ePS6000

RISC II Series Microcontroller

Product Specification

DOC. VERSION 1.7

ELAN MICROELECTRONICS CORP.

August 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
elanhk@emc.com.hk

USA:

Elan Information Technology Group (U.S.A.)
P.O. 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

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

Contents

Specification Revision History

Doc. Version	Revision Description	Date
1.0	Modify DC electrical characteristic Modify DC electrical characteristic Add the power current of Fast mode (200kHz) Modify DC electrical characteristic Modify pin assignment	2004/11/30 2004/12/09 2004/12/10 2004/12/27 2004/12/31
1.1	Add LCD driving methods Modify the value for each bias voltage of LCD	2005/01/18 2005/02/03
1.2	Add Pad Diagram and modify pin assignment	2005/03/11
1.3	Modify DC electrical characteristic (VDD=3V), Key I/O code example Add the 1/5 duty to LCD driver	2005/05/30 2005/08/21
1.4	Add Crystal stable timing and Slow to Fast mode Timing diagrams	2005/09/26
1.5	Add Fast to Slow mode Code Example to MCU Operation Mode (Section 7.3.1)	2005/11/01
1.6	Modify DC electrical characteristic (VDD=1.5V): Supply current Modify Application circuit	2006/07/03
1.7	Modify the pop interrupt register of the code example (Section 7.4.1) Modified the max. supply voltage on section 9 Added a Note on Section 6 Code Option	2008/04/07 2008.08.26

1 General Description

IMPORTANT NOTES !!

- *Do not use Register BSR (05h) Bit7 ~ Bit4.*
- *Do not use Register BSR1 (07h) Bit7 ~ Bit4.*
- *Do not use LCD RAM 3Ch ~ 3Fh.*
- *Do not use Registers JDNZ at FSR1 (04h) special register.*

ePS6000 is an 8-bit RISC MCU embedded with an 11*60 LCD driver along with two 8-bit timers, one 16-bit general timer, and a watch dog timer. Furthermore, ePS6000 is equipped with 2K bytes RAM and 16K words program ROM. It is highly ideal for advance scientific calculator application, particularly those that need a high performance at low cost solution.

The MCU core is a one of ELAN's second generation RISC based ICs, known as RISC II (RII) series. The core was specifically designed for low power and portable device applications. The ePS6000 also supports FAST, SLOW, and IDLE modes, as well as SLEEP mode to enhance its low power consumption features.

1.1 Applications

- Scientific calculating machine

2 Features

2.1 MCU Features

- 8 bit RISC MCU
- Operating voltage and speed: 1.2V~3.6V
- Clock Source: Dual clock system
  - Low-frequency: 32KHz Internal RC oscillator / External RC oscillator / Crystal oscillator.
  - High-frequency 200KHz / 300KHz / 500KHz External RC oscillator.
- One Instruction cycle time = 2 * System clock time
- Program ROM addressing: Maximum 16K words
- 128 bytes un-banked RAM including special registers and common registers
- 16 *128 bytes banked RAM
- RAM stack can achieve maximum of 32 stack levels

- TABLE LOOK UP 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 (max. 256 repeat times)
- Decimal ADD & SUB instruction
- Full range CALL and JUMP ability (2 cycles)

2.2 Peripheral

- 24 general I/O pins (PORTA, PORTB, PORTC)
- 11/10/5 COM * 60 SEG LCD driver (embedded)
- One (Timer 0)16-bit timer
- One (Timer1) 8-bit timer with wake-up function
- One (Timer2) 8-bit timer
- One 8-bit watch dog timer
- Key I/O function with maximum 64 keys

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
- Support RC oscillation and crystal oscillation for clock system
- MCU Wake-up function consisting of input wake-up and Timer1 wake-up
- MCU interrupt function consisting of Input port interrupt and Timer interrupt (Timer0 ~ 2).
- MCU reset function includes power on reset, RSTB pin reset, and Watchdog timer reset.

3 Block Diagram

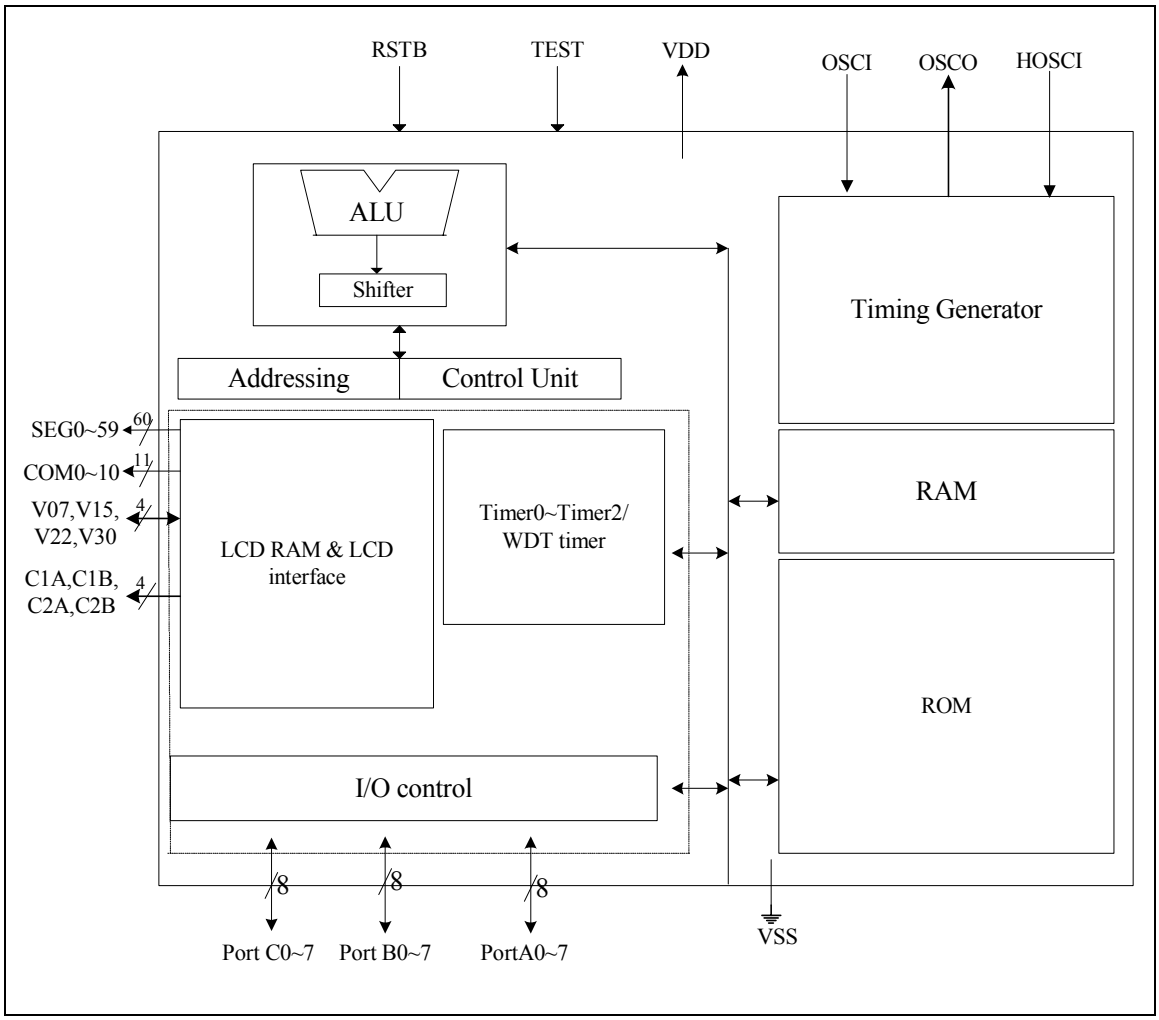


Figure 3-1 ePS6000 Block Diagram

4 Package Pinout and Pin Assignment

4.1 ePS6000 Pinout

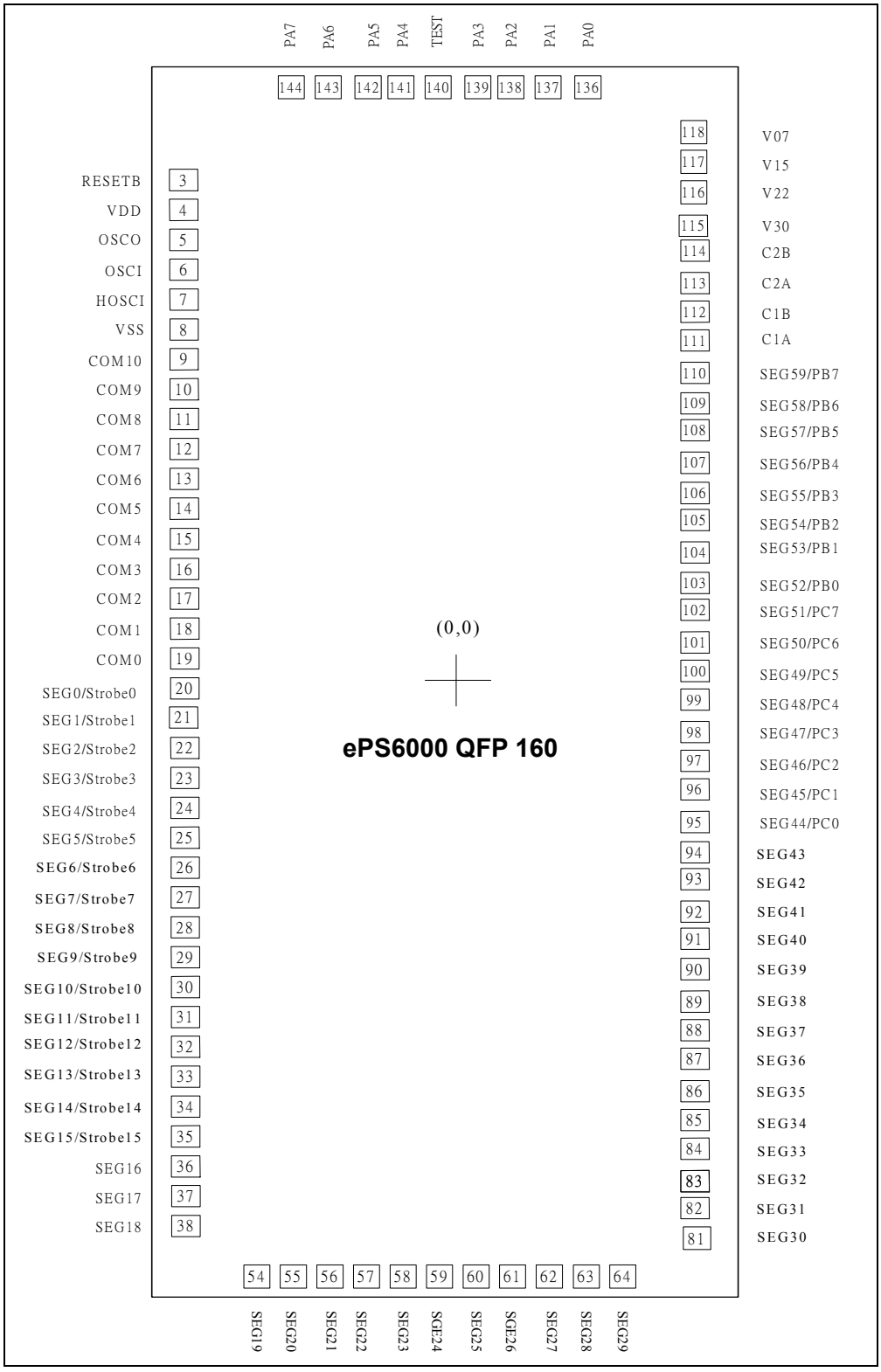
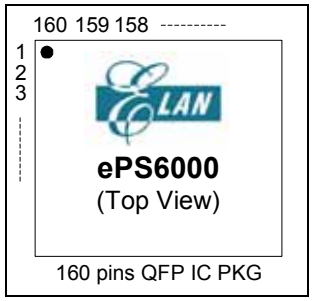


Figure 4-1 ePS6000 QFP Pinout

4.2 Pin Assignment



No.	Pin Name	No.	Pin Name	No.	Pin Name	No.	Pin Name
1	NC	41	NC	81	SEG30	121	NC
2	NC	42	NC	82	SEG31	122	NC
3	RESETB	43	NC	83	SEG32	123	NC
4	VDD	44	NC	84	SEG33	124	NC
5	OSCO	45	NC	85	SEG34	125	NC
6	OSCI	46	NC	86	SEG35	126	NC
7	HOSCI	47	NC	87	SEG36	127	NC
8	VSS	48	NC	88	SEG37	128	NC
9	COM10	49	NC	89	SEG38	129	NC
10	COM9	50	NC	90	SEG39	130	NC
11	COM8	51	NC	91	SEG40	131	NC
12	COM7	52	NC	92	SEG41	132	NC
13	COM6	53	NC	93	SEG42	133	NC
14	COM5	54	SEG19	94	SEG43	134	NC
15	COM4	55	SEG20	95	SEG44/PortC.0	135	NC
16	COM3	56	SEG21	96	SEG45/PortC.1	136	PortA.0
17	COM2	57	SEG22	97	SEG46/PortC.2	137	PortA.1
18	COM1	58	SEG23	98	SEG47/PortC.3	138	PortA.2
19	COM0	59	SEG24	99	SEG48/PortC.4	139	PortA.3
20	SEG0/Strobe0	60	SEG25	100	SEG49/PortC.5	140	TEST
21	SEG1/Strobe1	61	SEG26	101	SEG50/PortC.6	141	PortA.4
22	SEG2/Strobe2	62	SEG27	102	SEG51/PortC.7	142	PortA.5
23	SEG3/Strobe3	63	SEG28	103	SEG52/PortB.0	143	PortA.6
24	SEG4/Strobe4	64	SEG29	104	SEG53/PortB.1	144	PortA.7
25	SEG5/Strobe5	65	NC	105	SEG54/PortB.2	145	NC
26	SEG6/Strobe6	66	NC	106	SEG55/PortB.3	146	NC
27	SEG7/Strobe7	67	NC	107	SEG56/PortB.4	147	NC
28	SEG8/Strobe8	68	NC	108	SEG57/PortB.5	148	NC
29	SEG9/Strobe9	69	NC	109	SEG58/PortB.6	149	NC
30	SEG10/Strobe10	70	NC	110	SEG59/PortB.7	150	NC
31	SEG11/Strobe11	71	NC	111	C1A	151	NC
32	SEG12/Strobe12	72	NC	112	C1B	152	NC
33	SEG13/Strobe13	73	NC	113	C2A	153	NC
34	SEG14/Strobe14	74	NC	114	C2B	154	NC
35	SEG15/Strobe15	75	NC	115	V30	155	NC
36	SEG16	76	NC	116	V22	156	NC
37	SEG17	77	NC	117	V15	157	NC
38	SEG18	78	NC	118	V07	158	NC
39	NC	79	NC	119	NC	159	NC
40	NC	80	NC	120	NC	160	NC

5 Pin Description

5.1 MCU System Pins (7 Pins)

Name	I/O/P Type	Description	Note
VDD	P	Digital and Analog positive power supply, the range is from 1.2V~1.8V/ 2.4V~3.6V.	
VSS	P	Digital and Analog negative power supply.	
RSTB	I	System reset pin. Low active	Int. pull-up
TEST	I	Test mode select pin (High active). For chip internal test only, Normal connect to VSS.	Int. Pull Down
OSCI	I	External RC oscillator /Crystal oscillator connecting pin	Ext. R to VDD
OSCO	O	Crystal oscillator connecting pin	
HOSCI	I	Hi-Speed RC oscillator connecting pin.	Ext. R to VDD

5.2 Embedded LCD Pins (79 Pins)

Name	I/O/P Type	Description	Note
COM0~COM10	O	LCD common signal output pin	
SEG0~SEG15	O	LCD segment signal output pin shared with key strobe 0~15	
SEG16~ SEG43	O	LCD segment signal output pin	
SEG44~SEG51/ PortC.0~7	I/O	LCD segment signal output pin or I/O pin; define by code option	
SEG52~SEG59/ PortB.0~7	I/O	LCD segment signal output pin or I/O pin; define by code option	
C1A,C1B		LCD voltage charge-pump pin. Connect 0.1 uF between C1A and C1B.	
C2A,C2B		LCD voltage charge-pump pin. Connect 0.1 uF between C2A and C2B.	
V30,V22,V15,V07	O	LCD bias Pin. Connect 0.1 uF to Vss	

5.3 I/O Port (8 Pins)

PORT	Bit	Function	I/O Type	Power Source	Description	Note
Port A	Bit3~0 (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	
		General Output	O	VDD		
	Bit7~4	General Input	I	VDD		Int. Pull up (R2: Large resistor) controllable
		Interrupt and wake-up	I	VDD	Input port interrupt and wake-up pin	
		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
Suggest that user setting “Slow mode” for Initial mode after reset.

- Low Frequency Oscillator:
  - Select “External RC” oscillator or “Crystal” oscillator or “Internal RC” oscillator
- Maximum duty ratio option:
  - Select “1/10” duty or “1/11” duty or “1/5” duty
- Port B.0 control bit (SEG52):
  - Select “LCD segment signal output” or “general I/O function”
- Port B.1 control bit (SEG53):
  - Select “LCD segment signal output” or “general I/O function”
- Port B.2 control bit (SEG54):
  - Select “LCD segment signal output” or “general I/O function”
- Port B.3 control bit (SEG55):
  - Select “LCD segment signal output” or “general I/O function”
- Port B.4 control bit (SEG56):
  - Select “LCD segment signal output” or “general I/O function”
- Port B.5 control bit (SEG57):
  - Select “LCD segment signal output” or “general I/O function”
- Port B.6 control bit (SEG58):
  - Select “LCD segment signal output” or “general I/O function”
- Port B.7 control bit (SEG59):
  - Select “LCD segment signal output” or “general I/O function”
- Port C.0 control bit (SEG44):
  - Select “LCD segment signal output” or “general I/O function”
- Port C.1 control bit (SEG45):
  - Select “LCD segment signal output” or “general I/O function”
- Port C.2 control bit (SEG46):
  - Select “LCD segment signal output” or “general I/O function”
- Port C.3 control bit (SEG47):
  - Select “LCD segment signal output” or “general I/O function”

- Port C.4 control bit (SEG48):
  - Select “LCD segment signal output” or “general I/O function”
- Port C.5 control bit (SEG49):
  - Select “LCD segment signal output” or “general I/O function”
- Port C.6 control bit (SEG50):
  - Select “LCD segment signal output” or “general I/O function”
- Port C.7 control bit (SEG51):
  - Select “LCD segment signal output” or “general I/O function”

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.

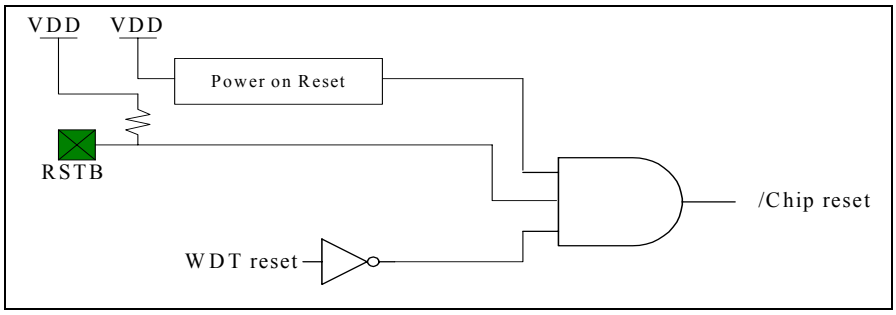


Figure 7-1 On-Chip RESET Schematic

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 decreases to below Vpor, RESET will occur.

7.1.2 RSTB Pin

In normal condition, RSTB pin is pulled up to VDD. Whenever the RSTB is at low condition, RESET will occur.

7.1.3 WDT Time Out

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

7.1.4 32768 Crystal Stable Time

■ **Power On Reset Timing:**

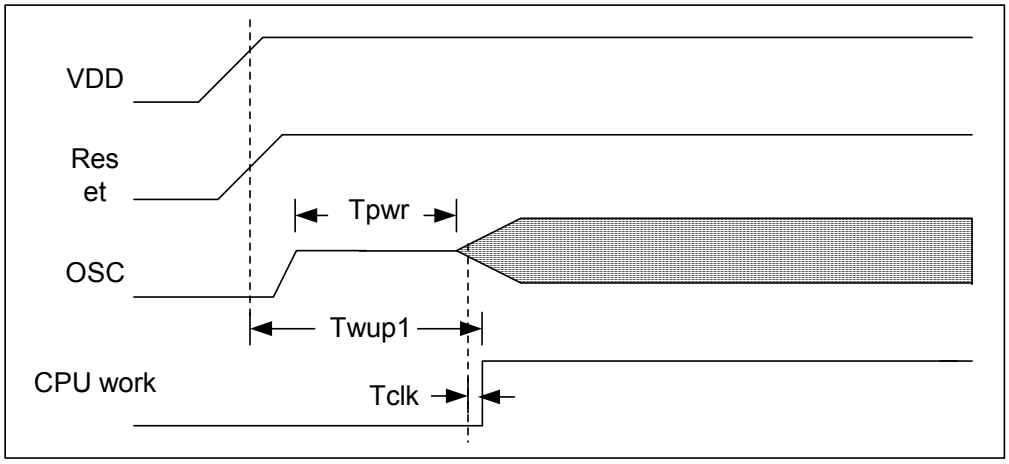


Figure 7-2a Power on Reset Timing Diagram

■ **Sleep Mode Wake-Up Timing:**

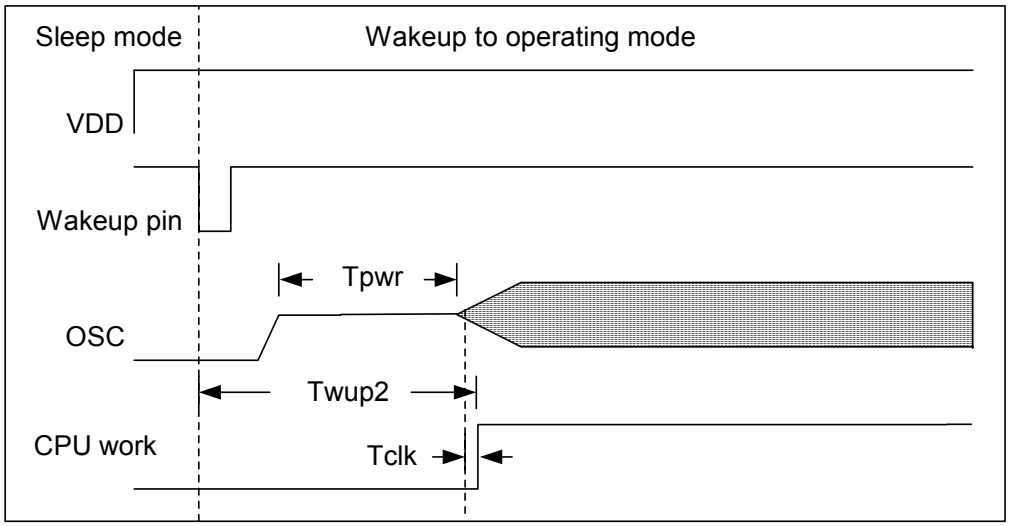


Figure 7-2b Sleep Mode Wake-Up Timing Diagram

Condition: Vdd = 1.5V, Cosc = 20pF & Ta = 25°C

Symbol	Characteristics	Min.	Typ.	Max.	Unit
Tpwr	Oscillator start up time	280	480	780	ms
Twup1	CPU warm up time (Power ON reset)	300	500	800	ms
Twup2	CPU warm up time (Sleep mode wakeup)	285	485	785	ms
Tclk	Detect slow clock time	0.9	1.0	1.1	ms

7.1.5 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

Bit0 (C): Carry flag or inverse of Borrow flag (B)
Under SUB operation, borrow flag is indicated by the inverse of carry bit. (B = /C).

Bit1 (DC): Auxiliary carry flag.

Bit2 (Z): Zero flag

Bit3 (OV): Overflow flag. Use in signed operation when Bit6 is carried into or borrows from signed bit (Bit7).

Bit4 (SLE): Computation result is less than or equal to zero (negative value) after signed arithmetic. Affected by HEX arithmetic instruction only.

Bit5 (SGE): Computation result is greater than or equal to zero (positive value) after signed arithmetic. Affected by HEX arithmetic instruction only.

NOTE

1. *When OV=1 after signed arithmetic, you can check 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.
2. *When overflow took place, you should clear the MSB of Accumulator to obtain the correct value.*
   When underflow took place, you should set the MSB of accumulator to obtain the correct value.

Example 1: ADD positive value with a positive value, and the ACC signed bit will be affected.

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

Unsigned bit results after execution of the instruction**:**

ACC = 0D0h SGE=1, means the result is greater than or equal to '**0**' (positive value)
OV=1, means overflow occurred and the result is carried into signed bit (Bit 7)

Signed bit results after execution of the instruction**:**

ACC = 50h (signed bit is cleared)

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

Example 2: SUB positive value from 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)
```

Unsigned bit results after execution of the instruction**:**

ACC = 40h SLE=1, means the result is less than or equal to '**0**' (negative value)

OV=1, Underflow occurred and the result is borrowed from signed bit (Bit 7)

Signed bit results after execution of the instruction**:**
ACC = 0C0h (the signed bit is set)

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

Bit6 (/PD): Reset to "**0**" when /PD enters SLEEP mode. Set to "**1**" by "WDTC" instruction, power-on reset, or by Reset pin low condition.

Bit7 (/TO): Reset to "**0**" at WDT time out reset. Set to "**1**" by "WDTC" instruction; power-on reset, Reset pin low condition, or when MCU enters into SLEEP MODE.

When reset occurs, special function register is reset to initial value except for the **/TO** and **/PD** bits of 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 from SLEEP mode)
1	0	Reserved.
1	1	Power up or RSTB pin low condition

7.1.6 Initialization after RESET Occurs

- 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 cleared to all "**0**."
- The other register initial values are as indicated in the following sections**:**

■ Special Registers

Addr.	Name	Initial Value	Addr.	Name	Initial Value
00h	INDF0	---- ----[1]	10h	PORTA	xxxx xxxx
01h	FSR0	0000 0000	11h	PORTB	xxxx xxxx
02h	BSR	0000 0000	12h	PORTC	xxxx xxxx
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

■ Control Register

Addr.	Name	Initial Value	Addr.	Name	Initial Value
20h	STBCON	0000 0000	2Bh	PAINTEN	0000 0000
21h	INTCON	---- -000	2Ch	PAINTSTA	0000 0000
22h	INTSTA	---- -000	2Dh	DCRA	1111 1111
23h	TR01CON	0000 0000	2Eh	PBCON	0000 0000
24h	TRL0L	uuuu uuuu	2Fh	DCRB	1111 1111
25h	TRL0H	uuuu uuuu	30h	PCCON	0000 0000
26h	TRL1	uuuu uuuu	31h	DCRC	1111 1111
27h	TR2WCON	0–00 0000	32h	LCDCON	–00– 00–0
28h	TRL2	uuuu uuuu	33h	POST_ID	–111 –000
29h	PACON	0000 0000	34h	CPUCON	---- –00c [3]
2Ah	PAWAKE	0000 0000			

Legend: x: unknown **–:** unimplemented read as "0"
u: unchanged, **c:** value depends on actual condition

[1] *Not a physical register*

[2] *If it is a power-on reset or the 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.*

[3] *Bit0 (MS0) of RE (CPUCON) is reloaded from "**INIM**" bit of code option when MCU resets.*

7.2 Oscillator System

The oscillator system is used to generate the device clock. The oscillator system may use an Internal RC, external RC, or a crystal oscillator for SLOW mode and use an external RC oscillator for FAST mode as illustrated in the diagram below.

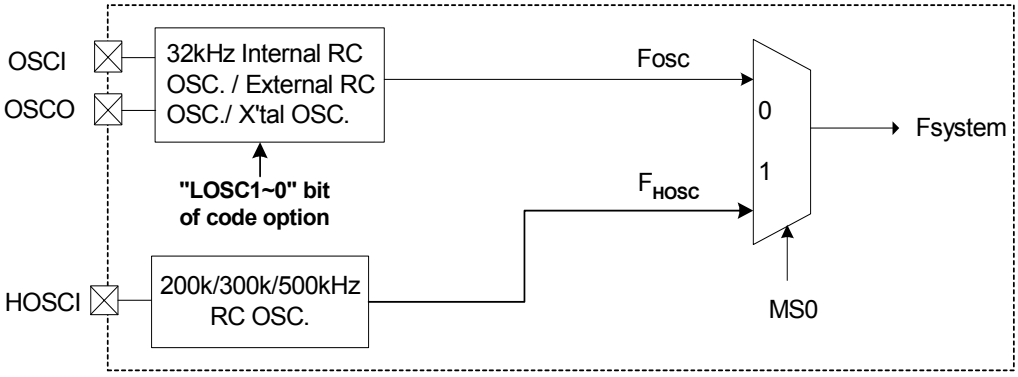


Figure 7-3 Oscillator System Function Block Diagram

The **MS0** bit (mode select bit) of **CPUCON** register (R34h) 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 FHOSC).

7.2.1 32.8kHz RC or 32768Hz Crystal Oscillator

- 32.8kHz RC Internal oscillator**:**
  Select "RC oscillator for FOSC" in the code option and allow OSCI and OSCO pins to stay floating.
- 32.8kHz RC external oscillator**:**
  Select "RC oscillator for FOSC" in the code option and connect a 2.2MΩ resistor between OSCI and Vdd pin while OSCO pin stays floating.
- 32768kHz Crystal oscillator:
  Select "Crystal oscillator for FOSC" in the code option and connect a crystal between OSCI and OSCO pins. The OSCI and OSCO pins are also connected to ground through a 20pF capacitor respectively.

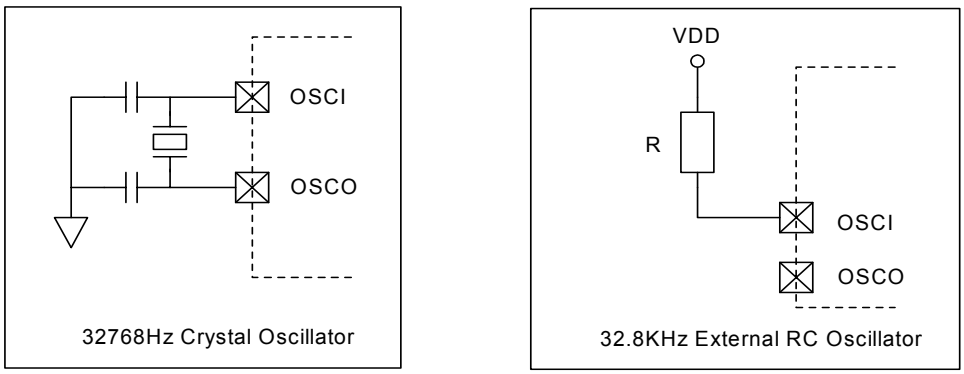


Figure 7-4 Slow Mode Crystal and External RC Oscillators Circuit Diagram

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

A resistor should be connected between HOSCI and Vdd pin.

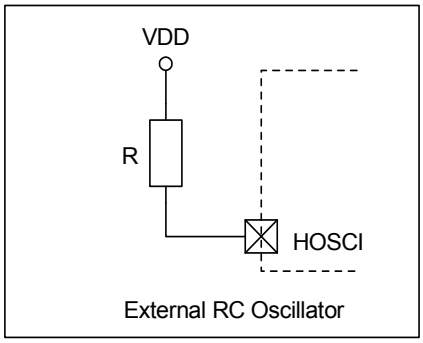


Figure 7-5 Fast Mode RC Oscillators Circuit Diagram

7.3 MCU Operation Mode:

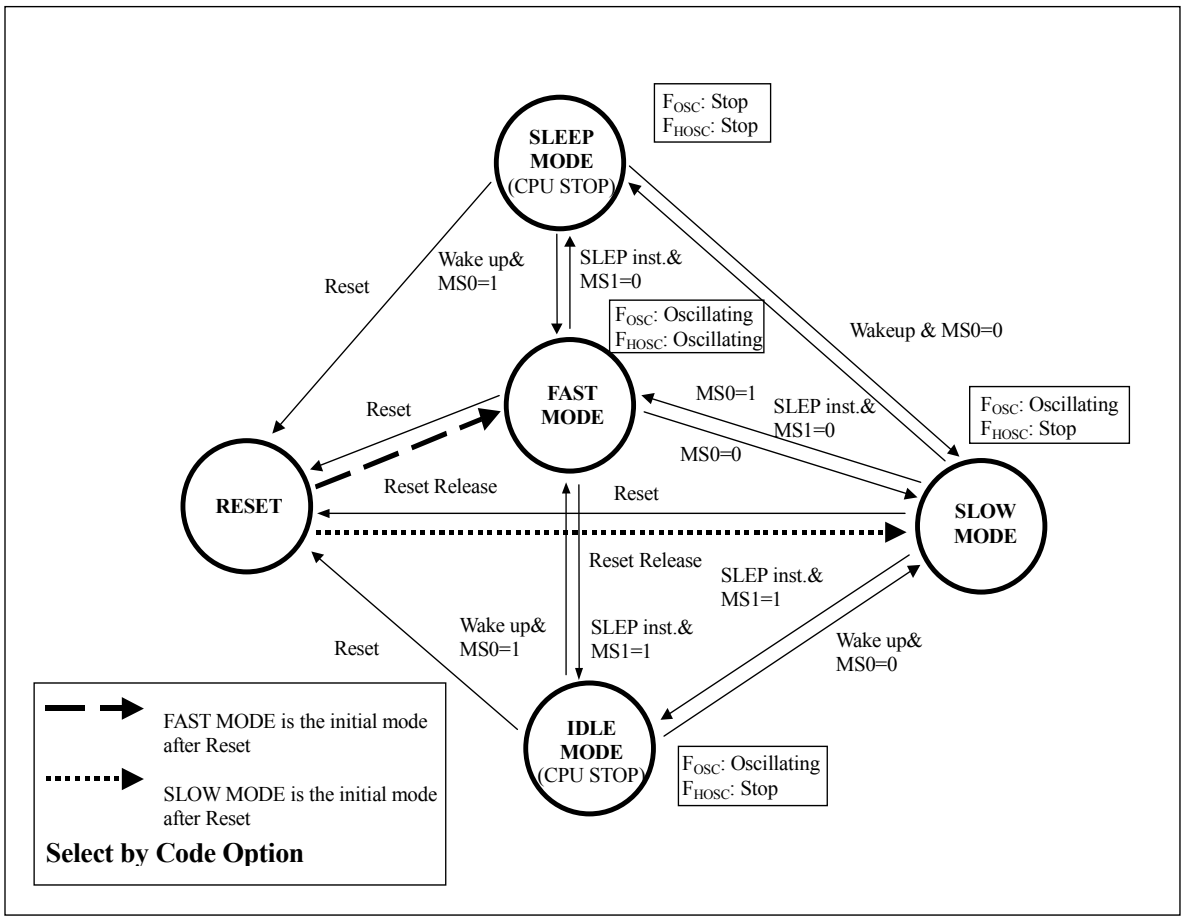


Figure 7-6 ePS6000 MCU Operation Block Diagram

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

Device \ Mode	SLEEP	IDLE	SLOW	FAST
Osc.(32768Hz)	X	O	O	O
Fsystem	X	X	From Osc.	From Hosc.
Timer0~2	X	X	O	O
INT	X*	X*	O	O
I/O wake-up	O	O	X	X
Timer1 wake-up	X	O	X	X

Legend: O = Function is available if enabled **X:** Function NOT supported

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

7.3.1 Slow, Fast, Sleep, and Idle Mode of Operations

■ **CPUCON (R34h): 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 MS0 bit of 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 to follow "BC CPUCON,MS0" instruction when 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 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 fast speed performance of all the system operations, but under highest power consumption.

■ IDLE MODE:

When MS1 bit of 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 awaken by the Timer 1 wake-up or by I/O pins 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 MS1 bit of 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 awaken by I/O pins wake-up.

NOTE

- *The /PD bit of STATUS Register (RFh) is cleared when 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:

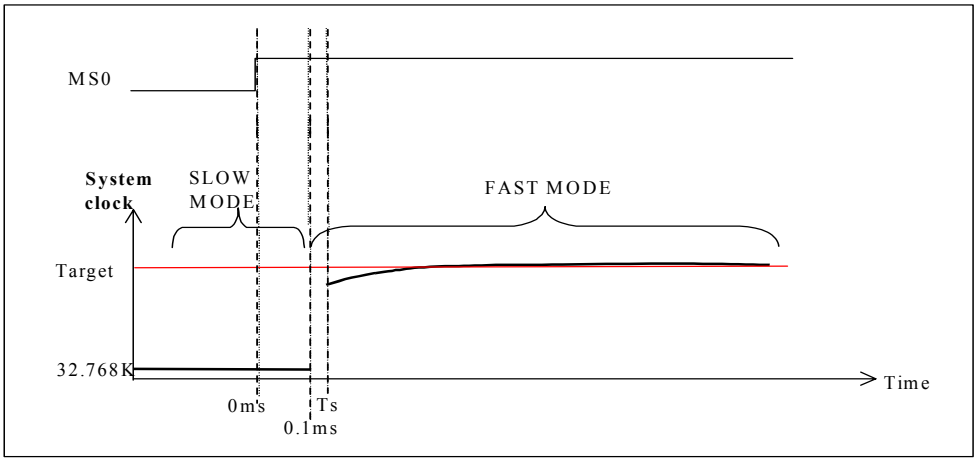


Figure 7-7 SLOW MODE to FAST MODE Timing Diagram

NOTE: *1. SLOW MODE switches to FAST MODE at Time=0ms.*

2. System clock will switch to FAST MODE after 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 stabilize 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 awaken by input port (Port A), then returns to FAST MODE or SLOW MODE (as determined by MS0 bit of CPUCON register described in previous section).

When in IDLE MODE, the 32khz oscillator keeps on running. The MCU is awaken by input port (Port A) or Timer1, then returns to FAST MODE or SLOW MODE (as determined by MS0 bit of CPUCON register described in the 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	WKEN6	WKEN5	WKEN4	WKEN3	WKEN2	WKEN1	WKEN0

Bit 7 (WKEN7) ~ Bit 0 (WKEN0): Wake-up function control bit of PortA.7 ~ PortA.0

"0": Disable PortA.7 ~ PortA.0 wake-up function

"1": Enable PortA.7 ~ PortA.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): Timer1 underflow wake-up function control bit under IDLE MODE

"0": Disable Timer1 wake-up function

"1": Enable Timer1 wake-up function.

7.4 Interrupts

When interrupt occurs, the GLINT bit of 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 (R34h) 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	PortA.7 ~ 0	0x00002	PAINT
Level 2	reserved	0x00004	Reserved
Level 3	reserved	0x00006	Reserved
Level 4	Timer0~2	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:
     MOV    AccBuf,A
     MOVPR  StatusBuf,Status
     RET
```

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

7.4.2 Input Port (PortA.7 ~ PortA.0) Interrupt

PortA.0 ~ PortA.7 are used as external interrupt/wake-up input. If PA7IE ~ PA0IE bits of PAINTEN register are set to "**1**," PortA.0 ~ PortA.7 are the external interrupt input port format.

■ **PAINTSTA (R2Ch): PortA.7 ~ PortA.0 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 (PA7I) ~ Bit 0 (PA0I): PortA.7 ~ PortA.0 Interrupt status

Set to "**1**" when pin falling edge is detected
Clear ("**0**") by software

■ **PAINTEN (R2Bh): PortA.7 ~ PortA.0 Interrupt 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 (PA7IE) ~ Bit 0 (PA0IE): PortA.7 ~ PortA.0 Interrupt control bits

"0": Disable interrupt function
"1": Enable interrupt function

■ **Code Example:**

```
;  ===  Input PortA Interrupt
PAINT:
     S0CALL PUSH
     CLR PAINTSTA
       :
     SJMP   POP
     RETI
```

7.4.3 Timer0 Interrupt

Timer 0 is a 16-bit timer used for general time counting. When the counting value underflows, the Timer0 interrupt takes place and the TRL0H**:** TRL0L value is automatically reloaded into the timer.

■ **TMR0IE Bit of INTCON (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 Timer0 interrupt

"0": Disable Timer0 interrupt function
"1": Enable Timer0 interrupt function

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

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

Bit0 (TMR0I): Status bit of Timer0 interrupt

Set to "**1**" when Timer0 counter underflows
Clear ("**0**") by software

7.4.4 Timer1 Interrupt

Timer1 is an 8-bit timer used for time counting and wake-up functions. When the counting value of Timer1 underflows, interrupt occurs and the TRL1 value is reloaded to the timer.

■ **TMR1IE Bit of INTCON (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 Timer1 interrupt function
"1": Enable Timer1 interrupt function

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

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

Bit1 (TMR1I): Status bit of Timer1 interrupt

Set to "**1**" when Timer1 counter underflows
Clear ("**0**") by software

7.4.5 Timer2 Interrupt

Timer2 is an 8-bit timer for time counting. When the counting value of Timer2 underflows, the interrupt occurs and the TRL2 value is reloaded into the timer.

■ **TMR2IE Bit of INTCON (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 Timer2 interrupt

"0": Disable Timer2 interrupt function
"1": Enable Timer2 interrupt function

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

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

Bit2 (TMR2I): Status bit of Time2 interrupt

Set to "**1**" when Timer2 counter underflows

Clear ("**0**") by software

■ **Code Example:**

```
;  ===  Timer-0,1,2 Interrupt
TIMERINT:
     S0CALL PUSH
     JBS    INTSTA,TMR0I,toTM0INT
     JBS    INTSTA,TMR1I,toTM1INT
     JBS    INTSTA,TMR2I,toTM2INT
     SJMP   POP

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

7.5 Program ROM Map

ROM Size = 16K Words.	
Address.	**Description**
0000h ⋮ 000Bh	Interrupt Vector (12 words)
000Ch ⋮ 000Fh	Code Option (4 words)
0010h ⋮ 001Fh	Test Program (16 words)
0020h ⋮ 3FFFh	Program or Fixed data region

7.6 RAM Map for Special and Control Registers
(RAM Size: 88 Bytes + 16 Banks * 128 Bytes = 2136 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 Pointer0.							
1	FSR0	R/W							
		File select register 0 for INDF0 (R0)							
2	BSR	R	R	R	R	R/W	R/W	R/W	R/W
		Fixed 0	Fixed 0	Fixed 0	Fixed 0	Bank select register (for INDF0 & general)			
3	INDF1	R/W							
		Indirect addressing Pointer1.							
4	FSR1	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		Fixed 1	File select register 1 for INDF1 (R3)						
5	BSR1	R	R	R	R	R/W	R/W	R/W	R/W
		Fixed 0	Fixed 0	Fixed 0	Fixed 0	Bank select Register 1 (for INDF1)			
6	STKPTR	R/W							
		Stack pointer							
7	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
8	PCM	R	R	R/W	R/W	R/W	R/W	R/W	R/W
		Fixed 0	Fixed 0	PC13	PC12	PC11	PC10	PC9	PC8
9	LCDARL	R/W							
		LCD Ram column Address							
A	ACC	R/W							
		Accumulator							
B	TABPTRL	R/W							
		Low byte of table pointer							
C	TABPTRM	R	R/W						
		Fixed 0	Middle byte of table pointer						
D	TABPTRH	R/W							
E	LCDDATA	R/W							
		Indirect register to LCD Ram							
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	PORTA	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		Port A.7	Port A.6	Port A.5	Port A.4	Port A.3	Port A.2	Port A.1	Port A.0
11	PORTB	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		Port B.7	Port B.6	Port B.5	Port B.4	Port B.3	Port B.2	Port B.1	Port B.0
12	PORTC	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		Port C.7	Port C.6	Port C.5	Port C.4	Port C.3	Port C.2	Port C.1	Port C.0

(Continuation)

Addr.	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
20	STBCON	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		SCAN	KE	R1EN	BitST	STB3	STB2	STB1	STB0
21	INTCON	-	-	-	-	-	R/W	R/W	R/W
		-	-	-	-	-	TMR2IE	TMR1IE	TMR0IE
22	INTSTA	-	-	-	-	-	R/W	R/W	R/W
		-	-	-	-	-	TMR2I	TMR1I	TMR0I
23	TR01CON	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		T1WKEN	T1EN	T1PSR1	T1PSR0	T0EN	T0CS	T0PSR1	T0PSR0
24	TRL0L	R/W							
		Timer0 auto-reload register low byte							
25	TRL0H	R/W							
		Timer0 auto-reload register high byte							
26	TRL1	R/W							
		Timer1 auto-reload register							
27	TR2WCON	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
		WDTEN	-	WDTPSR 1	WDTPSR 0	T2EN	T2CS	T2PSR1	T2PSR0
28	TRL2	R/W							
		Timer2 auto-reload register							
29	PACON	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		PA7PU	PA6PU	PA5PU	PA4PU	PA3PU	PA2PU	PA1PU	PA0PU
2A	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
2B	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
2C	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
2D	DCRA	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		PA7DC	PA6DC	PA5DC	PA4DC	PA3DC	PA2DC	PA1DC	PA0DC
2E	PBCON	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		PB7PU	PB6PU	PB5PU	PB4PU	PB3PU	PB2PU	PB1PU	PB0PU
2F	DCRB	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		PB7DC	PB6DC	PB5DC	PB4DC	PB3DC	PB2DC	PB1DC	PB0DC
30	PCCON	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		PC7PU	PC6PU	PC5PU	PC4PU	PC3PU	PC2PU	PC1PU	PC0PU
31	DCRC	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		PC7DC	PC6DC	PC5DC	PC4DC	PC3DC	PC2DC	PC1DC	PC0DC
32	LCDCON	-	R/W	R/W	-	R/W	R/W	-	R/W
		-	BLANK	LCDON	-	LCR1	LCR0	-	LBVON
33	POST_ID	-	R/W	R/W	R/W	-	R/W	R/W	R/W
		-	LCD_ID	FSR1_ID	FSR0_ID	-	LCD_PE	FSR1_PE	FSR0_PE
34	CPUCON	-	-	-	-	-	R/W	R/W	R/W
		-	-	-	-	-	GLINT	MS1	MS0

7.6.2 Other Unbanked General RAM

Address	Unbanked
13h ⋮ 1Fh	General purpose RAM
35h ⋮ 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

■ **COM0 (Bit0) ~ COM7 (Bit7)**

RAM Address LCDARL		COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7
		Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
SEG0	**00H**								
⋮	⋮								
SEG59	**3BH**								

■ **COM8 (Bit0) ~ COM10 (Bit2)**

RAM Address LCDARL		COM8	COM9	COM10
		Bit0	Bit1	Bit2
SEG0	**40H**			
⋮	⋮			
SEG59	**7BH**			

7.8 Special Register Descriptions

7.8.1 ACC (R0Ah): Accumulator Register

Internal data transfer or instruction operand holding

7.8.2 POST_ID (R33h): 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 (Bank0 ~ Bank15).

FSR0 (R01h) is an address register for INDF0. You can select up to 256 bytes (Address: 00 ~ 0FFh).

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 general register.

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

INDF1 (R03h) is not a physically implemented register.

■ **Code Example:**

```
Data transform bank0 to bank1:
     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 Bank1
       :
```

■ INDF1 Linear Address Capabilities

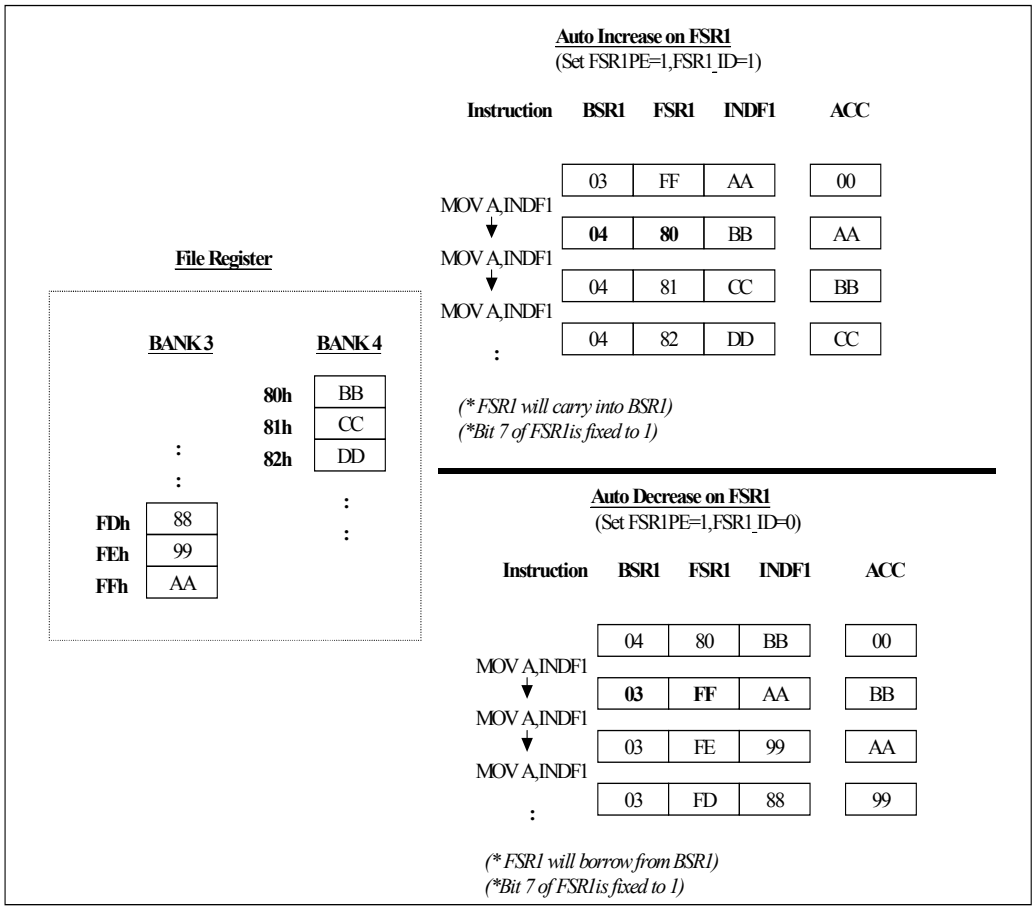


Figure 7-8 INDF1 Linear Address Capabilities Diagram

■ 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
     MOVRP  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 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. You cannot see the stack RAM.*

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							

Generates up to 16K*16 on chip ROM addresses for 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 14 bits of the PC (16K*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
0	TABPTRM							TABPTRL							

Program ROM or Internal ROM address register.

Bit 14 ~ Bit 1 are used to point the address of memory.

Bit 0 is used to select the low or high byte of the pointed word (see TBRD instruction in Section 12, *Instruction Set*)

■ **Code Example:**

```
;  ***  Program ROM
       :
       :
     TBPTM  #(PROMTabB*2)/100H
     TBPTL  #PROMTabB*2
       :
       :
     TBRD   0,ACC                    ; not 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 PortA, PortB, PortC (R10h, R11h, R12h): General I/O Pins Registers

PortA (R10h) PortA.0 ~ 7 are general I/O pins registers

PortB (R11h) PortB.0 ~ 7 are general I/O pins registers

PortC (R12h) PortC.0 ~ 7 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	KE	R1EN	BitST	STB3	STB2	STB1	STB0

Bit 7 (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 6 (KE): Key input enable/disable control bit

"0": Disable Key input function (PORTA.0~3 do NOT correspond with Key input in software scan mode)

"1": Enable Key input function (PORTA.0~3 correspond with Key input in software scan mode)

Bit 5 (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 4 (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 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	PA6PU	PA5PU	PA4PU	PA3PU	PA2PU	PA1PU	PA0PU

Bit 7 ~ 0 (PA7PU ~ PA0PU): Enable PortA.0 ~ PortA.7 pull up resistor bits

"0": Disable PortA.0 ~ PortA.7 pull up resistor

"1": Enable PortA.0 ~ PortA.7 pull up resistor

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	WKEN6	WKEN5	WKEN4	WKEN3	WKEN2	WKEN1	WKEN0

Bit 7 ~ 0 (WKEN7 ~ WKEN0): Wake-up enable control bits of PortA.7~PortA.0.

"0": Disable PortA.7 ~ PortA.0 wake-up function

"1": Enable PortA.7 ~ PortA.0 wake-up function.

NOTE

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

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	PA6IE	PA5IE	PA4IE	PA3IE	PA2IE	PA1IE	PA0IE

Bit 7 ~ Bit 0 (PA7IE ~ PA0IE): Interrupt Control bits

0: Disable Port A interrupt function

1: Enable Port A interrupt function

NOTE

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

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	PA6I	PA5I	PA4I	PA3I	PA2I	PA1I	PA0I

Bit 7 ~ Bit 0 (PA7I ~ PA0I): INT status of Port A.7 ~ PortA.0 interrupts bits

Set to "**1**" when pin falling edge detected
Clear ("**0**") by software

7.8.14 DCRA (R2Dh): Port A Direction Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PA7DC	PA6DC	PA5DC	PA4DC	PA3DC	PA2DC	PA1DC	PA0DC

Bit 7 ~ Bit 0 (PA7DC ~ PA0DC): PortA.0~PortA.7 direction control bits

"0": Set to output pin
"1": Set to input pin

7.8.15 PBCON (R2Eh): Port B Pull up Resistor Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PB7PU	PB6PU	PB5PU	PB4PU	PB3PU	PB2PU	PB1PU	PB0PU

Bit 7 ~ Bit 0 (PB7PU ~ PB0PU): Port B.0 ~ PortB.7 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.

7.8.16 DCRB (R2Fh): Port B Direction Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PB7DC	PB6DC	PB5DC	PB4DC	PB3DC	PB2DC	PB1DC	PB0DC

Bit 7 ~ Bit 0 (PB7DC ~ PB0DC): Port B.0 ~PortB.7 direction control bits

"0": Set to output pin
"1": Set to input pin

NOTE
When a PortB bit is set to input pin, a 5 µsec delay in reading the PortB data must be provided. Otherwise, read data will be inaccurate. See Example below.

■ **Code Example:**

```
; *** Set PortB to input pins
     MOV    A,#0XFF
     MOV    DCRB,A
     MOV    PBCON,A
Read_PB:
     JBS    PORTB,0,Read_PB
       Delay 5usec
     JBS    PORTB,0,Read_PB
     SJMP   Read_PB
```

7.8.17 PCCON (R30h): Port C Pull up Resistor Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PC7PU	PC6PU	PC5PU	PC4PU	PC3PU	PC2PU	PC1PU	PC0PU

Bit 7 ~ Bit 0 (PC7PU ~ PC0PU): PortC.0 ~ PortC.7 pull up resistor control bits

"0": Disable pull up resistor

"1": Enable pull up resistor

NOTE

This function is only available with the Port C selected as input pin.

7.8.18 DCRC (R31h): Port C Direction Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PC7DC	PC6DC	PC5DC	PC4DC	PC3DC	PC2DC	PC1DC	PC0DC

Bit 7 ~ Bit 0 (PC7DC ~ PC0DC): Port C.0 ~ PortC.7 direction control bits

"0": Set to output pin

"1": Set to input pin

NOTE

When a PortC bit is set to input pin, a 5 µsec delay in reading the PortC data must be provided. Otherwise, read data will be inaccurate. See example below.

■ **Code Example:**

```
; *** Set PortC to input pins
     MOV    A,#0XFF
     MOV    DCRC,A
     MOV    PCCON,A
Read_PC:
     JBS    PORTC,0,Read_PC
       Delay 5usec
     JBS    PORTC,0,Read_PC
     SJMP   Read_PC
```

8 Peripheral

8.1 Timer 0 (16 Bits Timer)

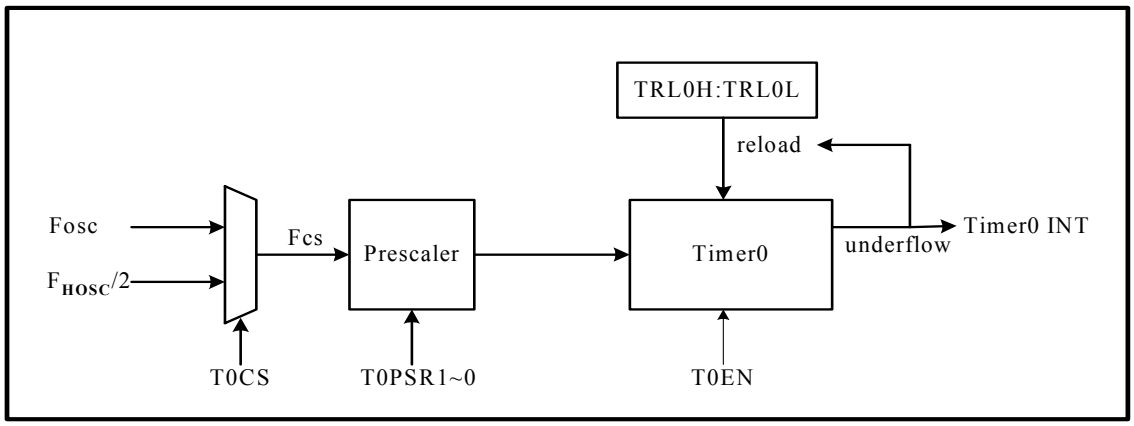


Figure 8-1 Timer 0 Function Block Diagram

Timer 0 is a general-purpose 16 bits down counter used on applications that require time counting with interrupt. The clock source (FCS) can be selected from the oscillator clock (FOCS) or half of the system clock (FHOSC/2).

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

The counter value decrements 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 reload into 16 bits counter.

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

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 registers will automatically reload into 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 (T0PSR1 ~ T0PSR0): Timer 0 prescaler select bits

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

■ **CPUCON (R34h): 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

■ **INTCON (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): Timer0 interrupt control bit
"0": Disable interrupt function
"1": Enable interrupt function

■ **INTSTA (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
Cleared ("**0**") by software

■ **Code Example:**

```
;  ===  Timer 0 interrupt
TIMERINT:
     PUSH
     JBC  INTSTA,TMR0I,Q_Time
     BC   INTSTA,TMR0I
     BTG  PORTA,7
Q_Time:
     POP
     RETI
;  ===  Timer0 = [1/(300K/2)] * [1 x(1FFFh + 1)]
Timer0SR:
       :
     System setting 300KHz
     PA.7 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       ; Timer0 enable.
     BS     INTCON,TMR0IE      ; Timer0 interrupt enable.
     BC     INTSTA,TMR0I       ; Clear timer0 interrupt status.
     BS     CPUCON,GLINT       ; Enable global interrupt.
TimeLoop:
     SJMP   TimeLoop
```

8.2 Timer 1 (8 Bits)

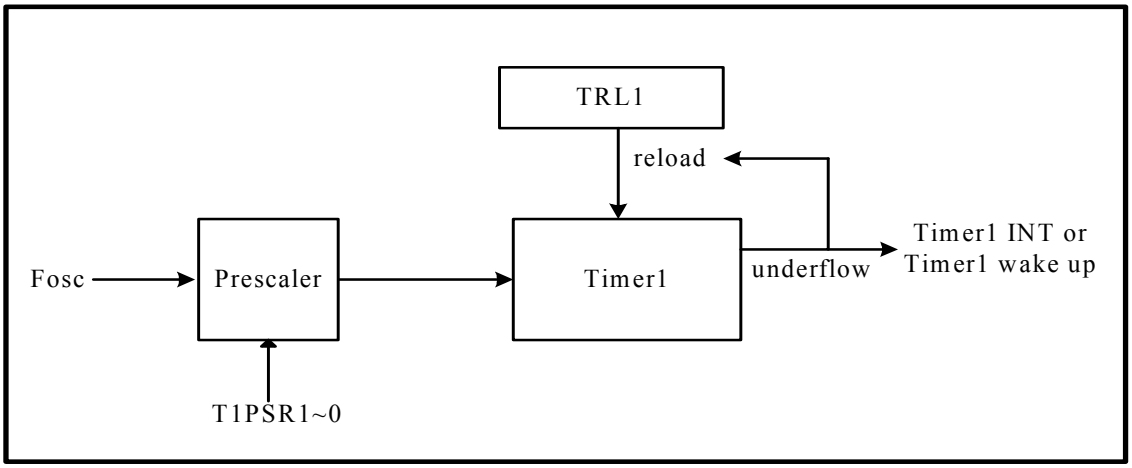


Figure 8-2 Timer 1 Function Block Diagram

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 (Focs).

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.

Counter value decrements 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 will automatically reload into 8 bits counter.

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

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 will automatically reload into 8 bits counter.

■ **CPUCON (R34h): 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

■ **INTCON (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

■ **INTSTA (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
Cleared ("**0**") by software

■ **Code Example:**

```
;  ===  Timer 1 interrupt
TIMERINT:
     PUSH
     JBC     INTSTA,TMR1I,Q_Time
     BC      INTSTA,TMR1I
     BTG     PORTA,7
Q_Time:
     POP
     RETI
;  ===  Timer1 = 32.768K / [256 x (3Fh + 1)]
Timer1SR:
       :
     PA.7 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       ; Timer1 enable.
     BS      INTCON,TMR1IE      ; Timer1 interrupt enable.
     BC      INTSTA,TMR1I       ; Clear timer1 interrupt status.
     BS      CPUCON,GLINT       ; Enable global interrupt.
     BS      CPUCON,MS1         ; Idle mode.
T1Wloop:
     SLEP
     NOP
       :
     SJMP    T1Wloop
```

8.3 Timer 2 (8 Bits)

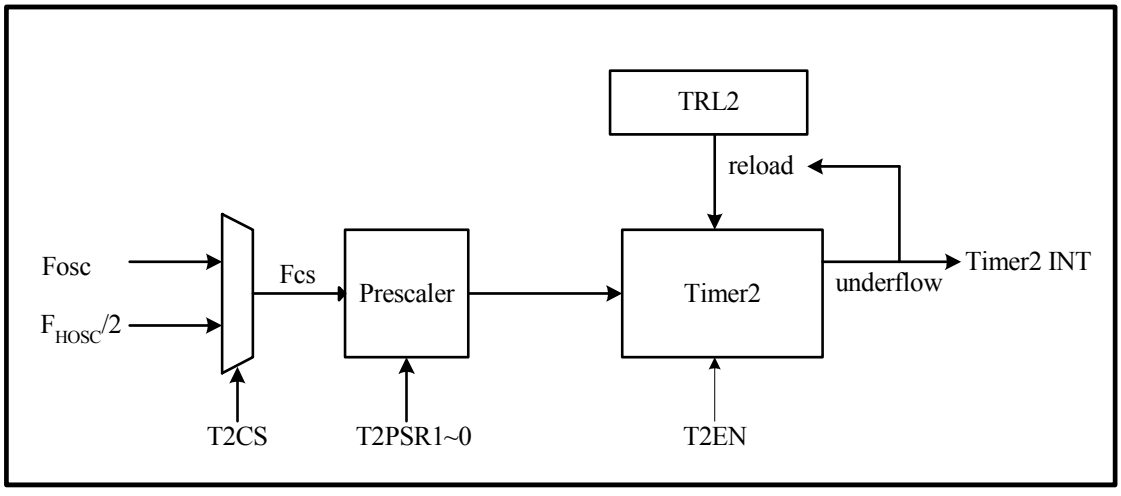


Figure 8-3 Timer 2 Function Block Diagram

Timer 2 is a general-purpose 8 bits down counter used on applications that require a time counter with interrupt. The clock source (Fcs) may be selected from the oscillator clock (FOSC) or half of the system clock (FHOSC/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 decrements by one (count down) according to the timer clock source frequency. When counter value underflows, the timer interrupt is triggered (if Timer 2 interrupt is enabled).

$$T = \frac{1}{Fcs} \times Prescaler \times (TRL2 + 1)$$

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 will automatically reload into 8 bits 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): Timer2 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 (R34h): 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

■ **INTCON (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 Timer2 interrupt
"0": Disable interrupt function
"1": Enable interrupt function

■ **INTSTA (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 Timer2 interrupt occurs, this bit will be set
Clear ("**0**") by software

■ **Code Example:**

```
;  ===  Timer 2 interrupt
TIMERINT:
     PUSH
     JBC     INTSTA,TMR2I,Q_Time
     BC      INTSTA,TMR2I
     BTG     PORTA,7
Q_Time:
     POP
     RETI
;  ===  Timer2 = (1/32.768K)x[4x(FFh+1)]
Timer2SR:
       :
     PA.7 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       ; Timer2 Enable
     BS      INTCON,TMR2IE     ; Timer2 interrupt Enable
     BC      INTSTA,TMR2I      ; Clear Timer2 interrupt Status
     BS      CPUCON,GLINT      ; Global interrupt Enable
TMR2Loop:
     SJMP    TMR2Loop
```

8.4 Watchdog Timer (WDT)

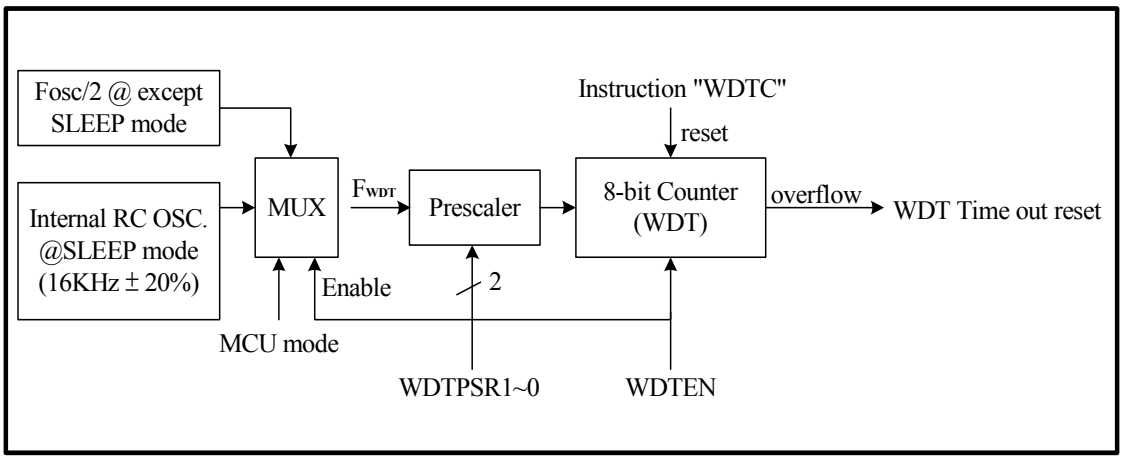


Figure 8-4 Watchdog Timer Function Block Diagram

The watchdog timer (WDT) clock source comes from 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 WDT's enable/disable functions. The initial state of WDT is disabled. When WDT is enabled, its time-out will cause the MCU to reset. You should use "WDTC" instruction 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).

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

8.4.1 Watchdog Timer (WDT) Registers

■ **TR2WCON (R27h): Timer2/Watch Dog 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     INTSTA,TMR1I,Q_Time
     BC      INTSTA,TMR1I
     WDTC
        :
        :
Q_Time:
     POP
     RETI
```

8.5 Input/Output Key

- 4 pins key input (Port A.3 ~ 0) and 16 pins key strobe (shared with LCD segment) can achieve a maximum of 64 keys matrix
- 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).

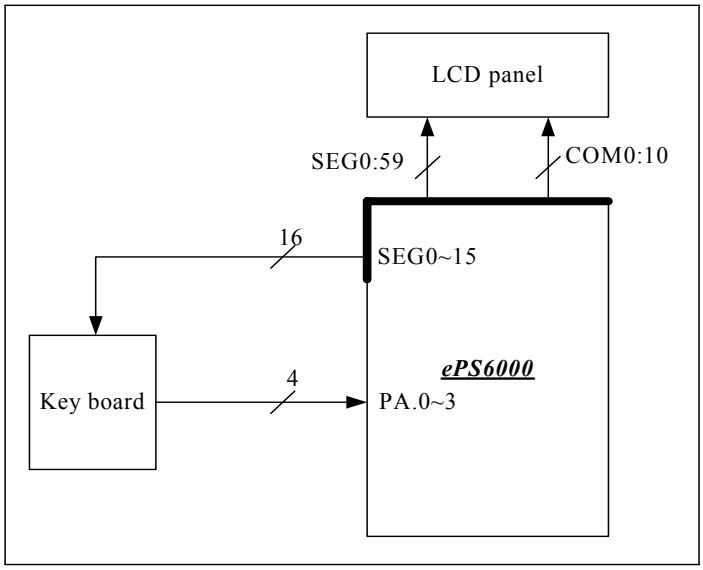


Figure 8-5a Input/Output Key Function Block Diagram

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 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 (RIN >> RON + KON). 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 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

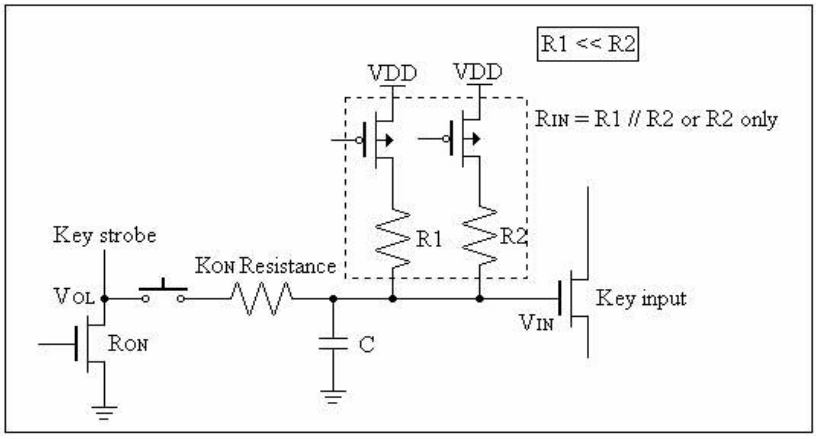


Figure 8-5b Key Circuit Diagram

8.5.1 Key Functions

SCAN	KE	R1EN	PA3PU~PA0PU	IEN[1]	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	1	1	R1 // R2 [2]	PA.0~.3	
1	x	0	0	0	Floating	High-Z	A[3]
		1	1	0	R1//R2 [2]	High-Z	B[3]
		0	1	1	R2	PA.0~ 3	C[3]

x: Don't care.

[1] *Internal signal. Refer to the Automatic Key Scan Timing Diagram (Figure 8-6) below*

[2] *R1 // R2 = R1R2 / (R1+R2).*

[3] *Sub clock signal. Refer to the Automatic Key Scan Timing Diagram (Figure 8-6) below*

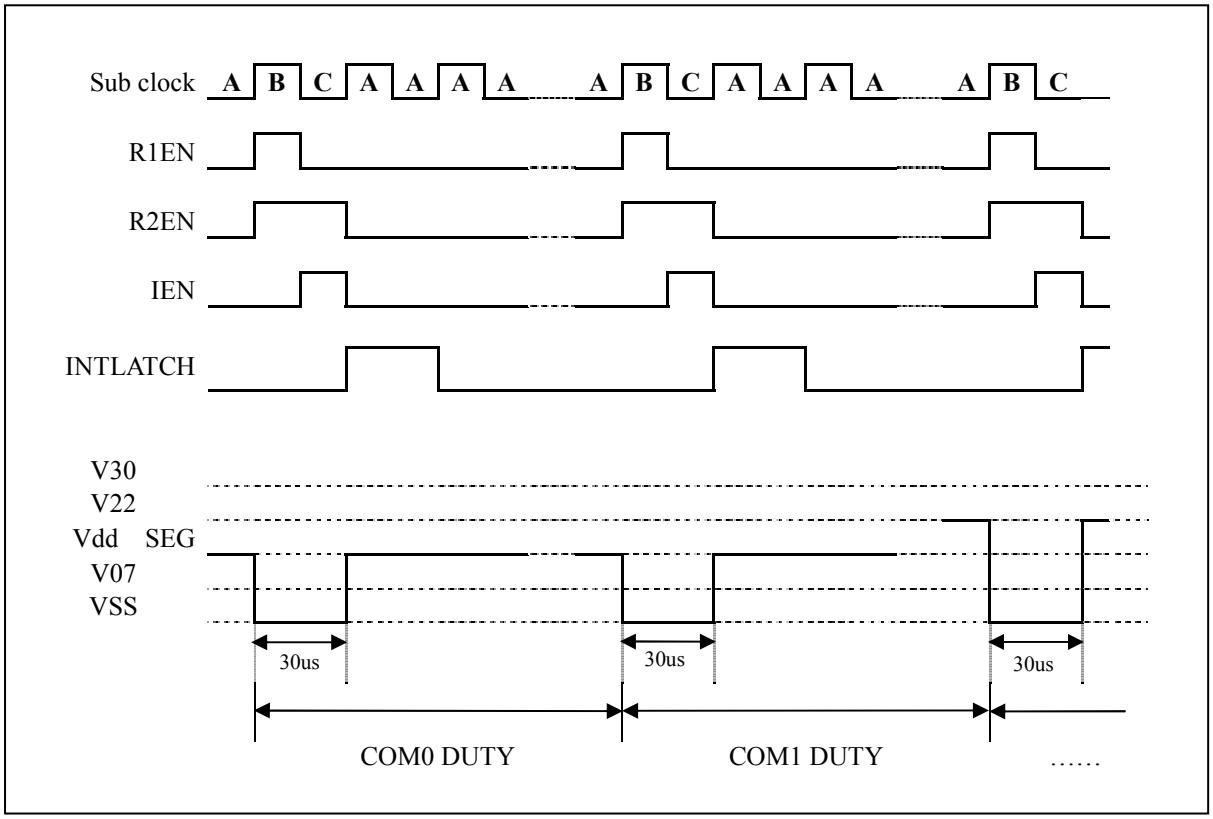


Figure 8-6 Automatic KeyScan Timing (SCAN = 1)

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.

The strobe signal output can be carried out in two ways, or described below.

■ **Automatic Key Scan**

LCD waveform has a 30*μ*s low pulse at the beginning of every common duty signal through setting of the SCAN bit of STBCON register. The strobe timing is shown in the following figure (Figure 8-7, *Automatic Key Scan Strobe Signal (SCAN = 1*).

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

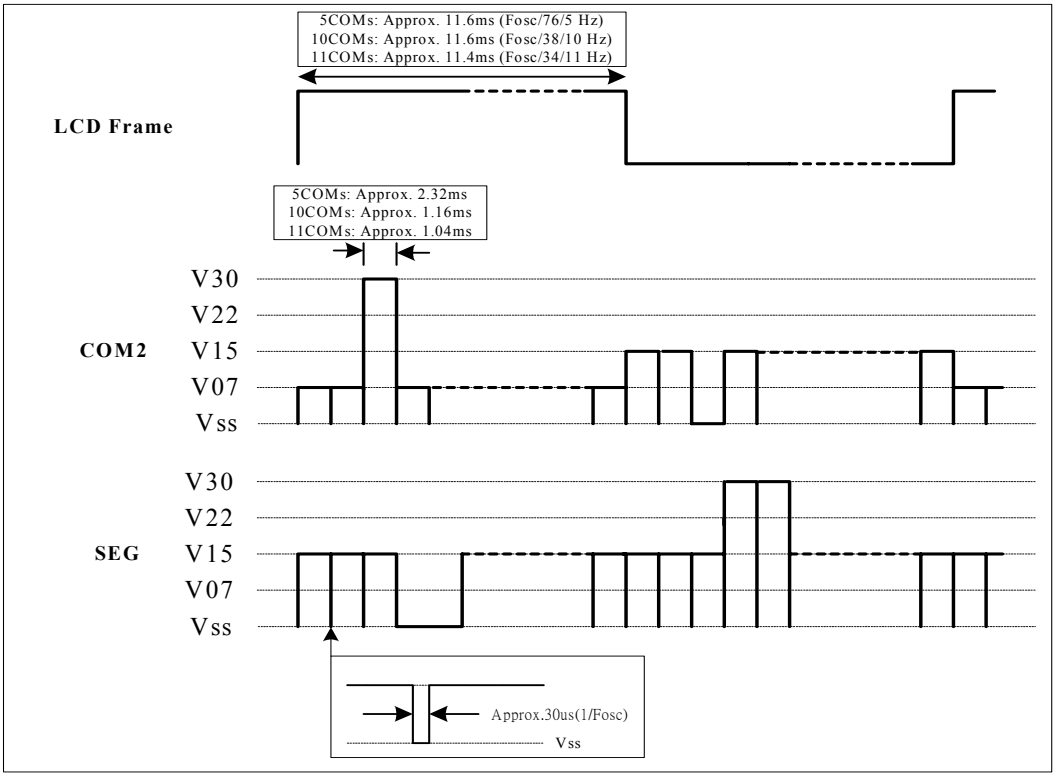


Figure 8-7 Automatic Key Scan Strobe Signal (SCAN = 1)

■ Software Key Scan

Segment is switched to strobe signal temporally by setting the BitST bit of STBCON register to "**1**" and the SCAN bit to "**0**." Set the STB3 ~ STB0 bits of STBCON register to select which pin to assign as strobe.

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

◇ **Key Strobe Pin Function**

STBCON			Key Strobe (Shared with Segment 0 ~ 15)																LCD	
SCAN	BitST	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	0	xxxx	Display waveform																	
	1	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	x	xxxx	Display waveform with automatic key scan																	

8.5.3 Input/Output Key Registers

■ **DCRA (R2Dh): Port A Direction Control Register**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PA7DC	PA6DC	PA5DC	PA4DC	**PA3DC**	**PA2DC**	**PA1DC**	**PA0DC**

Bit 3 ~ Bit 0 (PA3DC ~ PA0DC): Direction control of PortA.0~3
"0": output pin
"1": input pin

■ **PortA (R10h): Port A Register**

Bit 3 ~ Bit 0: Port A as input is selected through PA3DC~PA0DC bits of DCRA register (see above)

The input structure and two-stage pull up resistor are controlled together by PA3PU ~ PA0PU bits of PACON register and R1EN, KE bits of STBCON 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	PA6PU	PA5PU	PA4PU	PA3PU	PA2PU	PA1PU	PA0PU

Bit3 ~ Bit0 (PA3PU ~ PA0PU): Pull-up resistor (R2 large resistor) control bits
"0": Disable PortA.0 ~ PortA.3 pull-up resistor
"1": Enable PortA.0 ~ PortA.3 pull-up resistor

■ **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	PA6IE	PA5IE	PA4IE	PA3IE	PA2IE	PA1IE	PA0IE

Bit 3 ~ Bit 0 (PA3IE ~ PA0IE): Interrupt control bit

"0": Disable interrupt function

"1": Enable interrupt function

NOTE

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

■ **PAINTSTA (R2Ch): 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 3 ~ Bit 0 (PA3I ~ PA0I): INT status of Port A interrupt

Set to ("**1**") when pin falling edge is detected

Clear ("**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	KE	R1EN	BitST	STB3	STB2	STB1	STB0

Bit 7 (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 6 (KE): Key input enable/disable control bit

"0": Disable Key input function (PORTA register does NOT correspond with Key input in software scan mode)

"1": Enable Key input function (PORTA register corresponds with Key input in software scan mode)

Bit 5 (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 4 (BitST): Enable bit strobe

"0": Display waveform

"1": Strobe signal specified as STB3 ~ 0 defined.

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

■ **Code Example:**

```
; Key matrix 1 (Port A and Ground):
;  ===  Sleep mode
PAIN_SR:
       :
;  ---  Port A 0~7 input pins
     MOV    A,#0XFF
     MOV    DCRA,A
;  ---  PortA wakeup
     MOV    A,#11111111B
     MOV    PAWAKE,A
;  ---  R1EN & R2EN Pull-up & KE enable.
     MOV    A,#0XFF
     MOV    PACON,A
     BS     STBCON,R1EN
     BS     STBCON,KE
;  ---  Port A interrupt enable.
     MOV    A,#11111111B
     MOV    PAINTEN,A
     CLR    PAINTSTA
     BS     CPUCON,GLINT
;  ---  Sleep MODE
     BC     CPUCON,MS1
PAINloop:
     SLEP
     NOP
       :
     SJMP   PAINloop
```

```
;  ***  Interrupt PortA data
INPTINT:
     PUSH
     MOVRP  A,PAINTSTA
     MOV    Key_No,A
     CLR    PAINTSTA
     POP
     RETI
```

(Continuation)

```
; Key matrix 2 (Port A.0~3 and SEG0 ~ SEG15):
;  ***  Key scan function
       :
     LCD display setting
       :
     MOV    A,#0X0F
     MOV    DCRA,A              ; Port A 0~3 input pins
     OR     PACON,A             ; R2EN enable
     MOV    A,#0X0F
     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
     MOV    A,PORTA             ; Port A input data
     JE     A,#0X0F,KeyIdle
;  ===  Key scan routine
KeyScan:
     CLR  STBCON                ; Auto-key scan disable
KeyLoop:
     BS     STBCON,R1EN         ; R1EN enable
     BS     STBCON,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

ePS6000 Series provides direct drive LCD. It supports multiplexed drive for 60SEGs*11COMs which allows you to use pads as an LCD driver pin or as key input port. The available LCD RAM corresponds directly with LCD Pixel. The LCD frame rate is as follows:

Duty	LCD Frame Rate
1/5	Approx. 11.6ms (Fosc/76/5 Hz)
1/10	Approx. 11.6ms (Fosc/38/10 Hz)
1/11	Approx. 11.4ms (Fosc/34/11 Hz)

This embedded LCD driver generates waveforms to drive the display.

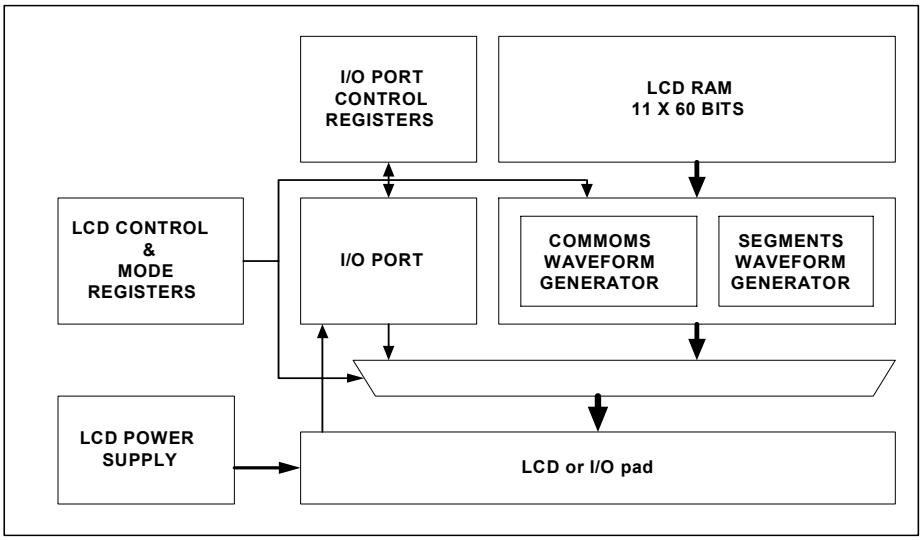


Figure 8-8a LCD Driver Function Block Diagram

The following is the LCD pin configuration:

SEG0 / Key Strobe0 ⋮ SEG15 / Key Strobe15
SEG16 ⋮ SEG43
SEG44 / PortC.0 ⋮ SEG51 / PortC.7
SEG52 / PortB.0 ⋮ SEG59 / PortB.7
COM0 ⋮ COM10

Figure 8-8b LCD Pin Configuration

8.6.1 LCD Driver Registers

■ **LCDCON (R32h): LCD Control Register**

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

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 in Section 8.6.2 below)

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).

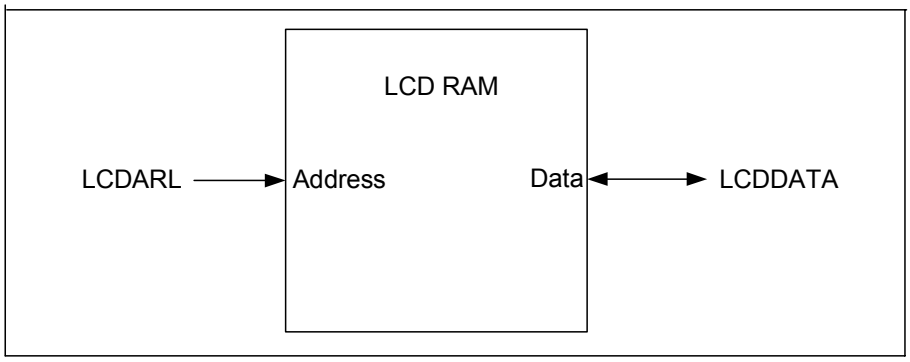


Figure 8-9 LCDDATA Register Access through LCD RAM

■ **POST_ID (R33h): 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 through setting of 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,#00000001B
     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  #0x40
;  ===  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

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

RAM Address LCDARL		COM 8	COM 9	COM 10
		Bit 0	Bit 1	Bit 2
SEG0	00H			
:	:			
SEG59	7BH			

8.6.3 LCD Driving Method Circuit

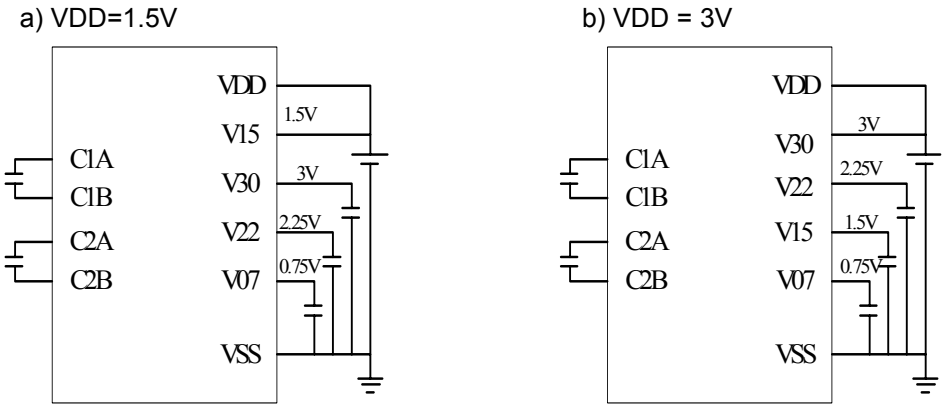


Figure 8-10 LCD Driving Method Circuit

8.6.4 LCD Waveforms for 1/11 Duty

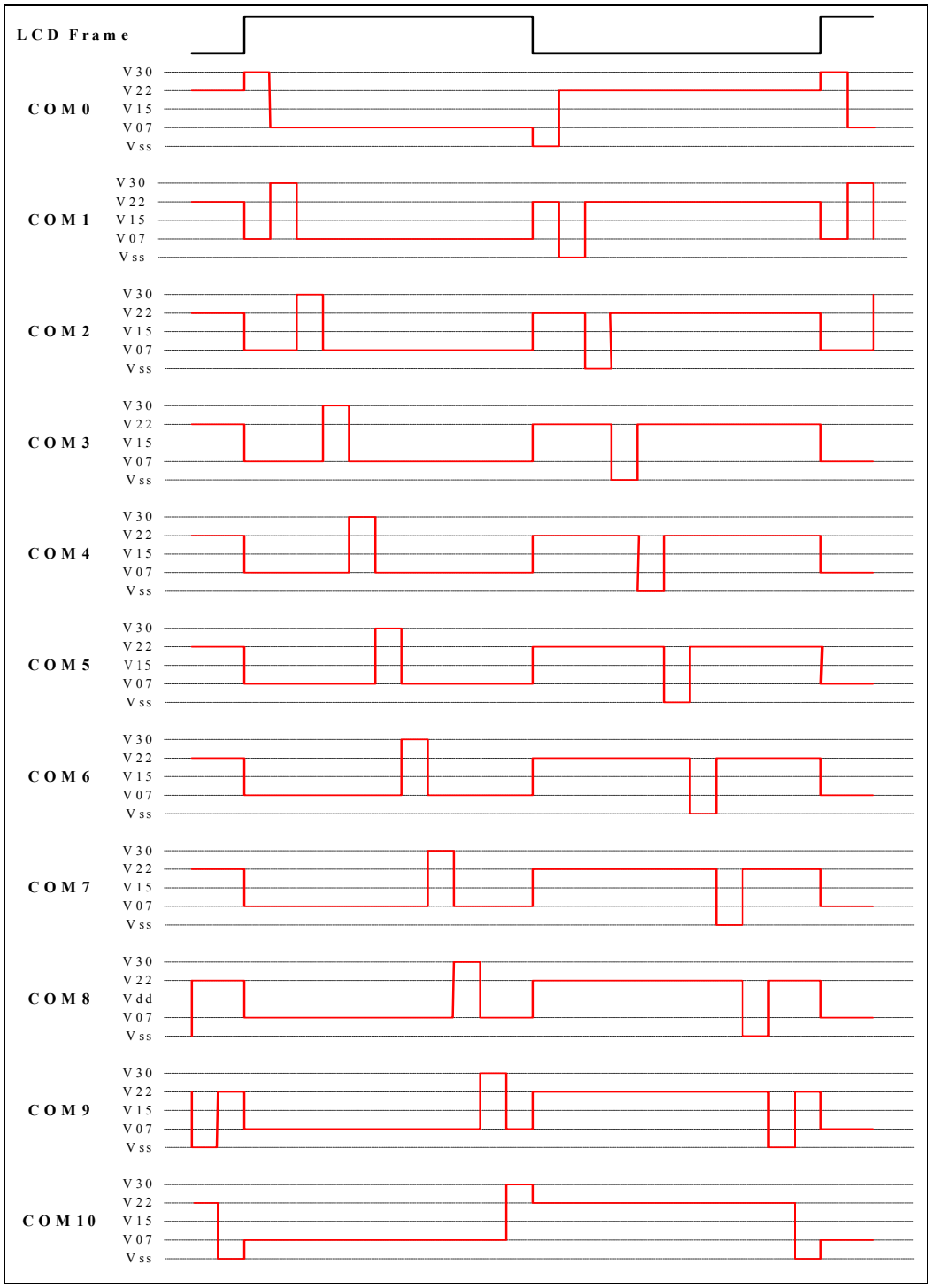
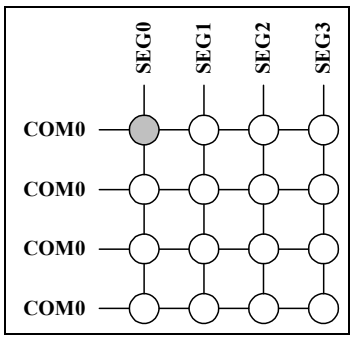


Figure 8-11a LCD COM Waveform



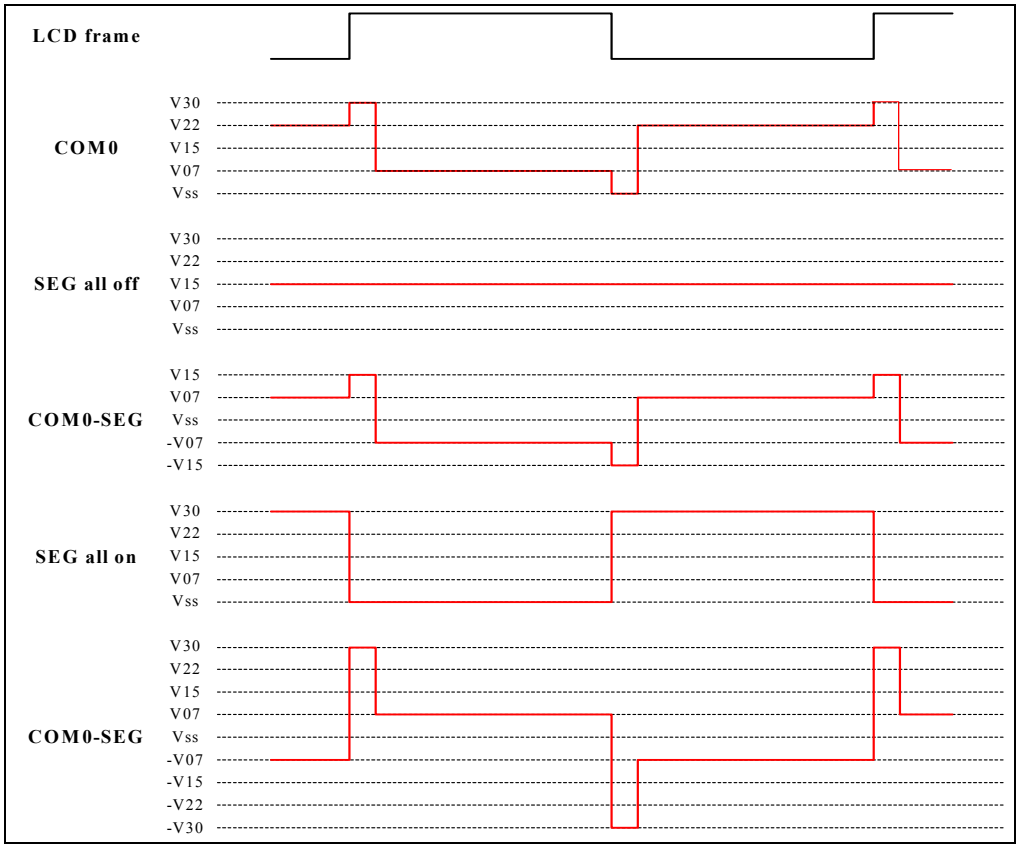


Figure 8-11b LCD COM & SEG Waveform

9 Electrical Characteristics

9.1 VDD=1.5V Electrical Characteristics

■ Absolute Maximum Ratings

Items	Sym.	Condition	Limits	Unit
Supply voltage	VDD		–0.3 to +2.0	V
Supply voltage for ICE	VCC		–0.3 to +3.6	
Input voltage (general input port)	VIN		–0.5 to VDD +0.5	V
Input voltage for ICE	VIN1		–0.5 to VCC +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
Supply voltage for ICE	VCC		2.2 to 3.6	
Input voltage	VIH		VDD x 0.9 to VDD	V
	VIL		0 to VDD x 0.1	V
Input voltage for ICE	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_{HOSC}	Main-clock frequency	RC OSC., R=910kΩ	140	200	260	KHz
			RC OSC., R=560kΩ	210	300	390	
			RC OSC., R=330kΩ	350	500	650	
	Fosc	Sub-clock frequency	RC OSC., R=2.2MΩ	24.6	32.8	41	KHz
			Internal RC OSC.	24.6	32.8	41	
			Crystal OSC.	-	32.768	-	
Supply Current	Idd1	SLEEP mode	VDD=1.5V, no load	-	-	1	μA
	Idd2	IDLE mode	VDD=1.5V RC OSC, LCD enabled, no load	-	3	4	
	Idd3		VDD=1.5V, X'tal OSC, LCD enabled, no load	-	2.5	3.5	
	Idd4	SLOW mode	VDD=1.5V, RC OSC, LCD enabled, no load, utility rate of RAM 25%	-	4	6	
	Idd5		VDD=1.5V, X'tal OSC, LCD enabled, no load, utility rate of RAM 25%	-	4	5.5	
	Idd6	FAST mode	VDD=1.5V, F_{HOSC} =200KHz, LCD enabled, no load	-	16	22	
	Idd7		VDD=1.5V, F_{HOSC} =300KHz, LCD enabled, no load	-	20	35	
	Idd8		VDD=1.5V, F_{HOSC} =500KHz, LCD enabled, no load	-	30	50	
Input Voltage	VIH1	PA[0:7] ,PB[0:7],PC[0:7] (as general input port)		VDD×0.7	-	VDD	V
	VIL1			0	-	VDD×0.3	

(Continuation)

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	450	750	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
Data Retention Voltage	Vret			1.2	-	-	V
Power On Reset Voltage	Vpor			0.9	1.0	1.1	V
LCD Enabled							
Output Current	IOH1	PA[0:7], PB[0:7], PC[0:7] (as general output port)	VDD=1.5V, VOH=1.2V, LCD enabled	-0.3	-0.5	-0.9	mA
	IOL1		VDD=1.5V, VOL=0.2V, LCD enabled	0.6	0.9	1.3	
Large Pull-up Resistance	RPU1	PA[0:7]	Key high resistance, pulled up by R2, LCD enabled, Vin2=0.5V	300	550	1100	KΩ
	RPU3	PB[0:7], PC[0:7]	Vin=0.5V, LCD enabled	90	180	360	
Small Pull-up Resistance	RPU2	PA[0:3]	Key high resistance, pulled up by R2//R1, LCD enabled, Vin2=0 V	50	90	180	KΩ
	RPU4	PA[4:7]	Vin=1V, LCD enabled	20	35	70	
LCD Disabled							
Output Current	IOH1	PA[0:7], PB[0:7], PC[0:7] (as general output port)	VDD=1.5V, VOH=1.2V, LCD disabled	-0.6	-0.9	-1.3	mA
	IOL1		VDD=1.5V, VOL=0.2V, LCD disabled	0.6	0.9	1.3	
Large Pull-up Resistance	RPU1	PA[0:7]	Key high resistance, pulled up by R2, LCD disabled, Vin2=0.5V	180	280	450	KΩ
	RPU3	PB[0:7], PC[0:7]	Vin=0.5V, LCD disabled	60	90	150	
Small Pull-up Resistance	RPU2	PA[0:3]	Key high resistance, pulled up by R2//R1, LCD disabled, Vin2=0 V	30	45	80	KΩ
	RPU4	PA[4:7]	Vin=1V, LCD disabled	14	19	28	
LCD Driver							
LCD Display Output ON-resistance	ROC	Com[0:10]	VOH=V30 +/- 0.1V	0.35	0.4	0.50	KΩ
			VOM=V22 +/- 0.1V	0.55	0.65	0.75	
			VOM=V07 +/- 0.1V	0.35	0.4	0.45	
			VOL=0.1V	0.25	0.3	0.35	
	ROS	Seg[0:59]	VOH=V30 +/- 0.1V	0.35	0.4	0.50	KΩ
			VOM=V15 +/- 0.1V	0.45	0.60	0.85	
			VOL=0.1V	0.25	0.3	0.35	
Strobe Output ON-resistance	ROP	Seg[0:15] (as key strobe)	V=VDD-0.2V	100	200	350	KΩ
	RON		V=0.2V	0.7	1.3	2	

9.2 VDD=3.0V Electrical Characteristics

■ Absolute Maximum Ratings

Items	Sym.	Condition	Limits	Unit
Supply voltage	VDD		–0.3 to +3.6	V
Supply voltage for ICE	VCC		–0.3 to +3.6	
Input voltage (general input port)	VIN		–0.5 to VDD +0.5	V
Input voltage for ICE	VIN1		–0.5 to VCC +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		2.4 to 3.6	V
Supply voltage for ICE	VCC		2.2 to 3.6	
Input voltage	VIH		VDD x 0.9 to VDD	V
	VIL		0 to VDD x 0.1	V
Input voltage for ICE	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= 3.0V)

Parameter	Sym.	Condition		Min	Typ	Max	Unit
CLOCK	F_{HOSC}	Main-clock frequency	RC OSC., R=910kΩ	140	200	260	KHz
			RC OSC., R=560kΩ	210	300	390	
			RC OSC., R=330kΩ	350	500	650	
	F_{OSC}	Sub-clock frequency	RC OSC., R=2.2MΩ	24.6	32.8	41	KHz
			Internal RC OSC.	24.6	32.8	41	
			Crystal OSC.	-	32.768	-	
Supply Current	Idd1	SLEEP mode	VDD=3.0V, no load	-	-	3	μA
	Idd3	IDLE mode	VDD=3.0V RC OSC, LCD enabled, no load	-	8	12	
	Idd5		VDD=3.0V, X'tal OSC, LCD enabled, no load	-	7	11	
	Idd6	SLOW mode	VDD=3.0V, RC OSC, LCD enabled, no load	-	12	14	
	Idd7		VDD=3.0V, X'tal OSC, LCD enabled, no load		12	15	
	Idd8	FAST mode	VDD=3.0V, F_{HOSC} =200KHz, LCD enabled, no load	-	40	50	
	Idd9		VDD=3.0V, F_{HOSC} =300KHz, LCD enabled, no load	-	60	70	
	Idd10		VDD=3.0V, F_{HOSC} =500KHz, LCD enabled, no load	-	90	110	
Input Voltage	VIH1	PA[0:7] ,PB[0:7],PC[0:7] (as general input port)		VDD×0.7	-	VDD	V
	VIL1			0	-	VDD×0.3	

(Continuation)

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
Output Current	IOH1	PA[0:7], PB[0:7], PC[0:7] (as general output port)	VDD=3V, VOH=2.7V	-1.8	-2.0	-2.3	mA
	IOL1		VDD=3V, VOL=0.2V	1.8	2.0	2.3	
Large Pull-up Resistance	RPU1	PA[0:7]	Key high resistance, pulled-up by R2, LCD enabled, Vin2=0.5V	90	120	160	KΩ
	RPU3	PB[0:7], PC[0:7]	Vin=0.5V	30	40	60	
	RPU5	RSTB	Vin=GND	120	150	200	
Small Pull-up Resistance	RPU2	PA[0:3]	Key high resistance, pulled up by R2//R1, LCD enable, Vin2=0 V	14	17	22	KΩ
	RPU4	PA[4:7]	Vin=2V	6	8	10	
	RPU6	RSTB	Vin=2V	10	13	16	
Large Pull-down Resistance	RPD1	TEST	Vin=VDD	100	120	150	KΩ
Small Pull-down Resistance	RPD2	TEST	Vin=1V	1.6	1.8	2.2	KΩ
Data Retention Voltage	Vret			2.1	-	-	V
Power On Reset Voltage	Vpor			0.9	1.0	1.1	V
LCD Driver							
LCD Display Output ON-resistance	ROC	Com[0:10]	VOH=V30 +/- 0.1V	0.35	0.4	0.50	KΩ
			VOM=V22 +/- 0.1V	0.55	0.65	0.75	
			VOM=V07 +/- 0.1V	0.35	0.4	0.45	
			VOL=0.1V	0.25	0.3	0.35	
	ROS	Seg[0:59]	VOH=V30 +/- 0.1V	0.35	0.4	0.50	KΩ
			VOM=V15 +/- 0.1V	0.45	0.60	0.85	
			VOL=0.1V	0.25	0.3	0.35	
Strobe Output ON-resistance	ROP	Seg[0:15] (as key strobe)	V=VDD-0.2V	30	45	60	KΩ
	RON		V=0.2V	0.3	0.45	0.65	

10 Pin Type Circuit Diagrams

■ **Reset Pin Type**

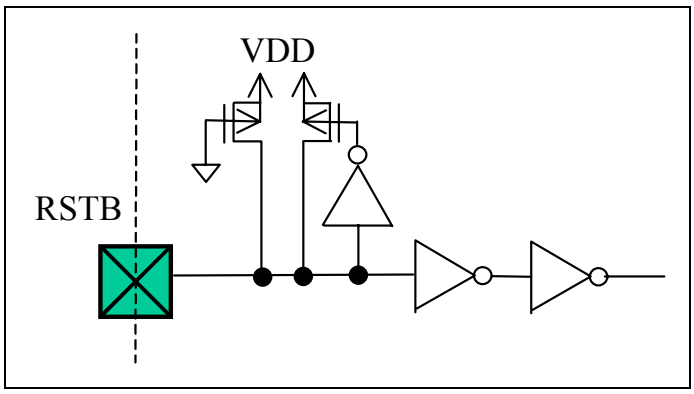


Figure 10-1a Reset Pin Type Circuit Diagram

■ **TEST Pin Type**

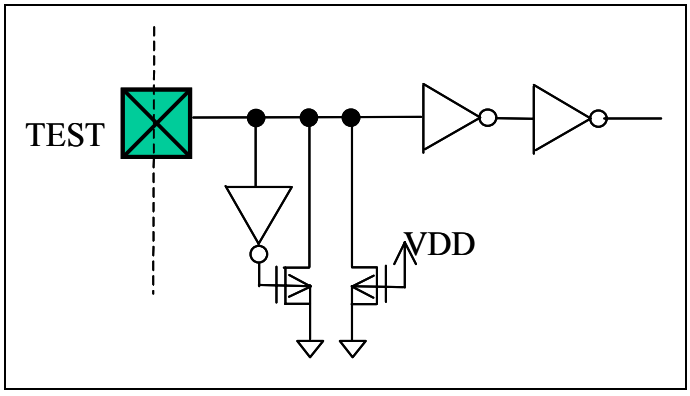


Figure 10-1b Test Pin Type Circuit Diagram

■ **Oscillator Pin Type**

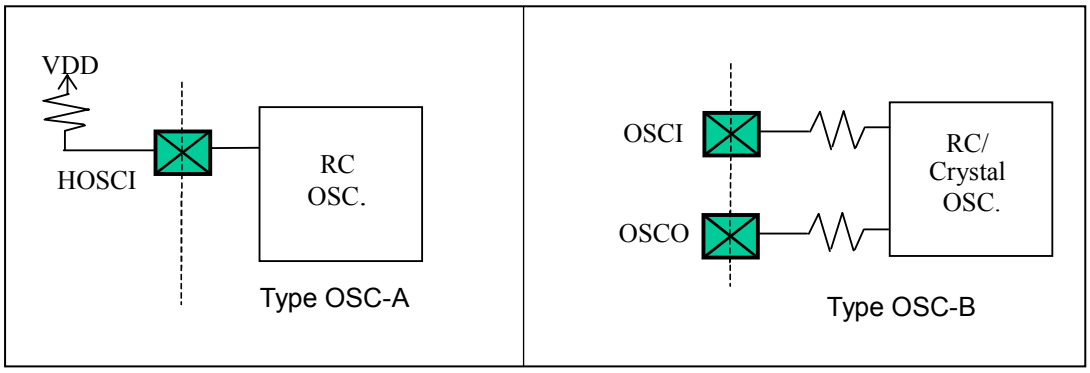


Figure 10-1c Oscillator Pin Type Circuit Diagram

■ **Input Pin Type**

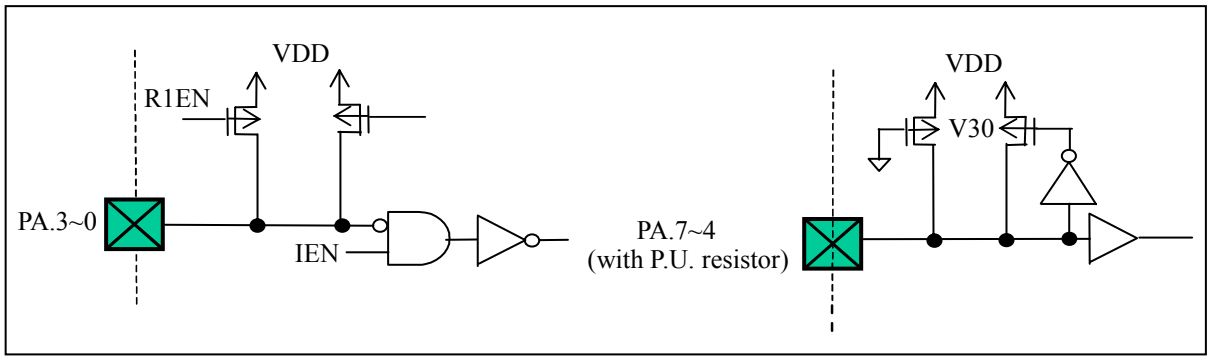


Figure 10-1d Input Pin Type Circuit Diagram

■ **SEG and I/O Share Pin Type**

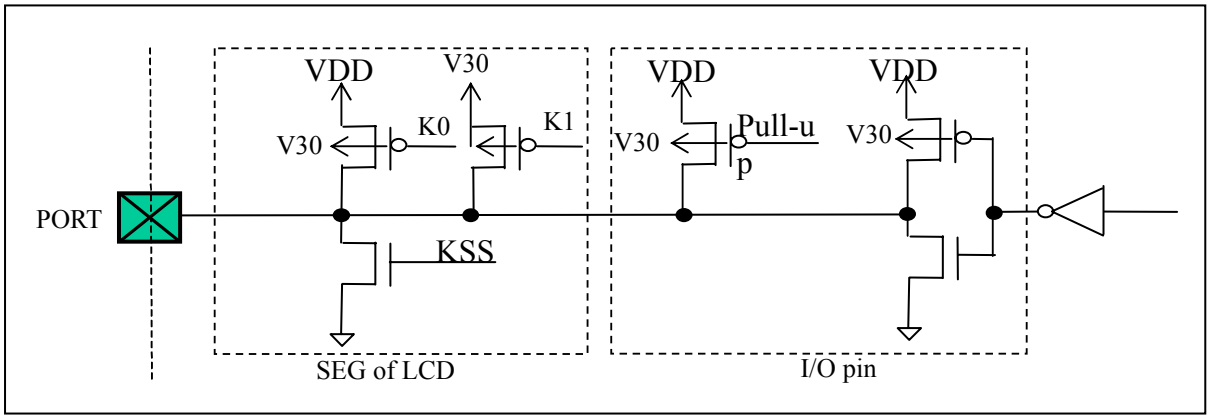


Figure 10-1e SEG & I/O Share Pin Type Circuit Diagram

■ **General SEG and COM Pin Type**

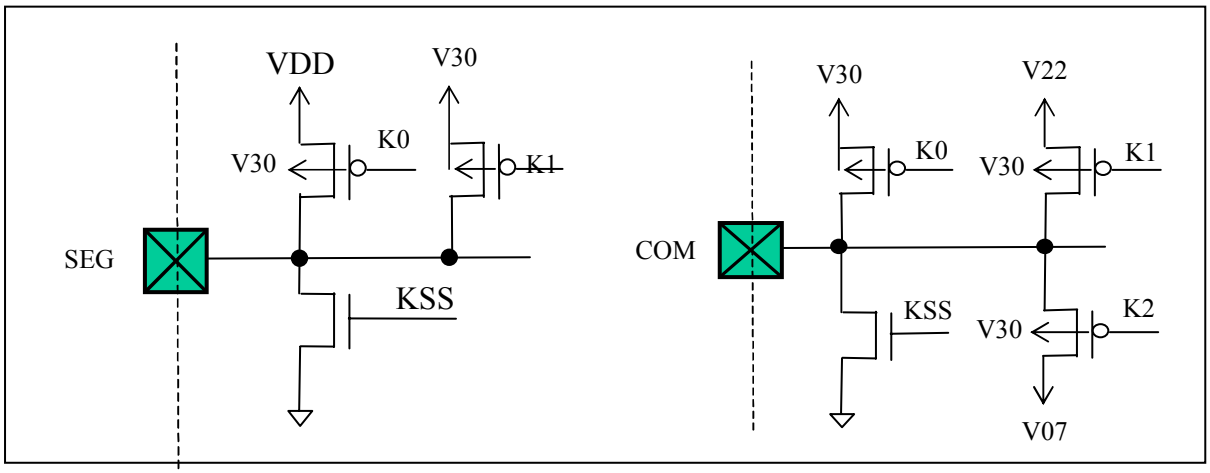


Figure 10-1f General SEG & COM Share Pin Type Circuit Diagram

11 Application Circuit

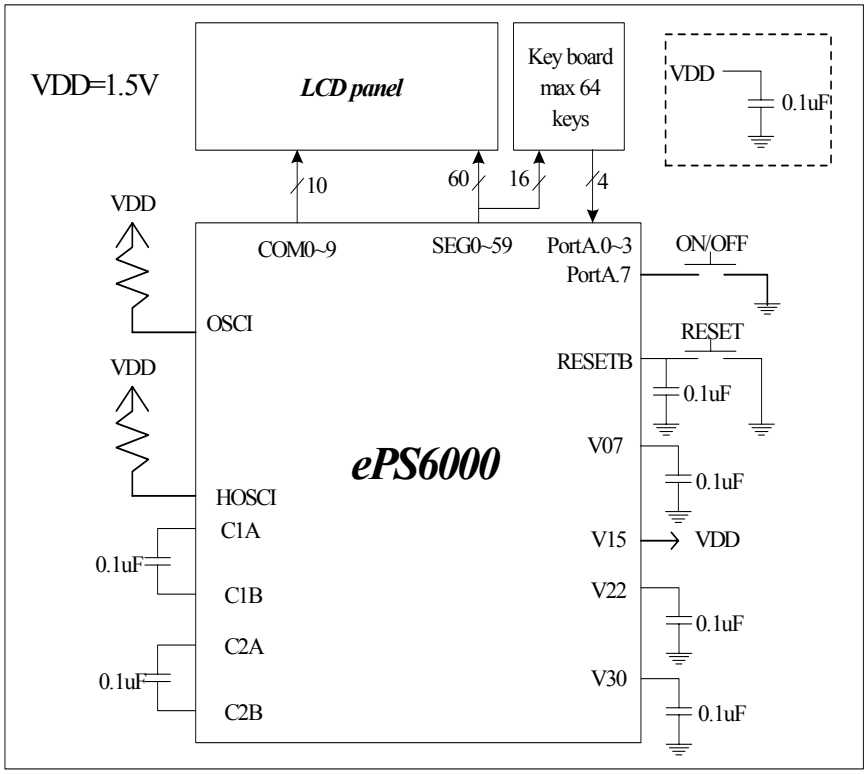


Figure 11-1a VDD=1.5 Application Circuit Diagram

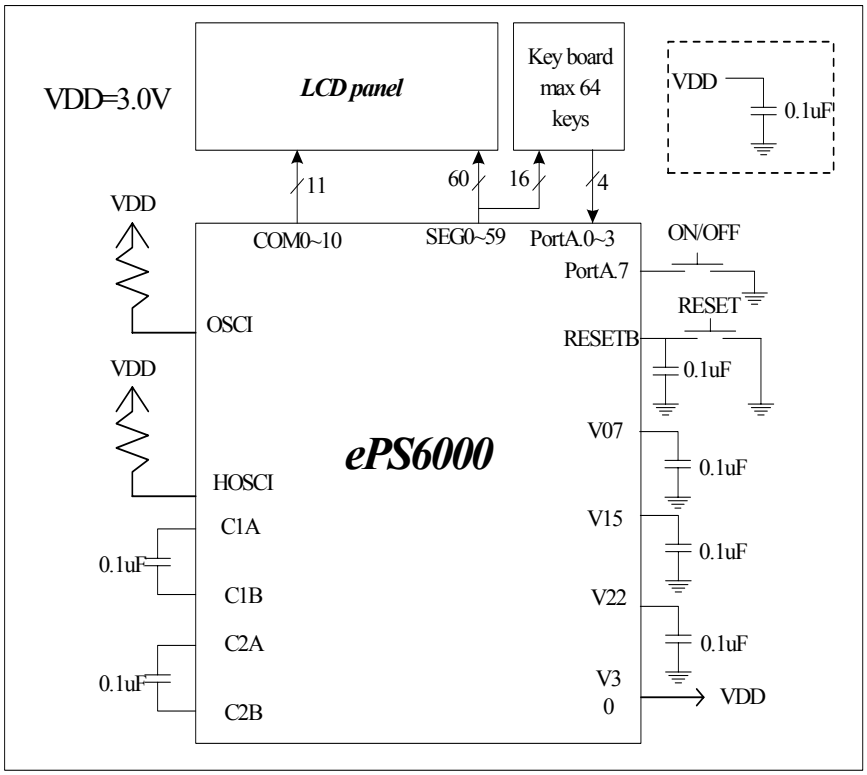


Figure 11-1b VDD=3.0 Application Circuit Diagram

12 Instruction Set

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

Type	Instruction Binary	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 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 00aa 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
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 00aa aaaa aaaa aaaa	LJMP addr (2 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
Data Transfer	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
	0100 1110 kkkk kkkk	MOV A,#k	A ← k	None	1
	0010 0100 rrrr rrrr	CLR r	r ← 0	Z	1

[1] *S0CALL addressing ability is from 0x000 to 0xFFF (4K space)*

[2] *The maximum jump range is 16K absolute address*

(Continuation)

Type	Instruction Binary	Mnemonic	Operation	Status Affected	Cycles
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 11ii rrrr rrrr	TBRD i,r	r←ROM[(TABPTR)] [3, 4]	None	2
	0010 1111 rrrr rrrr	TBRD A,r	r←ROM[(TABPTR+ACC)] [4]	None	2
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
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 [5]	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
	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

[3] *TBRD i, r:*

r← ROM [(TABPTR)]

i=00: TABPTR not change

wi=01: TABPTR←TABPTR+1

i=10: TABPTR←TABPTR-1

[4] *TABPTR=(TABPTRM: TABPTRL)*

Bit0=0: Low byte of the pointed ROM data

Bit0=1: High byte of the pointed ROM data

NOTE

- *Bit 0 of TABPTRL is used to select low byte or high byte of the pointed ROM data.*
- *The maximum table look up space is internal 32K bytes (16K 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, i.e., B=/C*

(Continuation)

Type	Instruction Binary	Mnemonic	Operation	Status Affected	Cycles
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
Exchange	0101 0100 rrrr rrrr	EX r	r(7-0) ←→ A(7-0)	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
Nibble Operation	0101 0010 rrrr rrrr	EXL r	r(3-0) ←→ A(3-0)	None	1
	0101 0011 rrrr rrrr	EXH r	r(7-4) ←→ A(3-0)	None	1
	0010 0110 rrrr rrrr	MOVL r,A	r(3-0) ← A(3-0)	None	1
	0010 1000 rrrr rrrr	MOVH r,A	r(7-4) ← A(3-0)	None	1
	0010 1001 rrrr rrrr	MOVL A,r	A(3-0) ← r(3-0); A(7-4) ← 0	None	1
	0010 1010 rrrr rrrr	MOVH A,r	A(3-0) ← r(7-4); A(7-4) ← 0	None	1
	0000 0001 rrrr rrrr	SFR4 r	r(7-4)←A(3-0);r(3-0)←r(7-4); A(3-0)←r(3,0)	None	1
	0100 1111 rrrr rrrr	SFL4 r	r(3-0)←A(3-0);r(7-4)←r(3-0); A(3-0)←r(7-4)	None	1
	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

13 Pad Diagram and Locations

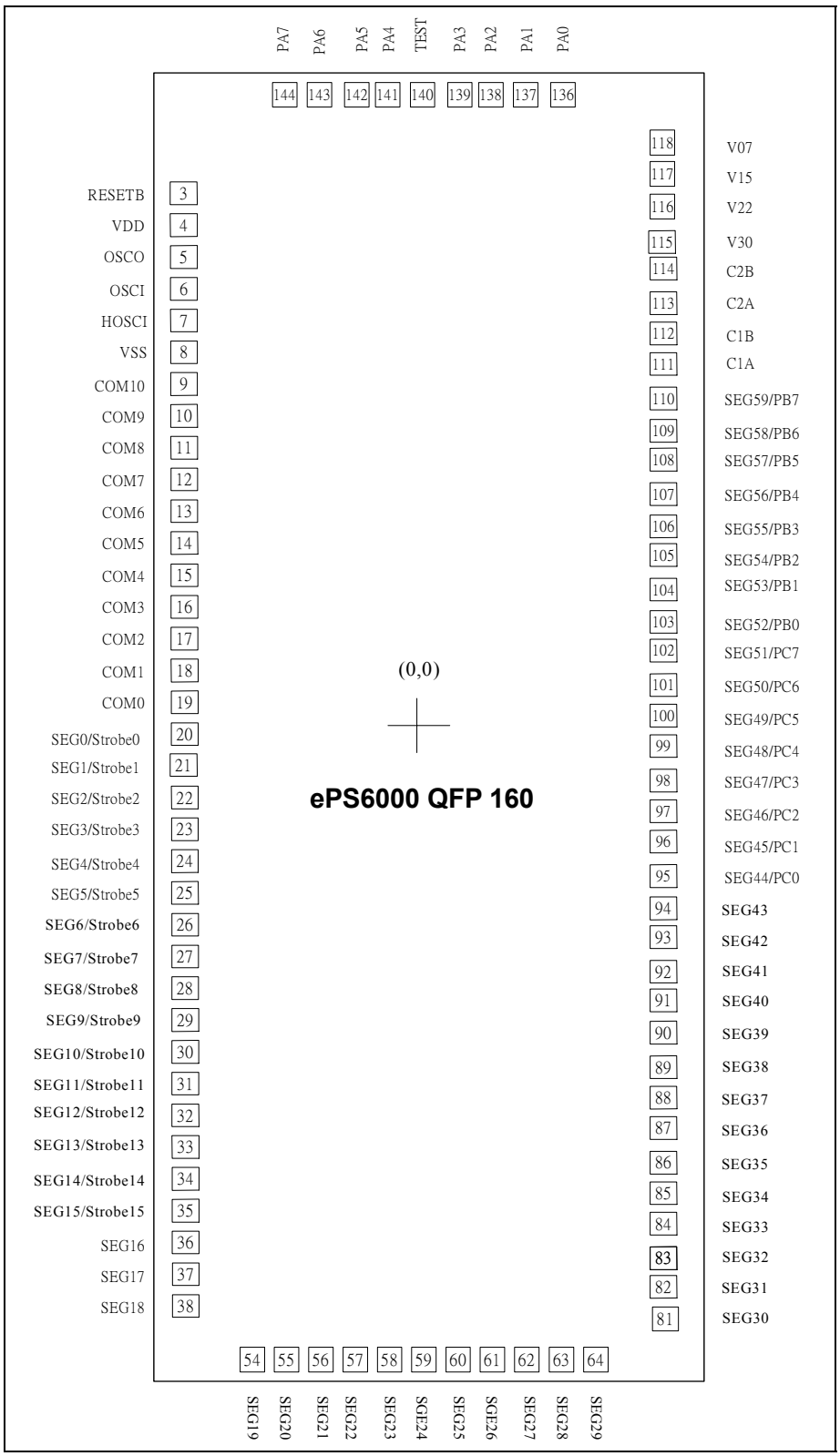


Figure 13-1 ePS6000 Pad Diagram

Chip Size: 1490 * 4850um^2

Pin N0.	Symbol	X	Y	Pin No.	Symbol	X	Y
1	NC			41	NC		
2	NC			42	NC		
3	RESETB	-635.0	1623.15	43	NC		
4	VDD	-635.0	1515.65	44	NC		
5	OSCO	-635.0	1408.15	45	NC		
6	OSCI	-635.0	1303.15	46	NC		
7	HOSCI	-635.0	1198.15	47	NC		
8	VSS	-635.0	1090.65	48	NC		
9	COM10	-635.1	975.1	49	NC		
10	COM9	-635.1	870.1	50	NC		
11	COM8	-635.1	765.1	51	NC		
12	COM7	-635.1	660.1	52	NC		
13	COM6	-635.1	555.1	53	NC		
14	COM5	-635.1	450.1	54	SEG19	-577.05	-2316.1
15	COM4	-635.1	345.1	55	SEG20	-472.05	-2316.1
16	COM3	-635.1	240.1	56	SEG21	-367.05	-2316.1
17	COM2	-635.1	135.1	57	SEG22	-262.05	-2316.1
18	COM1	-635.1	30.1	58	SEG23	-157.05	-2316.1
19	COM0	-635.1	-74.9	59	SEG24	-52.05	-2316.1
20	SEG0/Strobe0	-635.1	-179.9	60	SEG25	52.95	-2316.1
21	SEG1/Strobe1	-635.1	-284.9	61	SEG26	157.95	-2316.1
22	SEG2/Strobe2	-635.1	-389.9	62	SEG27	262.95	-2316.1
23	SEG3/Strobe3	-635.1	-494.9	63	SEG28	367.95	-2316.1
24	SEG4/Strobe4	-635.1	-599.9	64	SEG29	472.95	-2316.1
25	SEG5/Strobe5	-635.1	-704.9	65	NC		
26	SEG6/Strobe6	-635.1	-809.9	66	NC		
27	SEG7/Strobe7	-635.1	-914.9	67	NC		
28	SEG8/Strobe8	-635.1	-1019.9	68	NC		
29	SEG9/Strobe9	-635.1	-1124.9	69	NC		
30	SEG10/Strobe	-635.1	-1229.9	70	NC		
31	SEG11/Strobe	-635.1	-1334.9	71	NC		
32	SEG12/Strobe	-635.1	-1439.9	72	NC		
33	SEG13/Strobe	-635.1	-1544.9	73	NC		
34	SEG14/Strobe	-635.1	-1649.9	74	NC		
35	SEG15/Strobe	-635.1	-1754.9	75	NC		
36	SEG16	-635.1	-1859.9	76	NC		
37	SEG17	-635.1	-1964.9	77	NC		
38	SEG18	-635.1	-2069.9	78	NC		
39	NC			79	NC		
40	NC			80	NC		

(Continuation)

Pin NO.	Symbol	X	Y	Pin NO.	Symbol	X	Y
81	SEG30	635.1	-2157.45	121	NC		
82	SEG31	635.1	-2052.45	122	NC		
83	SEG32	635.1	-1947.45	123	NC		
84	SEG33	635.1	-1842.45	124	NC		
85	SEG34	635.1	-1737.45	125	NC		
86	SEG35	635.1	-1632.45	126	NC		
87	SEG36	635.1	-1527.45	127	NC		
88	SEG37	635.1	-1422.45	128	NC		
89	SEG38	635.1	-1317.45	129	NC		
90	SEG39	635.1	-1212.45	130	NC		
91	SEG40	635.1	-1107.45	131	NC		
92	SEG41	635.1	-1002.45	132	NC		
93	SEG42	635.1	-897.45	133	NC		
94	SEG43	635.1	-792.45	134	NC		
95	SEG44/PC0	635.0	-678.55	135	NC		
96	SEG45/PC1	635.0	-571.35	136	PA0	383.8	2315.0
97	SEG46/PC2	635.0	-464.15	137	PA1	278.8	2315.0
98	SEG47/PC3	635.0	-356.95	138	PA2	169.4	2315.0
99	SEG48/PC4	635.0	-249.75	139	PA3	64.4	2315.0
100	SEG49/PC5	635.0	-142.55	140	TEST	-42.8	2315.0
101	SEG50/PC6	635.0	-35.35	141	PA4	-150.0	2315.0
102	SEG51/PC7	635.0	71.85	142	PA5	-255.0	2315.0
103	SEG52/PB0	635.0	179.05	143	PA6	-364.4	2315.0
104	SEG53/PB1	635.0	286.25	144	PA7	-469.4	2315.0
105	SEG54/PB2	635.0	393.45	145	NC		
106	SEG55/PB3	635.0	500.65	146	NC		
107	SEG56/PB4	635.0	607.85	147	NC		
108	SEG57/PB5	635.0	715.05	148	NC		
109	SEG58/PB6	635.0	822.25	149	NC		
110	SEG59/PB7	635.0	929.45	150	NC		
111	C1A	635.0	1036.75	151	NC		
112	C1B	635.0	1141.75	152	NC		
113	C2A	635.0	1246.75	153	NC		
114	C2B	635.0	1351.75	154	NC		
115	V30	635.0	1459.25	155	NC		
116	V22	635.0	1566.75	156	NC		
117	V15	635.0	1671.75	157	NC		
118	V07	635.0	1776.75	158	NC		
119	NC			159	NC		
120	NC			160	NC		

NOTE: For PCB layout, IC substrate must be connected to Vss