

---

**ePS6009**

**RISC II  
Series MCU**

**Product  
Specification**

**DOC. VERSION 1.1**

**ELAN MICROELECTRONICS CORP.**

April 2012

---



### 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.

Copyright © 2010~2012 by **ELAN Microelectronics Corporation**

**All Rights Reserved**

Printed in Taiwan

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 1<sup>st</sup> Road  
Hsinchu Science Park  
Hsinchu, TAIWAN 30076  
Tel: +886 3 563-9977  
Fax: +886 3 563-9966  
[webmaster@emc.com.tw](mailto:webmaster@emc.com.tw)  
<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

### Korea:

#### **ELAN Korea Electronics Company, Ltd.**

301 Dong-A Building  
632 Kojan-Dong, Namdong-ku  
Incheon City, KOREA  
Tel: +82 32 814-7730  
Fax: +82 32 813-7730

### Shenzhen:

#### **ELAN Microelectronics Shenzhen, Ltd.**

8A Floor, Microprofit Building  
Gaoxin South Road 6  
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](mailto:elan-sz@elanic.com.cn)

### Shanghai:

#### **ELAN Microelectronics Shanghai, Ltd.**

Rm101, #3 Lane 289, Bisheng Rd..  
Zhangjiang Hi-Tech Park  
Pudong New Area, Shanghai,  
CHINA 201204  
Tel: +86 21 5080-3866  
Fax: +86 21 5080-0273  
[elan-sh@elanic.com.cn](mailto:elan-sh@elanic.com.cn)

# Contents

<b>1</b>	<b>General Description</b> .....	<b>1</b>
1.1	Application .....	1
<b>2</b>	<b>Features</b> .....	<b>1</b>
2.1	MCU .....	1
2.2	Peripheral .....	2
2.3	Internal Specification .....	2
<b>3</b>	<b>Block Diagram</b> .....	<b>3</b>
<b>4</b>	<b>Pin Assignment</b> .....	<b>4</b>
<b>5</b>	<b>Pin Description</b> .....	<b>5</b>
5.1	MCU System Pins (6 Pins) .....	5
5.2	Embedded LCD Pins (77 Pins) .....	5
5.3	I/O Port (7 Pins) .....	5
<b>6</b>	<b>Code Option</b> .....	<b>6</b>
<b>7</b>	<b>Function Description</b> .....	<b>6</b>
7.1	Reset Function .....	6
7.1.1	Power-on Reset .....	6
7.1.2	RSTB Pin .....	7
7.1.3	WDT Time-out .....	7
7.1.4	Status (R0Fh) .....	7
7.1.5	Initialization after Reset .....	9
7.1.5.1	Special Register .....	9
7.1.5.2	Control Register .....	9
7.2	Oscillator System .....	10
7.2.1	32.8kHz Internal RC or 32.8kHz External RC Oscillator .....	10
7.2.2	200kHz/300kHz/500kHz RC External Oscillator .....	11
7.3	MCU Operation Mode .....	11
7.3.1	Slow, Fast, Sleep, and Idle Mode Operation .....	12
7.3.2	Wake-up Operation .....	14
7.4	Interrupts .....	14
7.4.1	Global Interrupt .....	15
7.4.2	Input Port (Port A.7, Port A.3 ~ Port A.0) Interrupt .....	15
7.4.3	Timer 0, Timer 1, and Timer 2 Interrupts .....	16
7.4.3.1	Timer 0 Interrupt .....	16
7.4.3.2	Timer 1 Interrupt .....	16
7.4.3.3	Timer 2 Interrupt .....	17
7.4.3.4	Code Example: .....	18
7.5	Program ROM Map .....	18



7.6	RAM Map for Special and Control Registers .....	19
7.6.1	Special and Control Registers .....	19
7.6.2	Other Unbanked General RAM .....	21
7.6.3	Banked General RAM .....	21
7.7	LCD RAM Map .....	21
7.8	Special Function Registers .....	21
7.8.1	ACC (R0Ah): Accumulator .....	21
7.8.2	POST_ID (R30h): Post Increase / Decrease Control Register .....	21
7.8.3	BSR, FSR0, INDF0 (R02h, R01h, R00h): Indirect Address Pointer 0 Registers .....	22
7.8.4	BSR1, FSR1, INDF1 (R05h, R04h, R03h): Indirect Address Pointer 1 Registers .....	22
7.8.5	STKPTR (R06h): Stack Pointer Register .....	24
7.8.6	PCL, PCM (R07h, R08h): Program Counter Registers .....	24
7.8.7	TABPTRL, TABPTRM, (R0Bh, R0Ch): Table Pointer Registers .....	25
7.8.8	Port A, Port B (R10h, R11h): General I/O Pin Registers .....	25
7.8.9	STBCON (R20): Strobe Output Control Register .....	25
7.8.10	PACON (R29h): Port A Control Register .....	26
7.8.11	PAWAKE (R2Ah): Port A Wake-up Control Register .....	26
7.8.12	PAINTEN (R2Bh): Port A Interrupt Enable Control Register .....	27
7.8.13	PAINTSTA (R2Ch): Port A Interrupt Status Register .....	27
7.8.14	PBCON (R2Dh): Port B Pull up Resistor Control Register .....	27
7.8.15	DCRB (R2Eh): Port B Direction Control Register .....	27
<b>8</b>	<b>Peripheral .....</b>	<b>28</b>
8.1	Timer 0 (16 Bits Timer with Event Counter Function) .....	28
8.1.1	Timer 0 Registers .....	28
8.2	Timer 1 (8 Bits) .....	30
8.2.1	Timer 1 Registers .....	31
8.3	Timer 2 (8 Bits) .....	33
8.3.1	Timer 2 Registers .....	33
8.4	Watchdog Timer (WDT) .....	35
8.4.1	Watchdog Timer (WDT) Registers .....	36
8.5	Input/Output Key .....	37
8.5.1	Key Functions .....	38
8.5.2	Key Strobe .....	39
8.5.2.1	Automatic Key Scan .....	39
8.5.2.2	Software Key Scan .....	40
8.5.2.3	Key Strobe Pin Function .....	41
8.5.3	Input/Output Key Registers .....	41



8.6	LCD Driver.....	44
8.6.1	LCD Driver Registers .....	45
8.6.2	LCD RAM Map .....	48
8.6.3	LCD Driving Method Circuit.....	48
8.6.4	LCD COM Waveforms.....	49
8.6.5	LCD COM and SEG Waveforms .....	50
<b>9</b>	<b>Electrical Characteristics .....</b>	<b>51</b>
<b>10</b>	<b>Pin Type Circuit Diagrams.....</b>	<b>53</b>
<b>11</b>	<b>Application Circuit .....</b>	<b>55</b>
<b>12</b>	<b>Instruction Set .....</b>	<b>56</b>
<b>13</b>	<b>Pad Diagram .....</b>	<b>59</b>
<b>14</b>	<b>Pad Coordinates.....</b>	<b>60</b>

### Specification Revision History

Doc. Version	Revision Description	Date
1.0	Initial Release of Official Version	2010/10/21
1.1	Modified the Block Diagram Modified the Fast mode current	2012/04/03

---

# 1 General Description

The **ePS6009** is an 8-bit RISC MCU which embedded 10\*60 LCD driver along with two 8-bit timers, one 16-bit general timer, and a watchdog timer. It also has on-chip 2K bytes RAM and 32K words program ROM. It is highly ideal for advance scientific calculator application, particularly those requiring high performance and low cost solution.

The MCU core is one of ELAN's second generation RISC based IC's, known as RISC II (RII) series. The core was specifically designed for low power and portable device applications. The ePS6009 supports Fast, Slow, and Idle modes, as well as Sleep mode to enhance its low power consumption features.

### IMPORTANT NOTES

- Do not use Register BSR1 (07h) Bit 7 ~ Bit 6.
- Check the ranges of BSR and BSR1, which should be from 0x00 ~ 0x0F.
- Do not use LCD RAM 3Ch ~ 3Fh.
- Do not use Register JDNZ at FSR1 (04h) special register.

## 1.1 Application

- Scientific calculating gadgets

---

# 2 Features

## 2.1 MCU

- 8 bit RISC MCU
- Operating voltage and speed: 1.2V~1.8V
- Clock Source: Dual system clock
  - Low-frequency: 32kHz Internal RC oscillator / Crystal oscillator
  - High-frequency: 200kHz / 300kHz / 500kHz External RC oscillator
- One Instruction cycle time = 2 × System clock time
- Program ROM addressing: Maximum 32K words
- 128 bytes un-banked RAM including special registers and common registers
- 16×128 bytes banked RAM
- Max. of 32-level RAM stack
- Table Lookup function is fast and highly efficient when combined with Repeat instruction

- Register-to-Register move instruction
- Compare and Branch in one instruction (2 cycles)
- Single Repeat function (256 repeat times max.)
- Decimal ADD & SUB instruction
- Full range Call and Jump ability (2 cycles)

## **2.2 Peripheral**

- 5 input pins (Port A.0~3, Port A.7) and 2 general I/O pins (Port B.0~1)
- 10 COM × 60 SEG LCD driver (embedded)
- One 16-bit timer (Timer 0) with event counter function
- One 8-bit timer (Timer 1) with wake-up function
- One 8-bit timer (Timer 2)
- One 8-bit Watchdog Timer
- Key I/O function with 64 keys max (Key matrix: Port A.0~3 and SEG0~15 with automatic key scan)

## **2.3 Internal Specification**

- Watchdog Timer with its own on-chip RC oscillator
- MCU operating modes: Sleep Mode, Idle Mode, Slow Mode, and Fast Mode
- Supports RC and Crystal oscillation for system clock
- MCU wake-up function consists of input wake up and Timer 1 wake up
- MCU interrupt function consist of Input port interrupt and Timer interrupt (Timers 0 ~ 2)
- MCU reset function includes power-on reset, RSTB pin reset, and Watchdog timer reset

### 3 Block Diagram

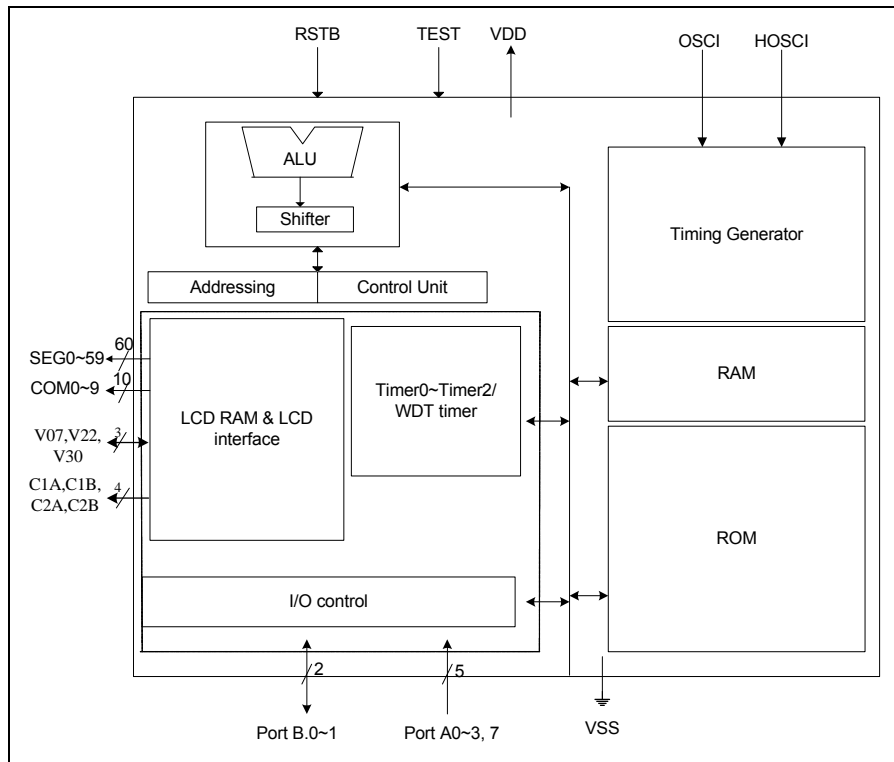


Figure 3-1 ePS6009 Block Diagram

## 4 Pin Assignment

■ 90-pin Chip form

No.	Pin NAME	No.	Pin NAME	No.	Pin NAME	No.	Pin NAME
1	NC	41	NC	81	NC	121	NC
2	NC	42	NC	82	SEG43	122	NC
3	NC	43	NC	83	SEG42	123	NC
4	NC	44	NC	84	SEG41	124	NC
5	COM1	45	NC	85	SEG40	125	NC
6	COM0	46	NC	86	SEG39	126	NC
7	C1A	47	NC	87	SEG38	127	NC
8	C1B	48	NC	88	SEG37	128	NC
9	C2A	49	NC	89	SEG36	129	NC
10	C2B	50	NC	90	SEG35	130	NC
11	V30	51	NC	91	SEG34	131	NC
12	V22	52	NC	92	SEG33	132	NC
13	V07	53	NC	93	SEG32	133	SEG7/Strobe7
14	NC	54	NC	94	SEG31	134	SEG6/Strobe6
15	NC	55	SEG56	95	SEG30	135	SEG5/Strobe5
16	PortA.0	56	SEG55	96	SEG29	136	SEG4/Strobe4
17	PortA.1	57	SEG54	97	SEG28	137	SEG3/Strobe3
18	PortA.2	58	SEG53	98	SEG27	138	SEG2/Strobe2
19	PortA.3	59	SEG52	99	SEG26	139	SEG1/Strobe1
20	PortA.7	60	SEG51	100	SEG25	140	SEG0/Strobe0
21	PortB.0	61	SEG50	101	SEG24	141	COM6
22	PortB.1	62	SEG49	102	SEG23	142	COM5
23	VDD	63	SEG48	103	SEG22	143	COM4
24	OSCI	64	SEG47	104	SEG21	144	COM3
25	NC	65	SEG46	105	SEG20	145	COM2
26	NC	66	SEG45	106	SEG19	146	NC
27	NC	67	SEG44	107	SEG18	147	NC
28	NC	68	NC	108	SEG17	148	NC
29	HOSCI	69	NC	109	SEG16	149	NC
30	RESETB	70	NC	110	SEG15/Strobe15	150	NC
31	TEST	71	NC	111	SEG14/Strobe14	151	NC
32	VSS	72	NC	112	SEG13/Strobe13	152	NC
33	NC	73	NC	113	SEG12/Strobe12	153	NC
34	COM7	74	NC	114	SEG11/Strobe11	154	NC
35	COM8	75	NC	115	SEG10/Strobe10	155	NC
36	COM9	76	NC	116	SEG9/Strobe9	156	NC
37	SEG59	77	NC	117	SEG8/Strobe8	157	NC
38	SEG58	78	NC	118	NC	158	NC
39	SEG57	79	NC	119	NC	159	NC
40	NC	80	NC	120	NC	160	NC

## 5 Pin Description

### 5.1 MCU System Pins (6 Pins)

Name	I/O/P Type	Description	Note
VDD	P	Digital and Analog positive power supply, the range is from 1.2V~1.8V. Connect to VSS through the capacitors (0.1μF).	-
VSS	P	Digital and Analog negative power supply.	-
RSTB	I	System reset pin. Low active, Connect 0.1 μF to VSS.	Int. pull-up
TEST	I	Test mode select pin (High active). For chip internal test only, Normally connect to VSS.	Int. Pull Down
OSCI	I	External RC oscillator/Crystal oscillator connecting pin	Ext. R to VDD
HOSCI	I	Hi-Speed RC oscillator connecting pin.	Ext. R to VDD

### 5.2 Embedded LCD Pins (77 Pins)

Name	I/O/P Type	Description	Note
COM0~COM9	O	LCD common signal output pin	
SEG0~SEG15	O	LCD segment signal output pin shared with Key Strobe 0~15	
SEG16~ SEG59	O	LCD segment signal output pin	
C1A, C1B		LCD voltage charge-pump pin. Connect 0.1 μF between C1A and C1B.	
C2A, C2B		LCD voltage charge-pump pin. Connect 0.1 μF between C2A and C2B.	
V30, V22, V07	O	LCD bias Pin. Connect 0.1 μF to Vss	

### 5.3 I/O Port (7 Pins)

Port	Bit	Function	I/O Type	Power Source	Description	Note
Port A	Bits 3~0 (for key scan)	General Input	I	VDD	Key input	Int. Pull-up (R1: small resistor, R2: Large resistor) controllable
		Interrupt and wake up	I	VDD	Input port interrupt and wake-up pin	
	Bit 7	General Input	I	VDD		
		Interrupt and wake up	I	VDD	Input port interrupt and wake-up pin	
Port B	Bits 1~0	General Input	I	VDD		Int. Pull up controllable
		General Output	O	VDD		

## 6 Code Option

Located at Address 0x000C~0x000F of Program ROM

- Initial mode after reset:
  - Select “Slow” mode or “Fast” mode

**NOTE**

*For Initial mode after reset, it is recommended that you set the setting to “Slow mode.”*

- Low Frequency Oscillator:
  - Select “External RC” oscillator or “Internal RC” oscillator
- High Frequency Oscillator:
  - Select “External RC” oscillator or “Internal RC” oscillator
- The Frequency of Internal High RC Oscillator:
  - Select “200KHz” or “300KHz” or “500KHz”

## 7 Function Description

### 7.1 Reset Function

Reset can be generated by one of the following:

- Power-on voltage detector reset and power-on reset
- WDT time out
- RSTB pin pull low

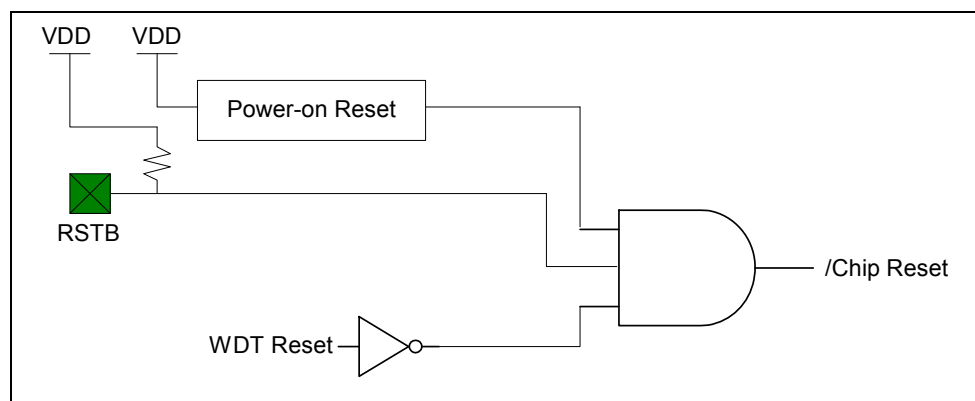


Figure 7-1 On-chip Reset Schematic

#### 7.1.1 Power-on Reset

The power-on reset circuit holds the device under reset condition until VDD is above Vpor (power-on reset voltage). Whenever the voltage supply lowers to below Vpor, a Reset will occur.

### 7.1.2 RSTB Pin

In normal condition, the RSTB pin is pulled up to VDD. Whenever the RSTB is at a low condition (level hold), a Reset will occur.

### 7.1.3 WDT Time-out

When the Watchdog Timer is enabled, the WDT time-out will cause the chip to reset. To prevent reset from occurring, the WDT value should be cleared with the “WDTCl” instruction before WDT time-out. WDT time-out can also be used to flag software malfunction.

### 7.1.4 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)  
Under 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. Used in signed operation when Bit 6 is carried into or borrows from a signed bit (Bit 7).

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

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

#### NOTE

- When  $OV=1$  after a signed arithmetic, check the SGE bit and SLE bit to verify whether overflow (carry into sign bit) or underflow (borrow from sign bit) occurred.  
If  $OV=1$  and  $SGE=1$  → overflow occurred.  
If  $OV=1$  and  $SLE=1$  → underflow occurred.
- When overflow occurred, the MSB of the Accumulator should be cleared to obtain the correct value.  
When underflow occurred, the MSB of the Accumulator should be set to obtain the correct value.



### 7.1.5 Initialization after Reset

- The oscillator is running, or will be started.
- The Watchdog timer is cleared.
- During power-on reset or RSTB pin low condition, the /TO bit and /PD bit of RF (Status) are set to “1.” At WDT time out reset, the /TO bit is cleared.
- The program counter (PCM: PCL) is clear to all “0.”
- The subsequent table shows the other registers’ initial values.

#### 7.1.5.1 Special Register

Addr.	Name	Initial Value	Addr.	Name	Initial Value
00h	INDF0	---- ----1	10h	Port A	x--- xxxx
01h	FSR0	0000 0000	11h	Port B	---- --xx
02h	BSR	0000 0000	12h	General RAM	uuuu uuuu
03h	INDF1	---- ----1	13h	General RAM	uuuu uuuu
04h	FSR1	1000 0000	14h	General RAM	uuuu uuuu
05h	BSR1	0000 0000	15h	General RAM	uuuu uuuu
06h	STKPTR	0000 0000	16h	General RAM	uuuu uuuu
07h	PCL	0000 0000	17h	General RAM	uuuu uuuu
08h	PCM	0000 0000	18h	General RAM	uuuu uuuu
09h	LCDARL	0000 0000	19h	General RAM	uuuu uuuu
0Ah	ACC	xxxx xxxx	1Ah	General RAM	uuuu uuuu
0Bh	TABPTRL	0000 0000	1Bh	General RAM	uuuu uuuu
0Ch	TABPTRM	0000 0000	1Ch	General RAM	uuuu uuuu
0Dh	TABPTRH	uuuu uuuu	1Dh	General RAM	uuuu uuuu
0Eh	LCDDATA	---- ----1	1Eh	General RAM	uuuu uuuu
0Fh	STATUS	cuxx xxxx 2	1Fh	General RAM	uuuu uuuu

#### 7.1.5.2 Control Register

Addr.	Name	Initial Value	Addr.	Name	Initial Value
20h	STBCON	-000 0000	29h	PACON	---- 1110
21h	TRINTCON	---- -000	2Ah	PAWAKE	0--- 0000
22h	TRINTSTA	---- -000	2Bh	PAINTEN	0--- 0000
23h	TR01CON	0000 0000	2Ch	PAINTSTA	0--- 0000
24h	TR0LL	uuuu uuuu	2Dh	PBCON	---- --00
25h	TR0LH	uuuu uuuu	2Eh	DCRB	---- --11
26h	TR1L	uuuu uuuu	2Fh	LCDCON	000- 00-0
27h	TR2WCON	0-00 0000	30h	POST_ID	-111 -000
28h	TR2L	uuuu uuuu	31h	CPUCON	---- -00c <sup>3</sup>

**Legend:** x: unknown      -: unimplemented, read as “0”  
u: unchanged,      c: value depends on actual condition

<sup>1</sup> Not a physical register.

<sup>2</sup> 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 remains unchanged.

<sup>3</sup> Bit 0 (MS0) of RE (CPUCON) is reloaded from “INIM” bit of code option when the MCU resets.

## 7.2 Oscillator System

The oscillator system is used to generate the device clock. The oscillator system is composed of an Internal RC, or External RC oscillator for Slow mode and an Internal RC, or external RC oscillator for Fast mode as shown in the diagram below.

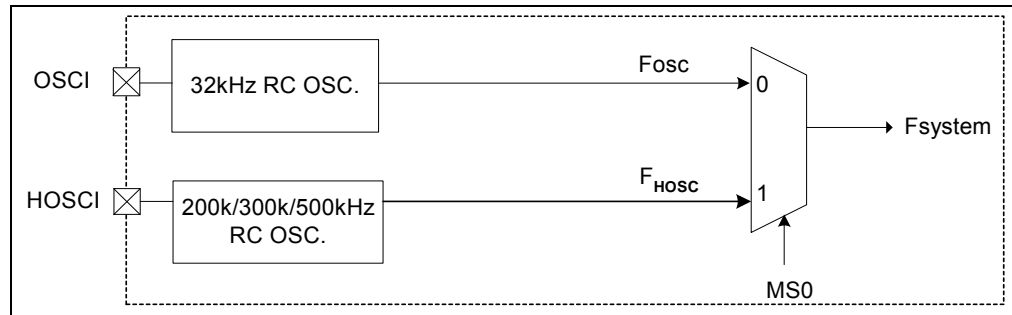


Figure 7-3 Oscillator System Function Block Diagram

The **MS0** bit (mode select bit) of **CPUCON** register (R31h) is used to set the Slow or Fast mode (see Section 7.3.1).

**0:** Slow mode (MCU system Clock is from Fosc)

**1:** Fast mode (MCU system Clock is from F<sub>HOSC</sub>)

### 7.2.1 32.8kHz Internal RC or 32.8kHz External RC Oscillator

- 32.8kHz RC Internal oscillator:  
Select “RC oscillator for Fosc” in the code option and allow OSCI pin to remain floating.
- 32.8kHz RC External oscillator:  
Select “RC oscillator for Fosc” in the code option and connect a 3MΩ resistor between OSCI and Vdd pins.

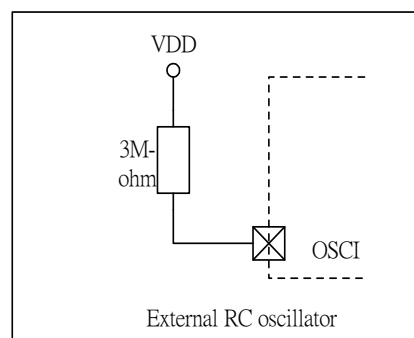


Figure 7-4 Slow Mode RC Oscillators Circuit Diagram

### 7.2.2 200kHz/300kHz/500kHz RC External Oscillator

A resistor should be connected between HOSCI and Vdd pin.

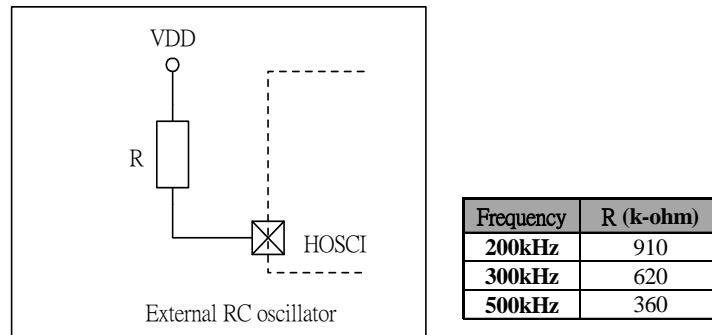


Figure 7-5 Fast Mode RC Oscillators Circuit Diagram

## 7.3 MCU Operation Mode

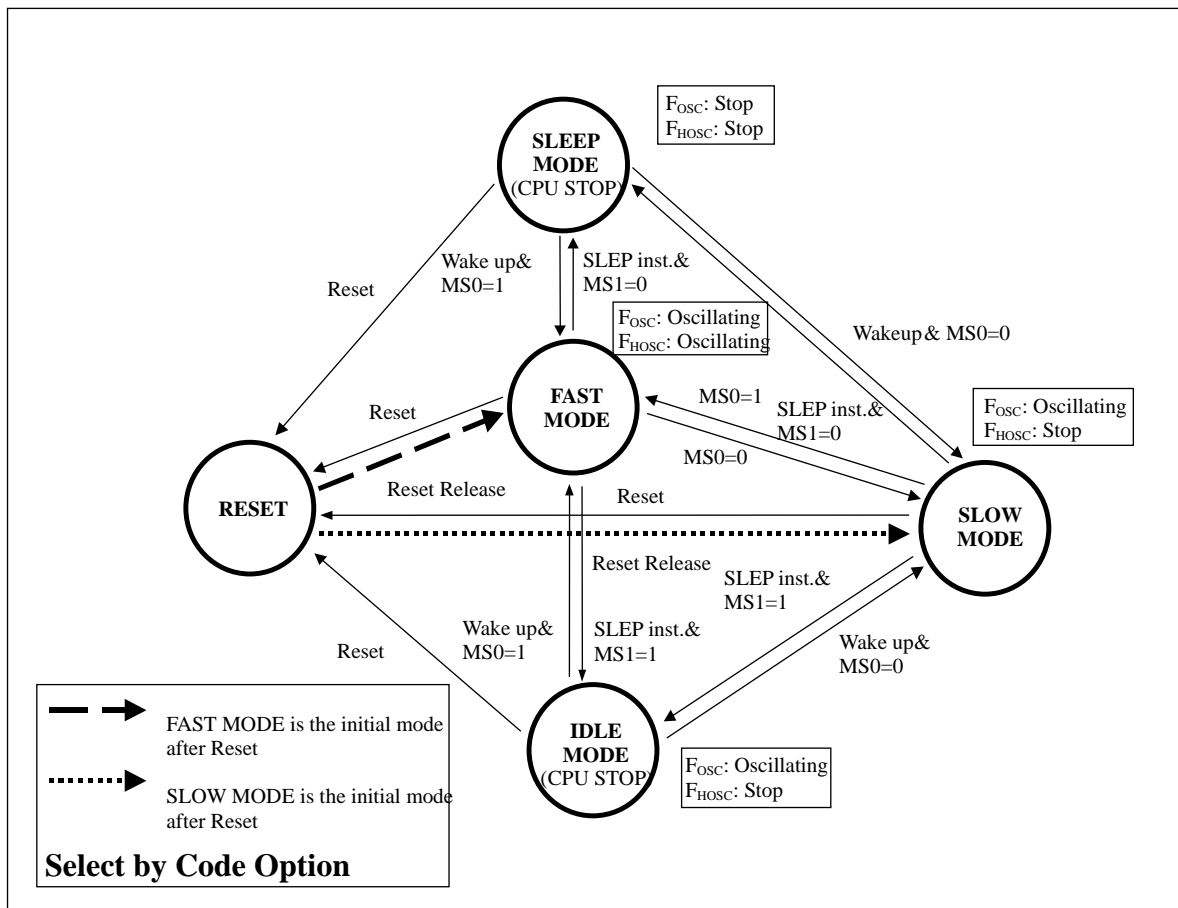


Figure 7-6 MCU Operation Block Diagram

The following table shows the supported device functions for each MCU Mode.

Device \ Mode	Sleep	Idle	Slow	Fast
Osc.(32768Hz)	×	✓	✓	✓
Fsystem	×	×	From Osc.	From Hosc.
Timers 0~2	×	×	✓	✓
INT	×*	×*	✓	✓
I/O wake up	✓	✓	×	×
Timer 1 wake up	×	✓	×	×

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

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

### 7.3.1 Slow, Fast, Sleep, and Idle Mode Operation

#### ■ CPUCON (R31h): MCU Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	–	–	–	–	GLINT	MS1	MS0

**Bit 0 (MS0):** Select Slow Mode or Fast Mode

0: Slow Mode

1: Fast Mode

**Bit 1 (MS1):** Select Sleep Mode or Idle Mode after executing “SLEP” instruction.

0: Sleep Mode

1: Idle Mode

#### ■ Slow Mode:

When the MS0 bit of the CPUCON register is set to “0,” the MCU will enter into Slow Mode and the corresponding system clock is at 32kHz. The Slow mode feature allows performance of all system operations at reduced power consumption.

#### NOTE

*The instruction “NOP” should be added after the “BC CPUCON, MS0” instruction when the MCU is made to enter into Slow Mode from Fast Mode. See the code example at the end of this Section.*

#### ■ Fast Mode:

When MS0 bit of the CPUCON register is set to “1,” the MCU will enter into Fast Mode. After setting the MS0 bit, it needs to count 32 clocks from HOSC, then the system clock switches from slow to high frequency. This mode allows performance of all the system operations at fast speed, but under highest consumption of power.

■ **Idle Mode:**

When the MS1 bit of the CPUCON register is set to “1.” and the “SLEP” instruction is executed, the MCU will enter into Idle Mode. The Idle Mode suspends all system operations except for the 32kHz oscillator. It retains the internal status under low power consumption without stopping the clock function.

The Idle Mode is awoken by the Timer 1 wake up or by I/O pin wake up (if enabled) and returns to the either Slow Mode (MS0=0) or Fast Mode (MS0=1)

**NOTE**  
*All registers remain unchanged during Sleep Mode.*

■ **Sleep Mode:**

When the MS1 bit of the CPUCON register is set to “0,” and the “SLEP” instruction is executed, the MCU will enter into Sleep Mode. The Sleep Mode suspends all system operation and put on hold the internal status immediately before the suspension of operation. Sleep Mode operates under very low power consumption and is awoken by I/O pin wake up.

**NOTE**

- *The /PD bit of the Status Register (R0Fh) is cleared when the MCU enters Sleep Mode.*
- *This /PD bit is set to “1” by “WDTC” instruction, power-on reset, or by RSTB pin low condition.*
- *All registers remain unchanged during Sleep Mode.*

■ **Slow Mode to Fast Mode Timing:**

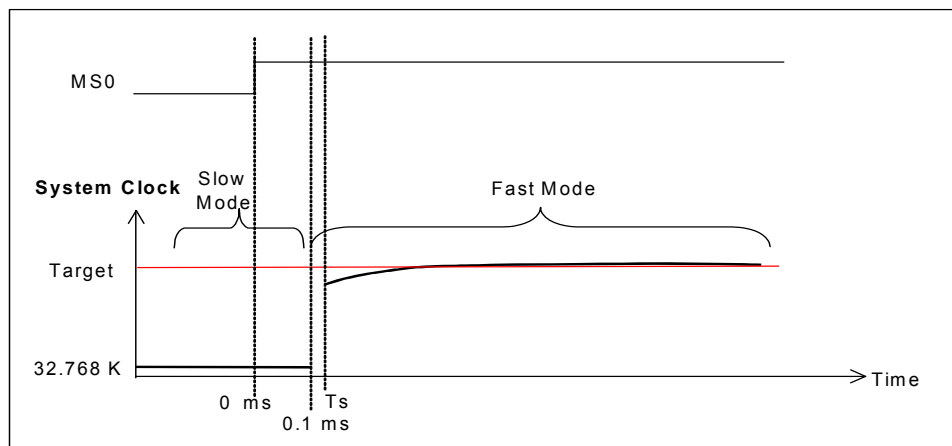


Figure 7-7 Slow Mode to Fast Mode Timing Diagram

**NOTE**

1. *Slow Mode switches to Fast Mode at Time=0ms.*
2. *System clock will switch to Fast Mode after a delay of 0.1ms by oscillator and enters into Fast Mode (i.e., system clock will be at 200, 300, or 500 kHz).*
3. *High frequency RC will be stabilized at Time=Ts (around 15μs~30μs).*

■ **Code Example:**

```

;Entry FAST mode
    BS    CPUCON,MS0

;Entry SLOW mode
    BC    CPUCON,MS0

;FAST mode Entry SLOW mode
    BS    CPUCON,MS0
    :
    :
    BC    CPUCON,MS0
    NOP

;Entry IDLE mode
    BS    CPUCON,MS1
    SLEP
    NOP

;Entry SLEEP mode
    BC    CPUCON,MS1
    SLEP
    NOP
    
```

### 7.3.2 Wake-up Operation

Oscillator is off during Sleep Mode. The MCU is awoken by input port (Port A), then returns to Fast Mode or Slow Mode (as determined by MS0 bit of CPUCON register as described in previous section).

When in Idle Mode, the 32kHz oscillator keeps on running. The MCU is awoken by input port (Port A) or Timer1, then returns to Fast Mode or Slow Mode (as determined by MS0 bit of CPUCON register as described in previous section).

■ **PAWAKE (R2Ah): Port A Wake-Up Function Control Register**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WKEN7	–	–	–	WKEN3	WKEN2	WKEN1	WKEN0

**Bit 7 (WKEN7), Bit 3 (WKEN3) ~ Bit 0 (WKEN0):** Wake-up function control bit of Port A.7, Port A.3 ~ Port A.0

**0:** Disable Port A.7, Port A.3 ~ Port A.0 wake-up function

**1:** Enable Port A.7, Port A.3 ~ Port A.0 wake-up function

■ **T1WKEN Bit of (R23h): Timer 0 & 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 7 (T1WKEN):** Timer 1 underflow wake-up function control bit in Idle Mode

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

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

## 7.4 Interrupts

When interrupt occurs, the GLINT bit of the CPUCON register is reset to “0”. It disables all interrupts, including Levels 1 ~ 5. Setting this bit to “1” will enable all un-mask interrupts.

### 7.4.1 Global Interrupt

■ **GLINT Bit of CPUCON (R31h) MCU Control Register**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	-	-	-	GLINT	MS1	MS0

**Bit 2 (GLINT):** Global interrupt control bit

0: disables all interrupts, including Level 1 ~ Level 5

1: enables all un-mask interrupts

■ **Interrupt Vector**

Interrupt Level	Interrupt Source	Start Address	Remarks
-	RESET	0x00000	-
Level 1	Port A.7, 3 ~ 0	0x00002	PAINT
Level 2	reserved	0x00004	Reserved
Level 3	reserved	0x00006	Reserved
Level 4	Timers 0~2	0x00008	TMR0I, TMR1I, TMR2I
Level 5	reserved	0x0000A	Reserved

■ **Code Example:**

```

; ***** Reset program
ResetSEG CSEG 0X00
    LJMP  RESET           ; (0x00) Initialize
    LJMP  PAINT          ; (0x02) Port A Interrupt
    LJMP  RESERVED      ; (0x04) Reserved
    LJMP  RESERVED      ; (0x06) Reserved
    LJMP  TIMERINT      ; (0x08) Timer-0,1,2 Interrupt
    LJMP  RESERVED      ; (0x0A) Reserved
INT    CSEG 0x20

; --- Push interrupt register
PUSH:
    MOVPR StatusBuf,Status
    MOV  AccBuf,A
    RET

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

### 7.4.2 Input Port (Port A.7, Port A.3 ~ Port A.0) Interrupt

Port A.7, Port A.3 ~ Port A.0 are used as external interrupt/wake up input. If PA7IE, PA3IE ~ PA0IE bits of PAINTEN register are set to “1,” Port A.7, Port A.3 ~ Port A.0 are the external interrupt input port format.

■ **PAINTSTA (R2Ch): Port A.7, Port A.3 ~ Port A.0 Interrupt Status Register**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PA7I	-	-	-	PA3I	PA2I	PA1I	PA0I

**Bit 7 (PA7I), Bit 3 (PA3I) ~ Bit 0 (PA0I):** Port A.7, Port A.3 ~ Port A.0 Interrupt status

Set to “1” when a pin falling edge is detected.

Clear to “0” by software.

■ **PAINTEN (R2Bh): Port A.7, Port A.3 ~ Port A.0 Interrupt Control Register**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PA7IE	–	–	–	PA3IE	PA2IE	PA1IE	PA0IE

**Bit 7 (PA7IE), Bit 3 (PA3IE) ~ Bit 0 (PA0IE):** PortA.7, Port A.3 ~ PortA.0 Interrupt control bits

**0:** Disable interrupt function

**1:** Enable interrupt function

■ **Code Example:**

```

; == Input Port A Interrupt
PAINT:
    SOCALL PUSH
    CLR     PAINTSTA
    :
    SJMP   POP
    RETI
    
```

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

#### 7.4.3.1 Timer 0 Interrupt

Timer 0 is a 16-bit timer used for general time counting. When the counting value underflows, the Timer 0 interrupt takes place and the TRL0H: TRL0L value is reloaded into the timer automatically.

■ **TMR0IE Bit of TRINTCON (R21h) Timer Interrupt Control Register**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	–	–	–	–	TMR2IE	TMR1IE	TMR0IE

**Bit 0 (TMR0IE):** Control bit of Timer 0 interrupt

**0:** Disable Timer 0 interrupt function

**1:** Enable Timer 0 interrupt function

■ **TMR0I Bit of TRINTSTA (R22h) Timer Interrupt Status Register**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	–	–	–	–	TMR2I	TMR1I	TMR0I

**Bit 0 (TMR0I):** Status bit of Timer 0 interrupt

Set to “1” when Timer 0 counter underflows.

Clear to “0” by software.

#### 7.4.3.2 Timer 1 Interrupt

Timer 1 is an 8-bit timer used for time counting and wake-up functions. When the counting value of Timer 1 underflows, interrupt occurs and the TRL1 value is reloaded to the timer.

■ **TMR1IE Bit of TRINTCON (R21h) Timer Interrupt Control Register**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	–	–	–	–	TMR2IE	TMR1IE	TMR0IE

**Bit 1 (TMR1IE):** Control bit of Timer 1 interrupt  
**0:** Disable Timer 1 interrupt function  
**1:** Enable Timer 1 interrupt function

■ **TMR1I Bit of TRINTSTA (R22h) Timer Interrupt Status Register**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	–	–	–	–	TMR2I	TMR1I	TMR0I

**Bit 1 (TMR1I):** Status bit of Timer 1 interrupt  
Set to “1” when Timer 1 counter underflows.  
Clear to “0” by software.

### 7.4.3.3 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 the timer.

■ **TMR2IE Bit of TRINTCON (R21h) Timer Interrupt Control Register**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	–	–	–	–	TMR2IE	TMR1IE	TMR0IE

**Bit 2 (TMR2IE):** Control bit of Timer 2 interrupt  
**0:** Disable Timer 2 interrupt function  
**1:** Enable Timer 2 interrupt function

■ **TMR2I Bit of TRINTSTA (R22h) Timer Interrupt Status Register**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	–	–	–	–	TMR2I	TMR1I	TMR0I

**Bit 2 (TMR2I):** Status bit of Time 2 interrupt  
Set to “1” when Timer 2 counter underflows.  
Clear to “0” by software.

#### 7.4.3.4 Code Example:

```

; === Timer-0,1,2 Interrupt
TIMERINT:
    SOCALL PUSH
    JBS    TRINTSTA,TMR0I ,toTM0INT
    JBS    TRINTSTA,TMR1I ,toTM1INT
    JBS    TRINTSTA,TMR2I ,toTM2INT
    SJMP  POP

; --- Timer 0 Interrupt
toTM0INT:
    BC    TRINTSTA,TMR0I
    :
    SJMP  POP
    RETI

; --- Timer 1 Interrupt
toTM1INT:
    BC    TRINTSTA,TMR1I
    :
    SJMP  POP
    RETI

; --- Timer 2 Interrupt
toTM2INT:
    BC    TRINTSTA,TMR2I
    :
    SJMP  POP
    RETI
    
```

## 7.5 Program ROM Map

ROM Size = 32K Words	
Address.	Description
0000h   000Bh	Interrupt Vector (12 words)
000Ch   000Fh	Code Option (4 words)
0010h   001Fh	Test Program (16 words)
0020h   7FFFh	Program or Fixed data region

## 7.6 RAM Map for Special and Control Registers

RAM Size: 93 Bytes + 16 Banks × 128 Bytes = 2141 Bytes

### 7.6.1 Special and Control Registers

Legend: R = Readable bit W = Writable bit – = Not implemented

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 (R0)							
2	BSR	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		General Flag				Bank select register (for INDF0 & general)			
3	INDF1	R/W Indirect Addressing Pointer 1							
4	FSR1	R Fixed 1	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		File Select Register 1 for INDF1 (R3)							
5	BSR1	R Fixed 0	R Fixed 0	R/W	R/W	R/W	R/W	R/W	R/W
		General Flag			Bank select Register 1 (for INDF1)				
6	STKPTR	R/W Stack Pointer							
7	PCL	R/W PC7	R/W PC6	R/W PC5	R/W PC4	R/W PC3	R/W PC2	R/W PC1	R/W PC0
8	PCM	R/W PC15	R/W PC14	R/W PC13	R/W PC12	R/W PC11	R/W PC10	R/W PC9	R/W PC8
9	LCDARL	R/W LCD RAM Column Address							
A	ACC	R/W Accumulator							
B	TABPTRL	R/W Low Byte of Table Pointer							

(Continued)

Addr.	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
C	TABPTRM	R/W Middle Byte of Table Pointer							
D	TABPTRH	R/W General purpose RAM							
E	LCDDATA	R/W Indirect Register to LCD RAM							
F	STATUS	R /TO	R /PD	R/W SGE	R/W SLE	R/W OV	R/W Z	R/W DC	R/W C
10	Port A	R Port A.7	-	-	-	R Port A.3	R Port A.2	R Port A.1	R Port A.0
11	Port B	-	-	-	-	-	-	R/W Port B.1	R/W Port B.0
20	STBCON	-	R/W SCAN	R/W BitST	R/W All	R/W STB3	R/W STB2	R/W STB1	R/W STB0
21	TRINTCON	-	-	-	-	-	R/W TMR2IE	R/W TMR1IE	R/W TMR0IE
22	TRINTSTA	-	-	-	-	-	R/W TMR2I	R/W TMR1I	R/W TMR0I
23	TR01CON	R/W T1WKEN	R/W T1EN	R/W T1PSR1	R/W T1PSR0	R/W T0EN	R/W T0CS	R/W T0PSR1	R/W T0PSR0
24	TRL0L	R/W Timer 0 Auto-reload Register Low Byte							
25	TRL0H	R/W Timer 0 Auto-reload Register High Byte							
26	TRL1	R/W Timer 1 Auto-reload Register							
27	TR2WCON	R/W WDTEN	-	R/W WDTPSR1	R/W WDTPSR0	R/W T2EN	R/W T2CS	R/W T2PSR1	R/W T2PSR0
28	TRL2	R/W Timer 2 Auto-reload Register							
29	PACON	-	-	-	-	R/W PA7PU	R/W R2EN	R/W R1EN	R/W EK
2A	PAWAKE	R/W WKEN7	-	-	-	R/W WKEN3	R/W WKEN2	R/W WKEN1	R/W WKEN0
2B	PAINTEN	R/W PA7IE	-	-	-	R/W PA3IE	R/W PA2IE	R/W PA1IE	R/W PA0IE
2C	PAINTSTA	R/W PA7I	-	-	-	R/W PA3I	R/W PA2I	R/W PA1I	R/W PA0I
2D	PBCON	-	-	-	-	-	-	R/W PB1PU	R/W PB0PU
2E	DCRB	-	-	-	-	-	-	R/W PB1DC	R/W PB0DC
2F	LCDCON	R/W REV	R/W BLANK	R/W LCDON	-	R/W LCR1	R/W LCR0	-	R/W LBVON
30	POST_ID	-	R/W LCD_ID	R/W FSR1_ID	R/W FSR0_ID	-	R/W LCD_PE	R/W FSR1_PE	R/W FSR0_PE
31	CPUCON	-	-	-	-	-	R/W GLINT	R/W MS1	R/W MS0

### 7.6.2 Other Unbanked General RAM

Address	Unbanked
0Dh	General purpose RAM
13h   1Fh	General purpose RAM
32h   7Fh	General purpose RAM

### 7.6.3 Banked General RAM

Address	Bank 0	Bank 1	Bank 2	Bank 3	.....	Bank 15
80h   FFh	General Purpose RAM	General Purpose RAM	General Purpose RAM	General Purpose RAM	.....	General Purpose RAM

## 7.7 LCD RAM Map

#### ■ 1/10 Duty

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								
:	:								
SEG39	27H								

#### ■ COM8 (Bit0) ~ COM9 (Bit1)

RAM Address		COM8	COM9
LCDARL		Bit0	Bit1
SEG0	40H		
:	:		
SEG59	7BH		

## 7.8 Special Function Registers

### 7.8.1 ACC (R0Ah): Accumulator

Internal data transfer operation, or instruction operand holding usually involves the temporary storage function of the Accumulator, which is not an addressable register.

### 7.8.2 POST\_ID (R30h): Post Increase / Decrease 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 4 (FSR0\_ID):** 1: auto increase FSR0  
0: auto decrease FSR0

**Bit 5 (FSR1\_ID):** 1: auto increase FSR1  
0: auto decrease FSR1

### 7.8.3 BSR, FSR0, INDF0 (R02h, R01h, R00h): Indirect Address Pointer 0 Registers

**BSR (R02h)** determines which bank is active (working bank) among the 16 banks (Bank 0 ~ Bank 15). Bit 7 ~ 4 can be used as general flag.

**FSR0 (R01h)** is an address register for INDF0. Up to 256 bytes (Address: 00 ~ 0FFh) can be selected.

**INDF0 (R00h)** is not a physically implemented register.

### 7.8.4 BSR1, FSR1, INDF1 (R05h, R04h, R03h): Indirect Address Pointer 1 Registers

**BSR1 (R05h)** is a bank register for INDF1. It cannot determine the working bank for the general register. Bit 5 ~ 4 can be used as general flag.

**FSR1 (R04h)** is an address register for INDF1. Up to 128 bytes (Address: 80 ~ 0FFh) can be selected. Bit 7 of FSR1 is fixed to "1."

**INDF1 (R03h)** is not a physically implemented register.

#### ■ Code Example:

```
-----  
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  
: MOV    INDF1,INDF0          ; Move 80H ~ 0FFH data to Bank1  
: :  
-----
```

■ **INDF1 Linear Address Capabilities**

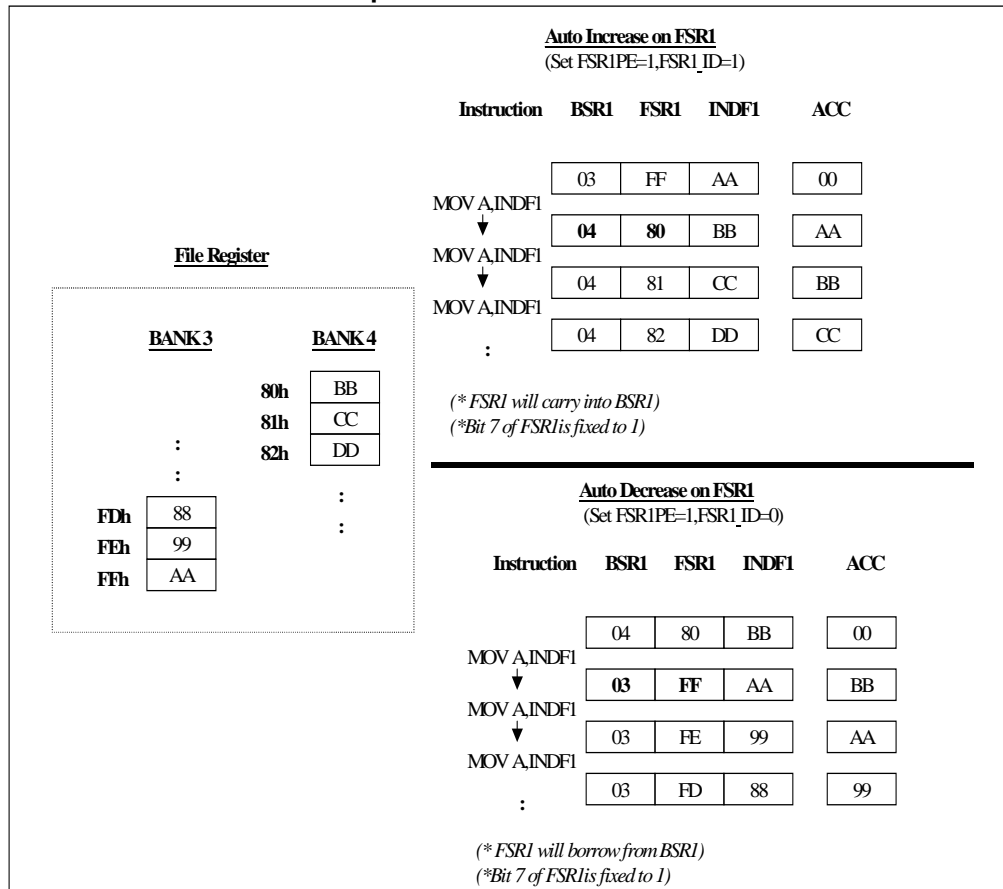


Figure 7-8 INDF1 Linear Address Capabilities Diagram

■ **Code Example:**

```

;*****
;* Const => Working bank setting
;* REG => Save or Recall register
;*****
; **** RAM stack macro
; *** Initial RAM stack
IniRAMsk MACRO #Const
    MOV    A,#Const
    MOV    BSR1,A
    CLR    FSR1
    BS     POST_ID,FSR1PE
    ENDM

; *** Push RAM stack
PushRAM MACRO REG
    BS     POST_ID,FSR1_ID
    MOVPR INDF1,REG
    ENDM

; *** Pop RAM stack
PopRAM MACRO REG
    BC     POST_ID,FSR1_ID
    MOVPR REG,INDF1
    ENDM

; *** Main start program
Mstart:
    :
    :
    IniRAMsk #29
    :
    :
MnLoop:
    :
    :
    LJMP  MnLoop

; *** Interrupt routine
IntSR:
    PushRAM ACC
    PushRAM Status
    :
    :
    PopRAM Status
    PopRAM ACC
    RETI
    
```

### 7.8.5 STKPTR (R06h): Stack Pointer Register

The initial stack pointer is 00h. Each INT/CALL will stack two bytes of address with a total capacity of 32 levels. When stack overflows, it will replace the first stack level.

**NOTE**

*This Bank RAM does not include the stack RAM. The stack RAM is independent and cannot be seen.*

### 7.8.6 PCL, PCM (R07h, R08h): Program Counter Registers

Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCM								PCL							

The configuration structure can generate up to 32K×16 on-chip ROM addresses for the relative programming instruction codes.

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

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

“**LCALL**” or “**LJMP**” loads the full 14 bits of the PC (32K×16 ROM)

“**ADD R7, A**” or “**ADC R7, A**” allows a relative address to be added into the current PC. The carry bit of R7 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
    
```

### 7.8.7 TABPTRL, TABPTRM, (R0Bh, R0Ch): Table Pointer Registers

Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TABPTRM								TABPTRL							

Program ROM or Internal ROM address register.

**Bit 15 ~ Bit 1** are used to point the memory address.

**Bit 0** is used to select whether low byte or high byte (see TBRD instruction in the Instruction Set under Section 12)

#### ■ Code Example:

```

; *** Program ROM
:
:
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

```

### 7.8.8 Port A, Port B (R10h, R11h): General I/O Pin Registers

**Port A (R10h)** Port A.0 ~ 3, 7 are general I/O pins registers

**Port B (R11h)** Port B.0 ~ 1 are general I/O pins registers

### 7.8.9 STBCON (R20): 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 6 (SCAN):** Automatic key scan or specify the scan signal bit by bit

**0:** Key scan is specified as “Bits STB3 ~ 0 defined”

**1:** Auto strobe scanning

**Bit 5 (BitST):** Enable SEG0 ~ SEG15 as key strobe pins

**0:** SEG0 ~ SEG15 are used as LCD segment signal pins only

**1:** SEG0 ~ SEG15 are used as key strobe pins and LCD segment pins. Strobe signal is STB3 ~ 0 defined

**Bit 4 (ALL):** Set all strobe.

- 0: Bit Strobe
- 1: All Strobe

**Bits 3 ~ 0 (STB3 ~ 0):** 16 to 1 multiplexing selector of key strobe pin

#### 7.8.10 PACON (R29h): Port A Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	–	–	–	PA7PU	R2EN	R1EN	KE

**Bit 3 (PA7PU):** Enable Port A.7 pull-up resistor bits

- 0: Disable Port A.7 pull up resistor
- 1: Enable Port A.7 pull up resistor

**Bit 2 (R2EN):** R2 pull up resistor (large resistor) control bit for Port A.3 ~ Port A.0.

- 0: Disable R2 pull up resistor
- 1: Enable R2 pull up resistor

**Bit 1 (R1EN):** R1 pull up resistor (small resistor) control bit for Port A.3 ~ Port A.0.

- 0: Disable R1 pull up resistor
- 1: Enable R1 pull up resistor

**Bit 0 (KE):** Key input enable/disable control bit

- 0: Disable key input function (Port A.0~3 do **not** correspond with Key input in software scan mode)
- 1: Enable key input function (Port A.0~3 correspond with Key input in software scan mode)

#### 7.8.11 PAWAKE (R2Ah): Port A Wake-up Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WKEN7	–	–	–	WKEN3	WKEN2	WKEN1	WKEN0

**Bit 7, Bit 3 ~ 0 (WKEN7, WKEN3 ~ WKEN0):** Wake-up enable control bits of Port A.7, Port A.3 ~ Port A.0.

- 0: Disable Port A.7, Port A.3 ~ Port A.0 wake-up function
- 1: Enable Port A.7, Port A.3 ~ Port A.0 wake-up function

### 7.8.12 PAINTEN (R2Bh): Port A Interrupt Enable Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PA7IE	–	–	–	PA3IE	PA2IE	PA1IE	PA0IE

**Bit 7, Bit 3 ~ Bit 0 (PA7IE, PA3IE ~ PA0IE):** Interrupt Control bits

**0:** Disable Port A.7, Port A.3 ~ Port A.0 interrupt function

**1:** Enable Port A.7, Port A.3 ~ Port A.0 interrupt function

### 7.8.13 PAINTSTA (R2Ch): Port A Interrupt Status Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PA7I	–	–	–	PA3I	PA2I	PA1I	PA0I

**Bit 7, Bit 3 ~ Bit 0 (PA7I ~ PA0I):** INT status of Port A.7, Port A.3 ~ Port A.0 interrupt bits

Set to “1” when pin falling edge detected

Clear (“0”) by software

### 7.8.14 PBCON (R2Dh): Port B Pull up Resistor Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	–	–	–	–	–	PB1PU	PB0PU

**Bit 1 ~ Bit 0 (PB1PU ~ PB0PU):** Port B.1 ~ Port B.0 pull up resistor control bits

**0:** Disable pull up resistor

**1:** Enable pull up resistor

**NOTE**

*This function is only available with Port B selected as input pin.*

### 7.8.15 DCRB (R2Eh): Port B Direction Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	–	–	–	–	–	PB1DC	PB0DC

**Bit 1 ~ Bit 0 (PB1DC ~ PB0DC):** Port B.1 ~PortB.0 direction control bits

**0:** Set to output pin

**1:** Set to input pin

## 8 Peripheral

### 8.1 Timer 0 (16 Bits Timer with Event Counter Function)

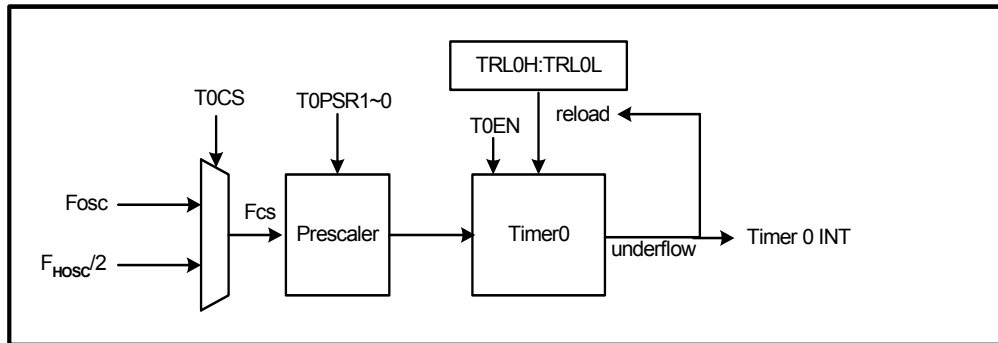


Figure 8-1 Timer 0 Function Block Diagram

Timer 0 is a general-purpose 16 bits down counter used on applications that require time counting with interrupt. The clock source ( $F_{CS}$ ) is selectable from the oscillator clock ( $F_{OCS}$ ) or half of the system clock ( $F_{HOSC}/2$ ).

A prescaler for the timer is also provided. The  $T0PSR1 \sim T0PSR0$  bits of  $TR01CON$  register determine the prescaler ratio and generate different clock rates as clock source for the timer.

The Counter value is decremented by one (count down) according to the timer clock source frequency. When underflow occurs, the timer interrupt is triggered if the global interrupt and Timer 0 interrupt are both enabled. At the same time,  $TRL0H:TRL0L$  will automatically be reloaded into the 16-bit counter.

$$T = \frac{1}{F_{CS}} \times Prescaler \times (TRL0H : TRL0L + 1)$$

#### 8.1.1 Timer 0 Registers

##### ■ $TRL0H:TRL0L$ (R25h, R24h): Timer 0 Reload Registers

Reload registers are used to store the auto-reload value of Timer 0. When Timer 0 is enabled or underflow occurs,  $TRL0H:TRL0L$  register values will automatically be reloaded into the 16 bits counter.

##### ■ $TR01CON$ (R23h): 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 3 (T0EN):** Timer 0 enable control bit

0: Disable

1: Enable

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

**0:** Clock source is from Fosc

**1:** Clock source is from Fhosc/2

**Bit 1 ~ Bit 0 (TOPSR1 ~ TOPSR0):** Timer 0 prescaler select bits

TOPSR1: TOPSR0	Prescaler Value
00	1:1
01	1:4
10	1:16
11	1:64

■ **CPUCON (R31h): MCU Control Register**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	-	-	-	GLINT	MS1	MS0

**Bit 2 (GLINT):** Global interrupt enable/disable bit

**0:** Disable all interrupts

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

■ **TRINTCON (R21h): Timer Interrupt Control Register**

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

**Bit 0 (TMR0IE):** Timer 0 interrupt control bit

**0:** Disable interrupt function

**1:** Enable interrupt function

■ **TRINTSTA (R22h): Timer Interrupt Status Register**

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

**Bit 0 (TMR0I):** When Timer 0 interrupt occurs, this bit will be set, and Cleared to "0" by software.

■ **Code Example:**

```

; === Timer 0 interrupt
TIMERINT:
    PUSH
    JBC    TRINTSTA,TMR0I,Q_Time
    BC     TRINTSTA,TMR0I
    BTG    PORT B,0
Q_Time:
    POP
    RETI
; === Timer0 = [1/(300K/2)] * [1 x(1FFFh + 1)]
Timer0SR:
    :
    System setting 300KHz
    PB.0 setting output pin
    :
    MOV    A,#0B00000100
    AND    TR01CON,A                ; FHOSC & Pre-scale 1:1
    MOV    A,#0X1F
    MOV    TRL0H,A
    MOV    A,#0XFF
    MOV    TRL0L,A                ; 13.65ms=[1x(8191 + 1)/(300K/2)]
    BS    TR01CON,T0EN            ; Timer 0 enable
    BS    TRINTCON,TMR0IE        ; Timer 0 interrupt enable
    BC    TRINTSTA,TMR0I        ; Clear Timer 0 interrupt status
    BS    CPUCON,GLINT          ; Enable global interrupt
TimeLoop:
    SJMP  TimeLoop
    
```

## 8.2 Timer 1 (8 Bits)

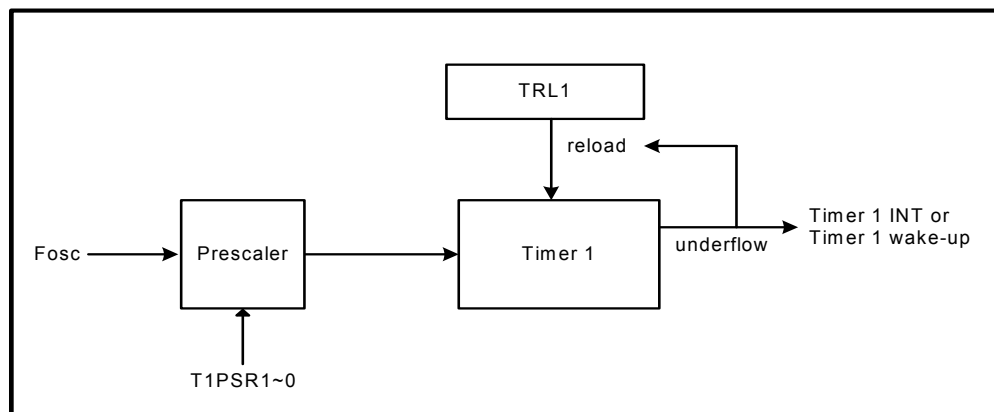


Figure 8-2 Timer 1 Function Block Diagram

Timer 1 is a general-purpose 8-bit down counter used on applications that require time counting with interrupt and wake-up functions. The clock source is from the oscillator clock ( $F_{OCs}$ ).

A prescaler for the timer is also available. The T1PSR1 ~ T1PSR0 bits of TR01CON register determine the pre-scale ratio and generate different clock rates as clock source for the timer. Setting T1WKEN bit of TR01CON register to “1” will enable the Timer 1 underflow wake-up function in Idle Mode.

The Counter value will be decremented by one (count down) according to timer clock source frequency. When the counter underflows, the timer interrupt is triggered if the global interrupt and Timer 1 interrupt are both enabled. At the same time, TRL1 value will be automatically reloaded into the 8-bit counter.

$$T = \frac{1}{F_{osc}} \times Prescaler \times (TRL1 + 1)$$

### 8.2.1 Timer 1 Registers

#### ■ TRL1 (R26h): Timer 1 Reload Register

This register is used to store the auto-reload value of Timer 1. When Timer 1 is enabled or underflow occurs, TRL1 register value will be automatically reloaded into the 8-bit counter.

#### ■ CPUCON (R31h): MCU Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	-	-	-	GLINT	MS1	MS0

**Bit 2 (GLINT):** Global interrupt enable/disable bit

**0:** Disable all interrupt

**1:** Enable all un-mask interrupt

#### ■ TR01CON (R23h): 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 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

**Bit 6 (T1EN):** Timer 1 enable control bit

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

**1:** Enable Timer 1

**Bit 5 ~ Bit 4 (T1PSR1 ~ T1PSR0):** Timer 1 prescaler select bits

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

■ TRINTCON (R21h): Timer Interrupt Control Register

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

**Bit 1 (TMR1IE):** Control bit of Timer1 interrupt.

0: Disable interrupt function

1: Enable interrupt function

■ TRINTSTA (R22h): Timer Interrupt Status Register

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

**Bit 1 (TMR1I):** When Timer 1 interrupt occurs, this bit will be set, and Cleared to "0" by software.

■ Code Example:

```

; === Timer 1 interrupt
TIMERINT:
    PUSH
    JBC    TRINTSTA,TMR1I,Q_Time
    BC     TRINTSTA,TMR1I
    BTG    Port B,0
Q_Time:
    POP
    RETI
; === Timer1 = 32.768K / [256 x (3Fh + 1)]
Timer1SR:
    :
    PB.0 setting output pin
    :
    MOV    A,#10110000B
    MOV    TR01CON,A                ; Fosc & Pre-scale 1:256 & wakeup
    MOV    A,#03FH
    MOV    TRL1,A                    ; 0.5sec=[256x(63+1)]/32.768K
    BS     TR01CON,T1EN              ; Timer 1 enable
    BS     TRINTCON,TMR1IE           ; Timer 1 interrupt enable
    BC     TRINTSTA,TMR1I            ; Clear Timer 1 interrupt status
    BS     CPUCON,GLINT              ; Enable global interrupt
    BS     CPUCON,MS1                ; Idle mode
T1Wloop:
    SLEP
    NOP
    :
    SJMP  T1Wloop
    
```

### 8.3 Timer 2 (8 Bits)

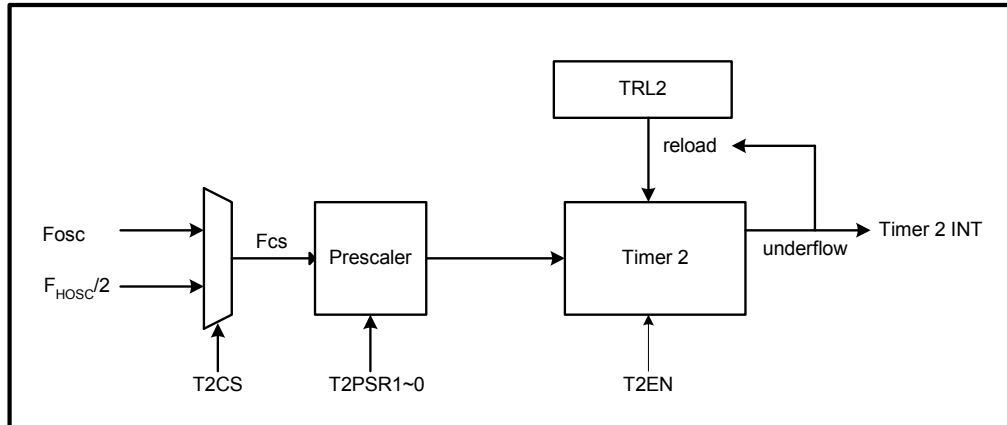


Figure 8-3 Timer 2 Function Block Diagram

Timer 2 is a general-purpose 8-bit down counter used on applications that require a time counter with interrupt. The clock source (Fcs) may be selected from the oscillator clock (F<sub>OSC</sub>) or half of the system clock (F<sub>HOSC</sub>/2).

A prescaler for the timer is also available. The T2PSR1 ~ T2PSR0 bits of TR2WCON register determine the prescaler ratio and generate different clock rates as clock source for the timer.

Counter value is decreased by one (counting down) according to the timer clock source frequency. When counter value underflows, the timer interrupt is triggered (if Timer 2 interrupt is enabled).

$$T = \frac{1}{F_{CS}} \times \text{Prescaler} \times (TRL2 + 1)$$

#### 8.3.1 Timer 2 Registers

##### ■ TRL2 (R28h): Timer 2 Reload Register

This register is used to store the auto-reload value of Timer 2. When Timer 2 is enabled or underflow occurs, TRL2 register value will be automatically reloaded into the 8-bit counter.

##### ■ TR2WCON (R27h): Timer 2/Watchdog Timer Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WDTEN	–	WDTPSR1	WDTPSR0	T2EN	T2CS	T2PSR1	T2PSR0

**Bit 3 (T2EN):** Timer 2 enable control bits

0: Disable Timer 2 (stop counting)

1: Enable Timer 2

**Bit 2 (T2CS):** Timer 2 clock source select bit

**0:** Clock source is from Fosc

**1:** Clock source is from Fhosc/2

**Bit 1 ~ Bit 0 (T2PSR1 ~ T2PSR0):** Timer 2 prescaler select bits

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

■ **CPUCON (R31h): MCU Control Register**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	–	–	–	–	GLINT	MS1	MS0

**Bit 2 (GLINT):** Global interrupt enable/disable bit

**0:** Disable all interrupts

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

■ **TRINTCON (R21h): Timer Interrupt Control Register**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	–	–	–	–	TMR2IE	TMR1IE	TMR0IE

**Bit 2 (TMR2IE):** Control bit of Timer 2 interrupt

**0:** Disable interrupt function

**1:** Enable interrupt function

■ **TRINTSTA (R22h): Timer Interrupt Status Register**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	–	–	–	–	TMR2I	TMR1I	TMR0I

**Bit 2 (TMR2I):** When Timer 2 interrupt occurs, this bit will be set, and Cleared to “0” by software.

■ **Code Example:**

```

; === Timer 2 interrupt
TIMERINT:
    PUSH
    JBC    TRINTSTA,TMR2I,Q_Time
    BC    TRINTSTA,TMR2I
    BTG    Port B,0
Q_Time:
    POP
    RETI
; === Timer2 = (1/32.768K) X [4 x (FFh + 1)]
Timer2SR:
    :
    PB.0 setting output pin
    :
    MOV    A,#00000010B
    MOV    TR2CON,A                ; Fosc & Pre-scale=1:4
    MOV    A,#0XFF
    MOV    TRL2,A                  ; 31.25ms=[4x(255+1)]/32768
    BS    TR2CON,T2EN              ; Timer 2 Enable
    BS    TRINTCON,TMR2IE          ; Timer 2 Interrupt Enable
    BC    TRINTSTA,TMR2I          ; Clear Timer 2 Interrupt Status
TMR2Loop:
    SJMP  TMR2Loop

```

## 8.4 Watchdog Timer (WDT)

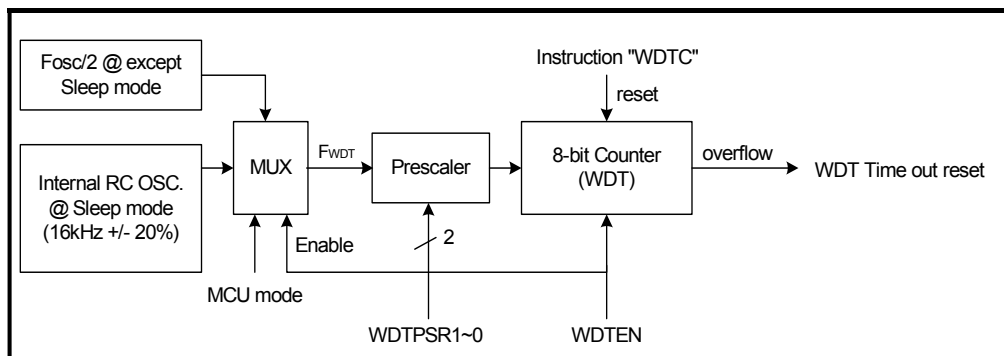


Figure 8-4 Watchdog Timer Function Block Diagram

The watchdog timer (WDT) clock source comes from an on-chip RC oscillator (16kHz ± 20%, MCU in Sleep mode) or FOSC/2 (MCU in Fast, Slow, or Idle mode). Therefore the WDT will keep on running even after the oscillator has been turned off.

The WDTEN bit controls the WDT's enable/disable functions. The initial state of the WDT is disabled. When WDT is enabled, its time-out will cause the MCU to reset. The "WDTC" instruction should be used to clear the WDT value before WDT time-out. A prescaler is provided to generate different clock rates for the WDT clock source. The prescaler ratio is defined by WDTPSR1 and WDTPSR0.

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

$$T = \frac{1}{F_{WDT}} \times \text{Prescaler} \times (WDT + 1)$$

### 8.4.1 Watchdog Timer (WDT) Registers

#### ■ TR2WCON (R27h): Timer2/Watchdog Timer Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WDTEN	–	WDTPSR1	WDTPSR0	T2EN	T2CS	T2PSR1	T2PSR0

**Bit 7 (WDTEN):** Watchdog Timer enable bit

0: Disable watchdog timer (stop running)

1: Enable watchdog timer

**Bit 5 ~ Bit 4 (WDTPSR1 ~ WDTPSR0):** Watchdog timer prescaler select bits

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

#### ■ Code Example:

```

; === WDT setting 2.048sec
:
Timer1 (0.5sec wakeup)
:
BS    TR2WCON,WDTPSR1
BS    TR2WCON,WDTPSR0      ; Pre-scale 1:128
BC    CPUCON,MS1          ; Change to sleep mode
WDTC
SLEP
WDT_Loop:
    SJMP    WDT_Loop

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

## 8.5 Input/Output Key

- Three pins key input (Port A.3 ~ 0) and 16 pins key strobe (shared with LCD segment) can achieve a maximum of 64 keys matrix with automatic key scan
- Automatic key scan or software key scan
- Interrupt available under automatic key scan mode (SCAN=1)
- Wake-up available when key input falling edge is detected under automatic key scan mode (SCAN=1).

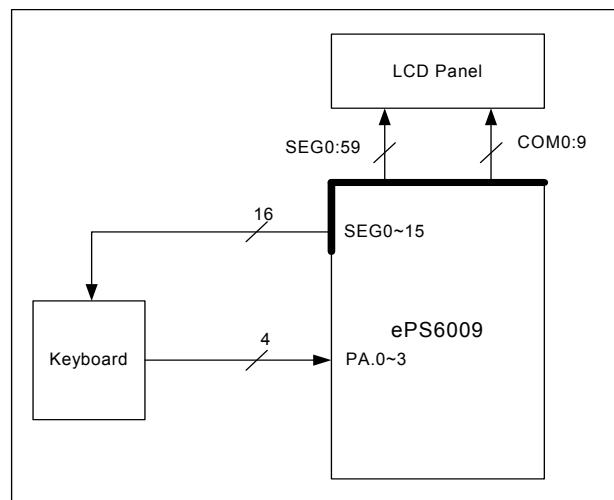


Figure 8-5a Key Function Block Diagram

As shown in the circuit diagram below, it is assumed that the key strobe output has resistance  $R_{ON}$ , while each key has resistance  $K_{ON}$  and capacitance  $C$ . A long strobe output duration will cause the LCD display to malfunction. Hence, strobe output time should be made as short as possible. Therefore,  $R_{IN}$  (pull-up resistance) should be low enough to allow quick charge to capacitor. On the contrary,  $R_{IN}$  should be high enough for  $V_{IN}$  to be considered as "L" level ( $R_{IN} \gg R_{ON} + K_{ON}$ ). Therefore, the value of  $R_{IN}$  should remain changeable.

The following is the normal key input processing:

1. Output the strobe signal
2. Pull up the input port by lowest resistance (both  $R_1$  and  $R_2$  enabled). Capacitance is charged quickly.
3. Pull up the input port by highest resistance (only  $R_2$  is enabled)
4. Read the key
5. Disable the pulled-up resistance
6. Stop the strobe signal

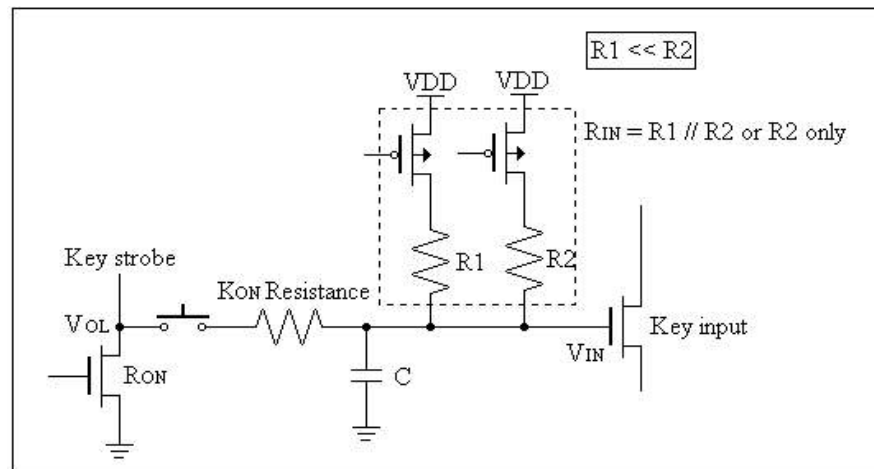


Figure 8-5b Key Circuit Diagram

### 8.5.1 Key Functions

SCAN	KE	R1EN	R2EN	IEN <sup>1</sup>	Total Pull-up Resistor	PORTA.0 ~ 3	Note
0	0	X	X	0	Floating	High-Z	
	1	0	0	1	Floating	Floating	Prohibited
		0	1	1	R2	PA.0~.3	
		1	0	1	R1	PA.0~.3	
1	x	0	0	0	R1 // R2 <sup>2</sup>	PA.0~.3	
		1	1	0	Floating	High-Z	A <sup>3</sup>
		0	1	1	R1//R2 <sup>2</sup>	High-Z	B <sup>3</sup>
		0	1	1	R2	PA.0~ 3	C <sup>3</sup>

x : Don't care

<sup>1</sup> Internal signal. Refer to the Automatic Key Scan Timing Diagram (Figure 8-6) below

<sup>2</sup>  $R1 // R2 = R1R2 / (R1+R2)$ .

<sup>3</sup> Sub clock signal. Refer to the Automatic Key Scan Timing Diagram (Figure 8-6) below

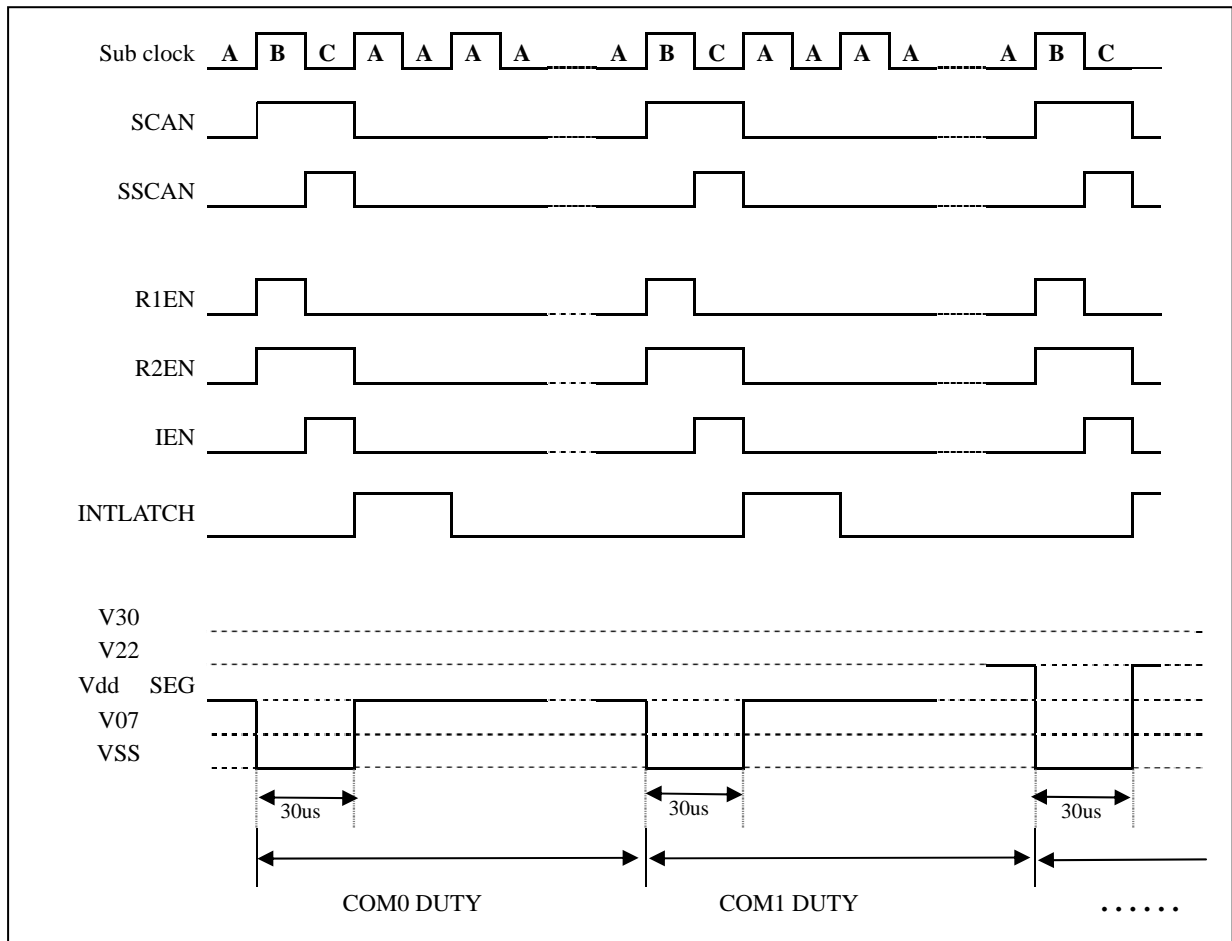


Figure 8-6 Automatic Key Scan Timing (SCAN = 1)

## 8.5.2 Key Strobe

The key strobe pin shares with LCD segment pin in the CPU embedded with LCD driver model. When sharing with LCD segment, strobe output should be as short as possible to prevent LCD display error.

There are two ways to output a strobe signal, by Automatic Key scan and by software Key scan.

### 8.5.2.1 Automatic Key Scan

The LCD waveform has a 30 $\mu$ s low pulse at the beginning of every common duty signal by setting the SCAN bit of STBCON register. The strobe timing is as shown in the following figure (Figure 8-7) on Automatic Key Scan Strobe Signal.

When in automatic key scan mode, Bits 3 ~ 0 of PAINT or PAWAKE must be enabled. During key scan, wake up and interrupt will occur if any of the falling edge of the key input pins (Port A.3 ~ Port A.0) is detected.

**NOTE**

The "SCAN" bit turn on/off in the key loop should be avoided.

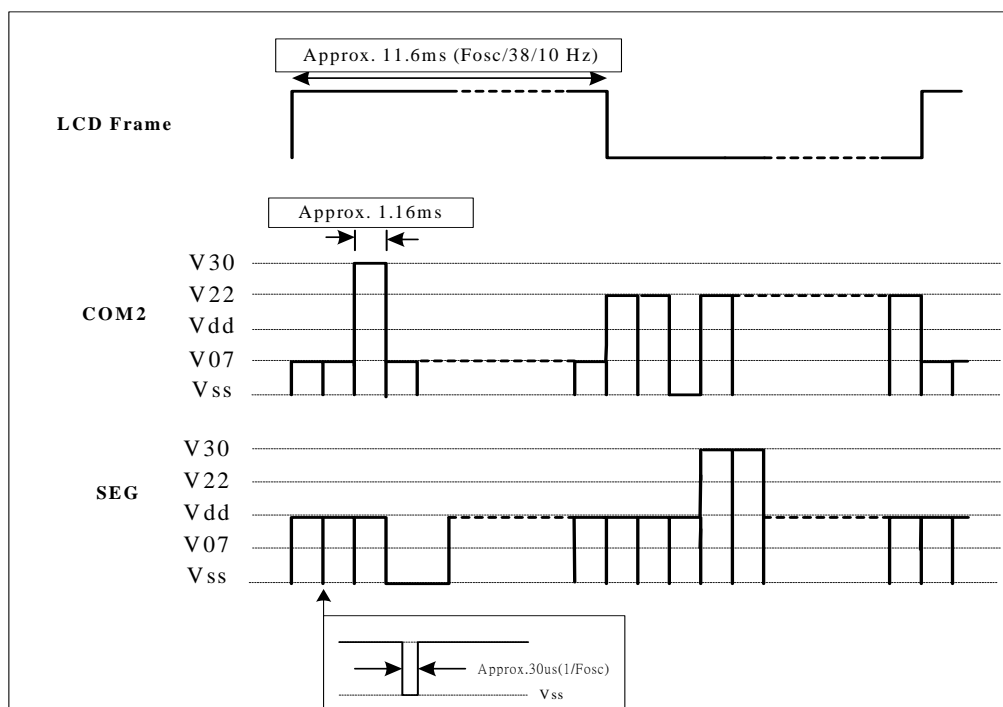


Figure 8-7 Automatic Key Scan Strobe Signal (SCAN = 1)

### 8.5.2.2 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." Set the STB3 ~ STB0 bits of the STBCON register to select which pin will be strobed.

#### ■ In Idle Mode

During automatic key scanning, if any of the falling edge of the PA.0 ~ 3 pins is detected (when PAINTEN=1), wake-up will occur. Then the CPU runs and interrupt is triggered (if enabled).

#### ■ In Slow Mode or Fast Mode

Both automatic and software key scans are applicable.

- Automatic key scan is used to determine "whether any key is pressed." If a key is pressed, PA.0~3 pin falling edge is detected, and then interrupt is triggered.
- Software key scan is used to determine "which key was pressed."

### 8.5.2.3 Key Strobe Pin Function

STBCON				Key Strobe (Shared 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	1	0	0	x	xxxx	Display waveform																Display waveform						
			0000	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		1	1				
			0001	1	0	1	1	1	1	1	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	1	1		1	1	1	1		
			0011	1	1	1	0	1	1	1	1	1	1	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	1	1		1	1	1	1	1	
			0101	1	1	1	1	1	0	1	1	1	1	1	1	1	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	1	1		1	1	1	1	1	1
			0111	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1
			1000	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1		1	1	1	1	1	1
			1001	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1		1	1	1	1	1	1
			1010	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1		1	1	1	1	1	1
			1011	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1		1	1	1	1	1	1
			1100	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1		1	1	1	1	1	1
			1101	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1		1	1	1	1	1	1
			1110	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1		1	1	1	1	1	1
			1111	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		0	1	1	1	1	1
1	x	X	xxxx	Display waveform with automatic key scan																								

### 8.5.3 Input/Output Key Registers

#### ■ Port A (R10h): Port A Register

**Bit 3 ~ Bit 0:** The input structure and two-stage pull-up resistor are controlled together by R1EN, R2EN, KE bits of the PACON register (see below).

**Bit 7:** The input structure and one-stage pull-up resistor are controlled together by PA7PU bit of the PACON register (see below).

#### ■ PACON (R29h): Port A Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
—	—	—	—	PA7PU	R2EN	R1EN	KE

**Bit 3 (PA7PU):** Enable Port A.7 pull-up resistor bits

**0:** Disable Port A.7 pull up resistor

**1:** Enable Port A.7 pull up resistor

**Bit 2 (R2EN):** R2 pull up resistor (large resistor) control bit for Port A.3 ~ Port A.0.

**0:** Disable R2 pull up resistor

**1:** Enable R2 pull up resistor

**Bit 1 (R1EN):** R1 pull up resistor (small resistor) control bit for Port A.3 ~ Port A.0.

- 0: Disable R1 pull up resistor
- 1: Enable R1 pull up resistor

**Bit 0 (KE):** Key input enable/disable control bit

- 0: Disable key input function (Port A.0~3 do **not** correspond with Key input in software scan mode)
- 1: Enable key input function (Port A.0~3 correspond with Key input in software scan mode)

■ **PAINTEN (R2Bh): Port A Interrupt Enable Control Register**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PA7IE	—	—	—	PA3IE	PA2IE	PA1IE	PA0IE

**Bit 7, Bit 3 ~ Bit 0 (PA7IE, PA3IE ~ PA0IE):** Interrupt control bit

- 0: Disable interrupt function
- 1: Enable interrupt function

■ **PAINTSTA (R2Ch): Port A Interrupt Status Register**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PA7I	—	—	—	PA3I	PA2I	PA1I	PA0I

**Bit 7, Bit 3 ~ Bit 0 (PA7I, PA3I ~ PA0I):** INT status of Port A interrupt

- Set to (1) when pin falling edge is detected, and
- Cleared to "0" by software.

■ **STBCON (R20h): 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 6 (SCAN):** Automatic key scan or specify the scan signal bit by bit

- 0: Key scan specified as Bit STB3 ~ 0 defined
- 1: Auto strobe scanning

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

- 0: Display waveform
- 1: Strobe signal specified as STB3 ~ 0 defined.

**Bits 4 (ALL):** Set all strobe.

- 0: Bit Strobe
- 1: All Strobe

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

■ **Code Example:**

```

; Key matrix 1 (Port A and Ground):
; === Sleep mode
PAIN_SR:
    :
; --- R2EN Pull-up & KE enable
    BS    PACON,R2EN
    BS    PACON,PA7PU
    BS    STBCON,KE
; --- Port A interrupt enable
    MOV   A,#10001111B
    MOV   PAINTEN,A
    CLR   PAINTSTA
    BS    CPUCON,GLINT
; --- Port A wakeup
    MOV   A,#10001111B
    MOV   PAWAKE,A
; --- Sleep MODE
    BC    CPUCON,MS1
PAINloop:
    SLEP
    NOP
    :
    SJMP PAINloop
; *** Interrupt PortA data
INPTINT:
    PUSH
    MOV   A,PAINTSTA
    MOV   Key_No,A
    CLR   PAINTSTA
    POP
    RETI

```

```

; Key matrix 2 (Port A.0~3 and SEG0 ~ SEG15):
; *** Key scan function
:
LCD display setting
:
BS    PACON,R2EN          ; R2EN enable
MOV   A,#00001111B
MOV   PAWAKE,A           ; Port A setting wakeup function
:
; === Idle mode auto key scan routine
BS    STBCON,SCAN        ; Auto-key scan enable
BS    CPUCON,MS1         ; Idle mode
KeyIdle:
SLEP
NOP
:
; === Key scan routine
KeyScan:
CLR   STBCON             ; Auto-key scan disable
KeyLoop:
BS    PACON,R1EN         ; R1EN enable
BS    PACON,KE           ; Key enable
BS    STBCON,BitST       ; Strobe ON
LCALL DLY50US
BC    PACON,R1EN         ; R1EN disable
MOVL  A,PortA           ; Port A input data
BC    STBCON,BitST       ; Strobe OFF
BC    PACON,KE           ; Key disable
JLE   A,#0X0E,KeyScan    ; If A <= PORTA Goto KeyScan
INC   STBCON
SJMP  KeyLoop
KeyScan:
; --- Clear key number
CLR   Key_No
; --- Key Scan is finish
KeyScanOk:
MOV   Key_No,A
MOV   A,STBCON
MOVH  Key_No,A          ; Key_No: XXXX XXXX
:

```

## 8.6 LCD Driver

The ePS6009 provides directly driven LCD. It supports multiplexed drive for 60SEGs × 10COMs which allows the use of pads as an LCD driver pin or as key input port. The available LCD RAM corresponds directly with LCD Pixel. The LCD voltage is 2xVDD. The LCD frame rate is as follows.

Duty	LCD Frame Rate
1/10	Approx. 11.6ms (Fosc/38/10 Hz)

This embedded LCD driver generates waveforms to drive the display.

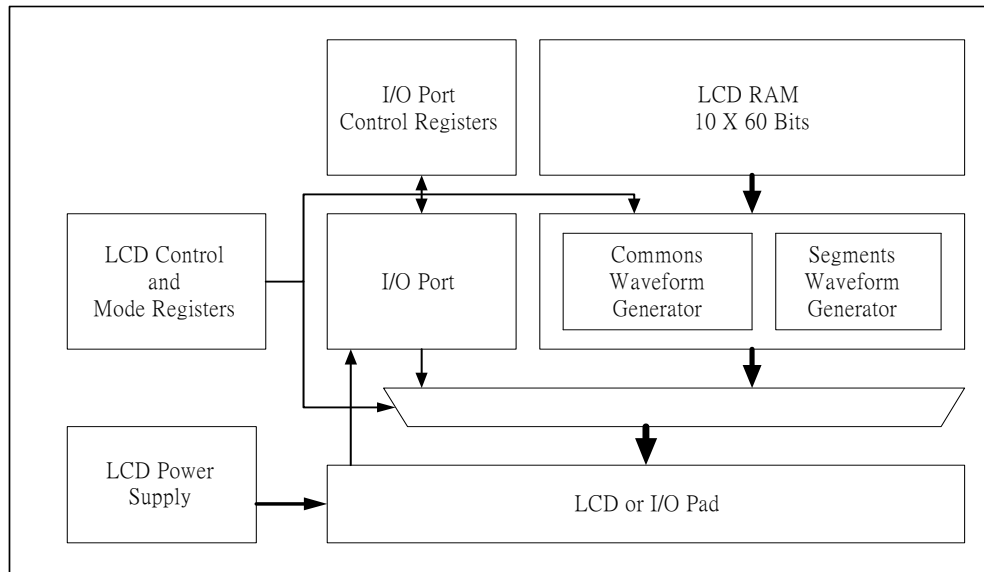


Figure 8-8a LCD Driver Function Block Diagram

The following is the LCD pin configuration:

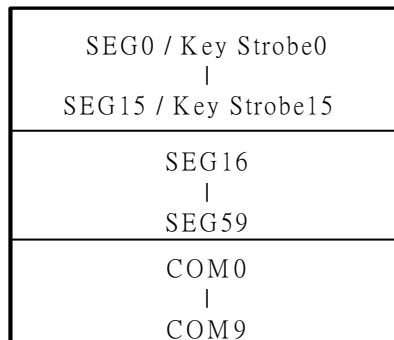


Figure 8-8b LCD Pin Configuration

### 8.6.1 LCD Driver Registers

■ **LCDCON (R2Fh): LCD Control Register**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
REV	BLANK	LCDCON	-	LCR1	LCR0	-	LBVON

- Bit 7 (REV):** LCD display status control bit  
**0:** Normal → LCD data = 1 means LCD dot light  
**1:** Inverse → LCD data = 1 means LCD dot dark

- Bit 6 (BLANK):** LCD Blanking control bit  
**0:** Disable  
**1:** Enable (All SEG pins output "0" signal)

**Bit 5 (LCDON):** LCD display control bit

**0:** LCD display off

**1:** LCD display on

**NOTE**

*All COM & SEG pins are tied to ground when LCD display is off.*

**Bit 3, Bit 2 (LCR1, LCR0):** LCD Bias Voltage Charge-pump Rate select bits

LCR1:LCR0	Charge-Pump Rate(Hz)
00	8K
01	4K
10	2K
11	16K

**Bit 0 (LBVON):** Bias Voltage Charge-pump control bit

**0:** Disable

**1:** Enable

■ **LCDARL (R09h):** LCD RAM Column Address Register (see LCD RAM Map)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LCDARL7	LCDARL6	LCDARL5	LCDARL4	LCDARL3	LCDARL2	LCDARL1	LCDARL0

■ **LCDDATA (R0Eh):** LCDDATA register is an indirect address pointer of LCD RAM  
Any instruction that uses LCDDATA as register, actually accesses LCD RAM via the address pointed by LCDARL (see figure below).

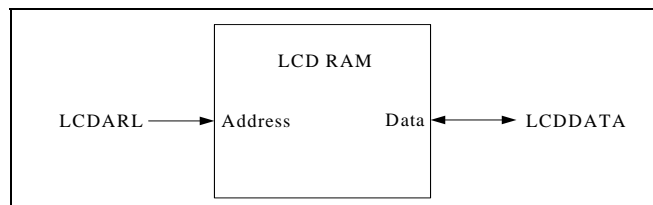


Figure 8-9 LCDDATA Register Access through LCD RAM

■ **POST\_ID (R30h):** Post Increase / Decrease 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

After accessing (read or write) the LCD RAM, the LCDARL register can be automatically increased or decreased by setting the POST\_ID register.

**Bit 6 (LCD\_ID):** Set to "1" to auto-increase the LCDARL register

Reset to "0" to auto-decrease the LCDARL register

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

■ **Code Example:**

```

; === LCD Setting
L_Initial:
; --- LCD Off, Normal Display Mode, Charge-Pump rate=8K
    MOV    A,#0000001B
    MOV    LCDCON,A
    SCALL  DspRAMdot
; --- LCD turn-on
    BS     LCDCON,LCDON
    LCALL  Delay1sec
    :
DspLoop:
; --- LCD Blanking
    BS     LCDCON,BLANK
    LCALL  Delay1sec
; --- Normal display
    BC     LCDCOM,BLANK
    LCALL  Delay1sec
    :
    SJMP   DspLoop
; *** Display LCD RAM is data 55 & AA
DspRAMdot:
; --- LCD increase enable.
    BS     POST_ID,LCDPE
    BS     POST_ID,LCD_ID
DspRAMd1:
    CLR    LCDARL
    TBPTH  #0x1E
; === Write LCD RAM is dot matrix
WrLRAMd:
    MOV    A,#0XAA
    MOV    LCDDATA,A
    MOV    A,#0X55
    MOV    LCDDATA,A
    JDNZ   TABPTRH, WrLRAMd
    CLR    LCDARL
    RET

```

### 8.6.2 LCD RAM Map

■ 1/10 Duty

RAM Address		COM 0	COM 1	COM 2	COM 3	COM 4	COM 5	COM 6	COM 7
LCDARL		Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
SEG0	00H								
:	:								
SEG59	3BH								

RAM address		COM 8	COM 9
LCDARL		Bit 0	Bit 1
SEG0	40H		
:	:		
SEG59	7BH		

### 8.6.3 LCD Driving Method Circuit

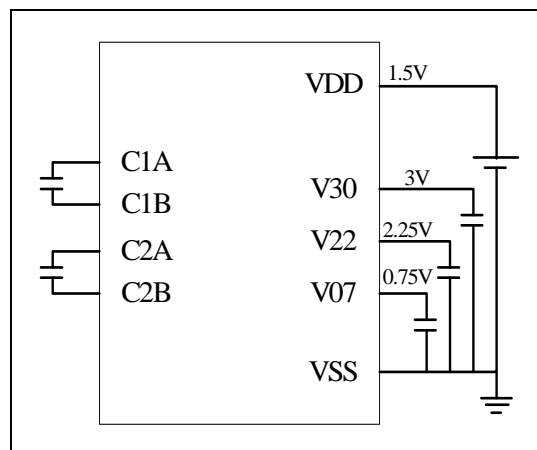


Figure 8-10 LCD Driving Method Circuit for 1/4 Bias

### 8.6.4 LCD COM Waveforms

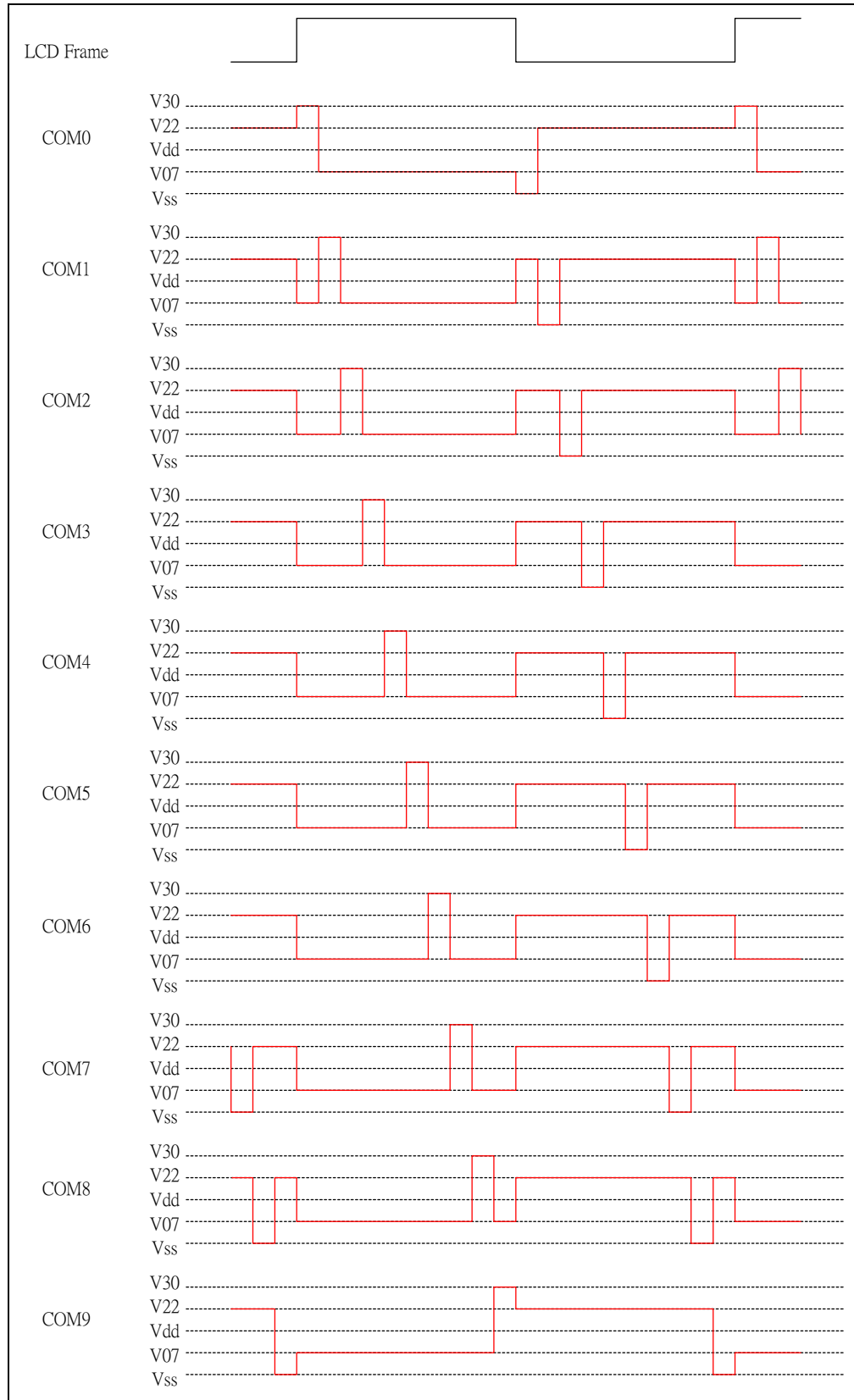


Figure 8-11 LCD COM Waveform

### 8.6.5 LCD COM and SEG Waveforms

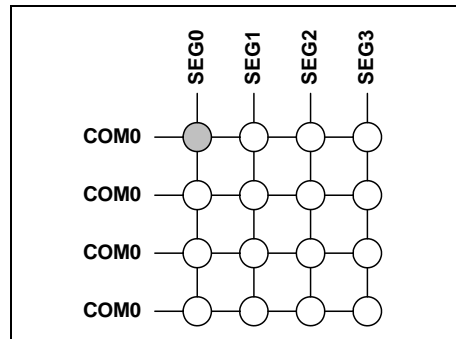


Figure 8-14a LCD COM & SEG Waveform Matrix

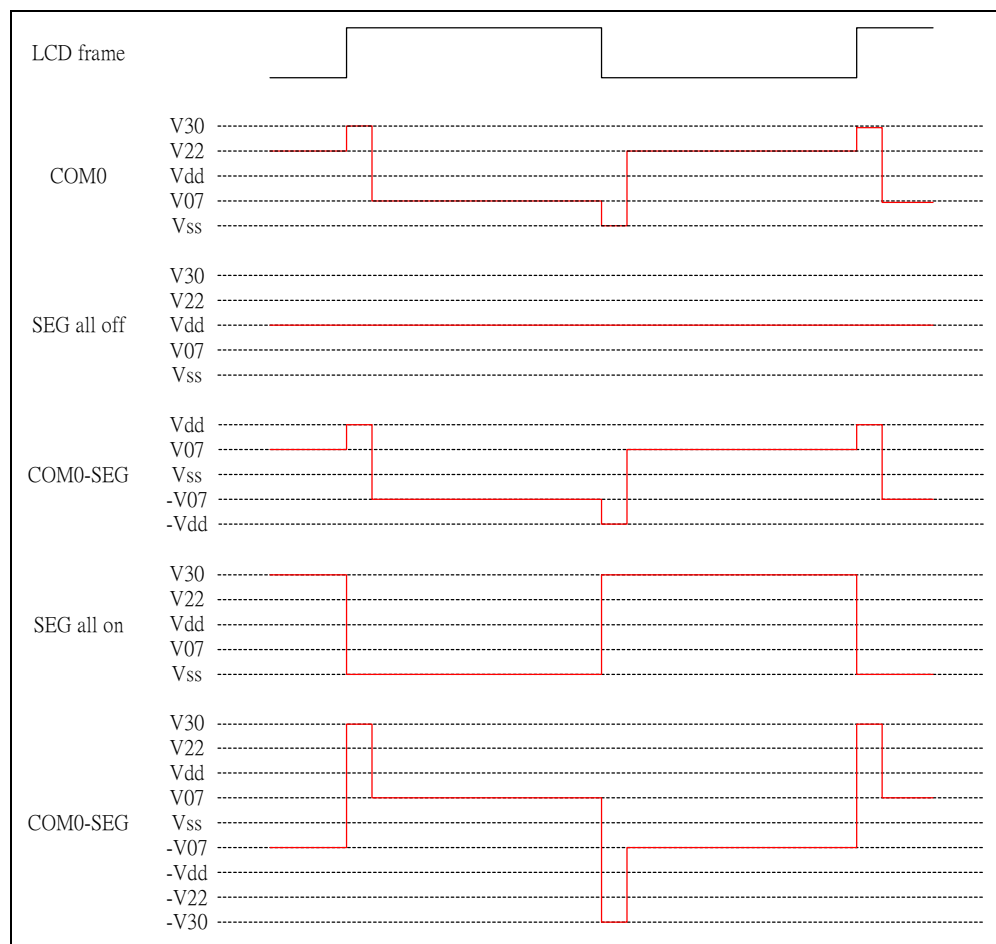


Figure 8-14b LCD COM & SEG Waveform

## 9 Electrical Characteristics

### ■ Absolute Maximum Ratings

Items	Sym.	Condition	Limits	Unit
Supply voltage	VDD	-	-0.3 to +2.0	V
Input voltage (general input port)	VIN	-	-0.5 to VDD +0.5	V
Operating temperature range	TOPR	-	-10 to +70	°C
Storage temperature range	TSTR	-	-55 to +125	°C

### ■ Recommended Operating Conditions

Items	Sym.	Condition	Limits	Unit
Supply voltage	VDD	-	1.2 to 1.8	V
Input voltage	VIH	-	VDD x 0.9 to VDD	V
	VIL	-	0 to VDD x 0.1	V
Operating temperature	TOPR	-	-10 to +70	°C

### ■ DC Electrical Characteristics (Condition: Ta=25°C, VDD= 1.5V)

Parameter	Sym.	Condition		Min	Typ	Max	Unit
Clock	F <sub>HOSC</sub>	Main-clock frequency	RC OSC. R=910KΩ	140	200	260	kHz
			RC OSC. R=620KΩ	210	300	390	
			RC OSC. R=360KΩ	350	500	650	
	F <sub>osc</sub>	Sub-clock frequency	Internal RC OSC	24.6	32.8	41	kHz
External RC OSC. R=3MΩ			24.6	32.8	41		
Supply Current	I <sub>dd1</sub>	Sleep mode	VDD=1.5V, no load	-	-	1	μA
	I <sub>dd2</sub>	Idle mode	VDD=1.5V, RC / Crystal OSC, LCD enabled, no load	-	4	10	
	I <sub>dd3</sub>	Slow mode	VDD=1.5V, RC OSC, LCD disabled, no load	-	5	11	
			VDD=1.5V, RC OSC, LCD enabled, no load	-	5	11	
	I <sub>dd5</sub>	Fast mode	VDD=1.5V, F <sub>HOSC</sub> =200kHz, LCD enabled, no load	-	16	30	
			VDD=1.5V, F <sub>HOSC</sub> =300kHz, LCD enabled, no load	-	20	40	
			VDD=1.5V, F <sub>HOSC</sub> =500kHz, LCD enabled, no load	-	30	50	
Input Voltage	VIH1	PA [0:3], PA [7], PB[0:1] (as general input port)		VDD×0.7	-	VDD	V
	VIL1			0	-	VDD×0.3	

(Continued)

Parameter	Sym.	Condition		Min	Typ	Max	Unit	
Input Threshold Voltage (Schmitt)	VT+	RSTB		0.5×VDD	-	0.75×VDD	V	
	VT-			0.2×VDD	-	0.4×VDD		
Input Leakage Current	IIL	ALL Input port (without pull up/down resistor) Vin= VDD or GND		-	-	±1	μA	
Large Pull-up Resistance	RPU5	RSTB	Vin=GND	300	400	560	KΩ	
Small Pull-up Resistance	RPU6	RSTB	Vin=1V	10	30	60	KΩ	
Large Pull-down Resistance	RPD1	TEST	Vin=VDD	250	500	750	KΩ	
Small Pull-down Resistance	RPD2	TEST	Vin=0.5V	3	6	12	KΩ	
Output Current	IOH1	PB [0:1] (as general output port)	VDD=1.5V, VOH=1.2V, LCD enabled	-0.7	-1	-1.3	mA	
	IOL1		VDD=1.5V, VOL=0.2V, LCD enabled	0.6	0.9	1.2		
Large Pull-up Resistance	RPU1	PA [0:3]	Key high resistance, pulled up by R2, LCD enabled, Vin2=0.5V	150	300	450	KΩ	
	RPU3	PB [0:1]	Vin=0.5V, LCD enabled	90	180	270		
Small Pull-up Resistance	RPU2	PA [0:3]	Key high resistance, pulled up by R2//R1, LCD enabled, Vin2=0 V	40	80	120	KΩ	
	RPU4	PA [7]	Vin=1V, LCD enabled	10	30	60		
Data Retention Voltage	Vret	-		1.2	-	-	V	
Power on reset voltage	Vpor	-		0.9	1.0	1.0	V	
<b>LCD Driver</b>								
LCD Display Output ON-resistance	ROC	Com [0:4]	VOH=V30 ± 0.2V	1	2	3	KΩ	
			VOM=V22 ± 0.2V					
			VOM=V07 ± 0.2V					
			VOL=0.2V					
Strobe Output ON-resistance	ROP	Seg [0:15] (as key strobe)	V=VDD-0.2V	100	200	300	KΩ	
			RON	V=0.2V	0.7	1.3		2

## 10 Pin Type Circuit Diagrams

### ■ Reset Pin Type

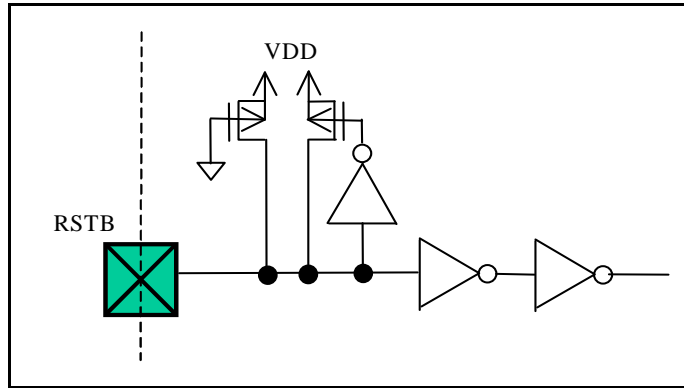


Figure 10-1a Reset Pin Type Circuit Diagram

### ■ Test Pin Type

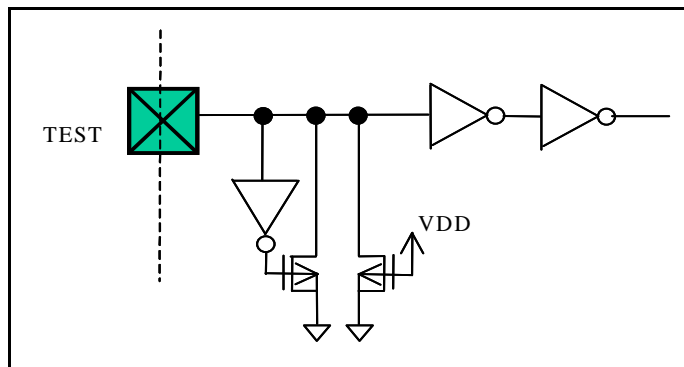


Figure 10-1b Test Pin Type Circuit Diagram

### ■ Oscillator Pin Type

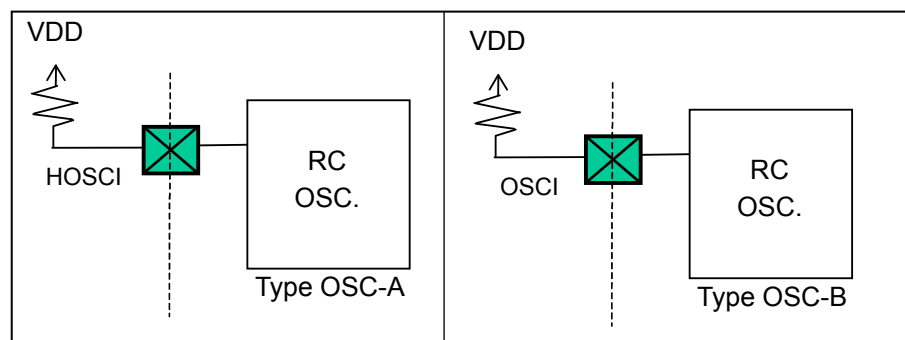


Figure 10-1c Oscillator Pin Type Circuit Diagram

■ Input Pin Type(PA.0~3, 7)

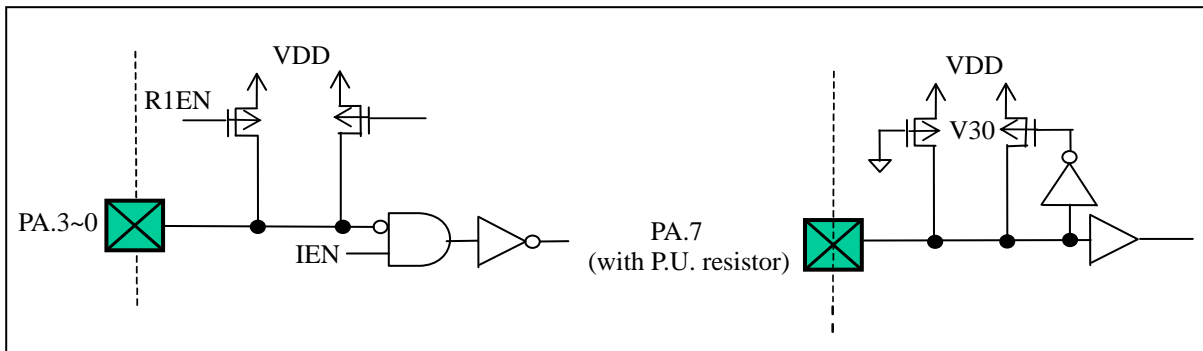


Figure 10-1d Input Pin Type Circuit Diagram

■ I/O Pin Type(PB.0~1)

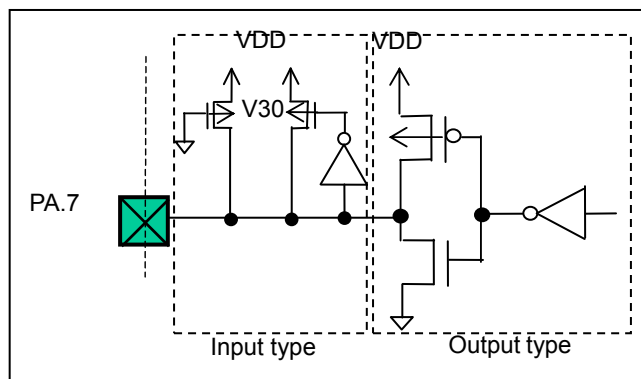


Figure 10-1e I/O Pin Type Circuit Diagram

■ General SEG and COM Pin Type

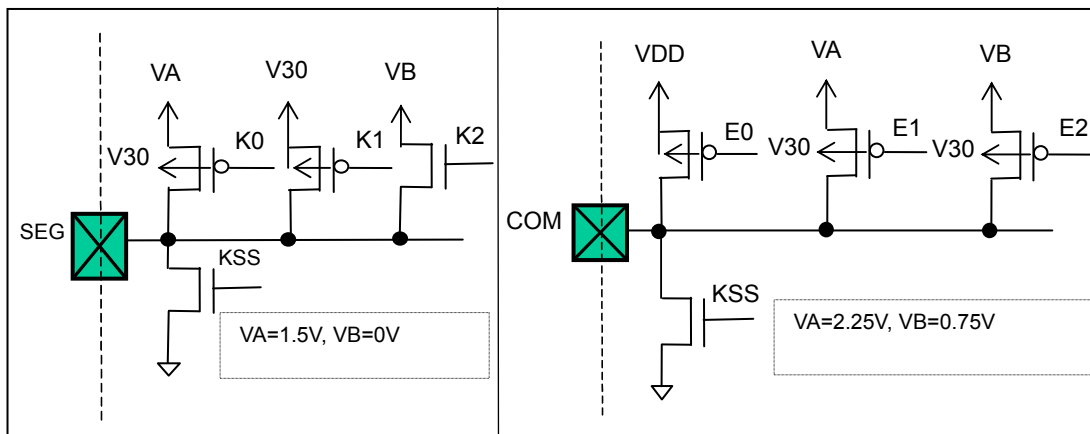


Figure 10-1f General SEG & COM Share Pin Type Circuit Diagram

## 11 Application Circuit

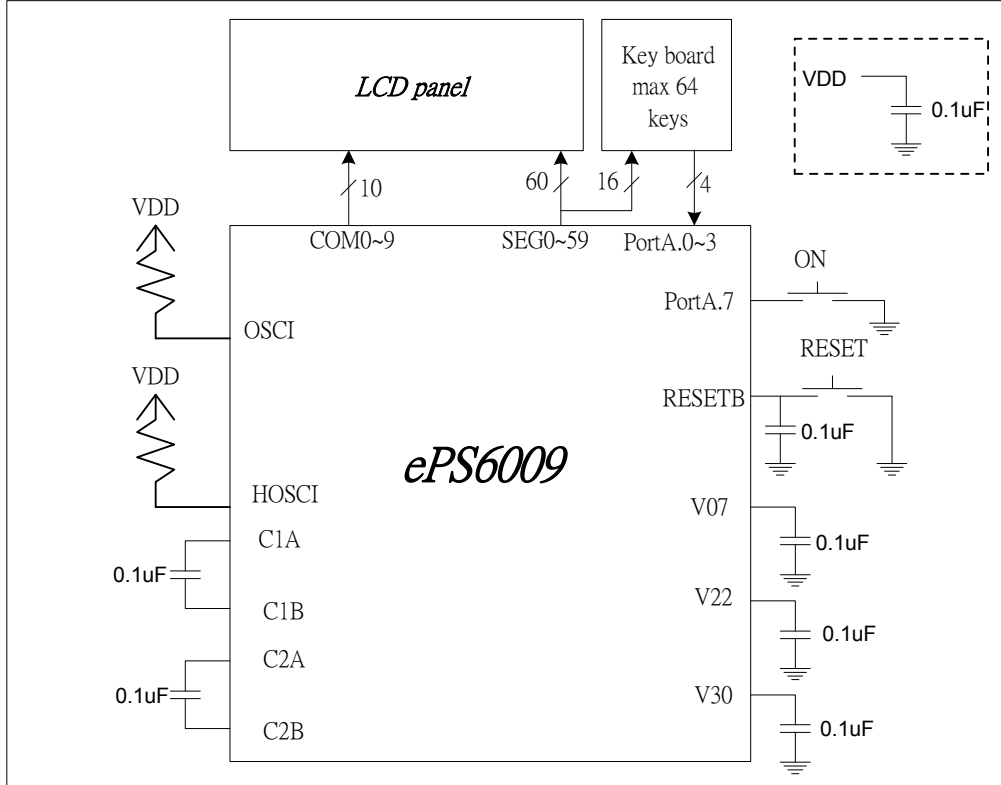


Figure 11-1 Application Circuit Diagram

## 12 Instruction Set

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

Type	Binary Instruction	Mnemonic	Operation	Status Affected	Cycle
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 Enter SLEEP MODE if MS1=0	None	1
	0010 0111 rrrr rrrr	RPT r ("r" is the content of register r)	Single repeat (r) times on next instruction	None	1
	0100 0011 kkkk kkkk	BANK #k	BSR ← k	None	1
Subroutine	0011 aaaa aaaa aaaa	S0CALL addr	Top of Stack] ← PC+1 PC[11:0] ← addr PC[12:16] ← 00000 *1	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 0000 aaaa aaaa aaaa aaaa	LCALL addr (two 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
Compare	0010 0101 rrrr rrrr	TEST r	Z ← 0 if r <> 0; Z ← 1 if r = 0	Z	1
Jump	110a aaaa aaaa aaaa	SJMP addr	PC ← addr PC[13..15] unchange	None	1
	0000 0000 0010 0000 aaaa aaaa aaaa aaaa	LJMP addr (two words)	PC ← addr	None	2
Compare & 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 *2	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 *2	None	2
	0100 0111 kkkk kkkk aaaa aaaa aaaa aaaa	JGE A,#k,addr	Jump to addr if A ≥ k PC[15:0] ← addr *2	None	2
	0100 1000 kkkk kkkk aaaa aaaa aaaa aaaa	JLE A,#k,addr	Jump to addr if A ≤ k PC[15:0] ← addr *2	None	2
	0100 1001 kkkk kkkk aaaa aaaa aaaa aaaa	JE A,#k,addr	Jump to addr if A = k PC[15:0] ← addr *2	None	2
	0101 0101 rrrr rrrr aaaa aaaa aaaa aaaa	JGE A,r,addr	Jump to addr if A ≥ r PC[15:0] ← addr *2	None	2
	0101 0110 rrrr rrrr aaaa aaaa aaaa aaaa	JLE A,r,addr	Jump to addr if A ≤ r PC[15:0] ← addr *2	None	2
	0101 0111 rrrr rrrr aaaa aaaa aaaa aaaa	JE A,r,addr	Jump to addr if A = r PC[15:0] ← addr *2	None	2
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 *2	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 *2	None	2

(Continued)

Type	Binary Instruction	Mnemonic	Operation	Status Affected	Cycle
Data Transfer	0010 0000 rrrr rrrr	MOV A,r	$A \leftarrow r$	Z	1
	0010 0001 rrrr rrrr	MOV r,A	$r \leftarrow A$	None	1
	100p pppp rrrr rrrr	MOVPR p,r	Register p $\leftarrow$ Register r	None	1
	101p pppp rrrr rrrr	MOVPR r,p	Register r $\leftarrow$ Register p	None	1
	0100 1110 kkkk kkkk	MOV A,#k	$A \leftarrow k$	None	1
	0010 0100 rrrr rrrr	CLR r	$r \leftarrow 0$	Z	1
Rom Table Look Up	0100 0000 kkkk kkkk	TBPTL #k	TABPTRL $\leftarrow$ k	None	1
	0100 0001 kkkk kkkk	TBPTM #k	TABPTRM $\leftarrow$ k	None	1
	0100 0010 kkkk kkkk	TBPTH #k	TABPTRH $\leftarrow$ k	None	1
	0010 11ii rrrr rrrr	TBRD i,r	$r \leftarrow \text{ROM}[(\text{TABPTR})^{*3, *4}]$	None	2
	0010 1111 rrrr rrrr	TBRD A,r	$r \leftarrow \text{ROM}[(\text{TABPTR}+\text{ACC})^{*4}]$	None	2
Logic Operation	0000 0010 rrrr rrrr	OR A,r	$A \leftarrow A .\text{or. } r$	Z	1
	0000 0011 rrrr rrrr	OR r,A	$r \leftarrow r .\text{or. } A$	Z	1
	0100 0100 kkkk kkkk	OR A,#k	$A \leftarrow A .\text{or. } k$	Z	1
	0000 0100 rrrr rrrr	AND A,r	$A \leftarrow A .\text{and. } r$	Z	1
	0000 0101 rrrr rrrr	AND r,A	$r \leftarrow r .\text{and. } A$	Z	1
	0100 0101 kkkk kkkk	AND A,#k	$A \leftarrow A .\text{and. } k$	Z	1
	0000 0110 rrrr rrrr	XOR A,r	$A \leftarrow A .\text{xor. } r$	Z	1
	0000 0111 rrrr rrrr	XOR r,A	$r \leftarrow r .\text{xor. } A$	Z	1
	0100 0110 kkkk kkkk	XOR A,#k	$A \leftarrow A .\text{xor. } k$	Z	1
	0000 1000 rrrr rrrr	COMA r	$A \leftarrow /r$	Z	1
	0000 1001 rrrr rrrr	COM r	$r \leftarrow /r$	Z	1
Arithmetic Operation	0001 1100 rrrr rrrr	INCA r	$A \leftarrow r+1$	C,Z	1
	0001 1101 rrrr rrrr	INC r	$r \leftarrow r+1$	C,Z	1
	0001 0000 rrrr rrrr	ADD A,r	$A \leftarrow A+r$	C,DC,Z,OV,SGE,SLE	1
	0001 0001 rrrr rrrr	ADD r,A	$r \leftarrow r+A^{*5}$	C,DC,Z,OV,SGE,SLE	1
	0100 1010 kkkk kkkk	ADD A,#k	$A \leftarrow A+k$	C,DC,Z,OV,SGE,SLE	1
	0001 0010 rrrr rrrr	ADC A,r	$A \leftarrow A+r+C$	C,DC,Z,OV,SGE,SLE	1
	0001 0011 rrrr rrrr	ADC r,A	$r \leftarrow r+A+C$	C,DC,Z,OV,SGE,SLE	1
	0100 1011 kkkk kkkk	ADC A,#k	$A \leftarrow A+k+C$	C,DC,Z,OV,SGE,SLE	1
	0001 1110 rrrr rrrr	DECA r	$A \leftarrow r-1$	C,Z	1
	0001 1111 rrrr rrrr	DEC r	$r \leftarrow r-1$	C,Z	1
	0001 0110 rrrr rrrr	SUB A,r	$A \leftarrow r-A^{*6}$	C,DC,Z,OV,SGE,SLE	1
	0001 0111 rrrr rrrr	SUB r,A	$r \leftarrow r-A^{*6}$	C,DC,Z,OV,SGE,SLE	1
	0100 1100 kkkk kkkk	SUB A,#k	$A \leftarrow k-A^{*6}$	C,DC,Z,OV,SGE,SLE	1
	0001 1000 rrrr rrrr	SUBB A,r	$A \leftarrow r-A-/C^{*6}$	C,DC,Z,OV,SGE,SLE	1
	0001 1001 rrrr rrrr	SUBB r,A	$r \leftarrow r-A-/C^{*6}$	C,DC,Z,OV,SGE,SLE	1
	0100 1101 kkkk kkkk	SUBB A,#k	$A \leftarrow k-A-/C^{*6}$	C,DC,Z,OV,SGE,SLE	1
	0001 0100 rrrr rrrr	ADDDC A,r	$A \leftarrow (\text{Decimal ADD}) A+r+C$	C,DC,Z	1
	0001 0101 rrrr rrrr	ADDDC r,A	$r \leftarrow (\text{Decimal ADD}) r+A+C$	C,DC,Z	1
	0001 1010 rrrr rrrr	SUBDB A,r	$A \leftarrow (\text{Decimal SUB}) r-A-/C$	C,DC,Z	1
0001 1011 rrrr rrrr	SUBDB r,A	$r \leftarrow (\text{Decimal SUB}) r-A-/C$	C,DC,Z	1	

(Continuation)

Type	Binary Instruction	Mnemonic	Operation	Status Affected	Cycle
Rotate	0000 1010 rrrr rrrr	RRCA r	$A(n-1) \leftarrow r(n); C \leftarrow r(0); A(7) \leftarrow C$	C	1
	0000 1011 rrrr rrrr	RRC r	$r(n-1) \leftarrow r(n); C \leftarrow r(0); r(7) \leftarrow C$	C	1
	0000 1100 rrrr rrrr	RLCA r	$A(n+1) \leftarrow r(n); C \leftarrow r(7); A(0) \leftarrow C$	C	1
	0000 1101 rrrr rrrr	RLC r	$r(n+1) \leftarrow r(n); C \leftarrow r(7); r(0) \leftarrow C$	C	1
Shift	0010 0010 rrrr rrrr	SHRA r	$A(n-1) \leftarrow r(n); A(7) \leftarrow C$	None	1
	0010 0011 rrrr rrrr	SHLA r	$A(n+1) \leftarrow r(n); A(0) \leftarrow C$	None	1
Exchange	0101 0100 rrrr rrrr	EX r	$r(7-0) \leftrightarrow A(7-0)$	None	1
Bit Manipulation	0110 1bbb rrrr rrrr	BC r,b	$r(b) \leftarrow 0$	None	1
	0111 0bbb rrrr rrrr	BS r,b	$r(b) \leftarrow 1$	None	1
	0111 1bbb rrrr rrrr	BTG r,b	$r(b) \leftarrow /r(b)$	None	1
Nibble Operation	0101 0010 rrrr rrrr	EXL r	$r(3-0) \leftrightarrow A(3-0)$	None	1
	0101 0011 rrrr rrrr	EXH r	$r(7-4) \leftrightarrow A(3-0)$	None	1
	0010 0110 rrrr rrrr	MOVL r,A	$r(3-0) \leftarrow A(3-0)$	None	1
	0010 1000 rrrr rrrr	MOVH r,A	$r(7-4) \leftarrow A(3-0)$	None	1
	0010 1001 rrrr rrrr	MOVL A,r	$A(3-0) \leftarrow r(3-0); A(7-4) \leftarrow 0$	None	1
	0010 1010 rrrr rrrr	MOVH A,r	$A(3-0) \leftarrow r(7-4); A(7-4) \leftarrow 0$	None	1
	0000 0001 rrrr rrrr	SFR4 r	$r(7-4) \leftarrow A(3-0); r(3-0) \leftarrow r(7-4); A(3-0) \leftarrow r(3,0)$	None	1
	0100 1111 rrrr rrrr	SFL4 r	$r(3-0) \leftarrow A(3-0); r(7-4) \leftarrow r(3-0); A(3-0) \leftarrow r(7-4)$	None	1
	0000 1111 rrrr rrrr	SWAP r	$r(0:3) \leftrightarrow r(4:7)$	None	1
0000 1110 rrrr rrrr	SWAPA r	$r(0:3) \rightarrow A(4:7); r(4:7) \rightarrow A(0:3)$	None	1	

\*1 SOCALL addressing ability is from 0x000 to 0xFFFF (4K words space)

\*2 The maximum jump range is 32K words absolute address

\*3 TBRD i, r :

$r \leftarrow ROM [(TABPTR)]$

i=00: TABPTR no change

wi=01: TABPTR  $\leftarrow$  TABPTR+1

i=10: TABPTR  $\leftarrow$  TABPTR-1

\*4 TABPTR = (TABPTRM: TABPTRL)

Bit 0 = 0: Low byte of the pointed ROM data

Bit 0 = 1: High byte of the pointed ROM data

#### NOTE

- Bit 0 of TABPTRL is used to select either low byte or high byte of the pointed ROM data.
- The maximum table look up space is internal 64K bytes (32K words).

\*5 Carry bit of "ADD PCL, A" or "ADD TABPTRL, A" will automatically carry into PCM or TABPTRM. The Instruction cycle of write to PC (program counter) takes two cycles.

\*6 When in SUB operation, borrow flag is indicated by the inverse of carry bit, that is B=/C

## 13 Pad Diagram

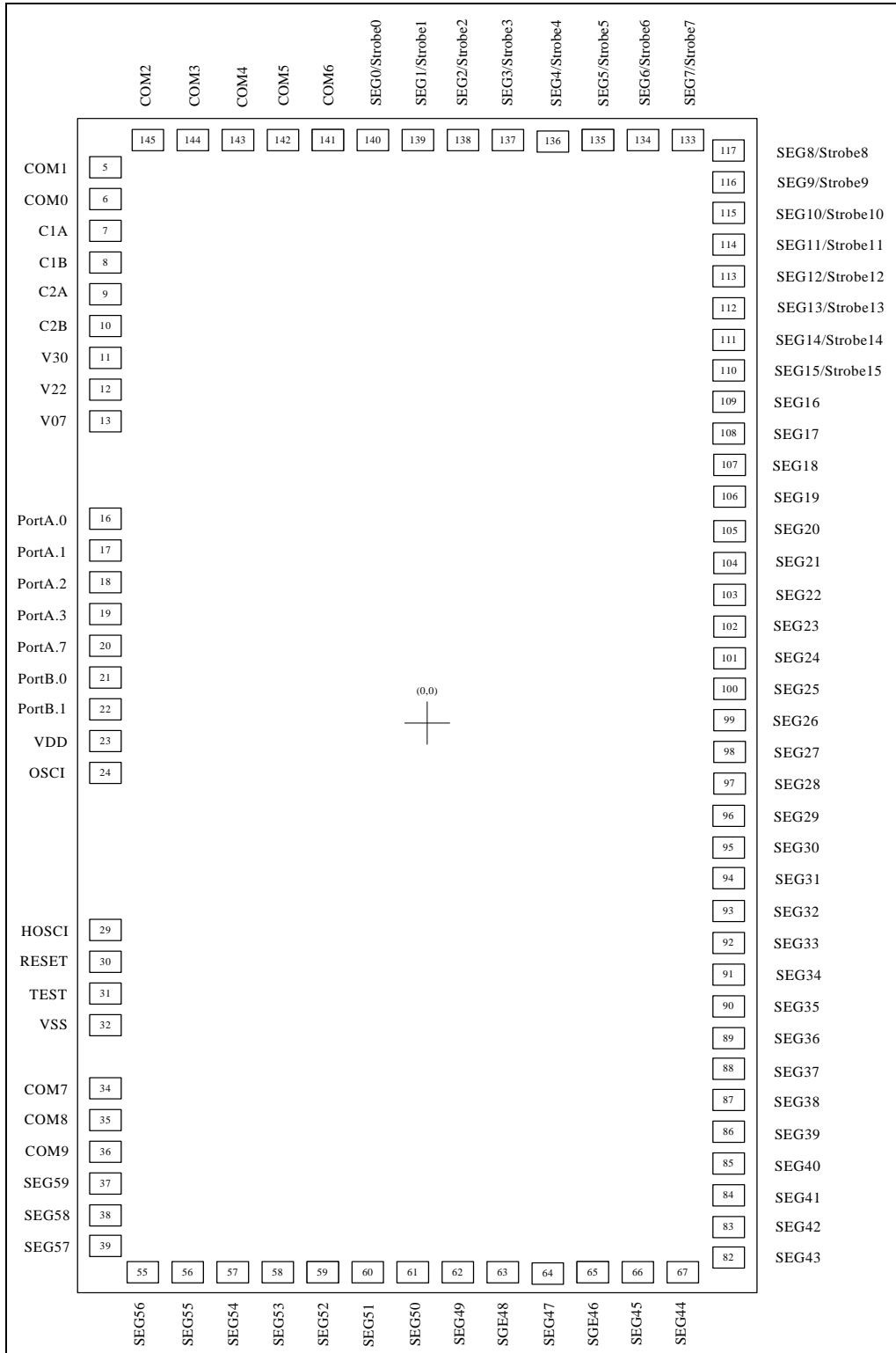


Figure 13-1 ePS6009 Pad Diagram

## 14 Pad Coordinates

Chip Size: 1750 × 453 μm<sup>2</sup>

Pin No	Pad Name	Coordinate ( X , Y )	Pin No	Pad Name	Coordinate ( X , Y )
1			26		
2			27		
3			28		
4			29	HOSCI	-765.0 , -828.2
5	COM_1_	-765.0 , 2124.7	30	RESETB	-765.0 , -938.2
6	COM_0_	-765.0 , 1999.7	31	TEST	-765.0 , -1048.2
7	C1A	-765.0 , 1879.6	32	GND	-765.0 , -1158.2
8	C1B	-765.0 , 1759.6	33		
9	C2A	-765.0 , 1644.6	34	COM_7_	-765.0 , -1514.0
10	C2B	-765.0 , 1529.6	35	COM_8_	-765.0 , -1624.0
11	V30	-765.0 , 1414.6	36	COM_9_	-765.0 , -1736.0
12	V22	-765.0 , 1299.7	37	SEG_59_	-765.0 , -1851.0
13	V07	-765.0 , 1189.6	38	SEG_58_	-765.0 , -1971.0
14			39	SEG_57_	-765.0 , -2096.0
15			40		
16	PA_0_	-765.0 , 673.3	41		
17	PA_1_	-765.0 , 558.3	42		
18	PA_2_	-765.0 , 443.3	43		
19	PA_3_	-765.0 , 328.3	44		
20	PA_7_	-765.0 , 213.4	45		
21	PB_0_	-765.0 , 98.4	46		
22	PB_1_	-765.0 , -16.6	47		
23	VDD	-765.0 , -131.6	48		
24	OSCI	-765.0 , -241.6	49		
25			50		

Pin No	Pad Name	Coordinate ( X , Y )	Pin No	Pad Name	Coordinate ( X , Y )
51			76		
52			77		
53			78		
54			79		
55	SEG_56_	-618.2 , -2140.0	80		
56	SEG_55_	-513.2 , -2140.0	81		
57	SEG_54_	-408.2 , -2140.0	82	SEG_43_	765.0 , -1960.5
58	SEG_53_	-303.2 , -2140.0	83	SEG_42_	765.0 , -1840.5
59	SEG_52_	-198.2 , -2140.0	84	SEG_41_	765.0 , -1720.5
60	SEG_51_	-93.2 , -2140.0	85	SEG_40_	765.0 , -1605.5
61	SEG_50_	11.8 , -2140.0	86	SEG_39_	765.0 , -1490.5
62	SEG_49_	116.8 , -2140.0	87	SEG_38_	765.0 , -1375.5
63	SEG_48_	221.8 , -2140.0	88	SEG_37_	765.0 , -1260.5
64	SEG_47_	326.8 , -2140.0	89	SEG_36_	765.0 , -1145.5
65	SEG_46_	431.8 , -2140.0	90	SEG_35_	765.0 , -1030.5
66	SEG_45_	536.8 , -2140.0	91	SEG_34_	765.0 , -920.5
67	SEG_44_	651.8 , -2140.0	92	SEG_33_	765.0 , -810.5
68			93	SEG_32_	765.0 , -700.5
69			94	SEG_31_	765.0 , -590.5
70			95	SEG_30_	765.0 , -480.5
71			96	SEG_29_	765.0 , -370.5
72			97	SEG_28_	765.0 , -260.5
73			98	SEG_27_	765.0 , -150.5
74			99	SEG_26_	765.0 , -40.5
75			100	SEG_25_	765.0 , 64.5



