# EPD8000

## ELAN RISC II™ Series

# Product Specification

**DOC. VERSION 2.2**

**ELAN MICROELECTRONICS CORP.**

October 2008

**Trademark Acknowledgments:**
IBM is a registered trademark and PS/2 is a trademark of IBM
Windows is a trademark of Microsoft Corporation
ELAN and ELAN logo are trademarks of ELAN Microelectronics Corporation



The contents of this specification are subject to change without further notice. ELAN Microelectronics assumes no responsibility concerning the accuracy, adequacy, or completeness of this specification. ELAN Microelectronics makes no commitment to update, or to keep current the information and material contained in this specification. Such information and material may change to conform to each confirmed order.

In no event shall ELAN Microelectronics be made responsible for any claims attributed to errors, omissions, or other inaccuracies in the information or material contained in this specification. ELAN Microelectronics shall not be liable for direct, indirect, special incidental, or consequential damages arising from the use of such information or material.

The software (if any) described in this specification is furnished under a license or nondisclosure agreement, and may be used or copied only in accordance with the terms of such agreement.

ELAN Microelectronics products are not intended for use in life support appliances, devices, or systems. Use of ELAN Microelectronics product in such applications is not supported and is prohibited.
NO PART OF THIS SPECIFICATION MAY BE REPRODUCED OR TRANSMITTED IN ANY FORM OR BY ANY MEANS WITHOUT THE EXPRESSED WRITTEN PERMISSION OF ELAN MICROELECTRONICS.

**ELAN MICROELECTRONICS CORPORATION**

**Headquarters:**
No. 12, Innovation Road 1
Hsinchu Science Park
Hsinchu, TAIWAN 308
Tel: +886 3 563-9977
Fax: +886 3 563-9966
http://www.emc.com.tw

**Hong Kong:**
**Elan (HK) Microelectronics Corporation, Ltd.**
Flat A, 19F., World Tech Centre
95 How Ming Street, Kwun Tong
Kowloon, HONG KONG
Tel: +852 2723-3376
Fax: +852 2723-7780

**USA:**
**Elan Information Technology Group (U.S.A.)**
PO Box 601
Cupertino, CA 95015
U.S.A.
Tel: +1 408 366-8225
Fax: +1 408 366-8225

**Shenzhen:**
**Elan Microelectronics Shenzhen, Ltd.**
3F, SSMEC Bldg., Gaoxin S. Ave. I
Shenzhen Hi-tech Industrial Park
(South Area), Shenzhen
CHINA 518057
Tel: +86 755 2601-0565
Fax: +86 755 2601-0500
elan-sz@elanic.com.cn

**Shanghai:**
**Elan Microelectronics Shanghai, Ltd.**
#23, Zone 115, Lane 572, Bibo Rd.
Zhangjiang Hi-Tech Park
Shanghai, CHINA 201203
Tel: +86 21 5080-3866
Fax: +86 21 5080-4600
elan-sh@elanic.com.cn

# Contents


**1 General Description ..... 1**
**2 Features ..... 1**
**3 Applications ..... 1**
**4 Block Diagram ..... 2**
**4 Pin Assignment ..... 3**
**5 Pin Description ..... 4**
**6 Code Option ..... 5**
**7 Function Description ..... 6**
7.1 Reset Function: ..... 6
7.1.1 Power-up and Reset Timing ..... 7
7.1.2 Register Initial Values ..... 9
7.2 Oscillator System ..... 10
7.2.1 32.768kHz Crystal or 32.8kHz RC ..... 10
7.2.2 Phase Locked Loop (PLL) ..... 11
7.3 MCU Operation Mode ..... 12
7.4 Wake-up Function ..... 14
7.5 Interrupt ..... 15
7.5.1 Input Port A Interrupt ..... 15
7.5.2 Timer 0, Timer 1, and Timer 2 Interrupts ..... 16
7.5.3 Peripheral Interrupt ..... 16
7.6 Program ROM Map: ..... 17
7.7 Data ROM Map ..... 17
7.8 RAM Map Register ..... 18
7.9 LCD RAM Map: 392 Bytes ..... 21
7.10 Special Register Description ..... 21
**8 Peripheral ..... 27**
8.1 Timer 0 (16 Bits): ..... 27
8.2 Timer 1 (8 Bits): ..... 29
8.3 Timer 2 (8 Bits) ..... 32
8.4 Melody Timer (8 Bits) ..... 34
8.5 EL or IR Generator Timer (6 Bits) ..... 36
8.6 Watchdog Timer (WDT) ..... 38
8.7 Low Voltage Detection (LVD) ..... 40
8.8 Serial Peripheral Interface (SPI) ..... 44
8.8.1 SPI Pin Description ..... 47
8.8.2 SPI Applicable Registers ..... 47

8.9 Key I/O ..... 52
8.9.1 Automatic Key Scan or Software Key Scan ..... 52
8.9.2 Input/Output Key Applicable Registers ..... 56
8.10 Embedded LCD Driver ..... 60
8.10.1 LCD Code Option ..... 61
**9 Electrical Characteristic ..... 65**
9.1 Absolute Maximum Ratings ..... 65
9.2 Recommended Operating Conditions ..... 65
9.3 DC Electrical Characteristics ..... 66
9.4 AC Electrical Characteristics ..... 69
**10 Pin Type Circuit Diagrams ..... 70**
**11 Application Circuit ..... 72**
**12 Instruction Set: ..... 74**
**13 Pad Diagram ..... 77**

## Specification Revision History

| Doc. Version | Revision Description | Date |
|---|---|---|
| 1.0 | Initial version | 2001/02/07 |
| 1.1 | 1. New CPU series renamed as ELAN RISC II™ Series<br>2. Updated the TEST pin type<br>3. Exchanged the pin number of V4 & VCA pad<br>4. Modified the LVD spec to Vdet ± 0.1V, and added the Vdet hysteresis voltage description.<br>5. Modified the DC Specification temperature range to Ta=0~+40°C<br>6. Added Pad Diagram<br>7. Specified the V0 spec.: VDD=2.3V~3.3V, V0= Typ ± 4%<br>8. Specified the Vref temperature coefficient: 7mV/°C | 2001/05/15 |
| 1.2 | EPD-8000 renamed as EPD8000. | 2001/06/20 |
| 1.3 | Package Type: COB or QFP208<br>Added Application field | 2002/03/14 |
| 1.4 | Changed the Fast mode current consumption DC Spec.<br>Changed the TEST pin small pull down resistance DC Spec. | 2002/04/09 |
| 1.5 | Corrected the test condition of Idd4 on the DC Spec. | |
| 1.6 | Added the note illustration when users disable all interrupts (BC CPUCON,GLINTD). | 2003/01/06 |
| 2.0 | New format for EPD8000 Spec.<br>Modified the code option setting description.<br>Modified the LVD function timing description. | 2005/02/15 |
| 2.1 | New released version | 2005/12/15 |
| 2.2 | Modified the POP and PUSH interrupt macro of the Code Example in Section 7.5. | 2008/10/16 |

# 1 General Description

The **EPD8000** is an 8-bit RISC MCU with embedded two 8-bit timers and one 16-bit general timer, melody timer, EL or IR generator, Watchdog Timer, SPI, low voltage detector and a 32×98 LCD driver. In addition, the EPD8000 has an embedded large size user RAM, and program/data memory. It is suitable for educational learning tools application requiring high performance and low cost solution.

The MCU core is ELAN's second generation RISC based, or RISCII (RII) MCU. The core is designed for low power and portable devices. The device supports Fast mode, Slow mode, Idle mode and Sleep mode for low power applications.

# 2 Features

- MCU Features
  - Operating voltage: 2.2V~3.6V
  - Maximum Operating Speed: 10 MHz clock
  - One Instruction cycle time = 2 × System clock time
  - 8-bit CPU with 80 instructions
  - 48K words program ROM
  - 512K words data ROM
  - 128 bytes un-banked RAM, 8K bytes banked RAM
  - 128-level RAM stack
  - Enhanced Table Look-up function (one or two cycles)
  - High capacity memory access ability
  - Register-to-Register instruction
  - Compare and Branch in one instruction (two cycles)
  - Single Repeat function
  - 8 × 8 hardware Multiplier
  - Decimal ADD and SUB instruction
  - Full range CALL and JUMP capability (two cycles)
- Peripheral Features
  - 16 I/O pins (Port A ~ Port B)
  - One 16 bits general-purpose timer, two 8 bits general-purpose timer, one melody timer
  - EL or IR generator
  - LVD (Low voltage detector)
  - SPI (Serial Peripheral Interface)
  - Clock output for external device (LCD driver, etc.)
  - Key input and strobe function
  - Embedded LCD Driver (1/11, 1/16, 1/24 and 1/32 selectable duty)
- Internal Specifications
  - Watchdog Timer with its own on-chip RC oscillator
  - CPU mode: Sleep Mode, Idle Mode, Slow Mode and Fast Mode
  - Adjustable PLL frequency
- Package Type:
  - 208-pin QFP
  - 87-pin chip form

# 3 Applications

- Educational Learning Aids
- Dictionary, Data Bank
- LCD game

# 4 Block Diagram

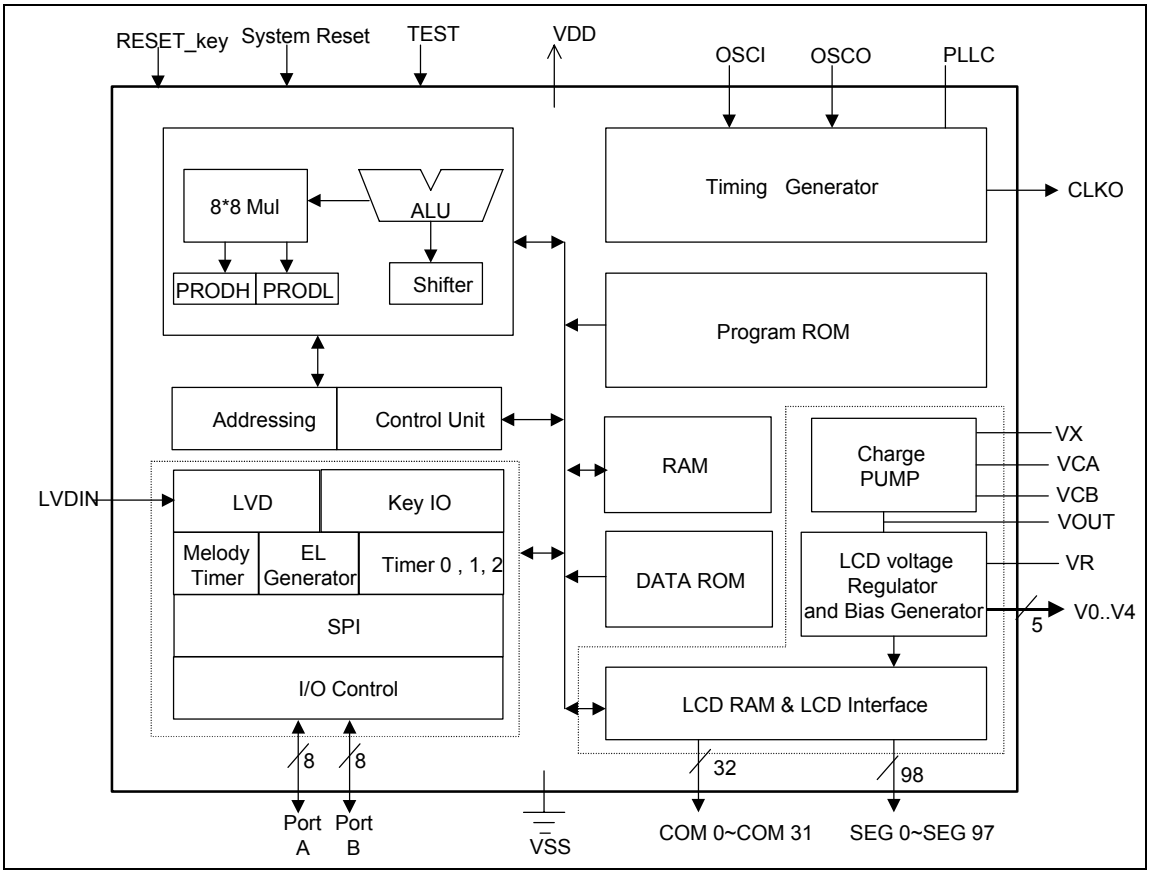
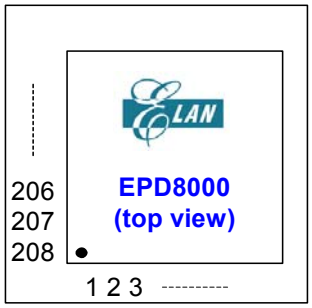



*Fig. 4-1 Functional Block Diagram*

# 4 Pin Assignment




| No. | Pin Name | No. | Pin Name | No. | Pin Name | No. | Pin Name |
|---|---|---|---|---|---|---|---|
| 1 | COM31 | 53 | SEG61 | 105 | SEG29 | 157 | PA.6 |
| 2 | COM30 | 54 | SEG60 | 106 | SEG28 | 158 | PA.7 (ON_key) |
| 3 | COM29 | 55 | SEG59 | 107 | SEG27 | 159 | CLKO |
| 4 | COM28 | 56 | SEG58 | 108 | SEG26 | 160 | PB.0 (/SPISS) |
| 5 | COM27 | 57 | SEG57 | 109 | SEG25 | 161 | PB.1 (SPISCK) |
| 6 | COM26 | 58 | SEG56 | 110 | SEG24 | 162 | PB.2 (SPISDO) |
| 7 | COM25 | 59 | SEG55 | 111 | SEG23 | 163 | PB.3 (SPISDI) |
| 8 | COM24 | 60 | SEG54 | 112 | SEG22 | 164 | PB.4 (CK) |
| 9 | COM23 | 61 | SEG53 | 113 | SEG21 | 165 | PB.5 (CHOP) |
| 10 | COM22 | 62 | SEG52 | 114 | SEG20 | 166 | PB.6 (MDO2) |
| 11 | COM21 | 63 | N.C. | 115 | SEG19 | 167 | PB.7 (MDO1) |
| 12 | COM20 | 64 | N.C. | 116 | SEG18 | 168 | Reset_key |
| 13 | COM19 | 65 | N.C. | 117 | SEG17 | 169 | System Reset |
| 14 | COM18 | 66 | N.C. | 118 | SEG16 | 170 | VDD |
| 15 | COM17 | 67 | N.C. | 119 | SEG15 (Strobe 15) | 171 | N.C. |
| 16 | COM16 | 68 | N.C. | 120 | SEG14 (Strobe 14) | 172 | N.C. |
| 17 | SEG97 | 69 | N.C. | 121 | SEG13 (Strobe 13) | 173 | N.C. |
| 18 | SEG96 | 70 | SEG51 | 122 | SEG12 (Strobe 12) | 174 | N.C. |
| 19 | SEG95 | 71 | SEG50 | 123 | SEG11 (Strobe 11) | 175 | LVDIN |
| 20 | SEG94 | 72 | SEG49 | 124 | SEG10 (Strobe 10) | 176 | Reserved |
| 21 | SEG93 | 73 | SEG48 | 125 | SEG9 (Strobe 9) | 177 | Reserved |
| 22 | SEG92 | 74 | SEG47 | 126 | SEG8 (Strobe 8) | 178 | Reserved |
| 23 | SEG91 | 75 | SEG46 | 127 | SEG7 (Strobe 7) | 179 | Reserved |
| 24 | SEG90 | 76 | SEG45 | 128 | SEG6 (Strobe 6) | 180 | Reserved |
| 25 | SEG89 | 77 | SEG44 | 129 | SEG5 (Strobe 5) | 181 | N.C. |
| 26 | SEG88 | 78 | SEG43 | 130 | SEG4 (Strobe 4) | 182 | N.C. |
| 27 | SEG87 | 79 | SEG42 | 131 | SEG3 (Strobe 3) | 183 | N.C. |
| 28 | SEG86 | 80 | SEG41 | 132 | SEG2 (Strobe 2) | 184 | N.C. |
| 29 | SEG85 | 81 | SEG40 | 133 | SEG1 (Strobe 1) | 185 | N.C. |
| 30 | SEG84 | 82 | SEG39 | 134 | SEG0 (Strobe 0) | 186 | PLLC |

| No. | Pin Name | No. | Pin Name | No. | Pin Name | No. | Pin Name |
|---|---|---|---|---|---|---|---|
| 31 | SEG83 | 83 | SEG38 | 135 | COM0 | 187 | GND |
| 32 | SEG82 | 84 | SEG37 | 136 | COM1 | 188 | OSCO |
| 33 | SEG81 | 85 | SEG36 | 137 | COM2 | 189 | OSCI |
| 34 | SEG80 | 86 | SEG35 | 138 | COM3 | 190 | TEST |
| 35 | SEG79 | 87 | SEG34 | 139 | COM4 | 191 | N.C. |
| 36 | SEG78 | 88 | SEG33 | 140 | COM5 | 192 | N.C. |
| 37 | SEG77 | 89 | SEG32 | 141 | COM6 | 193 | N.C. |
| 38 | SEG76 | 90 | SEG31 | 142 | COM7 | 194 | N.C. |
| 39 | SEG75 | 91 | SEG30 | 143 | COM8 | 195 | N.C. |
| 40 | SEG74 | 92 | N.C. | 144 | COM9 | 196 | N.C. |
| 41 | SEG73 | 93 | N.C. | 145 | COM10 | 197 | N.C. |
| 42 | SEG72 | 94 | N.C. | 146 | COM11 | 198 | N.C. |
| 43 | SEG71 | 95 | N.C. | 147 | COM12 | 199 | VCA |
| 44 | SEG70 | 96 | N.C. | 148 | COM13 | 200 | VX |
| 45 | SEG69 | 97 | N.C. | 149 | COM14 | 201 | V4 |
| 46 | SEG68 | 98 | N.C. | 150 | COM15 | 202 | VR |
| 47 | SEG67 | 99 | N.C. | 151 | PA.0 | 203 | V0 |
| 48 | SEG66 | 100 | N.C. | 152 | PA.1 | 204 | Vout |
| 49 | SEG65 | 101 | N.C. | 153 | PA.2 | 205 | V1 |
| 50 | SEG64 | 102 | N.C. | 154 | PA.3 | 206 | V2 |
| 51 | SEG63 | 103 | N.C. | 155 | PA.4 | 207 | V3 |
| 52 | SEG62 | 104 | N.C. | 156 | PA.5 | 208 | VCB |

**Note:** Pads 176~180 are for testing only. Do Not bond these pads on the application circuit.

# 5 Pin Description

| Name | I/O/P Type | Description |
|---|---|---|
| VDD<br>VSS | P | Digital power supply, the range is from 2.2V to 3.6V. Connect to VSS through the capacitors (0.1μF). |
| TEST | I | Normally connect to VSS, reserved for testing. |
| CLKO | O | Clock output pin |
| LVDIN | I | Low voltage detection input pin |
| System Reset<br>Reset Key | I<br>I | System reset input with built-in pull up resistor (100 KΩ Typical).<br>Low voltage related reset pin<br>Low: RESET asserted<br>High: RESET released |
| OSCI/RC<br>OSCO | I<br>O | RC or Crystal selection by Code Option.<br>32768Hz oscillator pins. Connect to VSS through the capacitors (20 pF).<br>RC oscillator connect pin. Connect to VDD through a resistor (2 MΩ). |
| PLLC | I | PLL capacitor connecting pin. Connect to VSS through the capacitors (0.047 μF). |
| VCA, VCB | - | Charge Pump capacitor. Connect VCA to 0.1μF Cap. and to VCB |

| Name | I/O/P Type | Description |
|---|---|---|
| Vout | - | Charge pump output voltage. Ext. C (0.22μF) to VSS. |
| VX | - | Clamping circuit output voltage. Ext. C (0.1μF) to VSS. |
| VR | - | V0 voltage adjustment pin |
| V4…V0 | O | LCD bias pin. Ext. C (0.1μF) to VSS. |
| COM0 ~COM31 | O | LCD common signal |
| SEG0 ~ SEG97 | O | LCD segment signal<br>(SEG0~SEG15 are shared with Key strobe) |
| Port A | I<br>I<br>I | General Input port<br>Bit 7: On key input, Input port interrupt and wake-up pin<br>Bits 6~0: Special function are Wake-up, Interrupt and Key matrix input pin |
| Port B | I/O<br>O<br>O<br>O<br>O<br>I<br>O<br>I/O<br>I | General Input/Input port<br>Bit 7: MDO1: Melody positive output<br>Bit 6: MDO2: Melody negative output<br>Bit 5: EL CHOP output pin<br>Bit 4: EL CK output pin<br>Bit 3: Serial data input pin<br>Bit 2: Serial data output pin<br>Bit 1: Serial clock input/output pin<br>Bit 0: /Slave Select pin |

# 6 Code Option

- Oscillator (OSCSEL): Select “RC” oscillator or “Crystal” oscillator
- Initial mode after reset: Select “Slow” mode or “Fast” mode
- EL Output Timing

  Select “CHOP output from carrier gating with 128Hz and CKP” or

  “CHOP output directly from carrier” (for IR function)
- LCD Duty Ratio: Maximum duty ratio option
- Select “1/11 duty” or “1/16 duty” or “1/24 duty” or “1/32 duty”
- PLL stable accuracy select bit
- Select “PS bit will be set when PLL frequency is within target ± 10%” or
- “PS bit will be set when PLL frequency is within target ± 3%”
- EXTVH: Vout is from an external power source or internal charge pump selection
- Select “Internal” or “External”
- V1; V2; V3 and V4 OP buffer

  Select “Source, Sink, Source and Sink” or “For auto key scan” or

  “Class A/B” or “OFF” or “V1, V4: auto key scan, V2, V3: OFF”

| | Small Current | Normal Current | Large Current | No Current |
|---|---|---|---|---|
| V1 | Source | * | Class A/B | OFF |
| V2 | Sink | * | Class A/B | OFF |
| V3 | Source | * | Class A/B | OFF |
| V4 | Sink | * | Class A/B | OFF |

* When in normal display : V1 = Source V2 = Sink
V3 = Source V4 = Sink

When in auto key scan: Every time, during a 30 μs strobe start, the OP Amp. will change to class A/B for 60 μs, then return to normal display.

# 7 Function Description

## 7.1 Reset Function:

**A reset can be caused by:**

- Power-on voltage detector reset and power-on reset
- WDT timeout
- System Reset or Reset Key pin pull low

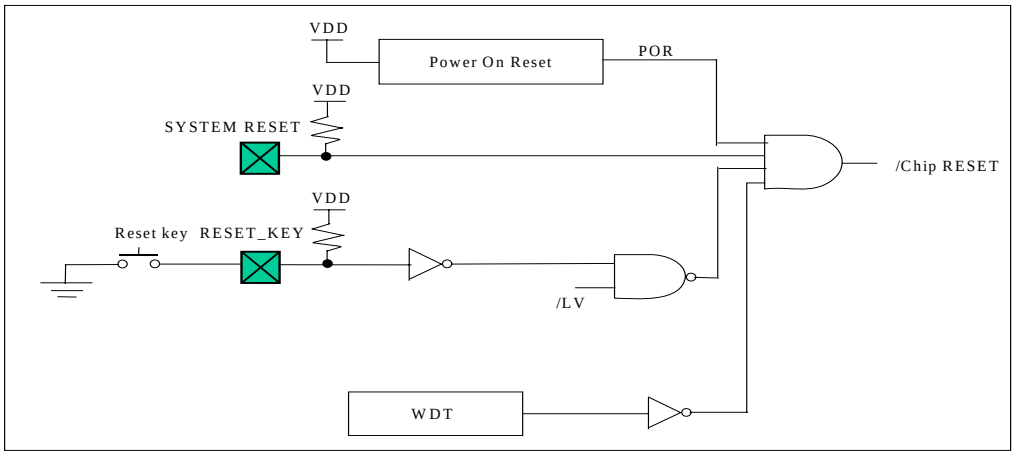



*Figure 7-1 On-chip Reset Circuit*

### 7.1.1 Power-up and Reset Timing

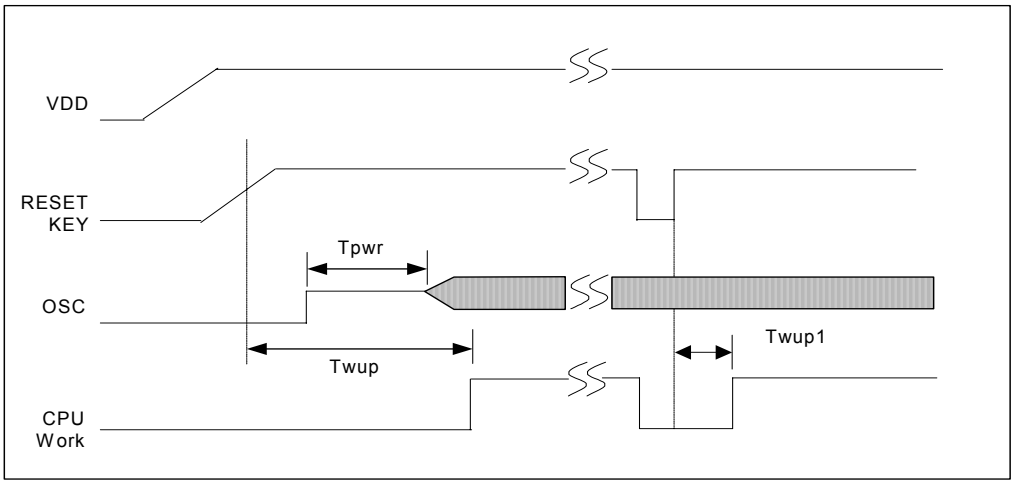



*Figure 7-2 Power-up and Reset Timing*

| Symbol | Characteristics | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|
| Tpwr | Oscillator start up time | 100 | 226 | 300 | ms |
| Twup | CPU warm up time | 260 | 340 | 550 | ms |
| Twup1 | CPU reset time | 18 | 22 | 44 | ms |

■ **Status (R0Fh)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| /TO | /PD | SGE | SLE | OV | Z | DC | C |

**Bit 0 (C):** Carry flag or inverse of Borrow flag (B)
When in SUB operation, borrow flag is indicated by the inverse of carry bit (B = /C)

**Bit 1 (DC):** Auxiliary carry flag

**Bit 2 (Z):** Zero flag

**Bit 3 (OV):** Overflow flag. Use in signed operation when Bit 6 carry into or borrow from a signed bit (Bit 7).

**Bit 4 (SLE):** Computation result is less than or equal to zero (Negative value) after a signed arithmetic. It is only affected by a HEX arithmetic instruction.

**Bit 5 (SGE):** Computation result is greater than or equal to zero (Positive value) after a signed arithmetic. It is only affected by a HEX arithmetic instruction.

**NOTE**

1. *When OV=1 after a signed arithmetic, user can check the SGE and SLE bits to determine whether an overflow (carry into a signed bit) or underflow (borrow from a signed bit) occurs.*
   *OV=1 and SGE=1 → overflow occurs*
   *OV=1 and SLE=1 → underflow occurs*
2. *When overflow occurs, user should clear the MSB of the Accumulator in order to get the correct value.*
   *When underflow occurs, user should set the MSB of the Accumulator in order to get the correct value.*

**Example 1**. ADD a positive value to another positive value, and ACC signed bit will be affected.

```
MOV    ACC, #60h   ; Signed number +60h.
ADD    ACC, #70h   ; +60h ADD WITH +70h.
```

**After instruction:** ACC = 0D0h

SGE=1, means the result is greater than or equal to 0 (positive value)
OV=1, means the result is carry into a signed bit (Bit 7), overflow occurs.

**Correct the signed bit:** ACC = 50h (Clear the signed bit)

The actual result= +80h (OV=1) + 50h = +0D0h

**Example 2**. SUB a positive value from a negative value, and ACC signed bit will be affected.

```
MOV    ACC, #50h   ; Signed number +50h.
SUB    ACC, #90h   ; +50h SUB from –70h (Signed number of 90h)
```

**After instruction:** ACC = 40h

SLE=1, means the result is less than or equal to 0 (negative value)
OV=1, means the result is borrow from a signed bit (Bit 7), underflow occurs.

**Correct the signed bit:** ACC = 0C0h (Set the signed bit)

The actual result = –80h (OV=1) + 0C0h (signed number of 0C0h) = 40h

**Bit 6 (/PD):** Reset to 0 when entering Sleep mode. Set to 1 by "WDTC" instruction, power-on reset or during a Reset pin low condition.

**Bit 7 (/TO):** Reset to 0 during WDT time out reset. Set to 1 by "WDTC" instruction, entering Sleep Mode, power-on reset or during a Reset pin low condition.

When a reset occurs, the special function register will be reset to its initial value except for the /TO and /PD bits of the Status register.

| Bit 7 (/TO) | Bit 6 (/PD) | Event |
|---|---|---|
| 0 | 0 | WDT time out reset from Sleep mode |
| 0 | 1 | WDT time out reset (not Sleep mode) |
| 1 | 0 | Reserved |
| 1 | 1 | Power up or RSTB pin low condition |

## 7.1.2 Register Initial Values

### Special Register

| Addr. | Name | Initial Value | Addr. | Name | Initial Value |
|---|---|---|---|---|---|
| 00h | INDF0 | ---- ----[1] | 10h | LCDDATA | ---- ----[1] |
| 01h | FSR0 | 0000 0000 | 11h | TRL2 | uuuu uuuu |
| 02h | PCL | 0000 0000 | 12h | PRODL | uuuu uuuu |
| 03h | PCM | 0000 0000 | 13h | PRODH | uuuu uuuu |
| 04h | PCH | ---- ---- | 14h | SPRL | xxxx xxxx |
| 05h | BSR | --00 0000 | 15h | SPRM | xxxx xxxx |
| 06h | STKPTR | 0000 0000 | 16h | SPRH | xxxx xxxx |
| 07h | BSR1 | --00 0000 | 17h | Reserved | ---- ---- |
| 08h | INDF1 | ---- ----[1] | 18h | Reserved | ---- ---- |
| 09h | FSR1 | 1000 0000 | 19h | Port A | xxxx xxxx |
| 0Ah | ACC | xxxx xxxx | 1Ah | Port B | xxxx xxxx |
| 0Bh | TABPTRL | 0000 0000 | 1Bh | Reserved | ---- ---- |
| 0Ch | TABPTRM | 0000 0000 | 1Ch | Reserved | ---- ---- |
| 0Dh | TABPTRH | 0000 0000 | 1Dh | Reserved | ---- ---- |
| 0Eh | CPUCON | 000x 0x0c[2] | 1Eh | Reserved | ---- ---- |
| 0Fh | STATUS | cuxx xxxx[3] | 1Fh | Reserved | ---- ---- |

### Control Register

| Addr. | Name | Initial Value | Addr. | Name | Initial Value |
|---|---|---|---|---|---|
| 20h | PFS | 0010 0000 | 30h | POST_ID | -111 -000 |
| 21h | PACON | 0000 0110 | 31h | TRL0L | uuuu uuuu |
| 22h | STBCON | -000 0000 | 32h | TRL0H | uuuu uuuu |
| 23h | DCRB | 1111 1111 | 33h | TRL1 | uuuu uuuu |
| 24h | Reserved | ---- ---- | 34h | TRCON | 0000 0000 |
| 25h | Reserved | ---- ---- | 35h | MTRL | uuuu uuuu |
| 26h | Reserved | ---- ---- | 36h | MWTCON | 0000 0000 |
| 27h | Reserved | ---- ---- | 37h | ELCON | 00uu uuuu |
| 28h | LCDCON | 0000 0000 | 38h | PAINTSTA | 0000 0000 |
| 29h | Reserved | ---- ---- | 39h | PAINTEN | 0000 0000 |
| 2Ah | Reserved | ---- ---- | 3Ah | INTSTA | ---0 0000 |
| 2Bh | SPICON | 0000 0000 | 3Bh | INTCON | ---0 0000 |
| 2Ch | SPISTA | ---- 0000 | 3Ch | Reserved | ---- ---- |
| 2Dh | PAWAKE | 0000 0000 | 3Dh | Reserved | ---- ---- |
| 2Eh | LCDARL | 0000 0000 | 3Eh | Reserved | ---- ---- |
| 2Fh | LCDARH | ---- --00 | 3Fh | PBCON | 0000 0000 |

**Legend:** **"x"** = unknown **"–"** = unimplemented, read as "0"

**"u"** = unchanged **"c"** = value depending on the condition

[1] Not a physical register

[2] Bit 0 (MS0) of RE (CPUCON) is reloaded from "**INIM**" bit of code option when the MCU is reset

[3] If it is a power-on reset or RSTB pin is at low condition, the /**TO** bit and /**PD** bit of RF (STATUS) are set to "1". If it is a WDT time out reset, the /**TO** bit is cleared and /**PD** bit is unchanged.

## 7.2 Oscillator System

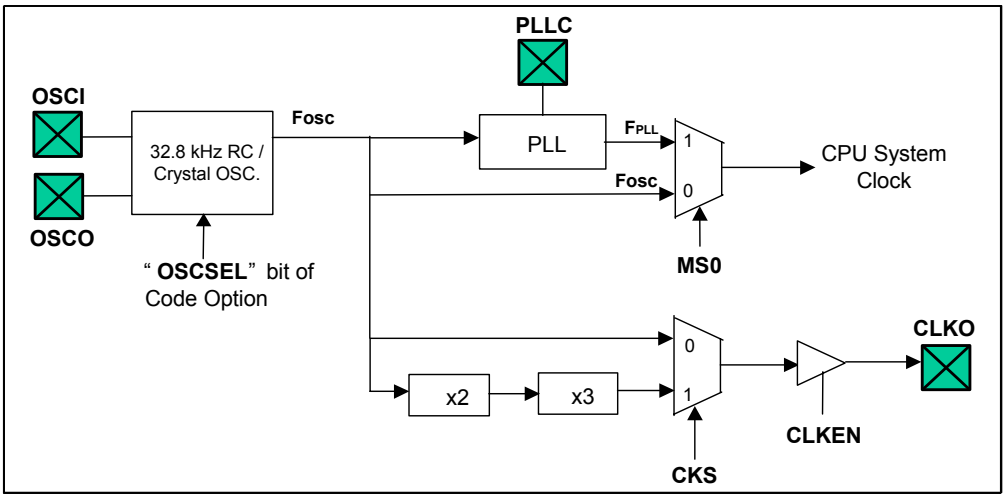



*Figure 7-3 Oscillator System Function Block Diagram*

### *7.2.1 32.768kHz Crystal or 32.8kHz RC*

For the 32.8kHz RC oscillator, connect a 2MΩ pull-up resistor to OSCI pin and the OSCO pin should be floating.

For the 32.768kHz Crystal oscillator, connect a crystal between OSCI and OSCO pins. Then connect the OSCI and OSCO pins to ground through a 20pF capacitor.

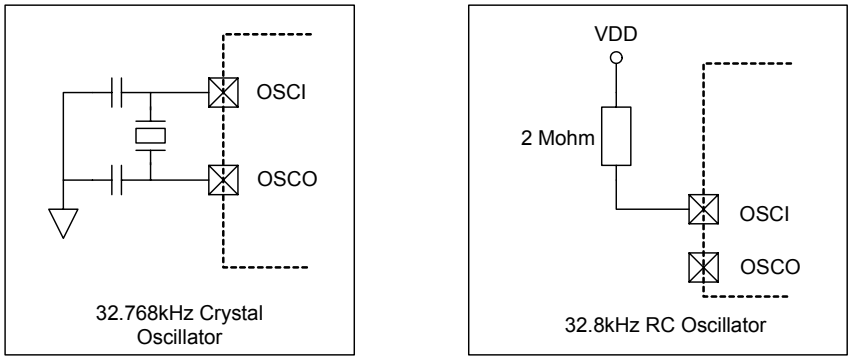



*Figure 7-4 Crystal and RC Oscillator Circuit Diagram*

### 7.2.2 Phase Locked Loop (PLL)

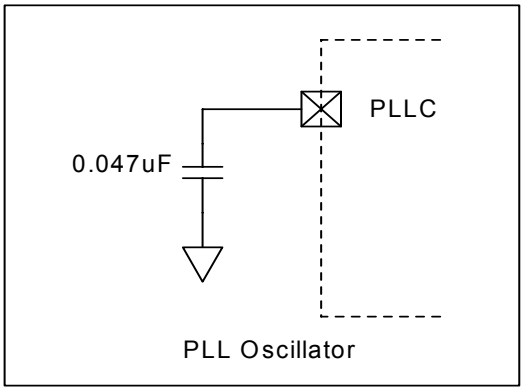



*Figure 7-5 PLL Oscillator Circuit Diagram*

- **PFS (R20h):** Target PLL frequency selects a register. The System clock can be fine tuned from 0.983MHz to 10MHz.
  The initial value of PFS register after a Chip reset will be set to "20h" (FPLL=2.097 MHz)

$$F_{target} = 2 \times PFS \times F_{OSC}$$

| PFS Register | Ftarget (MHz) | PFS Register | Ftarget (MHz) |
|---|---|---|---|
| 0~14 | **N.A.** [1] | 92 | 6.029 |
| | | 107 | 7.012 |
| 15 | 0.983 | 122 | 7.995 |
| 31 | 2.032 | 137 | 8.978 |
| 46 | 3.015 | 150 | 9.83 [2] |
| 61 | 3.998 | 153 | 10.027 |
| 76 | 4.981 | 154 | **N.A.** [1] |

[1] PFS=0~14 and > 154 are not available.

[2] The Maximum range of PLL is from 9.83 MHz ~ 10.027 MHz.

The table is based on 32.768kHz oscillator frequency.

## 7.3 MCU Operation Mode

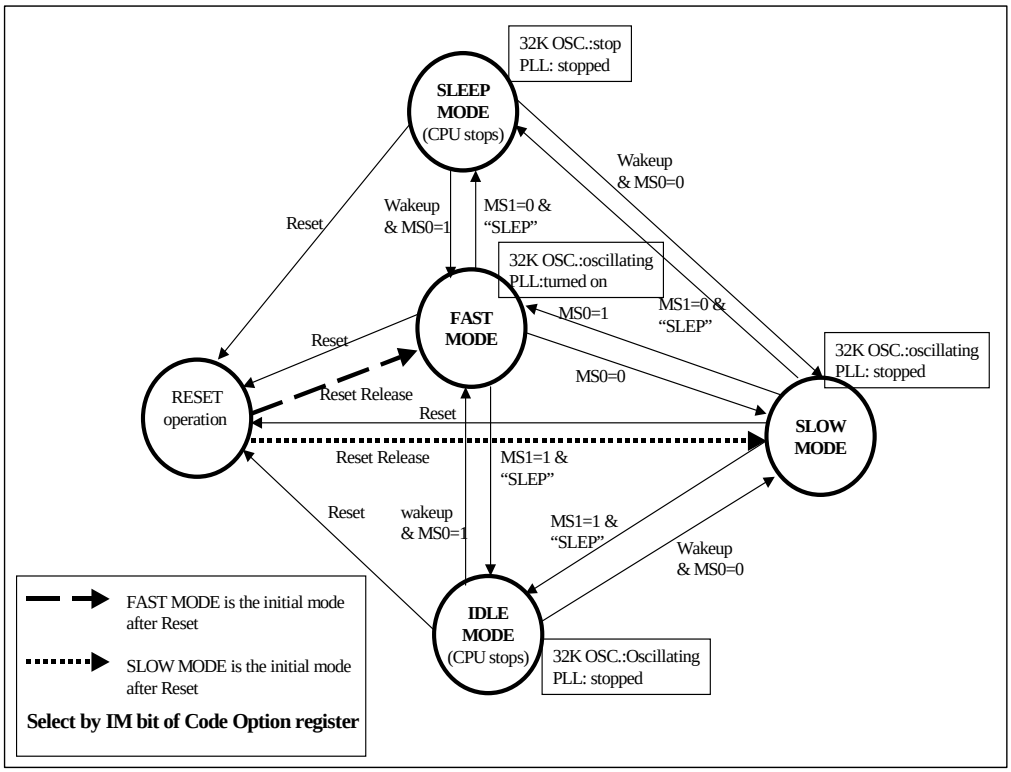



*Figure 7-6 Operation Block Diagram*

### MCU Mode with Function Table

| Mode / Device | SLEEP | IDLE | SLOW | FAST |
|---|---|---|---|---|
| OSC (32768Hz) | × | ✓ | ✓ | ✓ |
| Fsystem | × | × | From OSC | From PLL |
| PLL | × | × | × | ✓ |
| LVD | ✓ | ✓ | ✓ | ✓ |
| Clock output | × | ✓ | ✓ | ✓ |
| Timers 0~2; Melody timer; EL timer | × | ✓ | ✓ | ✓ |
| INT | ×* | ×* | ✓ | ✓ |
| SPI | ✓ (slave) | ✓ (slave) | ✓ | ✓ |

**Legend:** " ✓ " = function is available if enabled " × " = function is Not available

* Interrupt flag will be recorded but not executed until the MCU wakes up.

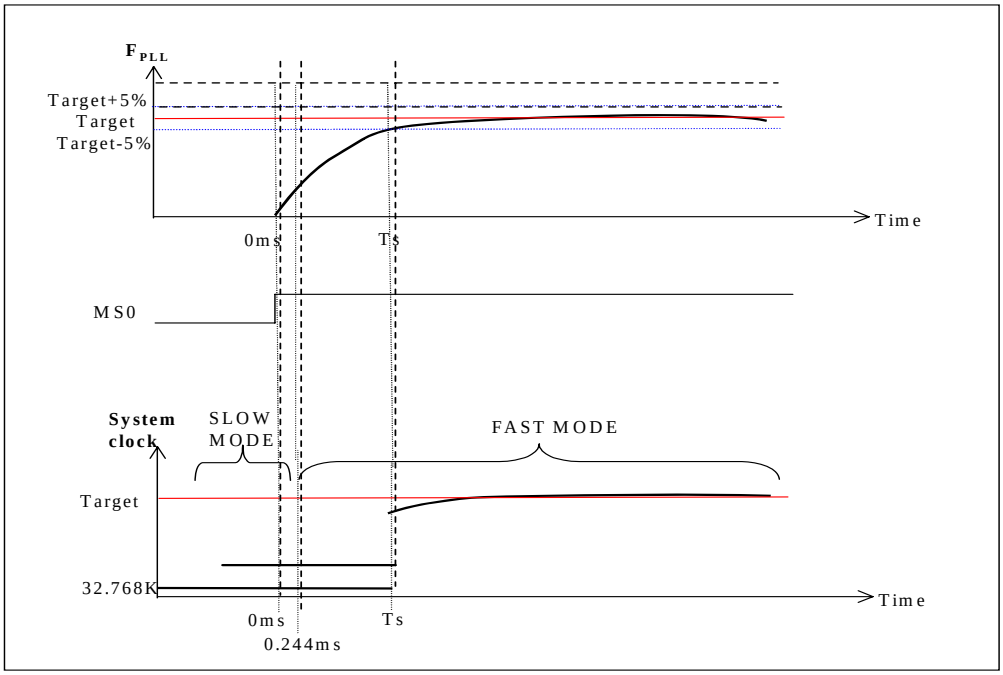



*Figure 7-7 MCU Operation Timing Diagram*

**NOTE**

1. *Switch from Slow mode to Fast mode at Time=0ms*
2. *The System clock will switch to FPLL after 8 oscillation clocks, and the system clock will then increase to about hundreds of kHz.*
3. *The PLL frequency will be stable (±5%) at Time=Ts (2ms~5ms).*

■ **CPUCON (R0Eh):** CPU Control Register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| CLKOEN | CKS | LVDEN | /LV | /GLINTD | PS | MS1 | MS0 |

**Sleep Mode:** When **MS1** bit is set to '0' and "SLEP" instruction is executed, the MCU will enter Sleep mode.

**Idle Mode:** When **MS1** bit is set to '1' and "SLEP" instruction is executed, the MCU will enter IDLE mode.

**Slow Mode:** When **MS0** bit is set to '0', the MCU will enter Slow mode.

**Fast Mode:** When **MS0** bit is set to '1', the MCU will enter Fast mode.

**Bit 2 (PS):** PLL stable flag. This bit will be cleared when PLL is not turned on or PLL is turned on but not stable. It will be set after PLL is turned on and the frequency is stable. PLL stable means that the actual frequency is within the target frequency ± 3%.

"**0**" : PLL is not turned on or PLL frequency is not stable yet
"**1**" : PLL is turned on and frequency is stable

**Bit 6 (CKS):** Clock select bit of clock output pin (CLKO)

"**0**" : CLKO pin output frequency at 32.768kHz (Fosc)
"**1**" : CLKO pin output frequency at 21.845kHz (Fosc/1.5)

**Bit 7 (CLKOEN):** Clock Output Enable control bit

CLKO pin is tri-state when clock output is disabled (CLKOEN=0).

## 7.4 Wake-up Function

| Mode / Device | Sleep | Idle | Slow | Fast |
|---|---|---|---|---|
| I/O wake up | ✓ | ✓ | × | × |
| Timer 1 wake up | × | ✓ | × | × |
| SPI wake up | ✓ (Slave) | ✓(Slave) | × | × |

**Legend:** ✓ = Function is available if enabled × = Function is NOT available

**Code Example:**

```
Entry FAST mode
    MOV     A,#122      ;8MHz
    MOV     PFS,A
    BS      CPUCON,MS0
Entry SLOW mode
    BC      CPUCON,MS0
```

```
Entry IDLE mode
    BS      CPUCON,MS1
    SLEP
    NOP
Entry SLEEP mode
    BC      CPUCON, MS1
    SLEP
    NOP
```

## 7.5 Interrupt

■ **CPUCON (R0Eh):** CPU Control Register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| CLKOEN | CKS | LVDEN | /LV | /GLINTD | PS | MS1 | MS0 |

**Bit 3 (/GLINTD):** Global interrupt disable bit

**"0"** : Disable all interrupts

**"1"** : Enable all un-mask interrupts

**NOTE**

*Give the command "BC CPUCON, GLINTD" twice continuously if all interrupt are to be disabled.*

When an interrupt occurs, the **/GLINTD** bit of the CPUCON register is reset to 0, which disables all interrupts, including Level 1 ~ Level 3. Setting this bit to 1 can enable all un-masked interrupts.

| Interrupt Level | Interrupt Source | Start Address | Remark |
|---|---|---|---|
| – | RESET | 0x00000 | – |
| Level 1 | Input Port | 0x00002 | PAINT |
| Level 2 | Timers 0~2 | 0x00004 | TMR0I, TMR1I, TMR2I |
| Level 3 | Peripheral | 0x00006 | SRBFI, LVDI |

**Code Example:**

```
; ***** Reset program
ResetSEG   CSEG      0X00
   LJMP    MSTART  ;(0X00) Initialize
   LJMP    INPTINT ;(0X02) Input Port
                     A Interrupt
   LJMP    TIMERINT;(0X04) Timer-
                     0,1,2 Interrupt
   LJMP    PERIPH  ;(0X06) Peripheral
                     Interrupt
```

```
; --- Push interrupt register
PUSH:
    MOVPR    AccBuf,ACC
    MOVPR    StatusBuf,Status
    RET

; --- Pop interrupt register
POP:
    MOVRP    Status,StatusBuf
    MOVRP    ACC,AccBuf
    RETI
```

### *7.5.1 Input Port A Interrupt*

Port A Interrupt (Falling edge trigger): Port A is used as external interrupt/wake-up input.

**Code Example:**

```
;=== Input Port And Touch Panel Interrupt
INPTINT:
     S0CALL   PUSH
     JBC      STATUS,F_Z,toPAINT
     SJMP     POP
```

```
;--- Port A interrupt
toPAINT:
     CLR      PAINTSTA
      :
     SJMP     POP
```

### 7.5.2 Timer 0, Timer 1, and Timer 2 Interrupts

1. Timer 0 Interrupt: Timer 0 is a 16-bit timer for general time counting. When the counting value is larger than TRL0H : TRL0L value, a Timer 0 interrupt occurs.
2. Timer 1 Interrupt: Timer 1 is an 8 bit-timer for time counting and wake-up function. When the counting value of Timer 1 underflows, an interrupt occurs and the TRL1 value will be reloaded to counting value.
3. Timer 2 Interrupt: Timer 2 is an 8-bit timer for time counting. When the counting value of Timer 2 underflows, an interrupt occurs and the TRL2 value will be reloaded to counting value.

**Code Example:**

```
;=== Timer-0,1,2 Interrupt
TIMERINT:
     S0CALLPUSH
     JBS     INTSTA,TMR0I,toTM0INT
     JBS     INTSTA,TMR1I,toTM1INT
     JBS     INTSTA,TMR2I,toTM2INT
     SJMP    POP
```

```
;--- Timer 0 Interrupt
toTM0INT:
     BC      INTSTA,TMR0I
      :
     SJMP    POP
;--- Timer 1 Interrupt
toTM1INT:
     BC      INTSTA,TMR1I
      :
     SJMP    POP
;--- Timer 2 Interrupt
toTM2INT:
     BC      INTSTA,TMR2I
      :
     SJMP    POP
```

### 7.5.3 Peripheral Interrupt

1. SRBFI Interrupt: SPI read buffer full interrupt
2. LVDI Interrupt: When LVD low level is detected (Falling edge)

**Code Example:**

```
;=== Peripheral Interrupt
PERIPH:
     S0CALL  PUSH
     JBS     INTSTA,UERRI,toUERRINT
     JBS     INTSTA,UTXI,toUTXINT
     JBS     INTSTA,URXI,toURXINT
     JBS     SPISTA,SRBFI,toSPINT
     JBS     INTSTA,LVDI,toLVDINT
     SJMP    POP
```

```
;--- LVD interrupt
toLVDINT:
     BC      INTSTA,LVDI
      :
     SJMP    POP

;--- SPI interrupt
toSPINT:
     BC      SPISTA,SRBFI
      :
     SJMP    POP
```

## 7.6 Program ROM Map:

| 8K words × 6 Segments = 48K Words | |
|---|---|
| **Addr.** | **Segment** |
| 0000h<br>\|<br>1FFFh | Segment 0 |
| 2000h<br>\|<br>3FFFh | Segment 1 |
| 4000h<br>\|<br>5FFFh | Segment 2 |
| 6000h<br>\|<br>7FFFh | Segment 3 |
| 8000h<br>\|<br>9FFFh | Segment 4 |
| A000h<br>\|<br>BFDFh | Segment 5 |
| BFE0h -BFFBh | Test Program (28 Words) |
| BFFCh -BFFFh | Code Option |

## 7.7 Data ROM Map

| Maximum size is 512K Words | |
|---|---|
| **Address** | |
| 100000h<br>\|<br>17FFFFh | Data ROM (8M bits) |

## 7.8 RAM Map Register

**RAM Size: 128 Bytes + 64 Banks × 128 Bytes = 8320 Bytes**

✧ **Special and Control Register of RAM:**

**Legend: R** = Readable bit **W** = Writable bit **–** = unimplemented, read as "0"

| Addr. | Register Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | INDF0 | R/W | | | | | | | |
| | | Indirect Addressing Pointer 0 | | | | | | | |
| 1 | FSR0 | R/W | | | | | | | |
| | | File Select Register 0 for INDF0 | | | | | | | |
| 2 | PCL | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| | | PC7 | PC6 | PC5 | PC4 | PC3 | PC2 | PC1 | PC0 |
| 3 | PCM | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| | | PC15 | PC14 | PC13 | PC12 | PC11 | PC10 | PC9 | PC8 |
| 4 | PCH | | | | | | | | |
| | | - | - | - | - | - | - | - | - |
| 5 | BSR | | | R/W | | | | | |
| | | - | - | Bank Select Register for INDF0 & General RAM | | | | | |
| 6 | STKPTR | R/W | | | | | | | |
| | | Stack Pointer | | | | | | | |
| 7 | BSR1 | | | R/W | | | | | |
| | | - | - | Bank Select Register 1 for INDF1 | | | | | |
| 8 | INDF1 | R/W | | | | | | | |
| | | Indirect Addressing Pointer 1 | | | | | | | |
| 9 | FSR1 | R | R/W | | | | | | |
| | | 1 | File Select Register 1 for INDF1 | | | | | | |
| A | ACC | R/W | | | | | | | |
| | | Accumulator | | | | | | | |
| B | TABPTRL | R/W | | | | | | | |
| | | Table Pointer Low | | | | | | | |
| C | TABPTRM | R/W | | | | | | | |
| | | Table Pointer Middle | | | | | | | |
| D | TABPTRH | R/W | | | | | | | |
| | | Table Pointer High | | | | | | | |
| E | CPUCON | R/W | R/W | R/W | R | R/W | R | R/W | R/W |
| | | CLKOEN | CKS | LVDEN | /LV | /GLINTD | PS | MS1 | MS0 |
| F | STATUS | R | R | R/W | R/W | R/W | R/W | R/W | R/W |
| | | /TO | /PD | SGE | SLE | OV | Z | DC | C |
| 10 | LCDDATA | R/W | | | | | | | |
| | | Indirect Register to LCD RAM | | | | | | | |
| 11 | TRL2 | R/W | | | | | | | |
| | | Timer 2 Reload Register | | | | | | | |
| 12 | PRODL | R/W | | | | | | | |
| | | Multiplier Product Low | | | | | | | |
| 13 | PRODH | R/W | | | | | | | |
| | | Multiplier Product High | | | | | | | |
| 14 | SPRL | R/W | | | | | | | |
| | | Shift Register Low Byte of SPI | | | | | | | |
| 15 | SPRM | R/W | | | | | | | |
| | | Shift register middle byte of SPI | | | | | | | |

| Addr | Register Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|
| 16 | SPRH | R/W | | | | | | | |
| | | Shift Register High Byte of SPI | | | | | | | |
| 17 | Reserved | - | | | | | | | |
| 18 | Reserved | - | | | | | | | |
| 19 | Port A | R | R | R | R | R | R | R | R |
| | | A.7 | A.6 | A.5 | A.4 | A.3 | A.2 | A.1 | A.0 |
| 1A | Port B | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| | | B.7 | B.6 | B.5 | B.4 | B.3 | B.2 | B.1 | B.0 |
| 1B | Reserved | - | | | | | | | |
| 1C | Reserved | - | | | | | | | |
| 1D | Reserved | - | | | | | | | |
| 1E | Reserved | - | | | | | | | |
| 1F | Reserved | - | | | | | | | |
| 20 | PFS | R/W | | | | | | | |
| | | Target PLL Frequency Selection Register | | | | | | | |
| 21 | PACON | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| | | LVD2 | LVD1 | LVD0 | LVDSEL | Bit7PU | /R2EN | /R1EN | KE |
| 22 | STBCON | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| | | - | SCAN | BitST | ALL | STB3 | STB2 | STB1 | STB0 |
| 23 | DCRB | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| | | Bit7DC | Bit6DC | Bit5DC | Bit4DC | Bit3DC | Bit2DC | Bit1DC | Bit0DC |
| 24 | Reserved | - | | | | | | | |
| 25 | Reserved | - | | | | | | | |
| 26 | Reserved | - | | | | | | | |
| 27 | Reserved | - | | | | | | | |
| 28 | LCDCON | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| | | BSEL2 | BSEL1 | BSEL0 | ADJ4 | ADJ3 | ADJ2 | ADJ1 | ADJ0 |
| 29 | Reserved | - | | | | | | | |
| 2A | Reserved | - | | | | | | | |
| 2B | SPICON | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| | | TLS1 | TLS0 | BRS2 | BRS1 | BRS0 | EDS | DORD | SE |
| 2C | SPISTA | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R |
| | | LCDM1 | LCDM0 | SFR1 | SFR0 | SPWKEN | SMP | DCOL | RBF |
| 2D | PAWAKE | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| | | WKEN7 | WKEN6 | WKEN5 | WKEN4 | WKEN3 | WKEN2 | WKEN1 | WKEN0 |

| Addr | Register Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|
| 2E | LCDARL | R/W | | | | | | | |
| | | Address Low to LCD RAM | | | | | | | |
| 2F | LCDARH | | | | | | | R/W | R/W |
| | | - | - | - | - | - | - | LCDARH1 | LCDARH0 |
| 30 | POST_ID | | R/W | R/W | R/W | | R/W | R/W | R/W |
| | | - | LCD_ID | FSR1_ID | FSR0_ID | - | LCDPE | FSR1PE | FSR0PE |
| 31 | TRL0L | R/W | | | | | | | |
| | | Timer 0 Reload Low Byte Register | | | | | | | |
| 32 | TRL0H | R/W | | | | | | | |
| | | Timer 0 Reload High Byte Register | | | | | | | |
| 33 | TRL1 | R/W | | | | | | | |
| | | Timer 1 Reload Register | | | | | | | |
| 34 | TRCON | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| | | T1WKEN | T1EN | T1PSR1 | T1PSR0 | T0EN | T0CS | T0PSR1 | T0PSR0 |
| 35 | MTRL | R/W | | | | | | | |
| | | Melody Timer Reload Register | | | | | | | |
| 36 | MWTCON | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| | | T2EN | T2PSR1 | T2PSR0 | WDTEN | WDTPSR1 | WDTPSR0 | MTEN | MDO2EN |
| 37 | ELCON | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| | | ELTEN | CKP | ELTRL5 | ELTRL4 | ELTRL3 | ELTRL2 | ELTRL1 | ELTRL0 |
| 38 | PAINTEN | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| | | PA7IE | PA6IE | PA5IE | PA4IE | PA3IE | PA2IE | PA1IE | PA0IE |
| 39 | PAINTSTA | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| | | PA7I | PA6I | PA5I | PA4I | PA3I | PA2I | PA1I | PA0I |
| 3A | INTSTA | | | | R/W | R/W | R/W | R/W | R/W |
| | | - | - | - | SRBFI | LVDI | TMR2I | TMR1I | TMR0I |
| 3B | INTCON | | | | R/W | R/W | R/W | R/W | R/W |
| | | - | - | - | SRBFIE | LVDIE | TMR2IE | TMR1IE | TMR0IE |
| 3C | Reserved | - | | | | | | | |
| 3D | Reserved | - | | | | | | | |
| 3E | Reserved | - | | | | | | | |
| 3F | PBCON | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| | | Bit7PU | Bit6PU | Bit5PU | Bit4PU | Bit3PU | Bit2PU | Bit1PU | Bit0PU |

- Other un-banked register of RAM

| Addr | Un-banked |
|---|---|
| 40h<br>\|<br>7Fh | General Purpose RAM |

- Banked Register of RAM:(selected by BSR)

| Addr. | Bank 0 | Bank 1 | Bank 2 | Bank 3 | …….. | Bank 63 |
|---|---|---|---|---|---|---|
| 80h<br>\|<br>FFh | General Purpose RAM | General Purpose RAM | General Purpose RAM | General Purpose RAM | …….. | General Purpose RAM |

## 7.9 LCD RAM Map: 392 Bytes

| LCD RAM | LCDARH [1:0] | | | |
|---|---|---|---|---|
| Address | 11 (Page 3) | 10 (Page 2) | 01 (Page 1) | 00 (Page 0) |
| LCDARL | Bit 7 ~ Bit 0 | Bit 7 ~ Bit 0 | Bit 7 ~ Bit 0 | Bit 7 ~ Bit 0 |
| 00h | | | | |
| : | | | | |
| 61h | | | | |

## 7.10 Special Register Description

- **STKPTR (R06h)**

The stack level starts from the bottom going up (in a decreasing order), starting from 0FFh of Bank 63.

Stack is located at Bank 62 and 63 from Address FFh~80h. Initial top position of stack pointer is located at 00h.

Bits 0~6 of STKPTR are used as address pointer from 80h~FFh, Bit 7=1 is used to select Bank 63, Bit 7=0 is used to select Bank 62.

Each INT/CALL will stack two bytes address, total capacity is 128 levels.

- **PCL, PCM PCH (R02h, R03h, R04h): Program Counter Register**

| Bit15 | | | | | | Bit8 | Bit7 | | | | | | Bit0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PCM | | | | | | | PCL | | | | | | |

Generates up to 64K×16 on-chip ROM addresses at the relative programming instruction codes.

“S0CALL” loads the low 12 bits of the PC (4K×16 ROM).

“SCALL” or ”SJUMP” loads the low 13 bits of the PC (8K×16 ROM).

“LCALL” or ”LJUMP” loads the full 18 bits of the PC (64K×16 ROM).

“ADD R2, A” or “ADC R2, A” allows a relative address to be added to the current PC. The carry bit of R2 will automatically carry into PCM.

**Code Example:**

```
START: MOV    A,entry
       MOV    number,a      ;number <-- entry
       LCALL  Indirect_JUMP
AAA:   ......
       .....
```

```
Indirect_JUMP:
     MOV       A,number
     ADD       A,ACC       ;A<-- 2*A
     ADD       PCL,A       ;PCL<-- PCL+A
function_table:
     LJMP  Function_Address_1   ; Number=0
     LJMP  Function_Address_2   ; Number=1
     LJMP  Function_Address_3   ; Number=2
     LJMP  Function_Address_4   ; Number=3
     LJMP  Function_Address_5   ; Number=4
     LJMP  Function_Address_6   ; Number=5
     LJMP  Function_Address_7   ; Number=6
     ....
Function_Address_1:
     ....     ;Function 1 operation

     ....
     RET      ;PC will return to AAA label
```

- **ACC (R0Ah):** Accumulator. Internal data transfer, or instruction operand holding

- **POST_ID (R30h):** Post increase / decrease the control register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| - | LCD_ID | FSR1_ID | FSR0_ID | - | LCDPE | FSR1PE | FSR0PE |

**Bit 0 (FSR0PE):** Enable FSR0 post increase/decrease function, FSR0 will *NOT* carry into or borrow from BSR.

**Bit 1 (FSR1PE):** Enable FSR1 post increase/decrease function, FSR1 will carry into or borrow from BSR1.

**Bit 2 (LCDPE):** Enable LCDARH: LCDARL post increase/decrease function.

**Bit 4 (FSR0_ID):** Set to 1 means auto increase, reset to 0 means auto decrease of FSR0.

**Bit 5 (FSR1_ID):** Set to 1 means auto increase, reset to 0 means auto decrease of FSR1.

**Bit 6 (LCD_ID):** Set to 1 means auto increase, reset to 0 means auto decrease of **LCDARH**: LCDARL.

✧ **Indirect Addressing Pointer 0**

- **BSR (R05h)** determines which bank is active (working bank) among the 64 banks (Bank 0 ~ Bank 63).
- **FSR0 (R01h)** is an Address Register for INDF0. User can select up to 256 bytes (Address: 00 ~ 0FFh).
- **INDF0 (R00h)** is not a physically implemented register.

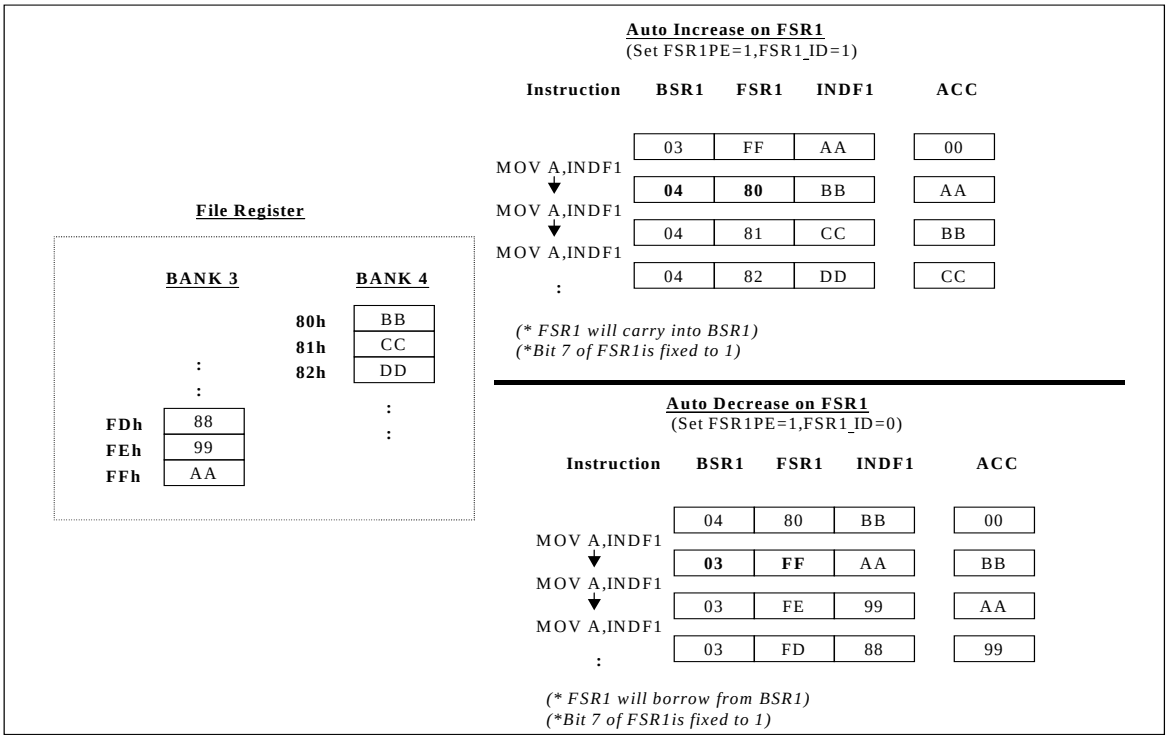
✧ **Indirect Addressing Pointer 1**

- **BSR1 (R07h)** is a Bank Register for INDF1. Cannot determine the working bank for the general register.
- **FSR1 (R09h)** is an Address Register for INDF1. User can select up to 128 bytes (Address: 80 ~ 0FFh); Bit 7 of FSR1 is fixed to 1.
- **INDF1 (R08h)** is not a physically implemented register.

✧ **LCD RAM Addressing Pointer**

- **LCDARH (R2Fh)** is a Page address for the LCD RAM (valid from 00H to 03H).
- **LCDARL (R2Eh)** is a Column address for the LCD RAM (valid from 00H to 61H).
- **LCDDATA (R10h)** is an indirect addressing pointer of the LCD RAM.

Linear addressing capability of INDF1 is shown below:



### Code Example

```
;*********************************       ;*** Main start program
;* Const => Working bank setting         Mstart:
;* REG => Save or Recall register                :
;*********************************                :
;***** RAM stack macro                           IniRAMsk      #29
;*** Initial RAM stack                           :
IniRAMsk  MACRO       #Const                    :
            MOV       A,#Const          MnLoop:
            MOV       BSR1,A                    :
            CLR       FSR1                      :
            BS        POST_ID,FSR1PE            LJMP          MnLoop
            ENDM
;***Push RAM stack                       ;*** Interrupt routine
PushRAM     MACRO     REG               IntSR:
            BS        POST_ID,FSR1_ID           PushRAM       ACC
            MOVRP     INDF1,REG                 PushRAM       Status
            ENDM                                :
;*** Pop RAM stack                              :
PopRAM    MACRO       REG                       :
            BC        POST_ID,FSR1_ID           :
            MOVPR     REG,INDF1                 PopRAM        Status
            ENDM                                PopRAM        ACC
                                                RETI
```

```
Data transform Bank 0 to Bank 1:
       MOV            A,#00110011B          ; Enable FSR0 & FSR1 post increase
       MOV            POST_ID,A
       BANK           #0                    ; BSR = 0 working bank
       MOV            A,#1
       MOV            BSR1,A                ; BSR1 = 1 is Bank 1
       MOV            A,#80H
       MOV            FSR0,A                ; FSR0 = 80H
       CLR            FSR1                  ; FSR1 = 80H
       MOV            A,#80H
       RPT            ACC
       MOVRP          INDF1,INDF0           ; Move 80H ~ 0FFH data to Bank 1
        :
```

■ **TABPTRL, TABPTRM, TABPTRH (R0Bh, R0Ch, R0Dh):** Table Pointer Register

| Bit 23 | | Bit 16 | Bit15 | | Bit 8 | Bit 7 | | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | TABPTRH | | | TABPTRM | | | TABPTRL | |

Program ROM or External memory address register:

**Bit 23** is used to select the internal/external memory. (External memory table look up function is reserved for future models).

**Bit 22 ~ Bit 1** are used to point the memory address.

**Bit 0** is used to select the low byte or high byte of the pointed word (see TBRD instruction).

**Code Example**

```
*** Program ROM
      :
      :
     TBPTH      #(PROMTabB*2)/10000H
     TBPTM      #(PROMTabB*2)/100H
     TBPTL      #PROMTabB*2
      :
      :
     TBRD       0,ACC      ;no change
     TBRD       1,ACC      ;auto-increase
     TBRD       2,ACC      ;auto-decrease
      :
      :
;*** Program ROM data
PROMTabB:
     DB  0x00,0x01,0x02,0x03,0x04,0x05
     DB  0x10,0x11,0x12,0x13,0x14,0x15
     DB  0x20,0x21,0x22,0x23,0x24,0x25
```

```
;*** Internal data ROM
INCLUDE "DROM_I.hdr"   ;to ROMConverter
      :
      :
    TBPTL    #_Data_l
    TBPTM    #_Data_m
    TBPTH    #_Data_h
      :
      :
    TBRD     0,ACC      ;no change
    TBRD     1,ACC      ;auto-increase
    TBRD     2,ACC      ;auto-decrease
    :
```

- **PRODL, PRODH (R12h, R13h):** An unsigned 8×8 hardware multiplier is included in the microcontroller. The result is stored into the 16 bits product register.
- **Port A (R19h):** General Input Register
- **PACON (R21h):** Port A Control Register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| LVD2 | LVD1 | LVD0 | LVDSEL | Bit7PU | /R2EN | /R1EN | KE |

**Bit 0 (KE):** Key input enable/disable control bit
"**0**" : Disable Key input function (Port A register does NOT correspond with Key input in software scan mode).
"**1**" : Enable Key input function (Port A register corresponds with Key input in software scan mode).

**Bit 1 (/R1EN):** R1 pull-up resistor (small resistor) control bit
"**0**" : Enable R1 pull-up resistor
"**1**" : Disable R1 pull-up resistor

**Bit 2 (/R2EN):** R2 pull-up resistor (large resistor) control bit
"**0**" : Enable R2 pull up resistor
"**1**" : Disable R2 pull up resistor

**Bit 3 (Bit 7PU):** Enable Port A.7 pull-up resistor
"**0**" : Disable pull-up resistor
"**1**" : Enable pull up resistor

- **PAINTEN (R38h)** is Port A interrupt control register
"**0**" : Disable interrupt function
"**1**" : Enable interrupt function

- **PAINTSTA (R39h)** is Port A interrupt status register
  Set to "1" when pin falling edge is detected
  Cleared to "0" by software

- **PAWAKE (R2Dh)** is Port A wake-up control register
  "**0**" : Disable wake-up function
  "**1**" : Enable wake-up function

- **Port B (R1Ah)** are General I/O registers

- **DCRB (R23h):** Direction Control of Port B

**Bit 7 ~ Bit 0 (Bit 7DC ~ Bit 0DC):**
"**0**" : Output pin setting
"**1**" : Input pin setting

- **PBCON (R3Fh):** Pull-up Resistor Control of Port B

**Bit 7 ~ Bit 0 (Bit 7PU ~ Bit 0PU):**
"**0**" : Disable pull-up resistor
"**1**" : Enable pull-up resistor

**Code Example**

```
;*** Port A function
;--- Port A interrupt
INPTINT:
     PUSH
     MOV      A,PAINTSTA
;--- Port A interrupt
     JBS      STATUS,F_Z,Q_PAINT
     MOV      PORTB,A
Q_PAINT:
     POP
     RETI
MainSR:
;--- Port B output
     CLR      DCRB
;--- Port A pull-up enable
     BC       PACON,R1EN
     BC       PACON,R2EN
     BS       PACON,Bit7PU
     CLR      PAINTSTA
     MOV      A,#11111111B
;--- Port A interrupt
     MOV      PAINTEN,A
;--- Port A wakeup
     MOV      PAWAKE,A
     BS       CPUCON,GLINTD
```

```
;--- Sleep mode
        BC          CPUCON,MS1
        SLEP
        NOP
         :

;*** Output function => 0XAAh to all port
        CLR         DCRB
        MOV         A,#0XAA
        MOV         PORTB,A
;
;*** Input function => Input all port to RAM 80 ~ 81h
        BC          PACON,REN
        BC          PACON,Bit7PU
        MOV         A,#0XFF
        MOV         DCRB,A
        MOV         PBCON,A
        MOV         DCRB,A
        BS          POST_ID,FSR1_ID
        BS          POST_ID,FSR1PE
        CLR         BSR1
        CLR         FSR1
        MOVRP       INDF1,PORTA
        MOVRP       INDF1,PORTB
```

# 8 Peripheral

## 8.1 Timer 0 (16 Bits):

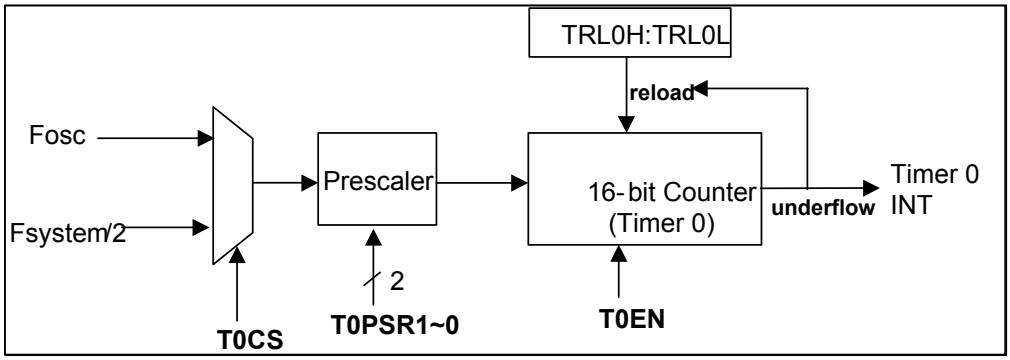



*Figure 8-1 Timer 0 Function Block Diagram*

✧ **Timer 0 Mode**

Timer 0 is a general-purpose 16 bits down counter for some applications requiring time counting. There is an interrupt available for user's application. The clock source is selectable from the oscillator clock or half of the system clock.

There is a prescaler for the timer. The T0PSR1~T0PSR0 bits of TRCON register determine the prescaler ratio and generate different clock rate as the timer clock source. Counter value will be decremented by one (count down) according to the timer clock source. When underflow occurs, the timer interrupt will be triggered if the global interrupt and Timer 0 interrupt are both enabled. At the same time, TRL0H:TRL0L will be automatically reloaded into the 16 bits counter.

$$T = \frac{1}{Fcs} \times \Pr escaler \times (TRL0H : TRL0L + 1)$$

The Timer 0 frequency range is from 1/128Hz (clock source is from Fosc, TRL0H: TRL0L = 0FFFFh, prescaler = 1: 64) to 5 MHz (clock source is from FPLL/2, system clock is 10 MHz, TRL0H: TRL0L = 0000h, prescaler = 1:1).

- **TRL0H, TRL0L (R32h, R31h):** Used to store the auto-reloaded values of Timer 0. When enabled, Timer 0 underflow occurs, TRL0H: TRL0L register will be automatically reloaded into the 16- bit counter.

■ **TRCON (R34h):** Timer 0 and Timer 1 Control Register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| T1WKEN | T1EN | T1PSR1 | T1PSR0 | T0EN | T0CS | T0PSR1 | T0PSR0 |

**Bit 1 ~ Bit 0 (T0PSR1~T0PSR0):** Timer 0 Prescaler select bit

| T0PSR1: T0PSR0 | Prescaler Value |
|---|---|
| 00 | 1:1 |
| 01 | 1:4 |
| 10 | 1:16 |
| 11 | 1:64 |

**Bit 2 (T0CS):** Timer 0 clock source select bit

"**0**" : Clock source is from Fosc

"**1**" : Clock source is from FPLL/2

**Bit 3 (T0EN):** Timer 0 enable control bit

"**0**" : Disable Timer 0

"**1**" : Enable Timer 0

■ **CPUCON (R0Eh):** CPU Control Register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| CLKOEN | CKS | LVDEN | /LV | /GLINTD | PS | MS1 | MS0 |

**Bit 3 (GLINTD):** Global Interrupt Disable Bit

"**0**" : Disable all interrupts

"**1**" : Enable all un-masked interrupts

■ **INTCON (R3Bh):** Interrupt Control Register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| - | - | - | SRBFIE | LVDIE | TMR2IE | TMR1IE | TMR0IE |

**Bit 0 (TMR0IE):** Timer 0 Interrupt Control Bit

"**0**" : Disable Timer 0 interrupt

"**1**" : Enable Timer 0 interrupt

■ **INTSTA (R3Ah):** Interrupt Status Register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| - | - | - | SRBFI | LVDI | TMR2I | TMR1I | TMR0I |

**Bit 0 (TMR0I):** Set to 1 when Timer 0 is larger than TRL0H ~ TRL0L value
Cleared to 0 by software or disable Timer 0

**Code Example:**

```
;=== Timer 0 = (8M/2)/[4 x 3FFF + 1]
Timer0SR:
      :
     System setting 8MHz
     PC.2 setting output port
      :
;--- Fpll & Pre-scale 1:4
     MOV       A,#00000101B
     MOV       TRCON,A
;--- 8ms = (4 x 16383 + 1)/(8M/2)
     MOV       A,#0FFH
     MOV       TRL0L,A
     MOV       A,#03FH
     MOV       TRL0H,A
;--- Timer 0 enable
     BS        TRCON,T0EN
;--- Timer 0 interrupt enable
     BS        INTCON,TMR0IE
;--- Clear Timer 0 interrupt status
     BC        INTSTA,TMR0I
;--- Enable global interrupt
     BS        CPUCON,GLINTD
TimeLoop:
     SJMP      TimeLoop
```

```
;=== Timer 0 interrupt
TIMERINT:
     PUSH
     JBC    INTSTA,TMR0I,Q_Time
     BC     INTSTA,TMR0I
     BTG    PORTC,2
Q_Time:
     POP
     RETI
```

## 8.2 Timer 1 (8 Bits):

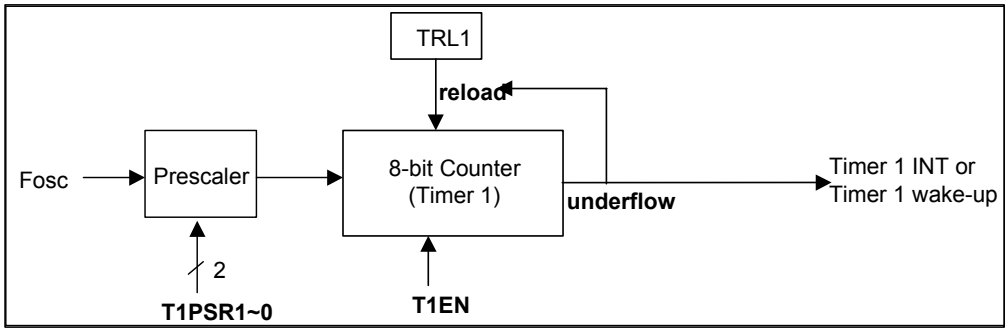



*Figure 8-2 Timer 1 Function Block Diagram*

Timer 1 is a general-purpose 8-bit down counter for some applications requiring time counting. There are interrupt and wake-up function available for user's application. The clock source is from the oscillator clock.

There is a prescaler for the timer. The **T1PSR1~T1PSR0** bits of the TRCON register determine the prescaler ratio and generate different clock rates for the timer clock source. Setting **T1WKEN** bit of the TRCON register to 1 will enable the Timer 1 underflow wake-up function in Idle Mode.

The counting value will be decremented by one (count down) according to the timer clock source. When an underflow occurs, the timer interrupt will be triggered if the global interrupt and Timer 1 interrupt are both enabled. At the same time, the TRL1 value will be automatically reloaded into the 8 bits counter.

$$T = \frac{1}{Focs} \times \Pr escaler \times (TRL1 + 1)$$

The Timer 1 frequency range is from 0.5Hz (TRL1 = 0FFh, prescaler = 1:256) to 8.192kHz (TRL1 = 0h, prescaler = 1:4). The clock source is from the oscillator clock (Fosc).

- **TRL1 (R33h):** Used to store the auto-reload value of Timer 1. When enabling Timer 1 or an underflow occurs, TRL1 register value will be automatically reloaded into the 8 bits counter.
- **TRCON (R34h):** Timer 0 and Timer 1 control register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| T1WKEN | T1EN | T1PSR1 | T1PSR0 | T0EN | T0CS | T0PSR1 | T0PSR0 |

**Bit 5 ~ Bit 4 (T1PSR1~T1PSR0):** Timer 1 Pre-scale Select Bit

| T1PSR1: T1PSR0 | Prescaler Value |
|---|---|
| 00 | 1:4 |
| 01 | 1:16 |
| 10 | 1:64 |
| 11 | 1:256 |

**Bit 6 (T1EN):** Timer 1 Enable Control Bit

**"0"** : Disable Timer 1 (stop counting)

**"1"** : Enable Timer 1

**Bit 7 (T1WKEN):** Enable bit of Timer 1 underflow wake-up function in Idle Mode

**"0"** : Disable Timer 1 wake-up function

**"1"** : Enable Timer 1 wake-up function

- **CPUCON (R0Eh):** CPU Control Register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| CLKOEN | CKS | LVDEN | /LV | /GLINTD | PS | MS1 | MS0 |

**Bit 3 (/GLINTD):** Global Interrupt Disable Bit

**"0"** : Disable all interrupts

**"1"** : Enable all un-mask interrupts

■ **INTCON (R3Bh):** Interrupt Control Register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| - | - | - | SRBFIE | LVDIE | TMR2IE | TMR1IE | TMR0IE |

**Bit 1 (TMR1IE):** Timer 1 Interrupt Control Bit

**"0"** : Disable Timer 1 interrupt

**"1"** : Enable Timer 1 interrupt

■ **INTSTA (R3Ah):** Interrupt Status Register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| - | - | - | SRBFI | LVDI | TMR2I | TMR1I | TMR0I |

**Bit 1 (TMR1I):** Set to 1 when Timer 1 interrupt occurs

Clear to 0 by software or disable Timer 1

**Code Example**

```
;=== Timer 1 interrupt
TIMERINT:
     PUSH
     JBC    INTSTA,TMR1I,Q_Time
     BC     INTSTA,TMR1I
     BTG    PORTC,2
Q_Time:     POP
     RETI
;=== Timer1 = 32.768K / [256 x 3F + 1]
Timer1SR:
      :
     PC.2 setting output port
      :
     MOV    A,#10110000B
     MOV    TRCON,A          ; Fosc & Pre-scale 1:256 & wakeup
     MOV    A,#03FH
     MOV    TRL1,A           ; 0.5sec = (256 x 63 + 1)/32.768K
     BS     TRCON,T1EN       ; Timer 1 enable
     BS     INTCON,TMR1IE    ; Timer 1 interrupt enable
     BC     INTSTA,TMR1I     ; Clear Timer 1 interrupt status
     BS     CPUCON,GLINTD    ; Enable global interrupt
     BS     CPUCON,MS1       ; Idle mode
T1WLoop:
     SLEP
     NOP
      :
     SJMP   T1Wloop
```

## 8.3 Timer 2 (8 Bits)

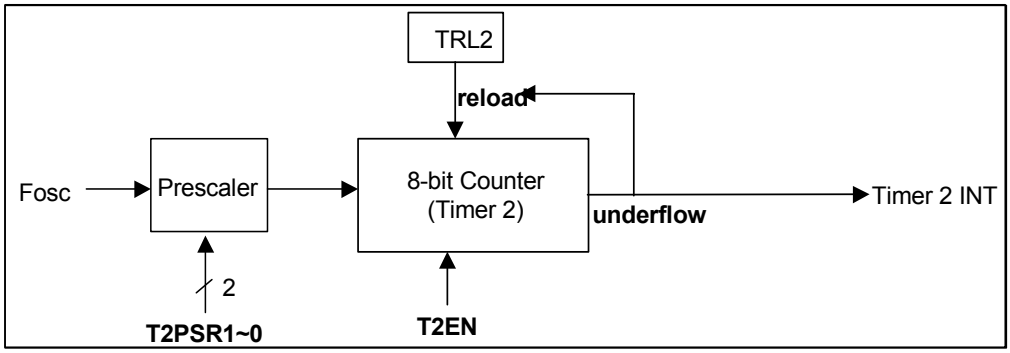



*Figure 8-3 Timer 2 Function Block Diagram*

Timer 2 is a general-purpose 8-bit down counter for some applications requiring time counting. There are interrupt functions available for user's application. The clock source is from the oscillator clock.

There is a prescaler for the timer. The **T2PSR1~T2PSR0** bits of the MWTCON register determine the prescaler ratio and generate different clock rates for the timer clock source.

The counting value will be decremented by one (count down) according to the timer clock source. When the counter value underflows, a timer interrupt will occur (if a global interrupt and a Timer 2 interrupt are both enabled). At the same time, TRL2 will be automatically reloaded into the 8 bits counter.

$$T = \frac{1}{Fosc} \times \Pr escaler \times (TRL2 + 1)$$

The Timer 2 frequency range is from 16Hz (clock source is from Fosc, TRL2 = 0FFh, prescaler = 1:8) to 32.768kHz (clock source is from $F_{OSC}$, TRL2 = 0h, prescaler = 1:1)

- **TRL2 (R11h):** Used to store the auto-reload value of Timer 2. When enabling Timer 2 or an underflow occurs, TRL2 register will automatically be reloaded into the 8 bits counter.
- **MWTCON (R36h):** Melody Timer, WDT and Timer 2 Control Register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| T2EN | T2PSR1 | T2PSR0 | WDTEN | WDTPSR1 | WDTPSR0 | MTEN | MDO2EN |

**Bit 6 ~ Bit 5 (T2PSR1~T2PSR0):** Timer 2 Prescaler Select Bit

| T2PSR1: T2PSR0 | Prescaler Value |
|---|---|
| 00 | 1:1 |
| 01 | 1:2 |
| 10 | 1:4 |
| 11 | 1:8 |

**Bit 7 (T2EN):** Timer 2 Enable Control Bit

"**0**" : Disable Timer 2 (stop counting)

"**1**" : Enable Timer 2

■ **CPUCON (R0Eh):** CPU Control Register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| CLKOEN | CKS | LVDEN | /LV | /GLINTD | PS | MS1 | MS0 |

**Bit 3 (/GLINTD):** Global Interrupt Disable Bit

"**0**" : Disable all interrupts

"**1**" : Enable all un-mask interrupts

■ **INTCON (R3Bh):** Interrupt Control Register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| - | - | - | SRBFIE | LVDIE | TMR2IE | TMR1IE | TMR0IE |

**Bit 2 (TMR2IE):** Timer 2 Interrupt Control bit

"**0**" : Disable Timer 2 interrupt

"**1**" : Enable Timer 2 interrupt

■ **INTSTA (R3Ah):** Interrupt Status Register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| - | - | - | SRBFI | LVDI | TMR2I | TMR1I | TMR0I |

**Bit 2 (TMR2I):** Set to 1 when Timer 2 interrupt occurs

Clear to 0 by software or disable Timer 2

**Code Example**

```
;=== Timer 2 interrupt
TIMERINT:
    PUSH
    JBC     INTSTA,TMR2I,Q_Time
    BC      INTSTA,TMR2I
    BTG     PORTC,2
Q_Time:
    POP
    RETI
;=== Timer 2 = 32.768K / [4 x 255 + 1]
Timer2SR:
     :
    System setting 8MHz
    Port C.2 setting output port
     :
    MOV     A,# 01000000B
    MOV     MWTCON,A          ; Prescaler 1:4
    MOV     A,#0FFH
    MOV     TRL2,A            ; 31.25ms = (4 x 255 + 1)/32.768K
    BS      MWTCON,T2EN       ; Timer 2 enable
    BS      INTCON,TMR2IE     ; Timer 2 interrupt enable
    BC      INTSTA,TMR2I      ; Clear Timer 2 interrupt status
    BS      CPUCON,GLINTD     ; Enable global interrupt
TMR2Loop:
    SJMP    TMR2Loop
```

## 8.4 Melody Timer (8 Bits)

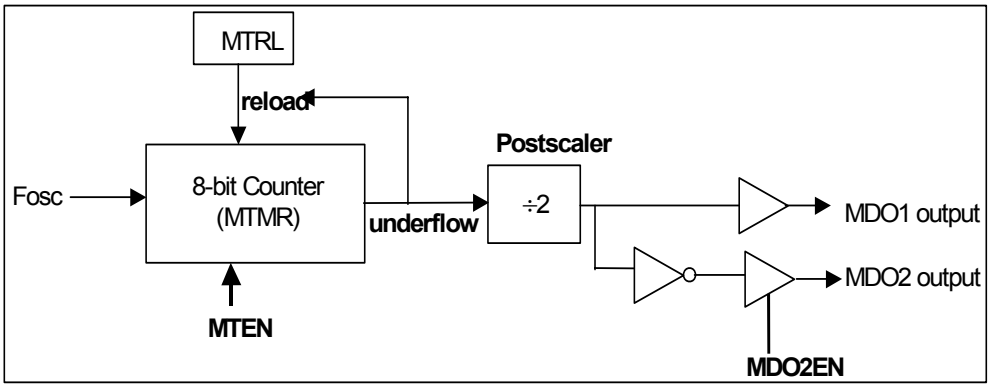



*Figure 8-4 Melody Timer Function Block Diagram*

The Melody Timer is used for counting the tempo and rhythm of music. There are positive/negative outputs available for user's application. The clock source is from the oscillator clock (32.768kHz). When the MTEN is set to 1, the Melody Timer is enabled.

The counting value will be decremented by one (count down) in accordance with the timer clock source. When the Melody Timer is enabled or an underflow occurs, the MTRL will be automatically reloaded into the 8-bit counter. MDO1 and MDO2 pin output level will be toggled simultaneously. MDO2 pin output is the inverse of MDO1 pin.

The melody timer frequency range is from 64 Hz (MTRL = 0FFh, post scalar = 1:2) to 16kHz (MTRL=0h, postscaler = 1:2).

$$F = Fosc \times \frac{1}{(MTRL+1)} \times \frac{1}{2}$$

**MDO1 (Port B.7):** Melody positive output pin

**MDO2 (Port B.6):** Melody negative output pin

- **MTRL (R35h):** Melody Timer Reload Register. It is used to store the auto-reload value of the melody timer. When the Melody Timer is enabled or an underflow occurs, the MTRL will automatically be reloaded into the 8-bit counter.
- **MWTCON (R36h):** Melody Timer, WDT and Timer 2 Control Register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| T2EN | T2PSR1 | T2PSR0 | WDTEN | WDTPSR1 | WDTPSR0 | MTEN | MDO2EN |

**Bit 0 (MDO2EN):** Melody negative output port enable control bit
"**0**" : Disable the melody negative output port
"**1**" : Enable the melody negative output port

**Bit 1 (MTEN):** Melody Timer Enable Control Bit
"**0**" : Disable
"**1**" : Enable

**Melody Table:** Clock source is 32768 Hz

| Musical Note | Standard Frequency (Hz) | MTRL | Postscaler | FMELODY (Hz) | DEV (%) |
|---|---|---|---|---|---|
| L1 | 130.813 | 124 | 1:2 | 131.07 | 0.20% |
| L1# | 138.591 | 117 | 1:2 | 138.85 | 0.18% |
| L2 | 146.832 | 111 | 1:2 | 146.29 | -0.37% |
| L2# | 155.563 | 104 | 1:2 | 156.04 | 0.31% |
| L3 | 164.814 | 98 | 1:2 | 165.49 | 0.41% |
| L4 | 174.614 | 93 | 1:2 | 174.30 | -0.18% |
| L4# | 184.997 | 88 | 1:2 | 184.09 | -0.49% |
| L5 | 195.998 | 83 | 1:2 | 195.05 | -0.48% |
| L5# | 207.652 | 78 | 1:2 | 207.39 | -0.13% |
| L6 | 220.000 | 73 | 1:2 | 221.41 | 0.64% |
| L6# | 233.082 | 69 | 1:2 | 234.06 | 0.42% |
| L7 | 246.942 | 65 | 1:2 | 248.24 | 0.53% |
| M1 | 261.626 | 62 | 1:2 | 260.06 | -0.60% |
| M1# | 277.183 | 58 | 1:2 | 277.69 | 0.18% |
| M2 | 293.665 | 55 | 1:2 | 292.57 | -0.37% |
| M2# | 311.127 | 52 | 1:2 | 309.13 | -0.64% |
| M3 | 329.628 | 49 | 1:2 | 327.68 | -0.59% |
| M4 | 349.228 | 46 | 1:2 | 348.60 | -0.18% |
| M4# | 369.994 | 43 | 1:2 | 372.36 | 0.64% |
| M5 | 391.995 | 41 | 1:2 | 390.10 | -0.48% |
| M5# | 415.305 | 38 | 1:2 | 420.10 | 1.16% |
| M6 | 440.000 | 36 | 1:2 | 442.81 | 0.64% |
| M6# | 466.164 | 34 | 1:2 | 468.11 | 0.42% |
| M7 | 493.883 | 32 | 1:2 | 496.48 | 0.53% |
| H1 | 523.251 | 30 | 1:2 | 528.52 | 1.01% |
| H1# | 554.365 | 29 | 1:2 | 546.13 | -1.48% |
| H2 | 587.330 | 27 | 1:2 | 585.14 | -0.37% |
| H2# | 622.254 | 25 | 1:2 | 630.15 | 1.27% |
| H3 | 659.255 | 24 | 1:2 | 655.36 | -0.59% |
| H4 | 698.456 | 22 | 1:2 | 712.35 | 1.99% |
| H4# | 739.989 | 21 | 1:2 | 744.73 | 0.64% |
| H5 | 783.991 | 20 | 1:2 | 780.19 | -0.48% |
| H5# | 830.609 | 19 | 1:2 | 819.20 | -1.37% |
| H6 | 880.000 | 18 | 1:2 | 862.32 | -2.01% |
| H6# | 932.328 | 17 | 1:2 | 910.22 | -2.37% |
| H7 | 987.767 | 16 | 1:2 | 963.76 | -2.43% |

**Code Example:**

```
;=== Beep tone 1kHz generator
Beep1K:
     :
    System setting 8MHz
     :
    MOV         A,#0FH
    MOV         MTRL,A          ; 1 kHz = 32768 / (15 +1) / 2
    BS          MWTCON,MTEN     ; Melody enable
     :
    LCALL       Dly1ms
     :
    BC          MWTCON,MTEN     ; Melody disable
     :
     :
```

## 8.5 EL or IR Generator Timer (6 Bits)

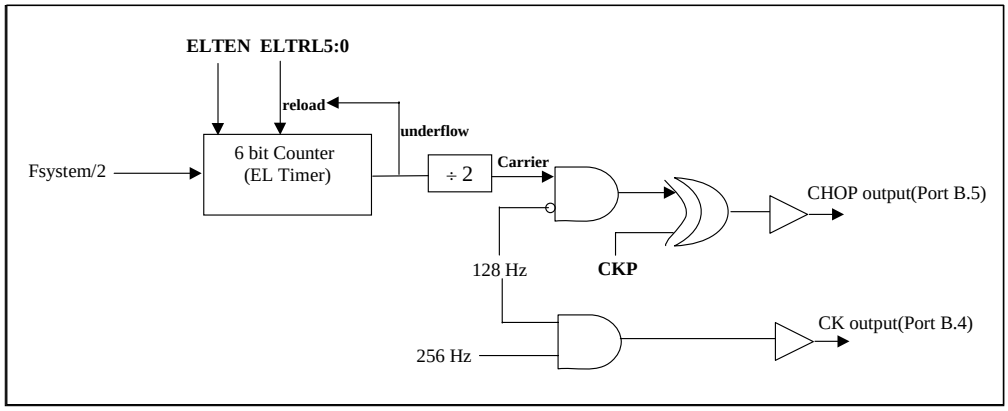



*Figure 8-5 EL Timer or IR Generator Timer Function Block Diagram*

The EL Timer is a 6-bit down counter used for counting the CHOP signal output. The clock source is generated from the system clock divided by 2 (1MHz ~ 10MHz).

The counting value will be decremented by one (count down) in accordance with the timer clock source. When underflow occurs, the CHOP output level will be toggled. When the EL timer is enabled or an underflow occurs, ELTRL will be automatically reloaded into the 6-bit counter.

The **ELTEN** bit is used to enable the EL timer and change Port B.5 and Port B.4 to CHOP and CK output pin.

**CKP** bit is used to select the clock polarity of the CHOP output (See timing diagram for details).

The frequency range of the CHOP carrier signal is 128Hz (System clock at 32768Hz, ELTRL=3Fh) to 2.5MHz (System clock at 10MHz, ELTRL=0h).

$$F_{CHOP} = (F_{SYSTEM} / 2) \times \frac{1}{(ELTRL + 1)} \times \frac{1}{2}$$

**CHOP (Port B.5):** Chop output pin of the EL driver

**CK (Port B.4):** CK output pin of the EL driver

✧ **EL Code Option**

**EL Output Timing**

Select "CHOP output from the carrier gating with 128Hz and CKP" or "CHOP output is directly from the carrier" (for IR function)

■ **ELCON (R37h):** EL Control Register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| ELTEN | CKP | ELTRL5 | ELTRL4 | ELTRL3 | ELTRL2 | ELTRL1 | ELTRL0 |

**Bit 5 ~ Bit 0 (ELTRL5~ELTRL0):** Used to store the auto reload value of EL/IR timer. When EL/IR timer is enabled or an underflow occurs, ELTRL5~0 will automatically be reloaded into the 6-bit counter.

**Bit 6 (CKP):** EL/IR clock polarity select bit

**"0"** : Idle state for CHOP pin is low level

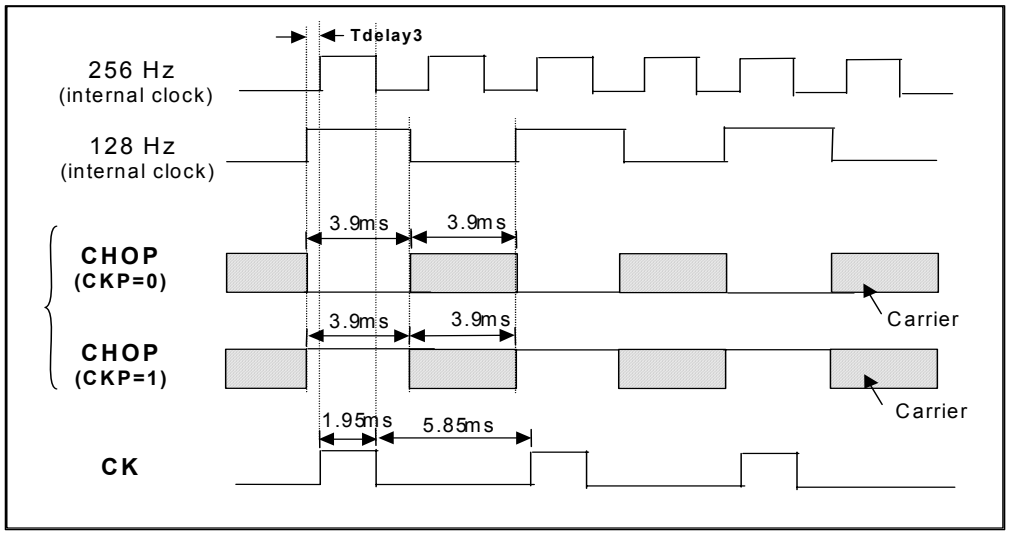
**"1"** : Idle state for CHOP pin is high level

**Bit 7 (ELTEN):** EL/IR Timer enable control bit

**"0"** : Disable EL/IR Timer (stop counting) and recover CK and CHOP pins as general I/O pin.

**"1"** : Enable EL/IR Timer and change Port B.4 and Port B.5 to CK and CHOP output pins.

**EL Generator Timing**



**Code Example**

```
EL code option -- EL Output Timing:
   Select "CHOP output from the carrier gating with 128Hz and CKP"
;=== EL generator 200kHz
    :
   System setting 4MHz
    :
   MOV       A,#00000100B
   MOV       ELCON,A          ; (4MHz / 2) / [ (4 + 1) x 2 ] = 200kHz
   BS        ELCON,ELTEN      ; Enable EL generator
    :
```

```
EL code option -- EL Output Timing:
   Select "CHOP output directly from the carrier"
;=== IR generator 32.25kHz
    :
   System setting 8MHz
    :
   MOV       A,#00111111B
   MOV       ELCON,A         ; (8MHz / 2) / [ (63 + 1) x2 ] = 31.25kHz
   BS        ELCON,ELTEN     ; Enable IR generator
    :
```

## 8.6 Watchdog Timer (WDT)

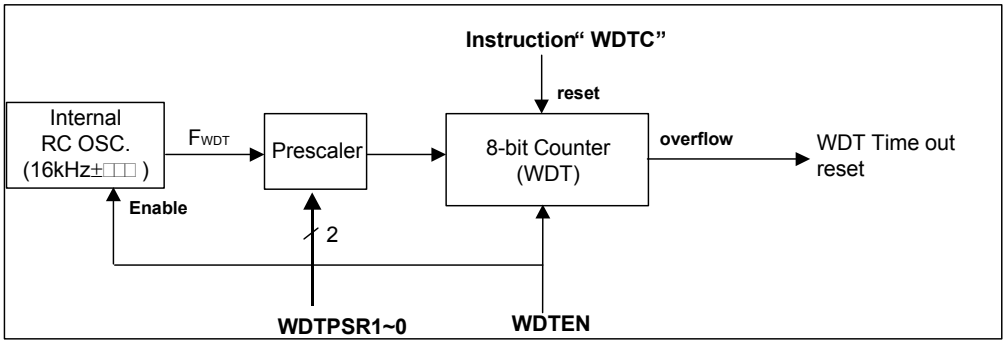



*Figure 8-6 Watchdog Timer Function Block Diagram*

The Watchdog Timer (WDT) clock source is from the on-chip RC oscillator (16kHz ± 20%). The WDT will keep on running even when the oscillator has been turned off (i.e. in Sleep Mode). WDT time-out will cause the MCU to reset (if WDT is enabled). To avoid a reset to occur, user should clear the WDT value by using the "WDTC" instruction before WDT time-out. Setting the WDTEN bit will enable the WDT function. Disable is the WDT default condition. There is also a prescaler to generate different clock rates for the WDT clock source. The prescaler ratio is defined by WDTPSR1 ~ WDTPSR0.

$$T = \frac{1}{F_{WDT}} \times \Pr escaler \times (WDT + 1)$$

The WDT time-out range is 64 ms (prescaler = 1:4) to 2.048 second (prescaler = 1:128).

- **MWTCON (R36h):** Melody Timer, WDT and Timer 2 Control Register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| T2EN | T2PSR1 | T2PSR0 | WDTEN | WDTPSR1 | WDTPSR0 | MTEN | MDO2EN |

**Bit 3 ~ Bit 2 (WDTPSR1~WDTPSR0):** Watchdog Timer Prescaler select bit

| WDTPSR1: WDTPSR0 | Prescaler Value |
|---|---|
| 00 | 1:4 |
| 01 | 1:16 |
| 10 | 1:64 |
| 11 | 1:128 |

**Bit 4 (WDTEN):** Watchdog Timer enable bit

**"0"** : Disable the Watchdog Timer (stop running)

**"1"** : Enable the Watchdog Timer

**Code Example**

```
;=== WDT setting 2.048 sec
    :
   Timer 1 (0.5 sec wake-up)
    :
   MOV     A,#00011100b
   AND     A,MWTCON        ; Prescaler 1:128
   BC      CPUCON,MS1      ; Change to sleep
                           ; mode
WDT_Loop:
   WDTC
   SLEP
   NOP
   SJMP    WDT_Loop
```

```
;=== Timer 1 interrupt 0.5 sec
TIMERINT:
   PUSH
   JBC       INTSTA,TMR1I,Q_Time
   BC        INTSTA,TMR1I
    :
    :
Q_Time:
   POP
   RETI
```

## 8.7 Low Voltage Detection (LVD)

- Selectable Built-in LVD (Low Voltage Detector), Vdet (Vth-) = 2.2V ~ 2.9V ± 0.1V
- When the voltage of the LVDIN pin is lower than the selected detection voltage (/LV=0), PA.7 interrupt / Wake up is disabled and the reset function of the RESET_KEY pin is unavailable.
- Selectable Built-in / External LVD
- Optional LVD interrupt

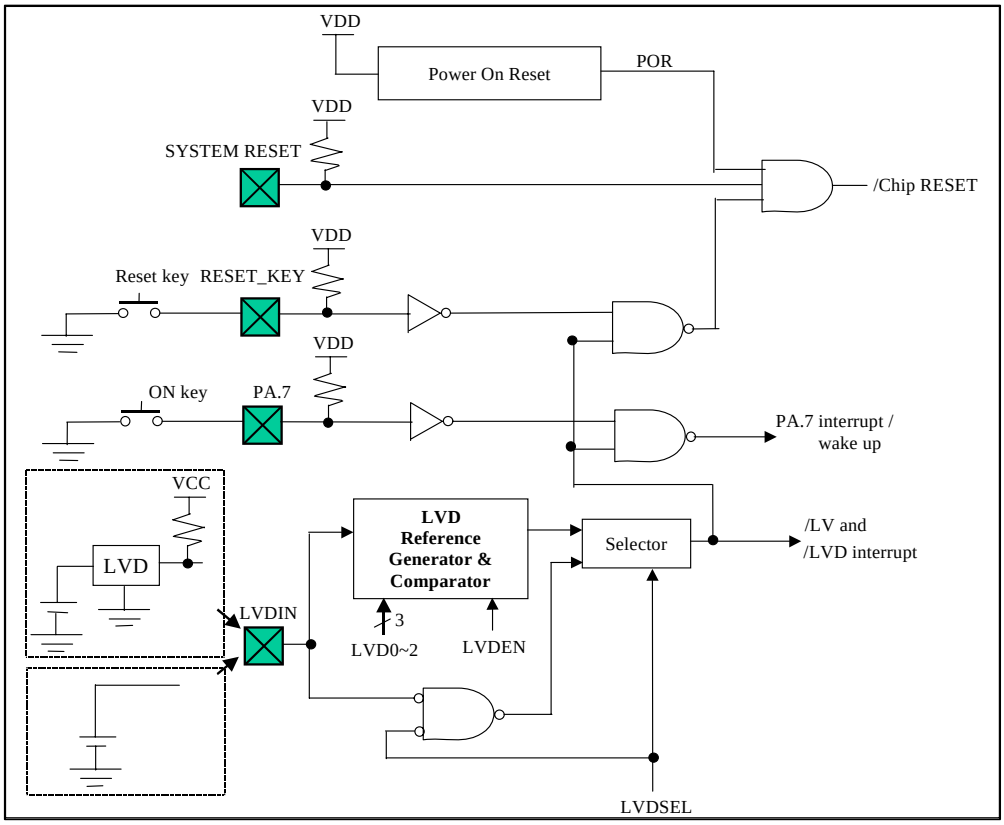



*Figure 8-7 LVD Function Block Diagram*

The Low Voltage Detection function is used to detect Battery Low level. When the **LVDEN** bit of the CPUCON Register is enabled, the **/LV** bit is the voltage detection result of the **LVDIN** pin. When the voltage of the **LVDIN** pin is lower than Vdet (Vth-), the **/LV** bit will be cleared.

Setup the **LVDSEL** bit of the STBCON register to select either external LVD device or internal built-in LVD.

Setup **LVD2~LVD0** bit of the PACON register to determine the detection voltage.

User can enable the Low Voltage Detect Interrupt function. If **/LV** bit becomes low, interrupt will simultaneously occur.

When low level is detected (/LV=0), the PA.7 interrupt/wake up and reset caused by the Reset_Key pin are disabled.

**LVDIN pin:** Input pin of Low Voltage Detection, used to examine whether the voltage level is higher than Vdet (Vth+) or lower than Vdet (Vth-).

■ **CPUCON (R0Eh):** CPU Control Register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| CLKOEN | CKS | LVDEN | /LV | /GLINTD | PS | MS1 | MS0 |

**Bit 3 (/GLINTD):** Global Interrupt Disable bit

"**0**" : Disable all interrupts

"**1**" : Enable all un-mask interrupts

**Bit 4 (/LV):** Low voltage detection (Read only bit)

"**0**" : LVDIN pin is lower than "Vdet (Vth-)", this bit will be cleared

"**1**" : LVDIN pin is higher than "Vdet (Vth+)", this bit will be set or LVD function is disabled

**Bit 5 (LVDEN):** Enable Low Voltage Detector. LVDEN bit will be cleared after the CPU is reset

"**0**" : Disable LVD function

"**1**" : Enable LVD function

■ **PACON (R21h):** Port A Control Register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| LVD2 | LVD1 | LVD0 | LVDSEL | Bit7PU | /R2EN | /R1EN | KE |

**Bit 4 (LVDSEL):** Built-in / External LVD select bit

"**0**" : External LVD selected

"**1**" : Built-in LVD selected

**Bit 7 ~ Bit 5 (LVD2 ~ LVD0):** LVD detection voltage select bits (limited value only)

| LVD 2 ~ 0 | Vdet (Vth-) | Vdet (Vth+) | Deviation |
|---|---|---|---|
| 000 | 2.2V | 2.3V | ± 0.1V |
| 001 | 2.3V | 2.4V | |
| 010 | 2.4V | 2.5V | |
| 011 | 2.5V | 2.6V | |
| 100 | 2.6V | 2.7V | |
| 101 | 2.7V | 2.8V | |
| 110 | 2.8V | 2.9V | |
| 111 | 2.9V | 3.0V | |

**Note:** The default value of LVD2 ~ 0 after the CPU is reset is "000".

■ **INTCON (R3Bh):** Interrupt Control Register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| - | - | - | SRBFIE | LVDIE | TMR2IE | TMR1IE | TMR0IE |

**Bit 3 (LVDIE):** LVD interrupt control bit

"**0**" : Disable LVD interrupt

"**1**" : Enable LVD interrupt

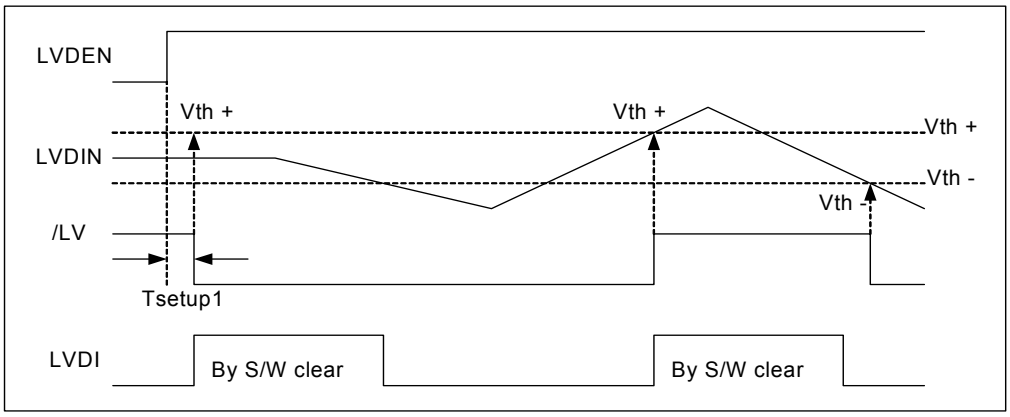
■ **INTSTA (R3Ah):** Interrupt Status Register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| - | - | - | SRBFI | LVDI | TMR2I | TMR1I | TMR0I |

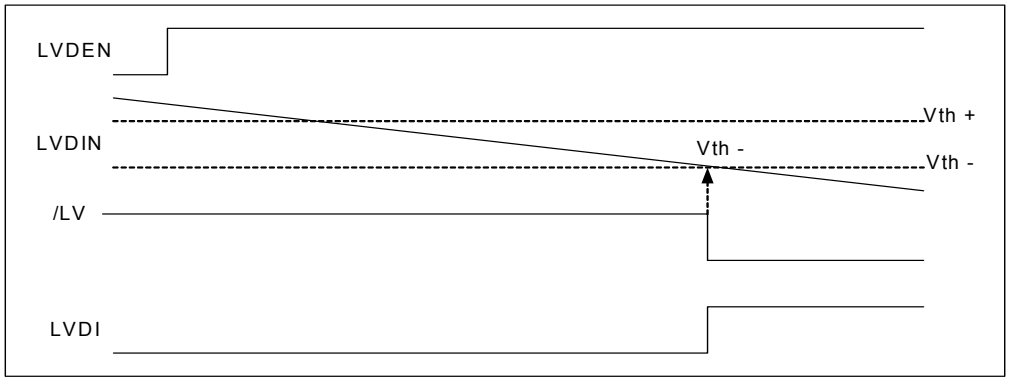
**Bit 3 (LVDI):** Set to 1 when LVD low level detect (falling edge) interrupt occurs.
Clear to 0 by software or disable LVD.

**Timing Diagram:**

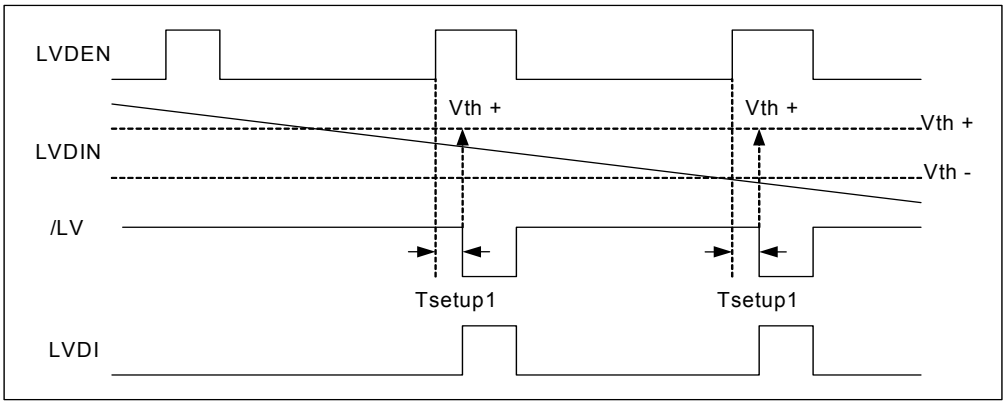
LVDIN voltage is within the Vdet threshold region when enabled.




Continuous LVD Detection



Power Saving Control of LVD




**Code Example**

```
;=== Low voltage detection (2.2V ~2.3V)
     :
    System setting 4MHz
     :
    MOV  A,#00001111b
    AND  A,PACON         ; LVD 2.2 ~ 2.3V
    BS   PACON,LVDSEL    ; Built-in LVD
    BS   CPUCON,LVDEN    ; LVD enable
    BC   INTSTA,LVDI
    BS   INTCON,LVDIE
    BS   CPUCON,GLINTD
MainLp:
    JBS  F_LVD/16,F_LVD%16,toBatteryLow
     :
```

```
PERIPH:
    JBS    INTSTA,LVDI,LVDINT
    RETI
;
;=== LVD interrupt routine
LVDINT:
    PUSH
    BC     INTSTA,LVDI
     :
    BS     F_LVD/16,F_LVD%16
     :
    POP
    RETI
```

## 8.8 Serial Peripheral Interface (SPI)

- Operation in either Master mode or Slave mode
- Three-wire or Four-wire full duplex synchronous communication
- Programmable Shift Register Length (24/16/8 bits)
- Programmable bit rates of communication
- Programmable clock polarity
- Programmable shift direction
- Programmable sample phase
- Interrupt flag available for the read buffer full
- Up to 2.5 MHz (system clock at 10 MHz) bit frequency

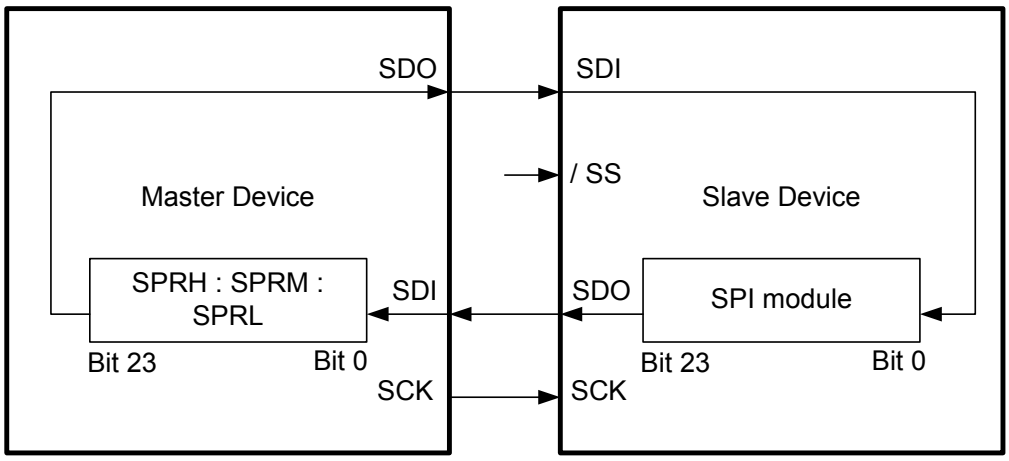



*Figure 8-8 Single SPI Master/Slave Communication*

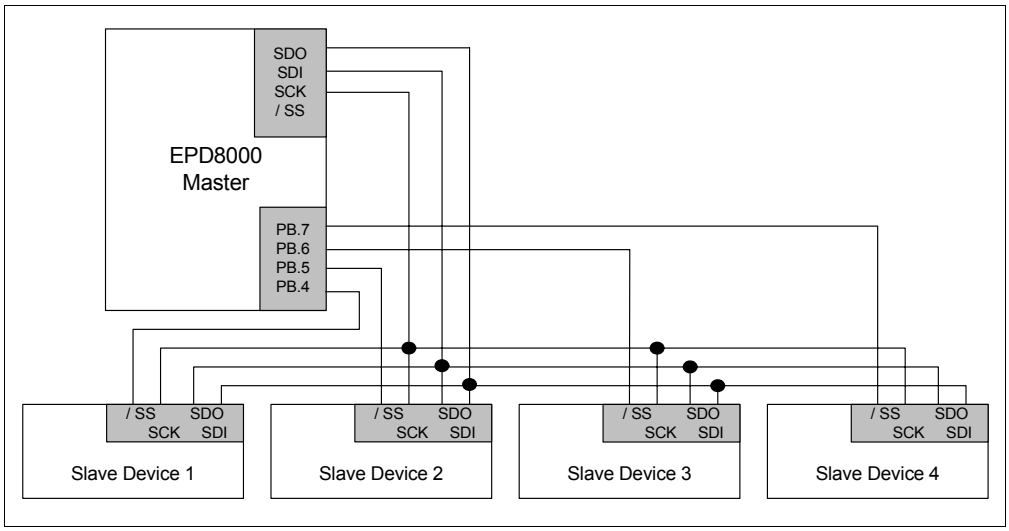



*Figure 8-9 SPI Configuration Example of Single-Master and Multi-Slaves*

The MCU communicates with other devices through an SPI module. If the MCU is defined as the master controller, it sends clock through the SCK pin. An 8-bit data is transmitted and received at the same time. If the MCU, however, is defined as a slave, its SCK pin could be programmed as an input pin. Data will continue to be shifted at selected clock rate and selected edge.

Setting up the **TLS1 ~ TLS0** bits of the SPICON register can select the shift register length of the SPI and enable/disable the SPI function. Setting up the **BRS2 ~ BRS0** bits of the SPICON register can select the SPI mode (master/slave) and Bit Rate. When in master mode, the clock source can be selected from the system clock or half of Timer 0 interval. When in slave mode, the **/SS** pin can be enabled or disabled. Setting up the **DORD** bit of the SPICON register can determine the shift direction. Setting up the **EDS** bit of the SPICON register can select either rising edge or falling edge to latch the data.

Setting up the **SMP** bit of the SPISTA register can select the sample phase whether at the middle or at the end of data output time.

✧ **Master Mode**

In master mode, the SCK pin functions as a clock output pin.

If a 24-bit shift register length is selected, SPRH, SPRM and SPRL registers are the high, middle and low bytes of the shift register. (Likewise, if an 8-bit shift register length is selected, SPRL register is the content of the shift register). Data are written to SPRH, SPRM and SPRL registers, after writing data into the SPRL register, the **SE** bit of the SPICON register will automatically be set by hardware and starts shifting. After a shift buffer is empty, the **SE** bit will be cleared by hardware and stops clock output from the **SCK** pin.

The receiver is active during SPI transfer. When the receiving buffer is full, the **RBF** flag will be set and an interrupt occurs (if enabled). During a read out of the shift register contents, and after the SPRL register has been read out, the hardware will automatically clear the **RBF** flag. If SPRL register has not been read out, **RBF** bit still remains set. Data collision will occur during the next clock input.

✧ **Slave Mode**

In slave mode, the input clock is from the MASTER device. **SCK** pin is a clock input pin. The **SE** bit is not used to control the starting shift in this mode. But it is a Transfer buffer empty status bit.

The same as with the Master Mode, user can select the shift register length. Transfer data are written to SPRH, SPRM, SPRL registers. After writing data into the SPRL register, the **SE** bit of SPICON register will be set by hardware. But the shifting start is controlled by the MASTER device clock input.

While the shift buffer is empty the **SE** bit will be cleared. At the same time, when the receive buffer is full, the **RBF** flag will be set and an interrupt occurs (if enabled). The received data is at SPRH, SPRM and SPRL register; user should read them out before the next clock input. Otherwise, data collision will occur and the **DCOL** bit of the SPISTA register will be set.

### 8.8.1 SPI Pin Description

**SDI (I):** Serial Data Input pin. Receives data serially.

**SDO (O):** Serial Data Output pin. Transmits data serially. In Slave mode, defined as high-impedance, if not selected.

**SCK (I/O):** Serial Clock input/output pin. When in Master mode, sends clock through the SCK pin. However, in Slave mode, SCK pin is programmed as an input pin).

**/SS (I):** /Slave Select pin. This pin becomes active when /SS function is enabled. (BRS=110), else /SS pin is a general purpose I/O.

Master device remains low for /SS pin to signify the slave(s) for transmit/receive data. Ignore the data on the SDI and SDO pins when /SS pin is high, since the SDO is no longer driven.

### 8.8.2 SPI Applicable Registers

- **SPRH; SPRM; SPRL (R41h; R42h; R43h):** SPI shift buffer for 24/16/8 bits length.

The buffer will ignore any write operation until shifting is completed. If user selects 24 bits shift buffer, it will include the SPRH: SPRM: SPRL. However, if 8 bits shift buffer is selected, only the SPRL register is included.

When writing data into the SPRL register, the **SE** bit of the SPICON register will be set by hardware and shifting starts. When the shift buffer is empty, and the receive buffer is full at the same time, the received data is shifted into SPRH, SPRM and SPRL registers. After the SPRL register has been read out, the hardware will automatically clear the **RBF** flag.

- **SPICON (R2Bh):** SPI Control Register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| TLS1 | TLS0 | BRS2 | BRS1 | BRS0 | EDS | DORD | SE |

**Bit 0 (SE):** Shift enable. Set to “1” automatically when writing data into the SPRL register and shifting starts. Reset to “0” when a transfer buffer empty is detected.

**NOTE**

*The SE bit is read-only and is cleared by hardware when SPI is enabled. Hence, writing to the SPRL register is necessary when user wants to start shifting the data.*

**Bit 1 (DORD):** Data transmission order

**"0"** : Shift left (MSB first)

**"1"** : Shift right (LSB first)

**Bit 2 (EDS):** Select the rising / falling edge latch by programming the EDS bit

**"0"** : Falling edge

**"1"** : Rising edge

**Bit 5 ~ Bit 3 (BRS2 ~ BRS0):** Bit rate select. Programming the clock frequency/rates and sources.

**000**: Master, TMR0/2

**001**: Master, Fsystem/4

**010**: Master, Fsystem/16

**011**: Master, Fsystem/64

**100**: Master, Fsystem/256

**101**: Master, Fsystem/1024

**110**: Slave, /SS enable

**111**: Slave, /SS disable

**SPI Bit Rate Table**

| Prescaler | | Fper | | |
|---|---|---|---|---|
| BRS2:0 | Bit Rate | 10MHz | 4MHz | 32.768kHz |
| 001 | Fper/4 | 2500000 | 1000000 | 8196 |
| 010 | Fper/16 | 625000 | 250000 | 2048 |
| 011 | Fper/64 | 156250 | 62500 | 512 |
| 100 | Fper/256 | 39063 | 15625 | 128 |
| 101 | Fper/1024 | 9766 | 3096 | 32 |

**Bit 7 ~ Bit 6 (TLS1 ~ TLS0):** Shift buffer length select. The Shift buffer length is programmable.

**00**: SPI disable

**01**: Enable SPI and shift buffer length = 24 bits

**10**: Enable SPI and shift buffer length = 16 bits

**11**: Enable SPI and shift buffer length = 8 bits

■ **SPISTA (R40h):** SPI Status Register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| LCDM1 | LCDM0 | SFR1 | SFR0 | SPWKEN | SMP | DCOL | RBF |

**Bit 0 (RBF):** Set to "1" by Buffer Full Detector, and automatically cleared to "0" when data are read from the SPRL register.

**NOTE**

*The RBF bit is cleared by hardware when SPI is enabled and this bit is read-only. Hence, reading the SPRL register is necessary to avoid data collision (DCOL) condition.*

**Bit 1 (DCOL):** SPI Data collision

**Bit 2 (SMP):** SPI data input sample phase

**"0"** : Input data sampled at the middle of data output time

**"1"** : Input data sampled at the end of data output time

| NOTE |
|---|
| *In Slave mode, data input sample is fixed at the middle of data output time.* |

**Bit 3 (SPWKEN):** SPI wake up enable control bit

**"0"** : Disable SPI (Slave mode) read buffer full wakeup

**"1"** : Enable SPI (Slave mode) read buffer full wakeup

- **INTCON (R3Bh):** Interrupt Control Register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| - | - | - | SRBFIE | LVDIE | TMR2IE | TMR1IE | TMR0IE |

**Bit 4 (SRBFIE):** SPI read Buffer Full Interrupt control bit

**"0"** : Disable interrupt function

**"1"** : Enable interrupt function

- **INTSTA (R3Ah):** Interrupt Status Register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| - | - | - | SRBFI | LVDI | TMR2I | TMR1I | TMR0I |

**Bit 4 (SRBFI):** Set to 1 when SPI read buffer full occurs. Clear to 0 by software or disable SPI.

**"0"** : Data collision didn't occur

**"1"** : Data collision occurs. Should be cleared by software.

- **CPUCON (R0Eh):** CPU Control Register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| CLKOEN | CKS | LVDEN | /LV | /GLINTD | PS | MS1 | MS0 |

**Bit 3 (/GLINTD):** Global interrupt disable bit

**"0"** : Disable all interrupts

**"1"** : Enable all un-mask interrupts

- **Master Mode (Shift Buffer Length = 24Bits)**

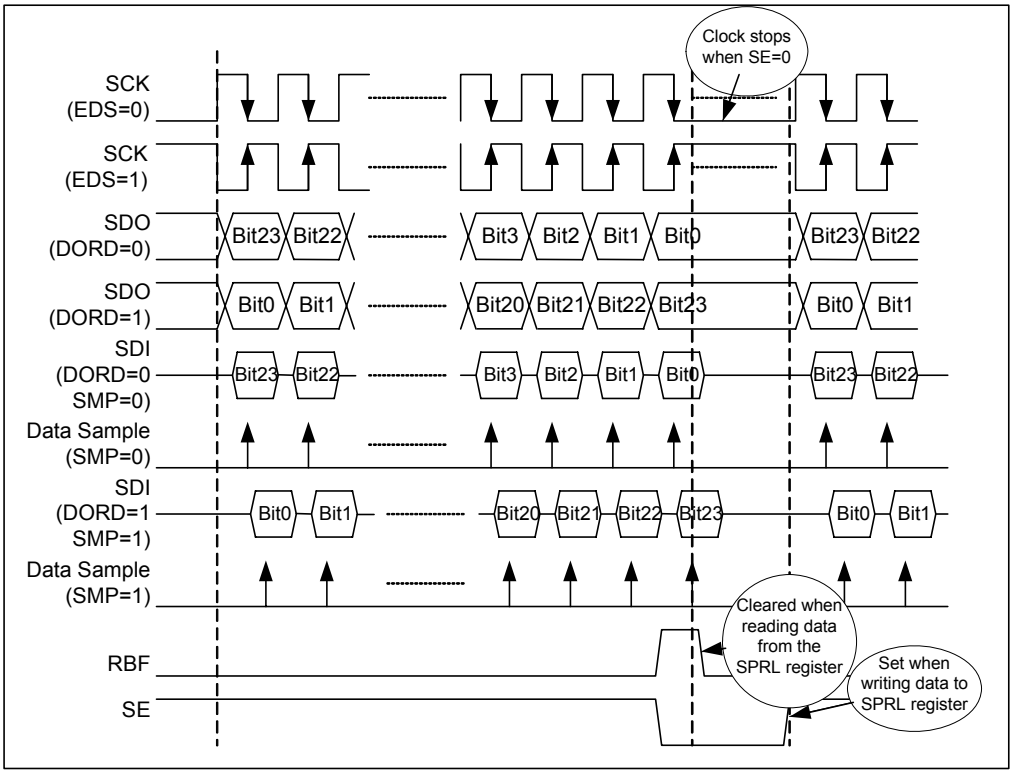



*Figure 8-10 SPI Master Mode Timing Diagram*

**Code Example: Master Mode (8 bit)**

```
;***Interrupt SPI
PERIPH:
     PUSH
     COMA    DATACNT
;---SPI read buffer full
     JBC     INTSTA,SRBFI,Q_SPINT
     BC      INTSTA,SRBFI
     BS      INTFLAG,F_SPI
;---SPI Data collision
     JBC     INTSTA,DCOL,Q_SPINT
     MOV     A,#0XFF
Q_SPINT:
     MOV     DATACNT,A
     POP
     RETI
```

```
;===8MHz/4 = 2000000 bit rate
SPIM_SR:
      :
     System setting 8MHz
     Port G setting output port
      :
;---8bit, Fper/2, Rising edge & MSB
     MOV     A,# 11001100B
     MOV     SPICON,A
```

```
;---SPI full interrupt
     MOV     A,#00000000B
     MOV     SPISTA,A
     BC      INTSTA,SRBFI
     BS      INTCON,SRBFIE
;---Global interrupt
     BS      CPUCON,GLINTD
;---SPI data output => 55
     MOV     A,#0X55
     MOV     DATACNT,A
SPI8LOOP:
     MOV     A,DATACNT
     MOV     SPRL,A
;---SPI Data collision
     JBC     SPISTA,DCOL,SPI8LP1
     BC      SPISTA,DCOL
;---SPI data output resend => 55
     MOV     A,#0X55
     MOV     DATACNT,A
SPI8LP1:
     JBC     INTFLAG,F_SPI,SPI8LP1
SPI8LP2:
     BC      INTFLAG,F_SPI
     MOVRP   PORTG,SPRL
     SJMP    SPI8LOOP
```

■ **Slave Mode (Shift Buffer Length = 8Bits, /SS Enabled)**

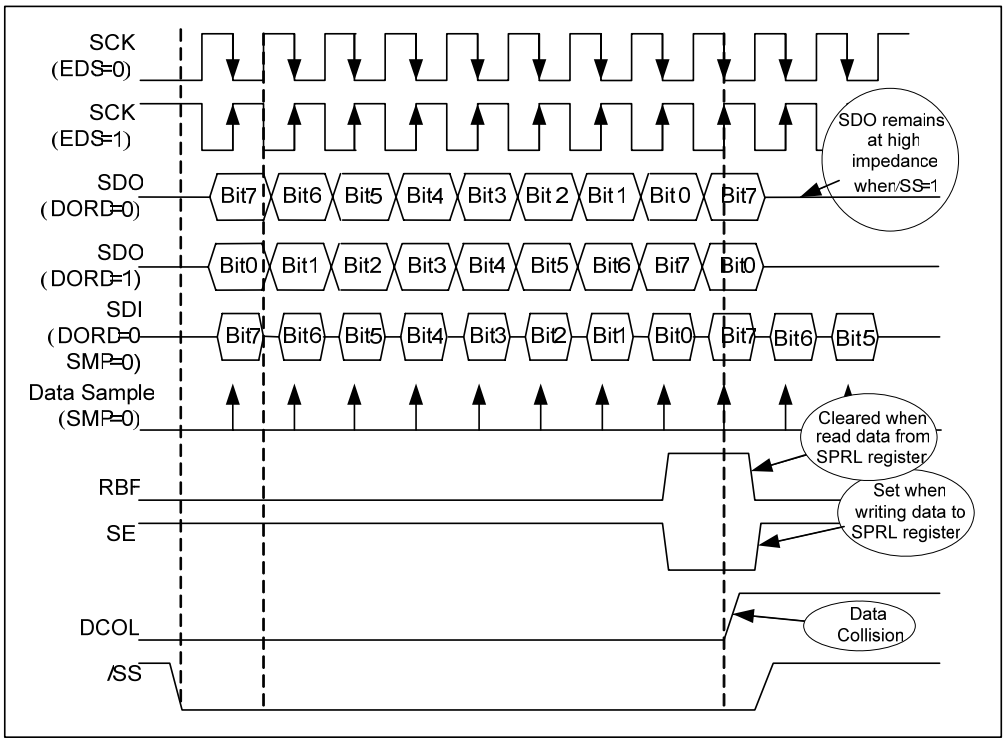



*Figure 8-11 SPI Slave Mode Timing Diagram*

**Code Example: Slave Mode (8 Bit)**

```
;***SPI slave mode
      :
     System setting 8 MHz
     Port G setting output port
      :
;===SPI 8bit & Sleep mode
SPIS_SR:
;---8bit, Slave /SS enable, Rising edge & LSB
     MOV       A,#11110100B
     MOV       SPICON,A
;---SPI Wakeup & SPI full interrupt
     MOV       A,#00001000B
     MOV       SPISTA,A
     BC        INTSTA,SRBFI
     BS        INTCON,SRBFIE
;---Global interrupt
     BS        CPUCON,GLINTD
;---Sleep mode
     BC        CPUCON,MS1
     SJMP      SPIS8Lp
```

```
SPIS8Lp:
     SLEP
     NOP
     MOVRP     PORTG,SPRL
;---SPI Data collision
     JBC       SPISTA,DCOL,SPIS8Lp
     MOV       A,#0XFF
     MOV       SPRL,A
      :
      :
```

```
;***Interrupt SPI
PERIPH:
     PUSH
     JBC       INTSTA,SRBFI,Q_SPINT
     BC        INTSTA,SRBFI
Q_SPINT:
     POP
     RETI
```

## 8.9 Key I/O

The 7-pin key input (Port A.0~6) and 16-pin key strobe (share with LCD segment) has a maximum of 112-key matrix.

### *8.9.1 Automatic Key Scan or Software Key Scan*

Interrupt is enabled when in automatic key scan (**SCAN**=1). Wake-up is also enabled when key input falling edge is detected at automatic key scan mode (**SCAN**=1).

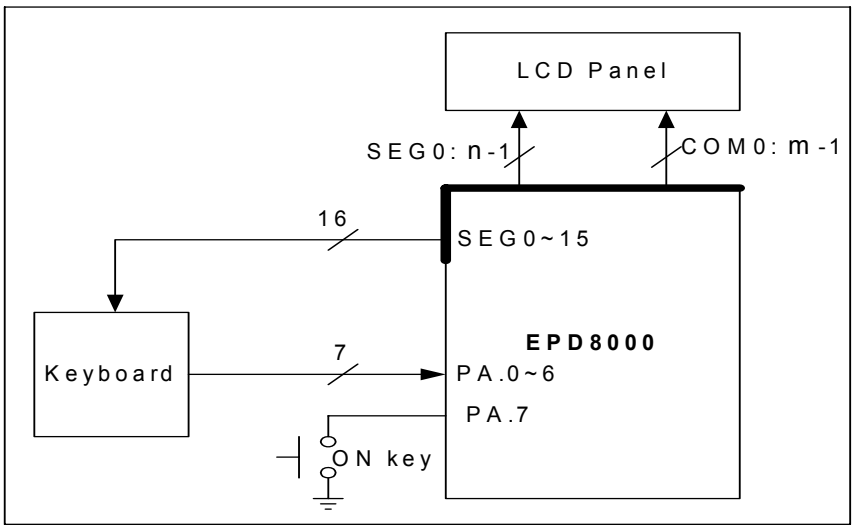



*Figure 8-12 Keyboard Function Block Diagram*

As shown in the figure, it is taken into consideration that the Key strobe output has resistance $R_{ON,}$ as well as each key has resistance $K_{ON}$ and capacitance C. Since a prolong strobe output makes an LCD display confusing, strobe output should be as short as possible. So $R_{IN}$ (pull-up resistance) should be low enough for the capacitance to be charged quickly. But $R_{IN}$ should be high enough for $V_{IN}$ to be ascertained in "L" level. ($R_{IN}$ >> $R_{ON}$ + $K_{ON}$) Hence, the value of $R_{IN}$ should be flexible.

The following are the “Key input process”:

1. Output the strobe signal
2. Pull up the input port by lowest resistance (R1 and R2 enabled): Capacitance is charged quickly.
3. Pull up the input port by highest resistance (only R2 is enabled).
4. Read the key
5. Disable the pulled-up resistance
6. Stop the strobe signal

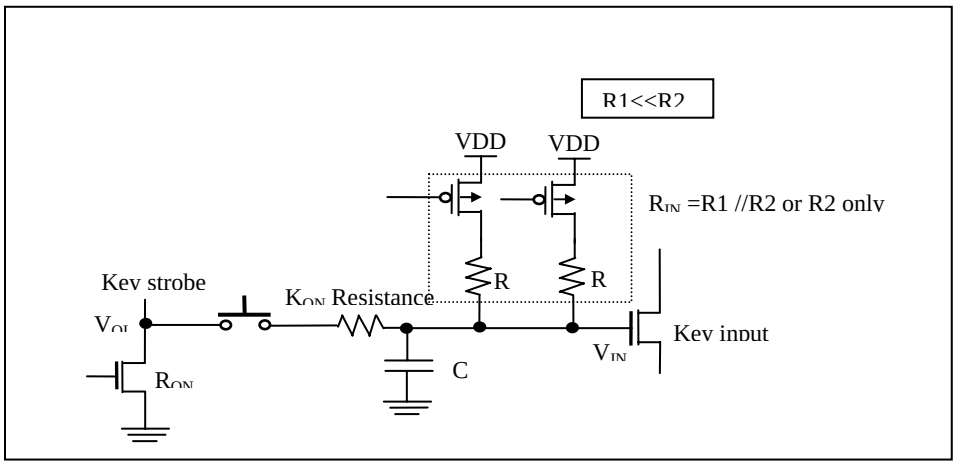



*Figure 8-13 Key Circuit Diagram*

The detailed function is summarized in the following table:

| SCAN | BitST | KE | /R1EN | /R2EN | SSCAN[1] | SCAN[1] | IEN[1] | Total Pull-up Resistor | Port A.0~6 | Note |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | × | 1 | 1 | 0 | 0 | 0 | Floating | 0 | |
| | 1 | 0 | 1 | 1 | | | 0 | Floating | 0 | A |
| | | | 1 | 0 | | | 0 | R2 | 0 | |
| | | | 0 | 1 | | | 0 | R1 | 0 | |
| | | | 0 | 0 | | | 0 | R1 // R2[2] | 0 | B |
| | | 1 | 1 | 1 | | | 1 | Floating | Floating | Prohibited |
| | | | 1 | 0 | | | 1 | R2 | PA.0~6 | C |
| | | | 0 | 1 | | | 1 | R1 | PA.0~6 | |
| | | | 0 | 0 | | | 1 | R1 // R2[2] | PA.0~6 | |
| 1 | × | × | 1 | 1 | 0 | 0 | 0 | Floating | 0 | A |
| | | | 0 | 0 | 0 | 1 | 0 | R1//R2[2] | 0 | B |
| | | | 1 | 0 | 1 | 1 | 1 | R2 | PA.0~6 | C |

[1] Internal signal - Refer to the "Automatic Key Scan Timing"
[2] R1 // R2 = R1R2 / (R1+R2)

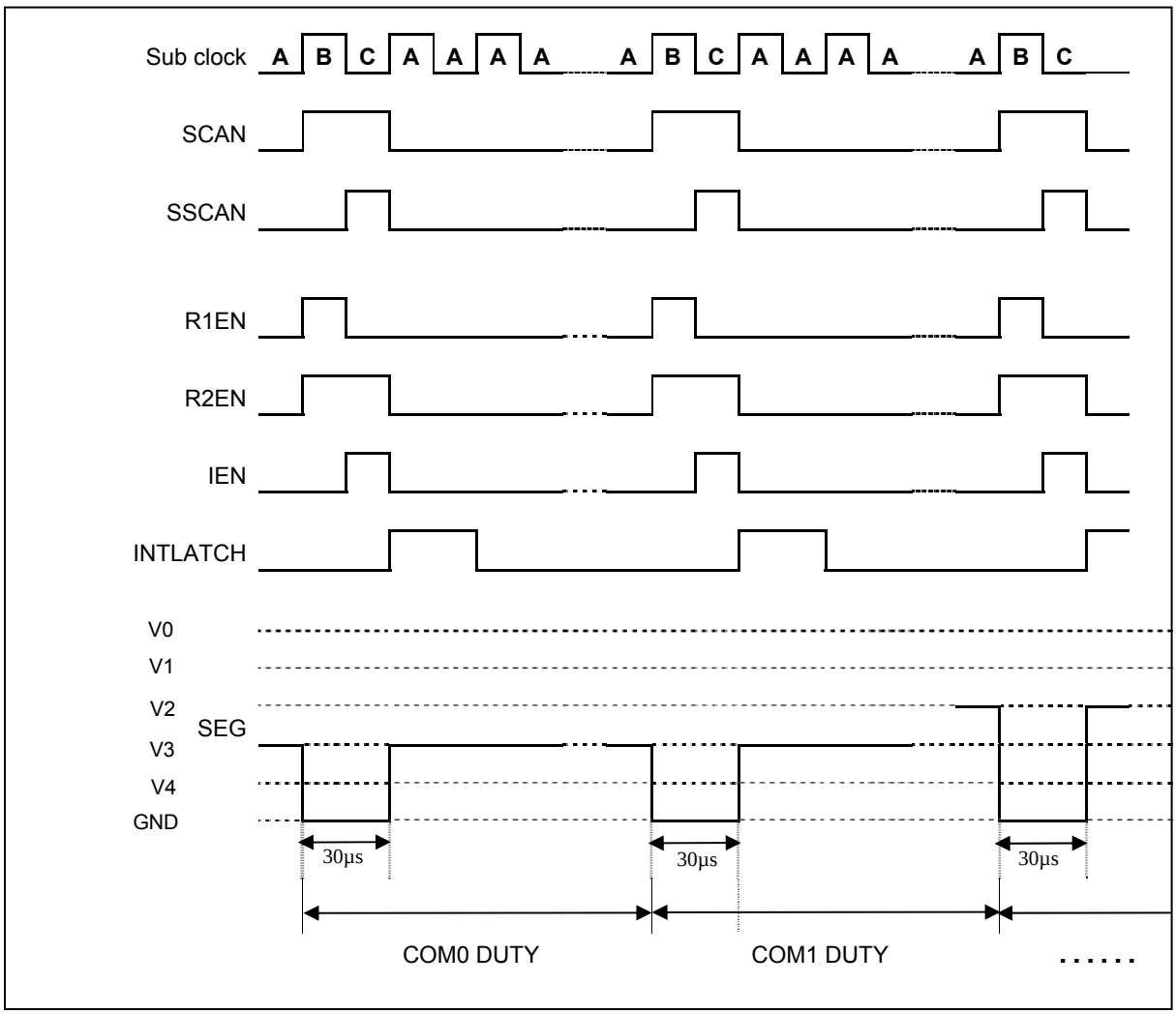



*Figure 8-14 Automatic Key Scan Timing Diagram (SCAN = 1)*

The Key strobe pin shares with the LCD segment pin in the CPU with embedded LCD driver model.

When pin-shared with an LCD segment, to avoid having a confusing LCD display, the strobe output should be as short as possible.

There are two ways to output a strobe signal:

✧ **Automatic Key Scan**

The LCD waveform has a 30µs low pulse at the beginning of every common duty signal by setting the **SCAN** bit of the STBCON register. The strobe timing is as shown in the figure.

When in automatic key scan mode, the PAINT or Port A wake-up must be enabled. During Key scan, wake-up and interrupt will occur if key input pin (Port A) falling edge is detected.

✧ **Software Key Scan**

Segment is switched to strobe signal temporarily by setting the **BitST** bit of the STBCON register to 1 and the **SCAN** bit to 0.

Setup the **STB3~STB0** bits of the STBCON register to select which pin will be strobes.

In this mode user can set **ALL** bits of the STBCON register to let Segments 0~15 to be kept at GND level.

**In Idle Mode**, during automatic key scanning, if PA.0~7 pin falling edge is detected (when IEN=1), wake-up will occur. Then the CPU runs and interrupt occurs (if enabled).

**In Slow Mode or Fast Mode,** both automatic key scan and software key scan will be used.

Automatic key scan will be used to examine **If there is any key pressed?** If a key is pressed, PA.0~7 pin falling edge is detected**,** then an **interrupt will occur**.

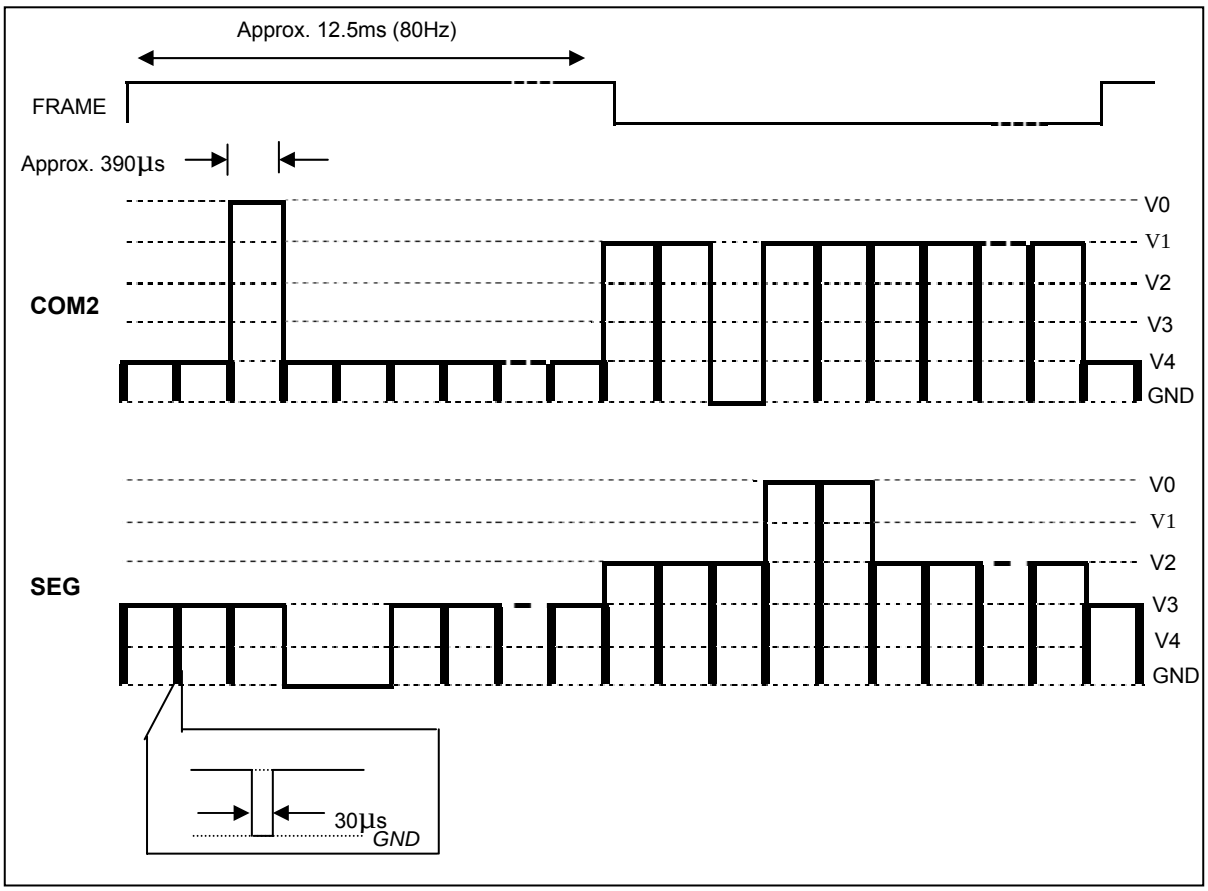



*Figure 8-15 Automatic Strobe Signal (SCAN = 1)*

Software key scan will examine “which key was pressed”.

The function of key strobe pin is as shown in the following table.

| STBCON | | | | Key Strobe (Share with Segments 0~15) | | | | | | | | | | | | | | | LCD | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SCAN | BitST | ALL | STB3~0 | seg 0 | seg 1 | seg 2 | seg 3 | seg 4 | seg 5 | seg 6 | seg 7 | seg 8 | seg 9 | seg 10 | seg 11 | seg 12 | seg 13 | seg 14 | seg 15 | seg 16:n-1 | com 0:m-1 |
| | 0 | × | ×××× | Display waveform | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | 0000 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | Display waveform | |
| | | | 0001 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | |
| | | | 0010 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | |
| | | | 0011 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | |
| | | | 0100 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | |
| | | | 0101 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | |
| | | | 0110 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | |
| | | | 0111 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | |
| | | | 1000 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | |
| | | | 1001 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | | |
| | | | 1010 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | | |
| | | | 1011 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | | |
| | | | 1100 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | | |
| | | | 1101 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | | |
| | | | 1110 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | | |
| | | | 1111 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | | |
| | | 1 | ×××× | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 1 | × | × | ×××× | Display waveform with automatic key scan | | | | | | | | | | | | | | | | | |

### *8.9.2 Input/Output Key Applicable Registers*

- **Port A (R19h):** Port A Register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |

**Bit 7 ~ Bit 0:** Key input. Input falling edge interrupt or wake-up pin.

- **PACON (R21h):** Port A Control Register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| LVD2 | LVD1 | LVD0 | LVDSEL | Bit7PU | /R2EN | /R1EN | KE |

**Bit 0 (KE):** Key input enable/disable control bit
“**0**” : Disable Key input function (Port A register does NOT correspond with Key input in software scan mode)
“**1**” : Enable Key input function (Port A register corresponds with the Key input in software scan mode)

**Bit 1 (/R1EN):** R1 pull-up resistor (small resistor) control bit
“**0**” : Enable R1 pull-up resistor
“**1**” : Disable R1 pull-up resistor

**Bit 2 (/REN):** Port A.0~Port A.6 Pull up resistor control bit
“**0**” : Enable pull-up resistor
“**1**” : Disable pull-up resistor

■ **PAWAKE (R2Dh):** Port A Wake-up Control Register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| WKEN7 | WKEN6 | WKEN5 | WKEN4 | WKEN3 | WKEN2 | WKEN1 | WKEN0 |

**Bit 7 ~ Bit 0 (WKEN7 ~ WKEN0):** Port A wake-up function control bit

"**0**" : Disable wake-up function

"**1**" : Enable wake-up function

■ **PAINTEN (R39h):** Port A Interrupt Enable Control Register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| PA7IE | PA6IE | PA5IE | PA4IE | PA3IE | PA2IE | PA1IE | PA0IE |

**Bit 7 ~ Bit 0 (PA7IE ~ PA0IE):** Interrupt control bit

"**0**" : Disable

"**1**" : Enable

■ **PAINTSTA (R2Eh):** Port A Interrupt Status Register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| PA7I | PA6I | PA5I | PA4I | PA3I | PA2I | PA1I | PA0I |

**Bit 7 ~ Bit 0 (PA7I ~ PA0I):** Port A interrupt INT status

Set to 1 when pin falling edge is detected
Cleared to 0 by software

■ **STBCON (R22h):** Strobe Output Control Register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| - | SCAN | BitST | ALL | STB3 | STB2 | STB1 | STB0 |

**Bit 3 ~ Bit 0 (STB3 ~ STB0):** Strobe output selector bit

**Bit 4 (ALL):** Set All strobes

"**0**" : Bit strobe

"**1**" : All strobes

**Bit 5 (BitST):** Enable Bit strobe

"**0**" : Port H and Port I are general I/O port

"**1**" : Port H and Port I are key strobe pin. Strobe signal is defined as STB3~0

**Bit 6 (SCAN):** Automatic key scan or specify the scan signal bit by bit.

"**0**" : Key scan is specified by the defined bits STB3~0

"**1**" : Auto strobe scanning

■ **CPUCON (R0Eh):** CPU Control Register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| CLKOEN | CKS | LVDEN | /LV | /GLINTD | PS | MS1 | MS0 |

**Bit 3 (/GLINTD):** Global interrupt disable bit

**"0"** : Disable all interrupts

**"1"** : Enable all un-mask interrupts

**Code Example**

```
; Key matrix 1 (Port A and Ground):
;===Sleep mode
PAIN_SR:
      :
;---Port B output
     CLR       DCRB
;---Port A wakeup
     MOV       A,#11111111B
     MOV       PAWAKE,A
;---/R1EN & /R2EN Pull-up & KE enable
     MOV       A,#00000001B
     MOV       PACON,A
;---Port A interrupt enable
     MOV       A,#11111111B
     MOV       PAINTEN,A
     CLR       PAINTSTA
     BS        CPUCON,GLINTD
;---Sleep MODE
     BC        CPUCON,MS1
PAINloop:
     BS        STBCON,BitST
     SLEP
     NOP
      :
     SJMP      PAINloop
```

```
;***Interrupt Port A data
INPTINT:
     PUSH
     MOVRP     PORTB,PAINTSTA
     BC        STBCON,BitST
     CLR       PAINTSTA
     POP
     RETI
```

```
; Key matrix 2 (Port A and SEG0 ~ SEG15):
;***Key scan function
     :
    LCD display setting
     :
    MOV      A,#0XFF
    MOV      PAWAKE,A              ; Port A setting wake-up function
     :
;===Key strobe all routine
KeyAll:
    CLR      PACON                 ; /R1EN, /R2EN enable
    BS       STBCON,ALL            ; Strobe all
    LCALL    DLY10US
    BS       PACON,R1EN            ; /R1EN disable
    BS       PACON,KE              ; Key enable
    LCALL    DLY10US
    MOV      A,PORTA               ; Port A input data
    BC       STBCON,ALL            ; Strobe all disable
    BS       PACON,R2EN            ; /R2EN disable
    BC       PACON,KE              ; Key disable
    JE       A,#0XFE,KeyScan
;
;===Idle mode auto key scan routine
    BS       STBCON,SCAN           ; Auto-key scan enable
    BS       CPUCON,MS1            ; Idle mode
KeyIdle:
    SLEP
    NOP
    MOV      A,PORTA               ; Port A input data
    JE       A,#0XFF,KeyIdle
;
;===Key scan routine
    CLR      STBCON                ; Auto-key scan disable
KeyLoop:
    CLR      PACON                 ; /R1EN, /R2EN enable
    BS       STBCON,BitST          ; Strobe ON
    LCALL    DLY10US
    BS       PACON,R1EN            ; /R1EN disable
    BS       PACON,KE              ; Key enable
    LCALL    DLY10US
    MOV      A,PORTA               ; Port A input data
    BS       STBCON,BitST          ; Strobe OFF
    BS       PACON,R2EN            ; /R2EN disable
    BC       PACON,KE              ; Key disable
    JLE      A,#0FEH,KeyScan       ; If A >= PORTA Goto KeyScan
    INC      STBCON
    LCALL    DLY400US
    SJMPKeyLoop
KeyScan:
;---Check key number
    CLR      Key_No
ChKeyNo:
    RRC      ACC
    JBC      STATUS,F_C,KeyScanOk
    INC      Key_No
    SJMPChKeyNo
;---Key Scan is finish
KeyScanOk:
    SWAPA    STBCON
    AND      A,#11110000B
    OR       Key_No,A
    SAWPKey_No                     ; Key_No:0XXX XXXX
     :
```

## 8.10 Embedded LCD Driver

The **EPD8000** provides a directly driven LCD. It supports 98 SEG × 32 COM. There is an LCD RAM for direct correspondence with LCD Pixel. Charge pump can pump to 2 times of VDD. The LCD contrast has 32 levels adjustable. And the LCD bias is selectable. The maximum LCD operating voltage can be determined by external resistor RA and RB.

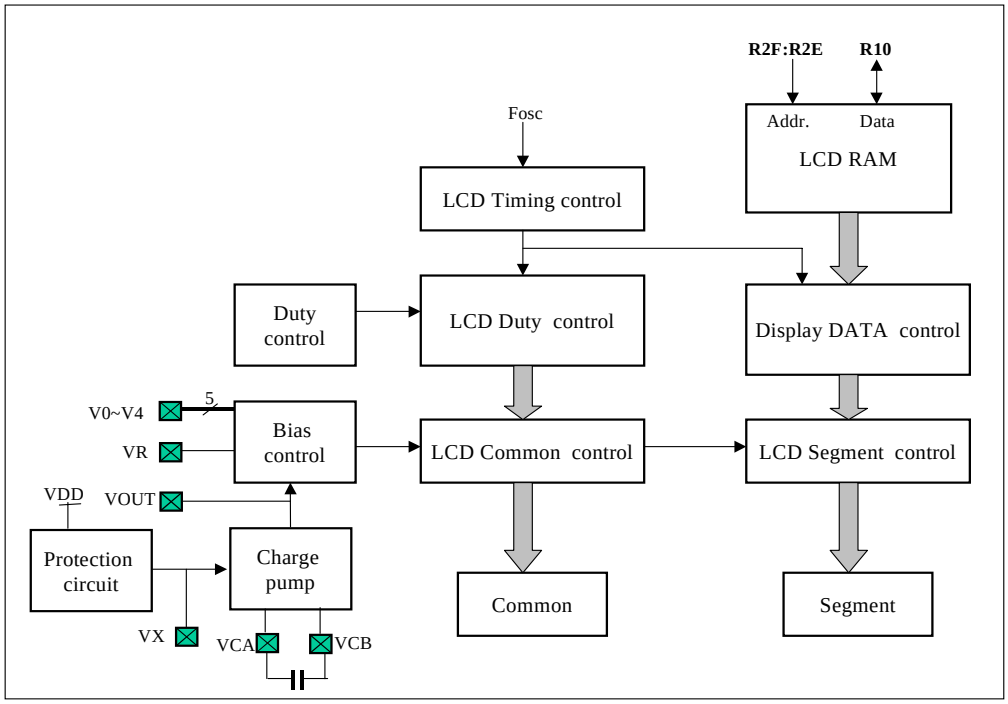



*Figure 8-16 Function Block Diagram*

The internal power supply circuits generate the voltage levels to drive liquid crystal driver circuits with low-power consumption and the least components. There are voltage converter (V/C) circuits, voltage regulator (V/R) circuits, and voltage follower (V/F) circuits.

The internal power supply circuits are shown in the following figure.

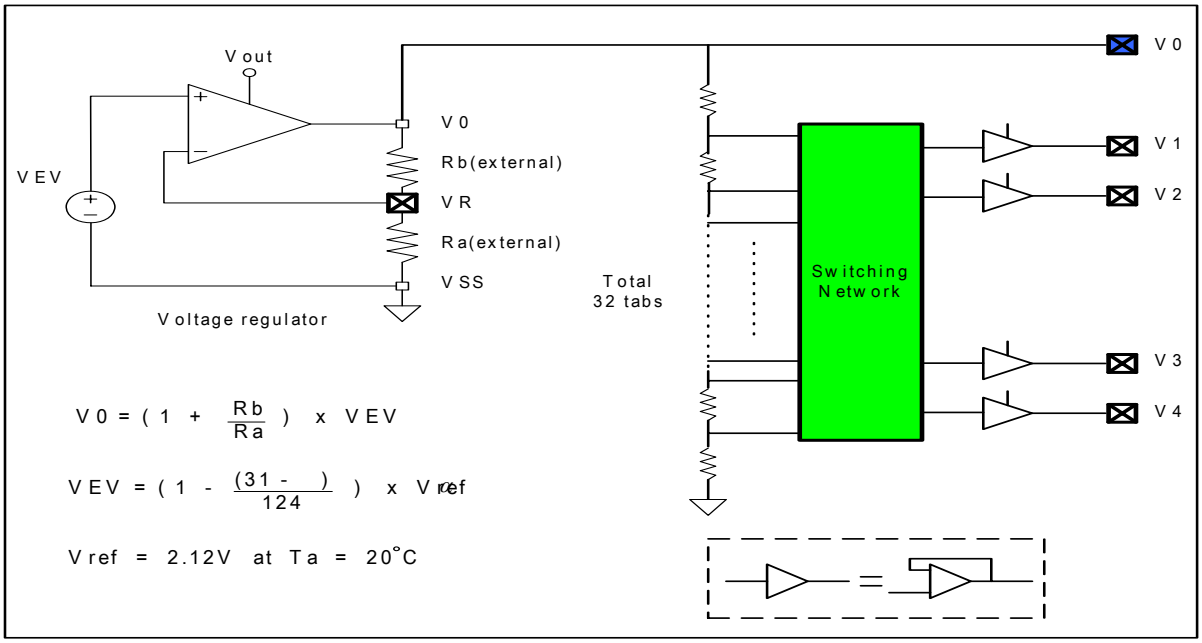



*Figure 8-17 Internal Power Supply Circuitry*

### *8.10.1 LCD Code Option*

- Duty Ratio: Maximum Duty Ratio Option

  Select "**1/11 duty**" or "**1/16 duty**" or "**1/24 duty**" or "**1/32 duty**".

- EXTVH: Vout is from an external power source or internal charge pump selection.

  Select "Internal" or "External".

- V1; V2; V3 and V4 OP Buffer

  Select "**Source, Sink, Source & Sink"** or **"For auto key scan**" or "**Class A/B**" or "**OFF**" or "**V1, V4: Auto Key Scan, V2, V3: OFF**"

| | Small Current | Normal Current | Large Current | No Current |
|---|---|---|---|---|
| V1 | Source | * | Class A/B | OFF |
| V2 | Sink | * | Class A/B | OFF |
| V3 | Source | * | Class A/B | OFF |
| V4 | Sink | * | Class A/B | OFF |

* When in normal display: V1 = source, V2 = sink, V3 = source, V4 = sink

When in auto key scan: During every 30 µs strobe start, the OP Amp. changes to class A/B for 60µs, then return to normal display.

■ **LCDCON (R19h):** LCD Control Register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| BSEL2 | BSEL1 | BSEL0 | ADJ4 | ADJ3 | ADJ2 | ADJ1 | ADJ0 |

**Bit 7 ~ Bit 5 (BSEL2 ~ 0):** LCD Bias select

| BSEL2 | BSEL1 | BSEL0 | LCD Bias |
|---|---|---|---|
| 0 | 0 | 0 | 1/3 |
| 0 | 0 | 1 | 1/3.5 |
| 0 | 1 | 0 | 1/4 |
| 0 | 1 | 1 | 1/4.5 |
| 1 | 0 | 0 | 1/5 |
| 1 | 0 | 1 | 1/5.5 |
| 1 | 1 | 0 | 1/6 |
| 1 | 1 | 1 | 1/6.5 |

**Bit 4 ~ Bit 0 (ADJ4 ~ 0):** LCD contrast adjustment

| ADJ4 | ADJ3 | ADJ2 | ADJ1 | ADJ0 | α | VEV | Contrast |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0.75 Vref | Low |
| 0 | 0 | 0 | 0 | 1 | 1 | 0.758 Vref | |
| . | . | . | . | . | . | . | |
| . | . | . | . | . | . | . | |
| 1 | 1 | 1 | 1 | 0 | 30 | 0.992Vref | |
| 1 | 1 | 1 | 1 | 1 | 31 | Vref | High |

$$V0 = (1 + \frac{Rb}{Ra}) \times VEV$$ ($\frac{Rb}{Ra}$ is external resistor ratio)

$$VEV = (1 - \frac{(31 - \alpha)}{124}) \times Vref$$ (Vref = 2.12V at 20°C)

■ **SPISTA (R2Ch):** SPI Status Register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| LCDM1 | LCDM0 | SFR1 | SFR0 | SPWKEN | SMP | DCOL | RBF |

**Bit 5 ~ Bit 4 (SFR1 ~ 0):** Frame Frequency Adjustment

| SFR1 | SFR0 | CL Frequency | Frame Frequency (Hz) | | |
|---|---|---|---|---|---|
| | | | Min | Typ | Max |
| 0 | 0 | Fosc /13 | 59.1 | 78.8 | 98.6 |
| 0 | 1 | Fosc / 14 | 54.9 | 73.2 | 91.5 |
| 1 | 0 | Fosc / 15 | 51.3 | 68.3 | 85.4 |
| 1 | 1 | Fosc / 16 | 48.0 | 64.1 | 80.1 |

**Note:** 1. Fosc = 32.8kHz ± 25% (all conditions)
2. This table is based on 1/32 duty.

**Bit 7 ~ Bit 6 (LCDM1 ~ 0):** LCD operation mode control register.

| LCDM1: LCDM0 | Operation Mode |
|---|---|
| 00 | Disable (LCD off) |
| 01 | Blanking |
| 10 | LCD enable |
| 11 | |

**Note:** Blanking means All COM and SEG pins are tied to ground.

■ **POST_ID (R30h):** Post increase/decrease the control register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| - | LCD_ID | FSR1_ID | FSR0_ID | - | LCDPE | FSR1PE | FSR0PE |

**Bit 2 (LCDPE):** Enable LCDARL: LCDARL post increase/decrease function

**Bit 6 (LCD_ID):** Set to 1 means auto-increase, reset to 0 means auto-decrease the LCDARH: LCDARL register.

**LCDARH (R2Fh)** is Page address for the LCD RAM (valid from 00H to 03H).

**LCDARL (R2Eh)** is Column address for the LCD RAM (valid from 00H to 61H).

**LCDDATA (R10h)** is an indirect addressing pointer of the LCD RAM.

**LCD RAM MAP:**

PAGE 00 (LCDARH=00H)

| RAM Address | | COM0 | COM1 | COM2 | COM3 | COM4 | COM5 | COM6 | COM7 |
|---|---|---|---|---|---|---|---|---|---|
| | LCDARL | Bit 0 | Bit 1 | Bit 2 | Bit 3 | Bit 4 | Bit 5 | Bit 6 | Bit 7 |
| SEG0 | 00H | | | | | | | | |
| SEG1 | 01H | | | | | | | | |
| SEG2 | 02H | | | | | | | | |
| : | : | | | | | | | | |
| : | : | | | | | | | | |
| SEG96 | 60H | | | | | | | | |
| SEG97 | 61H | | | | | | | | |

PAGE 01 (LCDARH=01H)

| RAM address | | COM8 | COM9 | COM10 | COM11 | COM12 | COM13 | COM14 | COM15 |
|---|---|---|---|---|---|---|---|---|---|
| | LCDARL | Bit 0 | Bit 1 | Bit 2 | Bit 3 | Bit 4 | Bit 5 | Bit 6 | Bit 7 |
| SEG0 | 00H | | | | | | | | |
| SEG1 | 01H | | | | | | | | |
| SEG2 | 02H | | | | | | | | |
| : | : | | | | | | | | |
| : | : | | | | | | | | |
| SEG96 | 60H | | | | | | | | |
| SEG97 | 61H | | | | | | | | |

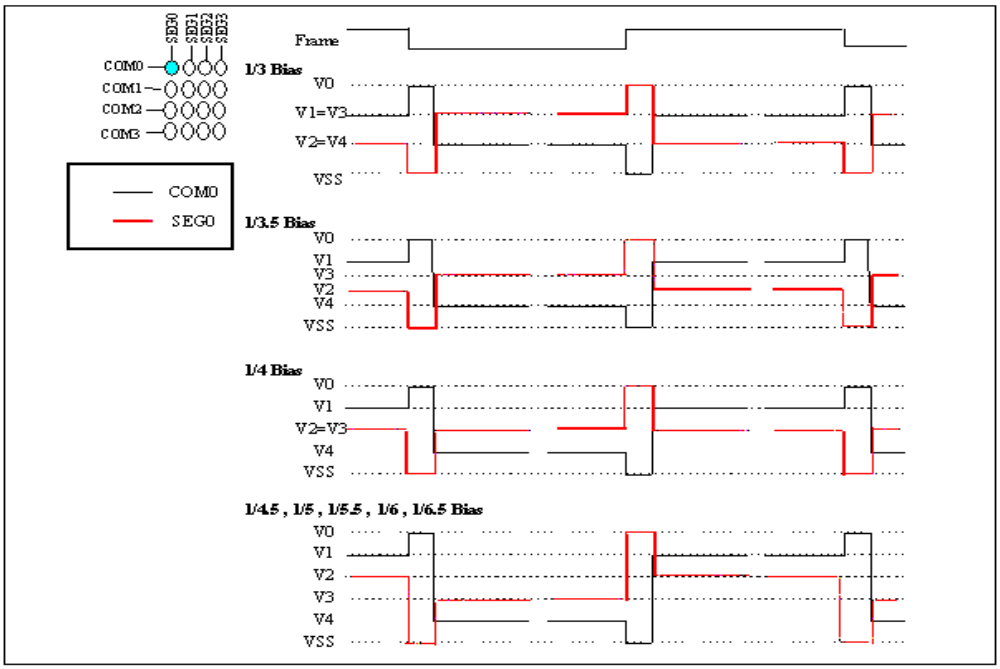
Page 02 (LCDARH=02H)

| RAM Address | | COM16 | COM17 | COM18 | COM19 | COM20 | COM21 | COM22 | COM23 |
|---|---|---|---|---|---|---|---|---|---|
| | LCDARL | Bit 0 | Bit 1 | Bit 2 | Bit 3 | Bit 4 | Bit 5 | Bit 6 | Bit 7 |
| SEG0 | 00H | | | | | | | | |
| SEG1 | 01H | | | | | | | | |
| SEG2 | 02H | | | | | | | | |
| : | : | | | | | | | | |
| : | : | | | | | | | | |
| SEG96 | 60H | | | | | | | | |
| SEG97 | 61H | | | | | | | | |

Page 03 (LCDARH=03H)

| RAM Address | | COM24 | COM25 | COM26 | COM27 | COM28 | COM29 | COM30 | COM31 |
|---|---|---|---|---|---|---|---|---|---|
| | LCDARL | Bit 0 | Bit 1 | Bit 2 | Bit 3 | Bit 4 | Bit 5 | Bit 6 | Bit 7 |
| SEG0 | 00H | | | | | | | | |
| SEG1 | 01H | | | | | | | | |
| SEG2 | 02H | | | | | | | | |
| : | : | | | | | | | | |
| : | : | | | | | | | | |
| SEG96 | 60H | | | | | | | | |
| SEG97 | 61H | | | | | | | | |

LCD Waveform:



**Code Example**

```
;== 1/32 Duty & 1/6.5 Bias
L32Duty:
;---1/6.5 Bias
       MOV        A,#11110000B
       MOV        LCDCON,A
;--- LCD Off, Normal Display Mode
       MOV        A,#00110000B
       MOV        SPISTA,A
       SCALL  DspRAMdot
;---LCD turn-on
       BS         SPISTA,LCDM1
       LCALL  Delay1sec
        :
DspLoop:
        :
       SJMP       DspLoop

;===Write LCD RAM is dot matrix
WrLRAMd:
       MOV        A,#0X55
       MOV        LCDDATA,A
       MOV        A,#0XAA
       MOV        LCDDATA,A
       JDNZ       CNT_HI,WrLRAMd
       RET
```

```
;*** Display LCD RAM is data 55 & AA
DspRAMdot:
;---LCD increase enable
      MOV        A,#01000100B
      MOV        POST_ID,A
;---LCD page 00
      CLR        LCDARH
DspRAMd1:
      CLR        LCDARL
      MOV        A,#0X20
      MOV        CNT_HI,A
      SCALL      WrLRAMd
      MOV        A,#050H
      MOV        LCDARL,A
      MOV        A,#0X10
      MOV        CNT_HI,A
      SCALL      WrLRAMd
      INC        LCDARH
      MOV        A,LCDARH
      JLE        A,#00000011B,DspRAMd1
      RET
```

# 9 Electrical Characteristic

## 9.1 Absolute Maximum Ratings

| Items | Sym. | Condition | Limits | Unit |
|---|---|---|---|---|
| Supply voltage | VDD | | -0.3 to +3.6 | V |
| Input voltage | VIN | | -0.5 to VDD +0.5 | V |
| Operating temperature range | TOPR | | -10 to +60 | °C |
| Storage temperature range | TSTR | | -55 to +125 | °C |

## 9.2 Recommended Operating Conditions

| Items | Sym. | Condition | Limits | Unit |
|---|---|---|---|---|
| Supply Voltage | VDD | | 2.6 to 3.6 | V |
| Input Voltage | VIH | | VDD*0.9 to VDD | V |
| | VIL | | 0 to VDD*0.1 | V |
| LVDIN Voltage | VLVD | LVDSEL=1 (Internal LVD) | 2.0V~3.6V | V |
| Operating Temperature | TOPR | | -10 to 60 | °C |

## 9.3 DC Electrical Characteristics

Condition: Ta=-10~+60ºC, VDD= 2.9 ± 0.3V

| Parameter | Sym. | Condition | | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|---|
| Clock | Fmain | Main-clock Frequency | | 1 | - | 10 | MHz |
| | Fsub | Sub-clock Frequency | RC OSC | 24.6 | 32.8 | 41 | kHz |
| | | | Crystal OSC | - | 32.768 | - | |
| Supply Current | Idd1 | Sleep mode | VDD=3V, No load (Ta=25°C) | - | - | 1 | μA |
| | Idd2 | Idle mode | VDD=3V, LCD disable, RC OSC | - | 8 | 12 | |
| | Idd3 | | VDD=3V, LCD disable, Crystal OSC | - | 5 | 8 | |
| | Idd4 | | VDD=3V, LCD enable, Auto Scan On, RC/Crystal OSC (Vo=3.8V, 1/3 Bias, V1 and V4 OP=*1, V2 and V3 OP=Off V1 connected to V3, V2 connected to V4, No load) | - | 80 | 100 | |
| | Idd5 | Slow mode | VDD=3V, LCD disable, RC/Crystal OSC, No load | - | 20 | 30 | |
| | Idd6 | Fast mode | VDD=3V, Fmain=4MHz, No load | - | 1200 | 1500 | |
| | Idd7 | | VDD=3V, Fmain=10MHz, No load | - | 2200 | 2750 | |
| Input Voltage | VIH1 | PA[7], PB[0:7] (as general input port) | | VDD×0.7 | - | VDD | V |
| | VIL1 | | | 0 | - | VDD×0.3 | |
| | VIH2 | PA[0:6] (as general input port, LCD enable) | | VDD×0.7 | - | V0 | |
| | VIL2 | | | 0 | - | VDD×0.3 | |
| Input Threshold Voltage (Schmitt) | VT+ | Port A interrupt, System Reset and Reset_Key | | 0.5×VDD | - | 0.75×VDD | V |
| | VT- | | | 0.2×VDD | - | 0.4×VDD | |
| Output Current | IOH1 | PB[0:3] (as general output port) | VDD=3V , VOH=2.4V | -1 | -2 | -3 | mA |
| | IOL1 | | VDD=3V , VOL=0.2V | +1 | +2 | +3 | |
| | IOH2 | PB[6:7] | VDD=3V , VOH=2.5V | -1 | -2 | -3 | |
| | IOL2 | | VDD=3V , VOL=0.5V | +1 | +2 | +3 | |
| | IOH3 | PB[4:5] | VDD=3V, VOH=1V | -3 | -6 | -9 | |
| | IOL3 | | VDD=3V, VOL=0.5V | +1.5 | +3 | +4.5 | |
| Input Leakage Current | IIL | ALL Input port ( without pull up/down resistor) Vin = VDD or GND | | - | - | ±1 | μA |
| Large Pull up Resistance | RPU1 | PA[0:6] | Key high resistance, pulled up by R2 , LCD enable, VDD=3V | 150. | 300 | 450 | KΩ |
| | RPU3 | PA[7], PB[0:7], | Vin=GND | 500 | 1000 | 1500 | |
| | RPU5 | System Reset & Reset_Key | Vin=GND | 200 | 400 | 800 | |

| Parameter | Sym. | Condition | | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|---|
| Small Pull up Resistance | RPU2 | PA[0:6] | Key low resistance, pulled up by R1//R2, LCD enable, VDD=3V | 40 | 75 | 110 | KΩ |
| | RPU4 | PA[7], PB[0:7] | Vin=2V | 50 | 100 | 200 | |
| | RPU6 | System Reset and Reset_Key | Vin=2V | 50 | 100 | 200 | |
| Large Pull-down Resistance | RPD1 | TEST | Vin=VDD | 250 | 500 | 750 | KΩ |
| Small Pull-down Resistance | RPD2 | TEST | Vin=1V | 5 | 10 | 15 | |
| Low Battery Detect Voltage | Vdet | Ta=0 to 40°C | | Vdet –0.1V | Vdet [1] | Vdet +0.1V | V |
| LVD Supply Current | Ilvd | LVDIN=2V | | - | 3 | 5 | μA |
| Data Retention Voltage | Vret | | | 1.6 | - | - | V |
| Power-on Reset Voltage | Vpor | | | 1.4 | 1.5 | 1.6 | V |
| Charge Pump Output | Vout | Capacitance of charge pump C1 and C2:0.1μF | | 2×VX-5% | VX×2 | - | V |
| Regulated Voltage | V0 | VDD=2.3V~3.3V, Ta=0 to 40 °C | | V0-4% | V0 [2] | V0+4% | V |
| Protection Circuit Voltage | VX | | | 2.6 | 2.75 | 2.9 | V |
| Reference voltage | Vref1 | Ta=20 °C [2] | | 2.035 | 2.12 | 2.205 | V |
| | Vref2 | Ta=0 °C [2] | | 2.169 | 2.26 | 2.351 | |
| | Vref3 | Ta=40 °C [2] | | 1.900 | 1.98 | 2.060 | |
| LCD Display Output ON-resistance | ROC | Com [0:31] | VOH=V0 ± 0.2V | - | 2 | 3 | KΩ |
| | | | VOM=V1 ± 0.2V | | | | |
| | | | VOM=V4 ± 0.2V | | | | |
| | | | VOL=0.2V | | | | |
| | ROS | Seg[0:97] | VOH=V0 ± 0.2V | - | 2 | 3 | KΩ |
| | | | VOM=V2 ± 0.2V | | | | |
| | | | VOM=V3 ± 0.2V | | | | |
| | | | VOL=0.2V | | | | |
| Strobe Output ON-resistance | ROP | Seg[0:15] | V=VDD-0.2V | 85 | 165 | 250 | KΩ |
| | RON | | V=0.2V | 0.7 | 1.3 | 2 | |
| Display Frame Frequency | Frame | Sub-Clock: RC OSC | | 57.6 | - | 87.8 | Hz |
| | | Sub-Clock: Crystal OSC | | 64 | - | 78.8 | |
| Op. Amp LCD Voltage Output Power Supply | Vout0 | V0 | No load | – [4] | V0 [3] | – [4] | mV |
| | Vout1 | V1 | | – [4] | V1 [3] | – [4] | |
| | Vout2 | V2 | | – [4] | V2 [3] | – [4] | |
| | Vout3 | V3 | | – [4] | V3 [3] | – [4] | |
| | Vout4 | V4 | | – [4] | V4 [3] | – [4] | |

**Note:** [1]: Typical detected voltage is chosen by software from the following table.

| LVD2~0 | Vdet(Vth-) | Vdet(Vth+) | Deviation | LVD2~0 | Vdet(Vth-) | Vdet(Vth+) | Deviation |
|---|---|---|---|---|---|---|---|
| 000 | 2.2V | 2.3V | ±0.1V | 100 | 2.6V | 2.7V | ±0.1V |
| 001 | 2.3V | 2.4V | | 101 | 2.7V | 2.8V | |
| 010 | 2.4V | 2.5V | | 110 | 2.8V | 2.9V | |
| 011 | 2.5V | 2.6V | | 111 | 2.9V | 3.0V | |

[2]: Typical regulated voltage for V0 is chosen by software from the following equation.

$$VEV = (1+\frac{Rb}{Ra})\times Vref \qquad (\frac{Rb}{Ra} \text{ is external resistor ratio})$$

$$V0 = (1-\frac{(31-\alpha)}{124})\times VEV$$

0.75 Vmax ≤ V0 ≤ Vmax (32 steps)

**NOTE**

*Vref temperature coefficient = -7mV/°C (Refer to Vref1, Vref2, Vref3 DC spec.)*

[3]: V0~V4 Theoretical value

| Bias | V0 | V1 | V2 | V3 | V4 |
|---|---|---|---|---|---|
| 1/3 | V0 | 2/3 × V0 | 1/3 × V0 | 2/3 × V0 | 1/3 × V0 |
| 1/3.5 | | 2.5/3.5 × V0 | 1.5/3.5 × V0 | 2/3.5 × V0 | 1/3.5 × V0 |
| 1/4 | | 3/4 × V0 | 2/4 × V0 | 2/4 × V0 | 1/4 × V0 |
| 1/4.5 | | 3.5/4.5 × V0 | 2.5/4.5 × V0 | 2/4.5 × V0 | 1/4.5 × V0 |
| 1/5 | | 4/5 × V0 | 3/5 × V0 | 2/5 × V0 | 1/5 × V0 |
| 1/5.5 | | 4.5/5.5 × V0 | 3.5/5.5 × V0 | 2/5.5 × V0 | 1/5.5 × V0 |
| 1/6 | | 5/6 × V0 | 4/6 × V0 | 2/6 × V0 | 1/6 × V0 |
| 1/6.5 | | 5.5/6.5 × V0 | 4.5/6.5 × V0 | 2/6.5 × V0 | 1/6.5 × V0 |

[4]: The Target value of V0~V4 is Theoretical value ± 20mV

## 9.4 AC Electrical Characteristics

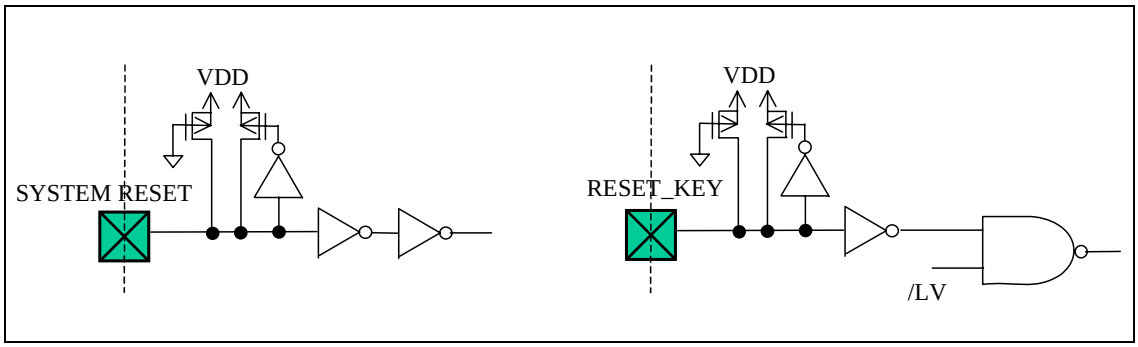
Condition: Ta=-10~+60°C, VDD= 2.9 ± 0.3V

| Parameter | Sym. | Condition | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| LVD Setup Time | Tstup1 | LVDEN↑to /LV↓ (LVDIN=2.0V) | – | 30 | 100 | μs |
| SDI Data Setup Time | Tstup2 | Setup time of SDI data input to SCK↑or SCK↓ | – | 25 | 50 | ns |
| SDI Data Hold Time | Thold2 | Hold time of SDI data input to SCK↓or SCK↑ | – | 25 | 50 | ns |
| SDO Output Valid Time | Tvalid1 | SCK↑or SCK↓to SDO data output | – | 25 | 50 | ns |
| SCK Input High Time | Tsckh | Slave mode (Fmain=10 MHz) | 200 | – | – | ns |
| SCK Input Low Time | Tsckl | Slave mode (Fmain=10 MHz) | 200 | – | – | ns |
| Slave Mode Setup Time | Tsetup3 | /SS↓ to SCK↑or SCK↓ (Fmain=10 MHz) | 400 | – | – | ns |
| Slave Mode Unselect Delay Time | Tdelay1 | /SS↑ to SDO output hi-impedance delay time | – | 25 | 50 | ns |
| PF[7:0]:PE[7:0] Auto Increase / Decrease Delay Time | Tdelay2 | Q4 end of read Port G instruction to Addr[15:0] Auto Increase/Decrease (Fmain=10 MHz) | – | 50 | 100 | ns |
| CK Delay Time | Tdelay3 | CHOP carrier stop to CK↑delay time | 22.9 | 30.5 | 38.2 | μs |
| Latch Write Pulse Low Time | Twpl | Q4 end of read/write Port G instruction to Latch↑(Fmain=10 MHz) | – | 100 | – | ns |
| LATCH Write Pulse High Time | Twph | Latch↑to Q4 end of current instruction (Fmain=10 MHz) | – | 100 | – | ns |
| Instruction Cycle Time | Tcycle | Fmain=1MHz | – | 2* | – | μs |
| | | Fmain=4MHz | – | 0.5* | – | |
| | | Fmain=10MHz | – | 0.2* | – | |

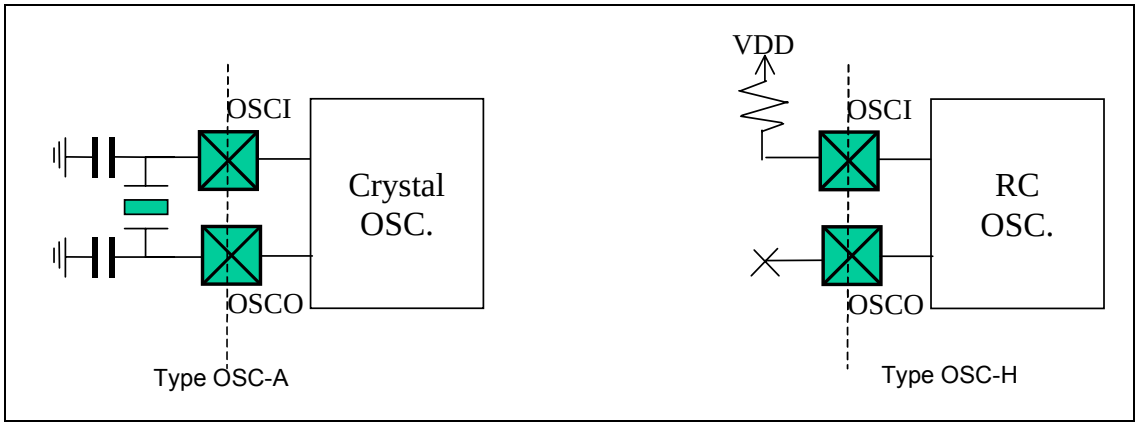
**Note:** *: Instruction cycle time= 2 × System clock time

# 10 Pin Type Circuit Diagrams

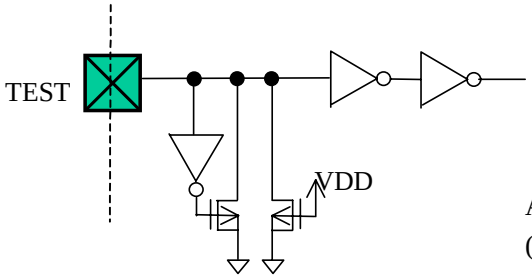
Reset Pin Type



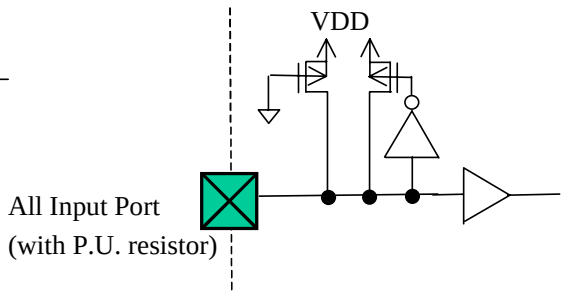

Oscillator Pin Type



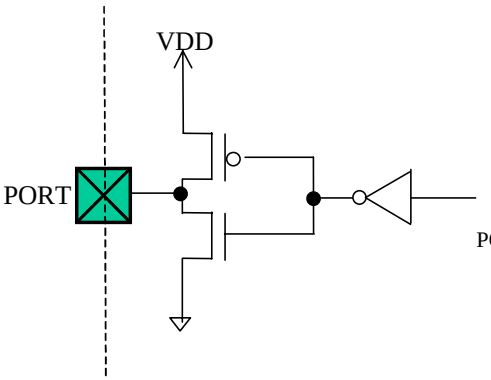
Test Pin Type



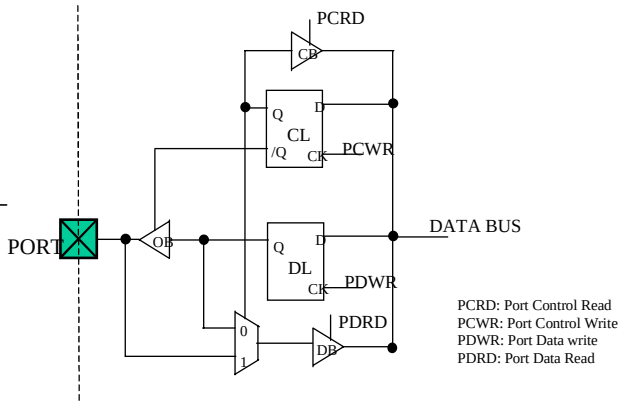

Input Pin Type




Output Pin Type

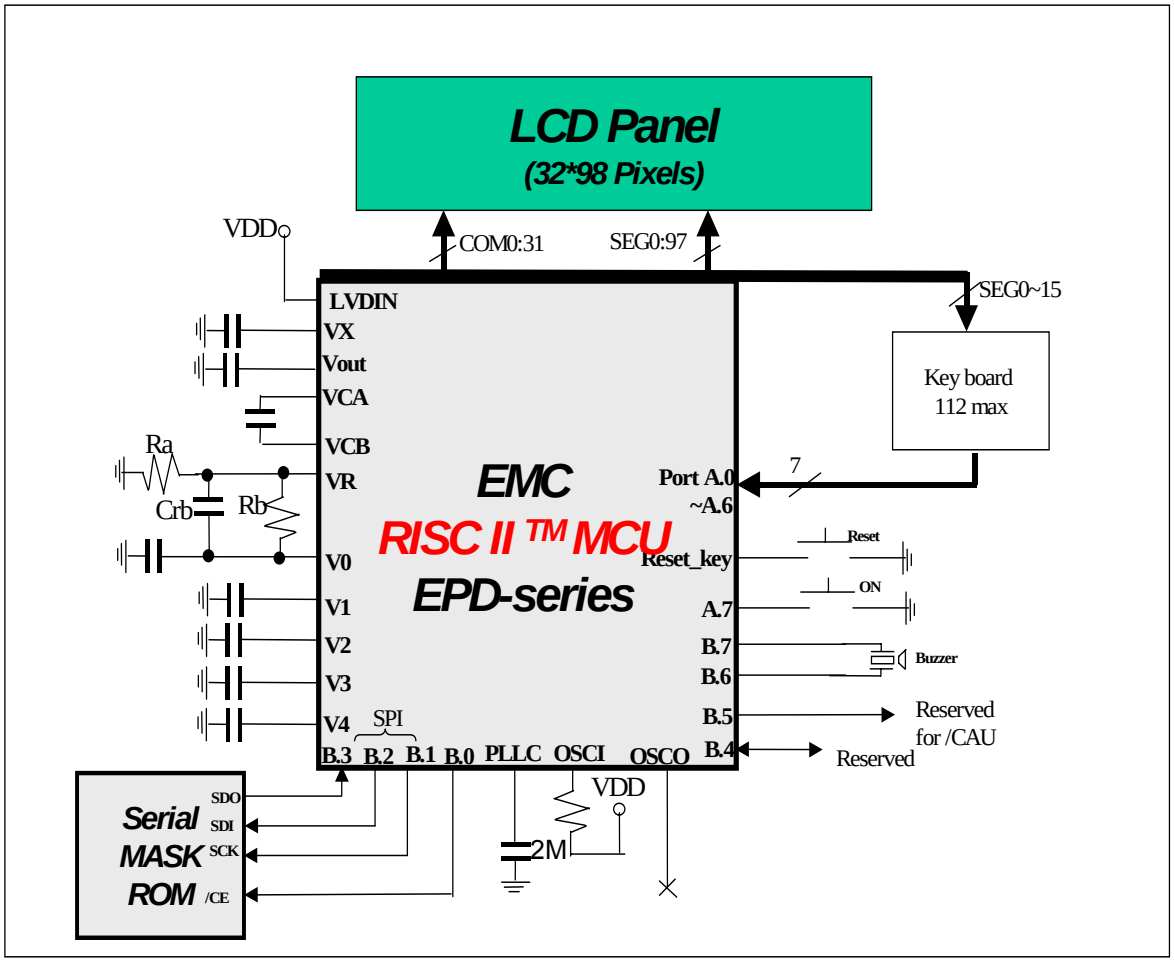



General I/O Pin Type

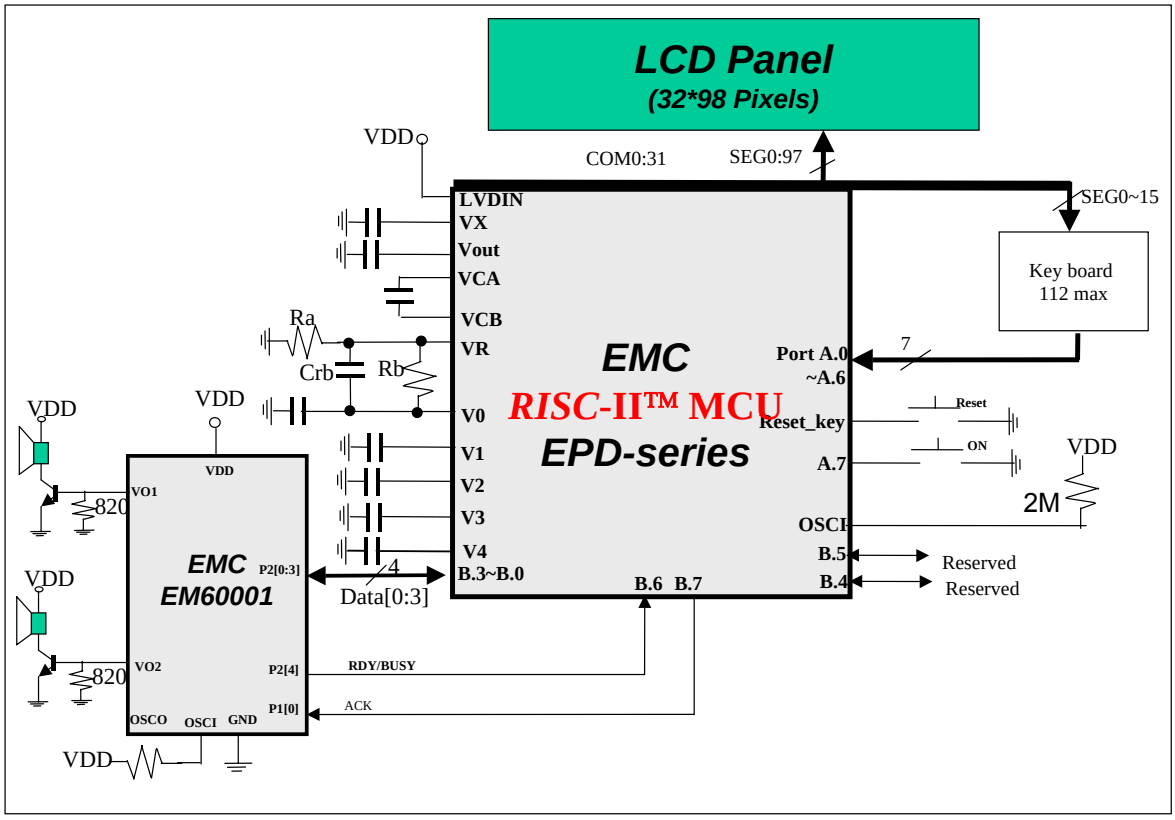


# 11 Application Circuit

(1) Dictionary



(2) ELAN Product



# 12 Instruction Set:

**Legend:** **addr:** address **i:** Table pointer control **p:** special file register (0h~1Fh)
**b:** bit **k:** constant **r:** File Register

| Type | Binary Instruction | Mnemonic | Operation | Status Affected | Cycles |
|---|---|---|---|---|---|
| System Control | 0000 0000 0000 0000 | NOP | No operation | None | 1 |
| | 0000 0000 0000 0001 | WDTC | WDT ← 0; /TO←1; /PD←1 | None | 1 |
| | 0000 0000 0000 0010 | SLEP | Enter Idle Mode if MS1=1<br>Enter Sleep Mode if MS1=0 | None | 1 |
| | 0010 0111 rrrr rrrr | RPT r | Single repeat<br>*(r) times on next instruction.<br>*(r) is the content of register r | None | 1 |
| | 0100 0011 kkkk kkkk | BANK #k | BSR ← k | None | 1 |
| Rom Table Look Up | 0100 0000 kkkk kkkk | TBPTL #k | TABPTRL ← k | None | 1 |
| | 0100 0001 kkkk kkkk | TBPTM #k | TABPTRM ← k | None | 1 |
| | 0100 0010 kkkk kkkk | TBPTH #k | TABPTRH ← k | None | 1 |
| | 0010 11 i i rrrr rrrr | TBRD i,r | r←ROM[(TABPTR)]. **(*1) (*2)** | None | 2 |
| | 0010 1111 rrrr rrrr | TBRD A,r | r←ROM[(TABPTR+ACC)] **(*2)** | None | 2 |
| Data Transfer | 0010 0100 rrrr rrrr | CLR r | r ← 0 | Z | 1 |
| | 0100 1110 kkkk kkkk | MOV A,#k | A ← k | None | 1 |
| | 0010 0000 rrrr rrrr | MOV A,r | A ← r | Z | 1 |
| | 0010 0001 rrrr rrrr | MOV r,A | r ← A | None | 1 |
| | 100p pppp rrrr rrrr | MOVRP p,r | Register p ← Register r | None | 1 |
| | 101p pppp rrrr rrrr | MOVPR r,p | Register r ← Register p | None | 1 |
| Exchange | 0000 1111 rrrr rrrr | SWAP r | r(0:3)←→r(4:7) | None | 1 |
| | 0000 1110 rrrr rrrr | SWAPA r | r(0:3)→A(4:7);r(4:7)→A(0:3) | None | 1 |
| Bit Manipulation | 0110 1bbb rrrr rrrr | BC r,b | r(b) ← 0 | None | 1 |
| | 0111 0bbb rrrr rrrr | BS r,b | r(b) ← 1 | None | 1 |
| | 0111 1bbb rrrr rrrr | BTG r,b | r(b) ← /r(b) | None | 1 |
| Arithmetic Operation | 0001 1100 rrrr rrrr | INCA r | A ← r+1 | C,Z | 1 |
| | 0001 1101 rrrr rrrr | INC r | r ← r+1 | C,Z | 1 |
| | 0001 0000 rrrr rrrr | ADD A,r | A ← A+r | C,DC,Z,OV,SGE,SLE | 1 |
| | 0001 0001 rrrr rrrr | ADD r,A | r ← r+A **(*4)** | C,DC,Z,OV,SGE,SLE | 1 |
| | 0100 1010 kkkk kkkk | ADD A,#k | A ← A+k | C,DC,Z,OV,SGE,SLE | 1 |
| | 0001 0010 rrrr rrrr | ADC A,r | A ← A+r+C | C,DC,Z,OV,SGE,SLE | 1 |
| | 0001 0011 rrrr rrrr | ADC r,A | r ← r+A+C | C,DC,Z,OV,SGE,SLE | 1 |
| | 0100 1011 kkkk kkkk | ADC A,#k | A ← A+k+C | C,DC,Z,OV,SGE,SLE | 1 |
| | 0001 1110 rrrr rrrr | DECA r | A ← r-1 | C,Z | 1 |
| | 0001 1111 rrrr rrrr | DEC r | r← r-1 | C,Z | 1 |
| | 0001 0110 rrrr rrrr | SUB A,r | A ← r-A **(*6)** | C,DC,Z,OV,SGE,SLE | 1 |
| | 0001 0111 rrrr rrrr | SUB r,A | r ← r-A **(*6)** | C,DC,Z,OV,SGE,SLE | 1 |
| | 0100 1100 kkkk kkkk | SUB A,#k | A ← k-A **(*6)** | C,DC,Z,OV,SGE,SLE | 1 |
| | 0001 1000 rrrr rrrr | SUBB A,r | A ← r-A-/C **(*6)** | C,DC,Z,OV,SGE,SLE | 1 |
| | 0001 1001 rrrr rrrr | SUBB r,A | r ← r-A-/C **(*6)** | C,DC,Z,OV,SGE,SLE | 1 |
| | 0100 1101 kkkk kkkk | SUBB A,#k | A ← k-A-/C **(*6)** | C,DC,Z,OV,SGE,SLE | 1 |

**Legend:** **addr:** address **i:** Table pointer control **p:** special file register (0h~1Fh)
**b:** bit **k:** constant **r:** File Register

| Type | Instruction Binary | Mnemonic | Operation | Status Affected | Cycles |
|---|---|---|---|---|---|
| Arithmetic Operation | 0010 0110 rrrr rrrr | MUL A,r | PRODH:PRODL ← A*r | None | 1 |
| | 0100 1111 kkkk kkkk | MUL A,#k | PRODH:PRODL ← A*k | None | 1 |
| | 0001 0100 rrrr rrrr | ADDDC A,r | A ← (Decimal ADD) A+r+C | C, DC, Z | 1 |
| | 0001 0101 rrrr rrrr | ADDDC r,A | r ← (Decimal ADD) r+A+C | C, DC, Z | 1 |
| | 0001 1010 rrrr rrrr | SUBDB A,r | A ← (Decimal SUB) r-A-/C | C, DC, Z | 1 |
| | 0001 1011 rrrr rrrr | SUBDB r,A | r ← (Decimal SUB) r-A-/C | C, DC, Z | 1 |
| Logic Operation | 0000 0010 rrrr rrrr | OR A,r | A ← A or r | Z | 1 |
| | 0000 0011 rrrr rrrr | OR r,A | r ← r .or A | Z | 1 |
| | 0100 0100 kkkk kkkk | OR A,#k | A ← A or k | Z | 1 |
| | 0000 0100 rrrr rrrr | AND A,r | A ← A and r | Z | 1 |
| | 0000 0101 rrrr rrrr | AND r,A | r ← r .and A | Z | 1 |
| | 0100 0101 kkkk kkkk | AND A,#k | A ← A .and k | Z | 1 |
| | 0000 0110 rrrr rrrr | XOR A,r | A ← A xor r | Z | 1 |
| | 0000 0111 rrrr rrrr | XOR r,A | r ← r xor A | Z | 1 |
| | 0100 0110 kkkk kkkk | XOR A,#k | A ← A xor k | Z | 1 |
| | 0000 1000 rrrr rrrr | COMA r | A ← /r | Z | 1 |
| | 0000 1001 rrrr rrrr | COM r | r ← /r | Z | 1 |
| Rotate | 0000 1010 rrrr rrrr | RRCA r | A(n-1) ←r(n);C ←r(0);A(7) ←C | C | 1 |
| | 0000 1011 rrrr rrrr | RRC r | r(n-1) ←r(n);C ←r(0);r(7) ←C | C | 1 |
| | 0000 1100 rrrr rrrr | RLCA r | A(n+1) ←r(n);C ←r(7);A(0) ←C | C | 1 |
| | 0000 1101 rrrr rrrr | RLC r | r(n+1) ←r(n);C ←r(7);r(0) ←C | C | 1 |
| Shift | 0010 0010 rrrr rrrr | SHRA r | A(n-1)←r(n); A(7)←C | None | 1 |
| | 0010 0011 rrrr rrrr | SHLA r | A(n+1)←r(n); A(0)←C | None | 1 |
| Bit Compare & Jump | 0101 1bbb rrrr rrrr aaaa aaaa aaaa aaaa | JBC r,b,addr | If r(b)=0,jump to addr; PC[15:0] ← addr **(*3)** | None | 2 |
| | 0110 0bbb rrrr rrrr aaaa aaaa aaaa aaaa | JBS r,b,addr | If r(b)=1,jump to addr; PC[15:0] ← addr **(*3)** | None | 2 |
| Compare | 0010 0101 rrrr rrrr | TEST r | Z ← 0 if r<>0;Z ← 1 if r=0 | Z | 1 |
| Compare and Jump | 0101 0000 rrrr rrrr aaaa aaaa aaaa aaaa | JDNZ A,r,addr | A ← r-1, jump to addr if not zero; PC [15:0] ← addr **(*3)** | None | 2 |
| | 0101 0001 rrrr rrrr aaaa aaaa aaaa aaaa | JDNZ r,addr | r ← r-1, jump to addr if not zero; PC [15:0] ← addr **(*3)** | None | 2 |
| | 0101 0010 rrrr rrrr aaaa aaaa aaaa aaaa | JINZ A,r,addr | A ← r+1,jump to addr if not zero; PC[15:0] ← addr **(*3)** | None | 2 |
| | 0101 0011 rrrr rrrr aaaa aaaa aaaa aaaa | JINZ r,addr | r ← r+1,jump to addr if not zero; PC[15:0] ← addr **(*3)** | None | 2 |
| | 0100 0111 kkkk kkkk aaaa aaaa aaaa aaaa | JGE A,#k,addr | Jump to addr if A ≥ k; PC [15:0] ← addr **(*3)** | None | 2 |
| | 0100 1000 kkkk kkkk aaaa aaaa aaaa aaaa | JLE A,#k,addr | Jump to addr if A ≤ k; PC [15:0] ← addr **(*3)** | None | 2 |
| | 0100 1001 kkkk kkkk aaaa aaaa aaaa aaaa | JE A,#k,addr | Jump to addr if A=k; PC[15:0] ← addr **(*3)** | None | 2 |

**Legend:** **addr:** address **i:** Table pointer control **p:** special file register (0h~1Fh)
**b:** bit **k:** constant **r:** File Register

| Type | Binary Instruction | Mnemonic | Operation | Status Affected | Cycles |
|---|---|---|---|---|---|
| Compare and Jump | 0101 0101 rrrr rrrr aaaa aaaa aaaa aaaa | JGE A,r,addr | Jump to addr if A ≥ r; PC[15:0] ← addr **(*3)** | None | 2 |
| | 0101 0110 rrrr rrrr aaaa aaaa aaaa aaaa | JLE A,r,addr | Jump to addr if A ≤ r; PC[15:0] ← addr **(*3)** | None | 2 |
| | 0101 0111 rrrr rrrr aaaa aaaa aaaa aaaa | JE A,r,addr | Jump to addr if A=r; PC[15:0] ← addr **(*3)** | None | 2 |
| Jump | 110a aaaa aaaa aaaa | SJMP addr | PC ← addr; PC [13..16] unchanged | None | 1 |
| | 0000 0000 0010 aaaa aaaa aaaa aaaa aaaa | LJMP addr (2 words) | PC ← addr | None | 2 |
| Subroutine | 0011 aaaa aaaa aaaa | S0CALL addr | Top of Stack] ← PC+1; PC [11:0] ← addr; PC [12:16] ← 00000 **(*5)** | None | 1 |
| | 111a aaaa aaaa aaaa | SCALL addr | [Top of Stack] ← PC+1; PC [12:0] ← addr; PC [13:16] unchanged | None | 1 |
| | 0000 0000 0011 aaaa aaaa aaaa aaaa aaaa | LCALL addr (2 words) | [Top of Stack] ← PC+1; PC ← addr | None | 2 |
| | 0010 1011 1111 1110 | RET | PC ← (Top of Stack) | None | 1 |
| | 0010 1011 1111 1111 | RETI | PC ← (Top of Stack); Enable Interrupt | None | 1 |

**(*1)** TBRD i, r:
r← ROM [(TABPTR)];
i=00: TABPTR no change
i=01: TABPTR←TABPTR+1
i=10: TABPTR←TABPTR-1

**(*2)** TABPTR=(TABPTRH: TABPTRM: TABPTRL)
Bit 0 = 0**:** Low byte of the pointed ROM data
Bit 0 = 1**:** High byte of the pointed ROM data
The maximum table look up space is internal 8Mbytes.

**(*3)** The maximum jump range is 64K absolute address, which means it can only jump within the same 64K range.

**(*4)** Carry bit of ADD PCL, A or ADD TABPTRL, A will automatically carry into PCM or TABPTRM. The Instruction cycle for writing to the PC (program counter) takes 2 cycles.

**(*5)** S0CALL address ability is from 0x000 to 0xFFF (4K space).

**(*6)** When in SUB operation, borrow flag is indicated by the inverse of carry bit, i.e., B = /C.

# 13 Pad Diagram

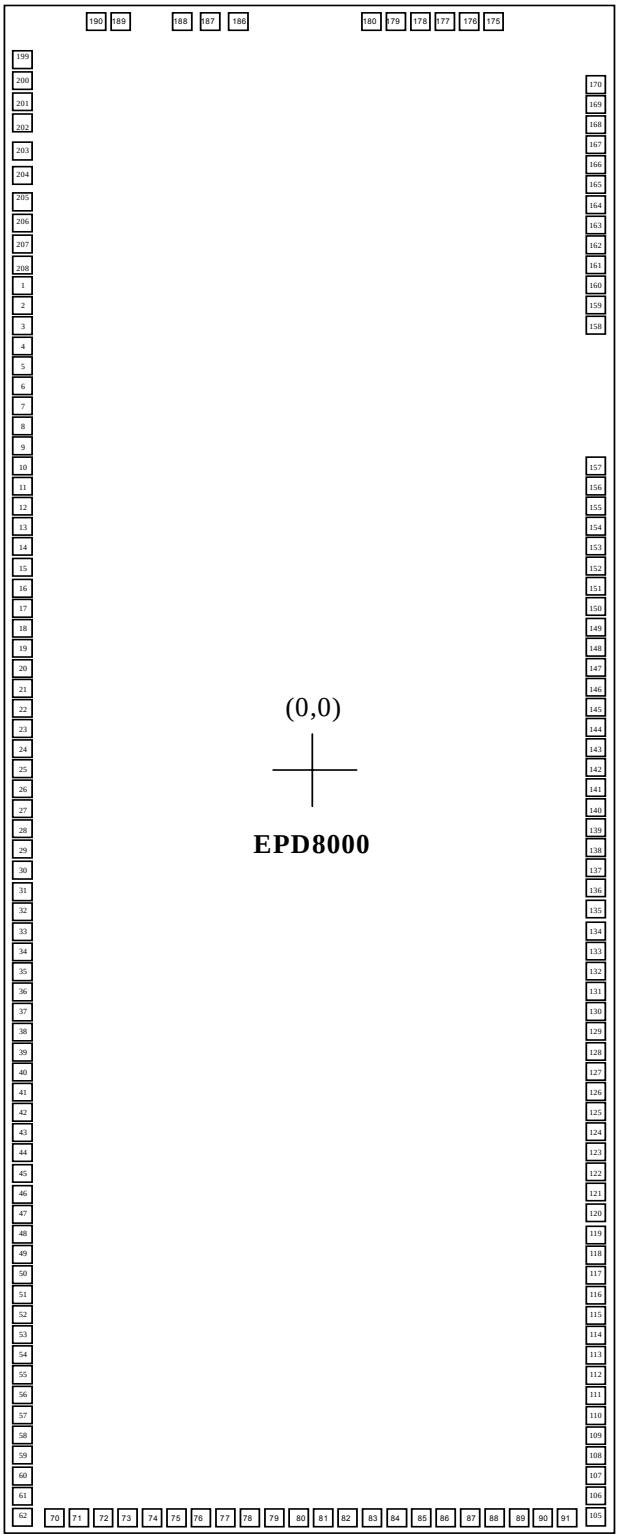



*Figure 13 EPD8000 Pad Locations*

**Chip size: 2810 * 8110 μm**$^2$

### Pad Coordinates

| Pin No. | Symbol | X | Y | Pin No. | Symbol | X | Y |
|---|---|---|---|---|---|---|---|
| 1 | COM31 | -1270.0 | 2505.0 | 105 | SEG29 | 1270.0 | -3920.0 |
| 2 | COM30 | -1270.0 | 2400.0 | 106 | SEG28 | 1270.0 | -3810.0 |
| 3 | COM29 | -1270.0 | 2295.0 | 107 | SEG27 | 1270.0 | -3700.0 |
| 4 | COM28 | -1270.0 | 2190.0 | 108 | SEG26 | 1270.0 | -3590.0 |
| 5 | COM27 | -1270.0 | 2085.0 | 109 | SEG25 | 1270.0 | -3480.0 |
| 6 | COM26 | -1270.0 | 1980.0 | 110 | SEG24 | 1270.0 | -3375.0 |
| 7 | COM25 | -1270.0 | 1875.0 | 111 | SEG23 | 1270.0 | -3270.0 |
| 8 | COM24 | -1270.0 | 1770.0 | 112 | SEG22 | 1270.0 | -3165.0 |
| 9 | COM23 | -1270.0 | 1665.0 | 113 | SEG21 | 1270.0 | -3060.0 |
| 10 | COM22 | -1270.0 | 1560.0 | 114 | SEG20 | 1270.0 | -2955.0 |
| 11 | COM21 | -1270.0 | 1455.0 | 115 | SEG19 | 1270.0 | -2850.0 |
| 12 | COM20 | -1270.0 | 1350.0 | 116 | SEG18 | 1270.0 | -2745.0 |
| 13 | COM19 | -1270.0 | 1245.0 | 117 | SEG17 | 1270.0 | -2640.0 |
| 14 | COM18 | -1270.0 | 1140.0 | 118 | SEG16 | 1270.0 | -2535.0 |
| 15 | COM17 | -1270.0 | 1035.0 | 119 | SEG15 | 1270.0 | -2430.0 |
| 16 | COM16 | -1270.0 | 930.0 | 120 | SEG14 | 1270.0 | -2325.0 |
| 17 | SEG97 | -1270.0 | 825.0 | 121 | SEG13 | 1270.0 | -2220.0 |
| 18 | SEG96 | -1270.0 | 720.0 | 122 | SEG12 | 1270.0 | -2115.0 |
| 19 | SEG95 | -1270.0 | 615.0 | 123 | SEG11 | 1270.0 | -2010.0 |
| 20 | SEG94 | -1270.0 | 510.0 | 124 | SEG10 | 1270.0 | -1905.0 |
| 21 | SEG93 | -1270.0 | 405.0 | 125 | SEG9 | 1270.0 | -1800.0 |
| 22 | SEG92 | -1270.0 | 300.0 | 126 | SEG8 | 1270.0 | -1695.0 |
| 23 | SEG91 | -1270.0 | 195.0 | 127 | SEG7 | 1270.0 | -1590.0 |
| 24 | SEG90 | -1270.0 | 90.0 | 128 | SEG6 | 1270.0 | -1485.0 |
| 25 | SEG89 | -1270.0 | -15.0 | 129 | SEG5 | 1270.0 | -1380.0 |
| 26 | SEG88 | -1270.0 | -120.0 | 130 | SEG4 | 1270.0 | -1275.0 |
| 27 | SEG87 | -1270.0 | -225.0 | 131 | SEG3 | 1270.0 | -1170.0 |
| 28 | SEG86 | -1270.0 | -330.0 | 132 | SEG2 | 1270.0 | -1065.0 |
| 29 | SEG85 | -1270.0 | -435.0 | 133 | SEG1 | 1270.0 | -960.0 |
| 30 | SEG84 | -1270.0 | -540.0 | 134 | SEG0 | 1270.0 | -855.0 |
| 31 | SEG83 | -1270.0 | -645.0 | 135 | COM0 | 1270.0 | -750.0 |
| 32 | SEG82 | -1270.0 | -750.0 | 136 | COM1 | 1270.0 | -645.0 |
| 33 | SEG81 | -1270.0 | -855.0 | 137 | COM2 | 1270.0 | -540.0 |
| 34 | SEG80 | -1270.0 | -960.0 | 138 | COM3 | 1270.0 | -435.0 |
| 35 | SEG79 | -1270.0 | -1065.0 | 139 | COM4 | 1270.0 | -330.0 |
| 36 | SEG78 | -1270.0 | -1170.0 | 140 | COM5 | 1270.0 | -225.0 |
| 37 | SEG77 | -1270.0 | -1275.0 | 141 | COM6 | 1270.0 | -120.0 |
| 38 | SEG76 | -1270.0 | -1380.0 | 142 | COM7 | 1270.0 | -15.0 |
| 39 | SEG75 | -1270.0 | -1485.0 | 143 | COM8 | 1270.0 | 90.0 |
| 40 | SEG74 | -1270.0 | -1590.0 | 144 | COM9 | 1270.0 | 195.0 |

| Pin No. | Symbol | X | Y | Pin No. | Symbol | X | Y |
|---|---|---|---|---|---|---|---|
| 41 | SEG73 | -1270.0 | -1695.0 | 145 | COM10 | 1270.0 | 300.0 |
| 42 | SEG72 | -1270.0 | -1800.0 | 146 | COM11 | 1270.0 | 405.0 |
| 43 | SEG71 | -1270.0 | -1905.0 | 147 | COM12 | 1270.0 | 510.0 |
| 44 | SEG70 | -1270.0 | -2010.0 | 148 | COM13 | 1270.0 | 615.0 |
| 45 | SEG69 | -1270.0 | -2115.0 | 149 | COM14 | 1270.0 | 720.0 |
| 46 | SEG68 | -1270.0 | -2220.0 | 150 | COM15 | 1270.0 | 825.0 |
| 47 | SEG67 | -1270.0 | -2325.0 | 151 | PA.0 | 1270.0 | 937.5 |
| 48 | SEG66 | -1270.0 | -2430.0 | 152 | PA.1 | 1270.0 | 1057.5 |
| 49 | SEG65 | -1270.0 | -2535.0 | 153 | PA.2 | 1270.0 | 1177.5 |
| 50 | SEG64 | -1270.0 | -2640.0 | 154 | PA.3 | 1270.0 | 1297.5 |
| 51 | SEG63 | -1270.0 | -2745.0 | 155 | PA.4 | 1270.0 | 1417.5 |
| 52 | SEG62 | -1270.0 | -2850.0 | 156 | PA.5 | 1270.0 | 1537.5 |
| 53 | SEG61 | -1270.0 | -2955.0 | 157 | PA.6 | 1270.0 | 1657.5 |
| 54 | SEG60 | -1270.0 | -3060.0 | 158 | PA.7 | 1270.0 | 2183.3 |
| 55 | SEG59 | -1270.0 | -3165.0 | 159 | CLKO | 1270.0 | 2303.4 |
| 56 | SEG58 | -1270.0 | -3270.0 | 160 | PB.0 | 1270.0 | 2423.4 |
| 57 | SEG57 | -1270.0 | -3375.0 | 161 | PB.1 | 1270.0 | 2543.3 |
| 58 | SEG56 | -1270.0 | -3480.0 | 162 | PB.2 | 1270.0 | 2663.4 |
| 59 | SEG55 | -1270.0 | -3590.0 | 163 | PB.3 | 1270.0 | 2783.3 |
| 60 | SEG54 | -1270.0 | -3700.0 | 164 | PB.4 | 1270.0 | 2903.4 |
| 61 | SEG53 | -1270.0 | -3810.0 | 165 | PB.5 | 1270.0 | 3023.4 |
| 62 | SEG52 | -1270.0 | -3920.0 | 166 | PB.6 | 1270.0 | 3143.3 |
| 63 | NC | | | 167 | PB.7 | 1270.0 | 3263.4 |
| 64 | NC | | | 168 | Reset-Key | 1270.0 | 3383.3 |
| 65 | NC | | | 169 | System Reset | 1270.0 | 3503.4 |
| 66 | NC | | | 170 | VDD | 1270.0 | 3623.4 |
| 67 | NC | | | 171 | NC | | |
| 68 | NC | | | 172 | NC | | |
| 69 | NC | | | 173 | NC | | |
| 70 | SEG51 | -1125.0 | -3920.0 | 174 | NC | | |
| 71 | SEG50 | -1015.0 | -3920.0 | 175 | LVDIN | 1190.7 | 3920.0 |
| 72 | SEG49 | -905.0 | -3920.0 | 176 | Reserved | 1058.7 | 3920.0 |
| 73 | SEG48 | -795.0 | -3920.0 | 177 | Reserved | 938.7 | 3920.0 |
| 74 | SEG47 | -685.0 | -3920.0 | 178 | Reserved | 818.7 | 3920.0 |
| 75 | SEG46 | -580.0 | -3920.0 | 179 | Reserved | 698.7 | 3920.0 |
| 76 | SEG45 | -475.0 | -3920.0 | 180 | Reserved | 578.7 | 3920.0 |
| 77 | SEG44 | -370.0 | -3920.0 | 181 | NC | | |
| 78 | SEG43 | -265.0 | -3920.0 | 182 | NC | | |
| 79 | SEG42 | -160.0 | -3920.0 | 183 | NC | | |
| 80 | SEG41 | -55.0 | -3920.0 | 184 | NC | | |

| Pin No. | Symbol | X | Y | Pin No. | Symbol | X | Y |
|---|---|---|---|---|---|---|---|
| 81 | SEG40 | 50.0 | -3920.0 | 185 | NC | | |
| 82 | SEG39 | 155.0 | -3920.0 | 186 | PLLC | -223.7 | 3920.0 |
| 83 | SEG38 | 260.0 | -3920.0 | 187 | GND | -370.7 | 3920.0 |
| 84 | SEG37 | 365.0 | -3920.0 | 188 | OSCO | -510.7 | 3920.0 |
| 85 | SEG36 | 470.0 | -3920.0 | 189 | OSCI | -821.7 | 3920.0 |
| 87 | SEG34 | 685.0 | -3920.0 | 191 | NC | | |
| 88 | SEG33 | 795.0 | -3920.0 | 192 | NC | | |
| 89 | SEG32 | 905.0 | -3920.0 | 193 | NC | | |
| 90 | SEG31 | 1015.0 | -3920.0 | 194 | NC | | |
| 91 | SEG30 | 1125.0 | -3920.0 | 195 | NC | | |
| 92 | NC | | | 196 | NC | | |
| 93 | NC | | | 197 | NC | | |
| 94 | NC | | | 198 | NC | | |
| 95 | NC | | | 199 | VCA | -1270.0 | 3702.7 |
| 96 | NC | | | 200 | VX | -1270.0 | 3592.7 |
| 97 | NC | | | 201 | V4 | -1270.0 | 3482.7 |
| 98 | NC | | | 202 | VR | -1270.0 | 3372.7 |
| 99 | NC | | | 203 | V0 | -1270.0 | 3223.5 |
| 100 | NC | | | 204 | VOUT | -1270.0 | 3074.2 |
| 101 | NC | | | 205 | V1 | -1270.0 | 2925.0 |
| 102 | NC | | | 206 | V2 | -1270.0 | 2820.0 |
| 103 | NC | | | 207 | V3 | -1270.0 | 2715.0 |
| 104 | NC | | | 208 | VCB | -1270.0 | 2610.0 |

**Note:** Pads 176~180 are for factory testing only. Do Not bond these pads on the Application Circuit

For PCB layout, the IC substrate must be connected to VSS.