PMICRO
Powerlink Microelectronics

# PL51T020

**Low Power High Performance ADC/Touch Key Flash 8051 MCU**

## Product Description:

PL51T020 is 8-bit high performance microcontroller with fully integrated touch key functions eliminating the need for external components, 11-bit up to 8 channels ADC and other modules. With the flexible configurable options integrated, PL51T020 offers a reliable and easy way of implementing touch keys, ADC, multi-function combinations for their product applications.

With integrating up to 16 flexible touch keys (which including 4 touch keys could be shift from P2.7~P2.4 to P0.0~P0.3), PL51T020 offers the customers a reliable and easy way of implementing touch keys for the product applications.

Special algorithms are employed to reduce the possibility of false detections, increasing the touch switch application reliability under adverse environmental conditions. With auto-calibration, low operating current and low power one-key operating state, PL51T020 provides a simple and effective means of implementing touch switches in a wide range of applications.

For high reliability and low cost issues, PL51T020 builds in reliable watchdog timer (WDT) and low power detect (LPD) and low voltage reset (LVR) function. The excellent noise immunity and ESD protection ensure reliable operations in the adverse electrical environments.

PL51T020 internal integrates high precision RC oscillator to operate and switch dynamically between a range of operating modes using different clock sources to optimize microcontroller operation and minimize power consumption. The excellent noise immunity and ESD protection ensure reliable operations in the adverse electrical environments.

PL51T020 also supports three low power modes, idle mode, stop mode and sleep mode, to reduce power consumption. It supports to be wakeup speedy by touch Key action when work in the low power mode.

PL51T020 is communicating with the outside world with UART, I2C and SPI interfaces.

Besides the 4K bytes Flash program memory, other memory includes 256 bytes RAM and 128 bytes data EEPROM.

In-Circuit-Programming (ICP) supports the users to upgrade the program code and data in circuit without removing the microcontroller from the actual application board.

For easy usage, POWERLINK provides the debugger and writer.

## Key Features:

- 1T Enhanced 8-Bit ET8051
- 4KB Flash and 128B EEPROM
- Fully integrated up to 16+4(shift) touch keys
- Operation Frequency@Voltage:
  ~4MHz@2.0~5.5V;
  ~8MHz@2.4~5.5V;
  ~12MHz@2.7~5.5V
- Operation Temperature: -40℃ ~+125℃
- Supports Crystal Oscillator, internal 32KHz and high precision RC oscillator(4/8/12MHz,±2%), external clock input
- Programmable System Clock
- Up to 22 bidirectional GPIO
- Four Priority Levels with 14 interrupt sources
- 8 Keyboard Interrupts
- 2 External Interrupts
- Support POR/LVR/LPD
- Three 16-bit Timers/Counters
- Four 12-bit PWM: PWM0/1/2/3
- Watchdog Timer with Prescaler
- Support UART/SPI/I2C interface
- Integrated Analog Comparator
- Integrated 11-bit 8 channels ADC
- Multi-mode Operation: Normal, Green, Stop, Sleep mode
- Support In-Circuit Programming
- Package: 24/20/16/8 PINs
- Memory Permission Control
- Flash Cycling: 100K @25℃
- EEPROM Cycling: 500K @25℃
- Data retention: 40 years @25℃

## Applications:

- Wireless Mice, Keyboards and Game Controllers
- RF Remote Controller
- Small Home Appliances

RoHS GREEN

# Product Types

| Product Name | Package | Program Flash | Data EEPROM[*5] | RAM | Timer | PWM | Freq@Voltage | I/O | Interface UART/SPI/I2C | ACMP | T.S. | Touch Key[*1] /Wakeup(Max) | ADC[*1] |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Touch Key Series** | | | | | | | | | | | | | |
| PL51T020B24 | SSOP24 | 4KB | 128B | 256B | 3 | 4+1 | ~ 4M@2.0~5.5V<br>~ 8M@2.4~5.5V<br>~12M@2.7~5.5V | 22 | 1/1/1 | 1 | 1 | 16+4[*3]/16 | 11b/8ch |
| PL51T020N24 | QFN24 | | | | | | | | | | | | |
| PL51T020T20 | TSSOP20 | 4KB | 128B | 256B | 3 | 4+1 | | 18 | 1/1/1 | 1 | 1 | 12+4[*2]/16 | 11b/4ch |
| PL51T020N20 | QFN20 | | | | | | | | | | | | |
| PL51T020S16 | SOP16 | 4KB | 128B | 256B | 3 | 4+1 | | 14 | 1/1/1 | - | 1 | 8+4[*2]/12 | 11b/4ch |
| PL51T020N16 | QFN16 | | | | | | | | | | | | |
| PL51T020S8 | SOP8 | 4KB | 128B | 256B | 3 | 4+1 | | 6 | - | 1[*4] | 1 | 3+2[*2]/5 | 11b/2ch |

**Note:** *1: Touch Key can't work with ADC at the same time, but can be set to work separately at different time slice.

*2: Shift Touch Keys <15:12> can be assigned as the touch keys <15:12> with wake-up function.

*3: Shift touch keys <15:12> or Original ones can be set to work separately at different time slice. Only Shift touch keys <15:12> or Original ones can be assigned as wake-up keys, separately.

*4: ACMP source, only between CMP1 and INTVREF (1.2V).

*5: In order to ensure that the Data EEPROM can be programmed stably, the LVR needs to be enabled and set to work greater than or equal to 2.4V(≥).

# Contents

# 1 Overview

PL51T020 series are a single-chip microcontroller based on a high performance 1T architecture 80C51 CPU which have a fully compatible instruction set with standard 80C51 series microcontroller, and execute instructions in 1~4 clock cycles (about 7~8 times the rate of a standard 8051 chip). The PL51T020 series have a 22 bi-direction GPIO, a 14-source (including 2 external interrupt sources), 4-priority-level interrupt structure, a fully integrated touch key function without the need for external components.

With integrating up to 16 flexible touch keys (which including 4 touch keys could be shift from P2.7~P2.4 to P0.0~P0.3), PL51T020 offers the customers a reliable and easy way of implementing touch keys for the product applications.

Special algorithms are employed to reduce the possibility of false touch action identifications, improving the touch key application reliability under adverse environmental conditions. With supporting auto calibration configuration, touch key could be working at a wide range of dynamic capacitance with low power consumption and high sensitivity identification.

For high reliability and low cost issues, PL51T020 builds in reliable watchdog timer (WDT), low power detect (LPD) and low voltage reset (LVR) functions. The excellent noise immunity and ESD protection ensure reliable operations in the adverse electrical environments.

PL51T020 integrates low and high frequency oscillators to operate and switch dynamically between a range of operating modes using different clock sources to optimize microcontroller operation and minimize power consumption.

In order to reduce power consumption, PL51T020 could be work in three low power modes, green mode, stop mode and sleep mode, it supports to be wakeup speedy by keyboard action when work in the low power mode.

PL51T020 is communicating with the outside world with UART, I2C and SPI interfaces.

Besides the 4K bytes flash program memory with Memory-Permission-Control, other memory includes 256 bytes RAM Data Memory as well as 128 bytes EEPROM memory is integrated.

In-Circuit-Programming (ICP) support the users to upgrade the program code and data in circuit without removing the microcontroller from the actual application board.

For easy usage, POWERLINK provides the debugger and writer.

PL51T020 is targeting at home appliance such as Wireless Mice, Keyboards and Game Controllers, RF Remote Controller, Induction cooker, Microwave oven, Washing machine, Clothes dryer, Dishwasher, Refrigerator, Air conditioner and etc.

# 2 Features

## Basic

- 1T 8-bit ET8051 core
- Fully integrated 16 touch key functions with no external components
- Operating Voltage @ Frequency:
  - ~4MHz@2.0~5.5V
  - ~8MHz@2.4~5.5V
  - ~12MHz@2.7~5.5V
- Oscillator Type
  - Crystal Oscillator: 400KHz to 12MHz
  - Internal RC Oscillator: 4/8/12MHz (±2%) and 32KHz
  - External Clock: 400KHz to 12MHz
- Up to 22 bidirectional General Purpose I/O
  - Input-Only with configurable pull high resistor
  - Push-Pull Output Drive Capacity: 20mA (@5V, Total: <100mA)
- Operation Temperature: -40℃ to +125℃

## Peripheral Features

- Four Priority Levels with 14 interrupt sources
  - Two External Interrupt: INT0B and INT1B
  - T0&T1 Overflow Interrupt
  - T2 Overflow, Reload, Compare/Capture Interrupt
  - UART Transmit and Receive Interrupt
  - EEPROM Write Finished Interrupt
  - Analog Comparator Interrupt
  - Keyboard Interrupt
  - Touch Key Interrupt
  - SPI Interrupt
  - I2C Interrupt
  - ADC Finish Converting Interrupt
- POR/LVR/LPD support
- Four LVR threshold Level by Fuse:
  - 2.1/2.4/3.7/4.3 V
- Two LPD threshold Level by Fuse:
  - 2.7/4.0 V
- Register Timed Access Protection
- Programmable System Clock
- Multi-mode Operation:
  - Normal/Idle/Stop/Sleep
- 16-bit Timers/Counters:
  - 80C51-like Timer 0 & 1
  - 8052-like Timer 2 with Compare/Capture Unit (CCU)
- Four 12-bit PWM: PWM0/1/2/3
- Watchdog Timer with Additional Configurable Prescaler: WDT
- UART/SPI/I2C Interface
- Analog Digital Converter: ADC
  - 11-bit resolution
  - Up to 8 multiplexed channels
  - support external input VREF

- ✧ Analog Comparator: ACMP
- ✧ Support In-Circuit Programming: ICP
- ✧ ESD: >2KV (HBM)
- ✧ EFT: >4KV
- ✧ Package Types
  - ✓ SOP8/SOP16/TSSOP20/SSOP24
  - ✓ QFN16/QFN20/QFN24

## Memory

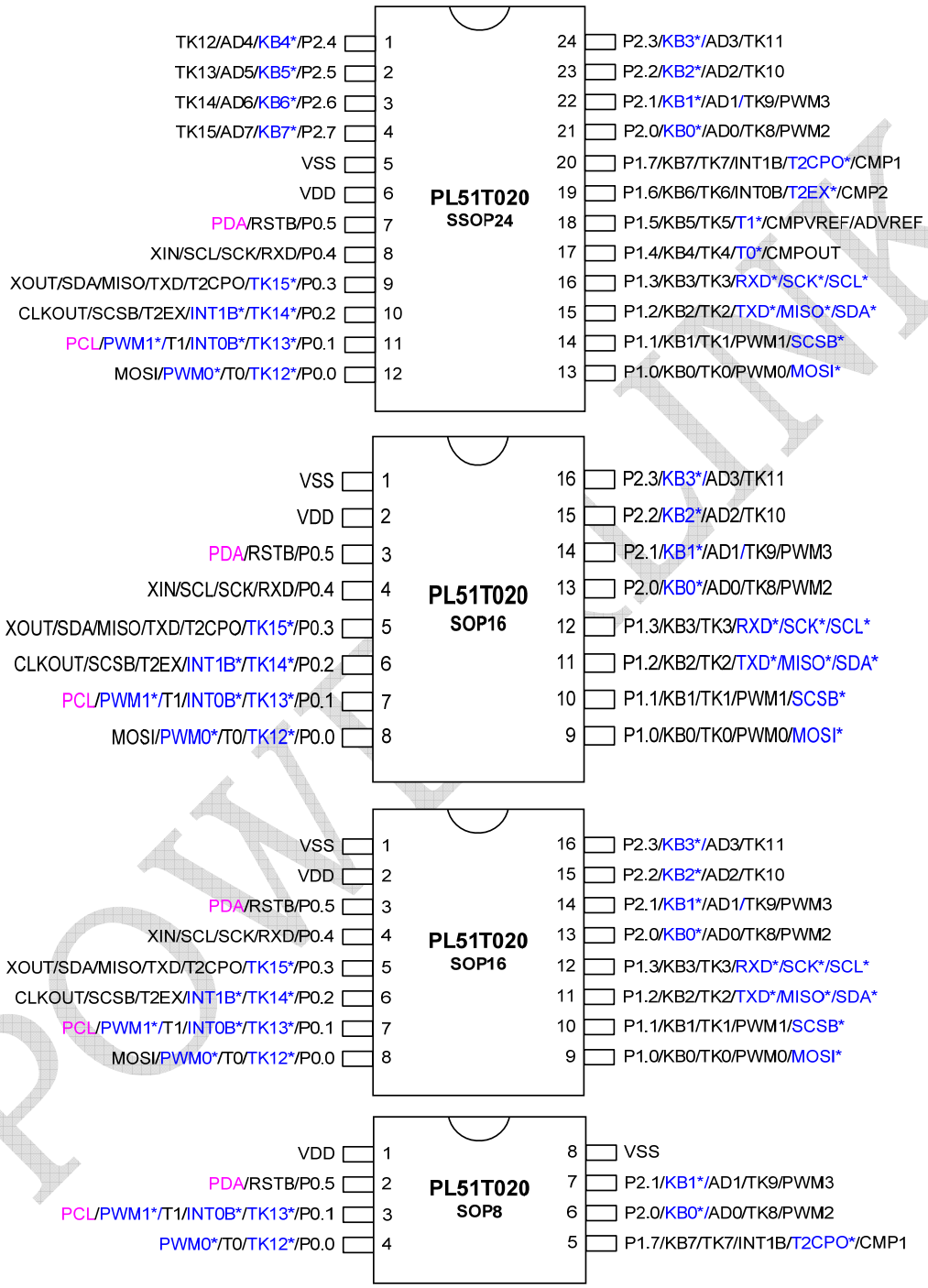
- ✧ 4K bytes Program Flash
- ✧ 128 bytes Data EEPROM（byte/page operation, 1page=32bytes)
- ✧ 256 bytes internal scratch-pad RAM
- ✧ Memory Programming Permission Control
- ✧ Flash Cycling: 100K @25℃
- ✧ EEPROM Cycling: 500K @25℃
- ✧ Data retention: 40 years @25℃

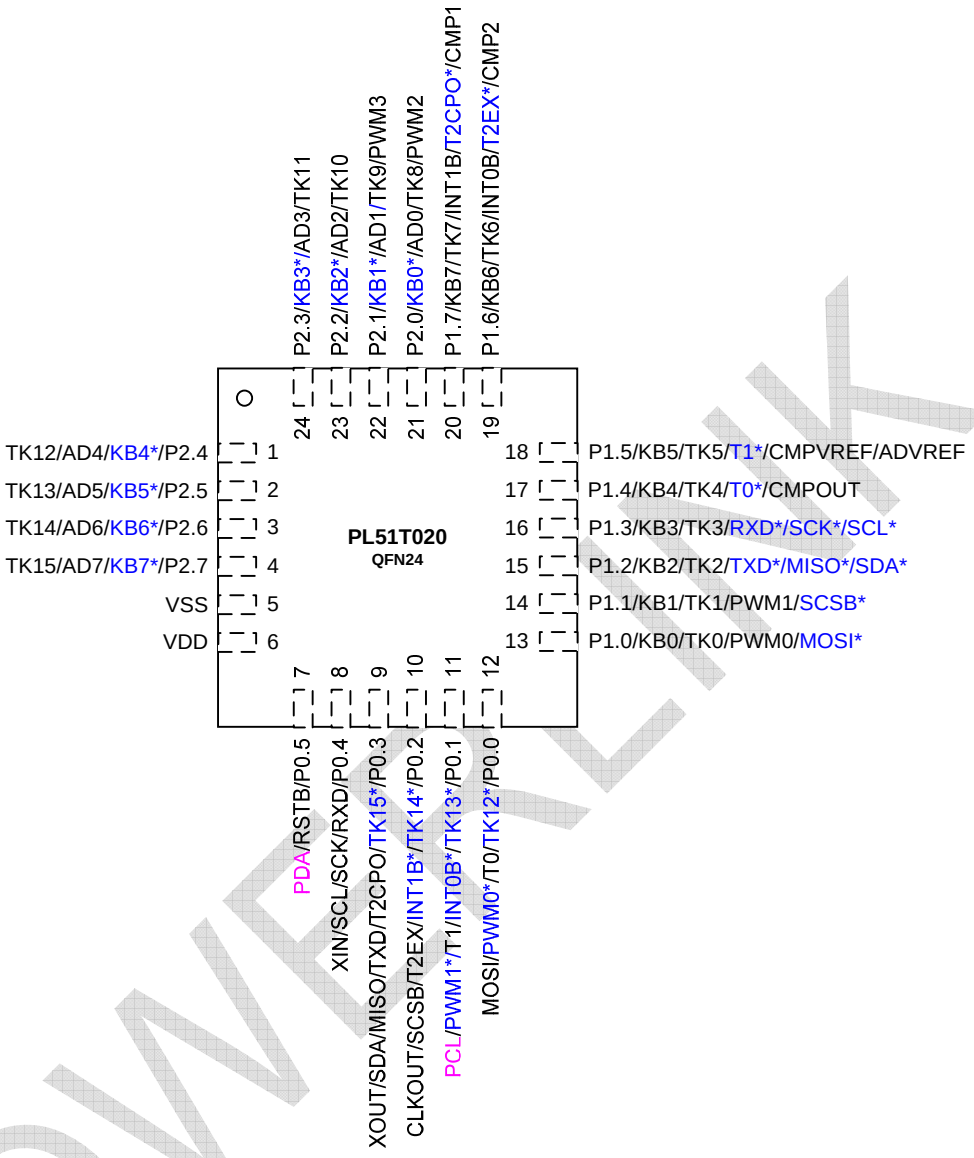
# 3 Quick Reference Data

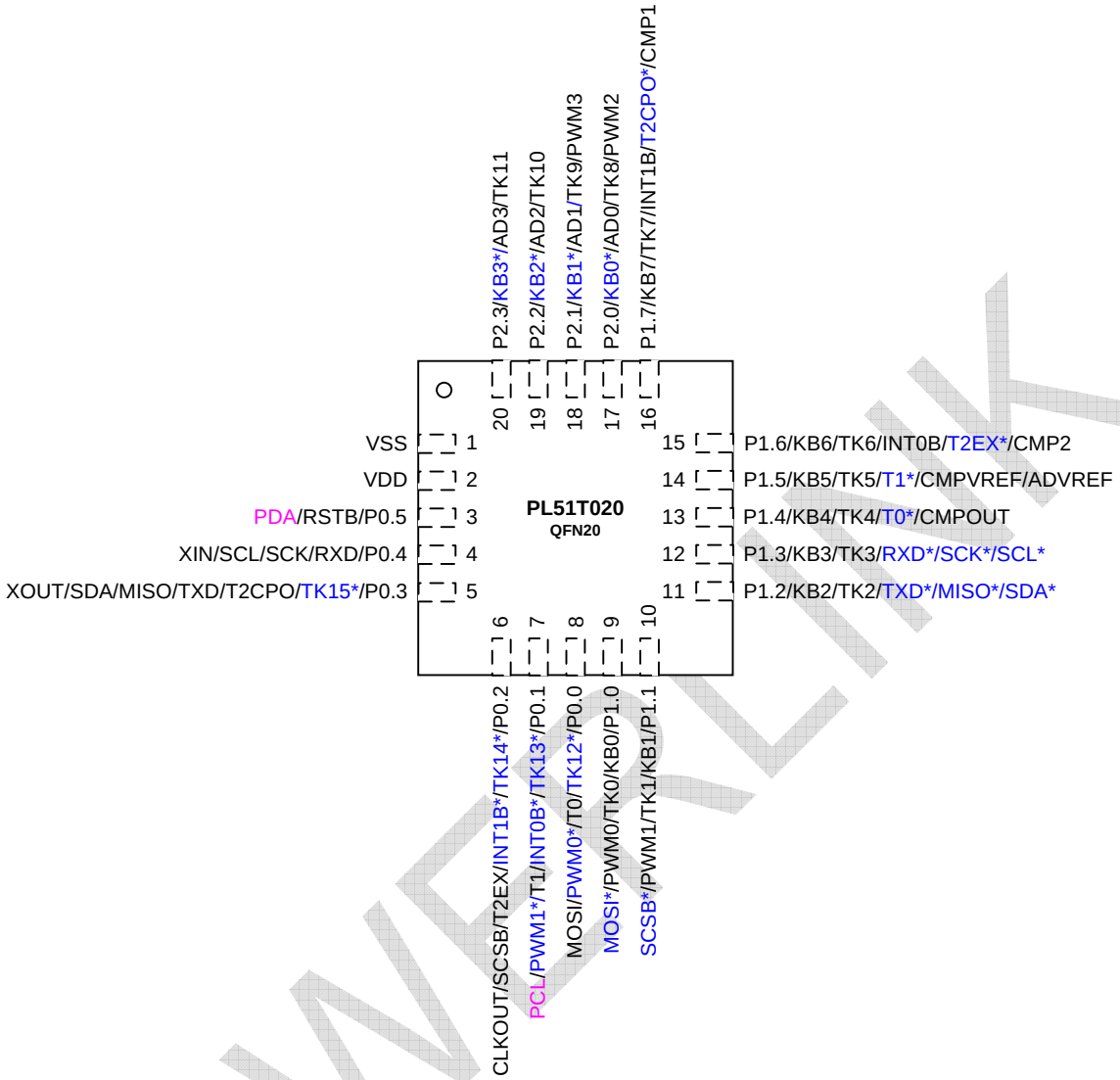
| Parameter | Value | Units |
|---|---|---|
| Min Supply Voltage | 2.0 | V |
| Operating Temperature Range | -40 to +125 | ℃ |
| Internal RC OSC Frequency | 4/8/12 | MHz |
| Internal RC OSC Precision @ 25℃ | ±2 | % |
| Push-Pull Output Drive Capacity @ 5V | 20 | mA |
| Push-Pull Output Drive Capacity @ 3.3V | 10 | mA |
| Total Push-Pull Output Drive Capacity | <100 | mA |
| Current Consumption @ Sleep Mode | 1 | uA |

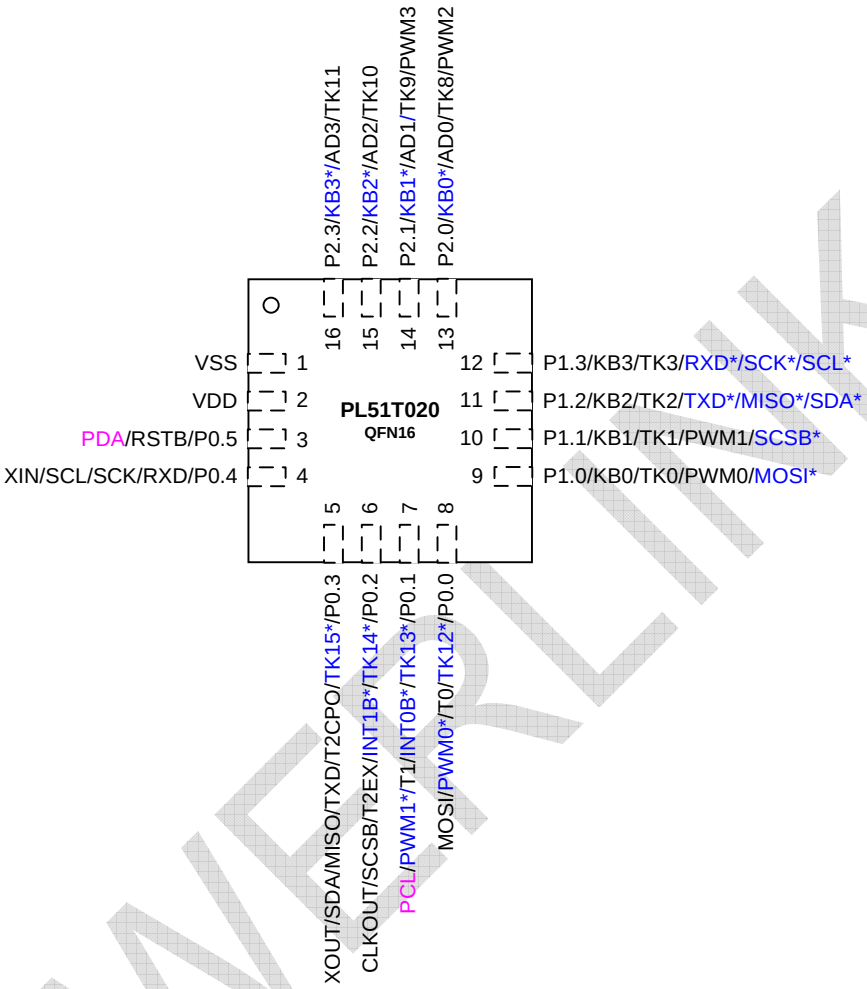
# 4 Pin Configurations

## 4.1 Pin Diagrams




PL51T020
QFN24
1 TK12/AD4/KB4*/P2.4
2 TK13/AD5/KB5*/P2.5
3 TK14/AD6/KB6*/P2.6
4 TK15/AD7/KB7*/P2.7
5 VSS
6 VDD
7 PDA/RSTB/P0.5
8 XIN/SCL/SCK/RXD/P0.4
9 XOUT/SDA/MISO/TXD/T2CPO/TK15*/P0.3
10 CLKOUT/SCSB/T2EX/INT1B*/TK14*/P0.2
11 PCL/PWM1*/T1/INT0B*/TK13*/P0.1
12 MOSI/PWM0*/T0/TK12*/P0.0
13 P1.0/KB0/TK0/PWM0/MOSI*
14 P1.1/KB1/TK1/PWM1/SCSB*
15 P1.2/KB2/TK2/TXD*/MISO*/SDA*
16 P1.3/KB3/TK3/RXD*/SCK*/SCL*
17 P1.4/KB4/TK4/T0*/CMPOUT
18 P1.5/KB5/TK5/T1*/CMPVREF/ADVREF
19 P1.6/KB6/TK6/INT0B/T2EX*/CMP2
20 P1.7/KB7/TK7/INT1B/T2CPO*/CMP1
21 P2.0/KB0*/AD0/TK8/PWM2
22 P2.1/KB1*/AD1/TK9/PWM3
23 P2.2/KB2*/AD2/TK10
24 P2.3/KB3*/AD3/TK11


PL51T020
QFN20
1 VSS
2 VDD
3 PDA/RSTB/P0.5
4 XIN/SCL/SCK/RXD/P0.4
5 XOUT/SDA/MISO/TXD/T2CPO/TK15*/P0.3
6 CLKOUT/SCSB/T2EX/INT1B*/TK14*/P0.2
7 PCL/PWM1*/T1/INT0B*/TK13*/P0.1
8 MOSI/PWM0*/T0/TK12*/P0.0
9 MOSI*/PWM0/TK0/KB0/P1.0
10 SCSB*/PWM1/TK1/KB1/P1.1
11 P1.2/KB2/TK2/TXD*/MISO*/SDA*
12 P1.3/KB3/TK3/RXD*/SCK*/SCL*
13 P1.4/KB4/TK4/T0*/CMPOUT
14 P1.5/KB5/TK5/T1*/CMPVREF/ADVREF
15 P1.6/KB6/TK6/INT0B/T2EX*/CMP2
16 P1.7/KB7/TK7/INT1B/T2CPO*/CMP1
17 P2.0/KB0*/AD0/TK8/PWM2
18 P2.1/KB1*/AD1/TK9/PWM3
19 P2.2/KB2*/AD2/TK10
20 P2.3/KB3*/AD3/TK11




**Note:**

1) The outside pin function has the highest priority, and the inner pin function has the lowest priority. It means that if the higher priority function is enabled, the lower priority function can't be used even when the lower priority function is also enabled.

2) The pin name colored blue with * denoted the shift ports, the pin function available only when the relative shift control bit in SFR "PSFT0~1" is set.

## 4.2 Pin Description

| Symbol | Type | Descriptions |
|---|---|---|
| VDD | Power | Power Supply (2.0~5.5V) |
| VSS | Power | Ground (0V) |
| RSTB | Digital Input | Reset Pin, Active Low |
| XIN | Analog Input | Crystal Oscillator Input |
| XOUT | Analog Output | Crystal Oscillator Output |
| CLKOUT | Digital Output | Internal Clock Output |
| SCL | Digital I/O | Clock for I2C Interface |
| SDA | Digital I/O | Data I/O for I2C Interface |
| SCSB | Digital Input | Enable Input for SPI Interface, active Low |
| SCK | Digital I/O | Clock for SPI Interface |
| MISO | Digital I/O | Master Data Input or Slave Data Output for SPI Interface |
| MOSI | Digital I/O | Master Data Output or Slave Data Input for SPI Interface |
| RXD | Digital Input | RXD of Serial Port |
| TXD | Digital Output | TXD of Serial Port |
| T0 | Digital Input | Timer 0 Input |
| T1 | Digital Input | Timer 1 Input |
| T2EX | Digital Input | Timer 2 External Capture Input |
| T2CPO | Digital Output | Timer 2 Compare/PWM Ouput |
| INT0B | Digital Input | External Interrupt 0 |
| INT1B | Digital Input | External Interrupt 1 |
| PWM0 | Digital Output | PWM 0 Output |
| PWM1 | Digital Output | PWM 1 Output |
| PWM2 | Digital Output | PWM 2 Output |
| PWM3 | Digital Output | PWM 3 Output |
| CMP1 | Analog Input | Comparator Positive 1 Input |
| CMP2 | Analog Input | Comparator Positive 2 Input |
| CMPVREF | Analog Input | Comparator Reference Voltage Input |
| CMPOUT | Digital Output | Comparator Output |
| TK0~15 | Analog Input | Touch Key Inputs |
| KB0~7 | Analog Input | Keyboard Inputs |
| P0.0~P0.5 | Digital I/O | General purpose I/O Port 0 |
| P1.0~P1.7 | Digital I/O | General purpose I/O Port 1 |
| P2.0~P2.7 | Digital I/O | General purpose I/O Port 2 |
| PCL | Digital Input | Clock Input for ICP (In Circuit Program) Mode |
| PDA | Digital I/O | Data I/O for ICP (In Circuit Program) Mode |
| ADVREF | Analog Input | ADC Reference Voltage Input |
| AD0~AD7 | Analog Input | ADC Inputs |

## 4.3 Terminology and Symbol Conventions

| Symbol | Description | Symbol | Description |
|---|---|---|---|
| **CPU** | Control Processor Unit | **PFL** | Program Flash |
| **ALU** | Arithmetic-Logic Unit | **DEE** | Data EEPROM |
| **MSB** | Most Significant Bit | **NVR** | NVR EEPROM |
| **LSB** | Least Significant Bit | **CEE** | Code Fuse EEPROM |
| **SFR** | Special Function Register | **TEE** | Trim Fuse EEPROM |
| **ISR** | Interrupt Service Routine unit | **ICP** | In-Circuit Programming |
| **POR** | Power On Reset | **ICD** | In-Circuit Debugging |
| **LVR** | Low Voltage Reset | **ISP** | In-System Programming |
| **LPD** | Low Power Detect | **TW** | Write operation is permitted only after open a window by 'TA' register |
| **PMU** | Power Management Unit | **SPI** | Serial Peripheral Interface |
| **PWM** | Pulse Width Modulation | **I2C** | Two-Wire Interface |
| **WDT** | Watch Dog Timer | **UART** | Universal Asynchronous Receiver Transmitter |
| **CCU** | Compare/Capture Unit | **ADC** | Analog Digital Converter |
| **TA** | Timed Access | **DAC** | Digital Analog Converter |
| **ACMP** | Analog Comparator | **CAP** | Capacitive Touch Sensor |

# 5 Block Diagram

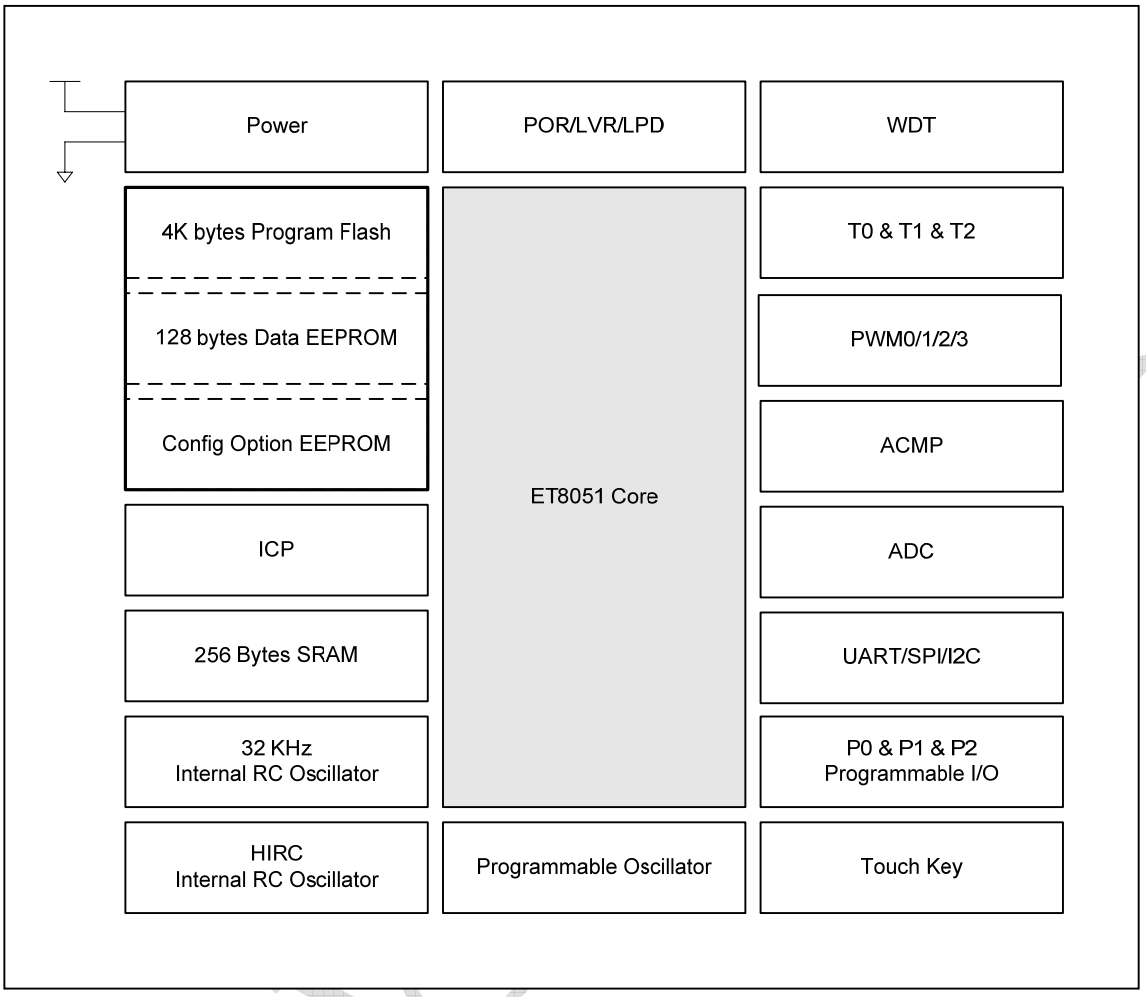



Figure 5-1 Block diagram

# 6 Memory Organization

The ET8051 microcontroller core incorporates the Harvard architecture, with separate code and data spaces.

Memory organization in the ET8051 is similar to that of the industry standard 8051. There are three memory areas: Program Memory (Internal FLASH), External Data Memory (External EEPROM) and Internal Data Memory (Internal RAM).

The Program Memory includes 4K bytes program FLASH.

The External Data Memory includes 128 bytes data EEPROM.

The Internal Data Memory includes 256 bytes internal scratch-pad RAM.

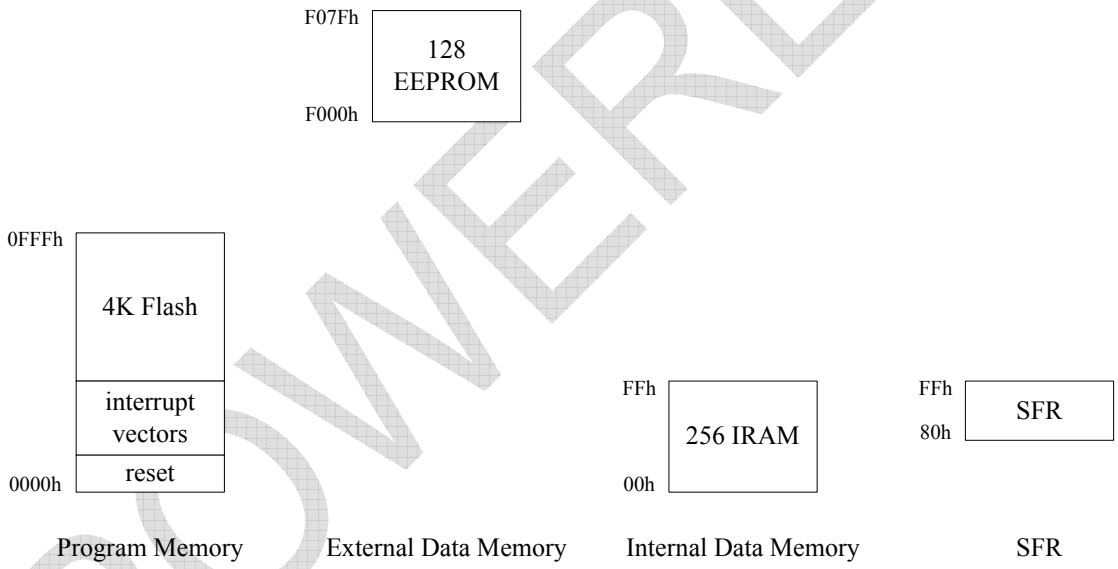



Figure 6-1 Memory Map

## 6.1 Program Memory

Program memory is used for storing program code, after reset, the CPU starts program execution at location 0000h. The lower part of program memory includes interrupt and reset vectors. The interrupt vectors are spaced at 8-byte intervals, starting from 0003h.

## 6.2 External Data Memory

External data memory is used for data storage.

The 128 bytes data memory, it's address mapped into F000h~F07Fh.

## 6.3 Data Pointer Registers

Data pointers accelerate data blocks moving. Data Pointer Register (DPTR) is a 16-bit register that is used to address external memory or peripherals.

The ET8051 includes one Data Pointer Register. The active Data Pointer Register can be accessed as SFRs: DPH, DPL.

## 6.4 Internal Data Memory

The internal data memory interface services up to 256 bytes data memory. The memory space accommodates also 128 bytes of Special Function Registers.

Addresses lower than 80h access lower 128 bytes of internal data memory. Both direct and indirect addressing can be used in this case.

Indirect addressing of locations higher than 7Fh accesses upper 128 bytes of internal data memory, while direct addressing of locations higher than 7Fh accesses SFR space.

The lower 128 bytes contain work registers (00h … 1Fh) and bit-addressable memory (20h … 2Fh). The lowest 32 bytes form four banks, each consisting of eight registers (R0-R7). Two bits of the program memory status word (PSW) select which bank is in use. The next 16 bytes of memory form a block of bit-addressable memory, accessible via 00h-7Fh addresses.

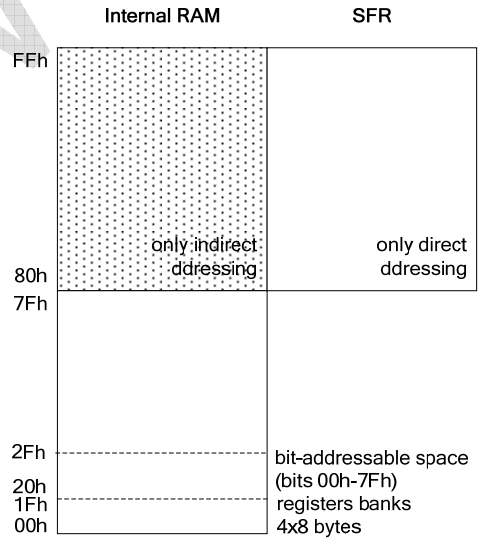



Figure 6-2 Internal Memory Map

# 7 Special Function Registers

## 7.1 Special Function Registers Locations

The map of Special Function Registers is shown in the following Table. Some addresses are occupied, while the others are not implemented. Read & write access to unimplemented addresses will target the External SFR Interface.

Table 7-1 Special Function Registers Locations

| Hex/ Bin | X000 | X001 | X010 | X011 | X100 | X101 | X110 | X111 | Bin/ Hex |
|---|---|---|---|---|---|---|---|---|---|
| **F8** | RSTCON | PWMCON1 | PWMCON0 | PWM0PL | PWM0PH | PWM0DL | PWM0DH | TA | **FF** |
| **F0** | B | | | PWM1PL | PWM1PH | PWM1DL | PWM1DH | SRST | **F7** |
| **E8** | PWM2PL | PWM2PH | PWM2DL | PWM2DH | PWM3PL | PWM3PH | PWM3DL | PWM3DH | **EF** |
| **E0** | ACC | SPSTA | SPCON | SPDAT | | | | | **E7** |
| **D8** | | | I2CDAT | I2CADR | I2CCON | I2CSTA | | | **DF** |
| **D0** | PSW | | TKWKL0 | TKWKH0 | TKWKL1 | TKWKH1 | DADAT0 | DADAT1 | **D7** |
| **C8** | T2CON | T2MOD | CRCL | CRCH | TL2 | TH2 | | | **CF** |
| **C0** | IRCON | IP1H | | | | | | | **C7** |
| **B8** | IE1 | IP1 | | | KBCON | CMPCON0 | CMPCON1 | | **BF** |
| **B0** | | IP0H | P0M1 | P0M0 | P1M0 | P1M1 | P2M0 | P2M1 | **B7** |
| **A8** | IE0 | IP0 | | | ASW0 | ASW1 | PSFT0 | PSFT1 | **AF** |
| **A0** | P2 | TKDATL | TKDATH | TKCHS0 | TKCHS1 | TKCON0 | TKCON1 | TKCON2 | **A7** |
| **98** | SCON | SBUF | | | | TKADCF | TKCSCF | TKCSOF | **9F** |
| **90** | P1 | | | | | | SCKCON | EECON | **97** |
| **88** | TCON | TMOD | TL0 | TL1 | TH0 | TH1 | CKCON | TCKCON | **8F** |
| **80** | P0 | SP | DPL | DPH | | | WDTCON | PCON | **87** |

The 16 addresses from SFR space are both byte- and bit-addressable. The bit-addressable SFRs are registers which addresses end with 000’b (80’h, 88’h, 90’h … F8’h). Those 16 registers (128 bits) together with 128 bits from internal data memory (locations 20’h … 2F’h) form the bit-addressable space.

Table 7-2 Bit-addressable Space

| Hex/ Bin | X000 | X001 | X010 | X011 | X100 | X101 | X110 | X111 | Bin/ Hex |
|---|---|---|---|---|---|---|---|---|---|
| **SFR** | | | | | | | | | |
| **F8** | | | | | | | | | **FF** |
| **F0** | | | | | | | | | **F7** |
| **E8** | | | | | | | | | **EF** |
| **E0** | | | | | | | | | **E7** |
| **D8** | | | | | | | | | **DF** |
| **D0** | | | | | | | | | **D7** |
| **C8** | | | | | | | | | **CF** |
| **C0** | | | | | | | | | **C7** |
| **B8** | | | | | | | | | **BF** |
| **B0** | | | | | | | | | **B7** |
| **A8** | | | | | | | | | **AF** |
| **A0** | | | | | | | | | **A7** |
| **98** | | | | | | | | | **9F** |
| **90** | | | | | | | | | **97** |
| **88** | | | | | | | | | **8F** |
| **80** | | | | | | | | | **87** |
| **Hex/ Bin** | **X000** | **X001** | **X010** | **X011** | **X100** | **X101** | **X110** | **X111** | **Bin/ Hex** |
| **Internal RAM** | | | | | | | | | |
| **78** | 2Fh.0 | 2Fh.1 | 2Fh.2 | 2Fh.3 | 2Fh.4 | 2Fh.5 | 2Fh.6 | 2Fh.7 | **7F** |
| **70** | 2Eh.0 | 2Eh.1 | 2Eh.2 | 2Eh.3 | 2Eh.4 | 2Eh.5 | 2Eh.6 | 2Eh.7 | **77** |
| **68** | 2Dh.0 | | | | | | | 2Dh.7 | **6F** |
| **60** | 2Ch.0 | | | | | | | 2Ch.7 | **67** |
| **58** | 2Bh.0 | | | | | | | 2Bh.7 | **5F** |
| **50** | 2Ah.0 | | | | | | | 2Ah.7 | **57** |
| **48** | 29h.0 | | | | | | | 29h.7 | **4F** |
| **40** | 28h.0 | | | | | | | 28h.7 | **47** |
| **38** | 27h.0 | | | | | | | 27h.7 | **3F** |
| **30** | 26h.0 | | | | | | | 26h.7 | **37** |
| **28** | 25h.0 | | | | | | | 25h.7 | **2F** |
| **20** | 24h.0 | | | | | | | 24h.7 | **27** |
| **18** | 23h.0 | | | | | | | 23h.7 | **1F** |
| **10** | 22h.0 | | | | | | | 22h.7 | **17** |
| **08** | 21h.0 | 21h.1 | 21h.2 | 21h.3 | 21h.4 | 21h.5 | 21h.6 | 21h.7 | **0F** |
| **00** | 20h.0 | 20h.1 | 20h.2 | 20h.3 | 20h.4 | 20h.5 | 20h.6 | 20h.7 | **07** |

## 7.2 Special Function Registers Reset Values

Table 7-3 Special Function Registers Reset Values

| SFR | ADR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | RST |
|---|---|---|---|---|---|---|---|---|---|---|
| CPU | | | | | | | | | | |
| **ACC** | E0H | ACC.7 | ACC.6 | ACC.5 | ACC.4 | ACC.3 | ACC.2 | ACC.1 | ACC.0 | 00H |
| **B** | F0H | B.7 | B.6 | B.5 | B.4 | B.3 | B.2 | B.1 | B.0 | 00H |
| **PSW** | D0H | CY | AC | F0 | RS1 | RS0 | OV | F1 | P | 00H |
| **SP** | 81H | SP.7 | SP.6 | SP.5 | SP.4 | SP.3 | SP.2 | SP.1 | SP.0 | 07H |
| **DPH** | 83H | DPH.7 | DPH.6 | DPH.5 | DPH.4 | DPH.3 | DPH.2 | DPH.1 | DPH.0 | 00H |
| **DPL** | 82H | DPL.7 | DPL.6 | DPL.5 | DPL.4 | DPL.3 | DPL.2 | DPL.1 | DPL.0 | 00H |
| **CKCON** | 8EH | CKCON.7 | CKCON.6 | CKCON.5 | CKCON.4 | CKCON.3 | CKCON.2 | CKCON.1 | CKCON.0 | 88H |
| **TA** | FFH | TA.7 | TA.6 | TA.5 | TA.4 | TA.3 | TA.2 | TA.1 | TA.0 | FFH |
| Clock Control | | | | | | | | | | |
| **SCKCON** | 96H | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00h |
| Reset Control | | | | | | | | | | |
| **RSTCON** | F8H | - | - | - | LPDF | PORF | LVRF | EXRF | WDRF | 2xh |
| **SRST** | F7H | - | - | - | - | - | - | - | D0 | 00h |
| Power Control | | | | | | | | | | |
| **PCON** | 87H | SMOD | - | ISR_TM | PMW | P2SEL | SLEEP | STOP | IDLE | 08h |
| Interrupt Control | | | | | | | | | | |
| **IE0** | A8H | EA | - | ET2 | ES0 | ET1 | EX1 | ET0 | EX0 | 00h |
| **IE1** | B8H | ET2R | - | ETK | EKB | ET2C | ECMP | ESPI | EI2C | 00h |
| **IRCON** | C0H | T2RF | TF2 | TKF | KBF | T2CF | CMPF | - | - | 00h |
| **IP0** | A9H | PEE | - | PT2 | PT2 | PT1 | PX1 | PT0 | PX0 | 00h |
| **IP0H** | B1H | PEEH | - | PT2H | PS0H | PT1H | PX1H | PT0H | PX0H | 00h |
| **IP1** | B9H | PT2R* | - | PTK | PKB | PT2C | PCMP | PSPI | PI2C | 00h |
| **IP1H** | C1H | PT2RH* | - | PTKH | PKBH | PT2CH | PCMPH | PSPIH | PI2CH | 00h |
| Keyboard Control | | | | | | | | | | |
| **KBCON** | BCH | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00h |
| Port Control | | | | | | | | | | |
| **P0** | 80H | - | - | D5 | D4 | D3 | D2 | D1 | D0 | 00h |
| **P0M0** | B2H | - | - | D5 | D4 | D3 | D2 | D1 | D0 | 00h |
| **P0M1** | B3H | - | - | D5 | D4 | D3 | D2 | D1 | D0 | 00h |

| SFR | ADR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | RST |
|---|---|---|---|---|---|---|---|---|---|---|
| **P1** | 90H | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00h |
| **P1M0** | B4H | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00h |
| **P1M1** | B5H | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00h |
| **P2** | A0H | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00h |
| **P2M0** | B6H | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00h |
| **P2M1** | B7H | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00h |
| **ASW0** | ACH | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00h |
| **ASW1** | ADH | D7 | D6 | D5- | D4 | D3 | D2 | D1 | D0 | 00h |
| **PSFT0** | AEH | SPI | I2C | UART | INT1B | INT0B | T2 | T1 | T0 | 00h |
| **PSFT1** | AFH | - | - | - | - | KEYB | TKCH | PWM1 | PWM0 | 00H |
| Timer0/1/2Control | | | | | | | | | | |
| **TCON** | 88H | TF1 | TR1 | TF0 | TR0 | IE1 | IT1 | IE0 | IT0 | 00h |
| **TMOD** | 89H | T1Gate | T1c/t | T1M1 | T1M0 | T0Gate | T0c/t | T0M1 | T0M0 | 00h |
| **TCKCON** | 8FH | - | T2PS2 | T2PS1 | T2PS0 | T1PS1 | T1PS0 | T0PS1 | T0PS0 | 4Fh |
| **TH0** | 8CH | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00h |
| **TL0** | 8AH | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00h |
| **TH1** | 8DH | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00h |
| **TL1** | 8BH | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00h |
| **T2CON** | C8H | T2EN | AES1 | AES0 | T2R1 | T2R0 | T2CM | - | - | 00h |
| **T2MOD** | C9H | - | - | - | - | - | D2 | D1 | D0 | 00h |
| **CRCH** | CBH | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00h |
| **CRCL** | CAH | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00h |
| **TH2** | CDH | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00h |
| **TL2** | CCH | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00h |
| PWM Control | | | | | | | | | | |
| **PWMCON0** | FAH | PWM1EN | PWM0EN | PWM1DS | PWM0DS | PWM1PS1 | PWM1PS0 | PWM0PS1 | PWM0PS0 | 00h |
| **PWMCON1** | F9H | PWM3EN | PWM2EN | PWM3DS | PWM2DS | PWM3PS1 | PWM3PS0 | PWM2PS1 | PWM2PS0 | 00h |
| **PWM0PH** | FCH | - | - | - | - | D3 | D2 | D1 | D0 | 00h |
| **PWM0PL** | FBH | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00h |
| **PWM0DH** | FEH | - | - | - | - | D3 | D2 | D1 | D0 | 00h |
| **PWM0DL** | FDH | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00h |
| **PWM1PH** | F4H | - | - | - | - | D3 | D2 | D1 | D0 | 00h |
| **PWM1PL** | F3H | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00h |
| **PWM1DH** | F6H | - | - | - | - | D3 | D2 | D1 | D0 | 00h |

| SFR | ADR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | RST |
|---|---|---|---|---|---|---|---|---|---|---|
| **PWM1DL** | F5H | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00h |
| **PWM2PH** | E9H | - | - | - | - | D3 | D2 | D1 | D0 | 00h |
| **PWM2PL** | E8H | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00h |
| **PWM2DH** | EBH | - | - | - | - | D3 | D2 | D1 | D0 | 00h |
| **PWM2DL** | EAH | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00h |
| **PWM3PH** | EDH | - | - | - | - | D3 | D2 | D1 | D0 | 00h |
| **PWM3PL** | ECH | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00h |
| **PWM3DH** | EFH | - | - | - | - | D3 | D2 | D1 | D0 | 00h |
| **PWM3DL** | EEH | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00h |
| WDT Control | | | | | | | | | | |
| **WDTCON** | 86H | WDTEN | - | WDTIEN | WDTIF | WDTPS3 | WDTPS2 | WDTPS1 | WDTPS0 | 00h |
| UART Control | | | | | | | | | | |
| **SCON** | 98H | SM0 | SM1 | SM2 | REN | TB8 | RB8 | TI | RI | 00h |
| **SBUF** | 99H | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00h |
| SPI Control | | | | | | | | | | |
| **SPSTA** | E1H | SPIF | WCOL | SSERR | MODF | - | - | - | - | 00h |
| **SPCON** | E2H | SPR2 | SPEN | SSDIS | MSTR | CPOL | CPHA | SPR1 | SPR0 | 14h |
| **SPDAT** | E3H | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00h |
| I2C Control | | | | | | | | | | |
| **I2CSTA** | DDH | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | F8h |
| **I2CCON** | DCH | CR2 | ENS1 | STA | STO | SI | AA | CR1 | CR0 | 00h |
| **I2CDAT** | DAH | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00h |
| **I2CADR** | DBH | ADR.6 | ADR.5 | ADR.4 | ADR.3 | ADR.2 | ADR.1 | ADR.0 | GC | 00h |
| **SPDAT** | E3H | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00h |
| Touch Key Control | | | | | | | | | | |
| **TKWKL0** | D2H | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00h |
| **TKWKH0** | D3H | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00h |
| **TKWKL1** | D4H | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00h |
| **TKWKH1** | D5H | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00h |
| **TKDATL** | A1H | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00h |
| **TKDATH** | A2H | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00h |
| **TKCHS0** | A3H | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00h |
| **TKCHS1** | A4H | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00h |
| **TKCON0** | A5H | TKEN | ADCEN | - | WAIT_CPURD | FREQ_SEL2 | FREQ_SEL1 | FREQ_SEL0 | MODE | 00h |

| SFR | ADR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | RST |
|---|---|---|---|---|---|---|---|---|---|---|
| **TKCON1** | A6H | START | ACCUM2 | ACCUM1 | ACCUM0 | AVE_DIS | TRIG_SEL | FUNC_FLG | OV | 00h |
| **TKCON2** | A7H | CAP_LP | SES_GAP2 | SES_GAP1 | SES_GAP0 | ISLE_STA | WAIT_CPU | STA_GAP1 | STA_GAP0 | 00h |
| **TKADCF** | 9DH | INJECT | ADC_PUMP | IREF_ADJ1 | IREF_ADJ0 | REF_GAP | ADC_VREF2 | ADC_VREF1 | ADC_VREF0 | 00h |
| **TKCSCF** | 9EH | C2V_CHA_SEL2 | C2V_CHA_SEL1 | C2V_CHA_SEL0 | - | - | CAP_SIZE2 | CAP_SIZE1 | CAP_SIZE0 | 00h |
| **TKCSOF** | 9FH | - | CPOL | CH_SW1 | CH_SW0 | CS_OFST3 | CS_OFST2 | CS_OFST1 | CS_OFST0 | 0Fh |
| Analog Comparator Control | | | | | | | | | | |
| **CMPCON0** | BDH | CEN | CPS | CNS | OEN | CPO | DBT | SYN | - | 00h |
| **CMPCON1** | BEH | - | TGS2 | TGS1 | TGS0 | VREF_EN | CDS2 | CDS1 | CDS0 | 00h |
| EEPROM Control | | | | | | | | | | |
| **EECON** | 97H | LOCK | FUSE | DENC | DSCR | EPGM | PGMF | CPF | PGM | 00h |

# 7.3 Special Function Registers Definition

## 7.3.1Accumulator – ACC

Table 7-4 ACC Register (E0h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| acc.7~0 | - | Accumulator | R/W | 00h |

Accumulator is used by most of the ET8051 instructions to hold the operand and to store the result of an operation. The mnemonics for accumulator-specific instructions refer to accumulator as A, not ACC.

## 7.3.2B Register – B

Table 7-5 B Register (F0h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| b.7~0 | - | B register for multiplying and division instructions | R/W | 00h |

The B register is used during multiplying and division instructions. It can also be used as a scratch-pad register to hold temporary data.

### 7.3.3Program Status Word Register - PSW

The PSW register contains status bits that reflect the current state of the CPU.

Note that the Parity bit can only be modified by hardware upon the state of ACC register.

Table 7-6 PSW Register (D0h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| psw.7 | cy | **Carry flag**<br>Carry bit in arithmetic operations and accumulator for Boolean operations. | R/W | 0 |
| psw.6 | ac | **Auxiliary Carry flag**<br>Set if there is a carry-out from 3rd bit of Accumulator in BCD operations | R/W | 0 |
| psw.5 | f0 | **General purpose Flag 0**<br>General purpose flag available for user | R/W | 0 |
| psw.4 | rs1 | **Register bank select** control bit 1, used to select working register bank | R/W | 0 |
| psw.3 | rs0 | **Register bank select** control bit 0, used to select working register bank | R/W | 0 |
| psw.2 | ov | **Overflow flag**<br>Set in case of overflow in Accumulator during arithmetic operations | R/W | 0 |
| psw.1 | f1 | **General purpose Flag 1**<br>General purpose flag available for user | R/W | 0 |
| psw.0 | p | **Parity flag**<br>Reflects the number of '1's in the Accumulator<br>P = '1' if Accumulator contains an odd number of '1's<br>P = '0' if Accumulator contains an even number of '1's | R | 0 |

The state of rs1 and rs0 bits selects the working register bank as follows:

Table 7-7 Register Bank Locations

| rs1 | rs0 | Selected Register Bank | Location |
|---|---|---|---|
| 0 | 0 | Bank 0 | (00H – 07H) |
| 0 | 1 | Bank 1 | (08H – 0FH) |
| 1 | 0 | Bank 2 | (10H – 17H) |
| 1 | 1 | Bank 3 | (18H – 1FH) |

### 7.3.4Stack Pointer - SP

Table 7-8 SP Register (81h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| sp.7~0 | - | Stack address | R/W | 07h |

This register points to the top of stack in internal data memory space.

It is used to store the return address of program before executing interrupt routine or subprograms. The SP is incremented before executing PUSH or CALL instruction and it is decremented after executing POP or RET(I) instruction (it always points the top of stack).

### 7.3.5Data Pointer – DPH, DPL

Table 7-9 DPL Register (82h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| dpl.7~0 | - | Data pointer low address | R/W | 00h |

Table 7-10 DPH Register (83h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| dph.7~0 | - | Data pointer high address | R/W | 00h |

Data Pointer Register can be accessed through DPL and DPH.

These registers are intended to hold 16-bit address in the indirect addressing mode used by MOVX (move external memory), MOVC (move program memory) or JMP (computed branch) instructions. They may be manipulated as 16-bit register or as two separate 8-bit registers. DPH holds higher byte and DPL holds lower byte of indirect address.

It is generally used to access external code or data space (e.g. MOVC A,@A+DPTR or MOV A,@DPTR respectively).

### 7.3.6Clock Control Register – CKCON

The contents of this register define the number of internally generated wait states that occur during read/write accesses to external data and program memory. It also controls the type of write access to either of the memory spaces.

The CKCON register is allocated in the SFR memory space when internal wait state generation is selected.

Table 7-11 CKCON Register (8Eh)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| ckcon.7 | - | - | R | 1 |
| ckcon.6 | - | Program memory wait state control | R/W | 0 |
| ckcon.5 | - | | | 0 |
| ckcon.4 | - | | | 0 |
| ckcon.3 | - | - | R | 1 |
| ckcon.2 | - | External data memory stretch cycle control | R/W | 0 |
| ckcon.1 | - | | | 0 |
| ckcon.0 | - | | | 0 |

### 7.3.7Timed Access Register – TA

Table 7-12 TA Register (FFh)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| TA.7~0 | - | The Timed Access register:<br>The Timed Access register controls the access to protected bits. To access protected bits, the user must first write AAH to the TA. This must be immediately followed by a write of 55H to TA. Now a window is opened in the protected bits for three machine cycles, during which the user can write to these bits | W | FFh |

The device has a new feature, like the Watchdog Timer which is a crucial to proper operation of the system. If left unprotected, errant code may write to the Watchdog control bits resulting in incorrect operation and loss of control. In order to prevent this, the device has a protection scheme which controls the write access to critical bits. This protection scheme is done using a timed access.

In this method, the bits which are to be protected have a timed write enable window. A write is successful only if this window is active, otherwise the write will be discarded. This write enable window is open for 3 machine cycles if certain conditions are met. After 3 machine cycles, this window automatically closes. The window is opened by writing AAh and immediately 55h to the Timed Access (TA) SFR. This SFR is located at address FFh. The suggested code for opening the timed access window is

```
TA   REG   0FFh          ;Define new register TA, located at 0C7h
     MOV   TA,    #0AAh
     MOV   TA,    #055h
```

When the software writes AAh to the TA SFR, a counter is started. This counter waits for 3 machine cycles looking for a write of 55h to TA. If the second write (55h) occurs within 3 machine cycles of the first write (AAh), then the timed access window is opened. It remains open for 3 machine cycles, during which the user may write to the protected bits. Once the window closes the procedure must be repeated to access the other protected bits.

Examples of Timed Assessing are shown below.

Example:

```
MOV     TA,         #0AAh
MOV     TA,         #055h
MOV     WDTCON,     #00h
```

### 7.3.8 SRST

Table 7-13 SRST egister (F7h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| srst.7~1 | - | Reserved, should be read as 0 | R/W | 00h |
| srst.0 | srst | Continuously set this bit twice involves a soft reset action, PC pointer will be reset to 0x0000, and SFR register will be reset to its reset value. | R/W | 0 |

# 8 Enhanced CPU

The ET8051 is a high performance, opcode compatible core version of the industry standard 8051 micro controller.

It provides software and hardware interrupts, interfaces for serial communication, timer system with compare-capture-reload resources, extended multiplication-division unit, multi-purpose I/O ports, watchdog timer and debugger interface.

The architecture eliminates redundant bus states and implements parallel execution of fetch and execution phases. Since a cycle is aligned with memory fetch when possible, most of the 1-byte instructions are performed in a single cycle. The ET8051 uses 1 clock per cycle.

# 9 System Clock

## 9.1 Overview

The chip has a single system clock that is generated directly from one of four selectable clock sources: on-chip crystal oscillator, on-chip ceramic resonator, internal 4/8/12 MHz RC oscillator and external clock source. The clock source is selected by the Fuses. The choice of clock source also affects the start-up time after a POR, LVR or STOP event.

In addition to this system clock, internal 32 KHz RC oscillator is used for WDT and time-out delay when power on reset.

The warm-up time is configured to maximum value to ensure that user can select all expected clock source.

## 9.2 Clock Definition

| Symbol | Description |
|---|---|
| clk_osc | **Main oscillator**<br>Four selectable clock sources: on-chip crystal oscillator, on-chip ceramic resonator, internal 4/8/12 MHz RC oscillator and external clock source. The clock source is selected by the Fuses. |
| clk_32k | **Sub-oscillator**<br>Internal 32 KHz RC oscillator is used for WDT and time-out delay when power on reset. |
| clk_sys | **System clock**<br>The system clock is generated from the prescaler clock of clk_osc. |
| clk_cpu | **CPU clock**<br>The CPU clock is generated from clk_sys. |
| clk_per | **Periphery clock**<br>The periphery clock is generated from clk_sys. |

Clock scheme within the system could see as below figure:

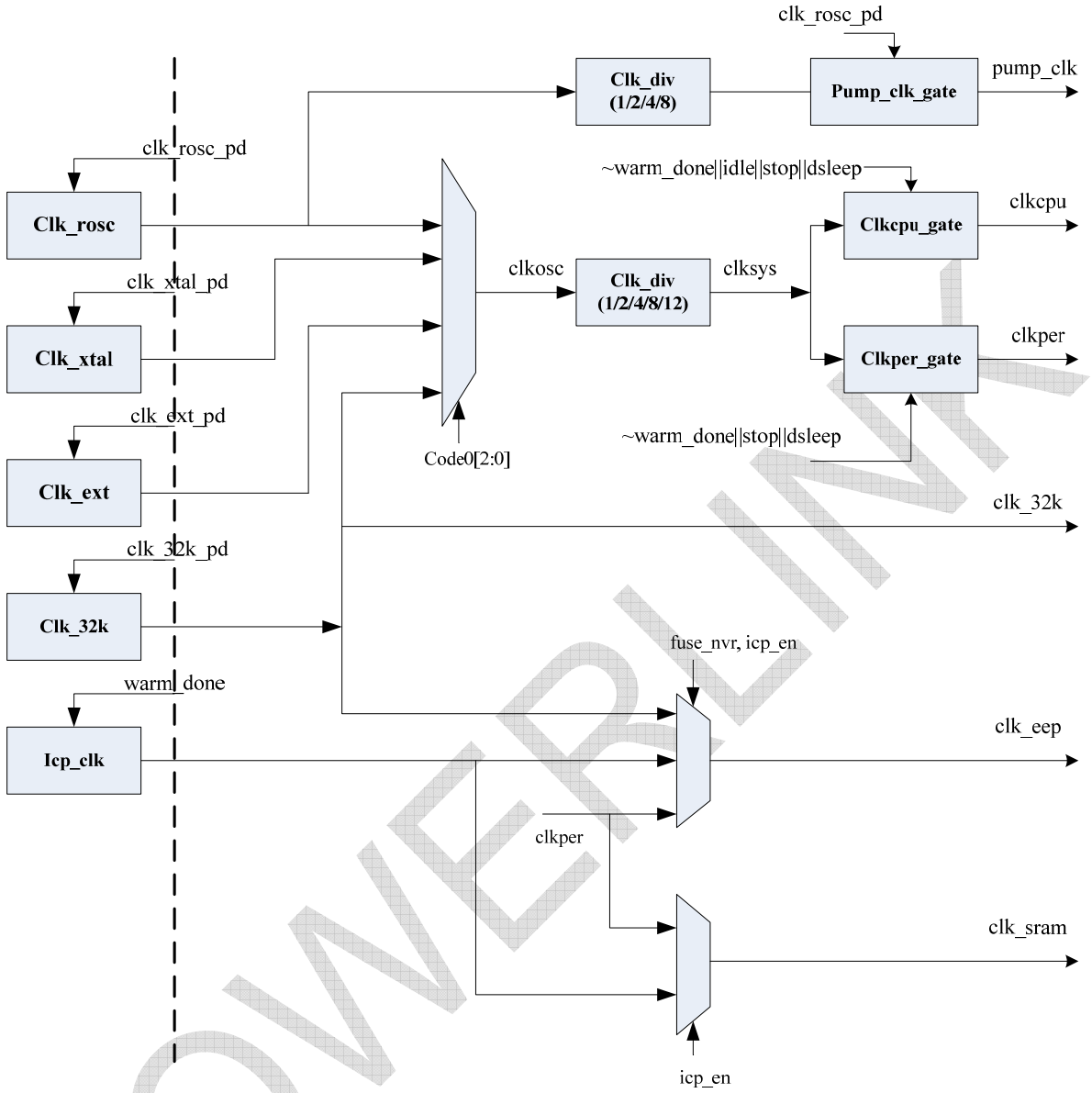



Figure 9-1 System Clock Scheme

**Note:**

1. clkcpu is gated when system enters into the idle/stop/sleep mode;
2. clkper is gated when system enters into the stop/sleep mode, but not gated in the idle mode;

## 9.3 Crystal Oscillator and Ceramic Resonator

When enabled, internal inverting oscillator amplifier is connected between XIN and XOUT for connection to an external quartz crystal or ceramic resonator. The oscillator may operate in either

high-speed or low-power mode. Low-power mode is intended for crystals of 4 MHz or less and consumes less power than the higher speed mode.

An optional 1MΩ or 4MΩ on-chip resistor can be selected to connect between XIN and XOUT. Two optional 15pF on-chip capacitors can be selected to connect between XIN/XOUT and GND. This resistor and capacitors can improve the startup characteristics of the oscillator especially at higher frequencies. The resistor and capacitors can be configured with the config options.

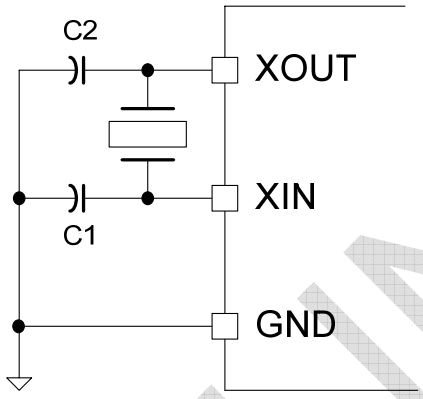



Figure 9-2 Crystal Oscillator and Ceramic Resonator

**Note:** C1, C2 = 0~15pF

## 9.4 Internal 4/8/12MHz RC Oscillator

A high accuracy internal 4/8/12MHz RC oscillator is integrated and calibrated with trimming config options.

Because EEPROM program will use the internal RC oscillator, it is always running when EEPROM is programming. When device start EEPROM program and then enter into STOP mode, the internal RC oscillator will not be disabled until EEPROM program is finished.

## 9.5 External Clock

When external clock source is selected by the config options, the external clock will be input from XIN. XOUT will be used for other pin functions.

## 9.6 Internal 32KHz RC Oscillator

The internal 32KHz RC oscillator is used for time-out delay time and WDT. When both WDT code fuse option is valid and WDTCON.WDTEN is set, it will run even device enter into STOP mode.

## 9.7 System Clock Output

When enable the oscillator clock out, the system clock can be output from P0.2. It will always output

1 during STOP mode.

## 9.8 Register Definition

### 9.8.1System Clock Prescaler Register - SCKCON

Table 9-1 SCKCON Register (96h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| sckcon.7~5 | - | - | R | 000b |
| sckcon.4 | - | - | R | 0 |
| sckcon.3 | - | - | R | 0 |
| sckcon.2 | - | System clock prescaler bits | R/TW | 0 |
| sckcon.1 | - | | | 0 |
| sckcon.0 | - | | | 0 |

The base peripheral clock is the same as the CPU clock. It is affected by the prescaler. However, individual peripherals can have their clock further modified using the prescaler in the SCKCON register. The prescaler is a 3-bit prescaler controlled by the SCKCON.

Table 9-2 System Clock Prescaler

| sckcon.2 | sckcon.1 | sckcon.0 | Prescaler |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 2 |
| 0 | 1 | 0 | 4 |
| 0 | 1 | 1 | 8 |
| 1 | 0 | 0 | 12 |
| 1 | 0 | 1 | Reserved. |
| 1 | 1 | 0 | It is the same as 000. |
| 1 | 1 | 1 | |

# 10 Reset

## 10.1 Overview

During reset, all I/O Registers are set to their initial values, the port pins are set to their default mode, and the program starts execution from the Reset Vector, 0000H. The device has four sources of reset: power-on reset, low voltage reset, external reset, hardware watchdog reset.

## 10.2 Power-on Reset

A Power-on Reset (POR) is generated by an on-chip detection circuit. The POR detection level is nominally 0.8V (Typ.) and the POE detection level is nominally 1.4V (Typ.). The POR is activated whenever VDD is below the detection level. The POR circuit can be used to trigger the start-up reset or to detect a major supply voltage failure. The POR circuit ensures that the device is reset from power-on.

When VDD reaches the Power-on Reset threshold voltage, the start-up time delay determines how long the device is kept in POR after VDD rise. The POR signal is activated again, without any delay, when VDD falls below the POR threshold level. A Power-on Reset (i.e. a cold reset) will set the PORF flag in RSTCON. The internally generated reset can be extended beyond the power-on period by holding the RST pin active longer than the time-out.

The start-up time delay is user-configurable with the start-up time fuses and depends on the clock source. The start-up time fuses also control the length of the start-up time after a Brown-out Reset or when waking up from Power-down during internally timed mode. The start-up delay should be selected to provide enough settling time for VDD and the selected clock source. The device operating environment (supply voltage, frequency, temperature, etc.) must meet the minimum system requirements before the device exits reset and starts normal operation.

Table 10-1 Time-out Delay Settings

| Time-out (CODE1.TOUT) | Time-out Clock | Time-out Delay |
|---|---|---|
| 00 | 2176 Clocks | 66 ms |
| 01 | 640 Clocks | 20 ms |
| 10 | 384 Clocks | 12 ms |
| 11 | 132 Clocks | 4 ms |

Table 10-2 Warm-up Time Settings

| Clock Source | Warm-up Time (CODE1.WARM) | | | |
|---|---|---|---|---|
| | 00 | 01 | 10 | 11 |

| Clock Source | Warm-up Time (CODE1.WARM) | | | |
|---|---|---|---|---|
| | 00 | 01 | 10 | 11 |
| Crystal Oscillator/Ceramic Resonator | 2048 Clocks | 1024 Clocks | 256 Clocks | 8 Clocks |
| Internal 8 MHz RC Oscillator | 1024 Clocks | 256 Clocks | 64 Clocks | 8 Clocks |
| External Clock | 8 Clocks | 8 Clocks | 8 Clocks | 8 Clocks |

The start-up time delay includes both time-out delay and the warm-up time when device starts up from reset. When device starts up from STOP mode, only warm-up time is needed.

The RST pin may be held active externally until these conditions are met.

## 10.3 Low Voltage Reset

The device has an on-chip low voltage reset (LVR) circuit for monitoring the VDD level during operation by comparing it to a fixed trigger level. The trigger level for the LVR is selected by config options. The purpose of the LVR is to ensure that if VDD fails or dips while executing at speed, the system will gracefully enter reset without the possibility of errors induced by incorrect execution.

A hysteresis of trigger level is designed to prevent LVR from spike. When VDD decreases to a value below the trigger level $V_{LVT}$, the internal reset is immediately activated. When VDD increases above the trigger level plus about $V_{HYST}$ of hysteresis, the device releases the internal reset after the specified time-out period has expired.

The LVR does not generate a reset output pulse except as part of a POR event.

A low voltage reset will set the LVRF flag in RSTCON.

## 10.4 Low Power Detect

The device has an on-chip low power detect (LPD) circuit for monitoring the VDD level during operation by comparing it to a fixed trigger level. The trigger level for the LPD is selected by config options. The purpose of the LPD is to ensure that if VDD fails or dips while executing at speed, the system will gracefully enter reset without the possibility of errors induced by incorrect execution.

The LPD does not generate a reset, it will only set the LPDF flag in RSTCON.

## 10.5 External Reset

The RSTB pin can function as either an active-low reset input. Entry into reset is completely asynchronous. The presence of the active reset level on the input will immediately reset the device. A glitch filter will suppress all reset input pulses of less than 50 ns. Exit from reset is synchronous. In Compatibility mode the reset pin is sampled every six clock cycles and must be held inactive for at least twelve clock cycles to deassert the internal reset. In Fast mode the reset pin is sampled every clock cycle and must be held inactive for at least two clock cycles to deassert the internal reset.

The device includes an on-chip Power-On Reset and Low Voltage Reset circuit that ensures that the device is reset from system power up. In most cases a RC startup circuit is not required on the RSTB pin, reducing system cost, and the RSTB pin may be left unconnected if a board-level reset is not present.

A external reset will set the EXRF flag in RSTCON.

## 10.6 Hardware Watchdog Reset

When the hardware watchdog times out, it will generate a reset pulse lasting 16 clock cycles. The WDT can be enabled by config options.

A hardware watchdog reset will set the WDRF flag in RSTCON.

## 10.7 Register Definition

### 10.7.1 Reset Control Register - RSTCON

Table 10-3 RSTCON Register (F8h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| rstcon.7~6 | - | - | R | 00b |
| rstcon.5 | - | - | R | 1 |
| rstcon.4 | LPDF | **Low Power Detect Flag**<br>This bit is set to 1 by hardware when a LPD is active. And it can only be cleared by software or POR | R/W | 0 |
| rstcon.3 | PORF | **Power-on Reset Flag**<br>This bit is set to 1 by hardware when a POR is active. And it can only be cleared by software. | R/W | 1 |
| rstcon.2 | LVRF | **Low Voltage Reset Flag**<br>This bit is set to 1 by hardware when a LVR is active. And it can only be cleared by POR or software. | R/W | x |
| rstcon.1 | EXRF | **External Reset Flag**<br>This bit is set to 1 by hardware when an external reset is active. And it can only be cleared by POR or software. | R/W | x |
| rstcon.0 | WDRF | **Watchdog Reset Flag**<br>This bit is set to 1 by hardware when a WDT reset is active. And it can only be cleared by POR or software.<br>Read or write this bit can clear hardware watchdog timer if WDT is enabled. | R/W | 0 |

Note:

The ’rstcon.2’ and ’rstcon.1’ perhaps read out as 1’b1 when LVR/EXT_RST are enabled in CODE

configuration, that means these resets happened once, clear these flags before using the flag is recomendded.

# 11 Power Saving Modes

## 11.1 Overview

The device supports three different software selectable power-reducing modes: IDLE, STOP and SLEEP. These modes are accessed through the PCON register.

Table 11-1 Power Saving Mode Type

| PSW types / PSW blocks | | IDLE | | STOP | | SLEEP | |
|---|---|---|---|---|---|---|---|
| | | **Clock** | **Power** | **Clock** | **Power** | **Clock*(1)** | **Power***(1) |
| Digital | CPU core | off | on | off | on | off | off |
| | Peripheral | on | on | off | on | off | off |
| Memory | EEPROM | on*(2) | on | off | on | off | off |
| | IRAM | off | on | off | on | off | off |
| Analog | Power supply | - | on | - | on | - | off |
| | IRC4/8/12Mhz | on | on | off | off | off | off |
| | IRC32Khz | on | on | off | off | off | off |
| | Oscillator | on | on | off | off | off | off |
| | ACMP | - | acmpcon0[6] | - | acmpcon0[6] | - | acmpcon0[6] |
| | CAP | on | on | on | on | on | on |

Note:

1. Power supply is about 1.8v (w/ 32K LIRC) or 1.2v (w/o 32K LIRC) in SLEEP mode, for Internal SRAM retention, and not be powered off actually.
2. The clock of EEPROM could be active even if 'cpuclk' is turnoff in IDLE mode;

## 11.2 IDLE Mode

Setting the IDLE bit in PCON enters IDLE mode. IDLE mode halts the internal CPU clock. The CPU state is preserved in its entirety, including the RAM, stack pointer, program counter, program status word, and accumulator. The Port pins hold the logic states they had at the time that IDLE was activated. IDLE mode leaves the peripherals running in order to allow them to wake up the CPU when an interrupt is generated. The timer and UART peripherals continue to function during IDLE. If these functions are not needed during IDLE, they should be explicitly disabled by clearing the appropriate control bits in their respective SFRs. The Low Voltage Reset is always active during IDLE.

Any enabled interrupt source or reset may terminate IDLE mode. When exiting IDLE mode with an interrupt, the interrupt will immediately be serviced, and following RETI the next instruction to be executed will be the one following the instruction that put the device into IDLE.

The power consumption during IDLE mode can be further reduced by prescaling down the system clock using the System Clock Prescaler. Be aware that the clock divider will affect all peripheral functions and baud rates may need to be adjusted to maintain their rate with the new clock frequency.

## 11.3 STOP Mode

Setting the STOP bit in PCON enters STOP mode. STOP mode stops the oscillator, disables the LVR and powers down the Flash memory in order to minimize power consumption. Only the power-on circuitry will continue to draw power during STOP. During STOP, the power supply voltage may be reduced to the RAM keep-alive voltage. The RAM contents will be retained, but the SFR contents are not guaranteed once VDD has been reduced. STOP could be exited by external reset, power-on reset, LVR reset, WDT reset or certain enabled interrupts.

Six enabled external interrupt sources could be configured to wakeup STOP mode directly: external interrupts INT0B and INT1B (only support level-activated), keyboard interrupt, analog comparator interrupt, touch key interrupt，WDT interrupt. When exiting STOP mode with an interrupt, the interrupt will immediately be serviced, and following RETI the next instruction to be executed will be the one following the instruction that put the device into STOP.

Note:

1) If set STOP and IDLE bits simultaneously, device will enters STOP mode. When device is waked up from STOP mode, both STOP and IDLE bits will be cleared by hardware.
2) To wake up from STOP mode, the enabled external interrupts must be active long enough for start-up time delay.
3) To wake up from STOP mode, RSTB pin must be active long enough for start-up time delay.

## 11.4 SLEEP Mode

Setting the SLEEP bit in PCON enters SLEEP mode. In SLEEP mode, reduced power supply of 1.8v to 1.2v (w/o 32K LIRC) more or less, in order to minimize power consumption but still keeping the contents stored in the IRAM, also all the other analog circuit including the Flash memory and the oscillators and Internal RC are also powered off, shut down all clocks supply.

When system entered into SLEEP mode, external interrupts INT0B and INT1B (only support level-activated), WDT interrupt, keyboard interrupt or touch key interrupt could wakeup system directly when they are enabled, and run program from the interrupt point after the system is return. Four types of reset including POR reset, LVR reset WDT reset or external reset will reset the system and run program from the start PC value.

## 11.5 Register Definition

### 11.5.1 Power Control Register - PCON

Table 11-2 PCON Register (87h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| pcon.7 | SMOD | **Serial Port 0 baud rate select**<br>(baud rate doubler) | R/W | 0 |
| pcon.6 | - | - | - | - |
| pcon.5 | ISR_TM | **Interrupt Service Routine Test Mode flag**<br>When set to 1, the interrupt vectors assigned to Timer 0 & 1, Serial Port 0 & 1, SPI and I2C interfaces can be triggered only with the use of external inputs of the core | R/TW | 0 |
| pcon.4 | - | - | R | 0 |
| pcon.3 | - | - | R | 1 |
| pcon.2 | SLEEP | **SLEEP mode control**<br>Setting this bit will auto-set STOP (pcon.1) bit, and let chip go into SLEEP mode. This bit is always read as 0 | R/TW | 0 |
| pcon.1 | STOP | **STOP mode control**<br>Setting this bit activates STOP mode. This bit is always read as 0 | R/W | 0 |
| pcon.0 | IDLE | **IDLE mode control**<br>Setting this bit activates IDLE mode. This bit is always read as 0 | R/W | 0 |

# 12 Interrupts

## 12.1 Overview

The device has four priority level interrupts structure with 14 interrupt sources. Each of the interrupt sources has an individual priority bit, flag, interrupt vector and enable bit. In addition, the interrupts can be globally enabled or disabled.

## 12.2 Interrupt Sources

The External Interrupts INT0B and INT1B can be either edge triggered or level triggered, depending on bits IT0 and IT1. The bits IE0 and IE1 in the TCON register are the flags which are checked to generate the interrupt. In the edge triggered mode, the INTx inputs are sampled in every machine cycle. If the sample is high in one cycle and low in the next, then a high to low transition is detected and the interrupts request flag IEx in TCON is set. The flag bit requests the interrupt. Since the external interrupts are sampled every machine cycle, they have to be held high or low for at least one complete machine cycle. The IEx flag is automatically cleared when the service routine is called. If the level triggered mode is selected, then the requesting source has to hold the pin low till the interrupt is serviced. The IEx flag will not be cleared by the hardware on entering the service routine. If the interrupt continues to be held low even after the service routine is completed, then the processor may acknowledge another interrupt request from the same source.

The Timer 0, 1 and 2 Interrupts are generated by the TF0, TF1 and TF2 flags. These flags are set by the overflow in the Timer 0, Timer 1 and Timer 2. The TF0 and TF1 flags are automatically cleared by the hardware when the timer interrupt is serviced, TF2 flag should be cleared by software.

The Serial block can generate interrupt on reception or transmission. There are two interrupt sources from the Serial block, which are obtained by the RI and TI bits in the SCON SFR. These bits are not automatically cleared by the hardware, and the user will have to clear these bits by software.

All the bits that generate interrupts can be set or reset by software, and thereby software initiated interrupts can be generated. Each of the individual interrupts can be enabled or disabled by setting or clearing a bit in the IEN0 and IEN1 SFR. IEN0 also has a global enable/disable bit EA, which can be cleared to disable all interrupts.

The EEPROM write finished can generate interrupt after finished EEPROM write operation. There is one interrupt source, which is obtained by the PGMF bit in the EECON SFR. This flag is automatically cleared by the hardware when the EEPROM program finish interrupt is serviced.

The analog comparator can generate interrupt after comparator output has toggle occurs by CMPF. This flag is automatically cleared by the hardware when the ACMP interrupt is serviced.

The keyboard can generate interrupt after keyboard output has toggle occurs by KBF. This bit is automatically cleared when pressed key is released.

The capsensor can generate interrupt after capsensor output has toggle occurs by TKF. This bit is not

automatically cleared by the hardware, and the user will have to clear this bit using software.

The I2C function can generate interrupt, if EI2C and EA bits are enabled, when SI Flag is set due to a new I2C status code is generated, SI flag is generated by hardware and must be cleared by software.

The SPI function can generate interrupt, if ESPI and EA bits are enabled, when 'spif' flag is set due to a new 8-bit data frame transfer completion, the flag is generated by hardware and must be cleared by software.

The WDT function can generate interrupt，if WDTIEN and EA bits are enabled, when WDTIF flag is set due to WDT time overflow ,the flag is generated by hardware and automatically cleared by the hardware when the WDT interrupt is serviced.

The interrupt flags are sampled every machine cycle. In the same machine cycle, the sampled interrupts are polled and their priority is resolved. If certain conditions are met then the hardware will execute an internally generated LCALL instruction which will vector the process to the appropriate interrupt vector address. The conditions for generating the LCALL are;

1. An interrupt of equal or higher priority is not currently being serviced.
2. The current polling cycle is the last machine cycle of the instruction currently being execute.
3. The current instruction does not involve a write to IE, EIE, IP0, IP0H, IP1 or IPH1 registers and is not a RETI.

If any of these conditions are not met, then the LCALL will not be generated. The polling cycle is repeated every machine cycle, with the interrupts sampled in the same machine cycle. If an interrupt flag is active in one cycle but not responded to, and is not active when the above conditions are met, the denied interrupt will not be serviced. This means that active interrupts are not remembered; every polling cycle is new.

The processor responds to a valid interrupt by executing an LCALL instruction to the appropriate service routine. This may or may not clear the flag which caused the interrupt. In case of Timer interrupts, the TF0, TF1 or TF2 flags are cleared by hardware whenever the processor vectors to the appropriate timer service routine. In case of external interrupts, INT0B and INT1B, the flags are cleared only if they are edge triggered. In case of Serial interrupts, the flags are not cleared by hardware. The hardware LCALL behaves exactly like the software LCALL instruction. This instruction saves the Program Counter contents onto the Stack, but does not save the Program Status Word PSW. The PC is reloaded with the vector address of that interrupt which caused the LCALL. These address of vector for the different sources are as follows:

Table 12-1 Vector Locations for Interrupt Sources

| Interrupt | Source | Vector | No. | | Interrupt | Source | Vector | No. |
|---|---|---|---|---|---|---|---|---|
| System Reset | RST | 0000H | | | | | | |
| External Interrupt0 | IE0 | 0003H | 0 | | Timer0 Overflow | TF0 | 000BH | 1 |
| External Interrupt1 | IE1 | 0013H | 2 | | Timer1 Overflow | TF1 | 001BH | 3 |
| Serial Port Interrupt | RI or TI | 0023H | 4 | | Timer2 Overflow or | TF2 | 002BH | 5 |

| Interrupt | Source | Vector | No. | | Interrupt | Source | Vector | No. |
|---|---|---|---|---|---|---|---|---|
| | TI | | | | External Reload | | | |
| I2C Interrupt | I2CF | 0033H | 6 | | Keyboard Interrupt | KBF | 003BH | 7 |
| SPI Interrupt | SPIF | 0043H | 8 | | Touch Key Interrupt | TKF(ADC) | 004BH | 9 |
| Timer2 Capture/Compare | T2CF | 0053H | 10 | | EE Write Finshed Interrupt | PGMF | 005BH | 11 |
| Comparator Interrupt | CMPF | 0063H | 12 | | Reserved | | 006BH | 13 |
| Reserved | | 0073H | 14 | | WDT interrupt | WDTIF | 007BH | 15 |
| Reserved | | 0083H | 16 | | Reserved | | 008BH | 17 |
| Reserved | | 0093H | 18 | | Reserved | | 009BH | 19 |

Execution continues from the vectored address till an RETI instruction is executed. On execution of the RETI instruction the processor pops the Stack and loads the PC with the contents at the top of the stack. The user must take care that the status of the stack is restored to what it was after the hardware LCALL, if the execution is return to the interrupted program. The processor does not notice anything if the stack contents are modified and will proceed with execution from the address put back into PC.

Note that a RET instruction would perform exactly the same process as a RETI instruction, but it would not inform the Interrupt Controller that the interrupt service routine is completed, and would leave the controller still thinking that the service routine is underway.

## 12.3 Priority Level Structure

The device uses a four priority level interrupt structure (highest, high, low and lowest) and supports up to 14 interrupt sources. The interrupt sources can be individually set to either high or low levels. Naturally, a higher priority interrupt cannot be interrupted by a lower priority interrupt. However there exists a pre-defined hierarchy amongst the interrupts themselves. This hierarchy comes into play when the interrupt controller has to resolve simultaneous requests having the same priority level. This hierarchy is defined as table below. This allows great flexibility in controlling and handling many interrupt sources.

Table 12-2 Four-level Interrupt Priority

| Priority Bits | | Interrupt Level Priority |
|---|---|---|
| IPXH | IPX | |
| 0 | 0 | Level 0 (lowest priority) |
| 0 | 1 | Level 1 |
| 1 | 0 | Level 2 |
| 1 | 1 | Level 3 (highest priority) |

Each interrupt source can be individually programmed to one of four priority levels by setting or clearing bits in the IP0, IP0H, IP1, and IP1H registers. An interrupt service routine in progress can

be interrupted by a higher priority interrupt, but not by another interrupt of the same or lower priority. The highest priority interrupt service cannot be interrupted by any other interrupt source. So, if two requests of different priority levels are received simultaneously, the request of higher priority level is serviced.

If requests of the same priority level are received simultaneously, an internal polling sequence determines which request is serviced. This is called the arbitration ranking. Note that the arbitration ranking is only used to resolve simultaneous requests of the same priority level.

As below Table summarizes the interrupt sources, flag bits, vector addresses, enable bits, priority bits, arbitration ranking, and whether each interrupt may wake up the CPU from STOP mode.

Table 12-3 Vector location for Interrupt sources and STOP mode Wakeup

| Source | Flag | Vector Address | Interrupt Enable Bits | Interrupt Priority | Flag Cleard By | Arbitration Ranking | STOP mode Wake-up |
|---|---|---|---|---|---|---|---|
| System Reset | RST | 0000H | / | / | Software | / | Yes |
| External Interrupt 0 | IE0 | 0003H | EX0 (IE0.0) | IP0H.0, IP0.0 | Hardware | 1 (highest) | Yes** |
| Timer0 Interrupt | TF0 | 000BH | ET0 (IE0.1) | IP0H.1, IP0.1 | Hardware, Software | 2 | No |
| External Interrupt 1 | IE1 | 0013H | EX1 (IE0.2) | IP0H.2, IP0.2 | Hardware | 3 | Yes** |
| Timer1 Interrupt | TF1 | 001BH | ET1 (IE0.3) | IP0H.3, IP0.3 | Hardware, Software | 4 | No |
| Serial Port Tx and Rx | TI&RI | 0023H | ES0 (IE0.4) | IP0H.4, IP0.4 | Software | 5 | No |
| Timer2 Overflow or External Reload | TF2 | 002BH | ET2(IE0.5) | IP0H.5, IP0.5 | Software | 6 | No |
| I2C Interrupt | I2CF | 0033H | EI2C (IE1.0) | IP1H.0, IP1.0 | Software | 7 | No |
| Keyboard Interrupt | KBF | 003BH | EKB (IE1.4) | IP1H.4, IP1.4 | Hardware, Software | 8 | Yes |
| SPI Interrupt | SPIF | 0043H | ESPI(IE1.1) | IP1H.1, IP1.1 | Software | 9 | No |
| Touch Key Interrupt | TKF | 004BH | ETK(IE1.5) | IP1H.5, IP1.5 | Hardware, Software | 10 | Yes |
| Timer2 Capture/Compare Interrupt | T2CF | 0053H | ET2C(IE1.3) | IP1H.5, IP1.5 | Hardware, Software | 11 | No |
| EE Write Finshed Interrupt | PGMF | 005BH | EPGM (EECON.3) | IP0H.7, IP0.7 | Hardware | 12 | No |
| Comparator | CMPF | 0063H | ECMP | IP1H.2, | Hardware, | 13 | Yes* |

| Source | Flag | Vector Address | Interrupt Enable Bits | Interrupt Priority | Flag Cleard By | Arbitration Ranking | STOP mode Wake-up |
|---|---|---|---|---|---|---|---|
| Interrupt | | | (IE1.2) | IP1.2 | Software | | |
| WDT Interrupt | WDTIF | 007BH | EWDT (wdtcon.5) | IP1H.7, IP1.7 | Hardware, Software | 14 (lowest) | Yes* |

Note:

* Only level interrupt generated from comparator could wakeup from STOP mode.

** Only low level interrupt generated from INT0 or INT1 could wakeup from STOP mode.

## 12.4 Response Time

The response time for each interrupt source depends on several factors, such as the nature of the interrupt and the instruction underway. For all of the 14 interrupt source signals, they are sampled at every system clock cycle and then their corresponding interrupt flags will be set or reset. These flag values are polled in the next cycle, and core responses it at the second cycle, interrupt spot is protect and 'PC' value is stored in the third cycle, then the interrupt vector is loaded into 'PC' in the forth cycle. So if an interrupt request is active and is selected as a active interrupt source according the configuration in SFR 'IEx', 'IPx', there is a minimum time of four system clock cycles between the interrupt flag being set and the interrupt service routine being executed.

A longer response time should be anticipated if any of the three conditions are not met. If a higher or equal priority is being serviced, then the interrupt latency time obviously depends on the nature of the service routine currently being executed.

## 12.5 Interrupt Inputs

The device has 14 interrupts source, they wire-OR together, and form one internal interrupt signal 'irq' connected to core. Only 6 of these input interrupts could be configured to wakeup the processor and resume operation when device is put into STOP or IDLE mode, they are INT0, INT1, touch key interrupt, keyboard interrupt, comparator interrupt, WDT interrupt.

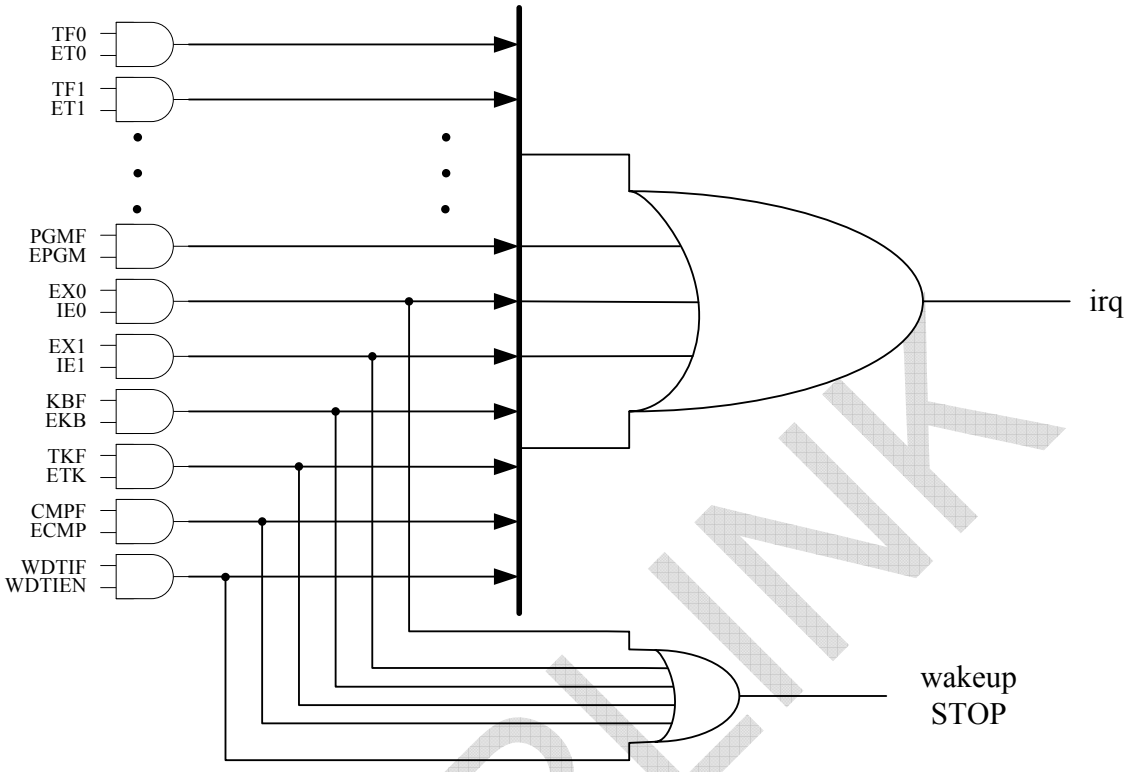



Figure 12-1 Interrupt IRQ and wakeup-STOP

Note:

1. 'irq' is interrupt signal to core, it could wakeup system when its in idle mode;
2. 'wakeup_stop' is used to wakeup system when its in stop mode;
3. WDT/keyboard/touch_key interrupt could wakeup system when its in sleep mode;

## 12.6 Register Definition

### 12.6.1 Interrupt Enable 0 Register – IE0

Table 12-4 IE0 Register (A8h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| ie0.7 | EA | **Interrupts enable**<br>When set to 0 – all interrupts are disabled<br>Otherwise enabling each interrupt is done by setting the corresponding interrupt enable bit | R/W | 0 |
| ie0.6 | - | - | R/W | 0 |
| ie0.5 | ET2 | **Timer2 interrupt enable**<br>When et2=0 timer2 interrupt is disabled. | R/W | 0 |

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| | | When et2=1 and ea=1 timer2 interrupt is enabled. | | |
| ie0.4 | ES0 | **Serial Port 0 interrupt enable**<br>When es0=0 Serial Port 0 interrupt is disabled.<br>When es0=1 and ea=1 Serial Port 0 interrupt is enabled. | R/W | 0 |
| ie0.3 | ET1 | **Timer1 overflow interrupt enable**<br>When et1=0 timer0 overflow interrupt is disabled.<br>When et1=1 and ea=1 timer1 overflow interrupt is enabled. | R/W | 0 |
| ie0.2 | EX1 | **External interrupt 1 enable**<br>When ex1=0 external interrupt 1 is disabled.<br>When ex1=1 and ea=1 external interrupt 1 is enabled. | R/W | 0 |
| ie0.1 | ET0 | **Timer0 overflow interrupt enable**<br>When et0=0 timer0 overflow interrupt is disabled.<br>When et0=1 and ea=1 timer0 overflow interrupt is enabled. | R/W | 0 |
| ie0.0 | EX0 | **External interrupt 0 enable**<br>When ex0=0 external interrupt 0 is disabled.<br>When ex0=1 and ea=1 external interrupt 0 is enabled. | R/W | 0 |

### 12.6.2 Interrupt Enable 1 Register – IE1

Table 12-5 IE1 Register (B8h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| ie1.7 | ET2R | **Timer2 external reload interrupt enable**<br>When et2r=0, timer2 external reload interrupt is disabled.<br>When et2r=1 and ea=1, timer2 external reload interrupt is enabled | R/W | 0 |
| ie1.6 | - | - | - | 0 |
| ie1.5 | ETK | **Touch key interrupt enable**<br>When etk=0 touch key interrupt is disabled.<br>When etk=1 and ea=1 touch key interrupt is enabled. | R/W | 0 |
| ie1.4 | EKB | **Keyboard interrupt enable**<br>When ekb=0 keyboard interrupt is disabled.<br>When ekb=1 and ea=1 keyboard interrupt is enabled. | R/W | 0 |
| ie1.3 | ET2C | **Timer2 capture and compare mode interrupt enable**<br>When et2c=0, timer2 capture and compare interrupt is disabled.<br>When et2c=1 and ea=1, timer2 capture and compare interrupt is enabled. | R/W | 0 |
| ie1.2 | ECMP | **Analog comparator interrupt enable**<br>When ecmp=0 comparator interrupt is disabled. | R/W | 0 |

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| | | When ecmp=1 and ea=1 comparator interrupt is enabled. | | |
| ie1.1 | ESPI | **SPI interrupt enable**<br>When espi=0 SPI interrupt is disabled.<br>When espi=1 and ea=1 SPI interrupt is enabled. | R/W | 0 |
| ie1.0 | EI2C | **I2C interrupt enable**<br>When ei2c=0 I2C interrupt is disabled.<br>When ei2c=1 and ea=1 I2C interrupt is enabled. | R/W | 0 |

### 12.6.3 Interrupt Request Control Register - IRCON

Table 12-6 IRCON Register (C0h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| ircon.7 | T2RF | Timer 2 external reload flag<br>software clear | R/W | 0 |
| ircon.6 | TF2 | Timer 2 overflow flag<br>software clear | R/W | 0 |
| ircon.5 | TKF | Touch key interrupt flag<br>Interrupt ack Hard self-clear, software clear | R/W | 0 |
| ircon.4 | KBF | Keyboard interrupt flag<br>Interrupt disappear Hard self-clear ,software clear | R/W | 0 |
| ircon.3 | T2CF | Timer2 capture and compare mode interrupt flag<br>Interrupt ack Hard self-clear, software clear | R/W | 0 |
| ircon.2 | CMPF | Analog comparator interrupt flag<br>Interrupt ack Hard self-clear, software clear | R/W | 0 |
| ircon.1 | - | - | R | 0 |
| ircon.0 | - | - | R | 0 |

### 12.6.4 Interrupt Priority 0 Register – IP0

Table 12-7 IP0 Register (A9h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| ip0.7 | PEE | EEPROM write finished interrupt priority | R/W | 0 |
| ip0.6 | - | - | R/W | 0 |
| ip0.5 | PT2 | Timer2 interrupt priority | R/W | 0 |
| ip0.4 | PS0 | Serial Port 0 interrupt priority | R/W | 0 |
| ip0.3 | PT1 | Timer1 overflow interrupt priority | R/W | 0 |

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| ip0.2 | PX1 | External interrupt 1 priority | R/W | 0 |
| ip0.1 | PT0 | Timer0 overflow interrupt priority | R/W | 0 |
| ip0.0 | PX0 | External interrupt 0 priority | R/W | 0 |

### 12.6.5 Interrupt High Priority 0 Register – IP0H

Table 12-8 IP0H Register (B1h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| ip0h.7 | PEEH | EEPROM write finished interrupt high priority | R/W | 0 |
| ip0h.6 | - | - | R/W | 0 |
| ip0h.5 | PT2H | Timer2 interrupt high priority | R/W | 0 |
| ip0h.4 | PS0H | Serial Port 0 interrupt high priority | R/W | 0 |
| ip0h.3 | PT1H | Timer1 overflow interrupt high priority | R/W | 0 |
| ip0h.2 | PX1H | External interrupt 1 high priority | R/W | 0 |
| ip0h.1 | PT0H | Timer0 overflow interrupt high priority | R/W | 0 |
| ip0h.0 | PX0H | External interrupt 0 high priority | R/W | 0 |

### 12.6.6 Interrupt Priority 1 Register – IP1

Table 12-9 IP1 Register (B9h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| ip1.7 | PWDT | WDT interrupt priority | R/W | 0 |
| ip1.6 | - | - | R/W | 0 |
| ip1.5 | PTK | Touch key interrupt priority | R/W | 0 |
| ip1.4 | PKB | Keyboard interrupt priority | R/W | 0 |
| ip1.3 | PT2C | Timer2 capture and compare mode interrupt priority | R/W | 0 |
| ip1.2 | PCMP | Analog comparator interrupt priority | R/W | 0 |
| ip1.1 | PSPI | SPI interrupt priority | R/W | 0 |
| ip1.0 | PI2C | I2C interrupt priority | R/W | 0 |

### 12.6.7 Interrupt High Priority 1 Register – IP1H

Table 12-10 IP1H Register (C1h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| ip1h.7 | PWDTH | WDT interrupt priority | R/W | 0 |
| ip1h.6 | - | - | R/W | 0 |
| ip1h.5 | PTKH | Touch key interrupt high priority | R/W | 0 |
| ip1h.4 | PKBH | Keyboard interrupt high priority | R/W | 0 |
| ip1h.3 | PT2CH | Timer2 capture and compare mode interrupt high priority | R/W | 0 |
| ip1h.2 | PCMPH | Analog comparator interrupt high priority | R/W | 0 |
| ip1h.1 | PSPIH | SPI interrupt high priority | R/W | 0 |
| ip1h.0 | PI2CH | I2C interrupt high priority | R/W | 0 |

# 13 External Interrupts

The INT0B and INT1B pins of the device may be used as external interrupt sources. The external interrupts can be programmed to be level-activated or transition activated by setting or clearing bit IT1 or IT0 in Register TCON. If ITx = 0, external interrupt x is triggered by a detected low at the INTx pin. If ITx = 1, external interrupt x is falling edge-triggered. In this mode if successive samples of the INTx pin show a high in one cycle and a low in the next cycle, interrupt request flag IEx in TCON is set. Flag bit IEx then requests the interrupt.

Since the external interrupt pins are sampled once each clock cycle, an input high or low should hold for at least 2 system periods to ensure sampling. If the external interrupt is transition-activated, the external source has to hold the request pin high for at least two clock cycles, and then hold it low for at least two clock cycles to ensure that the transition is seen so that interrupt request flag IEx will be set. IEx will be automatically cleared by the CPU when the service routine is called if generated in edge-triggered mode. If the external interrupt is level-activated, the external source has to hold the request active until the requested interrupt is actually generated. Then the external source must deactivate the request before the interrupt service routine is completed, or else another interrupt will be generated. Both INT0 and INT1 may wake up the device from the Power-down state.

# 14 Keyboard Interface

The device implements a keyboard interface allowing the connection of a 1 x n to 8 x n matrix keyboard. The keyboard function provides 8 configurable external interrupts, each key reuse with port1.0~port1.7.

When an interrupt condition on a pin is detected, and that pin is enabled in the keyboard control register (KBCON), the keyboard interrupt flag (KBF) is set. The KBF flag will be automatically cleared when pressed key is released. Any enabled keyboard interrupt may wake up the device from the IDLE, STOP or SLEEP mode.

User need to poll the keyboard pins to detect which key input is active.

## 14.1 Register Definition

### 14.1.1 Keyboard Interrupt Control Register – KBCON

Table 14-1 KBCON Register (BCh)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| kbcon.7~0 | - | Keyboard Interrupt control bits | R/W | 00h |

# 15 I/O Ports

## 15.1 Overview

The device has three I/O ports, port 0, port 1 and port 2. All pins of I/O ports can be configured by config options and port control registers. Maximal 22 general purpose I/O ports can be set.

The port output data is saved in port latch data register Px. The port modes are configured in port mode control registers PxM0 and PxM1.

When port is configured to analog functions pin by the port alternative control register ALTx, the digital input is disabled.

Some pins are shared with alternative functions, the outside pin function has the highest priority, and the inner pin function has the lowest priority. It means that if the higher priority function is enabled, the lower priority function can't be used even when the lower priority function is also enabled.

## 15.2 Port Configuration

All port pins on the device may be configured in one of four modes: input only, input with pull-up, push-pull output or open-drain output. The quasi-bidirectional of standard 8051 port output is not supported.

Each port pin also has a Schmitt-triggered input for improved input noise rejection. During STOP mode all the Schmitt-triggered inputs are disabled with the exception of INT0B, INT1B, RSTB, XIN and XOUT. The keyboard alternative pins configured as a keyboard interrupt input will also remain active during STOP mode to wake-up the device. These interrupt pins should either be disabled before entering STOP mode or they should not be left floating.

Table 15-1 Configuration Modes

| PxM0.y | PxM1.y | Port Mode |
|---|---|---|
| 0 | 0 | Input Only (High Impedance) |
| 0 | 1 | Input with Pull-up |
| 1 | 0 | Push-pull Output |
| 1 | 1 | Open-Drain Output |

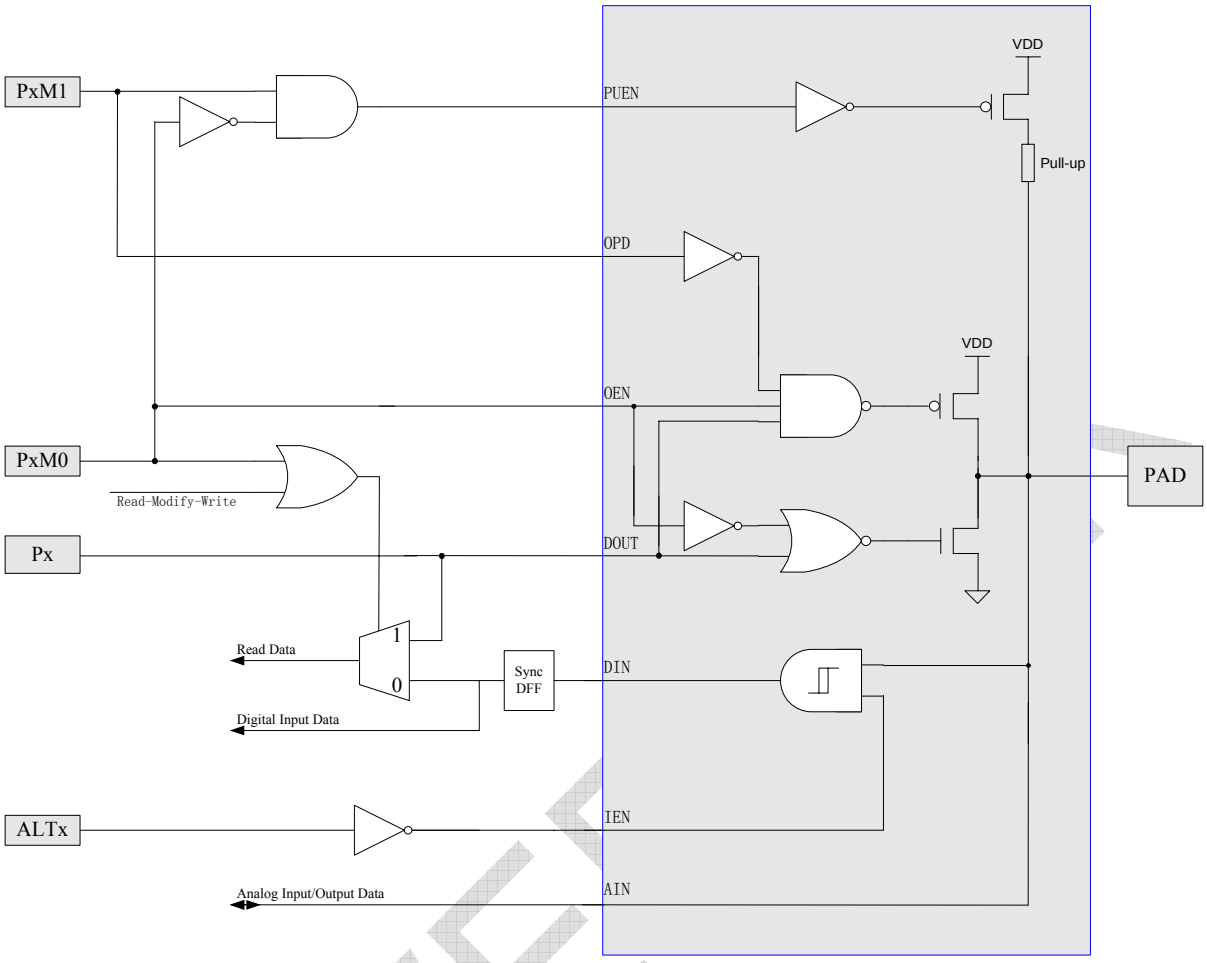



Figure 15-1 GPIO

## 15.3 Port Analog Functions

The device incorporates an analog comparator and 16-channel touch keys. In order to give the best analog performance and minimize power consumption, pins that are being used for analog functions must have both their digital outputs and digital inputs disabled.

Digital outputs are disabled by putting the port pins into the input-only mode. Digital inputs are disabled by ALTx whenever analog function pins are enabled and that pin is configured for input-only mode.

## 15.4 Port Read-Modify-Write

A read from a port will read either the state of the pins or the state of the port register depending on which instruction is used. Simple read instructions will always access the port pins directly.

Read-modify-write instructions, which read a value, possibly modify it, and then write it back, will

always access the port register. This includes bit write instructions such as CLR or SETB as they actually read the entire port, modify a single bit, then write the data back to the entire port.

## 15.5 Port Alternate Functions

Most general-purpose digital I/O pins of the device share functionality with the various I/Os needed for the peripheral units.

Table 15-2 Port Pin Alternate Functions

| Port Pin | Alternate Function | TYPE | Configuration Bits |
|---|---|---|---|
| P0.0 | MOSI | Digital(input )<br>Digital(output ) | SPCON.SPEN =1 && SPCON.MSTR =0(slave)<br>SPCON.SPEN =1 && SPCON.MSTR =1(master) |
| | PWM0* | Digital(output) | PSFT.PWM0=1 && PWMCON0.PWM0EN=1 |
| | T0 | Digital(input ) | PSFT.T0=0 |
| | TK12* | Analog(input) | PSFT.TKCH=1 && TKCHS1.4=1 |
| P0.1 | PCL | Digital(input ) | ICP mode enable |
| | PCL | Digital(input ) | CODE1.ICD=1 |
| | PWM1* | Digital(output) | PSFT.PWM1=1 && PWMCON0.PWM1EN=1 |
| | T1 | Digital(input ) | PSFT.T1=0 |
| | INT0B* | Digital(input ) | PSFT.INT0B=1 |
| | TK13* | Analog(input) | PSFT.TKCH=1 && TKCHS1.5=1 |
| P0.2 | CLKOUT | Digital(output) | CODE0.CLKOEN=1 |
| | SCSB | Digital(input ) | SPCON.SSDIS =0 |
| | T2EX | Digital(input ) | PSFT.T2=0 |
| | INT1B* | Digital(input ) | PSFT.INT1B=1 |
| | TK14* | Analog(input) | PSFT.TKCH=1 && TKCHS1.6=1 |
| P0.3 | XOUT | Analog(output) | CODE0.OSCSEL=3'b01x |
| | SDA | Digital(IO) | I2CCON. ENS1=1 |
| | MISO | Digital(input )<br>Digital(output ) | SPCON.SPEN =1 && SPCON.MSTR =1(master)<br>SPCON.SPEN =1 && SPCON.MSTR =0(slave) |
| | TXD | Digital(output) | SCON.REN=1 |

| Port Pin | Alternate Function | TYPE | Configuration Bits |
|---|---|---|---|
| | | | PSFT.UART=0 |
| | T2CPO | Digital(output) | T2CON.T2EN=1<br>PSFT.T2=0<br>T2MOD<2:0>=3'b10x |
| | TK15* | Analog(input) | PSFT.TKCH=1 && TKCHS1.7=1 |
| P0.4 | XIN | Analog(input) | CODE0.OSCSEL=3'b0xx |
| | SCL | Digital(IO) | I2CCON. ENS1=1 |
| | SCK | Digital(output )<br>Digital(input ) | SPCON.SPEN =1 && SPCON.MSTR =1(master)<br>SPCON.SPEN =1 && SPCON.MSTR =0(slave) |
| | RXD | Digital(input ) | SCON.REN=1<br>PSFT.UART=0 |
| P0.5 | PDA | Digital(output) | ICP mode enable |
| | PDA | Digital(output) | CODE1.ICD=1 |
| | RSTB | Analog(input) | CODE1.RST=1 |
| P1.0 | MOSI* | Digital(input )<br>Digital(output ) | SPCON.SPEN =1 && SPCON.MSTR=0(slave) && PSFT0.SPI =1<br>SPCON.SPEN =1 && SPCON.MSTR=1(master)&& PSFT0.SPI =1 |
| | PWM0 | Digital(output) | PWMCON0.PWM0EN=1 |
| | TK0 | Analog(input) | TKCON0.TKEN=1 && TKCON1.START=1 &&<br>TKCHS0.0=1<br>ASW0.KEY0=1 |
| P1.1 | SCSB* | Digital(input ) | SPCON.SSDIS =0 && PSFT0.SPI =1 |
| | PWM1 | Digital(output) | PWMCON0.PWM1EN=1 |
| | TK1 | Analog(input) | TKCON0.TKEN=1 && TKCON1.START=1 &&<br>TKCHS0.1=1<br>ASW0.KEY1=1 |
| P1.2 | SDA* | Digital(IO) | I2CCON. ENS1 =1 && PSFT.IIC =1 |
| | MISO* | Digital(input ) | SPCON.SPEN=1 && SPCON.MSTR=1(master) && |

| Port Pin | Alternate Function | TYPE | Configuration Bits |
|---|---|---|---|
| | | Digital(output ) | PSFT.SPI=1<br>SPCON.SPEN=1 && SPCON.MSTR=0(slave) && PSFT.SPI=1 |
| | TXD* | Digital(output) | SCON.REN=1<br>PSFT.UART=1 |
| | TK2 | Analog(input) | TKCON0.TKEN=1 && TKCON1.START=1 && TKCHS0.2=1<br>ASW0.KEY2=1 |
| P1.3 | SCL* | Digital(IO) | I2CCON. ENS1 =1 && PSFT0.IIC=1 |
| | SCK* | Digital(output ) | SPCON.SPEN=1 && SPCON.MSTR=1(master) && PSFT0.SPI=1 |
| | | Digital(input ) | SPCON.SPEN=1 && SPCON.MSTR=0(slave) && PSFT0.SPI=1 |
| | RXD* | Digital(output) | SCON.REN=1<br>PSFT.UART=1 |
| | TK3 | Analog(input) | TKCON0.TKEN=1 && TKCON1.START=1 &&<br>TKCHS0.3=1<br>ASW0.KEY3=1 |
| P1.4 | CMPOUT | Analog(output) | CMPCON0.CMPEN=1 |
| | T0* | Digital(input) | PSFT.T0=1 |
| | TK4 | Analog(input) | TKCON0.TKEN=1 && TKCON1.START=1 &&<br>TKCHS0.4=1<br>ASW0.KEY4=1 |
| P1.5 | ADVREF | Analog(input) | ADCSEL !=00 && ADCON0.ADCEN=1 |
| | CMPVREF | Analog(input) | CMPCON0.CMPEN=1<br>CMPCON0.CMPVREF=1 |
| | T1* | Digital(input) | PSFT.T1=1 |
| | TK5 | Analog(input) | TKCON0.TKEN=1 && TKCON1.START=1 &&<br>TKCHS0.5=1 |

| Port Pin | Alternate Function | TYPE | Configuration Bits |
|---|---|---|---|
| | | | ASW0.KEY5=1 |
| P1.6 | CMPN | Analog(input) | CMPCON0.CMPEN=1<br>CMPCON0.CMPVREF=0 |
| | T2EX* | Digital(input ) | PSFT.T2=1 |
| | INT0B | Digital(input) | PSFT.INT0B=0 |
| | TK6 | Analog(input) | TKCON0.TKEN=1 && TKCON1.START=1 &&<br>TKCHS0.6=1<br>ASW0.KEY6=1 |
| P1.7 | CMPP | Analog(input) | CMPCON0.CMPEN=1<br>CMPCON0.CMPIBG=0 |
| | T2CPO* | Digital(output) | T2CON.T2EN=1<br>PSFT.T2=1<br>T2MOD<2:0>=3'b10x |
| | INT1B | Digital(input) | PSFT.INT1B=0 |
| | TK7 | Analog(input) | TKCON0.TKEN=1 && TKCON1.START=1 &&<br>TKCHS0.7=1<br>ASW0.KEY7=1 |
| P2.0 | PWM2 | Digital(output) | PWMCON1.PWM2EN=1 |
| | AD0 | Analog(input) | TKCON0.ADCEN=1 && TKCON1.START=1 &&<br>TKCHS1.0=1 |
| | TK8 | Analog(input) | TKCON0.TKEN=1 && TKCON1.START=1 &&<br>TKCHS1.0=1<br>ASW1.KEY8=1 |
| P2.1 | PWM3 | Digital(output) | PWMCON1.PWM3EN=1 |
| | AD1 | Analog(input) | TKCON0.ADCEN=1 && TKCON1.START=1 &&<br>TKCHS1.1=1 |
| | TK9 | Analog(input) | TKCON0.TKEN=1 && TKCON1.START=1 &&<br>TKCHS1.1=1 |

| Port Pin | Alternate Function | TYPE | Configuration Bits |
|---|---|---|---|
| | | | ASW1.KEY9=1 |
| P2.2 | AD2 | Analog(input) | TKCON0.ADCEN=1 && TKCON1.START=1 && TKCHS1.2=1 |
| | TK10 | Analog(input) | TKCON0.TKEN=1 && TKCON1.START=1 && TKCHS1.2=1 && ASW1.KEY10=1 |
| P2.3 | AD3 | Analog(input) | TKCON0.ADCEN=1 && TKCON1.START=1 && TKCHS1.3=1 |
| | TK11 | Analog(input) | TKCON0.TKEN=1 && TKCON1.START=1 && TKCHS1.3=1 && ASW1.KEY11=1 |
| P2.4 | AD4 | Analog(input) | TKCON0.ADCEN=1 && TKCON1.START=1 && TKCHS1.4=1 |
| | TK12 | Analog(input) | TKCON0.TKEN=1 && TKCON1.START=1 && TKCHS1.4=1 && PSFT.TKCH=0 && ASW1.KEY12=1 |
| P2.5 | AD5 | Analog(input) | TKCON0.ADCEN=1 && TKCON1.START=1 && TKCHS1.5=1 |
| | TK13 | Analog(input) | TKCON0.TKEN=1 && TKCON1.START=1 && TKCHS1.5=1 && PSFT.TKCH=0 && ASW1.KEY13=1 |
| P2.6 | AD6 | Analog(input) | TKCON0.ADCEN=1 && TKCON1.START=1 && TKCHS1.6=1 |
| | TK14 | Analog(input) | TKCON0.TKEN=1 && TKCON1.START=1 && TKCHS1.6=1 && PSFT.TKCH=0 && ASW1.KEY14=1 |
| P2.7 | AD7 | Analog(input) | TKCON0.ADCEN=1 && TKCON1.START=1 && TKCHS1.7=1 |
| | TK15 | Analog(input) | TKCON0.TKEN=1 && TKCON1.START=1 && TKCHS1.7=1 && PSFT.TKCH=0 && ASW1.KEY15=1 |

Note: The above alternate function has high priority, when it is enabled the below alternate function will be disabled.

## 15.6 Register Definition

### 15.6.1 Port 0 Data Register – P0

Table 15-3 P0 Data Register (80h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| p0.7~6 | - | - | R | 2'b0 |
| p0.5~0 | - | Data register | R/W | 6'b0 |

### 15.6.2 Port 0 Control Registers – P0M0/P0M1

Table 15-4 P0 Control Register 0 – P0M0 (B2h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| p0m0.7~6 | - | - | R | 2'b0 |
| p0m0.5~0 | - | Control register 0 | R/W | 6'b0 |

Table 15-5 P0 Control Register 1 – P0M1 (B3h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| p0m1.7~6 | - | - | R | 2'b0 |
| p0m1.5~0 | - | Control register 1 | R/W | 6'b0 |

### 15.6.3 Port 1 Data Register – P1

Table 15-6 P1 Data Register (90h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| p1.7~0 | - | Data register | R/W | 00h |

### 15.6.4 Port 1 Control Registers – P1M0/P1M1

Table 15-7 P1 Control Register 0 – P1M0 (B4h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| p1m0.7~0 | - | Control register 0 | R/W | 00h |

Table 15-8 P1 Control Register 1 – P1M1 (B5h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| p1m1.7~0 | - | Control register 1 | R/W | 00h |

### 15.6.5 Port 2 Data Register – P2

Table 15-9 P2 Data Register (A0h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| p2.7~0 | - | Data register | R/W | 00h |

### 15.6.6 Port 2 Control Registers – P2M0/P2M1

Table 15-10 P2 Control Register 0 – P2M0 (B6h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| p2m0.7~0 | - | Control register 0 | R/W | 00h |

Table 15-11 P2 Control Register 1 – P2M1 (B7h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| p2m1.7~0 | - | Control register 1 | R/W | 00h |

### 15.6.7 Port Analog Switch Registers – ASW0/ASW1

The digital input enable of port1/port2 could be closed or opened accroding to configuration bit of ASW0/ASW1, one bit of asw corresponding to one pad, for example, set asw0.7 will close port1.7 digital input.

Table 15-12 Port Analog Switch Register 0 – ASW0 (ACh)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| asw0.7~0 | - | Analog switch register 0, control port1.7~0 | R/W | 00h |

Table 15-13 Port Analog Switch Register 1 – ASW1 (ADh)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| asw1.7~0 | - | Analog switch register 1, control port2.7~0 | R/W | 00h |

### 15.6.8 Port shift Register – PSFT0

Table 15-14 Port Shift 0 Register (AEh)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| psft0.7 | SPI | SCSB/SCK/MISO/MOSI shift control bit | R/W | 1'b0 |
| psft0.6 | I2C | SCL/SDA shift control bit | R/W | 1'b0 |
| psft0.5 | UART | RXD/TXD shift control bit | R/W | 1'b0 |
| psft0.4 | INT1B | INT1B shift control bit | R/W | 1'b0 |
| psft0.3 | INT0B | INT0B shift control bit | R/W | 1'b0 |
| psft0.2 | T2 | T2EX/T2CPO shift control bit | R/W | 1'b0 |
| psft0.1 | T1 | Timer 1 input shift control bit | R/W | 1'b0 |
| psft0.0 | T0 | Timer 0 input shift control bit | R/W | 1'b0 |

### 15.6.9 Port shift Register – PSFT1

Table 15-15 Port Shift 1 Register (AFh)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| psft 1.7~4 | - | - | R | 4'b0 |
| psft 1.3 | KEYB | Keyboard 7~0 shift control bit | R/W | 1'b0 |
| psft 1.2 | TKCH | Touch Key channel 15~12 shift control bit | R/W | 1'b0 |
| psft 1.1 | PWM1 | PWM1 shift control bit | R/W | 1'b0 |
| psft 1.0 | PWM0 | PWM0 shift control bit | R/W | 1'b0 |

# 16 Timer 0 and Timer 1

## 16.1 Overview

The device has two 16-bit Timer/Counters, Timer 0 and Timer 1.

In the timer mode, the Timer 0/1 is incremented at each active positive edge of clock which is pre-scale of system clock, the pre-scale bits store in the TCON register.

In the counter mode, the Timer 0/1 is incremented when the falling edge is detected at the corresponding input pin – "t0" for Timer 0, "t1" for Timer 1. Since it takes 2 clock cycles to recognize a 1-to-0 event, the maximum input count rate is 1/2 of the oscillator frequency. There are no restrictions on the duty cycle, however to ensure proper recognition of 0 or 1 state, an input should be stable for at least 1 clock cycle.

Four operating modes can be selected for Timer 0/1. Two Special Function Registers: tmod and tcon are used to select the appropriate mode.

## 16.2 Mode 0 and Mode 1

In mode 0, Timer 0 is configured as a 13-bit register ("tl0" = 5 bits, "th0" = 8 bits). The upper 3 bits of "tl0" are unchanged and should be ignored.

In mode 1, Timer 0 is configured as a 16- bit register.

In mode 0, Timer 1 is configured as a 13-bit register ("tl1" = 5 bits, "th1" = 8 bits). The upper 3 bits of "tl1" are unchanged and should be ignored.

In mode 1, Timer 1 is configured as a 16- bit register.

Timer 1 in Mode 0 or Mode 1 are the same as Timer 0, all have 13-bit prescaler.

The timer clock source is controlled by TCKCON.TPSx.

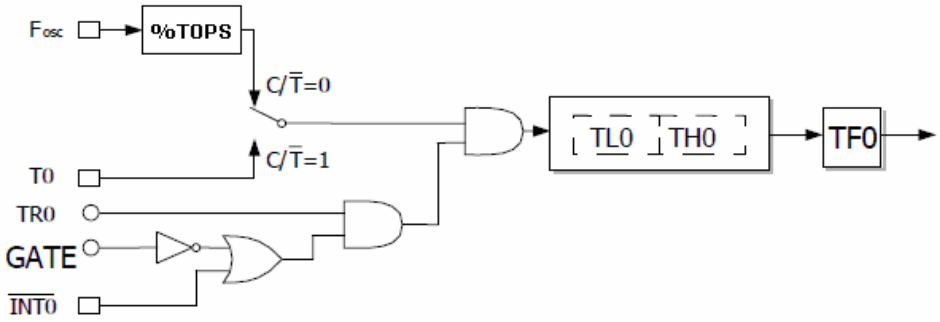



Figure 16-1 Timer 0 in Mode 0 and 1

## 16.3 Mode 2

In this mode the Timer 0 and Timer 1 are configured as an 8-bit register with auto-reload.

Timer 1 in Mode 2 are the same as Timer 0.

The timer clock source is controlled by TCKCON.TPSx.

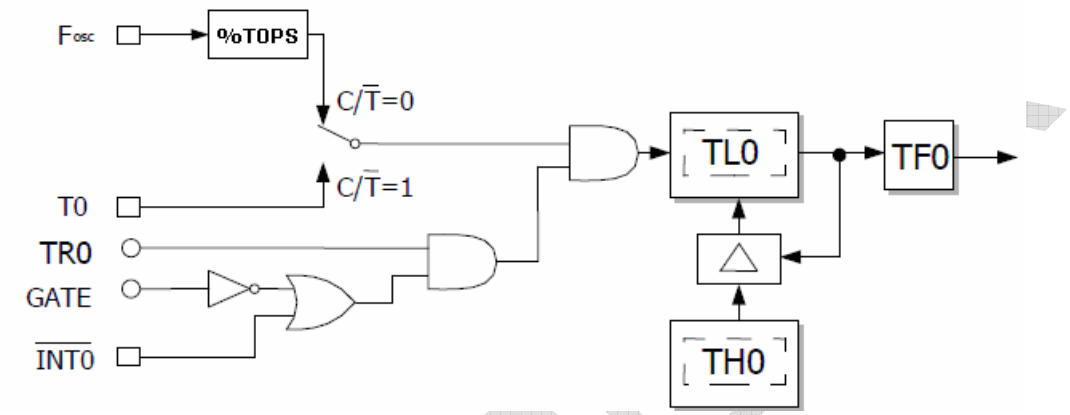



Figure 16-2 Timer 0 in Mode 2

## 16.4 Mode 3

In mode 3, Timer 0 is configured as one 8-bit timer/counter and one 8-bit timer.

When Timer 0 works in mode 3, Timer 1 can still be used in other mode by the serial port as a baud rate generator, or application not requiring an interrupt from Timer 1.

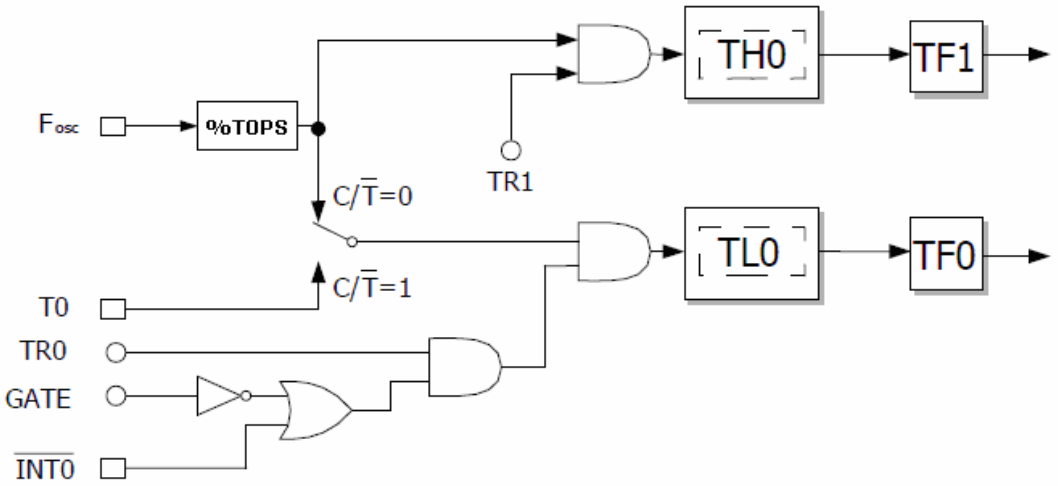



Figure 16-3 Timer 0 in Mode 3

## 16.5 Register Definition

### 16.5.1 Timer/Counter Control Register - TCON

Table 16-1 TCON Register (88h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| tcon.7 | tf1 | **Timer 1 overflow flag**<br>Bit set by hardware when Timer1 overflows. This flag can be cleared by software and is automatically cleared when interrupt is processed. | R/W | 0 |
| tcon.6 | tr1 | **Timer1 Run control**<br>If cleared, Timer 1 stops. | R/W | 0 |
| tcon.5 | tf0 | **Timer 0 overflow flag**<br>Bit set by hardware when Timer 0 overflows. This flag can be cleared by software and is automatically cleared when interrupt is processed. | R/W | 0 |
| tcon.4 | tr0 | **Timer 0 Run control**<br>If cleared, Timer 0 stops. | R/W | 0 |
| tcon.3 | ie1 | **External interrupt 1 flag**<br>Set by hardware, when external interrupt int1 (edge/level, depending on settings) is observed. Cleared by hardware when interrupt is processed. | R/W | 0* |
| tcon.2 | it1 | **External interrupt 1 type control**<br>If set, external interrupt 1 is activated at falling edge on input pin. If cleared, external interrupt 1 is activated at low level on input pin. | R/W | 0 |
| tcon.1 | ie0 | **External interrupt 0 flag**<br>Set by hardware, when external interrupt int0 (edge/level, depending on settings) is observed. Cleared by hardware when interrupt is processed. | R/W | 0* |
| tcon.0 | it0 | **External interrupt 0 type control**<br>If set, external interrupt 0 is activated at falling edge on input pin. If cleared, external interrupt 0 is activated at low level on input pin. | R/W | 0 |

Note:

After power on, the register perhaps read out as ’0Ah’ when the INT0B/INT1B port is input floating, that means there are interrupt flags for INT0B/INT1B, if happenning, clear these flags before using the INT0B/INT1B interrupt is strongly recomendded;

### 16.5.2 Timer Mode Register - TMOD

Table 16-2 TMOD Register (89h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| tmod.7 | Gate | **Timer 1 gate control**<br>If set, enables external gate control (pin "int(1)") for Counter 1. When "int(1)" is high, and "tr1" bit is set (Table 18), the Counter 1 is incremented every falling edge on "t1" input pin | R/W | 0 |
| tmod.6 | c/t | **Timer 1 counter/timer select**<br>Selects Timer or Counter operation. When set to 1, a Counter operation is performed, when cleared to 0, the Timer/Counter 1 will function as a Timer. | R/W | 0 |
| tmod.5 | m1 | **Timer 1 mode**<br>Selects mode for Timer/Counter 1, as shown in table below. | R/W | 0 |
| tmod.4 | m0 | | R/W | 0 |
| tmod.3 | Gate | **Timer 0 gate control**<br>If set, enables external gate control (pin "int(0)") for Counter 0. When "int(0)" is high, and "tr0" bit is set (Table 18), the Counter 0 is incremented every falling edge on "t0" input pin | R/W | 0 |
| tmod.2 | c/t | **Timer 0 counter/timer select**<br>Selects Timer or Counter operation. When set to 1, a Counter operation is performed, when cleared to 0, the Timer/Counter 0 will function as a Timer. | R/W | 0 |
| tmod.1 | m1 | **Timer 0 mode**<br>Selects mode for Timer/Counter 0, as shown in table below. | R/W | 0 |
| tmod.0 | m0 | | R/W | 0 |

Table 16-3 Timers/Counters Modes

| M1 | M0 | Mode | Function |
|---|---|---|---|
| 0 | 0 | Mode 0 | 13-bit Counter/Timer, with 5 lower bits in tl0 (tl1) register and 8 bits in th0 (th1) register (for Timer 0 or Timer 1, respectively). Note, that unlike in 80C51, the 3 high-order bits of tl0 (tl1) are zeroed whenever Mode 0 is enabled. |
| 0 | 1 | Mode 1 | 16-bit Counter/Timer. |
| 1 | 0 | Mode 2 | 8-bit auto-reload Counter/Timer. The reload value is kept in th0 (th1), while tl0 (tl1) is incremented every machine cycle. When tl0 (tl1) overflows, a value from th0 (th1) is copied to tl0 (tl1). |
| 1 | 1 | Mode 3 | For Timer1: Timer1 is stopped.<br>For Timer0: Timer 0 acts as two independent 8 bit Timers/Counters – tl0, th0.<br>- tl0 uses the Timer0 control bits and sets tf0 flag on overflow |

| M1 | M0 | Mode | Function |
|---|---|---|---|
| | | | - th0 operates as timer1. It is enabled by tr1 bit and sets tf1 flag on overflow. |

### 16.5.3 Timer Clock Prescaler Register - TCKCON

Table 16-4 TCKCON Register (8Fh)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| tckcon.7 | - | - | R | 0 |
| tckcon.6 | t2ps2 | **Timer 2 Prescaler selection** | R/W | 1 |
| tckcon.5 | t2ps1 | | R/W | 0 |
| tckcon.4 | t2ps0 | | R/W | 0 |
| tckcon.3 | t1ps1 | **Timer 1 Prescaler selection** | R/W | 1 |
| tckcon.2 | t1ps0 | | R/W | 1 |
| tckcon.1 | t0ps1 | **Timer 0 Prescaler selection** | R/W | 1 |
| tckcon.0 | t0ps0 | | R/W | 1 |

Table 16-5 Timer 2 Clock Prescaler Selection

| T2PS2 | T2PS1 | T2PS0 | Timer 2 Clock Prescaler Selection |
|---|---|---|---|
| 0 | 0 | 0 | RSV |
| 0 | 0 | 1 | clk_sys/2 |
| 0 | 1 | 0 | clk_sys/4 |
| 0 | 1 | 1 | clk_sys/8 |
| 1 | 0 | 0 | clk_sys/12 |
| 1 | 0 | 1 | clk_sys/16 |
| 1 | 1 | 0 | clk_sys/32 |
| 1 | 1 | 1 | clk_sys/128 |

Table 16-6 Timer 1/0 Clock Prescaler Selection

| T1PS1 | T1PS0 | Timer 1 Clock Prescaler | | T0PS1 | T0PS0 | Timer 0 Clock Prescaler |
|---|---|---|---|---|---|---|
| 0 | 0 | RSV | | 0 | 0 | RSV |
| 0 | 1 | clk_sys/4 | | 0 | 1 | clk_sys/4 |
| 1 | 0 | clk_sys/8 | | 1 | 0 | clk_sys/8 |
| 1 | 1 | clk_sys/12 | | 1 | 1 | clk_sys/12 |

### 16.5.4 Timer 0 Registers – TH0/TL0

Table 16-7 TH0 Register (8Ch)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| th0.7~0 | - | Timer 0 higher byte | R/W | 00h |

Table 16-8 TL0 Register (8Ah)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| tl0.7~0 | - | Timer 0 lower byte | R/W | 00h |

### 16.5.5 Timer 1 Registers – TH1/TL1

Table 16-9 TH1 Register (8Dh)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| th1.7~0 | - | Timer 1 higher byte | R/W | 00h |

Table 16-10 TL1 Register (8Bh)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| tl1.7~0 | - | Timer 1 lower byte | R/W | 00h |

# 17 Timer 2

## 17.1 Overview

The device includes a 16-bits Timer with Compare/Capture and PWM.

## 17.2 Timer Function

The Timer 2 can operate as 16-bits timer.

### 17.2.1 Timer Mode

In this mode, the count rate is derived from the prescaler of system clock. The prescaler mode is controlled by TCKCON.T2PS. Set T2CON.T2EN to start timer counting.

### 17.2.2 Timer 2 Reload Mode

A 16-bits reload from the Timer2 Compare/Capture registers can be executed in two reload modes:

- Mode 0: Reload signal is generated by Timer 2 overflow (auto reload)
- Mode 1: Reload signal is generated by negative transition at the corresponding input pin T2EX.

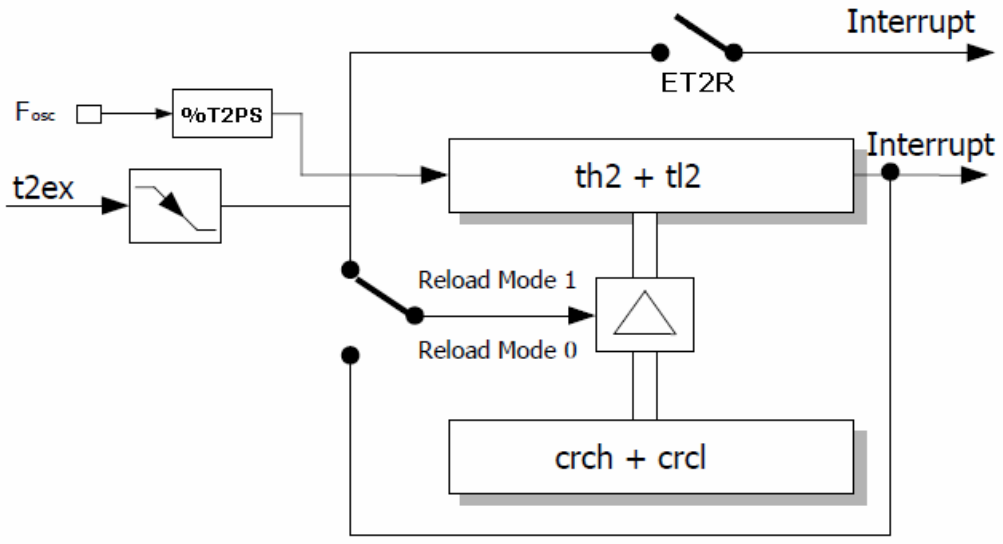



Figure 17-1 Timer 2 in Reload Mode

## 17.3 Compare Function

The Compare/Capture consists of one 16-bits Timer 2 compare data register. The register can be configured to work in comparator mode. In this mode the value stored in register is compared with the contents of the Timer2. The comparators outputs drive the T2CPO pin, where the T2CPO is output of the comparator associated with the register CRCH/CRCL.

There are two compare modes selected by T2CON.T2CM.

### 17.3.1 Compare Mode 0

In mode 0, when the value in Timer 2 equals the value of the compare register, the comparator output changes from low to high. It goes back low on timer 2 overflow.

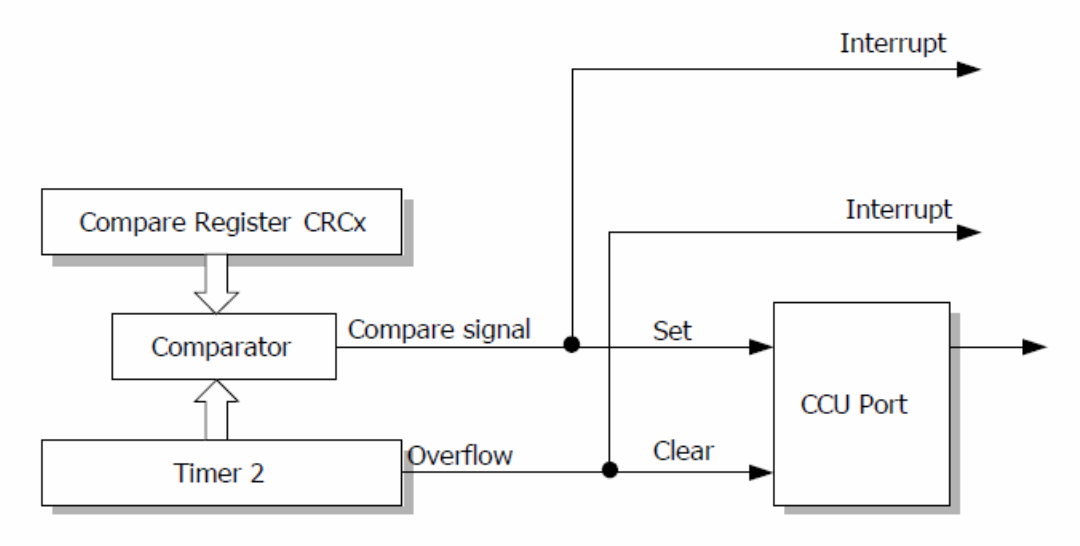



Figure 17-2 Timer 2 in Compare Mode 0

### 17.3.2 Compare Mode 1

In compare mode 1, the transition of the output signal can be determined by software. A Timer 2 overflow causes no output change. In this mode both transitions of output signal can be controlled. Figure below shows a functional diagram of a register/port configuration in compare mode 1. In compare mode 1 the value is written first to the “Shadow Register (p0.3/p1.7)”, and when the compare signal goes active this value is transferred to the output register.

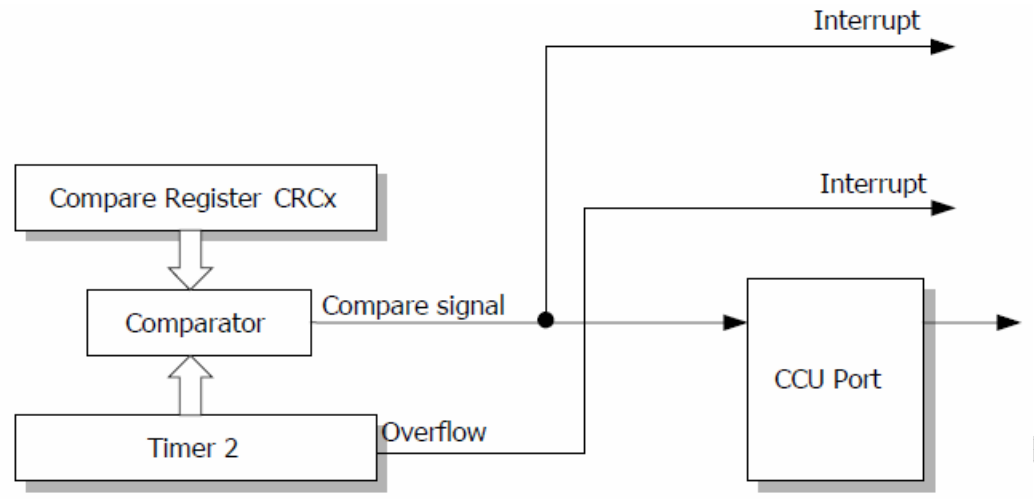



Figure 17-3 Timer 2 in Compare Mode 1

## 17.4 Capture Function

The 16-bits CRCH/CRCL register can be configured to work in capture mode.

In this mode the actual timer/counter contents are saved into the CRCH/CRCL register upon an external event (mode 0) or a software write operation (mode 1).

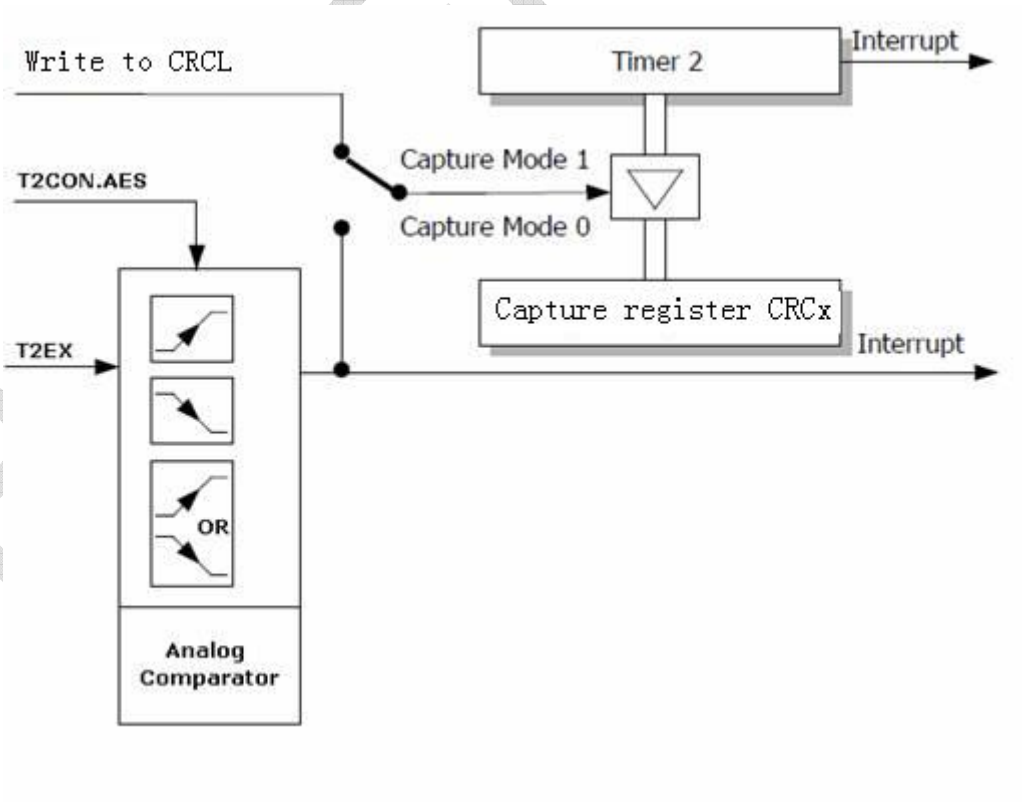



Figure 17-4 Timer 2 in Capture Mode

### 17.4.1 Capture Mode 0

In mode 0 capturing of Timer 2 contents is executed when rising edge, falling edge, toggle edge on T2EX or analog comparator output trigger depending on the T2CON.AES.
The timer 2 contents will be latched into appropriate capture register CRCH/CRCL. In this mode no interrupt request will be generated.

### 17.4.2 Capture Mode 1

In mode 1 capture of Timer 2 is caused by any write into the low-ordered byte of the dedicated capture register. The value written to capture register is irrelevant for this function. The Timer 2 contents will be latched into appropriate capture register CRCH/CRCL. In this mode no interrupt request will be generated.

## 17.5 PWM Function

The 16-bits CRCH/CRCL registers can be configured for 8-bit PWM mode.

In PWM mode, TL2 will be used as PWM period register, CRCL will be used as PWM duty register, and TH2 will be used as PWM counter.

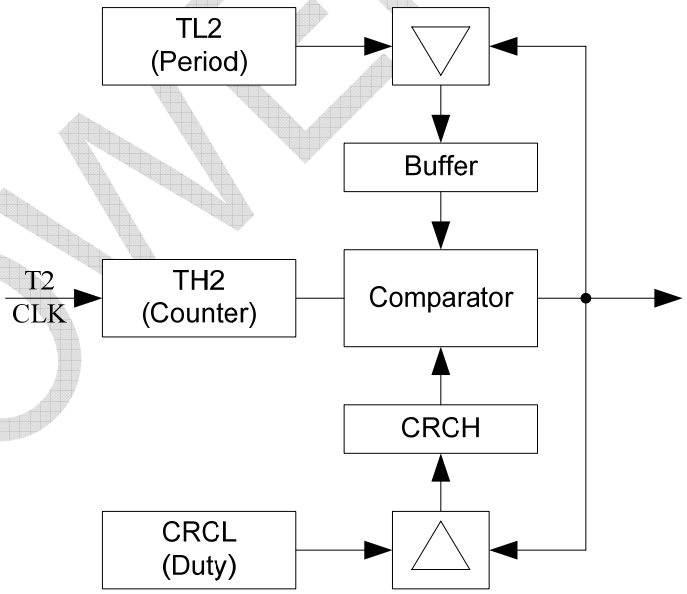



Figure 17-5 Timer 2 in PWM Mode

## 17.6 Register Definition

### 17.6.1 Timer 2 Control Register – T2CON

Table 17-1 T2CON Register (C8h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| t2con.7 | t2en | **Timer 2 enable**<br>0 – Timer/Compare/Capture/PWM mode disabled<br>1 – Timer/Compare/Capture/PWM mode enabled | R/W | 0 |
| t2con.6 | aes1 | **Active edge selection** | R/W | 0 |
| t2con.5 | aes0 | For Timer 2 Compare/Capture/PWM mode | R/W | 0 |
| t2con.4 | t2r1 | **Timer 2 reload mode selection** | R/W | 0 |
| t2con.3 | t2r0 | 0x – reload disabled<br>10 – Mode 0<br>11 – Mode 1 | R/W | 0 |
| t2con.2 | t2cm | **Timer 2 compare mode selection**<br>0 – Mode 0<br>1 – Mode 1 | R/W | 0 |
| t2con.1 | - | - | R | 0 |
| t2con.0 | - | - | R | 0 |

Table 17-2 T2 Active Edge Selection

| AES1 | AES0 | Compare Mode | Capture Mode | PWM Mode |
|---|---|---|---|---|
| 0 | 0 | - | Falling Edge | CCP overflow output low<br>Timer 2 Overflow output high |
| 0 | 1 | - | Rising Edge | CCP overflow output high<br>Timer 2 Overflow output low |
| 1 | 0 | - | Toggle Edge | - |
| 1 | 1 | - | Analog Comparator | - |

### 17.6.2 Timer 2 Mode Register – T2MOD

Table 17-3 T2MOD Register (C9h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| t2mod.7~3 | - | - | R | 0 |
| t2mod.2 | - | **Timer 2 Compare/Capture mode selection** | R/W | 0 |
| t2mod.1 | - | | R/W | 0 |
| t2mod.0 | - | | R/W | 0 |

Table 17-4 Timer 2 Compare/Capture Mode Selection

| T2MOD.2 | T2MOD.1 | T2MOD.0 | Timer 2 Mode |
|---|---|---|---|
| 0 | 0 | 0 | Timer mode |
| 0 | 0 | 1 | Capture mode 0 |
| 0 | 1 | 0 | Capture mode 1 |
| 0 | 1 | 1 | Compare mode, disable output |
| 1 | 0 | 0 | Compare mode, enable output to T2CPO |
| 1 | 0 | 1 | PWM output to T2CPO |
| 1 | 1 | 0 | Timer mode |
| 1 | 1 | 1 | Timer mode |

### 17.6.3 Timer 2 Compare/Reload/Capture/PWM Registers – CRCH/CRCL

Table 17-5 CRCH Register (CBh)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| crch.7~0 | - | Timer 2 Compare/Reload/Capture/PWM Register | R/W | 00h |

Table 17-6 CRCL Register (CAh)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| crcl.7~0 | - | Timer 2 Compare/Reload/Capture/PWM Register | R/W | 00h |

### 17.6.4 Timer 2 Registers – TH2/TL2

Table 17-7 TH2 Register (CDh)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| th2.7~0 | - | Timer 2 higher byte | R/W | 00h |

Table 17-8 TL2 Register (CCh)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| tl2.7~0 | - | Timer 2 lower byte | R/W | 00h |

# 18 PWM0/1/2/3

## 18.1 Overview

The device includes four 12-bit pulse width modulation (PWM) output. The SFR PWMCON0/1 is PWM mode control register, the SFRs PWMxDH/PWMxDL are PWM duty cycle control registers, and the SFRs PWMxPH/PWMxPL are PWM period cycle control registers.

In PWM mode, when set the duty cycle control registers or set the period cycle control registers, the high byte register should be set firstly, and then set the low byte register. Once the low byte register is set, the data will be loaded and PWM will start working.

PWM can work in IDLE mode, and be disabled in STOP mode.

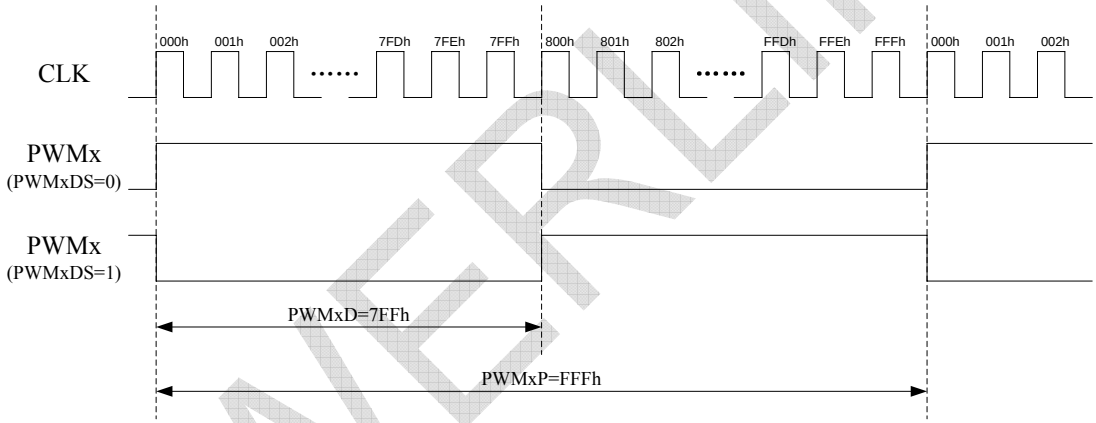



Figure 18-1 12-bit PWM

## 18.2 Register Definition

### 18.2.1 PWM Control Register – PWMCON0

Table 18-1 PWMCON0 Register (FAh)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| pwmcon0.7 | pwm1en | **PWM1 enable** | R/W | 0 |
| pwmcon0.6 | pwm0en | **PWM0 enable** | R/W | 0 |
| pwmcon0.5 | pwm1ds | **PWM1 duty cycle mode selection**<br>0 – normal duty cycle mode | R/W | 0 |

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| | | 1 – negative duty cycle mode | | |
| pwmcon0.4 | pwm0ds | **PWM0 duty cycle mode selection**<br>0 – normal duty cycle mode<br>1 – negative duty cycle mode | R/W | 0 |
| pwmcon0.3 | pwm1ps1 | **PWM1 clock prescaler selection** | R/W | 0 |
| pwmcon0.2 | pwm1ps0 | | R/W | 0 |
| pwmcon0.1 | pwm0ps1 | **PWM0 clock prescaler selection** | R/W | 0 |
| pwmcon0.0 | pwm0ps0 | | R/W | 0 |

Table 18-2 PWMCON1 Register (F9h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| pwmcon1.7 | pwm3en | **PWM3 enable** | R/W | 0 |
| pwmcon1.6 | pwm2en | **PWM2 enable** | R/W | 0 |
| pwmcon1.5 | pwm3ds | **PWM3 duty cycle mode selection**<br>0 – normal duty cycle mode<br>1 – negative duty cycle mode | R/W | 0 |
| pwmcon1.4 | pwm2ds | **PWM2 duty cycle mode selection**<br>0 – normal duty cycle mode<br>1 – negative duty cycle mode | R/W | 0 |
| pwmcon1.3 | pwm3ps1 | **PWM3 clock prescaler selection** | R/W | 0 |
| pwmcon1.2 | pwm3ps0 | | R/W | 0 |
| pwmcon1.1 | pwm2ps1 | **PWM2 clock prescaler selection** | R/W | 0 |
| pwmcon1.0 | pwm2ps0 | | R/W | 0 |

Table 18-3 PWM Prescaler Selection

| PWMxPS1 | PWMxPS0 | PWMx Clock Prescaler Selection |
|---|---|---|
| 0 | 0 | clk_sys/1 |
| 0 | 1 | clk_sys/2 |
| 1 | 0 | clk_sys/4 |
| 1 | 1 | clk_sys/8 |

### 18.2.2 PWM0 Period Registers – PWM0PH/PWM0PL

Table 18-4 PWM0PH Register (FCh)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| pwm0ph.7~4 | - | - | R | 4'b0 |
| pwm0ph.3~0 | - | PWM0 period cycle register higher byte | R/W | 4'b0 |

Table 18-5 PWM0PL Register (FBh)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| pwm0pl.7~0 | - | PWM0 period cycle register lower byte | R/W | 00h |

### 18.2.3 PWM0 Duty Registers – PWM0DH/PWM0DL

Table 18-6 PWM0DH Register (FEh)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| pwm0dh.7~4 | - | - | R | 4'b0 |
| pwm0dh.3~0 | - | PWM0 duty cycle register higher byte | R/W | 4'b0 |

Table 18-7 PWM0DL Register (FDh)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| pwm0dl.7~0 | - | PWM0 duty cycle register lower byte | R/W | 00h |

### 18.2.4 PWM1 Period Registers – PWM1PH/PWM1PL

Table 18-8 PWM1PH Register (F4h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| pwm1ph.7~4 | - | - | R | 4'b0 |
| pwm1ph.3~0 | - | PWM1 period cycle register higher byte | R/W | 4'b0 |

Table 18-9 PWM1PL Register (F3h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| pwm1pl.7~0 | - | PWM1 period cycle register lower byte | R/W | 00h |

### 18.2.5 PWM1 Duty Registers – PWM1DH/PWM1DL

Table 18-10 PWM1DH Register (F6h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| pwm1dh.7~4 | - | - | R | 4'b0 |

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| pwm1dh.3~0 | - | PWM1 duty cycle register higher byte | R/W | 4'b0 |

Table 18-11 PWM1DL Register (F5h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| pwm1dl.7~0 | - | PWM1 duty cycle register lower byte | R/W | 00h |

### 18.2.6 PWM2 Period Registers – PWM2PH/PWM2PL

Table 18-12 PWM2PH Register (E9h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| pwm2ph.7~4 | - | - | R | 4'b0 |
| pwm2ph.3~0 | - | PWM2 period cycle register higher byte | R/W | 4'b0 |

Table 18-13 PWM2PL Register (E8h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| pwm2pl.7~0 | - | PWM2 period cycle register lower byte | R/W | 00h |

### 18.2.7 PWM2 Duty Registers – PWM2DH/PWM2DL

Table 18-14 PWM2DH Register (EBh)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| pwm2dh.7~4 | - | - | R | 4'b0 |
| pwm2dh.3~0 | - | PWM2 duty cycle register higher byte | R/W | 4'b0 |

Table 18-15 PWM2DL Register (EAh)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| pwm2dl.7~0 | - | PWM2 duty cycle register lower byte | R/W | 00h |

### 18.2.8 PWM3 Period Registers – PWM3PH/PWM3PL

Table 18-16 PWM3PH Register EDh)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| pwm3ph.7~4 | - | - | R | 4'b0 |
| pwm3ph.3~0 | - | PWM3 period cycle register higher byte | R/W | 4'b0 |

Table 18-17 PWM3PL Register (ECh)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| pwm3pl.7~0 | - | PWM3 period cycle register lower byte | R/W | 00h |

### 18.2.9 PWM3 Duty Registers – PWM3DH/PWM3DL

Table 18-18 PWM3DH Register (EFh)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| pwm3dh.7~4 | - | - | R | 4'b0 |
| pwm3dh.3~0 | - | PWM3 duty cycle register higher byte | R/W | 4'b0 |

Table 18-19 PWM3DL Register (EEh)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| pwm3dl.7~0 | - | PWM3 duty cycle register lower byte | R/W | 00h |

# 19 Watchdog Timer

The watchdog timer is an 18-bit counter The device has a watchdog timer (WDT). It is used to provide the system supervision in case of software or hardware upset. If the software is not able to refresh the WDT before it is time-out, an internal reset is generated. Software can access (read or write) the RSTCON.WDRF to reset the WDT counter and re-open a WDT window.

## 19.1 Register Definition

### 19.1.1 WDT Control Register – WDTCON

Table 19-1 WDTCON Register (86h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| wdtcon.7 | wdten | **WDT enable**<br>0 – disabled<br>1 – enabled | R/TW | 0 |
| wdtcon.6 | - | - | R | 0 |
| wdtcon.5 | wdtien | WDT interrupt enable | R/W | 0 |
| wdtcon.4 | wdtif | WDT interrupt flag, when enabled, the flag indicating the WDT timeout reset will come after about 256 periods of clk_32k | R/W | 0 |
| wdtcon.3 | wdtps3 | **WDT prescaler selection** | R/TW | 0 |
| wdtcon.2 | wdtps2 | | R/TW | 0 |
| wdtcon.1 | wdtps1 | | R/TW | 0 |
| wdtcon.0 | wdtps0 | | R/TW | 0 |

Table 19-2 WDT Prescaler Selection

| WDTPS3 | WDTPS2 | WDTPS1 | WDTPS0 | Time-out Cycles | Time-out (Typ.) |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 512 | 16ms |
| 0 | 0 | 0 | 1 | 1024 | 31ms |
| 0 | 0 | 1 | 0 | 2048 | 63ms |
| 0 | 0 | 1 | 1 | 4096 | 125ms |
| 0 | 1 | 0 | 0 | 8192 | 250ms |
| 0 | 1 | 0 | 1 | 16384 | 500ms |
| 0 | 1 | 1 | 0 | 32768 | 1.0s |
| 0 | 1 | 1 | 1 | 65536 | 2.0s |
| 1 | 0 | 0 | 0 | 131072 | 4.0s |

| WDTPS3 | WDTPS2 | WDTPS1 | WDTPS0 | Time-out Cycles | Time-out (Typ.) |
|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 262144 | 8.0s |
| 1 | 0 | 1 | 0 | Reserved | Reserved |
| 1 | 0 | 1 | 1 | Reserved | Reserved |
| 1 | 1 | 0 | 0 | Reserved | Reserved |
| 1 | 1 | 0 | 1 | Reserved | Reserved |
| 1 | 1 | 1 | 0 | Reserved | Reserved |
| 1 | 1 | 1 | 1 | Reserved | Reserved |

# 20 UART

The Serial provides a flexible full-duplex synchronous/asynchronous receiver/transmitter. It can operate in four modes (one synchronous and three asynchronous). The Serial is buffered at the receive side, i.e. it can revive new data while the previously received is not damaged in the receive register until the completion of the 2nd transfer. The Serial is fully compatible with the standard 8051 serial channel.

The transmit register and the receive buffer are both addressed as SBUF Special Function Register. However any write to SBUF will be to the transmit register, while a read from SBUF will be from the receiver buffer register.

## 20.1 Mode 0

In mode 0 the Serial Port 0 operates as synchronous transmitter/receiver. The TXD outputs the shift clock. The RXD outputs data and inputs data. 8 bits are transmitted with LSB first. The baud rate is fixed at 1/12 of the main clock frequency. Reception is started by setting the SCON.REN, and clearing the SCON.RI. Transmission is started by writing data to SBUF register.

## 20.2 Mode 1

In mode 1 the Serial Port 0 operates as asynchronous transmitter/receiver with 8 data bits and programmable baud rate. Additionally the baud rate can be doubled with the use of the PCON.SMOD.

Transmission is started by writing to the SBUF register. The TXD pin outputs data. The first bit transmitted is a start bit (always 0), then 8 bits of data proceed, after which a stop bit (always 1) is transmitted.

The RXD pin inputs data. When reception starts, the Serial Port synchronizes with the falling edge detected at pin RXD. Input data are available after completion of the reception in the SBUF register, and the value of stop bit is available as the SCON.RB8. During the reception, the SBUF and SCON.RB8 remain unchanged until the completion.

## 20.3 Mode 2

In mode 2 the Serial Port operates as asynchronous transmitter/receiver with 9 data bits, and baud rate fixed to 1/32 or 1/64 of system clock, depending on the setting of PCON.SMOD. Transmission is started by writing to the SBUF register. The TXD pin outputs data. The first bit transmitted is a start bit (always 0), then 9 bits of data proceed where the 9th is taken from bit SCON.TB8 after which a stop bit (always 1) is transmitted.

The RXD pin inputs data. When reception starts, the Serial Port synchronizes with the falling edge detected at pin RXD. Input data are available after completion of the reception in the SBUF register, and the 9th bit is available as the SCON.RB8. During the reception, the SBUF and SCON.RB8 remain

unchanged until the completion.

## 20.4 Mode 3

The only difference between Mode 2 and Mode 3 is that in Mode 3 Timer 1 can be used to specify the baud rate.

In mode 3 the Serial Port operates as asynchronous transmitter/receiver with 9 data bits and programmable baud rate. Additionally the baud rate can be doubled with the use of the PCON.SMOD

Transmission is started by writing to the SBUF register. The TXD pin outputs data. The first bit transmitted is a start bit (always 0), then 9 bits of data proceed where the 9th is taken from bit SCON.TB8, after which a stop bit (always 1) is transmitted.

The RXD pin inputs data. When reception starts, the Serial Port synchronizes with the falling edge detected at pin RXD. Input data are available after completion of the reception in the SBUF register, and the 9th bit is available as the SCON.RB8. During the reception, the SBUF and SCON.RB8 remain unchanged until the completion.

## 20.5 Baud Rate

The baud rate for Serial Port working in mode 1 or mode 3:

$$for\ bd\ (adcon.7) = 0:$$

$$baud\ rate = \frac{2^{SMOD} * Fclk}{32} * (Timer1\ overflow\ rate)$$

## 20.6 The Serial Port Multiprocessor Communication

The feature of receiving 9 bits in Modes 2 and 3 of Serial Interface 0 can be used for multiprocessor communication.

When the SCON.SM2 is set, the receive interrupt is generated only when the 9th received bit (SCON.RB8) is 1. Otherwise, no interrupt is generated upon reception.

To utilize this feature to multiprocessor communication, the slave processors have their SCON.SM2 bit set to 1. The master processor transmits the slave's address, with the 9th bit set to 1, causing reception interrupt in all of the slaves. The slave processors' software compares the received byte with their network address. If there is a match, the addressed slave clears its SCON.SM2 and the rest of the message is transmitted from the master with the 9th bit set to 0. The other slaves keep their SCON.SM2 set to 1 so that they ignore the rest of the message sent by the master.

## 20.7 Register Definition

### 20.7.1 Serial Port Control Register – SCON

Table 20-1 SCON Register (98h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| s0con.7 | sm0 | **Serial Port 0 mode select** | R/W | 0 |
| s0con.6 | sm1 | | R/W | 0 |
| s0con.5 | sm2 | **Multiprocessor communication enable** | R/W | 0 |
| s0con.4 | ren | **Serial reception enable**<br>If set HIGH serial reception at Serial Port is enabled.<br>Otherwise serial reception at Serial Port is disabled. | R/W | 0 |
| s0con.3 | tb8 | **Transmitter bit 8**<br>This bit is used while transmitting data through Serial Port in Modes 2 and 3. The state of this bit corresponds with the state of the 9th transmitted bit (e.g. parity check or multiprocessor communication). It is controlled by software. | R/W | 0 |
| s0con.2 | rb8 | **Received bit 8**<br>This bit is used while receiving data through Serial Port 0 in Modes 2 and 3. It reflects the state of the 9th received bit.<br>In Mode 1, if multiprocessor communication is enabled (sm2 = 0), this bit is the stop bit that was received.<br>In Mode 0 this bit is not used. | R/W | 0 |
| s0con.1 | ti | **Transmit interrupt flag**<br>It indicates completion of a serial transmission at Serial Port.<br>It is set by hardware at the end of bit 8 in mode 0 or at the beginning of a stop bit in other modes. It must be cleared by software. | R/W | 0 |
| s0con.0 | ri | **Receive interrupt flag**<br>It is set by hardware after completion of a serial reception at Serial Port 0.<br>It is set by hardware at the end of bit 8 in mode 0 or in the middle of a stop bit in other modes.<br>It must be cleared by software. | R/W | 0 |

Table 20-2 Serial Port modes and Baud Rates

| sm0 | sm1 | Mode | Description | Baud Rate | |
|---|---|---|---|---|---|
| 0 | 0 | Mode 0 | Shift register | Fsys/12 | |
| 0 | 1 | Mode 1 | 8-bit UART | Timer 1 overflow | |
| 1 | 0 | Mode 2 | 9-bit UART | Depends on PCON.SMOD | |
| | | | | smod | Baud Rate |
| | | | | 0 | Fsys/64 |
| | | | | 1 | Fsys/32 |
| 1 | 1 | Mode 3 | 9-bit UART | Timer 1 overflow | |

### 20.7.2 Serial Port Data Buffer – SBUF

Table 20-3 SBUF Register (99h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| sbuf.7~0 | - | Serial port data buffer | R/W | 00h |

# 21 SPI

The Serial Peripheral Interface (SPI) allows high-speed, full-duplex synchronous data transfer between the device and peripheral devices or between multiple microcontroller devices, including multiple masters and slaves on a single bus.

The SPI interface is often used to communicate with external peripheral devices such as RF transceiver, sensors, Flash or EEPROM memory devices etc. Originally developed by Motorola, the four line SPI interface is a synchronous serial data interface that has a relatively simple communication protocol simplifying the programming requirements when communicating with external hardware devices.

The communication is full duplex and operates as a slave/master type, where the device can be either master or slave. Although the SPI interface specification can control multiple slave devices from a single master, but this device provided only one SCSB pin. If the master needs to control multiple slave devices from a single master, the master can use I/O pin to select the slave devices.

## 21.1 SPI Interface

The SPI interface is a full duplex synchronous serial data link. It is a four line interface with pin names MISO, MOSI, SCK and SCSB. Pins MISO and MOSI are the Serial Data Input and Serial Data Output lines, SCK is the Serial Clock line and SCSB is the Slave Select line. As the SPI interface pins are pin-shared with normal I/O pins and with the I2C function pins. Communication between devices connected to the SPI interface is carried out in a slave/master mode with all data transfer initiations being implemented by the master. The Master also controls the clock signal. As the device only contains a single SCSB pin only one slave device can be utilized. The SCSB pin is controlled by software.

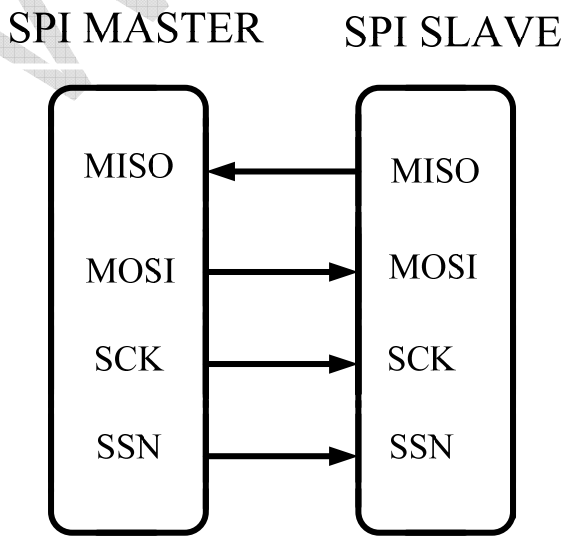



Figure 21-1 SPI Master/Slave Connection

## 21.2 SPI Communication

After the SPI interface is enabled by setting the SPCON.SPEN bit high, then in the Master Mode, when data is written to the SPDAT register, transmission/reception will begin simultaneously. When the data transfer is complete, the SPIF flag will be set automatically, but must be cleared using the application program.

In the Slave Mode, when the clock signal from the master has been received, any data in the SPDAT register will be transmitted and any data on the MISO pin will be shifted into the SPDAT register. The master should output an SCSB signal to enable the slave device before a clock signal is provided. The slave data to be transferred should be well prepared at the appropriate moment relative to the SCSB signal.

SPI communication timing waveform is shown as below:

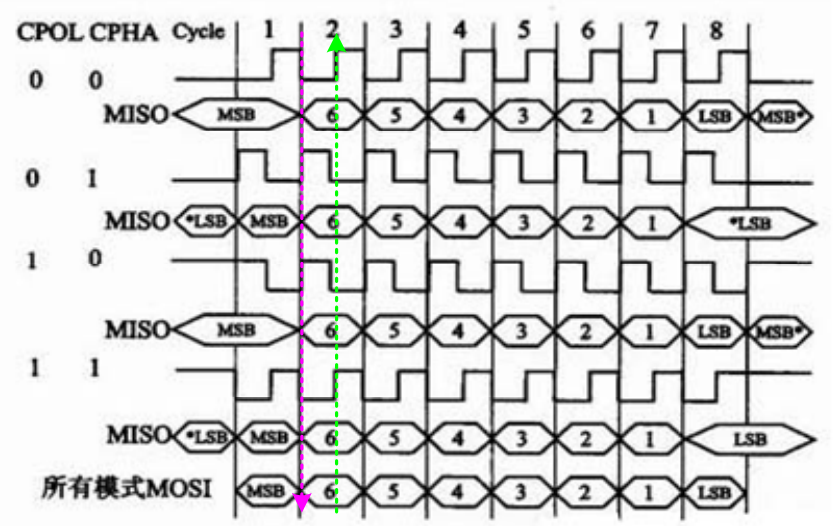



Figure 21-2 SPI bus protocol

## 21.3 Register Definition

### 21.3.1 SPI Serial Peripheral Status Register – SPSTA

Table 21-1 SPSTA Register (E1h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| spsta.7 | spif | **Serial Peripheral Data Transfer Flag** | R | 0 |

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| | | Set by hardware upon data transfer completion.<br>Cleared by reading the "spsta" register with the "spif" bit set, and then reading the "spdat" register. | | |
| spsta.6 | wcol | **Write Collision Flag**<br>Set by hardware upon write collision to "spdat".<br>Cleared by an access to "spsta" register and an access to "spdat" register. | R | 0 |
| spsta.5 | sserr | **Synchronous Serial Slave Error Flag**<br>Set by hardware when "ssn" input is deasserted before the end of receive sequence. Cleared by disabling the SPI module (clearing "spen" bit in "spcon" register). | R | 0 |
| spsta.4 | modf | **Mode Fault Flag**<br>Set by hardware when the SCSB pin level is in conflict with actual mode of the SPI controller (configured as master while externally selected as slave).<br>Cleared by hardware when the SCSB pin is at appropriate level. Can be also cleared by software by reading the "spsta" register with "modf" bit set. | R | 0 |
| spsta.3~0 | - | - | R | 4'b0 |

### 21.3.2 SPI Serial Peripheral Control Register – SPCON

Table 21-2 SPCON Register (E2h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| spcon.7 | spr2 | **Serial Peripheral Rate 2**<br>Together with "spr1" and "spr0" defines the clock rate in master mode. | R/W | 0 |
| spcon.6 | spen | **Serial Peripheral Enable**<br>When cleared disables the SPI interface. When set enables the SPI interface. | R/W | 0 |
| spcon.5 | ssdis | **SS Disable**<br>When cleared enables the SCSB input in both Master and Slave modes. When set disables the SCSB input in both Master and Slave modes. In Slave mode, this bit has no effect if "cpha"=1. When "ssdis" is set, no "modf" interrupt request will be generated. | R/W | 0 |
| spcon.4 | mstr | **Serial Peripheral Master**<br>When cleared configures the SPI as a Slave. When set configures the SPI as a Master. | R/W | 1 |

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| spcon.3 | cpol | **Clock Polarity**<br>When cleared, the SCK is set to 0 in idle state. When set, the SCK is set to 1 in idle state. | R/W | 0 |
| spcon.2 | cpha | **Clock Phase**<br>When cleared, data is sampled when the "scki"/"scko" leaves the idle state (see "cpol"). When set, data is sampled when the "scki"/"scko" returns to idle state (see "cpol"). | R/W | 1 |
| spcon.1 | spr1 | **Serial Peripheral Rate**<br>Together with "spr2" specify the serial clock rate in Master mode. | R/W | 0 |
| spcon.0 | spr0 | | R/W | 0 |

Table 21-3 Serial Peripheral Rate

| spr2 | spr1 | spr0 | Serial Peripheral Rate |
|---|---|---|---|
| 0 | 0 | 0 | Fclk/2 |
| 0 | 0 | 1 | Fclk/4 |
| 0 | 1 | 0 | Fclk/8 |
| 0 | 1 | 1 | Fclk/16 |
| 1 | 0 | 0 | Fclk/32 |
| 1 | 0 | 1 | Fclk/64 |
| 1 | 1 | 0 | Fclk/128 |
| 1 | 1 | 1 | the master clock is not generated (when "cpol" = '1' SCK is high level, otherwise is low level |

### 21.3.3 SPI Serial Peripheral Data Register – SPDAT

Table 21-4 SPDAT Register (E3h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| spdat.7~0 | - | SPI serial peripheral data | R/W | 00h |

# 22 I2C

The I2C interface is used to communicate with external peripheral devices such as RF transceiver, sensors, EEPROM memory etc. Originally developed by Philips, it is a two line low speed serial interface for synchronous serial data transfer. The advantage of only two lines for communication, relatively simple communication protocol and the ability to accommodate multiple devices on the same bus has made it an extremely popular interface type for many applications.

## 22.1 I2C Interface

The Two-Wire Interface (I2C) is a bi-directional 2-wire serial communication standard. It is designed primarily for simple but efficient integrated circuit (IC) control. The system is comprised of two lines, SCL (Serial Clock) and SDA (Serial Data) that carry information between the ICs connected to them. The only external hardware needed to implement the bus is a single pull-up resistor for each of the I2C bus lines. All devices connected to the bus have individual addresses, and mechanisms for resolving bus contention are inherent in the I2C protocol. Any of the devices connected to the bus can be master or slave.

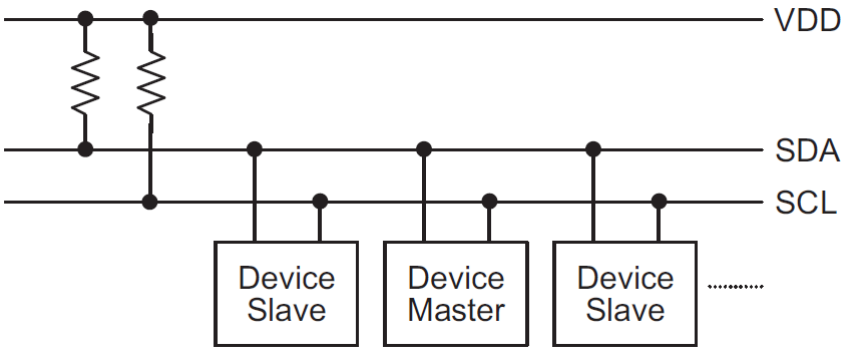



Figure 22-1 I2C Master Slave Bus Connection

## 22.2 I2C Communication

The I2C serial interface is a two line interface, a serial data line, SDA, and serial clock line, SCL. As many devices may be connected together on the same bus, their outputs are both open drain types. For this reason it is necessary that external pull-high resistors are connected to these outputs. Note that no chip select line exists, as each device on the I2C bus is identified by a unique address which will be transmitted and received on the I2C bus.

When two devices communicate with each other on the bidirectional I2C bus, one is known as the master device and one as the slave device. Both master and slave can transmit and receive data, however, it is the master device that has overall control of the bus. For these devices, which only operates in slave mode,

there are two methods of transferring data on the I2C bus, the slave transmit mode and the slave receive mode.

Communication on the I2C bus requires four separate steps, a START signal, a slave device address transmission, a data transmission and finally a STOP signal. When a START signal is placed on the I2C bus, all devices on the bus will receive this signal and be notified of the imminent arrival of data on the bus. The first seven bits of the data will be the slave address with the first bit being the MSB.

The START signal can only be generated by the master device connected to the I2C bus and not by the slave device. This START signal will be detected by all devices connected to the I2C bus. A START condition occurs when a high to low transition on the SDA line takes place when the SCL line remains high.

I2C communication timing waveform is shown as below:

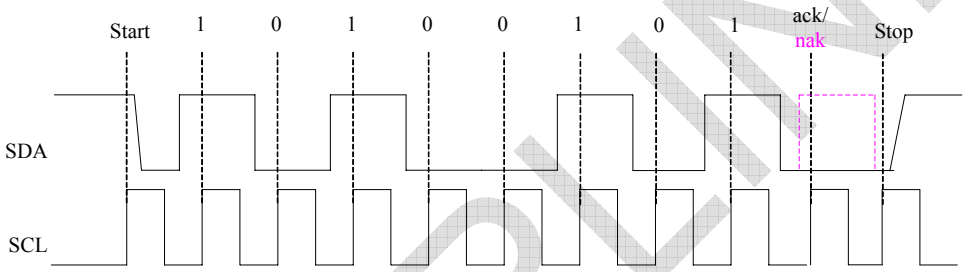



Figure 22-2 I2C bus protocol

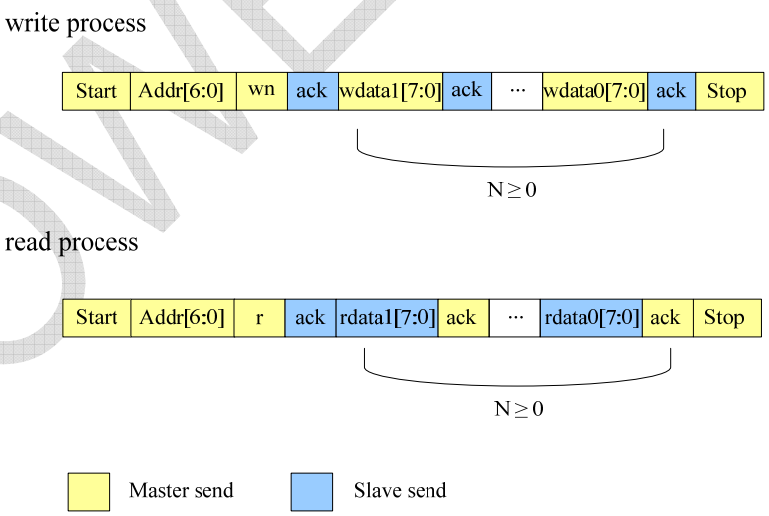



Figure 22-3 I2C communication timing waveform

## 22.3 Register Definition

### 22.3.1 I2C Status Register – I2CSTA

Table 22-1 I2CSTA Register (DDh)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| i2csta.7~3 | - | **I2C Status Code** | R | F8h |
| i2csta.2~0 | - | - | R | |

### 22.3.2 I2C Control Register – I2CCON

Table 22-2 I2CCON Register (DCh)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| i2ccon.7 | cr2 | **Clock rate bit 2** | R/W | 0 |
| i2ccon.6 | ens1 | **I2C enable bit** | R/W | 0 |
| i2ccon.5 | sta | **START Flag**<br>When sta='1', the I2C component checks the I2C bus status and if the bus is free a START condition is generated. | R/W | 0 |
| i2ccon.4 | sto | **STOP Flag**<br>When sto='1' and I2C interface is in master mode, a STOP condition is transmitted to the I2C bus. | R/W | 0 |
| i2ccon.3 | si | **Serial Interrupt Flag**<br>The "si "is set by hardware when one of 25 out of 26 possible I2C states is entered. The only state that does not set the "si" is state F8h, which indicates that no relevant state information is available. The "si" flag must be cleared by software. In order to clear the "si" bit, '0' must be written to this bit. Writing a '1' to si bit does not change value of the "si". | R/W | 0 |
| i2ccon.2 | aa | **Assert Acknowledge Flag**<br>When aa='1', an "acknowledge" will be returned when:<br>- the "own slave address" has been received<br>- the general call address has been received while gc bit in i2caddr register was set<br>- a data byte has been received while I2C was in master | R/W | 0 |

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| | | receiver mode<br>- a data byte has been received while I2C was in slave receiver mode<br>When aa='0', an "not acknowledge" will be returned when:<br>- a data byte has been received while I2C was in master receiver mode<br>- a data byte has been received while I2C was in slave receiver mode | | |
| i2ccon.1 | cr1 | **Clock rate bit 1** | R/W | 0 |
| i2ccon.0 | cr0 | **Clock rate bit 0** | R/W | 0 |

This programmable clock pulse generator provides the "sclo" clock pulses when the I2C is in the master mode. The clock generator is suppressed when the I2C is in the slave mode.

The function of the clock generator is controlled by bits "cr0", "cr1" and "cr2" of "i2ccon" register. The table below shows the possible rates of "clko" in the master mode.

The "bclk" input referenced in the table is connected to the Timer 1 overflow output. That means the baud rate of the I2C can be controlled by the Timer 1.

Table 22-3 I2CCON Register (DCh)

| cr2 | cr1 | cr0 | Bit Frequency | | | CLK Divided |
|---|---|---|---|---|---|---|
| | | | 4MHz | 8MHz | 12MHz | |
| 0 | 0 | 0 | 15.6 | 31 | 47 | 256 |
| 0 | 0 | 1 | 17.8 | 35.8 | 54 | 224 |
| 0 | 1 | 0 | 21 | 42 | 63 | 192 |
| 0 | 1 | 1 | 15 | 50 | 75 | 160 |
| 1 | 0 | 0 | 4.2 | 8.4 | 12.5 | 960 |
| 1 | 0 | 1 | 33.3 | 66.6 | 100 | 120 |
| 1 | 1 | 0 | 66.6 | 133.3 | 200 | 60 |
| 1 | 1 | 1 | "bclk"(T1 overflow) input divided by 8 | | | |

### 22.3.3 I2C Address Register – I2CADR

Table 22-4 I2CADR Register (DBh)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| i2cadr.7~1 | adr | **Own I2C slave address** (7 bit) | R/W | 7'b0 |
| i2cadr.0 | gc | **General Call Address Acknowledge**<br>If this bit is set, the general call address is recognized; | R/W | 1'b0 |

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| | | otherwise it is ignored. | | |

### 22.3.4 I2C Data Register – I2CDAT

Table 22-5 I2CDAT Register (DAh)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| i2cdat.7~0 | - | I2C data | R/W | 00h |

# 23 A/D & TouchKey

## 23.1 Overview

The chip supports the functions of ADC, touch keys, but just could be configured by CPU as one function separately at one time, since touch key function needs to use ADC module for calculating.

ADC function could only work in normal mode, and the operation clock could be selected as 1/2/4/8 pre-scale of system clock. However, Touch Key could work in both normal mode and low power mode (idle/stop/sleep mode).

In normal mode, both system clock and 32khz internal clock are always working. While only 32khz clock is always working in low power mode, though system clock could be used when needed, for reduce the power consumption. The touch action could wakeup the system from low power mode (idle/stop/sleep), and when this happens, the system will then enter into normal work mode.

The block diagram of Touch key function is shown as Figure 23-1.

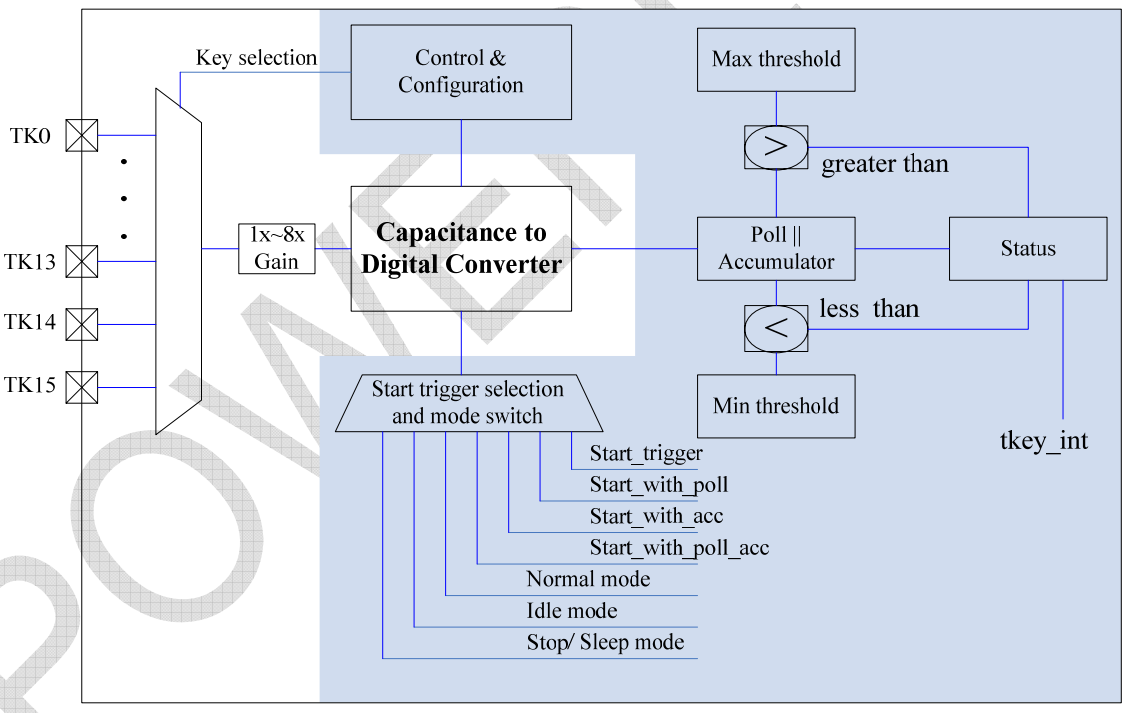



Figure 23-1 Touch Key Function Diagram

The block diagram of ADC function is shown as Figure 23-2.

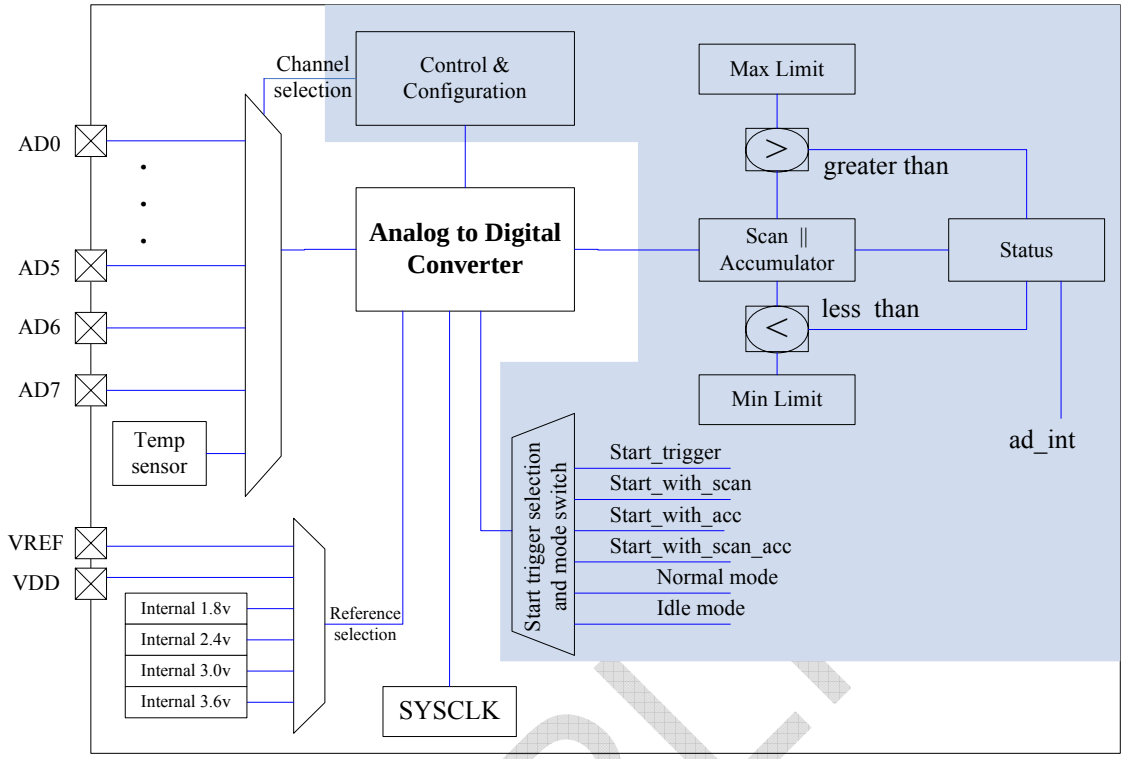



Figure 23-2 ADC Function Diagram

### 23.1.1 Touch Key Operation

When a finger touches or is in proximity to a touch pad, the capacitance of the pad will increase. By using this capacitance variation to change slightly the charge time of the internal sense comparator, touch actions can be sensed by measuring these charge time changes.

The device contains 16 touch key inputs which are shared with logical I/O pins, with the desired function selected using register bits. The Touch Key module also has its own interrupt vectors and set of interrupts flags.

### 23.1.2 Touch Key & ADC Interrupt

The interrupt could be involved by Touch key or ADC, depending on the function selected in SFR register ‘tkcon0’.

When ADC function is selected, the interrupt could be triggered by ADC data is ready to read, or be triggered by the end of all the session done.

The session done for ADC is defined as the end of accumulation if accumulation is enabled, or the end of channel scan if scan is enabled, or end of both of accumulation and channel scan if they are

enabled both.

When Touch key is selected, the interrupt source could be configured as touch action happened or touch key data is ready to read in normal/idle work mode, however, the interrupt could only be triggered by touch action in stop/sleep work mode;

### 23.1.3 Touch Key Work Mode

The modes in which the design supports to configure to work lists at the following table:

Table 23-1 Touch Key Work Mode

| Mode | Key | Polling | Accumulation | Interrupt Source | Work Mode States |
|---|---|---|---|---|---|
| Normal /IDLE | single | X | N | Data ready to read or touch action | trig_sel:<br>0 – Charging with single key, generating interrupt after completion of each data conversion;<br>1 – Continue to work，generating interrupt when the single key is judged to be a touch action(compare with the threshold of single key); |
| Normal /IDLE | single | X | Y | Data ready to read or touch action | trig_sel:<br>0 – Multiple charging with single key, accumulating the converted data, generating interrupt when the number of times is reached;<br>1 – Continue to work, generating interrupt when the single key is judged to be a touch action(compare with the accumulation threshold ); |
| Normal /IDLE | Comb * | N | N | Data ready to read or touch action | trig_sel:<br>0 – Charging with combined keys,generating interrupt after completion of each data conversion;<br>1 – Continue to work，generating interrupt when the combined keys are judged to be a touch action(compare with the threshold of combined keys); |
| Normal /IDLE | Comb | N | Y | Data ready to read or touch action | trig_sel:<br>0 – Multiple charging with combined keys, accumulating the converted data, generating interrupt when the number of times is reached;<br>1 – Continue to work，generating interrupt when the combined keys are judged to be a touch action after multiple accumulation (compare with the accumulation threshold of combined keys); |
| Normal | Comb | Y | N | Data ready | trig_sel: |

| Mode | Key | Polling | Accumulation | Interrupt Source | Work Mode States |
|---|---|---|---|---|---|
| /IDLE | | | | to read or touch action | 0 – Polling charge with single key，generating interrupt after completion of each data conversion;<br>1 – Continue to work，generating interrupt when the single key is polling judged to be a touch action(compare with the threshold of single key); |
| Normal /IDLE | Comb | Y | Y | Data ready to read or touch action | trig_sel:<br>0 – Multiple charging with single key, accumulating the converted data, generating interrupt when the number of times is reached;<br>1 – Continue to work，generating interrupt when the single key is judged to be a touch action after multiple accumulations (compare with the accumulation threshold); |
| STOP /SLEEP | single | X | N | Only key touch action | Continue to work, generating interrupt when the single key is judged to be a touch action(compare with the threshold of single key); |
| STOP /SLEEP | single | X | Y | Only key touch action | Continue to work, generating interrupt when the single key is judged to be a touch action(compare with the accumulation threshold); |
| STOP /SLEEP | Comb | N | N | Only key touch action | Continue to work，generating interrupt when the combined keys are judged to be a touch action(compare with the threshold of combined keys); |
| STOP /SLEEP | Comb | N | Y | Only key touch action | Continue to work，generating interrupt when the combined keys are judged to be a touch action after multiple accumulation (compare with the accumulation threshold of combined keys); |
| STOP /SLEEP | Comb | Y | N | Only key touch action | Continue to work，generating interrupt when the single key is polling judged to be a touch action(compare with the threshold of single key); |
| STOP /SLEEP | Comb | Y | Y | Only key touch action | Continue to work，generating interrupt when the single key is judged to be a touch action after multiple accumulations (compare with the accumulation threshold); |

Notes: *Combined key depend the selected channel

## 23.2 Register Definition

### 23.2.1 Touch Key Registers Address Map

Table 23-2 Touch Key Registers Address Map

| Name | SFR Address | Bit width | Function states |
|---|---|---|---|
| TKDATL | 0xA1 | 8 | Touch Key data low 8 bits |
| TKDATH | 0xA2 | 8 | Touch Key data high 8 bits |
| TKCHS0 | 0xA3 | 8 | Touch Key channels select register 0 |
| TKCHS1 | 0xA4 | 8 | Touch Key channels select register 1 |
| TKCON0 | 0xA5 | 8 | Touch Key control register 0 |
| TKCON1 | 0xA6 | 8 | Touch Key control register 1 |
| TKCON2 | 0xA7 | 8 | Touch Key control register 2 |
| TKADCF | 0x9D | 8 | Touch Key ADC configure register |
| TKCSCF | 0x9E | 8 | Touch Key capsensor configure register |
| TKCSOF | 0x9F | 8 | Touch Key capsensor offset adjustment |
| TKWKL0 | 0xD2 | 8 | Touch Key wakeup threshold 0 low 8 bits |
| TKWKH0 | 0xD3 | 8 | Touch Key wakeup threshold 0 high 8 bits |
| TKWKL1 | 0xD4 | 8 | Touch Key wakeup threshold 1 low 8 bits |
| TKWKH1 | 0xD5 | 8 | Touch Key wakeup threshold 1 high 8 bits |

### 23.2.2 Touch key Data Register – TKDATL

Table 23-3 TKDATL Register (A1h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| tkdatl.7~0 | - | When 'tken' is set, the register store the touch key low 8bits data;<br>When 'adcen' is set, the register store the ADC low 8bits data; | R/W | 00h |

### 23.2.3 Touch key Data Register – TKDATH

Table 23-4 TKDATH Register (A2h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| tkdath.7~0 | - | When 'tken' is set, the register store the touch key high 8bits | R/W | 00h |

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| | | data;<br>When 'adcen' is set, the register store the ADC high 8bits data; | | |

### 23.2.4 Touch key7-0 select Register – TKCHS0

Table 23-5 TKCHS0 Register (A3h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| tkchs0.7~0 | - | Touch key 7-0 channel select bit | R/W | 00h |

### 23.2.5 Touch key15-8 select Register – TKCHS1

Table 23-6 TKCHS1 Register (A4h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| tkchs1.7~0 | - | Touch key 15-8 channel select bit and ADC channel 7~0 select bit | R/W | 00h |

### 23.2.6 Touch key control Register – TKCON0

Table 23-7 TKCON0 Register (A5h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| tkcon0.7 | tken | If set, Touch key is enabled and 'adcen' could not be set; if clear, Touch key is disabled and 'adcen' can be set | R/W | 1'b0 |
| tkcon0.6 | adcen | If set AD Convert is enabled, otherwise is disabled | R/W | 1'b0 |
| tkcon0.5 | - | It always should be 0. | R/W | 1'b0 |
| tkcon0.4 | wait_cpurd | If set, wait till CPU to read out the data at present, then start the next touch key probe operation. | R/W | 1'b0 |
| tkcon0.3 | freq_sel2 | ADC analog clock frequency prescaler of system clock select options: | R/W | 3'b0 |

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| tkcon0.2 | freq_sel1 | 3’b000: no prescaler<br>3’b001: prescaler by 2<br>3’b010: prescaler by 4<br>3’b011: prescaler by 8<br>3’b100: prescaler by 16<br>3’b101: prescaler by 32<br>3’b110: no prescaler<br>3’b111: no prescaler | | |
| tkcon0.1 | freq_sel0 | | | |
| tkcon0.0 | mode | When ADC function is selected, this bit select the ADC work mode, ***must be cleared***, continuous conversion mode is selected;<br>When Touch Key function is selected, this bit select the Touch Key work mode, if set Polling mode is open, otherwise, Polling mode is closed | R/W | 1‘b0 |

### 23.2.7 Touch key control Register – TKCON1

Table 23-8 TKCON1 Register (A6h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| tkcon1.7 | start | If set, trigger the analog function start working, in the end of the operation, this bit will be auto cleared and wait the next start trigger. | R/W | 1’b0 |
| tkcon1.6 | accum2 | **Accumulative number:**<br>3’h0: single time convert data<br>3’h1: accumulate 2 times convert data<br>3’h2: accumulate 4 times convert data<br>3’h3: accumulate 8 times convert data<br>…<br>3’h7: accumulate 128 times convert data<br>Note: please configure the accumulation time carefully to avoid the overflow of sum, if overflow, the ‘ov’ in ‘TKSTA’ will be set to indicates the bad accumulation. | R/W | 3‘b0 |
| tkcon1.5 | accum1 | | | |
| tkcon1.4 | accum0 | | | |
| tkcon1.3 | ave_dis | **Average operation disable** | R/W | 1‘b0 |
| tkcon1.2 | trig_sel | In the normal/idle work mode, when configure as touch key function, interrupt could be triggered by two sources: data is valid or key touch action:<br>1’b0: data is valid and trigger interrupt<br>1’b1: key touch action trigger interrupt | R/W | 1’b0 |
| tkcon1.1 | func_flg | When read as 0, indicates the design is working as touch key function otherwise the design is working as ADC | R/W | 1‘b0 |

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| | | function. | | |
| tkcon1.0 | ov | Indicates the overflow is occurred in the accumulative operation | R/W | 1'b0 |

### 23.2.8 Touch key control Register – TKCON2

Table 23-9 TKCON2 Register (A7h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| tkcon2.7 | cap_lp | Capsense low power enable | R/W | 1'b0 |
| tkcon2.6 | ses_gap2 | The gap time between two session of accumulation operation, configure as:<br>3'h0: 16 clock period of 'clk_32k'<br>3'h1: 32 clock period of 'clk_32k'<br>3'h2: 64 clock period of 'clk_32k'<br>3'h3: 128 clock period of 'clk_32k'<br>…<br>3'h7: 2048 clock period of 'clk_32k'<br>NOTE: one session of accumulation includes the channel selected in registers 'tkchs1/tkchs0' charge one times at less. In other words, this configure number is the number of session contained in completely one touch action detection. | R/W | 3'b0 |
| tkcon2.5 | ses_gap1 | | | |
| tkcon2.4 | ses_gap0 | | | |
| tkcon2.3 | idle_sta | idle start flag, indicates the process triggered by previous go-idle command is not finished yet, and the next step will be continue after coming go-idle command, this bit will be auto-cleared by hardware when end of process, it also could be cleared by CPU | R/W | 1'b0 |
| tkcon2.2 | wait_cpu | Wait CPU flag, indicates process is handling wait for CPU read, it should be cleared by CPU setting 1'b0 after read out data. | R/W | 1'b0 |
| tkcon2.1 | sta_gap1 | The gap time between two start trigger, configure as:<br>2'b00: 2 clock period of 'tk_clk'<br>2'b01: 4 clock period of 'tk_clk'<br>2'b10: 8 clock period of 'tk_clk'<br>2'b11: 16 clock period of 'tk_clk' | R/W | 2'b0 |
| tkcon2.0 | sta_gap0 | | | |

### 23.2.9 Touch key status Register – TKADCF

Table 23-10 TKADCF Register (9Dh)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| tkadcf.7 | inject | If set, injection operation is enabled, the current normal channel conversion is interrupted and channel 7 is injected into ADC to start conversion until this bit is cleared | R/TW | 1‘b0 |
| tkadcf.6 | adc_pump | 1:support VDD50 higher than 1.8V;<br>0:support VDD50 higher than 2.7V; | R/W | 1‘b0 |
| tkadcf.5 | iref_adj1 | 2’b00:1uA;<br>2’b01:2uA;<br>2’b10:4uA;<br>2’b11:6uA; | R/W | 2‘b0 |
| tkadcf.4 | iref_adj0 | | | |
| tkadcf.3 | cons_sta* | This bit is used for ADC continuously convert mode only, and for other usage is forbidden, And it must be set immediately after the ‘tkcon1.7’(start) is set, otherwise, the ‘tkcon1.7’(start) can only work once, it means ADC converts data one time only, | R/W | 1‘b0 |
| tkadcf.2 | adc_vref2 | **ADC reference select:**<br>3’b00x:VDD pin;<br>3’b01x: VREF pin;<br>3’b100: internal 1.8V;<br>3’b101: internal 2.4V;<br>3’b110: internal 3.0V;<br>3’b111: internal 3.6V; | R/W | 3‘b0 |
| tkadcf.1 | adc_vref1 | | | |
| tkadcf.0 | adc_vref0 | | | |

Note:

Bit tkadcf.3(cons_sta) is used for ADC continuously convert mode only, and for other usage is forbidden. And it must be set immediately after the ‘tkcon1.7’(start) is set, and make sure there is no interrupt to break the sequence, otherwise, the ‘tkcon1.7’(start) can only work once, it means ADC converts data one time only, the following steps should be obeyed for ADC continuously convert mode:

```
Step1: ORL   TKCON0, #040H   // make sure the others configuration(like channel) is configured right before
Step2: CLR   EA              // clear all interrupt before trigger ADC start
Step3: ORL   TKCON1, #080H   // trigger ADC start
Step4: ORL   TKADCF, #008H   // trigger ADC continuously mode start ‘cons_sta’
Step5:   // wait until ADC interrupt comes and get the ADC convert data from ‘tkdath’, ‘tkdatl’
```

Step6: // cleared ADC continuously mode start 'cons_sta' if it the ADC data is enough and then clear 'adcen'

### 23.2.10 Touch key status Register – TKCSCF

Table 23-11 TKCSCF Register (9Eh)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| tkcscf.7 | c2v_cha_sel2 | **Trimming the c2v charge time:**<br>3'b000: analog charge time last 6 clocks of 'tk_clk'<br>3'b001: analog charge time last 8 clocks of 'tk_clk'<br>3'b010: analog charge time last 12 clocks of 'tk_clk'<br>3'b011: analog charge time last 16 clocks of 'tk_clk'<br>3'b100: analog charge time last 20 clocks of 'tk_clk'<br>3'b101: analog charge time last 24 clocks of 'tk_clk'<br>3'b110: analog charge time last 28 clocks of 'tk_clk'<br>3'b111: analog charge time last 32 clocks of 'tk_clk' | R/W | 3'b0 |
| tkcscf.6 | c2v_cha_sel1 | | | |
| tkcscf.5 | c2v_cha_sel0 | | | |
| tkcscf.4~3 | - | Reserved. | R/W | 2'b0 |
| tkcscf.2 | cap_size2 | Configure the capacitance size range of touch key used :<br>3'b000: 1pf ~ 10pf<br>3'b001: 10pf ~ 20pf<br>3'b010: 20pf ~ 30pf<br>3'b011: 30pf ~ 40pf<br>3'b100: 40pf ~ 50pf<br>3'b101: 50pf ~ 60pf<br>3'b110: 60pf ~ 70pf<br>3'b111: 70pf ~ 80pf | R/W | 3'b0 |
| tkcscf.1 | cap_size1 | | | |
| tkcscf.0 | cap_size0 | | | |

### 23.2.11 Touch key status Register – TKCSOF

Table 23-12 TKCSOF Register (9Fh)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| tkcsof.7 | - | Reserved. | R/W | 1'b0 |
| tkcsof.6 | cpol | Select the polarity of comparison between convert data and wakeup threshold value configured into 'tkwk0/tkwk1':<br>1'b0: involves interrupt when data is more than 'tkwk1' or less than 'tkwk0' (X≤tkwk0 or X≥tkwk1)<br>1'b1: involves interrupt when data is less than 'tkwk1' and more than 'tkwk0' (tkwk0≤X≤tkwk1) | R/W | 1'b0 |

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| tkcsof.5 | ch_sw1 | Four type sources could be select as ADC channel:<br>2'b00: switch to P2, configure 'tkchs1[7:0]' to select any P2.7~P2.0 as channel;<br>2'b01: switch to internal 1.2v;<br>2'b10: switch to ground (0v);<br>2'b11: switch to ADC reference voltage; | R/W | 2'b0 |
| tkcsof.4 | ch_sw0 | | | |
| tkcsof.3~0 | cs_ofst3~0 | Touch key cap sensor offset adjust bits | R/W | 4'hF |

### 23.2.12 Touch key Wakeup threshold Register – TKWKL0

Table 23-13 TKWKL0 Register (D2h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| tkwkl0.7~0 | - | When 'tken' is set, the register store the touch key wakeup 0 threshold low 8bits data;<br>When 'adcen' is set, the register store the ADC limit 0 threshold low 8bits data; | R/W | 00h |

### 23.2.13 Touch key Wakeup threshold Register – TKWKH0

Table 23-14 TKWKH0 Register (D3h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| tkwkh0.7~0 | - | When 'tken' is set, the register store the touch key wakeup 0 threshold high 8bits data;<br>When 'adcen' is set, the register store the ADC limit 0 threshold high 8bits data; | R/W | 00h |

### 23.2.14 Touch key Wakeup threshold Register – TKWKL1

Table 23-15 TKWKL1 Register (D4h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| tkwkl1.7~0 | - | When 'tken' is set, the register store the touch key wakeup 1 threshold low 8bits data;<br>When 'adcen' is set, the register store the ADC limit 1 threshold low 8bits data; | R/W | 00h |

### 23.2.15 Touch key Wakeup threshold Register – TKWKH1

Table 23-16 TKWKH1 Register (D5h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| tkwkh1.7~0 | - | When 'tken' is set, the register store the touch key wakeup 1 threshold high 8bits data;<br>When 'adcen' is set, the register store the ADC limit 1 threshold high 8bits data; | R/W | 00h |

# 24 Analog Comparator

## 24.1 Overview

The device is provided one analog comparator. Input and output options allow use of the comparator in a number of different configurations. The comparator output is a logical one when its positive input is greater than its negative input, otherwise the output is a zero. The comparator can be configured to cause to an interrupt when the output value change. The block diagram is as below.

The comparator has two control registers CMPCON0 and CMPCON1, Both Inputs are CMP1, CMP2, CMPVREF and internal reference voltage, and output is CMPOUT. After enable comparator the comparator need waited stable time to guarantee comparator output.

The value of internal reference voltage (Vref) is 1.2V (+/-2%, @25°C).

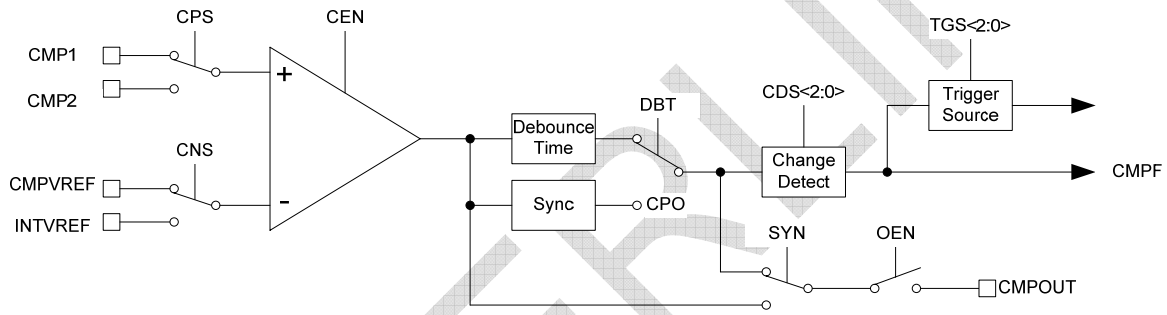



Figure 24-1 Analog Comparator

## 24.2 Register Definition

### 24.2.1 Comparator Control Register 0 – CMPCON0

Table 24-1 CMPCON0 Register (BDh)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| cmpcon0.7 | cen | **Comparator enable**<br>0: Disable Comparator<br>1: Enabled Comparator | R/W | 0 |
| cmpcon0.6 | cps | **Comparator positive input select**<br>0: CMP1 is selected as the positive comparator input<br>1: CMP2 is selected as the positive comparator input | R/W | 0 |
| cmpcon0.5 | cns | **Comparator negative input select**<br>0: CMPVREF is selected as the negative comparator input | R/W | 0 |

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| | | 1: INTVREF is selected as the negative comparator input | | |
| cmpcon0.4 | oen | **Comparator output enable**<br>1: CMPOUT output is enabled if CEN = 1<br>0: CMPOUT output is disabled | R/W | 0 |
| cmpcon0.3 | cpo | **Comparator output**<br>Synchronized to CPU clock to allow reading by software.<br>Cleared when the comparator is disabled (CEN = 0). | R/W | 0 |
| cmpcon0.2 | dbt | **Comparator debounce time enable**<br>1: enable 8 system clocks debounce time<br>0: disable debounce time, only synchronous | R/W | 0 |
| cmpcon0.1 | syn | **Comparator synchronous/asynchronous selection**<br>1: synchronous output<br>0: asynchronous output | R/W | 0 |
| cmpcon0.0 | - | - | R | 0 |

### 24.2.2 Comparator Control Register 1 – CMPCON1

Table 24-2 CMPCON1 Register (BEh)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| cmpcon1.7 | - | - | R | 0 |
| cmpcon1.6 | tgs2 | **Comparator trigger source selection** | R/W | 0 |
| cmpcon1.5 | tgs1 | | R/W | 0 |
| cmpcon1.4 | tgs0 | | R/W | 0 |
| cmpcon1.3 | vref_en | Internal vref12 enable, if set, vref12 enabled, if you intend to select internal vref12 as one reference voltage to compare, this bit should be set before the 'cen' is set for the reason that internal vref12 stable time is about 60us. | R/W | 0 |
| cmpcon1.2 | cds2 | **Comparator interrupt change detect selection** | R/W | 0 |
| cmpcon1.1 | cds1 | | R/W | 0 |
| cmpcon1.0 | cds0 | | R/W | 0 |

Table 24-3 Comparator Interrupt Change Selection

| CDS2 | CDS1 | CDS0 | Comparator Interrupt Change Detect Selection |
|---|---|---|---|
| 0 | 0 | 0 | Low level |
| 0 | 0 | 1 | High level |
| 0 | 1 | 0 | Falling Edge |
| 0 | 1 | 1 | Rising Edge |

| CDS2 | CDS1 | CDS0 | Comparator Interrupt Change Detect Selection |
|---|---|---|---|
| 1 | 0 | 0 | Toggle |
| 1 | 0 | 1 | - |
| 1 | 1 | 0 | - |
| 1 | 1 | 1 | - |

Table 24-4 Comparator Trigger Source Selection

| TGS2 | TGS1 | TGS0 | Comparator Trigger Source Selection |
|---|---|---|---|
| 0 | 0 | 0 | Normal mode |
| 0 | 0 | 1 | Switch-off PWM0 output, clear PWM0EN |
| 0 | 1 | 0 | Switch-off PWM1 output, clear PWM1EN |
| 0 | 1 | 1 | Switch-off PWM0 and PWM1 output, clear PWM0EN and PWM1EN |
| 1 | 0 | 0 | Trigger Timer 2 Capture Operation |
| 1 | 0 | 1 | Switch-on PWM0 output, set PWM0EN |
| 1 | 1 | 0 | Switch-on PWM1 output, set PWM1EN |
| 1 | 1 | 1 | Switch-on PWM0 and PWM1 output, set PWM0EN and PWM1EN |

# 25 Flash & EEPROM

The memory contains 4K bytes program Flash code area, 128 bytes data EEPROM code area.

- 4K bytes program Flash
- 128 bytes data EEPROM (page/byte operation)

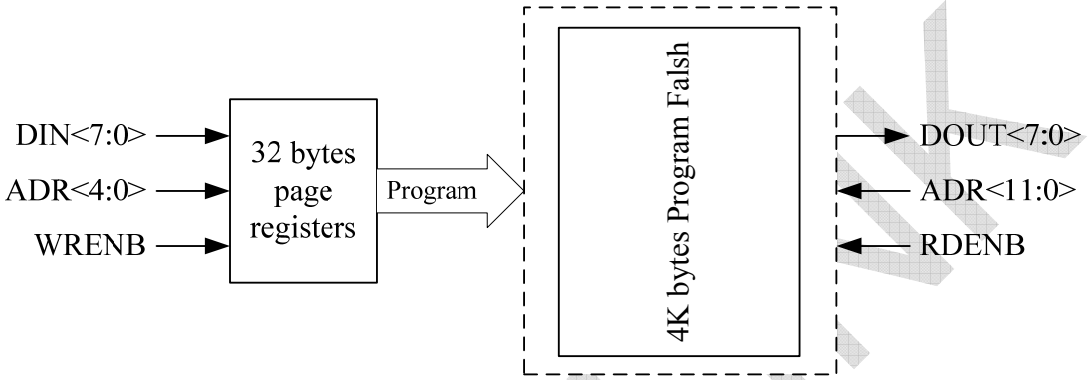



Figure 25-1 Flash Area

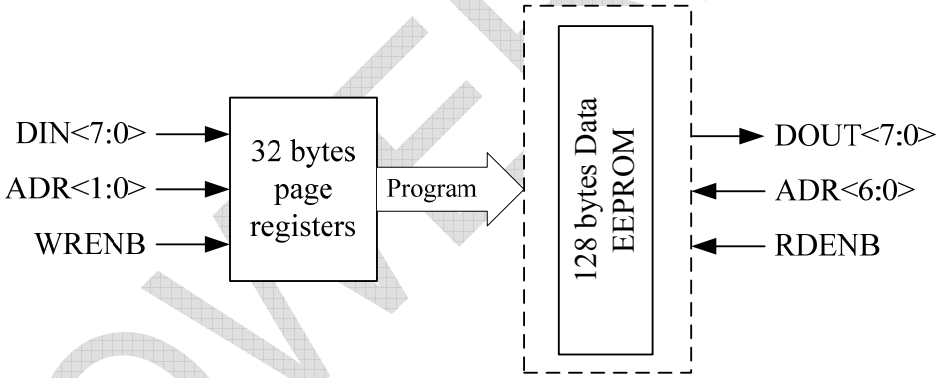



Figure 25-2 EEPROM Area

## 25.1 Memory Encryption

The program code area of Flash is encrypted in this device.

## 25.2 Register Definition

### 25.2.1 EEPROM Control Register – EECON

Table 25-1 EECON Register (97h)

| Bit | Symbol | Description | Type | Reset |
|---|---|---|---|---|
| eecon.7 | LOCK | **EEPROM program inhibit**<br>0 – EEPROM program is enabled<br>1 – EEPROM program is inhibited | R/W | 0 |
| eecon.6 | - | - | R | 0 |
| eecon.5 | - | - | R | 0 |
| eecon.4 | - | - | R | 0 |
| eecon.3 | EPGM | **EEPROM program interrupt enable**<br>When epgm=0 EEPROM program interrupt is disabled.<br>When epgm=1 and ea=1 EEPROM program interrupt is enabled. | R/W | 0 |
| eecon.2 | PGMF | **EEPROM program interrupt flag**<br>1 – EEPROM program is finished<br>It can only be set by hardware and can be cleared by software or interrupt. When set PGM to 1, it will be cleared automatically. | R/W | 0 |
| eecon.1 | CPF | **EEPROM program cross page flag**<br>1 – EEPROM program page is changed (cross page)<br>If CPF=1, PGM can not be set to 1 until CPF is cleared by software. CPF can only be set to 1 by hardware; it can not be set to 1 by software.<br>After cross page error occurred, more than 3 NOP must be followed close behind the CPF cleared instruction to avoid the reset operation of EEPROM. | R/W | 0 |
| eecon.0 | PGM | **EEPROM program enable**<br>1 – start EEPROM program<br>After write data to EEPROM buffer, set it to start EEPROM program. If EEPROM buffer is not written, software can not set it. When program is finished, it is cleared by hardware automatically. It can not be cleared by software. | R/TW | 0 |

# 26 ICP (In-Circuit Programming)

## 26.1 Overview

The contexts of flash in the device are empty by default. User must program the flash by external Writer device or by ICP (In-Circuit Programming) tool.

In the ICP tool, the user must take note of ICP's programming pins used in system board. In some application circuits, it is highly recommended customer power off then power on after ICP programming has completed on the system board.

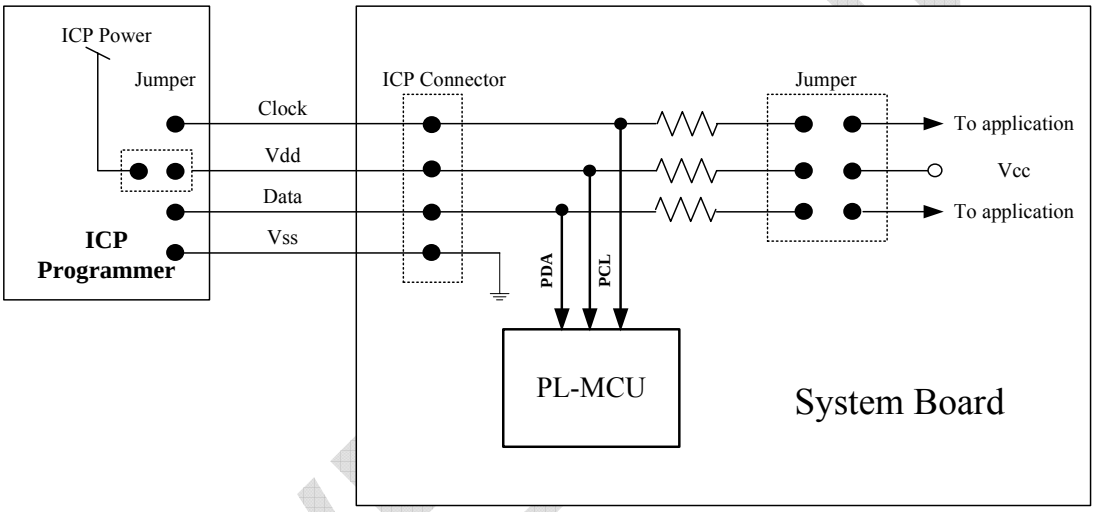



Figure 26-1 ICP Application Circuit

Note:

1. Circuitry separation is optionally needed between ICP and application during ICP operation.
2. Resistor is optional by application
3. When using ICP to upgrade code, the clock PCL and data PDA must be taken within design system board.
4. After program finished by ICP, to suggest system power must power off and remove ICP connector then power on.

The device supports programming of Flash (4K bytes AP Flash), and data EEPROM (128 bytes). User has the option to program the AP Flash and data EEPROM either individually or both.

# 27 Config Options

The config options are used for code configuration.

| Config Option0 | Config Option1 | Config Option2 |
|---|---|---|
| **Program Flash lock bit** | **Reserved** | **LVR enable**<br>0x – disabled<br>10 – always enabled<br>11 – enabled except EE program |
| **Data EEPROM lock bit** | **Reset pin enable** | |
| **XTAL options**<br>OPT[0]:<br>0 -> use inner feedback resistor<br>1 -> no inner feedback resistor<br>OPT[1]:<br>0 -> select inner 1M resistor<br>1 -> select inner 4M resistor<br>OPT[2]:<br>0 -> no 15pf inner cap<br>1 -> use 15pf inner cap | **Pump clock vision select**<br>0 –Not dived the pump clock<br>1 –Dived the pump clock by 2 | **LVR threshold level**<br>00 – 2.1v<br>01 – 2.4v<br>10 – 3.7v<br>11 – 4.3v |
| | **Clock output enable** | |
| | **Warm-up time**<br>00~11:<br>long warm-up time<br>→<br>short warm-up time | **LPD** enable |
| **Oscillator type selection**<br>000 – external clk input mode<br>001 – Internal RC 32KHz<br>01x – Crystal Oscillator<br>100 – Internal RC 4MHz<br>101 – Internal RC 8MHz<br>110 – Internal RC 12MHz<br>111 – reserved | | **LPD threshold level**<br>0 – 2.7v<br>1 – 4.0v |
| | **Time-out delay**<br>00 – 2176 clocks (66ms)<br>01 – 640 clocks (20ms)<br>10 – 384 clocks (12ms)<br>11 – 132 clocks (4ms) | **WDT enable**<br>0x – disabled<br>10 – enabled<br>controlled by WDTEN<br>11 – enabled<br>controlled by WDTEN<br>disabled in STOP mode |

# 28 Electrical Characteristics

## 28.1 Absolute Maximum Ratings

Absolute maximum ratings are the parameter values or ranges which can cause permanent damage and affect device reliability if exceeded.

| Parameter | Symbol | Min. | Max. | Unit |
|---|---|---|---|---|
| DC Power Supply | VDD-VSS | -0.3 | +6.0 | V |
| Input Voltage | $V_{IN}$ | VSS-0.3 | VDD+0.3 | V |
| Operating Temperature | $T_A$ | -40 | +125 | °C |
| Storage Temperature | $T_{ST}$ | -55 | +150 | °C |
| Maximum Current into VDD | | | 120 | mA |
| Maximum Current out of VSS | | | 120 | mA |
| Maximum Current suck by a I/O pin | | | 25 | mA |
| Maximum Current sourced by a I/O pin | | | 25 | mA |
| Maximum Current suck by total I/O pins | | | 75 | mA |
| Maximum Current sourced by total I/O pins | | | 75 | mA |

**Note:** These are stress ratings only. Stress beyond these limits may cause permanent damage to the device. Functional operation of the device at these or any conditions beyond those indicated under Recommended Operating Conditions is not implied. Exposure to absolute maximum rated conditions for extended periods of time may affect device reliability.

## 28.2 DC Electrical Characteristics

(VDD = 2.0V~5.5V, $T_A$ = 25°C, unless otherwise specified)

| Parameter | Symbol | Specification | | | | Test Conditions |
|---|---|---|---|---|---|---|
| | | Min. | Typ. | Max. | Units | |
| Operating Voltage | VDD | 2.0 | | 5.5 | V | Freq: ~4MHz |
| | | 2.4 | | 5.5 | | Freq: ~8MHz |
| | | 2.7 | | 5.5 | | Freq: ~12MHz |
| Operation Current | $I_{OP}$ | | 5 | 10 | mA | No load, VDD=5V@12MHz |
| IDLE Current | $I_{IDLE}$ | | 2.5 | 5 | mA | No load, VDD=5V@12MHz, IDLE |
| Power Down Current | $I_{STOP}$ | | 1 | 10 | uA | No load, VDD=5V@12MHz, STOP |
| Power Down Current | $I_{SLEEP}$ | | 0.5 | 5 | uA | No load, VDD=5V@12MHz, Sleep |
| Input High Voltage | $V_{IH}$ | 0.7*VDD | | VDD+0.2 | V | |
| Input Low Voltage | $V_{IL}$ | -0.5 | | 0.3*VDD | V | |
| Output High Voltage | $V_{OH}$ | 2.5 | 3.5 | | V | VDD=5V, $I_{OH}$=-20mA |
| Output Low Voltage | $V_{OL}$ | | 0.5 | 0.7 | V | VDD=5V, $I_{OL}$=+20mA |
| Port Pull up Resistor | $R_{PU}$ | | 50 | | KΩ | |
| POR slope rate | $S_{POR}$ | 0.025 | | 4.5 | V/ms | |
| POR threshold voltage of rising | $V_{PORH}$ | 0.4 | 0.8 | 1.2 | V | |
| POR threshold voltage of falling | $V_{PORL}$ | 0.4 | 0.7 | 1.0 | V | |
| POE threshold voltage | $V_{POE}$ | | 1.4 | | V | |
| Comparator Reference Voltage | Vref | 1.176 | 1.20 | 1.224 | V | $T_A$ = 25°C |

## 28.3 AC Electrical Characteristics

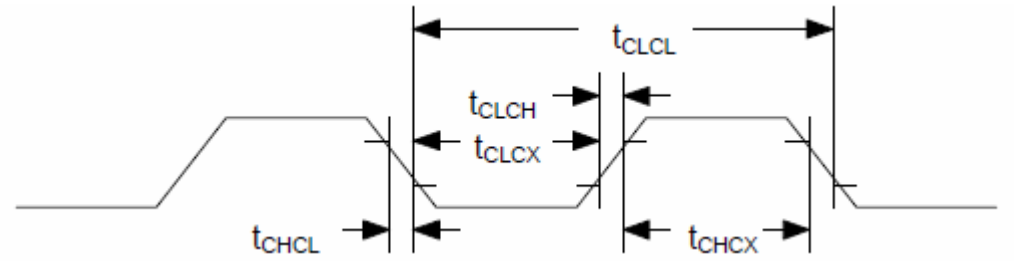



Figure 22-1 Clock Timing

**Note:** Duty cycle is 50%.

### 28.3.1 External Clock Characteristics

(VDD = 2.0V~5.5V, $T_A$ = 25°C, unless otherwise specified)

| Parameter | Symbol | Specification | | | | Test Conditions |
|---|---|---|---|---|---|---|
| | | Min. | Typ. | Max. | Units | |
| Clock Frequency | | | 12 | | MHz | |
| Clock High Time | $t_{CHCX}$ | 30 | | | ns | |
| Clock Low Time | $t_{CLCX}$ | 30 | | | ns | |
| Clock Rise Time | $t_{CLCH}$ | | | 10 | ns | |
| Clock Fall Time | $t_{CHCL}$ | | | 10 | ns | |

### 28.3.2 Internal RC OSC Characteristics

(VDD = 2.0V~5.5V, $T_A$ = 25°C, unless otherwise specified)

| Parameter | Symbol | Specification | | | | Test Conditions |
|---|---|---|---|---|---|---|
| | | Min. | Typ. | Max. | Units | |
| Clock Frequency | | | 4/8/12 | | MHz | |
| Clock Frequency | | | ±2 | | % | $T_A$ = 25°C@8MHz |

### 28.3.3 Crystal Oscillator/Ceramic Resonator Characteristics

(VDD = 2.0V~5.5V, $T_A$ = 25°C, unless otherwise specified)

| Parameter | Symbol | Specification | | | | Test Conditions |
|---|---|---|---|---|---|---|
| | | Min. | Typ. | Max. | Units | |
| Clock Frequency | | 400K | | 12M | Hz | |

## 28.4 Comparator Electrical Characteristics

TA = 25°C, VCC = 2.4V to 5.5V (unless otherwise noted)

| Parameter | Symbol | Specification | | | | Test Conditions |
|---|---|---|---|---|---|---|
| | | Min. | Typ. | Max. | Units | |
| Common mode range comparator inputs | $V_{CR}$ | 0 | | VDD | V | |
| Comparator response time | $T_{RS}$ | | 30 | | ns | |
| Comparator enable to output valid time | $T_{EN}$ | | 50 | | us | |

# 29 Typical Application

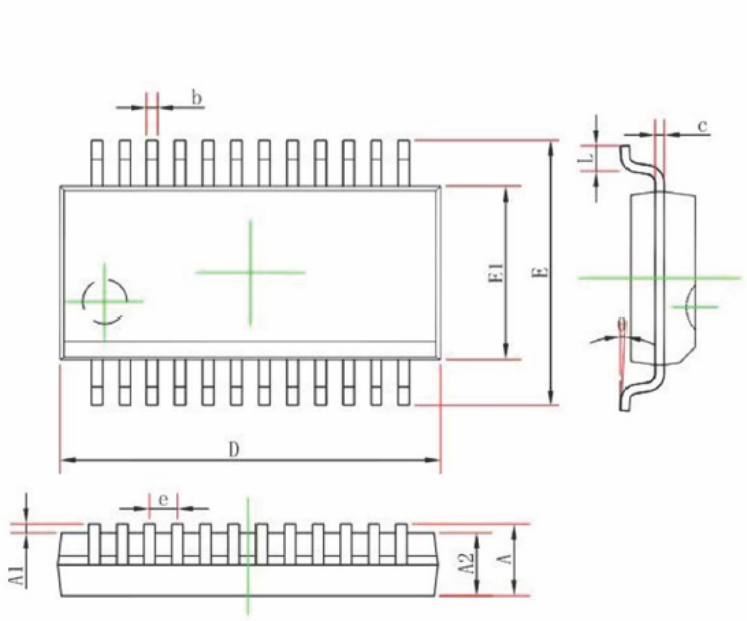
## 29.1 Touchkey Demo

N/A

# 30 Package Dimensions

## 30.1 SSOP24 Package

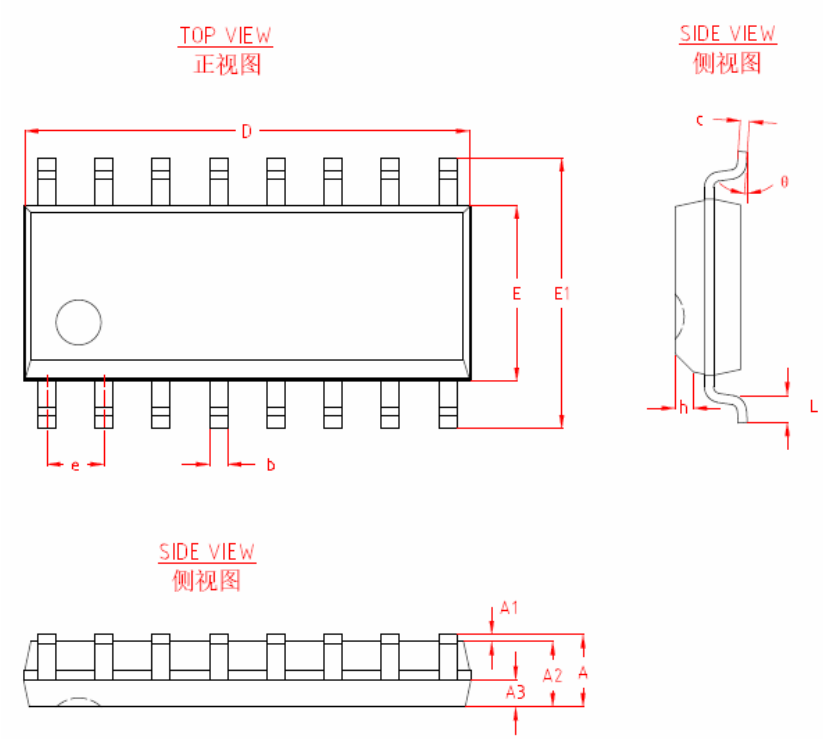
SSOP24 PACKAGE OUTLINE DIMENSIONS




| Symbol | Dimensions In Millimeters | |
|---|---|---|
| | **Min** | **Max** |
| A | —— | 1.750 |
| A1 | 0.050 | 0.080 |
| A2 | 1.400 | 1.500 |
| b | 0.203 | 0.305 |
| c | 0.102 | 0.254 |
| D | 8.550 | 8.650 |
| E1 | 3.800 | 4.000 |
| E | 5.800 | 6.200 |
| e | 0.635 (BSC) | |
| L | 0.400 | 1.270 |
| θ | 0° | 8° |

## 30.2 SOP16 Package

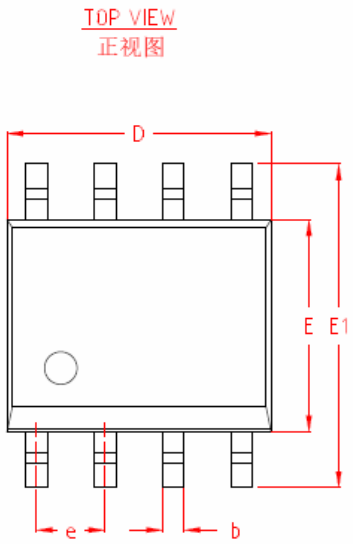
SOP16 PACKAGE OUTLINE DIMENSIONS

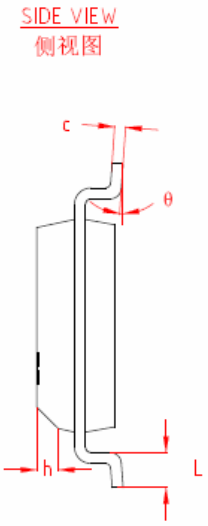



| 机械尺寸/mm Dimensions | | | |
|---|---|---|---|
| 字符 SYMBOL | 最小值 MIN | 典型值 NOMINAL | 最大值 MAX |
| A | – | – | 1.75 |
| A1 | 0.10 | – | 0.25 |
| A2 | 1.35 | 1.45 | 1.55 |
| A3 | 0.60 | 0.65 | 0.70 |
| b | 0.35 | – | 0.50 |
| c | 0.19 | – | 0.25 |
| D | 9.80 | 10.00 | 10.20 |
| E | 3.80 | 3.90 | 4.00 |
| E1 | 5.80 | 6.00 | 6.20 |
| e | 1.27 BSC | | |
| h | 0.30 | – | 0.50 |
| L | 0.40 | – | 0.80 |
| θ | 0° | – | 8° |

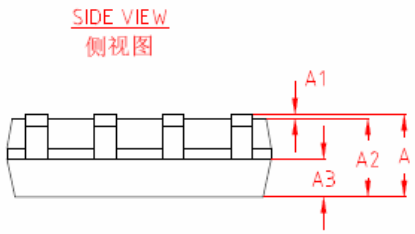
## 30.3 SOP8 Package

SOP8 PACKAGE OUTLINE DIMENSIONS



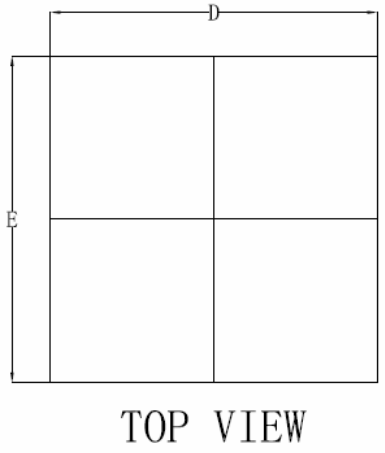



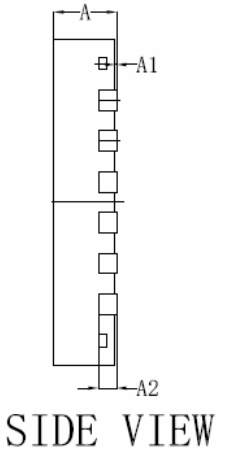

| 机械尺寸/mm Dimensions | | | |
|---|---|---|---|
| 字符 SYMBOL | 最小值 MIN | 典型值 NOMINAL | 最大值 MAX |
| A | 1.50 | 1.60 | 1.70 |
| A1 | 0.04 | – | 0.12 |
| A2 | 1.35 | 1.45 | 1.55 |
| A3 | 0.65 | 0.70 | 0.75 |
| b | 0.35 | – | 0.50 |
| c | 0.19 | – | 0.25 |
| D | 4.80 | 4.90 | 5.00 |
| E | 3.80 | 3.90 | 4.00 |
| E1 | 5.80 | 6.00 | 6.20 |
| e | 1.27 BSC | | |
| h | 0.30 | – | 0.50 |
| L | 0.50 | – | 0.80 |
| θ | 0° | – | 8° |



## 30.4 QFN24 Package

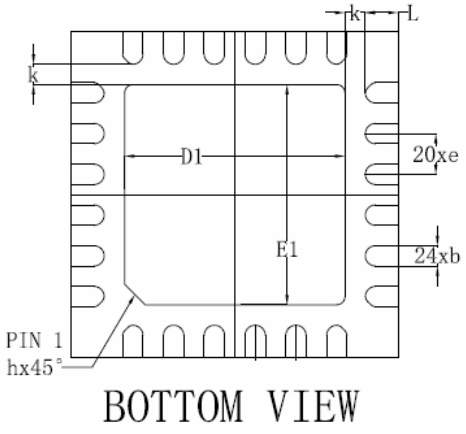
QFN24(4*4, P0.50T0.75) PACKAGE OUTLINE DIMENSIONS



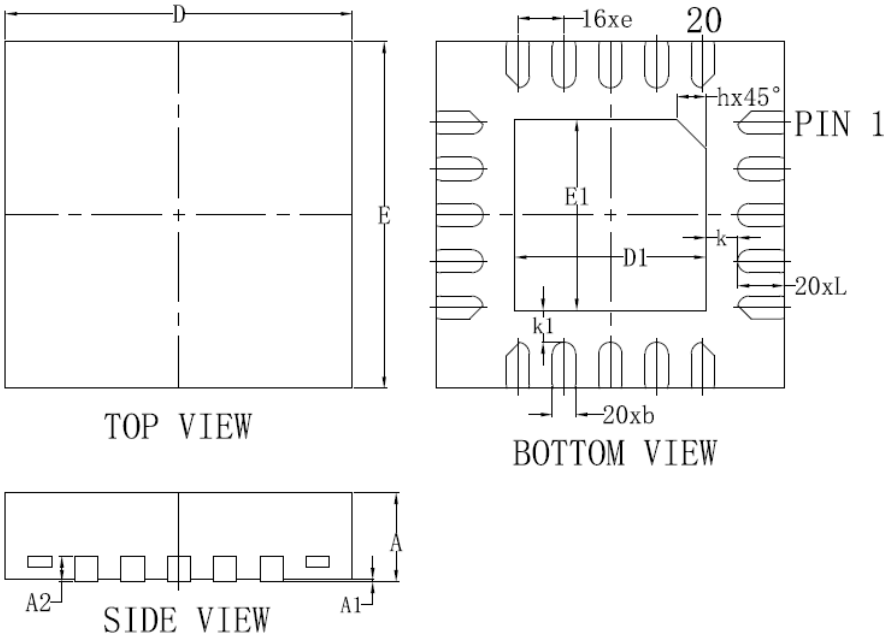

COMMON DIMENSIONS
(UNITS OF MEASURE=MILLIMETER)

| SYMBOL | MIN | NOM | MAX |
|---|---|---|---|
| A | 0.700 | 0.750 | 0.820 |
| A1 | 0.000 | / | 0.050 |
| A2 | 0.153 | 0.203 | 0.273 |
| b | 0.200 | 0.250 | 0.300 |
| D | 3.900 | 4.000 | 4.100 |
| D1 | 2.600 | 2.700 | 2.800 |
| E | 3.900 | 4.000 | 4.100 |
| E1 | 2.600 | 2.700 | 2.800 |
| e | 0.450 | 0.500 | 0.550 |
| h | 0.200 | 0.250 | 0.300 |
| k | 0.150 | 0.250 | 0.350 |
| L | 0.350 | 0.400 | 0.450 |



## 30.5 QFN20 Package

QFN20(3*3, P0.40T0.75) PACKAGE OUTLINE DIMENSIONS

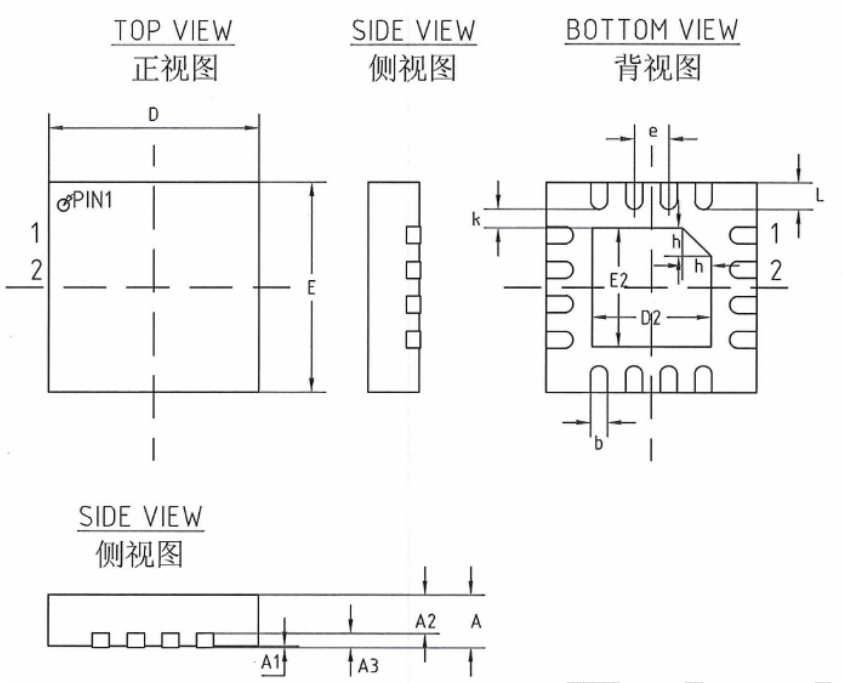



COMMON DIMENSIONS
(UNITS OF MEASURE=MILLIMETER)

| SYMBOL | MIN | NOM | MAX |
|---|---|---|---|
| A | 0.700 | 0.750 | 0.800 |
| A1 | 0.000 | / | 0.050 |
| A2 | 0.193 | 0.203 | 0.233 |
| b | 0.150 | 0.200 | 0.250 |
| D | 2.950 | 3.000 | 3.050 |
| D1 | 1.600 | 1.650 | 1.700 |
| E | 2.950 | 3.000 | 3.050 |
| E1 | 1.600 | 1.650 | 1.700 |
| e | 0.350 | 0.400 | 0.450 |
| h | 0.200 | 0.250 | 0.300 |
| k | 0.225 | 0.275 | 0.325 |
| k1 | 0.225 | 0.275 | 0.325 |
| L | 0.350 | 0.400 | 0.450 |

## 30.6 QFN16 Package

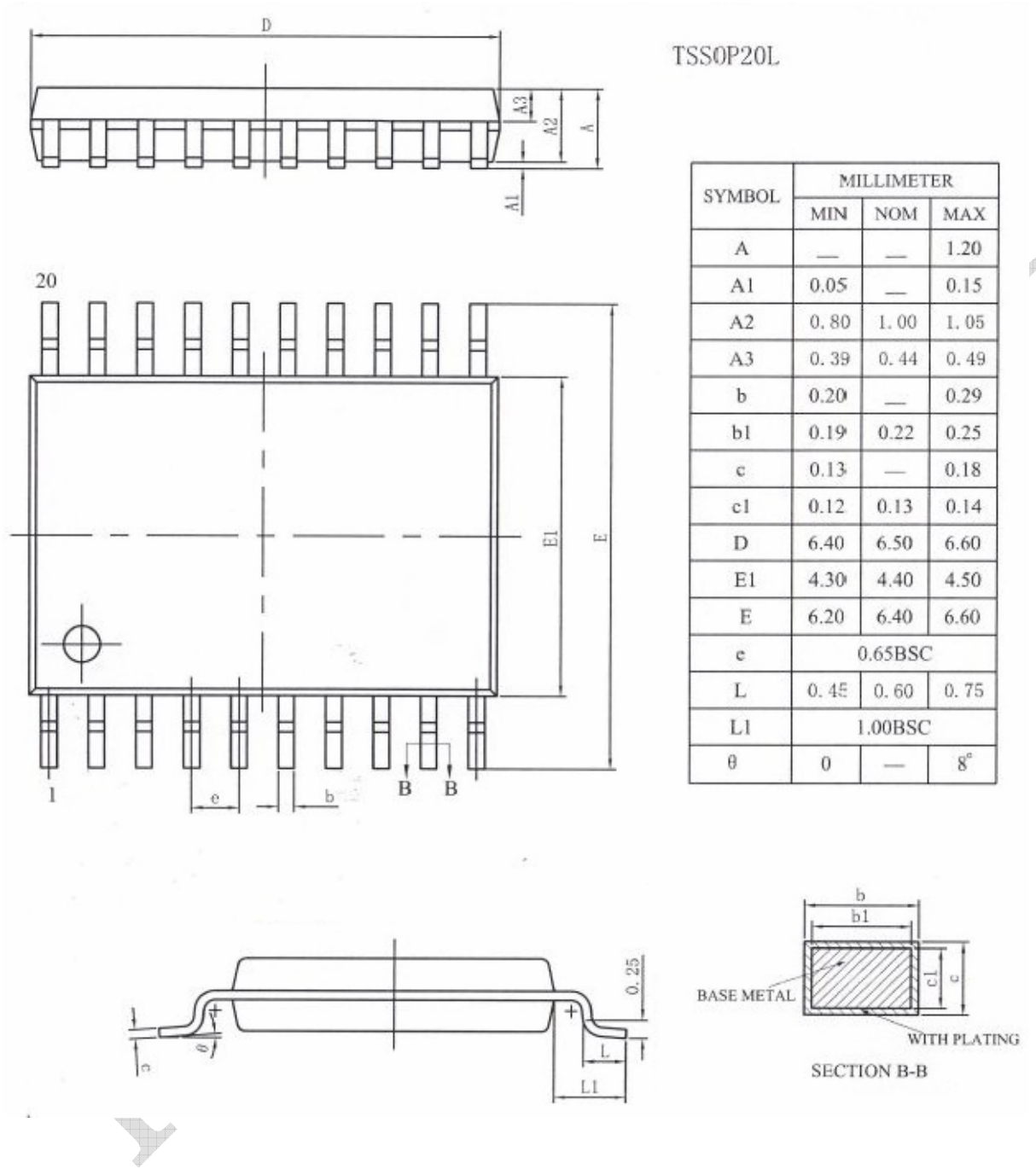
QFN16(3*3, P0.50T0.75) PACKAGE OUTLINE DIMENSIONS




| 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 | 0.02 | 0.05 |
| A2 | – | 0.55 | – |
| A3 | 0.203 REF | | |
| b | 0.18 | 0.24 | 0.30 |
| D | 3 BSC | | |
| E | 3 BSC | | |
| e | 0.50 BSC | | |
| D2 | 1.6 | 1.7 | 1.8 |
| E2 | 1.6 | 1.7 | 1.8 |
| K | 0.20 BCS | | |
| L | 0.30 | 0.40 | 0.50 |
| h | 0.35 | 0.40 | 0.45 |

# 30.7 TSSOP20 Package

TSSOP20 PACKAGE OUTLINE DIMENSIONS

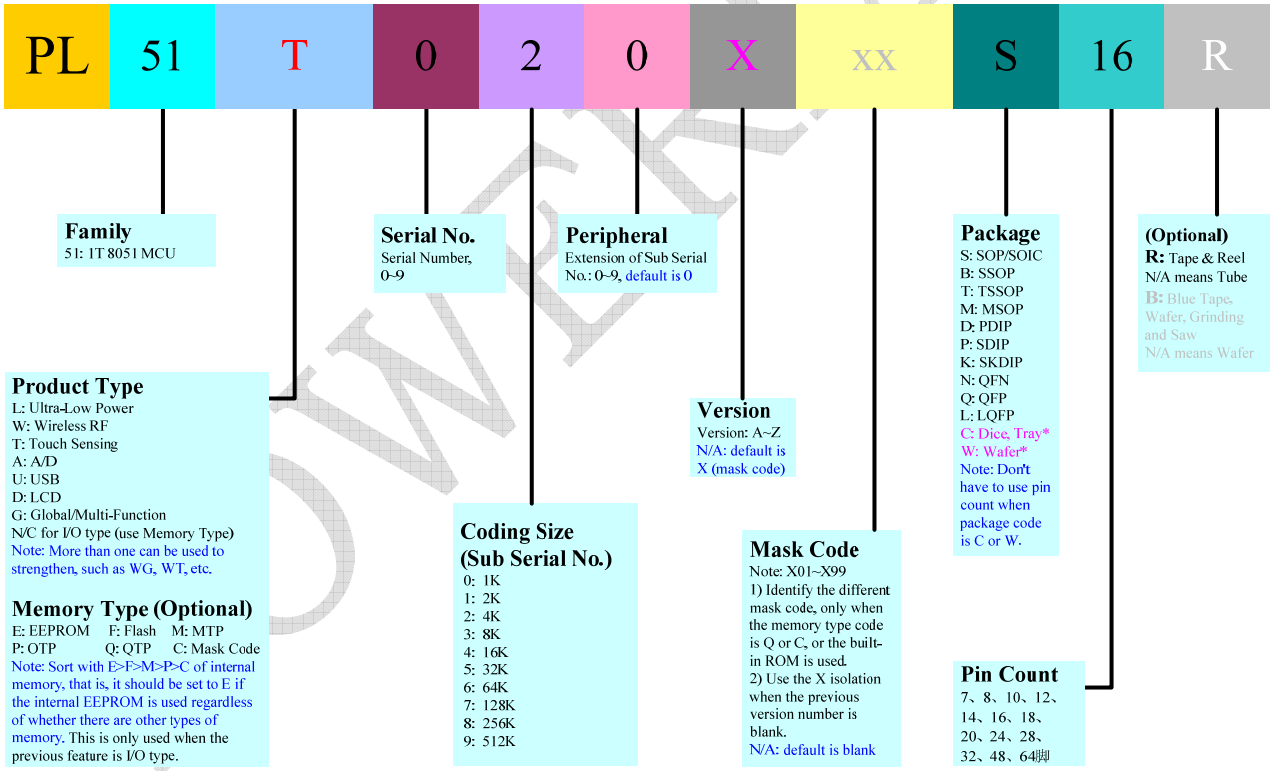



TSSOP20L

| SYMBOL | MILLIMETER | | |
|---|---|---|---|
| | MIN | NOM | MAX |
| A | — | — | 1.20 |
| A1 | 0.05 | — | 0.15 |
| A2 | 0.80 | 1.00 | 1.05 |
| A3 | 0.39 | 0.44 | 0.49 |
| b | 0.20 | — | 0.29 |
| b1 | 0.19 | 0.22 | 0.25 |
| c | 0.13 | — | 0.18 |
| c1 | 0.12 | 0.13 | 0.14 |
| D | 6.40 | 6.50 | 6.60 |
| E1 | 4.30 | 4.40 | 4.50 |
| E | 6.20 | 6.40 | 6.60 |
| e | 0.65BSC | | |
| L | 0.45 | 0.60 | 0.75 |
| L1 | 1.00BSC | | |
| θ | 0 | — | 8° |

# 31 Ordering Information

| Part Number | Packaging |
|---|---|
| PL51T020C | Dice,Tray |
| PL51T020W | Wafer |
| PL51T020WB | Wafer, Blue Tape |
| PL51T020S8 | SOP8, Tube |
| PL51T020S8R | SOP8, Tape&Reel |
| PL51T020S16 | SOP16, Tube |
| PL51T020B24 | SSOP24, Tube |
| PL51T020N16 | QFN16, Tube |
| PL51T020N20 | QFN20, Tube |
| PL51T020N24 | QFN24, Tube |
| PL51T020T20 | TSSOP20, Tube |
| …… | …… |



# 32 Document Revision History

| Rev. | Date | Comments |
|---|---|---|
| 0.4 | 2016/11/18 | Preliminary Version<br>Add Detail Description about Package and EEPROM config options |
| 0.5 | 2017/02/14 | PL51T020: Change Memory's description and Package Dimensions<br>SSOP24/TSSOP20/SOP16/SOP8/QFN16/QFN24 |
| 0.6 | 2017/03/08 | Updated some descriptions: Timeout Delay, CPF register, level-activated external interrupt wake-up from STOP&SLEEP, Vref to±2%@25℃, and data EEPROM program condition (enable LVR≥2.4V) |
| 0.7 | 2017/10/10 | Updated some descriptions: production description, overview, ADC scan mode, and SPI interface, etc. |
| 0.8 | 2018/07/11 | Operation Temperature |
| 0.9 | 2019/06/05 | Fixed Table 32-1 ADR: PWM2PH/PL/DH/DL & PWM3PH/PL/DH/DL<br>Removed TSSOP20 Package |
| 1.0 | 2019/11/01 | Operation Temperature: -40~125℃ |
| 1.1 | 2020/04/10 | Updated Package Dimensions |
| 1.2 | 2020/07/03 | Fixed TMOD.2/SPI/TCKCON |
| 1.3 | 2021/01/20 | Added TSSOP20 Package |
| 1.4 | 2021/02/03 | Added QFN20 Package |
| 1.5 | 2021/03/18 | Updated Part Number Description |
| | | |

# 33 Important Notice

POWERLINK reserves the right to make changes or corrections to its products at any time without notice. Customers should verify the datasheets are current and complete before placing order.