十速

# TM52F1364

## *DATA SHEET Rev 0.91*

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

# PRECAUTIONS

1. The chip cannot enter Halt/Stop mode if the INTn pin is low and the INTn wake-up function is enabled. (INTn=0 and EXn=1, n=0~2)
2. If you need to use LVR, it is recommended to set the LVR (SFR LVRSEL) first after the program is powered on, and then change the default value related to the pin.

# AMENDMENT HISTORY

| Version | Date | Description |
|---|---|---|
| V0.90 | Apr, 2023 | |
| V0.91 | Dec, 2023 | Flash IAP needs to be written twice @VCC = 5.0V~5.5V<br>Added EEPROM writing condition: SFR.IVCPD=1 |

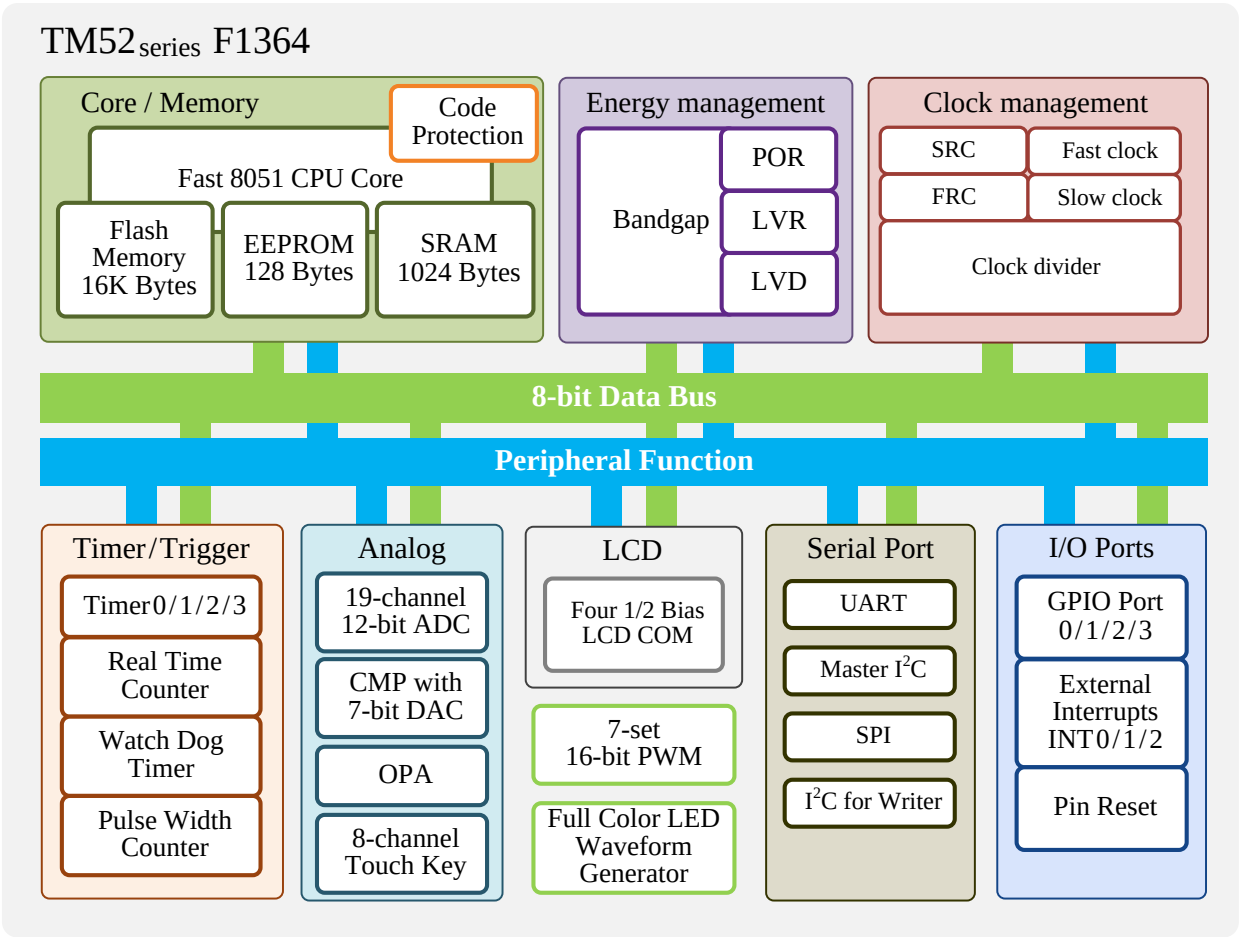
# CONTENTS

# GENERAL DESCRIPTION

**TM52 series F1364** 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-F1364** provides improved performance, lower cost and fast time-to-market by integrating features on the chip, including 16K Bytes Flash program memory, 128 Bytes EEPROM, 1024 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, 7 sets 16-bit PWMs, 19 channels 12-bit A/D Convertor, 8-channel Touch Key, master $I^2C$ interface, SPI interface, Full color LED communication format waveform generator, S/W control 1/2 bias LCD COM, OPA, CMP with DAC and Watch Dog Timer. It's a high reliability and low power consumption feature can be widely applied in consumer and home appliance products.

# SYSTEM BLOCK DIAGRAM



# FEATURES

1. **Standard 8051 Instruction set, fast machine cycle**
   - Executes instructions six times faster than the standard 8051.
2. **Flash Program Memory**
   - 16K Bytes Flash program memory
   - Support “In Circuit Programming” (ICP) or “In System Programming” (ISP) for the Flash code
   - Code Protection Capability
   - BOOT vector option
   - 10K erase times at least
   - 10 years data retention at least
3. **128 Bytes EEPROM Memory**
   - 50K erase times at least
   - 10 years data retention at least
4. **Total 1024 Bytes SRAM (IRAM + XRAM)**
   - 256 Bytes IRAM in the 8051 internal data memory area
   - 768 Bytes XRAM in the 8051 external data memory area (accessed by MOVX Instruction)
5. **Two System Clock type selections**
   - Fast clock from Internal RC (FRC, 16.588 MHz)
   - Slow clock from Internal RC (SRC, 41 KHz)
   - System Clock can be divided by 1/2/4/16 option
6. **8051 Standard Timer – Timer0/1/2**
   - 16-bit Timer0, also supports T0O clock output for Buzzer application
   - 16-bit Timer1, also supports T1O clock output for Buzzer application
   - 16-bit Timer2, also supports T2O clock output for Buzzer application
7. **15-bit Timer3**
   - Clock source is Slow clock
   - Interrupt period can be clock divided by 32768/4096/2048/512/25600/3200/1600/400 option
8. **One UART**
   - 8051 standard UART, One Wire UART option can be used for ISP or other application
   - Additional baud rate generator option

*Support one UART, pin select to P30/P31 or P02/P16 by TXRXSEL (SFR 93h.7)

9. **Seven 16-bit PWMs with prescaler/ period-adjustment**
10. **One Master $I^2C$ interface (MIIC)**

*Support one MIIC, pin select to P35/P16 by MSDASEL (SFR B7h.7) , pin select to P13/P02 by MSCLSEL (SFR B7h.6)

11. **One SPI interface**
12. **Full Color LED Communication Format Waveform Generator**
13. **12-bit ADC with 19 channels External Pin Input and 3 channels Internal Reference Voltage**
    - Internal Reference Voltage VBG, OPO, $1/4V_{CC}$
14. **8-channel Touch Key**
15. **Operational Amplifier**
    - Build-in 4-level OPA gain 1/20/50/100
16. **Comparator**
    - With 7-bit DAC output for comparator negative input
    - DAC reference voltage can select VCC or VBG (1.20V/2.49V)
    - DAC can output to PAD
17. **LDOC 1.2V High Driver Current Output (60mA)**
18. **4-level LDO Regulator for Internal Digital Circuit**
    - 1.70V/1.95V/2.20V/2.45V
19. **LCD Driver**
    - Software controlled COM0~3
    - 1/2 LCD Bias
20. **14 Sources, 4-level priority Interrupt**
    - Timer0/Timer1/Timer2/Timer3 Interrupt
    - INT0/INT1 pin Falling-Edge/Low-Level Interrupt
    - INT2 pin Falling-Edge Interrupt
    - Port0/1/2/3 Pin Change Interrupt
    - UART TX/RX Interrupt
    - ADC/TK Interrupt
    - MIIC/SPI Interrupt
    - LVD Interrupt
    - CMP Interrupt
    - PWM0/PWM1 Interrupt
21. **Pin Interrupt can Wake up CPU from Power-Down (Halt/Stop) mode**
    - INT0~INT2 Interrupt & Wake-up
    - Each Port0/1/2/3 pin can be defined as Interrupt & Wake-up pin (by pin change)
22. **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

**23. Independent RC Oscillating Watch Dog Timer**

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

**24. Five types Reset**

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

**25. 16-level Low Voltage Reset**

- 2.05V / 2.19V / 2.33V / 2.47V / 2.61V / 2.75V / 2.89V / 3.03V / 3.17V / 3.31V / 3.45V / 3.59V / 3.73V / 3.87V / 4.01V / 4.15V

**26. 15-level Low Voltage Detect**

- 2.19V / 2.33V / 2.47V / 2.61V / 2.75V / 2.89V / 3.03V / 3.17V / 3.31V / 3.45V / 3.59V / 3.73V / 3.87V / 4.01V / 4.15V
- LVD detect polarity option
- LVD Hysteresis 30mV~80mV

**27. Five Power Operation Modes**

- Fast/Slow/Idle/Halt/Stop mode

**28. Integrated 16-bit Cyclic Redundancy Check function**

**29. Multiplication and division**

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

**30. On-chip Debug/ICE interface**

- Use P3.0/P3.1 pin or P2.0/P2.1 pin
- Share with ICP programming pin
- Mass production writer only supports P3.0/P3.1

**31. Operating Voltage and Current**

- $V_{CC}$ =2.2V ~ 5.5V @$F_{SYS}$=16.588 MHz
- $I_{CC}$ =0.2μA @Stop mode, PWRSAV=1, $V_{CC}$=3V
- $I_{CC}$ =2.4μA @Halt mode, PWRSAV=1, $V_{CC}$=3V
- $I_{CC}$ =4.1μA @Idle mode, PWRSAV=1, PORPD=1, $V_{CC}$=3V

## 32. Operating Temperature Range

- –40℃ ~ +105℃

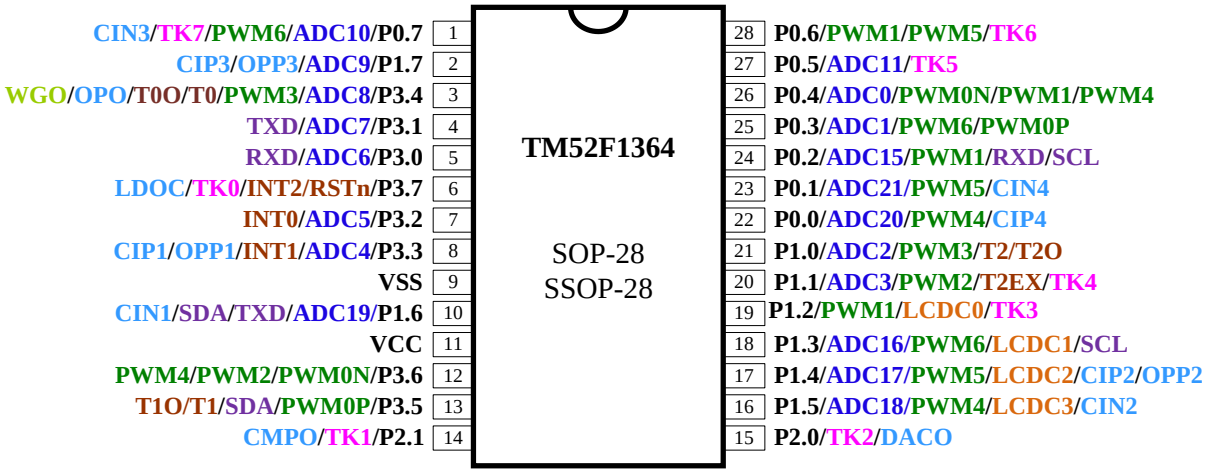
## 33. Package Types

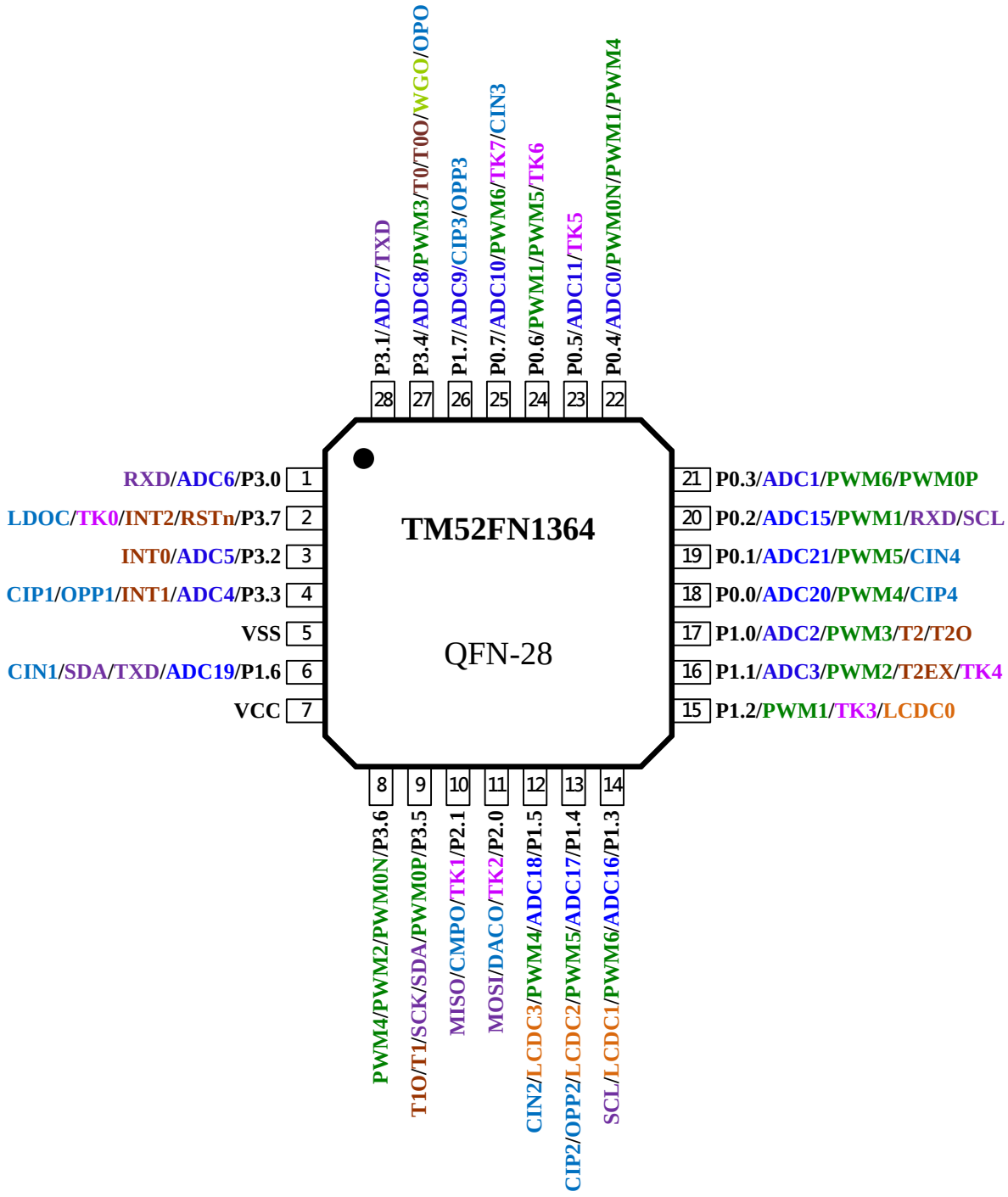
- 28-pin SOP (300 mil)
- 28-pin SSOP (150 mil)
- 28-pin QFN (4x4x0.75-0.4mm)

# PIN ASSIGNMENT

***UART default pin is P30, P31; user can set P02, P16 instead by TXRXSEL (SFR 93h.7)**
***Master I²C SDA default pin is P35; user can set P16 instead by MSDASEL (SFR B7h.7)**
***Master I²C SCL default pin is P13; user can set P02 instead by MSCLSEL (SFR B7h.6)**

**For low power applications, all digital I/Os (including unbonding or unused) should avoid high-impedance settings.**




TM52FN1364
QFN-28
1 RXD/ADC6/P3.0
2 LDOC/TK0/INT2/RSTn/P3.7
3 INT0/ADC5/P3.2
4 CIP1/OPP1/INT1/ADC4/P3.3
5 VSS
6 CIN1/SDA/TXD/ADC19/P1.6
7 VCC
8 PWM4/PWM2/PWM0N/P3.6
9 T1O/T1/SCK/SDA/PWM0P/P3.5
10 MISO/CMPO/TK1/P2.1
11 MOSI/DACO/TK2/P2.0
12 CIN2/LCDC3/PWM4/ADC18/P1.5
13 CIP2/OPP2/LCDC2/PWM5/ADC17/P1.4
14 SCL/LCDC1/PWM6/ADC16/P1.3
15 P1.2/PWM1/TK3/LCDC0
16 P1.1/ADC3/PWM2/T2EX/TK4
17 P1.0/ADC2/PWM3/T2/T2O
18 P0.0/ADC20/PWM4/CIP4
19 P0.1/ADC21/PWM5/CIN4
20 P0.2/ADC15/PWM1/RXD/SCL
21 P0.3/ADC1/PWM6/PWM0P
22 P0.4/ADC0/PWM0N/PWM1/PWM4
23 P0.5/ADC11/TK5
24 P0.6/PWM1/PWM5/TK6
25 P0.7/ADC10/PWM6/TK7/CIN3
26 P1.7/ADC9/CIP3/OPP3
27 P3.4/ADC8/PWM3/T0/T0O/WGO/OPO
28 P3.1/ADC7/TXD

# PIN DESCRIPTION

| Name | In/Out | Pin Description |
|---|---|---|
| P0.0~P0.7<br>P1.0~P1.7<br>P2.0~P2.1<br>P3.3~P3.7 | I/O | Bit-programmable I/O port for Schmitt-trigger input or CMOS push-pull output. Pull-up resistors are assignable by software. These pin's level change can interrupt/wake up CPU from Idle/Stop mode. |
| 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. These pin's level change can interrupt/wake up CPU from Idle/Stop mode. |
| INT0, INT1 | I | External low level or falling edge Interrupt input, Idle/Stop mode wake up input. |
| INT2 | I | External falling edge Interrupt input, Idle/Stop mode wake up input. |
| RXD | I/O | UART Mode0 transmit & receive data, Mode1/2/3 receive data |
| TXD | I/O | UART Mode0 transmit clock, Mode1/2/3 transmit data. In One Wire UART mode, this pin transmits and receives serial 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 |
| T1O | O | Timer1 overflow divided by 2 output |
| T2O | O | Timer2 overflow divided by 2 output |
| PWM1~PWM6<br>PWM0P/PWM0N | O | 16 bit PWM output |
| ADC0~ADC11,<br>ADC15~ADC21 | I | ADC input |
| LCDC0~LCDC3 | O | LCD 1/2 bias output |
| SCL | I/O | Master $I^2C$ (MIIC) SCL |
| SDA | I/O | Master $I^2C$ (MIIC) SDA |
| SCK | I/O | SPI clock |
| MISO, MOSI | I/O | SPI data input and output |
| TK0~TK7 | I | Touch Key input |
| CIN1~CIN4 | I | Comparator negative port input |
| CIP1~CIP4 | I | Comparator positive port input |
| CMPO | O | Comparator status output |
| DACO | O | DAC output |
| OPP1~OPP4 | I | OPA positive port input |
| OPO | O | OPA output |
| LDOC | O | LDO 1.2V high driver current output |
| WGO | O | Full color LED waveform generator output |
| RSTn | I | External active low reset input, Pull-up resistor is fixed enable. |
| VCC, VSS | P | Power input pin and ground |

# PIN SUMMERY

| Pin Number | Pin Name | Type | Input | | | Output | | | Alternative Function | | | | | | MISC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| QFN-28 | | | Pull-up Control | Wake up | Ext. Interrupt | CMOS Push-Pull | Pseudo Open Drain | Open Drain | LCD | ADC | Touch Key | PWM | OPA / CMP / DAC | MIIC / SPI | |
| 1 | RXD/ADC6/P3.0 | I/O | ● | ● | | ● | ● | ● | | ● | | | | | |
| 2 | INT2/RSTn/LDOC/TK0/P3.7 | I/O | ● | ● | ● | ● | | ● | | | ● | | | | Reset |
| 3 | INT0/VBGO/ADC5/P3.2 | I/O | ● | ● | ● | ● | ● | ● | | ● | | | | | VBGO |
| 4 | INT1/CIP1/OPP1/ADC4/P3.3 | I/O | ● | ● | ● | ● | | ● | | ● | | | ● | | |
| 5 | VSS | P | | | | | | | | | | | | | |
| 6 | CIN1/SDA/TXD/ADC19/P1.6 | I/O | ● | ● | | ● | | ● | | ● | | | ● | ● | |
| 7 | VCC | P | | | | | | | | | | | | | |
| 8 | PWM4/PWM2/PWM0N/P3.6 | I/O | ● | ● | | ● | | ● | | | | ● | | | |
| 9 | T1O/T1/SCK/SDA/PWM0P/P3.5 | I/O | ● | ● | | ● | | ● | | | | ● | | ● | T1O |
| 10 | CMPO/MISO/TK1/P2.1 | I/O | ● | ● | | ● | | ● | | | ● | | ● | ● | |
| 11 | DACO/MOSI/TK2/P2.0 | I/O | ● | ● | | ● | | ● | | | ● | | ● | ● | |
| 12 | CIN2/LCDC3/PWM4/ADC18/P1.5 | I/O | ● | ● | | ● | | ● | ● | ● | | ● | ● | | |
| 13 | CIP2/OPP2/LCDC2/PWM5/ADC17/P1.4 | I/O | ● | ● | | ● | | ● | ● | ● | | ● | ● | | |
| 14 | SCL/LCDC1/PWM6/ADC16/P1.3 | I/O | ● | ● | | ● | | ● | ● | ● | | ● | | ● | |
| 15 | LCDC0/PWM1/TK3/P1.2 | I/O | ● | ● | | ● | | ● | ● | | ● | ● | | | |
| 16 | T2EX/PWM2/TK4/ADC3/P1.1 | I/O | ● | ● | | ● | | ● | | ● | ● | ● | | | |
| 17 | T2O/T2/PWM3/ADC2/P1.0 | I/O | ● | ● | | ● | | ● | | ● | | ● | | | T2O |
| 18 | CIP4/PWM4/ADC20/P0.0 | I/O | ● | ● | | ● | | | | ● | | ● | ● | | |
| 19 | CIN4/PWM5/ADC21/P0.1 | I/O | ● | ● | | ● | | | | ● | | ● | ● | | |
| 20 | SCL/RXD/PWM1/ADC15/P0.2 | I/O | ● | ● | | ● | | | | ● | | ● | | ● | |
| 21 | PWM0P/PWM6/ADC1/P0.3 | I/O | ● | ● | | ● | | | | ● | | ● | | | |
| 22 | PWM4/PWM1/PWM0N/ADC0/P0.4 | I/O | ● | ● | | ● | | | | ● | | ● | | | |
| 23 | TK5/ADC11/P0.5 | I/O | ● | ● | | ● | | | | ● | ● | | | | |
| 24 | PWM5/PWM1/TK6/P0.6 | I/O | ● | ● | | ● | | | | | ● | ● | | | |
| 25 | CIN3/PWM6/TK7/ADC10/P0.7 | I/O | ● | ● | | ● | | | | ● | ● | ● | ● | | |
| 26 | CIP3/OPP3/ADC9/P1.7 | I/O | ● | ● | | ● | | ● | | ● | | | ● | | |
| 27 | T0O/T0/OPO/WGO/PWM3/ADC8/P3.4 | I/O | ● | ● | | ● | | ● | | ● | | ● | ● | | T0O/WGO |
| 28 | TXD /ADC7/P3.1 | I/O | ● | ● | | ● | ● | ● | | ● | | | | | |

# 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 | – | ADSOC | CLRPWM0 | CLRPWM1 | LDOCOUT | DPSEL |
| R/W | R/W | R/W | – | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | – | 0 | 1 | 1 | 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 | | |
|---|---|---|---|
| | C | OV | AC |
| ADD | X | X | X |
| ADDC | X | X | X |
| SUBB | X | X | X |
| MUL | 0 | X | |
| DIV | 0 | X | |
| DA | X | | |
| RRC | X | | |
| RLC | X | | |
| SETB C | 1 | | |

| Instruction | Flag | | |
|---|---|---|---|
| | C | OV | AC |
| CLR C | 0 | | |
| CPL C | X | | |
| ANL C, bit | X | | |
| ANL C, /bit | X | | |
| ORL C, bit | X | | |
| ORL C, /bit | X | | |
| MOV C, bit | X | | |
| CJNE | X | | |
| | | | |

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)

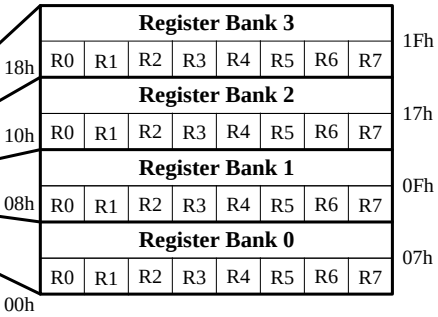
D0h.2 **OV:** ALU overflow flag

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.

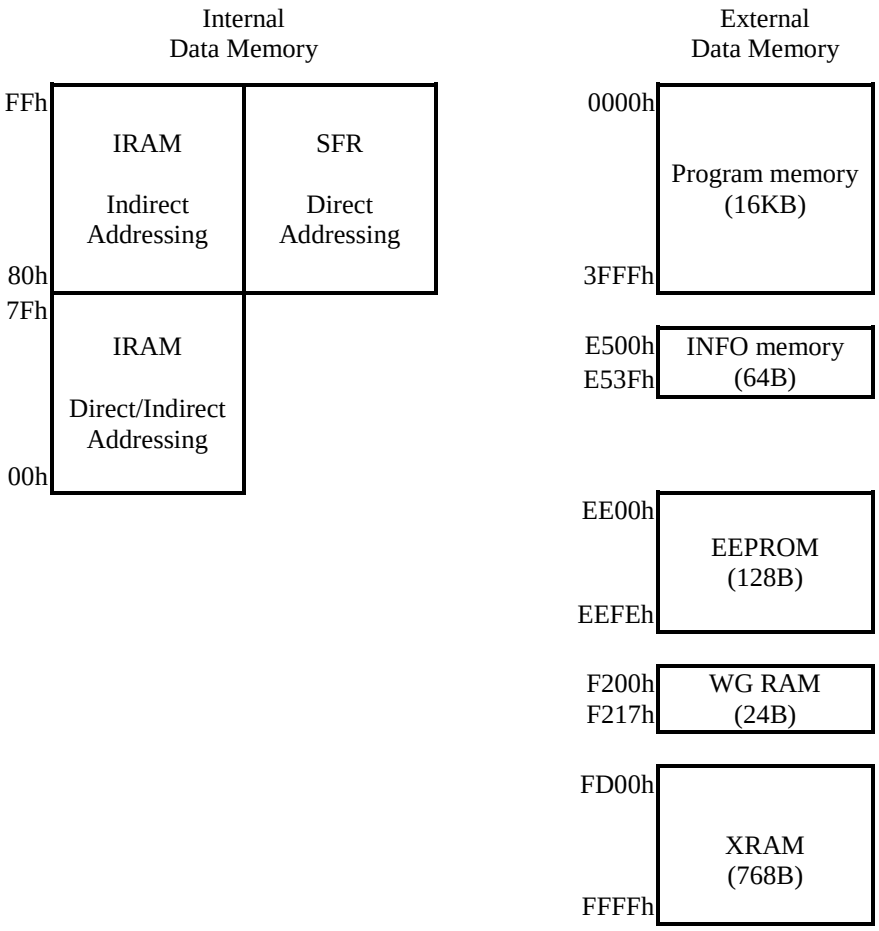
| PSW | | | | | | | |
|---|---|---|---|---|---|---|---|
| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| **CY** | **AC** | **FO** | **RS1** | **RS0** | **OV** | **F1** | **P** |

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



## 2. Memory

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 768 bytes XRAM, 128 bytes EEPROM, 24 bytes WG RAM, 64 bytes INFO memory and 16K bytes Program memory, which can be only accessed by MOVX instruction, Program memory also can be accessed by MOVC instruction.

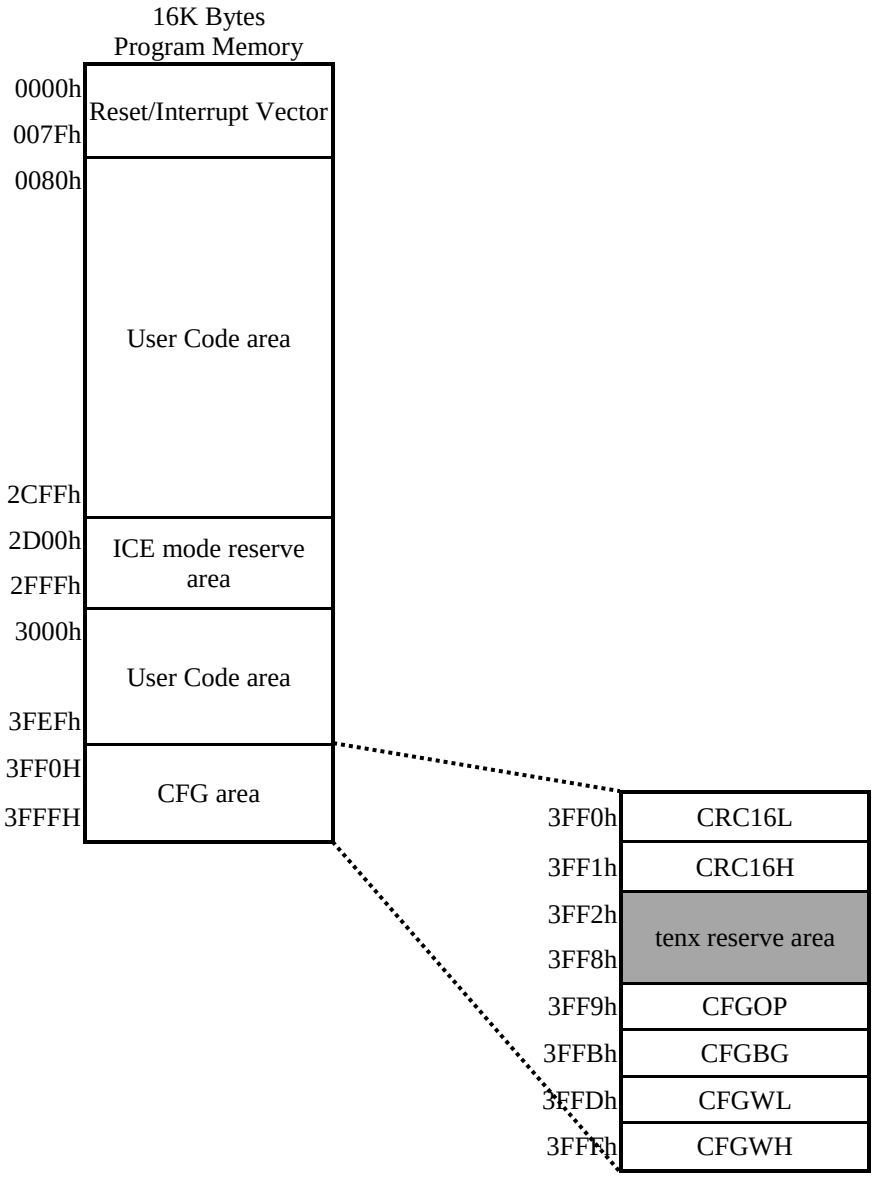


## 2.1 Program Memory (Support IAP)

The Chip has a 16K bytes Flash program memory which can support In Circuit Programming (ICP), In Application Programming (IAP) and In System Programming (ISP) function modes. The program memory address continuous space (0000h~3FFFh) is partitioned to several sectors for device operation.

### 2.1.1 Functional Partition

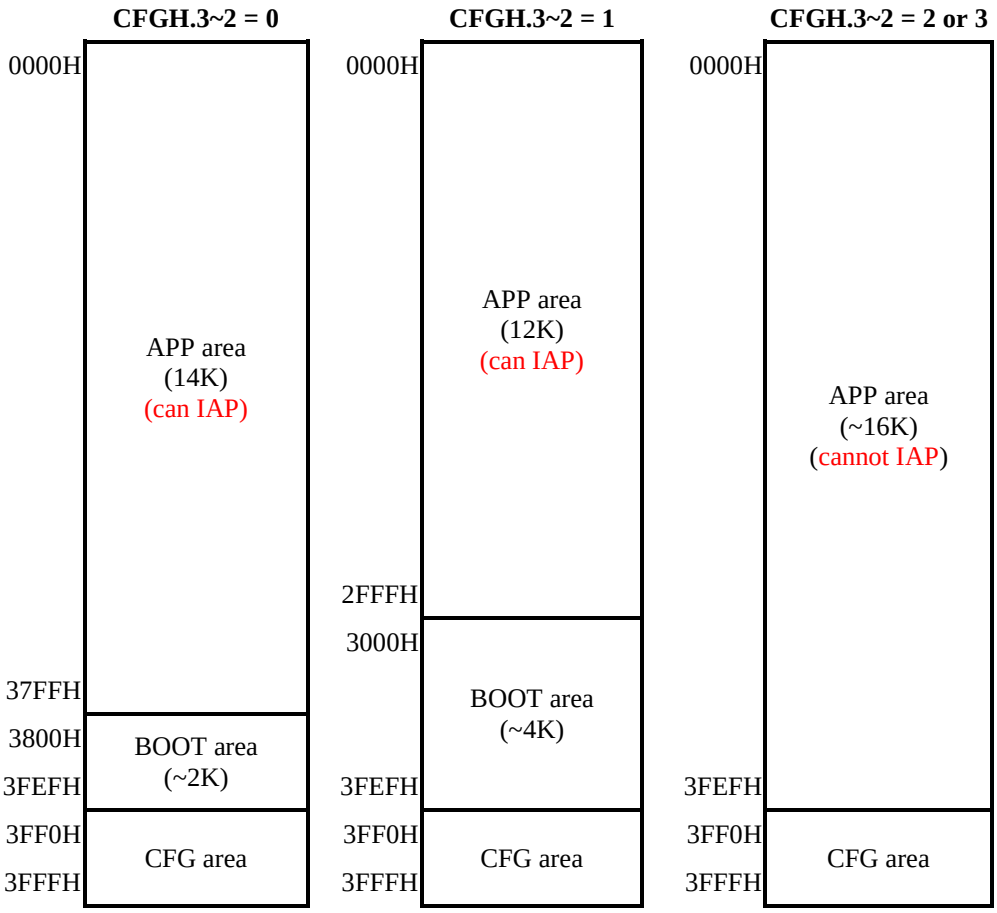
The last 16 bytes (3FF0h~3FFFh) of program memory is defined as chip Configuration Word (CFGW), which is loaded into the device control registers upon power on reset (POR). The 0000h~007Fh is occupied by Reset/Interrupt vectors as standard 8051 definition. In the in-circuit emulation (ICE) mode, user also needs to reserve the address space 2D00h~2FFFh 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.




**Program Memory Partition**

User can select different BOOT mode by CFGH.3~2. If CFGH.3~2=0, reset vector=0x3800 and BOOT area is 2KB, if CFGH.3~2=1, reset vector=0x3000 and BOOT area is 4KB, if CFGH.3~2=2 or 3, Boot mode is disable. User also can keep reset vector to 0x0000 in BOOT mode by set RSTV=0 (BFh.2).

Only App area can IAP write. In different BOOT modes, the writable area is also different; see the figure below for details.




**Flash partition determined by different BOOT modes**

### 2.1.2 Flash IAP Mode

The chip has "In Application Program" (IAP) capability, which allows software to read/write data from/to the Flash memory during CPU run time as conveniently as data EEPROM access. The IAP function is byte writable, meaning that the chip does not need to erase one Flash page before write.

Both write 47h and 74h to IAPCON (C9h.7~0) can let IAPWE=1, the difference is when user write 47h to IAPCON, user can write one byte at once, when user write 74h to IAPCON, user can write two byte at once to save write time.

When IAPALL=1 and IAPWE=1, the user is allowed to use the IAP function to write to the APP area in BOOT mode.

To use IAP function, user need to meet the following conditions:

1. In BOOT mode.
2. Only APP area can be written by IAP.
3. Set IAPALL=1 and IAPWE=1.

**Flash IAP Write** is simply achieved by a "MOVX @DPTR, A" instruction while the DPTR contains the target Flash address, and the ACC contains the data being written. The Flash IAP need to be written twice. Flash writing requires approximately 0.6 ms @$V_{CC}$=5.0V~5.5V, VCC capacitance greater than 220uF. During the period of IAP, the CPU stays in a waiting state, but all peripheral modules continue running during the writing time. The software must handle the pending interrupts after an IAP write. The chip has a build-in write Time-out function selected by IAPTE (F7h.2~1) to escape write fail state. Besides, S/W must disable WDT before IAP write. The IVC function must be turned off when writing to the Flash.

**Flash IAP Read** can be performed by the "MOVC" or "MOVX" instruction as long as the target address points to the 0000h~3FFFh area. A Flash IAP read does not require extra CPU wait time.

One-byte IAP Example: (In BOOT mode)

```
; need 5.0V < VCC < 5.5V & WDT disable
ANL     AUX2, #3Fh          ; Disable WDT
ORL     PWRCON, #80h        ; IVCPD=1
MOV     DPTR, #1F00h        ; DPTR=1F00h=target IAP address
MOV     A, #5Ah             ; A=5Ah=target IAP write data
ORL     AUX2,  #04h         ; IAP Time-Out function select
MOV     SWCMD, #65h         ; IAPALL flag=1
MOV     IAPCON, #47h        ; IAPWE flag=1,one-byte write
                            ; Flash IAP write enable if IAPALL=IAPWE=1
MOVX    @DPTR, A            ; IAP Write Flash
                            ; Flash[1F00h] =5Ah after IAP write
MOVX    @DPTR, A            ; IAP Write Flash twice
MOV     IAPCON, #00h        ; IAP write disable, immediately after IAP write
ANL     PWRCON, #7Fh        ; IVCPD=0
MOVX    A, @DPTR            ; Read Flash.  A=5Ah
```

Two-byte IAP Example: (In BOOT mode)

```
; need 5.0V < VCC < 5.5V & WDT disable
ANL     AUX2, #3Fh          ; Disable WDT
ORL     PWRCON, #80h        ; IVCPD=1
MOV     DPTR, #1F01h        ; DPTR=1F01h mean target IAP address=1F00h~1F01h
                            ; DPTR must be "odd" if user want to use two-byte IAP.
MOV     A, #ABh             ; A=ABh=target IAP write high byte data
MOV     B, #CDh             ; B=CDh=target IAP write low byte data
ORL     AUX2,  #04h         ; IAP Time-Out function select
MOV     SWCMD, #65h         ; IAPALL flag=1
MOV     IAPCON, #74h        ; IAPWE flag=1,two-byte write
                            ; Flash IAP write enable if IAPALL=IAPWE=1
MOVX    @DPTR, A            ; IAP Write Flash
                            ; Flash[1F00h] =5Ah after IAP write
MOVX    @DPTR, A            ; IAP Write Flash twice
MOV     IAPCON, #00h        ; IAP write disable, immediately after IAP write
ANL     PWRCON, #7Fh        ; IVCPD=0
MOVX    A, @DPTR            ; Read Flash.  A=ABh
```

```
MOV     DPTR, #1F00h        ;
MOVX    A, @DPTR            ; Read Flash.  A=CDh
```

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

97h.7~0 **IAPALL (W):**
Write 65h to set IAPALL flag. Write other value to clear IAPALL flag.

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

97h.0 **IAPALL (R):** Flag indicates Flash can be written by IAP or not
0: Flash IAP disable
1: Flash IAP enable, only for BOOT mode upgrade APP area.

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

C9h.7~0 **IAPCON (W):**
Write 47h or 74h to set IAPWE flag; Write 47h can write 1 byte at once, write 74h can write 2 bytes at once. Write other value to clear IAPWE flag. It is recommended to clear it immediately after IAP write.
Write A1h to set INFOWE flag; write other value to clear INFOWE flag. It is recommended to clear it immediately after IAP write.
Write E2h to set EEPWE flag; write other value to clear EEPWE flag. It is recommended to clear it immediately after EEPROM write.

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

C9h.7 **IAPWE (R):** Flag indicates Flash memory can be written by IAP or not
0: IAP Write disable
1: IAP Write enable

C9h.6 **IAPTO (R):** Time-Out flag of IAP write/EEPROM write/INFO write. Set by H/W when IAP or EEPROM write or INFO write Time-out occurs. Cleared this flag by H/W when IAPWE=0 or EEPWE=0 or INFOWE=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 | 1 | 1 | 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

F7h.2~1 **IAPTE:** IAP (or EEPROM) write watchdog timer enable
00: Disable

01: wait 1.6ms trigger watchdog time-out flag, and escape the write fail state
10: wait 3.1ms trigger watchdog time-out flag, and escape the write fail state
11: wait 12.5ms trigger watchdog time-out flag, and escape the write fail state

### 2.1.3 Flash ICP Mode

The Flash memory can be programmed by the tenx 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.

### 2.1.4 Flash ISP Mode

The “In System Programming” (ISP) usage is similar to IAP, except the purpose is to refresh the Program code. User can use UART or other method to get new Program code from external host, then writes code as the same way as IAP. ISP operation is complicated; basically it needs to assign a Boot code area to the Flash which does not change during the ISP process.

## 2.2 Information Memory (Support IAP)

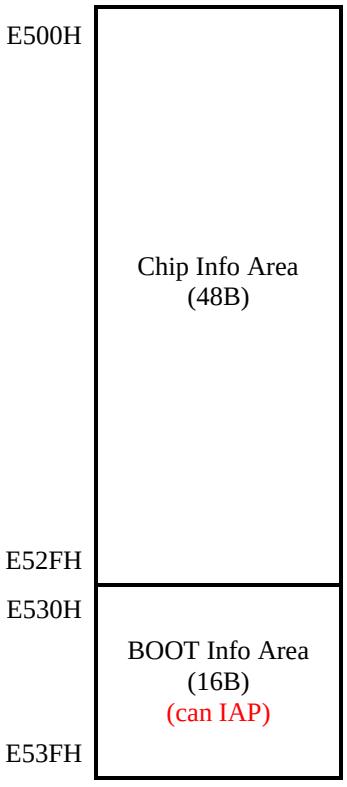
The Chip has a 64 bytes Information memory. The Information memory address continuous space (E500h~E53Fh) is partitioned to several sectors for device operation.

Chip Info area is tenx reserved defined as production information, such as ID, Special Regulations, Code Num, checksum. BOOT Info area allow IAP write in BOOT mode, user can store new checksum code in this area after Flash IAP.

The IVC function must be turned off when writing to the Information memory.

To use IAP function, user need to meet the following conditions:

1. In BOOT mode.
2. Only BOOT Info Area can be written by IAP.
3. Set INFOWE=1.




**Info ROM partition**

**Info ROM IAP Write** is simply achieved by a "MOVX @DPTR, A" instruction while the DPTR contains the target Flash address, and the ACC contains the data being written. Flash writing requires approximately 0.6 ms @$V_{CC}$=4.0V~5.5V, VCC capacitance greater than 220uF. During the period of IAP, the CPU stays in a waiting state, but all peripheral modules continue running during the writing time. The software must handle the pending interrupts after an IAP write. The chip has a build-in write Time-out function selected by IAPTE(F7h.2~1) to escape write fail state. Besides, S/W must disable WDT before IAP write.

**Info ROM IAP Read** only can be performed by the "MOVX" instruction as long as the target address points to the E500h~E53Fh area. A Info ROM IAP read does not require extra CPU wait time.

Info ROM IAP Example: (In BOOT mode)

```
; need 4.0V < VCC < 5.5V & WDT disable
ANL     AUX2, #3Fh          ; Disable WDT
ORL     PWRCON, #80h        ; IVCPD=1
MOV     DPTR, #E530h        ; DPTR=E530h=target IAP address
MOV     A, #5Ah             ; A=5Ah=target IAP write data
ORL     AUX2,  #04h         ; IAP Time-Out function select
MOV     IAPCON, #A1h        ; Info ROM IAP write enable.
MOVX    @DPTR, A            ; IAP Write Info ROM
                            ; Info ROM[E530h] =5Ah after IAP write
MOVX    @DPTR, A            IAP Write Info ROM twice
MOV     IAPCON, #00h        ; IAP write disable, immediately after IAP write
ANL     PWRCON, #7Fh        ; IVCPD=0
MOVX    A, @DPTR            ; Read Info ROM.  A=5Ah
```

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

C9h.7~0 **IAPCON (W):**
Write 47h or 74h to set IAPWE flag; Write 47h can write 1 byte at once, write 74h can write 2 bytes at once. Write other value to clear IAPWE flag. It is recommended to clear it immediately after IAP write.
Write A1h to set INFOWE flag; write other value to clear INFOWE flag. It is recommended to clear it immediately after IAP write.
Write E2h to set EEPWE flag; write other value to clear EEPWE flag. It is recommended to clear it immediately after EEPROM write.

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

C9h.6 **IAPTO (R):** Time-Out flag of IAP write/EEPROM write/INFO write. Set by H/W when IAP or EEPROM write or INFO write Time-out occurs. Cleared this flag by H/W when IAPWE=0 or EEPWE=0 or INFOWE=0.

C9h.4 **INFOWE (R):** Flag indicates INFO memory can be written by IAP or not
0: INFO IAP Write disable
1: INFO IAP Write enable

| 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 | 1 | 1 | 0 |

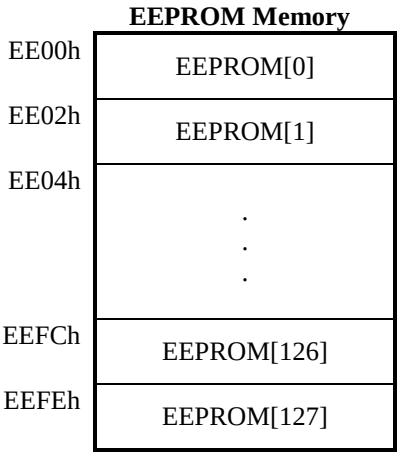
F7h.7~6 **WDTE:** Watchdog Timer Reset control
0x: Watchdog Timer Reset disable

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

F7h.2~1 **IAPTE:** IAP write/EEPROM write/INFO write watchdog timer enable
00: Disable
01: wait 1.6ms trigger watchdog time-out flag, and escape the write fail state
10: wait 3.1ms trigger watchdog time-out flag, and escape the write fail state
11: wait 12.5ms trigger watchdog time-out flag, and escape the write fail state

## 2.3 EEPROM Memory

The chip contains 128 bytes of data EEPROM memory. It is organized as a separate data space, in which single bytes can be read and written. The EEPROM has an endurance of at least 50K write/erase cycles.




**(Only even addresses can be used, odd addresses are invalid)**

**The EEPROM Write** usage is similar to Flash IAP mode. It is simply achieved by a "MOVX @DPTR, A" instruction while the DPTR contains the target EEPROM address, and the ACC contains the data being written. EEPROM writing requires approximately 0.6 ms @$V_{CC}$=3.5V~5.5V, VCC capacitance greater than 220uF. When the EEPROM is being written, the CPU stays in a waiting state, but all peripheral modules (Timers, LED, and others) continue running during the writing time. The software must handle the pending interrupts after an EEPROM write. The chip has a build-in EEPROM Time-out function shared with Flash IAP for escaping write fail state. The chip has a build-in write Time-out function selected by IAPTE (F7h.2~1) to escape write fail state. Besides, S/W must disable WDT before EEPROM write.

The IVC function must be turned off when writing to the EEPROM.

**The EEPROM Read** can be performed by the "MOVX A, @DPTR" instruction as long as the target address points to the EE00h~EEFEh area.

EEPROM example code:

```
; need 3.5V < VDD < 5.5V & WDT disable
ANL     AUX2, #3Fh          ; Disable WDT
ORL     PWRCON, #80h        ; IVCPD=1
MOV     DPTR, #EE00h        ; DPTR=EE00h=target EEPROM[0] address
MOV     A, #A5h             ; A=A5h=target EEPROM[0] write data
ORL     AUX2, #04h          ; IAP Time-Out function select.
MOV     IAPCON, #E2h        ; EEPROM write enable
MOVX    @DPTR, A            ; Write EEPROM.
MOV     IAPCON, #00h        ; EEPROM write disable, immediately after EEPROM write
ANL     PWRCON, #7Fh        ; IVCPD=0
MOVX    A, @DPTR            ; Read EEPROM. A=A5h.
```

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

C9h.7~0 **IAPCON (W):**
Write 47h or 74h to set IAPWE flag; Write 47h can write 1 byte at once, write 74h can write 2 bytes at once. Write other value to clear IAPWE flag. It is recommended to clear it immediately after IAP write.
Write A1h to set INFOWE flag; write other value to clear INFOWE flag. It is recommended to clear it immediately after IAP write.
Write E2h to set EEPWE flag; write other value to clear EEPWE flag. It is recommended to clear it immediately after EEPROM write.

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

C9h.6 **IAPTO (R):** Time-Out flag of IAP write/EEPROM write/INFO write. Set by H/W when IAP or EEPROM write or INFO write Time-out occurs. Cleared this flag by H/W when IAPWE=0 or EEPWE=0 or INFOWE=0.

C9h.5 **EEPWE (R):** Flag indicates EEPROM can be written or not
0: EEPROM Write disable
1: EEPROM Write enable

| 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 | 1 | 1 | 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

F7h.2~1 **IAPTE:** IAP write/EEPROM write/INFO write watchdog timer enable
00: Disable
01: wait 1.6ms trigger watchdog time-out flag, and escape the write fail state
10: wait 3.1ms trigger watchdog time-out flag, and escape the write fail state
11: wait 12.5ms trigger watchdog time-out flag, and escape the write fail state

## 2.4 IRAM

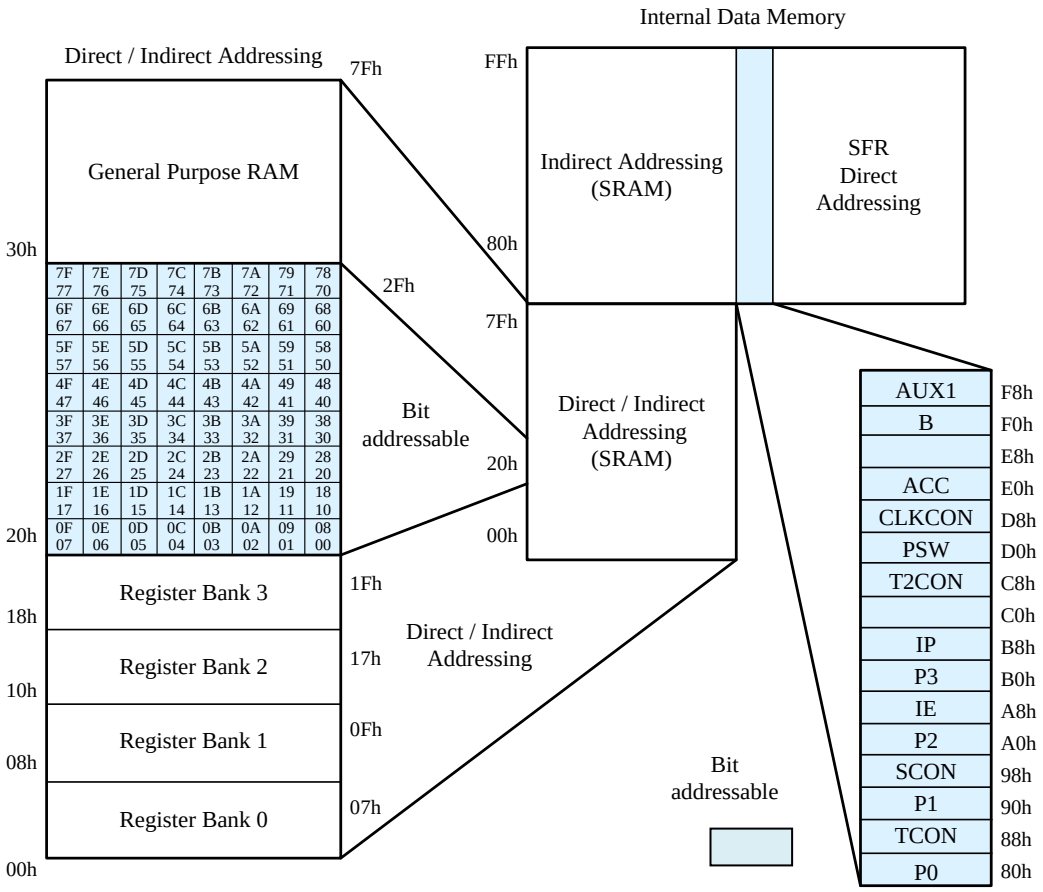
IRAM is located in the 8051 internal data memory space. The whole 256 Bytes IRAM are accessible using indirect addressing but only the lower 128 Bytes are accessible using direct addressing. There are four directly addressable register banks (switching by PSW), which occupy IRAM space from 00h to 1Fh.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

## 2.5 XRAM

XRAM is located in the 8051 external data memory space (address from FD00h to FFFFh). The 768 bytes XRAM can be only accessed by "MOVX" instruction.

## 2.6 Special Function Register (SFR)

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.



| | 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 | | PWM4DH | PWM4DL | PWM5DH | PWM5DL | PWM6DH | PWM6DL | PWRCON |
| E0h | ACC | MICON | MIDAT | LVRCON | LVDCON | EFTCON | EXA | EXB |
| D8h | CLKCON | PWM0PRDH | PWM0PRDL | PWM1PRDH | PWM1PRDL | PWM3DH | PWM3DL | UARTCON |
| D0h | PSW | PWM0DH | PWM0DL | PWM1DH | PWM1DL | PWM2DH | PWM2DL | CFGOP |
| C8h | T2CON | IAPCON | RCP2L | RCP2H | TL2 | TH2 | EXA2 | EXA3 |
| C0h | | TKBTMRL | TKBTMRH | TKBKCP | TKBREFC | P0WKUP | P2WKUP | P3WKUP |
| B8h | IP | IPH | IP1 | IP1H | SPCON | SPSTA | SPDAT | BOOTV |
| B0h | P3 | TKAREFC | TKADH | TKCHS | TKATMRL | TKATMRH | PWMOE1 | PWMOE2 |
| A8h | IE | INTE1 | ADCDL | ADCDH | TKADL | TKCON | CHSEL | P1LOE |
| A0h | P2 | PWMCON | P1MODL | P1MODH | P3MODL | P3MODH | PWMOE0 | PWMCON2 |
| 98h | SCON | SBUF | WGCON | WGCON2 | DACON | CMPCON | CMPPNS | OPCON |
| 90h | P1 | P0MODL | P0MODH | PINMOD | OPTION | INTFLG | P1WKUP | SWCMD |
| 88h | TCON | TMOD | TL0 | TL1 | TH0 | TH1 | TKBDL | TKBDH |
| 80h | P0 | SP | DPL | DPH | INTE2 | INTFLG2 | DACON2 | PCON |

**SFR table**

# 3. Reset

The chip has five types of reset (Reset) methods. Power-on reset (POR), external pin reset (XRST), software reset (SWRST), watchdog timer reset (WDTR) and low voltage reset (LVR), SFR returns to default values after reset.

After reset, the Program memory address will start at 0000h, 3000h or 3800h depended by RSTV (BFh.2) and CFGH.3~2.

## 3.1 Power on Reset (POR)

After power-on reset, the device stays in the reset state and the preheating time of this chip is about 40ms. A power-on reset requires the voltage on the VCC pin to discharge to near the VSS level before rising above 2.2V(TBD). POR is automatically turned off when the chip enters HALT/STOP mode and can be enabled or disabled by PORPD (E3h.5) when the chip is not in HALT/STOP mode.

## 3.2 External Pin Reset (XRST)

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 enabled or disabled by CFGWH.6.

## 3.3 Software Command Reset (SWRST)

Software reset is generated by writing data 56h to SWCMD (97h).

## 3.4 Watchdog Timer Reset (WDTR)

WDT overflow reset is controlled by WDTE (F7h.7~6). The WDT uses SRC as the count time base, runs in FAST/SLOW clock mode, and optionally runs or stops in IDLE/HALT/STOP clock mode. The watchdog timer overflow speed can be defined by WDTPSC (94h.5~4). WDT is cleared by CLRWDT (F8h.7) or reset.

## 3.5 Low Voltage Reset (LVR)

Low voltage reset (LVR) can select 16 different voltage thresholds through LVRCON (E3h.3~0). When PWRSAV (F7h.5) =1, the LVR will automatically turn off when the chip enters IDLE/HALT/STOP mode. It can be enabled or disabled by LVRPD (E3h.4).

***Note:*** *refer to AP-TM52XXXXX_02S for LVR setting information*

| Operation Mode | SFR | | CFGWH | LVR | Function | Note |
|---|---|---|---|---|---|---|
| | LVRPD | PWRSAV | LVRE | | | |
| Fast Slow | 0 | X | 0000 | ON | LVR 2.05V | |
| | 0 | X | 0001 | ON | LVR 2.19V | |
| | 0 | X | 0010 | ON | LVR 2.33V | |
| | 0 | X | 0011 | ON | LVR 2.47V | |
| | 0 | X | 0100 | ON | LVR 2.61V | |
| | 0 | X | 0101 | ON | LVR 2.75V | |
| | 0 | X | 0110 | ON | LVR 2.89V | |
| | 0 | X | 0111 | ON | LVR 3.03V | |
| | 0 | X | 1000 | ON | LVR 3.17V | |
| | 0 | X | 1001 | ON | LVR 3.31V | |
| | 0 | X | 1010 | ON | LVR 3.45V | |
| | 0 | X | 1011 | ON | LVR 3.59V | |
| | 0 | X | 1100 | ON | LVR 3.73V | |
| | 0 | X | 1101 | ON | LVR 3.87V | |
| | 0 | X | 1110 | ON | LVR 4.01V | |
| | 0 | X | 1111 | ON | LVR 4.15V | |
| Idle Halt Stop | 0 | 0 | 0000 | ON | LVR 2.05V | Current consumption about 60uA(TBD) |
| | 0 | 0 | 0001 | ON | LVR 2.19V | |
| | 0 | 0 | 0010 | ON | LVR 2.33V | |
| | 0 | 0 | 0011 | ON | LVR 2.47V | |
| | 0 | 0 | 0100 | ON | LVR 2.61V | |
| | 0 | 0 | 0101 | ON | LVR 2.75V | |
| | 0 | 0 | 0110 | ON | LVR 2.89V | |
| | 0 | 0 | 0111 | ON | LVR 3.03V | |
| | 0 | 0 | 1000 | ON | LVR 3.17V | |
| | 0 | 0 | 1001 | ON | LVR 3.31V | |
| | 0 | 0 | 1010 | ON | LVR 3.45V | |
| | 0 | 0 | 1011 | ON | LVR 3.59V | |
| | 0 | 0 | 1100 | ON | LVR 3.73V | |
| | 0 | 0 | 1101 | ON | LVR 3.87V | |
| | 0 | 0 | 1110 | ON | LVR 4.01V | |
| | 0 | 0 | 1111 | ON | LVR 4.15V | |
| Idle | 0 | 1 | XXXX | ON | Disable LVR Enable POR | Current consumption about 20uA(TBD) |
| Halt Stop | 0 | 1 | XXXX | OFF | Disable | Minimum current consumption about 0.1uA |
| Fast Slow Idle | 1 | X | XXXX | ON | Disable LVR Enable POR | Current consumption about 20uA(TBD) |
| Halt Stop | 1 | X | XXXX | OFF | Disable | Minimum current consumption about 0.1uA |

***Note:*** *The current consumption of Halt mode is more than STOP mode about 2 ~ 5uA, because SRC is enabled.*

| SFR **94h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **OPTION** | UART1W | TM3CKS | WDTPSC | | ADCKS | | TM3PSC | |
| R/W | R/W | R/W | R/W | | R/W | | R/W | |
| Reset | 0 | 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 **97h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **SWCMD** | IAPALL/SWRST | | | | | | | |
| R/W | W | | | | | | | |
| Reset | – | | | | | | | |

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

| SFR **E3h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **LVRCON** | – | – | PORPD | LVRPD | LVRSEL | | | |
| R/W | – | – | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | – | – | 0 | 0 | 0 | 0 | 0 | 0 |

E3h.5 **PORPD**： POR Power Down.
0: POR Enable, 1: POR Disable

E3h.4 **LVRPD**： LVR Power Down.
0: LVR Enable, 1: LVR Disable

E3h.3~0 **LVRSEL**： Low Voltage Reset function select.
0000: Set LVR at 2.05V
0001: Set LVR at 2.19V
0010: Set LVR at 2.33V
0011: Set LVR at 2.47V
0100: Set LVR at 2.61V
0101: Set LVR at 2.75V
0110: Set LVR at 2.89V
0111: Set LVR at 3.03V
1000: Set LVR at 3.17V
1001: Set LVR at 3.31V
1010: Set LVR at 3.45V
1011: Set LVR at 3.59V
1100: Set LVR at 3.73V
1101: Set LVR at 3.87V
1110: Set LVR at 4.01V
1111: Set LVR at 4.15V

| 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 | 1 | 1 | 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

F7h.5 **PWRSAV:** chip power-saving option
Set 1 to reduce the chip's power consumption at 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 | – | ADSOC | CLRPWM0 | CLRPWM1 | LDOCOUT | DPSEL |
| R/W | R/W | R/W | – | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | – | 0 | 1 | 1 | 0 | 0 |

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

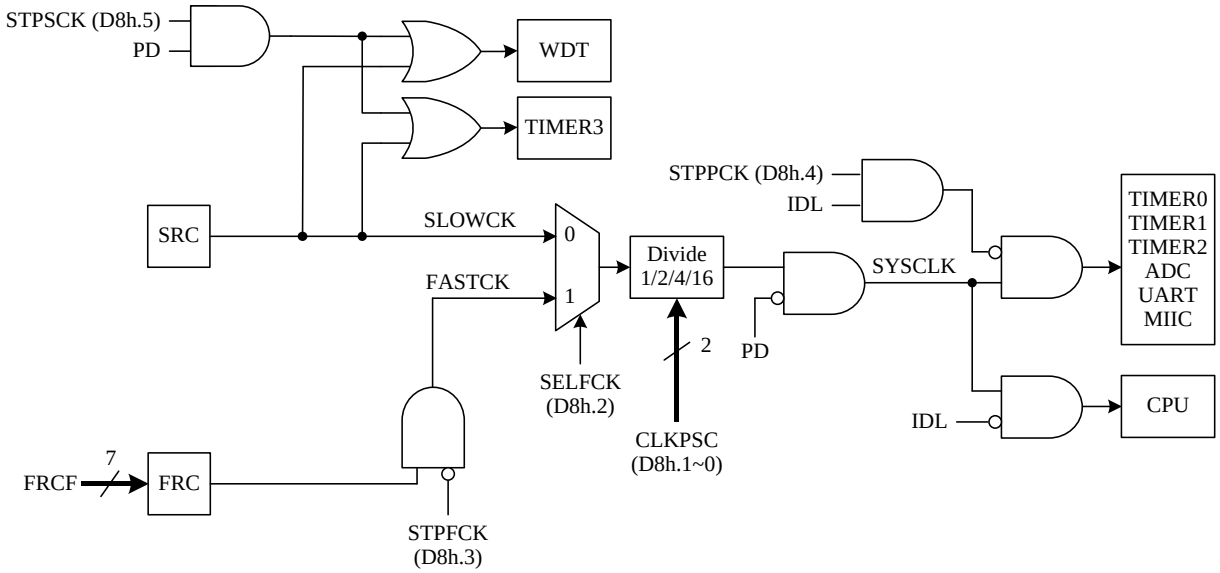
# 4. Clock Circuitry & Operation Mode

## 4.1 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 FRC (Fast Internal RC, 16.588 MHz). The Slow clock can be selected as SRC (Slow Internal RC, 32 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 32 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.

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.




**Clock Structure**

***Note:*** *Because of the CLKPSC delay, it needs to wait for 16 clock cycles (max.) before switching Slow clock to Fast clock. Also refer to AP-TM52XXXXX_01S and AP-TM52XXXXX_02S about System Clock Application Note.*

| SYSCLK | CLKCON (D8h) | |
|---|---|---|
| | bit3 STPFCK | bit2 SELFCK |
| Fast FRC | 0 | 1 |
| Slow SRC | 0/1 | 0 |
| Stop FRC | 0 → 1 | 0 |
| Switch to FRC | 0 | 0 → 1 |
| Switch to SRC | 0 | 1 → 0 |

| Flash **1FFDh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **CFGWL** | – | FRCF | | | | | | |

1FFDh.6~0 **FRCF:** FRC frequency adjustment.
FRC is trimmed to 16.588 MHz in chip manufacturing. FRCF records the adjustment data.

| 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
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** | – | – | STPSCK | STPPCK | STPFCK | SELFCK | CLKPSC | |
| R/W | – | – | R/W | R/W | R/W | R/W | R/W | |
| Reset | – | – | 0 | 0 | 0 | 0 | 1 | 1 |

D8h.5 **STPSCK:** Set 1 to stop slow clock in Stop mode.

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

## 4.2 Operation Modes

There are 5 operation modes for this device. The power consumption is lower when the system clock speed is lower.

**Fast Mode:**

Fast Mode is defined as the CPU running at Fast clock speed.

**Slow Mode:**

Slow Mode is defined as the CPU running at Slow clock speed.

**Idle Mode:**

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 Interrupts.

**Halt Mode:**

Halt Mode is entered by setting the **PD** bit in PCON SFR and clearing the **STPSCK** bit in CLKCON SFR. In Halt mode, all clocks are stopped, but Timer3 and WDT may be on if they are enabled. Halt mode can be terminated by Reset, Interrupt or Pin wakeup.

**Stop Mode:**

Stop Mode is entered by setting the **PD** bit in PCON SFR and setting the **STPSCK** bit in CLKCON 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 wakeup.

***Note:*** *The chip cannot enter Halt/Stop Mode if the INTn pin is low and the INTn wake-up function is enabled. (INTn=0 and EXn=1, n=0,1,2)*

| 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 HALT/STOP 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 | 1 | 1 | 0 |

F7h.4 **VBGOUT:** VBG voltage output to P3.2
0: Disable 1: Enable

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

D8h.5 **STPSCK:** Set 1 to stop Slow clock in Stop mode.

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

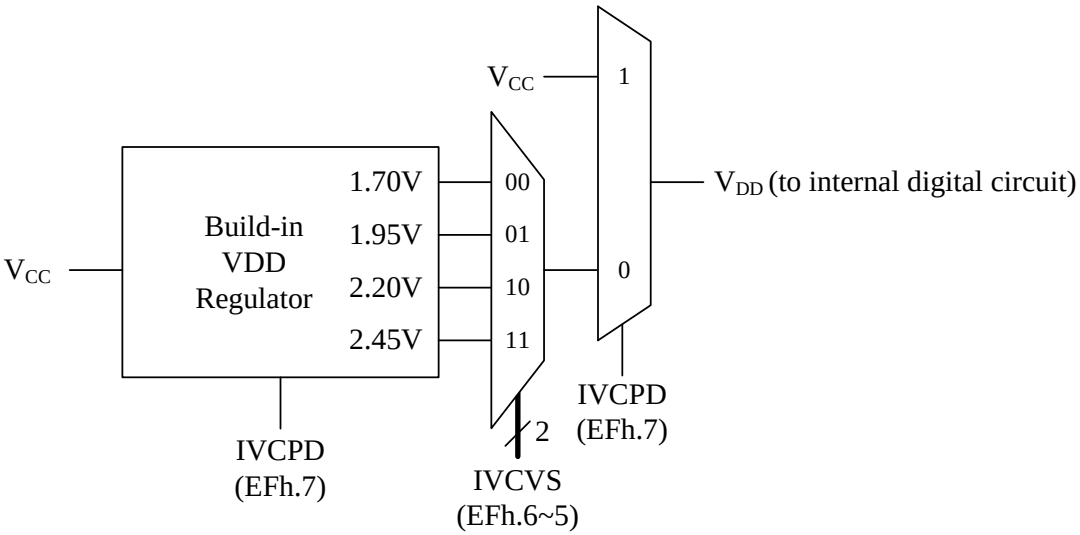
### 4.3 IVC (Build-in VDD regulator)

User can set IVCPD (EFh.7) = 0 to enable IVC function. it can reduce VDD voltage in **Fast mode** or **Slow mode** for power-saving, and be selected different VDD voltage by IVCVS (EFh.6~5).

User has to set IVCPD=1 before going to **Idle mode**, **Halt mode**, or **Stop mode**.

It is only recommended to use IVCVS=3 (highest level). Before using IVC, the LVR or LVD must be turned on first, and the user cannot turn off the LVD or LVR when IVC is enabled.

The IVC must be turned off temporarily during writing to Program Memory, Information Memory, or EEPROM.




**IVC Schematic**

| SFR **EFh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWRCON** | IVCPD | IVCVS | | – | WARMTIME | – | – | – |
| R/W | R/W | R/W | | – | R/W | – | – | – |
| Reset | 1 | 1 | 1 | – | 0 | – | – | – |

EFh.7 **IVCPD:** IVC(build-in VDD regulator) power down
0: IVC Enable ($V_{DD}$ = IVC's voltage)
1: IVC Disable ($V_{DD}$ = $V_{CC}$)

EFh.6~5 **IVCVS:** IVC Voltage select
00: 1.70V
01: 1.95V
10: 2.20V
11: 2.45V (recommended )

*Note: The VCC voltage must be higher than the IVC voltage, when IVC is enabled.*

*Note: F/W must turn on LVR or LVD (LVRPD=0 or LVDPD=0), when IVC is enabled.*

*Note: F/W must turn off IVC to obtain Tiny Current (IVCPD=1) in Idle/Halt/Stop mode.*

*Note: F/W must turn off IVC (IVCPD=1) before using IAP function.*

## 5. Interrupt & Wake-up

This Chip has a 14-source with 4-level priority interrupt structure. All enabled interrupts can wake up CPU from Idle mode, but only the Pin interrupts (IE0, IE1, IE2, PXIF) and Timer3 (TF3) can wake up CPU from Halt/Stop mode.

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.

| Vector | Flag | Description |
|---|---|---|
| 0003 | IE0 | INT0 external pin interrupt |
| 000B | TF0 | Timer0 interrupt |
| 0013 | IE1 | INT1 external pin interrupt |
| 001B | TF1 | Timer1 interrupt |
| 0023 | RI+TI | UART interrupt |
| 002B | TF2+EXF2 | Timer2 interrupt |
| 0033 | – | Reserved for ICE mode |
| 003B | TF3 | Timer3 interrupt |
| 0043 | PXIF | Port0~Port3 external pin change interrupt |
| 004B | IE2 | INT2 external pin interrupt |
| 0053 | ADIF / TKIF | ADC interrupt / TK interrupt |
| 005B | SPIF / MIIF | SPI interrupt / Master $I^2C$ interrupt |
| 0063 | LVDIF | LVD interrupt |
| 006B | CMPIF | Comparator interrupt |
| 0073 | PWM0IF+PWM1IF | PWM interrupt |

**Interrupt vector description**

| Vector | Item | Interrupt enable | Sub-interrupt enable | Interrupt flag |
|---|---|---|---|---|
| 0003 | IE0 | IE A8.0 | | TCON 88.1 |
| 000B | TF0 | IE A8.1 | | TCON 88.5 |
| 0013 | IE1 | IE A8.2 | | TCON 88.3 |
| 001B | TF1 | IE A8.3 | | TCON 88.7 |
| 0023 | RI+TI | IE A8.4 | | SCON 98.1~0 |
| 002B | TF2+EXF2 | IE A8.5 | | T2CON C8.7~6 |
| 0033 | – | | | |
| 003B | TF3 | INTE1 A9.0 | | INTFLG 95.0 |
| 0043 | PXIF | INTE1 A9.1 | | INTFLG 95.1 |
| 004B | IE2 | INTE1 A9.2 | | INTFLG 95.2 |
| 0053 | ADIF / TKIF | INTE1 A9.3 | INTE2 84.1<br>INTE2 84.0 | INTFLG 95.4<br>INTFLG 95.5 |
| 005B | SPIF / MIIF | INTE1 A9.4 | | SPSTA BD.7<br>MICON E1.5 |
| 0063 | LVDIF | INTE1 A9.5 | | INTFLG 95.7 |
| 006B | CMPIF | INTE1 A9.6 | | INTFLG 95.6 |
| 0073 | PWM0IF+PWM1IF | INTE1 A9.7 | INTE2 84.6<br>INTE2 84.5 | INTFLG2 85.6<br>INTFLG2 85.5 |

**Interrupt related SFRs**

## 5.1 Interrupt Enable and Priority Control

The IE and INTE1 SFRs decide whether the pending interrupt is serviced by CPU. 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.

| 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 (UART) interrupt enable
0: Disable Serial Port (UART) interrupt
1: Enable Serial Port (UART) interrupt

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

A8h.2 **EX1:** External INT1 pin Interrupt enable and Halt/Stop mode wake up enable
0: Disable INT1 pin Interrupt and Halt/Stop mode wake up
1: Enable INT1 pin Interrupt and Halt/Stop mode wake up, it can wake up CPU from Halt/Stop 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 Halt/Stop mode wake up enable
0: Disable INT0 pin Interrupt and Halt/Stop mode wake up
1: Enable INT0 pin Interrupt and Halt/Stop mode wake up, it can wake up CPU from Halt/Stop 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 | CMPIE | LVDIE | SPI2CE | ADTKIE | EX2 | PXIE | 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/PWM1~PWM6 interrupt enable
0: Disable PWM0/PWM1~PWM6 interrupt 1: Enable PWM0/PWM1~PWM6 interrupt

A9h.6 **CMPIE:** CMP interrupt enable
0: Disable CMP interrupt 1: Enable CMP interrupt

A9h.5 **LVDIE:** LVD interrupt enable
0: Disable LVD interrupt 1: Enable LVD interrupt

A9h.4 **SPI2CE:** I²C interrupt enable
0: Disable SPI/I²C interrupt 1: Enable SPI/I²C interrupt

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

A9h.2 **EX2:** External INT2 pin Interrupt enable and Halt/Stop mode wake up enable

0: Disable INT2 pin Interrupt and Halt/Stop mode wake up

1: Enable INT2 pin Interrupt and Halt/Stop mode wake up, it can wake up CPU from Halt/Stop mode no matter EA is 0 or 1.

A9h.1 **PXIE:** Port0~Port3 pin change interrupt enable. This bit does not affect the Port0~Port3 pin's Halt/Stop mode wake up capability.

0: Disable Port0~Port3 pin change interrupt

1: Enable Port0~Port3 pin change interrupt

A9h.0 **TM3IE:** Timer3 interrupt enable

0: Disable Timer3 interrupt

1: Enable Timer3 interrupt

| 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 (UART) 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 | PCMPH | PLVDH | PSPI2CH | PADTKIH | PX2H | PPXH | 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 | PCMP | PLVD | PSPI2C | PADTKI | PX2 | PPX | 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/PWM1 Interrupt Priority control. Definition as above.

BBh.6, BAh.6 **PCMPH, PCMP :** CMP Interrupt Priority control. Definition as above.

BBh.5, BAh.5 **PLVDH, PLVD :** LVD Interrupt Priority control. Definition as above.

BBh.4, BAh.4 **PSPI2CH, PSPI2C :** SPI / I2C Interrupt Priority control. Definition as above.

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

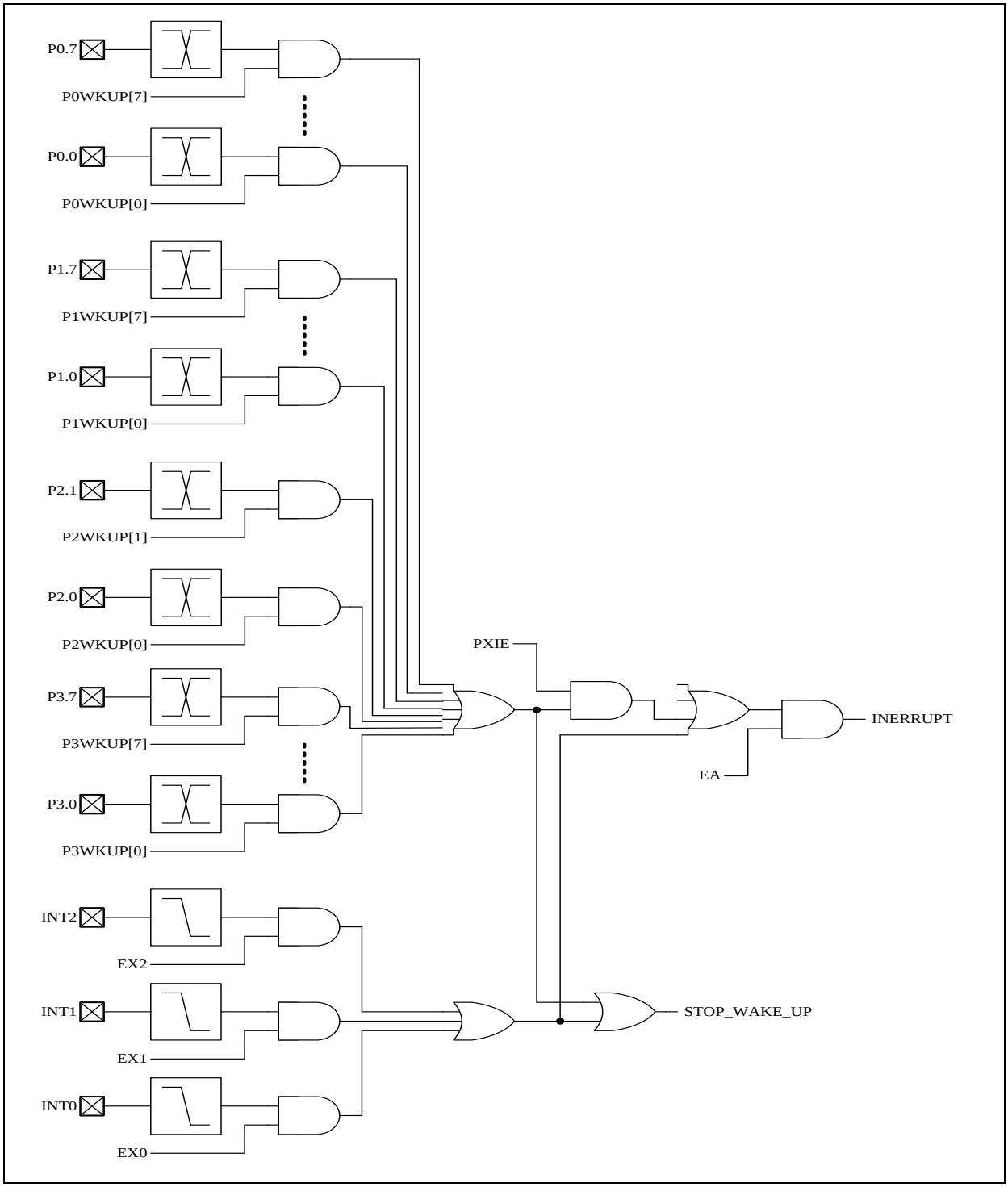
BBh.2, BAh.2 **PX2H, PX2 :** External INT2 pin Interrupt Priority control. Definition as above.

BBh.1, BAh.1 **PPXH, PPX :** Port0~Port3 Pin Change Interrupt Priority control. Definition as above.

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

## 5.2 Pin Interrupt

Pin Interrupts include Change Interrupt. These pins also have the Idle/Halt/Stop mode wake up capability. INT0 and INT1 are falling edge or low level triggered as the 8051 standard. INT2 is falling edge triggered and Port Change Interrupt is triggered by Port pin state change.



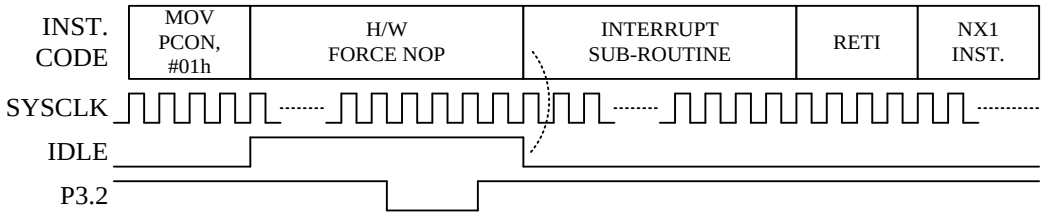

**Pin Interrupt & Wake up**

***Note:*** *The chip cannot enter Halt/Stop Mode if the INTn pin is low and the INTn wake-up function is enabled. (INTn=0 and EXn=1, n=0~2)*

### 5.3 Idle mode Wake up and Interrupt

Each interrupt enable bit (e.g. ET0, EX0) and the EA bit must be set to establish the wake-up function from Idle mode. All enabled interrupts (pins, timers, ADC, touch buttons, SPI and UART) can wake up the CPU from idle mode. When the idle is woken up, immediately enter the interrupt subroutine. When the interrupt subroutine returns, "the first instruction after IDL(PCON.0) is set" will be executed.

For all pin interrupts to be triggered, each interrupt enable bit (e.g. EX0) and the EA bit must be set to 1 and the pin trigger state must stay long enough (greater than 1 system clock) to be sampled by the system clock. When the EA is not set to 1 or the pin trigger state does not stay long enough, it will not wake up and will not generate an interrupt subroutine.




**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 HALT/STOP mode.

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

### 5.4 Halt/Stop mode Wake up and Interrupt

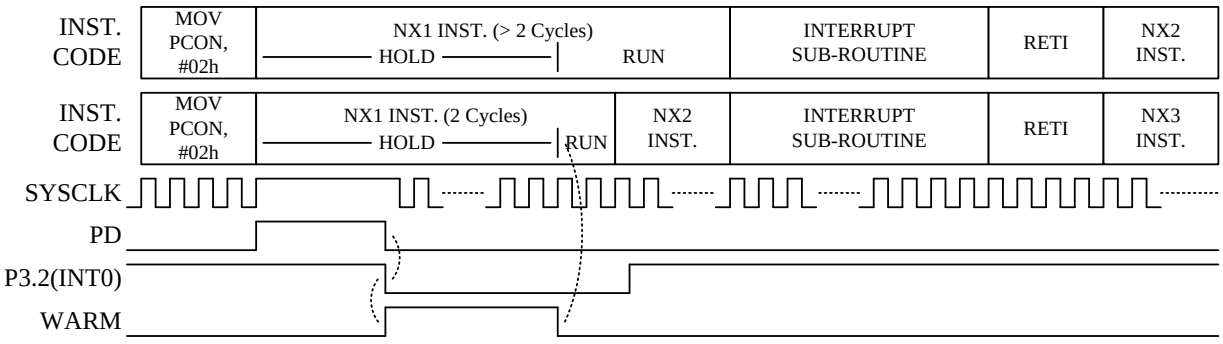
Each interrupt enable bit (e.g. ET3, EX0) and the EA bit must be set to 1 to establish the Halt/Stop mode interrupt function. All enabled interrupts (pins, Timer3) can wake up the CPU from Halt/Stop mode. Once Halt/Stop is woken up, if "the first instruction after PD (PCON.1) is set" is a two-cycle instruction, it will execute immediately before the interrupt is serviced, if "the first instruction after PD (PCON.1) is set" is a four-cycle or more long instruction, it will execute after the interrupt is serviced.

In addition to setting EX0/EX1/EX2, the INT0~2 pin interrupt needs to set EA=1 and the pin trigger state stays long enough (greater than 128 system clocks) to be sampled by the system clock, that is to say, when EA is not set to 1 or if the pin trigger state does not stay long enough, the CPU will only wake up without entering the interrupt subroutine.

In addition to setting P0WKUP/P1WKUP/P2WKUP/P3WKUP, Port0~3 WKUP pin interrupt needs to set EA=1, that is to say, when EA is not set to 1, the CPU will only be woken up and will not enter the interrupt subroutine.

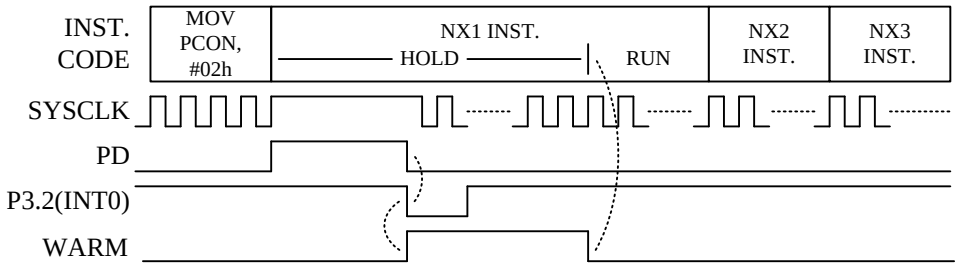
***Note:*** *It is recommended to place the NX1/NX2 with NOP Instruction in figures below.*
***Note:*** *The chip cannot enter Halt/Stop mode if the INTn pin is low and the INTn wake-up function is enabled. (INTn=0 and EXn=1, n=0~2)*

**INT0~2 Pin Interrupt:**




**EA=EX0=1**

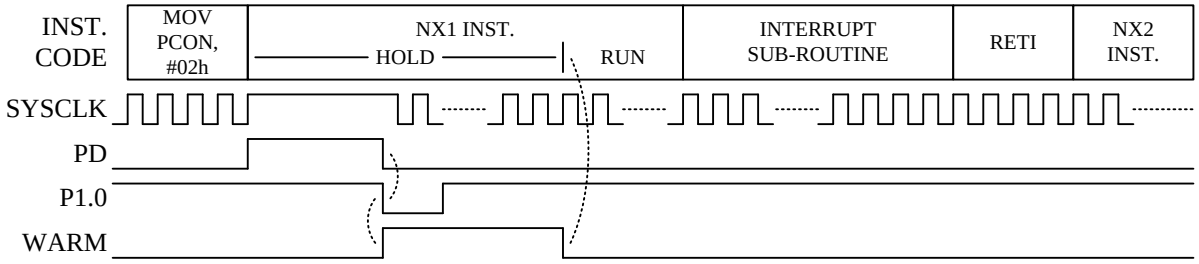
**Input the interrupt pulse whose width is greater than 64 system clocks, then the Idle/Stop mode will wake up and enter the interrupt subroutine**




**EA=EX0= 1**

**Input the interrupt pulse whose width is less than 64 system clocks, then the Idle/Stop mode will wake up but will not enter the interrupt subroutine**

**Port0~3 WKUP Pin Interrupt:**




**EA=PXIE=1, P1WKUP[0]=1**

**Input any width WKUP pin interrupt, the Idle/Stop mode will wake up and enter the interrupt subroutine**

| 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 HALT/STOP mode.

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

| 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 | CMPIF | TKIF | ADIF | – | IE2 | PXIF | TF3 |
| R/W | R/W | R/W | R/W | R/W | – | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | – | 0 | 0 | 0 |

95h.7 **LVDIF:** LVD interrupt flag
Set by H/W when VCC less than the LVD voltage. S/W writes 7Fh to INTFLG to clear this flag.

95h.6 **CMPIF:** CMP interrupt flag
Set by H/W while CMPO match trigger condition. It is cleared automatically when the program performs the interrupt service routine. S/W writes BFh to INTFLG to clear this flag.

95h.5 **TKIF:** Touch Key interrupt flag
Set by H/W at the end of TK conversion. S/W writes DFh to INTFLG or sets the TKSOC bit to clear this flag. When user clears this flag, H/W will automatically clear TKAIF and TKBIF.

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.

95h.2 **IE2:** External Interrupt 2 (INT2 pin) edge flag
Set by H/W when a falling edge is detected on the INT2 pin, no matter the EX2 is 0 or 1.
It is cleared automatically when the program performs the interrupt service routine.
S/W can write FBh to INTFLG to clear this bit.

95h.1 **PXIF:** Port0~Port3 pin change interrupt flag
Set by H/W when Port0~Port3 pin state change is detected and its interrupt enable bit is set (P0WKUP/P1WKUP/P2WKUP/P3WKUP). PXIE 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.

95h.0 **TF3:** Timer3 interrupt flag.
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.

***Note:*** *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 **C5h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **P0WKUP** | P0WKUP | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

C5h.7~0 **P0WKUP:** P0.7~P0.0 pin individual Wake-up / Interrupt enable control
0: Disable
1: Enable

| SFR **C6h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **P2WKUP** | P2WKUP | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

C6h.7~0 **P2WKUP:** P2.7~P2.0 pin individual Wake-up / Interrupt enable control
0: Disable
1: Enable

| SFR **C7h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **P3WKUP** | P3WKUP | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

C7h.7~0 **P3WKUP:** P3.7~P3.0 pin individual Wake-up / Interrupt enable control
0: Disable
1: Enable

| FR **84h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **INTE2** | – | PWM1IE | PWM0IE | – | – | – | TKBIE | TKAIE |
| R/W | – | R/W | R/W | – | – | – | R/W | R/W |
| Reset | – | 0 | 0 | – | – | – | 0 | 0 |

84h.6 **PWM1IE:** PWM1~PWM6 interrupt enable
0: Disable PWM1~PWM6 interrupt
1: Enable PWM1~PWM6 interrupt

84h.5 **PWM0IE:** PWM0 interrupt enable
0: Disable PWM0 interrupt
1: Enable PWM0 interrupt

84h.1 **TKBIE:** Touch Key B interrupt enable
0: Disable Touch Key B interrupt
1: Enable Touch Key B interrupt

84h.0 **TKAIE:** Touch Key A interrupt enable
0: Disable Touch Key A interrupt
1: Enable Touch Key A interrupt

| SFR **85h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **INTFLG2** | – | PWM1IF | PWM0IF | – | – | – | TKBIF | TKAIF |
| R/W | – | R/W | R/W | – | – | – | R/W | R/W |
| Reset | – | 0 | 0 | – | – | – | 0 | 0 |

85h.6 **PWM1IF:** PWM1~PWM6 interrupt flag
Set by H/W at the end of PWM1 period, S/W writes BFh to INTFLG2 to clear this flag.

85h.5 **PWM0IF:** PWM0 interrupt flag
Set by H/W at the end of PWM0 period, S/W writes DFh to INTFLG2 to clear this flag.

85h.1 **TKBIF:** Touch Key B interrupt flag
Set by H/W at the end of Touch Key B scan, S/W writes FDh to INTFLG2 to clear this flag.

85h.0 **TKAIF:** Touch Key A interrupt flag
Set by H/W at the end of Touch Key A scan, S/W writes FEh to INTFLG2 to clear this flag.

## 6. I/O Ports

The Chip has total 26 multi-function I/O pins. There are four Pin modes in this chip, and the functions are shown in the table below. In this table, Port Mode is defined by P0MODL, P0MODH, P1MODL, P1MODH, P2MODL, P3MODL, and P3MODH. Port Data is defined by P0, P1, P2, and P3.

**Port0 & Port1 & P2.1~P2.0 & Port 3**

| Port Mode | Port Data | Description | Output enable | Internal Pull-up Resistor | Digital Input |
|---|---|---|---|---|---|
| Mode0 | 0 | Output Low | Y | N | N |
| | 1 | Input with internal Pull-up resister | N | Y | Y |
| Mode1 | 0 | Output Low | Y | N | N |
| | 1 | Input | N | N | Y |
| Mode2 | 0 | Output Low | Y | N | N |
| | 1 | Output High | Y | N | N |
| Mode3 | X | Analog signal (digital input buffer is disabled) | N | N | N |

*P3.0~P3.2 is Pseudo Open Drain when user select Mode0 or Mode1.

**I/O Pin Function Table**

When user select Mode0 or Mode1, the function is Open drain output low, when Port data=0, the function is output low, when port data=1, the port type is Hi-Z, so user can use digital input in this setting. User can choose mode0 or mode1 for in-out type such as I2C SDA pin. The difference of Mode0 and Mode1 is whether have pull-up resistor or not, when port data = 1, Mode0 have an internal pull-up resister but mode1 haven't, user can add external pull-up resistors by yourself when using Mode1 if you need.

When user select Mode2, the function is CMOS output, user can choose output low or high by port data value. When user select Mode3, the function is for analog signal, such as ADC pin, the port type is Hi-Z and the digital input Schmitt-trigger is disabled in this mode.

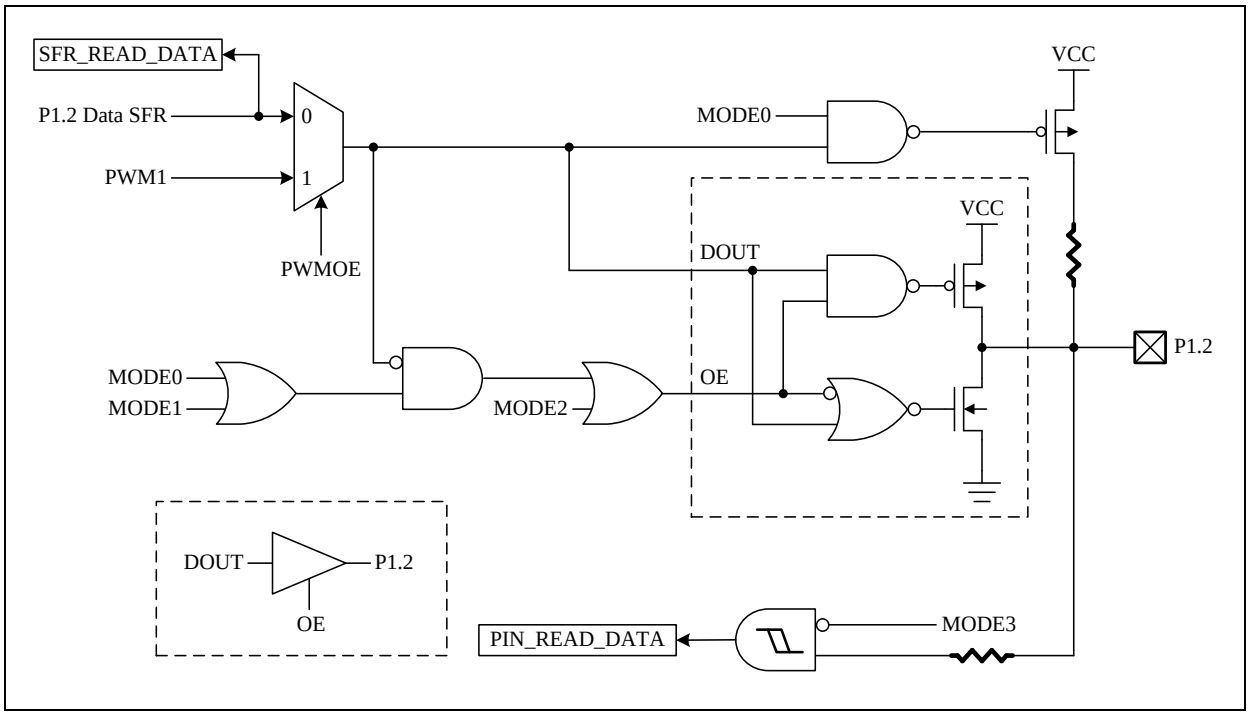
All I/O pins follow the standard 8051 "Read-Modify-Write" feature. If the final output value is related to the value before the command is processed, then this command is the so-called Read-Modify-Write command, such as ANL, INC, CPL. These Read-Modify-Write commands will read Port Data register value instead of the port state value.

| Alternative Function | Port Mode | Port Data | Description | Necessary SFR setting |
|---|---|---|---|---|
| T0, T1, T2, T2EX, INT0, INT1, INT2 | 0 | 1 | Input with Pull-up | - |
| | 1 | 1 | Input | |
| RSTn | x | x | Input with Pull-up | XRSTE |
| TXD | 0 | X | Pseudo Open Drain Output with Pull-up [1] | TXRXSEL |
| | 1 | X | Pseudo Open Drain Output [2] | |
| | 2 | X | CMOS output | |
| RXD | 0 | 1 | Input with Pull-up | TXRXSEL |
| | 1 | 1 | Input | |
| SCL (Master $I^2C$) | 2 | X | CMOS output | MSCLSEL |
| SDA (Master $I^2C$) | 0 | X | Open Drain Output with Pull-up | MSDASEL |
| | 1 | X | Open Drain Output | |
| Master Mode, MISO Slave Mode, SCK, MOSI | 0 | 1 | Input with Pull-up | SPEN MSTR |
| | 1 | 1 | Input | |
| Master Mode, SCK, MOSI Slave Mode, MISO | 2 | X | CMOS Output | |
| T0O, T1O, T2O | 2 | X | CMOS Output | T0OE T1OE T2OE |
| TK0~7 | 2 | 0 | CMOS output low when TK idling, HW auto switch to Mode3 for analog input during TK scanning | TKCHS |
| VBGO | X | X | Analog output | VBGOUT |
| LCDC0~ LCDC3 | X | X | Analog output | P1LOE |
| AD0~11,AD15~21 | 3 | X | Analog input | - |
| CIN1~4 CIP1~4 OPP1~3 | 3 | X | Analog input | - |
| LDOC | X | X | Analog output | LDOCOUT |
| OPOUT | X | X | Analog output | OPOUT |
| DACO | X | X | Analog output | DACOUT |
| CMPO | 2 | X | CMOS Output | CMPOE |
| PWM0~PWM6 | 2 | X | CMOS Output | PWMOE0 PWMOE1 PWMOE2 |

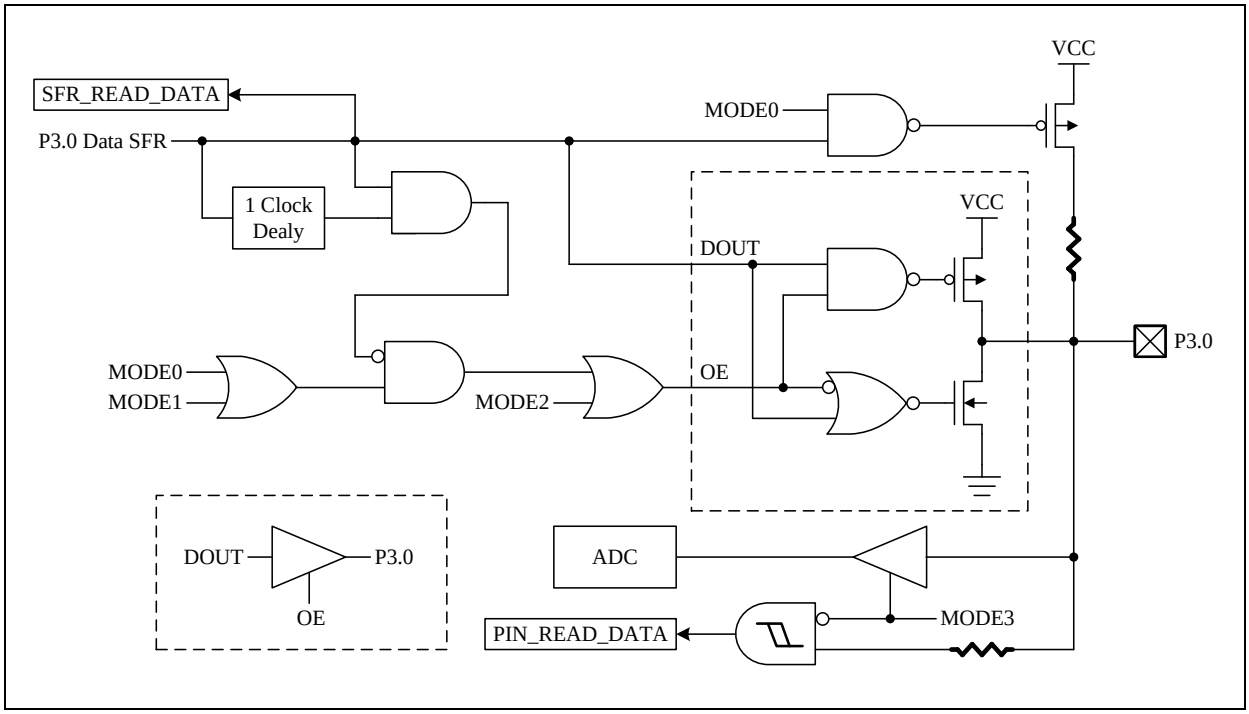
[1] If TXRXSEL=1, RXD and TXD is Open Drain Output with Pull-up

[2] If TXRXSEL=1, RXD and TXD is Open Drain Output

**Alternative Function Table**




**P1.2 Pin Structure**




**P3.0 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 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

80h.7~0 **P0:** Port0 data

| 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 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

A0h.1~0 **P2.1~P2.0:** P2.1~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 **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 | 1 | 1 | 0 |

F7h.4 **VBGOUT:** Bandgap voltage output control
0: Disable
1: Bandgap voltage output to P3.2 pin

| SFR **AFh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **P1LOE** | – | – | – | – | P1LOE3 | P1LOE2 | P1LOE1 | P1LOE0 |
| R/W | – | – | – | – | R/W | R/W | R/W | R/W |
| Reset | – | – | – | – | 0 | 0 | 0 | 0 |

AFh.3 **P1LOE3:** LCD 1/2 bais Output
0: Disable
1: P15 as LCD 1/2 bais Output

AFh.2 **P1LOE2:** LCD 1/2 bais Output
0: Disable
1: P14 as LCD 1/2 bais Output

AFh.1 **P1LOE1:** LCD 1/2 bais Output
0: Disable
1: P13 as LCD 1/2 bais Output

AFh.0 **P1LOE0:** LCD 1/2 bais Output
0: Disable
1: P12 as LCD 1/2 bais Output

| SFR **93h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PINMOD** | TXRXSEL | T2OE | T1OE | T0OE | P2MOD1 | | P2MOD0 | |
| R/W | R/W | R/W | R/W | R/W | R/W | | R/W | |
| Reset | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |

93h.7 **TXRXSEL:** UART TXD/RXD pin select
0: P31 as TXD, P30 as RXD
1: P16 as TXD, P02 as RXD

93h.6 **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

93h.5 **T1OE:** Timer1 signal output (T1O) control
0: Disable "Timer1 overflow divided by 2" output to P3.5 pin
1: Enable "Timer1 overflow divided by 2" output to P3.5 pin

93h.4 **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

93h.3~2 **P2MOD1:** P2.1 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

93h.1~0 **P2MOD0:** P2.0 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

| SFR **91h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **P0MODL** | P0MOD3 | | P0MOD2 | | P0MOD1 | | P0MOD0 | |
| R/W | R/W | | R/W | | R/W | | R/W | |
| Reset | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

A2h.7~6 **P1MOD3:** P0.3 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

A2h.5~4 **P1MOD2:** P0.2 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

A2h.3~2 **P1MOD1:** P0.1 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

A2h.1~0 **P1MOD0:** P0.0 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

| SFR **92h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **P0MODH** | P0MOD7 | | P0MOD6 | | P0MOD5 | | P0MOD4 | |
| R/W | R/W | | R/W | | R/W | | R/W | |
| Reset | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

A3h.7~6 **P0MOD7:** P0.7 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

A3h.5~4 **P0MOD6:** P0.6 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

A3h.3~2 **P0MOD5:** P0.5 pin control.
00: Mode0
01: Mode1
10: Mode2
11: Mode3

A3h.1~0 **P0MOD4:** P0.4 pin control.
00: Mode0
01: Mode1
10: Mode2
11: Mode3

| 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

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

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

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

| 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

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

| 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

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

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

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

| 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 | 1 | 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 **A6h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWMOE0** | PWM1OE3 | PWM1OE2 | PWM1OE1 | PWM1OE0 | PWM0NOE1 | PWM0POE1 | PWM0NOE0 | PWM0POE0 |
| 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 |

A6h.7 **PWM1OE3:** PWM1 output control
0: Disable 1: PWM1 enable and output to P1.2

A6h.6 **PWM1OE2:** PWM1 output control
0: Disable 1: PWM1 enable and output to P0.6

A6h.5 **PWM1OE1:** PWM1 output control
0: Disable 1: PWM1 enable and output to P0.4

A6h.4 **PWM1OE0:** PWM1 output control
0: Disable 1: PWM1 enable and output to P0.2

A6h.3 **PWM0NOE1:** PWM0N output control
0: Disable 1: PWM0N enable and output to P3.6

A6h.2 **PWM0POE1:** PWM0P output control
0: Disable 1: PWM0P enable and output to P3.5

A6h.1 **PWM0NOE0:** PWM0N output control
0: Disable 1: PWM0N enable and output to P0.4

A6h.0 **PWM0POE0:** PWM0P output control
0: Disable 1: PWM0P enable and output to P0.3

| SFR **B6h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWMOE1** | PWM4OE3 | PWM4OE2 | PWM4OE1 | PWM4OE0 | PWM3OE1 | PWM3OE0 | PWM2OE1 | PWM2OE0 |
| 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 |

B6h.7 **PWM4OE3:** PWM4 output control
0: Disable 1: PWM4 enable and output to P3.6

B6h.6 **PWM4OE2:** PWM4 output control
0: Disable 1: PWM4 enable and output to P1.5

B6h.5 **PWM4OE1:** PWM4 output control
0: Disable 1: PWM4 enable and output to P0.4

B6h.4 **PWM4OE0:** PWM4 output control
0: Disable 1: PWM4 enable and output to P0.0

B6h.3 **PWM3OE1:** PWM3 output control
0: Disable 1: PWM3 enable and output to P3.4

B6h.2 **PWM3OE0:** PWM3 output control
0: Disable 1: PWM3 enable and output to P1.0

B6h.1 **PWM2OE1:** PWM2 output control
0: Disable 1: PWM2 enable and output to P3.6

B6h.0 **PWM2OE0:** PWM2 output control
0: Disable 1: PWM2 enable and output to P1.1

| SFR **B7h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWMOE2** | MSDASEL | MSCLSEL | PWM6OE2 | PWM6OE1 | PWM6OE0 | PWM5OE2 | PWM5OE1 | PWM5OE0 |
| 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 |

B7h.7 **MSDASEL:** Master $I^2C$ SDA select
0: P3.5 as Master $I^2C$ SDA
1: P1.6 as Master $I^2C$ SDA

B7h.6 **MSCLSEL:** Master $I^2C$ SCL select

0: P1.3 as Master $I^2C$ SCL
1: P0.2 as Master $I^2C$ SCL

B7h.5 **PWM6OE2:** PWM6 output control
0: Disable 1: PWM6 enable and output to P1.3

B7h.4 **PWM6OE1:** PWM6 output control
0: Disable 1: PWM6 enable and output to P0.7

B7h.3 **PWM6OE0:** PWM6 output control
0: Disable 1: PWM6 enable and output to P0.3

B7h.2 **PWM5OE2:** PWM5 output control
0: Disable 1: PWM5 enable and output to P1.4

B7h.1 **PWM5OE1:** PWM5 output control
0: Disable 1: PWM5 enable and output to P0.6

B7h.0 **PWM5OE0:** PWM5 output control
0: Disable 1: PWM5 enable and output to P0.1

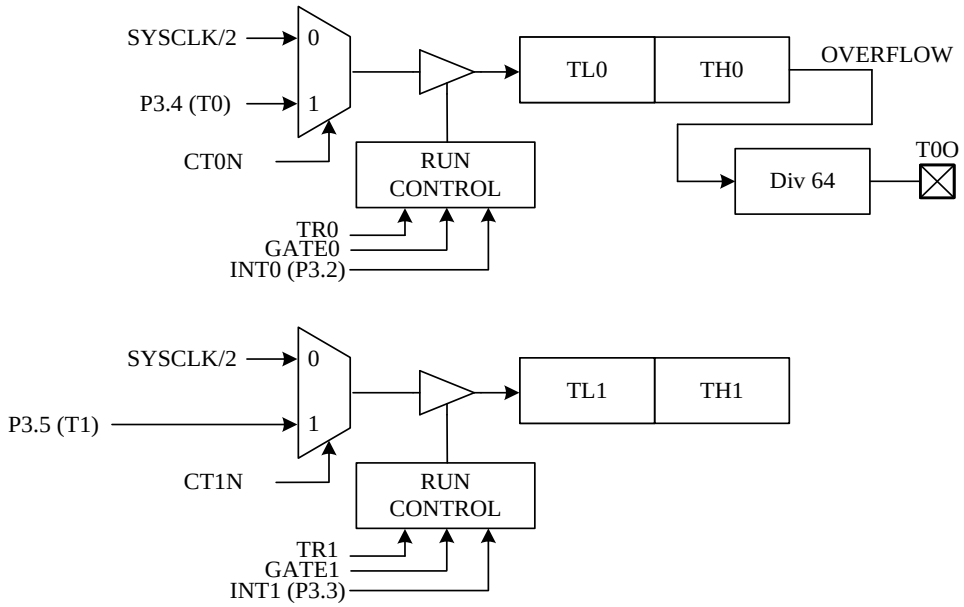
## 7. Timers

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.

This device can generate various frequency waveform pin output for Buzzer. The T0O, T1O, and T2O waveform is divided by Timer0/Timer1/Timer2 overflow signal. The T0O waveform is Timer0 overflow divided by 64, T1O waveform is Timer1 overflow divided by 2, and T2O waveform is Timer2 overflow divided by 2. User can control their frequency by Timers auto reload speed. Set T0OE (93h.6), T1OE (93h.5), and T2OE (93h.4) can output these waveforms.

### 7.1 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: 8-bit timer/counter (TH1) and 5-bit prescaler (TL1)
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: 8-bit timer/counter (TH0) and 5-bit prescaler (TL0)
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

| SFR **93h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PINMOD** | TXRXSEL | T2OE | T1OE | T0OE | P2MOD1 | | P2MOD0 | |
| R/W | R/W | R/W | R/W | R/W | R/W | | R/W | |
| Reset | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |

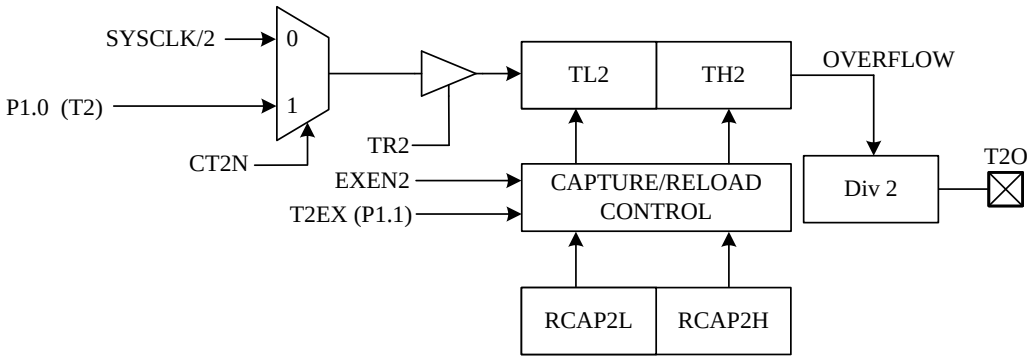
93h.5 **T1OE:** Timer1 signal output (T1O) control
0: Disable "Timer1 overflow divided by 2" output to P3.5 pin
1: Enable "Timer1 overflow divided by 2" output to P3.5 pin

93h.4 **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

***Note**: See also Chapter 5 for more information on Timer0/1 interrupt enable and priority.*
***Note**: See also Chapter 6 for details on T0O pin output settings.*

## 7.2 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 **93h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PINMOD** | TXRXSEL | T2OE | T1OE | T0OE | P2MOD1 | | P2MOD0 | |
| R/W | R/W | R/W | R/W | R/W | R/W | | R/W | |
| Reset | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |

93h.6 **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

***Note**: See also Chapter 5 for more information on Timer2 interrupt enable and priority.*
***Note**: See also Chapter 6 for details on T2O pin output settings.*

### 7.3 Timer3

Timer3 works as a time-base counter, which generates interrupts periodically. Timer3 has 8 kinds of Interrupt period to choose, user can select Timer3 clock source by TM3CKS (94h.6), and this clock source can be divided by TM3PSC (94h.1~0).

| unit: ms | | TM3CKS | |
|---|---|---|---|
| | | 0 | 1 |
| TM3PSC | 00 | 886 | 664 |
| | 01 | 111 | 83 |
| | 10 | 55 | 42 |
| | 11 | 14 | 10 |

**TM3 interrupt period @3V**

| unit: ms | | TM3CKS | |
|---|---|---|---|
| | | 0 | 1 |
| TM3PSC | 00 | 799 | 599 |
| | 01 | 100 | 75 |
| | 10 | 50 | 37 |
| | 11 | 12 | 9 |

**TM3 interrupt period @5V**

| SFR **94h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **OPTION** | UART1W | TM3CKS | WDTPSC | | ADCKS | | TM3PSC | |
| R/W | R/W | R/W | R/W | | R/W | | R/W | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

94h.6 **TM3CKS:** Timer3 clock source select.
0：Slow clock (SRC)
1：SRC/0.75

94h.1~0 **TM3PSC:** Timer3 prescaler.
00: Timer3 is 32768 clock cycle
01: Timer3 is 4096 clock cycle
10: Timer3 is 2048 clock cycle
11: Timer3 is 512 clock cycle

| SFR **95h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **INTFLG** | LVDIF | CMPIF | TKIF | ADIF | – | IE2 | PXIF | TF3 |
| R/W | R/W | R/W | R/W | R/W | – | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 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.

| SFR **F8h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **AUX1** | CLRWDT | CLRTM3 | – | ADSOC | CLRPWM0 | CLRPWM1 | LDOCOUT | DPSEL |
| R/W | R/W | R/W | – | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | – | 0 | 1 | 1 | 0 | 0 |

F8h.6 **CLRTM3:** Set 1 to clear Timer3.

***Note:*** *also refer to Chapter 5 for more information about Timer3 Interrupt enable and priority.*

## 8. UART

The UART 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.

In the 8051 standard, the calculation of the UART baud rate depends on Timer1 or Timer2, but the user can also use the UART dedicated Timer to define a new baud rate by UARTCON.

| 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:** UART double baud rate control bit
0: Disable UART double baud rate
1: Enable UART double baud rate

| SFR **93h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PINMOD** | TXRXSEL | T2OE | T1OE | T0OE | P2MOD1 | | P2MOD0 | |
| R/W | R/W | R/W | R/W | R/W | R/W | | R/W | |
| Reset | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |

93h.7 **TXRXSEL:** UART TXD/RXD pin select
0: P31 as TXD, P30 as RXD
1: P16 as TXD, P02 as RXD

| SFR **94h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **OPTION** | UART1W | TM3CKS | WDTPSC | | ADCKS | | TM3PSC | |
| R/W | R/W | R/W | R/W | | R/W | | R/W | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

94h.7 **UART1W:** One wire UART mode enable, both TXD/RXD use P3.1 or p1.6 pin
0: Disable one wire UART mode
1: Enable one wire UART 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:** Serial port mode select bit 0,1
00: Mode0: 8 bit shift register, Baud Rate=$F_{SYSCLK}$/2
01: Mode1: 8 bit UART, Baud Rate is variable
10: Mode2: 9 bit UART, Baud Rate=$F_{SYSCLK}$/32 or/64
11: Mode3: 9 bit UART, Baud Rate is variable

98h.5 **SM2:** 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:** UART reception enable
0: Disable reception
1: Enable reception

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

98h.2 **RB8:** 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:** 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:** 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:** UART 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 **DFh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **UARTCON** | UARTBRS | UARTBRP | | | | | | |
| 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 |

DFh.7 **UARTBRS:** UART Baud Rate Source Select.
0: 8051 default Baud Rate source select
1: UART Baud Rate select as UARTBRP

DFh.6~0 **UARTBRP:** Define UART Baud Rate Prescaler.
UART Baud Rate = Fsys/32/UARTBRP

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

- **Mode 0:**

  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

  Baud Rate=Timer2 overflow rate/16 = $F_{SYSCLK}$/ (32 x (65536 – RCP2H, RCP2L))

- **Mode 1, 3:** if using UART dedicated Timer (UARTBRS=1)

  Baud Rate= Fsys/32/UARTBRP

- **Mode 2:**

  Baud Rate= (SMOD + 1) x $F_{SYSCLK}$/64

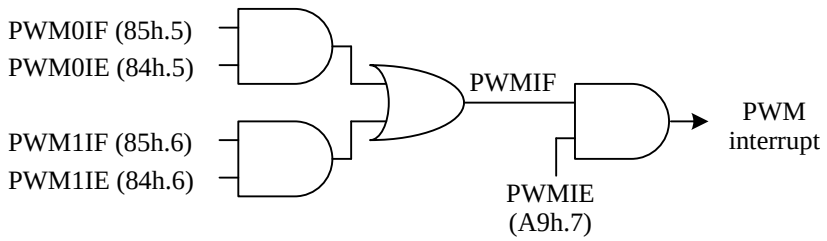
***Note:*** *also refer to Chapter 5 for more information about UART Interrupt enable and priority.*
***Note:*** *also refer to Chapter 7 for more information about how Timer2 controls UART clock.*

## 9. PWMs

This Chip has seven 16-bit PWM modules, PWM0 to PWM6. 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. Users should pay attention to the setting; the period of PWM must be greater than duty.

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.

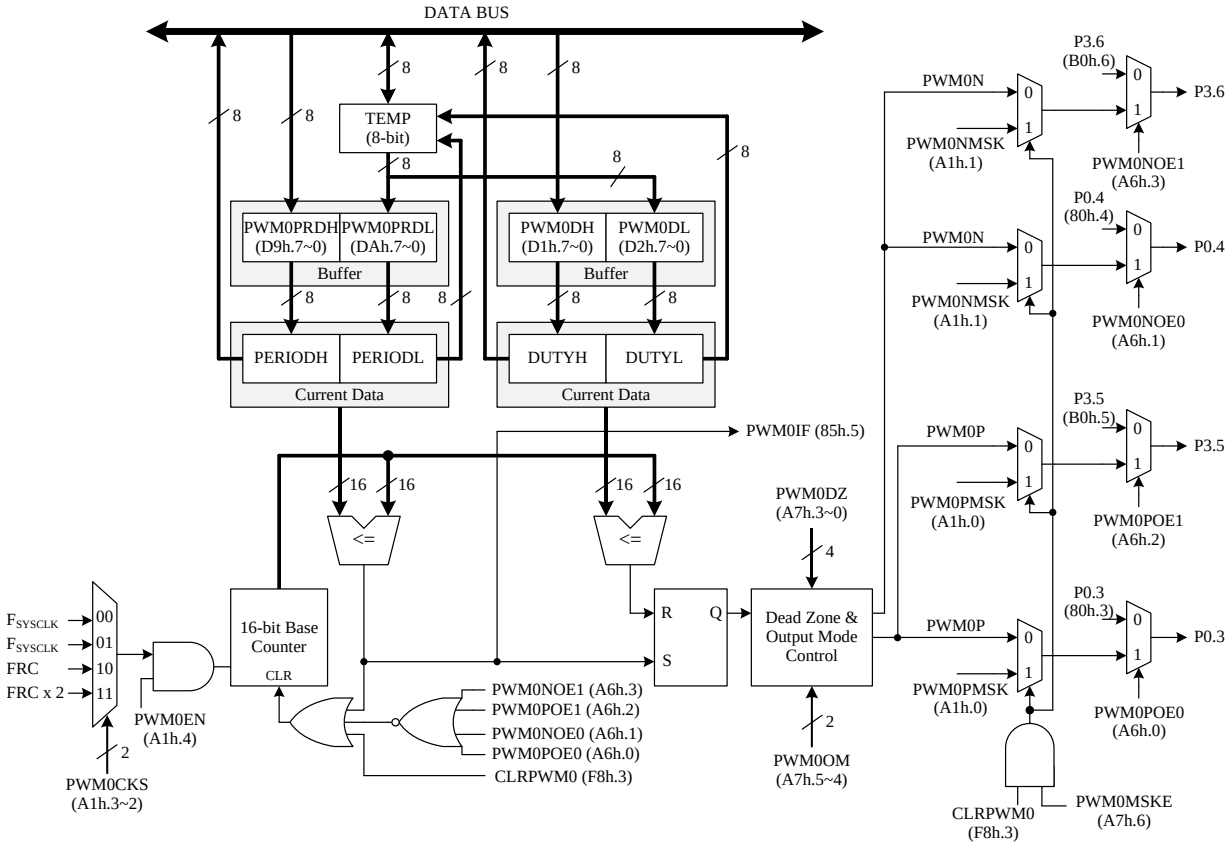
The 16-bit PWM0PRD, PWM1PRD and PWM0D ~ PWM6D registers all have a low and high byte structure. The high bytes can be directly accessed, but as the low bytes can only be accessed via an internal 8-bit buffer, reading or writing to these register pairs must be carried out in a specific way. The important 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.




**PWM interrupt structure**

### 9.1 PWM0

The PWM0POE0 / PWM0POE1 are used to select the output for PWM0P, and the PWM0NOE0 / PWM0NOE1 are used to select the output for PWM0N. These four bits also can be PWM0 control bit. If those four bits are cleared, the PWM0 will be cleared and stopped, otherwise the PWM0 is running. The CLRPWM0 bit has the same function. When CLRPWM0 bit is set, the PWM0 will be cleared and held, otherwise the PWM0 is running. The PWM0 structure is shown as follow.



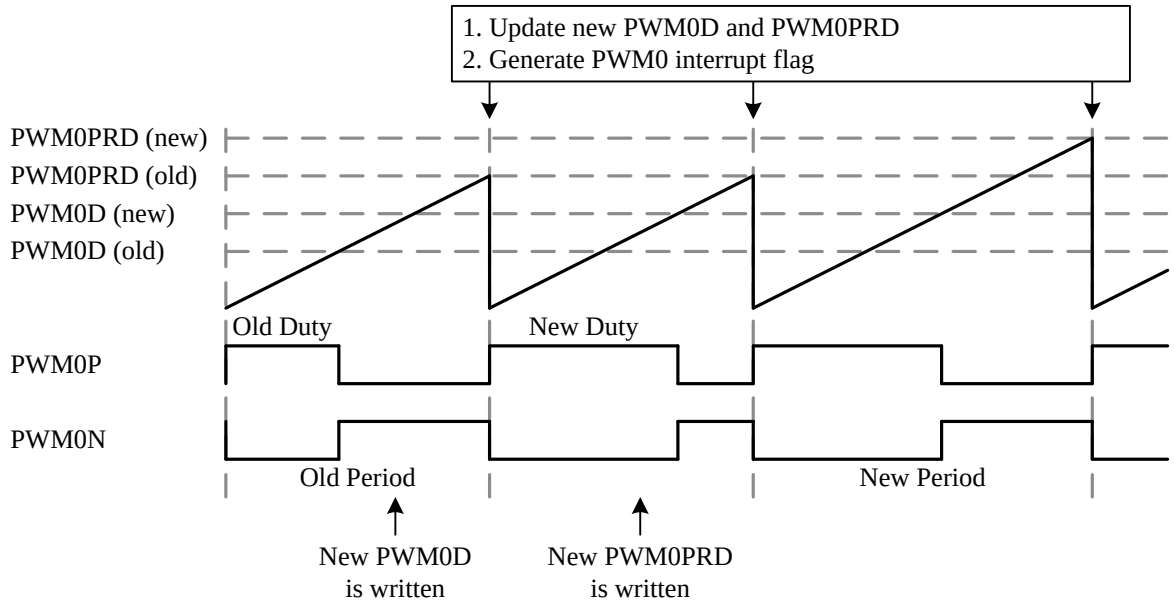

**PWM0 Structure**

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. At the end of current period, H/W will set the PWM0IF bit and generate an interrupt if a PWM0 interrupt is enabled.

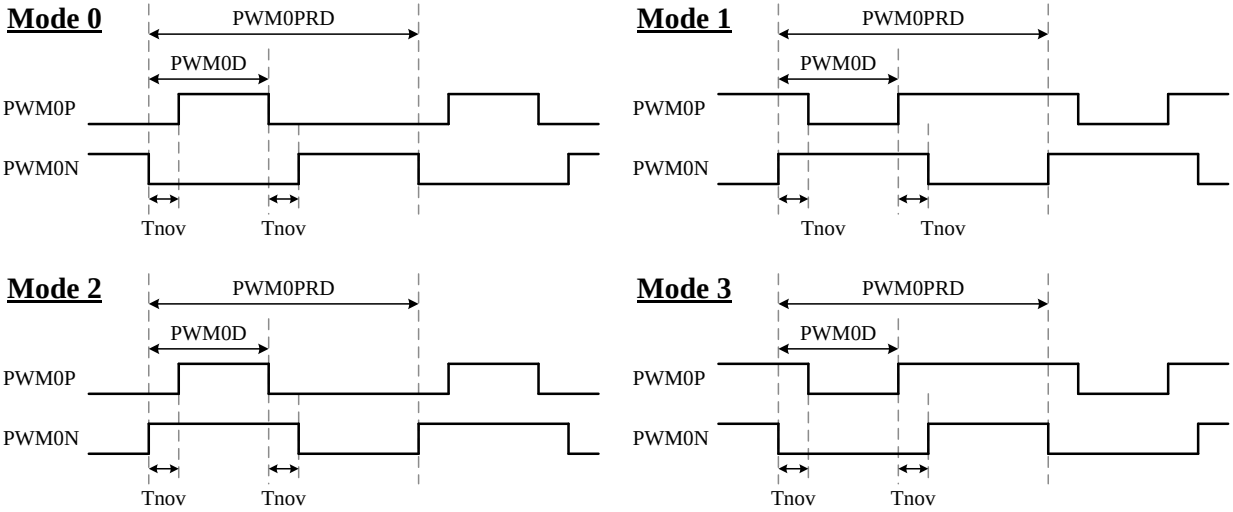
The PWM0 has two operation modes, normal mode and half-bridge mode. PWM0 output signal can be output via PWM0P and PWM0N with four differnet modes. These two outputs are non-overlapped with time interval Tnov. Non-overlapping time interval is also named as dead zone or dead band. Tnov is determined by setting PWM0DZ bits. The value 0~15 of PWM0DZ map onto 0~15, 16 PWM0CLK cycles respectively. If PWM0DZ=0, PWM0 outputs is directly passed to PWM0P and PWM0N so that waveforms of them have the same duty cycle. Note that, if high pulse width or low pulse width of PWM0 output is shorter than Tnov, the real waveforms of these two outputs will different from the expected waveforms. If the PWM0MSKE bit is set, the outputs can be masked to force output fix signal while S/W set the CLRPWM0 bit is set by H/W.

### Normal Mode

The normal mode PWM is a simple structure, which switches its output high and low at uniform repeatable intervals. The PWM0D is the output duty cycle, and the output period is PWM0PRD+1. The output waveform of PWM0 is shown below.



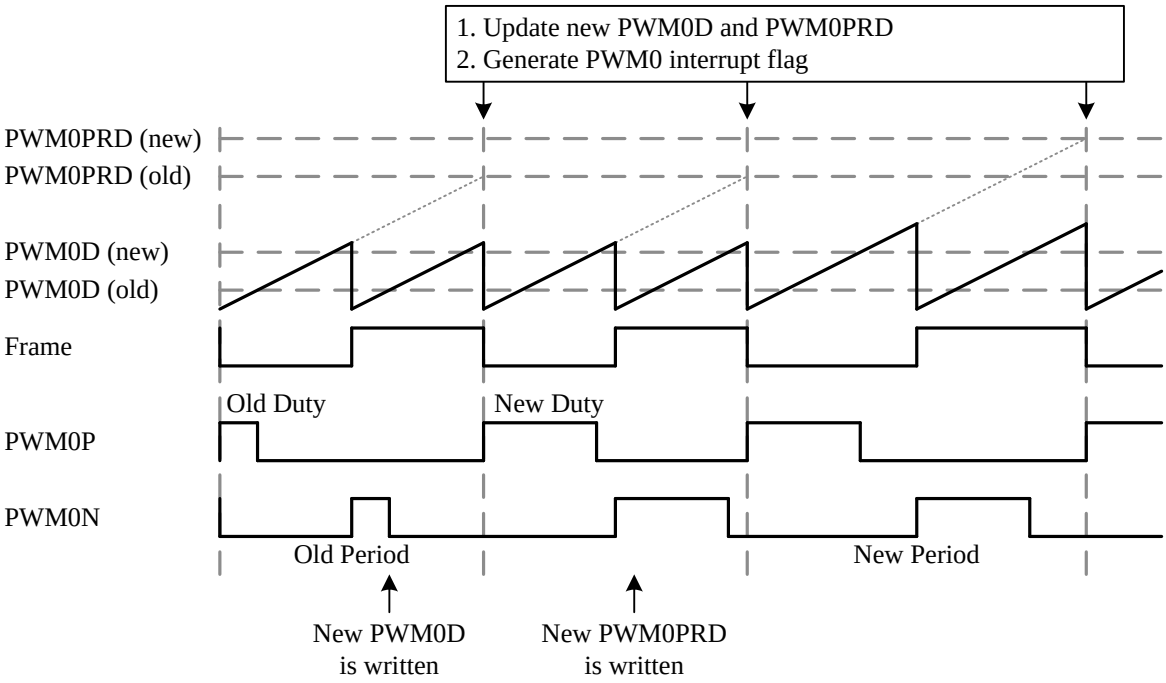

**PWM0 normal mode output waveform (PWM0OM=0, PWM0DZ=0)**



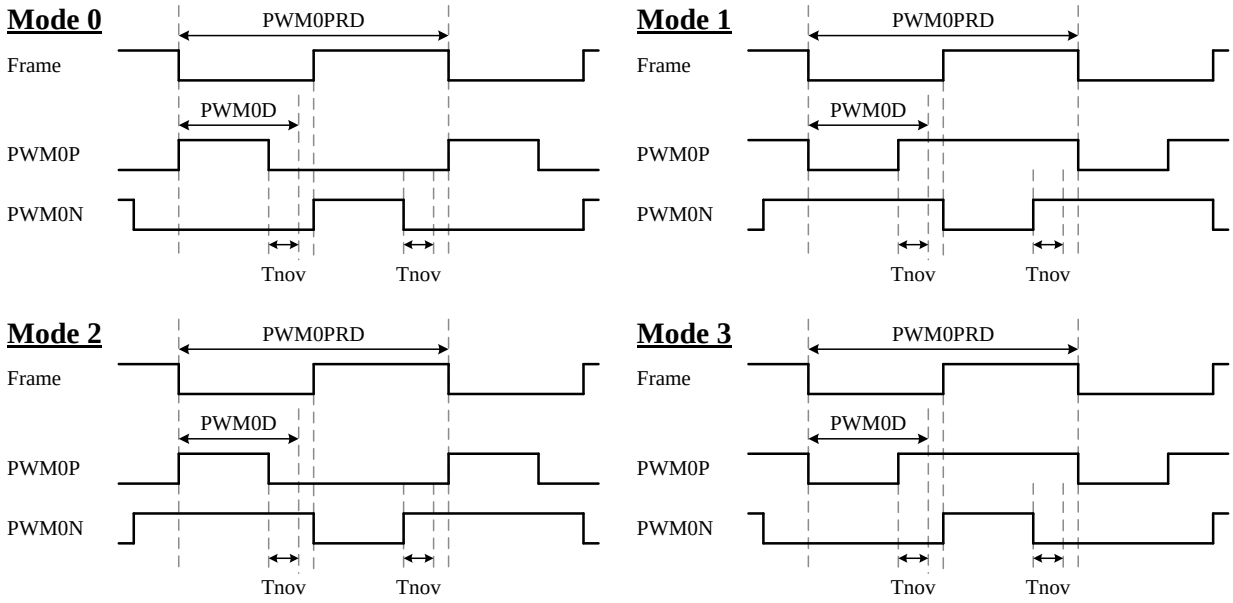

**PWM0 normal mode output modes**

### Half-Bridge Mode

The half-bridge mode PWM is similar to the normal mode but Dead zone is prohibited in half-bridge mode (SFR PWM0DZ must be 0). It has two frames in a period, PWM0P only output in the first frame, PWM0N only output in the second frame. The width of these two frames must be same, so their width is the integer part of PWM0PRD/2. Because each output channel only output in one frame, the maximum duty cycle is same as the width of a frame. If the PWM0D is larger than PWM0PRD/2, H/W will force set the duty cycle to PWM0PRD/2. Following figure shows the output waveform and the output modes.



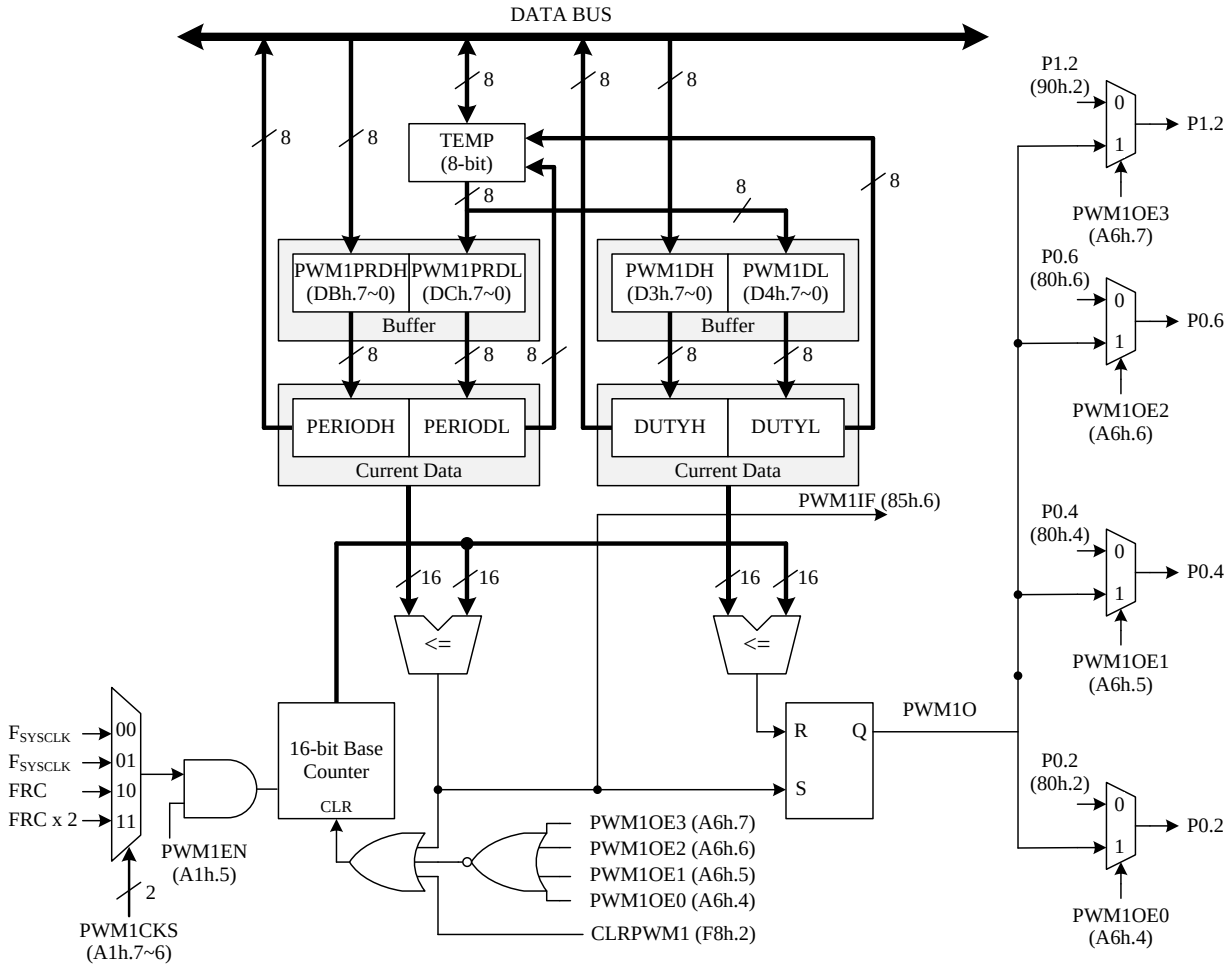

**PWM0 half-bridge mode output waveform (PWM0OM=0, PWM0DZ=0)**




**PWM0 half-bridge mode output modes**

## 9.2 PWM1~PWM6

The Chip has six 16-bit PWM modules PWM1~PWM6. PWM1~6 are sharing period, clock source and interrupt (PWM1IF). The following takes PWM1 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 double frequency (FRC x 2), FRC or $F_{SYSCLK}$ as its clock source.




**PWM1~6 Structure**

| SFR **84h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **INTE2** | – | PWM1IE | PWM0IE | – | – | – | TKBIE | TKAIE |
| R/W | – | R/W | R/W | – | – | – | R/W | R/W |
| Reset | – | 0 | 0 | – | – | – | 0 | 0 |

84h.6 **PWM1IE:** PWM1~PWM6 interrupt enable
0: Disable PWM1~PWM6 interrupt
1: Enable PWM1~PWM6 interrupt

84h.5 **PWM0IE:** PWM0 interrupt enable
0: Disable PWM0 interrupt
1: Enable PWM0 interrupt

| SFR **85h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **INTFLG2** | – | PWM1IF | PWM0IF | – | – | – | TKBIF | TKAIF |
| R/W | – | R/W | R/W | – | – | – | R/W | R/W |
| Reset | – | 0 | 0 | – | – | – | 0 | 0 |

85h.6 **PWM1IF:** PWM1~PWM6 interrupt flag
Set by H/W at the end of PWM1 period, S/W writes BFh to INTFLG2 to clear this flag.

85h.5 **PWM0IF:** PWM0 interrupt enable
Set by H/W at the end of PWM0 period, S/W writes DFh to INTFLG2 to clear this flag.

| SFR **A9h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **INTE1** | PWMIE | CMPIE | LVDIE | SPI2CE | ADTKIE | EX2 | PXIE | 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/PWM1~PWM6 interrupt enable
0: Disable PWM0/PWM1~PWM6 interrupt 1: Enable PWM0/PWM1~PWM6 interrupt

| SFR **A1h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWMCON** | PWM1CKS | | PWM1EN | PWM0EN | PWM0CKS | | PWM0NMSK | PWM0PMSK |
| R/W | R/W | | R/W | R/W | R/W | | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

A1h.7~6 **PWM1CKS:** PWM1~PWM6 clock source
00: $F_{SYSCLK}$
01: $F_{SYSCLK}$
10: FRC
11: FRCx2 (Vcc>2.7V)

A1h.5 **PWM1EN:** PWM1~6 enable
0: PWM1~6 disable
1: PWM1~6 enable

A1h.4 **PWM0EN:** PWM0 enable
0: PWM0 disable
1: PWM0 enable

A1h.3~2 **PWM0CKS:** PWM0 clock source
00: $F_{SYSCLK}$
01: $F_{SYSCLK}$
10: FRC
11: FRCx2 (Vcc>2.7V)

A1h.1 **PWM0NMSK:** PWM0N mask data. If CLRPWM0=1 and PMW0MSKE=1, PWM0N will output this mask data.

A1h.0 **PWM0PMSK:** PWM0P mask data. If CLRPWM0=1 and PMW0MSKE=1, PWM0P will output this mask data.

| SFR **A6h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWMOE0** | PWM1OE3 | PWM1OE2 | PWM1OE1 | PWM1OE0 | PWM0NOE1 | PWM0POE1 | PWM0NOE0 | PWM0POE0 |
| 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 |

A6h.7 **PWM1OE3:** PWM1 output control
0: Disable 1: PWM1 enable and output to P1.2

A6h.6 **PWM1OE2:** PWM1 output control
0: Disable 1: PWM1 enable and output to P0.6

A6h.5 **PWM1OE1:** PWM1 output control
0: Disable 1: PWM1 enable and output to P0.4

A6h.4 **PWM1OE0:** PWM1 output control
0: Disable 1: PWM1 enable and output to P0.2

A6h.3 **PWM0NOE1:** PWM0N output control
0: Disable 1: PWM0N enable and output to P3.6

A6h.2 **PWM0POE1:** PWM0P output control
0: Disable 1: PWM0P enable and output to P3.5

A6h.1 **PWM0NOE0:** PWM0N output control
0: Disable 1: PWM0N enable and output to P0.4

A6h.0 **PWM0POE0:** PWM0P output control
0: Disable 1: PWM0P enable and output to P0.3

| SFR **A7h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWMCON2** | PWM0MOD | PWM0MSKE | PWM0OM | | PWM0DZ | | | |
| R/W | R/W | R/W | R/W | | R/W | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

A7h.7 **PWM0MOD:** PWM0 mode select
0: Normal mode
1: Half-bridge mode

A7h.6 **PWM0MSKE:** PWM0 mask output enable
0: Disable
1: Enable, PWM0P/PWM0N output data by PWM0PMSK/PWM0NMSK while CLRPWM0=1

A7h.5~4 **PWM0OM:** PWM0 output mode select
00: Mode0
01: Mode1
10: Mode2
11: Mode3

A7h.3~0 **PWM0DZ:** PWM0 dead zone (Dead zone is prohibited in half-bridge mode)
0000: 0 x $T_{PWMCLK}$
0001: 1 x $T_{PWMCLK}$
...
1111: 15 x $T_{PWMCLK}$

| SFR **B6h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWMOE1** | PWM4OE3 | PWM4OE2 | PWM4OE1 | PWM4OE0 | PWM3OE1 | PWM3OE0 | PWM2OE1 | PWM2OE0 |
| 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 |

B6h.7 **PWM4OE3:** PWM4 output control
0: Disable 1: PWM4 enable and output to P3.6

B6h.6 **PWM4OE2:** PWM4 output control
0: Disable 1: PWM4 enable and output to P1.5

B6h.5 **PWM4OE1:** PWM4 output control
0: Disable 1: PWM4 enable and output to P0.4

B6h.4 **PWM4OE0:** PWM4 output control
0: Disable 1: PWM4 enable and output to P0.0

B6h.3 **PWM3OE1:** PWM3 output control
0: Disable 1: PWM3 enable and output to P3.4

B6h.2 **PWM3OE0:** PWM3 output control
0: Disable 1: PWM3 enable and output to P1.0

B6h.1 **PWM2OE1:** PWM2 output control
0: Disable 1: PWM2 enable and output to P3.6

B6h.0 **PWM2OE0:** PWM2 output control
0: Disable 1: PWM2 enable and output to P1.1

| SFR **B7h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWMOE2** | MSDASEL | MSCLSEL | PWM6OE2 | PWM6OE1 | PWM6OE0 | PWM5OE2 | PWM5OE1 | PWM5OE0 |
| 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 |

B7h.5 **PWM6OE2:** PWM6 output control
0: Disable 1: PWM6 enable and output to P1.3

B7h.4 **PWM6OE1:** PWM6 output control
0: Disable 1: PWM6 enable and output to P0.7

B7h.3 **PWM6OE0:** PWM6 output control
0: Disable 1: PWM6 enable and output to P0.3

B7h.2 **PWM5OE2:** PWM5 output control
0: Disable 1: PWM5 enable and output to P1.4

B7h.1 **PWM5OE1:** PWM5 output control
0: Disable 1: PWM5 enable and output to P0.6

B7h.0 **PWM5OE0:** PWM5 output control
0: Disable 1: PWM5 enable and output to P0.1

| 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 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

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

| 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: PWMxDL then PWMxDH
read sequence: PWMxDH then PWMxDL

| 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 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

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

| 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: PWMxDL then PWMxDH
read sequence: PWMxDH then PWMxDL

| 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 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

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

| 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: PWMxDL then PWMxDH
read sequence: PWMxDH then PWMxDL

| 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: PWMxPRDL then PWMxPRDH
read sequence: PWMxPRDH then PWMxPRDL

| 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: PWMxPRDL then PWMxPRDH
read sequence: PWMxPRDH then PWMxPRDL

| 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/PWM2/PWM3/PWM4/PWM5/PWM6 period high byte
write sequence: PWMxPRDL then PWMxPRDH
read sequence: PWMxPRDH then PWMxPRDL

| 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/PWM2/PWM3/PWM4/PWM5/PWM6 period low byte
write sequence: PWMxPRDL then PWMxPRDH
read sequence: PWMxPRDH then PWMxPRDL

| SFR **DDh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWM3DH** | PWM3DH | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

DDh.7~0 **PWM3DH:** PWM3 duty high byte
write sequence: PWMxDL then PWMxDH
read sequence: PWMxDH then PWMxDL

| SFR **DEh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWM3DL** | PWM3DL | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

DEh.7~0 **PWM3DL:** PWM3 duty low byte
write sequence: PWMxDL then PWMxDH
read sequence: PWMxDH then PWMxDL

| SFR **E9h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWM4DH** | PWM4DH | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

E9h.7~0 **PWM4DH:** PWM4 duty high byte
write sequence: PWMxDL then PWMxDH
read sequence: PWMxDH then PWMxDL

| SFR **EAh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWM4DL** | PWM4DL | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

EAh.7~0 **PWM4DL:** PWM4 duty low byte
write sequence: PWMxDL then PWMxDH
read sequence: PWMxDH then PWMxDL

| SFR **EBh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWM5DH** | PWM5DH | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

EBh.7~0 **PWM5DH:** PWM5 duty high byte
write sequence: PWMxDL then PWMxDH
read sequence: PWMxDH then PWMxDL

| SFR **ECh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWM5DL** | PWM5DL | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

ECh.7~0 **PWM5DL:** PWM5 duty low byte
write sequence: PWMxDL then PWMxDH
read sequence: PWMxDH then PWMxDL

| SFR **EDh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWM6DH** | PWM6DH | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

EDh.7~0 **PWM6DH:** PWM6 duty high byte
write sequence: PWMxDL then PWMxDH
read sequence: PWMxDH then PWMxDL

| SFR **EEh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWM6DL** | PWM6DL | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

EEh.7~0 **PWM6DL:** PWM6 duty low byte
write sequence: PWMxDL then PWMxDH
read sequence: PWMxDH then PWMxDL

| SFR **F8h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **AUX1** | CLRWDT | CLRTM3 | – | ADSOC | CLRPWM0 | CLRPWM1 | LDOCOUT | DPSEL |
| R/W | R/W | R/W | – | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | – | 0 | 1 | 1 | 0 | 0 |

F8h.3 **CLRPWM0:** PWM0 clear enable
0: PWM0 is running
1: PWM0 is cleared and held

F8h.2 **CLRPWM1:** PWM1/PWM2/PWM3/PWM4/PWM5/PWM6 clear enable
0: PWM1/PWM2/PWM3/PWM4/PWM5/PWM6 is running
1: PWM1/PWM2/PWM3/PWM4/PWM5/PWM6 is cleared and held

## 10. Touch Key

Touch Key provides a simple and reliable method to achieve finger touch detection. In most applications, it does not require any external components. The device supports 8-channel touch key detection.

To use the Touch Key, user must setup the Pin Mode (***see Section 6***) correctly as below table. Touch Key Pin need to set CMOS output Low (Pin Mode2) to reduce the mutual interference between the adjacent keys.

| PxMOD setting for TK0~TK7 | TK PIN |
|---|---|
| Pin is Touch Key, Idling | Drive Low (Pin Mode2) |
| Pin is Touch Key, Scanning | |

There are two oscillators: Reference Clock (RCK) and Touch Clock (TCK). They are connected to the Reference Counter and Data Counter respectively. The frequency of RCK can be adjusted by setting TKxREFC. Reference Counter is used to control conversion time. From starting touch key conversion to end, it will take 0 to 16384 RCK oscillation cycles by setting TKxTMR. After end of conversion, user can get TK data (TKxDH, TKxDL) from Data counter. TK data is affected by finger touching. As finger touching TCK is getting slower, the value of TK data is smaller than the no finger touching. According to the difference of TK data, user can check if it is touched of not. User can choose TK channel by TKxCHS. TK7 is connected a internal built-in reference capacitor, and its capacitance is selected by TKBKCP.

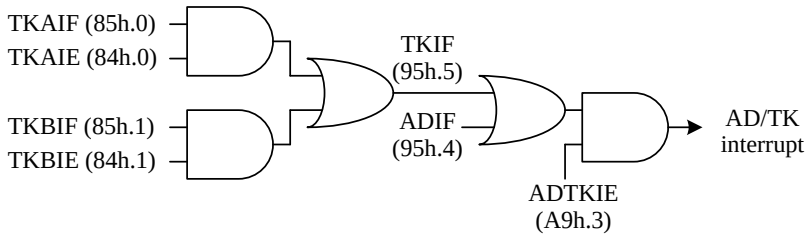
To start the Scanning, user assigns TKxPD=0, then set the TKxSOC bit to start touch key conversion, the TKxSOC bit can be automatically cleared after TKxEOC rising. TKxEOC=0 means conversion is in process, TKxEOC=1 means the conversion is finish, and the touch key counting result is stored into the 14-bit TK Data Counter TKxDH and TKxDL.

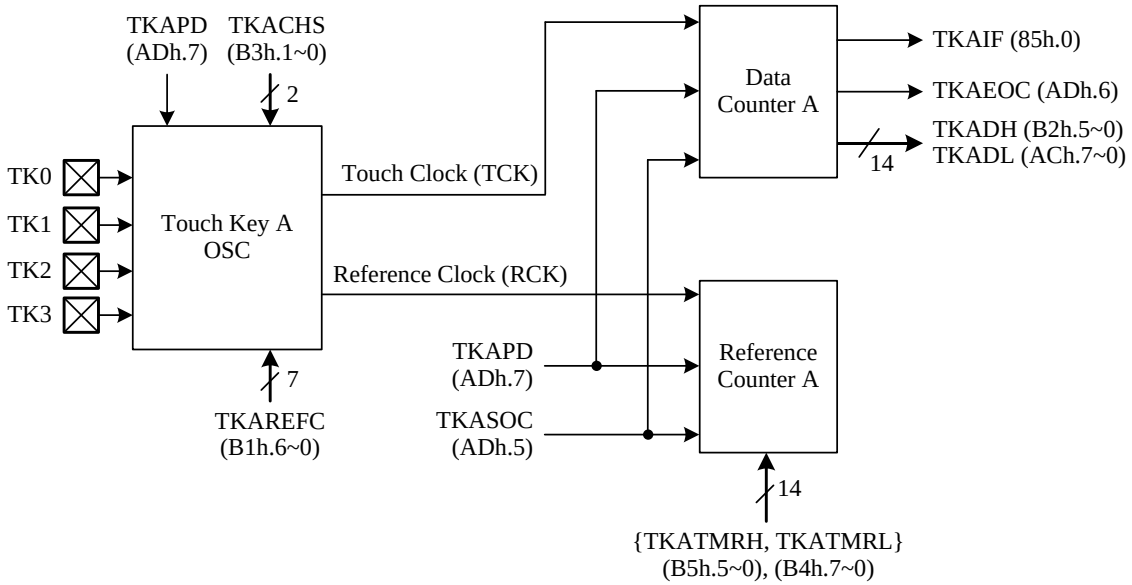
TKIF will active at the first time enable Touch Key function (TKxPD=0), user should clear TKIF after TKxPD cleared.

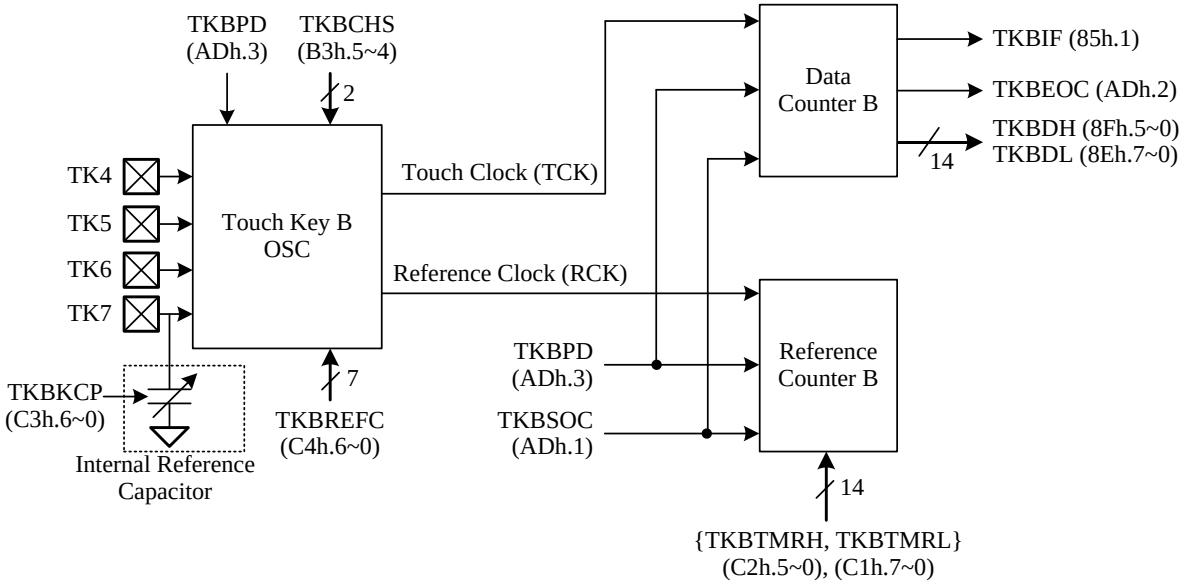
When TKxPD=0, and TKxCHS is set, the Touch Key module is connected to the I/O port through the selection of TKxCHS. If the I/O port is used as other functions, it must be affected. Therefore, when the Touch Key module is not in use, it is recommended to set TKxPD =1 to disconnect the TK module from the I/O port.

Example:

```
MOV     TKCON, #4Ch         ; TKAPD=0, TKBPD=1
                            ;
MOV     TKATMRH, #04h       ;
MOV     TKATMRL, #00h       ; TKATMR=400h
MOV     TKCHS, #33h         ; TKA channel select is TK3
MOV     INTFLG, #DFh        ; Clear TKIF
ORL     INTE1, #08h         ; ADTKIE=1
ORL     IE, #80h            ; EA=1
ORL     TKCON, #20h         ; TKASOC=1
```




**Touch Key Structure**

| FR **84h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **INTE2** | – | PWM1IE | PWM0IE | – | – | – | TKBIE | TKAIE |
| R/W | – | R/W | R/W | – | – | – | R/W | R/W |
| Reset | – | 0 | 0 | – | – | – | 0 | 0 |

84h.1 **TKBIE:** Touch Key B interrupt enable
0: Disable Touch Key B interrupt
1: Enable Touch Key B interrupt

84h.0 **TKAIE:** Touch Key A interrupt enable
0: Disable Touch Key A interrupt
1: Enable Touch Key A interrupt

| SFR **85h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **INTFLG2** | – | PWM1IF | PWM0IF | – | – | – | TKBIF | TKAIF |
| R/W | – | R/W | R/W | – | – | – | R/W | R/W |
| Reset | – | 0 | 0 | – | – | – | 0 | 0 |

85h.1 **TKBIF:** Touch Key B interrupt flag
Set by H/W at the end of Touch Key B scan, S/W writes FDh to INTFLG2 to clear this flag.

85h.0 **TKAIF:** Touch Key A interrupt flag
Set by H/W at the end of Touch Key A scan, S/W writes FEh to INTFLG2 to clear this flag.

| SFR **95h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **INTFLG** | LVD | CMPIF | TKIF | ADIF | – | IE2 | PXIF | TF3 |
| R/W | R/W | R/W | R/W | R/W | – | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | – | 0 | 0 | 0 |

95h.5 **TKIF:** Touch Key Interrupt Flag
Set by H/W at the end of Touch Key conversion. S/W writes DFh to INTFLG or sets the TKSOC bit to clear this flag. When user clears this flag, H/W will automatically clear TKAIF and TKBIF.

| SFR **A9h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **INTE1** | PWMIE | CMPIE | LVDIE | SPI2CE | ADTKIE | EX2 | PXIE | 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.3 **ADTKIE:** ADC interrupt enable
0: Disable ADC/TK interrupt 1: Enable ADC/TK interrupt

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

8Eh.7~0 **TKBDL:** Touch Key B data low byte

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

8Fh.5~0 **TKBDH:** Touch Key B data high byte

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

ACh.7~0 **TKADL:** Touch Key A data low byte

| SFR **ADh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TKCON** | TKAPD | TKAEOC | TKASOC | – | TKBPD | TKBEOC | TKBSOC | – |
| R/W | R/W | R/W | R/W | – | R/W | R/W | R/W | – |
| Reset | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |

ADh.7 **TKAPD:** Touch Key A power down.

ADh.6 **TKAEOC:** Touch Key A end of conversion.

ADh.5 **TKASOC:** Touch Key A start, HW clear while end of conversion.

ADh.3 **TKBPD:** Touch Key B power down.

ADh.2 **TKBEOC:** Touch Key B end of conversion.

ADh.1 **TKBSOC:** Touch Key B start, HW clear while end of conversion.

| SFR **B1h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TKAREFC** | – | TKAREFC | | | | | | |
| R/W | – | R/W | | | | | | |
| Reset | – | – | – | – | – | – | – | – |

B1h.6~0 **TKAREFC:** Touch Key A reference clock capacitor select

| SFR **B2h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TKADH** | – | –- | TKADH | | | | | |
| R/W | – | – | R | | | | | |
| Reset | – | – | – | – | – | – | – | – |

B2h.5~0 **TKADH:** Touch Key A data high byte

| SFR **B3h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TKCHS** | – | – | TKBCHS | | – | – | TKACHS | |
| R/W | R/W | R/W | R/W | | R/W | R/W | R/W | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

B3h.5~4 **TKBCHS:** Touch Key B channel select
00: TK4 (P1.1)
01: TK5 (P0.5)
10: TK6 (P0.6)
11: TK7 (P0.7) (Ref)

B3h.1~0 **TKACHS:** Touch Key A channel select
00: TK0 (P3.7)
01: TK1 (P2.1)
10: TK2 (P2.0)
11: TK3 (P1.2)

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

B4h.7~0 **TKATMRL:** Touch Key A reference counter data 7~0

| SFR **B5h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TKATMRH** | – | – | TKATMRH | | | | | |
| R/W | – | – | R/W | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

B5h.5~0 **TKATMRH:** Touch Key A reference counter data 13~8

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

C1h.7~0 **TKBTMRL:** Touch Key B reference counter data 7~0

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

C2h.5~0 **TKBTMRH:** Touch Key B reference counter data 13~8

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

C3h.6~0 **TKBKCP:** Touch Key B reference capacitor select (TK7)

| SFR **C4h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TKBREFC** | – | TKBREFC | | | | | | |
| R/W | – | R/W | | | | | | |
| Reset | – | – | – | – | – | – | – | – |

C4h.6~0 **TKBREFC:** Touch Key B reference clock capacitor select

## 11. Low Voltage Detection (LVD)

The chip also provides a low voltage detection (LVD) function, and the SFR LVDSEL can select 15 LVDs with different voltage thresholds.

| LVDPD (SFR E4h.4) | Operation Mode | PWRSAV (SFR F7.5) | LVDSEL (SFR E4h.3~0) | Function |
|---|---|---|---|---|
| 1 | X | X | X | LVD disable |
| 0 | Fast/Slow | X | 0000 | LVD 2.05V |
| | | | 0001 | LVD 2.19V |
| | | | 0010 | LVD 2.33V |
| | | | 0011 | LVD 2.47V |
| | | | 0100 | LVD 2.61V |
| | | | 0101 | LVD 2.75V |
| | | | 0110 | LVD 2.89V |
| | | | 0111 | LVD 3.03V |
| | | | 1000 | LVD 3.17V |
| | | | 1001 | LVD 3.31V |
| | | | 1010 | LVD 3.45V |
| | | | 1011 | LVD 3.59V |
| | | | 1100 | LVD 3.73V |
| | | | 1101 | LVD 3.87V |
| | | | 1110 | LVD 4.01V |
| | | | 1111 | LVD 4.15V |
| | Idle/Halt/Stop | 0 | 0000 | LVD 2.05V |
| | | | 0001 | LVD 2.19V |
| | | | 0010 | LVD 2.33V |
| | | | 0011 | LVD 2.47V |
| | | | 0100 | LVD 2.61V |
| | | | 0101 | LVD 2.75V |
| | | | 0110 | LVD 2.89V |
| | | | 0111 | LVD 3.03V |
| | | | 1000 | LVD 3.17V |
| | | | 1001 | LVD 3.31V |
| | | | 1010 | LVD 3.45V |
| | | | 1011 | LVD 3.59V |
| | | | 1100 | LVD 3.73V |
| | | | 1101 | LVD 3.87V |
| | | | 1110 | LVD 4.01V |
| | | | 1111 | LVD 4.15V |
| | Idle/Halt/Stop | 1 | xxxx | LVD disable |

**Low voltage detect table**

| SFR **E4h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **LVDCON** | LVDM | LVDO | LVDHYS | LVDPD | LVDSEL | | | |
| R/W | R/W | R | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

E4h.7 **LVDM:**
0: VCC < VLVD (LVDIF = 1 while LVDO = 1)
1: VCC > VLVD (LVDIF = 1 while LVDO = 0)

E4h.6 **LVDO:** LVD real-time Output

E4h.5 **LVDHYS:** LVD Hysteresis Enable.
0: LVD Hysteresis disable
1: LVD Hysteresis enable

E4h.4 **LVDPD:** LVD Power Down.
0: LVD Enable
1: LVD Disable

E4h.3~0 **LVDSEL:** Low Voltage Detect (LVD) select. (step=0.14V)
0000: Set LVD at 2.05V
0001: Set LVD at 2.19V
0010: Set LVD at 2.33V
0011: Set LVD at 2.47V
0100: Set LVD at 2.61V
0101: Set LVD at 2.75V
0110: Set LVD at 2.89V
0111: Set LVD at 3.03V
1000: Set LVD at 3.17V
1001: Set LVD at 3.31V
1010: Set LVD at 3.45V
1011: Set LVD at 3.59V
1100: Set LVD at 3.73V
1101: Set LVD at 3.87V
1110: Set LVD at 4.01V
1111: Set LVD at 4.15V

| 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 | 1 | 1 | 0 |

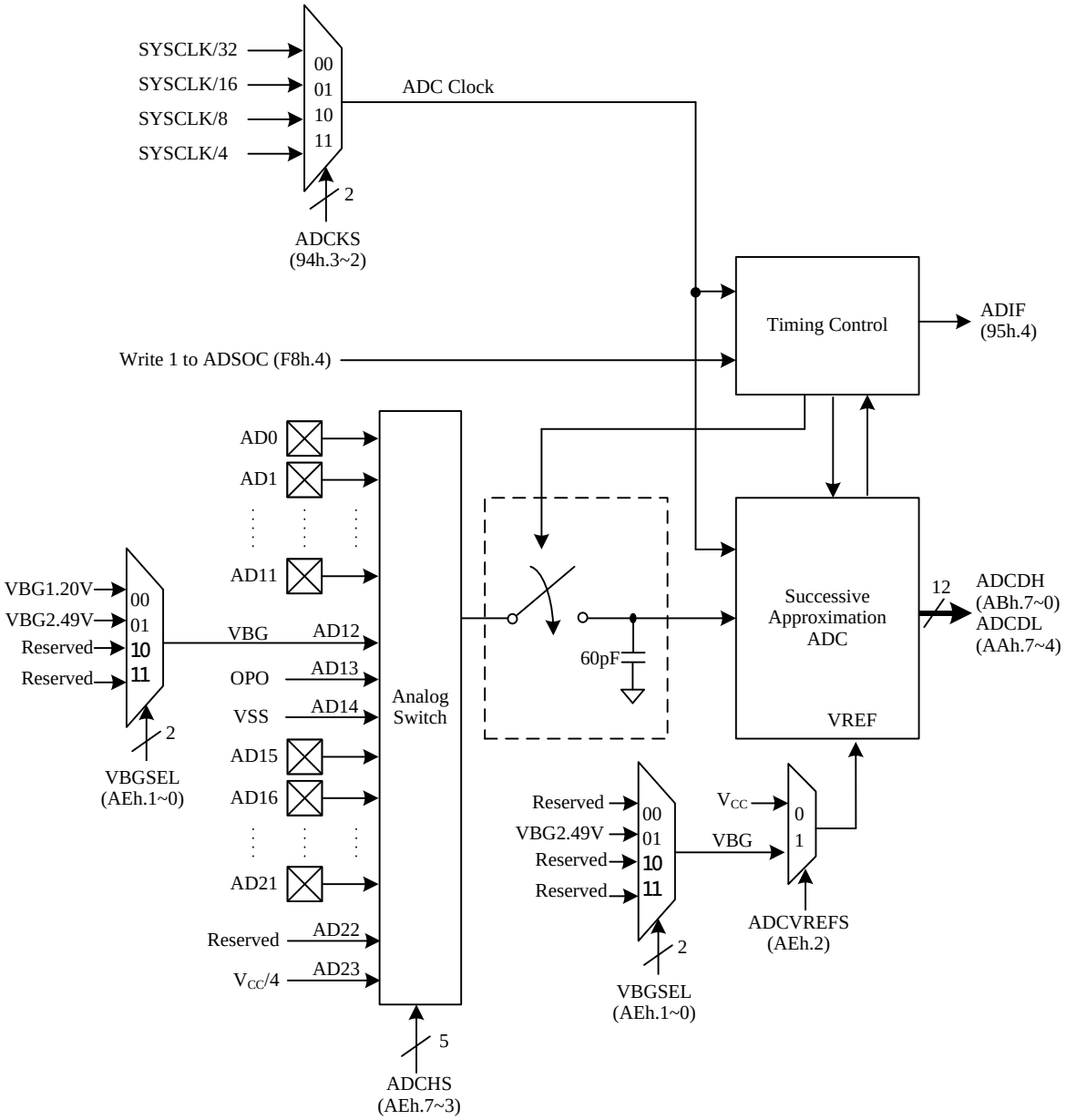
F7h.5 **PWRSAV:** Set 1 to reduce the chip's power consumption at Idle/Halt/Stop Mode

## 12. ADC

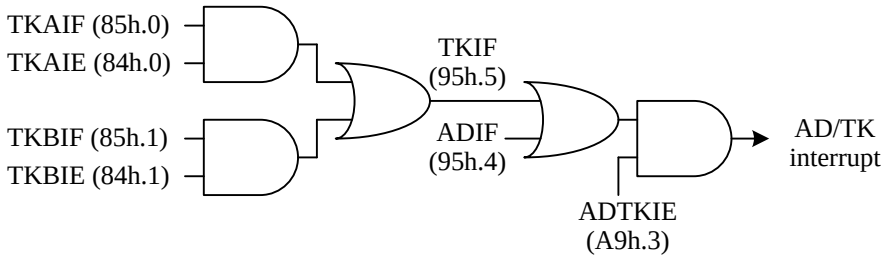
The Chip offers a 12-bit ADC consisting of a 19-channel analog input multiplexer, control register, clock generator, 12-bit successive approximation register, and output data register. Generally, ADC clock frequency is less than 1 MHz, user can refer to Electrical Characteristics Chapter for detail.

To use the ADC, set the ADCKS bits first to choose a proper ADC clock frequency. Then, user 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 set ADSOC bit. The analog input level must remain within the range from $V_{SS}$ to $V_{CC}$.

Using the ADCVREFS option, the ADC internal reference voltage source (VREF) can be selected as $V_{CC}$ or VBG 2.49V. When ADCVREFS=1, VBGSEL must be set to 2'b01.



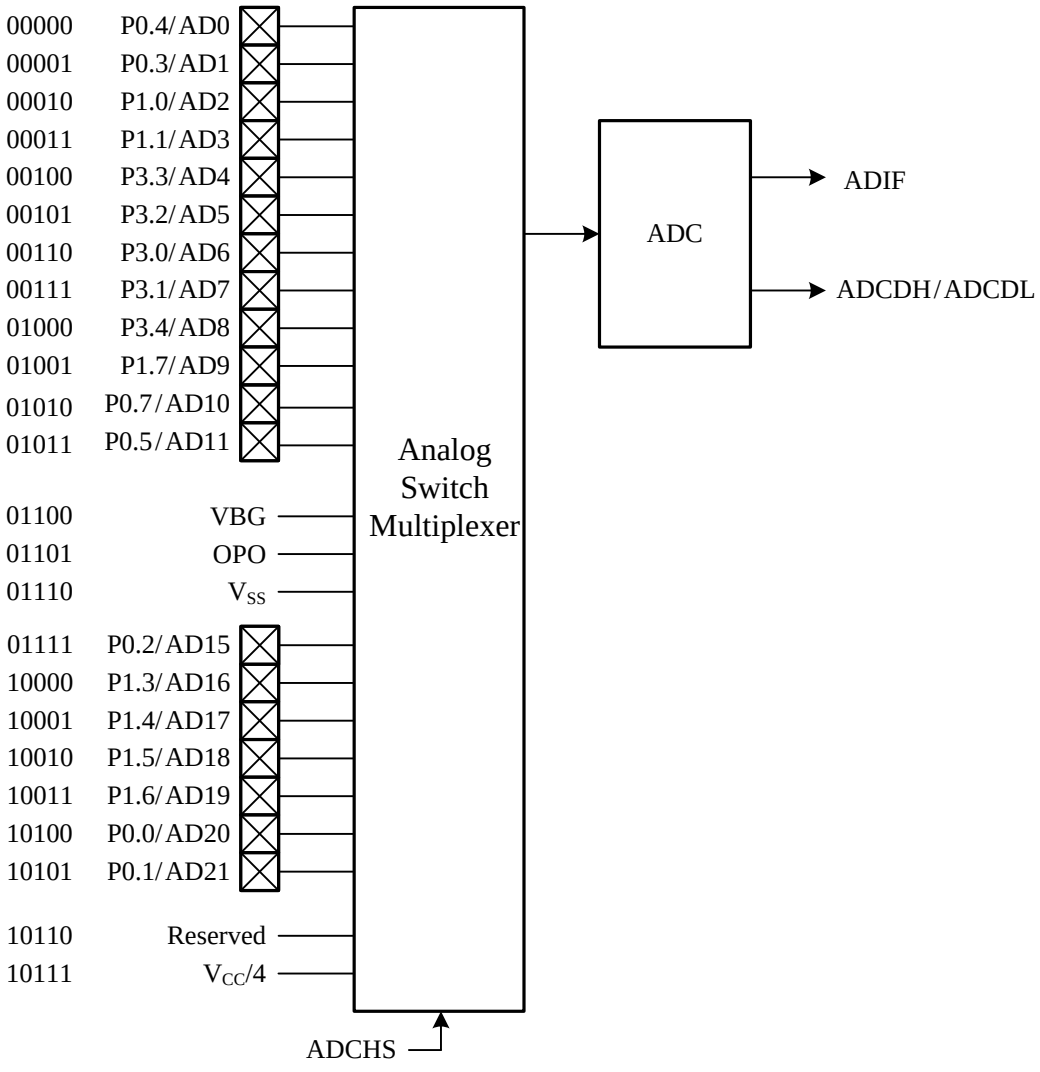

**ADC Structure**




**ADC Interrupt Structure**

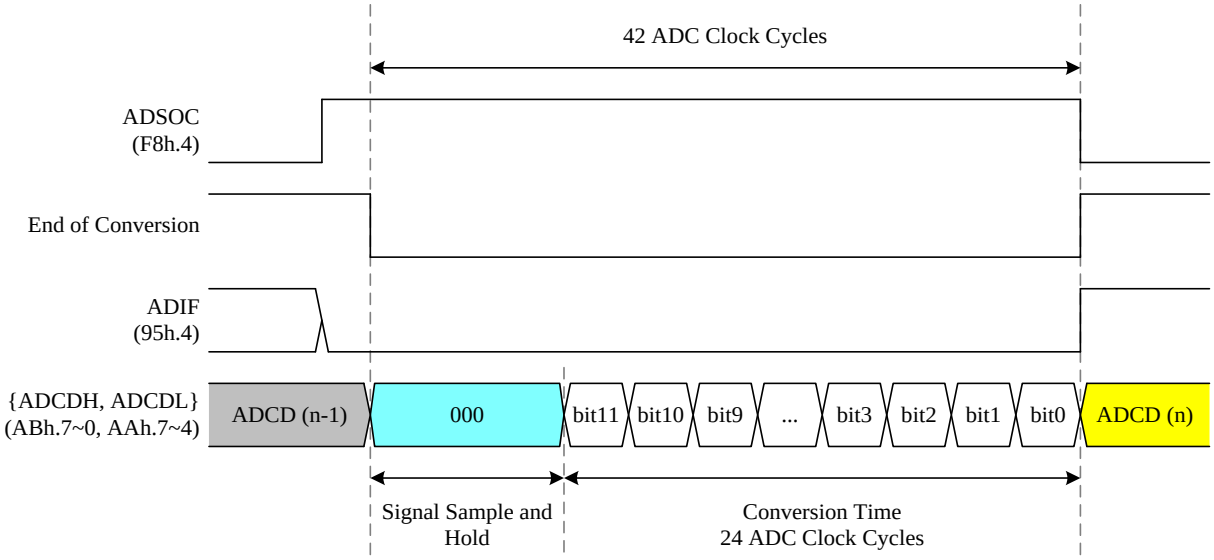
### ADC Channels

The ADC channels are connected to the analog input pins via the analog switch multiplexer. The analog switch multiplexer is controlled by ADCHS register. The Chip offers up to 19 IO input pins, designated AD0~AD11, AD15~AD21. In addition, there are 3 internal reference voltages (VBG, VSS, $V_{CC}/4$). When ADCHS is set to 1100b, the analog input will connect to VBG, and when ADCHS is set to 1101b, the analog input will connect to OPO, generated from internal operational amplifier.



### 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 42 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 | TM3CKS | WDTPSC | | ADCKS | | TM3PSC | |
| R/W | R/W | R/W | R/W | | R/W | | R/W | |
| Reset | 0 | 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 | CMPIF | TKIF | ADIF | – | IE2 | PXIF | TF3 |
| R/W | R/W | R/W | R/W | R/W | – | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 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.

| 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 | VBGSEL | |
| R/W | R/W | | | | | R/W | R/W | |
| Reset | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |

AEh.7~3 **ADCHS:** ADC channel select

00000: ADC0 (P04)
00001: ADC1 (P03)
00010: ADC2 (P10)
00011: ADC3 (P11)
00100: ADC4 (P33)
00101: ADC5 (P32)
00110: ADC6 (P30)
00111: ADC7 (P31)
01000: ADC8 (P34)
01001: ADC9 (P17)
01010: ADC10 (P07)
01011: ADC11 (P05)
01100: VBGO
01101: OPO
01110: VSS
01111: ADC15 (P02)
10000: ADC16 (P13)
10001: ADC17 (P14)
10010: ADC18 (P15)
10011: ADC19 (P16)
10100: ADC20 (P00)
10101: ADC21 (P01)
10110: Reserved
10111: VCC/4
other: Reserved

AEh.2 **ADCVREFS:** ADC reference voltage select

0: $V_{CC}$
1: VBG

AEh.1~0 **VBGSEL:** VBG voltage select

When ADCVREF is selected as VBG, VBGSEL is prohibited from using 1.20V.

00: 1.20V
01: 2.49V (need $V_{CC}$>2.8V)
10: Reserved
11: Reserved

| SFR **F8h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **AUX1** | CLRWDT | CLRTM3 | – | ADSOC | CLRPWM0 | CLRPWM1 | LDOCOUT | DPSEL |
| R/W | R/W | R/W | – | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | – | 0 | 1 | 1 | 0 | 0 |

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

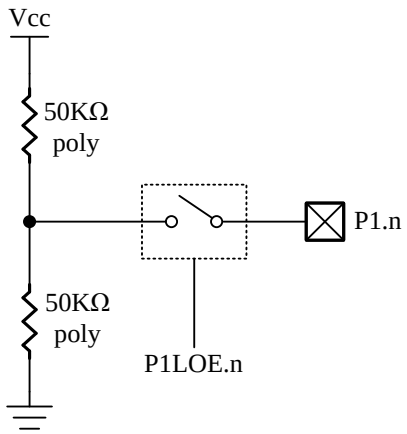
Set the ADSOC bit to start ADC conversion, and the ADSOC bit will be cleared by H/W at the end of conversion. S/W can also write 0 to clear this flag.

*Note: See also Chapter 5 for more information on ADC interrupt enable and priority.*

*Note: Also refer to Chapter 6 for details on ADC pin input settings.*

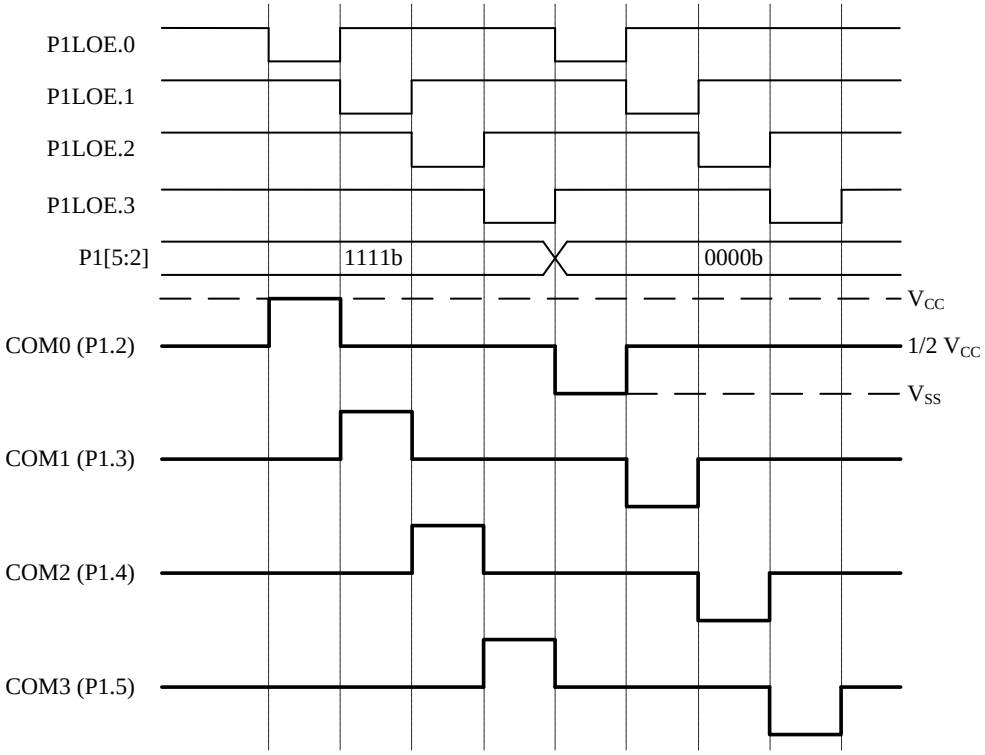
## 13. S/W Controller LCD Driver

The chip supports an S/W controlled method to driving LCD. It is capable of driving the LCD panel with 88 dots (Max.) by 4 Commons (COM) and 22 Segments (SEG). The P1.2~P1.5 are used for Common pins COM0~COM3 and others pins can be used for Segment pins. COM0~COM3 are capable of driving 1/2 bias when P1.2~P1.5's P1LOE=1. Refer to the following figures.



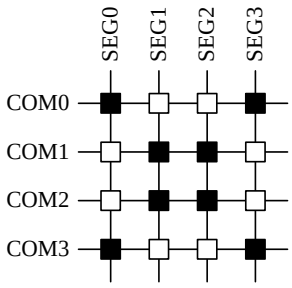

**LCD COM0~3 Circuit**

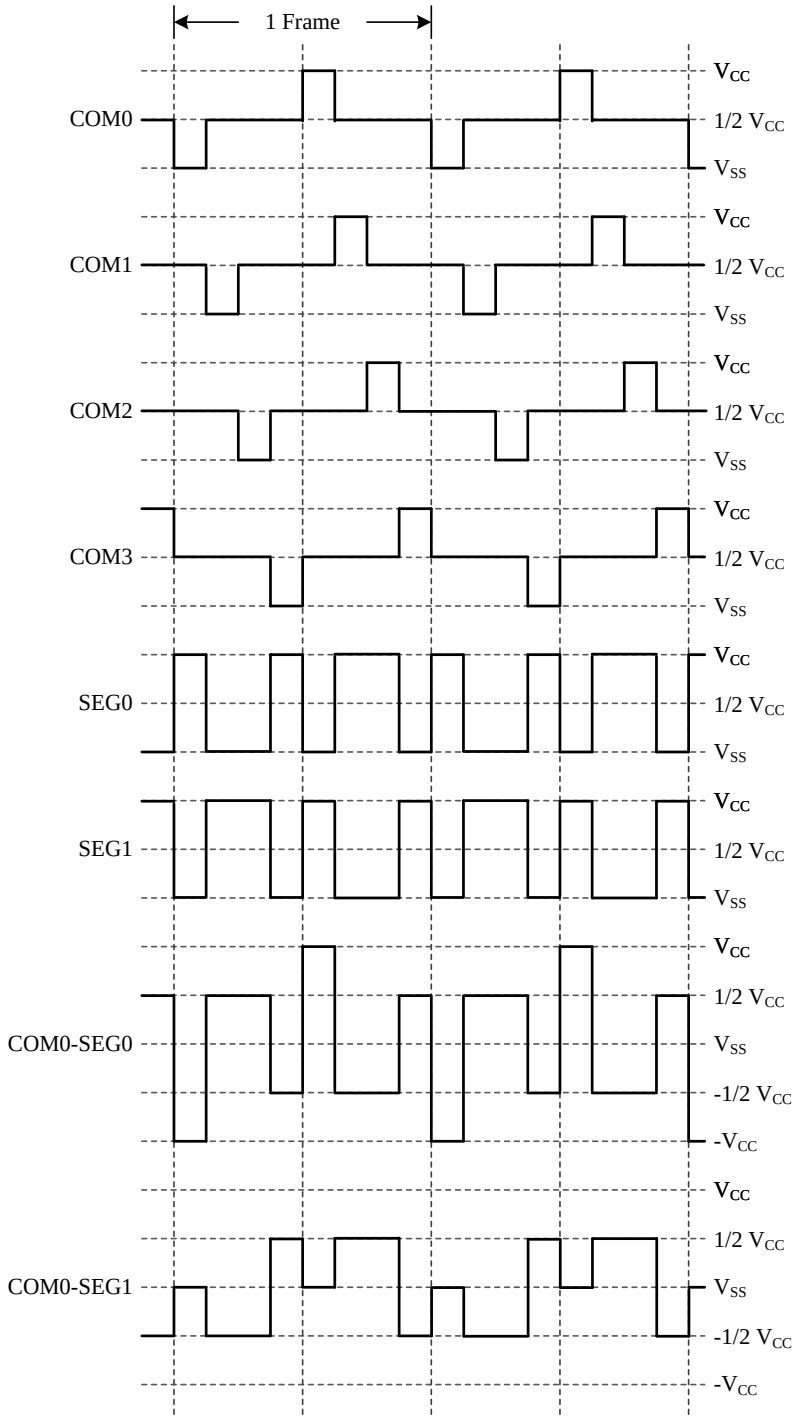
The frequency of any repeating waveform output on the COM pin can be used to represent the LCD frame rate. The figure below shows an LCD frame.




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

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



| SFR **AFh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **P1LOE** | – | – | – | – | P1LOE3 | P1LOE2 | P1LOE1 | P1LOE0 |
| R/W | – | – | – | – | R/W | R/W | R/W | R/W |
| Reset | – | – | – | – | 0 | 0 | 0 | 0 |

92h.3 **P1LOE3:** LCD 1/2 bais Output
0: Disable
1: P15 as LCD 1/2 bais Output

92h.2 **P1LOE2:** LCD 1/2 bais Output
0: Disable
1: P14 as LCD 1/2 bais Output

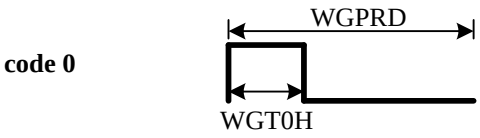
92h.1 **P1LOE1:** LCD 1/2 bais Output
0: Disable
1: P13 as LCD 1/2 bais Output

92h.0 **P1LOE0:** LCD 1/2 bais Output
0: Disable
1: P12 as LCD 1/2 bais Output

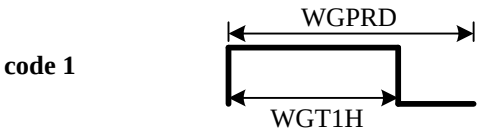
## 14. Full Color LED Communication Format Waveform Generator

The input information of the waveform generator is stored in the 0xF200~0xF217 area of RAM (24 bytes in total). The waveform generator will serially output it to P3.4 according to the following encoding rules.

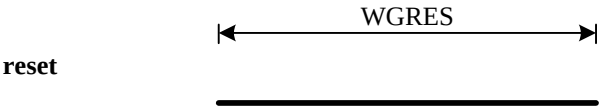
If data value is 0, then the output waveform will encode as shown in the figure below, Duty and Period can be adjusted by WGT0H and WGPRD.




If data value is 1, then the output waveform will encode as shown in the figure below, Duty and Period can be adjusted by WGT1H and WGPRD.




Use a long low level to separate different Transmit data, the length of the low level can be adjusted by WGRES.




The length of the Transmit data can be adjusted by WGRES, ranging from 3 bytes to 24 bytes. The RGB value of one LED is 3 bytes, that is, it can support 1~8 LEDs.

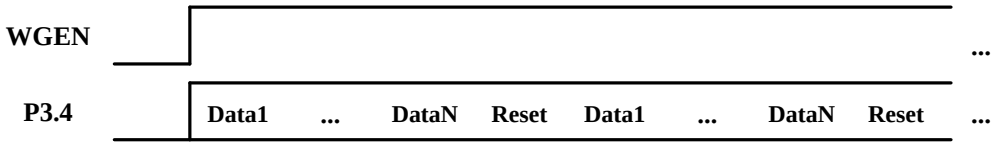
**WGDTS=3**


**P3.4 Serial output (WGDTS=3)**

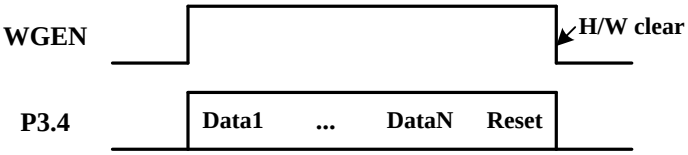
When user set WGMOD= 0 or 1, HW will send data code and reset code continuously, and it stop only if user clear WGEN bit manually; When user set WGMOD = 2, HW will send data code and reset code, then clear WGEN bit automatically by HW; When user set WGMOD = 3, HW only send data code, then clear WGEN bit automatically by HW.

**N=1~8 select by WGDTS**

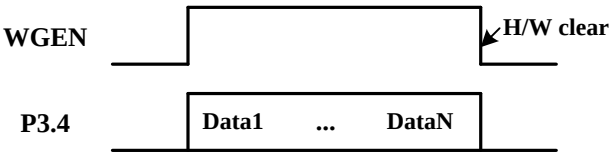
**WGMOD=0 or 1:**




**WGMOD=2:**




**WGMOD=3:**




**Three Mode Options**

| SFR **9Ah** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **WGCON** | WGRES | | WGMOD | | WGTDS | | | WGEN |
| R/W | R/W | | R/W | | R/W | | | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

9Ah.7~6 **WGRES:** WG reset time select ($T_{FRC}$ =60.28 ns)

0: 7*$T_{FRC}$ = 54.0us

1: 11*$T_{FRC}$ = 84.9us

2: 21*$T_{FRC}$ = 162.0us

3: 37*$T_{FRC}$ = 285.5us

9Ah.5~4 **WGMOD:** WG Mode select

0xb：Continue Mode. HW sends data code and reset code, repeating continuously, will not clear WGEN automatically.

10b：One Cycle Mode. HW will automatically clear WGEN after sending data code and reset code

11b：One Cycle Mode. HW will automatically clear WGEN after sending data code.

9Ah.3~1 **WGTDS:** WG Transmit data length select

0： 3 bytes (1 LED)

1： 6 bytes (2 LED)

2： 9 bytes (3 LED)
3：12 bytes (4 LED)
4：15 bytes (5 LED)
5：18 bytes (6 LED)
6：21 bytes (7 LED)
7：24 bytes (8 LED)

9Ah.0 **WGEN:** WG enable and output to P3.4, H/W will automatically clear WGEN in One Cycle mode
0：WG disable, P3.4 as GPIO。 1：WG enable, P3.4 as WG output.

| SFR **9Bh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **WGCON2** | WGPRD | | | WGT1H | | | WGT0H | |
| R/W | R/W | | | R/W | | | R/W | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

9Bh.7~5 **WGPRD:** WG period of each bit select ($T_{FRC}$ =60.28 ns)
0: 17*$T_{FRC}$ = 1025ns
1: 18*$T_{FRC}$ = 1085ns
2: 19*$T_{FRC}$ = 1145ns
3: 20*$T_{FRC}$ = 1206ns
4: 21*$T_{FRC}$ = 1266ns
5: 22*$T_{FRC}$ = 1326ns
6: 23*$T_{FRC}$ = 1386ns
7: 24*$T_{FRC}$ = 1447ns

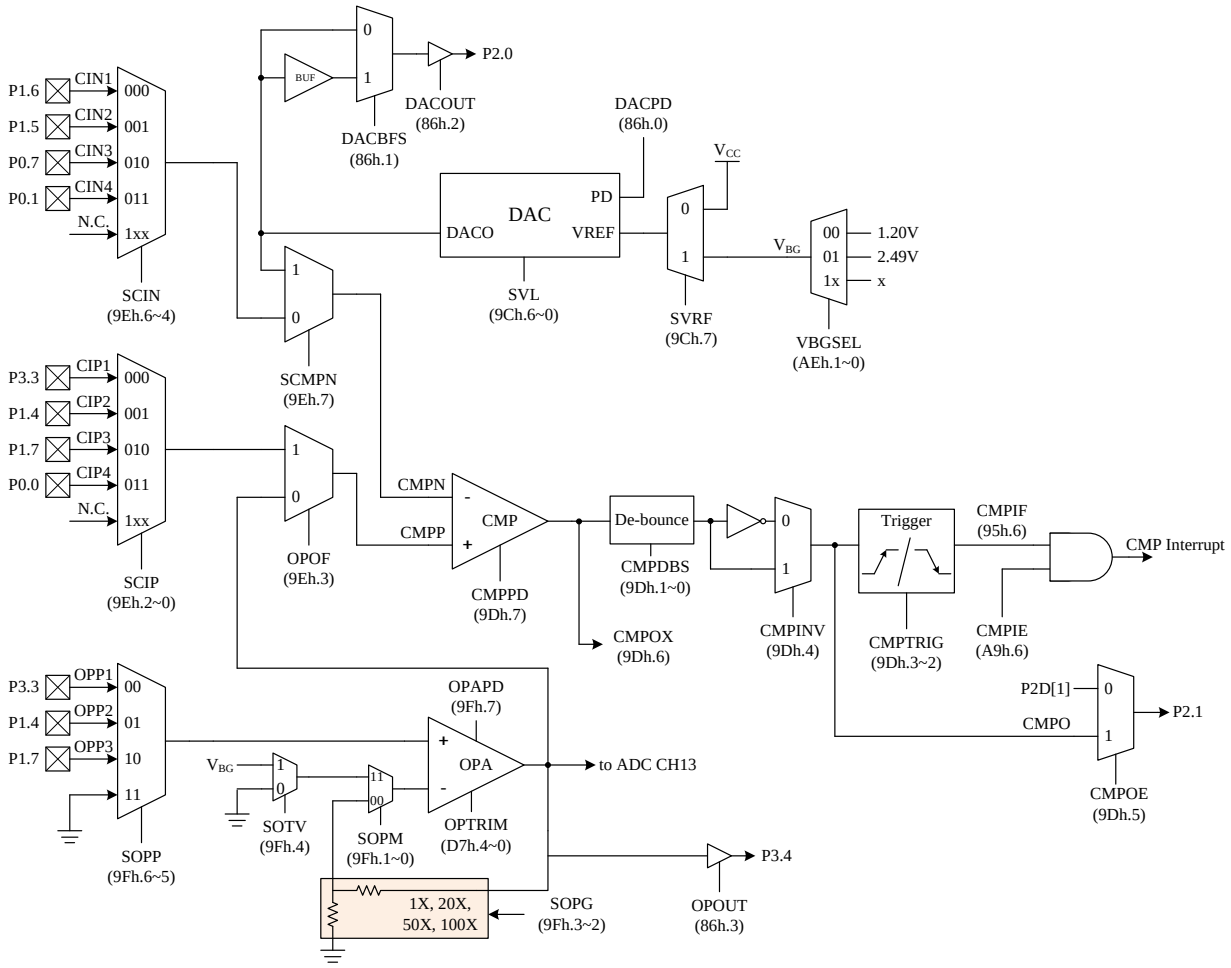
9Bh.4~2 **WGT1H:** WG code 1 high level time select ($T_{FRC}$ =60.28 ns)
0: 10*$T_{FRC}$ = 603ns
1: 11*$T_{FRC}$ = 663ns
2: 12*$T_{FRC}$ = 723ns
3: 13*$T_{FRC}$ = 784ns
4: 14*$T_{FRC}$ = 844ns
5: 15*$T_{FRC}$ = 904ns
6: 16*$T_{FRC}$ = 964ns
7: 17*$T_{FRC}$ =1025ns

9Bh.1~0 **WGT0H:** WG code 0 high level time select ($T_{FRC}$ =60.28 ns)
0: 4*$T_{FRC}$ = 241ns
1: 5*$T_{FRC}$ = 301ns
2: 6*$T_{FRC}$ = 362ns
3: 7*$T_{FRC}$ = 422ns

## 15. Operational Amplifier and Comparator

There is an Operational Amplifier (OPA) and a Comparator (CMP) in this device. The OPA can be set to normal mode or comparator mode by SOPM (9Fh.1~0). By setting the OPAPD=1 (9Fh.7), the OPA will enter power down mode. The SOPP (9Fh.6~5) register determined the OPA positive input channel (OPP) is P3.3, P1.4, P1.7, or VSS. The OPA negative input is VSS in normal mode, but it can be selected as VSS or VBG by SOTV (9Fh.4) in comparator mode. The 4-level OPA gain (1x/20x/50x/100x) used for normal mode is controlled by SOPG (9Fh.3~2).

The CMP built in a 7-bit DAC module, which output can be accessed to negative input port of the CMP. Reference Voltage of DAC can be selected as $V_{CC}$ or $V_{BG}$ by setting SVRF (9Ch.7). $V_{BG}$ will be configured as 1.20V or 2.48V by setting VBGSEL (AEh.1~0). A suitable level of voltage can be selected for proper operation of user application by setting SVL (9Ch.6~0), which will change the resistance to transform the value of voltage. Setting the CMPPD=1 (9Dh.7) will let DAC and CMP enter power down mode. By configuring SCMPN (9Eh.7), negative port input source will be external pin input or DAC output. And positive port input source can be external pin input or OPA output (OPO) by defining OPOF (9Eh.3). The SCIN (9Eh.6~4) and SCIP (9Eh.2~0) register determine negative and positive port external input source respectively. Because the input module of the CMP is composed of PMOS, the input voltage range will be affected by Vth of the PMOS. Thus, the maximum input voltage of the CMP will be ($V_{CC}$-0.5) V. Meanwhile, the Comparator's hysteresis voltage is about 30mV. The Comparator original output (CMPOX) can be read by CMPOX (9Dh.6) bit. The Chip provides a de-bounce module to de-bounce the CMPOX signal, user can select de-bounce time by CMPDBS (9Dh.1~0). The de-bounce output signal can select invert or not by CMPINV (9Dh.4) to generate CMPO signal. The CMPO can be output to pin (P2.1) by set CMPOE (9Dh.5) and the P2MOD1 should be set to 10b. The CMPO is also a trigger source for the interrupt trigger module to generate interrupt flag CMPIF (95h.6). The trigger mode is selected by CMPTRIG (9Dh.3~2). When Comparator power down, the interrupt flag will still be produced. Therefore, it is necessary to clear the interrupt flag first after turning on the CMP module each time to prevent using the dummy flag.




**Operational Amplifier and Comparator Structure**

| A9h | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| INTIE1 | PWMIE | CMPIE | LVDIE | SPI2CIE | ADTKIE | EX2 | PXIE | 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 **CMPIE:** Comparator interrupt enable
0: disable
1: enable

| 95h | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| INTFLG | LVDIF | CMPIF | TKIF | ADIF | – | IE2 | PXIF | TF3 |
| R/W | R/W | R/W | R/W | R/W | – | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | – | 0 | 0 | 0 |

95h.6 **CMPIF:** Comparator interrupt flag
Set by H/W while CMPO match trigger condition. It is cleared automatically when the program performs the interrupt service routine. S/W writes BFh to INTFLG to clear this flag.

| AEh | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| CHSEL | ADCHS | | | | | ADCVREFS | VBGSEL | |
| R/W | R/W | | | | | R/W | R/W | |
| Reset | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |

AEh.1~0 **VBGSEL:** VBG voltage select.
00: 1.20V
01: 2.49V (need VCC>2.8V)
10: Reserved
11:Reserved

| 86h | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| DACON2 | – | – | – | – | OPOUT | DACOUT | DACBFS | DACPD |
| R/W | – | – | – | – | R/W | R/W | R/W | R/W |
| Reset | – | – | – | – | 0 | 0 | 0 | 1 |

86h.3 **OPOUT:** OPO output enable
0: P3.4 as normal IO
1:OPO output to P3.4

86h.2 **DACOUT:** DAC output enable
0:P2.0 as normal IO
1:DAC output to P2.0

86h.1 **DACBFS:** DAC Output Buffer select.
0: Output without Buffer
1: Output with Buffer

86h.0 **DACPD:** DAC Power Down.
0: DAC enable.
1: DAC disable.

| 9Ch | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| DACON | SVRF | SVL | | | | | | |
| R/W | R/W | R/W | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

9Ch.7 **SVRF:** DAC reference voltage select
0: VDDA
1: VBGO (define by VBGSEL AEh.1~0)

9Ch.6~0 **SVL:** Select DAC output voltage (reference source can be selected as VDDA or VBGO)
000_0000: 0/128 * reference source
000_0001: 1/128 * reference source
…
111_1101: 125/128 * reference source
111_1110: Reserved
111_1111: Reserved

| 9Dh | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| CMPCON | CMPPD | CMPOX | CMPOE | CMPINV | CMPTRIG | | CMPDBS | |
| R/W | R/W | R | R/W | R/W | R/W | | R/W | |
| Reset | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

9Dh.7 **CMPPD:** Comparator & DAC power down enable control
0: disable Comparator & DAC power down

1: enable Comparator & DAC power down

9Dh.6 **CMPOX:** Comparator original output (CMPOX) status
0: $V_{CMPP} < V_{CMPN}$
1: $V_{CMPP} > V_{CMPN}$ or CMPPD =1

9Dh.5 **CMPOE:** Comparator output (CMPO) signal output to P2.1
0: disable
1: enable, P2MOD1 should be set to 10b

9Dh.4 **CMPINV:** Comparator de-bounce output invert select
0: no invert
1: invert

9Dh.3~2 **CMPTRIG:** Comparator interrupt trigger mode
00: Rising edge
01: Falling edge
10: Both edge
11: High level

9Dh.1~0 **CMPDBS:** Comparator original output (CMPOX) de-bounce time
00: none
01: 4 Fsys
10: 8 Fsys
11: 16 Fsys

| 9Eh | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| CMPPNS | SCMPN | SCIN | | | OPOF | SCIP | | |
| R/W | R/W | R/W | | | R/W | R/W | | |
| Reset | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |

9Eh.7 **SCMPN:** Comparator CMPN source select
0: Comparator CMPN source is external input (CINx)
1: Comparator CMPN source is DAC output

9Eh.6~4 **SCIN:** Comparator CMPN external input select
000: Comparator CMPN external input is CIN1 (P1.6)
001: Comparator CMPN external input is CIN2 (P1.5)
010: Comparator CMPN external input is CIN3 (P0.7)
011: Comparator CMPN external input is CIN4 (P0.1)
1xx: No connect

9Eh.3 **OPOF:** OPA output (OPO) connect to Comparator CMPP
0: Comparator CMPP source is OPA output (OPO)
1: Comparator CMPP source is external input (CIPx)

9Eh.2~0 **SCIP:** Comparator CMPP external input select
000: Comparator CMPP external input is CIP1 (PA1)
001: Comparator CMPP external input is CIP2 (PA2)
010: Comparator CMPP external input is CIP3 (PB6)
011: Comparator CMPP external input is CIP4 (PD1)
1xx: No connect

| 9Fh | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| OPCON | OPAPD | SOPP | | SOTV | SOPG | | SOPM | |
| R/W | R/W | R/W | | R/W | R/W | | R/W | |
| Reset | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |

9Fh.7 **OPAPD:** OPA power down enable control
0: disable OPA power down
1: enable OPA power down

9Fh.6~5 **SOPP:** select OPP input source
00: OPP input source is P3.3

01: OPP input source is P1.4
10: OPP input source is P1.7
11: OPP input source is VSS

9Fh.4 **SOTV:** OPN input voltage selection in Comparator mode
0: $V_{SS}$
1: $V_{BG}$ (voltage level is selected by VBGSEL)

9Fh.3~2 **SOPG:** select OPA gain
00: 1X
01: 20X
10: 50X
11: 100X

9Fh.1~0 **SOPM:** select OPA operating mode
00: Normal Mode
01: Reserved
10: Reserved
11: Comparator Mode

| D7h | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| CFGOP | – | – | – | OPTRIM | | | | |
| R/W | – | – | – | R/W | | | | |
| Reset | – | – | – | - | - | - | - | - |

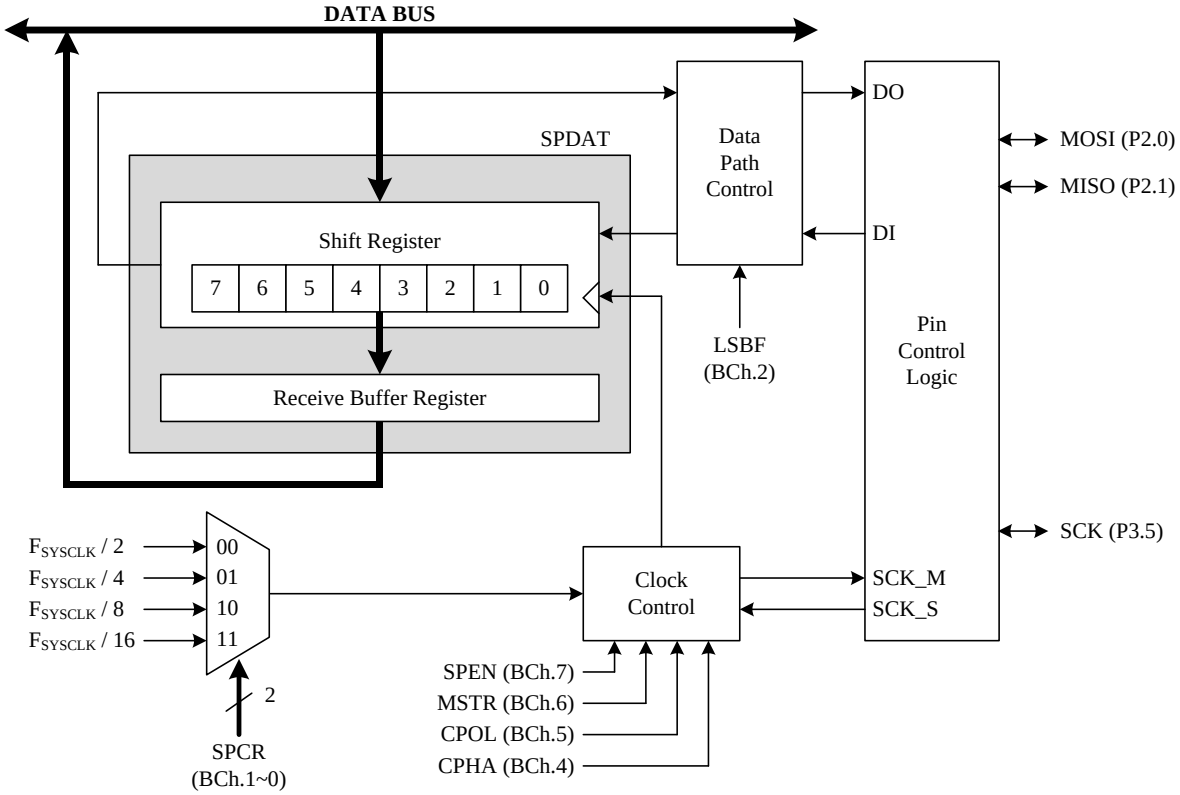
D7h.4~0 **OPTRIM:** OP trimming value.
00000: minimum
11111:maximum

## 16. Serial Peripheral Interface (SPI)

The SPI module is capable of full-duplex, synchronous, serial communication between the chip and peripheral devices. The peripheral devices can be other MCUs, A/D converter, sensors, or Flash memory, etc. The SPI runs at a baud rate up to the system clock divided by two. Firmware can read the status flags, or the operation can be interrupt driven.

The features of the SPI module include:

- Master or Slave mode operation
- 3-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




The MOSI (P2.0) signal is an output when SPI is operating in Master mode and an input when SPI is operating in Slave mode. The MISO (P2.1) signal is an input when SPI is operating in Master mode and an output when SPI is operating in Slave mode. Data is transferred MSB or LSB first by setting the LSBF bit. The SCK (P3.5) 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.

### Master Mode

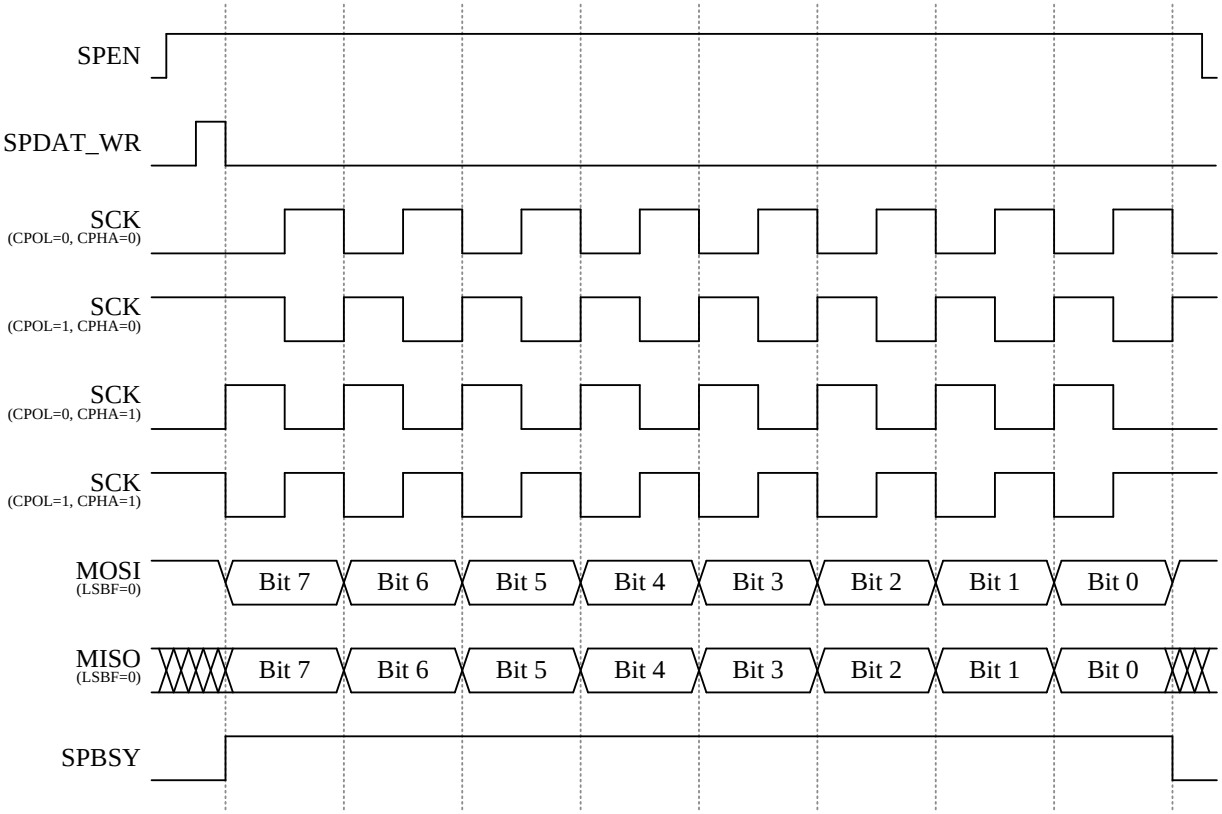
The SPI operates in Master mode by setting the MSTR bit in the SPCON. To start transmit, writing a data to the SPDAT. If SPBSY=0, the data will be transferred to the shift register and starts shift out on the MOSI line. The data of the Slave shift in from the MISO line at the same time. When the SPIF bit becomes set at the end of 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**

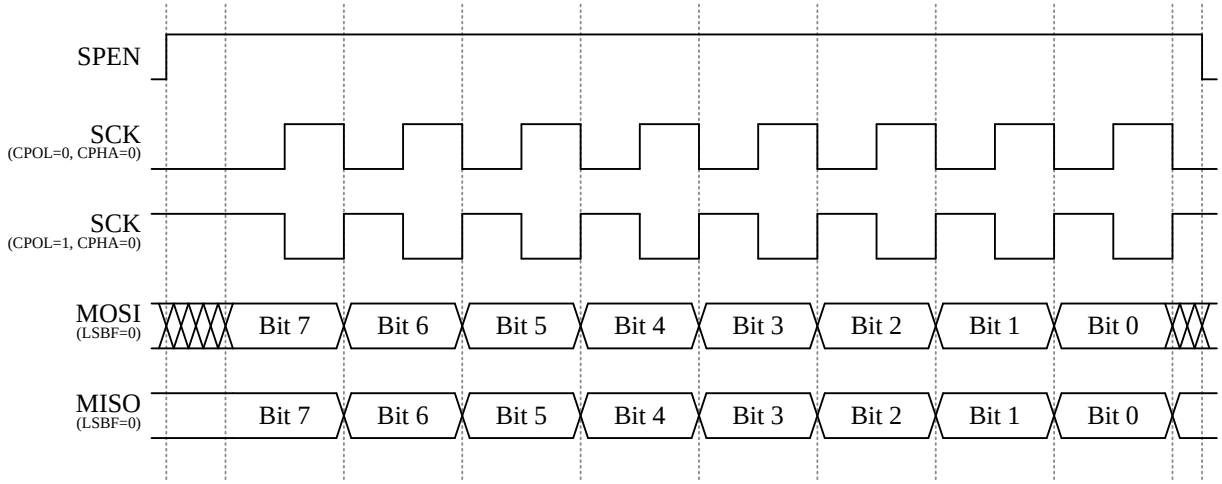
The SPI operates in Slave mode by clearing the MSTR bit in the SPCON. The transmission will start when the SPEN bit in the SPCON is set. The data from a Master will shift into the shift register through the MOSI line, and shift out from the shift register on the MISO line. When a byte enters the shift register, the data will be transferred to receiver buffer if RCVBF=0. If RCVBF=1, the newer received 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 $F_{SYSCLK}/4$.

**Serial Clock**

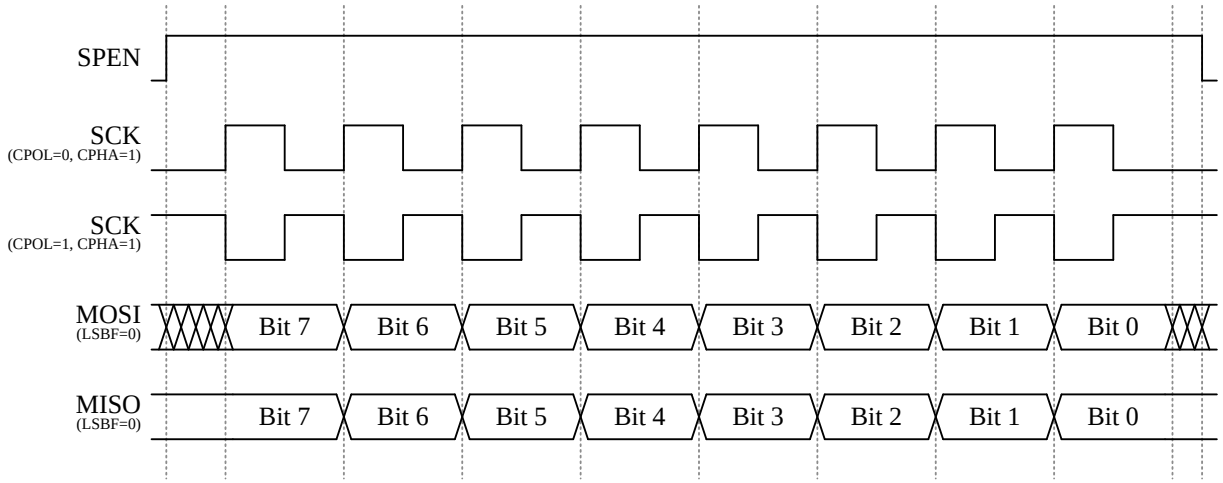
The SPI has four clock types by setting the CPOL and CPHA bits in the SPCON register. The CPOL bit defines the level of the SCK in SPI idle state. The level of the SCK in idle state is low when CPOL=0, and is high when CPOL=1. The CPHA bit defines the edges used to sample and shift data. The SPI sample data on the first edge of SCK period and shift data on the second edge of SCK period when CPHA=0. The SPI sample data on the second edge of SCK period and shift data on first edge of SCK period when CPHA=1. 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 interrupt flag is set by H/W at the end of a data transfer. If write data to SPDAT when SPBSY=1, the WCOL interrupt flag will be set by H/W. When this occurs, the data write to SPDAT will be ignored, and shift register will not be written.

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

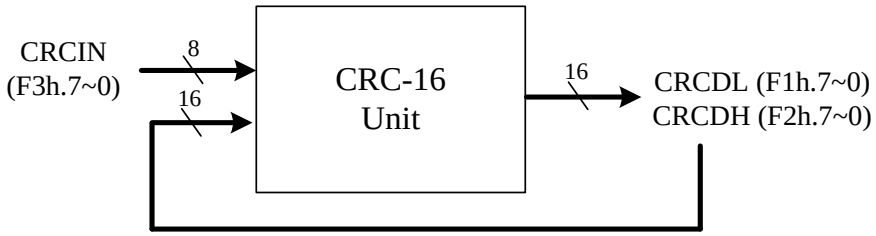
BDh.2 **SPBSY:** SPI busy flag
Set by H/W when a SPI transfer is in progress.

| 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.

## 17. 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

## 18. 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

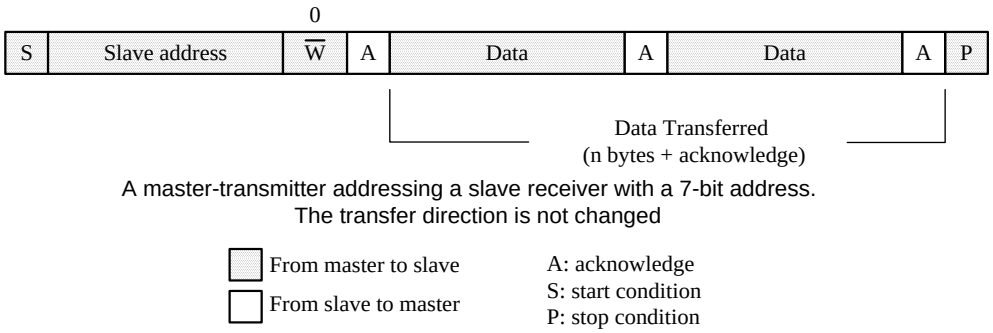
F7h.0 **MULDIV16:**
0: instruction MUL/DIV as 8*8, 8/8 operation
1: instruction MUL/DIV as 16*16, 16/16 or 32/16 operation

| **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 |

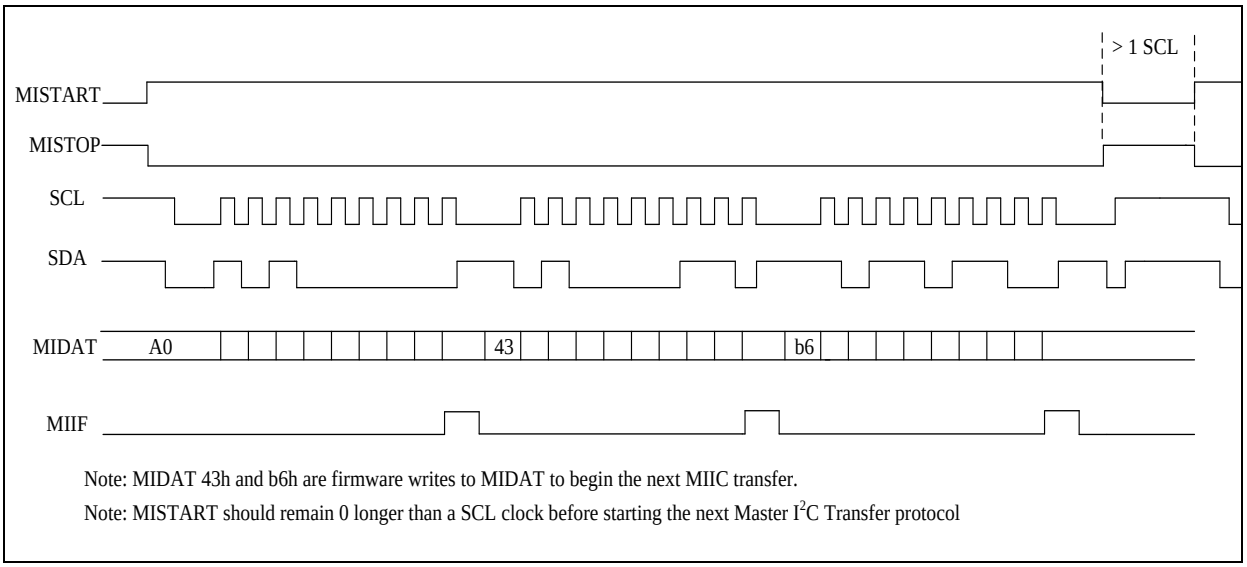
## 19. Master I²C Interface

**Master I²C interface Transmitter 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 transmitter 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 I²C protocol. SCL clock can be adjusted via MICR.




Master I²C Transmit flow:
(1) Write slave address and direction bit to MIDAT
(2) Clear MISTOP and set MISTART to start I²C transmission
(3) Wait until MIIF converter 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 converter to 1 (interrupt will be issued according to the user's request) and Clear MIIF, Loop (3) ~(4) for next transfer.
(6) Clear MISTART and set MISTOP to stop the I²C transfer



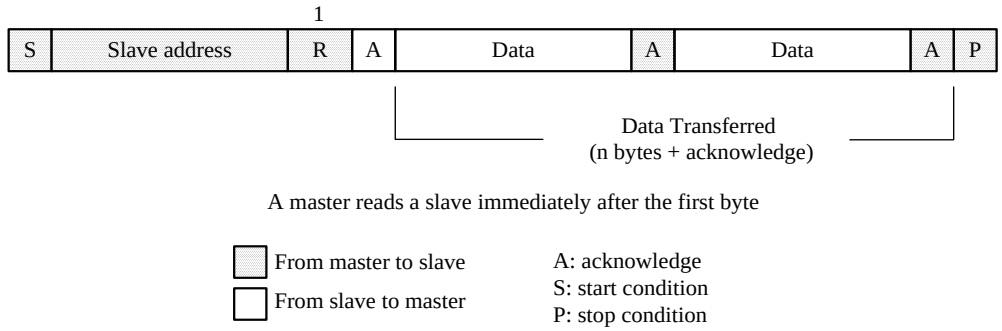

**Master Transmit Timing**

*Note: MISTART should remain 0 longer than a SCL period before starting the next Master I²C 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 $I^2C$ 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 converter to 1 (interrupt will be issued according to the user's request)

(4) Clear MIIF

(5) Read data from MIDAT to start first receive data

(The first reading of MIDAT does not represent the data returned by the slave)

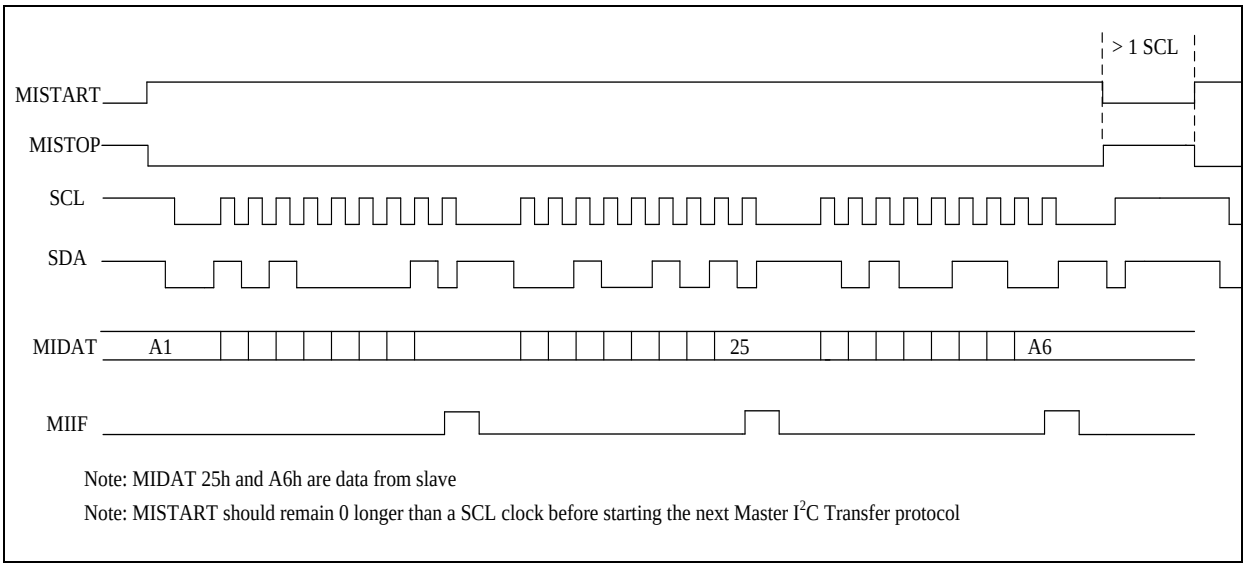
(6) Wait until MIIF converter to 1

(7) Clear MIIF

(8) Read slave data from MIDAT and receive next data

(9) Loop (6) ~(8)

(10) 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 $I^2C$ Transfer protocol

**Master Receive Timing**

| SFR **A9h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **INTE1** | PWMIE | CMPIE | LVDIE | SPI2CE | ADTKIE | EX2 | PXIE | 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.4 **SPI2CE:** I²C interrupt enable
0: Disable SPI/I²C interrupt 1: Enable SPI/I²C interrupt

| SFR **B7h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWMOE2** | MSDASEL | MSCLSEL | PWM6OE2 | PWM6OE1 | PWM6OE0 | PWM5OE2 | PWM5OE1 | PWM5OE0 |
| 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 |

B7h.7 **MSDASEL:** Master $I^2C$ SDA select
0: P3.5 as Master $I^2C$ SDA
1: P1.6 as Master $I^2C$ SDA

B7h.6 **MSCLSEL:** Master $I^2C$ SCL select
0: P1.3 as Master $I^2C$ SCL
1: P0.2 as Master $I^2C$ SCL

| 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
0: write 0 to clear it
1: Master $I^2C$ transfer one byte complete

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=16MHz, $I^2C$ clock is 4M Hz)
01: Fsys/16 (ex. If Fsys=16MHz, $I^2C$ clock is 1M Hz)
10: Fsys/64 (ex. If Fsys=16MHz, $I^2C$ clock is 250K Hz)
11: Fsys/256 (ex. If Fsys=16MHz, $I^2C$ clock is 62.5K Hz)

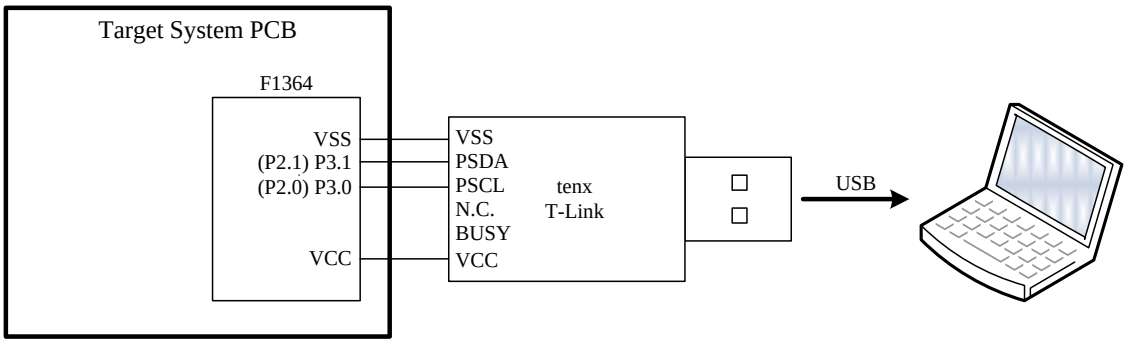
| SFR **E2h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **MIDAT** | MIDAT | | | | | | | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

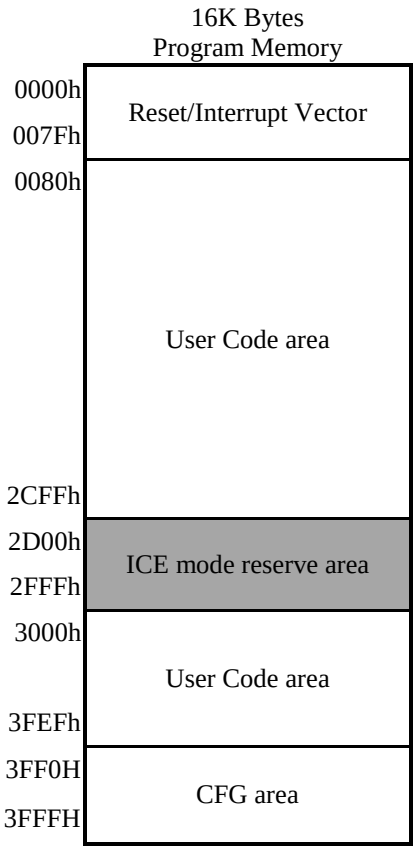
E2h.7~0 **MIDAT**: Master $I^2C$ data shift register
(W):After Start and before Stop condition, write this register will resume transmission to $I^2C$ bus
(R): After Start and before Stop condition, read this register will resume receiving from $I^2C$ bus

## 20. In Circuit Emulation (ICE) Mode

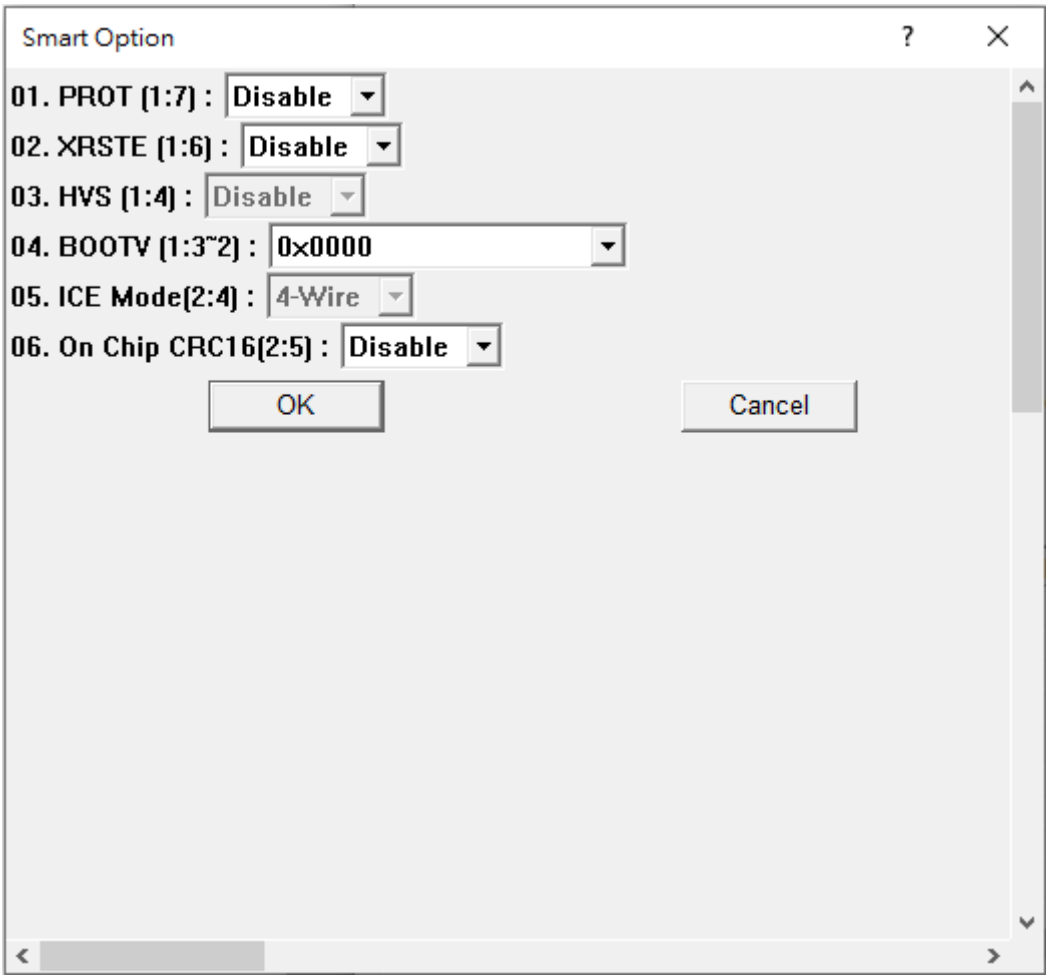
This device can support the In Circuit Emulation Mode. To use the ICE Mode, user just needs to connect P3.0 and P3.1 pin to the tenx proprietary EV Module. The benefit is that user can emulate the whole system without changing the on board target device. But there are some limits for the ICE mode as 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 2D00h~2FFFh 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 P2.0 and P2.1. (Only emulation can be replaced, mass production writer only supports P3.0/P3.1)
6. The VDD level is controlled by T-Link module.



**ICE tool settings introduction**




| No. | Item | Description |
|---|---|---|
| 01 | PROT | **Enable:** Flash code is protect, Writer cannot access the ROM code<br>**Disable:** Flash code is not protect, Writer can access the ROM code (default) |
| 02 | XRSTE | **Enable:** P3.7 is external reset pin<br>**Disable:** P3.7 is normal I/O pin (default) |
| 03 | HVS | Reserved |
| 04 | BOOTV | Reset Vector after POR<br>00: Reset Vector = 0x3800, BOOT Area Size = 2K<br>01: Reset Vector = 0x3000, BOOT Area Size = 4K<br>1x: Reset Vector = 0x0000, no BOOT Area |
| 05 | ICE Mode | Reserved |
| 06 | On Chip CRC16 | **Enable:** On chip CRC-16 function enable<br>**Disable:** On chip CRC-16 function disable (default) |

# SFR & CFGW MAP

| Adr | RST | NAME | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 80h | 1111-1111 | **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 | x00x-xx00 | **INTE2** | – | PWM1IE | PWM0IE | – | – | – | TKBIE | TKAIE |
| 85h | x00x-xx00 | **INTFLG2** | – | PWM1IF | PWM0IF | – | – | – | TKBIF | TKAIF |
| 86h | xxxx-0001 | **DACON2** | – | – | – | – | OPOUT | DACOUT | DACBFS | DACPD |
| 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 | xxxx-xxxx | **TKBDL** | TKBHL | | | | | | | |
| 8Fh | xxxx-xxxx | **TKBDH** | – | – | TKBDH | | | | | |
| 90h | 1111-1111 | **P1** | P1.7 | P1.6 | P1.5 | P1.4 | P1.3 | P1.2 | P1.1 | P1.0 |
| 91h | 0101-0101 | **P0MODL** | P0MOD3 | | P0MOD2 | | P0MOD1 | | P0MOD0 | |
| 92h | 0101-0101 | **P0MODH** | P0MOD7 | | P0MOD6 | | P0MOD5 | | P0MOD4 | |
| 93h | 0000-0101 | **PINMOD** | TXRXSEL | T2OE | T1OE | T0OE | P2MOD1 | | P2MOD0 | |
| 94h | 0000-0000 | **OPTION** | UART1W | TM3CKS | WDTPSC | | ADCKS | | TM3PSC | |
| 95h | 0000-x000 | **INTFLG** | LVDIF | CMPIF | TKIF | ADIF | – | IE2 | PXIF | TF3 |
| 96h | 0000-0000 | **P1WKUP** | P1WKUP | | | | | | | |
| 97h | xxxx-xx00 | **SWCMD** | SWRST / IAPALL / WDTO | | | | | | | |
| 98h | 0000-0000 | **SCON** | SM0 | SM1 | SM2 | REN | TB8 | RB8 | TI | RI |
| 99h | xxxx-xxxx | **SBUF** | SBUF | | | | | | | |
| 9Ah | 0000-0000 | **WGCON** | WGRES | | WGMOD | | WGTDS | | | WGEN |
| 9Bh | 0000-0000 | **WGCON2** | WGPRD | | | WGT1H | | | WGT0H | |
| 9Ch | 0000-0000 | **DACON** | SVRF | SVL | | | | | | |
| 9Dh | 1100-0000 | **CMPCON** | CMPPD | CMPOX | CMPOE | CMPINV | CMPTRIG | | CMPDBS | |
| 9Eh | 1111-0111 | **CMPPNS** | SCMPN | SCIN | | | OPOF | SCIP | | |
| 9Fh | 1110-1100 | **OPCON** | OPAPD | SOPP | | SOTV | SOPG | | SOPM | |
| A0h | 1111-1111 | **P2** | P2.7 | P2.6 | P2.5 | P2.4 | P2.3 | P2.2 | P2.1 | P2.0 |
| A1h | 0000-0000 | **PWMCON** | PWM1CKS | | PWM1EN | PWM0EN | PWM0CKS | | PWM0NMSK | PWM0PMSK |
| 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 | 0101-0101 | **P3MODH** | P3MOD7 | | P3MOD6 | | P3MOD5 | | P3MOD4 | |
| A6h | 0000-0000 | **PWMOE0** | PWM1OE3 | PWM1OE2 | PWM1OE1 | PWM1OE0 | PWM0NOE1 | PWM0POE1 | PWM0NOE0 | PWM0POE0 |
| A7h | 0000-0000 | **PWMCON2** | PWM0MOD | PWM0MSKE | PWM0OM | | PWM0DZ | | | |
| A8h | 0x00-0000 | **IE** | EA | – | ET2 | ES | ET1 | EX1 | ET0 | EX0 |
| A9h | 0000-0000 | **INTE1** | PWMIE | CMPIE | LVDIE | SPI2CIE | ADTKIE | EX2 | PXIE | TM3IE |
| AAh | xxxx-xxxx | **ADCDL** | ADCDL | | | | – | – | – | PWRDEC |
| ABh | xxxx-xxxx | **ADCDH** | ADCDH | | | | | | | |
| ACh | xxxx-xxxx | **TKADL** | TKADL | | | | | | | |
| ADh | 110x-110x | **TKCON** | TKAPD | TKAEOC | TKASOC | – | TKBPD | TKBEOC | TKBSOC | – |
| AEh | 1111-1000 | **CHSEL** | ADCHS | | | | ADCHS4 | ADCVREFS | VBGSEL | |
| AFh | xxxx-0000 | **P1LOE** | – | | | | P1LOE | | | |
| B0h | 1111-1111 | **P3** | P3.7 | P3.6 | P3.5 | P3.4 | P3.3 | P3.2 | P3.1 | P3.0 |
| B1h | x000-0000 | **TKAREFC** | – | TKAREFC | | | | | | |
| B2h | xxxx-xxxx | **TKADH** | – | – | TKADH | | | | | |
| B3h | xx11-xx11 | **TKCHS** | – | – | TKBCHS | | – | – | TKACHS | |
| B4h | 1111-1111 | **TKATMRL** | TKATMRL | | | | | | | |
| B5h | xx00-0000 | **TKATMRH** | – | – | TKATMRH | | | | | |
| B6h | 0000-0000 | **PWMOE1** | PWM4OE3 | PWM4OE2 | PWM4OE1 | PWM4OE0 | PWM3OE1 | PEM3OE0 | PWM2OE1 | PWM2OE0 |
| B7h | 0000-0000 | **PWMOE2** | MSDASEL | MSCLSEL | PWM6OE2 | PWM6OE1 | PWM6OE0 | PWM5OE2 | PWM5OE1 | PWM5OE0 |
| B8h | xx00-0000 | **IP** | – | – | PT2 | PS | PT1 | PX1 | PT0 | PX0 |
| B9h | xx00-0000 | **IPH** | – | – | PT2H | PSH | PT1H | PX1H | PT0H | PX0H |

| Adr | RST | NAME | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| BAh | 0000-0000 | **IP1** | PPWM | PCMP | PLVD | PSPI2C | PADTKI | PX2 | PPX | PT3 |
| BBh | 0000-0000 | **IP1H** | PPWMH | PCMPH | PLVDH | PSPI2CH | PADTKIH | PX2H | PPXH | PT3H |
| BCh | 0000-0000 | **SPCON** | SPEN | MSTR | CPOL | CPHA | – | LSBF | SPCR | |
| BDh | 0000-00xx | **SPSTA** | SPIF | WCOL | – | RCVOVF | RCVBF | SPBSY | – | – |
| BEh | 0000-0000 | **SPDAT** | SPDAT | | | | | | | |
| BFh | xxxx-x1xx | **BOOTV** | – | – | – | – | – | RSTV | BOOTVR | |
| C1h | 1111-1111 | **TKBTMRL** | TKBTMRL | | | | | | | |
| C2h | xx00-0000 | **TKBTMRH** | – | – | TKBTMRH | | | | | |
| C3h | x000-0000 | **TKBKCP** | – | TKBKCP | | | | | | |
| C4h | x000-0000 | **TKBREFC** | – | TKBREFC | | | | | | |
| C5h | 0000-0000 | **P0WKUP** | P0WKUP | | | | | | | |
| C6h | 0000-0000 | **P2WKUP** | P2WKUP | | | | | | | |
| C7h | 0000-0000 | **P3WKUP** | P3WKUP | | | | | | | |
| C8h | 0000-0000 | **T2CON** | TF2 | EXF2 | RCLK | TCLK | EXEN2 | TR2 | CT2N | CPRL2N |
| C9h | 0000-xxxx | **IAPCON** | IAPCON / IAPWE / EEPWE / INFOWE / 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 | 0000-0000 | **PWM0DH** | PWM0DH | | | | | | | |
| D2h | 0000-0000 | **PWM0DL** | PWM0DL | | | | | | | |
| D3h | 0000-0000 | **PWM1DH** | PWM1DH | | | | | | | |
| D4h | 0000-0000 | **PWM1DL** | PWM1DL | | | | | | | |
| D5h | 0000-0000 | **PWM2DH** | PWM2DH | | | | | | | |
| D6h | 0000-0000 | **PWM2DL** | PWM2DL | | | | | | | |
| D7h | xxxx-xxxx | **CFGOP** | – | – | – | OPTRIM | | | | |
| D8h | xxx0-0011 | **CLKCON** | – | – | 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 | 0000-0000 | **PWM3DH** | PWM3DH | | | | | | | |
| DEh | 0000-0000 | **PWM3DL** | PWM3DL | | | | | | | |
| DFh | 0000-0000 | **UARTCON** | UARTBRS | UARTBRP | | | | | | |
| 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 | | | | | | | |
| E3h | xx00-0000 | **LVRCON** | – | – | PORPD | LVRPD | LVRSEL | | | |
| E4h | 0000-0000 | **LVDCON** | LVDM | LVDO | LVDHYS | LVDPD | LVDSEL | | | |
| E5h | 0000-0000 | **EFTCON** | EFT2CS | EFT1CS | EFT1S | | EFTSLOW | EFTWCPU | EFTWOUT | CKHLDE |
| E6h | 0000-0000 | **EXA** | EXA | | | | | | | |
| E7h | 0000-0000 | **EXB** | EXB | | | | | | | |
| E9h | 0000-0000 | **PWM4DH** | PWM4DH | | | | | | | |
| EAh | 0000-0000 | **PWM4DL** | PWM4DL | | | | | | | |
| EBh | 0000-0000 | **PWM5DH** | PWM5DH | | | | | | | |
| ECh | 0000-0000 | **PWM5DL** | PWM5DL | | | | | | | |
| EDh | 0000-0000 | **PWM6DH** | PWM6DH | | | | | | | |
| EEh | 0000-0000 | **PWM6DL** | PWM6DL | | | | | | | |
| EFh | 1110-0011 | **PWRCON** | IVCPD | IVCVS | | – | WARMTIME | – | – | – |
| 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** | – | FRCTRIM | | | | | | |
| F7h | 0000-0110 | **AUX2** | WDTE | | PWRSAV | VBGOUT | DIV32 | IAPTE | | MULDIV16 |
| F8h | 0000-1100 | **AUX1** | CLRWDT | CLRTM3 | – | ADSOC | CLRPWM0 | CLRPWM1 | LDOCOUT | DPSEL |

| Flash Address | NAME | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|
| 3FF9h | **CFGOP** | – | – | – | OPTRIM | | | | |
| 3FFBh | **CFGBG** | – | – | – | BGTRIM | | | | |
| 3FFDh | **CFGWL** | – | FRCTRIM | | | | | | |
| 3FFFh | **CFGWH** | PROT | XRSTE | – | HVS | BOOTV | | – | – |

# SFR & CFGW DESCRIPTION

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| 80h | **P0** | 7~0 | P0 | R/W | FFh | 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 | **INTE2** | 6 | PWM1IE | R/W | 0 | PWM1~PWM6 interrupt enable<br>0: Disable PWM1~PWM6 interrupt<br>1: Enable PWM1~PWM6 interrupt |
| | | 5 | PWM0IE | R/W | 0 | PWM0 interrupt enable<br>0: Disable PWM0 interrupt<br>1: Enable PWM0 interrupt |
| | | 1 | TKBIE | R/W | 0 | Touch Key B interrupt enable<br>0: Disable Touch Key B interrupt<br>1: Enable Touch Key B interrupt |
| | | 0 | TKAIE | R/W | 0 | Touch Key A interrupt enable<br>0: Disable Touch Key A interrupt<br>1: Enable Touch Key A interrupt |
| 85h | **INTFLG2** | 6 | PWM1IF | R/W | 0 | PWM1~PWM6 interrupt flag<br>Set by H/W at the end of PWM1 period, S/W writes BFh to INTFLG2 to clear this flag. |
| | | 5 | PWM0IF | R/W | 0 | PWM0 interrupt enable<br>Set by H/W at the end of PWM0 period, S/W writes DFh to INTFLG2 to clear this flag. |
| | | 1 | TKBIF | R/W | 0 | Touch Key B interrupt flag<br>Set by H/W at the end of TKB scan, S/W writes FDh to INTFLG2 to clear this flag. |
| | | 0 | TKAIF | R/W | 0 | Touch Key A interrupt enable<br>Set by H/W at the end of TKA scan, S/W writes FEh to INTFLG2 to clear this flag. |
| 86h | **DACON2** | 3 | OPOUT | R/W | 0 | 0: P3.4 as normal IO<br>1:OPO output to P3.4 |
| | | 2 | DACOUT | R/W | 0 | 0:P2.0 as normal IO<br>1:DAC output to P2.0 |
| | | 1 | DACBFS | R/W | 0 | DAC Output Buffer select.<br>0: Output without Buffer, 1: Output with Buffer |
| | | 0 | DACPD | R/W | 1 | DAC Power Down.<br>0: DAC enable. 1: DAC disable. |
| 87h | **PCON** | 7 | SMOD | R/W | 0 | Set 1 to enable UART 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 HALT/STOP 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. |

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| | | 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 |
| 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: 8-bit timer/counter (TH1) and 5-bit prescaler (TL1)<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: 8-bit timer/counter (TH0) and 5-bit prescaler (TL0)<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 | **TKBDL** | 7~0 | TKBDL | R | - | TKB data low byte |
| 8Fh | **TKBDH** | 5~0 | TKBDH | R | - | TKB data high byte |
| 90h | **P1** | 7~0 | P1 | R/W | FFh | Port1 data |
| 91h | **P0MODL** | 7~6 | P0MOD3 | R/W | 01 | P0.3 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3 |
| | | 5~4 | P0MOD2 | R/W | 01 | P0.2 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3 |
| | | 3~2 | P0MOD1 | R/W | 01 | P0.1 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3 |
| | | 1~0 | P0MOD0 | R/W | 01 | P0.0 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3 |
| 92h | **P0MODH** | 7~6 | P0MOD7 | R/W | 01 | P0.7 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3 |
| | | 5~4 | P0MOD6 | R/W | 01 | P0.6 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3 |
| | | 3~2 | P0MOD5 | R/W | 01 | P0.5 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3 |
| | | 1~0 | P0MOD4 | R/W | 01 | P0.4 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3 |

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| 93h | **PINMOD** | 7 | TXRXSEL | R/W | 0 | UART TXD/RXD pin select<br>0: P31 as TXD, P30 as RXD<br>1: P16 as TXD, P02 as RXD |
| | | 6 | T2OE | R/W | 0 | Timer2 signal output (T2O) control<br>0: Disable "Timer2 overflow divided by 2" output to P1.0 pin<br>1: Enable "Timer2 overflow divided by 2" output to P1.0 pin |
| | | 5 | T1OE | R/W | 0 | Timer1 signal output (T1O) control<br>0: Disable "Timer1 overflow divided by 2" output to P3.5 pin<br>1: Enable "Timer1 overflow divided by 2" output to P3.5 pin |
| | | 4 | T0OE | R/W | 0 | Timer0 signal output (T0O) control<br>0: Disable "Timer0 overflow divided by 64" output to P3.4 pin<br>1: Enable "Timer0 overflow divided by 64" output to P3.4 pin |
| | | 3~2 | P2MOD1 | R/W | 01 | P2.1 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3 |
| | | 1~0 | P2MOD0 | R/W | 01 | P2.0 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3 |
| 94h | **OPTION** | 7 | UART1W | R/W | 0 | Set 1 to enable one wire UART mode, both TXD/RXD use P3.1 pin or P1.6. |
| | | 6 | TM3CKS | R/W | 0 | Timer3 clock source select.<br>0: Slow Clock (SRC) 1: SRC/0.75 |
| | | 5~4 | WDTPSC | R/W | 00 | Watchdog Timer pre-scalar time select<br>00: 400ms WDT overflow rate<br>01: 200ms WDT overflow rate<br>10: 100ms WDT overflow rate<br>11: 50ms 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 prescaler.<br>00: 32768 clock cycle<br>01: 4096 clock cycle<br>10: 2048 clock cycle<br>11: 512 clock cycle |
| 95h | **INTFLG** | 7 | LVDIF | R/W | 0 | LVD interrupt flag<br>Set by H/W when $V_{CC}$ less than the LVD voltage. S/W writes 7Fh to INTFLG to clear this flag. |
| | | 6 | CMPIF | R/W | 0 | CMP interrupt flag<br>Set by H/W while CMPO match trigger condition. It is cleared automatically when the program performs the interrupt service routine. S/W writes BFh to INTFLG to clear this flag. |
| | | 5 | TKIF | R/W | 0 | Touch Key interrupt flag<br>Set by H/W at the end of TK conversion. S/W writes DFh to INTFLG or sets the TKSOC bit to clear this flag. When user clears this flag, H/W will automatically clear TKAIF and TKBIF. |
| | | 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. |
| | | 2 | IE2 | R/W | 0 | External Interrupt 2 (INT2 pin) edge flag<br>Set by H/W when a falling edge is detected on the INT2 pin, no matter the EX2 is 0 or 1. It is cleared automatically when the program performs the interrupt service routine. S/W can write FBh to INTFLG to clear this bit. |

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| | | 1 | PXIF | R/W | 0 | Port0~3 pin change Interrupt flag<br>Set by H/W when a Port0~3 pin state change is detected and its interrupt enable bit is set (P0WKUP/P1WKUP/P2WKUP/P3WKUP). PXIE 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 | IAPALL | W | | Write 65h to set IAPALL flag. Write other value to clear IAPALL flag. |
| | | 1 | WDTO | R | 0 | Watchdog Time-Out flag |
| | | 0 | IAPALL | R | 0 | Flag indicates Flash can be written by IAP or not<br>0: Flash IAP disable<br>1: Flash IAP enable, only for BOOT mode upgrade APP area. |
| 98h | **SCON** | 7 | SM0 | R/W | 0 | UART 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 UART, Baud Rate is variable<br>10: Mode2: 9 bit UART, Baud Rate=$F_{SYSCLK}$/32 or /64<br>11: Mode3: 9 bit UART, 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 UART 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 | – | UART transmit and receive data. Transmit data is written to this location and receive data is read from this location, but the paths are independent. |
| 9Ah | **WGCON** | 7~6 | WGRES | R/W | 00 | WG reset time select ($T_{FRC}$ =60.28 ns)<br>0: 7*$T_{FRC}$ = 54.0us<br>1: 11*$T_{FRC}$ = 84.9us<br>2: 21*$T_{FRC}$ = 162.0us<br>3: 37*$T_{FRC}$ = 285.5us |
| | | 5~4 | WGMOD | R/W | 00 | WG Mode select<br>0xb：Continue Mode. HW sends data code and reset code, repeating continuously, will not clear WGEN automatically.<br>10b：One Cycle Mode. HW will automatically clear WGEN after sending data code and reset code<br>11b：One Cycle Mode. HW will automatically clear WGEN after sending data code. |
| | | 3~1 | WGTDS | R/W | 000 | WG Transmit data length select<br>0： 3 bytes (1 LED)<br>1： 6 bytes (2 LED)<br>2： 9 bytes (3 LED) |

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| | | | | | | 3：12 bytes (4 LED)<br>4：15 bytes (5 LED)<br>5：18 bytes (6 LED)<br>6：21 bytes (7 LED)<br>7：24 bytes (8 LED) |
| | | 0 | WGEN | R/W | 0 | WG enable and output to P3.4, H/W will automatically clear WGEN in One Cycle mode<br>0：WG disable, P3.4 as GPIO。 1：WG enable, P3.4 as WG output. |
| 9Bh | **WGCON2** | 7~5 | WGPRD | R/W | 000 | WG period of each bit select ($T_{FRC}$ =60.28 ns)<br>0: 17*TFRC = 1025ns<br>1: 18*TFRC = 1085ns<br>2: 19*TFRC = 1145ns<br>3: 20*TFRC = 1206ns<br>4: 21*TFRC = 1266ns<br>5: 22*TFRC = 1326ns<br>6: 23*TFRC = 1386ns<br>7: 24*TFRC = 1447ns |
| | | 4~2 | WGT1H | R/W | 000 | WG code 1 high level time select ($T_{FRC}$ =60.28 ns)<br>0: 10*$T_{FRC}$ = 603ns<br>1: 11*$T_{FRC}$ = 663ns<br>2: 12*$T_{FRC}$ = 723ns<br>3: 13*$T_{FRC}$ = 784ns<br>4: 14*$T_{FRC}$ = 844ns<br>5: 15*$T_{FRC}$ = 904ns<br>6: 16*$T_{FRC}$ = 964ns<br>7: 17*$T_{FRC}$ =1025ns |
| | | 1~0 | WGT0H | R/W | 00 | WG code 0 high level time select ($T_{FRC}$ =60.28 ns)<br>0: 4*$T_{FRC}$ = 241ns<br>1: 5*$T_{FRC}$ = 301ns<br>2: 6*$T_{FRC}$ = 362ns<br>3: 7*$T_{FRC}$ = 422ns |
| 9Ch | **DACON** | 7 | SVRF | R/W | 0 | Select comparator reference voltage level<br>0: VDDA<br>1: VBGO (define by VBGSEL AEh.1~0) |
| | | 6~0 | SVL | R/W | 0 | Select DAC output voltage<br>reference source can be selected as VDDA or VBGO<br>000_0000: 0/128 * reference source<br>000_0001: 1/128 * reference source<br>…<br>111_1101: 125/128 * reference source<br>111_1110: Reserved<br>111_1111: Reserved |
| 9Dh | **CMPCON** | 7 | CMPPD | R/W | 1 | Comparator & DAC power down enable control<br>0: disable Comparator & DAC power down<br>1: enable Comparator & DAC power down |
| | | 6 | CMPOX | R/W | 1 | Comparator original output (CMPOX) status<br>0: VCMPP < VCMPN<br>1: VCMPP > VCMPN or CMPPD =1 |
| | | 5 | CMPOE | R/W | 0 | Comparator output (CMPO) signal output to P2.1<br>0: disable<br>1: enable, P2MOD1 should be set to 10b |
| | | 4 | CMPINV | R/W | 0 | Comparator de-bounce output invert select<br>0: no invert<br>1: invert |
| | | 3~2 | CMPTRIG | R/W | 00 | Comparator interrupt trigger mode<br>00: Rising edge<br>01: Falling edge<br>10: Both edge<br>11: High level |
| | | 1`0 | CMPDBS | R/W | 00 | Comparator original output (CMPOX) de-bounce time<br>00: none<br>01: 4 Fsys |

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| | | | | | | 10: 8 Fsys<br>11: 16 Fsys |
| 9Eh | **CMPPNS** | 7 | SCMPN | R/W | 1 | Comparator CMPN source select<br>0: Comparator CMPN source is external input (CINx)<br>1: Comparator CMPN source is DAC output |
| | | 6~4 | SCIN | R/W | 111 | Comparator CMPN external input select<br>000: Comparator CMPN external input is CIN1 (P1.6)<br>001: Comparator CMPN external input is CIN2 (P1.5)<br>010: Comparator CMPN external input is CIN3 (P0.7)<br>011: Comparator CMPN external input is CIN4 (P0.1)<br>1xx: No connect |
| | | 3 | OPOF | R/W | 0 | OPA output (OPO) connect to Comparator CMPP<br>0: Comparator CMPP source is OPA output (OPO)<br>1: Comparator CMPP source is external input (CIPx) |
| | | 2~0 | SCIP | R/W | 111 | Comparator CMPP external input select<br>000: Comparator CMPP external input is CIP1 (PA1)<br>001: Comparator CMPP external input is CIP2 (PA2)<br>010: Comparator CMPP external input is CIP3 (PB6)<br>011: Comparator CMPP external input is CIP4 (PD1)<br>1xx: No connect |
| 9Fh | **OPCON** | 7 | OPAPD | R/W | 1 | OPA power down enable control<br>0: disable OPA power down<br>1: enable OPA power down |
| | | 6~5 | SOPP | R/W | 11 | select OPP input source<br>00: OPP input source is P3.3<br>01: OPP input source is P1.4<br>10: OPP input source is P1.7<br>11: OPP input source is VSS |
| | | 4 | SOTV | R/W | 0 | OPN input voltage selection in Comparator mode<br>0: VSS<br>1: VBG (voltage level is selected by VBGSEL) |
| | | 3~2 | SOPG | R/W | 11 | select OPA gain<br>00: 1X<br>01: 20X<br>10: 50X<br>11: 100X |
| | | 1~0 | SOPM | R/W | 00 | select OPA operating mode<br>00: Normal Mode<br>01: Reserved<br>10: Reserved<br>11: Comparator Mode |
| A0h | **P2** | 7~2 | P2.7~P2.2 | R/W | FFh | P2.7~P2.2 have no pin out, so these bits are used as general purpose register |
| | | 1~0 | P2.1~P2.0 | R/W | 11 | P2.1~P2.0 data |
| A1h | **PWMCON** | 7~6 | PWM1CKS | R/W | 00 | PWM1 clock source<br>00: $F_{SYSCLK}$<br>01: $F_{SYSCLK}$<br>10: FRC<br>11: FRCx2 (Vcc>2.7V) |
| | | 5 | PWM1EN | R/W | 0 | PWM1~6 Enable. 0: PWM1~6 Disable, 1: PWM1~6 Enable |
| | | 4 | PWM0EN | R/W | 0 | PWM0 Enable. 0: PWM0 Disable, 1: PWM0 Enable |
| | | 3~2 | PWM0CKS | R/W | 00 | PWM0 clock source<br>00: $F_{SYSCLK}$<br>01: $F_{SYSCLK}$<br>10: FRC<br>11: FRCx2 (Vcc>2.7V) |
| | | 1 | PWM0NMSK | R/W | 0 | PWM0N mask data. If CLRPWM0=1 and PMW0MSKE=1, PWM0N will output this mask data. |
| | | 0 | PWM0PMSK | R/W | 0 | PWM0P mask data. If CLRPWM0=1 and PMW0MSKE=1, PWM0P will output this mask data. |
| A2h | **P1MODL** | 7~6 | P1MOD3 | R/W | 01 | P1.3 Pin Control |

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| | | | | | | 00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3 |
| | | 5~4 | P1MOD2 | R/W | 01 | P1.2 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3 |
| | | 3~2 | P1MOD1 | R/W | 01 | P1.1 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3 |
| | | 1~0 | P1MOD0 | R/W | 01 | P1.0 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3 |
| A3h | **P1MODH** | 7~6 | P1MOD7 | R/W | 01 | P1.7 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3 |
| | | 5~4 | P1MOD6 | R/W | 01 | P1.6 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3 |
| | | 3~2 | P1MOD5 | R/W | 01 | P1.5 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3 |
| | | 1~0 | P1MOD4 | R/W | 01 | P1.4 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3 |
| A4h | **P3MODL** | 7~6 | P3MOD3 | R/W | 01 | P3.3 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3 |
| | | 5~4 | P3MOD2 | R/W | 01 | P3.2 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3 |
| | | 3~2 | P3MOD1 | R/W | 01 | P3.1 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3 |
| | | 1~0 | P3MOD0 | R/W | 01 | P3.0 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3 |
| A5h | **P3MODH** | 7~6 | P3MOD7 | R/W | 01 | 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 |
| A6h | **PWMOE0** | 7 | PWM1OE3 | R/W | 0 | PWM1 output control<br>0: Disable 1: PWM1 enable and output to P1.2 |
| | | 6 | PWM1OE2 | R/W | 0 | PWM1 output control<br>0: Disable 1: PWM1 enable and output to P0.6 |
| | | 5 | PWM1OE1 | R/W | 0 | PWM1 output control<br>0: Disable 1: PWM1 enable and output to P0.4 |
| | | 4 | PWM1OE0 | R/W | 0 | PWM1 output control<br>0: Disable 1: PWM1 enable and output to P0.2 |
| | | 3 | PWM0NOE1 | R/W | 0 | PWM0N output control<br>0: Disable 1: PWM0N enable and output to P3.6 |
| | | 2 | PWM0POE1 | R/W | 0 | PWM0P output control<br>0: Disable 1: PWM0P enable and output to P3.5 |
| | | 1 | PWM0NOE0 | R/W | 0 | PWM0N output control<br>0: Disable 1: PWM0N enable and output to P0.4 |
| | | 0 | PWM0POE0 | R/W | 0 | PWM0P output control<br>0: Disable 1: PWM0P enable and output to P0.3 |
| A7h | **PWMCON2** | 7 | PWM0MOD | R/W | 0 | PWM0 mode select<br>0: Normal mode<br>1: Half-bridge mode |
| | | 6 | PWM0MSKE | R/W | 0 | PWM0 mask output enable<br>0: Disable<br>1: Enable, PWM0P/PWM0N output data by PWM0PMSK/PWM0NMSK while CLRPWM0=1 |
| | | 5~4 | PWM0OM | R/W | 00 | PWM0 output mode select<br>00: Mode0<br>01: Mode1<br>10: Mode2<br>11: Mode3 |

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| | | 3~0 | PWM0DZ | R/W | 0000 | PWM0 dead zone (Dead zone is prohibited in half-bridge mode)<br>0000: 0 x $T_{PWMCLK}$<br>0001: 1 x $T_{PWMCLK}$<br>…<br>1111: 15 x $T_{PWMCLK}$ |
| 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 (UART) 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 & Halt/Stop 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 & Halt/Stop mode wake up capability |
| A9h | **INTE1** | 7 | PWMIE | R/W | 0 | Set 1 to enable PWM0/PWM1~PWM6 interrupt |
| | | 6 | CMPIE | R/W | 0 | Set 1 to enable CMP interrupt |
| | | 5 | LVDIE | R/W | 0 | Set 1 to enable LVD interrupt |
| | | 4 | SPI2CE | R/W | 0 | Set 1 to enable SPI/$I^2C$ interrupt |
| | | 3 | ADTKIE | R/W | 0 | Set 1 to enable ADC/TK Interrupt |
| | | 2 | EX2 | R/W | 0 | Set 1 to enable external INT2 pin Interrupt & Halt/Stop mode wake up capability |
| | | 1 | PXIE | R/W | 0 | Set 1 to enable Port0/Port1/Port2/Port3 Pin Change Interrupt |
| | | 0 | TM3IE | R/W | 0 | Set 1 to enable Timer3 Interrupt |
| AAh | **ADCDL** | 7~4 | ADCDL | R | – | ADC data bit 3~0 |
| | | 0 | PWRDEC | W | 0 | ROM parameter settings for high temperature writing. |
| ABh | **ADCDH** | 7~0 | ADCDH | R | – | ADC data bit 11~4 |
| ACh | **TKADL** | 7~0 | TKADL | R | – | |
| ADh | **TKCON** | 7 | TKAPD | R/W | 1 | Touch Key A power down. |
| | | 6 | TKAEOC | R | 1 | Touch Key A end of conversion. |
| | | 5 | TKASOC | R/W | 0 | Touch Key A start, HW clear while end of conversion. |
| | | 4 | - | - | 0 | Reserved |
| | | 3 | TKBPD | R/W | 1 | Touch Key B power down. |
| | | 2 | TKBEOC | R | 1 | Touch Key B end of conversion. |
| | | 1 | TKBSOC | R/W | 0 | Touch Key B start, HW clear while end of conversion. |
| | | 0 | - | - | 0 | Reserved |

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| AEh | **CHSEL** | 7~3 | ADCHS | R/W | 11111 | ADC channel select.<br>00000: AD0 (P0.4)<br>00001: AD1 (P0.3)<br>00010: AD2 (P1.0)<br>00011: AD3 (P1.1)<br>00100: AD4 (P3.3)<br>00101: AD5 (P3.2)<br>00110: AD6 (P3.0)<br>00111: AD7 (P3.1)<br>01000: AD8 (P3.4)<br>01001: AD9 (P1.7)<br>01010: AD10 (P0.7)<br>01011: AD11 (P0.5)<br>01100: VBG<br>01101: OPO<br>01110: $V_{SS}$<br>01111: AD15 (P0.2)<br>10000: AD16 (P1.3)<br>10001: AD17 (P1.4)<br>10010: AD18 (P1.5)<br>10011: AD19 (P1.6)<br>10100: AD20 (P0.0)<br>10101: AD21 (P0.1)<br>10110: Reserved<br>10111: $V_{CC}/4$<br>others: Reserved |
| | | 2 | ADCVREFS | R/W | 0 | ADC reference voltage<br>0: $V_{CC}$<br>1: VBG |
| | | 1~0 | VBGSEL | R/W | 00 | VBG voltage select, When ADCVREF is selected as VBG, VBGSEL is prohibited from using 1.20V.<br>00: 1.20V<br>01: 2.49V (need VCC>2.8V)<br>10: Reserved<br>11:Reserved |
| AFh | **P1LOE** | 3~0 | P1LOE | R/W | 0000 | P1.5~P1.2 (COM3~0) LCD 1/2 bias output enable. |
| B0h | **P3** | 7~0 | P3 | R/W | FFh | Port3 data |
| B1h | **TKAREFC** | 6~0 | TKAREFC | R/W | 00h | Touch Key A reference clock capacitor select. |
| B2h | **TKADH** | 5~0 | TKADH | R | - | Touch Key A Data bit13~bit8 |
| B3h | **TKCHS** | 5~4 | TKBCHS | R/W | 11 | Touch Key B channel select.<br>00:TK4 (P1.1)<br>01:TK5 (P0.5)<br>10:TK6 (P0.6)<br>11: TK7 (P0.7) (Ref) |
| | | 1~0 | TKACHS | R/W | 11 | Touch Key A channel select.<br>00: TK0 (P3.7)<br>01: TK1 (P2.1)<br>10: TK2 (P2.0)<br>11: TK3 (P1.2) |
| B4h | **TKATMRL** | 7~0 | TKATMRL | R/W | FF | Touch Key A reference counter data 7~0 |
| B5h | **TKATMRH** | 5~0 | TKATMRH | R/W | 00 | Touch Key A reference counter data 13~8 |
| B6h | **PWMOE1** | 7 | PWM4OE3 | R/W | 0 | PWM4 output control<br>0: Disable 1: PWM4 enable and output to P3.6 |
| | | 6 | PWM4OE2 | R/W | 0 | PWM4 output control<br>0: Disable 1: PWM4 enable and output to P1.5 |
| | | 5 | PWM4OE1 | R/W | 0 | PWM4 output control<br>0: Disable 1: PWM4 enable and output to P0.4 |

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| | | 4 | PWM4OE0 | R/W | 0 | PWM4 output control<br>0: Disable 1: PWM4 enable and output to P0.0 |
| | | 3 | PWM3OE1 | R/W | 0 | PWM3 output control<br>0: Disable 1: PWM3 enable and output to P3.4 |
| | | 2 | PWM3OE0 | R/W | 0 | PWM3 output control<br>0: Disable 1: PWM3 enable and output to P1.0 |
| | | 1 | PWM2OE1 | R/W | 0 | PWM2 output control<br>0: Disable 1: PWM2 enable and output to P3.6 |
| | | 0 | PWM2OE0 | R/W | 0 | PWM2 output control<br>0: Disable 1: PWM2 enable and output to P1.1 |
| B7h | **PWMOE2** | 7 | MSDASEL | R/W | 0 | Master $I^2C$ SDA select<br>0: P3.5 as Master $I^2C$ SDA<br>1: P1.6 as Master $I^2C$ SDA |
| | | 6 | MSCLSEL | R/W | 0 | Master $I^2C$ SCL select<br>0: P1.3 as Master $I^2C$ SCL<br>1: P0.2 as Master $I^2C$ SCL |
| | | 5 | PWM6OE2 | R/W | 0 | PWM6 output control<br>0: Disable 1: PWM6 enable and output to P1.3 |
| | | 4 | PWM6OE1 | R/W | 0 | PWM6 output control<br>0: Disable 1: PWM6 enable and output to P0.7 |
| | | 3 | PWM6OE0 | R/W | 0 | PWM6 output control<br>0: Disable 1: PWM6 enable and output to P0.3 |
| | | 2 | PWM5OE2 | R/W | 0 | PWM5 output control<br>0: Disable 1: PWM5 enable and output to P1.4 |
| | | 1 | PWM5OE1 | R/W | 0 | PWM5 output control<br>0: Disable 1: PWM5 enable and output to P0.6 |
| | | 0 | PWM5OE0 | R/W | 0 | PWM5 output control<br>0: Disable 1: PWM5 enable and output to P0.1 |
| B8h | **IP** | 5 | PT2 | R/W | 0 | Timer2 Interrupt Priority Low bit |
| | | 4 | PS | R/W | 0 | Serial Port (UART) 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 (UART) 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 | PWM0/PWM1 Interrupt Priority Low bit |
| | | 6 | PCMP | R/W | 0 | CMP Interrupt Priority Low bit |
| | | 5 | PLVD | R/W | 0 | LVD Interrupt Priority Low bit |
| | | 4 | PSPI2C | R/W | 0 | SPI/$I^2C$ Interrupt Priority Low bit |
| | | 3 | PADTKI | R/W | 0 | ADC/TK Interrupt Priority Low bit |
| | | 2 | PX2 | R/W | 0 | External INT2 Pin Interrupt Priority Low bit |
| | | 1 | PPX | R/W | 0 | Port0~Port3 pin change Interrupt Priority Low bit |
| | | 0 | PT3 | R/W | 0 | Timer3 Interrupt Priority Low bit |

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| BBh | **IP1H** | 7 | PPWMH | R/W | 0 | PWM0/PWM1 Interrupt Priority High bit |
| | | 6 | PCMPH | R/W | 0 | CMP Interrupt Priority High bit |
| | | 5 | PLVDH | R/W | 0 | LVD Interrupt Priority High bit |
| | | 4 | PI2CH | R/W | 0 | SPI/$I^2C$ Interrupt Priority High bit |
| | | 3 | PADTKIH | R/W | 0 | ADC/TK Interrupt Priority High bit |
| | | 2 | PX2H | R/W | 0 | External INT2 Pin Interrupt Priority High bit |
| | | 1 | PPXH | R/W | 0 | Port0~Port3 Interrupt Priority High bit |
| | | 0 | PT3H | R/W | 0 | Timer3 Interrupt Priority High bit |
| BCh | **SPCON** | 7 | SPEN | R/W | 0 | SPI Enable |
| | | 6 | MSTR | R/W | 0 | Master Mode Enable |
| | | 5 | CPOL | R/W | 0 | SPI Clock Polarity |
| | | 4 | CPHA | R/W | 0 | SPI Clock Phase |
| | | 2 | LSBF | R/W | 0 | LSB First. 0: MSB first, 1: LSB first |
| | | 1~0 | SPCR | R/W | 00 | SPI Clock Rate. 0: $F_{SYS}$/2, 1: $F_{SYS}$/4, 2: $F_{SYS}$/8, 3: $F_{SYS}$/16 |
| BDh | **SPSTA** | 7 | SPIF | R/W | 0 | SPI Interrupt Flag |
| | | 6 | WCOL | R/W | 0 | Write Collision Interrupt Flag |
| | | 4 | RCVOVF | R/W | 0 | Receive Buffer Overrun Flag |
| | | 3 | RCVBF | R/W | 0 | Receive Buffer Full Flag |
| | | 2 | SPBSY | R | 0 | SPI Busy Flag |
| BEh | **SPDAT** | 7~0 | SPDAT | R/W | 00h | SPI Transmit and Receive Data |
| BFh | **BOOTV** | 2 | RSTV | R/W | 1 | Reset Vector after Reset. No change while Reset (except POR)<br>0: Reset Vector = 0x0000<br>1: Reset Vector define by BOOTVR (CFG.BOOTV) |
| | | 1~0 | BOOTVR | R | - | Load from CFG.BOOTV after POR.<br>00: Reset Vector = 0x3800, BOOT Area Size = 2K<br>01: Reset Vector = 0x3000, BOOT Area Size = 4K<br>1x: Reset Vector = 0x0000, no BOOT Area |
| C1h | **TKBTMRL** | 7~0 | TKBTMRL | R/W | FFh | Touch Key B reference counter data 7~0 |
| C2h | **TKBTMRH** | 5~0 | TKBTMRH | R/W | 00h | Touch Key B reference counter data 13~8 |
| C3h | **TKBKCP** | 6~0 | TKBKCP | R/W | 00h | Touch Key B reference capacitor select (TK7) |
| C4h | **TKBREFC** | 6~0 | TKBREFC | R/W | 00h | Touch Key B reference clock capacitor select |
| C5h | **P0WKUP** | 7~0 | P0WKUP | R/W | 00h | P0.7~P0.0 pin individual Wake-up/Interrupt enable control<br>0: Disable;<br>1: Enable. |
| C6h | **P2WKUP** | 7~0 | P2WKUP | R/W | 00h | P2.7~P2.0 pin individual Wake-up/Interrupt enable control<br>0: Disable;<br>1: Enable. |
| C7h | **P3WKUP** | 7~0 | P3WKUP | R/W | 00h | P3.7~P3.0 pin individual Wake-up/Interrupt enable control<br>0: Disable;<br>1: Enable. |
| 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 |

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| | | 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 | **IAPCON** | 7~0 | IAPCON | W | – | Write 47h or 74h to set IAPWE flag; Write 47h can write 1 byte at once, write 74h can write 2 bytes at once. Write other value to clear IAPWE flag. It is recommended to clear it immediately after IAP write.<br>Write A1h to set INFOWE flag; write other value to clear INFOWE flag. It is recommended to clear it immediately after IAP write.<br>Write E2h to set EEPWE flag; write other value to clear EEPWE flag. It is recommended to clear it immediately after EEPROM write. |
| | | 7 | IAPWE | R | 0 | Flag indicates Flash memory can be written by IAP or not<br>0: IAP Write disable<br>1: IAP Write enable |
| | | 6 | IAPTO | R | 0 | Time-Out flag of IAP write/EEPROM write/INFO write.<br>Set by H/W when IAP or EEPROM or INFO write Time-out occurs. Cleared this flag by H/W when IAPWE=0 or EEPWE=0 or INFOWE=0. |
| | | 5 | EEPWE | R | 0 | Flag indicates EEPROM memory can be written or not<br>0: EEPROM Write disable<br>1: EEPROM Write enable |
| | | 4 | INFOWE | R | 0 | Flag indicates INFO memory can be written or not<br>0: INFO IAP Write disable<br>1: INFO IAP Write enable |
| 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 | 00h | PWM0 duty high byte |
| D2h | **PWM0DL** | 7~0 | PWM0DL | R/W | 00h | PWM0 duty low byte |
| D3h | **PWM1DH** | 7~0 | PWM1DH | R/W | 00h | PWM1 duty high byte |
| D4h | **PWM1DL** | 7~0 | PWM1DL | R/W | 00h | PWM1 duty low byte |
| D5h | **PWM2DH** | 7~0 | PWM2DH | R/W | 00h | PWM2 duty high byte |
| D6h | **PWM2DL** | 7~0 | PWM2DL | R/W | 00h | PWM2 duty low byte |
| D7h | **CFGOP** | 4~0 | OPTRIM | R/W | - | OP trimming value |
| D8h | **CLKCON** | 5 | STPSCK | R/W | 1 | Set 1 to stop Slow clock in Stop Mode. |
| | | 4 | STPPCK | R/W | 0 | Set 1 to stop UART/Timer0/1/2 clock in Idle mode for current reducing. |

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| | | 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 |
| DAh | **PWM0PRDL** | 7~0 | PWM0PRDL | R/W | FFh | PWM0 period low byte |
| DBh | **PWM1PRDH** | 7~0 | PWM1PRDH | R/W | FFh | PWM1/PWM2/PWM3/PWM4/PWM5/PWM6 period high byte |
| DCh | **PWM1PRDL** | 7~0 | PWM1PRDL | R/W | FFh | PWM1/PWM2/PWM3/PWM4/PWM5/PWM6 period low byte |
| DDh | **PWM3DH** | 7~0 | PWM3DH | R/W | 00h | PWM3 duty high byte |
| DEh | **PWM3DL** | 7~0 | PWM3DL | R/W | 00h | PWM3 duty low byte |
| DFh | **UARTCON** | 7 | UARTBRS | R/W | 0 | UART Baud Rate Source Select<br>0: 8051 default Baud Rate source select<br>1: UART Baud Rate select as UARTBRP |
| | | 6~0 | UARTBRP | R/W | 00h | Define UART Baud Rate Prescaler<br>UART Baud Rate = Fsys/32/UARTBRP |
| E0h | **ACC** | 7~0 | ACC | R/W | 00h | Accumulator |
| E1h | **MICON** | 7 | MIEN | R/W | 0 | Master $I^2C$ enable<br>0: disable<br>1: enable |
| | | 6 | MIACKO | R/W | 0 | When Master $I^2C$ receive data, send acknowledge to $I^2C$ Bus<br>0: ACK to slave device<br>1: NACK to slave device |
| | | 5 | MIIF | R/W | 0 | Master $I^2C$ Interrupt flag<br>0: write 0 to clear it<br>1: Master $I^2C$ transfer one byte complete |
| | | 4 | MIACKI | R | – | When Master $I^2C$ transfer, acknowledgement form $I^2C$ bus (read only)<br>0: ACK received<br>1: NACK received |
| | | 3 | MISTART | R/W | 0 | Master $I^2C$ Start bit<br>1: start $I^2C$ bus transfer |
| | | 2 | MISTOP | R/W | 1 | Master $I^2C$ Stop bit<br>1: send STOP signal to stop $I^2C$ bus |
| | | 1~0 | MICR | R/W | 00 | Master $I^2C$ (SCL) clock frequency selection<br>00: Fsys/4 (ex. If Fsys=16MHz, $I^2C$ clock is 4M Hz)<br>01: Fsys/16 (ex. If Fsys=16MHz, $I^2C$ clock is 1M Hz)<br>10: Fsys/64 (ex. If Fsys=16MHz, $I^2C$ clock is 250K Hz)<br>11: Fsys/256 (ex. If Fsys=16MHz, $I^2C$ clock is 62.5K Hz) |
| E2h | **MIDAT** | 7~0 | MIDAT | R/W | 00 | Master $I^2C$ data shift register<br>(W): After Start and before Stop condition, write this register will resume transmission to $I^2C$ bus<br>(R): After Start and before Stop condition, read this register will resume receiving from $I^2C$ bus |

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| E3h | **LVRCON** | 5 | PORPD | R/W | 0 | POR Power Down. 0: POR Enable, 1: POR Disable |
| | | 4 | LVRPD | R/W | 0 | LVR Power Down. 0: LVR Enable, 1: LVR Disable |
| | | 3~0 | LVRSEL | R/W | 0000 | Low Voltage Reset (LVR) select. (step=0.14V)<br>0000: Set LVR at 2.05V<br>0001: Set LVR at 2.19V<br>0010: Set LVR at 2.33V<br>0011: Set LVR at 2.47V<br>0100: Set LVR at 2.61V<br>0101: Set LVR at 2.75V<br>0110: Set LVR at 2.89V<br>0111: Set LVR at 3.03V<br>1000: Set LVR at 3.17V<br>1001: Set LVR at 3.31V<br>1010: Set LVR at 3.45V<br>1011: Set LVR at 3.59V<br>1100: Set LVR at 3.73V<br>1101: Set LVR at 3.87V<br>1110: Set LVR at 4.01V<br>1111: Set LVR at 4.15V |
| E4h | **LVDCON** | 7 | LVDM | R/W | 0 | 0: VCC < VLVD (LVDIF = 1 while LVDO = 1)<br>1: VCC > VLVD (LVDIF = 1 while LVDO = 0) |
| | | 6 | LVDO | R | 0 | LVD real-time Output |
| | | 5 | LVDHYS | R/W | 0 | LVD Hysteresis Enable. 0: LVD Hysteresis disable, 1: LVD Hysteresis enable |
| | | 4 | LVDPD | R/W | 0 | LVD Power Down. 0: LVD Enable, 1: LVD Disable |
| | | 3~0 | LVDSEL | R/W | 0h | Low Voltage Detect (LVD) select. (step=0.14V)<br>0000: Set LVD at 2.05V<br>0001: Set LVD at 2.19V<br>0010: Set LVD at 2.33V<br>0011: Set LVD at 2.47V<br>0100: Set LVD at 2.61V<br>0101: Set LVD at 2.75V<br>0110: Set LVD at 2.89V<br>0111: Set LVD at 3.03V<br>1000: Set LVD at 3.17V<br>1001: Set LVD at 3.31V<br>1010: Set LVD at 3.45V<br>1011: Set LVD at 3.59V<br>1100: Set LVD at 3.73V<br>1101: Set LVD at 3.87V<br>1110: Set LVD at 4.01V<br>1111: Set LVD at 4.15V |
| E5h | **EFTCON** | 7 | EFT2CS | R/W | 0 | EFT2 Detector enable<br>0: Disable EFT2<br>1: Enable EFT2 |
| | | 6 | EFT1CS | R/W | 0 | EFT1 Detector enable<br>0: Disable EFT1<br>1: Enable EFT1 |
| | | 5~4 | EFT1S | R/W | 00 | EFT1 Detector sensitivity adjustment |
| | | 3 | EFTSLOW | R/W | 0 | Force SYSCLK to SLOWCLK while EFT detected<br>0: Disable<br>1: Enable |
| | | 2 | EFTWCPU | R/W | 0 | CPU enter Wait state while EFT detected<br>0: Disable<br>1: Enable |
| | | 1 | EFTWOUT | R/W | 0 | EFTWAIT output to pin<br>0: P3.6 = normal I/O |

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| | | | | | | 1: P3.6 = EFTWAIT |
| | | 0 | CKHLDE | R/W | 0 | clock hold enable<br>0: Disable<br>1: Enable |
| E6h | **EXA** | 7~0 | EXA | R/W | 00h | Expansion accumulator |
| E7h | **EXB** | 7~0 | EXB | R/W | 00h | Expansion B register |
| E9h | **PWM4DH** | 7~0 | PWM4DH | R/W | 00h | PWM4 duty high byte |
| EAh | **PWM4DL** | 7~0 | PWM4DL | R/W | 00h | PWM4 duty low byte |
| EBh | **PWM5DH** | 7~0 | PWM5DH | R/W | 00h | PWM5 duty high byte |
| ECh | **PWM5DL** | 7~0 | PWM5DL | R/W | 00h | PWM5 duty low byte |
| EDh | **PWM6DH** | 7~0 | PWM6DH | R/W | 00h | PWM6 duty high byte |
| EEh | **PWM6DL** | 7~0 | PWM6DL | R/W | 00h | PWM6 duty low byte |
| EFh | **PWRCON** | 7 | IVCPD | R/W | 1 | IVC(build-in VDD regulator) power down<br>0: IVC Enable ($V_{DD}$ = IVC's voltage)<br>1: IVC Disable ($V_{DD}$ = $V_{CC}$) |
| | | 6~5 | IVCVS | R/W | 11 | IVC Voltage select<br>00: 1.70V<br>01: 1.95V<br>10: 2.20V<br>11: 2.45V (recommended) |
| | | 3 | WARMTIME | R/W | 0 | Warm-up time for wake-up from Halt/Stop mode<br>0: 128 Clock<br>1: 64 Clock |
| 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 |
| F3h | **CRCIN** | 7~0 | CRCIN | W | – | CRC input data |
| F5h | **CFGBG** | 4~0 | BGTRIM | R/W | – | VBG trimming value |
| F6h | **CFGWL** | 6~0 | FRCTRIM | R/W | – | FRC frequency adjustment<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 | – | Set 1 to reduce the chip's power consumption at 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 |
| | | 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 | 11 | IAP write/EEPROM write/INFO write watchdog timer enable<br>00: Disable<br>01: wait 1.6ms trigger watchdog time-out flag<br>10: wait 3.1ms trigger watchdog time-out flag<br>11: wait 12.5ms 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 |

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| 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. |
| | | 5 | - | R/W | 0 | Reserved |
| | | 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 | CLRPWM0 | R/W | 1 | PWM0 clear enable<br>0: PWM0 is running<br>1: PWM0 is cleared and held |
| | | 2 | CLRPWM1 | R/W | 1 | PWM1/PWM2/PWM3/PWM4/PWM5/PWM6 clear enable<br>0: PWM1/PWM2/PWM3/PWM4/PWM5/PWM6 is running<br>1: PWM1/PWM2/PWM3/PWM4/PWM5/PWM6 is cleared and held |
| | | 1 | LDOCOUT | R/W | 0 | 0: P3.7 as normal IO<br>1: LDOC output to P3.7 (active while XRSTE=0) |
| | | 0 | DPSEL | R/W | 0 | Active DPTR Select |

| Adr | Flash | Bit# | Bit Name | Description |
|---|---|---|---|---|
| 3FF9h | **CFGOP** | 4~0 | OPTRIM | OP adjustment. |
| 3FFBh | **CFGBG** | 4~0 | BGTRIM | VBG adjustment. VBG is trimmed to 1.20V in chip manufacturing. |
| 3FFDh | **CFGWL** | 6~0 | FRCTRIM | FRC frequency adjustment. FRC is trimmed to 16.588 MHz in chip manufacturing. |
| 3FFFh | **CFGWH** | 7 | PROT | Flash Code Protect, 1=Protect |
| | | 6 | XRSTE | External Pin Reset Enable, 1=Enable. |
| | | 5 | - | Reserved |
| | | 4 | HVS | 0: ROM speed up function enable for improving write speed at high voltage<br>1: ROM speed up function disable |
| | | 3~2 | BOOTV | Reset Vector after POR<br>00: Reset Vector = 0x3800, BOOT mode enable, BOOT Area Size = 2K<br>01: Reset Vector = 0x3000, BOOT mode enable, BOOT Area Size = 4K<br>1x: Reset Vector = 0x0000, BOOT mode disable, no BOOT Area. |
| | | 1~0 | - | Reserved |

# INSTRUCTION SET

Instructions are 1, 2 or 3 bytes long as listed in the 'byte' column below. Each instruction takes 2~32 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 | 8 | 93 |
| MOVC A,@A+PC | Move code byte relative PC to A | 1 | 8 | 83 |
| MOVX A,@Ri | Move external data(A8) to A | 1 | 8 | E2-E3 |
| MOVX A,@DPTR | Move external data(A16) to A | 1 | 8 | E0 |
| MOVX @Ri,A | Move A to external data(A8) | 1 | 8 | F2-F3 |
| MOVX @DPTR,A | Move A to external data(A16) | 1 | 8 | 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 (+2) | 11-F1 |
| LCALL addr 16 | Long jump to subroutine | 3 | 4 (+2) | 12 |
| RET | Return from subroutine | 1 | 4 (+2) | 22 |
| RETI | Return from interrupt | 1 | 4 (+2) | 32 |
| AJMP addr 11 | Absolute jump unconditional | 2 | 4 (+2) | 01-E1 |
| LJMP addr 16 | Long jump unconditional | 3 | 4 (+2) | 02 |
| SJMP rel | Short jump (relative address) | 2 | 4 (+2) | 80 |
| JC rel | Jump on carry = 1 | 2 | 4 (or 6) | 40 |
| JNC rel | Jump on carry = 0 | 2 | 4 (or 6) | 50 |
| JB bit,rel | Jump on direct bit = 1 | 3 | 4 (or 6) | 20 |
| JNB bit,rel | Jump on direct bit = 0 | 3 | 4 (or 6) | 30 |
| JBC bit,rel | Jump on direct bit = 1 and clear | 3 | 4 (or 6) | 10 |
| JMP @A+DPTR | Jump indirect relative DPTR | 1 | 4 (+2) | 73 |
| JZ rel | Jump on accumulator = 0 | 2 | 4 (or 6) | 60 |
| JNZ rel | Jump on accumulator … 0 | 2 | 4 (or 6) | 70 |
| CJNE A,dir,rel | Compare A,direct, jump not equal relative | 3 | 4 (or 6) | B5 |
| CJNE A,#data,rel | Compare A,immediate, jump not equal relative | 3 | 4 (or 6) | B4 |
| CJNE Rn,#data,rel | Compare register,immediate, jump not equal relative | 3 | 4 (or 6) | B8-BF |
| CJNE @Ri,#data,rel | Compare indirect,immediate, jump not equal relative | 3 | 4 (or 6) | B6-B7 |
| DJNZ Rn,rel | Decrement register, jump not zero relative | 2 | 4 (or 6) | D8-DF |
| DJNZ dir,rel | Decrement direct byte, jump not zero relative | 3 | 4 (or 6) | 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 | |
| Output current high per all PIN | –80 | mA |
| Output current low per all PIN | +150 | |
| Maximum Operating Voltage | 5.5 | V |
| Operating temperature | –40 ~ +105 | °C |
| Storage temperature | –65 ~ +150 | |

## 2. DC Characteristics ($T_A$=25 ℃, $V_{CC}$=2.2V ~ 5.5V) (TBD)

| Parameter | Symbol | Conditions | | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|---|
| Operating Voltage | $V_{CC}$ | $F_{SYS}$=16.588 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$V_{CC}$ | 6 | 12 | – | mA |
| | | | $V_{CC}$=3V, $V_{OH}$=0.9$V_{CC}$ | 2.5 | 5 | – | |
| I/O Port Sink Current | $I_{OL}$ | All Output, | $V_{CC}$=5V, $V_{OL}$=0.1$V_{CC}$ | 41 | 82 | – | mA |
| | | | $V_{CC}$=3V, $V_{OL}$=0.1$V_{CC}$ | 18 | 36 | – | |
| Supply Current | $I_{DD}$ | FAST mode $V_{CC}$=5V | FRC=16.588 MHz | | 7.3 | | mA |
| | | | FRC=8.294 MHz | – | 6.7 | – | |
| | | FAST mode $V_{CC}$=3V | FRC=16.588 MHz | – | 4.0 | – | |
| | | | FRC=8.294 MHz | | 3.0 | | |
| | | SLOW mode | SRC, $V_{CC}$=5V | – | 2.1 | – | |
| | | | SRC, $V_{CC}$=3V | – | 1.4 | – | |
| | | FAST mode $V_{CC}$=5V, IVC=2.45V | FRC=16.588 MHz | | 3.9 | | mA |
| | | | FRC=8.294 MHz | | 2.9 | | |
| | | FAST mode $V_{CC}$=3V, IVC=2.45V | FRC=16.588 MHz | | 3.7 | | |
| | | | FRC=8.294 MHz | | 2.7 | | |
| | | SLOW mode IVC=2.45V | SRC, $V_{CC}$=5V | | 1.3 | | |
| | | | SRC, $V_{CC}$=3V | | 1.3 | | |
| | | IDLE mode (PWRSAV=0) | SRC, $V_{CC}$=5V | | 138 | | µA |
| | | | SRC, $V_{CC}$=3V | | 102 | | |
| | | IDLE mode (PWRSAV=1) | SRC, $V_{CC}$=5V | | 11.8 | | |
| | | | SRC, $V_{CC}$=3V | | 4.7 | | |
| | | IDLE mode (PWRSAV=1,PORPD=1) | SRC, $V_{CC}$=5V | – | 10.8 | – | |
| | | | SRC, $V_{CC}$=3V | | 4.1 | | |
| | | HLAT mode (PWRSAV=1) | $V_{CC}$=5V | | 7.0 | | |
| | | | $V_{CC}$=3V | – | 2.4 | – | |
| | | STOP mode | $V_{CC}$=5V | | 0.4 | | |
| | | | $V_{CC}$=3V | | 0.2 | | |
| Pull-Up Resistor | $R_{UP}$ | $V_{IN}$=0V P3.7,P3.3,P1.7 | $V_{CC}$=5V | – | 1.16 | – | MΩ |
| | | | $V_{CC}$=3V | – | 1.16 | – | |
| | | $V_{IN}$=0V other GPIO | $V_{CC}$=5V | – | 31.4 | – | KΩ |
| | | | $V_{CC}$=3V | – | 54.0 | – | |

## 3. Clock Timing

| Parameter | Condition | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| FRC Frequency | 25℃, $V_{CC}$=4.5V | –1% | 16.588 | +1% | MHz |
| | 0℃ ~ 105℃, $V_{CC}$=4.5V | –1.5% | 16.588 | +1.5% | |
| | 0℃ ~ 105℃, $V_{CC}$=3.0 ~ 5.5V | –3.5% | 16.588 | +3.5% | |

| Parameter | Condition | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| SRC Frequency | $V_{CC}$=5V | | 41 | | KHz |
| | $V_{CC}$=3V | | 37 | | |

## 4. Reset Timing Characteristics ($T_A$= –40°C ~ +105°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 | – | 50 | – | ms |
| | $V_{CC}$=3V, WDTPSC=11 | – | 55 | – | |
| CPU start up time | $V_{CC}$ = 5V | – | 22 | – | ms |

## 5. LVR Circuit Characteristics ($T_A$ = 25℃)

| Parameter | Symbol | Condition | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| LVR Voltage | $LVR_{th}$ | $T_A$ = 25℃ | – | 2.05 | – | V |
| | | | – | 2.19 | – | |
| | | | – | 2.33 | – | |
| | | | – | 2.47 | – | |
| | | | – | 2.61 | – | |
| | | | – | 2.75 | – | |
| | | | – | 2.89 | – | |
| | | | – | 3.03 | – | |
| | | | – | 3.17 | – | |
| | | | – | 3.31 | – | |
| | | | – | 3.45 | – | |
| | | | – | 3.59 | – | |
| | | | – | 3.73 | – | |
| | | | – | 3.87 | – | |
| | | | – | 4.01 | – | |
| | | | – | 4.15 | – | |
| LVR Hysteresis Window | $V_{HYS\_LVR}$ | $T_A$ = 25℃ | – | 20 | – | mV |
| Low Voltage Detection time | $T_{LVR}$ | $T_A$ = 25℃ | 100 | – | – | μs |

## 6. LVD Circuit Characteristics ($T_A$ = 25℃)

| Parameter | Symbol | Condition | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| LVD Voltage | $LVD_{th}$ | $T_A$ = 25℃ | – | 2.05 | – | V |
| | | | | 2.19 | | |
| | | | – | 2.33 | – | |
| | | | – | 2.47 | – | |
| | | | – | 2.61 | – | |
| | | | – | 2.75 | – | |
| | | | – | 2.89 | – | |
| | | | – | 3.03 | – | |
| | | | – | 3.17 | – | |
| | | | – | 3.31 | – | |
| | | | – | 3.45 | – | |
| | | | – | 3.59 | – | |
| | | | – | 3.73 | – | |
| | | | – | 3.87 | – | |
| | | | – | 4.01 | – | |
| | | | – | 4.15 | – | |
| LVD Hysteresis Window | $V_{HYS\_LVD}$ | LVDHYS = 0 | – | 20 | – | mV |
| | | LVDHYS = 1 | – | 60 | – | |
| Low Voltage Detection time | $T_{LVD}$ | $T_A$ = 25℃ | 100 | – | – | μs |

## 7. 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 < 5KΩ) | | – | – | 4.2 | MHz |
| | Source impedance (Rs < 10KΩ) | | – | – | 2.1 | |
| | Source impedance (Rs < 25KΩ) | | – | – | 1.1 | |
| | Source is VBG (ADCHS=01100b) | | – | – | 4.2 | |
| Conversion Time | $F_{ADC}$ = 1MHz | | – | 50 | – | μs |
| Bandgap Reference Voltage ($V_{BG}$) | – | $V_{CC}$=2.5V~5.5V 25℃ | -1.5% | 1.20 | +1.5% | V |
| | | $V_{CC}$=2.5V~5.5V -40℃~105℃ | -1.8% | 1.20 | +1.8% | |
| ADC Reference Voltage ($V_{ADC}$) | ADCVREFS=1 | $V_{CC}$=3V~5.5V 25℃ | -1.7% | 2.49 | +1.7% | |
| | | $V_{CC}$=2.8V~5.5V -40℃~105℃ | -2.3% | 2.49 | +2.3% | |
| $V_{CC}$/4 Reference Voltage ($V_{1/4}$) | – | $V_{CC}$=5V, 25℃ | -0.8% | 1.252 | +0.8% | |
| | | $V_{CC}$=3.6V, 25℃ | -0.8% | 0.902 | +0.8% | |
| Input Voltage | – | | $V_{SS}$ | – | $V_{CC}$ | V |

## 8. OPA Electrical Characteristics ($T_A$ = 25℃, $V_{CC}$ = 5V, $V_{SS}$ = 0V)

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| Power supply $V_{CC}$ | – | 2.2 | – | 5.5 | V |
| Vicm | – | 0 | – | $V_{CC}$-0.7 | V |
| Vos2 | After trim | – | 2 | – | mV |
| Δ Vos/ Δ T | After trim | – | 4 | 8 | μV/C |
| AVOL | RL = 1M ohm, CL = 100 pF, Vi = 0.1 to 4V, Vo = 1 to 4V | – | 100 | – | dB |
| GBW | RL = 1M ohm, CL = 100 pF | – | 2 | – | MHz |
| CMRR | Vo = 2V | – | 80 | – | dB |
| PSRR | Vo = 2V | – | 80 | – | dB |
| ICC | Gain = 1, OPP = 5V, OPO>2.5V at $V_{CC}$ = 5V | – | 200 | – | uA |
| SR | No load | – | 1.2 | – | V/usec |
| IOH | Gain = 1, OPP = 5V, OPO>2.5V at $V_{CC}$ = 5V | – | 8 | – | mA |
| IOL | Gain = 1, OPP = 5V, OPO>2.5V at $V_{CC}$ = 5V | – | 14 | – | mA |

## 9. Comparator Characteristics ($T_A$ = 25℃, $V_{CC}$ = 3.0V to 5.5V, $V_{SS}$ = 0V)

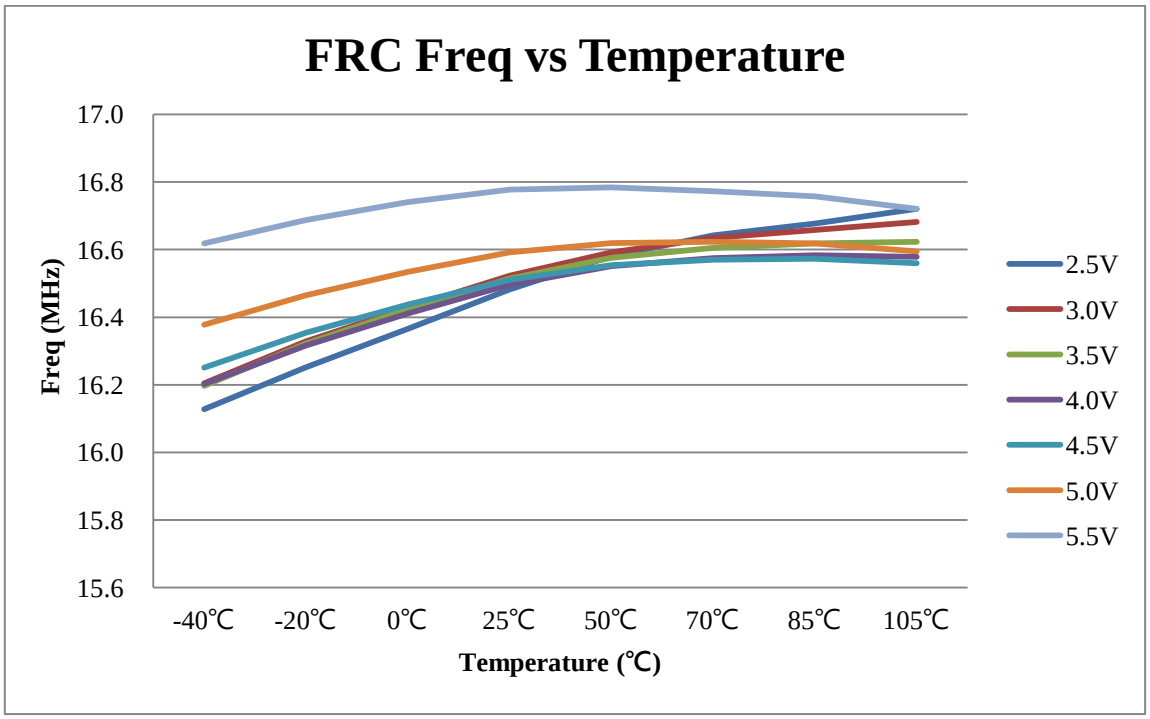
| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| Power supply $V_{CC}$ | – | 2.2 | – | 5.5 | V |
| Quiescent Current | $V_{CC}$ = 5.0V | – | 100 | – | μA |
| DAC Current | $V_{CC}$ = 5.0V | 60 | – | 220 | μA |
| $V_{OS\_CMP}$ | $V_{CC}$ = 5.0V | -15 | – | 15 | mV |
| $V_{CM\_CMP}$ | $V_{CC}$ = 5.0V | 0 | – | $V_{CC}$-0.5 | V |
| $V_{HYS\_CMP}$ | $V_{CC}$ = 5.0V | 20 | 30 | 40 | mV |

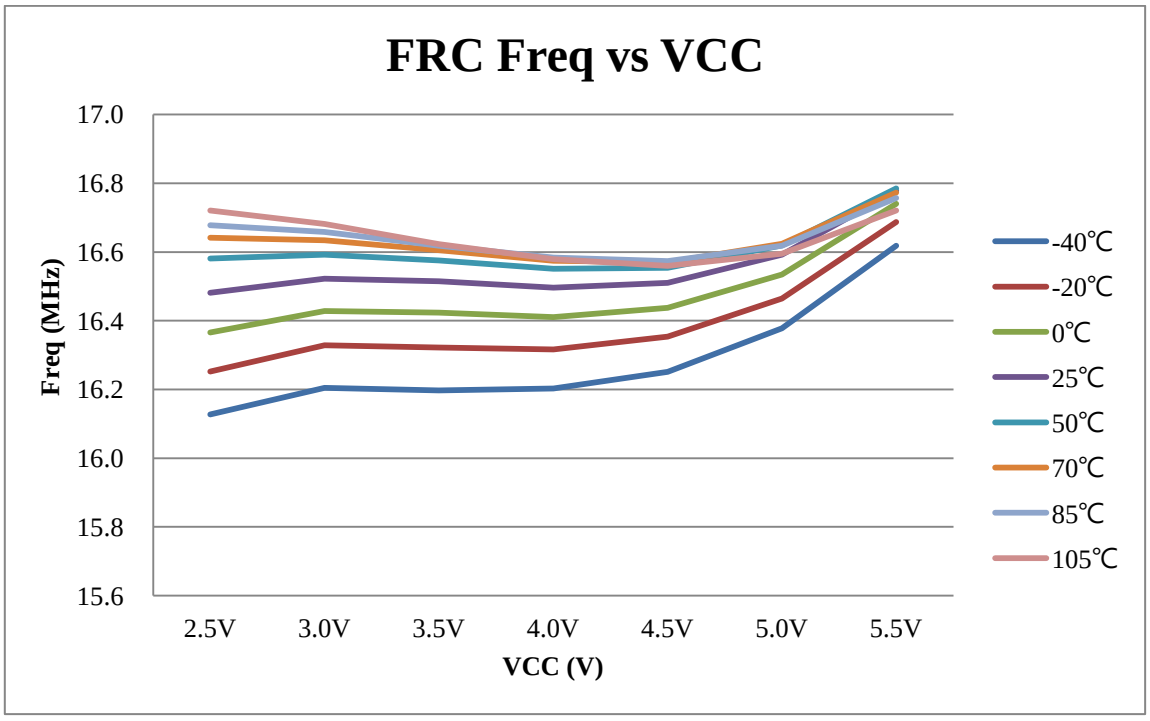
## 10. EEPROM Characteristics

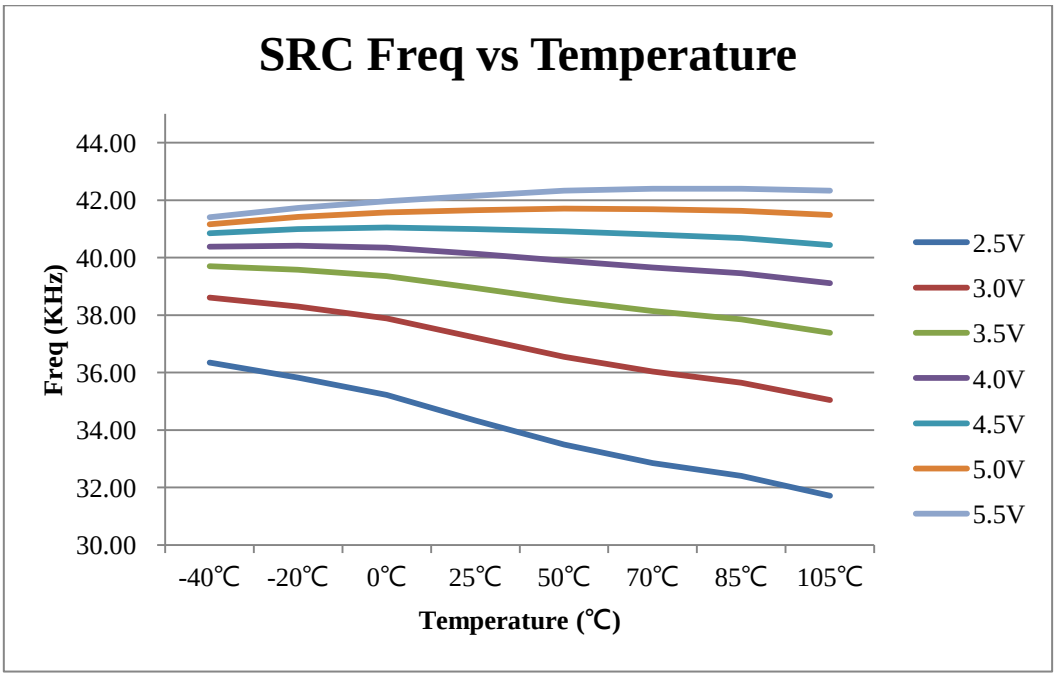
| Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| Write Voltage | –20℃ ~ 85℃, IVCPD=1 | 3.5 | 5 | 5.5 | V |
| | 0℃ ~105℃, IVCPD=1 | 4.5 | 5 | 5.5 | |
| Write Endurance* | $V_{CC}$=5V, -20℃ | 30K | – | – | cycles |
| | $V_{CC}$ =5V, -10℃ | 50K | – | – | |
| | $V_{CC}$ =3.5V~5V, 85℃ | 50K | – | – | |
| | $V_{CC}$ =4.5V, 0℃~105℃ | 50K | – | – | |

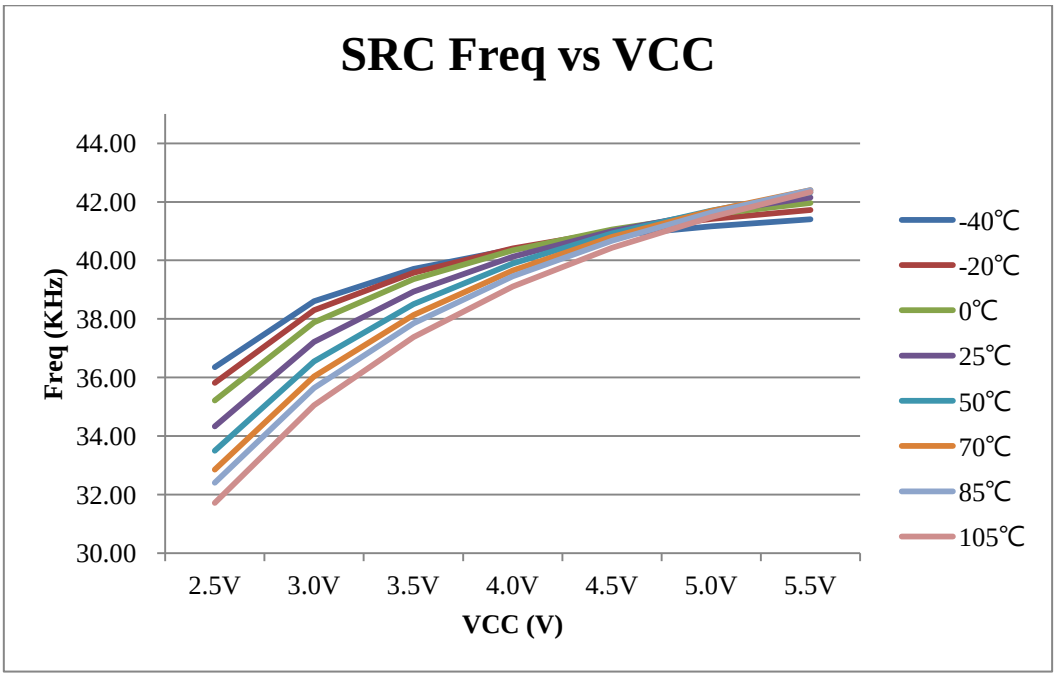
*Note: The value of this parameter is based on the characteristics of tested samples.*

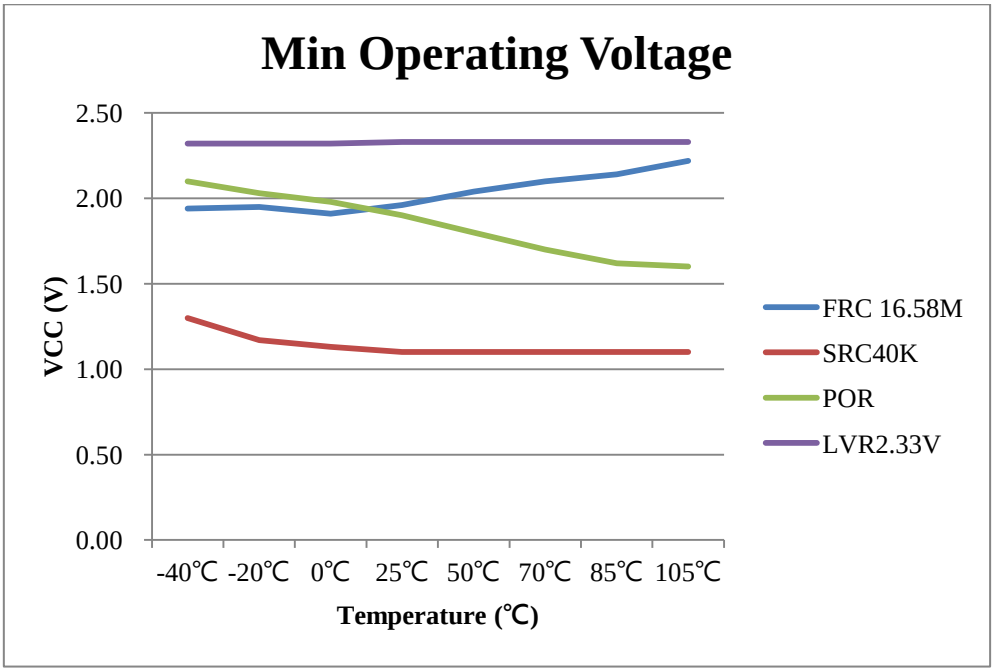
## 11. Characteristic Graphs



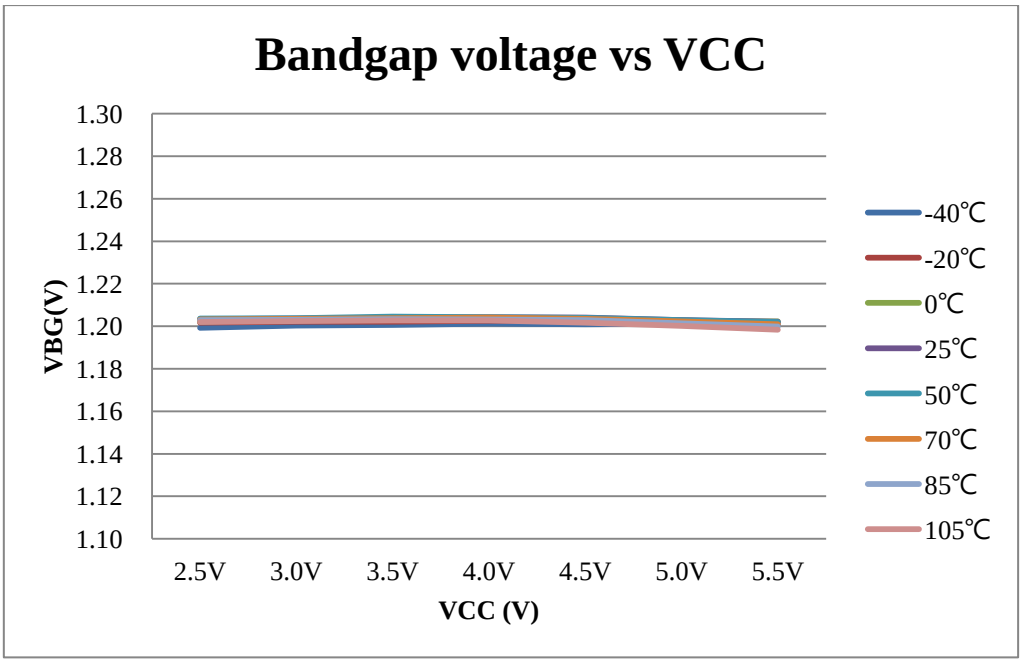




SRC Freq vs Temperature
44.00
42.00
40.00
38.00
36.00
34.00
32.00
30.00
Freq (KHz)
-40℃ -20℃ 0℃ 25℃ 50℃ 70℃ 85℃ 105℃
Temperature (℃)
2.5V
3.0V
3.5V
4.0V
4.5V
5.0V
5.5V


SRC Freq vs VCC
44.00
42.00
40.00
38.00
36.00
34.00
32.00
30.00
Freq (KHz)
2.5V 3.0V 3.5V 4.0V 4.5V 5.0V 5.5V
VCC (V)
-40℃
-20℃
0℃
25℃
50℃
70℃
85℃
105℃




*POR: Power on reset. VCC should greater than POR when power on. Due to the variation of the manufacturing process, the POR value will be slightly different between different chips.
*There are 16 levels of LVR to choose from by setting CFGWH



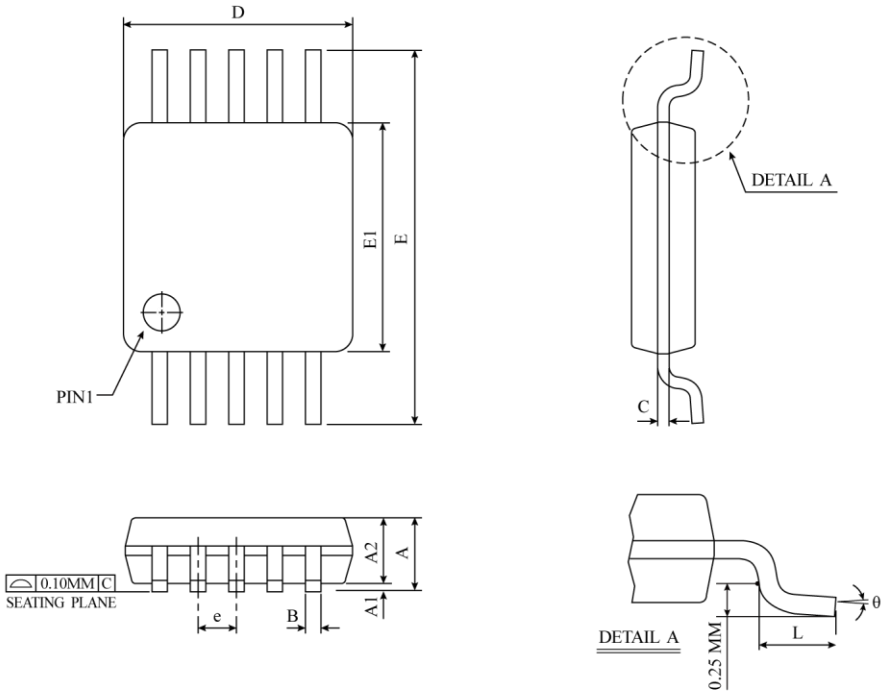
# Package and Dice Information

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 |
|---|---|
| TM52F1364-MTP | Wafer/Dice blank chip |
| TM52F1364-COD | Wafer/Dice with code |
| TM52F1364-MTP-53 | MSOP 10-pin (118mil) |
| TM52F1364-MTP-16 | SOP-16 (150mil) |
| TM52F1364-MTP-46 | TSSOP-20 ( 173mil ) |
| TM52F1364-MTP-21 | SOP-20 ( 300mil ) |
| TM52F1364-MTP-28 | SSOP-24 ( 150mil ) |
| TM52F1364-MTP-23 | SOP-28 ( 300mil ) |
| TM52F1364-MTP-29 | SSOP-28 ( 150mil ) |
| TM52F1364-MTP-D1 | QFN-20 (3*3*0.75-0.4mm)(L=0.25mm) |
| TM52F1364-MTP-C3 | QFN-28 (4x4x0.75-0.4mm) |

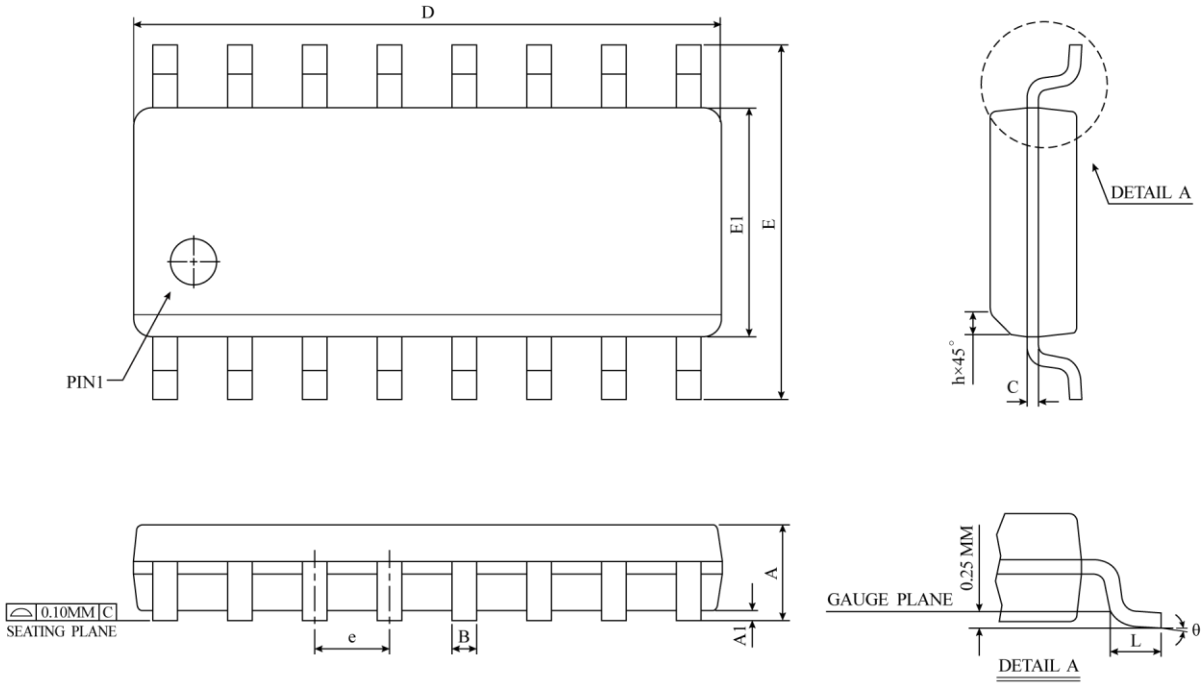
## MSOP-10 (118mil) Package Dimension




| SYMBOL | DIMENSION IN MM | | | DIMENSION IN INCH | | |
|---|---|---|---|---|---|---|
| | MIN | NOM | MAX | MIN | NOM | MAX |
| A | 0.81 | 0.96 | 1.10 | 0.032 | 0.038 | 0.043 |
| A1 | 0.05 | 0.10 | 0.15 | 0.002 | 0.004 | 0.006 |
| A2 | 0.75 | 0.85 | 0.95 | 0.030 | 0.034 | 0.037 |
| B | 0.17 | 0.22 | 0.27 | 0.007 | 0.009 | 0.011 |
| C | 0.13 | 0.18 | 0.23 | 0.005 | 0.007 | 0.009 |
| D | 2.90 | 3.00 | 3.10 | 0.114 | 0.118 | 0.122 |
| E | 4.75 | 4.90 | 5.05 | 0.187 | 0.193 | 0.199 |
| E1 | 2.90 | 3.00 | 3.10 | 0.114 | 0.118 | 0.122 |
| e | 0.50 BSC | | | 0.020 BSC | | |
| L | 0.40 | 0.55 | 0.70 | 0.016 | 0.022 | 0.028 |
| θ | 0° | 3° | 6° | 0° | 3° | 6° |
| JEDEC | | | | | | |

△ * NOTES : DIMENSION " D " DOES NOT INCLUDE MOLD PROTRUSIONS OR GATE BURRS.
MOLD PROTRUSIONS AND GATE BURRS SHALL NOT EXCEED 0.12 MM ( 0.005 INCH ) PER SIDE.
DIMENSION " E1 " DOES NOT INCLUDE MOLD PROTRUSIONS
MOLD PROTRUSIONS SHALL NOT EXCEED 0.25 MM ( 0.010 INCH ) PER SIDE.

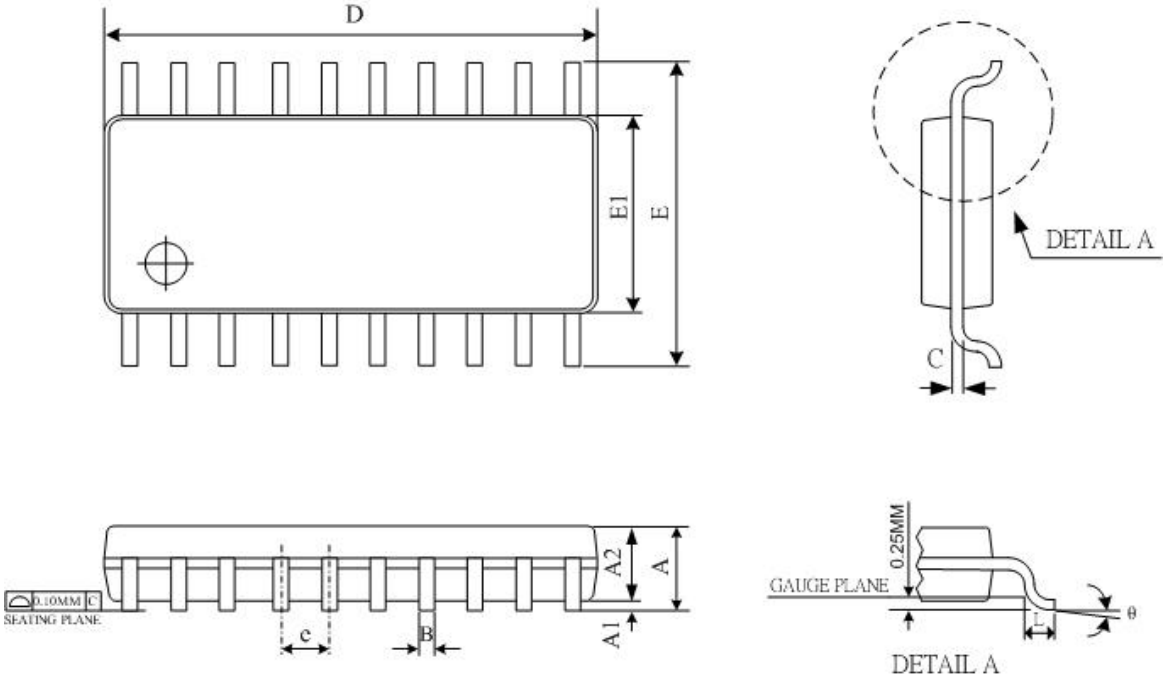
## SOP-16 ( 150mil ) Package Dimension




| SYMBOL | DIMENSION IN MM | | | DIMENSION IN INCH | | |
|---|---|---|---|---|---|---|
| | MIN | NOM | MAX | MIN | NOM | MAX |
| A | 1.35 | 1.55 | 1.75 | 0.0532 | 0.0610 | 0.0688 |
| A1 | 0.10 | 0.18 | 0.25 | 0.0040 | 0.0069 | 0.0098 |
| B | 0.33 | 0.42 | 0.51 | 0.0130 | 0.0165 | 0.0200 |
| C | 0.19 | 0.22 | 0.25 | 0.0075 | 0.0087 | 0.0098 |
| D | 9.80 | 9.90 | 10.00 | 0.3859 | 0.3898 | 0.3937 |
| E | 5.80 | 6.00 | 6.20 | 0.2284 | 0.2362 | 0.2440 |
| E1 | 3.80 | 3.90 | 4.00 | 0.1497 | 0.1536 | 0.1574 |
| e | 1.27 BSC | | | 0.050 BSC | | |
| h | 0.25 | 0.38 | 0.50 | 0.0099 | 0.0148 | 0.0196 |
| L | 0.40 | 0.84 | 1.27 | 0.0160 | 0.0330 | 0.0500 |
| θ | 0° | 4° | 8° | 0° | 4° | 8° |
| JEDEC | MS-012 (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.

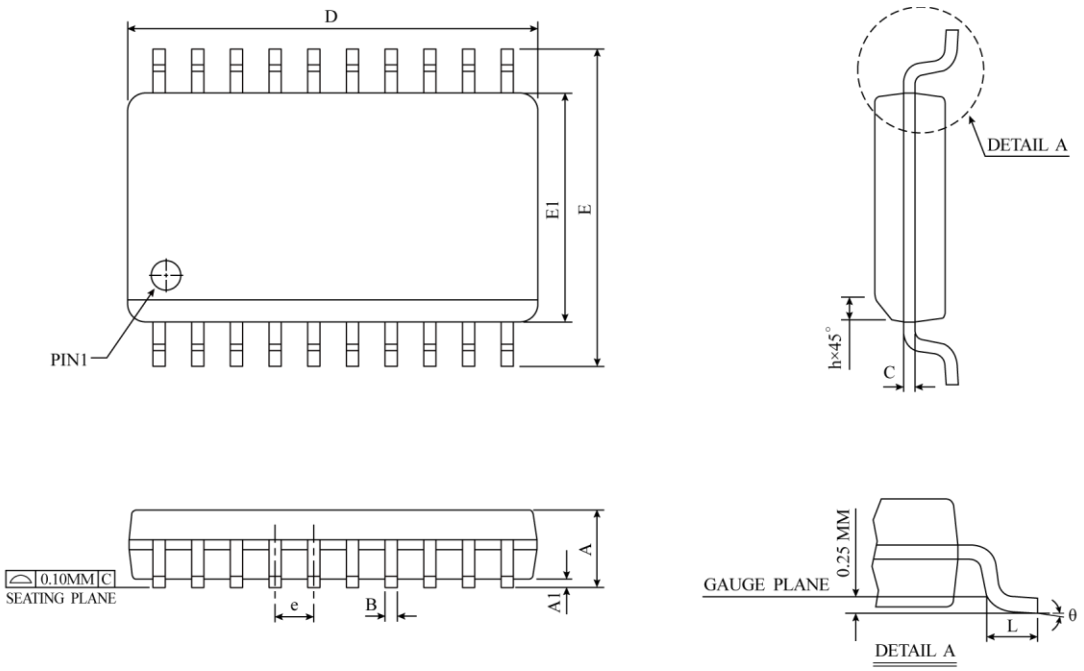
## TSSOP-20 ( 173mil ) Package Dimension




| SYMBOL | DIMENSION IN MM | | | DIMENSION IN INCH | | |
|---|---|---|---|---|---|---|
| | MIN | NOM | MAX | MIN | NOM | MAX |
| A | - | - | 1.2 | - | - | 0.047 |
| A1 | 0.05 | 0.10 | 0.15 | 0.002 | 0.004 | 0.006 |
| A2 | 0.8 | 0.93 | 1.05 | 0.031 | 0.036 | 0.041 |
| B | 0.19 | - | 0.3 | 0.007 | - | 0.012 |
| D | 6.4 | 6.5 | 6.6 | 0.252 | 0.256 | 0.260 |
| E | 6.25 | 6.4 | 6.55 | 0.246 | 0.252 | 0.258 |
| E1 | 4.3 | 4.4 | 4.5 | 0.169 | 0.173 | 0.177 |
| e | 0.65 BSC | | | 0.026 BSC | | |
| L | 0.45 | 0.60 | 0.75 | 0.018 | 0.024 | 0.030 |
| θ | 0 ° | | 8 ° | 0 ° | | 8 ° |
| JEDEC | MO-153 AC REV.F | | | | | |

Notes：
1.DIMENSION "D" DOES NOT INCLUDE MOLD FLASH, PROTRUSIONS OR GATE BURRS. MOLD FLASH, PROTRUSIONS OR GATE BURRS SHALL NOT EXCEED 0.15 PER SIDE.
2.DIMENSION "E1" DOES NOT INCLUDE INTERLEAD FLASH OR PROTRUSION. INTERLEAD FLASH OR PROTRUSION SHALL NOT EXCEED 0.25 PER SIDE.
3.DIMENSION "B" DOES NOT INCLUDE DAMBAR PROTRUSION.ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.08MM TOTAL IN EXCESS OF THE "B" DIMENSION AT MAXIMUM METERIAL CONDITION. DAMBAR CANNOT BE LOCATED ON THE LOWER RADIUS OF THE FOOT. MINIMUM SPACE BETWEEN PROTRUSION AND ADJACENT LEAD IS 0.07MM.

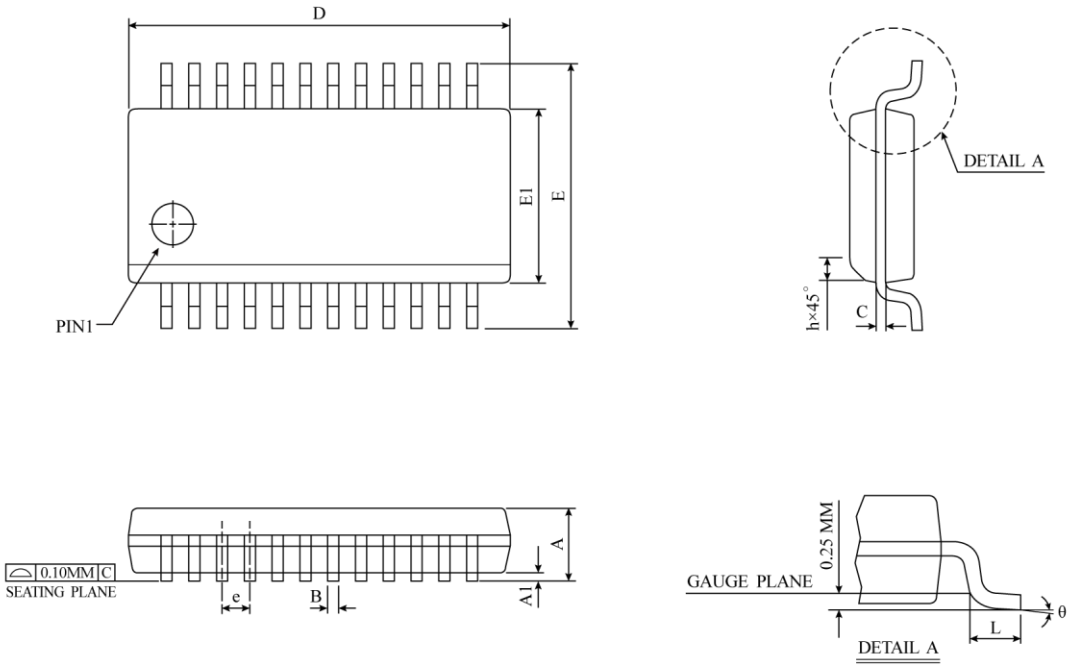
## SOP-20 ( 300mil ) Package Dimension




| 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.

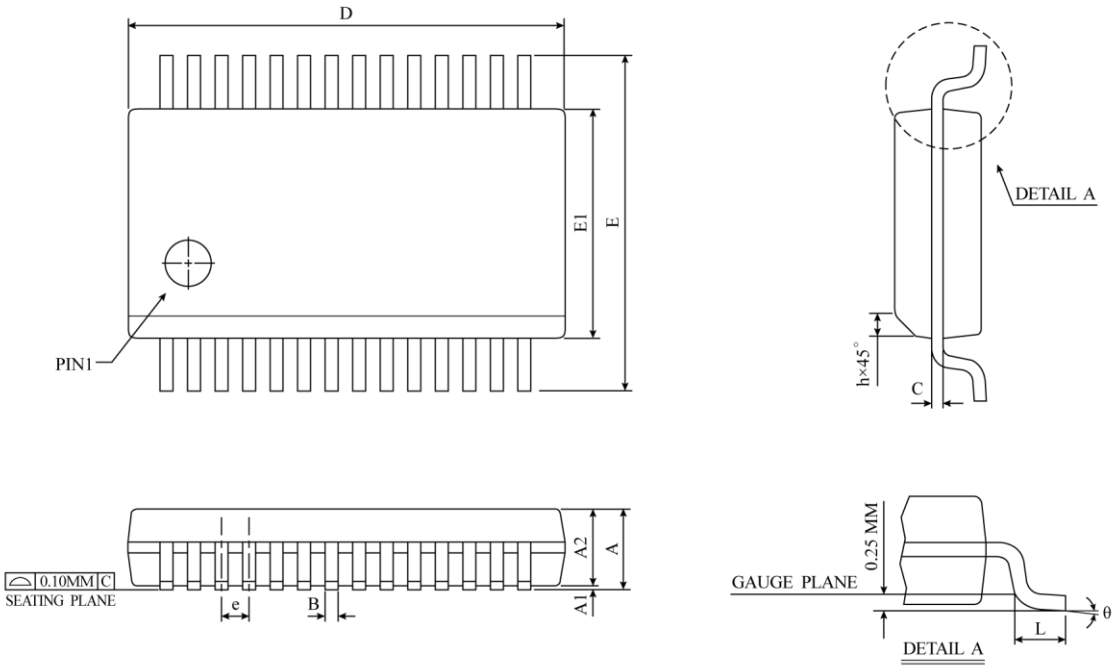
## SSOP-24 ( 150mil ) Package Dimension




| SYMBOL | DIMENSION IN MM | | | DIMENSION IN INCH | | |
|---|---|---|---|---|---|---|
| | MIN | NOM | MAX | MIN | NOM | MAX |
| A | 1.35 | 1.55 | 1.75 | 0.053 | 0.061 | 0.069 |
| A1 | 0.10 | 0.18 | 0.25 | 0.004 | 0.007 | 0.010 |
| A2 | - | - | 1.50 | - | - | 0.059 |
| B | 0.20 | 0.25 | 0.30 | 0.008 | 0.010 | 0.012 |
| C | 0.18 | 0.22 | 0.25 | 0.007 | 0.009 | 0.010 |
| D | 8.56 | 8.65 | 8.74 | 0.337 | 0.341 | 0.344 |
| E | 5.79 | 6.00 | 6.20 | 0.228 | 0.236 | 0.244 |
| E1 | 3.81 | 3.90 | 3.99 | 0.150 | 0.154 | 0.157 |
| e | 0.635 BSC | | | 0.025 BSC | | |
| L | 0.41 | 0.84 | 1.27 | 0.016 | 0.033 | 0.050 |
| θ | 0° | 4° | 8° | 0° | 4° | 8° |
| JEDEC | M0-137 (AE) | | | | | |

Ⓑ * NOTES : DIMENSION " D " DOES NOT INCLUDE MOLD PROTRUSIONS OR GAT BURRS.
MOLD PROTRUSIONS AND GATE BURRS SHALL NOT EXCEED 0.006 INCH PER SIDE.

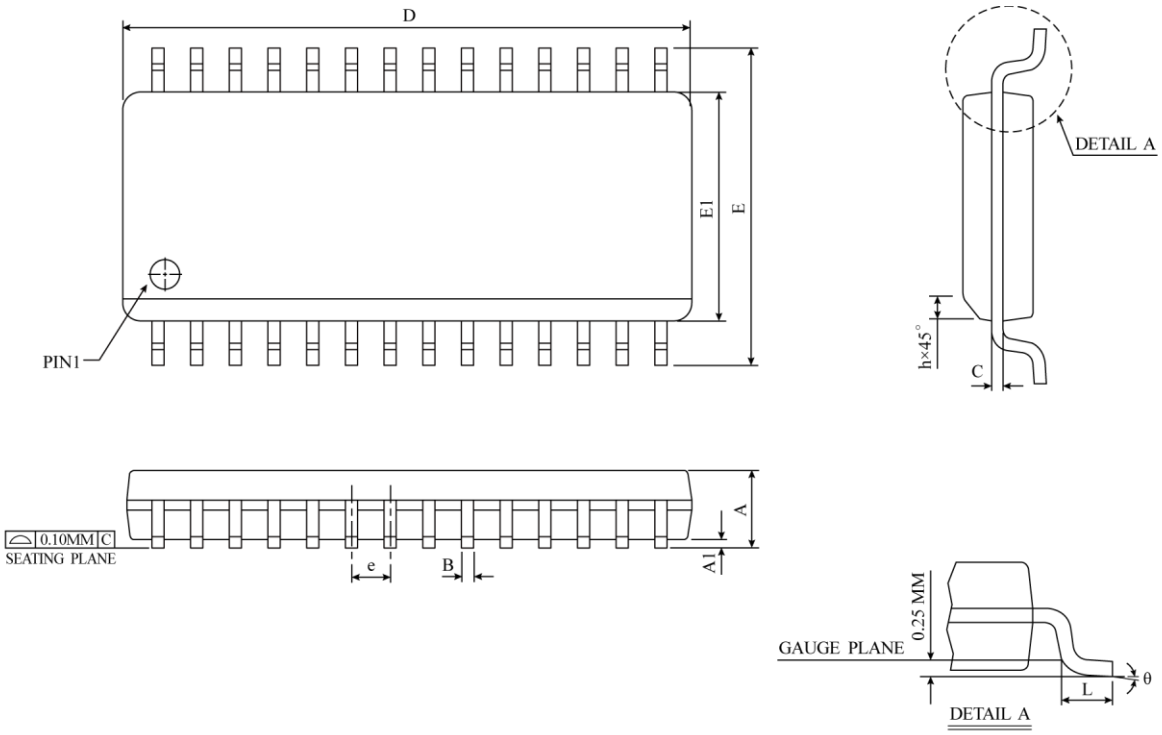
## SSOP-28 ( 150mil ) Package Dimension




| SYMBOL | DIMENSION IN MM | | | DIMENSION IN INCH | | |
|---|---|---|---|---|---|---|
| | MIN | NOM | MAX | MIN | NOM | MAX |
| A | 1.50 | 1.65 | 1.80 | 0.06 | 0.06 | 0.07 |
| A1 | 0.102 | 0.176 | 0.249 | 0.004 | 0.007 | 0.010 |
| A2 | 1.40 | 1.475 | 1.55 | 0.06 | 0.06 | 0.06 |
| B | 0.20 | 0.25 | 0.30 | 0.01 | 0.01 | 0.01 |
| C | 0.2TYP | | | 0.008TYP | | |
| e | 0.635TYP | | | 0.025TYP | | |
| D | 9.804 | 9.881 | 9.957 | 0.386 | 0.389 | 0.392 |
| E | 5.842 | 6.020 | 6.198 | 0.230 | 0.237 | 0.244 |
| E1 | 3.86 | 3.929 | 3.998 | 0.152 | 0.155 | 0.157 |
| L | 0.406 | 0.648 | 0.889 | 0.016 | 0.026 | 0.035 |
| θ | 0° | 4° | 8° | 0° | 4° | 8° |
| JEDEC | M0-137(AF) | | | | | |

△B *NOTES：DIMENSION "D" DOES NOT INCLUDE MOLD PROTRUSIONS OR GATE BURRS.
MOLD PROTRUSIONS AND GATE BURRS SHALL NOT EXCEED 0.006 INCH PER SIDE.

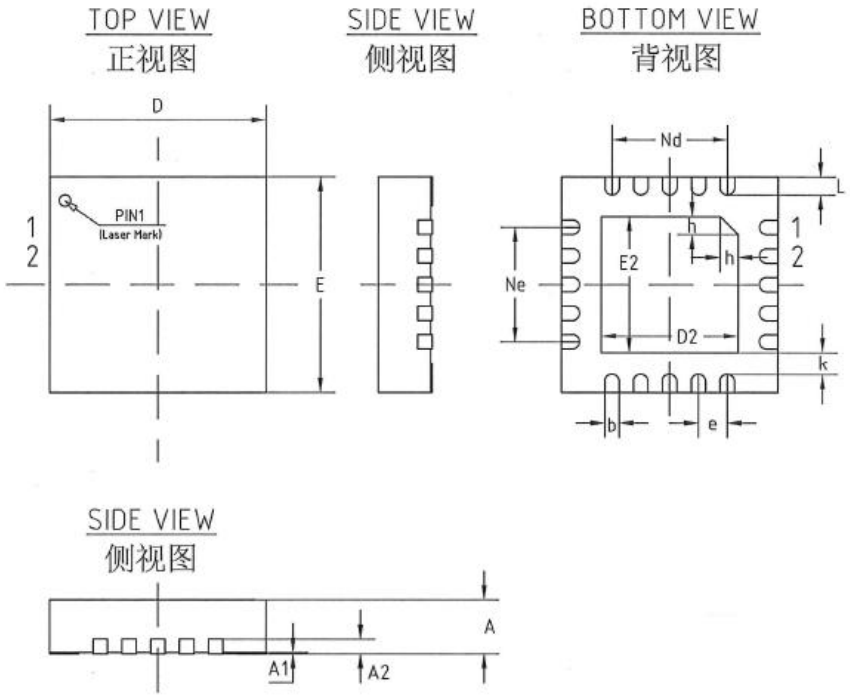
## SOP-28 ( 300mil ) Package Dimension




| 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) | | | | | |

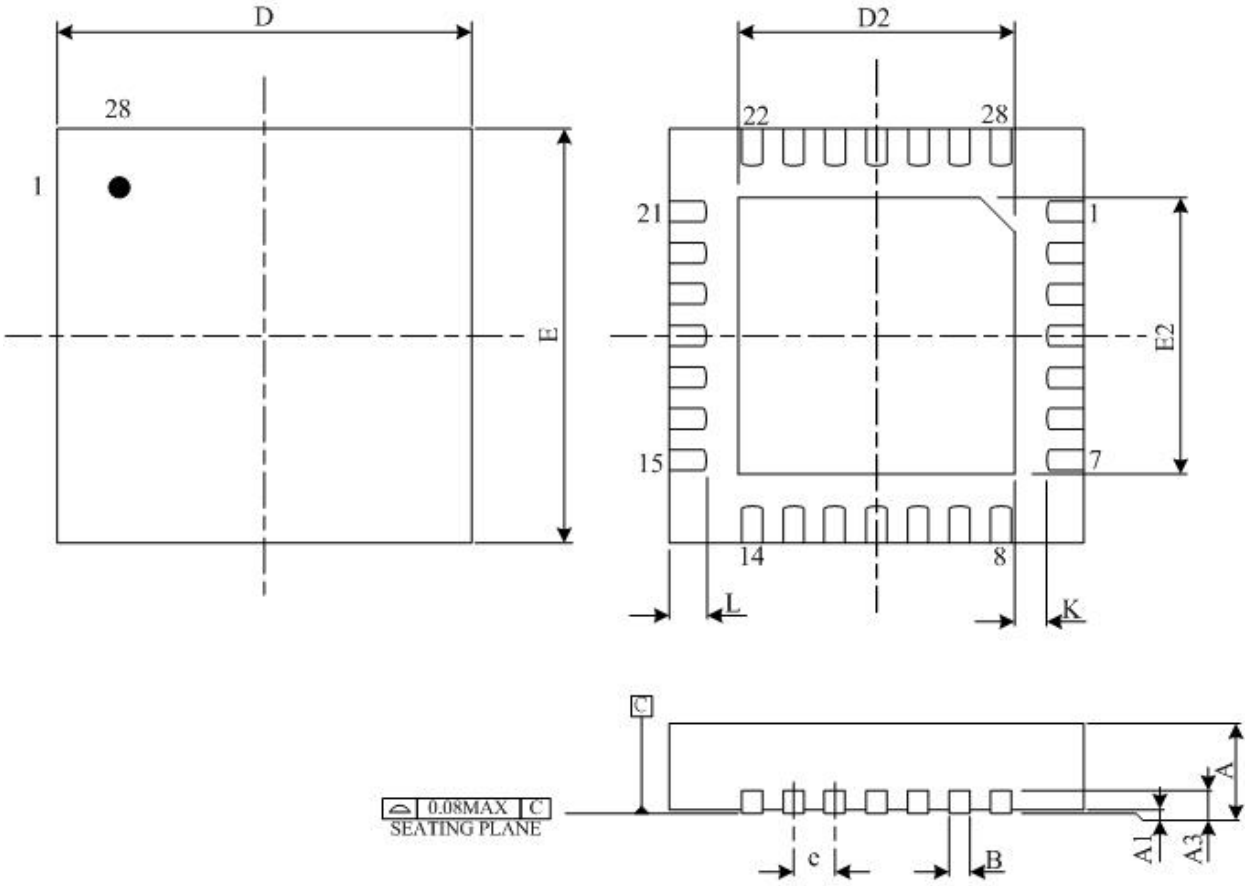
⚠ * NOTES : DIMENSION " D " DOES NOT INCLUDE MOLD FLASH, PROTRUSIONS OR GATE BURRS.
MOLD FLASH, PROTRUSIONS AND GATE BURRS SHALL
NOT EXCEED 0.15 MM ( 0.006 INCH ) PER SIDE.

**QFN 20 (3*3*0.75-0.4mm) (L=0.25mm) Package Dimension**




| 机械尺寸/mm | | | |
|---|---|---|---|
| 字符 SYMBOL | 最小值 MIN | 典型值 NOMINAL | 最大值 MAX |
| A | 0.70 | 0.75 | 0.80 |
| A1 | - | 0.02 | 0.05 |
| A2 | 0.203 REF | | |
| b | 0.15 | 0.20 | 0.25 |
| D | 2.90 | 3.00 | 3.10 |
| D2 | 1.80 | 1.90 | 2.00 |
| E | 2.90 | 3.00 | 3.10 |
| E2 | 1.80 | 1.90 | 2.00 |
| e | 0.40 BSC | | |
| K | 0.20 | 0.30 | 0.40 |
| L | 0.20 | 0.25 | 0.30 |
| h | 0.20 | 0.25 | 0.30 |
| Ne | 1.60 BSC | | |
| Nd | 1.60 BSC | | |

**QFN-28 (4x4x0.75-0.4mm) Package Dimension**




| SYMBOL | DIMENSION IN MM | | | DIMENSION IN INCH | | |
|---|---|---|---|---|---|---|
| | MIN | NOM | MAX | MIN | NOM | MAX |
| A | 0.7 | 0.75 | 0.8 | 0.028 | 0.030 | 0.031 |
| A1 | 0 | 0.02 | 0.05 | 0 | 0.001 | 0.002 |
| A3 | | 0.203 REF | | | 0.008 REF | |
| B | 0.15 | 0.2 | 0.25 | 0.006 | 0.008 | 0.010 |
| D | | 4 BSC | | | 0.157 | |
| E | | 4 BSC | | | 0.157 | |
| D2 | 2.2 | 2.3 | 2.4 | 0.087 | 0.091 | 0.094 |
| E2 | 2.2 | 2.3 | 2.4 | 0.087 | 0.091 | 0.094 |
| e | | 0.4 BSC | | | 0.016 | |
| L | 0.3 | 0.4 | 0.5 | 0.012 | 0.016 | 0.020 |
| K | | 0.45 REF | | | 0.018 | |
| JEDEC | | MO-220 | | | | |