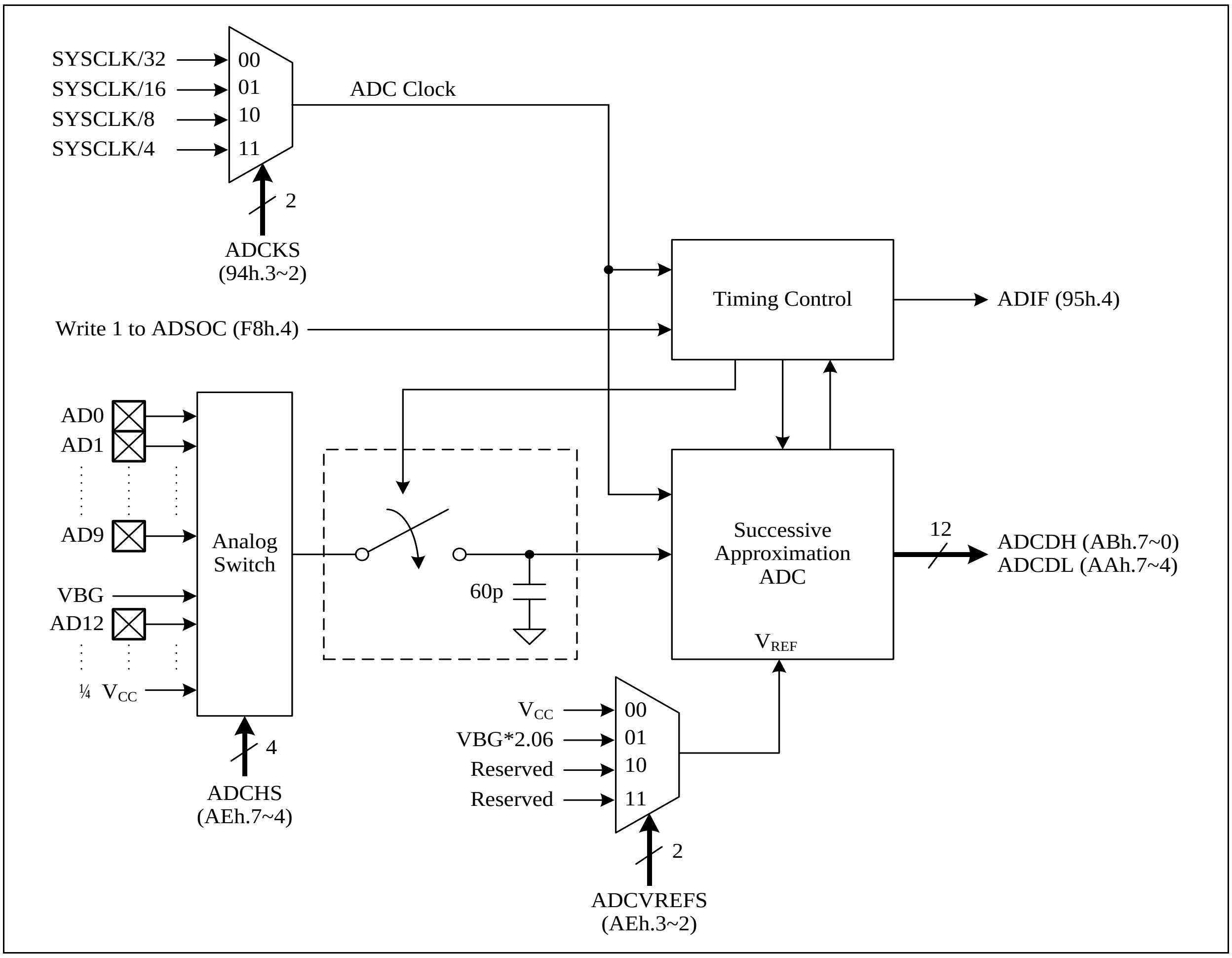
DDh.7~0 **PWM2PRDH:** PWM2 period high byte
write sequence: PWM2PRDL then PWM2PRDH
read sequence: PWM2PRDH then PWM2PRDL

| SFR **DEh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWM2PRDL** | PWM2PRDL | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

DEh.7~0 **PWM2PRDL:** PWM2 period low byte
write sequence: PWM2PRDL then PWM2PRDH
read sequence: PWM2PRDH then PWM2PRDL

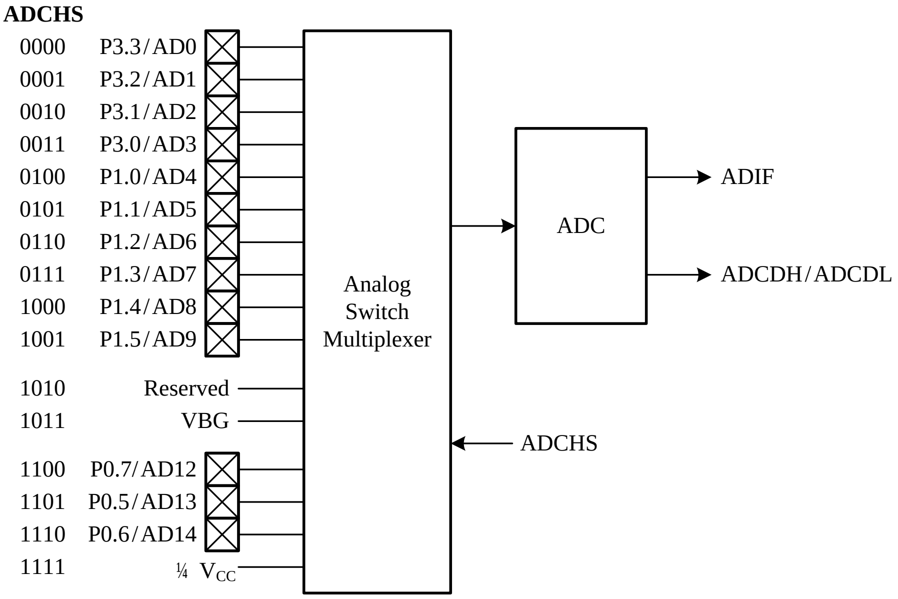
## 11. ADC

The Chip offers a 12-bit ADC consisting of a 16-channel analog input multiplexer, control register, clock generator, 12-bit successive approximation register, and output data register. To use the ADC, set the ADCKS bit first to choose a proper ADC clock frequency, which must be less than 1 MHz. Then, launch the ADC conversion by setting the ADSOC bit, and H/W will automatic clear it at the end of the conversion. After the end of the conversion, H/W will set the ADIF bit and generate an interrupt if an ADC interrupt is enabled. The ADIF bit can be cleared by writing 0 to this bit or 1 to the ADSOC bit. Because certain channels are shared with the Touch Key, the ADC channel must be configured differently from the Touch Key channel to avoid affecting the channel input sensitivity. The VREF of the ADC can be selected from the following two voltages: $V_{CC}$ and VBG*2.06V. When ADCHS is selected to VBG, ADCVREFS must be set to $V_{CC}$, otherwise ADC conversion will be invalid.



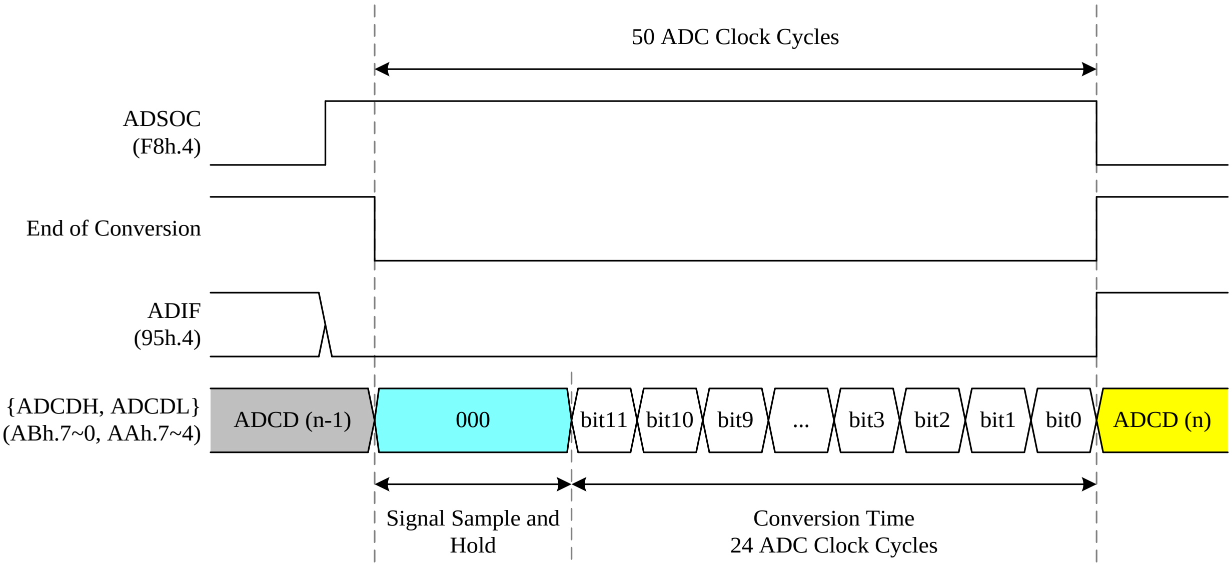
## 11.1 ADC Channels

The 12-bit ADC has a total of 16 channels, designated AD0~AD9, AD12~AD14, VBG and 1/4V$_{CC}$. The ADC channels are connected to the analog input pins via the analog switch multiplexer. The analog switch multiplexer is controlled by the ADCHS register. VBG is an internal voltage reference at 1.27V. When ADC channel select to VBG, VBG generator will enable automatically. User can get more stable VBG voltage by setting SFR VBGEN=1 to always enable VBG generator. When ADCHS is selected to VBG, ADCVREFS must be set to V$_{CC}$, otherwise ADC conversion will be invalid.



## 11.2 ADC Conversion Time

The conversion time is the time required for the ADC to convert the voltage. The ADC requires two ADC clock cycles to convert each bit and several clock cycles to sample and hold the input voltage. A total of 50 ADC clock cycles are required to perform the complete conversion. When the conversion time is complete, the ADIF interrupt flag is set by H/W, and the result is loaded into the ADCDH and ADCDL registers of the 12-bit A/D result.



| SFR **94h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **OPTION** | UART1W | – | WDTPSC | | ADCKS | | TM3PSC | |
| R/W | R/W | – | R/W | | R/W | | R/W | |
| Reset | 0 | – | 0 | 0 | 0 | 0 | 0 | 0 |

94h.3~2 **ADCKS:** ADC clock rate select

00: $F_{SYSCLK}/32$
01: $F_{SYSCLK}/16$
10: $F_{SYSCLK}/8$
11: $F_{SYSCLK}/4$

| SFR **95h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **INTFLG** | LVDIF | – | TKIF | ADIF | – | – | P1IF | TF3 |
| R/W | R | – | R/W | R/W | – | – | R/W | R/W |
| Reset | – | – | 0 | 0 | – | – | 0 | 0 |

95h.4 **ADIF:** ADC interrupt flag

Set by H/W at the end of ADC conversion. S/W writes EFh to INTFLG or sets the ADSOC bit to clear this flag. (***Note1***)

| SFR **AAh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **ADCDL** | ADCDL | | | | – | | | |
| R/W | R | | | | – | | | |
| Reset | – | – | – | – | – | – | – | – |

AAh.7~4 **ADCDL:** ADC data bit 3~0

| SFR **ABh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **ADCDH** | ADCDH | | | | | | | |
| R/W | R | | | | | | | |
| Reset | – | – | – | – | – | – | – | – |

ABh.7~0 **ADCDH:** ADC data bit 11~4

| SFR **AEh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **CHSEL** | ADCHS | | | | ADCVREFS | | VBGEN | – |
| R/W | R/W | | | | R/W | R/W | R/W | – |
| Reset | 1 | 1 | 1 | 1 | 0 | 0 | 0 | – |

AEh.7~4 **ADCHS:** ADC channel select

0000: AD0 (P3.3)
0001: AD1 (P3.2)
0010: AD2 (P3.1)
0011: AD3 (P3.0)
0100: AD4 (P1.0)
0101: AD5 (P1.1)
0110: AD6 (P1.2)
0111: AD7 (P1.3)
1000: AD8 (P1.4)
1001: AD9 (P1.5)
1010: Reserved
1011: $V_{BG}$ (Internal Bandgap Reference Voltage)
1100: AD12 (P0.7)
1101: AD13 (P0.5)
1110: AD14 (P0.6)
1111: 1/4 $V_{CC}$

AEh.3~2 **ADCVREFS:** ADC reference voltage. When ADCHS is selected to VBG, ADCVREFS must be set to VCC, otherwise ADC conversion will be invalid

00: VCC
01: VBG*2.06V
10: Reserved
11: Reserved

AEh.1 **VBGEN:** force VBG generator enable

0: VBG generator is automatically enable and disable
1: Force VBG generator enable included in IDLE mode but disabled in Stop/Halt mode

| SFR **F8h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **AUX1** | CLRWDT | CLRTM3 | TKSOC | ADSOC | LVRPD | T2SEL | T1SEL | DPSEL |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

F8h.4 **ADSOC:** Start ADC conversion

Set the ADSOC bit to start ADC conversion, and the ADSOC bit will be cleared by H/W at the end of conversion. S/W can also write 0 to clear this flag.

## 12. Touch Key (FTK)

The Touch Key offers an easy simple and reliable method to implement finger touch detection. During the key scan operation, the device support 20 channels touch key detection.

To use the Touch Key, user should setup correctly. There are two ways to set IO as TK channel. Set SFR PxMODx to 11b or set SFR TKPINSEL0~2 to force IO as TK channel. If TKPINSEL0~2 are set, the corresponding IO pins will be fixed as TK channels and will no longer be affected by PxNMODx.

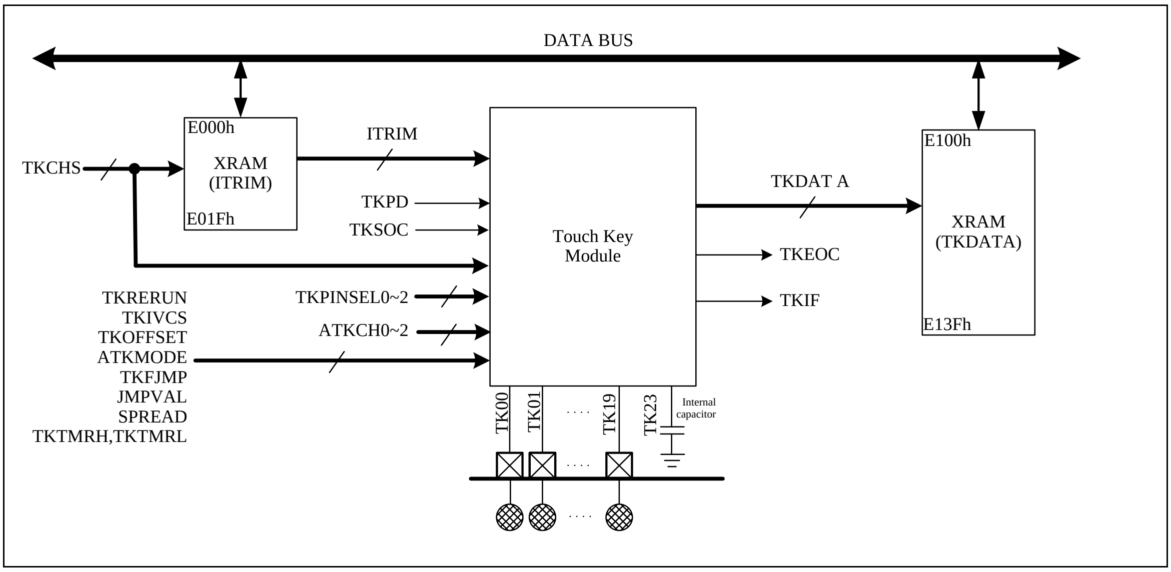
| TKPINSEL | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TKPINSEL0** | TK07 | TK06 | TK05 | TK04 | TK03 | TK02 | TK01 | TK00 |
| **TKPINSEL1** | TK15 | TK14 | TK13 | TK12 | TK11 | TK10 | TK09 | TK08 |
| **TKPINSEL2** | | | | | TK19 | TK18 | TK17 | TK16 |

**Set TKPINSEL0~2 to fix IO as TK channel**

In the TK Mode, user assigns TKPD=0 to turn on the TK module, then set the TKSOC bit to start touch key conversion, the TKSOC bit can be automatically cleared while end of conversion. However, if the SYSCLK is too slow, H/W might fail to clear TKSOC due to clock sampling rate. TKEOC=0 means conversion is in process. TKEOC=1 means the conversion is finish, and the touch key counting result is stored into the XRAM. After TKEOC=1, user must wait at least 50 µs for next conversion. But if TKRERUN = 1, TK will always be converted, and there is no need to set TKSOC for each conversion. Reducing/increasing TKTMR can reduce/increase the TKDATA to accommodate the condition of the system.

The FTK has an internal built-in reference capacitor to simulate the KEY behavior. Set TKCHS=17h and start the scanning can get the TK Data Count of internal reference capacitor (TKCAP). Since the internal capacitor would not be affected by water or mobile phone, it is useful for comparing the environment background noise. Setting the TKFJMP, the frequency of Touch Key clock can be change automatically by H/W controlled. It may help to improve the ability to resist noise.

ITRIM are 7 bits data for TK channel reference voltage fine tune. E000h.6~0 is TK00 reference voltage fine tune. E001h.6~0 is TK01 reference voltage fine tune. E017h.6~0 is TKCAP (TK23) reference voltage fine tune etc. Users can use ITRIM to obtain similar reference voltages for different TK channels



**FTK Structure**

SFR ATKCH0~2 are used to specify scan TK channel, and each bit is mapped to TK pin. TK scan will scan from low bit to high bit. If ATKMODE = 0, TK can scan up to 21 channels, TK00~TK19 and TKCAP (TK23), each channel is scanned once. If ATKMODE = 1, TK can scan up to 16 channels, each channel is scanned twice. If ATKMODE = 2, TK can scan up to 8 channels, each channel is scanned 4 times. If ATKMODE = 3, TK can scan up to 4 channels, each channel is scanned 8 times. TKCHS is used to specify the first channel for TK to start scanning.

For example:

Condition ATKMODE=0, scan TK16/TK14/TK08/TK07/TK06/TK02

⇨ TKPINSEL2=0000_0001, TKPINSEL1=0100_0001, TKPINSEL0=1100_0100

⇨ ATKCH2=0000_0001, ATKCH1=0100_0001, ATKCH0=1100_0100

⇨ TKCHS=0x02 (Specify the first scan channel)

The arrangement of TK data stored in XRAM is as follows.

| | XRAM |
|---|---|
| E100h | TK00 DATAL |
| E101h | TK00 DATAH |
| E102h | TK01 DATAL |
| E103h | TK01 DATAH |
| | … |
| E128h | TK20 DATAL |
| E129h | TK20 DATAH |
| | … |
| E12Eh | TK23 DATAL |
| E12Fh | TK23 DATAH |

The TK scan result is 14-bit data, which are DATAH 6-bit and DATAL 8-bit. DATAH/L must be read in order to get the correct 14-bit data: first read the low byte (DATAL), then read the high word byte (DATAH)

Condition ATKMODE=1, scan TK16/TK14/TK08/TK07/TK06/TK02

⇨ TKPINSEL2=0000_0001, TKPINSEL1=0100_0001, TKPINSEL0=1100_0100

⇨ ATKCH2=0000_0001, ATKCH1=0100_0001, ATKCH0=1100_0100

⇨ TKCHS=0x02 (Specify the first scan channel)

The arrangement of TK data stored in XRAM is as follows.

| | XRAM |
|---|---|
| E100h | TK02 1st DATAL |
| E101h | TK02 1st DATAH |
| E102h | TK02 2nd DATAL |
| E103h | TK02 2nd DATAH |
| E104h | TK06 1st DATAL |
| E105h | TK06 1st DATAH |
| E106h | TK06 2nd DATAL |
| E107h | TK06 2nd DATAH |
| | … |
| E114h | TK16 1st DATAL |
| E115h | TK16 1st DATAH |
| E116h | TK16 2nd DATAL |
| E117h | TK16 2nd DATAH |
| | … |

The TK scan result is 14-bit data, which are DATAH 6-bit and DATAL 8-bit. DATAH/L must be read in order to get the correct 14-bit data: first read the low byte (DATAL), then read the high word byte (DATAH)

Condition ATKMODE=2, scan TK16/TK14/TK08/TK07/TK06/TK02

⇨ TKPINSEL2=0000_0001, TKPINSEL1=0100_0001, TKPINSEL0=1100_0100

⇨ ATKCH2=0000_0001, ATKCH1=0100_0001, ATKCH0=1100_0100

⇨ TKCHS=0x02 (Specify the first scan channel)

The arrangement of TK data stored in XRAM is as follows.

| | XRAM |
|---|---|
| E100h | TK02 1st DATAL |
| E101h | TK02 1st DATAH |
| E102h | TK02 2nd DATAL |
| E103h | TK02 2nd DATAH |
| E104h | TK02 3rd DATAL |
| E105h | TK02 3rd DATAH |
| E106h | TK02 4th DATAL |
| E107h | TK02 4th DATAH |
| E108h | TK06 1st DATAL |
| E109h | TK06 1st DATAH |
| E10Ah | TK06 2nd DATAL |
| E10Bh | TK06 2nd DATAH |
| E10Ch | TK06 3rd DATAL |
| E10Dh | TK06 3rd DATAH |
| E10Eh | TK06 4th DATAL |
| E10Fh | TK06 4th DATAH |
| | … |
| E128h | TK16 1st DATAL |
| E129h | TK16 1st DATAH |
| E12Ah | TK16 2nd DATAL |
| E12Bh | TK16 2nd DATAH |
| E12Ch | TK16 3rd DATAL |
| E12Dh | TK16 3rd DATAH |
| E12Eh | TK16 4th DATAL |
| E12Fh | TK16 4th DATAH |

The TK scan result is 14-bit data, which are DATAH 6-bit and DATAL 8-bit. DATAH/L must be read in order to get the correct 14-bit data: first read the low byte (DATAL), then read the high word byte (DATAH)

Condition ATKMODE=3, scan TK08/TK07/TK06/TK02

⇨ TKPINSEL2=0000_0000, TKPINSEL1=0000_0001, TKPINSEL0=1100_0100

⇨ ATKCH2=0000_0000, ATKCH1=0000_0001, ATKCH0=1100_0100

⇨ TKCHS=0x02 (Specify the first scan channel)

The arrangement of TK data stored in XRAM is as follows.

| | XRAM |
|---|---|
| E100h | TK02 1st DATAL |
| E101h | TK02 1st DATAH |
| E102h | TK02 2nd DATAL |
| E103h | TK02 2nd DATAH |
| E104h | TK02 3rd DATAL |
| E105h | TK02 3rd DATAH |
| E106h | TK02 4th DATAL |
| E107h | TK02 4th DATAH |
| E108h | TK02 5th DATAL |
| E109h | TK02 5th DATAH |
| E10Ah | TK02 6th DATAL |
| E10Bh | TK02 6th DATAH |
| E10Ch | TK02 7th DATAL |
| E10Dh | TK02 7th DATAH |
| E10Eh | TK02 8th DATAL |
| E10Fh | TK02 8th DATAH |
| | … |
| E130h | TK08 1st DATAL |
| E131h | TK081st DATAH |
| E132h | TK08 2nd DATAL |
| E133h | TK08 2nd DATAH |
| E134h | TK08 3rd DATAL |
| E135h | TK08 3rd DATAH |
| E136h | TK08 4th DATAL |
| E137h | TK08 4th DATAH |
| E138h | TK08 5th DATAL |
| E139h | TK08 5th DATAH |
| E13Ah | TK08 6th DATAL |
| E13Bh | TK08 6th DATAH |
| E13Ch | TK08 7th DATAL |
| E13Dh | TK08 7th DATAH |
| E13Eh | TK08 8th DATAL |
| E13Fh | TK08 8th DATAH |

The TK scan result is 14-bit data, which are DATAH 6-bit and DATAL 8-bit. DATAH/L must be read in order to get the correct 14-bit data: first read the low byte (DATAL), then read the high word byte (DATAH)

| SFR 95h | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **INTFLG** | LVDIF | – | TKIF | ADIF | – | – | P1IF | TF3 |
| R/W | R | – | R/W | R/W | – | – | R/W | R/W |
| Reset | – | – | 0 | 0 | – | – | 0 | 0 |

95h.5 **TKIF:** Touch Key Interrupt Flag

Set by H/W at the end of Touch Key conversion if SYSCLK is fast enough. S/W writes DFh to INTFLG or sets the TKSOC bit to clear this flag.

| SFR **ADh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TKCON** | TKPD | TKEOC | TKRERUN | TKIVCS | TKXCAP | TKOFFSET | ATKMODE | |
| R/W | R/W | R | R/W | R/W | R/W | R/W | R/W | |
| Reset | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

ADh.7 **TKPD:** Touch Key power down

0: Touch Key enable

1: Touch Key disable

ADh.6 **TKEOC:** Touch Key end of conversion flag, TKEOC may have 3uS delay after TKSOC=1, so F/W must wait enough time before polling this Flag.

0: Indicates conversion is in progress

1: Indicates conversion is finished

ADh.5 **TKRERUN:** TK Auto re-start, doesn’t need to set TKSOC again to restart TK converter.

0: Auto re-start disable. TKSOC needs to be executed once for each TK conversion

1: Auto re-start enable. After TKSOC is executed once, TK will be converted continuously without re-executing TKSOC

ADh.4 **TKIVCS:** Touch Key internal voltage control select

0: VCHG=2.8V; VINT=1.4V

1: VCHG=3.6V; VINT=1.8V

ADh.3 **TKXCAP:** Touch Key external capacitor select

0: Keep 0, disable Touch Key external capacitor

1: reserved (Do not set to 1)

ADh.2 **TKOFFSET:** status of non-scan TK

0: connect to VSS

1: connect to AC shielding , connect to VSS@EOC

ADh.1~0 **ATKMODE:** Touch Key Scan Mode

00: TK scan method, each channel scan 1 time, max 21 TK channels
01: TK scan method, each channel scan 2 times, max 16 TK channels
10: TK scan method, each channel scan 4 times, max 8 TK channels
11: TK scan method, each channel scan 8 times, max 4 TK channels

***Note:*** *also refer to Section 6 for more information about Touch Key Interrupt enable and priority.*

| SFR **B4h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TKTMRL** | TKTMRL | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

B4h.7~0 **TKTMRL:** Touch Key Scan length bit 7~0 adjustment.
00: shortest, FF: longest

| SFR **B5h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TKCON2** | TKFJMP | JMPVAL | | SPREAD | TKTMRH | | | |
| R/W | R/W | R/W | | R/W | R/W | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

B5h.7 **TKFJMP:** Internal Touch Key clock frequency auto adjust option
0: Disable
1: Enable

B5h.6~5 **JMPVAL :** Touch Key Clock frequency fine tune , only available in TKFJMP=0
00=frequency slowest, 11=frequency fastest

B5h.4 **SPREAD:** TK spread spectrum
0: Disable
1: Enable

B5h.3~0 **TKTMRH:** Touch Key Scan length 11~8 adjustment.
0000: shortest, 1111: longest

| SFR **F8h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **AUX1** | CLRWDT | CLRTM3 | TKSOC | ADSOC | LVRPD | T2SEL | T1SEL | DPSEL |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

F8h.5 **TKSOC:** Touch Key Start of Conversion
Set 1 to start Touch Key conversion. If SYSCLK is fast enough, this bit will be cleared by H/W at the end of conversion while TKRERUN=0. S/W can also write 0 to clear this flag.

| SFR **A7h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TKCHS** | – | – | – | TKCHS | | | | |
| R/W | – | – | – | R/W | | | | |
| Reset | – | – | – | 1 | 1 | 1 | 1 | 1 |

A7h.4~0 **TKCHS:** Specify the first touch key scan channel

00000: TK00
00001: TK01
00010: TK02
00011: TK03
00100: TK04
00101: TK05
00110: TK06
00111: TK07
01000: TK08
01001: TK09
01010: TK10
01011: TK11
01100: TK12
01101: TK13
01110: TK14
01111: TK15
10000: TK16
10001: TK17
10010: TK18
10011: TK19
10111: TKCAP: internal reference capacitor channel

| SFR **C1h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TKPINSEL0** | **TKPINSEL0** | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

C1h.7 TK07 Pin fix as TK channel: 0: disable 1: enable

C1h.6 TK06 Pin fix as TK channel: 0: disable 1: enable

C1h.5 TK05 Pin fix as TK channel: 0: disable 1: enable

C1h.4 TK04 Pin fix as TK channel: 0: disable 1: enable

C1h.3 TK03 Pin fix as TK channel: 0: disable 1: enable

C1h.2 TK02 Pin fix as TK channel: 0: disable 1: enable

C1h.1 TK01 Pin fix as TK channel: 0: disable 1: enable

C1h.0 TK00 Pin fix as TK channel: 0: disable 1: enable

| SFR **C2h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TKPINSEL1** | **TKPINSEL1** | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

C2h.7 TK15 Pin fix as TK channel: 0: disable 1: enable

C2h.6 TK14 Pin fix as TK channel: 0: disable 1: enable

C2h.5 TK13 Pin fix as TK channel: 0: disable 1: enable

C2h.4 TK12 Pin fix as TK channel: 0: disable 1: enable

C2h.3 TK11 Pin fix as TK channel: 0: disable 1: enable

C2h.2 TK10 Pin fix as TK channel: 0: disable 1: enable

C2h.1 TK09 Pin fix as TK channel: 0: disable 1: enable

C2h.0 TK08 Pin fix as TK channel: 0: disable 1: enable

| SFR **C3h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TKPINSEL2** | **TKPINSEL2** | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

C3h.7~5 Reservd

C3h.4 TK20 Pin fix as TK channel: 0: disable 1: enable

C3h.3 TK19 Pin fix as TK channel: 0: disable 1: enable

C3h.2 TK18 Pin fix as TK channel: 0: disable 1: enable

C3h.1 TK17 Pin fix as TK channel: 0: disable 1: enable

C3h.0 TK16 Pin fix as TK channel: 0: disable 1: enable

| SFR **C5h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **ATKCH0** | **ATKCH0** | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

C5h.7 TK07 scan enable: 0: disable 1: enable
C5h.6 TK06 scan enable: 0: disable 1: enable
C5h.5 TK05 scan enable: 0: disable 1: enable
C5h.4 TK04 scan enable: 0: disable 1: enable
C5h.3 TK03 scan enable: 0: disable 1: enable
C5h.2 TK02 scan enable: 0: disable 1: enable
C5h.1 TK01 scan enable: 0: disable 1: enable
C5h.0 TK00 scan enable: 0: disable 1: enable

| SFR **C6h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **ATKCH1** | **ATKCH1** | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

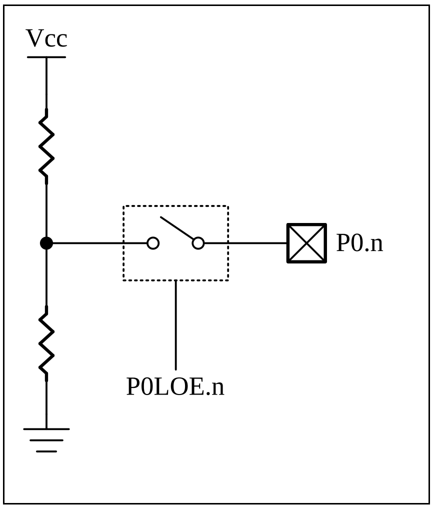
C6h.7 TK15 scan enable: 0: disable 1: enable
C6h.6 TK14 scan enable: 0: disable 1: enable
C6h.5 TK13 scan enable: 0: disable 1: enable
C6h.4 TK12 scan enable: 0: disable 1: enable
C6h.3 TK11 scan enable: 0: disable 1: enable
C6h.2 TK10 scan enable: 0: disable 1: enable
C6h.1 TK09 scan enable: 0: disable 1: enable
C6h.0 TK08 scan enable: 0: disable 1: enable

| SFR **C7h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **ATKCH2** | **ATKCH2** | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

C7h.7 TKCAP (TK23) internal reference capacitor channel scan enable: 0: disable 1: enable
C7h.6~5 Reservd
C7h.4 TK20 scan enable: 0: disable 1: enable
C7h.3 TK19 scan enable: 0: disable 1: enable
C7h.2 TK18 scan enable: 0: disable 1: enable
C7h.1 TK17 scan enable: 0: disable 1: enable
C7h.0 TK16 scan enable: 0: disable 1: enable

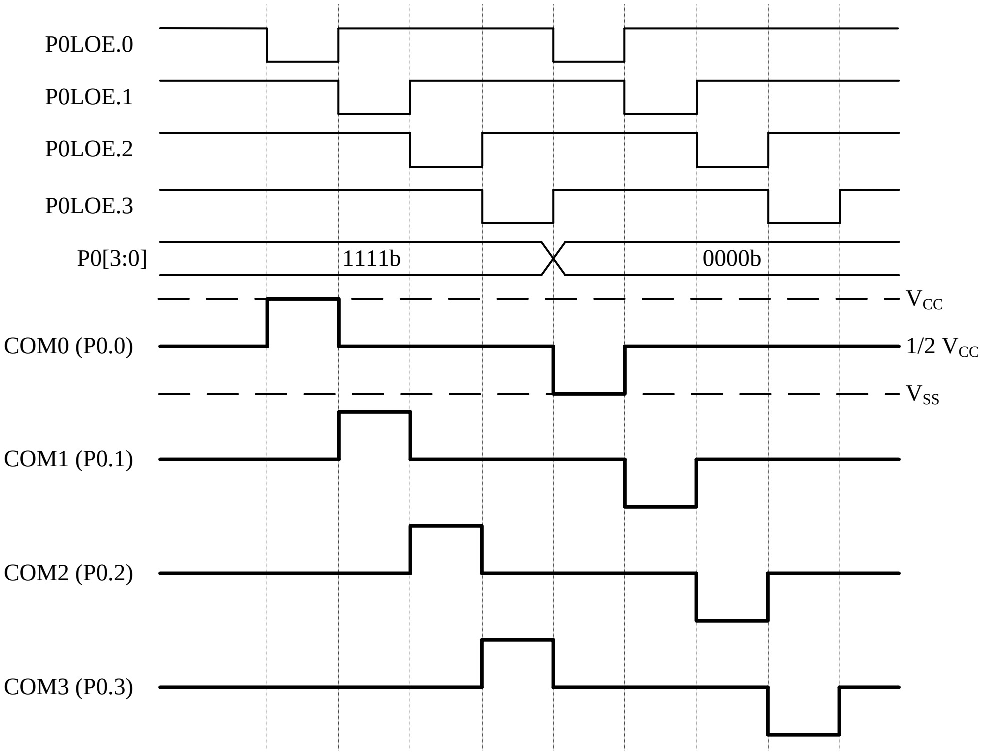
## 13. S/W Controller LCD Driver

The chip supports an S/W controlled method to driving LCD. It is capable of driving the LCD panel with 144 dots (Max.) by 8 Commons (COM) and 18 Segments (SEG). The P0.0~P0.7 are used for Common pins COM0~COM7 and others pins can be used for Segment pins. COM0~COM7 are capable of driving 1/2 bias when P0.0~P0.7's P0LOE=1. Refer to the following figures.



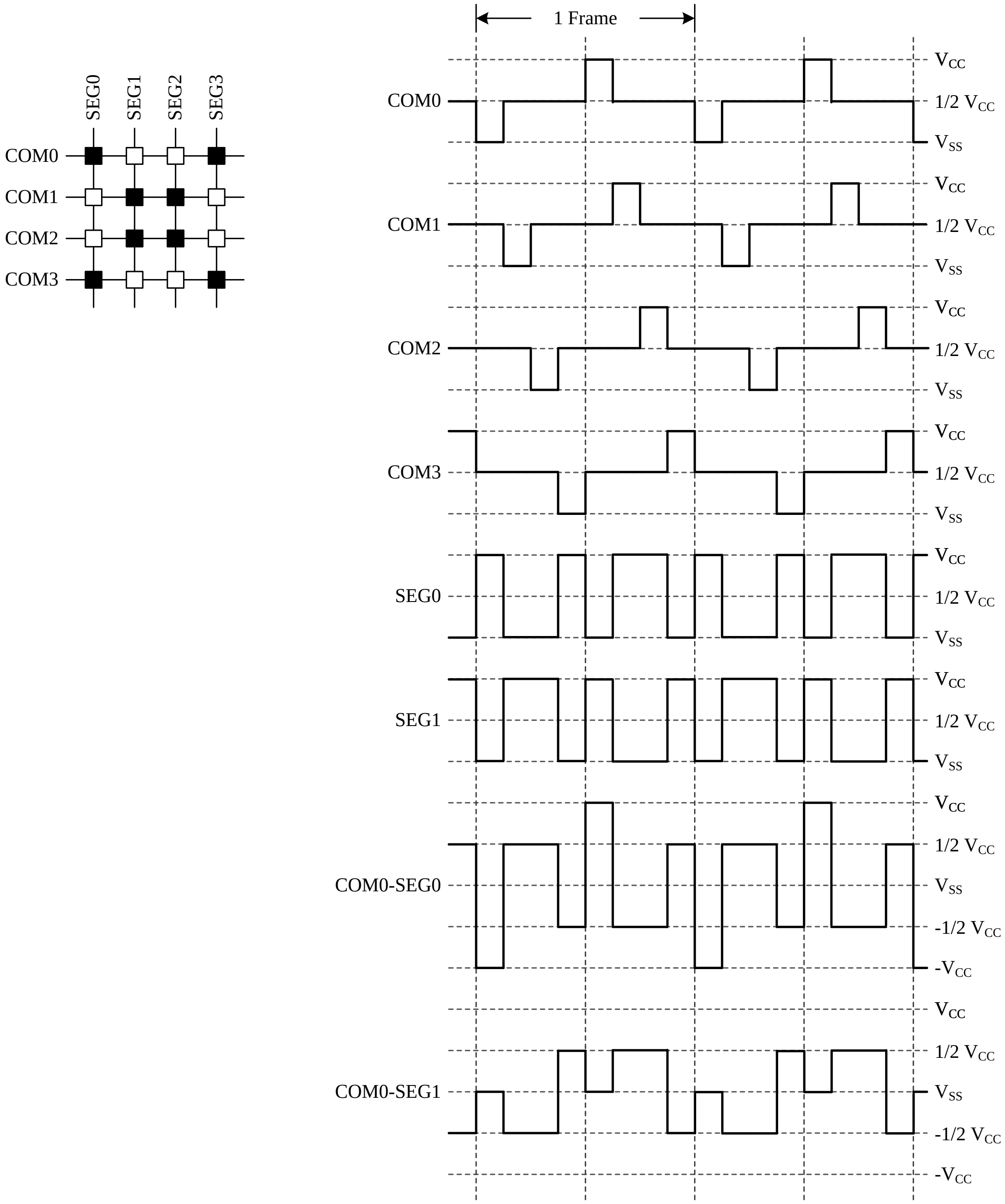
**LCD COM0~7 Circuit**

The frequency of any repeating waveform output on the COM pin can be used to represent the LCD frame rate. The figure below shows an LCD frame.



**S/W Controlled LCD COM0~3 Scanning**

## 1/4 Duty, 1/2 Bias Output Waveform



| SFR 92h | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **P0LOE** | P0LOE | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

92h.7~0 **P0LOE:** P0.7~P0.0 LCD 1/2 bias output enable control

0: Disable

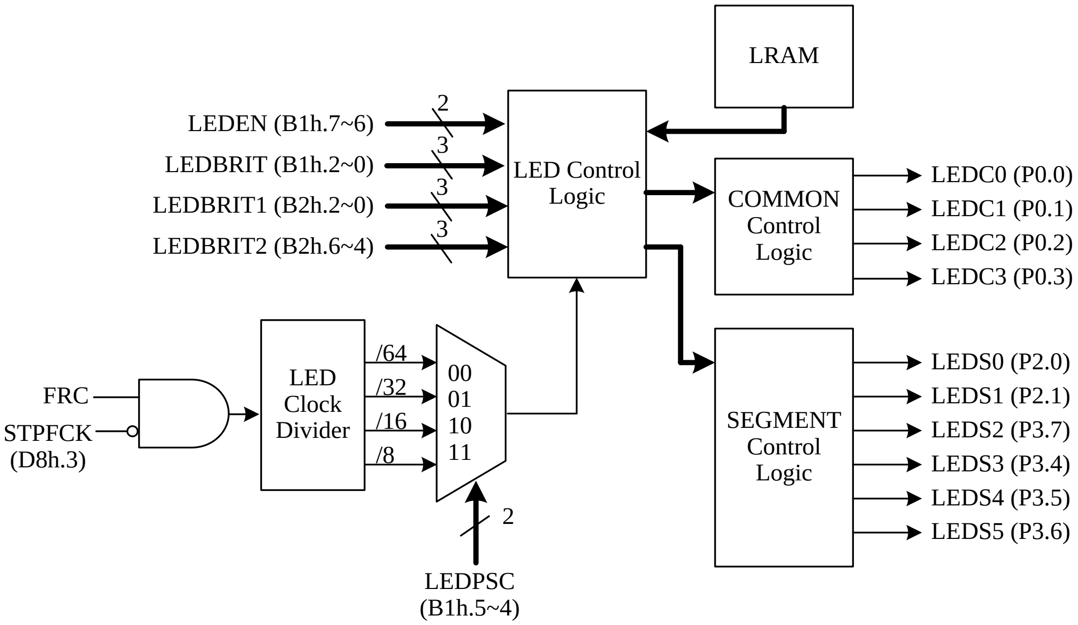
1: Enable

# 14. LED Controller/Driver

The module can be configured with two drive modes: LED BiD (Bi-Direction) matrix mode and LED dot matrix mode. By register configuration, it only supports one mode of operation at the same time.

## 14.1 LED Bi-Direction (BiD) Mode

The LED BiD mode can drive more number of LED pixels than the tradition mode, when they use the same number of pins. In this mode, it provides maximum 10 pins (LEDC0~C3, LEDS0~S5) to drive a LED module with 48 pixels. All 10 pins have a high sink current for driving LED directly. This LED controller also provides 3groups 8-level of brightness adjustment for all 10 pin. To avoid LED flicker when the common signal is changing, the chip provides a dead time control. In the dead time period, segment pins will output a short inactive signal instead of changing the signal immediately. To start the LED scanning, it only has to set the LEDEN. Then H/W will control the Pin mode automatically.

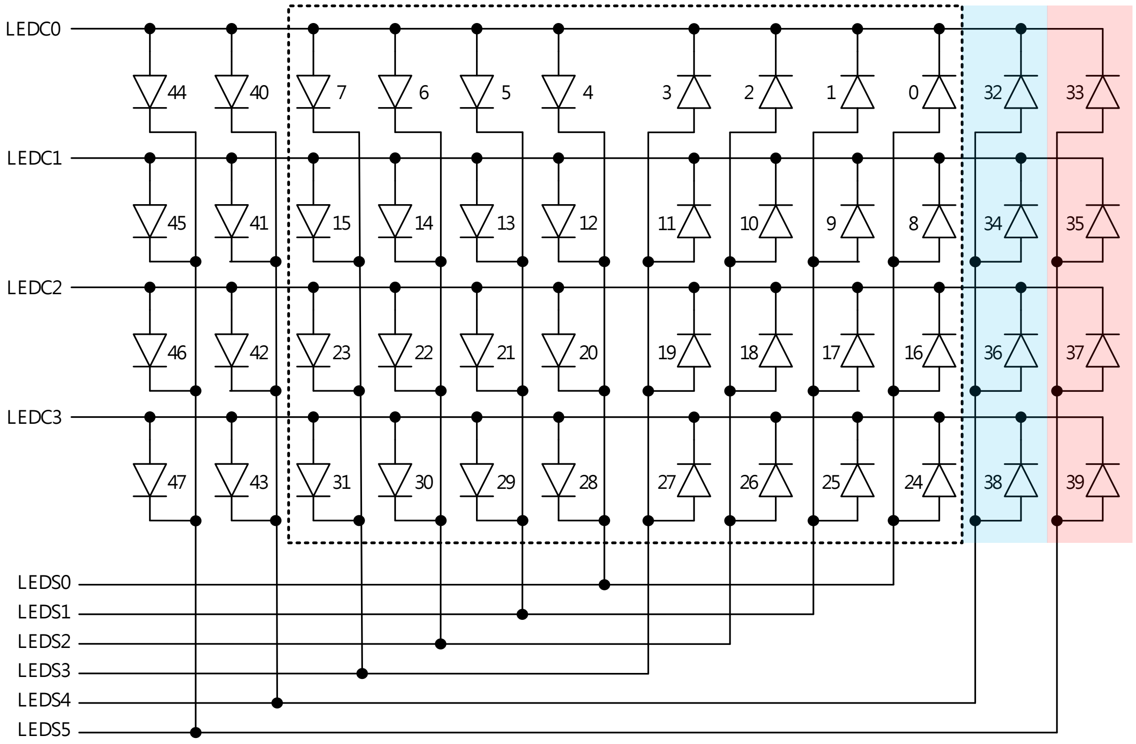


| LEDEN | Duty | Matrix | Max pixels |
|---|---|---|---|
| 0 | Disable | - | - |
| 1 | 1/8 | 4COM x 4SEG | 32 (4x4x2) |
| 2 | 1/9 | 4COM x 5SEG | 40 (4x5x2) |
| 3 | 1/10 | 4COM x 6SEG | 48 (4x6x2) |

| LRAM Addr. | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| C800h | SEG3-COM0+ | SEG2-COM0+ | SEG1-COM0+ | SEG0-COM0+ | COM0-SEG3+ | COM0-SEG2+ | COM0-SEG1+ | COM0-SEG0+ |
| C801h | SEG3-COM1+ | SEG2-COM1+ | SEG1-COM1+ | SEG0-COM1+ | COM1-SEG3+ | COM1-SEG2+ | COM1-SEG1+ | COM1-SEG0+ |
| C802h | SEG3-COM2+ | SEG2-COM2+ | SEG1-COM2+ | SEG0-COM2+ | COM2-SEG3+ | COM2-SEG2+ | COM2-SEG1+ | COM2-SEG0+ |
| C803h | SEG3-COM3+ | SEG2-COM3+ | SEG1-COM3+ | SEG0-COM3+ | COM3-SEG3+ | COM3-SEG2+ | COM3-SEG1+ | COM3-SEG0+ |
| C804h | COM3-SEG5+ | COM3-SEG4+ | COM2-SEG5+ | COM2-SEG4+ | COM1-SEG5+ | COM1-SEG4+ | COM0-SEG5+ | COM0-SEG4+ |
| C805h | SEG5-COM3+ | SEG5-COM2+ | SEG5-COM1+ | SEG5-COM0+ | SEG4-COM3+ | SEG4-COM2+ | SEG4-COM1+ | SEG4-COM0+ |

| LRAM Addr. | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| C800h | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| C801h | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| C802h | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| C803h | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| C804h | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| C805h | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 |

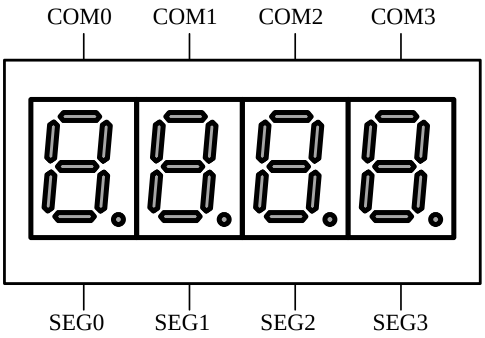
**LED BiD matrix mode corresponding display configuration table**

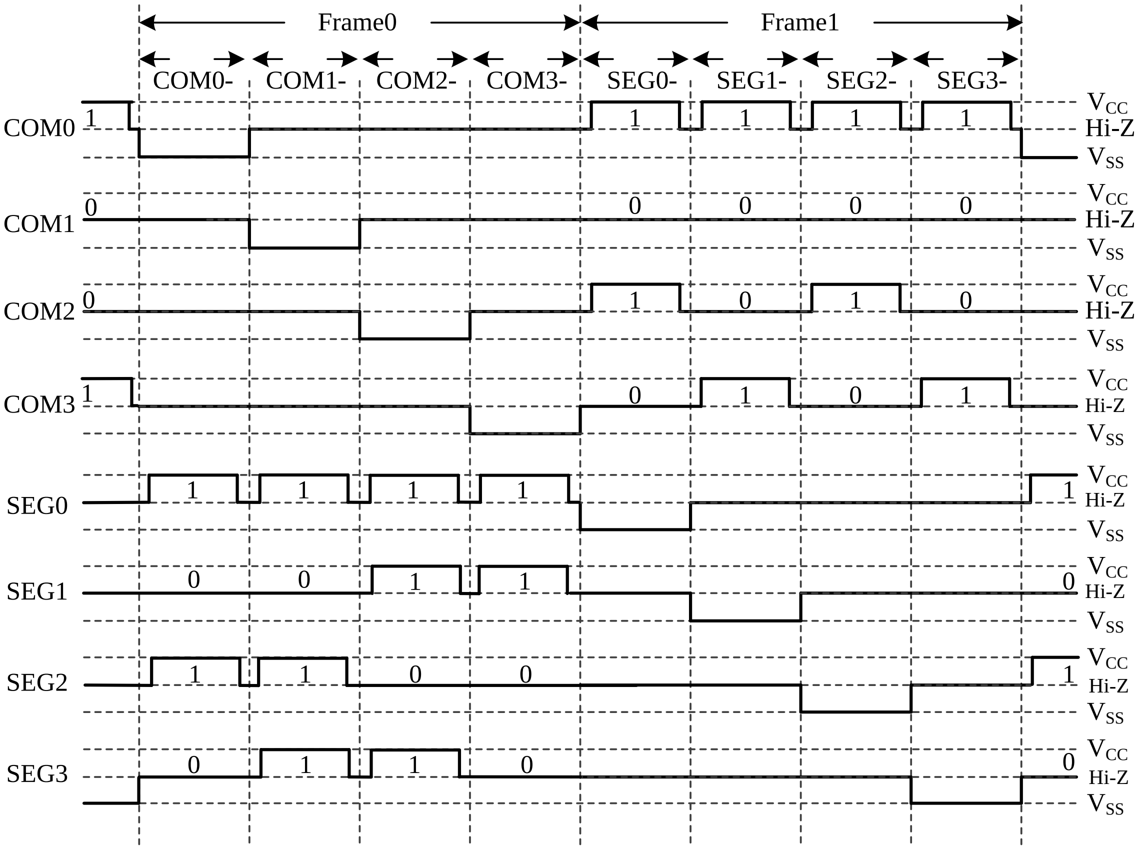


**LED 4*6 BiD matrix**

***Note***: *LEDBRIT (B1h.2~0) : LED number 0~31, 40~47 brightness control*
*LEDBRIT1 (B2h.2~0): LED number 32, 34, 36, 38 brightness control*
*LEDBRIT2 (B2h.6~4): LED number 33, 35, 37, 39 brightness control*

## Application Circuit: 4COM x 4SEG (1/8 Duty)





◇ Example:

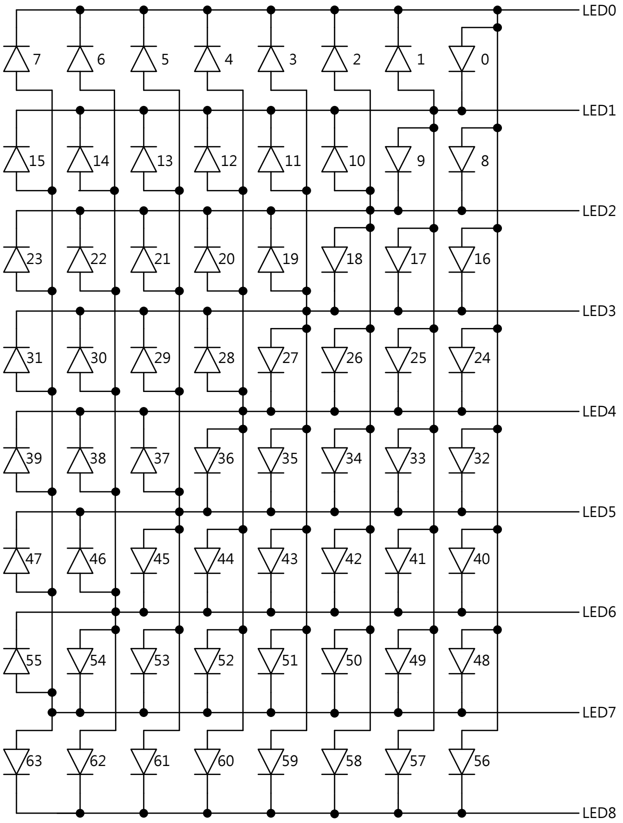
```
MOV     DPTR,#0C800h      ; LEDRAM0
MOV     A,#0FFh
MOVX    @DPTR, A          ; C800h = FFh

MOV     LEDCON,#056h      ; LED duty = 1/8
                          ; LEDPSC = FRC/32
                          ; Brightness=6
```

### 14.2 LED Dot Matrix (DMX) Mode

If LEDMTEN=1, LED dot matrix mode will enable. The LED dot matrix is a universal 8*8 dot matrix. Corresponding to LED0~LED8 ports, up to 8x8=56 LED dots can be configured to drive, the corresponding position of the LED is marked in the 8*8 dot matrix in the figure below Address, the display configuration in XRAM corresponds to the lighting status of the corresponding address (1 means lighting, 0 means not lighting). Support up to 64 lights LED drive. Using LEDCON3 to choose dot matrix 4*5, 5*6, 6*7, 7*8 or 8*8, the corresponding LED address remains unchanged. The brightness of the LED can be set by LCDBRIT2. When it is set to 1111b, it is the highest brightness. In addition, LEDBRITM is used to set the brightness and uniformity bit. When LEDBRITM=0, better display uniformity can be obtained. When LEDBRITM= 1, better display brightness can be obtained.

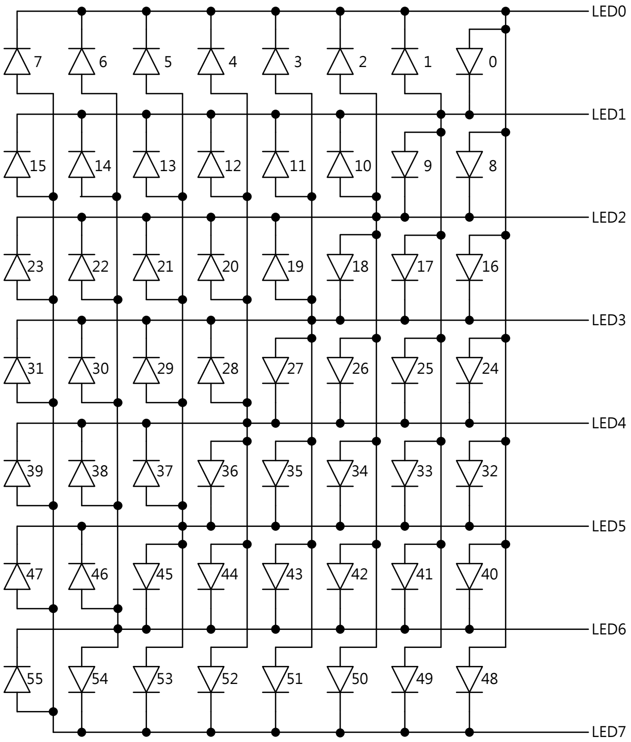
| XRAM Addr. | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| C800h | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| C801h | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| C802h | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| C803h | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| C804h | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| C805h | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 |
| C806h | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 |
| C807h | 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 |

**LED Dot matrix mode corresponding display configuration table**

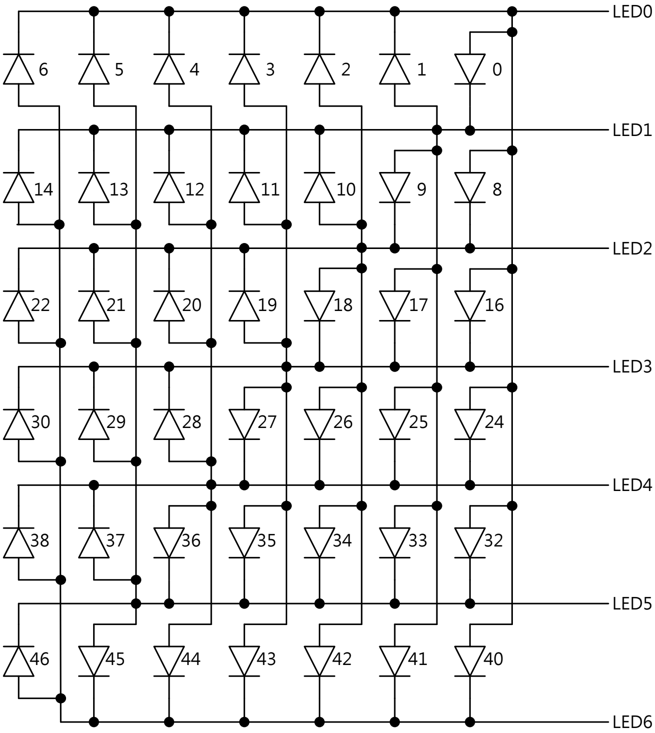


**LED 8*8 dot matrix**

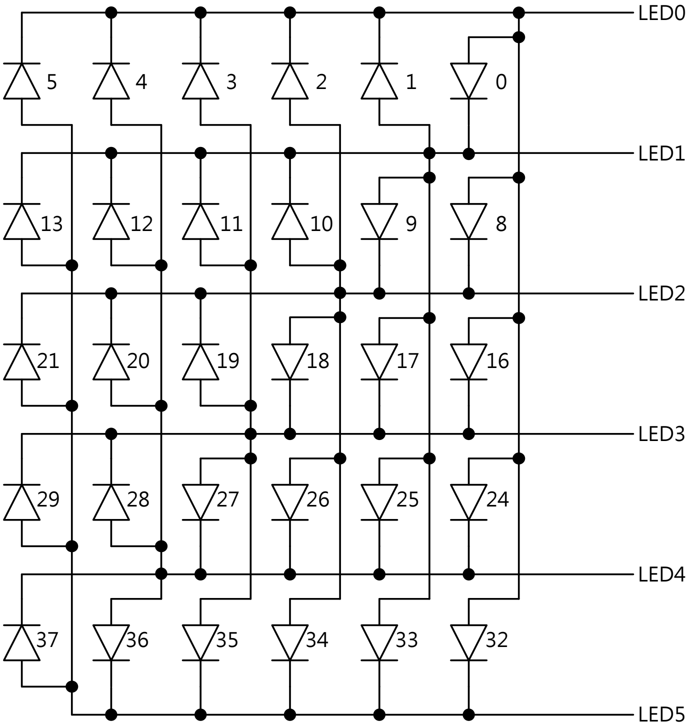
***Note***: *LEDBRIT2 (B2h.6~4): LED number 0~63 brightness control*



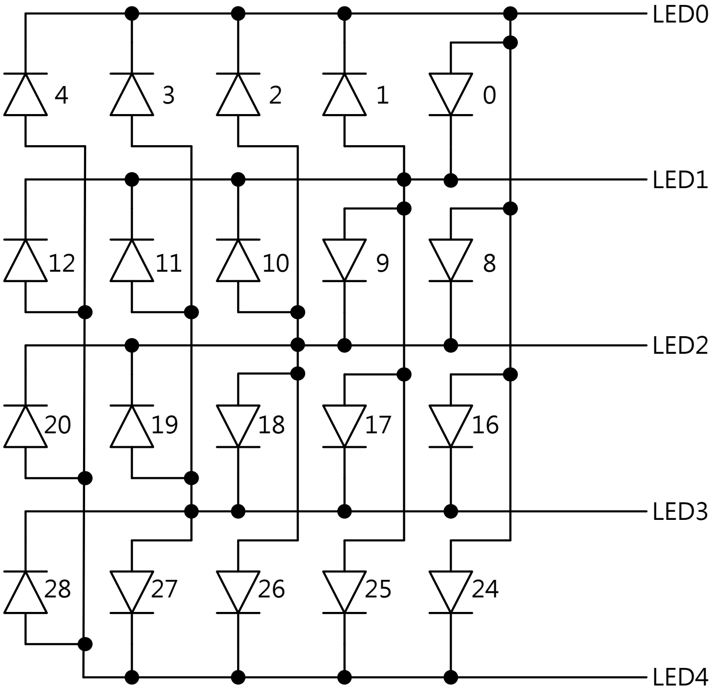
LED 7*8 dot matrix



LED 6*7 dot matrix



**LED 5*6 dot matrix**



**LED 4*5 dot matrix**

| SFR **B1h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **LEDCON** | LEDEN | | LEDPSC | | LEDHOLD | LEDBRIT | | |
| R/W | R/W | | R/W | | R/W | R/W | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

B1h.7~6 **LEDEN:** LED BiD matrix mode enable and duty select
00: LED BiD matrix mode disable
01: LED 1/8 duty (4COM x 4SEG)
10: LED 1/9 duty (4COM x 5SEG)
11: LED 1/10 duty (4COM x 6SEG)

B1h.5~4 **LEDPSC:** LED clock prescaler select
00: LED clock is FRC divided by 64
01: LED clock is FRC divided by 32
10: LED clock is FRC divided by 16
11: LED clock is FRC divided by 8

B1h.3 **LEDHOLD:** LED clock hold
0: LED scan
1: LED clock hold

B1h.2~0 **LEDBRIT:**
BiD matrix mode: LED  number 0~31, 40~47 brightness control
000: Level 0 (Darkest)
…
111: Level 7 (Brightest)

| SFR **B2h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **LEDCON2** | LEDBRITM | LEDBRIT2 | | | – | LEDBRIT1 | | |
| R/W | R/W | R/W | | | – | R/W | | |
| Reset | 0 | 1 | 0 | 0 | – | 1 | 0 | 0 |

B2h.7 **LEDBRITM:** Brightness mode control
0: Uniform brightness mode
1: Brightness enhancement mode

B2h.6~4 **LEDBRIT2:**
BiD matrix mode: LED number 33, 35, 37, 39  brightness control
Dot matrix mode: LED number 0~63  brightness control
000: Level 0 (Darkest)
…
111: Level 7 (Brightest)

B2h.2~0 **LEDBRIT1:**
BiD matrix mode: LED number 32, 34, 36, 38 brightness control
000: Level 0 (Darkest)
…
111: Level 7 (Brightest)

| SFR **D8h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **CLKCON** | SCKTYPE | FCKTYPE | STPSCK | STPPCK | STPFCK | SELFCK | CLKPSC | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Reset | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |

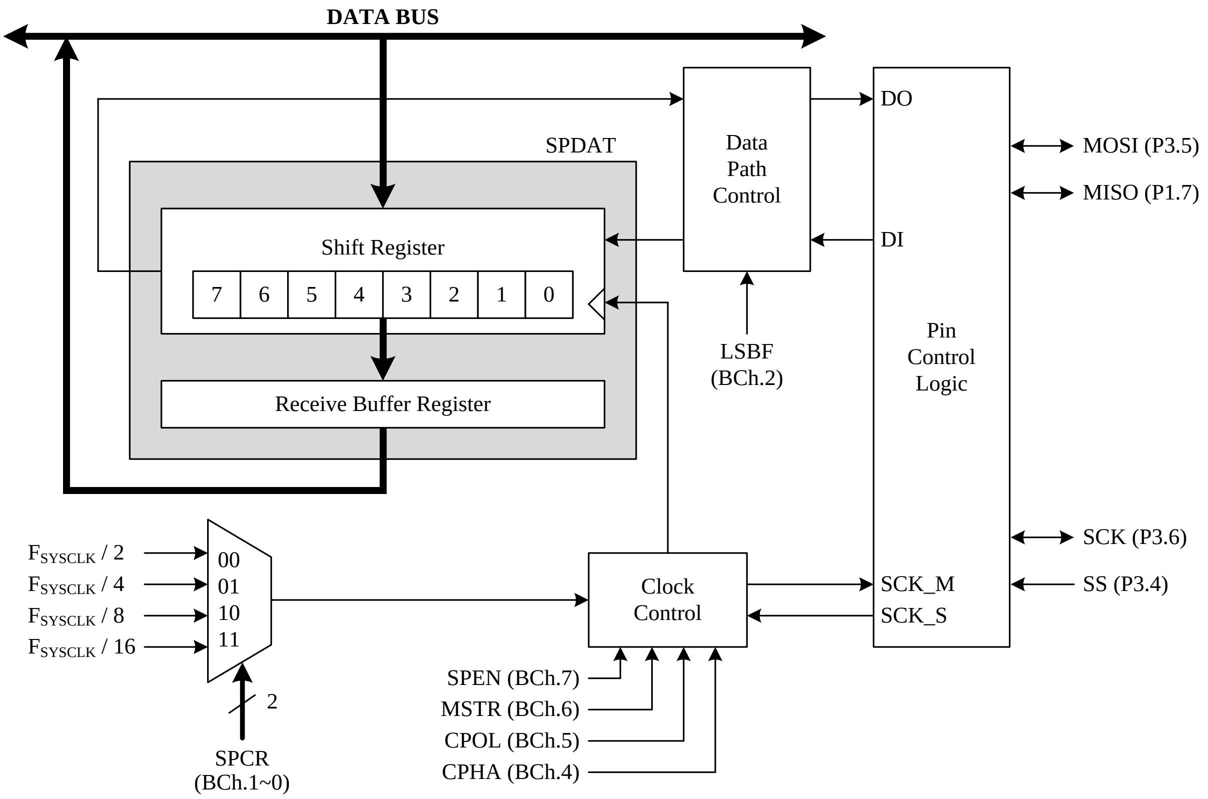
D8h.3 **STPFCK:** Set 1 to stop Fast clock for power saving in Slow/Idle mode. This bit can be changed only in Slow mode.

## 15. Serial Peripheral Interface (SPI)

The Serial Peripheral Interface (SPI) module is capable of full-duplex, synchronous, serial communication between the MCU and peripheral devices. The peripheral devices can be other MCUs, A/D converter, sensors, or flash memory, etc. The SPI runs at a clock rate up to the system clock divided by two. Firmware can read the status flags, or the operation can be interrupt driven. Following figure shows the SPI system block diagram.

The features of the SPI module include:

- Master or Slave mode operation
- 3-wire or 4-wire mode operation
- Full-duplex operation
- Programmable transmit bit rate
- Single buffer receive
- Serial clock phase and polarity options
- MSB-first or LSB-first shifting selectable



| SPI Function Pin | PxMODx | Px.n SFR data | Pin State |
|---|---|---|---|
| Master Mode MISO | **1** | 1 | SPI Data Input |
| Master Mode SCK, MOSI | **2** | X | SPI Clock/Data Output (CMOS Push-Pull) |
| Slave Mode MISO | **2** | X | SPI Data Output (CMOS Push-Pull) |
| Slave Mode SCK, MOSI | **1** | 1 | SPI Clock/Data Input |
| SS | **1** | 1 | SPI Chip Selection |

**Pin Mode Setting for SPI**

The four signals used by SPI are described below. The MOSI signal is an output from a Master Device and an input to Slave Devices. The signal is an output when SPI is operating in Master mode and an input when SPI is operating in Slave mode. The MISO signal is an output from a Slave Device and an input to a Master Device. The signal is an input when SPI is operating in Master mode and an output when SPI is operating in Slave mode. Data is transferred most-significant bit (MSB) or least-significant bit (LSB) first by setting the LSBF bit. The SCK signal is an output from a Master Device and an input to Slave Devices. It is used to synchronize the data on the MOSI and MISO lines of Master and Slave. SPI generates the signal with eight programmable clock rates in Master mode. The SS signal is a low active slave select pin. In 4-wire Slave mode, the signal is ignored when the Slave is not selected (SS=1). The SS is ignored when the SSDIS in SPCON is set in both Master and Slave modes. In Slave mode and the SSDIS is clear, the SPI active when SS stay low. For multiple-slave mode, only one slave device is selected at a time to avoid bus collision on the MISO line. In Master mode and the SSDIS is cleared, the MODF in SPSTA is set when this signal is low. For multiple-master mode, enable SS line to avoid multiple driving on MOSI and SCK lines from multiple masters.

## Master Mode

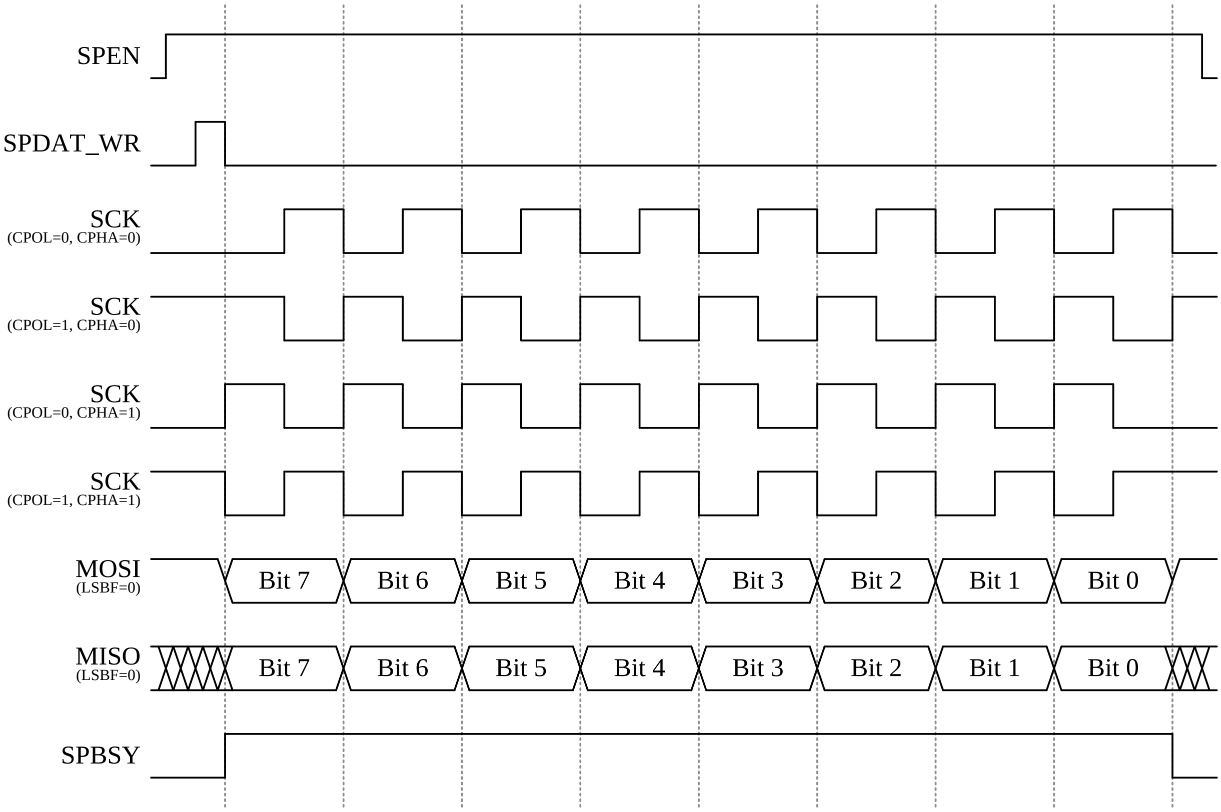
The SPI operates in Master mode by setting the MSTR bit in the SPCON. To start transmit, writing a data to the SPDAT. If the SPBSY bit is cleared, the data will be transferred to the shift register and starts shift out on the MOSI line. The data of the slave shift in from the MISO line at the same time. When the SPIF bit in the SPSTA becomes set at the end of the transfer, the receive data is written to receiver buffer and the RCVBF bit in the SPSTA is set. To prevent an overrun condition, software must read the SPDAT before next byte enters the shift register. The SPBSY bit will be set when writing a data to SPDAT to start transmit, and be cleared at the end of the eighth SCK period in Master mode.

## Slave Mode

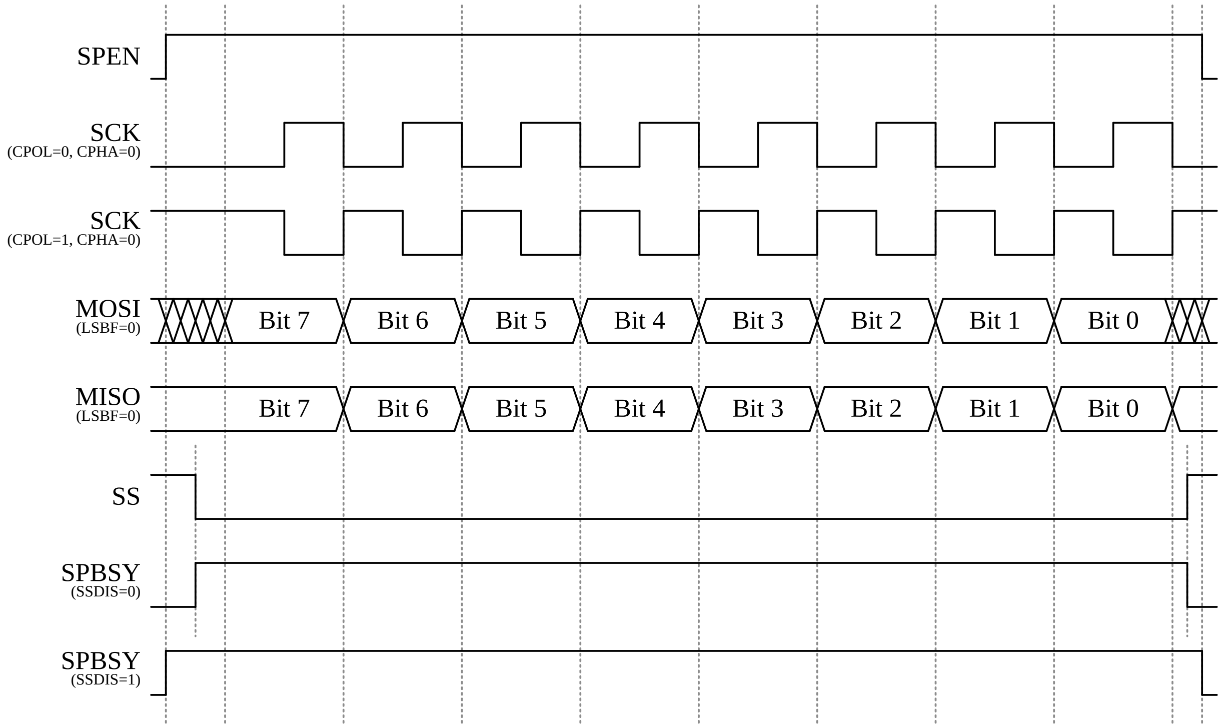
The SPI operates in Slave mode by clearing the MSTR bit in the SPCON. If the SSDIS is cleared, the transmission will start when the SS become low and remain low until the end of a data transfer. If the SSDIS is set, the transmission will start when the SPEN bit in the SPCON is set, and don’t care the SS. The data from a master will shift into the shift register through the MOSI line, and shift out from the shift register on the MISO line. When a byte enters the shift register, the data will be transferred to receiver buffer if the RCVBF is cleared. If the RCVBF is set, the newer receive data will not be transferred to receiver buffer and the RCVOVF bit is set. After a byte enters the shift register, the SPIF and RCVBF bits are set. To prevent an overrun condition, software must read the SPDAT or write 0 to RCVBF before next byte enters the shift register. The maximum SCK frequency allowed in Slave mode is $F_{SYSCLK}/4$. In Slave mode, the SPBSY bit refers to the SS pin when the SSDIS bit is cleared, and refer to the SPEN bit when SSDIS bit is set.

## Serial Clock

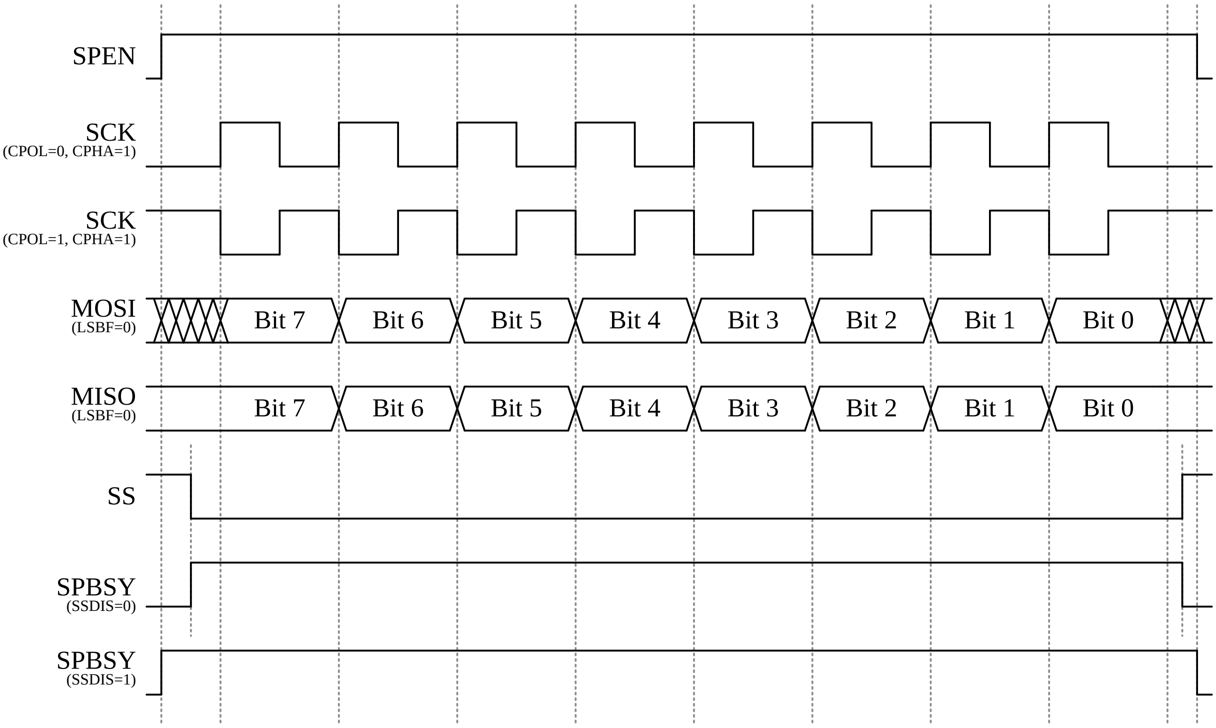
The SPI has four clock types by setting the CPOL and CPHA bits in the SPCON register. The CPOL bit defines the level of the SCK in SPI idle state. The level of the SCK in idle state is low when the CPOL bit is cleared, and is high when the CPOL bit is set. The CPHA bit defines the edges used to sample and shift data. The SPI sample data on the first edge of SCK period and shift data on the second edge of SCK period when the CPHA bit is cleared. The SPI sample data on the second edge of SCK period and shift data on first edge of SCK period when the CPHA bit is set. The figures below show the detail timing in Master and Slave modes. Both Master and Slave devices must be configured to use the same clock type before the SPEN bit is set. The SPCR controls the Master mode serial clock frequency. This register is ignored when operating in Slave mode. The SPI clock can select System clock divided by 2, 4, 8, or 16 in Master mode.



**Master Mode Timing**



**Slave Mode Timing (CPHA=0)**



**Slave Mode Timing (CPHA=1)**

In both Master and Slave modes, the SPIF bit is set by H/W at the end of a data transfer and generates an interrupt if SPI interrupt is enabled. The SPIF bit is cleared automatically when the program performs the interrupt service routines. S/W can also write 0 to clear this flag. If write data to SPDAT when the SPBSY is set, the WCOL bit will be set by H/W and generates an interrupt if SPI interrupt is enabled. When this occurs, the data write to SPDAT will be ignored, and shift register will not be written. Write 0 to this bit or when SPBSY is cleared and rewrite data to SPDAT will clear this flag. The MODF bit is set when SSDIS is cleared and SS pin is pulled low in Master mode. If SPI interrupt is enabled, an interrupt will be generated. When this bit is set, the SPEN and MSTR in SPCON will be cleared by H/W. Write 0 to this bit will clear this flag.

| SFR **BCh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **SPCON** | SPEN | MSTR | CPOL | CPHA | SSDIS | LSBF | SPCR | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

BCh.7 **SPEN:** SPI enable
0: SPI disable 1: SPI enable

BCh.6 **MSTR:** Master mode enable
0: Slave mode 1: Master mode

BCh.5 **CPOL:** SPI clock polarity
0: SCK is low in idle state
1: SCK is high in idle state

BCh.4 **CPHA:** SPI clock phase
0: Data sample on first edge of SCK period
1: Data sample on second edge of SCK period

BCh.3 **SSDIS:** SS pin disable
0: Enable SS pin 1: Disable SS pin

BCh.2 **LSBF:** LSB first
0: MSB first
1: LSB first

BCh.1~0 **SPCR:** SPI clock rate
00: $F_{SYSCLK}/2$
01: $F_{SYSCLK}/4$
10: $F_{SYSCLK}/8$
11: $F_{SYSCLK}/16$

| SFR **BDh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **SPSTA** | SPIF | WCOL | MODF | RCVOVF | RCVBF | SPBSY | – | – |
| R/W | R/W | R/W | R/W | R/W | R/W | R | – | – |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | – | – |

BDh.7 **SPIF:** SPI interrupt flag
This is set by H/W at the end of a data transfer. Cleared by H/W when an interrupt is vectored into. Writing 0 to this bit will clear this flag.

BDh.6 **WCOL:** Write collision interrupt flag
Set by H/W if write data to SPDAT when SPBSY is set. Write 0 to this bit or rewrite data to SPDAT when SPBSY is cleared will clear this flag.

BDh.5 **MODF:** Mode fault interrupt flag
Set by H/W when SSDIS is cleared and SS pin is pulled low in Master mode. Write 0 to this bit will clear this flag. When this bit is set, the SPEN and MSTR in SPCON will be cleared by H/W.

BDh.4 **RCVOVF:** Received buffer overrun flag
Set by H/W at the end of a data transfer and RCVBF is set. Write 0 to this bit or read SPDAT register will clear this flag.

BDh.3 **RCVBF:** Receive buffer full flag
Set by H/W at the end of a data transfer. Write 0 to this bit or read SPDAT register will clear this flag.

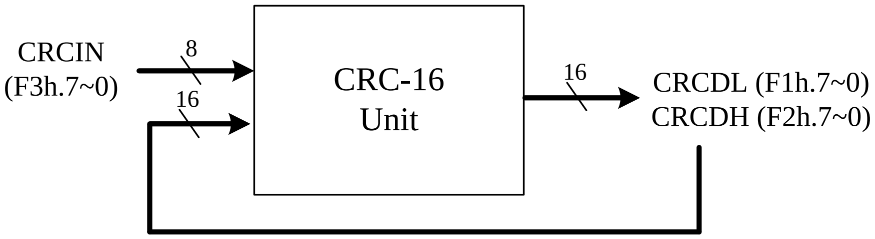
BDh.2 **SPBSY:** SPI busy flag
Set by H/W when a SPI transfer is in progress.

| SFR **BEh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **SPDAT** | SPDAT | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

BEh.7~0 **SPDAT:** SPI transmit and receive data
The SPDAT register is used to transmit and receive data. Writing data to SPDAT place the data into shift register and start a transfer when in master mode. Reading SPDAT returns the contents of the receive buffer.

## 16. Cyclic Redundancy Check (CRC)

The chip supports an integrated 16-bit Cyclic Redundancy Check function. The Cyclic Redundancy Check (CRC) calculation unit is an error detection technique test algorithm and uses to verify data transmission or storage data correctness. The CRC calculation takes a 8-bit data stream or a block of data as input and generates a 16-bit output remainder. The data stream is calculated by the same generator polynomial.



**CRC Block Diagram**

The CRC generator provides the 16-bit CRC result calculation based on the CRC-16-IBM polynomial. In this CRC generator, there are only one polynomial available for the numeric values calculation. It can't support the 16-bit CRC calculations based on any other polynomials. Each write operation to the CRCIN register creates a combination of the previous CRC value stored in the CRCDH and CRCDL registers. It will take one MCU instruction cycle to calculate.

**CRC-16-IBM (Modbus) Polynomial representation: $X^{16} + X^{15} + X^{2} + 1$**

| SFR **F1h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **CRCDL** | CRCDL | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

F1h.7~0 **CRCDL:** 16-bit CRC checksum data bit 7~0

| SFR **F2h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **CRCDH** | CRCDH | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

F2h.7~0 **CRCDL:** 16-bit CRC checksum data bit 15~8

| SFR **F3h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **CRCIN** | CRCIN | | | | | | | |
| W | W | | | | | | | |
| Reset | – | – | – | – | – | – | – | – |

F3h.7~0 **CRCIN:** CRC input data register

## 17. Multiplier and divider

The chip provide multiplier and divider have the following functions. The 8 bit operation is fully compatible with industry standard 8051.

- 8 bits × 8 bits = 16 bit (standard 8051)
- 8 bits ÷ 8 bits = 8 bits, 8 bits remainder (standard 8051)
- 16 bits × 16 bits = 32 bit
- 16 bits ÷ 16 bits = 16 bits, 16 bits remainder
- 32 bits ÷ 16 bits = 32 bits, 16 bits remainder

No matter 8bit / 16bit / 32bit operation, it's easy to execute by MUL AB and DIV AB instruction. There is extra SFR EXA/EXA2/EXA3/EXB for 16bit / 32bit multiply and divide operation.

For 8 bit multiplier/divider operation, be sure SFR bit muldiv16=0 and div32=0.

For 16 bit multiplier operation, multiplicand, multiplier and product as follows. 16 bit multiplier takes 16 System clock cycles to execute.

| **Condition** | **SFR bit muldiv16=1 and div32=0** | | | |
|---|---|---|---|---|
| **Multiplication** | **Byte3** | **Byte2** | **Byte1** | **Byte0** |
| Multiplicand | - | - | EXA | A |
| Multiplier | - | - | EXB | B |
| Product | EXB | B | A | EXA |
| OV | Product (EXB or B) !=0 | | - | - |

For 16 bit divider operation, dividend, divisor, quotient, remainder read as follows. 16 bit divider takes 16 System clock cycles to execute.

| **Condition** | **SFR bit muldiv16=1 and div32=0** | | | |
|---|---|---|---|---|
| **Division** | **Byte3** | **Byte2** | **Byte1** | **Byte0** |
| Dividend | - | - | EXA | A |
| Divisor | - | - | EXB | B |
| Quotient | - | - | A | EXA |
| Remainder | - | - | B | EXB |
| OV | Divisor EXB = B =0 | | | |

For 32 bits ÷ 16 bits operation, dividend, divisor, quotient, remainder read as follows. 32 bit divider takes 32 System clock cycles to execute.

| **Condition** | **SFR bit muldiv16=1 and div32=1** | | | |
|---|---|---|---|---|
| **Division** | **Byte3** | **Byte2** | **Byte1** | **Byte0** |
| Dividend | EXA3 | EXA2 | EXA | A |
| Divisor | - | - | EXB | B |
| Quotient | A | EXA | EXA2 | EXA3 |
| Remainder | - | - | B | EXB |
| OV | Divisor EXB=B =0 | | | |

| SFR CEh | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **EXA2** | EXA2 | | | | | | | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

CEh.7~0 **EXA2:** Expansion accumulator 2

| SFR CFh | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **EXA3** | EXA3 | | | | | | | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

CFh.7~0 **EXA3:** Expansion accumulator 3

| SFR E6h | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **EXA** | EXA | | | | | | | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

E6h.7~0 **EXA:** Expansion accumulator

| SFR E7h | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **EXB** | EXB | | | | | | | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

E7h.7~0 **EXB:** Expansion B register

| SFR F7h | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **AUX2** | WDTE | | PWRSAV | VBGOUT | DIV32 | IAPTE | | MULDIV16 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

F7h.3 **DIV32:**
only active when MULDVI16 =1
0: instruction DIV as 16/16 bit division operation
1: instruction DIV as 32/16 bit division operation

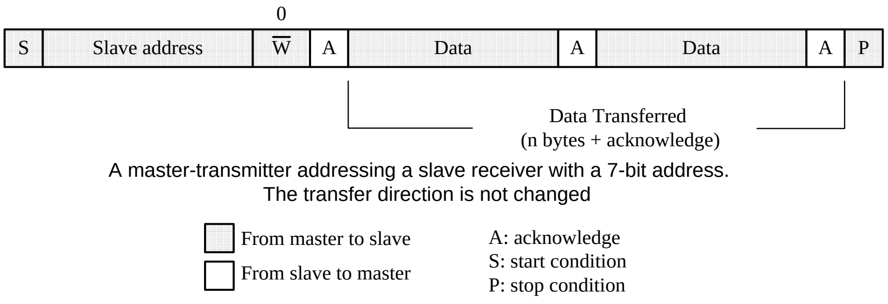
F7h.0 **MULDIV16:**
0: instruction MUL/DIV as 8*8, 8/8 operation
1: instruction MUL/DIV as 16*16, 16/16 or 32/16 operation

| **ARITHMETIC** | | | | |
|---|---|---|---|---|
| **Mnemonic** | **Description** | **byte** | **cycle** | **opcode** |
| MUL AB | Multiply A by B | 1 | 8/16 | A4 |
| DIV AB | Divide A by B | 1 | 8/16/32 | 84 |

## 18. Master I$^2$C Interface

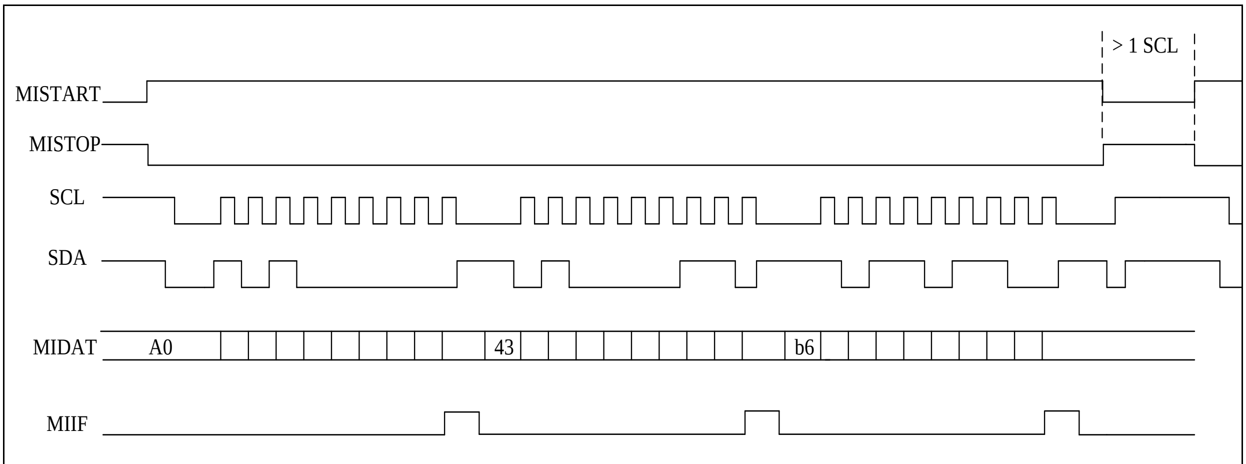
**Master I$^2$C interface transmit mode:**
At the beginning write slave address and direction bit to MIDAT and set MISTART. After the START condition (MISTART), the 7 bits slave address and one bit direction bit are sent. When MIIF convert to 1, address and direction bit transmission was complete. After sending the address and direction bit, user should clear MIIF and write MIDAT to start first data transmission. When MIIF convert to 1, data transfer to slave was complete. User can write MIDAT again to transfer next data to slave. Set MISTOP to finish transmit mode.

MISTART must remain at 1 for the next transfer. After the final data transmit/receive, set MISTOP to finish transmit/receive protocol. MISTART should remain 0 longer than a SCL clock before starting the next Master I$^2$C protocol. SCL clock can be adjusted via MICR**.**



A master-transmitter addressing a slave receiver with a 7-bit address.
The transfer direction is not changed

Master I$^2$C Transmit flow:
(1) Write slave address and direction bit to MIDAT
(2) Clear MISTOP and set MISTART to start I$^2$C transmission
(3) Wait until MIIF convert to 1 (interrupt will be issued according to the user's request) and Clear MIIF
(4) Write data to MIDAT to start next transfer (MISTART must remain at 1)
(5) Wait until MIIF convert to 1 (interrupt will be issued according to the user's request) and Clear MIIF, Loop (4) ~(5) for next transfer.
(6) Clear MISTART and set MISTOP to stop the I$^2$C transfer



Note: MIDAT 43h and b6h are firmware writes to MIDAT to begin the next MIIC transfer.
Note: MISTART should remain 0 longer than a SCL clock before starting the next Master I$^2$C Transfer protocol

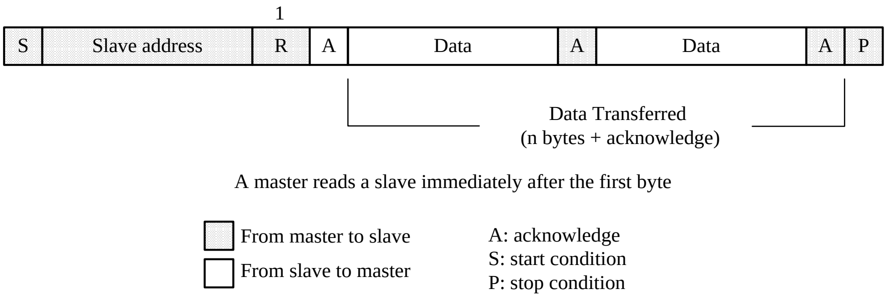
**Master Transmit Timing**

*Note: MISTART should remain 0 longer than a SCL period before starting the next Master I$^2$C protocol.*

**Master I²C interface Receive mode:**

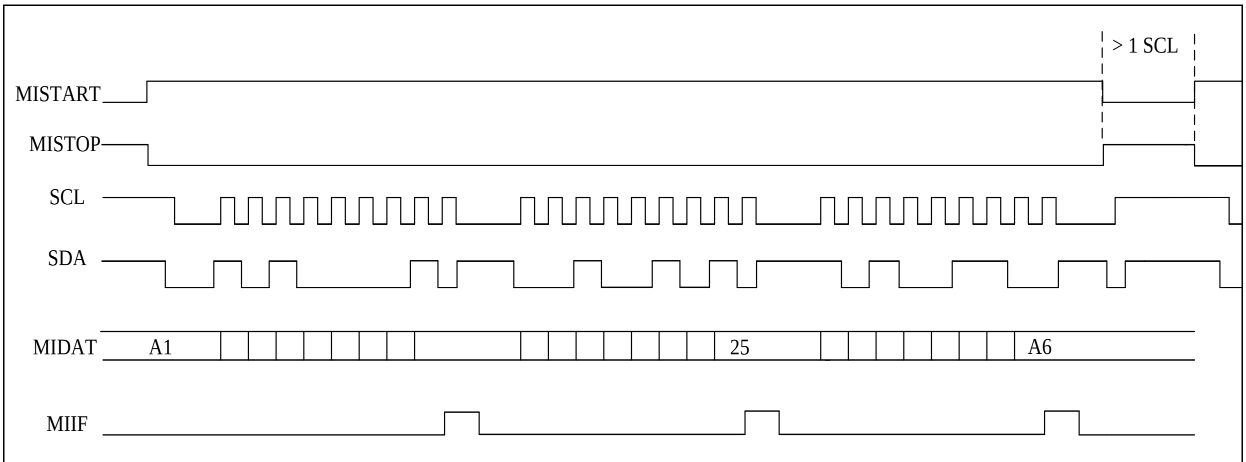
At the beginning write slave address and direction bit to MIDAT and set MISTART. After the START condition (MISTART), the 7 bits slave address and one bit direction bit are sent. When MIIF convert to 1, address and direction bit transmission was complete. After sending the address and direction bit, user should clear MIIF and read MIDAT to start first receive data (The first reading of MIDAT does not represent the data returned by the slave). When MIIF convert to 1, data receive from slave was complete. User can read MIDAT to get data from slave, and start next receive. Set MISTOP to finish receive mode.

MISTART must remain at 1 for the next transfer. After final data transmit/receive, set MISTOP to finish transmit/receive protocol. MISTART should remain 0 longer than a SCL clock before starting the next Master I$^2$C protocol. SCL clock can be adjusted via MICR.



Master I²C Receive flow:

(1) Write slave address and direction bit to MIDAT
(2) Clear MISTOP and set MISTART to start I²C transmission
(3) Wait until MIIF convert to 1 (interrupt will be issued according to the user's request), Clear MIIF
(4) Read data from MIDAT to start first receive data
(receiving data has not been completed at this time, and the read MIDAT should be discarded)
(5) Wait until MIIF convert to 1, Clear MIIF
(6) Read slave data from MIDAT and Loop (5) ~(6) to receive next data
(7) Set MISTOP to stop the I²C transfer



Note: MIDAT 25h and A6h are data from slave

Note: MISTART should remain 0 longer than a SCL clock before starting the next Master I$^2$C Transfer protocol

**Master Receive Timing**

| $I^2C$ Function Pin | P3modx | P3.n SFR data | Pin State |
|---|---|---|---|
| $I^2C$ Master SCL | **0** | X | Clock Output (Open Drain Output) |
| | **2** | X | Clock Output (CMOS Push-Pull) |
| $I^2C$ Master/Slaver SDA | **0** | 1 | DATA (Pull-up) |

**Pin Mode Setting for Master I²C**

| SFR **A9h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **INTE1** | PWMIE | I2CE | ES2 | SPIE | ADTKIE | EXLVDIE | P1IE | TM3IE |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

A9h.6 **I2CE:** I²C interrupt enable
0: Disable I²C interrupt
1: Enable I²C interrupt

| SFR **E1h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **MICON** | MIEN | MIACKO | MIIF | MIACKI | MISTART | MISTOP | MICR | |
| R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

E1h.7 **MIEN**:Master $I^2C$ enable
0: disable 1: enable

E1h.6 **MIACKO**: When Master $I^2C$ receive data, send acknowledge to $I^2C$ Bus
0: ACK to slave device 1: NACK to slave device

E1h.5 **MIIF**: Master $I^2C$ Interrupt flag
When the master $I^2C$ sends or receives a byte, it is set by H/W. Writing "0" to this bit will clear the flag

E1h.4 **MIACKI**: When Master $I^2C$ transfer, acknowledgement form $I^2C$ bus (read only)
0: ACK received 1: NACK received

E1h.3 **MISTART**: Master $I^2C$ Start bit
1: start $I^2C$ bus transfer

E1h.2 **MISTOP**: Master $I^2C$ Stop bit
1: send STOP signal to stop $I^2C$ bus

E1h.1~0 **MICR:** Master $I^2C$ (SCL) clock frequency selection
00: Fsys/4 (ex. If Fsys=18MHz, $I^2C$ clock is 4.5M Hz)
01: Fsys/16 (ex. If Fsys=18MHz, $I^2C$ clock is 1.1M Hz)
10: Fsys/64 (ex. If Fsys=18MHz, $I^2C$ clock is 281K Hz)
11: Fsys/256 (ex. If Fsys=18MHz, $I^2C$ clock is 70K Hz)

| SFR **E2h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **MIDAT** | MIDAT | | | | | | | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

E2h.7~0 **MIDAT**: Master $I^2C$ data shift register
(W):After Start and before Stop condition, write this register will resume transmission to $I^2C$ bus
(R): After Start and before Stop condition, read this register will resume receiving from $I^2C$ bus

| SFR **EAh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **SICON** | MIIE | TXDIE | RCD2IE | RCD1IE | – | TXDF | RCD2F | RCD1F |
| R/W | R/W | R/W | R/W | R/W | – | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | – | 1 | 0 | 0 |

EAh.7 **MIIE:** $I^2C$ Master interrupt enable
0: disable 1: enable

## 19. Slave I$^2$C Interface

The chip provides Slave I$^2$C interface receive protocol as following. Slave I$^2$C module allow to receive one or two byte data each time after start condition. Before receiving DATA1, be aware that RCD1F must be 0. After DATA1 reception is completed, RCD1F will be converted to 1 and an interrupt will be issued according to the user's request. User can use firmware to clear RCD1F before receiving next DATA1 again. User can write RCD1F to 0 to clear RCD1F. DATA2 and RCD2F operate in the same way as DATA1 and RCD1. After DATA1 or DATA2 reception is completed, the Master side should restart the transfer protocol to transmit the next DATA1 and DATA2.

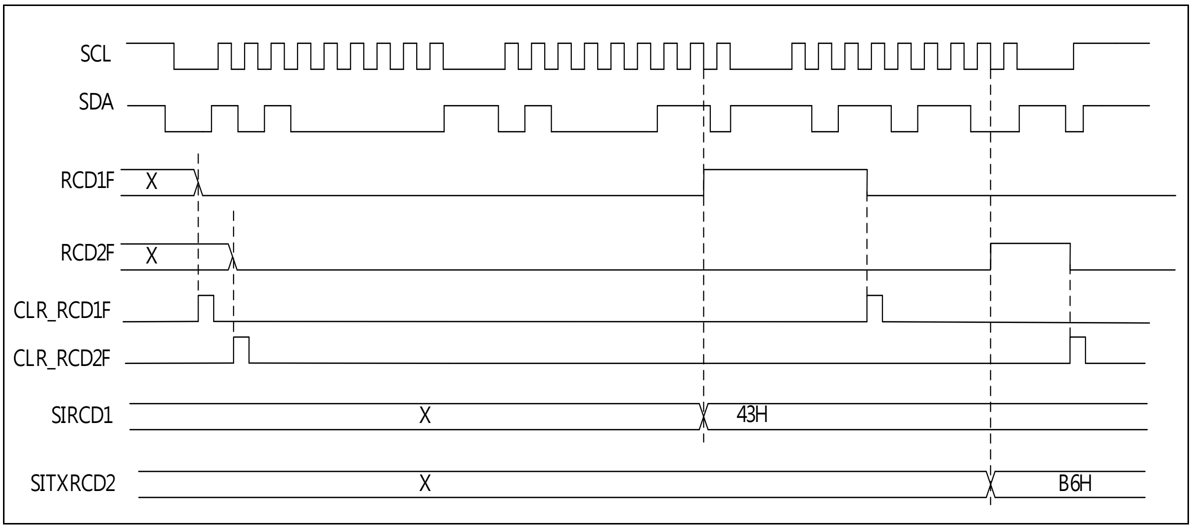
|  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
| S | Slave address | $\overline{W}$ | A | Data 1 | A | P |

Slave I$^2$C Receive Byte protocol

|  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|
| S | Slave address | $\overline{W}$ | A | Data 1 | A | Data 2 | A | P |

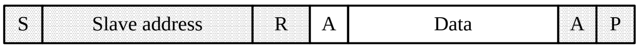
Slave I$^2$C Receive Two Byte protocol

| | | |
|---|---|---|
| ▨ | From master to slave | A: acknowledge |
| □ | From slave to master | S: start condition |
| | | P: stop condition |



**Slave Receive Timing**

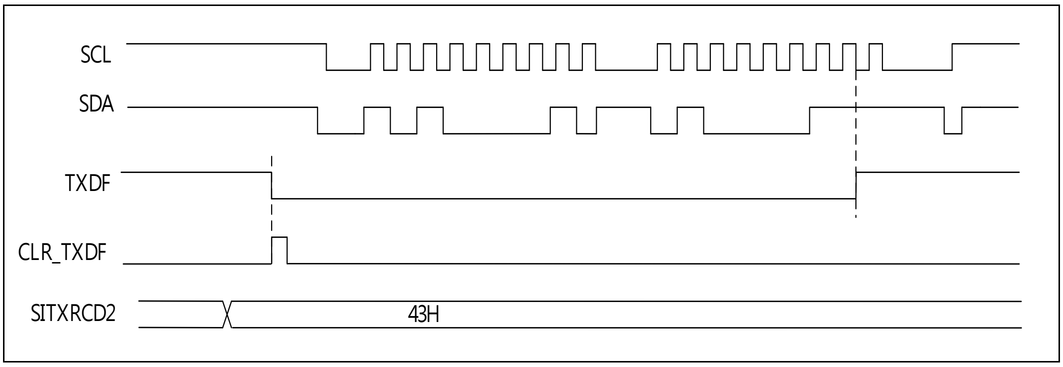
The chip provides Slave I²C interface transmission protocol as following. Slave I²C module allow to transmit one byte data each time after start condition. Before data transmitting, be aware that TXDF must be 0. After data transmission is completed, TXDF will be converted to 1 and an interrupt will be issued according to the user's request. User can use firmware to clear TXDF before transmitting next data again. User can write TXDF to 0 to clear TXDF. After each transmission is completed, the host should restart the transmission protocol to transmit the next data.



Slave $I^2C$ Transmit protocol

▨ From master to slave
□ From slave to master

A: acknowledge
S: start condition
P: stop condition



**Slave Transmit Timing**

| $I^2C$ Function Pin | P3MODx | P3.n<br>SFR data | Pin State |
|---|---|---|---|
| $I^2C$ Slave SCL | **1** | 1 | Clock input |
| $I^2C$ Master/Slaver SDA | **0** | 1 | DATA (Pull-up) |

**Pin Mode Setting for Slave I²C**

| SFR **A9h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **INTE1** | PWMIE | I2CE | ES2 | SPIE | ADTKIE | EXLVDIE | P1IE | TM3IE |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

A9h.6 **I2CE:** I²C interrupt enable
0: Disable I²C interrupt
1: Enable I²C interrupt

| SFR **E9h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **SIADR** | SA | | | | | | | SIEN |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |

E9h.7~1 **SA:** Slave I²C address assigned

E9h.0 **SIEN:** Slave I²C enable
0: disable
1: enable

| SFR **EAh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **SICON** | MIIE | TXDIE | RCD2IE | RCD1IE | – | TXDF | RCD2F | RCD1F |
| R/W | R/W | R/W | R/W | R/W | – | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | – | 1 | 0 | 0 |

EAh.6 **TXDIE:** Slave I²C transmission completed interrupt enable
0: disable
1: enable

EAh.5 **RCD2IE:** Slave I²C DATA2(SITXRCD2) reception completed interrupt enable
0: disable
1: enable

EAh.4 **RCD1IE:** Slave I²C DATA1(SIRCD1) reception completed interrupt enable
0: disable
1: enable

EAh.2 **TXDF:** Slave I²C transmission completed interrupt flag
Set by H/W when Slave I²C transmission complete, write 0 to clear it

EAh.1 **RCD2F:** Slave I²C DATA2(SITXRCD2) reception completed interrupt flag
Set by H/W when Slave I²C DATA2(SITXRCD2) reception complete, write 0 to clear it

EAh.0 **RCD1F:** Slave I²C DATA1(SIRCD1) reception completed interrupt flag
Set by H/W when Slave I²C DATA1(SIRCD1) reception complete, write 0 to clear it

| SFR **EBh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **SIRCD1** | SIRCD1 | | | | | | | |
| R/W | R | R | R | R | R | R | R | R |
| Reset | – | – | – | – | – | – | – | – |

EBh.7~0 **SIRCD1:** Slave I²C data receive register1 (DATA1)

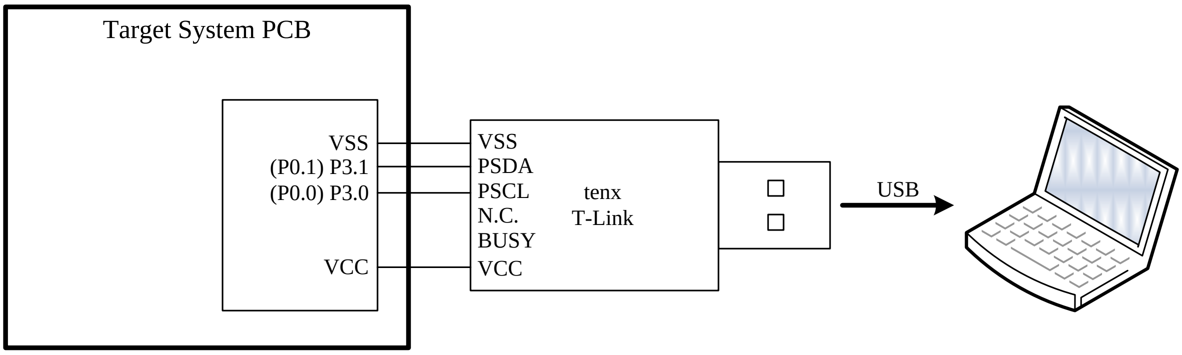
| SFR **ECh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **SITXRCD2** | SITXRCD2 | | | | | | | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | – | – | – | – | – | – | – | – |

ECh.7~0 **SITXRCD2:** Slave I²C transmit and receive data register
Read: Slave I²C data receive register2 (DATA2)
Write: Slave I²C data transmission register (TXD)

## 20. In Circuit Emulation (ICE) Mode

This device can support the In Circuit Emulation Mode. To use the ICE Mode, user just needs to connect P3.0 and P3.1 pin to the tenx proprietary EV Module. The benefit is that user can emulate the whole system without changing the on board target device. But there are some limits for the ICE mode as below.

1. The device must be un-protect.
2. The device’s P3.0 and P3.1 pins must work in input Mode (P3MOD0 = 0/1 and P3MOD1=0/1).
3. The Program Memory’s addressing space 6C00h~6FFFh and 0033h~003Ah are occupied by tenx EV module. So user Program cannot access these spaces.
4. The T-Link communication pin’s function cannot be emulated.
5. The P3.0 and P3.1 pin’s can be replaced by P0.0 and P0.1.
   (P0.0/P0.1 can only support ICE function, not for Writer)
6. SFR PWRSAV (F7h.5) will be cleared when use T-Link module.



# SFR & CFGW MAP

| Adr | RST | NAME | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 80h | 0000-0000 | **P0** | P0.7 | P0.6 | P0.5 | P0.4 | P0.3 | P0.2 | P0.1 | P0.0 |
| 81h | 0000-0111 | **SP** | SP | | | | | | | |
| 82h | 0000-0000 | **DPL** | DPL | | | | | | | |
| 83h | 0000-0000 | **DPH** | DPH | | | | | | | |
| 84h | 0000-0000 | **INTEX** | EX9 | EX8 | EX7 | EX6 | EX5 | EX4 | EX3 | EX2 |
| 85h | 0000-0000 | **INTEXF** | IE9 | IE8 | IE7 | IE6 | IE5 | IE4 | IE3 | IE2 |
| 86h | xxxx-x000 | **INTPWM** | – | – | – | – | – | PWM2IF | PWM1IF | PWM0IF |
| 87h | 0xxx-0000 | **PCON** | SMOD | – | – | – | GF1 | GF0 | PD | IDL |
| 88h | 0000-0000 | **TCON** | TF1 | TR1 | TF0 | TR0 | IE1 | IT1 | IE0 | IT0 |
| 89h | 0000-0000 | **TMOD** | GATE1 | CT1N | TMOD1 | | GATE0 | CT0N | TMOD0 | |
| 8Ah | 0000-0000 | **TL0** | TL0 | | | | | | | |
| 8Bh | 0000-0000 | **TL1** | TL1 | | | | | | | |
| 8Ch | 0000-0000 | **TH0** | TH0 | | | | | | | |
| 8Dh | 0000-0000 | **TH1** | TH1 | | | | | | | |
| 8Eh | 0100-0000 | **SCON2** | SM | – | – | REN2 | TB82 | RB82 | TI2 | RI2 |
| 8Fh | xxxx-xxxx | **SBUF2** | SBUF2 | | | | | | | |
| 90h | 1111-1111 | **P1** | P1.7 | P1.6 | P1.5 | P1.4 | P1.3 | P1.2 | P1.1 | P1.0 |
| 91h | 0000-0000 | **P0OE** | P0OE | | | | | | | |
| 92h | 0000-0000 | **P0LOE** | P0LOE | | | | | | | |
| 93h | 0000-0101 | **P2MOD** | – | – | – | – | P2MOD1 | | P2MOD0 | |
| 94h | 0000-0000 | **OPTION** | UART1W | – | WDTPSC | | ADCKS | | TM3PSC | |
| 95h | xx00-x000 | **INTFLG** | LVDIF | – | TKIF | ADIF | – | – | P1IF | TF3 |
| 96h | 0000-0000 | **P1WKUP** | P1WKUP | | | | | | | |
| 97h | xxxx-xx00 | **SWCMD** | IAPEN / SWRST / WDTO | | | | | | | |
| 98h | 0000-0000 | **SCON** | SM0 | SM1 | SM2 | REN | TB8 | RB8 | TI | RI |
| 99h | xxxx-xxxx | **SBUF** | SBUF | | | | | | | |
| 9Eh | 0000-0000 | **PWMOE** | PWM1IE | PWM0IE | – | – | – | PWM2OE | PWM1OE | PWM0OE |
| 9Fh | 0x00-0000 | **PWMCLR** | PWM2IE | – | – | – | – | PWM2CLR | PWM1CLR | PWM0CLR |
| A0h | 0000-0011 | **P2** | P2.7 | P2.6 | P2.5 | P2.4 | P2.3 | P2.2 | P2.1 | P2.0 |
| A1h | xx10-1010 | **PWMCON** | – | – | PWM2CKS | | PWM1CKS | | PWM0CKS | |
| A2h | 0101-0101 | **P1MODL** | P1MOD3 | | P1MOD2 | | P1MOD1 | | P1MOD0 | |
| A3h | 0101-0101 | **P1MODH** | P1MOD7 | | P1MOD6 | | P1MOD5 | | P1MOD4 | |
| A4h | 0101-0101 | **P3MODL** | P3MOD3 | | P3MOD2 | | P3MOD1 | | P3MOD0 | |
| A5h | 0001-0101 | **P3MODH** | P3MOD7 | | P3MOD6 | | P3MOD5 | | P3MOD4 | |
| A6h | 0000-xxx0 | **PINMOD** | – | I2CSEL | TCOE | T2OE | – | – | – | T0OE |
| A7h | xxx1-1111 | **TKCHS** | – | | | TKCHS | | | | |
| A8h | 0x00-0000 | **IE** | EA | – | ET2 | ES | ET1 | EX1 | ET0 | EX0 |
| A9h | xx00-0000 | **INTE1** | PWMIE | I2CE | ES2 | SPIE | ADTKIE | EXLVDIE | P1IE | TM3IE |
| AAh | xxxx-xxxx | **ADCDL** | ADCDL | | | | – | | | |
| ABh | xxxx-xxxx | **ADCDH** | ADCDH | | | | | | | |
| ADh | 1100-0100 | **TKCON** | TKPD | TKEOC | TKRERUN | TKIVCS | TKXCAP | TKOFFSET | ATKMODE | |
| AEh | 1111-000x | **CHSEL** | ADCHS | | | | ADCVREFS | | VBGEN | |
| AFh | 000x-xxxx | **P0ADIE** | P0ADIE | | | – | – | – | – | – |
| B0h | 1111-1111 | **P3** | P3.7 | P3.6 | P3.5 | P3.4 | P3.3 | P3.2 | P3.1 | P3.0 |
| B1h | 0000-x100 | **LEDCON** | LEDEN | | LEDPSC | | LEDHOLD | LEDBRIT | | |
| B2h | 0100-x100 | **LEDCON2** | LEDBRITM | LEDBRIT2 | | | – | LEDBRIT1 | | |
| B3h | 0000-0000 | **LEDCON3** | LEDMTEN | LED8EN | LED7EN | LED6EN | LED5EN | LED4EN | LED3EN | LED2EN |
| B4h | 0000-0000 | **TKTMRL** | TKTMRL | | | | | | | |
| B5h | 0000-0000 | **TKCON2** | TKFJMP | JMPVAL | | SPREAD | TKTMRH | | | |
| B8h | xx00-0000 | **IP** | – | – | PT2 | PS | PT1 | PX1 | PT0 | PX0 |
| B9h | xx00-0000 | **IPH** | – | – | PT2H | PSH | PT1H | PX1H | PT0H | PX0H |
| BAh | xx00-0000 | **IP1** | – | – | PS2 | PSPI | PADTKI | PX2_9LVD | PP1 | PT3 |
| BBh | xx00-0000 | **IP1H** | – | – | PS2H | PSPIH | PADTKIH | PX2_9LVDH | PP1H | PT3H |
| BCh | 0000-0000 | **SPCON** | SPEN | MSTR | CPOL | CPHA | SSDIS | LSBF | SPCR | |
| BDh | 0000-0xxx | **SPSTA** | SPIF | WCOL | MODF | RCVOVF | RCVBF | SPBSY | – | – |
| BEh | 0000-0000 | **SPDAT** | SPDAT | | | | | | | |
| BFh | 0xxx-0000 | **LVDS** | LVDIE | LVDO | – | – | LVDS | | | ENVPULL |
| C1h | 0000-0000 | **TKPINSEL0** | TKPINSEL0 | | | | | | | |

| Adr | RST | NAME | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| C2h | 0000-0000 | **TKPINSEL1** | TKPINSEL1 | | | | | | | |
| C3h | 0000-0000 | **TKPINSEL2** | TKPINSEL2 | | | | | | | |
| C5h | 0000-0000 | **ATKCH0** | ATKCH0 | | | | | | | |
| C6h | 0000-0000 | **ATKCH1** | ATKCH1 | | | | | | | |
| C7h | 0000-0000 | **ATKCH2** | ATKCH2 | | | | | | | |
| C8h | 0000-0000 | **T2CON** | TF2 | EXF2 | RCLK | TCLK | EXEN2 | TR2 | CT2N | CPRL2N |
| C9h | 00xx-xxxx | **IAPWE** | IAPWE / IAPTO | | | | | | | |
| CAh | 0000-0000 | **RCP2L** | RCP2L | | | | | | | |
| CBh | 0000-0000 | **RCP2H** | RCP2H | | | | | | | |
| CCh | 0000-0000 | **TL2** | TL2 | | | | | | | |
| CDh | 0000-0000 | **TH2** | TH2 | | | | | | | |
| CEh | 0000-0000 | **EXA2** | EXA2 | | | | | | | |
| CFh | 0000-0000 | **EXA3** | EXA3 | | | | | | | |
| D0h | 0000-0000 | **PSW** | CY | AC | F0 | RS1 | RS0 | OV | F1 | P |
| D1h | 1000-0000 | **PWM0DH** | PWM0DH | | | | | | | |
| D2h | 0000-0000 | **PWM0DL** | PWM0DL | | | | | | | |
| D3h | 1000-0000 | **PWM1DH** | PWM1DH | | | | | | | |
| D4h | 0000-0000 | **PWM1DL** | PWM1DL | | | | | | | |
| D5h | 1000-0000 | **PWM2DH** | PWM2DH | | | | | | | |
| D6h | 0000-0000 | **PWM2DL** | PWM2DL | | | | | | | |
| D8h | 00x0-0011 | **CLKCON** | SCKTYPE | FCKTYPE | STPSCK | STPPCK | STPFCK | SELFCK | CLKPSC | |
| D9h | 1111-1111 | **PWM0PRDH** | PWM0PRDH | | | | | | | |
| DAh | 1111-1111 | **PWM0PRDL** | PWM0PRDL | | | | | | | |
| DBh | 1111-1111 | **PWM1PRDH** | PWM1PRDH | | | | | | | |
| DCh | 1111-1111 | **PWM1PRDL** | PWM1PRDL | | | | | | | |
| DDh | 1111-1111 | **PWM2PRDH** | PWM2PRDH | | | | | | | |
| DEh | 1111-1111 | **PWM2PRDL** | PWM2PRDL | | | | | | | |
| E0h | 0000-0000 | **ACC** | ACC.7 | ACC.6 | ACC.5 | ACC.4 | ACC.3 | ACC.2 | ACC.1 | ACC.0 |
| E1h | 000x-0100 | **MICON** | MIEN | MIACKO | MIIF | MIACKI | MISTART | MISTOP | MICR | |
| E2h | 0000-0000 | **MIDAT** | MIDAT | | | | | | | |
| E6h | 0000-0000 | **EXA** | EXA | | | | | | | |
| E7h | 0000-0000 | **EXB** | EXB | | | | | | | |
| E9h | 0110-1000 | **SIADR** | SA | | | | | | | SIEN |
| EAh | 0000-x100 | **SICON** | MIIE | TXDIE | RCD2IE | RCD1IE | – | TXDF | RCD2F | RCD1F |
| EBh | xxxx-xxxx | **SIRCD1** | SIRCD1 | | | | | | | |
| ECh | xxxx-xxxx | **SITXRCD2** | SITXRCD2 | | | | | | | |
| F0h | 0000-0000 | **B** | B.7 | B.6 | B.5 | B.4 | B.3 | B.2 | B.1 | B.0 |
| F1h | 1111-1111 | **CRCDL** | CRCDL | | | | | | | |
| F2h | 1111-1111 | **CRCDH** | CRCDH | | | | | | | |
| F3h | 0000-0000 | **CRCIN** | CRCIN | | | | | | | |
| F5h | xxxx-xxxx | **CFGBG** | – | – | – | BGTRIM | | | | |
| F6h | xxxx-xxxx | **CFGWL** | – | FRCF | | | | | | |
| F7h | 0000-1110 | **AUX2** | WDTE | | PWRSAV | VBGOUT | DIV32 | IAPTE | | MULDIV16 |
| F8h | 0000-0000 | **AUX1** | CLRWDT | CLRTM3 | TKSOC | ADSOC | LVRPD | T2SEL | T1SEL | DPSEL |

| Flash Address | NAME | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|
| 7FFFh | **CFGWH** | PROTN | XRSTEN | LVRE | | | – | MVCLOCKN | FRCPSC |

# SFR & CFGW DESCRIPTION

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| 80h | **P0** | 7~0 | P0 | R/W | 00h | Port0 data |
| 81h | **SP** | 7~0 | SP | R/W | 07h | Stack Point |
| 82h | **DPL** | 7~0 | DPL | R/W | 00h | Data Point low byte |
| 83h | **DPH** | 7~0 | DPH | R/W | 00h | Data Point high byte |
| 84h | **INTEX** | 7 | EX9 | R/W | 0h | External INT9~INT2 pin Interrupt enable and Stop/Halt mode wake up enable.<br>0: Disable INTx pin Interrupt and Stop/Halt mode wake up<br>1: Enable INTx pin Interrupt and Stop/Halt mode wake up, it can wake up CPU from Stop/Halt mode no matter EA is 0 or 1<br>(note: EXLVDIE must be 1 at the same time to generate INTx interrupt and wake up) |
| | | 6 | EX8 | R/W | 0h | |
| | | 5 | EX7 | R/W | 0h | |
| | | 4 | EX6 | R/W | 0h | |
| | | 3 | EX5 | R/W | 0h | |
| | | 2 | EX4 | R/W | 0h | |
| | | 1 | EX3 | R/W | 0h | |
| | | 0 | EX2 | R/W | 0h | |
| 85h | **INTEXF** | 7 | IE9 | R/W | 0h | INT9 Interrupt Flag. 1: interrupt asserted, write 0 to clear int flag |
| | | 6 | IE8 | R/W | 0h | INT8 Interrupt Flag. 1: interrupt asserted, write 0 to clear int flag |
| | | 5 | IE7 | R/W | 0h | INT7 Interrupt Flag. 1: interrupt asserted, write 0 to clear int flag |
| | | 4 | IE6 | R/W | 0h | INT6 Interrupt Flag. 1: interrupt asserted, write 0 to clear int flag |
| | | 3 | IE5 | R/W | 0h | INT5 Interrupt Flag. 1: interrupt asserted, write 0 to clear int flag |
| | | 2 | IE4 | R/W | 0h | INT4 Interrupt Flag. 1: interrupt asserted, write 0 to clear int flag |
| | | 1 | IE3 | R/W | 0h | INT3 Interrupt Flag. 1: interrupt asserted, write 0 to clear int flag |
| | | 0 | IE2 | R/W | 0h | INT2 Interrupt Flag. 1: interrupt asserted, write 0 to clear int flag |
| 86h | **INTPWM** | 2 | PWM2IF | R/W | 0h | PWM2 Interrupt Flag. 1: interrupt asserted, write 0 to clear int flag |
| | | 1 | PWM1IF | R/W | 0h | PWM1 Interrupt Flag. 1: interrupt asserted, write 0 to clear int flag |
| | | 0 | PWM0IF | R/W | 0h | PWM0 Interrupt Flag. 1: interrupt asserted, write 0 to clear int flag |
| 87h | **PCON** | 7 | SMOD | R/W | 0 | Set 1 to enable UART1 double baud rate |
| | | 3 | GF1 | R/W | 0 | General purpose flag bit |
| | | 2 | GF0 | R/W | 0 | General purpose flag bit |
| | | 1 | PD | R/W | 0 | Power down control bit, set 1 to enter STOP (or Halt) mode |
| | | 0 | IDL | R/W | 0 | Idle control bit, set 1 to enter IDLE mode |
| 88h | **TCON** | 7 | TF1 | R/W | 0 | Timer1 overflow flag<br>Set by H/W when Timer/Counter 1 overflows. Cleared by H/W when CPU vectors into the interrupt service routine. |
| | | 6 | TR1 | R/W | 0 | Timer1 run control. 1: timer runs; 0: timer stops |
| | | 5 | TF0 | R/W | 0 | Timer0 overflow flag<br>Set by H/W when Timer/Counter 0 overflows. Cleared by H/W when CPU vectors into the interrupt service routine. |
| | | 4 | TR0 | R/W | 0 | Timer0 run control. 1:timer runs; 0:timer stops |
| | | 3 | IE1 | R/W | 0 | External Interrupt 1 (INT1 pin) edge flag<br>Set by H/W when an INT1 pin falling edge is detected. Cleared by H/W when CPU vectors into the interrupt service routine. |
| | | 2 | IT1 | R/W | 0 | External Interrupt 1 control bit<br>0: Low level active (level triggered) for INT1 pin<br>1: Falling edge active (edge triggered) for INT1 pin |
| | | 1 | IE0 | R/W | 0 | External Interrupt 0 (INT0 pin) edge flag<br>Set by H/W when an INT0 pin falling edge is detected. Cleared by H/W when CPU vectors into the interrupt service routine. |
| | | 0 | IT0 | R/W | 0 | External Interrupt 0 control bit<br>0: Low level active (level triggered) for INT0 pin<br>1: Falling edge active (edge triggered) for INT0 pin |

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| 89h | **TMOD** | 7 | GATE1 | R/W | 0 | Timer1 gating control bit<br>0: Timer1 enable when TR1 bit is set<br>1: Timer1 enable only while the INT1 pin is high and TR1 bit is set |
| | | 6 | CT1N | R/W | 0 | Timer1 Counter/Timer select bit<br>0: Timer mode, Timer1 data increases at 2 System clock cycle rate<br>1: Counter mode, Timer1 data increases at T1 pin's negative edge |
| | | 5~4 | TMOD1 | R/W | 00 | Timer1 mode select<br>00: 13-bit timer/counter<br>01: 16-bit timer/counter<br>10: 8-bit auto-reload timer/counter (TL1). Reloaded from TH1 at overflow.<br>11: Timer1 stops |
| | | 3 | GATE0 | R/W | 0 | Timer0 gating control bit<br>0: Timer0 enable when TR0 bit is set<br>1: Timer0 enable only while the INT0 pin is high and TR0 bit is set |
| | | 2 | CT0N | R/W | 0 | Timer0 Counter/Timer select bit<br>0: Timer mode, Timer0 data increases at 2 System clock cycle rate<br>1: Counter mode, Timer0 data increases at T0 pin's negative edge |
| | | 1~0 | TMOD0 | R/W | 00 | Timer0 mode select<br>00: 13-bit timer/counter<br>01: 16-bit timer/counter<br>10: 8-bit auto-reload timer/counter (TL0). Reloaded from TH0 at overflow.<br>11: TL0 is an 8-bit timer/counter.<br>TH0 is an 8-bit timer/counter using Timer1's TR1 and TF1 bits. |
| 8Ah | **TL0** | 7~0 | TL0 | R/W | 00h | Timer0 data low byte |
| 8Bh | **TL1** | 7~0 | TL1 | R/W | 00h | Timer1 data low byte |
| 8Ch | **TH0** | 7~0 | TH0 | R/W | 00h | Timer0 data high byte |
| 8Dh | **TH1** | 7~0 | TH1 | R/W | 00h | Timer1 data high byte |
| 8Eh | **SCON2** | 7 | SM | R/W | 0 | UART2 Serial port mode select bit<br>0: Mode1: 8 bit UART2, Baud Rate is variable<br>1: Mode3: 9 bit UART2, Baud Rate is variable |
| | | 4 | REN2 | R/W | 0 | UART2 reception enable<br>0: Disable reception<br>1: Enable reception |
| | | 3 | TB82 | R/W | 0 | Transmit Bit 8, the ninth bit to be transmitted in Mode3 |
| | | 2 | RB82 | R/W | 0 | Receive Bit 8, contains the ninth bit that was received in Mode3 |
| | | 1 | TI2 | R/W | 0 | Transmit interrupt flag<br>Set by H/W at the beginning of the stop bit in Mode 1 & 3. Must be cleared by S/W. |
| | | 0 | RI2 | R/W | 0 | Receive interrupt flag<br>Set by H/W at the sampling point of the stop bit in Mode 1 & 3. Must be cleared by S/W. |
| 8Fh | **SBUF2** | 7~0 | SBUF2 | R/W | – | UART2 transmit and receive data. Transmit data is written to this location and receive data is read from this location, but the paths are independent. |
| 90h | **P1** | 7~0 | P1 | R/W | FFh | Port1 data |
| 91h | **P0OE** | 7~0 | P0OE | R/W | 00h | Port0 CMOS Push-Pull output enable control<br>0: Disable<br>1: Enable |
| 92h | **P0LOE** | 7~0 | P0LOE | R/W | 00h | Port0 LCD 1/2 bias output enable control<br>0: Disable<br>1: Enable |
| 93h | **P2MOD** | 3~2 | P2MOD1 | R/W | 01 | P2.1 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2<br>11: not defined |
| | | 1~0 | P2MOD0 | R/W | 01 | P2.0 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2<br>11: not defined |

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| 94h | **OPTION** | 7 | UART1W | R/W | 0 | Set 1 to enable one wire UART1 mode, both TXD/RXD use P3.1 pin. |
| | | 5~4 | WDTPSC | R/W | 00 | Watchdog Timer pre-scalar time select<br>00: 480ms WDT overflow rate<br>01: 240ms WDT overflow rate<br>10: 120ms WDT overflow rate<br>11: 60ms WDT overflow rate |
| | | 3~2 | ADCKS | R/W | 00 | ADC clock rate select<br>00: $F_{SYSCLK}$ /32<br>01: $F_{SYSCLK}$ /16<br>10: $F_{SYSCLK}$ /8<br>11: $F_{SYSCLK}$ /4 |
| | | 1~0 | TM3PSC | R/W | 00 | Timer3 Interrupt rate<br>00: Timer3 Interrupt rate is 32768 Slow clock cycle<br>01: Timer3 Interrupt rate is 16384 Slow clock cycle<br>10: Timer3 Interrupt rate is 8192 Slow clock cycle<br>11: Timer3 Interrupt rate is 128 Slow clock cycle |
| 95h | **INTFLG** | 7 | LVDIF | R | – | Low Voltage Detect flag<br>Set by H/W when a low voltage occurs. |
| | | 5 | TKIF | R/W | 0 | Touch Key Interrupt Flag<br>Set by H/W at the end of TK conversion if SYSCLK is fast enough. S/W writes DFh to INTFLG or sets the TKSOC bit to clear this flag. |
| | | 4 | ADIF | R/W | 0 | ADC interrupt flag<br>Set by H/W at the end of ADC conversion. S/W writes EFh to INTFLG or sets the ADSOC bit to clear this flag. |
| | | 1 | P1IF | R/W | 0 | Port1 pin change Interrupt flag<br>Set by H/W when a Port1 pin state change is detected and its interrupt enable bit is set (P1WKUP). P1IE does not affect this flag’s setting. It is cleared automatically when the program performs the interrupt service routine. S/W can write FDh to INTFLG to clear this bit. |
| | | 0 | TF3 | R/W | 0 | Timer3 Interrupt Flag<br>Set by H/W when Timer3 reaches TM3PSC setting cycles. It is cleared automatically when the program performs the interrupt service routine. S/W can write FEh to INTFLG to clear this bit. |
| 96h | **P1WKUP** | 7~0 | P1WKUP | R/W | 00h | P1.7~P1.0 pin individual Wake-up/Interrupt enable control<br>0: Disable;<br>1: Enable. |
| 97h | **SWCMD** | 7~0 | SWRST | W | | Write 56h to generate S/W Reset |
| | | 7~0 | IAPEN | W | | Write 65h to set IAPEN control flag; Write other value to clear IAPEN flag. It is recommended to clear it immediately after IAP access. |
| | | 1 | WDTO | R | 0 | WatchDog Time-Out flag |
| | | 0 | IAPEN | R | 0 | Flag indicates Flash memory sectors can be accessed by IAP or not. This bit combines with MVCLOCK to define the accessible IAP area. |

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| 98h | **SCON** | 7 | SM0 | R/W | 0 | UART1 Serial port mode select bit 0, 1 (SM0, SM1) =<br>00: Mode0: 8 bit shift register, Baud Rate=$F_{SYSCLK}$/2<br>01: Mode1: 8 bit UART1, Baud Rate is variable<br>10: Mode2: 9 bit UART1, Baud Rate=$F_{SYSCLK}$/32 or /64<br>11: Mode3: 9 bit UART1, Baud Rate is variable |
| | | 6 | SM1 | R/W | 0 | |
| | | 5 | SM2 | R/W | 0 | Serial port mode select bit 2<br>SM2 enables multiprocessor communication over a single serial line and modifies the above as follows. In Modes 2 & 3, if SM2 is set then the received interrupt will not be generated if the received ninth data bit is 0. In Mode 1, the received interrupt will not be generated unless a valid stop bit is received. In Mode 0, SM2 should be 0. |
| | | 4 | REN | R/W | 0 | Set 1 to enable UART1 Reception |
| | | 3 | TB8 | R/W | 0 | Transmitter bit 8, ninth bit to transmit in Modes 2 and 3 |
| | | 2 | RB8 | R/W | 0 | Receive Bit 8, contains the ninth bit that was received in Mode 2 and 3 or the stop bit is Mode 1 if SM2=0 |
| | | 1 | TI | R/W | 0 | Transmit Interrupt flag<br>Set by H/W at the end of the eighth bit in Mode 0, or at the beginning of the stop bit in other modes. Must be cleared by S/W |
| | | 0 | RI | R/W | 0 | Receive Interrupt flag<br>Set by H/W at the end of the eighth bit in Mode 0, or at the sampling point of the stop bit in other modes. Must be cleared by S/W. |
| 99h | **SBUF** | 7~0 | SBUF | R/W | – | UART1 transmit and receive data. Transmit data is written to this location and receive data is read from this location, but the paths are independent. |
| 9Eh | **PWMOE** | 7 | PWM1IE | R/W | 0 | PWM1 Interrupt Enable.<br>0: disable<br>1: enalbe (note: PWMIE must be 1 at the same time to generate PWM interrupt) |
| | | 6 | PWM0IE | R/W | 0 | PWM0 Interrupt Enable<br>0: disable<br>1: enable (note: PWMIE must be 1 at the same time to generate PWM interrupt) |
| | | 2 | PWM2OE | R/W | 0 | PWM2 enable and signal output to P1.6 pin<br>0: disable<br>1: enable |
| | | 1 | PWM1OE | R/W | 0 | PWM1 enable and signal output to P1.3 pin<br>0: disable<br>1: enable |
| | | 0 | PWM0OE | R/W | 0 | PWM0 enable and signal output to P1.2 pin<br>0: disable<br>1: enable |
| 9Fh | **PWMCLR** | 7 | PWM2IE | R/W | 0 | PWM2 Interrupt Enable<br>0: disable<br>1: enable (note: PWMIE must be 1 at the same time to generate PWM interrupt) |
| | | 2 | PWM2CLR | R/W | 0 | PWM2 clear enable<br>0: PWM2 is running<br>1: PWM2 is cleared and held |
| | | 1 | PWM1CLR | R/W | 0 | PWM1 clear enable<br>0: PWM1 is running<br>1: PWM1 is cleared and held |
| | | 0 | PWM0CLR | R/W | 0 | PWM0 clear enable<br>0: PWM0 is running<br>1: PWM0 is cleared and held |
| A0h | **P2** | 7~0 | P2 | R/W | 00h | P2 data |

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| A1h | **PWMCON** | 5~4 | PWM2CKS | R/W | 10 | PWM2 clock source<br>00: $F_{SYSCLK}$<br>01: $F_{SYSCLK}$<br>10: FRC<br>11: FRC x 2 |
| | | 3~2 | PWM1CKS | R/W | 10 | PWM1 clock source<br>00: $F_{SYSCLK}$<br>01: $F_{SYSCLK}$<br>10: FRC<br>11: FRC x 2 |
| | | 1~0 | PWM0CKS | R/W | 10 | PWM0 clock source<br>00: $F_{SYSCLK}$<br>01: $F_{SYSCLK}$<br>10: FRC<br>11: FRC x 2 |
| A2h | **P1MODL** | 7~6 | P1MOD3 | R/W | 01 | P1.3 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2<br>11: Mode3, P1.3 is ADC input |
| | | 5~4 | P1MOD2 | R/W | 01 | P1.2 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2<br>11: Mode3, P1.2 is ADC input |
| | | 3~2 | P1MOD1 | R/W | 01 | P1.1 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2<br>11: Mode3, P1.1 is ADC input |
| | | 1~0 | P1MOD0 | R/W | 01 | P1.0 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2<br>11: Mode3, P1.0 is ADC input |
| A3h | **P1MODH** | 7~6 | P1MOD7 | R/W | 01 | P1.7 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2<br>11: Mode3 |
| | | 5~4 | P1MOD6 | R/W | 01 | P1.6 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2<br>11: Mode3 |
| | | 3~2 | P1MOD5 | R/W | 01 | P1.5 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2<br>11: Mode3, P1.5 is ADC input |
| | | 1~0 | P1MOD4 | R/W | 01 | P1.4 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2<br>11: Mode3, P1.4 is ADC input |
| A4h | **P3MODL** | 7~6 | P3MOD3 | R/W | 01 | P3.3 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2<br>11: Mode3, P3.3 is ADC input |
| | | 5~4 | P3MOD2 | R/W | 01 | P3.2 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2<br>11: Mode3, P3.2 is ADC input |
| | | 3~2 | P3MOD1 | R/W | 01 | P3.1 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2<br>11: Mode3, P3.1 is ADC input |
| | | 1~0 | P3MOD0 | R/W | 01 | P3.0 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2<br>11: Mode3, P3.0 is ADC input |
| A5h | **P3MODH** | 7~6 | P3MOD7 | R/W | 00 | P3.7 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3 |
| | | 5~4 | P3MOD6 | R/W | 01 | P3.6 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3 |
| | | 3~2 | P3MOD5 | R/W | 01 | P3.5 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3 |
| | | 1~0 | P3MOD4 | R/W | 01 | P3.4 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3 |

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| A6h | **PINMOD** | 6 | I2CSEL | R/W | 0 | I2C Pin Select<br>0: SCL/SDA = P3.4/P3.5<br>1: SCL/SDA = P3.0/P3.1 |
| | | 5 | TCOE | R/W | 0 | Set 1 to enable "System clock divided by 2" (CKO) output to P1.4 pin |
| | | 4 | T2OE | R/W | 0 | Set 1 to enable "Timer2 overflow divided by 2" (T2O) output to P1.0 pin |
| | | 0 | T0OE | R/W | 0 | Set 1 to enable "Timer0 overflow divided by 64" (T0O) output to P3.4 pin |
| A7h | **TKCHS** | 4~0 | TKCHS | R/W | 1Fh | Specify the first touch key scan channel<br>00000: TK0 (P3.3)<br>00001: TK1 (P3.2)<br>00010: TK2 (P3.1)<br>00011: TK3 (P3.0)<br>00100: TK4 (P1.0)<br>00101: TK5 (P1.1)<br>00110: TK6 (P1.2)<br>00111: TK7 (P1.3)<br>01000: TK8 (P1.4)<br>01001: TK9 (P1.6)<br>01010: TK10 (P1.7)<br>01011: TK11 (P3.6)<br>01100: TK12 (P3.5)<br>01101: TK13 (P3.4)<br>01110: TK14 (P1.5)<br>01111: TK15 (P3.7)<br>10000: TK16 (P0.3)<br>10001: TK17 (P0.5)<br>10010: TK18 (P0.6)<br>10011: TK19 (P0.7)<br>10111: TK reference capacitor |
| A8h | **IE** | 7 | EA | R/W | 0 | Global interrupt enable control.<br>0: Disable all Interrupts.<br>1: Each interrupt is enabled or disabled by its own interrupt control bit. |
| | | 5 | ET2 | R/W | 0 | Set 1 to enable Timer2 interrupt |
| | | 4 | ES | R/W | 0 | Set 1 to enable Serial Port (UART1) Interrupt |
| | | 3 | ET1 | R/W | 0 | Set 1 to enable Timer1 Interrupt |
| | | 2 | EX1 | R/W | 0 | Set 1 to enable external INT1 pin Interrupt & Stop/Halt mode wake up capability |
| | | 1 | ET0 | R/W | 0 | Set 1 to enable Timer0 Interrupt |
| | | 0 | EX0 | R/W | 0 | Set 1 to enable external INT0 pin Interrupt & Stop/Halt mode wake up capability |
| A9h | **INTE1** | 7 | PWMIE | R/W | 0 | Set 1 to enable PWM0~PWM2 interrupt |
| | | 6 | I2CE | R/W | 0 | Set 1 to enable I$^{2}$C (master/slave) interrupt |
| | | 5 | ES2 | R/W | 0 | Set 1 to enable Serial Port (UART2) interrupt |
| | | 4 | SPIE | R/W | 0 | Set 1 to enable SPI interrupt |
| | | 3 | ADTKIE | R/W | 0 | Set 1 to enable ADC/Touch Key Interrupt |
| | | 2 | EXLVDIE | R/W | 0 | Set 1 to enable external INT2~INT9 pin Interrupt, Stop/Halt mode wake up capability and LVD interrupt. |
| | | 1 | P1IE | R/W | 0 | Set 1 to enable Port1 Pin Change Interrupt |
| | | 0 | TM3IE | R/W | 0 | Set 1 to enable Timer3 Interrupt and Halt mode wake up enable, it can wake up CPU from Halt mode no matter EA is 0 or 1. |
| AAh | **ADCDL** | 7~4 | ADCDL | R | – | ADC data bit 3~0 |
| ABh | **ADCDH** | 7~0 | ADCDH | R | – | ADC data bit 11~4 |

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| ADh | **TKCON** | 7 | TKPD | R/W | 1 | Touch Key Power Down<br>0: Touch Key enable;<br>1: Touch Key disable |
| | | 6 | TKEOC | R | 1 | Touch Key end of conversion flag<br>0: Indicates conversion is in progress<br>1: Indicates conversion is finished |
| | | 5 | TKRERUN | R/W | 0 | TK Auto re-start, doesn't need to set TKSOC again to restart TK converter.<br>0: Auto re-start disable. TKSOC needs to be executed once for each TK conversion<br>1: Auto re-start enable. After TKSOC is executed once, TK will be converted continuously without re-executing TKSOC |
| | | 4 | TKIVCS | R/W | 0 | Touch Key internal voltage control select<br>0: VCHG=2.8V; VINT=1.4V<br>1: VCHG=3.6V; VINT=1.8V |
| | | 3 | TKXCAP | R/W | 0 | Touch Key external capacitor select<br>0: Keep 0, disable Touch Key external capacitor<br>1: reserved (Do not set to 1) |
| | | 2 | TKOFFSET | R/W | 0 | status of non-scan TK<br>0: connect to VSS<br>1: connect to AC shielding , connect to VSS@EOC |
| | | 1~0 | ATKMODE | R/W | 00 | Touch Key Scan Mode<br>00: TK scan method, each channel scan 1 time, max 21 TK channels<br>01: TK scan method, each channel scan 2 times, max 16 TK channels<br>10: TK scan method, each channel scan 4 times, max 8 TK channels<br>11: TK scan method, each channel scan 8 times, max 4 TK channels |
| AEh | **CHSEL** | 7~4 | ADCHS | R/W | 1111 | ADC channel select<br>0000: AD0 (P3.3)<br>0001: AD1 (P3.2)<br>0010: AD2 (P3.1)<br>0011: AD3 (P3.0)<br>0100: AD4 (P1.0)<br>0101: AD5 (P1.1)<br>0110: AD6 (P1.2)<br>0111: AD7 (P1.3)<br>1000: AD8 (P1.4)<br>1001: AD9 (P1.5)<br>1010: Reserved<br>1011: $V_{BG}$ (Internal Bandgap Reference Voltage)<br>1100: AD12 (P0.7)<br>1101: AD13 (P0.5)<br>1110: AD14 (P0.6)<br>1111: 1/4 $V_{CC}$ |
| | | 3~2 | ADCVREFS | R/W | 00 | ADC reference voltage. When ADCHS is selected to VBG, ADCVREFS must be set to VCC, otherwise ADC conversion will be invalid<br>00: VCC<br>01: $V_{BG}$*2.06V<br>10: Reserved<br>11: Reserved |
| | | 1 | VBGEN | R/W | 0 | force VBG generator enable<br>0: VBG generator is automatically enable and disable<br>1: Force VBG generator enable included in IDLE mode but disabled in stop mode |
| AFh | **P0ADIE** | 7~5 | P0ADIE | R/W | 000 | ADC channel input Enable<br>000: P0.7~P0.4 are digital input<br>1xx: P0.7 is ADC input<br>x1x: P0.6 is ADC input<br>xx1: P0.5 is ADC input |
| B0h | **P3** | 7~0 | P3 | R/W | FFh | Port3 data |

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| B1h | **LEDCON** | 7~6 | LEDEN | R/W | 00 | LED BiD matrix mode enable and duty select<br>00: LED BiD matrix mode disable<br>01: LED 1/8 duty (4COM x 4SEG)<br>10: LED 1/9 duty (4COM x 5SEG)<br>11: LED 1/10 duty (4COM x 6SEG) |
| | | 5~4 | LEDPSC | R/W | 00 | LED clock prescaler select<br>00: LED clock is FRC divided by 64<br>01: LED clock is FRC divided by 32<br>10: LED clock is FRC divided by 16<br>11: LED clock is FRC divided by 8 |
| | | 3 | LEDHOLD | R/W | 0 | LED clock hold<br>0: LED scan<br>1: LED clock hold |
| | | 2~0 | LEDBRIT | R/W | 100 | BiD matrix mode: LED number 0~31, 40~47 brightness control<br>000: Level 0 (Darkest)<br>…<br>111: Level 7 (Brightest) |
| B2h | **LEDCON2** | 7 | LEDBRITM | R/W | 0 | Brightness smooth control<br>0: Uniform brightness mode<br>1: Brightness enhancement mode |
| | | 6~4 | LEDBRIT2 | R/W | 100 | BiD matrix mode: LED number 33, 35, 37, 39 brightness control<br>Dot matrix mode: LED number 0~63 brightness control<br>000: Level 0 (Darkest)<br>…<br>111: Level 7 (Brightest) |
| | | 2~0 | LEDBRIT1 | R/W | 100 | BiD matrix mode: LED number 32, 34, 36, 38 brightness control<br>000: Level 0 (Darkest)<br>…<br>111: Level 7 (Brightest) |
| B3h | **LEDCON3** | 7 | LEDMTEN | R/W | 0 | LED Dot matrix mode enable<br>0: disable<br>1: enable |
| | | 6 | LED8EN | R/W | 0 | LED Dot matrix mode enable<br>0: LED8 disable<br>1: LED8 enable |
| | | 5 | LED7EN | R/W | 0 | LED Dot matrix mode enable<br>0: LED7 disable<br>1: LED7 enable |
| | | 4 | LED6EN | R/W | 0 | LED Dot matrix mode enable<br>0: LED6 disable<br>1: LED6 enable |
| | | 3 | LED5EN | R/W | 0 | LED Dot matrix mode enable<br>0: LED5 disable<br>1: LED5 enable |
| | | 2 | LED4EN | R/W | 0 | LED Dot matrix mode enable<br>0: LED4 disable<br>1: LED4 enable |
| | | 1 | LED3EN | R/W | 0 | LED Dot matrix mode enable<br>0: LED3 disable<br>1: LED3 enable |
| | | 0 | LED2EN | R/W | 0 | LED Dot matrix mode enable<br>0: LED2 disable<br>1: LED2 enable |
| B4h | **TKTMRL** | 7~0 | TKTMRL | R/W | 0 | Touch Key Scan length bit 7~0 adjustment.<br>00: shortest, FF: longest |

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| B5h | **TKCON2** | 7 | TKFJMP | R/W | 0 | Internal Touch Key clock frequency auto adjust option<br>0: Disable<br>1: Enable |
| | | 6~5 | JAMVAL | R/W | 0 | Touch Key Clock frequency fine tune , only available in TKFJMP=0<br>00=frequency slowest, 11=frequency  fastest |
| | | 4 | SPREAD | R/W | 0 | TK spread spectrum<br>0: Disable<br>1: Enable |
| | | 3~0 | TKTMRH | R/W | 0 | Touch Key Scan length 11~8 adjustment.<br>0000: shortest, 1111: longest |
| B8h | **IP** | 5 | PT2 | R/W | 0 | Timer2 Interrupt Priority Low bit |
| | | 4 | PS | R/W | 0 | Serial Port (UART1) Interrupt Priority Low bit |
| | | 3 | PT1 | R/W | 0 | Timer1 Interrupt Priority Low bit |
| | | 2 | PX1 | R/W | 0 | External INT1 Pin Interrupt Priority Low bit |
| | | 1 | PT0 | R/W | 0 | Timer0 Interrupt Priority Low bit |
| | | 0 | PX0 | R/W | 0 | External INT0 Pin Interrupt Priority Low bit |
| B9h | **IPH** | 5 | PT2H | R/W | 0 | Timer2 Interrupt Priority High bit |
| | | 4 | PSH | R/W | 0 | Serial Port (UART1) Interrupt Priority High bit |
| | | 3 | PT1H | R/W | 0 | Timer1 Interrupt Priority High bit |
| | | 2 | PX1H | R/W | 0 | External INT1 Pin Interrupt Priority High bit |
| | | 1 | PT0H | R/W | 0 | Timer0 Interrupt Priority High bit |
| | | 0 | PX0H | R/W | 0 | External INT0 Pin Interrupt Priority High bit |
| BAh | **IP1** | 7 | PPWM | R/W | 0 | PWM Interrupt Priority Low bit |
| | | 6 | PI2C | R/W | 0 | I2C Interrupt Priority Low bit |
| | | 5 | PS2 | R/W | 0 | Serial Port (UART2) interrupt priority low bit |
| | | 4 | PSPI | R/W | 0 | SPI interrupt priority low bit |
| | | 3 | PADTKI | R/W | 0 | ADC/Touch Key Interrupt Priority Low bit |
| | | 2 | PX2_9LVD | R/W | 0 | External INT2~INT9 Pin Interrupt Priority Low bit |
| | | 1 | PP1 | R/W | 0 | Port1 pin change Interrupt Priority Low bit |
| | | 0 | PT3 | R/W | 0 | Timer3 Interrupt Priority Low bit |
| BBh | **IP1H** | 7 | PPWMH | R/W | 0 | PWM Interrupt Priority High bit |
| | | 6 | PI2CH | R/W | 0 | I2C Interrupt Priority High bit |
| | | 5 | PS2H | R/W | 0 | Serial Port (UART2) interrupt priority high bit |
| | | 4 | PSPIH | R/W | 0 | SPI interrupt priority high bit |
| | | 3 | PADTKIH | R/W | 0 | ADC/Touch Key Interrupt Priority High bit |
| | | 2 | PX2_9LVDH | R/W | 0 | External INT2~INT9 Pin Interrupt Priority High bit |
| | | 1 | PP1H | R/W | 0 | Port1 Interrupt Priority High bit |
| | | 0 | PT3H | R/W | 0 | Timer3 Interrupt Priority High bit |

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| BCh | **SPCON** | 7 | SPEN | R/W | 0 | SPI enable<br>0: SPI disable<br>1: SPI enable |
| | | 6 | MSTR | R/W | 0 | Master mode enable<br>0: Slave mode<br>1: Master mode |
| | | 5 | CPOL | R/W | 0 | SPI clock polarity<br>0: SCK is low in idle state<br>1: SCK is high in idle state |
| | | 4 | CPHA | R/W | 0 | SPI clock phase<br>0: Data sample on first edge of SCK period<br>1: Data sample on second edge of SCK period |
| | | 3 | SSDIS | R/W | 0 | SS pin disable<br>0: Enable SS pin<br>1: Disable SS pin |
| | | 2 | LSBF | R/W | 0 | LSB first<br>0: MSB first<br>1: LSB first |
| | | 1~0 | SPCR | R/W | 00 | SPI clock rate<br>00: FSYSCLK/2<br>01: FSYSCLK/4<br>10: FSYSCLK/8<br>11: FSYSCLK/16 |
| BDh | **SPSTA** | 7 | SPIF | R/W | 0 | SPI interrupt flag<br>This is set by H/W at the end of a data transfer. Cleared by H/W when an interrupt is vectored into. Writing 0 to this bit will clear this flag. |
| | | 6 | WCOL | R/W | 0 | Write collision interrupt flag<br>Set by H/W if write data to SPDAT when SPBSY is set. Write 0 to this bit or rewrite data to SPDAT when SPBSY is cleared will clear this flag. |
| | | 5 | MODF | R/W | 0 | Mode fault interrupt flag<br>Set by H/W when SSDIS is cleared and SS pin is pulled low in Master mode. Write 0 to this bit will clear this flag. When this bit is set, the SPEN and MSTR in SPCON will be cleared by H/W. |
| | | 4 | RCVOVF | R/W | 0 | Received buffer overrun flag<br>Set by H/W at the end of a data transfer and RCVBF is set. Write 0 to this bit or read SPDAT register will clear this flag. |
| | | 3 | RCVBF | R/W | 0 | Receive buffer full flag<br>Set by H/W at the end of a data transfer. Write 0 to this bit or read SPDAT register will clear this flag. |
| | | 2 | SPBSY | R | 0 | SPI busy flag<br>Set by H/W when a SPI transfer is in progress. |
| BEh | **SPDAT** | 7~0 | SPDAT | R/W | 0 | SPI transmit and receive data<br>The SPDAT register is used to transmit and receive data. Writing data to SPDAT place the data into shift register and start a transfer when in master mode. Reading SPDAT returns the contents of the receive buffer. |

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| BFh | **LVDS** | 7 | LVDIE | R/W | 0 | Low Voltage Detect interrupt enable<br>0: Disable<br>1: Enable (note: EXLVDIE must be 1 at the same time to generate LVD interrupt) |
| | | 6 | LVDO | R | - | Low Voltage Detect output |
| | | 3~1 | LVDS | R/W | 0 | Low Voltage Detect select<br>000: Set LVD at 2.3V<br>001: Set LVD at 2.54V<br>010: Set LVD at 2.78V<br>011: Set LVD at 3.04V<br>100: Set LVD at 3.28V<br>101: Set LVD at 3.54V<br>110: Set LVD at 3.8V<br>111: Set LVD at 4.04V |
| | | 0 | ENVPULL | R/W | 0 | Power control, force VPULL enable, Must be set to 0<br>0: Disable<br>1: Don’t use, cannot be set to 1 |
| C1h | **TKPINSEL0** | 7~0 | TKPINSEL0 | R/W | 00 | Touch Key TK7~TK0 Channel Select<br>0: Normal IO<br>1: Touch Key |
| C2h | **TKPINSEL1** | 7~0 | TKPINSEL1 | R/W | 00 | Touch Key TK15~TK8 Channel Select<br>0: Normal IO<br>1: Touch Key |
| C3h | **TKPINSEL2** | 7~0 | TKPINSEL2 | R/W | 00 | Touch Key TK23~TK16 Channel Select<br>0: Normal IO<br>1: Touch Key |
| C5h | **ATKCH0** | 7~0 | ATKCH0 | R/W | 00 | Auto Touch Key TK7~TK0 Channel Select<br>0: Disable auto scan<br>1: Enable auto scan |
| C6h | **ATKCH1** | 7~0 | ATKCH1 | R/W | 00 | Auto Touch Key TK15~TK8 Channel Select<br>0: Disable auto scan<br>1: Enable auto scan |
| C7h | **ATKCH2** | 7~0 | ATKCH2 | R/W | 00 | Auto Touch Key TK23~TK16 Channel Select<br>0: Disable auto scan<br>1: Enable auto scan |

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| C8h | **T2CON** | 7 | TF2 | R/W | 0 | Timer2 overflow flag<br>Set by H/W when Timer/Counter 2 overflows unless RCLK=1 or TCLK=1. This bit must be cleared by S/W. |
| | | 6 | EXF2 | R/W | 0 | T2EX interrupt pin falling edge flag<br>Set when a capture or a reload is caused by a negative transition on T2EX pin if EXEN2=1. This bit must be cleared by S/W. |
| | | 5 | RCLK | R/W | 0 | UART receive clock control bit<br>0: Use Timer1 overflow as receive clock for serial port in mode 1 or 3<br>1: Use Timer2 overflow as receive clock for serial port in mode 1 or 3 |
| | | 4 | TCLK | R/W | 0 | UART transmit clock control bit<br>0: Use Timer1 overflow as transmit clock for serial port in mode 1 or 3<br>1: Use Timer2 overflow as transmit clock for serial port in mode 1 or 3 |
| | | 3 | EXEN2 | R/W | 0 | T2EX pin enable<br>0: T2EX pin disable<br>1: T2EX pin enable, it cause a capture or reload when a negative transition on T2EX pin is detected if RCLK=TCLK=0 |
| | | 2 | TR2 | R/W | 0 | Timer2 run control<br>0:timer stops<br>1:timer runs |
| | | 1 | CT2N | R/W | 0 | Timer2 Counter/Timer select bit<br>0: Timer mode, Timer2 data increases at 2 System clock cycle rate<br>1: Counter mode, Timer2 data increases at T2 pin's negative edge |
| | | 0 | CPRL2N | R/W | 0 | Timer2 Capture/Reload control bit<br>0: Reload mode, auto-reload on Timer2 overflows or negative transitions on T2EX pin if EXEN2=1.<br>1: Capture mode, capture on negative transitions on T2EX pin if EXEN2=1.<br>If RCLK=1 or TCLK=1, CPRL2N is ignored and timer is forced to auto-reload on Timer2 overflow. |
| C9h | **IAPWE** | 7~0 | IAPWE | W | – | Write 4Ah to enable one byte IAP write to ROM[7A00~7BFF]<br>Write 4Ch to enable one byte IAP write to ROM[7C00~7DFF]<br>Write BAh to enable ERASE 512 byte of ROM[7A00~7BFF]<br>Write BCh to enable ERASE 512 byte of ROM[7C00~7DFF]<br>Write other value to disable IAP write |
| | | 7 | IAPWE | R | 0 | Flag indicates Flash memory can be written by IAP or not<br>0: IAP Write/Erase disable<br>1: IAP Write/Erase enable |
| C9h | **IAPWE** | 6 | IAPTO | R | 0 | IAP (or EEPROM write) Time-Out flag<br>Set by H/W when IAP (or EEPROM write) Time-out occurs.<br>Cleared by H/W when IAPWE=0 (or EEPWE=0). |
| CAh | **RCP2L** | 7~0 | RCP2L | R/W | 00h | Timer2 reload/capture data low byte |
| CBh | **RCP2H** | 7~0 | RCP2H | R/W | 00h | Timer2 reload/capture data high byte |
| CCh | **TL2** | 7~0 | TL2 | R/W | 00h | Timer2 data low byte |
| CDh | **TH2** | 7~0 | TH2 | R/W | 00h | Timer2 data high byte |
| CEh | **EXA2** | 7~0 | EXA2 | R/W | 00h | Expansion accumulator 2 |
| CFh | **EXA3** | 7~0 | EXA3 | R/W | 00h | Expansion accumulator 3 |
| D0h | **PSW** | 7 | CY | R/W | 0 | ALU carry flag |
| | | 6 | AC | R/W | 0 | ALU auxiliary carry flag |
| | | 5 | F0 | R/W | 0 | General purpose user-definable flag |
| | | 4 | RS1 | R/W | 0 | Register Bank Select bit 1 |
| | | 3 | RS0 | R/W | 0 | Register Bank Select bit 0 |
| | | 2 | OV | R/W | 0 | ALU overflow flag |
| | | 1 | F1 | R/W | 0 | General purpose user-definable flag |
| | | 0 | P | R/W | 0 | Parity flag |
| D1h | **PWM0DH** | 7~0 | PWM0DH | R/W | 80H | PWM0 duty high byte<br>write sequence: PWM0DL then PWM0DH<br>read sequence: PWM0DH then PWM0DL |
| D2h | **PWM0DL** | 7~0 | PWM0DL | R/W | 00H | PWM0 duty low byte<br>write sequence: PWM0DL then PWM0DH<br>read sequence: PWM0DH then PWM0DL |

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| D3h | **PWM1DH** | 7~0 | PWM1DH | R/W | 80H | PWM1 duty high byte<br>write sequence: PWM1DL then PWM1DH<br>read sequence: PWM1DH then PWM1DL |
| D4h | **PWM1DL** | 7~0 | PWM1DL | R/W | 00H | PWM1 duty low byte<br>write sequence: PWM1DL then PWM1DH<br>read sequence: PWM1DH then PWM1DL |
| D5h | **PWM2DH** | 7~0 | PWM2DH | R/W | 80H | PWM2 duty high byte<br>write sequence: PWM2DL then PWM2DH<br>read sequence: PWM2DH then PWM2DL |
| D6h | **PWM2DL** | 7~0 | PWM2DL | R/W | 00H | PWM2 duty low byte<br>write sequence: PWM2DL then PWM2DH<br>read sequence: PWM2DH then PWM2DL |
| D8h | **CLKCON** | 7 | SCKTYPE | R/W | 0 | Slow clock Type. This bit can be changed only in Fast mode (SELFCK=1)<br>0: SRC<br>1: SXT, P2.0 and P2.1 are crystal pins |
| | | 6 | FCKTYPE | R/W | 0 | Fast clock type. This bit can be changed only in Slow mode (SELFCK=0).<br>0: FRC<br>1: FXT, P2.0 and P2.1 are crystal pins, oscillator gain is high for FXT |
| | | 5 | STPSCK | R/W | 1 | Set 1 to stop Slow clock after PD=1 (Halt / Stop mode control) |
| | | 4 | STPPCK | R/W | 0 | Set 1 to stop UART/Timer0/1/2 clock in Idle mode for current reducing. |
| | | 3 | STPFCK | R/W | 0 | Set 1 to stop Fast clock for power saving in Slow/Idle mode. This bit can be changed only in Slow mode. |
| | | 2 | SELFCK | R/W | 0 | System clock select. This bit can be changed only when STPFCK=0.<br>0: Slow clock<br>1: Fast clock |
| | | 1~0 | CLKPSC | R/W | 11 | System clock prescaler. Effective after 16 clock cycles (Max.) delay.<br>00: System clock is Fast/Slow clock divided by 16<br>01: System clock is Fast/Slow clock divided by 4<br>10: System clock is Fast/Slow clock divided by 2<br>11: System clock is Fast/Slow clock divided by 1 |
| D9h | **PWM0PRDH** | 7~0 | PWM0PRDH | R/W | FFH | PWM0 period high byte<br>write sequence: PWM0PRDL then PWM0PRDH<br>read sequence: PWM0PRDH then PWM0PRDL |
| DAh | **PWM0PRDL** | 7~0 | PWM0PRDL | R/W | FFH | PWM0 period low byte<br>write sequence: PWM0PRDL then PWM0PRDH<br>read sequence: PWM0PRDH then PWM0PRDL |
| DBh | **PWM1PRDH** | 7~0 | PWM1PRDH | R/W | FFH | PWM1 period high byte<br>write sequence: PWM1PRDL then PWM1PRDH<br>read sequence: PWM1PRDH then PWM1PRDL |
| DCh | **PWM1PRDL** | 7~0 | PWM1PRDL | R/W | FFH | PWM1 period low byte<br>write sequence: PWM1PRDL then PWM1PRDH<br>read sequence: PWM1PRDH then PWM1PRDL |
| DDh | **PWM2PRDH** | 7~0 | PWM2PRDH | R/W | FFH | PWM2 period high byte<br>write sequence: PWM2PRDL then PWM2PRDH<br>read sequence: PWM2PRDH then PWM2PRDL |
| DEh | **PWM2PRDL** | 7~0 | PWM2PRDL | R/W | FFH | PWM2 period low byte<br>write sequence: PWM2PRDL then PWM2PRDH<br>read sequence: PWM2PRDH then PWM2PRDL |
| E0h | **ACC** | 7~0 | ACC | R/W | 00h | Accumulator |

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| E1h | **MICON** | 7 | MIEN | R/W | 0 | Master I$^2$C enable<br>0: disable<br>1: enable |
| | | 6 | MIACKO | R/W | 0 | When Master I$^2$C receive data, send acknowledge to I$^2$C Bus<br>0: ACK to slave device<br>1: NACK to slave device |
| | | 5 | MIIF | R/W | 0 | Master I$^2$C Interrupt flag<br>0: write 0 to clear it<br>1: Master I$^2$C transfer one byte complete |
| | | 4 | MIACKI | R | – | When Master I$^2$C transfer, acknowledgement form I$^2$C bus (read only)<br>0: ACK received<br>1: NACK received |
| | | 3 | MISTART | R/W | 0 | Master I$^2$C Start bit<br>1: start I$^2$C bus transfer |
| | | 2 | MISTOP | R/W | 1 | Master I$^2$C Stop bit<br>1: send STOP signal to stop I$^2$C bus |
| | | 1~0 | MICR | R/W | 00 | Master I$^2$C (SCL) clock frequency selection<br>00: Fsys/4 (ex. If Fsys=18MHz, I$^2$C clock is 4.5M Hz)<br>01: Fsys/16 (ex. If Fsys=18MHz, I$^2$C clock is 1.1M Hz)<br>10: Fsys/64 (ex. If Fsys=18MHz, I$^2$C clock is 281K Hz)<br>11: Fsys/256 (ex. If Fsys=18MHz, I$^2$C clock is 70K Hz) |
| E2h | **MIDAT** | 7~0 | MIDAT | R/W | 00 | Master I$^2$C data shift register<br>(W): After Start and before Stop condition, write this register will resume transmission to I$^2$C bus<br>(R): After Start and before Stop condition, read this register will resume receiving from I$^2$C bus |
| E6h | **EXA** | 7~0 | EXA | R/W | 00h | Expansion accumulator |
| E7h | **EXB** | 7~0 | EXB | R/W | 00h | Expansion B register |
| E9h | **SIADR** | 7~1 | SA | R/W | 64h | Slave I$^2$C address assigned |
| | | 0 | SIEN | R/W | 0 | Slave I$^2$C enable<br>0: disable<br>1: enable |
| EAh | **SICON** | 7 | MIIE | R/W | 0 | I$^2$C Master interrupt enable<br>0: disable<br>1: enable |
| | | 6 | TXDIE | R/W | 0 | Slave I$^2$C transmission completed interrupt enable<br>0: disable<br>1: enable |
| | | 5 | RCD2IE | R/W | 0 | Slave I$^2$C DATA2(SITXRCD2) reception completed interrupt enable<br>0: disable<br>1: enable |
| | | 4 | RCD1IE | R/W | 0 | Slave I$^2$C DATA1(SIRCD1) reception completed interrupt enable<br>0: disable<br>1: enable |
| | | 2 | TXDF | R/W | 1 | Slave I$^2$C transmission completed interrupt flag<br>0: write 0 to clear it<br>1: Set by H/W when Slave I$^2$C transmission complete |
| | | 1 | RCD2F | R/W | 0 | Slave I$^2$C DATA2(SITXRCD2) reception completed interrupt flag<br>0: write 0 to clear it<br>1: Set by H/W when Slave I$^2$C DATA2(SITXRCD2) reception complete enable |
| | | 0 | RCD1F | R/W | 0 | Slave I$^2$C DATA1(SIRCD1) reception completed interrupt flag<br>0: write 0 to clear it<br>1: Set by H/W when Slave I$^2$C DATA1(SIRCD1) reception complete |
| EBh | **SIRCD1** | 7~0 | SIRCD1 | R | – | Slave I$^2$C data receive register1 (DATA1) |
| ECh | **SITXRCD2** | 7~0 | SITXRCD2 | R/W | – | Slave I$^2$C transmit and receive data register<br>Read: Slave I$^2$C data receive register2 (DATA2)<br>Write: Slave I$^2$C data transmission register (TXD) |
| F0h | **B** | 7~0 | B | R/W | 00h | B register |
| F1h | **CRCDL** | 7~0 | CRCDL | R/W | FFh | 16-bit CRC data bit 7~0 |
| F2h | **CRCDH** | 7~0 | CRCDH | R/W | FFh | 16-bit CRC data bit 15~8 |

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| F3h | **CRCIN** | 7~0 | CRCIN | W | – | CRC input data |
| F5h | **CFGBG** | 3~0 | BGTRIM | R/W | – | VBG trimming value (Chip Reserved) |
| F6h | **CFGWL** | 6~0 | FRCF | R/W | – | FRC frequency adjustment, automatically load the calibration value after power-on<br>00h: lowest frequency<br>7Fh: highest frequency |
| F7h | **AUX2** | 7~6 | WDTE | R/W | – | Watchdog Timer Reset control<br>0x: WDT disable<br>10: WDT enable in Fast/Slow mode, disable in Idle/Halt/Stop mode<br>11: WDT always enable |
| | | 5 | PWRSAV | R/W | – | Power saving mode control<br>0: No power saving<br>1: Power saving, disable LVR in IDLE/HALT/STOP mode |
| | | 4 | VBGOUT | R/W | 0 | Bandgap voltage output control<br>0: P3.2 as normal I/O<br>1: Bandgap voltage output to P3.2 pin, The additional condition VBGEN=1 (AEh.1) should be set. |
| | | 3 | DIV32 | R/W | 0 | only active when MULDVI16 =1<br>0: instruction DIV as 16/16 bit division operation<br>1: instruction DIV as 32/16 bit division operation |
| | | 2~1 | IAPTE | R/W | 00 | IAP watchdog timer enable<br>00: Disable<br>01: wait 0.8mS trigger watchdog time-out flag<br>10: wait 3.2mS trigger watchdog time-out flag<br>11: wait 6.4mS trigger watchdog time-out flag |
| | | 0 | MULDIV16 | R/W | 0 | 0: instruction MUL/DIV as 8*8, 8/8 operation<br>1: instruction MUL/DIV as 16*16, 16/16 or 32/16 operation |
| F8h | **AUX1** | 7 | CLRWDT | R/W | 0 | Set 1 to clear WDT, H/W auto clear it at next clock cycle |
| | | 6 | CLRTM3 | R/W | 0 | Set 1 to clear Timer3, HW auto clear it at next clock cycle. |
| | | 5 | TKSOC | R/W | 0 | Touch Key Start of Conversion<br>Set 1 to start Touch Key conversion. If SYSCLK is fast enough, this bit will be cleared by H/W at the end of conversion. S/W can also write 0 to clear this flag. |
| | | 4 | ADSOC | R/W | 0 | ADC Start of Conversion<br>Set 1 to start ADC conversion. Cleared by H/W at the end of conversion. S/W can also write 0 to clear this flag. |
| | | 3 | LVRPD | R/W | 0 | Low Voltage Reset function select<br>0: enable LVR<br>1: disable LVR |
| | | 2 | T2SEL | R/W | 0 | Timer2 counter mode (CT2N=1) input select<br>0: P1.0 (T2) pin (8051standard)<br>1:Slow clock divide by 16 (SLOWCLK/16) |
| | | 1 | T1SEL | R/W | 0 | Timer1 counter mode (CT1N=1) input select<br>0: P3.5 (T1) pin (8051 standard)<br>1: Slow clock divide by 16 (SLOWCLK/16) |
| | | 0 | DPSEL | R/W | 0 | Active DPTR Select |

| Adr | Flash | Bit# | Bit Name | Description |
|---|---|---|---|---|
| 7FFFh | **CFGWH** | 7 | PROTN | Flash Code Protect, 0=Protect |
| | | 6 | XRSTEN | External Pin Reset enable, 0=enable. |
| | | 5~3 | LVRE | Low Voltage Reset function select<br>000: Set LVR at 2.3V<br>001: Set LVR at 2.54V<br>010: Set LVR at 2.78V<br>011: Set LVR at 3.04V<br>100: Set LVR at 3.28V<br>101: Set LVR at 3.54V<br>110: Set LVR at 3.8V<br>111: Set LVR at 4.04V |
| | | 1 | MVCLOCKN | If 0, the MOVC & MOVX instruction’s accessibility to MOVC-Lock area is limited. |
| | | 0 | FRCPSC | FRC frequency select<br>0: 9.216MHz<br>1: 18.432MHz |

# INSTRUCTION SET

Instructions are 1, 2 or 3 bytes long as listed in the 'byte' column below. Each instruction takes 1~8 System clock cycles to execute as listed in the 'cycle' column below.

| ARITHMETIC | | | | |
|---|---|---|---|---|
| **Mnemonic** | **Description** | **byte** | **cycle** | **opcode** |
| ADD A,Rn | Add register to A | 1 | 2 | 28-2F |
| ADD A,dir | Add direct byte to A | 2 | 2 | 25 |
| ADD A,@Ri | Add indirect memory to A | 1 | 2 | 26-27 |
| ADD A,#data | Add immediate to A | 2 | 2 | 24 |
| ADDC A,Rn | Add register to A with carry | 1 | 2 | 38-3F |
| ADDC A,dir | Add direct byte to A with carry | 2 | 2 | 35 |
| ADDC A,@Ri | Add indirect memory to A with carry | 1 | 2 | 36-37 |
| ADDC A,#data | Add immediate to A with carry | 2 | 2 | 34 |
| SUBB A,Rn | Subtract register from A with borrow | 1 | 2 | 98-9F |
| SUBB A,dir | Subtract direct byte from A with borrow | 2 | 2 | 95 |
| SUBB A,@Ri | Subtract indirect memory from A with borrow | 1 | 2 | 96-97 |
| SUBB A,#data | Subtract immediate from A with borrow | 2 | 2 | 94 |
| INC A | Increment A | 1 | 2 | 04 |
| INC Rn | Increment register | 1 | 2 | 08-0F |
| INC dir | Increment direct byte | 2 | 2 | 05 |
| INC @Ri | Increment indirect memory | 1 | 2 | 06-07 |
| DEC A | Decrement A | 1 | 2 | 14 |
| DEC Rn | Decrement register | 1 | 2 | 18-1F |
| DEC dir | Decrement direct byte | 2 | 2 | 15 |
| DEC @Ri | Decrement indirect memory | 1 | 2 | 16-17 |
| INC DPTR | Increment data pointer | 1 | 4 | A3 |
| MUL AB | Multiply A by B | 1 | 8 / 16 | A4 |
| DIV AB | Divide A by B | 1 | 8/16/32 | 84 |
| DA A | Decimal Adjust A | 1 | 2 | D4 |

| LOGICAL | | | | |
|---|---|---|---|---|
| **Mnemonic** | **Description** | **byte** | **cycle** | **opcode** |
| ANL A,Rn | AND register to A | 1 | 2 | 58-5F |
| ANL A,dir | AND direct byte to A | 2 | 2 | 55 |
| ANL A,@Ri | AND indirect memory to A | 1 | 2 | 56-57 |
| ANL A,#data | AND immediate to A | 2 | 2 | 54 |
| ANL dir,A | AND A to direct byte | 2 | 2 | 52 |
| ANL dir,#data | AND immediate to direct byte | 3 | 4 | 53 |
| ORL A,Rn | OR register to A | 1 | 2 | 48-4F |
| ORL A,dir | OR direct byte to A | 2 | 2 | 45 |
| ORL A,@Ri | OR indirect memory to A | 1 | 2 | 46-47 |
| ORL A,#data | OR immediate to A | 2 | 2 | 44 |
| ORL dir,A | OR A to direct byte | 2 | 2 | 42 |
| ORL dir,#data | OR immediate to direct byte | 3 | 4 | 43 |
| XRL A,Rn | Exclusive-OR register to A | 1 | 2 | 68-6F |
| XRL A,dir | Exclusive-OR direct byte to A | 2 | 2 | 65 |
| XRL A, @Ri | Exclusive-OR indirect memory to A | 1 | 2 | 66-67 |
| XRL A,#data | Exclusive-OR immediate to A | 2 | 2 | 64 |
| XRL dir,A | Exclusive-OR A to direct byte | 2 | 2 | 62 |
| XRL dir,#data | Exclusive-OR immediate to direct byte | 3 | 4 | 63 |
| CLR A | Clear A | 1 | 2 | E4 |
| CPL A | Complement A | 1 | 2 | F4 |
| SWAP A | Swap Nibbles of A | 1 | 2 | C4 |

| LOGICAL | | | | |
|---|---|---|---|---|
| **Mnemonic** | **Description** | **byte** | **cycle** | **opcode** |
| RL A | Rotate A left | 1 | 2 | 23 |
| RLC A | Rotate A left through carry | 1 | 2 | 33 |
| RR A | Rotate A right | 1 | 2 | 03 |
| RRC A | Rotate A right through carry | 1 | 2 | 13 |

| DATA TRANSFER | | | | |
|---|---|---|---|---|
| **Mnemonic** | **Description** | **byte** | **cycle** | **opcode** |
| MOV A,Rn | Move register to A | 1 | 2 | E8-EF |
| MOV A,dir | Move direct byte to A | 2 | 2 | E5 |
| MOV A,@Ri | Move indirect memory to A | 1 | 2 | E6-E7 |
| MOV A,#data | Move immediate to A | 2 | 2 | 74 |
| MOV Rn,A | Move A to register | 1 | 2 | F8-FF |
| MOV Rn,dir | Move direct byte to register | 2 | 4 | A8-AF |
| MOV Rn,#data | Move immediate to register | 2 | 2 | 78-7F |
| MOV dir,A | Move A to direct byte | 2 | 2 | F5 |
| MOV dir,Rn | Move register to direct byte | 2 | 4 | 88-8F |
| MOV dir,dir | Move direct byte to direct byte | 3 | 4 | 85 |
| MOV dir,@Ri | Move indirect memory to direct byte | 2 | 4 | 86-87 |
| MOV dir,#data | Move immediate to direct byte | 3 | 4 | 75 |
| MOV @Ri,A | Move A to indirect memory | 1 | 2 | F6-F7 |
| MOV @Ri,dir | Move direct byte to indirect memory | 2 | 4 | A6-A7 |
| MOV @Ri,#data | Move immediate to indirect memory | 2 | 2 | 76-77 |
| MOV DPTR,#data | Move immediate to data pointer | 3 | 4 | 90 |
| MOVC A,@A+DPTR | Move code byte relative DPTR to A | 1 | 4 | 93 |
| MOVC A,@A+PC | Move code byte relative PC to A | 1 | 4 | 83 |
| MOVX A,@Ri | Move external data(A8) to A | 1 | 4 | E2-E3 |
| MOVX A,@DPTR | Move external data(A16) to A | 1 | 4 | E0 |
| MOVX @Ri,A | Move A to external data(A8) | 1 | 4 | F2-F3 |
| MOVX @DPTR,A | Move A to external data(A16) | 1 | 4 | F0 |
| PUSH dir | Push direct byte onto stack | 2 | 4 | C0 |
| POP dir | Pop direct byte from stack | 2 | 4 | D0 |
| XCH A,Rn | Exchange A and register | 1 | 2 | C8-CF |
| XCH A,dir | Exchange A and direct byte | 2 | 2 | C5 |
| XCH A,@Ri | Exchange A and indirect memory | 1 | 2 | C6-C7 |
| XCHD A,@Ri | Exchange A and indirect memory nibble | 1 | 2 | D6-D7 |

| BOOLEAN | | | | |
|---|---|---|---|---|
| **Mnemonic** | **Description** | **byte** | **cycle** | **opcode** |
| CLR C | Clear carry | 1 | 2 | C3 |
| CLR bit | Clear direct bit | 2 | 2 | C2 |
| SETB C | Set carry | 1 | 2 | D3 |
| SETB bit | Set direct bit | 2 | 2 | D2 |
| CPL C | Complement carry | 1 | 2 | B3 |
| CPL bit | Complement direct bit | 2 | 2 | B2 |
| ANL C,bit | AND direct bit to carry | 2 | 4 | 82 |
| ANL C,/bit | AND direct bit inverse to carry | 2 | 4 | B0 |
| ORL C,bit | OR direct bit to carry | 2 | 4 | 72 |
| ORL C,/bit | OR direct bit inverse to carry | 2 | 4 | A0 |
| MOV C,bit | Move direct bit to carry | 2 | 2 | A2 |
| MOV bit,C | Move carry to direct bit | 2 | 4 | 92 |

| BRANCHING | | | | |
|---|---|---|---|---|
| **Mnemonic** | **Description** | **byte** | **cycle** | **opcode** |
| ACALL addr 11 | Absolute jump to subroutine | 2 | 4 | 11-F1 |
| LCALL addr 16 | Long jump to subroutine | 3 | 4 | 12 |
| RET | Return from subroutine | 1 | 4 | 22 |
| RETI | Return from interrupt | 1 | 4 | 32 |
| AJMP addr 11 | Absolute jump unconditional | 2 | 4 | 01-E1 |
| LJMP addr 16 | Long jump unconditional | 3 | 4 | 02 |
| SJMP rel | Short jump (relative address) | 2 | 4 | 80 |
| JC rel | Jump on carry = 1 | 2 | 4 | 40 |
| JNC rel | Jump on carry = 0 | 2 | 4 | 50 |
| JB bit,rel | Jump on direct bit = 1 | 3 | 4 | 20 |
| JNB bit,rel | Jump on direct bit = 0 | 3 | 4 | 30 |
| JBC bit,rel | Jump on direct bit = 1 and clear | 3 | 4 | 10 |
| JMP @A+DPTR | Jump indirect relative DPTR | 1 | 4 | 73 |
| JZ rel | Jump on accumulator = 0 | 2 | 4 | 60 |
| JNZ rel | Jump on accumulator ≠ 0 | 2 | 4 | 70 |
| CJNE A,dir,rel | Compare A,direct, jump not equal relative | 3 | 4 | B5 |
| CJNE A,#data,rel | Compare A,immediate, jump not equal relative | 3 | 4 | B4 |
| CJNE Rn,#data,rel | Compare register,immediate, jump not equal relative | 3 | 4 | B8-BF |
| CJNE @Ri,#data,rel | Compare indirect,immediate, jump not equal relative | 3 | 4 | B6-B7 |
| DJNZ Rn,rel | Decrement register, jump not zero relative | 2 | 4 | D8-DF |
| DJNZ dir,rel | Decrement direct byte, jump not zero relative | 3 | 4 | D5 |

| MISCELLANEOUS | | | | |
|---|---|---|---|---|
| **Mnemonic** | **Description** | **byte** | **cycle** | **opcode** |
| NOP | No operation | 1 | 2 | 00 |

In the above table, an entry such as E8-EF indicates a continuous block of hex opcodes used for 8 different registers, the register numbers of which are defined by the lowest three bits of the corresponding code. Non-continuous blocks of codes, shown as 11-F1 (for example), are used for absolute jumps and calls with the top 3 bits of the code being used to store the top three bits of the destination address.

# ELECTRICAL CHARACTERISTICS

## 1. Absolute Maximum Ratings ($T_A$=25℃)

| Parameter | Rating | Unit |
|---|---|---|
| Supply voltage | $V_{SS}$ –0.3 ~ $V_{SS}$ +5.5 | V |
| Input voltage | $V_{SS}$ –0.3 ~ $V_{CC}$ +0.3 | |
| Output voltage | $V_{SS}$ –0.3 ~ $V_{CC}$ +0.3 | |
| All pins output current high | –80 | mA |
| All pins output current low | +150 | |
| Maximum Operating Voltage | 5.5 | V |
| Operating temperature | –40 ~ +85 | °C |
| Storage temperature | –65 ~ +150 | |

## 2. DC Characteristics ($T_A$=25 ℃, $V_{CC}$=2.2V ~ 5.5V)

| Parameter | Symbol | Conditions | | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|---|
| Operating Voltage | $V_{CC}$ | $F_{SYSCLK}$=18.432 MHz | | 2.2 | – | 5.5 | V |
| Input High Voltage | $V_{IH}$ | All Input | $V_{CC}$=5V | 0.6$V_{CC}$ | – | – | V |
| | | | $V_{CC}$=3V | 0.6$V_{CC}$ | – | – | V |
| Input Low Voltage | $V_{IL}$ | All Input | $V_{CC}$=5V | – | – | 0.2$V_{CC}$ | V |
| | | | $V_{CC}$=3V | – | – | 0.2$V_{CC}$ | V |
| I/O Port Source Current | $I_{OH}$ | All Output | $V_{CC}$=5V, $V_{OH}$=0.9$V_{CC}$ | 5.5 | 11 | – | mA |
| | | | $V_{CC}$=5V, $V_{OH}$=0.6$V_{CC}$ | 15 | 30 | | |
| | | | $V_{CC}$=3V, $V_{OH}$=0.9$V_{CC}$ | 2.5 | 4.5 | – | |
| | | | $V_{CC}$=3V, $V_{OH}$=0.6$V_{CC}$ | 7 | 13 | | |
| I/O Port Sink Current | $I_{OL}$ | All Output, | $V_{CC}$=5V, $V_{OL}$=0.1$V_{CC}$ | 40 | 65 | – | mA |
| | | | $V_{CC}$=3V, $V_{OL}$=0.1$V_{CC}$ | 20 | 30 | – | |
| Supply Current | $I_{DD}$ | FAST mode | FRC=18.432 MHz<br>$V_{CC}$=5V | – | 4 | – | mA |
| | | | FRC=18.432 MHz<br>$V_{CC}$=3V | – | 3.5 | – | |
| | | SLOW mode | $V_{CC}$=3V | – | 0.22 | – | |
| | | | $V_{CC}$=5V | – | 0.2 | – | |
| | | IDLE mode PWRSAV=0 | SRC, $V_{CC}$=5V | – | 200 | – | μA |
| | | | SRC, $V_{CC}$=3V | – | 183 | – | |
| | | IDLE mode PWRSAV=1 | $V_{CC}$=5V | – | 183 | – | |
| | | | $V_{CC}$=3V | – | 166 | – | |
| | | STOP mode PWRSAV=0 | $V_{CC}$=5V | – | 65 | – | |
| | | | $V_{CC}$=3V | – | 55 | – | |
| | | STOP mode PWRSAV=1 | $V_{CC}$=5V | – | 10 | – | |
| | | | $V_{CC}$=3V | – | 4 | – | |
| | | HALT mode PWRSAV=0 | $V_{CC}$=5V | – | 68 | – | |
| | | | $V_{CC}$=3V | – | 57 | – | |
| | | HALT mode PWRSAV=1 | $V_{CC}$=5V | – | 13 | – | |
| | | | $V_{CC}$=3V | – | 6 | – | |
| System Clock Frequency | $F_{SYSCLK}$ | $V_{CC}$ >$LVR_{TH}$ | $V_{CC}$=2.2V | – | – | 18.432 | MHz |

| Parameter | Symbol | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| LVR Reference Voltage | $V_{LVR}$ | $T_A$=25°C | – | 2.3 | – | V |
| | | | – | 2.54 | – | |
| | | | – | 2.78 | – | |
| | | | – | 3.04 | – | |
| | | | – | 3.28 | – | |
| | | | – | 3.54 | – | |
| | | | – | 3.8 | – | |
| | | | – | 4.04 | – | |
| LVR Hysteresis Voltage | $V_{HYST}$ | $T_A$=25°C | – | ±0.1 | – | V |
| LVD Reference Voltage | $V_{LVD}$ | $T_A$=25°C | – | 2.3 | – | V |
| | | | – | 2.54 | – | |
| | | | – | 2.78 | – | |
| | | | – | 3.04 | – | |
| | | | – | 3.28 | – | |
| | | | – | 3.54 | – | |
| | | | – | 3.8 | – | |
| | | | – | 4.04 | – | |
| Low Voltage Detection time | $t_{LVR}$ | $T_A$=25°C | 100 | – | – | μs |
| Pull-Up Resistor | $R_P$ | $V_{IN}$=0V, $V_{CC}$=5V | – | 35 | – | KΩ |
| | | $V_{CC}$=3V | | 60 | | |

## 3. Clock Timing ($T_A$= –40°C ~ +85°C)

| Parameter | Condition | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| FRC Frequency | -40℃ ~ 105℃, $V_{CC}$=5.0V | –1% | 18.432 | +1% | MHz |
| | -20℃ ~ 85℃, $V_{CC}$=3.0 ~ 5.0V | –1% | 18.432 | +3% | |

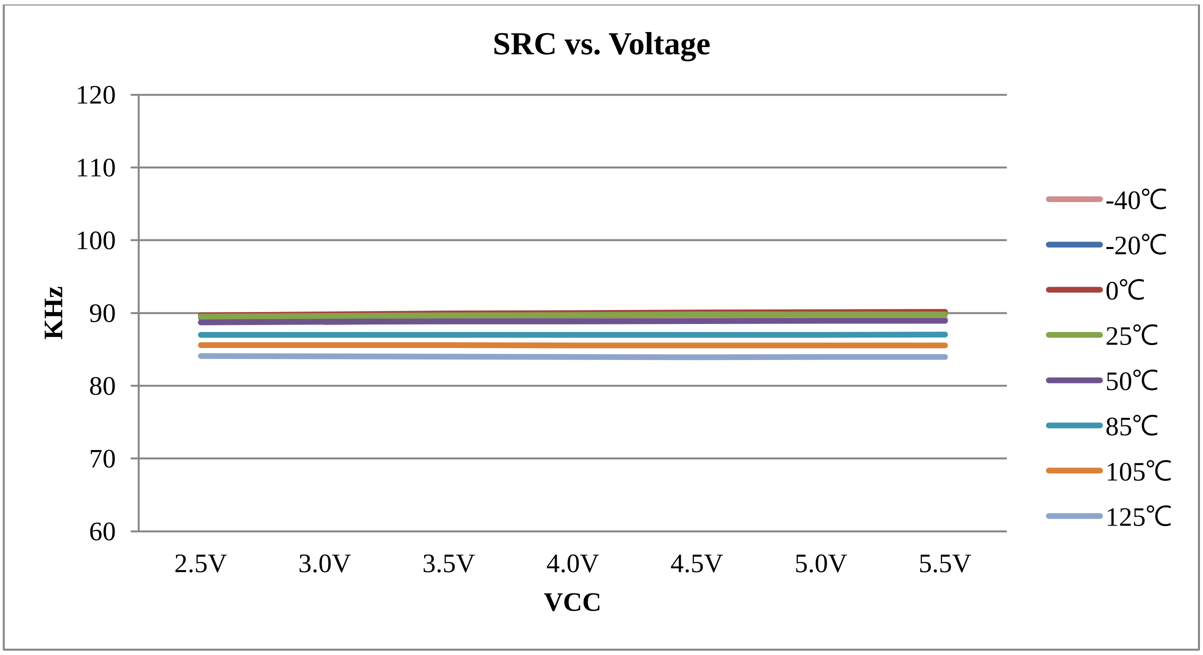
## 4. Reset Timing Characteristics ($T_A$= –40°C ~ +85°C)

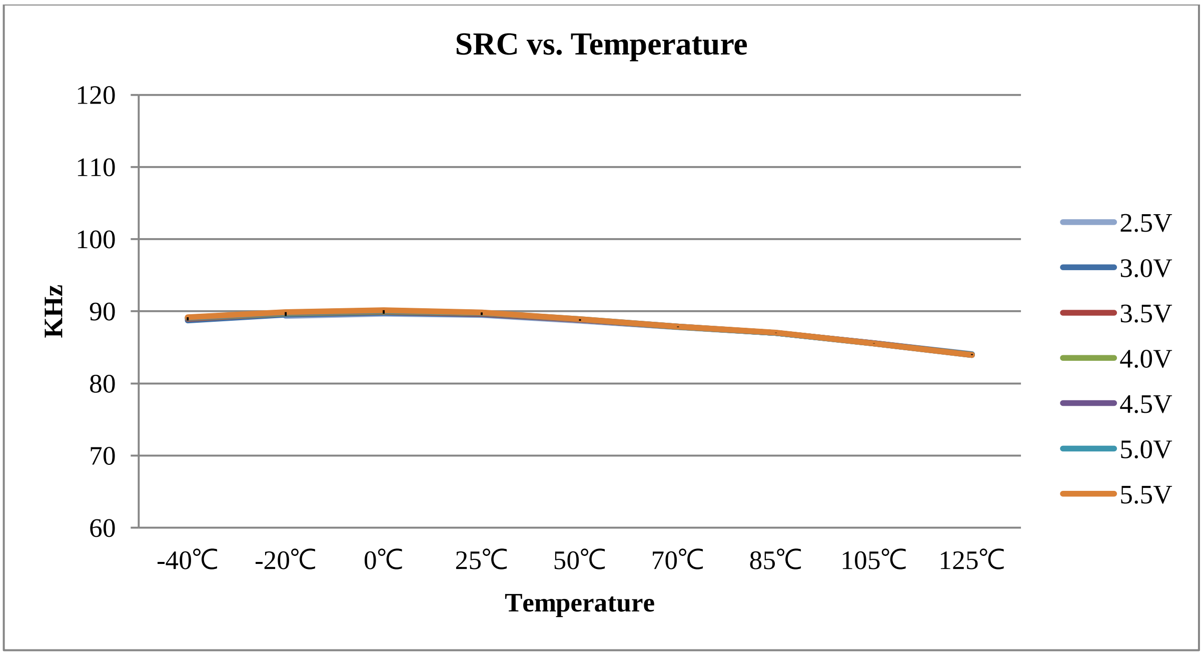
| Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| RESET Input Low width | Input $V_{CC}$=5V ± 10 % | 30 | – | – | µs |
| WDT wakeup time | $V_{CC}$=5V, WDTPSC=11 | – | 55 | – | ms |
| | $V_{CC}$=3V, WDTPSC=11 | – | 57 | – | |

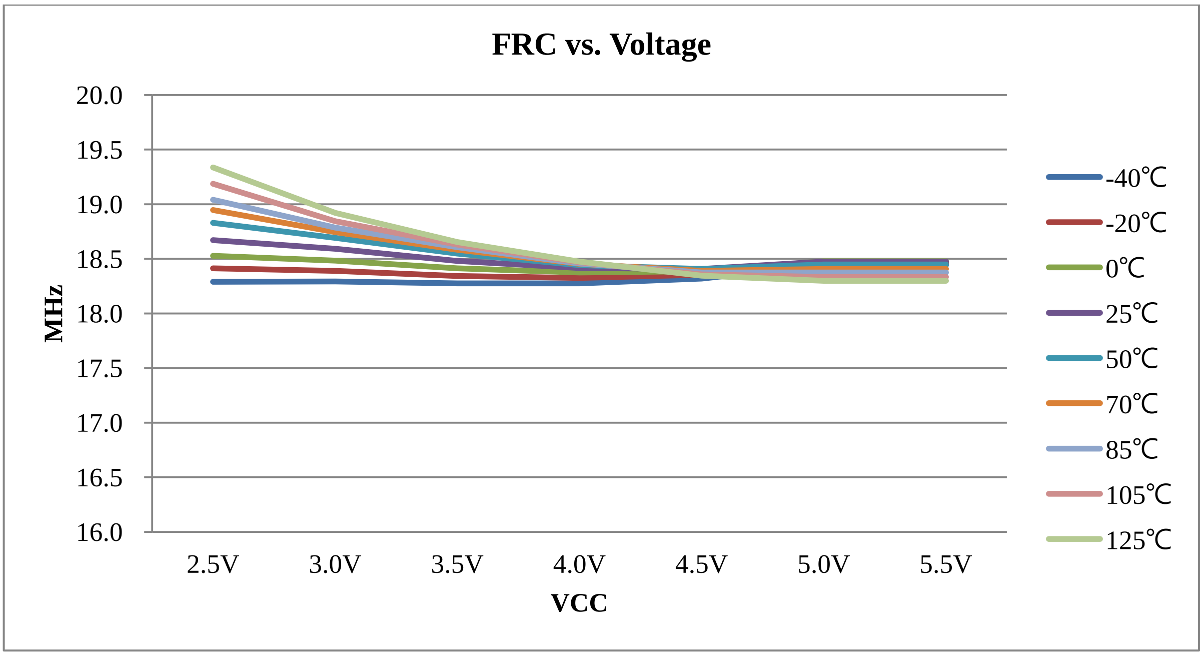
## 5. ADC Electrical Characteristics ($T_A$= 25°C, $V_{CC}$= 3.0V ~ 5.5V, $V_{SS}$= 0V)

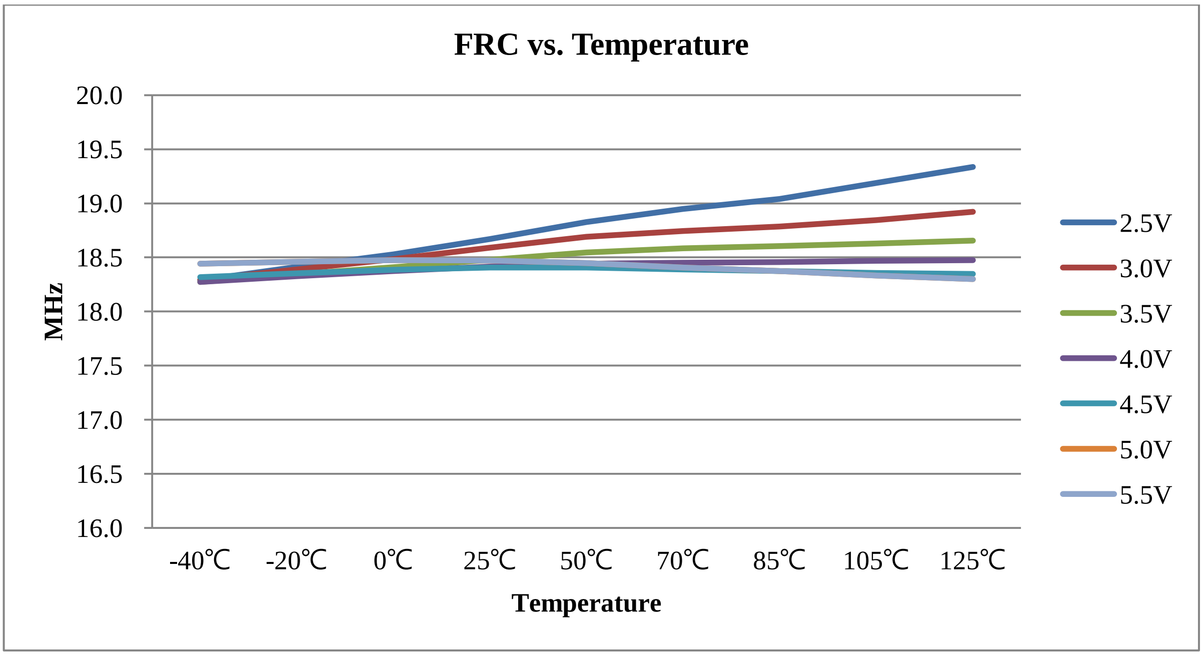
| Parameter | Conditions | | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| Total Accuracy | $V_{CC}$=5.12 V, $V_{SS}$=0V | | – | ±2.5 | ±4 | LSB |
| Integral Non-Linearity | | | – | ±3.2 | ±5 | |
| Max Input Clock ($f_{ADC}$) | Source impedance (Rs < 10K omh) | | – | – | 2 | MHz |
| | Source impedance (Rs < 20K omh) | | – | – | 1 | |
| | Source impedance (Rs < 50K omh) | | – | – | 0.5 | |
| | Source is VBG (ADCHS=1011b) | | – | – | 0.5 | |
| Conversion Time | $F_{ADC}$ = 1MHz | | – | 50 | – | µs |
| Bandgap Reference Voltage ($V_{BG}$) | $V_{CC}$=3V~5V<br>-40°C ~85°C | | -2% | 1.27 | +2% | V |
| ADC Reference Voltage ($V_{ADC}$) | ADCVREFS=1 | $V_{CC}$=5V<br>0°C ~85°C | -1.5% | 2.47 | +1.5% | |
| $V_{CC}$/4 Reference Voltage ($V_{1/4}$) | – | $V_{CC}$=5V, 25°C | -0.8% | 1.26 | +0.8% | |
| | | $V_{CC}$=3.6V, 25°C | -0.8% | 0.907 | +0.8% | |
| Input Voltage | – | | $V_{SS}$ | – | $V_{CC}$ | |

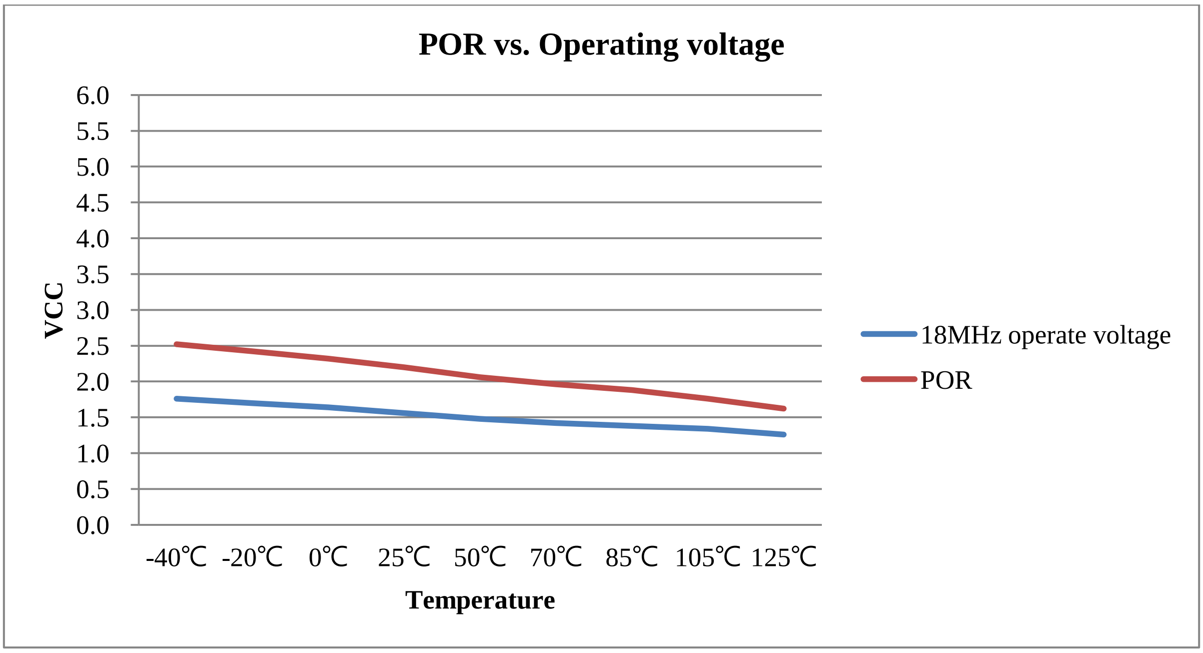
## 6. Characteristic Graphs

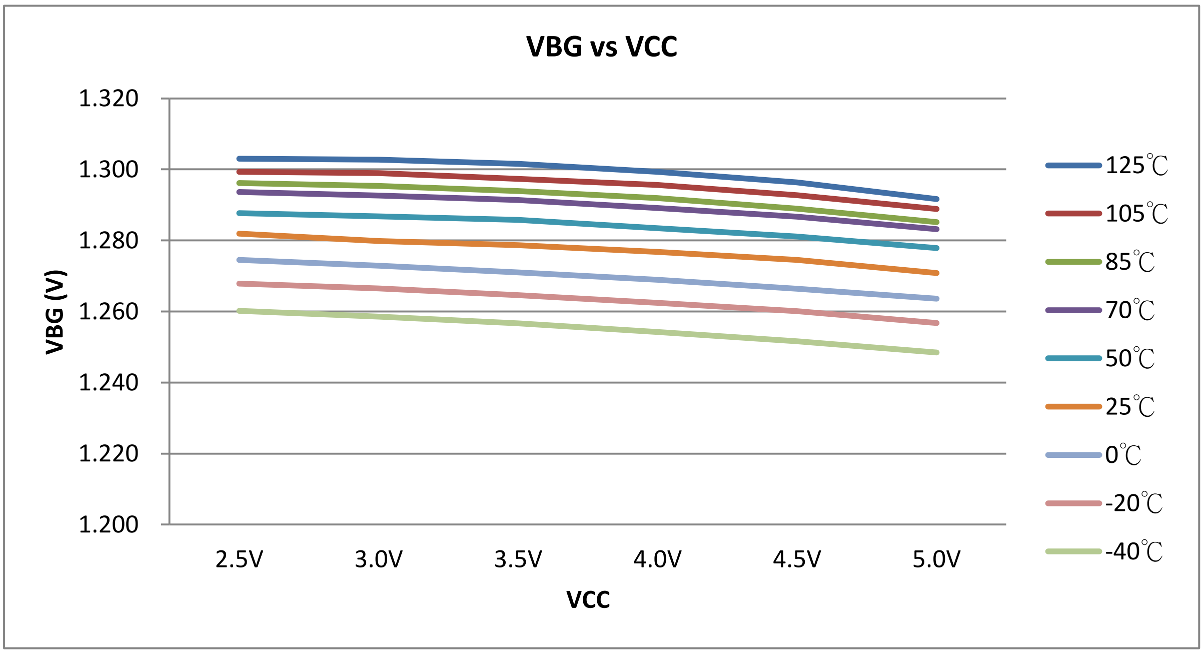





FRC vs. Voltage
20.0
19.5
19.0
18.5
18.0
17.5
17.0
16.5
16.0
MHz
2.5V
3.0V
3.5V
4.0V
4.5V
5.0V
5.5V
VCC
-40℃
-20℃
0℃
25℃
50℃
70℃
85℃
105℃
125℃


FRC vs. Temperature
20.0
19.5
19.0
18.5
18.0
17.5
17.0
16.5
16.0
MHz
-40℃
-20℃
0℃
25℃
50℃
70℃
85℃
105℃
125℃
Temperature
2.5V
3.0V
3.5V
4.0V
4.5V
5.0V
5.5V


POR vs. Operating voltage
VCC
6.0
5.5
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
-40℃
-20℃
0℃
25℃
50℃
70℃
85℃
105℃
125℃
Temperature
18MHz operate voltage
POR


VBG vs VCC
VBG (V)
1.320
1.300
1.280
1.260
1.240
1.220
1.200
2.5V
3.0V
3.5V
4.0V
4.5V
5.0V
VCC
125℃
105℃
85℃
70℃
50℃
25℃
0℃
-20℃
-40℃

# Package and Dice Information

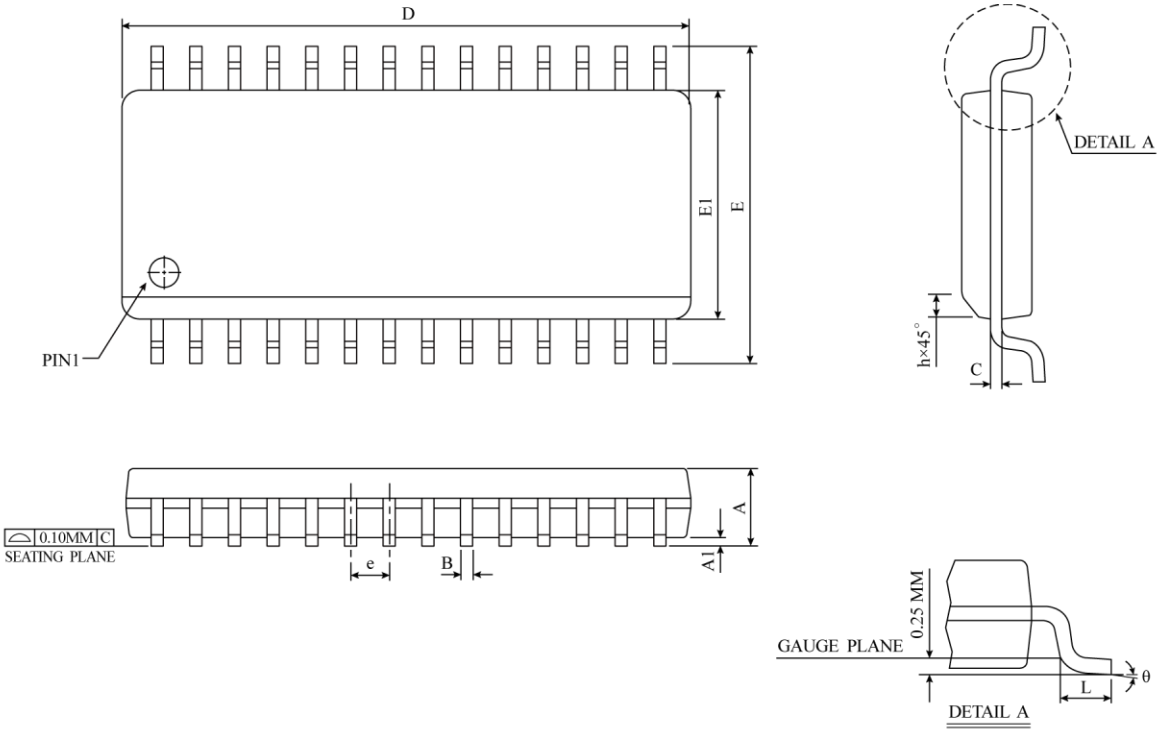
Please note that the package information provided is for reference only. Since this information is frequently updated, users can contact Sales to consult the latest package information and stocks.

## Ordering information

| Ordering number | Package |
|---|---|
| TM52eF1375G-MTP-23 | SOP 28-pin (300 mil) |
| TM52eF1374G-MTP-23 | |
| TM52eF1375G-MTP-21 | SOP 20 pin (300mil) |
| TM52eF1374G-MTP-21 | |

## Package Information

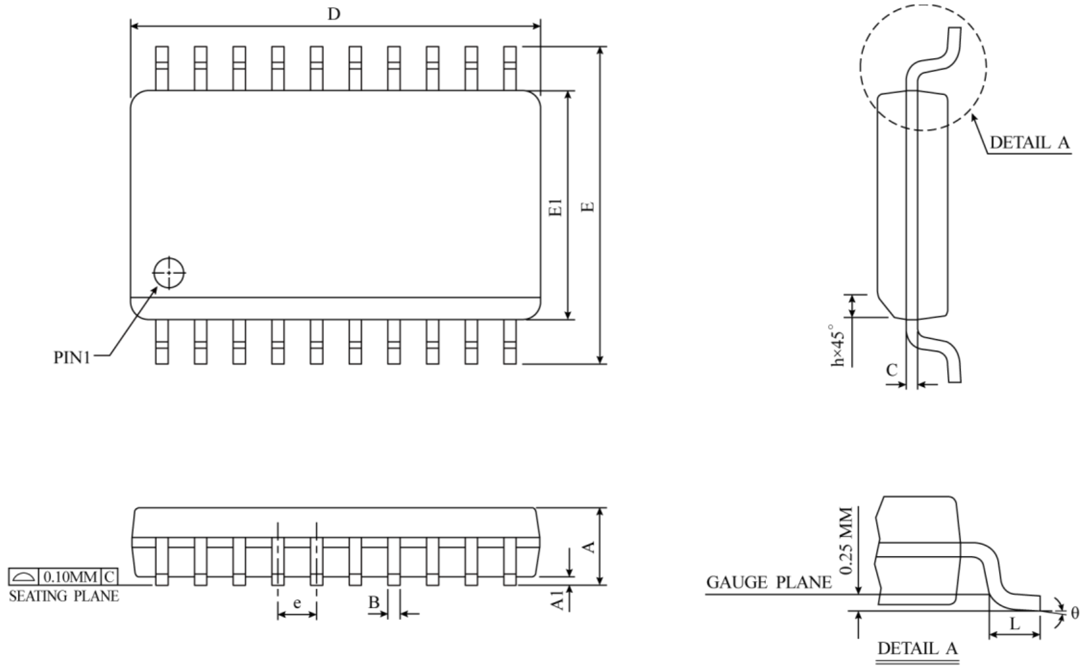
### SOP-28 ( 300mil ) Package Dimension



| SYMBOL | DIMENSION IN MM | | | DIMENSION IN INCH | | |
|---|---|---|---|---|---|---|
| | MIN | NOM | MAX | MIN | NOM | MAX |
| A | 2.35 | 2.50 | 2.65 | 0.0926 | 0.0985 | 0.1043 |
| A1 | 0.10 | 0.20 | 0.30 | 0.0040 | 0.0079 | 0.0118 |
| B | 0.33 | 0.42 | 0.51 | 0.0130 | 0.0165 | 0.0200 |
| C | 0.23 | 0.28 | 0.32 | 0.0091 | 0.0108 | 0.0125 |
| D | 17.70 | 17.90 | 18.10 | 0.6969 | 0.7047 | 0.7125 |
| E | 10.00 | 10.33 | 10.65 | 0.3940 | 0.4425 | 0.4910 |
| E1 | 7.40 | 7.50 | 7.60 | 0.2914 | 0.2953 | 0.2992 |
| e | 1.27 BSC | | | 0.050 BSC | | |
| h | 0.25 | 0.50 | 0.75 | 0.0100 | 0.0195 | 0.0290 |
| L | 0.40 | 0.84 | 1.27 | 0.0160 | 0.0330 | 0.0500 |
| θ | 0° | 4° | 8° | 0° | 4° | 8° |
| JEDEC | MS-013 (AE) | | | | | |

⚠ * NOTES : DIMENSION " D " DOES NOT INCLUDE MOLD FLASH, PROTRUSIONS OR GATE BURRS.
MOLD FLASH, PROTRUSIONS AND GATE BURRS SHALL
NOT EXCEED 0.15 MM ( 0.006 INCH ) PER SIDE.

## SOP-20 ( 300mil ) Package Dimension



| SYMBOL | DIMENSION IN MM | | | DIMENSION IN INCH | | |
|---|---|---|---|---|---|---|
| | MIN | NOM | MAX | MIN | NOM | MAX |
| A | 2.35 | 2.50 | 2.65 | 0.0926 | 0.0985 | 0.1043 |
| A1 | 0.10 | 0.20 | 0.30 | 0.0040 | 0.0079 | 0.0118 |
| B | 0.33 | 0.42 | 0.51 | 0.0130 | 0.0165 | 0.0200 |
| C | 0.23 | 0.28 | 0.32 | 0.0091 | 0.0108 | 0.0125 |
| D | 12.60 | 12.80 | 13.00 | 0.4961 | 0.5040 | 0.5118 |
| E | 10.00 | 10.33 | 10.65 | 0.3940 | 0.4425 | 0.4910 |
| E1 | 7.40 | 7.50 | 7.60 | 0.2914 | 0.2953 | 0.2992 |
| e | 1.27 BSC | | | 0.050 BSC | | |
| h | 0.25 | 0.50 | 0.75 | 0.0100 | 0.0195 | 0.0290 |
| L | 0.40 | 0.84 | 1.27 | 0.0160 | 0.0330 | 0.0500 |
| θ | 0° | 4° | 8° | 0° | 4° | 8° |
| JEDEC | MS-013 (AC) | | | | | |

△ * NOTES : DIMENSION 〝D〞 DOES NOT INCLUDE MOLD FLASH, PROTRUSIONS OR GATE BURRS.
MOLD FLASH, PROTRUSIONS AND GATE BURRS SHALL
NOT EXCEED 0.15 MM ( 0.006 INCH ) PER SIDE.