ePS6010

RISC II Series Microcontroller

Product Specification

DOC. VERSION 1.0

ELAN MICROELECTRONICS CORP.

October 2015

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

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

Shenzhen:
ELAN Microelectronics Shenzhen, Ltd.
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

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	2015/04/09
1.0	Initial Release of Offical Version	2015/10/05

1 General Description

The **ePS6010** is an 8-bit RISC MCU embedded with two 8-bit timers, one 16-bit general timer, and a Watchdog Timer. It also has on-chip 3K bytes RAM and 56K 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 ePS6010 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 5.*
- *Do not use Register BSR1 (05h) Bit 7 ~ Bit 5.*
- *Do not use Register BSR2 (12h) Bit 7 ~ Bit 5.*
- *Do not use JDNZ at FSR1 (04h) special register.*
- *Do not use JDNZ at FSR2 (11h) special register.*
- *Check the range of BSR and BSR1 which should be from 0x00~0x017.*
- *Do not use TBRD to read the ROM value when TABPRTM: L>0x1BFFF*
- *Do not use PUSH, POP by "MOV A,r" to avoid affecting S_Z.*
- *Do not use the RAMs: Address=09h,0Eh,22h,23h,2Fh,3Ah,3Bh,3Ch*
- *Do not use these bits: Bits 1~7 of TABPTRH (0Dh), Bit 7 of CPUCON (20h), Bit 2 and Bit 6 of POST_ID (21h), Bit 5 of TR0CON (25h), Bits 0~6 of PACON2 (2Eh), Bits 0~6 of PACON3 (30h)*

1.1 Application

- Calculating machine's controller

2 Features

2.1 MCU

- 8 bit RISC MCU
- Operating voltage and speed: 1.2V~3.6V
- Clock Source: Dual system clock:
  - Low-frequency: 32kHz Internal RC oscillator / Crystal oscillator.
  - High-frequency: 200kHz / 300kHz / 500kHz / 1MHz / 1.5 MHz / 2 MHz External RC oscillator (≧1 MHz only for VDD=2.4~3.6V)
- One Instruction cycle time = 2 × System clock time
- Program ROM addressing: Maximum of 56K words
- 128 bytes un-banked RAM including special registers and common registers
- 24×128 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

- 32 general I/O pins (Port A.0~7, Port B.0~7, Port D.0~7, Port E.0~7)
- 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 56 keys (Key matrix: Port A.0~6 and Port B.0~7)

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 and crystal oscillation for system clock
- MCU wake-up function consists of input wake up and Timer 1 wake up
- MCU interrupt function consist of Input port interrupt and Timer interrupt (Timers 0 ~ 2)
- MCU reset function includes power-on reset, RSTB pin reset, and Watchdog timer reset

3 Block Diagram

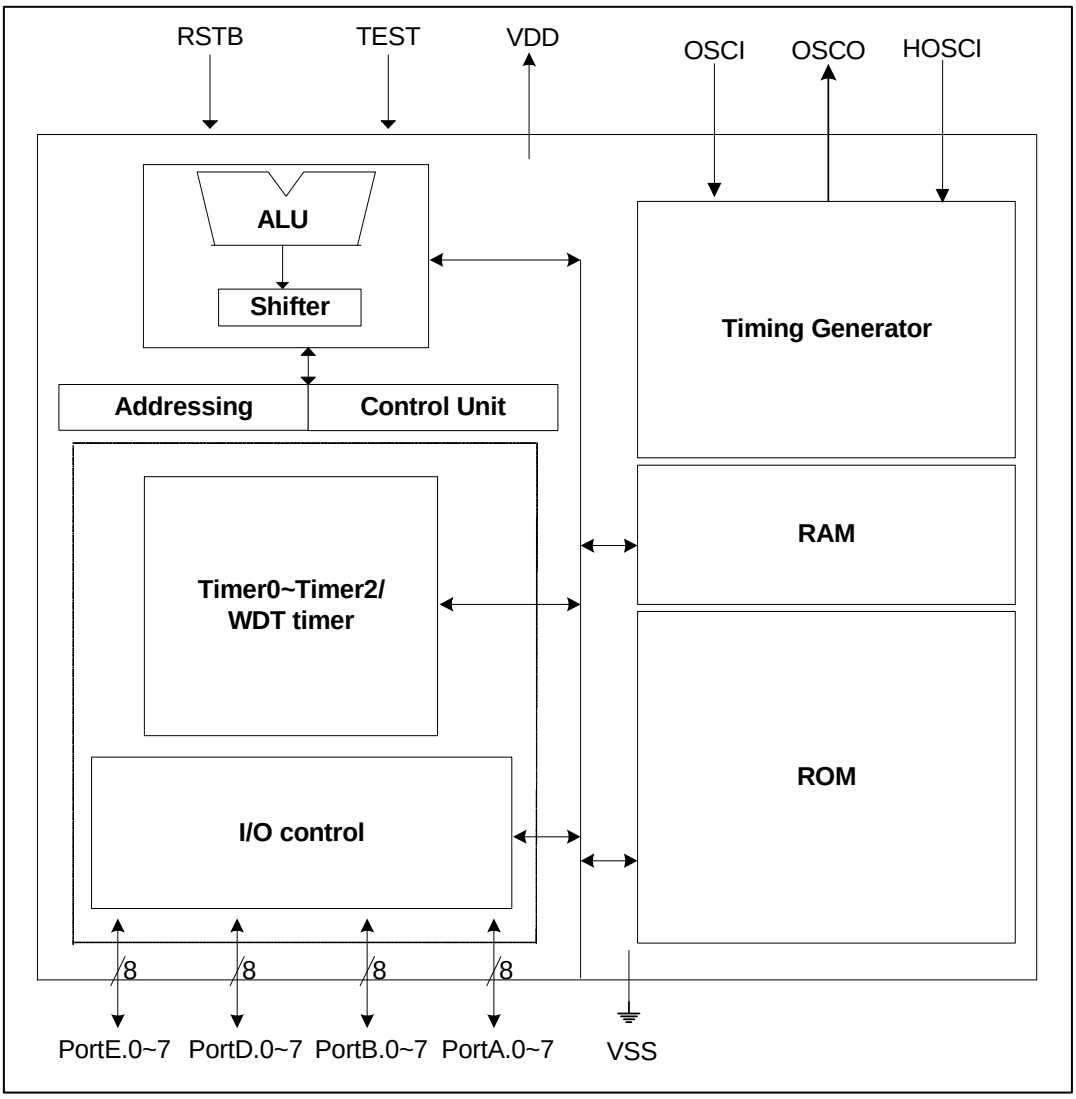


Figure 3-1 ePS6010 Block Diagram

4 Pin Assignment

■ **39-pin Chip Type**

No.	Pin Name	No.	Pin Name	No.	Pin Name	No.	Pin Name
1	VDD	11	Port A.3	21	Port B.5	31	Port D.7
2	HOSCI	12	Port A.4	22	Port B.6	32	Port E.0
3	RESETB	13	Port A.5	23	Port B.7	33	Port E.1
4	OSCI	14	Port A.6	24	Port D.0	34	Port E.2
5	OSCO	15	Port A.7	25	Port D.1	35	Port E.3
6	VSS	16	Port B.0	26	Port D.2	36	Port E.4
7	TEST	17	Port B.1	27	Port D.3	37	Port E.5
8	Port A.0	18	Port B.2	28	Port D.4	38	Port E.6
9	Port A.1	19	Port B.3	29	Port D.5	39	Port E.7
10	Port A.2	20	Port B.4	30	Port D.6		

5 Pin Description

5.1 MCU System Pins (7 Pins)

Name	I/O/P Type	Description	Note
VDD	P	The positive power supply, ranging from 1.2V~3.6V. 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
OSCI	I	Crystal oscillator connecting pin	–
OSCO	O	Crystal oscillator connecting pin	–
HOSCI	I	Hi-Speed RC oscillator connecting pin.	Ext. R to VDD

5.2 I/O Port (32 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 (R2: Controllable Large resister)
		Interrupt and wake up	I	VDD	Input port interrupt and wake-up pin	
		General Output	O	VDD		
Port B	Bits 7~0	General Input	I	VDD		–
		General Output	O	VDD		
Port D	Bits 7~0	General Input	I	VDD		–
		General Output	O	VDD		
Port E	Bits 7~0	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
- Operating voltage option:
  - Select “1.5V” or “3V”
- Select the oscillator frequency:
  - Select “200kHz” or “300kHz” or “500kHz” or “1 MHz” or “1.5 MHz” or “2 MHz” (≧1 MHz only for VDD=2.4~3.6V)

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

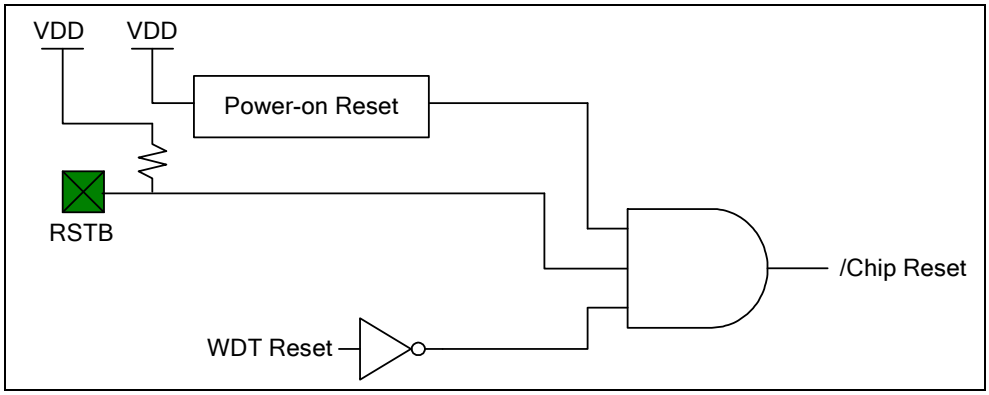


Figure 7-1 On-chip Reset Schematic Diagram

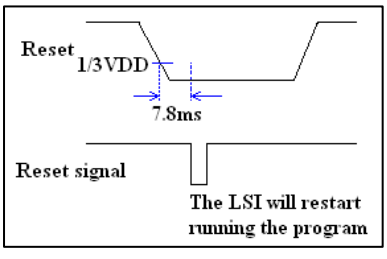
7.1.1 Power-on Reset

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 Registers

Addr.	Name	Initial Value	Addr.	Name	Initial Value
00h	INDF0	uuuu uuuu [1]	10h	INDF2	uuuu uuuu [1]
01h	FSR0	0000 0000	11h	FSR2	1000 0000
02h	BSR	---0 0000	12h	BSR2	---0 0000
03h	INDF1	uuuu uuuu [1]	13h	General RAM	uuuu uuuu
04h	FSR1	1000 0000	14h	General RAM	uuuu uuuu
05h	BSR1	---0 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	–	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	–	uuuu uuuu [1]	1Eh	General RAM	uuuu uuuu
0Fh	STATUS	cuuu uuuu [2]	1Fh	General RAM	uuuu uuuu

7.1.5.2 Control Registers

Addr.	Name	Initial Value	Addr.	Name	Initial Value
20h	CPUCON	0--- -00c [3]	30h	PACON3	0000 0000
21h	POST_ID	1111 0000	31h	PORTA	xxxx xxxx
22h	–	0000 0000	32h	PACON	0000 0000
23h	–	0000 0000	33h	DCRA	1111 1111
24h	INTSTA	---- -000	34h	PAWAKE	0000 0000
25h	TR0CON	--00 0000	35h	PAINTEN	0000 0000
26h	TRL0L	uuuu uuuu	36h	PAINTSTA	0000 0000
27h	TRL0H	uuuu uuuu	37h	PORTB	xxxx xxxx
28h	T0CL	0000 0000	38h	PBCON	0000 0000
29h	T0CH	0000 0000	39h	DCRB	1111 1111
2Ah	TR1CON	0--0 0-00	3Ah	–	---- ----
2Bh	TRL1	uuuu uuuu	3Bh	–	---- ----
2Ch	TR2WCON	0000 0000	3Ch	–	---- ----
2Dh	TRL2	uuuu uuuu	3Dh	PORTD	xxxx xxxx
2Eh	PACON2	0000 0000	3Eh	PORTE	xxxx xxxx
2Fh	–	0000 0000	3Fh	DCRDE	0011 0011

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.

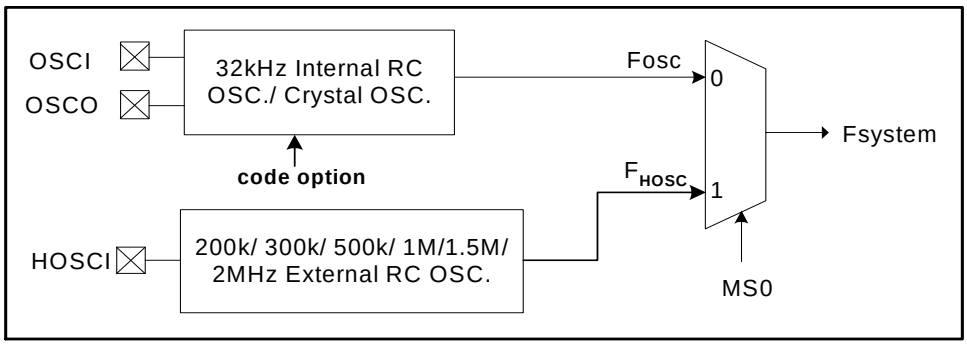


Figure 7-3 Oscillator System Function Block Diagram

The **MS0** bit (mode select bit) of **CPUCON** register (R20h) 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 32.768kHz Crystal Oscillator

- 32.8kHz Internal RC oscillator**:**
  Select "RC oscillator for FOSC" in the code option and allow OSCI and OSCO pins to stay floating.
- 32.768kHz Crystal oscillator**:**
  Select "Crystal oscillator for FOSC" in the code option and connect a crystal and a resistor (10 MΩ) between OSCI and OSCO pins. The OSCI and OSCO pins are also connected to ground through a 20 pF capacitor respectively.

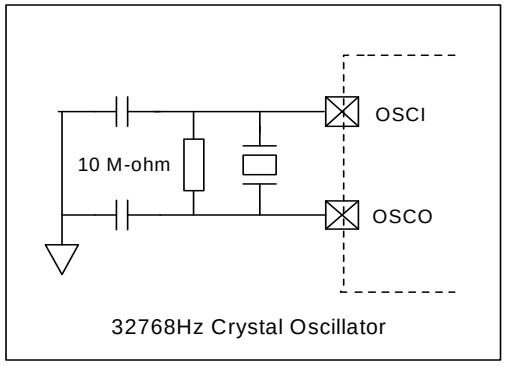


Figure 7-4 Slow Mode Crystal Circuit Diagram

7.2.2 200kHz/300kHz/500kHz/1MHz/1.5MHz/2MHz RC Oscillator

- 200kHz/ 300kHz / 500kHz / 1 MHz / 1.5 MHz / 2 MHz RC External oscillator**:**

  Select "RC oscillator for FHOSC" in the code option. A resistor should be connected between HOSCI and VDD pin.

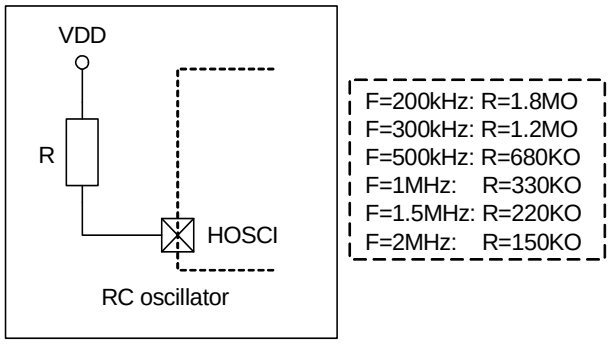


Figure 7-7 Fast Mode RC Oscillators Circuit Diagram

7.3 MCU Operation Mode

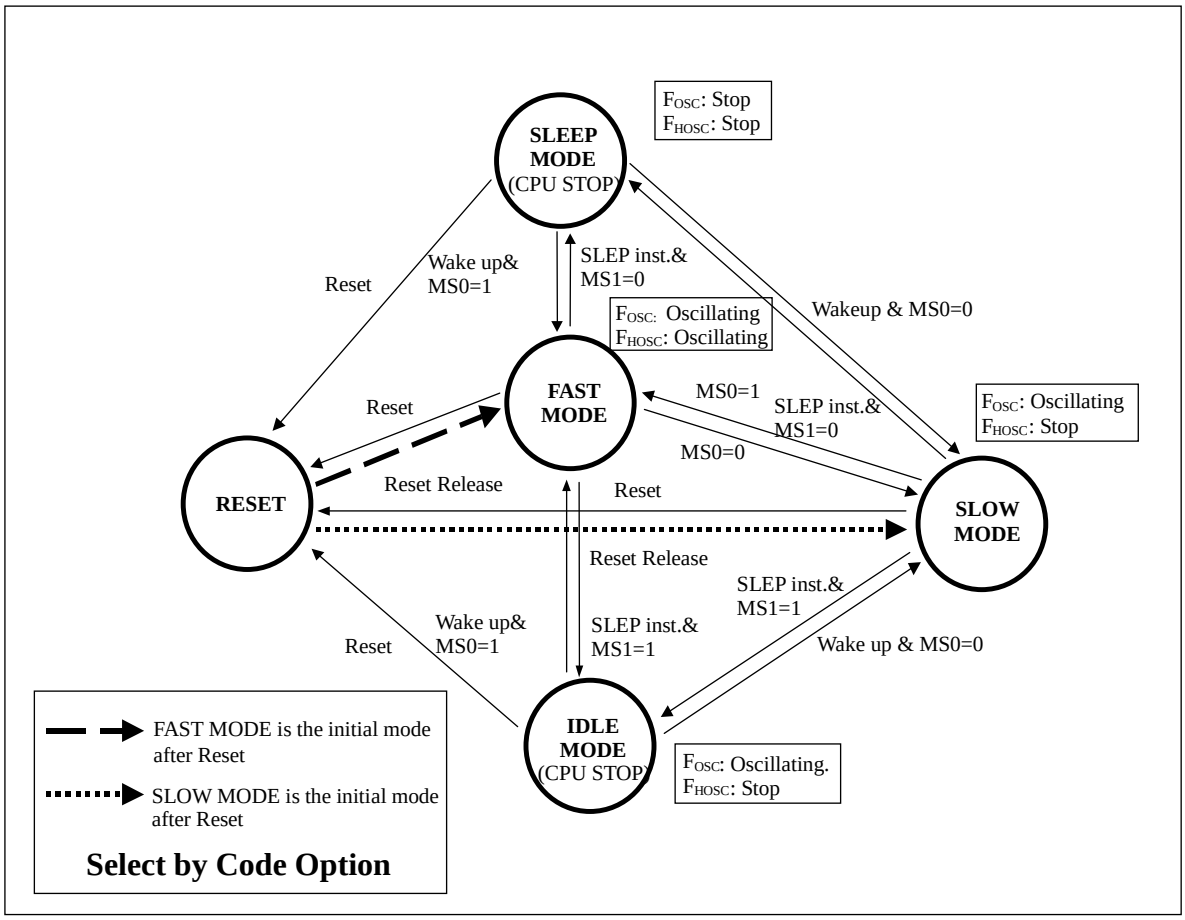


Figure 7-4 MCU Operation Block Diagram

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

Device \ Mode	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 (R20h): MCU Control Register**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Fixed to 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 32 kHz. 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 32kHz 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 32 kHz 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 (R34h): 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 (R2Ah): Timer 1 Control Register**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1WKEN	–	–	TMR1IE	T1EN	–	T1PSR1	T1PSR0

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 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 (R20h) MCU Control Register**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Fixed to 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 (R36h): 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 (R35h): 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 TR0CON (R25h) Timer 0 Control Register**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	–	Fixed to 0	**TMR0IE**	T0EN	T0CS	T0PSR1	T0PSR0

Bit 4 (TMR0IE): Control bit of Timer 0 interrupt

0: Disable Timer 0 interrupt function

1: Enable Timer 0 interrupt function

■ **TMR0I Bit of INTSTA (R24h) 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 Timer 0 interrupt

Set to "**1**" when Timer 0 counter underflows

Clear to "**0**" by software

7.4.3.2 Timer 1 Interrupt

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

■ **TMR1IE Bit of TR1CON (R2Ah) Timer 1 Control Register**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1WKEN	–	–	**TMR1IE**	T1EN	–	T1PSR1	T1PSR0

Bit 4 (TMR1IE): Control bit of Timer 1 interrupt

0: Disable Timer 1 interrupt function

1: Enable Timer1 interrupt function

■ **TMR1I Bit of INTSTA (R24h) 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 Timer 1 interrupt

Set to "**1**" when Timer1 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 TR2WCON (R2Ch) 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	**TMR2IE**	T2EN	T2CS	T2PSR1	T2PSR0

Bit 4 (TMR2IE): Control bit of Timer 2 interrupt

0: Disable Timer 2 interrupt function

1: Enable Timer 2 interrupt function

■ **TMR2I Bit of INTSTA (R24h) 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 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 = 56K Words	
Address.	**Description**
0000h | 000Bh	Interrupt Vector (12 words)
000Ch | 000Fh	Code Option (4 words)
0010h | 001Fh	Test Program (16 words)
0020h | DFFFh	Program or Fixed data region

7.6 RAM Map for Special and Control Registers

RAM Size: 77 Bytes + 24 Banks × 128 Bytes = 3149 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/W	R/W	R/W	R/W	R/W
		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/W	R/W	R/W	R/W	R/W
		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	-	R	R	R	R	R	R	R	R
		Fixed 0	Fixed 0	Fixed 0	Fixed 0	Fixed 0	Fixed 0	Fixed 0	Fixed 0
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	R	R	R	R	R	R	R/W
		Fixed 0	Fixed 0	Fixed 0	Fixed 0	Fixed 0	Fixed 0	Fixed 0	High byte of table pointer
E	-	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
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	INDF2	R/W							
		Indirect Addressing Pointer 2							
11	FSR2	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		Fixed 1	File select Register 1 for INDF2 (R15)						
12	BSR2	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		Fixed 0	Bank select Register 2 (for INDF2)						

Addr.	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
20	CPUCON	R	-	-	-	-	R/W	R/W	R/W
		Fixed 0	-	-	-	-	GLINT	MS1	MS0
21	POST_ID	R/W	-	R/W	R/W	R/W	-	R/W	R/W
		FSR2_ID	-	FSR1_ID	FSR0_ID	FSR2_PE	-	FSR1_PE	FSR0_PE
22	-	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
23	-	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
24	INTSTA	-	-	-	-	-	R/W	R/W	R/W
		-	-	-	-	-	TMR2I	TMR1I	TMR0I
25	TR0CON	-	-	R	R/W	R/W	R/W	R/W	R/W
		-	-	Fixed 0	TMR0IE	T0EN	T0CS	T0PSR1	T0PSR0
26	TRL0L	R/W							
		Timer 0 auto-reload register low byte							
27	TRL0H	R/W							
		Timer 0 auto-reload register high byte							
28	T0CL	R/W							
		Timer 0 counting value register low byte							
29	T0CH	R/W							
		Timer 0 counting value register high byte							
2A	TR1CON	R/W	-	-	R/W	R/W	-	R/W	R/W
		T1WKEN	-	-	TMR1IE	T1EN	-	T1PSR1	T1PSR0
2B	TRL1	R/W							
		Timer 1 auto-reload register							
2C	TR2WCON	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		WDTEN	WDTPSR1	WDTPSR0	TMR2IE	T2EN	T2CS	T2PSR1	T2PSR0
2D	TRL2	R/W							
		Timer 2 auto-reload register							
2E	PACON2	R/W	-	-	-	-	-	-	-
		R1EN	-	-	-	-	-	-	-
2F	-	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
30	PACON3	R/W	-	-	-	-	-	-	-
		KE	-	-	-	-	-	-	-
31	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
32	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
33	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
34	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
35	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
36	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

Addr.	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
37	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
38	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
39	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
3A	-	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
3B	-	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
3C	-	-	-	-	-	-	-	-	-
		-	-	-	-	-	-	-	-
3D	PORTD	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		Port D.7	Port D.6	Port D.5	Port D.4	Port D.3	Port D.2	Port D.1	Port D.0
3E	PORTE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		Port E.7	Port E.6	Port E.5	Port E.4	Port E.3	Port E.2	Port E.1	Port E.0
3F	DCRDE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		EHNPU	ELNPU	EHNDC	ELNDC	DHNPU	DLNPU	DHNDC	DLNDC

7.6.2 Other Unbanked General RAM

Address	Unbanked
13h | 1Fh	General purpose RAM
40h | 7Fh	General purpose RAM

7.6.3 Banked General RAM

Address	Bank 0	Bank 1	Bank 2	……	Bank 22	Bank 23
80h | FFh	General Purpose RAM	General Purpose RAM	General Purpose RAM	……	General Purpose RAM	General Purpose RAM

7.7 Special Function Registers

7.7.1 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.

7.7.2 POST_ID (R21h): Post Increase / Decrease Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FSR2_ID	–	FSR1_ID	FSR0_ID	FSR2PE	–	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 3 (FSR2PE): Enable FSR2 post increase/decrease function. FSR2 will carry into or borrow from BSR2.

Bit 4 (FSR0_ID): **1:** auto increase FSR0

0: auto decrease FSR0

Bit 5 (FSR1_ID): **1:** auto increase FSR1

0: auto decrease FSR1

Bit 7 (FSR2_ID): **1:** auto increase FSR2

0: auto decrease FSR2

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

BSR (R02h) determines which bank is active (working bank) among the 24 banks (Bank 0 ~ Bank 17).

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.

7.7.4 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.

7.7.5 BSR2, FSR2, INDF2 (R12h, R11h, R10h): Indirect Address Pointer 2 Registers

BSR2 (R12h) is a bank register for INDF2. It cannot determine the working bank for general register.

FSR2 (R11h) is an address register for INDF2. You can select up to 128 bytes (Address**:** 80 ~ 0FFh). Bit 7 of FSR2 is fixed to "1".

INDF2 (R10h) is not a physically implemented register.

■ **Code Example :**

```
Data transform Bank 0 to Bank 1:
     MOV    A,#00110011B          ; Enable FSR0 & FSR1 post increase
     MOV    POST_ID,A
     BANK   #0                    ; BSR = 0 working Bank
     MOV    A,#1
     MOV    BSR1,A                ; BSR1 = 1 is Bank 1
     MOV    A,#80H
     MOV    FSR0,A                ; FSR0 = 80H
     CLR    FSR1                  ; FSR1 = 80H
     MOV    A,#80H
     RPT    ACC
     MOVRP  INDF1,INDF0           ; Move 80H ~ 0FFH data to Bank 1
         :
Data transform bank 0 to bank 2:
     MOV    A,#10101010B          ; Enable FSR0 & FSR2 post increase
     MOV    POST_ID,A
     BANK   #0                    ; BSR = 0 working Bank
     MOV    A,#2
     MOV    BSR2,A                ; BSR2 = 1 is Bank 2
     MOV    A,#80H
     MOV    FSR0,A                ; FSR0 = 80H
     CLR    FSR2                  ; FSR2 = 80H
     MOV    A,#80H
     RPT    ACC
     MOVRP  INDF2,INDF0           ; Move 80H ~ 0FFH data to Bank2
         :
```

■ INDF1 Linear Address Capabilities

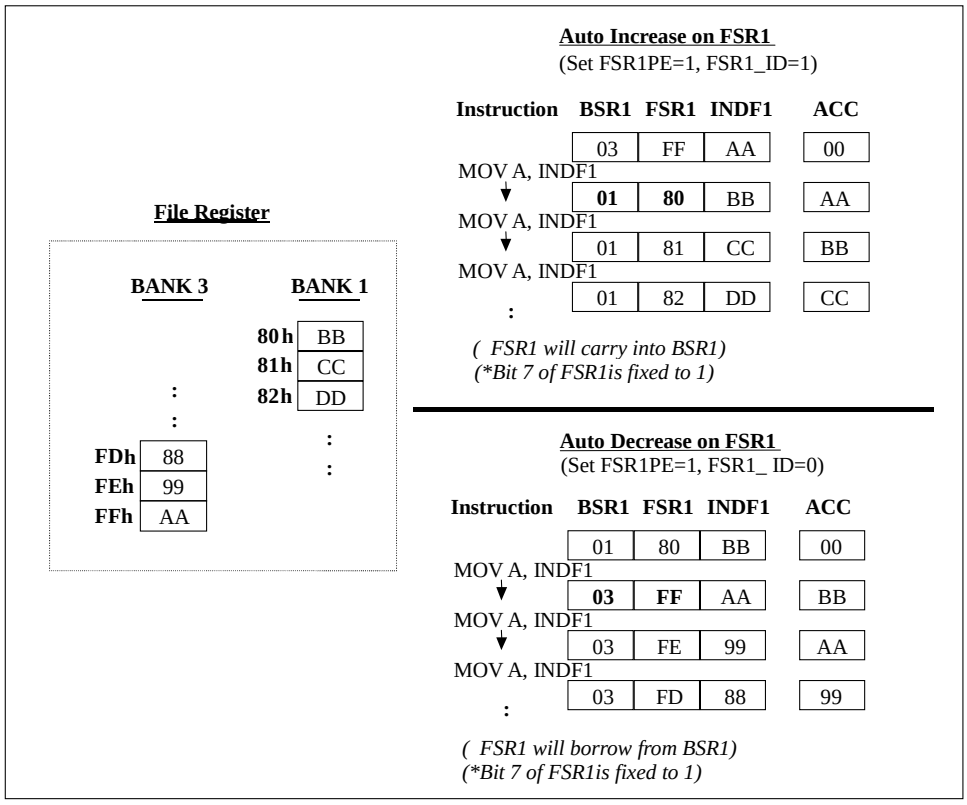


Figure 7-5 INDF1 Linear Address Capabilities Diagram

■ 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  #19
          :
          :
MnLoop:
          :
          :
     LJMP   MnLoop

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

■ INDF2 Linear Address Capabilities

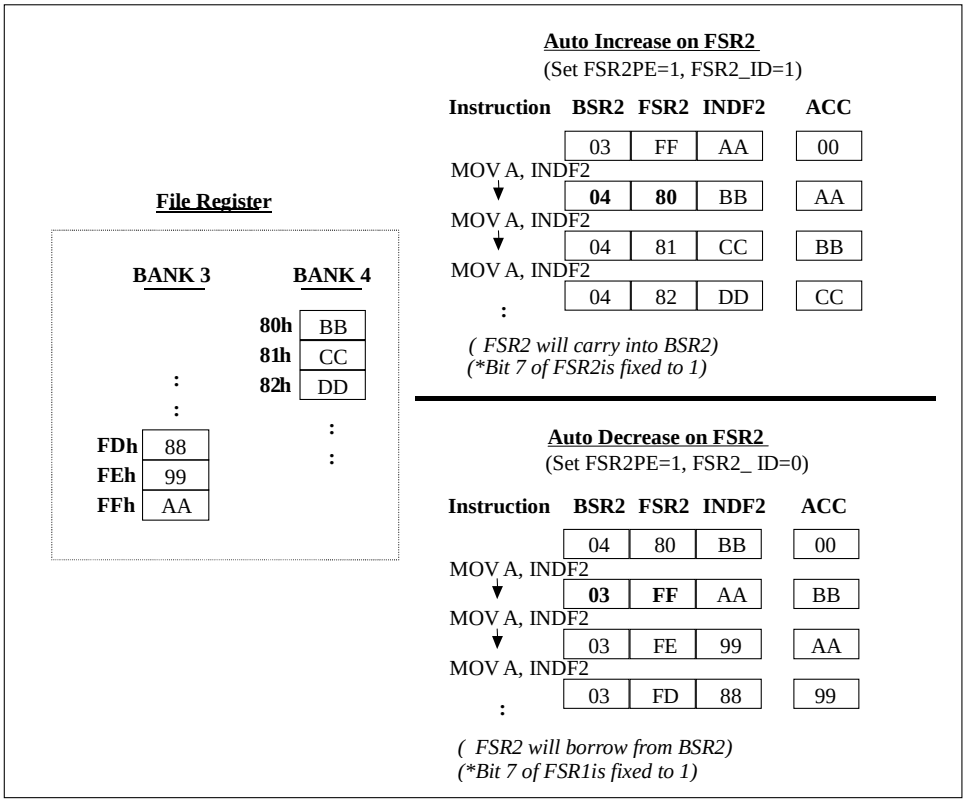


Fig. 7-6 INDF2 Linear Address Capabilities Diagram

■ Code Example :

```
;************************************
;*   Const => Working bank setting
  ;*  REG => Save or Recall register
;************************************
;  *****  RAM stack macro
;  ***  Initial RAM stack
IniRAMsk  MACRO   #Const
     MOV    A,#Const
     MOV    BSR2,A
     CLR    FSR2
     BS     POST_ID,FSR2PE
     ENDM
  ; *** Push RAM stack
PushRAM   MACRO  REG
     BS     POST_ID,FSR2_ID
     MOVRP  INDF2,REG
     ENDM
;  ***  Pop RAM stack
PopRAM    MACRO  REG
     BC     POST_ID,FSR2_ID
     MOVPR  REG,INDF2
     ENDM
```

```
;  ***  Main start program
Mstart:
          :
          :
     IniRAMsk  #19
          :
          :
MnLoop:
          :
          :
     LJMP   MnLoop

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

7.7.6 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.

7.7.7 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 56K×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 (56K×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
```

7.7.8 TABPTRL, TABPTRM, TABPTH (R0Bh, R0Ch, R0Dh): Table Pointer Registers

Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0							TABPTRH	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
        :
        :
     TBPTH  #(PROMTabB*2)/10000H
     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
```

7.7.9 CPUCON (R20h): MCU Control Register

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

Bit 7 (WBK): Select Working RAM

0: Set unbanked (24h~3Fh) change to Control registers

1: Set unbanked (24h~3Fh) change to General purpose RAM

Bit 2 (GLINT): Global interrupt control bit

0: disables all interrupts

1: enables all un-mask interrupts

Bit 1 (MS1): Select SLEEP MODE or IDLE MODE after executing “SLEP” instruction

0: SLEEP MODE

1: IDLE MODE

Bit 0 (MS0): Select SLOW MODE or FAST MODE

0: SLOW MODE

1: FAST MODE

7.7.10 PACON3 (R30): Strobe Output Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
KE	–	–	–	–	–	–	–

Bit 7 (KE): Port A input enable/disable control bit (control at Port A.6 ~ 0)

0: Disable Port A input function.

1: Enable Port A input function.

7.7.11 Port A (R31h): General I/O Pins Register

Port A (R31h) Port A.0 ~ 7 are general I/O pins registers

7.7.12 PACON (R32h): 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 Port A.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

7.7.13 DCRA (R33h): 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 to output pin

1: Set to input pin

7.7.14 PAWAKE (R34h): 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 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.

7.7.15 PAINTEN (R35h): 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.

7.7.16 PAINTSTA (R36h): 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

7.7.17 Port B (R37h): General I/O Pins Register

Port B (R37h) PortB.0 ~ 7 are general I/O pins register

7.7.18 PBCON (R38h): 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 pull up resistor

1: Enable pull up resistor.

NOTE

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

7.7.19 DCRB (R39h): 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 ~Port B.7 direction control bits

0: Set to output pin

1: Set to 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
```

7.7.20 Port D, Port E (R3Dh, R3Eh): General I/O Pins Registers

Port D (R3Dh) Port D.0 ~ 7 are general I/O pins register

Port E (R3Eh) Port E.0 ~ 7 are general I/O pins register

7.7.21 DCRDE (R3Fh): Port D/Port E Direction Control and Pull-up Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EHNPU	ELNPU	EHNDC	ELNEC	DHNPU	DLNPU	DHNDC	DLNDC

Bit 1, Bit 5 (DHNDC, EHNDC) and Bit 0, Bit 4 (DLNDC, ELNDC): Port D and E high / low nibbles direction control.

0: Set to output pin

1: Set to input pin

Bit 3, Bit 7 (DHNPU, EHNPU) and Bit 2, Bit 6 (DLNPU, ELNPU): Enable Ports D and E high/low nibbles pull-high resistor.

0: Disable pull-up resistor

1: Enable pull-up resistor

NOTE

1. *The pull-up function is only available with Port D and E selected as input pin.*
2. *When a Ports D and E bits are set to input pin, a 15 µsec delay in reading the Port D and E data must be provided. Otherwise, the read data will be inaccurate. See Example below.*

■ **Code Example:**

```
; *** Set Port D to input pins
     MOV    A,#0X0F
     MOV    DCRDE,A
Read_PD:
     JBS    PORTD,0,Read_PD
        Delay 15usec
     JBS    PORTD,0,Read_PD
     SJMP   Read_PD
```

```
; *** Set Port E to input pins
     MOV    A,#0XF0
     MOV    DCRDE,A
Read_PE:
     JBS    PORTE,0,Read_PE
        Delay 15usec
     JBS    PORTE,0,Read_PE
     SJMP   Read_PE
```

8 Peripheral

8.1 Timer 0 (16-Bit Timer)

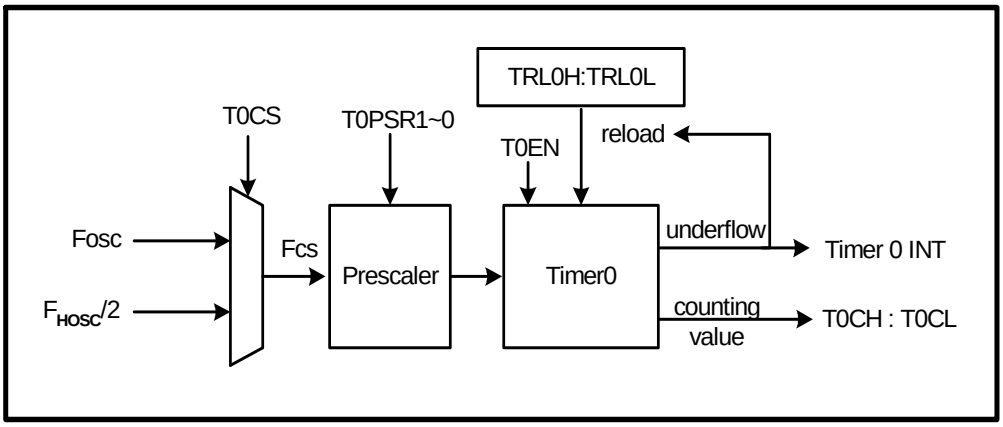


Figure 8-1a Timer 0 Function Block Diagram

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.

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

8.1.1 Timer 0 Registers

- **TRL0H:TRL0L (R26h, R27h): 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.

- **T0CH:T0CL(R28H, R29H): Timer 0 Counter Value Register**

Used to store the value compared with Timer 0 register.

■ **TR0CON (R25h): Timer Control Register**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	–	–	**TMR0IE**	**T0EN**	**T0CS**	**T0PSR1**	**T0PSR0**

Bit 4 (TMR0IE): Timer 0 interrupt control bit

0: Disable Interrupt function

1: Enable Interrupt function

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 (R20h): MCU Control Register**

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

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

0: Disable all interrupts

1: Enable all un-mask interrupts

■ **INTSTA (R24h): 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 to "**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     TR0CON,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      TR0CON,T0EN             ; Timer0 enable.
     BS      TR0CON,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 Timer)

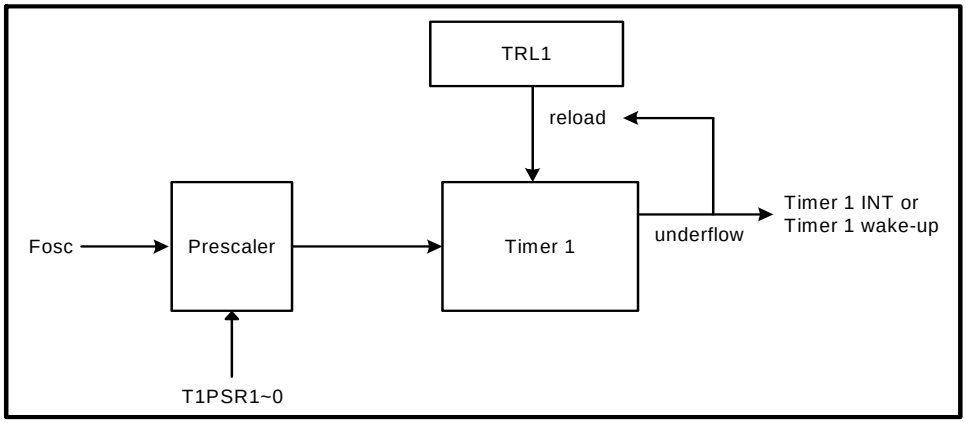


Figure 8-2 Timer 1 Function Block Diagram

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.

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

8.2.1 Timer 1 Registers

■ **TRL1 (R2Bh): 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 (R20h): MCU Control Register**

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

Bit 2 (GLINT) Global interrupt enable/disable bit

0: Disable all interrupt

1: Enable all un-mask interrupt

■ **TR1CON (R2Ah): Timer 1 Control Register**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1WKEN	–	–	**TMR1IE**	**T1EN**	–	**T1PSR1**	**T1PSR0**

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 4 (TMR1IE): Control bit of Timer 1 interrupt

0: Disable Interrupt function

1: Enable Interrupt function

Bit 3 (T1EN): Timer 1 enable control bit

0: Disable Timer 1 (stop counting)

1: Enable Timer 1

Bit 1 ~ Bit 0 (T1PSR1 ~ T1PSR0): Timer 1 prescaler select bits

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

■ **INTSTA (R24h): 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 to "**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,#10000011B
     MOV     TR1CON,A              ; Fosc & Pre-scale 1:256 & wakeup
     MOV     A,#03FH
     MOV     TRL1,A                ; 0.5sec=[256x(63+1)]/32.768K
     BS      TR1CON,T1EN           ; Timer1 enable.
     BS      TR1CON,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 Timer)

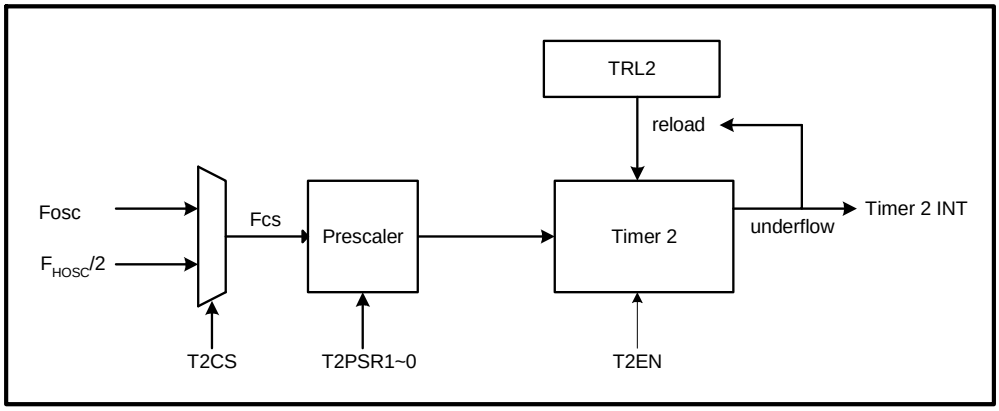


Figure 8-3 Timer 2 Function Block Diagram

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).

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

8.3.1 Timer 2 Registers

■ **TRL2 (R2Dh): 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 (R2Ch): 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	**TMR2IE**	**T2EN**	**T2CS**	**T2PSR1**	**T2PSR0**

Bit 4 (TMR2IE): Control bit of Timer 2 interrupt

0: Disable Interrupt function

1: Enable Interrupt function

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 F_{OSC}

1: Clock source is from $F_{HOSC}/2$

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 (R20h): MCU Control Register**

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

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

0: Disable all interrupts

1: Enable all un-mask interrupts

■ **INTSTA (R24h): 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.
Clear to "**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 [4 x (FFh + 1)]
      Timer2SR:
        :
     PA.7 setting output pin.
        :
     MOV     A,#00000010B
     MOV     TR2WCON,A           ;Fosc & Pre-scale=1:4
     MOV     A,#0XFF
     MOV     TRL2,A              ;31.25ms=[4x(255+1)]/32768
     BS      TR2WCON,T2EN        ;Timer2 Enable
     BS      TR2WCON,TMR2IE      ;Timer2 interrupt Enable
     BC      INTSTA,TMR2I        ;Clear Timer2 interrupt Statu
TMR2Loop:
     SJMP    TMR2Loop
```

8.4 Watchdog Timer (WDT)

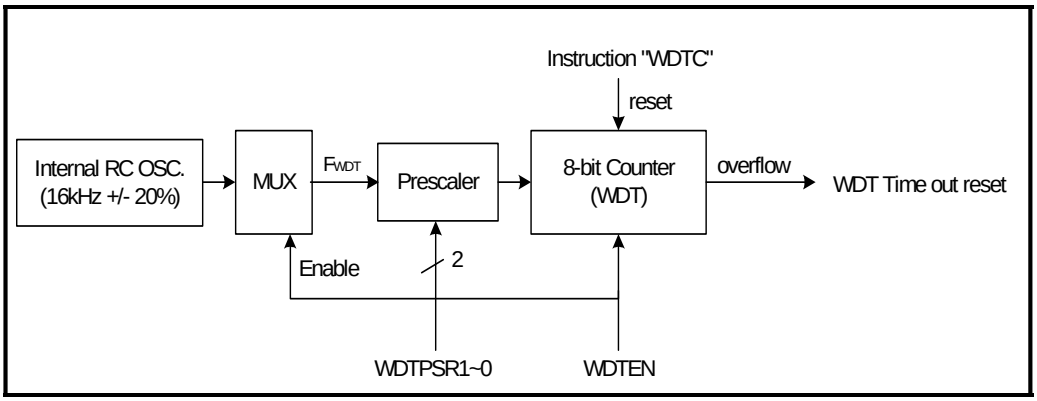


Figure 8-4 Watchdog Timer Functional Block Diagram

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 (R2Ch): 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**	TMR2IE	T2EN	T2CS	T2PSR1	T2PSR0

Bit 7 (WDTEN): Watchdog Timer enable bit

0: Disable Watchdog Timer (stop running)

1: Enable Watchdog Timer

Bit 6 ~ Bit 5 (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.6 ~ 0 and Port B.7 ~ 0) can have a maximum of 56keys 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.

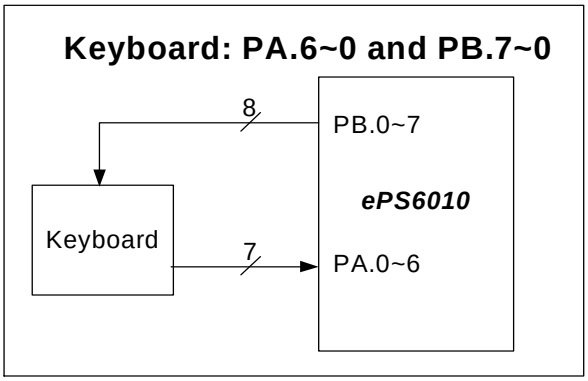


Figure 8-5 Key Function Block Diagram

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

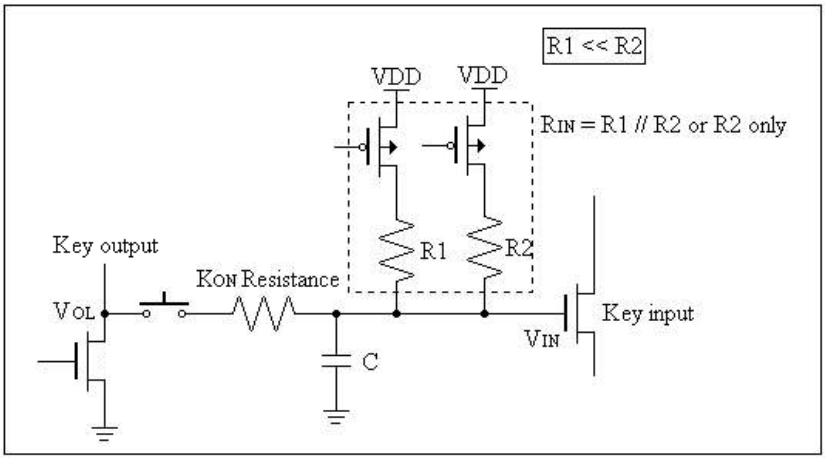


Figure 8-6 Key Circuit Diagram

8.5.1 Key Functions

KE	R1EN	PA6PU ~ PA0PU	Total Pull-up Resistor	Port A.0 ~ 6	Note
0	×	×	Floating	High-Z	-
1	0	0	**Floating**	**Floating**	**Prohibited**
	0	1	R2	PA.0~.6	-
	1	0	R1	PA.0~.6	-
	1	1	R1 // R2 [1]	PA.0~.6	-

× : Don't care

[1] *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 B.0~7) in determining which Key output pins must be output pin with low voltage.

Key Intput					KEY Output									
KE	R1EN	PA6PU~ PA0PU	Total Pull-up Resistor	PORTA. 6 ~ 0	PBCON	DCRB	PORTB							
							Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
0	X	X	Floating	High-Z	**0xFF**	**0xFF**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
1	0	1	R2	PA.0~.6	**0xFF**	**0xFE**	**0**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
						0xFD	**1**	**0**	**1**	**1**	**1**	**1**	**1**	**1**
						0xFB	**1**	**1**	**0**	**1**	**1**	**1**	**1**	**1**
						0xF7	**1**	**1**	**1**	**0**	**1**	**1**	**1**	**1**
						0xEF	**1**	**1**	**1**	**1**	**0**	**1**	**1**	**1**
						0xDF	**1**	**1**	**1**	**1**	**1**	**0**	**1**	**1**
						0xBF	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**1**
						0x7F	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**0**

x: Don't care.

8.5.3 Input/Output Key Registers

■ **DCRA(R33): Port A Direction Control Register**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PA7DC	**PA6DC**	**PA5DC**	**PA4 DC**	**PA3DC**	**PA2DC**	**PA1DC**	**PA0DC**

Bit 6 ~ Bit 0: Direction control of Port A.0~6

0: output pin

1: input pin

■ **Port A (R31h): Port A Register**

Bit 6 ~ Bit 0: Port A is input is selected by PA6DC~PA0DC bits of DCRA register (above). The input structure and two-stage pull up resistor are controlled together by PA6PU ~ PA0PU bits of PACON register and R1EN bit of PACON2 register, KE bit of PACON3 register (see below).

■ **PACON (R32h): 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 6 ~ Bit 0 (PA6PU ~ PA0PU): Pull-up resistor (R2 large resistor) control bits

0: Disable PortA.0 ~ PortA.6 pull-up resistor

1: Enable PortA.0 ~ PortA.6 pull-up resistor

■ **PAINTEN (R35h): 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 (R36h): 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,

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

■ **Port B (R37h): Port B Register**

Bit 7 ~ Bit 0: Port B is output that is selected by PB7DC ~ PB0DC bits of DCRB register (see DCRB below).

■ **PBCON (R38h): Port B 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 pull up resistor

1: Enable pull up resistor

■ **DCRB(R39): 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: Direction control of Port B.0~7

0: output pin

1: input pin

■ **PACON2 (R2Eh): LCD Control Register**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
R1EN	–	–	–	–	–	–	–

Bit 7 (R1EN): R1 pull up resistor (small resistor) control bit for Port A.6 ~ Port A.0.

0: Disable R1 pull up resistor

1: Enable R1 pull up resistor

■ **PACNO3 (R30h): Strobe Output Control Register**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
KE	–	–	–	–	–	–	–

Bit 7 (KE): Port A input enable/disable control bit (control at Port A.6 ~ 0)

0: Disable Port A input function.

1: Enable Port A input function.

■ **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     PACON2,R1EN
     BS     PACON3,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
```

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
Input voltage (general input port)	VIN	-	–0.5 to VDD +0.5	V
Operating temperature range	TOPR	-	–10 to +70	℃
Storage temperature range	TSTR	-	–55 to +125	℃

■ Recommended Operating Conditions

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

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

Parameter	Sym.	Condition		Min	Typ	Max	Unit
Clock	F_{HOSC}	Main-clock frequency	RC OSC. R=1.8MΩ	140	200	260	kHz
			RC OSC. R=1.2MΩ	210	300	390	
			RC OSC. R=680KΩ	350	500	650	
	Fosc	Sub-clock frequency	Internal RC OSC	24.6	32.8	41	kHz
			Crystal OSC.	–	32.768	–	
Supply Current	Idd1	Sleep mode	VDD=1.5V	–	–	1	µA
	Idd2	Idle mode	VDD=1.5V RC OSC./ Crystal OSC.	–	2	4	
	Idd3	Slow mode	VDD=1.5V, RC OSC./ Crystal OSC.	–	4	6	
	Idd4	Fast mode	VDD=1.5V, F_{HOSC} =200KHz	–	16	22	
	Idd5		VDD=1.5V, F_{HOSC} =300KHz	–	22	32	
	Idd6		VDD=1.5V, F_{HOSC} =500KHz	–	32	52	
Input Voltage	VIH1	PA [0:7], PB [0:7], PD[0:7] , PE[0:7] (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

(Continuation)

Parameter	Sym.	Condition		Min.	Typ.	Max.	Unit
Large Pull-up Resistance	RPU5	RSTB	Vin=GND	400	640	1000	KΩ
Small Pull-up Resistance	RPU6	RSTB	Vin=1V	12	40	80	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], PD[0:7] , PE[0:7] (as general output port)	VDD=1.5V, VOH=1.2V	-0.7	-0.9	-1.3	mA
	IOL1		VDD=1.5V, VOL=0.2V	0.7	0.9	1.3	
Large Pull-up Resistance	RPU1	PA[0:7]	Key high resistance, pulled up by R2, Vin2=0.5V	180	280	430	KΩ
	RPU3	PB[0:7], PD[0:7] , PE[0:7]	Vin=0.5V	200	320	500	
Small Pull-up Resistance	RPU2	PA[0:6]	Key high resistance, pulled up by R2//R1, Vin2=0 V	12	18	26	KΩ
	RPU4	PA[7]	Vin=1V	40	60	95	
Data retention voltage	Vret			1.0	–	–	V
Power on reset voltage	Vpor			1.0	1.08	1.15	

9.2 VDD=3.0V Electrical Characteristics

■ Absolute Maximum Ratings

Items	Sym.	Condition	Limits	Unit
Supply voltage	VDD		–0.3 to +3.6	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		2.4 to 3.6	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= 3.0V)

Parameter	Sym.	Condition		Min	Typ	Max	Unit
CLOCK	F_{HOSC}	Main-clock frequency	RC OSC., R=1.8MΩ	140	200	260	KHz
			RC OSC., R=1.2MΩ	210	300	390	
			RC OSC., R=680kΩ	350	500	650	
			RC OSC., R=330KΩ	0.7	1	1.3	MHz
			RC OSC., R=220kΩ	1.05	1.5	1.95	
			RC OSC., R=150kΩ	1.4	2	2.6	
	F_{OSC}	Sub-clock frequency	Internal RC OSC.	24.6	32.8	41	KHz
			Crystal OSC.	–	32.768	–	
Supply Current	Idd1	SLEEP mode	VDD=3V	–	–	1	µA
	Idd2	IDLE mode	VDD=3V, RC OSC./ Crystal OSC.	–	8	12	
	Idd3	SLOW mode	VDD=3V, RC OSC/ Crystal OSC.	–	10	14	
	Idd4	FAST mode	VDD=3V, F_{HOSC} =200KHz	–	40	50	
	Idd5		VDD=3V, F_{HOSC} =300KHz	–	60	70	
	Idd6		VDD=3V, F_{HOSC} =500KHz	–	90	110	
	Idd7		VDD=3V, F_{HOSC} =1MHz	–	140	170	
	Idd8		VDD=3V, F_{HOSC} =1.5MHz	–	210	255	
	Idd9		VDD=3V, F_{HOSC} =2MHz	–	280	340	

(Continuation)

Parameter	Sym.	Condition		Min	Typ	Max	Unit
Input Voltage	VIH1	PA[0:7], PB[0:7], PD[0:7], PE[0:7] (as general input port)		VDD×0.7	-	VDD	V
	VIL1			0	-	VDD×0.3	
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], PD[0:7] , PE[0:7] (as general output port)	VDD=3V, VOH=2.7V	-2.0	-2.6	-3.2	mA
	IOL1		VDD=3V, VOL=0.2V	1.4	1.8	2.3	
Large Pull-up Resistance	RPU1	PA[0:7]	Key high resistance, pulled-up by R2, Vin2=0.5V	180	280	430	KΩ
	RPU3	PB[0:7], PD[0:7], PE[0:7]	Vin=0.5V	200	320	500	
	RPU5	RSTB	Vin=GND	60	100	150	
Small Pull-up Resistance	RPU2	PA[0:6]	Key high resistance, pulled up by R2//R1, Vin2=0 V	12	18	26	KΩ
	RPU4	PA[7]	Vin=2V	40	60	95	
	RPU6	RSTB	Vin=2V	5	9	13	
Large Pull-down Resistance	RPD1	TEST	Vin=VDD	110	160	240	KΩ
Small Pull-down Resistance	RPD2	TEST	Vin=1V	2	2.7	3.6	KΩ
Data Retention Voltage	Vret			2.2	–	–	V
Power On Reset Voltage	Vpor			2.2	2.4	2.6	V

10 Application Circuits

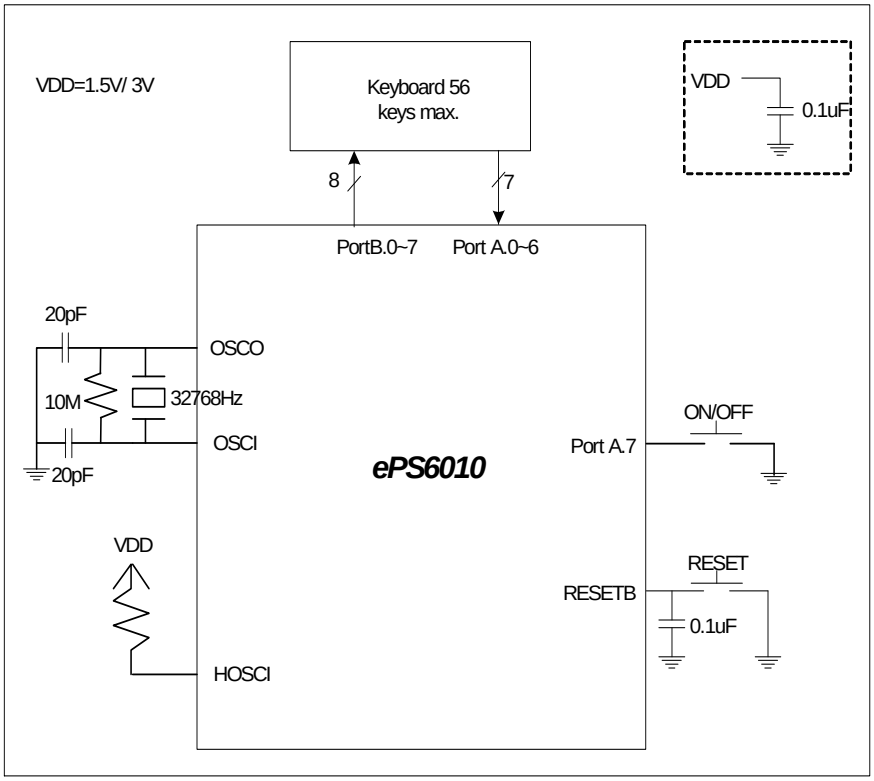


Figure 10-1 Application Circuit Diagram

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 aaaa aaaa aaaa aaaa	LCALL addr (two words)	[Top of Stack] ← PC+1; PC ← addr	None	2
	0010 1011 1111 1110	RET	PC ← (Top of Stack)	None	1
	0010 1011 1111 1111	RETI	PC ← (Top of Stack); Enable Interrupt	None	1
Compare	0010 0101 rrrr rrrr	TEST r	Z ← 0 if r<>0;Z ← 1 if r=0	Z	1
Jump	110a aaaa aaaa aaaa	SJMP addr	PC ← addr PC[13..15] unchanged	None	1
	0000 0000 0010 0000 aaaa aaaa aaaa aaaa	LJMP addr (two words)	PC ← addr	None	2
Compare & Jump	0101 0000 rrrr rrrr aaaa aaaa aaaa aaaa	JDNZ A,r,addr	A ← r-1, jump to addr if not zero PC[15:0] ← addr (*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 112K bytes (56K words).*

*(*3) The maximum jump range is 56K 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

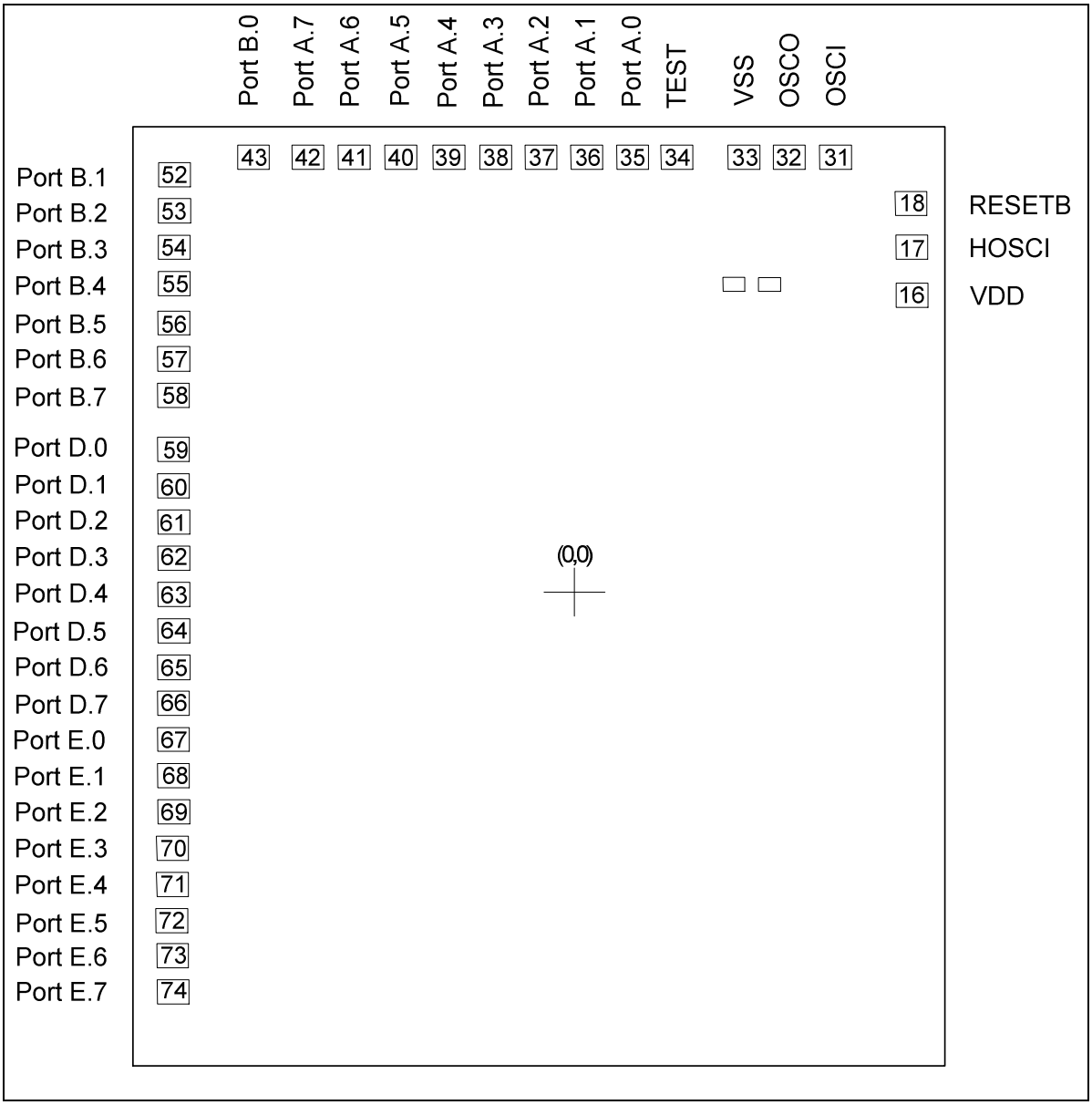


Figure 12-1 ePS6010 39-Pin out Diagram

12.1 Pad Coordinates

Chip Size: 1640 × 2280 μm^2

Pin NO.	Symbol	X	Y
1	NC	-	-
2	NC	-	-
3	NC	-	-
4	NC	-	-
5	NC	-	-
6	NC	-	-
7	NC	-	-
8	NC	-	-
9	NC	-	-
10	NC	-	-
11	NC	-	-
12	NC	-	-
13	NC	-	-
14	NC	-	-
15	NC	-	-
16	VDD	732.478	739.35
17	HOSCI	732.353	868.025
18	RESETB	732.478	973.025
19	NC	-	-
20	NC	-	-
21	NC	-	-
22	NC	-	-
23	NC	-	-
24	NC	-	-
25	NC	-	-
26	NC	-	-
27	NC	-	-
28	NC	-	-
29	NC	-	-
30	NC	-	-
31	OSCI	554.728	1049.525
32	OSCO	456.928	1049.525
33	VSS	355.653	1049.525
34	TEST	237.878	1049.525
35	PortA.0	152.878	1049.525
36	PortA.1	67.878	1049.525
37	PortA.2	-17.122	1049.525
38	PortA.3	-102.122	1049.525
39	PortA.4	-187.122	1049.525
40	PortA.5	-272.122	1049.525

Pin NO.	PortA.6	X	Y
41	PortA.6	-357.122	1049.525
42	PortA.7	-442.122	1049.525
43	PortB.0	-537.322	1049.525
44	NC	-	-
45	NC	-	-
46	NC	-	-
47	NC	-	-
48	NC	-	-
49	NC	-	-
50	NC	-	-
51	NC	-	-
52	PortB.1	-732.497	1009.425
53	PortB.2	-732.497	914.425
54	PortB.3	-732.497	819.425
55	PortB.4	-732.497	729.425
56	PortB.5	-732.497	644.425
57	PortB.6	-732.497	559.425
58	PortB.7	-732.497	474.425
59	PortD.0	-732.497	345.55
60	PortD.1	-732.497	260.55
61	PortD.2	-732.497	175.55
62	PortD.3	-732.497	90.55
63	PortD.4	-732.497	5.55
64	PortD.5	-732.497	-79.45
65	PortD.6	-732.497	-164.45
66	PortD.7	-732.497	-249.45
67	PortE.0	-732.497	-334.45
68	PortE.1	-732.497	-419.45
69	PortE.2	-732.497	-504.45
70	PortE.3	-732.497	-589.45
71	PortE.4	-732.497	-674.45
72	PortE.5	-732.497	-764.45
73	PortE.6	-732.497	-854.45
74	PortE.7	-732.497	-944.45
75	NC	-	-
76	NC	-	-
77	NC	-	-
78	NC	-	-
79	NC	-	-
80	NC	-	-

(Continuation)

Pin NO.	Symbol	X	Y	Pin NO.	Symbol	X	Y
81	NC	-	-	91	NC	-	-
82	NC	-	-	92	NC	-	-
83	NC	-	-	93	NC	-	-
84	NC	-	-	94	NC	-	-
85	NC	-	-	95	NC	-	-
86	NC	-	-	96	NC	-	-
87	NC	-	-	97	NC	-	-
88	NC	-	-	98	NC	-	-
89	NC	-	-	99	NC	-	-
90	NC	-	-	100	NC	-	-

NOTE

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