ePS7200

RISC II Series Microcontroller

Product Specification

DOC. VERSION 1.0

ELAN MICROELECTRONICS CORP.
December 2014

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 1st Road
Hsinchu Science Park
Hsinchu, TAIWAN 30076
Tel: +886 3 563-9977
Fax: +886 3 563-9966
webmaster@emc.com.tw
http://www.emc.com.tw

Hong Kong:

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

Shenzhen:

ELAN Microelectronics Shenzhen, Ltd.

8A Floor, Microprofit Building
Gaoxin South Road 6
Shenzhen Hi-Tech Industrial Park
South Area, Shenzhen
CHINA 518057
Tel: +86 755 2601-0565
Fax: +86 755 2601-0500
elan-sz@elanic.com.cn

USA:

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

Shanghai:

ELAN Microelectronics Shanghai, Ltd.

6F, Ke Yuan Building
No. 5 Bibo Road
Zhangjiang Hi-Tech Park
Shanghai, CHINA 201203
Tel: +86 21 5080-3866
Fax: +86 21 5080-0273
elan-sh@elanic.com.cn

Contents

Specification Revision History

Doc. Version	Revision Description	Date
0.1	Initial version	2012/08/03
0.2	1. Modified the Pin Assignment 2. Added the Electrical Characteristics 3. Added the Pad Diagram and Pad Locations	2013/06/13
0.3	1. Modified the capacitor value that connected between Vreg and Vss 2. Modified the VDD range on the Absolute Maximum Ratings 3. Modified the operating voltage	2014/03/26
1.0	1. Deleted the oscillator frequency: 400kHz and 500kHz 2. Modified the operating voltage 3. Modified the register initial status	2014/12/12

1 General Description

The **ePS7200** is an 8-bit RISC MCU embedded with a 10×58 LCD driver along with two 8-bit timers, one 16-bit general timer, and a Watchdog Timer. It also has on-chip 1.5K bytes RAM and 24K words program ROM. It is highly ideal for advance scientific calculator application, particularly those requiring high performance and low cost solution.

The MCU core is one of ELAN's second generation RISC based IC's, known as RISC II (RII) series. The core is specifically designed for low power and portable device applications. The ePS7200 also supports Fast, Slow, and Idle modes, as well as Sleep mode to enhance its low power consumption features.

IMPORTANT NOTES

- *Do not use Register BSR (02h) Bit 7 ~ Bit 4.*
- *Do not use Register BSR1 (05h) Bit 7 ~ Bit 4.*
- *Check the range of BSR and BSR1 which should be from 0x00~0x0B.*
- *Do not use LCD RAM 3Ah~3Fh and 7Ah~ FFh.*
- *Do not use Register JDNZ at FSR1 (04h) special register.*
- *Do not use PUSH, POP by "MOV A,r" to avoid affecting S_Z.*
- *Do not use TBRD to read the ROM value when TABPRTM: L>0xBFFF*

1.1 Application

- Calculating machine

2 Features

2.1 MCU

- 8 bit RISC MCU
- Operating voltage and speed: 1.15V~2.2V
- Clock Source: Single system clock
- Idle and Slow mode: 8kHz/ 16kHz/ 32kHz
- Fast mode: 60kHz/ 100kHz/ 150kHz/ 200kHz/ 300kHz
- One Instruction cycle time = 2 × System clock time
- Program ROM addressing: Maximum of 24K words
- 128 bytes un-banked RAM including special registers and common registers

- 12×128 bytes banked RAM
- Max. of 32-level RAM stack
- Table Lookup function is fast and highly efficient when combined with Repeat instruction
- Register-to-Register move instruction
- Compare and Branch in one instruction (2 cycles)
- Single Repeat function (256 repeat times max.)
- Decimal ADD and SUB instruction
- Full range Call and Jump ability (2 cycles)

2.2 Peripheral Configuration

- 24 general I/O pins (Port A.0~7, Port B.0~7 share with segment pins, Port C.0~1 share with common pins, Port D.0~5)
- 10 / 9 / 8 COM × 58 SEG LCD driver (embedded), 1/4 bias
- One 16-bit timer (Timer 0) with event counter function
- One 8-bit timer (Timer 1) with wake-up function
- One 8-bit timer (Timer 2)
- One 8-bit Watchdog Timer
- Key I/O function with a maximum of 48 keys (Key matrix: Port A.0~7 and Port D.0~5) when LCD driver is 10 x 58

2.3 Internal Specification

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

3 Block Diagram

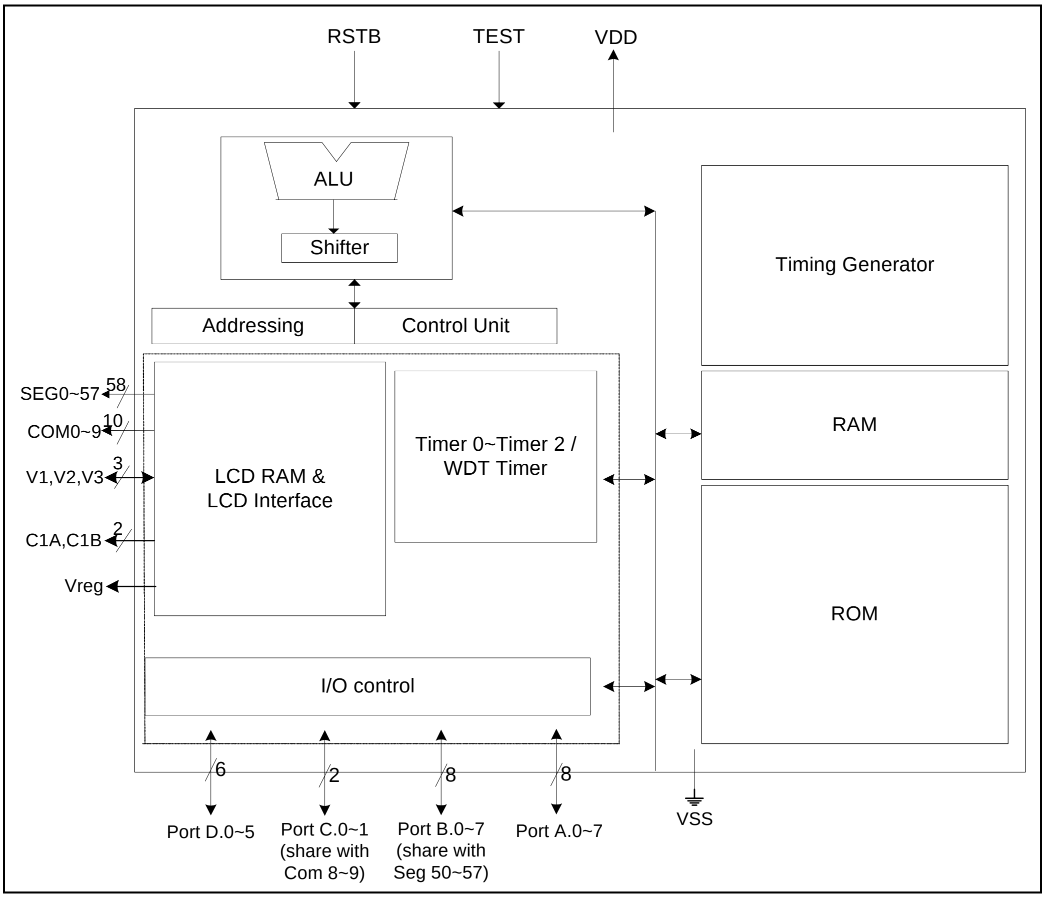


Figure 3-1 ePS7200 Block Diagram

4 Pin Assignment

- **92-pin Chip Type**

No.	Pin Name	No.	Pin Name	No.	Pin Name	No.	Pin Name
1	Port D.3	24	SEG4/Strobe4	47	SEG27	70	C1A
2	Port D.2	25	SEG5/Strobe5	48	SEG28	71	C1B
3	Port D.1	26	SEG6/Strobe6	49	SEG29	72	V1
4	Port D.0	27	SEG7/Strobe7	50	SEG30	73	V2
5	VDD	28	SEG8/Strobe8	51	SEG31	74	V3
6	RESETB	29	SEG9/Strobe9	52	SEG32	75	SEG50/Port B.0
7	Vreg	30	SEG10/Strobe10	53	SEG33	76	SEG51/Port B.1
8	VSS	31	SEG11/Strobe11	54	SEG34	77	SEG52/Port B.2
9	Test	32	SEG12/Strobe12	55	SEG35	78	SEG53/Port B.3
10	COM9/Port C.1	33	SEG13/Strobe13	56	SEG36	79	SEG54/Port B.4
11	COM8/Port C.0	34	SEG14/Strobe14	57	SEG37	80	SEG55/Port B.5
12	COM7	35	SEG15/Strobe15	58	SEG38	81	SEG56/Port B.6
13	COM6	36	SEG16	59	SEG39	82	SEG57/Port B.7/EVIN
14	COM5	37	SEG17	60	SEG40	83	Port A.7
15	COM4	38	SEG18	61	SEG41	84	Port A.6
16	COM3	39	SEG19	62	SEG42	85	Port A.5
17	COM2	40	SEG20	63	SEG43	86	Port A.4
18	COM1	41	SEG21	64	SEG44	87	Port A.3
19	COM0	42	SEG22	65	SEG45	88	Port A.2
20	SEG0/Strobe0	43	SEG23	66	SEG46	89	Port A.1
21	SEG1/Strobe1	44	SEG24	67	SEG47	90	Port A.0
22	SEG2/Strobe2	45	SEG25	68	SEG48	91	Port D.5
23	SEG3/Strobe3	46	SEG26	69	SEG49	92	Port D.4

5 Pin Description

5.1 MCU System Pins (4 Pins)

Name	I/O/P Type	Description	Note
VDD	P	The positive power supply, ranging from 1.15V~2.2V. Connect 0.1 µF to Vss.	-
VSS	P	Digital and Analog negative power supply.	-
RSTB	I	System reset pin. Low active, Connect 0.1 µF to VSS.	Int. Pull-up
TEST	I	Test mode select pin (High active). For chip internal test only, Normally connect to VSS.	Int. Pull Down

5.2 Embedded LCD Pins (74 Pins)

Name	I/O/P Type	Description	Note
COM0~COM7	O	LCD common signal output pin	-
COM8~COM9 / Port C.0~1	I/O	LCD common signal output pin or I/O pin; defined by code option	-
SEG0~SEG15	O	LCD segment signal output pin shared with Key Strobe 0~15	-
SEG16~ SEG49	O	LCD segment signal output pin	-
SEG50~SEG57 / Port B.0~7	I/O	LCD segment signal output pin or I/O pin; defined by code option	-
C1A, C1B		LCD voltage charge-pump pin. Connect 0.1µF or 1µF between C1A and C1B.	-
V3, V2, V1	O	LCD bias Pins. Connect 0.1µF or 1µF to Vss	-
Vreg	P	LCD charge-pump power supply. Connect 1~10 µF to Vss.	-

5.3 I/O Port (14 Pins)

Port	Bit	Function	I/O Type	Power Source	Description	Note
Port A	Bits 6~0	General Input	I	VDD	Key input	Int. Pull-up (R1: small resistor, R2: Controllable Large resistor)
		Interrupt and wake up	I	VDD	Input port interrupt and wake-up pin	
		General Output	O	VDD		
	Bit 7	General Input	I	VDD		Int. Pull up (R1: small resister set by code option, R2: Controllable Large resister)
		Interrupt and wake up	I	VDD	Input port interrupt and wake-up pin	
		General Output	O	VDD		
Port D	Bits 5~0	General Input	I	VDD		-
		General Output	O	VDD		

6 Code Option

Located at Address 0x000C~0x000F of Program ROM

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

> **NOTE**
>
> *For Initial mode after reset, it is recommended that the setting be adjusted to “Slow mode.”*

- Reset pin’s condition:
  - Select “Level hold” or “One short” for the reset pin
- Select the oscillator frequency:
  - Select “60kHz” or “100kHz” or “150kHz” or “200kHz” or “300kHz”
- Select the frequency for Slow and Idle mode:
  - Select “8kHz” or “16kHz” or “32kHz”
- Maximum duty ratio option:
  - Select “1/10 duty” or “1/9 duty” or “1/8 duty”
- Port B.0 control bit (SEG50):
  - Select “LCD segment signal output” or “general I/O function”
- Port B.1 control bit (SEG51):
  - Select “LCD segment signal output” or “general I/O function”
- Port B.2 control bit (SEG52):
  - Select “LCD segment signal output” or “general I/O function”
- Port B.3 control bit (SEG53):
  - Select “LCD segment signal output” or “general I/O function”
- Port B.4 control bit (SEG54):
  - Select “LCD segment signal output” or “general I/O function”
- Port B.5 control bit (SEG55):
  - Select “LCD segment signal output” or “general I/O function”
- Port B.6 control bit (SEG56):
  - Select “LCD segment signal output” or “general I/O function”
- Port B.7 control bit (SEG57):
  - Select “LCD segment signal output” or “general I/O function”

- Port C.0 control bit (COM8):
  - Select “LCD common signal output” or “general I/O function”
- Port C.1 control bit (COM9):
  - Select “LCD common signal output” or “general I/O function”
- Key matrix combination:
  - Select “Port A and SEG” or “Port A and GPIO”
- The PA.0’s Key matrix combination:
  - Select “PA.0 and SEG” or “PA.0 and GPIO or none”
- The PA.1’s Key matrix combination:
  - Select “PA.1 and SEG” or “PA.1 and GPIO or none”
- The PA.2’s Key matrix combination:
  - Select “PA.2 and SEG” or “PA.2 and GPIO or none”
- The PA.3’s Key matrix combination:
  - Select “PA.3 and SEG” or “PA.3 and GPIO or none”
- The PA.4’s Key matrix combination:
  - Select “PA.4 and SEG” or “PA.4 and GPIO or none”
- The PA.5’s Key matrix combination:
  - Select “PA.5 and SEG” or “PA.5 and GPIO or none”
- The PA.6’s Key matrix combination:
  - Select “PA.6 and SEG” or “PA.6 and GPIO or none”
- The PA.7’s Key matrix combination:
  - Select “PA.7 and SEG” or “PA.7 and GPIO or none”
- The PA.7 control function option:
  - Select “PA.7 can be controlled by KE and PA.7’s small R can be controlled by R1EN” or “PA.7 cannot be controlled by KE and PA.7’s small R cannot be controlled by R1EN”.

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

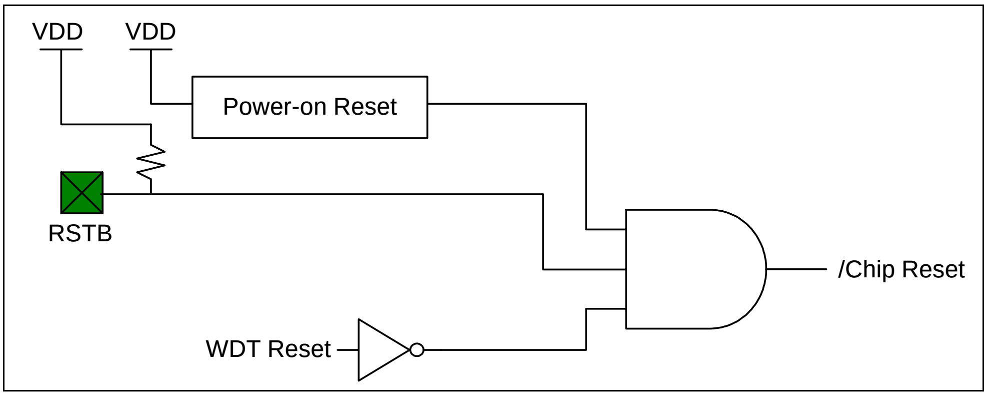


Figure 7-1 On-chip Reset Schematic Diagram

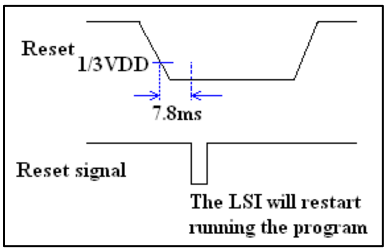
7.1.1 Power-on Reset

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

7.1.2 RSTB Pin

In normal condition, the RSTB pin is pulled up to VDD. Whenever the RSTB is at a low condition (level hold or one short, set by code option), a Reset will occur.

- Level hold: When user presses and holds the reset key, the LSI will stop running the program. After the reset key is released, the LSI will generate a reset signal to restart running the program.
- One short: When user presses and holds the reset key while the reset pin's voltage is lower than 1/3VDD, the LSI will generate a reset signal when the time lapses above 7.8ms, and restart running the program. Note that the LSI will generate a reset signal to restart running the program without having to wait for the release of the reset key.



7.1.3 WDT Time-out

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

7.1.4 Status (R0Fh)

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

Bit 0 (C): Carry flag or inverse of Borrow flag (B). Under SUB operation, Borrow flag is indicated by the inverse of Carry bit (B = /C).

Bit 1 (DC): Auxiliary carry flag

Bit 2 (Z): Zero flag

Bit 3 (OV): Overflow flag. Used in signed operation when Bit 6 is carried into or borrows from a signed bit (Bit 7).

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

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

NOTE

1. *When OV=1 after a signed arithmetic, check the SGE bit and SLE bit to verify whether overflow (carry into sign bit) or underflow (borrow from sign bit) occurred.*

   If OV=1 and SGE=1 → overflow occurred.
   If OV=1 and SLE=1 → underflow occurred.

2. *When overflow occurred, the MSB of the Accumulator should be cleared to obtain the correct value.*

   When underflow occurred, the MSB of the Accumulator should be set to obtain the correct value.

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 that the result is greater than or equal to 0 (positive value)

OV=1,   means that 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 the 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 that the result is less than or equal to 0 (negative value)

OV=1,   Underflow occurred and the result borrowed from a 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

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

Bit 7 (/TO): Reset to “**0**” at WDT time out reset. Set to “**1**” by “WDTC” instruction, power-on reset, Reset pin low condition, or when the MCU enters into Sleep Mode.

When a reset occurs, the special function registers are reset to their initial value except for the /**TO** **a**nd /**PD** bits of the Status register.

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

7.1.5 Initialization after Reset

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

7.1.5.1 Special Register

Addr.	Name	Initial Value	Addr.	Name	Initial Value
00h	INDF0	uuuu uuuu [1]	10h	Port A	uuuu uuuu
01h	FSR0	0000 0000	11h	Port B	uuuu uuuu
02h	BSR	---- 0000	12h	Port C	xxxx xxuu
03h	INDF1	uuuu uuuu [1]	13h	Port D	xxuu uuuu
04h	FSR1	1000 0000	14h	General RAM	uuuu uuuu
05h	BSR1	---- 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	uuuu uuuu	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	uuuu uuuu [1]	1Eh	General RAM	uuuu uuuu
0Fh	STATUS	cuuu uuuu [2]	1Fh	General RAM	uuuu uuuu

7.1.5.2 Control Register

Addr.	Name	Initial Value	Addr.	Name	Initial Value
20h	STBCON	0000 0000	2Dh	DCRA	1111 1111
21h	INTCON	0100 -000	2Eh	PBCON	0000 0000
22h	INTSTA	---- -000	2Fh	DCRB	1111 1111
23h	TR01CON	0000 0000	30h	PCCON	xxxx xx00
24h	TRL0L	uuuu uuuu	31h	DCRC	xxxx xx11
25h	TRL0H	uuuu uuuu	32h	LCDCON	-00- 0000
26h	TRL1	uuuu uuuu	33h	POST_ID	x111 x000
27h	TR2WCON	0000 0000	34h	CPUCON	xxxx x00c[3]
28h	TRL2	uuuu uuuu	35h	T0CL	0000 0000
29h	PACON	0000 0000	36h	T0CH	0000 0000
2Ah	PAWAKE	0000 0000	37h	PDCON	xx00 0000
2Bh	PAINTEN	0000 0000	38h	DCRD	xx11 1111
2Ch	PAINTSTA	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] *Bit 0 (MS0) of RE (CPUCON) is reloaded from "****INIM****" bit of Code Option when the MCU resets.*

7.2 Oscillator System

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

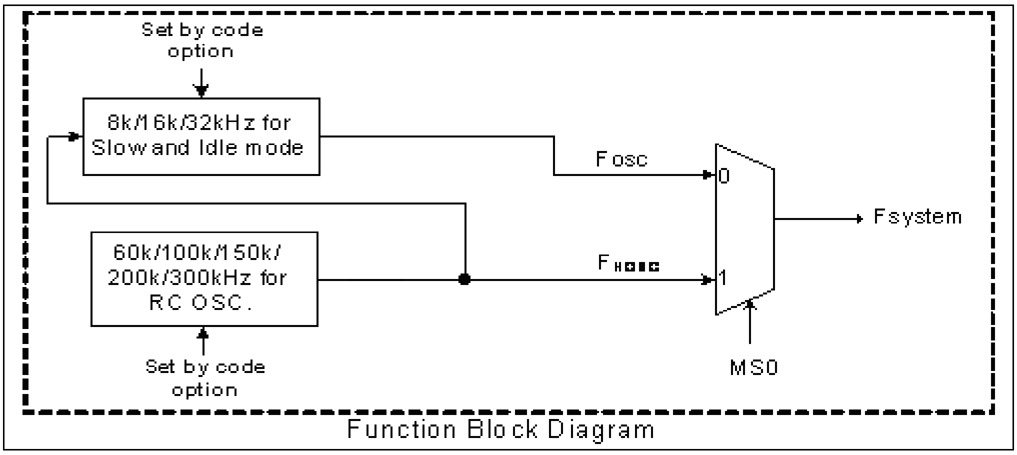


Figure 7-3 Oscillator System Function Block Diagram

The **MS0** bit (mode select bit) of **CPUCON** register (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 60kHz/100kHz/150kHz/200kHz/300kHz RC Oscillator

- 60kHz / 100kHz/ 150kHz/ 200kHz/ 300kHz Internal RC oscillator**:**
  Select the internal RC oscillator frequency by Code Option.

7.3 MCU Operation Mode

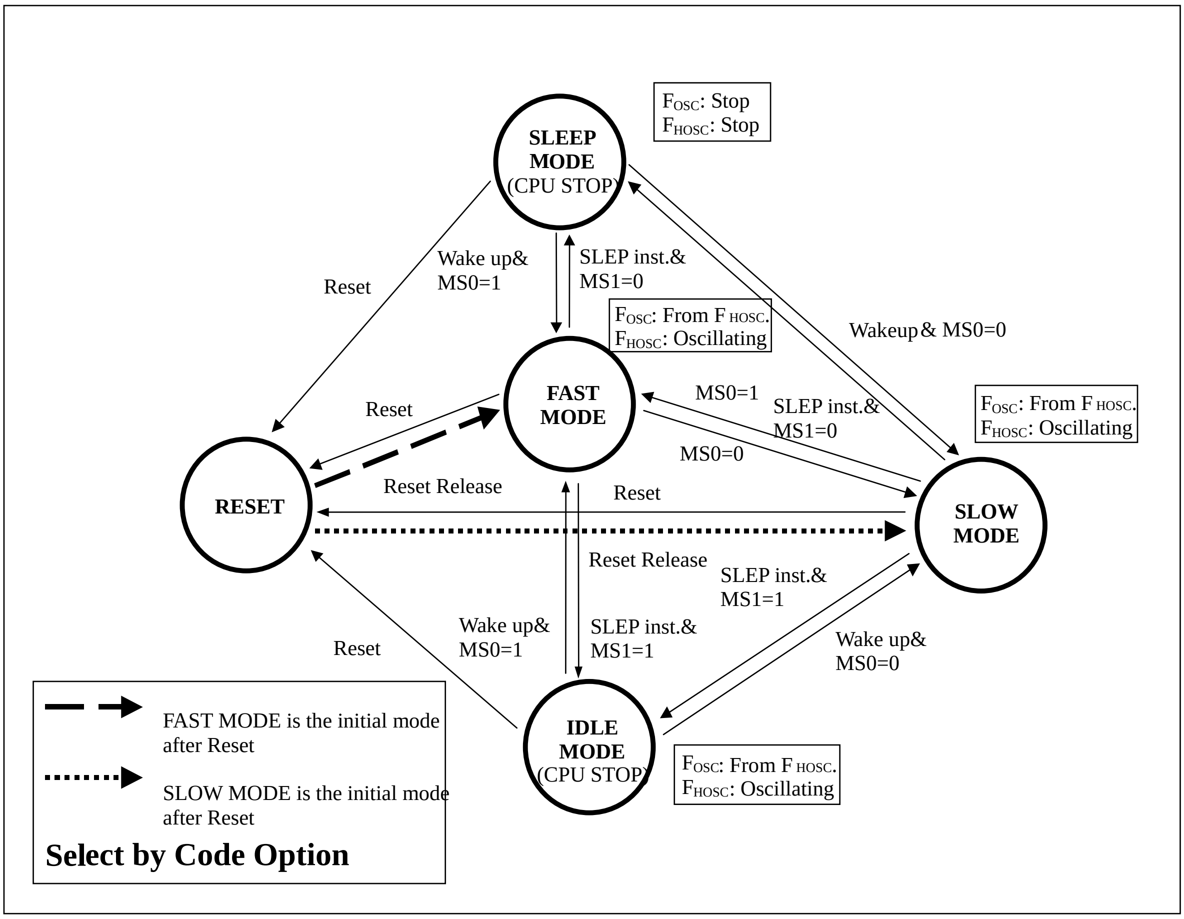


Figure 7-4 MCU Operation Block Diagram

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

Mode / Device	Sleep	Idle	Slow	Fast
Osc.	×	✓	✓	✓
Fsystem	×	×	From F_{OSC}	From F_{HOSC}
Timers 0~2	×	×	✓	✓
INT	×*	×*	✓	✓
I/O wake up	✓	✓	×	×
Timer 1 wake up	×	✓	×	×

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

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

7.3.1 Slow, Fast, Sleep, and Idle Mode Operation

- **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 the MS0 bit of the CPUCON register is set to “0,” the MCU will enter into Slow Mode and the corresponding system clock is at 8k/ 16k/ 32kHz (from F_{HOSC}). The Slow mode feature allows performance of all system operations at reduced power consumption.

NOTE

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

- **Fast Mode:**

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

- **Idle Mode:**

When the MS1 bit of the CPUCON register is set to “1” and the “SLEP” instruction is executed, the MCU will enter into Idle Mode. The Idle Mode suspends all system operations except for the 8k/ 16k/ 32kHz (from F_{HOSC}) and RC oscillator. It retains the internal status under low power consumption without stopping the clock function.

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

NOTE

All registers remain unchanged during Sleep Mode.

Sleep Mode:

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

NOTE

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

- **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 8k/ 16k/ 32kHz (from F_{HOSC}) keeps on running. The MCU is waken by input port (Port A) or Timer1, then returns to Fast Mode or Slow Mode (as determined by MS0 bit of the 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 Port A.7 ~ Port A.0

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

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

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

0: Disable Timer 1 wake-up function

1: Enable Timer 1 wake-up function.

7.4 Interrupts

When interrupt occurs, the GLINT bit of the CPUCON register is reset to **"0".** It disables all interrupts, including Levels 1 ~ 5. Setting this bit to **"1"** will enable all un-masked 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: Disable all interrupts, including Level 1 ~ Level 5

1: Enable all un-masked interrupts

- **Interrupt Vector**

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

- **Code Example:**

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

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

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

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

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

- **PAINTSTA (R2Ch): Port A.7 ~ Port A.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): Port A.7 ~ Port A.0 Interrupt status

Set to "**1**" when a pin falling edge is detected.

Clear to "**0**" by software.

- **PAINTEN (R2Bh): Port A.7 ~ Port A.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): Port A.7 ~ Port A.0 Interrupt control bits

0: Disable Interrupt function

1: Enable Interrupt function

- **Code Example:**

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

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

7.4.3.1 Timer 0 Interrupt

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

- **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
0	1	0	0	–	TMR2IE	TMR1IE	**TMR0IE**

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

0: Disable Timer 0 interrupt function

1: Enable Timer 0 interrupt function

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

Set to "**1**" when Timer 0 counters underflows.

Clear to "**0**" by software.

7.4.3.2 Timer 1 Interrupt

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

- **TMR1IEBit of INTCON (R21h) Timer Interrupt Control Register**

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

Bit 1 (TMR1IE): Control bit of Timer 1 interrupt

0: Disable Timer 1 interrupt function

1: Enable Timer 1 interrupt function

- **TMR1I Bit of 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): Status bit of Timer 1 interrupt

Set to "**1**" when Timer 1 counter underflows.

Clear to "**0**" by software.

7.4.3.3 Timer 2 Interrupt

Timer 2 is an 8-bit timer for time counting. When the counting value of Timer 2 underflows, an interrupt occurs and the TRL2 value will be reloaded to the timer.

- **TMR2IE Bit of INTCON (R21h) Timer Interrupt Control Register**

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

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

0: Disable Timer 2 interrupt function

1: Enable Timer 2 interrupt function

- **TMR2I Bit of 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): Status bit of Timer 2 interrupt
Set to "**1**" when Timer 2 Counter underflows.
Clear to "**0**" by software.

7.4.3.4 Code Example:

```
; === Timer-0,1,2 Interrupt
TIMERINT:
     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 = 24K Words	
Address.	**Description**
0000h | 000Bh	Interrupt Vector (12 words)
000Ch | 000Fh	Code Option (4 words)
0010h | 001Fh	Test Program (16 words)
0020h | 5FFFh	Program or Fixed data region

7.6 RAM Map for Special and Control Registers

RAM Size: 83 Bytes + 12 Banks × 128 Bytes = 1619 Bytes

7.6.1 Special and Control Registers

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

Addr.	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	INDF0	R/W							
		Indirect Addressing Pointer 0							
1	FSR0	R/W							
		File Select Register 0 for INDF0 (R0)							
2	BSR	R	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 Pointer 1							
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/W	R/W	R/W	R/W	R/W	R/W	R/W
		Fixed 0	PC14	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/W							
		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	Port A	R/W	R/W	R/W	R/W	R	R	R	R
		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	Port B	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	Port C	-	-	-	-	-	-	R/W	R/W
		-	-	-	-	-	-	Port C.1	Port C.0
13	Port D	-	-	R/W	R/W	R/W	R/W	R/W	R/W
		-	-	Port D.5	Port D.4	Port D.3	Port D.2	Port D.1	Port D.0

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
		ALL	KE	R1EN	BitST	STB3	STB2	STB1	STB0
21	INTCON	R	R	R	R	-	R/W	R/W	R/W
		Fixed 0	Fixed 1	Fixed 0	Fixed 0	-	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							
		Timer 0 Auto-reload Register Low Byte							
25	TRL0H	R/W							
		Timer 0 Auto-reload Register High Byte							
26	TRL1	R/W							
		Timer 1 Auto-reload Register							
27	TR2WCON	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		WDTEN	T0ENMD	WDTPSR1	WDTPSR0	T2EN	T2CS	T2PSR1	T2PSR0
28	TRL2	R/W							
		Timer 2 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
		-	-	-	-	-	-	PC1PU	PC0PU
31	DCRC	-	-	-	-	-	-	R/W	R/W
		-	-	-	-	-	-	PC1DC	PC0DC
32	LCDCON	-	R/W	R/W	-	R/W	R/W	R/W	R/W
		-	BLANK	LCDON	-	LCR1	LCR0	LRVON	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
35	T0CL	R							
		Timer 0 Counting Value Low Byte Register							
36	T0CH	R							
		Timer 0 Counting Value High Byte Register							
37	PDCON	-	-	R/W	R/W	R/W	R/W	R/W	R/W
		-	-	PD5PU	PD4PU	PD3PU	PD2PU	PD1PU	PD0PU
38	DCRD	-	-	R/W	R/W	R/W	R/W	R/W	R/W
		-	-	PD5DC	PD4DC	PD3DC	PD2DC	PD1DC	PD0DC

7.6.2 Other Unbanked General RAM

Address	Unbanked
14h | 1Fh	General purpose RAM
39h | 7Fh	General purpose RAM

7.6.3 Banked General RAM

Address	Bank 0	Bank 1	Bank 2	……	Bank 11
80h | FFh	General Purpose RAM	General Purpose RAM	General Purpose RAM	……	General Purpose RAM

7.7 LCD RAM Map

- **1/10 Duty**

RAM Address LCDARL		COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7
		Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
SEG0	00H								
:	:								
SEG57	39H								

RAM Address LCDARL		COM8	COM9	-	-	-	-	-	-
		Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
SEG0	40H								
:	:								
SEG57	79H								

- **1/9 Duty**

RAM Address LCDARL		COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7
		Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
SEG0	00H								
:	:								
SEG57	39H								

RAM Address LCDARL		COM8	-	-	-	-	-	-	-
		Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
SEG0	40H								
:	:								
SEG57	79H								

- **1/8 Duty**

RAM Address LCDARL		COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7
		Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
SEG0	00H								
:	:								
SEG57	39H								

7.8 Special Function Registers

7.8.1 *ACC (R0Ah): Accumulator*

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

7.8.2 *POST_ID (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): **0:** Auto decrease FSR0

1: Auto increase FSR0

Bit 5 (FSR1_ID): **0:** Auto decrease FSR1

1: Auto increase 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 12 banks (Bank 0 ~ Bank B).

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

INDF0 (R00h) is not a physically implemented register.

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

BSR1 (R05h) is a bank register for INDF1. It cannot determine the working bank for the general register.

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

INDF1 (R03h) is not a physically implemented register.

Code Example 1:

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

INDF1 Linear Address Capabilities

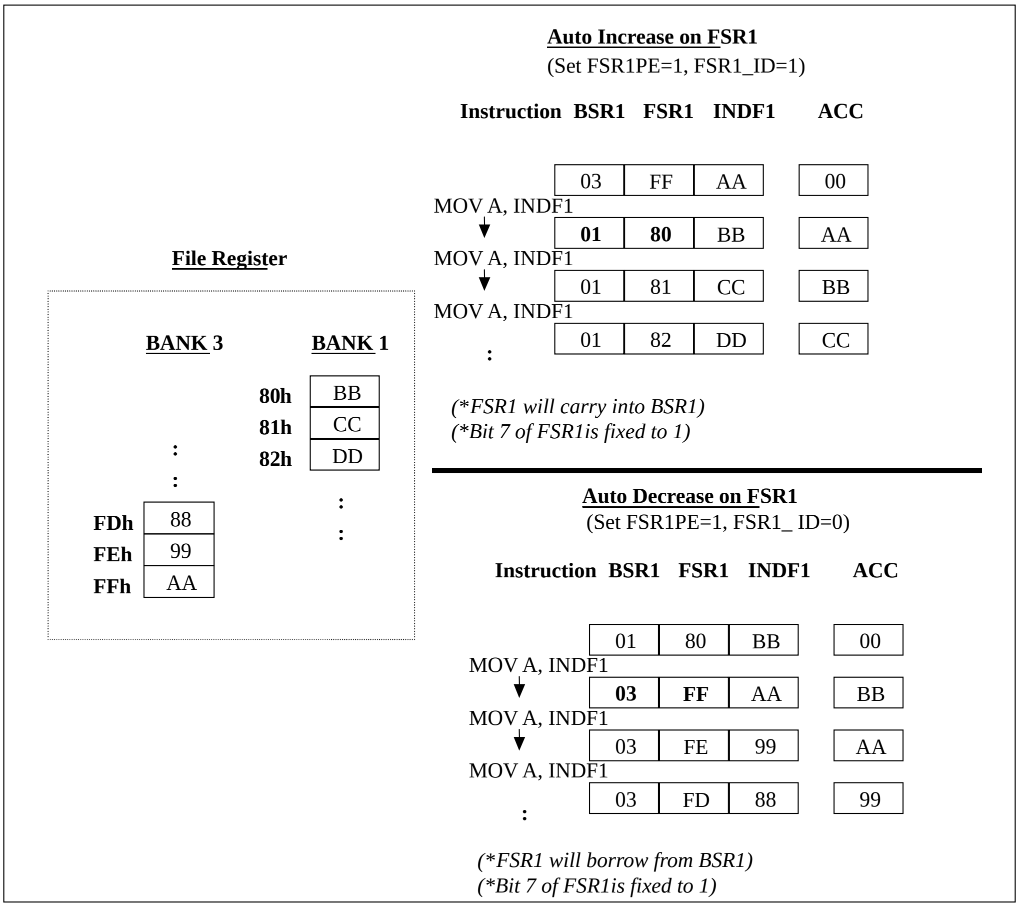


Figure 7-5 INDF1 Linear Address Capabilities Diagram

- **Code Example 2:**

```
;*************************************
;*   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 a total capacity of 32 levels. When stack overflows, it will replace the first stack level.

NOTE

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

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

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

The configuration structure can generate up to 24K×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 **“SJMP”** loads the low 13 bits of the PC (8K×16 ROM)

“LCALL” or **“LJMP”** 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
TABPTRM								TABPTRL							

Program ROM or Internal ROM address register.

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

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

- **Code Example:**

```
; *** Program ROM
        :
        :
     TBPTM  #(PROMTabB*2)/100H
     TBPTL  #PROMTabB*2
        :
        :
     TBRD   0,ACC                        ; no change
     TBRD   1,ACC                        ; auto-increase
     TBRD   2,ACC                        ; auto-decrease
        :
        :

; *** Program ROM data
PROMTabB:
     DB     0x00,0x01,0x02,0x03,0x04,0x05
     DB     0x10,0x11,0x12,0x13,0x14,0x15
     DB     0x20,0x21,0x22,0x23,0x24,0x25
```

7.8.8 Port A, Port B, Port C, Port D (R10h, R11h, R12h, R13h): General I/O Pin Registers

Port A (R10h) Port A.0 ~ 7 are general I/O pin registers

Port B (R11h) Port B.0 ~ 7 are general I/O pin registers

Port C (R12h) Port C.0 ~ 1 are general I/O pin registers

Port D (R13h) Port D.0 ~ 5 are general I/O pin 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
ALL	KE	R1EN	BitST	STB3	STB2	STB1	STB0

Bit 7 (ALL): Set All Strobe.

0: Bit Strobe

1: All Strobe

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

0: Disable Key input function (Port A.0~7 do not correspond with Key input in software scan mode)

1: Enable Key input function (Port A.0~7 correspond with Key input in software scan mode)

(The Key input control can include PA.7 set by Code Option)

Bit 5 (R1EN): R1 pull up resistor (small resistor) control bit for Port A.7 ~ Port A.0.

0: Disable R1 pull up resistor

1: Enable R1 pull up resistor

(The PA.7's small resistor can control by R1EN set by Code Option)

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.

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

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

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

Bits 7 ~ 0 (PA7PU ~ PA0PU): Enable PortA.0 ~ Port A.7 pull-up resistor bits

0: Disable Port A.0 ~ Port A.7 pull-up resistor

1: Enable Port A.0 ~ Port A.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

Bits 7 ~ 0 (WKEN7 ~ WKEN0): Wake-up enable control bits of Port A.7~Port A.0.

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

1: Enable Port A.7 ~ Port A.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 ~ Port A.0 interrupts bits

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

Cleared **("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 as output pin

1: Set as 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 ~ Port B.7 pull-up resistor control bits

0: Disable the pull-up resistor

1: Enable the 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 as output pin

1: Set as input pin

NOTE

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

- **Code Example:**

```
; *** Set Port B as 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
-	-	-	-	-	-	PC1PU	PC0PU

Bit 1 ~ Bit 0 (PC1PU ~ PC0PU): Port C.0 ~ Port C.1 pull-up resistor control bits

0: Disable the pull-up resistor

1: Enable the pull-up resistor

NOTE

This function is only available with 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
-	-	-	-	-	-	PC1DC	PC0DC

Bit 1 ~ Bit 0 (PC1DC ~ PC0DC): Port C.0 ~Port C.1 direction control bits

0: Set as output pin

1: Set as input pin

> **NOTE**
> *When Port C bit is set as input pin, a 5 µsec delay in reading Port C data must be provided. Otherwise, the read data will be inaccurate. See the Example below.*

- **Code Example:**

```
; *** Set Port C as input pins
     MOV    A,#0X03
     MOV    DCRC,A
     MOV    PCCON,A
Read_PC:
     JBS    PORTC,0,Read_PC
        Delay 5usec
     JBS    PORTC,0,Read_PC
     SJMP   Read_PC
```

7.8.19 PDCON (R37h): Port D Pull up Resistor Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	PD5PU	PD4PU	PD3PU	PD2PU	PD1PU	PD0PU

Bit 5 ~ Bit 0 (PD5PU ~ PD0PU): Port D.0 ~ Port D.5 pull-up resistor control bits

0: Disable the pull-up resistor

1: Enable the pull-up resistor

> **NOTE**
> *This function is only available with Port D selected as input pin.*

7.8.20 DCRD (R38h): Port D Direction Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	PD5DC	PD4DC	PD3DC	PD2DC	PD1DC	PD0DC

Bit 5 ~ Bit 0 (PD5DC ~ PD0DC): PortD.0 ~PortD.5 direction control bits

0: Set as output pin

1: Set as input pin

NOTE

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

- **Code Example:**

```
; *** Set Port D as input pins
     MOV    A,#0X3F
     MOV    DCRD,A
     MOV    PDCON,A
Read_PD:
     JBS    PORTD,0,Read_PD
        Delay 5usec
     JBS    PORTD,0,Read_PD
     SJMP   Read_PD
```

8 Peripheral

8.1 Timer 0 (16-Bit Timer with Event Counter Function)

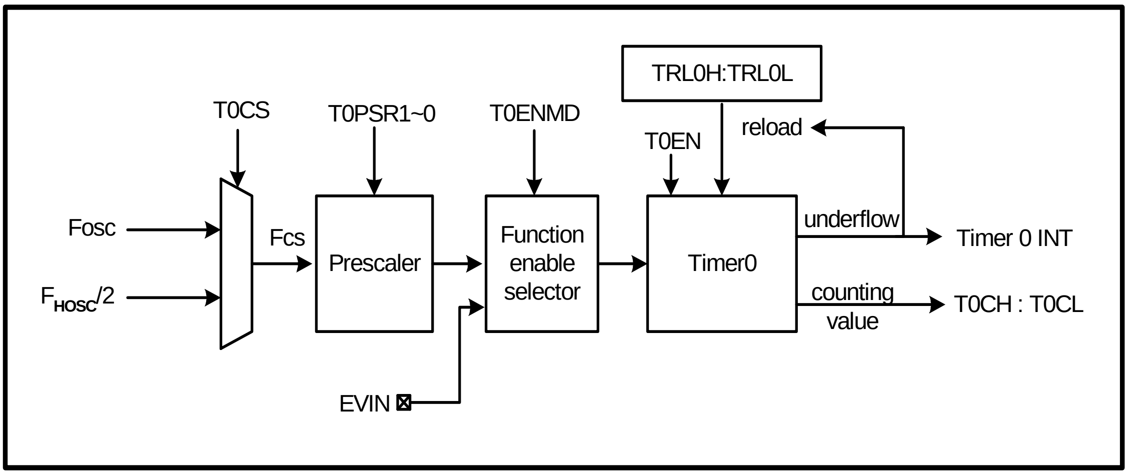


Figure 8-1a Timer 0 Function Block Diagram

- **Timer 0 Mode**

Timer 0 is a general-purpose 16-bit down counter used on applications that require time counting with interrupt. The clock source (FCS) is selectable from the oscillator clock (FOSC) or half of the system clock (FHOSC).

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

$$T = \frac{1}{F_{CS}} \times \mathrm{Pr}\,escaler \times (TRL0H : TRL0L + 1)$$

- **Event Counter Mode, EVIN (Port B.7) Pin**

The Event counter is a function that allows the 16-bit counter value to be decremented by one when an event occurs on the EVIN pin at every rising edge. In other words, the clock source of Timer 0 is from an external event (EVIN pin).

The EVIN pin can be configured into an event counter input function by setting the T0ENMD bit of the TR2WCON register (see next page) to **“1.”** The counter value of Timer 0 will be stored in T0CH:T0CL registers.

NOTE

If the program uses the event counter mode, Port B.7 will be fixed as input pin and Port B.7 cannot be controlled by the "Port B high nibble control bit" code option.

- **Event Counter Mode Example:**

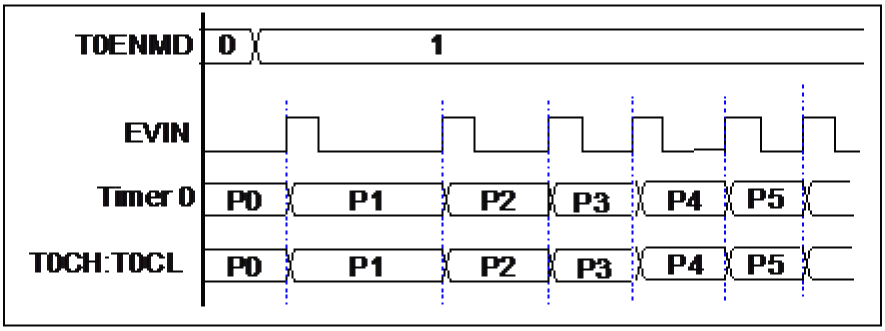


Figure 8-1b Event Counter Mode Example Timing Diagram

8.1.1 Timer 0 Registers

- **TRL0H:TRL0L (R25h, R24h): Timer 0 Reload Registers**

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

- **T0CH:T0CL (R36H, R35H): Timer 0 Counter Value Register**

Used to store the value compared with Timer 0 register.

- **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

■ 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	**T0ENMD**	WDTPSR1	WDTPSR0	T2EN	T2CS	T2PSR1	T2PSR0

Bit 6 (T0ENMD): Timer 0/ Event counter select bit

0: Timer 0 mode

1: Event counter mode

■ 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
0	1	0	0	–	TMR2IE	TMR1IE	**TMR0IE**

Bit 0 (TMR0IE): Timer 0 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, and Cleared to "**0**" by software.

- **Code Example:**

```
; === Timer 0 interrupt
TIMERINT:
     PUSH
     JBC    INTSTA,TMR0I,Q_Time
     BC     INTSTA,TMR0I
     BTG    PORT A,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)
     BC     TR2TWCON,T0ENMD          ; 0=>Timer 0 mode
     BS     TR01CON,T0EN             ; Timer 0 enable
     BS     INTCON,TMR0IE            ; Timer 0 interrupt enable
     BC     INTSTA,TMR0I             ; Clear Timer 0 interrupt status
     BS     CPUCON,GLINT             ; Enable global interrupt
TimeLoop:
     SJMP   TimeLoop
```

```
; === Event mode interrupt
TIMERINT:
     PUSH
     JBC    INTSTA,TMR0I,Q_Time
     BC     INTSTA,TMR0I
     BTG    PORT A,7
Q_Time:
     POP
     RETI
; === Event counter set
TR0_Event:
      :
     PA.7 setting output pin
       :
     MOV    A,#0XFF                  ; Switch 256 times reload
     MOV    TRL0L,A
     CLR    TRL0H                    ; Count start 00FFh
     BS     TR2TWCON,T0ENMD          ; 1=>Event counter mode
     BS     INTCON,TMR0IE            ; Enable Timer 0 interrupt
     BS     CPUCON,GLINT             ; Enable global interrupt
EventWait:
     SJMP   EventWait
```

8.2 Timer 1 (8 Bits)

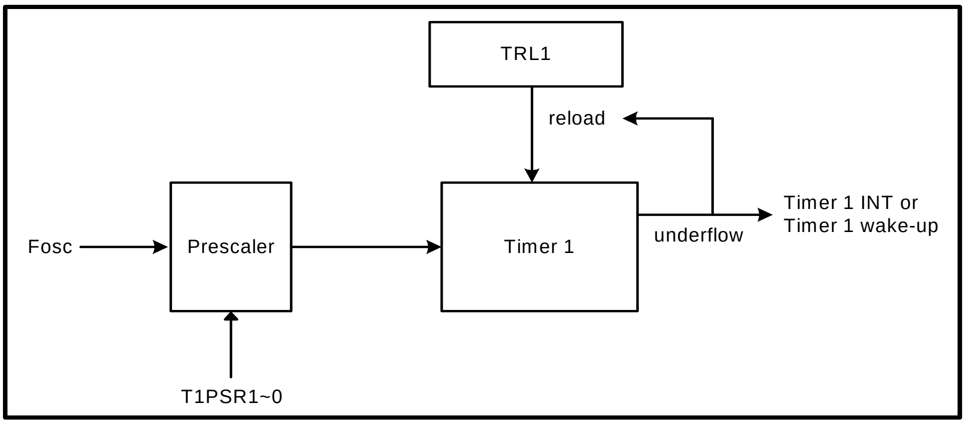


Figure 8-2 Timer 1 Function Block Diagram

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 (FOSC).

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

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

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

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, the TRL1 register value will be automatically reloaded into the 8-bit 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
0	1	0	0	–	TMR2IE	**TMR1IE**	TMR0IE

Bit 1 (TMR1IE): Control bit of Timer 1 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, and Cleared to “**0**” by software.

- **Code Example:**

```
; === Timer 1 interrupt
TIMERINT:
     PUSH
     JBC    INTSTA,TMR1I,Q_Time
     BC     INTSTA,TMR1I
     BTG    Port A,7
Q_Time:
     POP
     RETI
; === Timer1 = 32K / [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                       ; 0.512sec=[256x(63+1)]/32K
     BS     TR01CON,T1EN                 ; Timer 1 enable
     BS     INTCON,TMR1IE                ; Timer 1 interrupt enable
     BC     INTSTA,TMR1I                 ; Clear Timer 1 interrupt status
     BS     CPUCON,GLINT                 ; Enable global interrupt
     BS     CPUCON,MS1                   ; Idle mode
T1Wloop:
     SLEP
     NOP
      :
     SJMP   T1Wloop
```

8.3 Timer 2 (8 Bits)

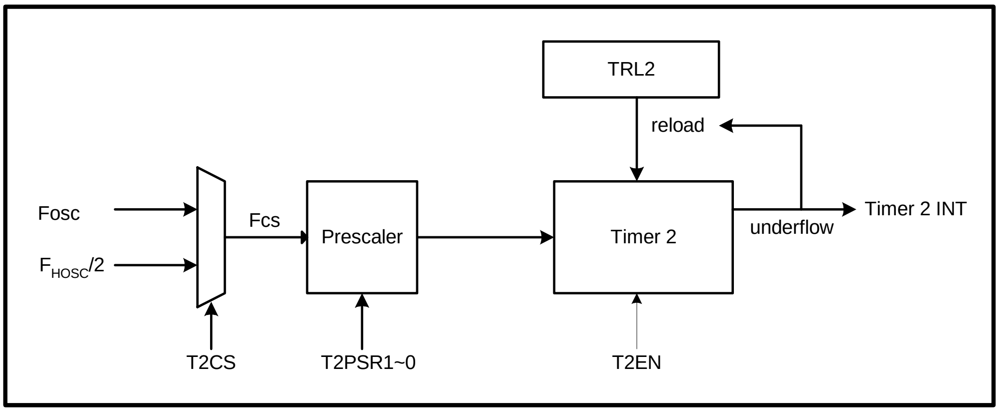


Figure 8-3 Timer 2 Function Block Diagram

Timer 2 is a general-purpose 8-bit down counter used on applications that require a time counter with interrupt. The clock source (FCS) may be selected from the oscillator clock (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 is decreased by one (counting down) according to the timer clock source frequency. When counter value underflows, the timer interrupt is triggered (if Timer 2 interrupt is enabled).

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

8.3.1 Timer 2 Registers

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

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

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

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WDTEN	T0ENMD	WDTPSR1	WDTPSR0	**T2EN**	**T2CS**	**T2PSR1**	**T2PSR0**

Bit 3 (T2EN): Timer 2 Enable Control Bits

0: Disable Timer 2 (stop counting)

1: Enable Timer 2

Bit 2 (T2CS): Timer 2 Clock Source Select Bit

0: Clock source is from FOSC

1: Clock source is from FHOSC/2

Bit 1 ~ Bit 0 (T2PSR1 ~ T2PSR0): Timer 2 Prescaler Select Bits

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

■ CPUCON (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-masked interrupts

■ INTCON (R21h): Timer Interrupt Control Register

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

Bit 2 (TMR2IE): Control bit of Timer 2 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 Timer 2 interrupt occurs, this bit will be set, and Cleared to "**0**" by software.

- **Code Example:**

```
; === Timer 2 interrupt
TIMERINT:
     PUSH
     JBC    INTSTA,TMR2I,Q_Time
     BC     INTSTA,TMR2I
     BTG    Port A,7
Q_Time:
     POP
     RETI
; === Timer2 = (1/32K) X [4 x (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              ;32ms=[4x(255+1)]/32K
     BS     TR2CON,T2EN         ;Timer 2 Enable
     BS     INTCON,TMR2IE       ;Timer 2 Interrupt Enable
     BC     INTSTA,TMR2I        ;Clear Timer 2 Interrupt Status
TMR2Loop:
     SJMP   TMR2Loop
```

8.4 Watchdog Timer (WDT)

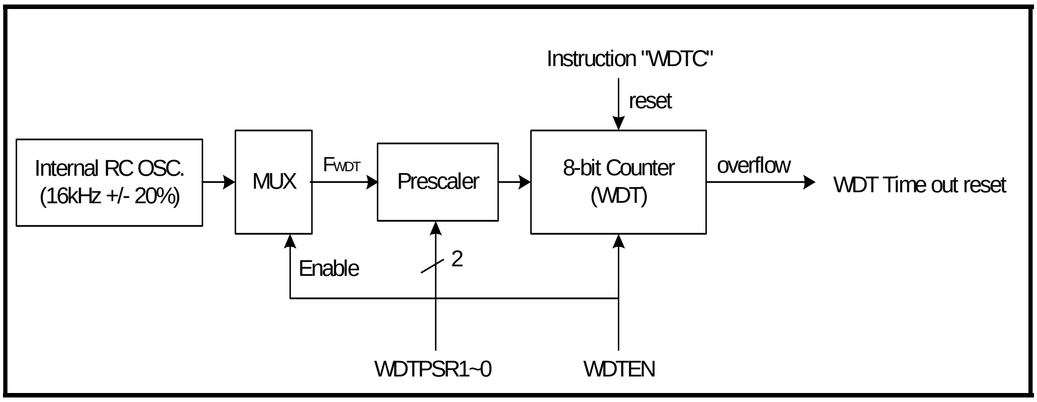


Figure 8-4 Watchdog Timer Functional Block Diagram

The watchdog timer (WDT) clock source comes from an on-chip RC oscillator (16kHz ± 20%). Therefore the WDT will keep on running even after the oscillator has been turned off.

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

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

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

8.4.1 Watchdog Timer (WDT) Registers

■ **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	T0ENMD	**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

- Four pins key input / output (Port A.7 ~ 0 and Port D.5 ~ 0) can have a maximum of 48 keys matrix.
- Interrupt available when Port A input falling edge detected at wait key in.
- Wake-up available when Port A input falling edge detected at wait key in.

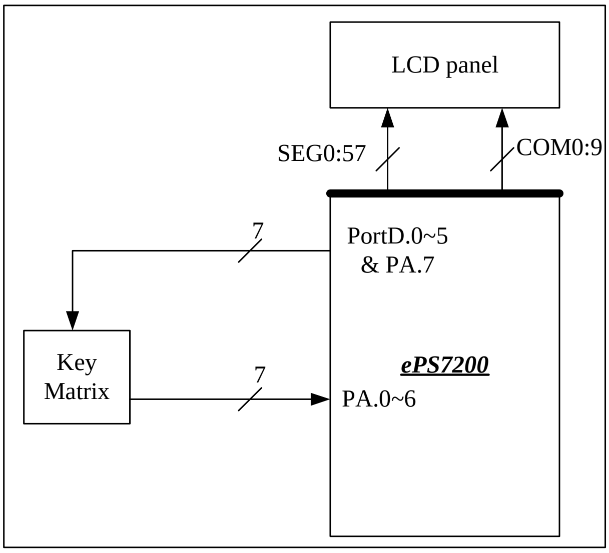


Figure 8-5 Key Function Block Diagram

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. 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 process**:**

1. Pull up the input port by lowest resistance (both R1 and R2 enabled). Capacitance is charged quickly.
2. Enable the key input control bit (KE=1).
3. Set the output pin to low signal
4. Pull up the input port by highest resistance (only R2 is enabled)
5. Read the key
6. Disable the key input control bit (KE=0).
7. Disable the pulled-up resistance

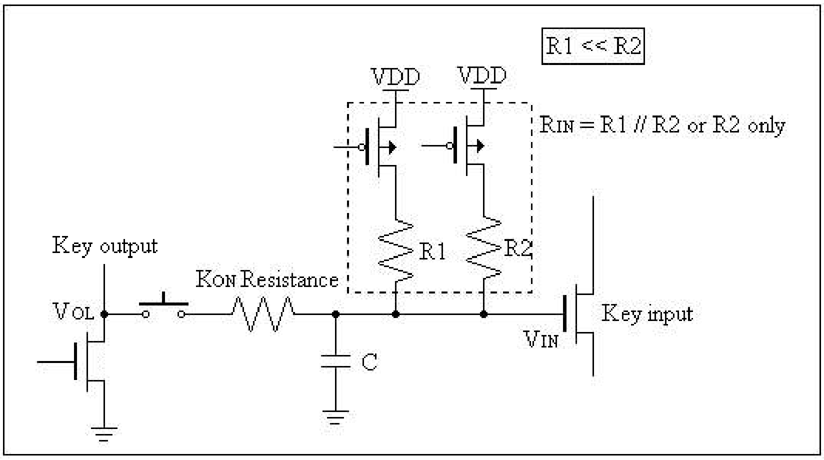


Figure 8-6 Key Circuit Diagram

8.5.1 Key Functions

KE	R1EN	PA7PU ~ PA0PU	Total Pull-up Resistor	Port A.0 ~ 7	Note
0	×	×	Floating	High-Z	-
1	0	0	**Floating**	**Floating**	**Prohibited**
	0	1	R2	PA.0~.7	-[1]
	1	0	R1	PA.0~.7	-[1]
	1	1	R1 // R2 [2]	PA.0~.7	-[1]

× **:** Don't care

[1] *For the table: the Port A.7 can be controlled by KE and small resistor can be controlled by R1EN set by Port A.7 control's Code Option.*

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

8.5.2 Key Strobe

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

8.5.2.1 Waiting Key in mode

When in waiting key-in mode, PAINT or PAWAKE for Port A must be enabled and Key output must set to output low. During key in, the wake-up and interrupt will occur if any of the falling edge of the key input pins (Port A) is detected.

8.5.2.2 Software Key Scan

Software key scan is used to determine "which key is pressed." Refer to the table below (Key matrix ➔ Key input is Port A.0~6, Key output is Port A. 7 and Port D.0~5) in determining which Key output pins must be output pin with low voltage.

Key Intput						KEY Output										
BifST	KE	R1EN	PA5PU~ PA0PU	Total Pull-up Resistor	PortA. 6 ~ 0	PD5PU~ PD0PU	PA7PU	DCRD	DCRA	PortD						PortA
										Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit7
0	0	X	X	Floating	High-Z	1	1	0xFF	0xFF	1	1	1	1	1	1	1
	1	0	1	R2	PA.0~ 6	1	1	0xFE	0xFF	0	1	1	1	1	1	1
								0xFD	0xFF	1	0	1	1	1	1	1
		1	0	R1	PA.0~ 6			0xFB	0xFF	1	1	0	1	1	1	1
								0xF7	0xFF	1	1	1	0	1	1	1
		1	1	R1//R2	PA.0~ 6			0xEF	0xFF	1	1	1	1	0	1	1
								0xDF	0xFF	1	1	1	1	1	0	1
								0xFF	0x7F	1	1	1	1	1	1	0

x: Don't care.

8.5.3 Input/Output Key Registers

■ **Port A (R10h): Port A Register**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0

Bit 7 ~ Bit 0: The input structure and two-stage pull-up resistor are controlled together by PA7PU ~ PA0PU bits of the PACON register and R1EN, KE bits of the STBCON register (see below).

■ **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): Port A.0~Port A.7 direction control bits

0: Set as output pin

1: Set as input pin

■ **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 ~ Bit 0 (PA7PU ~ PA0PU): Pull-up resistor (R2 large resistor) control bits

0: Disable Port A.0 ~ Port A.7 pull-up resistor

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

NOTE

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

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

Set to "**1**" when pin falling edge is detected, and

Cleared to "**0**" by software.

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

■ STBCON (R20h): Strobe Output Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ALL	KE	R1EN	BitST	STB3	STB2	STB1	STB0

Bit 7 (ALL): Set All Strobe.

0: Bit Strobe

1: All Strobe

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

0: Disable Key input function (Port A.0~7 do NOT correspond with Key input in software scan mode)

1: Enable Key input function (Port A.0~7 corresponds with Key input in software scan mode)

(The Key input control can include PA.7 set by code option)

Bit 5 (R1EN): R1 pull-up resistor (small resistor) control bit for Port A.7 ~ Port A.0

0: Disable R1 pull-up resistor

1: Enable R1 pull-up resistor

(The PA.7 small resistor can control by R1EN set by code option)

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 specified as STB3 ~ 0 defined.

Bit 3 ~ Bit 0 (STB0 ~ STB3): Strobe output selector bits. These are 16-to-1 multiplexing selector of the key strobe pins.

■ Code Example:

```
; Key Matrix (Port A and Ground):
;  ===  Sleep mode
PAIN_SR:
      :
; --- Port A 0~7 input pins
     MOV    A,#0XFF
     MOV    DCRA,A
; --- Port A 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 Port A data
INPTINT:
     PUSH
     MOVRP  A,PAINTSTA
     MOV    Key_No,A
     CLR    PAINTSTA
     POP
     RETI
```

NOTE

The PA.0~7's code option must be set to "PA.0~7 and GPIO or none" by setting the bit when the Key matrix combination is PA[7:0] and GPIO or Ground.

8.6 LCD Driver

The ePS7200 provides directly driven LCD. It supports multiplexed drive for 58SEGs × 10COMs which allows the use of pads as an LCD driver pin or as key input port. The available LCD RAM corresponds directly with the LCD Pixel. The LCD frame rate is as follows.

Duty	LCD Frame Rate
1/8	Approx. 11.5ms (32k /46/8 Hz)
1/9	Approx. 11.81ms (32k /42/9 Hz)
1/10	Approx. 11.88ms (32k /38/10 Hz)

This embedded LCD driver generates waveforms to drive the display.

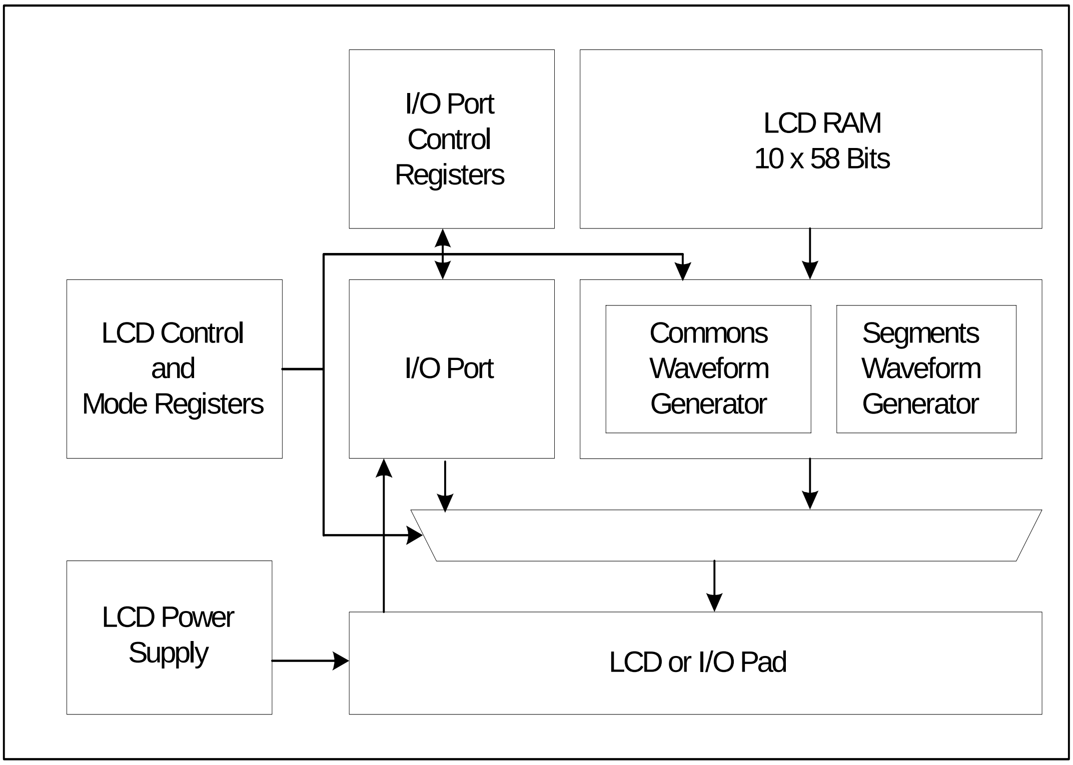

I/O Port Control Registers
LCD RAM 10 x 58 Bits
LCD Control and Mode Registers
I/O Port
Commons Waveform Generator
Segments Waveform Generator
LCD Power Supply
LCD or I/O Pad

Figure 8-7a LCD Driver Function Block Diagram

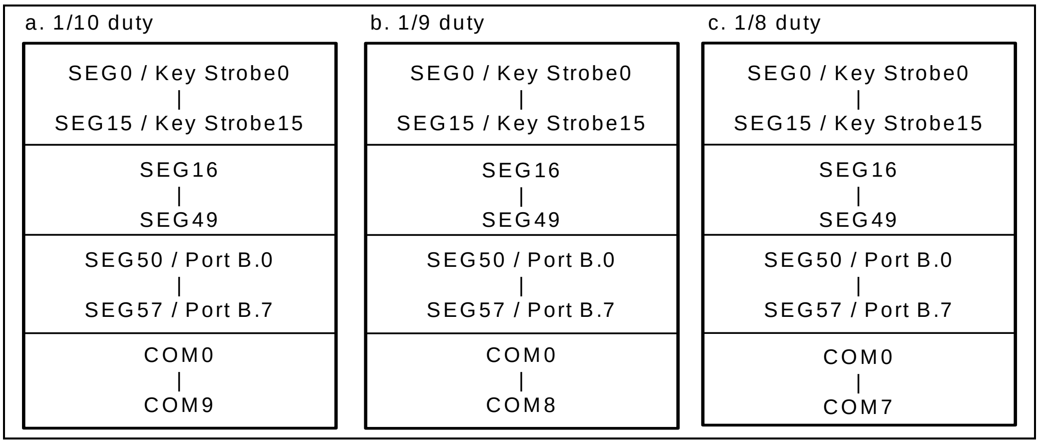


Figure 8-7b LCD Pin Configuration

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
0	BLANK	LCDCON	–	LCR1	LCR0	LRVON	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 and 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	1K

Bit 1 (LRVON): Regulator Voltage control bit

0: Disable

1: Enable

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

0: Disable

1: Enable

■ LCDARL (R09h): LCD RAM Column Address Register

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

See next Section 8.6.2, *LCD RAM Map* for details.

■ **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 the figure below).

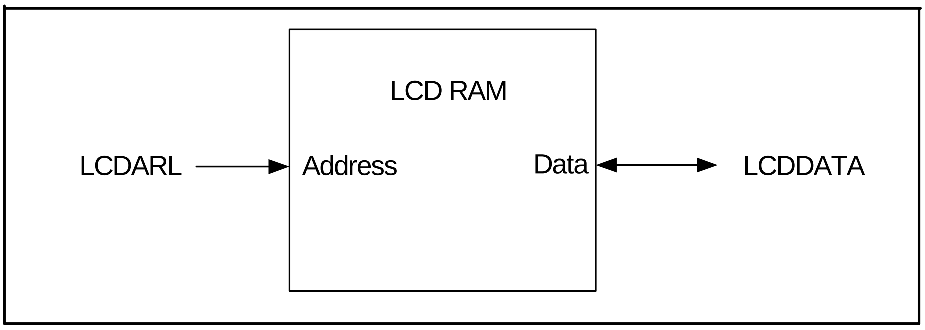


Figure 8-8 LCDDATA Register Access through LCD RAM

■ 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 by setting the POST_ID register.

Bit 6 (LCD_ID): Set to "**1**" to auto-increase the LCDARL register.

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

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

- **Code Example:**

```
; === LCD Setting
L_Initial:
; --- LCD Off, Normal Display Mode, Charge-Pump rate=2K
     MOV    A,#00001011B
     MOV    LCDCON,A
     SCALL  DspRAMdot
; --- LCD turn-on
     BS     LCDCON,LCDON
     LCALL  Delay1sec
      :
DspLoop:
; --- LCD Blanking
     BS     LCDCON,BLANK
     LCALL  Delay1sec
; --- Normal display
     BC     LCDCON,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  #0x14
; === Write LCD RAM is dot matrix
WrLRAMd:
     MOV    A,#0XAA
     MOV    LCDDATA,A
     MOV    A,#0X55
     MOV    LCDDATA,A
     JDNZ   TABPTRH, WrLRAMd
     CLR    LCDARL
     RET
```

8.6.2 LCD RAM Map

- **1/10 Duty**

RAM Address LCDARL		COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7
		Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
SEG0	**00H**								
:	**:**								
SEG57	**39H**								

RAM Address LCDARL		COM8	COM9	-	-	-	-	-	-
		Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
SEG0	**40H**								
:	**:**								
SEG57	**79H**								

- **1/9 Duty**

RAM Address LCDARL		COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7
		Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
SEG0	**00H**								
:	**:**								
SEG57	**39H**								

RAM Address LCDARL		COM8	-	-	-	-	-	-	-
		Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
SEG0	**40H**								
:	**:**								
SEG57	**79H**								

- **1/8 Duty**

RAM Address LCDARL		COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7
		Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
SEG0	**00H**								
:	**:**								
SEG57	**39H**								

8.6.3 LCD Driving Method Circuit

- **1/4 Bias**

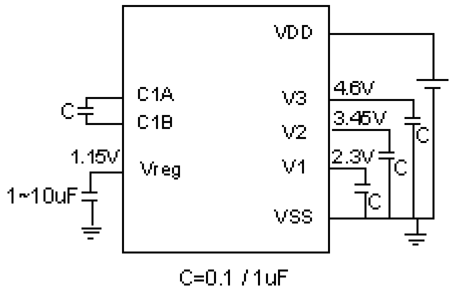


Figure 8-9 LCD Driving Method Circuit for 1/4 Bias

8.6.4 LCD COM Waveforms

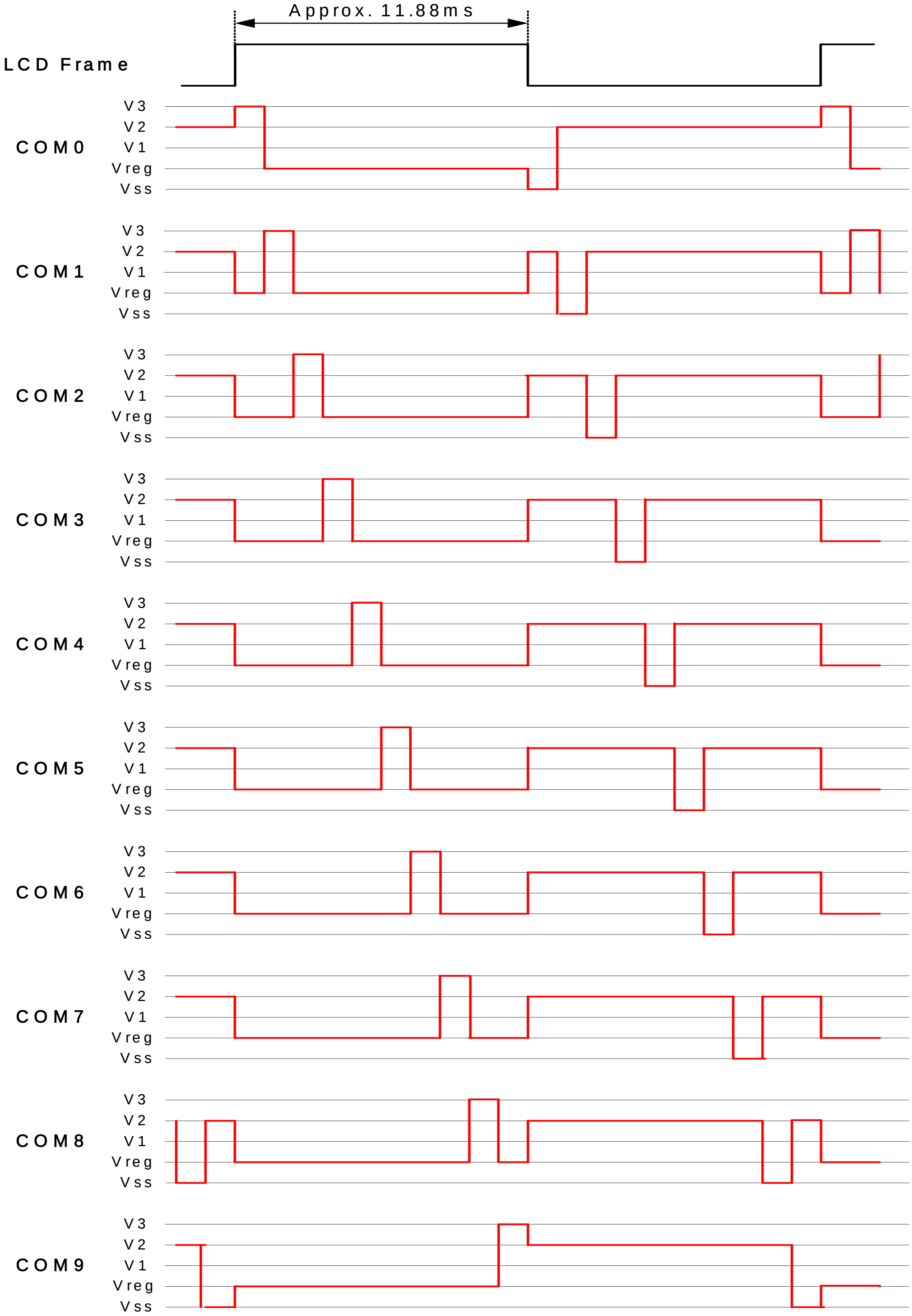


Figure 8-10 LCD COM Waveform for 1/10 Duty and 1/4 Bias

8.6.5 LCD COM and SEG Waveforms

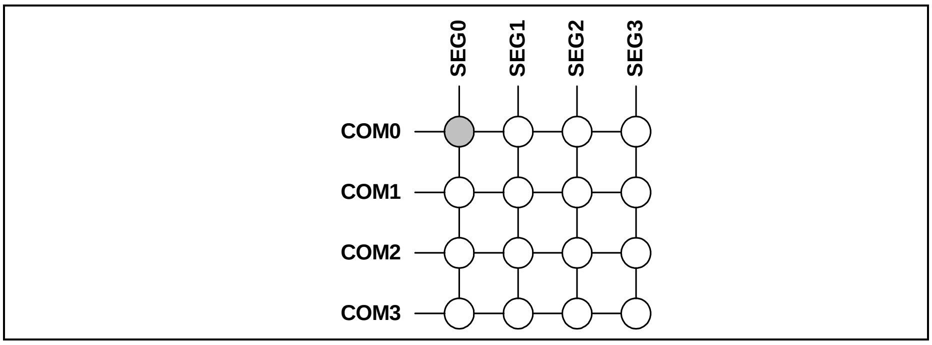


Figure 8-11a LCD COM and SEG Waveform Matrix

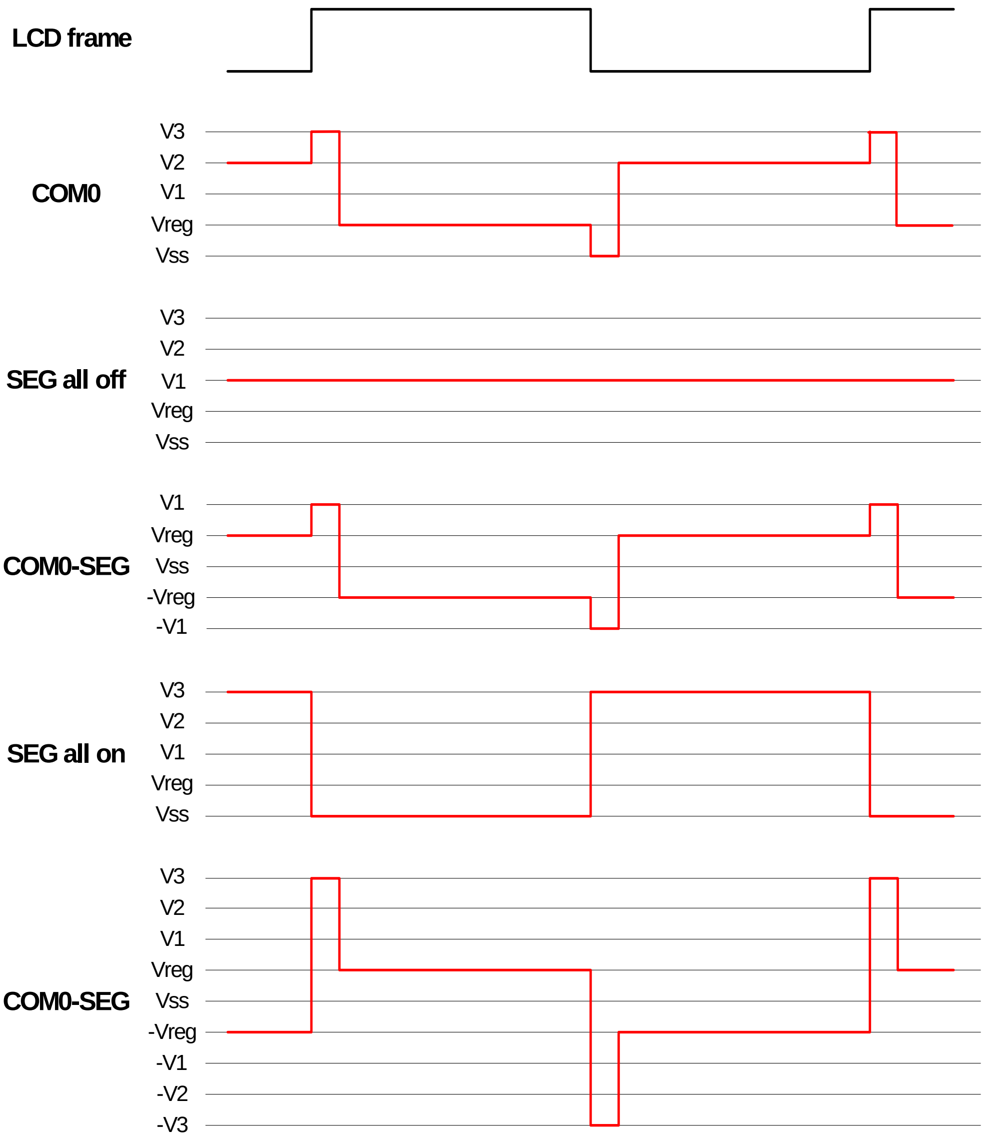


Figure 8-11b LCD COM and SEG Waveform for 1/10 Duty and 1/4 Bias

9 Electrical Characteristics

■ Absolute Maximum Ratings

Items	Sym.	Condition	Limits	Unit
Supply voltage	VDD	-	–0.3 to +2.4	V
Input voltage (general input port)	VIN	-	–0.5 to VDD +0.5	V
Operating temperature range	TOPR	-	–10 to +70	°C
Storage temperature range	TSTR	-	–55 to +125	°C

■ Recommended Operating Conditions

Items	Sym.	Condition	Limits	Unit
Supply voltage	VDD	-	1.15 to 2.2	V
Input voltage	VIH	-	VDD x 0.9 to VDD	V
	VIL	-	0 to VDD x 0.1	V
Operating temperature	TOPR	-	-10 to +70	°C

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

Parameter	Sym.	Condition		Min.	Typ.	Max.	Unit
Clock	F_{HOSC}	Main-clock frequency	RC OSC	48	60	72	kHz
			RC OSC	80	100	120	
			RC OSC	120	150	180	
			RC OSC	160	200	240	
			RC OSC	240	300	360	
	Fosc	Sub-clock frequency	From FHOSC	5.6	8	10.4	kHz
			From FHOSC	11.2	16	20.8	
			From FHOSC	22.4	32	41.6	
Supply Current	Idd1	Sleep mode	VDD=1.5V, no load	–	–	1	µA
	Idd2	Idle mode	VDD=1.5V, Fosc=8~32kHz, LCD enable, no load	–	5	8	
	Idd3	Slow mode	VDD=1.5V, Fosc=8kHz, LCD enable, no load	–	5.5	8.8	
	Idd4		VDD=1.5V, Fosc=16kHz, LCD enable, no load	–	6	9.6	
	Idd5		VDD=1.5V, Fosc=32kHz, LCD enable, no load	–	6.5	10.4	
	Idd6	Fast mode	VDD=1.5V, FHOSC =60kHz, LCD enable, no load	–	8	12.8	
	Idd7		VDD=1.5V, FHOSC =100kHz, LCD enable, no load	–	10	16	
	Idd8		VDD=1.5V, FHOSC =150kHz, LCD enable,no load	–	13	20.8	
	Idd9		VDD=1.5V, FHOSC =200kHz, LCD enable, no load	–	15	24	
	Idd10		VDD=1.5V, FHOSC =300kHz, LCD enable, no load	–	20	32	

(Continuation)

Parameter	Sym.	Condition		Min.	Typ.	Max.	Unit
Input Voltage	VIH1	PA [0:7], PB [0:7], PC[0:1] , PD[0:5] (as general input port)		0.7xVDD	–	VDD	V
	VIL1			0	–	0.3xVDD	
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	260	400	720	KΩ
Small Pull-up Resistance	RPU6	RSTB	Vin=1V	10	30	60	KΩ
Large Pull-down Resistance	RPD1	TEST	Vin=VDD	250	500	800	KΩ
Small Pull-down Resistance	RPD2	TEST	Vin=0.5V	3	6	12	KΩ
Output Current	IOH1	PA[0:7], PB[0:7], PC[0:1], PD[0:5] (as general output port)	VDD=1.5V, VOH=1.2V, LCD enabled	-0.7	-0.9	-1.4	mA
	IOL1		VDD=1.5V, VOL=0.2V, LCD enabled	0.7	0.9	1.3	
Large Pull-up Resistance	RPU1	PA[0:7]	Key high resistance, pulled up by R2, LCD enabled, Vin2=0.5V	180	280	430	KΩ
	RPU3	PB[0:7], PC[0:1], PD[0:5]	Vin=0.5V, LCD enabled	150	250	400	
Small Pull-up Resistance	RPU2	PA[0:7]	Key high resistance, pulled up by R2//R1, LCD enabled, Vin2=0 V	12	18	26	KΩ
	RPU4	PA[7]	Vin=1V, LCD enabled	40	60	95	
Data retention voltage	Vret			1.0	–	–	V
Power on reset voltage	Vpor			1.0	1.05	1.1	
LCD Driver							
LCD Voltage	Vreg	Regulator voltage		1.15-2.1%	1.15	1.15+2.1%	V
	V1	LCD no load		–	2xVreg	–	
	V2			–	3xVreg	–	
	V3			V3-0.1	4xVreg	V3+0.1	
LCD Display Output ON-resistance	ROC	Com[0:9]	VOH=V3 ± 0.2V	0.35	0.4	0.5	KΩ
			VOM=V2 ± 0.2V	0.55	0.65	0.75	
			VOM=Vreg ± 0.2V	0.35	0.40	0.45	
			VOL=0.2V	0.25	0.3	0.35	
	ROS	Seg[0:57]	VOH=V3 ± 0.2V	0.35	0.4	0.5	KΩ
			VOM=V1 ± 0.2V	0.45	0.6	0.85	
			VOL=0.2V	0.25	0.3	0.35	
Strobe Output ON-resistance	ROP	Seg[0:15] (as key strobe)	V=VDD-0.2V	145	200	280	KΩ
	RON		V=0.2V	1	1.2	1.5	

10 Application Circuits

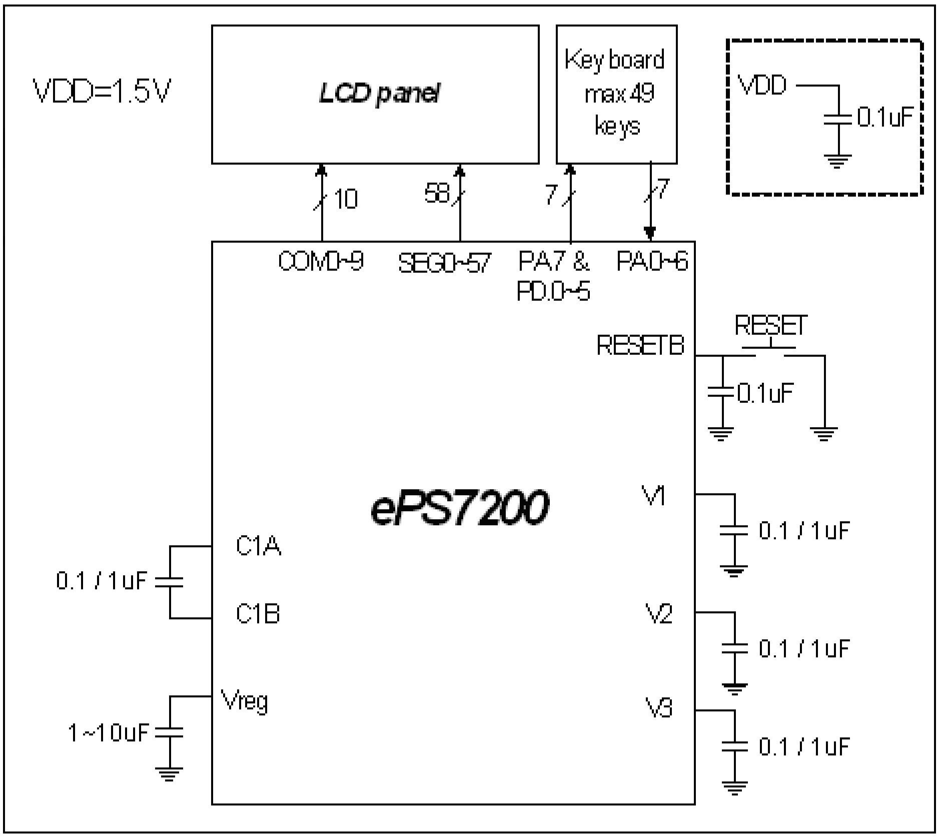


Figure 10-1 Application Circuit Diagram

11 Instruction Set

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

Type	Binary Instruction	Mnemonic	Operation	Status Affected	Cycle
System Control	0000 0000 0000 0000	NOP	No operation	None	1
	0000 0000 0000 0001	WDTC	WDT ← 0; /TO←1; /PD←1	None	1
	0000 0000 0000 0010	SLEP	Enter IDLE MODE if MS1=1 Enter SLEEP MODE if MS1=0	None	1
	0010 0111 rrrr rrrr	RPT r (“r” is the content of register r)	Single repeat (r) times on next instruction	None	1
	0100 0011 kkkk kkkk	BANK #k	BSR ← k	None	1
Subroutine	0011 aaaa aaaa aaaa	S0CALL addr	Top of Stack] ← PC+1 PC[11:0] ← addr PC[12:16] ← 00000 (*5)	None	1

	111a aaaa aaaa aaaa	SCALL addr	[Top of Stack] ← PC+1;
			PC[12:0] ← addr;
			PC[13:16] unchanged
				None	1

	0000 0000 0011 0000 00aa aaaa aaaa aaaa	LCALL addr (two words)	[Top of Stack] ← PC+1; PC ← addr	None	2
	0010 1011 1111 1110	RET	PC ← (Top of Stack)	None	1
	0010 1011 1111 1111	RETI	PC ← (Top of Stack); Enable Interrupt	None	1
Compare	0010 0101 rrrr rrrr	TEST r	Z ← 0 if r<>0;Z ← 1 if r=0	Z	1
Jump	110a aaaa aaaa aaaa	SJMP addr	PC ← addr PC[13..15] unchanged	None	1
	0000 0000 0010 0000 00aa aaaa aaaa aaaa	LJMP addr (two words)	PC ← addr	None	2
Compare & Jump	0101 0000 rrrr rrrr aaaa aaaa aaaa aaaa	JDNZ A,r,addr	A ← r-1, jump to addr if not zero PC[15:0] ← addr (*3)	None	2
	0101 0001 rrrr rrrr aaaa aaaa aaaa aaaa	JDNZ r,addr	r ← r-1, jump to addr if not zero PC[15:0] ← addr (*3)	None	2
	0100 0111 kkkk kkkk aaaa aaaa aaaa aaaa	JGE A,#k,addr	Jump to addr if A≧k PC[15:0] ← addr (*3)	None	2
	0100 1000 kkkk kkkk aaaa aaaa aaaa aaaa	JLE A,#k,addr	Jump to addr if A≦k PC[15:0] ← addr (*3)	None	2
	0100 1001 kkkk kkkk aaaa aaaa aaaa aaaa	JE A,#k,addr	Jump to addr if A=k PC[15:0] ← addr (*3)	None	2
	0101 0101 rrrr rrrr aaaa aaaa aaaa aaaa	JGE A,r,addr	Jump to addr if A≧r PC[15:0] ← addr (*3)	None	2
	0101 0110 rrrr rrrr aaaa aaaa aaaa aaaa	JLE A,r,addr	Jump to addr if A≦r PC[15:0] ← addr (*3)	None	2
	0101 0111 rrrr rrrr aaaa aaaa aaaa aaaa	JE A,r,addr	Jump to addr if A=r PC[15:0] ← addr (*3)	None	2
Bit Compare and Jump	0101 1bbb rrrr rrrr aaaa aaaa aaaa aaaa	JBC r,b,addr	If r(b)=0,jump to addr PC[15:0] ← addr (*3)	None	2
	0110 0bbb rrrr rrrr aaaa aaaa aaaa aaaa	JBS r,b,addr	If r(b)=1,jump to addr PC[15:0] ← addr (*3)	None	2
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
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)] (*1)(*2)	None	2
	0010 1111 rrrr rrrr	TBRD A,r	r←ROM[(TABPTR+ACC)] (*2)	None	2

Type	Binary Instruction	Mnemonic	Operation	Status Affected	Cycle
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 (*4)	C,DC,Z,OV,SGE,SLE	1
	0100 1010 kkkk kkkk	ADD A,#k	A ← A+k	C,DC,Z,OV,SGE,SLE	1
	0001 0010 rrrr rrrr	ADC A,r	A ← A+r+C	C,DC,Z,OV,SGE,SLE	1
	0001 0011 rrrr rrrr	ADC r,A	r ← r+A+C	C,DC,Z,OV,SGE,SLE	1
	0100 1011 kkkk kkkk	ADC A,#k	A ← A+k+C	C,DC,Z,OV,SGE,SLE	1
	0001 1110 rrrr rrrr	DECA r	A ← r-1	C,Z	1
	0001 1111 rrrr rrrr	DEC r	r← r-1	C,Z	1
	0001 0110 rrrr rrrr	SUB A,r	A ← r-A (*6)	C,DC,Z,OV,SGE,SLE	1
	0001 0111 rrrr rrrr	SUB r,A	r ← r-A (*6)	C,DC,Z,OV,SGE,SLE	1
	0100 1100 kkkk kkkk	SUB A,#k	A ← k-A (*6)	C,DC,Z,OV,SGE,SLE	1
	0001 1000 rrrr rrrr	SUBB A,r	A ← r-A-/C (*6)	C,DC,Z,OV,SGE,SLE	1
	0001 1001 rrrr rrrr	SUBB r,A	r ← r-A-/C (*6)	C,DC,Z,OV,SGE,SLE	1
	0100 1101 kkkk kkkk	SUBB A,#k	A ← k-A-/C (*6)	C,DC,Z,OV,SGE,SLE	1
	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
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

*(*1) TBRD i, r:*

r← ROM [(TABPTR)]

i=00: TABPTR no change

wi=01: TABPTR←TABPTR+1

i=10: TABPTR←TABPTR-1

*(*2) TABPTR = (TABPTRM: TABPTRL)*

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

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

NOTE

- *Bit 0 of TABPTRL is used to select either low byte or high byte of the pointed ROM data.*
- *The maximum table look-up space is internal 48K bytes (24K words).*

*(*3) The maximum jump range is 8K words absolute address*

*(*4) 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.*

*(*5) S0CALL addressing ability is from 0x000 to 0xFFF (4K words space)*

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

12 Pad Diagram and Pad Locations

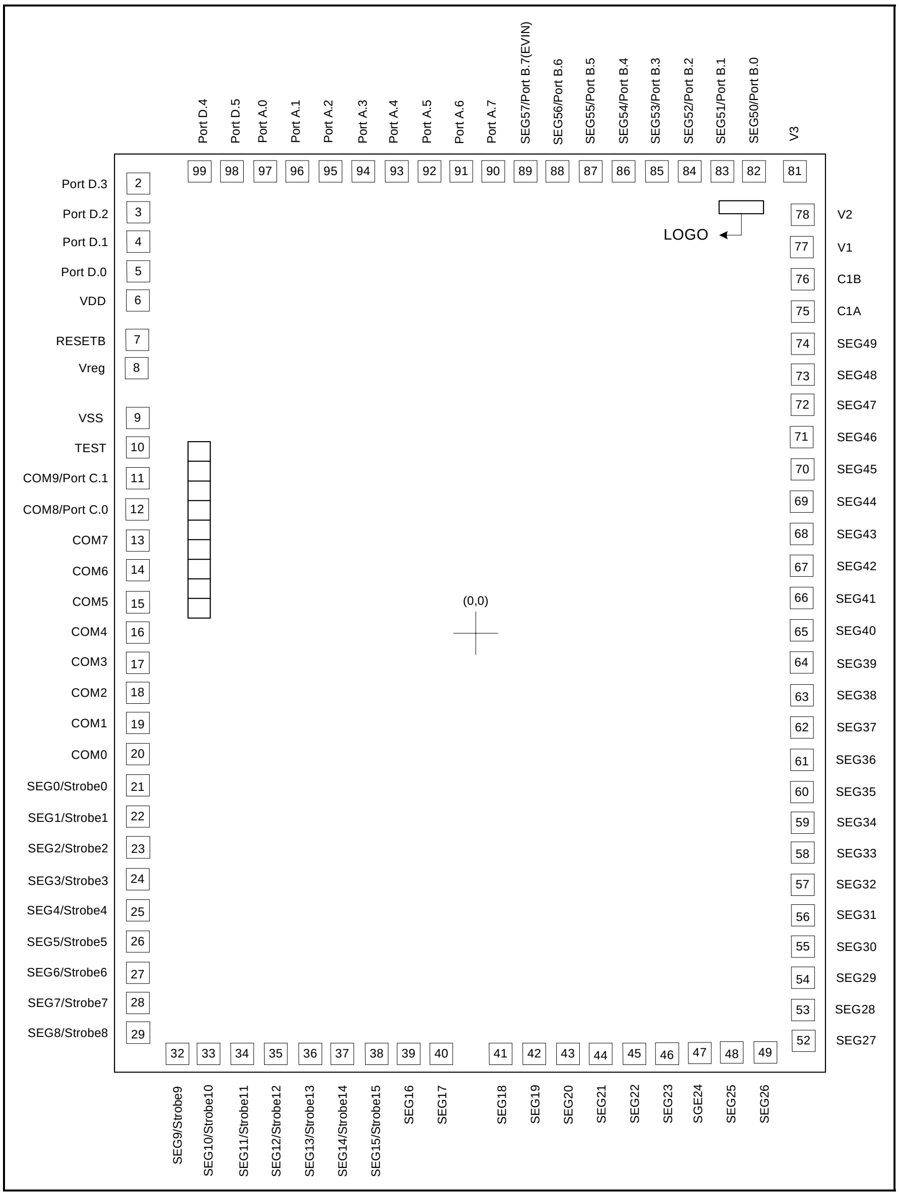


Figure 12-1 ePS7200 92-Pin out Diagram

12.1 Pad Coordinates

- Chip Size: 2000 *2640 um^2

Pin NO.	Symbol	X	Y
1	NC	-	-
2	Port D.3	-920	1224
3	Port D.2	-920	1139
4	Port D.1	-920	1054
5	Port D.0	-920	969
6	VDD	-920	879
7	RESETB	-920	764
8	VREG	-920	674
9	GND	-920	534
10	TEST	-920	449
11	COM9/Port C.1	-920	361.1
12	COM8/Port C.0	-920	276.1
13	COM7	-900.5	170.75
14	COM6	-900.5	85.75
15	COM5	-900.5	0.75
16	COM4	-900.5	-84.25
17	COM3	-900.5	-169.25
18	COM2	-900.5	-254.25
19	COM1	-900.5	-339.25
20	COM0	-900.5	-424.25
21	SEG0/Strobe0	-900.5	-509.25
22	SEG1/Strobe1	-900.5	-594.25
23	SEG2/Strobe2	-900.5	-679.25
24	SEG3/Strobe3	-900.5	-764.25
25	SEG4/Strobe4	-900.5	-849.25
26	SEG5/Strobe5	-900.5	-934.25
27	SEG6/Strobe6	-900.5	-1019.25
28	SEG7/Strobe7	-900.5	-1104.25
29	SEG8/Strobe8	-900.5	-1189.25
30	NC	-	-
31	NC	-	-
32	SEG9/Strobe9	-781.5	-1220.5
33	SEG10/Strobe10	-696.5	-1220.5
34	SEG11/Strobe11	-611.5	-1220.5
35	SEG12/Strobe12	-526.5	-1220.5
36	SEG13/Strobe13	-441.5	-1220.5
37	SEG14/Strobe14	-356.5	-1220.5
38	SEG15/Strobe15	-271.5	-1220.5
39	SEG16	-186.5	-1220.5
40	SEG17	-101.5	-1220.5
41	SEG18	106.45	-1220.5
42	SEG19	191.45	-1220.5
43	SEG20	276.45	-1220.5
44	SEG21	361.45	-1220.5
45	SEG22	446.4	-1220.5
46	SEG23	531.4	-1220.5
47	SEG24	616.4	-1220.5
48	SEG25	701.4	-1220.5
49	SEG26	786.4	-1220.5
50	NC	-	-
51	NC	-	-
52	SEG27	900.5	-1205.9
53	SEG28	900.5	-1120.9
54	SEG29	900.5	-1035.9
55	SEG30	900.5	-950.9
56	SEG31	900.5	-865.9
57	SEG32	900.5	-780.9
58	SEG33	900.5	-695.9
59	SEG34	900.5	-610.9
60	SEG35	900.5	-525.9
61	SEG36	900.5	-440.9
62	SEG37	900.5	-355.9
63	SEG38	900.5	-270.9
64	SEG39	900.5	-185.9
65	SEG40	900.5	-100.9
66	SEG41	900.5	-15.9
67	SEG42	900.5	69.1
68	SEG43	900.5	154.1
69	SEG44	900.5	239.1
70	SEG45	900.5	324.1
71	SEG46	900.5	409.1
72	SEG47	900.5	494.1
73	SEG48	900.5	579.1
74	SEG49	900.5	664.1
75	C1A	900.5	749.1
76	C1B	900.5	834.1
77	V1	900.5	919.1
78	V2	900.5	1004.1
79	NC	-	-
80	NC	-	-

Pin NO.	Symbol	X	Y
81	V3	887.3	1240
82	SEG50/Port B.0	764.15	1240
83	SEG51/Port B.1	679.15	1240
84	SEG52/Port B.2	594.15	1240
85	SEG53/Port B.3	509.15	1240
86	SEG54/Port B.4	424.15	1240
87	SEG55/Port B.5	339.15	1240
88	SEG56/Port B.6	254.15	1240
89	SEG57/Port B.7_EVIN	169.15	1240
90	Port A.7	84.15	1240
91	Port A.6	-0.85	1240
92	Port A.5	-85.85	1240
93	Port A.4	-170.85	1240
94	Port A.3	-255.85	1240
95	Port A.2	-340.85	1240
96	Port A.1	-425.85	1240
97	Port A.0	-510.85	1240
98	Port D.5	-597.15	1240
99	Port D.4	-682.15	1240
100	NC	-	-

NOTE

For PCB layout, the die pad must be connected to VSS (the IC substrate must be connected to VSS or kept floating).