EPD3338

RISC II Series Microcontroller

Product Specification

DOC. VERSION 1.1

ELAN MICROELECTRONICS CORP.
September 2009

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.

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
elanhk@emc.com.hk

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

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

Shanghai:
Elan Microelectronics Shanghai, Ltd.
#34, First Fl., 2nd Bldg.,
Lane 122, Chunxiao Rd.
Zhangjiang Hi-Tech Park
Shanghai, CHINA 201203
Tel: +86 21 5080-3866
Fax: +86 21 5080-4600

Contents

Specification Revision History

Doc. Version	Revision Description	Date
0.0	Initial version	2002/09/25
0.1	1. Added a Note on Not using TABPTRH (0Dh) Bits 6, 5. 2. Modified Port G and Port H used as segments shared with I/O. 3. Modified the RAM memory size from 2KB to 4KB.	2005/09/09
0.2	1. New Specification 2. Removed the Single-ended reference mode on Bit 3 (S/DB) of ADCON (02Ch). 3. Added code example for Port H and Port G set input.	2005/09/20
	4. Modified PROM memory size 32kW to 64kW. 5. Modified the Pin assignment.	2005/10/15
	6. Modified the Note about FA/D value, which when greater than 1.4MHz is invalid. 7. Modifed the Watchdog Timer Function Block Diagram, added in particular the WDT clock source.	2005/11/03
	8. Added Pad Diagram for EPD3338.	2005/11/04
0.3	Modified the Pin Assignments.	2005/11/22
0.4	Modified the Pin Diagram and QFP_128 package.	2005/12/26

Doc. Version	Revision Description	Date
0.5	1. Modified the operation system frequency to 16MHz max and note the operation limit condition. 2. Added a 16MHz max.clock source for Timer 0. 3. Added a 16MHz max.clock source for Timer 2. 4. Added a 16MHz max. clock source for IR. 5. Added the UART operation frequency to Fpll=14.745MHz. 6. Modified the AD operation frequency to 16MHz max. 7. Modified the SPI operation frequency to 16MHz max. 8. Modified some values of the Electrical characteristics table particularly RPU1, RPU2, RPU3. 9. Modified the single key control method for Port A. 10. Added UART buad rate setting in the Code Option.	2006/02/16
0.6	Modified the LCD RAM Map Tables (Page 00 ~ Page 03) and Note, i.e., RAM Address LCDARL=18~2Fh is Not for use.	2006/03/14
0.7	1. Modified the operation temperature range: -10°C to +70°C 2. Modified the curernt consumption in Idle mode (idd4) with LCD enabled. 3. Added the ADPCM encoder / decoder in Section 2.4, Elan Software Support. 4. Added LCD code option and circuit design method : 1/3 bias small current setting. 5. Removed Bit 2 (DLNPU) and Bit 0 (DLNDC) of DCRDE (R36h).	2006/05/23
0.8	1. Modified and added the code option from 0x000C ~ 0x000F to 0x000C~0x0013. 2. Improved the Speech efficiency played by PWM when the system is from slow/idle/sleep mode. 3. Improved the SEG and COM status (Hi-z) while in soft-key scan during LCD display. 4. Added the Port A control selection on the LCD-BitST code option. 5. Added a high power mode selection on the LCD driver code option.	2006/10/05
0.9	1. Modified the PWM circuit and add the resistor (8~32Ω) connecting to the speaker. 2. Modified the LCD external circuit and add the capacitor (0.1uF,104) between V0 and VR pins. 3. Delete a high power mode selection on the LCD driver code option.	2008/03/18
1.0	Added the following Note: Before entering Sleep mode, the LCD function control must be set : a. The LCDON bit of LCDCONB set to 0 (LCDoff) b. The LCDPM[2:0] bit of LCDCONB set to [000] (LCD power off mode)	2008/05/14
1.1	1. Added a new circuit on the current consumption application. 2. Modified the DC characteristics for Idle 4 LCD enable condition.	2009/09/16

1 General Description

The **EPD3338** is an 8-bit RISC MCU embedded with the following features**:**

- 10-bit SAR A/D converter with touch screen controller
- Analog front end, 32 × 40 LCD driver
- Two 8-bit timers and one 16-bit general timer with capture and event counter functions
- Embedded large size user RAM and program/data memory
- Watchdog timer, SPI, UART, and four melody timers
- PWM or current D/A
- IR generator

The **EPD3338** is ideally suitable for educational learning tools application that requires high performance and low cost solution.

The MCU core is ELAN's second generation RISC (RISC II) based IC. The core is specifically designed as a low power and portable device. It supports Fast, Slow, and Idle mode, as well as Sleep mode for low power consumption application.

IMPORTANT NOTES

- *Do not use Register BSR (05h) Bit 7 ~ Bit 5.*
- *Do not use Register BSR1 (07h) Bit 7 ~ Bit 5.*
- *Do not use Special Register (04h).*
- *Do not use Special Register (1Bh).*
- *Do not use Special Register (1Ch).*
- *Do not use Special Register (1Fh).*
- *Do not use Special Register (2Ah).*
- *Do not use Special Register (32h).*
- *Do not use Special Register (33h).*
- *Do not use Special Register (39h).*
- *Do not use Special Register (4Fh).*
- *Do no use LCD RAM Page 00 (18~2Fh, 40h~4Fh, 68~6Fh).*
- *Do not use LCD RAM Page 01 (18~2Fh, 40h~4Fh, 68~6Fh).*
- *Do not use LCD RAM Page 02 (18~2Fh, 40h~4Fh, 68~6Fh).*
- *Do not use LCD RAM Page 03 (18~2Fh, 40h~4Fh, 68~6Fh).*
- *Do not use LCD RAM Page 04.*
- *Do not use LCD RAM Page 05.*
- *Do not use Port B.3~4.*
- *Do not use Port C.0~1.*
- *Do not use Port D.0~3.*
- *Do not use JDNZ and JINZ at FSR1 (09h) special register.*
- *Do not use Register TABPTRH (0Dh) Bit 6.*
- *Do not to use "MOV A,r" with PUSH,POP to avoid effecting S_Z.*

1.1 Applications

- Educational Learning Tools
- Kids PDA, Kids computer
- Electronics book
- Dictionary, Data Bank

2 Features

2.1 MCU Features

- 8 bit RISC MCU
- 8×8 multiplier with controllable signed or unsigned operation
- Operating voltage and speed: 16~11MHz @ 2.9V~3.6V, 10MHz @ 2.2V~3.6V
- One Instruction cycle time = 2 × System clock time
- Program ROM addressing: 64K words max.
- Data ROM addressing: 256K words max.
- 128 bytes un-banked RAM including special registers and common registers
- 32×128 bytes banked RAM
- RAM stack has a maximum of 128 levels
- Look-up Table function is fast and highly efficient when implemented 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 capability (2 cycles)

2.2 Peripheral

- One input port (Port A) and 32 general I/O pins (Port B.7~5, Port B.2~0, Port C.7~2, Port D.7~4, Port G ~ Port H)
- 4-channel Melody/Speech Synthesizer
- 32 COM × 40 SEG LCD driver embedded
- 16-bit timer (Timer 0) with capture and event counter functions
- 8-bit timer (Timer 1) with wake-up function
- 8-bit timer (Timer 2) as beat counter for Melody function

- 8-bit IR generator
- 8-bit PWM or a current D/A for melody and speech application
- 8-bit Watchdog Timer
- 10 bits resolution SAR A/D converter with 4 channels general analog input and 2 channels for touch panel application
- Key I/O function with 112 keys maximum
- SPI (Serial Peripheral Interface)
- UART (Universal Asynchronous Receiver and Transmitter)

2.3 Internal Specification

- Watchdog Timer with on-chip RC oscillator
- MCU mode: Sleep Mode, Idle Mode, Slow Mode, and Fast Mode
- Supports RC oscillation and crystal oscillation for system clock
- PLL is turned on at Fast Mode, and controlled by PEN bit when the MCU is in Slow Mode and Idle Mode
- MCU wake-up function includes input wake up, Timer 1 wake-up, touch panel wake-up, SPI wake-up, and A/D wake-up
- MCU interrupt function includes Input port interrupt, touch panel interrupt, Capture interrupt, speech timer interrupt, Timer interrupt (Timers 0~2), A/D interrupt, SPI interrupt and UART interrupt
- MCU reset function includes power-on reset, RSTB pin reset, and Watchdog timer reset

2.4 Elan Software Support (Option)

- Hand writing recognition
- LCD display
- 4-channel Melody or 3-channel Melody + 1-channel Speech
- ADPCM decoder
- ADPCM encoder

3 Block Diagram

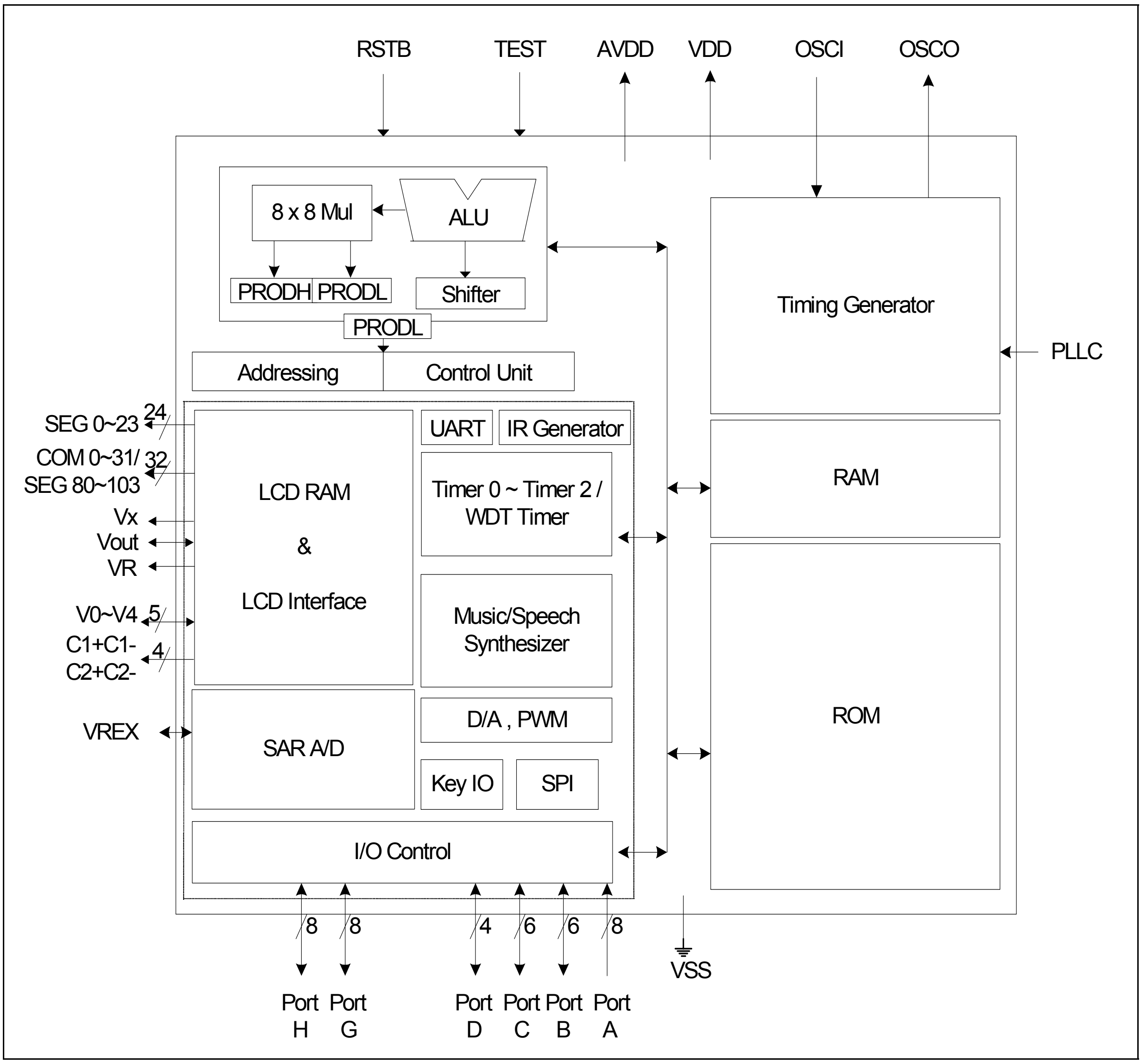
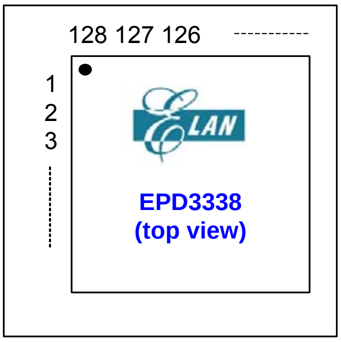


Figure 3-1 EPD3338 Block Diagram

4 Pin Assignment



No.	Pin Name	No.	Pin Name	No.	Pin Name	No.	Pin Name
1	Port A.0	33	Port B.7 (URXD)	65	N.C.	97	COM4
2	Port A.1	34	Port C.2 (ADIN6)	66	N.C.	98	COM5
3	Port A.2	35	Port C.3 (ADIN5)	67	N.C.	99	COM6
4	Port A.3	36	N.C.	68	N.C.	100	N.C.
5	Port A.4	37	N.C.	69	SEG23	101	N.C.
6	Port A.5	38	N.C.	70	SEG22	102	N.C.
7	Port A.6	39	Port C.4 (ADIN4/YP)	71	SEG21	103	COM7
8	Port A.7	40	Port C.5 (ADIN3/XP)	72	SEG20	104	COM8/SEG103
9	VOUT	41	Port C.6 (YN)	73	SEG19	105	COM9/SEG102
10	C1+	42	Port C.7 (XN)	74	SEG18	106	COM10/SEG101
11	C1-	43	VREX	75	SEG17	107	COM11/SEG100
12	C2+	44	AVDD	76	SEG16	108	COM12/SEG99
13	C2-	45	Port D.7 (SPISDI)	77	SEG15 (Strobe 15)	109	COM13/SEG98
14	Vx	46	Port D.6 (SPISDO)	78	SEG14 (Strobe 14)	110	COM14/SEG97
15	VR	47	Port D.5 (SPISCK)	79	SEG13 (Strobe 13)	111	COM15/SEG96
16	V0	48	Port D.4 (SPISS)	80	SEG12 (Strobe 12)	112	COM16/SEG80
17	V1	49	Port H.7 (SEG63)	81	SEG11 (Strobe 11)	113	COM17/SEG81
18	V2	50	Port H.6 (SEG62)	82	SEG10 (Strobe 10)	114	COM18/SEG82
19	V3	51	Port H.5 (SEG61)	83	SEG9 (Strobe 9)	115	COM19/SEG83
20	V4	52	Port H.4 (SEG60)	84	SEG8 (Strobe 8)	116	COM20/SEG84
21	TEST	53	Port H.3 (SEG59)	85	SEG7 (Strobe 7)	117	COM21/SEG85
22	PLLC	54	Port H.2 (SEG58)	86	SEG6 (Strobe 6)	118	COM22/SEG86
23	OSCI	55	Port H.1 (SEG57)	87	SEG5 (Strobe 5)	119	COM23/SEG87
24	OSCO	56	Port H.0 (SEG56)	88	SEG4 (Strobe 4)	120	COM24/SEG88
25	RSTB	57	Port G.7 (SEG55)	89	SEG3 (Strobe 3)	121	COM25/SEG89
26	VDD	58	Port G.6 (SEG54)	90	SEG2 (Strobe 2)	122	COM26/SEG90
27	Port B.0 (VO2)	59	Port G.5 (SEG53)	91	SEG1 (Strobe 1)	123	COM27/SEG91
28	Port B.1 (VO1/DAO)	60	Port G.4 (SEG52)	92	SEG0 (Strobe 0)	124	COM28/SEG92
29	VSS	61	Port G.3 (SEG51)	93	COM0	125	COM29/SEG93
30	Port B.2 (IROT)	62	Port G.2 (SEG50)	94	COM1	126	COM30/SEG94
31	Port B.5 (EVIN/CPIN)	63	Port G.1 (SEG49)	95	COM2	127	COM31/SEG95
32	Port B.6 (UTXD)	64	Port G.0 (SEG48)	96	COM3	128	N.C.

5 Pin Description

5.1 MCU System Pins (9 Pins)

Name	I/O/P Type	Description
AVDD VSS	P	Analog positive power supply ranging from 2.2V~3.6V. Connect to VSS through the capacitors (0.1µF)
VDD VSS	P	Digital and Analog positive power supply, ranging from 2.2V~3.6V. Connect to VSS through the capacitors (0.1µF)
RSTB	I	System reset input with built-in pull-up resistor (100KΩ Typical) Low: RESET asserted High: RESET released
TEST	I	Normally connected to VSS. Reserved for testing use.
OSCI/RC OSCO	I O	RC or Crystal selection by Code Option 32768 Hz oscillator pins: Connect to VSS through capacitors (20pF) RC oscillator connector pin: Connect to VDD through a resistor (2MΩ).
PLLC	I	PLL capacitor connector pin: Connect to VSS through a capacitor (0.047µF).
VREX	I/O	External or internal reference voltage for A/D converter: Connect to VSS through a capacitor (0.1µF).

5.2 Embedded LCD Pins (68 Pins)

Name	I/O/P Type	Description
COM0~COM7	O	LCD common/segment signal output pin.
COM8~COM31/ SEG103~SEG80	O	LCD common/segment signal output pin. Multiplexed: Common and segment pins.
SEG0~SEG23	O	LCD segment signal output pins. (SEG0~SEG15 are shared with key strobe)
Vx	-	Clamping circuit output voltage. Ext. C (0.1µF) to VSS
Vout	-	Charge pump output voltage. Ext. C (0.22µF) to VSS
VR	-	V0 voltage adjusting pin
V0~V4	O	LCD bias pin. Ext. C (0.1µF) to VSS
C1+, C1-, C2+, C2-	-	Charge pump capacitor (0.1µF) 2 times pumping (charge pump capacitor C1:0.1µF) 3 times pumping (charge pump capacitor C1 & C2:0.1µF)

5.3 I/O Ports (40 Pins)

Name	I/O/P Type	Description
Port A	I	General Input port for special functions, i.e., Wake-up and Interrupt Bit 7: ON key input Bits 6~0: Key matrix input pins
Port B (7~5, 2~0)	I/O I O I O O O	General Input/Output port Bit 7: UART Rx pin Bit 6: UART Tx pin Bit 5: Event counter/Capture input pin Bit 2: IR output pin Bit 1: PWM or Current D/A output pin Bit 0: PWM output pin
Port C (7~2)	I/O O O I I I I	General Input/Output port Bit 7: Touch screen X direction negative pin Bit 6: Touch screen Y direction negative pin Bit 5: Touch screen X direction positive pin and A/D input Channel 3 Bit 4: Touch screen Y direction positive pin and A/D input Channel 4 Bit 3: A/D input Channel 5 Bit 2: A/D input Channel 6
Port D (7~4)	I/O I O I/O I	General Input/Output port Bit 7: Serial data input pin Bit 6: Serial data output pin Bit 5: Serial clock Input/Output pin Bit 4: /Slave Select pin
Port G	I/O O	General Input/Output port SEG 55~48: LCD segment signal output pins
Port H	I/O O	General Input/Output port SEG 63~56: LCD segment signal output pins

6 Code Option

The Code Options are located at address 0x000C~0x0013 of the Program ROM:

- Oscillator (OSCSEL): “RC” oscillator
  “Crystal” oscillator **(Default)**
- Initial mode after reset: “Slow” mode
  “Fast” mode (**Default**)
- Port C.7 function selection bit: “XN for touch panel”
  “General I/O function” (**Default**)
- Port C.6 function selection bit: “YN for touch panel”
  “General I/O function” (**Default**)
- Port C.5 function selection bit: “XP for touch panel/ADIN3”
  “General I/O function” (**Default**)
- Port C.4 function selection bit: “YP for touch panel/ADIN4”
  “General I/O function” (**Default**)
- Port C.3 function selection bit: “ADIN5”
  “General I/O function” (**Default**)
- Port C.2 function selection bit: “ADIN6”
  “General I/O function” (**Default**)
- DAC and PWM function selection bits:

DAC or PWM Function Selection	Port B.0 and Port B.1 Function
DAC is used	Port B.1 is DAO for D/A, Port B.0 is General I/O
PWM is used	Port B.1 is VO1 and Port B.0 is VO2 for PWM
DAC and PWM are prohibited for use	General I/O **(Default)**

- Duty Ratio: Maximum Duty Ratio Option (**1/32, Default**)

COM 8-15/SEG 103-96; COM 16-23/SEG 87-80, and COM 24-31/SEG 95-88 status setting**:**

Duty Ratio	Display Size (max.)	Common Driver Used			
		COM 0~7	COM 8~15/ SEG 103~96	COM 16~23/ SEG 87~80	COM 24~31/ SEG 95~88
1/4	64 × 4	COM 0~7	SEG103~96	SEG87~80	SEG95~88
1/8	64 × 8	COM 0~7	SEG103~96	SEG87~80	SEG95~88
1/9	56 × 9	COM 0~15		SEG87~80	SEG95~88
1/11	56 × 11	COM 0~15		SEG 87~80	SEG 95~88
1/16	56 × 16	COM 0~15		SEG 87~80	SEG 95~88
1/24	48 × 24	COM 0~23			SEG 95~88
1/32	40 × 32	COM 0~31			

■	V1; V2; V3 & V4 OP Buffer:	"ON (Normal current)" (**Default**) "OFF"
■	V0 OP buffer control bit:	"V0 OP buffer turn off" "V0 OP buffer turn on" (**Default**)
■	Port G low nibble control bits:	"LCD segment signal output" (**Default**) "General I/O function"
■	Port G high nibble control bits:	"LCD segment signal output" (**Default**) "General I/O function"
■	Port H low nibble control bits:	"LCD segment signal output" (**Default**) "General I/O function"
■	Port H high nibble control bits:	"LCD segment signal output" (**Default**) "General I/O function"
■	Select UART standard baud rate:	"PLL frequency is 9.83MHz" (**Default**) "PLL frequency is 14.745MHz"
■	Read Port A [6:0] control bit:	"Port A connected with LCD-BitST" "Port A divided from LCD-BitST" (**Default**)

> **NOTE**
> *The Code Option Read Port A control can only be used on PMEPD38 or PMEPD33B PM board.*

7 Function Description

7.1 Reset Function

A reset can be caused by:

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

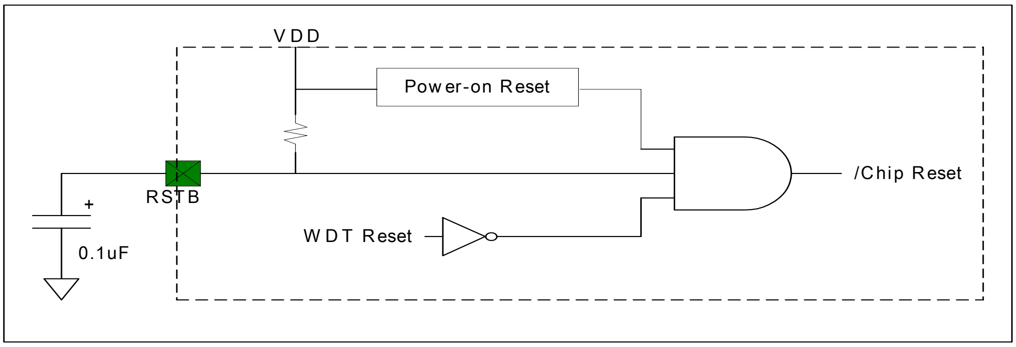


Figure 7-1 On-chip Reset Circuit

7.1.1 Power-up and Reset Timing

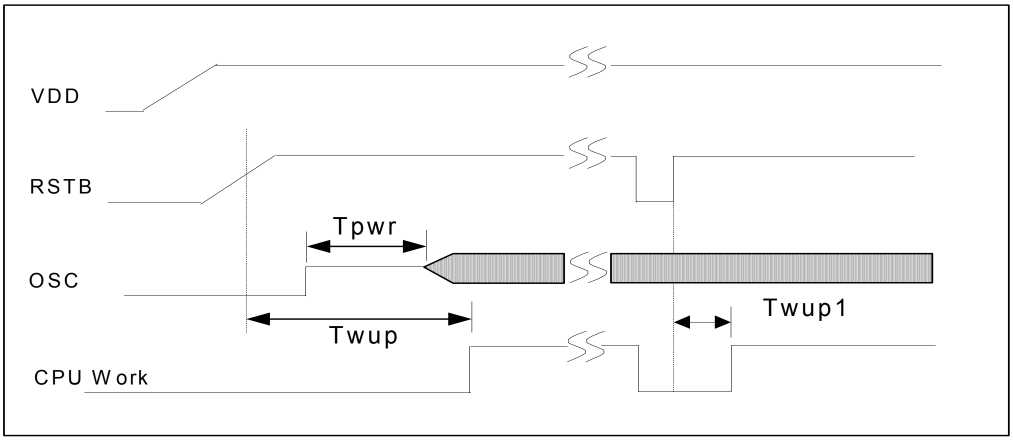


Figure 7-2 Power-up and Reset Timing

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

- **Status (R0Fh)**

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

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

Bit 1 (DC): Auxiliary carry flag

Bit 2 (Z): Zero flag

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

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

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

NOTE

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

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

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

After instruction: ACC = 0D0h
SGE=1, means the result is greater than or equal to 0 (positive value)
OV=1, means the result is carried into a signed bit (Bit 7), overflow occurs.

Correct the signed bit: ACC = 50h (Clear the signed bit)
The actual result= +80h (OV=1) + 50h = +0D0h

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

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

After instruction: ACC = 40h

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

OV=1, means the result is borrow from a signed bit (Bit 7), underflow occurs.

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

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

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

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

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

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

7.1.2 Register Initial Values

- **Special Register:**

Addr.	Name	Initial Value	Addr.	Name	Initial Value
00h	INDF0	–––– ––––[1]	10h	TRL2	uuuu uuuu
01h	FSR0	0000 0000	11h	PRODL	uuuu uuuu
02h	PCL	0000 0000	12h	PRODH	uuuu uuuu
03h	PCM	0000 0000	13h	ADOTL	0–0– –0uu
04h	(Reserved)	–––– ––––	14h	ADOTH	uuuu uuuu
05h	BSR	–––0 0000	15h	UARTTX	xxxx xxxx
06h	STKPTR	0000 0000	16h	UARTRX	xxxx xxxx
07h	BSR1	–––0 0000	17h	Port A	xxxx xxxx
08h	INDF1	–––– ––––[1]	18h	Port B	xxx– –xxx
09h	FSR1	1000 0000	19h	Port C	xxxx xx––
0Ah	ACC	xxxx xxxx	1Ah	Port D	xxxx ––––
0Bh	TABPTRL	0000 0000	1Bh	Reserved	–––– ––––
0Ch	TABPTRM	0000 0000	1Ch	Reserved	–––– ––––
0Dh	TABPTRH	0–00 0000	1Dh	Port G	xxxx xxxx
0Eh	CPUCON	0––0 000c [2]	1Eh	Port H	xxxx xxxx
0Fh	STATUS	cuxx xxxx [3]	1Fh	Reserved	–––– ––––

■ Control Register

Addr.	Name	Initial Value	Addr.	Name	Initial Value
20h	PFS	0010 0000	3Bh	PCCON	0000 0000
21h	STBCON	0000 0000	3Ch	PLLF	×××× ××××
22h	INTCON	0000 0000	3Dh	T0CL	0000 0000
23h	INTSTA	0000 0000	3Eh	T0CH	0000 0000
24h	TRL0L	uuuu uuuu	3Fh	SPICON	0000 0000
25h	TRL0H	uuuu uuuu	40h	SPISTA	––00 0000
26h	TRL1	uuuu uuuu	41h	SPRL	×××× ××××
27h	TR01CON	0000 0000	42h	SPRM	×××× ××××
28h	TR2CON	0000 0000	43h	SPRH	×××× ××××
29h	TRLIR	uuuu uuuu	44h	SFCR	0000 0000
2Ah	Reserved	–––– ––––	45h	ADDL1~ADDL4	×××× ××××
2Bh	POST_ID	–111 –000	46h	ADDM1~ADDM4	×××× ××××
2Ch	ADCON	0101 0000	47h	ADDH1~ADDH4	×××× ××××
2Dh	PAINTEN	0000 0000	48h	ENV1~4 / SPHDR	0000 0000 / 0000 0000
2Eh	PAINTSTA	0000 0000	49h	MTCON1~4/SPHTCON	–––– 0000 / ––00 0000
2Fh	PAWAKE	0000 0000	4Ah	MTRL1~4 / SPHTRL	0000 0000 / 0000 0000
30h	UARTCON	0000 0010	4Bh	VOCON	0–00 0111
31h	UARTSTA	0000 0000	4Ch	TR1C	1111 1111
32h	Reserved	–––– ––––	4Dh	TR2C	1111 1111
33h	Reserved	–––– ––––	4Eh	ADCF	uuuu uuuu
34h	DCRB	111– –111	4Fh	Reserved	–––– ––––
35h	DCRC	1111 11––	50h	LCDCONA	0000 0000
36h	DCRDE	–––– 0–1–	51h	LCDCONB	0–00 0000
37h	DCRFG	0011 ––––	52h	LCDCONC	–000 0000
38h	DCRHI	–––– 0011	53h	LCDARL	0000 0000
39h	Reserved	–––– ––––	54h	LCDDATA	–––– ––––[1]
3Ah	PBCON	0000 0000	55h	PACON	–––– 0110

Legend: **"×"** = unknown **"–"** = unimplemented, read as "0"
"u" = unchanged **"c"** = value depending on the condition

[1] *Not a physical register*

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

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

7.2 Oscillator System

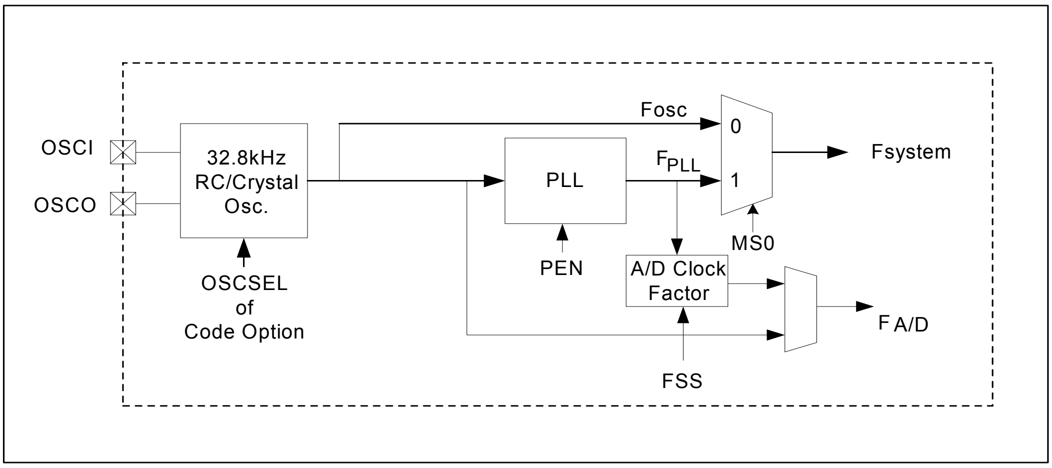


Figure 7-3 Oscillator System Function Block Diagram

7.2.1 32.768kHz Crystal or 32.8kHz RC

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

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

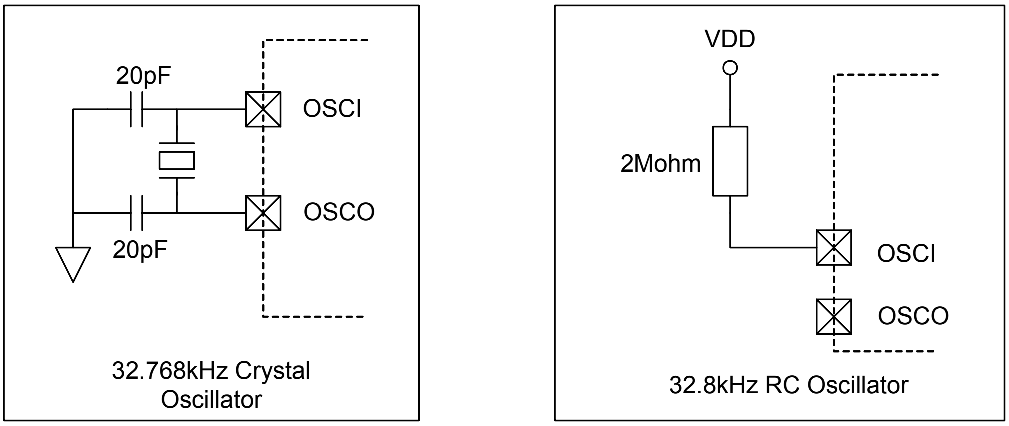


Figure 7-4 Crystal and RC Oscillator Circuit Diagram

7.2.2 Phase-Locked Loop (PLL)

- **PLLF (R3Ch):** Store the actual PLL frequency value. It is used to check whether the PLL frequency is stable or not.

$$F_{actual} = 2 \times PLLF \times F_{OSC}$$

- **PFS (R20h):** Target PLL frequency select register. System clock can be fine tuned from 0.983MHz to 16MHz. The initial value of the PFS register after a chip reset is set at "20h" (FPLL=2.097 MHz)

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

PFS Register	Ftarget (MHz)	PFS Register	Ftarget (MHz)
0~14	N.A.[1]	92	6.029
		107	7.012
15	0.983	122	7.995
31	2.032	137	8.978
46	3.015	150	9.83 [2]
61	3.998	153	10.027
76	4.981	255	16.712

[1] *PFS=0~14 is not available.*

[2] *When UART is enabled, the system clock should be 9.83 MHz (PFS=150) or 14.745 MHz (PFS=225)*

The table is based on 32.768kHz oscillator frequency.

The Maximum range of PLL is from 983kHz ~ 16.712 MHz.

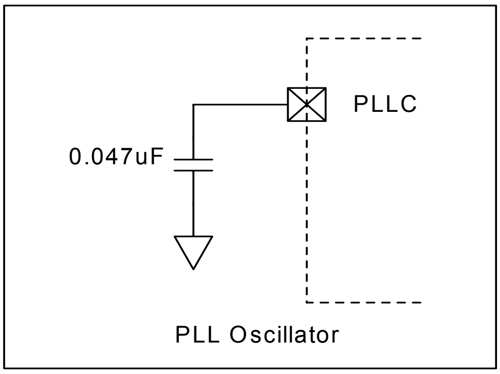


Figure 7-5 PLL Oscillator Circuit Diagram

7.3 MCU Operation Mode

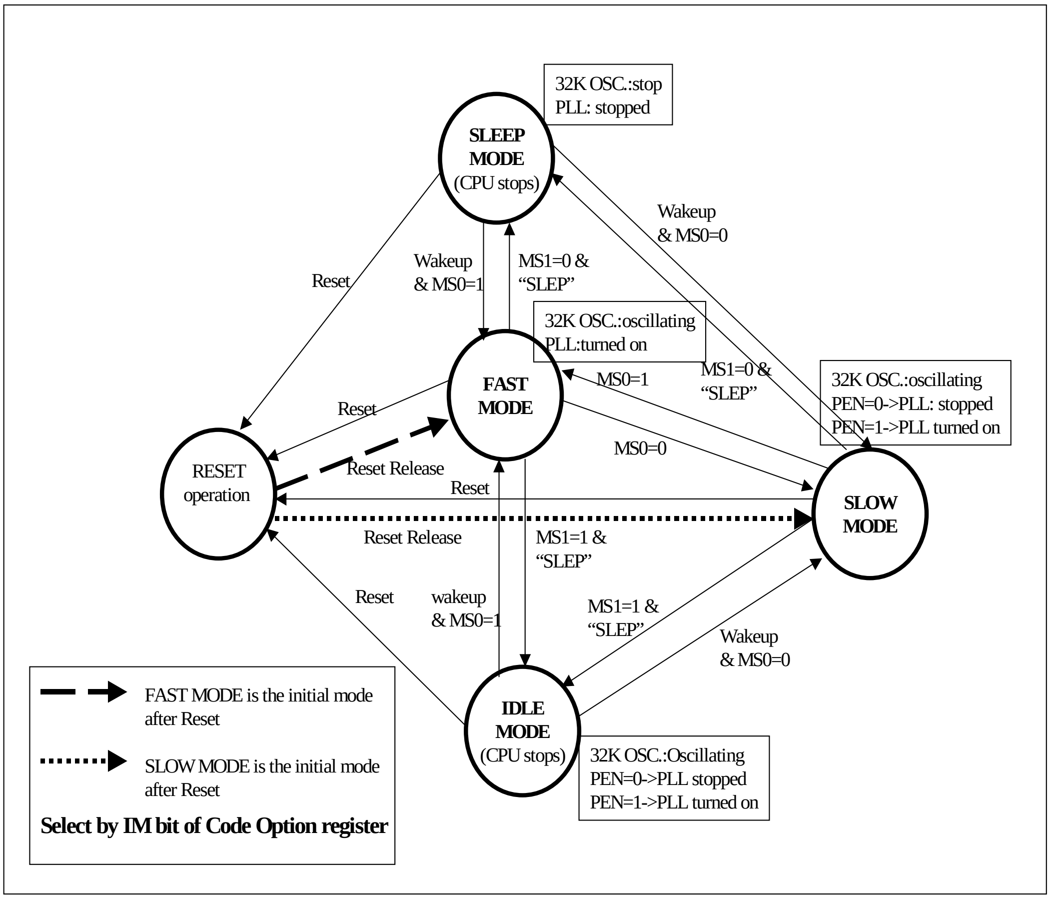


Figure 7-6 Operation Block Diagram

7.3.1 MCU Operation Mode Table

Device \ Mode	Sleep	Idle	Slow	Fast
OSC (32.768kHz)	×	✓	✓	✓
Fsystem	×	×	From OSC	From PLL
PLL	×	✓	✓	✓
A/D conversion	×	✓ [2]	✓	✓
Timers 0~2, IR generator	×	✓	✓	✓
INT	× [1]	× [1]	✓	✓
SPI	✓ (slave)	✓ (slave)	✓	✓
UART	×	×	×	✓
Melody Synthesizer	×	×	×	✓
PWM, current D/A	×	×	✓	✓

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

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

[2] *It is recommended to operate the A/D converter in Idle mode to lower the noise couple from the MCU clock.*

■ **CPUCON (R0Eh):**

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

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

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

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

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

PLL enable**:** It is only effective when the MCU is in Idle mode or **Slow** mode.

MCU Mode	PEN Bit	PLL On/Off
Sleep	×	Off
Idle/Slow	0	Off
	1	On
Fast	×	On

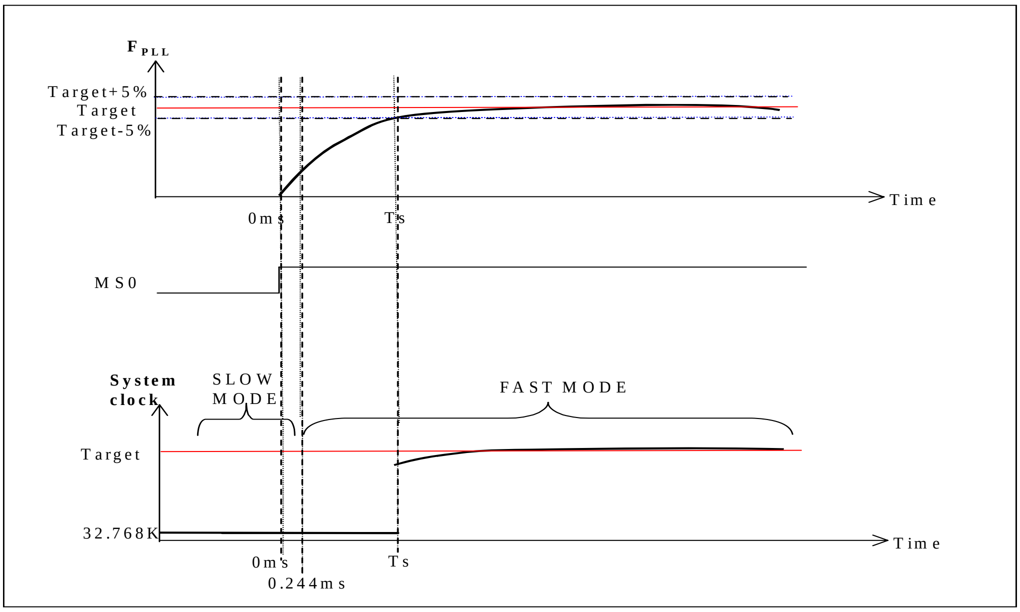


Figure 7-7 MCU Operation Timing Diagram

NOTE

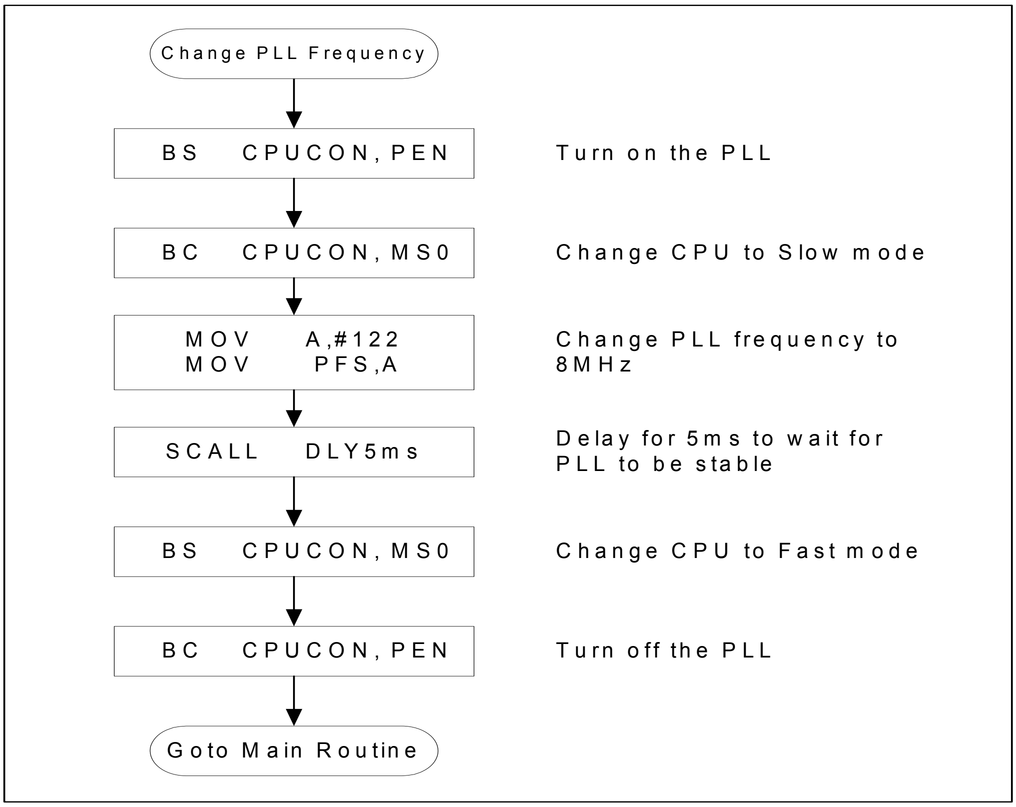
1. *Switch from Slow mode to Fast mode at Time = 0ms*
2. *The System clock will switch to FPLL after 8 oscillation clocks, and the system clock will then increase to about hundreds of kHz.*
3. *The PLL frequency will be stable (± 5%) at Time = Ts (2ms ~ 5ms).*

7.4 Wake-up Function

Device \ Mode	Sleep	Idle	Slow	Fast
I/O wake up	✓	✓	×	×
Touch panel wake up	✓	✓	×	×
Timer1 wake up	×	✓	×	×
A/D wake up	×	✓	×	×
SPI wake up	✓ (Slave)	✓ (Slave)	×	×

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

- **Flowchart:**



- **Code Example:**

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

Entry SLOW mode
     BC        CPUCON,MS0
```

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

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

7.5 Interrupt

When an interrupt occurs, the **GLINT** bit of the CPUCON register is reset to '**0**', which disables all interrupts, including Level 1 ~ Level 5. Setting this bit to '**1**' will enable all un-mask interrupts.

Interrupt Level	Interrupt Source	Start Address	Remark
	RESET	0x00000	
Level 1	Input Port	0x00002	PAINT, PIRQB
Level 2	Capture	0x00004	CPIF
Level 3	Speech Timer	0x00006	SPHTI
Level 4	Timers 0~2	0x00008	TMR0I, TMR1I, TMR2I
Level 5	Peripheral	0x0000A	UERRI, UTXI, URXI, ADIF, SRBFI

- **Code Example:**

```
; ***** Reset program
ResetSEG  CSEG      0X00
    LJMP  MSTART   ;(0X00) Initialize
    LJMP  INPTINT  ;(0X02) Input Port and Touch Panel INT
    LJMP  CAPINT   ;(0X04) Capture Input INT
    LJMP  SPHINT   ;(0X06) Speech Timer INT
    LJMP  TIMERINT ;(0X08) Timer-0,1,2 INT
    LJMP  PERIPH   ;(0X0A) Peripheral INT

PgmSEG    CSEG      0X20
```

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

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

7.5.1 Input Port A Interrupt

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

2. Touch Panel Interrupt (Level trigger)**:** When Port C.7~Port C.4 (X+, X-, Y+ and Y-) are connected to touch panel input pins and the touch panel is touched, PIRQB interrupt occurs.

- **Code Example:**

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

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

7.5.2 Capture Input Interrupt

The Capture function is used to capture an input event at rising to falling edge, falling to rising edge, rising to rising edge, or falling to falling edge. When every event input edge is detected, a Capture interrupt occurs.

■ **Code Example:**

```
; === Capture Input Interrupt
CAPINT:
    S0CALL   PUSH
    JBS      INTSTA,CPIF,toCAPINT
    SJMP     POP
```

```
;---Capture input interrupt
toCAPINT:
     BS      INTSTA,CPIF
      :
     SJMP    POP
```

7.5.3 Speech Timer Interrupt

Speech Timer is an 11-bit timer for time counting. When the counting value of the Speech Timer underflows, an interrupt occurs and the SPHTRL value will be reloaded to counting value.

■ **Code Example:**

```
; === Speech Timer Interrupt
SPHINT:
   S0CALL   PUSH
   JBS      PHTCON,SPHTI,toSPHINT
   SJMP     POP
```

```
; --- To speech timer interrupt
toSPHINT:
    BC      SPHTCON,SPHTI
     :
    SJMP    POP
```

7.5.4 Timer 0, Timer 1, and Timer 2 Interrupts

1. Timer 0 Interrupt: Timer 0 is a 16-bit timer for general time counting. When the counting value is larger than TRL0H : TRL0L value, a Timer 0 interrupt occurs.

2. Timer 1 Interrupt: Timer 1 is an 8 bit-timer for time counting and wake-up function. When the counting value of Timer 1 underflows, an interrupt occurs and the TRL1 value will be reloaded to counting value.

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

■ **Code Example:**

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

7.5.5 Peripheral Interrupt

1. A/D (Analog to Digital converter) Interrupt: A/D is used to convert analog input signal to digital output bits. When the conversion is completed, an A/D interrupt occurs.
2. UERRI Interrupt: UART receiving error interrupt
3. UTXI Interrupt: UART transfer buffer empty interrupt
4. URXI Interrupt: UART receiver buffer full interrupt
5. SRBFI Interrupt: SPI read buffer full interrupt

■ **Code Example:**

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

```
;--A/D interrupt
toADINT:
      BC       INTSTA,ADIF
       :
      SJMP     POP
;--UART Receiving Error Interrupt
toUERRINT:
      BC       INTSTA,UERRI
       :
      SJMP     POP
;--UART Tx Buffer Full Interrupt
toUTXINT:
      BC       INTSTA,UTXI
       :
      SJMP     POP
;--UART Rx Buffer Full Interrupt
toURXINT:
      BC       INTSTA,URXI
       :
      SJMP     POP
;--SPI Interrupt
toSPINT:
      BC       SPISTA,SRBFI
       :
      SJMP     POP
```

7.6 Program ROM Map

8K Words × 8 Segments = 64K Words	
Address	**Segment**
0000h ↓ 000Bh	Interrupt Vector (12 words)
000Ch ↓ 0013h	Code Option (8 words)
0014h ↓ 001Fh	Test Program (12 words)
0020h ↓ 3FFFh	Segment 0 ↓ Segment 1
4000h ↓ 7FFFh	Segment 2 ↓ Segment 3

Address	Segment
8000h ↓ BFFFh	Segment 4 ↓ Segment 5
C000h ↓ FFFFh	Segment 6 ↓ Segment 7

7.7 Data ROM Map

Maximum Size is 256K Words	
Address	
100000h ↓ 13FFFFh	Data ROM (4M bits)

7.8 RAM Map Register

RAM Size: 128 Bytes + 32 Banks × 128 Bytes = 4224 Bytes

■ **Special and Control Register of RAM**

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

Addr.	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	INDF0	R/W							
		Indirect Addressing Pointer 0							
1	FSR0	R/W							
		File Select Register 0 for INDF0							
2	PCL	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
3	PCM	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		PC15	PC14	PC13	PC12	PC11	PC10	PC9	PC8
4	(Reserved)	–							
5	BSR				R/W				
		–			Bank Select Register for INDF0 & General RAM				
6	STKPTR	R/W							
		Stack Pointer							
7	BSR1				R/W				
		–			Bank Select Register 1 for INDF1				
8	INDF1	R/W							
		Indirect Addressing Pointer 1							
9	FSR1	R	R/W						
		1	File Select Register 1 for INDF1						
A	ACC	R/W							
		Accumulator							
B	TABPTRL	R/W							
		Table Pointer Low							

(Continuation)

Addr.	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
C	TABPTRM	R/W							
		Table Pointer Middle							
D	TABPTRH	R/W	–	R/W	R/W	R/W	R/W	R/W	R/W
		Table Pointer High							
E	CPUCON	R/W			R/W	R/W	R/W	R/W	R/W
		PEN	–	–	SMCAND	SMIER	GLINT	MS1	MS0
F	STATUS	R	R	R/W	R/W	R/W	R/W	R/W	R/W
		/TO	/PD	SGE	SLE	OV	Z	DC	C
10	TRL2	R/W							
		Timer 2 Reload Register							
11	PRODL	R/W							
		Multiplier Product Low							
12	PRODH	R/W							
		Multiplier Product High							
13	ADOTL	R/W		R/W			R/W	R	R
		WDTEN	–	ADWKEN	–	–	FSS	ADOT1	ADOT0
14	ADOTH	R	R	R	R	R	R	R	R
		ADOT9	ADOT8	ADOT7	ADOT6	ADOT5	ADOT4	ADOT3	ADOT2
15	UARTTX	W	W	W	W	W	W	W	W
		TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0
16	UARTRX	R	R	R	R	R	R	R	R
		RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
17	Port A	R	R	R	R	R	R	R	R
		A.7	A.6	A.5	A.4	A.3	A.2	A.1	A.0
18	Port B	R/W	R/W	R/W			R/W	R/W	R/W
		B.7	B.6	B.5	–	–	B.2	B.1	B.0
19	Port C	R/W	R/W	R/W	R/W	R/W	R/W		
		C.7	C.6	C.5	C.4	C.3	C.2	–	–
1A	Port D	R/W	R/W	R/W	R/W				
		D.7	D.6	D.5	D.4	–			
1B	Reserved	–							
1C	Reserved	–							
1D	Port G	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		G.7	G.6	G.5	G.4	G.3	G.2	G.1	G.0
1E	Port H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		H.7	H.6	H.5	H.4	H.3	H.2	H.1	H.0
1F	Reserved	–							
20	PFS	R/W							
		Target PLL Frequency Selection Register							
21	STBCON	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		UINVEN	/SCAN	BitST	ALL	STB3	STB2	STB1	STB0
22	INTCON	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		CPIE	ADIE	URXIE	UTXIE	UERRIE	TMR2IE	TMR1IE	TMR0IE

(Continuation)

Addr.	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
23	INTSTA	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		CPIF	ADIF	URXI	UTXI	UERRI	TMR2I	TMR1I	TMR0I
24	TRL0L	R/W							
		Timer 0 Reload Low Byte Register							
25	TRL0H	R/W							
		Timer 0 Reload High Byte Register							
26	TRL1	R/W							
		Timer 1 Reload Register							
27	TR01CON	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		T1WKEN	T1EN	T1PSR1	T1PSR0	IREN	T0CS	T0PSR1	T0PSR0
28	TR2CON	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		IRPSR1	IRPSR0	T0FNEN1	T0FNEN0	T2EN	T2CS	T2PSR1	T2PSR0
29	TRLIR	R/W							
		IR Reload Register							
2A	(Reserved)	–							
2B	POST_ID		R/W	R/W	R/W		R/W	R/W	R/W
		–	LCD_ID	FSR1_ID	FSR0_ID	–	LCDPE	FSR1PE	FSR0PE
2C	ADCON	R/W	R/W	R/W	R		R/W	R/W	R/W
		DET	VRS	ADEN	PIROB	–	CHS2	CHS1	CHS0
2D	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
2E	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
2F	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
30	UARTCON	W	R/W	R/W	R/W	R/W	R/W	R	R/W
		TB8	UMODE1	UMODE0	BRATE2	BRATE1	BRATE0	UTBE	TXE
31	UARTSTA	R	R/W	R/W	R/W	R/W	R/W	R	R/W
		RB8	EVEN	PRE	PRERR	OVERR	FMERR	URBF	RXE
32	(Reserved)	–							
33	(Reserved)	–							
34	DCRB	R/W	R/W	R/W			R/W	R/W	R/W
		Bit7DC	Bit6DC	Bit5DC	–	–	Bit2DC	Bit1DC	Bit0DC
35	DCRC	R/W	R/W	R/W	R/W	R/W	R/W		
		Bit7DC	Bit6DC	Bit5DC	Bit4DC	Bit3DC	Bit2DC	–	–
36	DCRDE					R/W		R/W	
		–				DHNPU	–	DHNDC	–
37	DCRFG	R/W	R/W	R/W	R/W				
		GHNPU	GLNPU	GHNDC	GLNDC	–			
38	DCRHI					R/W	R/W	R/W	R/W
		–				HHNPU	HLNPU	HHNDC	HLNDC
39	(Reserved)	–							
3A	PBCON	R/W	R/W	R/W			R/W	R/W	R/W
		Bit7PU	Bit6PU	Bit5PU	–	–	Bit2PU	Bit1PU	Bit0PU
3B	PCCON	R/W	R/W	R/W	R/W	R/W	R/W		
		Bit7PU	Bit6PU	Bit5PU	Bit4PU	Bit3PU	Bit2PU	–	–

(Continuation)

Addr.	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
3C	PLLF	R							
		Actual PLL Frequency Value Register							
3D	TOCL	R							
		Timer 0 Counting Value Low Byte Register							
3E	TOCH	R							
		Timer 0 Counting Value High Byte Register							
3F	SPICON	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		TLS1	TLS0	BRS2	BRS1	BRS0	EDS	DORD	SE
40	SPISTA			R/W	R/W	R/W	R/W	R/W	R
		–	–	SRBFIE	SRBFI	SPWKEN	SMP	DCOL	RBF
41	SPRL	R/W							
		Shift Register Low Byte of SPI							
42	SPRM	R/W							
		Shift Register Middle Byte of SPI							
43	SPRH	R/W							
		Shift Register High Byte of SPI							
44	SFCR	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		AGMD2	AGMD1	AGMD0	WDTPRS1	WDTPRS0	SPHSB	CSB1	CSB0
45	ADDL	R/W							
		Melody Channels 1~4 Address Low Byte Register							
46	ADDM	R/W							
		Melody Channels1~4 Address Middle Byte Register							
47	ADDH	R/W							
		Melody Channels 1~4 Address High Byte Register							
48	ENV/SPHDR	R/W							
		Melody Channels 1~4 Envelope Register/Speech Data Register							
49	MTCON /SPHTCON	R/W							
		Melody Channels 1~4 Control Register/Speech Control Register							
4A	MTRL/SPHTRL	R/W							
		Melody Channels 1~4 Reload Register/Speech Reload Register							
4B	VOCON	R/W		R/W	R/W	R/W	R/W	R/W	R/W
		VOEN	–	SETR1	SETR0	PWMPSR	VOL2	VOL1	VOL0
4C	TR1C	R							
		Timer 1 Counting Value Register							
4D	TR2C	R							
		Timer 2 Counting Value Register							
4E	ADCF	R/W							
		A/D Clock Factor Register							
4F	(Reserved)	–							

(Continuation)

Addr.	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
50	LCDCONA	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		BSEL2	BSEL1	BSEL0	ADJ4	ADJ3	ADJ2	ADJ1	ADJ0
51	LCDCONB	R/W		R/W	R/W	R/W	R/W	R/W	R/W
		REV	–	LCDON	LCDPM2	LCDPM1	LCDPM0	SFR1	SFR0
52	LCDCONC		R/W	R/W	R/W	R/W	R/W	R/W	R/W
		–	DRSEL2	DRSEL1	DRSEL0	BOOST	Fix 0	LCDARH1	LCDARH0
53	LCDARL	R/W							
		LCD RAM Column Address							
54	LCDDATA	R/W							
		Indirect Register to LCD RAM							
55	PACON					R/W	R/W	R/W	R/W
		–				Bit7PU	/R2EN	/R1EN	KE

■ **Other Un-banked Register of RAM**

Address	Un-banked
56h ⇣ 7Fh	General Purpose RAM

■ **Banked Register of RAM (Selected by BSR)**

Address	Bank 0	Bank 1	Bank 2	Bank 3	……..	Bank 31
80h ⇣ FFh	General Purpose RAM	General Purpose RAM	General Purpose RAM	General Purpose RAM	……..	General Purpose RAM

7.9 LCD RAM Map

LCD RAM	LCDARH [2:0]			
Address	011 (Page 03)	010 (Page 02)	001 (Page 01)	000 (Page 00)
⇣	COM31~COM24	COM23~COM16	COM15~COM8	COM7~COM0
LCDARL	Bit 7 ~ Bit 0	Bit 7 ~ Bit 0	Bit 7 ~ Bit 0	Bit 7 ~ Bit 0
00H (SEG0)				
⇣				
17H (SEG23)				
18h~2Fh	**Not Used**			
30H (SEG48)				
⇣				
3FH (SEG63)				
40H ~ 4FH	**Not Used**			
50H (SEG80)				
⇣				
67H (SEG103)				

Note: *LCDARL = 18F~2F and 40h ~ 4Fh are not used*

7.10 Special Register Description

■ **STKPTR (R06h)**

The stack level starts from the bottom going up (in a decreasing order), starting from 0FFh of Bank 31.

Stack is located at Bank 30 and 31 from address FFh~80h. Initial top position of stack pointer is located at 00h.

Bits 0~6 of STKPTR are used as address pointer from 80h~FFh, Bit 7=1 is used to select Bank 31, Bit 7=0 is used to select Bank 30.

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

■ **PCL, PCM (R02h, R03h): Program Counter Register**

Bit 15	…	Bit 8	Bit 7	…	Bit 0
PCM			PCL		

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

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

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

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

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

■ **Code Example**

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

```
Indirect_JUMP:
    MOV    A,number
    ADD    A,ACC     ;A<-- 2*A
    ADD    PCL,A     ;PCL<-- PCL+A
Function_table:
    LJMP   Function_Address_1  ; number=0
    LJMP   Function_Address_2  ; number=1
    LJMP   Function_Address_3  ; number=2
    LJMP   FUNCTION_ADDRESS_4  ; number=3
    LJMP   Function_Address_5  ; number=4
    LJMP   Function_Address_6  ; number=5
    LJMP   Function_Address_7  ; number=6
    .........
Function_Address_1:
   ........;Function 1 operation

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

- **ACC (R0Ah): Accumulator. Internal Data Transfer, or instruction Operand Holding**

- **POST_ID (R2Bh): Post Increase / Decrease the Control Register**

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

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

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

Bit 4 (FSR0_ID): Setting to '**1**' means auto increase, resetting to '**0**' means auto decrease of FSR0.

Bit 5 (FSR1_ID): Setting to '**1**' means auto increase, resetting to '**0**' means auto decrease of FSR1.

◇ Indirect Addressing Pointer 0

- **BSR (R05h):** is determining which bank is active (working bank) among the 32 banks (Bank 0 ~ Bank 31).
- **FSR0 (R01h):** is an address register for INDF0. User can select up to 256 bytes (Address: 00 ~ 0FFh).
- **INDF0 (R00h):** is not a physically implemented register.

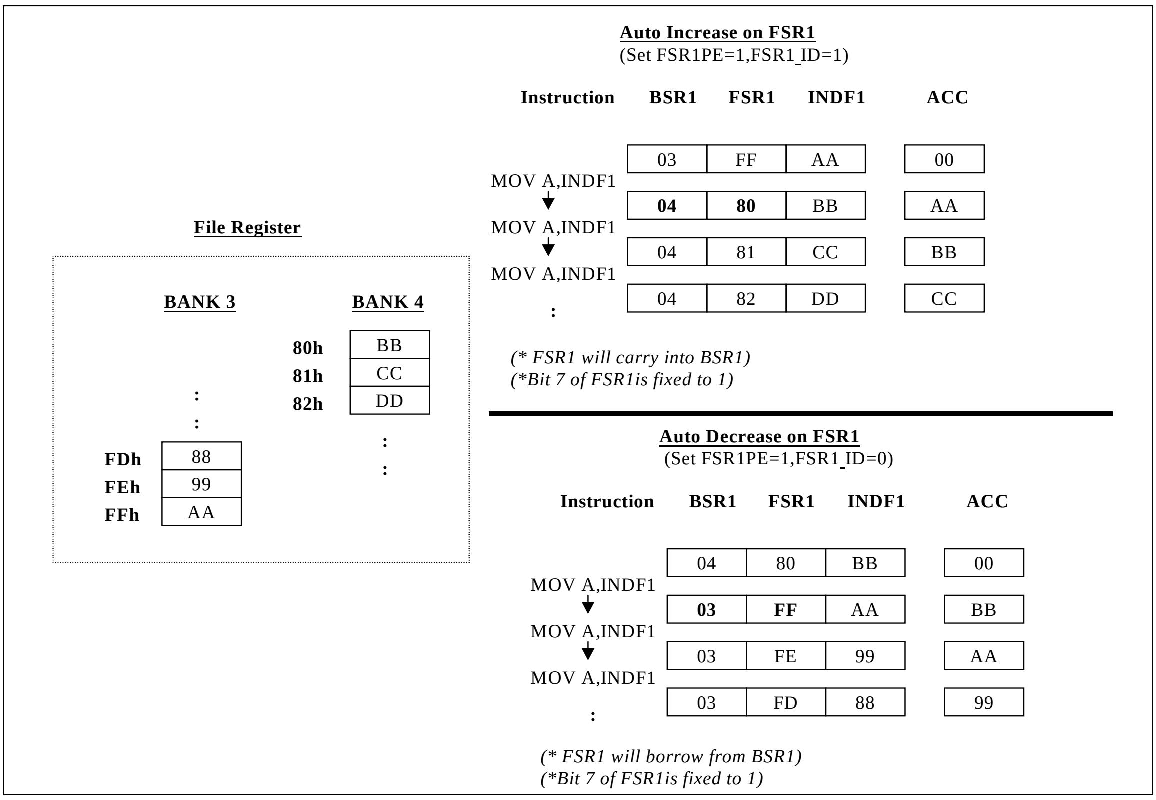
◇ Indirect Addressing Pointer 1

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

◇ 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
     :
```

Linear addressing capability of INDF1 is shown below:



- **Code Example**

```
;********************************************
;*        Const => Working bank setting
;*        REG => Save or Recall register
;********************************************
; *****   RAM stack macro
; *** Initial RAM stack
IniRAMsk  MACRO   #Const
          MOV     A,#Const
          MOV     BSR1,A
          CLR     FSR1
          BS      POST_ID,FSR1PE
          ENDM
;  ***  Push RAM stack
PushRAM   MACRO   REG
          BS      POST_ID,FSR1_ID
          MOVRP   INDF1,REG
          ENDM
; *** Pop RAM stack
PopRAM    MACRO   REG
          BC      POST_ID,FSR1_ID
          MOVPR   REG,INDF1
          ENDM
```

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

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

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

Bit 23	Bit 22	…	Bit 16	Bit15	…	Bit 8	Bit 7	…	Bit 0
-	-	TABPTRH		TABPTRM			TABPTRL		

Program ROM or Internal ROM address register:

Bit 23 is used to select the internal/external memory.

Bit 21 ~ Bit 1 are used to point the memory address.

Bit 0 is used to select the low byte or high byte of the pointed word (see TBRD instruction in Section 11, *Instruction Set*).

◇ **Code Example:**

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

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

- **PRODL, PRODH (R11h, R12h)**: An unsigned or signed 8 × 8 hardware multiplier is included in the microcontroller. The result is stored into the 16 bits product register.

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

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

Bit 3 (SMIER): Signed or unsigned selection bit of the Multiplier. (ACC)
"**0**": Multiplier is unsigned
"**1**": Multiplier is signed

Bit 4 (SMCAND): Signed or unsigned selection bit of the Multiplicand. (Constant or Register)
"**0**": Multiplier is unsigned
"**1**": Multiplier is signed

◇ **Code Example:**

```
; *** Signed multiplier operation
; === PRODH:PRODL = A x REG
      BS            CPUCON,SMIER
      BS            CPUCON,SMCAND
      MUL           A,REG
```

```
; *** Unsigned multiplier operation
; === PRODH:PRODL = A x #k
      BC            CPUCON,SMIER
      BC            CPUCON,SMCAND
      MUL           A,# 88
```

- **STBCON (R21h):** Strobe Output Control Register

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

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

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

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

- **PACON (R55h):** Port A Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	-	-	Bit7PU	/R2EN	/R1EN	KE

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

"**0**": Disable Key input function (Port A register does NOT correspond with Key input in software scan mode).

"**1**": Enable Key input function (Port A register corresponds with Key input in software scan mode).

Bit 1 (/R1EN): R1 pull-up resistor (small resistor) control bit.

"**0**": Enable R1 pull-up resistor

"**1**": Disable R1 pull-up resistor

Bit 2 (/R2EN): R2 pull-up resistor (large resistor) control bit.

"**0**": Enable R2 pull-up resistor

"**1**": Disable R2 pull-up resistor

Bit 3 (Bit 7PU): Enable Port A.7 pull-up resistor.

"**0**": Disable pull-up resistor

"**1**": Enable pull up resistor

- **PAINTEN (R2Dh):** is Port A interrupt control register

"0": Disable interrupt function

"1": Enable interrupt function

- **PAINTSTA (R2Eh):** is Port A interrupt status register

Set to "1" when pin falling edge is detected

Clear to "0" by software

- **PAWAKE (R2Fh):** is Port A wake-up control register

"0": Disable wake-up function

"1": Enable wake-up function

- **Port B.7 ~ 5, Port B.2 ~ 0 (R18h):** are General I/O Registers

- **DCRB (R34h):** Direction Control of Port B

Bit 7 ~ Bit 5, Bit 2 ~ Bit 0 (Bit 7DC ~ Bit 5DC, Bit 2DC ~ Bit 0DC):

"0": Output pin setting
"1": Input pin setting

- **PBCON (R3Ah):** Pull-up Resistor Control of Port B

Bit 7 ~ Bit 5, Bit 2 ~ Bit 0 (Bit 7PU ~ Bit 5PU, Bit 2PU ~ Bit 0PU):

"0": Disable the pull-up resistor
"1": Enable the pull-up resistor

- **Port C.7~2 (R19h):** are General I/O Registers

- **DCRC (R35h):** Direction Control of Port C

Bit 7 ~ Bit 2 (Bit 7DC ~ Bit 2DC):

"0": Output pin setting
"1": Input pin setting

- **PCCON (R3Bh):** Pull-up Resistor Control of Port C.

Bit 7 ~ Bit 2 (Bit 7PU ~ Bit 2PU):

"0": Disable the pull-up resistor
"1": Enable the pull-up resistor

- **Port D 7~4 (R1Ah):** is a General I/O Register

- **DCRDE (R36h):** Direction Control & Pull-up Resistor Control of Port D

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	-	-	DHNPU	-	DHNDC	-

Bit 1 (DHNDC): Port D high nibble direction control

"0": Output pin setting
"1": Input pin setting

Bit 3 (DHNPU): Enable Port D high nibble pull-up resistor

"0": Disable the pull up resistor
"1": Enable the pull-up resistor

- **Port G (R1Dh):** General I/O Register

- **DCRFG (R37h):** Direction Control and Pull-up Resistor Control of Port G

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
GHNPU	GLNPU	GHNDC	GLNDC	-	-	-	-

Bit 5 (GHNDC) and Bit 4 (GLNDC): Port G high / low nibble direction control

"0": Output pin setting
"1": Input pin setting

Bit 7 (GHNPU) and Bit 6 (GLNPU): Enable Port G high / low nibble pull-up resistor

"0": Disable the pull-up resistor
"1": Enable the pull-up resistor

- **Port H (R1Eh):** is a General I/O Register
- **DCRHI (R38h):** Direction Control and Pull-up Resistor Control of Port H

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	-	-	HHNPU	HLNPU	HHNDC	HLNDC

Bit 1 (HHNDC) & Bit 0 (HLNDC): Port H high / low nibbles direction control.

"0": Output pin setting

"1": Input pin setting

Bit 3 (HHNPU) & Bit 2 (HLNPU): Enable Port H high / low nibbles pull up resistor.

"0": Disable pull-up resistor

"1": Enable pull-up resistor

◇ **Code Example:**

```
; *** Port A function
; --- Port A interrupt
INPTINT:
        PUSH
        MOV             A,PAINTSTA
        BC              STBCON,BitST
; --- Port A interrupt
        JBS             STATUS,F_Z,Q_PAINT
        MOV             PORTH,A
Q_PAINT:
        POP
        RETI
; --- Port H output
        CLR             DCRHI
; --- Port A pull-up enable
        MOV             A,#00001001B
        MOV             PACON,A
; --- Port A interrupt
        MOV             A,#11111111B
        MOV             PAINTEN,A
        CLR             PAINTSTA
; --- Port A wakeup
        MOV             PAWAKE,A
        BS              CPUCON,GLINT
; --- Sleep mode
        BC              CPUCON,MS1
KeyLoop:
        BS              STBCON,BitST
        SLEP
        NOP
          :
        SJMP            KeyLoop
```

```
; *** Output function => 0XAAh to all port
        CLR             DCRC
        CLR             DCRB
        CLR             DCRDE
        CLR             DCRFG
        CLR             DCRHI
        MOV             A,#0XAA
        MOV             PORTC,A
        MOV             PORTB,A
        MOV             PORTD,A
        MOV             PORTG,A
        MOV             PORTH,A
;
; *** Input function => Input port to RAM 80 ~ 83h
        BS              POST_ID,FSR1_ID
        BS              POST_ID,FSR1PE
        CLR             BSR1
        CLR             FSR1
        MOV             A,#00001011B
        MOV             PACON,A
        MOV             A,#0XFF
        MOV             DCRB,A
        MOV             PBCON,A
        MOV             DCRC,A
        MOV             PCCON,A
        MOV             DCRDE,A
        MOVRP           INDF1,PORTA
        MOVRP           INDF1,PORTB
        MOVRP           INDF1,PORTC
        MOVRP           INDF1,PORTD
```

8 Peripheral

8.1 Timer 0 (16-bit Timer with Capture and Event Counter Functions)

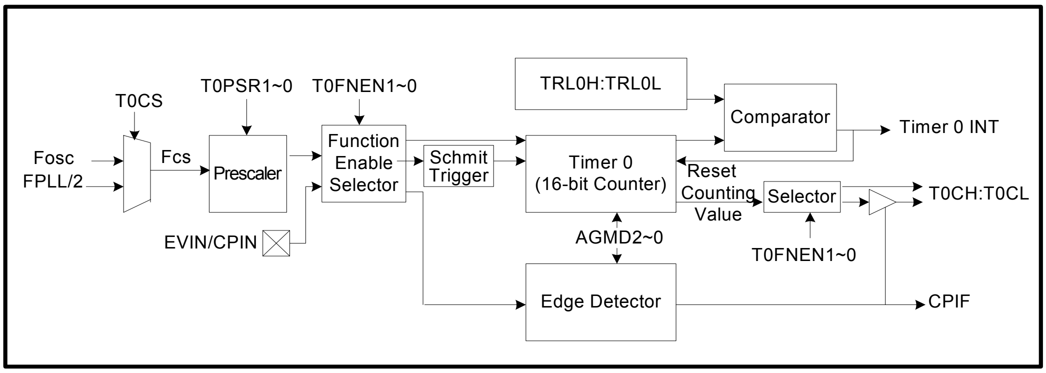


Figure 8-1 Timer 0 Function Block Diagram

8.1.1 Timer 0 Mode

In this mode, Timer 0 is used as a general-purpose 16-bit up counter. There is an interrupt available for your application.

A prescaler is also available for the timer. The T0PSR2~T0PSR0 bits of the TR01CON register determine the prescaler ratio and generate different clock rates for the timer clock source. Counter value will be incremented by one (counting up) according to the timer clock source and stored into the T0CH: T0CL register. The clock source (Fcs) is selected from Fosc or $F_{PLL}/2$ by T0CS and pre-scaled by T0PSR1~0. When the counting value is larger than TRL0H**:** TRL0L value, Timer 0 interrupt will occur, and the counter value will automatically reset to zero.

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

- **Timer 0 Frequency:**

Clock Source	Fper / 2	TRL0H:TRL0L	Prescaler	Timer 0 Freq.
Fosc (32.768kHz)	-	FFFFh	1:64	128Hz
Fpll (8MHz)	4 MHz	00FFh	1:1	15.6kHz
Fpll (16MHz)	8 MHz	00FFh	1:1	31.2kHz

8.1.2 Capture Mode: CPIN (Port B.5) Pin

Capture is a function that captures the Timer 0 value when an event occurs on CPIN pin.

The counter value is captured at: 1st rising edge, 2nd falling edge, etc.; 1st falling edge, 2nd rising edge, etc.; every rising edge or every falling edge selected by AGMD2~0 bit of the SFCR register. When an event edge is detected from the CPIN input pin, the interrupt flag CPIF is set. If a new event edge is detected before the old value in T0CH: T0CL register is read, the old captured value will be lost.

The CPIN pin should be configured in capture function input by setting T0FNEN1~0 bits of TR2CON register.

$$T = \frac{1}{Fcs} \times \mathrm{Prescaler} \times \left[(T0CH : T0CL)_{NEW} - (T0CH : T0CL)_{OLD} \right]$$

- **Capture Mode Example**

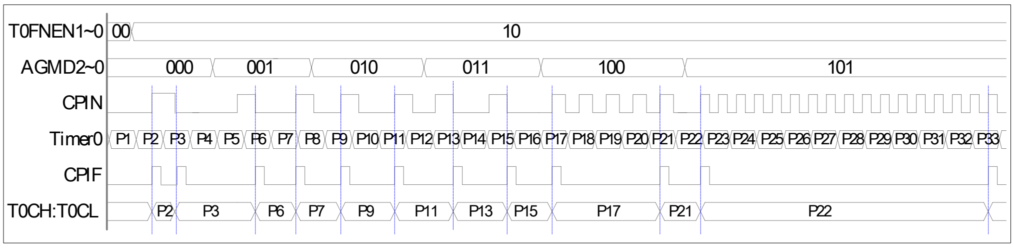


Figure 8-2 Capture Mode Example Timing Diagram

8.1.3 Event Counter Mode: EVIN (Port B.5) Pin

Event counter is a function wherein the 16-bit counter value increments by one when an event occurs on EVIN pin at: every rising edge or every falling edge selected by AGMD2~0 bit of the SFCR register. In other words, the Timer 0 clock source is from an external event (EVIN pin).

The EVIN pin should be configured in event counting function input by setting the T0FNEN1~0 bits of the TR2CON register. The counting value of Timer 0 will be stored in T0CH: T0CL registers.

■ **Event Counter Mode Example:**

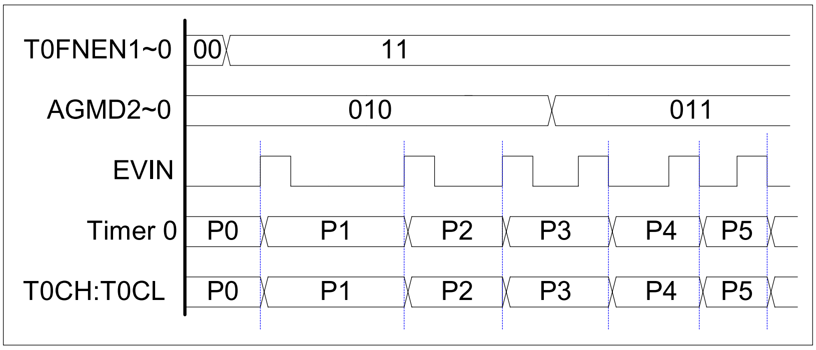


Figure 8-3 Event Counter Mode Example Timing Diagram

8.1.4 Timer 0 Registers Description

■ **TRL0H, TRL0L (R25h, R24h):** Used to store the values compared with Timer 0 register.

■ **T0CH, T0CL (R3Eh, R3Dh):** Used to store the Timer 0 counting value in Timer 0 mode and Event counter mode. But in Capture mode, it is used to store the captured value.

■ **TR01CON (R27h):** Timer 0 and Timer 1 Control Registers

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

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

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

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

"0": Clock source is from Fosc

"1": Clock source is from FPLL/2

■ **TR2CON (R28h):** Timer 2 Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IRPSR1	IRPSR0	T0FNEN1	T0FNEN0	T2EN	T2CS	T2PSR1	T2PSR0

Bit 5 ~ Bit 4 (T0FNEN1 ~ T0FNEN0): Timer 0 and Capture, event counter mode selection bits.

■ **SFCR (R44h):** Special Function Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AGMD2	AGMD1	AGMD0	WDTPSR1	WDTPSR0	SPHSB	CSB1	CSB0

Bit 7 ~ Bit 5 (AGMD2 ~ AGMD0): Capture and Event Counter function edge detector selection bits.

T0FNEN 1 ~ 0	Mode	AGMD 2~0	Edge Mode
00	Disable	-	-
01	Timer 0	-	-
10	Capture	000	1st Rising edge, 2nd falling edge, etc.
		001	1st Falling edge, 2nd rising edge, etc.
		010	Every rising edge
		011	Every falling edge
		100	Every 4th rising edge
		101	Every 16th rising edge
11	Event Counter	010	Every rising edge
		011	Every falling edge

Note: *1. In changing from one mode to another, it is necessary to disable the Timer 0.*
2. To avoid error, simultaneously setup T0FNEN1 and T0FNEN0.

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

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

Bit 2 (GLINT): Global Interrupt Control Bit

"0": Disable all interrupts

"1": Enable all un-masked interrupt

■ **INTCON (R22h):** Interrupt Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CPIE	ADIE	URXIE	UTXIE	UERRIE	TMR2IE	TMR1IE	TMR0IE

Bit 0 (TMR0IE): Timer 0 Interrupt Control Bit

"0": Disable Timer 0 interrupt

"1": Enable Timer 0 interrupt

Bit 7 (CPIE): Capture Interrupt Control bit

"0": Disable Capture interrupt

"1": Enable Capture interrupt

■ **INTSTA (R23h):** Interrupt Status Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CPIF	ADIF	URXI	UTXI	UERRI	TMR2I	TMR1I	TMR0I

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

Bit 7 (CPIF): Set to '**1**' when Capture input edge is detected
Clear to '**0**' by software or disable Capture

■ **Code Example:**

```
;===Timer 0 interrupt
TIMERINT:
      PUSH
      JBC    INTSTA,TMR0I,Q_Time
      BC     INTSTA,TMR0I
      BTG    PORTC,3
Q_Time:
      POP
      RETI
;===Timer 0 = (8M/2) / [4 x 3FFF + 1]
Timer0SR:
        :
      System setting 8MHz
      PC.2 Port H & G setting output port
        :
; --- Fpll & Prescaler 1:4
      MOV    A,#00000101B
      MOV    TR01CON,A
; --- 4ms = (4 x 16383 + 1)/(8M/2)
      MOV    A,#0FFH
      MOV    TRL0L,A
      MOV    A,#03FH
      MOV    TRL0H,A
; --- Timer 0 mode
      MOV    A,#00010000B
      MOV    TR2CON,A
; --- Timer 0 interrupt enable
      BS     INTCON,TMR0IE
; --- Clear Timer 0 interrupt status.
      BC     INTSTA,TMR0I
; --- Enable global interrupt
      BS     CPUCON,GLINT
TimeLoop:
; --- Out Timer 0 count to Port H:G
      MOVRP PORTH,T0CH
      MOVRP PORTG,T0CL
      SJMP   TimeLoop
```

```
;===Capture Input Interrupt
CAPINT:
      PUSH
      JBS    INTSTA,CPIF,Q_ICAP
      BC     INTSTA,CPIF
      BTG    PORTC,3
      BS     INTFLAG,F_ICAP
Q_ICAP:
      POP
      RETI
;
;===1st falling edge,2nd rising edge, etc.
CAP_SR:
      System setting 8MHz
      PC.2 Port H & G setting output port
      User setting F_ICAP flag.
        :
; --- Count end => 0FFFFH
      MOV    A,#0XFF
      MOV    TRL0H,A
      MOV    TRL0L,A
; --- PLL/2 & Prescaler 1:1
; --- (8MHz/2)/65536=61Hz
      MOV    A,#00000100B
      MOV    TR01CON,A
; --- 1st Falling - 2nd Rising
      MOV    A,#00100000B
      MOV    SFCR,A
      BS     INTCON,CPIE
; --- 10->Capture Enable
      MOV    A,#00100000B
      MOV    TR2CON,A
      BC     INTFLAG,F_ICAP
      BS     CPUCON,GLINT
CAP_LOOP:
      JBC    INTFLAG,F_ICAP,CAP_LOOP
      BC     INTFLAG,F_ICAP
; --- Out capture count to Port H:G
      MOVRP PORTH,T0CH
      MOVRP PORTG,T0CL
      SJMP   CAP_LOOP
```

(Continuation)

```
; === Every rising edge
EVcntSR:
        :
      System setting 8MHz
      Port H & G setting output port
        :
      MOV         A,#0XFF          ; Switch 256 times reload
      MOV         TRL0L,A
      CLR         TRL0H            ; Count start 0000H
      BS          TR01CON,T0CS     ; PLL/2
      MOV         A,#01000000B
      MOV         SFCR,A           ; Rising edge
      MOV         A,#00110000B
      MOV         TR2CON,A         ; 11->Event count Enable
EV_LOOP:
      MOVRP       PORTH,T0CH       ; Out event count to Port H:G
      MOVRP       PORTG,T0CL
      SJMP        EV_LOOP
```

8.2 Timer 1 (8 Bits)

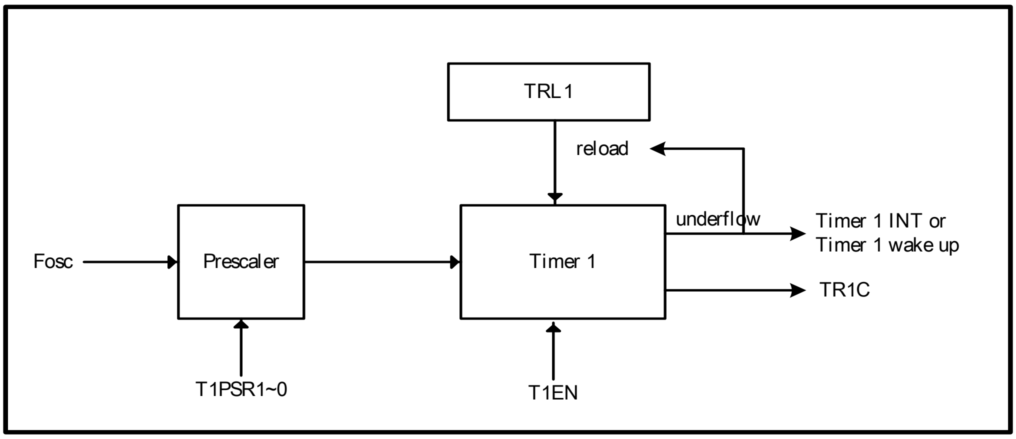


Figure 8-4 Timer 1 Function Block Diagram

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

There is also a prescaler for the timer. The **T1PSR1~T1PSR0** bits of the TR01CON register determine the prescaler ratio and generate different clock rates for the timer clock source. Setting **T1WKEN** bit of the TR01CON register to '**1**' will enable the Timer 1 underflow wake-up function in Idle Mode.

Counting value will be decremented by one (count down) according to the real timer clock source. When the counter underflows, the timer interrupt is triggered if the global interrupt and Timer 1 interrupt are both enabled. At the same time, the TRL1 value will automatically reload into the 8 bits counter.

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

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

8.2.1 Timer 1 Registers Description

- **TRL1 (R26h):** is used to store the auto-reload value of Timer 1. When enabling Timer 1 or an underflow occurs, TRL1 register value will automatically reload into the 8 bits counter.

- **TR1C (R4Ch):** is used to store the Timer 1 Counting Value

■ **TR01CON (R27h):** 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	IREN	T0CS	T0PSR1	T0PSR0

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

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

Bit 6 (T1EN): Timer 1 Enable Control Bit
"0": Disable Timer 1 (stop counting)
"1": Enable Timer 1

Bit 7 (T1WKEN): Enable bit of Timer 1 underflow wake-up function in IDLE MODE.
"0": Disable Timer 1 wake-up function
"1": Enable Timer 1 wake-up function

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

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

Bit 2 (GLINT): Global Interrupt Control Bit

■ **INTCON (R22h):** Interrupt Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CPIE	ADIE	URXIE	UTXIE	UERRIE	TMR2IE	TMR1IE	TMR0IE

Bit 1 (TMR1IE): Timer 1 Interrupt Control Bit
"0": Disable Timer 1 interrupt
"1": Enable Timer 1 interrupt

■ **INTSTA (R23h):** Interrupt Status Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CPIF	ADIF	URXI	UTXI	UERRI	TMR2I	TMR1I	TMR0I

Bit 1 (TMR1I): Set to '**1**' when Timer 1 interrupt occurs
Clear to '**0**' by software or disable Timer 1

■ **Code Example:**

```
; === Timer 1 interrupt
TIMERINT:
      PUSH
      JBC   INTSTA,TMR1I,Q_Time
      BC    INTSTA,TMR1I
      BTG   PORTC,3
Q_Time:
      POP
      RETI
; === Timer 1 = 32.768K/[256 x 3F + 1]
Timer1SR:
       :
      PC.2 setting output port
       :
      MOV   A,#10110000B
      MOV   TR01CON,A          ; Fosc & Prescaler 1:256 & wakeup
      MOV   A,#03FH
      MOV   TRL1,A             ; 0.5sec = (256 x 63 + 1)/32.768K
      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)

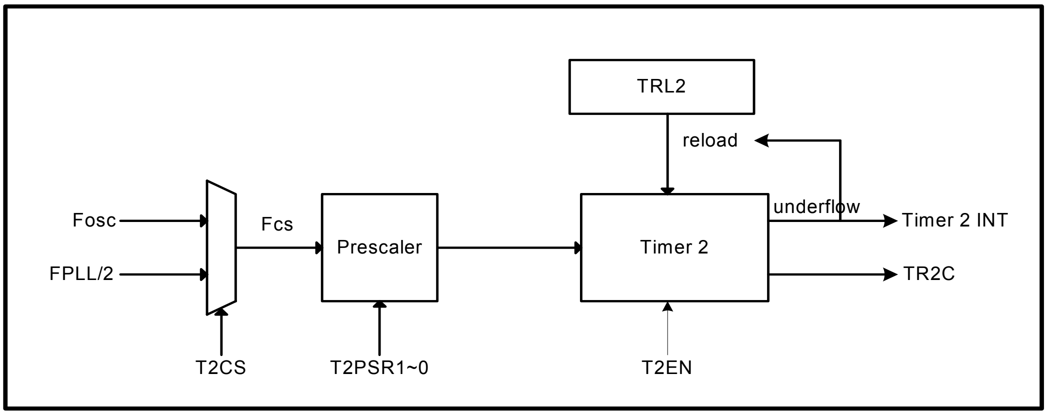


Figure 8-5 Timer 2 Function Block Diagram

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

A prescaler is also available for the timer. The T2PSR1~T2PSR0 bits of the TR2CON register determine the prescaler ratio and generate different clock rates for the timer clock source.

Counting value will be decremented by one (counting down) according to the timer clock source. When the counter value underflows, a timer interrupt will occur (if Timer 2 interrupt is enabled).

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

8.3.1 Timer 2 Registers Description

■ **Timer 2 Frequency:**

Clock Source	Fper / 2	TRL2	Prescaler	Timer 2 Freq.
Fosc (32.768kHz)	-	FFh	1:8	16Hz
Fpll (8 MHz)	4 MHz	0Fh	1:1	250kHz
Fpll (16 MHz)	8 MHz	0Fh	1:1	500kHz

■ **TRL2 (R10h):** is used to store the auto-reload value of Timer 2. When enabling Timer 2 or an underflow occurs, TRL2 register is automatically reloaded into the 8 bits counter.

■ **TR2C (R4Dh):** is used to store the Timer 2 counting value

■ **TR2CON (R28h):** Timer 2 Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IRPSR1	IRPSR0	T0FNEN1	T0FNEN0	T2EN	T2CS	T2PSR1	T2PSR0

Bit 1 ~ Bit 0 (T2PSR1~T2PSR0): Timer 2 Prescaler select bit.

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

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

"0": Clock source is from Fosc
"1": Clock source is from FPLL/2

Bit 3 (T2EN): Timer 2 Enable Control Bit

"0": Disable Timer 2 (stop counting)
"1": Enable Timer 2

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

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

Bit 2 (GLINT): Global Interrupt Control Bit

■ **INTCON (R22h):** Interrupt Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CPIE	ADIE	URXIE	UTXIE	UERRIE	TMR2IE	TMR1IE	TMR0IE

Bit 2 (TMR2IE): Timer 2 Interrupt Control bit
"0": Disable Timer 2 interrupt
"1": Enable Timer 2 interrupt

■ **INTSTA (R23h):** Interrupt Status Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CPIF	ADIF	URXI	UTXI	UERRI	TMR2I	TMR1I	TMR0I

Bit 2 (TMR2I): Set to '**1**' when Timer 2 interrupt occurs
Clear to '**0**' by software or disable Timer 2

■ **Code Example:**

```
; === Timer 2 interrupt
TIMERINT:
      PUSH
      JBC   INTSTA,TMR2I,Q_Time
      BC    INTSTA,TMR2I
      BTG   PORTC,3
Q_Time:
      POP
      RETI
; === Timer 2 = (8M/2)/[4 x 3F + 1]
Timer2SR:
       :
      System setting 8MHz
      Port G setting output port
       :
      MOV   A,#00000110B
      MOV   TR2CON,A           ; Fpll & Prescaler 1:4
      MOV   A,#03FH
      MOV   TRL2,A             ; 16us = (4 x 63 + 1)/(8M/2)
      BS    TR2CON,T2EN        ; Timer 2 enable
      BS    INTCON,TMR2IE      ; Timer 2 interrupt enable
      BC    INTSTA,TMR2I       ; Clear Timer 2 interrupt status
      BS    CPUCON,GLINT       ; Enable global interrupt
TMR2Loop:
      MOVRP PORTH,TR2C         ; Out Timer 2 count to Port H
      SJMP  TMR2Loop
```

8.4 IR Generator: IROT (Port B.2) Pin

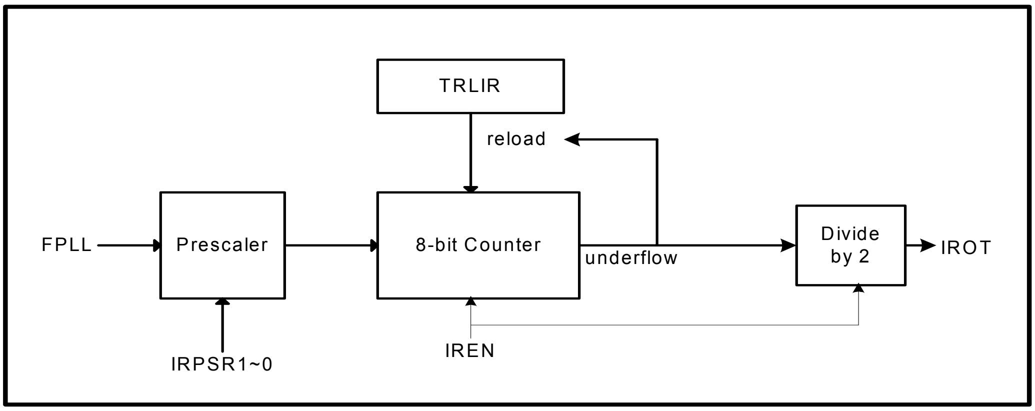


Figure 8-6 IR Generator Function Block Diagram

IR function is enabled by IREN bit and output on the IROT (Port B.2) pin by a general-purpose 8-bit down counter. When IREN is low, the T-flip-flop should be initialized, as IROT equals zero. The clock source is from the PLL clock. The IRPSR1 ~ IRPSR0 bits of the TR2CON register determine the prescaler ratio and generate different clock rates for the timer clock source. The counting value will be decremented by one (counting down) according to the clock source. When the counter value underflows, the IR reload register value will be reloaded into the counter.

$$T = \frac{2}{F_{PLL}} \times Prescaler \times (TRLIR + 1)$$

IR Carrier Signal Frequency:

Clock Source	Fper / 2	TRLIR	Prescaler	IR Freq.
Fpll (8 MHz)	4 MHz	0Fh	1:1	250kHz
Fpll (16 MHz)	8 MHz	0Fh	1:1	500kHz

8.4.1 IR Registers Description

- **TRLIR (R29h):** is used to store the auto-reload value of the IR generator. When enabling the IR generator or when an underflow occurs, the TRLIR register value is automatically reloaded into the 8 bits counter.

- **TR01CON (R27h):** 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	IREN	T0CS	T0PSR1	T0PSR0

Bit 3 (IREN): IR function enable control bit

"0": Disable IR function and recover IROT pin as a general I/O pin

"1": Enable IR function and change Port B.2 as IROT output pin

■ **TR2CON (R28h): Timer 2 Control Register**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IRPSR1	IRPSR0	T0FNEN1	T0FNEN0	T2EN	T2CS	T2PSR1	T2PSR0

Bit 7 ~ Bit 6 (IRPSR1~IRPSR0): IR Generator Prescaler Select Bit

IRPSR1: IRPSR0	Prescaler Value
00	1:1
01	1:4
10	1:16
11	1:64

■ **Code Example:**

```
; === IR generator 31kHz
        :
      System setting 10MHz
        :
      MOV   A,#10000000B
      MOV   TR2CON,A    ; Prescaler 1: 16
      MOV   A,#9
      MOV   TRLIR,A     ; 10MHz /[ 2 x 16 x ( 9 + 1 ) ] = 31kHz
      BS    TR01CON,IREN
IR_Loop:
      SJMP  IR_Loop
```

8.5 Watchdog Timer (WDT)

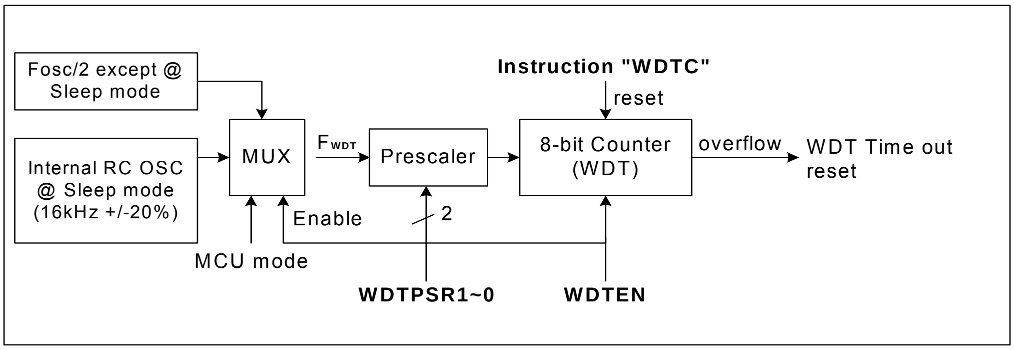


Figure 8-7 Watchdog Timer Function Block Diagram

The watchdog timer (WDT) clock source is from on-chip RC oscillator (16kHz ± 20%, MCU in Sleep mode) or $F_{OSC}/2$ (MCU in Fast, Slow or Idle mode). The WDT will keep on running even when the oscillator has been turned off (i.e., in Sleep Mode). WDT time-out will cause the MCU to reset (if WDT is enabled). To prevent a reset from occurring, you should clear the WDT value by using the “WDTC” instruction before WDT time-out. The WDTEN bit must be set to ‘**1**’ (Enable) as the WDT function is disabled by default. A prescaler that generates different clock rates for the WDT clock source is also available. The prescaler ratio is defined by WDTPSR1 ~ WDTPSR0.

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

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

■ **ADOTL (R13h):** A/D Output Data Low Byte Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WDTEN	-	ADWKEN	-	-	FSS	ADOT1	ADOT0

Bit 7 (WDTEN): Watchdog Timer enable bit.

"0": Disable (stop running) Watchdog Timer

"1": Enable Watchdog Timer

■ **SFCR (R44h):** Special Function Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AGMD2	AGMD1	AGMD0	WDTPSR1	WDTPSR0	SPHSB	CSB1	CSB0

Bit 4 ~ Bit 3 (WDTPSR1~WDTPSR0): Watchdog Timer Prescaler select bit

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

■ **Code Example:**

```
; === WDT setting 2.048sec
    :
  Timer 1 (0.5sec wakeup)
    :
  BS    SFCR,WDTPSR0
  BS    SFCR,WDTPSR1 ; Prescaler 1:128
  BC    CPUCON,MS1   ; Change to sleep mode
  WDTC
  SLEEP
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
```

8.6 Universal Asynchronous Receiver Transmitter (UART)

- RS232C compatible
- Mode selectable (7/8/9-bit) with/without parity bit
- Baud rate selectable
- Error detect function
- Interrupt available for Tx buffer empty, Rx buffer full, and receiver error
- TXD and RXD port inverse output control

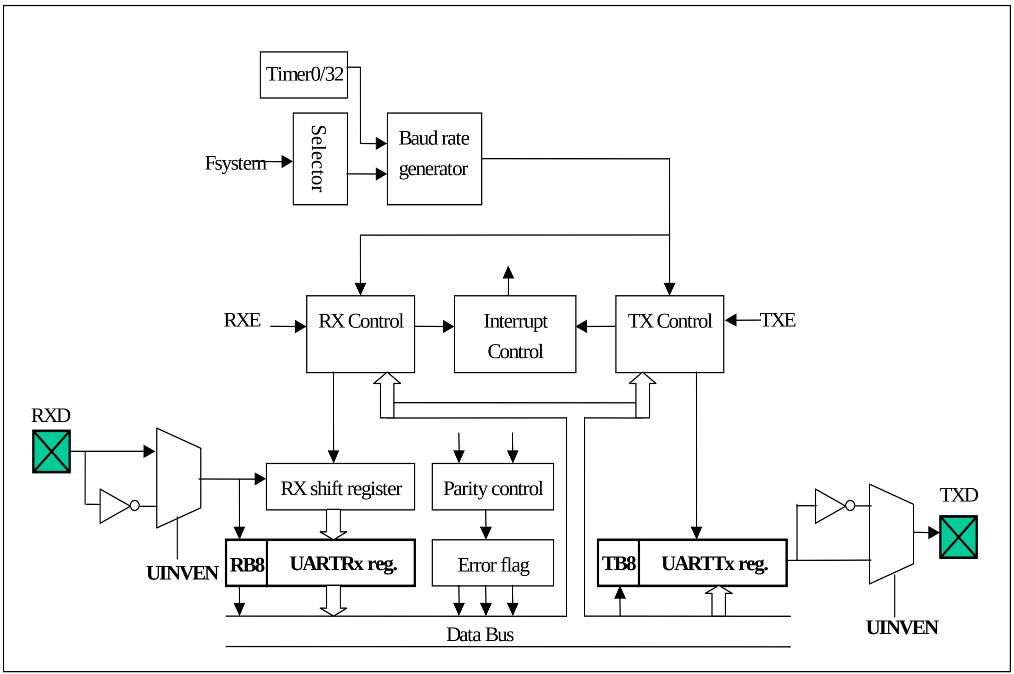


Figure 8-8 UART Function Block Diagram

In Universal Asynchronous Receiver Transmitter (UART), each transmitted or received character is individually synchronized by framing it with a start bit and stop bit.

Full duplex data transfer is possible because the UART has independent transmit and receive sections. Double buffering in both sections enables the UART to be programmed for continuous data transfer.

The figure below shows the general format of one character sent or received. The communication channel is normally held in the marked state (high). Character transmission or reception starts with a transition to the space state (low).

The first bit transmitted or received is the start bit (low). It is followed by the data bits, in which the least significant bit (LSB) comes first. The data bits are followed by the parity bit. If present, then the stop bit or bits (high) confirm the end of the frame.

In receiving, the UART synchronizes on the falling edge of the start bit. When two or more "0's" are detected during three sampling, it is recognized as a normal start bit and receiving operation starts.

8.6.1 Data Format in UART

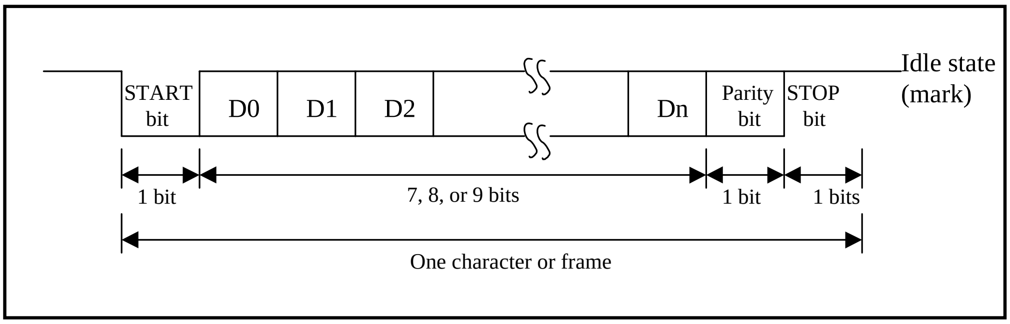


Figure 8-9 UART Data Format Diagram

8.6.2 UART Modes

There are three modes in UART. Mode 1 (7 bits data) and Mode 2 (8 bits data) allow the addition of a parity bit. The parity bit addition is not available in Mode 3. The figure below shows the data format in each mode.

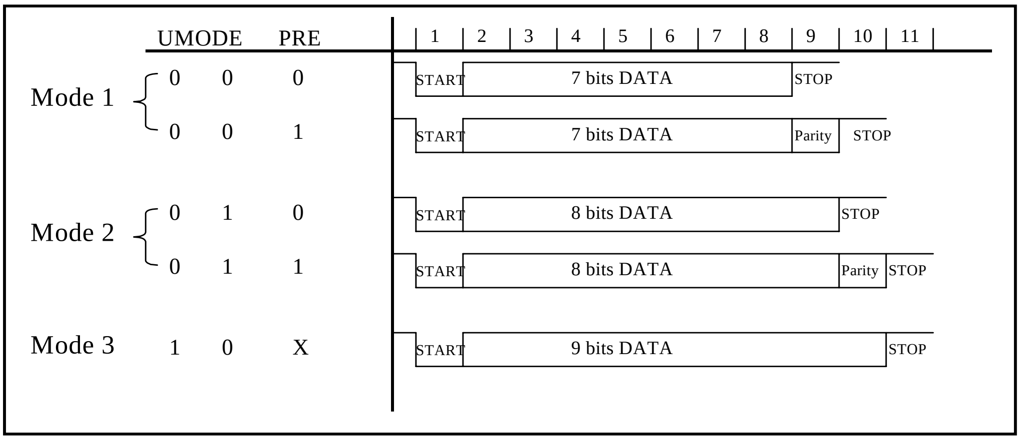


Figure 8-10 UART Modes Data Format

8.6.3 UART Transmit Data

In transmitting serial data, the UART operates as follows:

1. Set the **TXE** bit of the UARTCON register to enable UART transmission function.
2. Write data into the UARTTX register, and the **TBE** bit of the UARTCON register will be set by hardware. Then start transmitting.
3. Serially transmitted data are transmitted in the following order from the TXD pin:
   - (a) Start bit: one “0” bit is output
   - (b) Transmit data: 7, 8, or 9 bits data are output from LSB to MSB
   - (c) Parity bit: one parity bit (odd or even selectable) is output
   - (d) Stop bit: one “1” bit (stop bit) is output
   - (e) Mark state: output “1” continues until the start bit of the next transmit data
4. After transmitting the stop bit, the UART generates a **UTXI** interrupt (if enabled)

8.6.4 UART Receive Data

1. Sets the **RXE** bit of the UARTCON register to enable the UART receiving function.

2. The UART monitors the RXD pin and synchronizes internally when it detects a start bit.

3. Received data is shifted into the UARTRX register in LSB to MSB sequence.

4. The parity bit and the stop bit are received. After one character is received, the UART generates a **URXI** interrupt (if enabled). And the **URBF** bit of the UARTSTA register is set to 1.

5. The UART performs the following checks**:**

   a) Parity check**:** The number of "**1**" in the receive data must match with the even or odd parity setting of the **EVEN** bit in the UARTSTA register.

   b) Frame check**:** The start bit must be "0" and the stop bit must be "**1**."

   c) Overrun check: the **URBF** bit of the UARTCON register must be cleared (i.e., the UARTRX register should be read out) before the next received data are loaded into the UARTRX register.

   If any checks failed, the UERRI interrupt will be generated (if enabled). And the error flag is indicated in **PRERR, OVERR** or **FMERR** bit. The error flag should be cleared by software, otherwise, a UERRI interrupt will occur when the next byte is received.

6. Read the received data from the UARTRX register. The **URBF** bit will be cleared by hardware.

8.6.5 UART Baud Rate Generator

- The baud rate generator comprises of a circuit that generates a clock pulse which determines the transfer speed of the transmitted/received data in the UART.

- The input clock of the baud rate generator is derived from the system clock divided by 64 or from Timer 0 divided by 32.

- The system clock should be at 9.83 MHz (PFS = 150) or 14.745 MHz (PFS = 225) when UART is enabled.

- The BRATE2 ~ BRATE0 bits of the UARTCON register determines the desired baud rate.

8.6.6 *UART Applicable Registers*

■ **UARTCON (R30h):** UART Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TB8	UMODE1	UMODE0	BRATE2	BRATE1	BRATE0	UTBE	TXE

Bit 0 (TXE): Enables transmit data function

Bit 1 (UTBE): UART transfer buffer empty flag. Set to "**1**" when the transfer buffer is empty. Reset to "**0**" automatically when writing into the UARTTX register.

> **NOTE**
> *When transmit data is enabled, the UTBE (read-only) bit will be cleared by hardware. Hence, writing to the UARTTX register is required when you want to start transmitting data.*

Bit 4 ~ Bit 2 (BRATE 2 ~ 0): Baud Rate Selector

SELBR3 (for Code Option)		0: Fper = Fpll		1: Fper = Fpll x 2/3
BRATE2 ~ 0	Fper (PFS = 4 ~ 255)	Fper = 9.83MHz (PFS = 150)	Fper = 14.745MHz (PFS = 225)	Fper = 14.745MHz (PFS = 225)
000	Timer 0/32	Timer 0/32	Timer 0/32	Timer 0/32
001	Fper/4096 baud	2400 baud	3600 baud	2400 baud
010	Fper/2048 baud	4800 baud	7200 baud	4800 baud
011	Fper/1024 baud	9600 baud	14400 baud	9600 baud
100	Fper/512 baud	19200 baud	28800 baud	19200 baud
101	Fper/256 baud	38400 baud	57600 baud	38400 baud
110	Fper/128 baud	76800 baud	115200 baud	76800 baud
111	Fper/64 baud	153600 baud	230400 baud	153600 baud

Bit 6 ~ Bit 5 (UMODE 1 ~ 0): UART Mode

UMODE 1: UMODE 0	UART Mode
00	Mode 1**:** 7-bit data
01	Mode 2**:** 8-bit data
10	Mode 3**:** 9-bit data
11	Reserved

Bit 7 (TB8): Transmission Data Bit 8

■ **UARTSTA (R31h):** UART STATUS Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RB8	EVEN	PRE	PRERR	OVERR	FMERR	URBF	RXE

Bit 0 (RXE): Enable receive data function

Bit 1 (URBF): UART read buffer full flag. Set to 1 when one character is received. Reset to 0 automatically when read from the UARTRX register.

> **NOTE**
> *When receive data is enabled, URBF (read-only) bit will be cleared by hardware. Hence, reading from the UARTRX register is required to avoid overrun error.*

Bit 2 (FMERR): Framing error flag. Set to '**1**' when framing error occurs
Clear to '**0**' by software

Bit 3 (OVERR): Overrun error flag. Set to '**1**' when overrun error occurs
Clear to '**0**' by software

Bit 4 (PRERR): Parity error flag. Set to '**1**' when parity error occurs
Clear to '**0**' by software

Bit 5 (PRE): Enable parity addition

"0": Disable

"1": Enable

Bit 6 (EVEN): Select parity check

"0": Odd parity

"1": Even parity

Bit 7 (RB8): Receiving Data Bit 8

■ **UARTTX (R15h):** UART Transfer Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0

Bit 7 ~ Bit 0 (TB7 ~ TB0): Transmit data register. UARTTX register is write-only.

■ **UARTRX (R16h):** UART Receiver Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0

Bit 7 ~ Bit 0 (RB7 ~ RB0): Receive data register. UARTRX register is read-only.

■ **STBCON (R21h):** Strobe Output Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
UINVEN	/REN	BitST	ALL	STB3	STB2	STB1	STB0

Bit 7 (UINVEN): Enable UART **TXD** and **RXD** port inverse output.

"0": Disable **TXD** and **RXD** port inverse output.

"1": Enable **TXD** and **RXD** port inverse output.

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

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

Bit 2 (GLINT): Global Interrupt Control Bit

"0": Disable all interrupts

"1": Enable all un-masked interrupt

■ **INTCON (R22h):** Interrupt Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CPIE	ADIE	URXIE	UTXIE	UERRIE	TMR2IE	TMR1IE	TMR0IE

Bit 3 (UERRIE): Control bit of UART receiving error interrupt

"0": Disable
"1": Enable

Bit 4 (UTXIE): Control bit of UART Transfer buffer empty interrupt
"0": Disable
"1": Enable

Bit 5 (URXIE): Control bit of UART Receiver buffer full interrupt
"0": Disable
"1": Enable

■ **INTSTA (R23h):** Interrupt Status Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CPIF	ADIF	URXI	UTXI	UERRI	TMR2I	TMR1I	TMR0I

Bit 3 (UERRI): Set to '**1**' when UART receiving error occurs
Clear to '**0**' by software or disable UART

Bit 4 (UTXI): Set to '**1**' when UART transfer buffer empty occurs
Clear to '**0**' by software or disable UARTTX (TXE=0)

Bit 5 (URXI): Set to '**1**' when UART receiver buffer full occurs
Clear to '**0**' by software or disable UARTRX (RXE=0)

8.6.7 Transmit Counter Timing

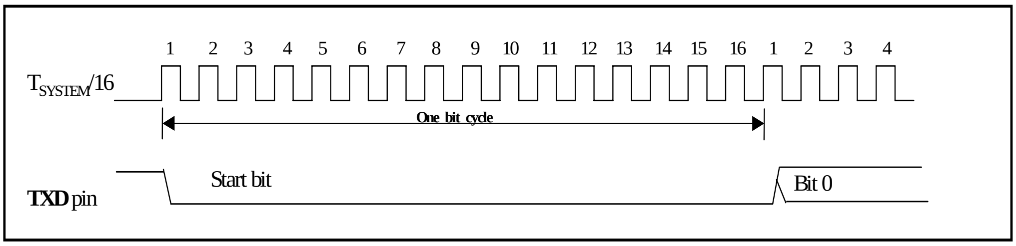


Figure 8-11 UART Transmit Counter Timing

8.6.8 UART Transmit Operation (8-Bit Data with Parity Bit)

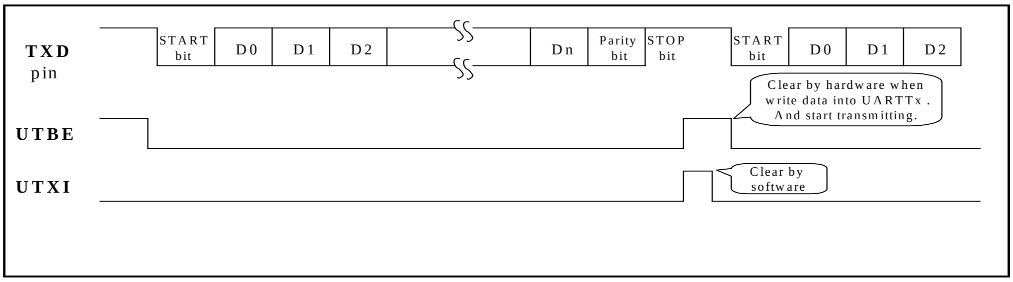


Figure 8-12 UART Transmit Operation

■ **Code Example:**

```
; === UART Transfer buffer empty interrupt
PERIPH:
      PUSH
      JBC         INTSTA,UTXI,Q_UTXINT
      BC          INTSTA,UTXI
      MOV         A,UTX_NO
      COMA        ACC
      MOV         UTX_NO,A
      MOV         UARTTX,A            ;Tx data 55,AA,55,AA
Q_UTXINT:
      POP
      RETI
; === UART 38400 baud 8bit inverse
UTX_SR:
        :
      System setting 9.83MHz
        :
      BS          STBCON,UINVEN       ; TXD & RXD inverse
      MOV         A,#00110101B        ; Enable Tx
      MOV         UARTCON,A           ; 8bit, 38400baud
      MOV         A,#01100000B        ; Disable Rx
      MOV         UARTSTA,A           ; Even Parity
      BC          INTSTA,UTXI         ; TX buffer empty occurs
      BS          INTCON,UTXIE        ; En. TX interrupt
      BS          CPUCON,GLINT        ; Global interrupt
      MOV         A,#0X55
      MOV         UTX_NO,A
      MOV         UARTTX,A            ; Tx data 55
TX_loop:
      SJMP        TX_loop
```

8.6.9 Receive Counter Timing

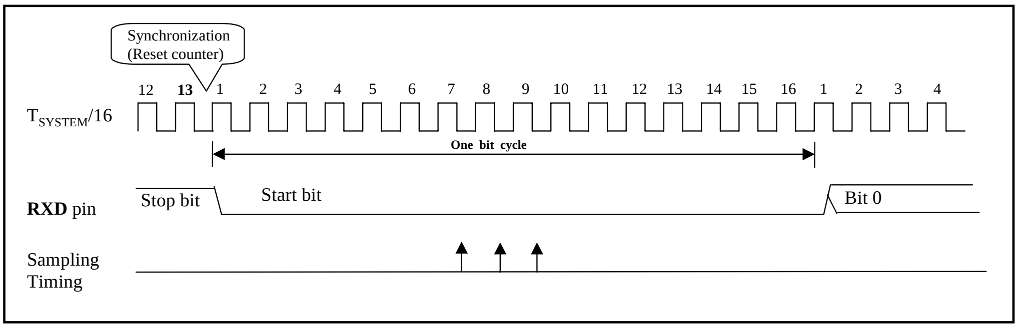


Figure 8-13 UART Receive Counter Timing

- **UART Receive Operation (8 bits data with parity and stop bit):**

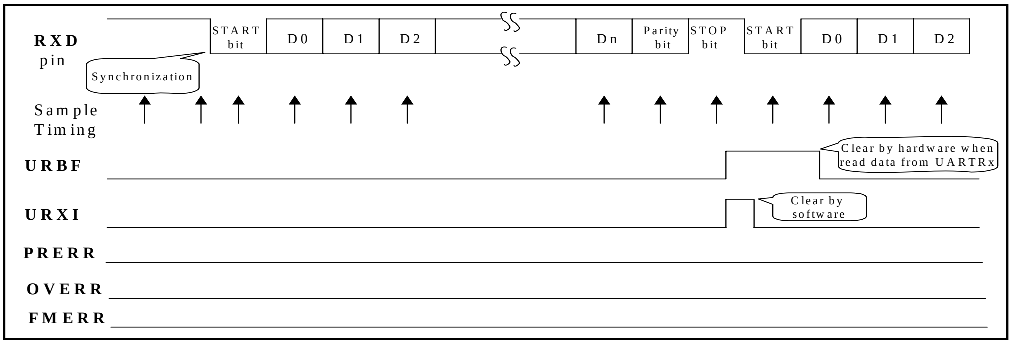


Figure 8-14 UART Receive Operation

- **Code Example:**

```
;===UART Receiver buffer full interrupt
PERIPH:
     PUSH
     JBC     INTSTA,URXI,UERRINT
     BC      INTSTA,URXI
     MOVPR   URX_NO,UARTRX
     SJMP    Q_RXINT
;
;===UART error interrupt
UERRINT:
     JBC     INTSTA,UERRI,Q_RXINT
     BC      INTSTA,UERRI
;---Framing error flag
;---Over run error flag
;---Parity error flag
     MOV     A,UARTSTA
     AND     A,#00011100B
     MOV     PORTH,A
     BC      UARTSTA,FMERR
     BC      UARTSTA,OVERR
     BC      UARTSTA,PRERR
Q_RXINT:
     POP
     RETI
```

```
;===UART 38400 baud 8bit inverse
URX_SR:
      :
   System setting 9.83MHz
   Port H & G setting output port
     :
;---TXD & RXD inverse
   BS        STBCON,UINVEN
;---Disable Tx, 8bit, 38400baud
   MOV       A,#00110100B
   MOV       UARTCON,A
;---Enable Rx, Even Parity
   MOV       A,#01100001B
   MOV       UARTSTA,A
;---UART RX buffer empty interrupt
   BS        INTSTA,URXI
   BS        INTCON,URXIE
;---UART RX error interrupt
   BS        INTSTA,UERRI
   BS        INTCON,UERRIE
;---Global interrupt
   BS        CPUCON,GLINT
RX_loop:
   MOVRP     PORTG,URX_NO
   SJMP      RX_loop
```

8.7 A/D Converter

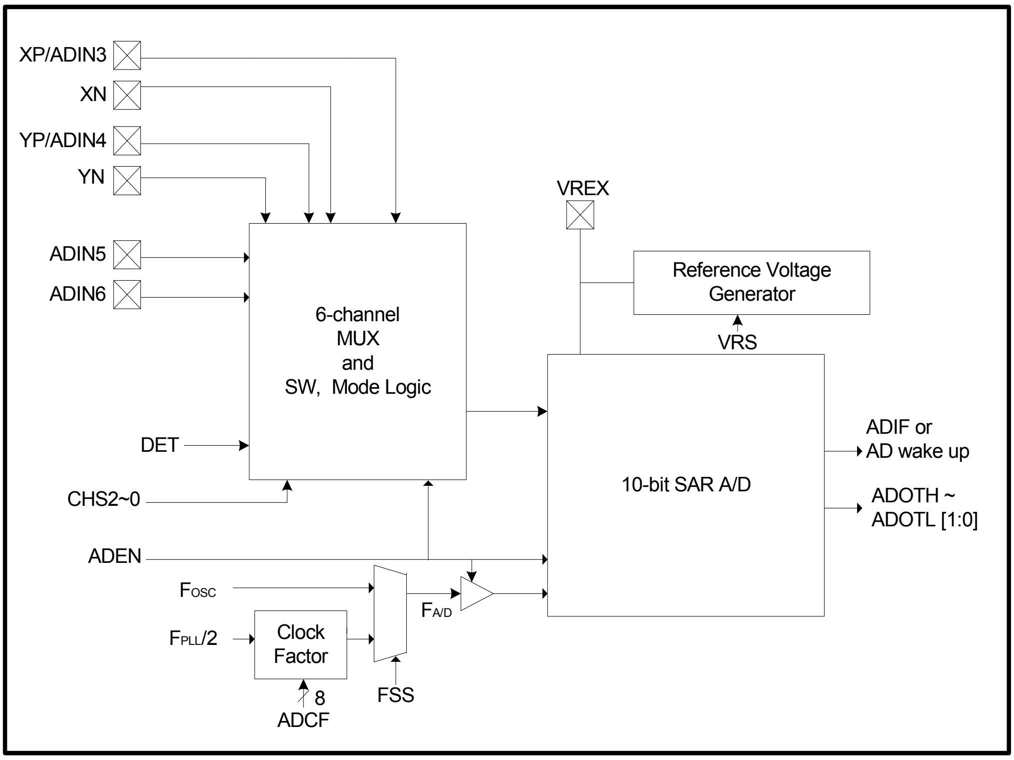


Figure 8-15 A/D Converter Function Block Diagram

Where:

VREX: Reference voltage I/O pin
When VRS=1; it is input pin
When VRS=0, it is output pin

XN (Port C.7): X negative position input

YN (Port C.6): Y negative position input

XP/ADIN3 (Port C.5): X positive position input or A/D input Channel 3

YP/ADIN4 (Port C.4): Y positive position input or A/D input Channel 4

ADIN5 (Port C.3): A/D converter input Channel 5

ADIN6 (Port C.2): A/D converter input Channel 6

This A/D has 6 channels and 10-bit resolution. When the MCU is in Slow mode and ADEN=1, A/D conversion runs immediately. When the MCU is in Fast mode or Idle mode, ADEN=1, A/D conversion will also run. The A/D resolution is better in Idle mode than in Fast mode. The two channels; XP and YP have low resistance switches for driving the touch screens. The other 4 channels are for general application.

The A/D converter operation for touch panel application is as follows:

Step 1: Pen down detection
If the panel is not tapped, the PIRQB is high. When the touch panel is tapped, the PIRQB is low and PIRQB interrupt occurs (if INT is enabled).

Step 2: Measure the X position
If the PIRQB remains low and steady for awhile, the DET bit is cleared, then the PIRQB returns to high and the X position is measured.

Step 3: Measure the Y position
Y position is measured immediately after Step 2.

Step 4: Back to Step 1

8.7.1 A/D Converter Applicable Registers

■ **ADCON (R2Ch):** A/D Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DET	VRS	ADEN	PIRQB	-	CHS2	CHS1	CHS0

Bit 2 ~ Bit 0 (CHS2 ~ CHS0): 2-channel touch screen & 4-channel A/D input selection.

Bit 4 (PIQRB): Touch screen status bit. It is a read bit

"**0**"**:** Touch screen is tapped

"**1**"**:** Touch screen is not tapped

Bit 5 (ADEN): A/D enable control bit. Automatically clears to "0" when ADIF occurs.

"**0**"**:** A/D disable

"**1**"**:** A/D enable

Bit 6 (VRS): A/D input reference voltage selection and enable/disable internal reference generator bit

"**0**"**:** Enable the internal reference generator and the reference voltage is from the internal reference voltage generator

"**1**"**:** Disable the internal reference generator and the reference voltage is from the external VREX pin

Bit 7 (DET): Touch panel pen down detection mode control bit. Enables/disables PIRQB interrupt and wake-up functions.

"**0**": Disable the detection mode. "S" switches are **off** to interrupts and wake-up functions

"**1**": Enable the detection mode. "S" switches are **on** to interrupts and wake-up functions

ADEN	DET	CHS [2:0]	Vin	VRS	Mode
0	0	-	-	1	Standby mode
0	1	-	-	1	Pen-down detection
1	0	000	YP	1	Measure X position (Touch panel)
1	0	001	XP	1	Measure Y position (Touch panel)
1	0	010	ADIN3	0/1	Measure ADIN3
1	0	011	ADIN4	0/1	Measure ADIN4
1	0	100	ADIN5	0/1	Measure ADIN5
1	0	101	ADIN6	0/1	Measure ADIN6

■ **ADOTH (R14h):** A/D Output High Byte Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADOT9	ADOT8	ADOT7	ADOT6	ADOT5	ADOT4	ADOT3	ADOT2

■ **ADOTL (R13h):** A/D Output Low Byte Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WDTEN	-	ADWKEN	-	-	FSS	ADOT1	ADOT0

Bit 7H ~ Bit 0H ~ Bit 1L ~ Bit 0L (ADOT9 ~ ADOT0): 10-bit resolution A/D output data.

Bit 2 (FSS): A/D clock source select bit

"0": A/D clock source is from Fosc

"1": A/D clock source is from F PLL/2

NOTE

When MCU is in FAST mode, the A/D clock source must be from PLL (FSS = 1, PEN = 1). Sourcing A/D clock from Oscillator (FSS = 0, PEN = 0) is prohibited.

Bit 5 (ADWKEN): A/D wake up control bit

"0": Disable A/D wake-up function

"1" : Enable A/D wake-up function

■ **ADCF (R4Eh):** A/D Clock Factor Register

The ADCF is used as a clock factor, such as:

$$F_{A/D} = \frac{F_{PLL}}{2(ADCF+1)}$$

A/D Throughput rate = $F_{A/D}/12$

ADCF Value	Fper=2.03M (PFS=31)	Fper=3.99M (PFS=61)	Fper=7.99M (PFS=122)	Fper=9.83M (PFS=150)	Fper=11.99M (PFS=183)	Fper=14.02M (PFS=214)	Fper=16.7M (PFS=255)
ADCF=3	FA/D=254k	FA/D=499k	FA/D=999k	FA/D=1229k	FA/D=1499k	FA/D=1753k	FA/D=2089k
ADCF=7	FA/D=127k	FA/D=250k	FA/D=499k	FA/D=614k	FA/D=749k	FA/D=876k	FA/D=1044k
ADCF=15	FA/D=63k	FA/D=125k	FA/D=250k	FA/D=307k	FA/D=374k	FA/D=438k	FA/D=522k
ADCF=31	FA/D=31k	FA/D=62k	FA/D=125k	FA/D=154k	FA/D=187k	FA/D=219k	FA/D=261k
ADCF=63	FA/D=15k	FA/D=31k	FA/D=62k	FA/D=77k	FA/D=93k	FA/D=109k	FA/D=130k
ADCF=95	FA/D=11k	FA/D=21k	FA/D=42k	FA/D=60k	FA/D=73k	FA/D=86k	FA/D=102k
ADCF=127	FA/D=10k	FA/D=21k	FA/D=31k	FA/D=51k	FA/D=62k	FA/D=73k	FA/D=87k
ADCF=159	FA/D=6k	FA/D=12k	FA/D=25k	FA/D=31k	FA/D=37k	FA/D=44k	FA/D=52k
ADCF=191	FA/D=5k	FA/D=10k	FA/D=21k	FA/D=25k	FA/D=31k	FA/D=37k	FA/D=44k
ADCF=223	FA/D=4.5k	FA/D=8.9k	FA/D=17.8k	FA/D=21.9k	FA/D=26.8k	FA/D=31.3k	FA/D=37.3k
ADCF=255	FA/D=3.9k	FA/D=7.8k	FA/D=15.6k	FA/D=19.2k	FA/D=23.4k	FA/D=27.3k	FA/D=32.6k

NOTE
Any FA/D value greater than 1.4MHz is invalid.

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

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

Bit 0 (MS0): CPU Fast/Slow mode setting

"**0**"**:** Slow mode

"**1**"**:** Fast mode

Bit 1 (MS1): CPU Sleep and Idle mode setting

"**0**"**:** Sleep mode

"**1**"**:** Idle mode

Bit 2 (GLINT): Global interrupt control bit

"**0**"**:** Disable all interrupts

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

Bit 7 (PEN): PLL enable (only effective when the MCU is in Idle or Slow mode)

"**0**"**:** Disable PLL

"**1**"**:** Enable PLL

■ **INTCON (R22h):** Interrupt Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CPIE	ADIE	URXIE	UTXIE	UERRIE	TMR2IE	TMR1IE	TMR0IE

Bit 6 (ADIE): A/D interrupt control bit
"**0**": Disable
"**1**": Enable

■ **INTSTA (R23h):** Interrupt Status Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CPIF	ADIF	URXI	UTXI	UERRI	TMR2I	TMR1I	TMR0I

Bit 6 (ADIF): Set to "**1**" when A/D output data is ready to be read
Clear to "**0**" by software or disable A/D

8.7.2 Timing Diagram of General A/D Converter Application

CHS [2:0] = 010 ~ 101

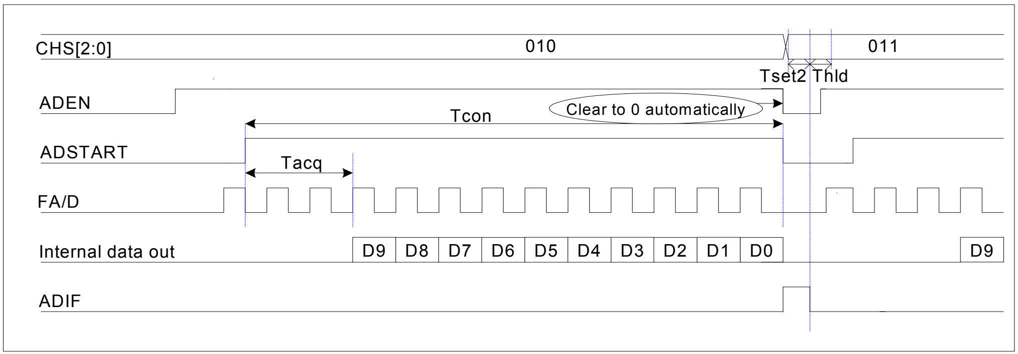


Figure 8-16 A/D Converter General Application Timing Diagram

8.7.3 Correlation between A/D Converter and MCU Mode

When MCU is in Fast mode

When MCU is in Slow mode

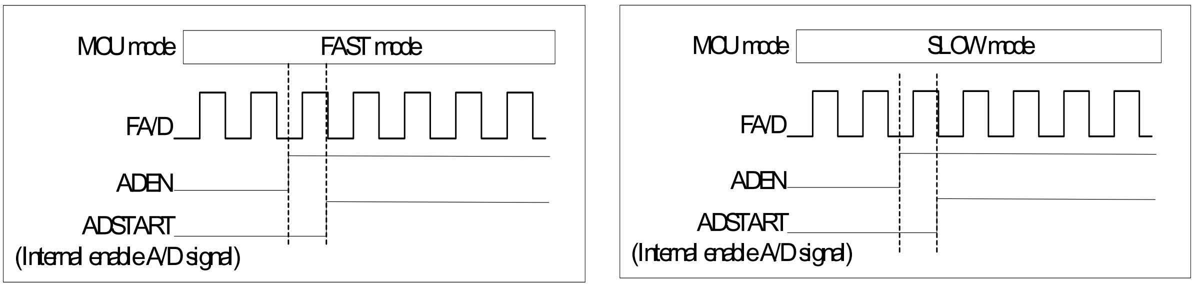


Figure 8-17 A/D Converter vs. MCU Mode

- **Code Example:**

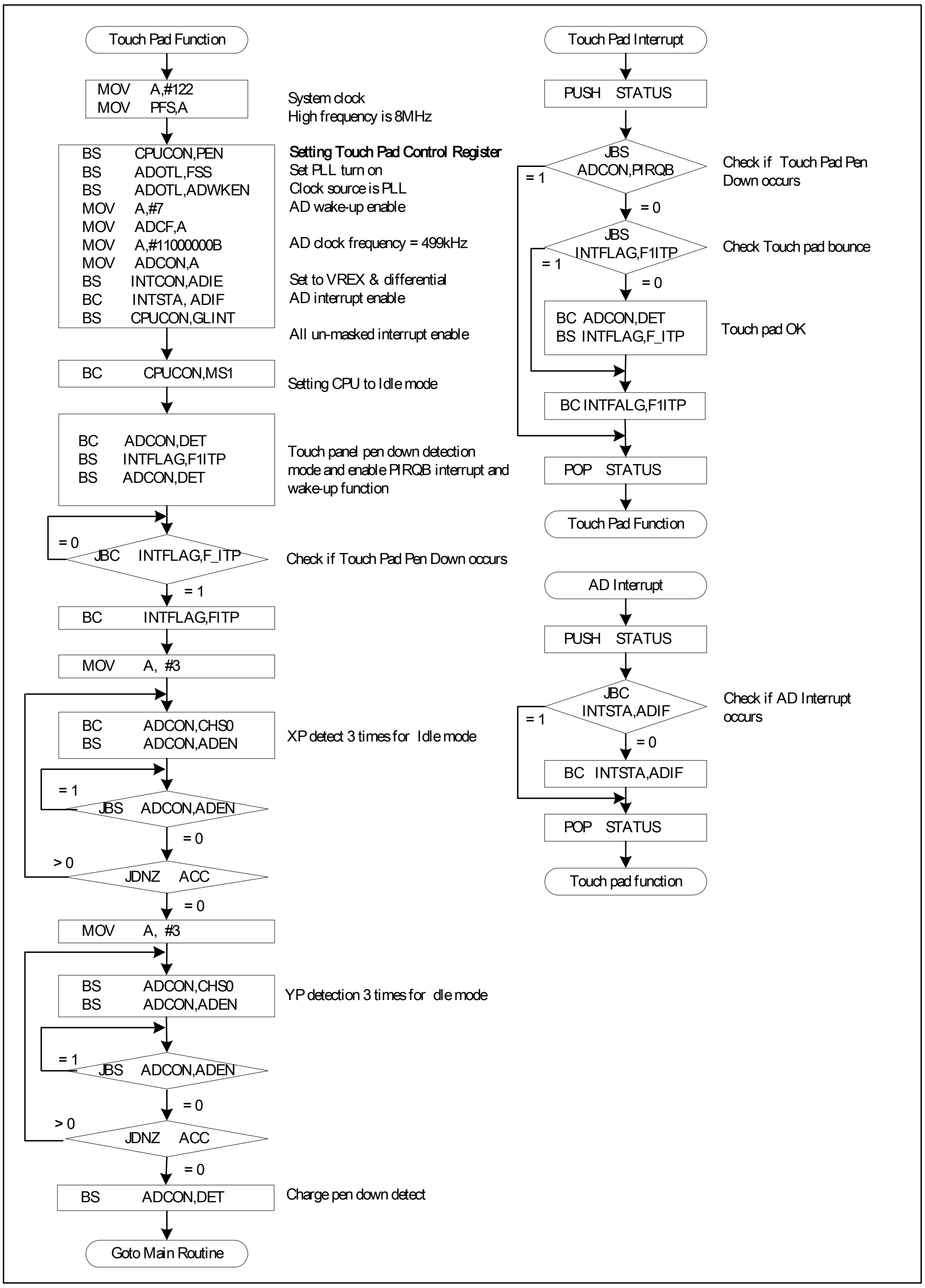
```
;===A/D interrupt
PERIPH:
   PUSH
   JBC        INTSTA,ADIF,Q_ADINT
   BC         INTSTA,ADIF
   BS         INTFLAG,F_IAD
Q_ADINT:
   POP
   RETI
; === Fpll=8MHz & ADCF=7 => FA/D=499kHz
AD_SR:
      :
   System setting 8MHz
   Port H & G setting output port
      :
;---PLL enable
   BS         CPUCON,PEN
;---Clock source is PLL
   BS         ADOTL,FSS
;---FA/D=499kHz
   MOV        A,#7
   MOV        ADCF,A
;---VRIN, Differential, ADIN3
   MOV        A,#00000010B
   MOV        ADCON,A
;---AD interrupt enable
   BS         INTCON,ADIE
   BC         INTSTA,ADIF
   BS         CPUCON,GLINT
```

```
;===Fast mode: MCU in fast mode
    BS     CPUCON,MS1
;---AD wakeup
    BS     ADOTL,ADWKEN
;--- Repeat detect A/D 3 times
    MOV    A,#3
AD3times:
;---AD enable
    BS     ADCON,ADEN
Chk_AD:
    JBC    INTFLAG,F_IAD,Chk_AD
    BC     INTFLAG,F_IAD
    JDNZ   ACC,AD3times
```

```
;===Slow mode: MCU in slow mode
    BC     CPUCON,MS0
;---Repeat detect A/D 3 times
    MOV    A,#3
AD3times:
;---AD enable
    BS     ADCON,ADEN
Chk_AD:
    JBC    INTFLAG,F_IAD,Chk_AD
    BC     INTFLAG,F_IAD
    JDNZ   ACC,AD3times
```

```
;---Out AD to Port H : G
    MOVRP  PORTH,ADOTH
    MOV    A,ADOTL
    AND    A,#00000011B
    MOV    PORTG,A
      :
```

8.7.4 A/D Converter Flowchart



■ **Code Example:**

```
:; *** Touch panel Interrupt
INPTINT:
        PUSH
        JBS       ADCON,PIRQB,Q_TPINT       ; Touch screen status bit
        JBS       INTFLAG,F1ITP,TPINT1
        BC        ADCON,DET                 ; Pen down detection disable
        BS        INTFLAG,F_ITP             ; Pen down ok flag

TPINT1:
        BC        INTFLAG,F1ITP             ; Pen down detect 2 times
Q_TPINT:
        POP
        RETI
; === A/D interrupt
PERIPH:
        PUSH
        JBC       INTSTA,ADIF,Q_ADINT
        BC        INTSTA,ADIF
Q_ADINT:
        POP
        RETI
; === Touch panel routine
TP_SR:
         :
        System setting 8MHz
        Port H & G setting output port
         :
        BS        CPUCON,PEN                ; PLL enable
        BS        ADOTL,FSS                 ; Clock source is PLL
        BS        ADOTL,ADWKEN              ; AD wake-up
        MOV       A,#7
        MOV       ADCF,A                    ; FA/D=499kHz
        MOV       A,#11000000B
        MOV       ADCON,A                   ; VREX, Differential
        BS        INTCON,ADIE               ; AD interrupt enable
        BC        INTSTA,ADIF
        BS        CPUCON,GLINT
TPILoop:
        BC        ADCON,DET                 ; Pen down detection disable
        BS        INTFLAG,F1ITP
        BS        ADCON,DET                 ; Pen down detection enable
TPILp1:
        JBC       INTFLAG,F_ITP,TPILp1
        BC        INTFLAG,F_ITP             ; Clear Pen down flag
; --- Repeat YP detect A/D 3 times
        MOV       A,#3
YP3times:
        BS        ADCON,CHS0                ; YP detection
        BS        ADCON,ADEN                ; AD enable
WaitYAD:
        JBS       ADCON,ADEN,WaitYAD
        JDNZ      ACC,YP3times
        MOVRP     PORTG,ADOTH
; --- Repeat XP detect A/D 3 times
        MOV       A,#3
XP3times:
        BC        ADCON,CHS0                ; XP detection
        BS        ADCON,ADEN                ; AD enable
WaitXAD:
        JBS       ADCON,ADEN,WaitXAD
        JDNZ      ACC,XP3times
        MOVRP     PORTH,ADOTH
        BS        ADCON,DET
         :
        SJMP      TPILoop
```

8.8 Key I/O

The 7-pin key input (Port A.0~6) and 16-pin key strobe (share with LCD SEG0~15) can achieve a maximum of 112-key matrix.

8.8.1 Automatic Key Scan or Software Key Scan

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

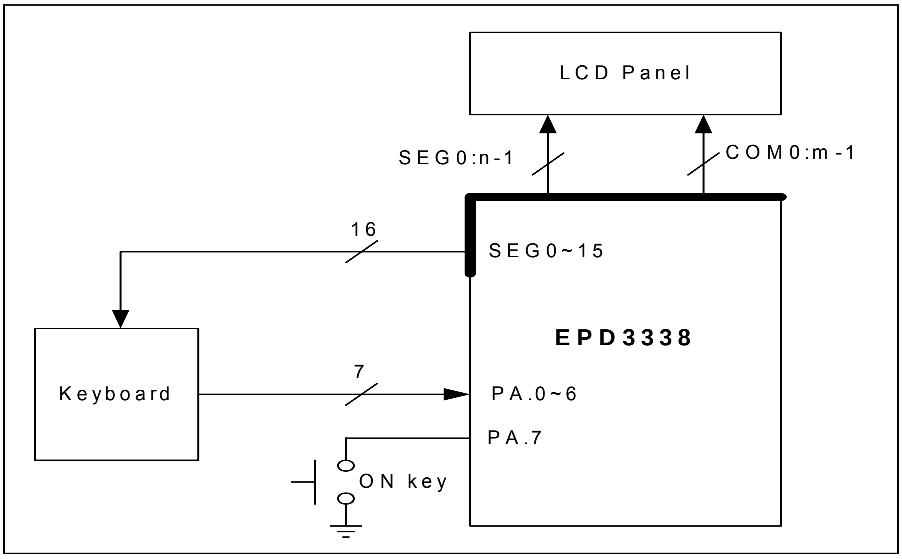


Figure 8-18 Keyboard Function Block Diagram

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

The following are the "Key input process":

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

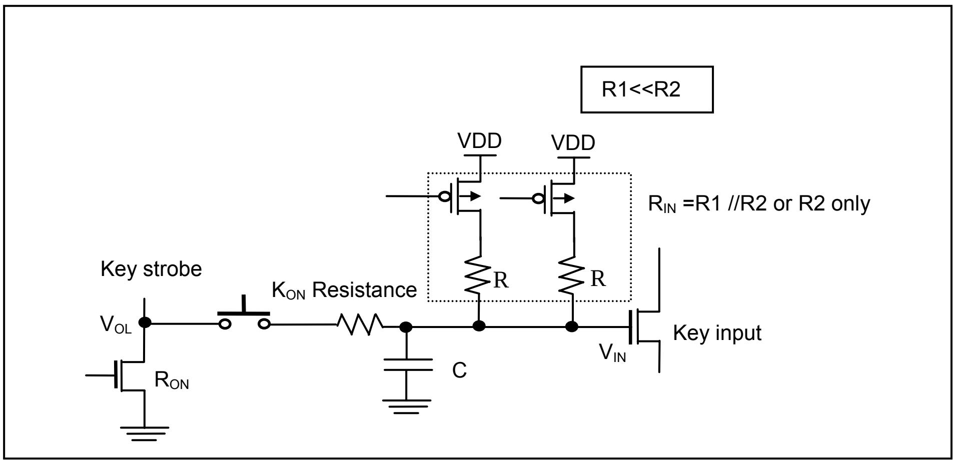


Figure 8-19 Key Circuit Diagram

The detailed function is summarized in the following table:

(Code Option: Select "Port A_KE divided from LCD_BitST")

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

(Code Option: Select "Port A_KE connected with LCD_BitST")

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

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

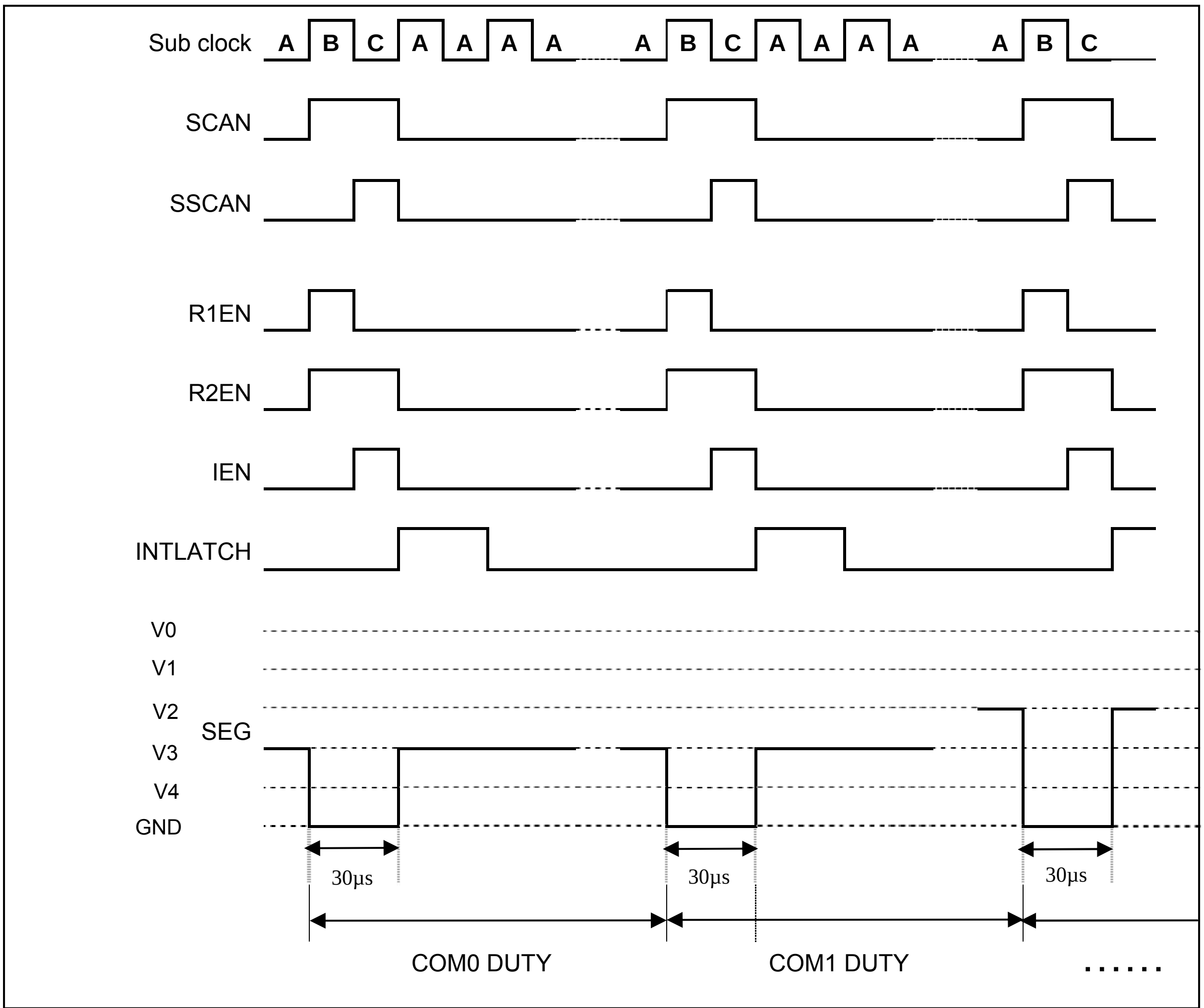


Figure 8-20 Automatic Key Scan Timing Diagram (SCAN = 1)

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

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

There are two ways to output a strobe signal**:**

■ Automatic Key Scan

The LCD waveform has a 30µs low pulse at the beginning of every common duty signal when the **SCAN** bit of the STBCON register is set. The strobe timing is as shown in the following figure (Figure 8-21).

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

■ **Software Key Scan**

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

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

Under this mode you can set **All** bits of the STBCON register to let Segments 0~15 to be kept at GND level.

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

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

Automatic key scan is used to check **whether any key is pressed**. If a key is pressed, PA.0~7 pin falling edge will be detected**,** then an **interrupt occurs**.

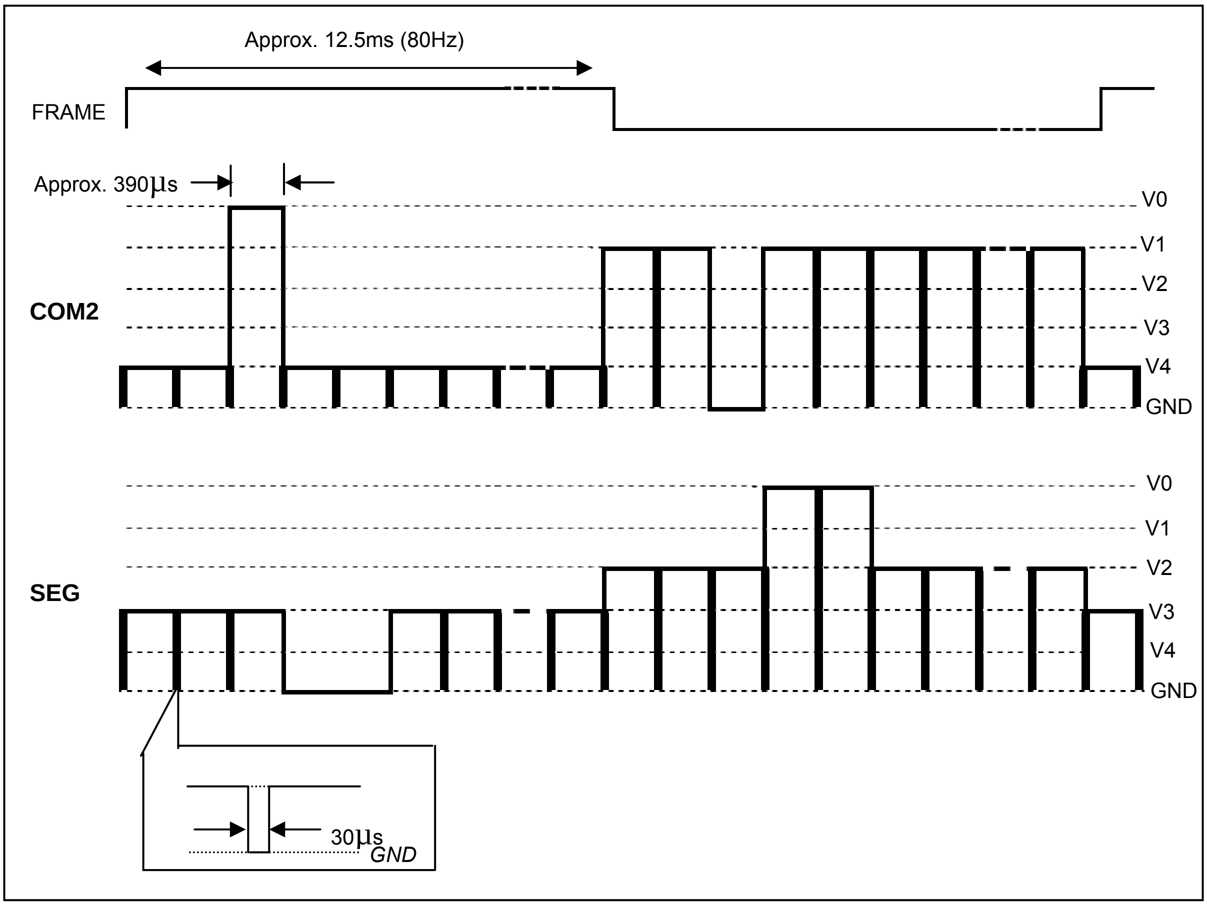


Figure 8-21 Automatic Strobe Signal (SCAN = 1)

Software key scan will examine "which key was pressed".

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

STBCON				Key Strobe (Share with Segment 0~15)																LCD	
SCAN	BitST	ALL	STB3~0	seg 0	seg 1	seg 2	seg 3	seg 4	seg 5	seg 6	Seg 7	seg 8	seg 9	seg 10	Seg 11	seg 12	seg 13	seg 14	seg 15	seg 16:n-1	com 0:m-1
0	0	×	××××	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		
1	×	×	××××	Display waveform with automatic key scan																	

8.8.2 Input/Output Key Applicable Registers

■ **Port A (R17h):** 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 (PA7 ~ PA0): Key input. Input falling edge interrupt or wake-up pin.

■ **PAINTEN (R2Dh):** Port A Interrupt Enable Control Register

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

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

"**0**": Disable

"**1**": Enable

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

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

Bit 7 ~ Bit 0 (PA7I ~ PA0I): Port A interrupt INT status
Set to "**1**" when pin falling edge is detected
Clear to "**0**" by software

■ **PAWAKE (R2Fh):** Port A Wakeup Control Register

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

Bit 7 ~ Bit 0 (WKEN7 ~ WKEN0): Port A wakeup function control bit
"**0**"**:** Disable wake-up function
"**1**"**:** Enable wake-up function

■ **PACON (R55h):** Port A Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	-	-	Bit7PU	/R2EN	/R1EN	KE

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

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

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

■ **STBCON (R21h):** Strobe Output Control Register

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

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

Bit 4 (ALL): Set All strobe
"**0**"**:** Bit strobe
"**1**"**:** All strobe

Bit 5 (BitST): Enable Bit strobe
"**0**"**:** Display waveform
"**1**"**:** Strobe signal is defined by STB3 ~ 0 registers.

Bit 6 (SCAN): Automatic key scan or specify the scan signal bit by bit.
"**0**"**:** Key scan is specified by the defined bits STB3~0.
"**1**"**:** Auto strobe scanning

■ **Code Example:**

(Code Option: Select **"Port A_KE divided from LCD_BitST")**

```
; Key matrix 1 (Port A and Ground):
; === Sleep mode
PAIN_SR:
       :
; --- Port A wakeup
       MOV    A,#11111111B
       MOV    PAWAKE,A
;--- /R2EN, PA.7 Pull-up & KE enable
       MOV    A,#00001011B
       MOV    PACON,A
; --- Port A interrupt enable
       MOV    A,#11111111B
       MOV    PAINTEN,A
       CLR    PAINTSTA
       BS     CPUCON,GLINT
; ---  Sleep MODE
       BC     CPUCON,MS1
PAINloop:
      ;BS     STBCON,BitST
       SLEP
       NOP
        :
       SJMP   PAINloop
```

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

*(**Continued**)*

```
; Key matrix 2 (Port A and SEG0 ~ SEG15):
; *** Key scan function
;     LCD display setting
      MOV     A,#0XFF
      MOV     PAWAKE,A              ; Port A wake-up function setting
      CLR     PACON                 ; /R1EN, /R2EN enable
; === Key strobe all routine
KeyAll:
      BS      STBCON,ALL            ; Strobe all
      BS      STBCON,BitST
      BS      PACON,KE              ; Key enable
      LCALL   DLY10US
      BS      PACON,R1EN            ; /R1EN disable
      MOV     A,PORTA               ; Port A input data
      BC      PACON,KE              ; Key disable
      BC      STBCON,BitST
      BC      STBCON,ALL            ; Strobe all disable
      OR      A,#10000000B
      JLE     A,#0xFE,KeyScan
; === Idle mode auto key scan routine
KeyIdle:
      BS      STBCON,SCAN           ; Auto-key scan enable
      BS      CPUCON,MS1            ; Idle mode
      SLEP
      NOP
; === Key scan routine
KeyScan:
      CLR     STBCON                ; Auto-key scan disable
KeyLoop:
      BS      STBCON,BitST          ; Strobe ON(STB3~0)
      BS      PACON,KE              ; Key enable
      LCALL   DLY10US
      BS      PACON,R1EN            ; /R1EN disable
      MOV     A,PORTA               ; Port A input data
      BC      PACON,KE              ; Key disable
      BC      STBCON,BitST          ; Strobe OFF(STB3~0)
      JLE     A,#0FEH,KeyNmval      ; If PORTA=<0xFE Goto KeyNmval
      INC     STBCON
      JBC     STBCON,ALL,KeyLoop
      CLR     STBCON
      SJMP    ScanNoKey
KeyNmval:
; --- Check key number
      CLR     KEY_NO
ChKeyNo:
      RRC     ACC
      JBC     STATUS,F_C,KeyScanOk
      INC     Key_No
      SJMP    ChKeyNo
; --- Key Scan is finished
KeyScanOk:
      INC     KEY_NO
      SWAP    KEY_NO
      MOV     A,STBCON
      AND     A,#00001111B
      OR      Key_No,A
      SJMP    QKeyScan
        :
ScanNoKey:
      CLR     KEY_NO
QKeyScan:
      RET
```

8.9 LCD Driver

The **EPD3338** provides a directly driven LCD. It supports multiplex drive for 40SEG×32COM. It is also able to use pads as an LCD driver pin or as input/output port. The duty ratio is selected by the LCDCONC register. An LCD RAM for direct correspondence is available with LCD Pixel. Charge pump can pump 2 or 3 times of Vx. The LCD contrast has 32 adjustable levels and the LCD bias is selectable. The maximum LCD operating voltage can be determined by external resistor Ra and Rb. Furthermore, multiple frame frequencies are provided to reduce power consumption.

This embedded LCD driver contains power supply circuits and generates waveforms for display drive.

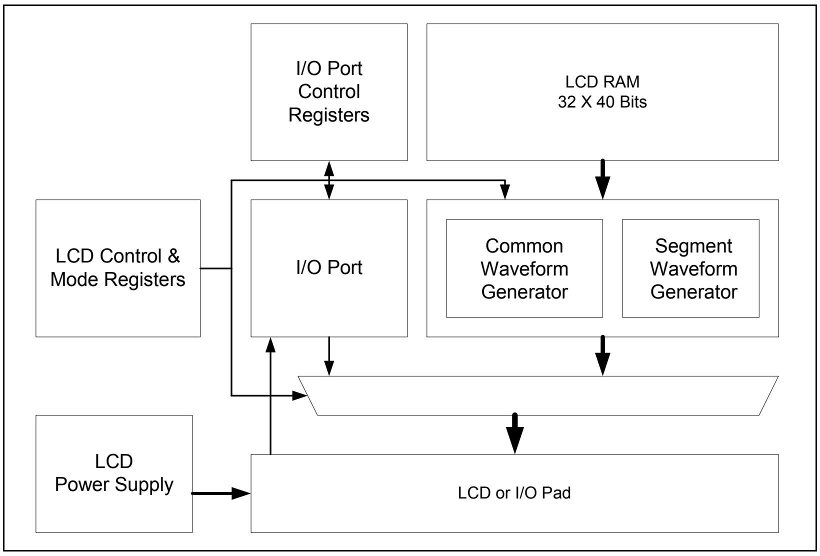


Figure 8-22 Function Block Diagram

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

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

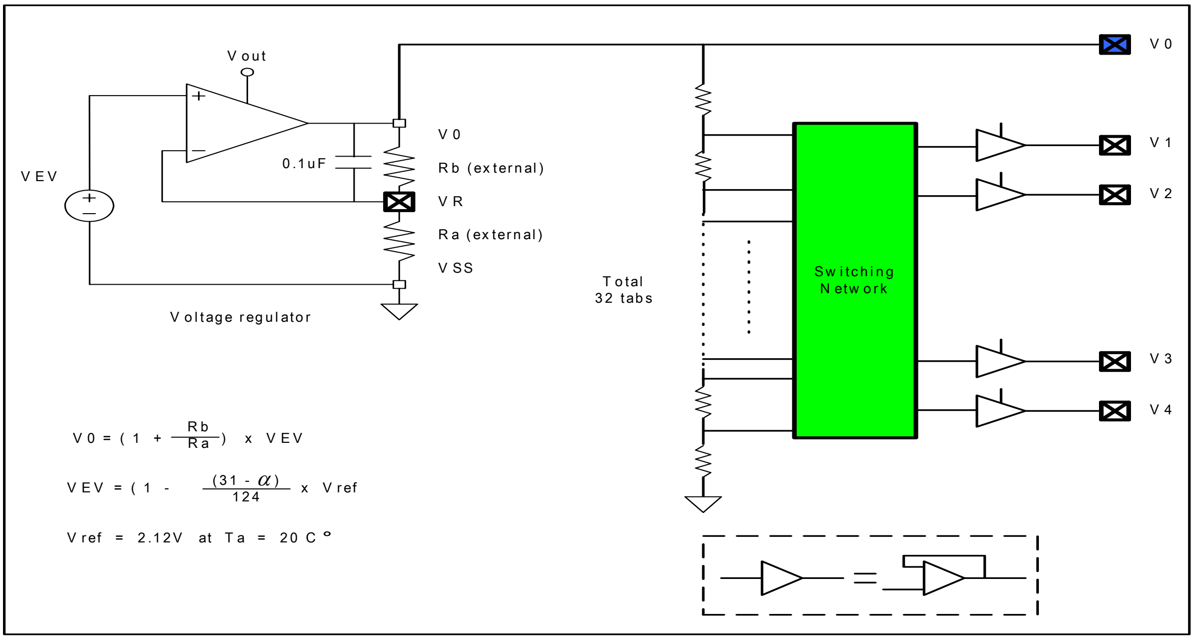


Figure 8-23 Internal Power Supply Circuitry

■ Switching Network:

LCD Bias	V1	V2	V3	V4
1/3	0.665*V0	0.335*V0	0.665*V0	0.335*V0
1/3.5	0.715*V0	0.430*V0	0.570*V0	0.285*V0
1/4	0.750*V0	0.500*V0	0.500*V0	0.250*V0
1/4.5	0.780*V0	0.555*V0	0.445*V0	0.220*V0
1/5	0.800*V0	0.600*V0	0.400*V0	0.200*V0
1/5.5	0.820*V0	0.635*V0	0.365*V0	0.180*V0
1/6	0.835*V0	0.665*V0	0.335*V0	0.165*V0
1/6.5	0.845*V0	0.690*V0	0.310*V0	0.155*V0

The built-in voltage converter (boost circuit), generates twice or triple boosted voltage output to VOUT pin. And VOUT provides the operating voltage for the operational-amplifier circuits. The external capacitors must be connected as shown in the figure at right.

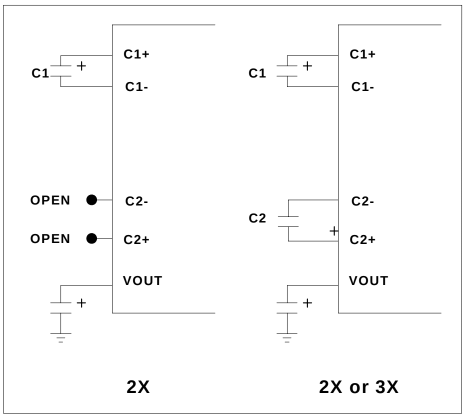


Figure 8-24 Voltage Converter External Capacitors

8.9.1 LCD Code Option

■ Duty Ratio: Maximum Duty Ratio Option

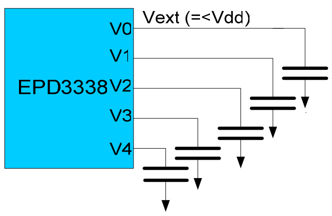
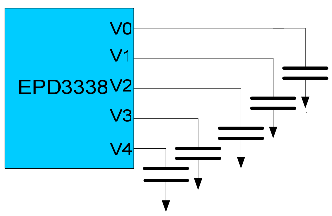
COM8-15/SEG103-96; COM16-23/SEG87-80 & COM24-31/SEG95-88 Status Setting

Duty Ratio	Max. Display Size	Common Driver Used			
		COM0~7	COM8~15/ SEG103~96	COM16~23/ SEG87~80	COM24~31/ SEG95~88
1/4	64x4	COM 0~7	SEG103~96	SEG87~80	SEG95~88
1/8	64x8	COM 0~7	SEG103~96	SEG87~80	SEG95~88
1/9	56x9	COM 0~15		SEG87~80	SEG95~88
1/11	56x11	COM 0~15		SEG87~80	SEG95~88
1/16	56x16	COM 0~15		SEG87~80	SEG95~88
1/24	48x24	COM 0~23			SEG95~88
1/32	40x32	COM 0~31			

■ V1; V2; V3 & V4 OP Buffer:

OP Buffer	Normal Current	No Current
V1	ON	OFF
V2	ON	OFF
V3	ON	OFF
V4	ON	OFF

■ V0~V4 OP buffer type configuration for different application

OP Buffer	Small Current	Normal Current
VOUT	Vdd (Partial display mode)	2Vdd or 3Vdd
V0	OFF (Vext : external power =<Vdd)	ON
V1~V4	ON	ON
Circuit		
Consumption	Low Current	Normal Current

■ **V0 OP Buffer Control Bit:** Select "V0 OP buffer turn off" or "V0 OP buffer turn on".

■ **Port G Low Nibble Control Bits (SEG48~51):** Select "LCD segment signal output" or "General I/O function"

■ **Port G High Nibble Control Bits (SEG52~55):** Select "LCD segment signal output" or "General I/O function"

■ **Port H Low Nibble Control Bits (SEG56~59):** Select "LCD segment signal output" or "general I/O function"

■ **Port H High Nibble Control Bits (SEG60~63):** Select "LCD segment signal output" or "general I/O function"

The LCD segment pin configuration is as follows:

SEG0 ~ 15	SEG16 ~ 23	SEG48 ~ 63	SEG64 ~ 79	SEG80 ~ 95	SEG96 ~ 111
SEG0/Key Strobe0 ↓ SEG7/Key Strobe7	SEG16 ↓ SEG23	SEG48/Port G.0 ↓ SEG55/Port G.7	Prohibited	SEG80/COM16 ↓ SEG87/COM23	SEG96/COM15 ↓ SEG103/COM8
SEG8/Key Strobe8 ↓ SEG15/Key Strobe15	-	SEG56/Port H.0 ↓ SEG63/Port H.7	-	SEG88/COM24 ↓ SEG95/COM31	SEG80/COM7 ↓ SEG111/COM0

■ **LCDCONA (R50h):** LCD Control Register A

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

Bit 4 ~ Bit 0 (ADJ4 ~ ADJ0): LCD Contrast Adjustment

ADJ4	ADJ3	ADJ2	ADJ1	ADJ0	α	Contrast
0	0	0	0	0	0	Low
0	0	0	0	1	1	
:	:	:	:	:	:	
:	:	:	:	:	:	
1	1	1	1	0	30	
1	1	1	1	1	31	High

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

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

Bit 7 ~ Bit 5 (BSEL2 ~ BSEL0): LCD Bias select

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

Different duty radio requires different bias level. For optimum bias level, B_L can be calculated from the equation:

$$B_L = \frac{1}{\sqrt{Duty\ ratio} + 1}$$

Setting to the optimum bias level will have a better result on contrast and view angle.

■ **LCDCONB (R51h):** LCD Control Register B

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
REV	-	LCDON	LCDPM2	LCDPM1	LCDPM0	SFR1	SFR0

Bit 1 ~ Bit 0 (SFR1 ~ SFR0): Frame Frequency Adjustment

CL Frequency vs. Duty Ratio Table:

		CL Frequency						
SFR1	SFR0	1/4 duty	1/8 duty	1/9 duty	1/11 duty	1/16 duty	1/24 duty	1/32 duty
0	0	Fosc /104	Fosc / 52	Fosc / 46	Fosc / 38	Fosc / 26	Fosc /17	Fosc /13
0	1	Fosc /112	Fosc / 56	Fosc / 50	Fosc / 40	Fosc / 28	Fosc / 19	Fosc /14
1	0	Fosc /120	Fosc / 60	Fosc / 54	Fosc / 44	Fosc / 30	Fosc / 20	Fosc /15
1	1	Fosc /128	Fosc / 64	Fosc / 56	Fosc / 46	Fosc / 32	Fosc / 21	Fosc /16

NOTE

Fosc = 32.8kHz ± 25% (all conditions)

The display clock CL affects the current consumption and the frame frequency affects the flicker. Therefore, fine tuning of the display clock CL and the frame frequency are required.

Bit 4 ~ Bit 2 (LCDPM2 ~ LCDPM0): LCD Power Control Mode

LCDPM [2:0]	Power Control Mode	V/C Circuits	V/R Circuits	V/F Circuits (V0~V4)	VOUT Status	Discharge
000	LCD power off mode	Off	Off	Off	VDD	Off
001	External power mode	Off	Off	Off (From external)	Off (From external)	Off
010	Discharge mode	Off	Off	Off	Off	On
100	Partial display mode (Vout=VDD)	Off	On	On	VDD	Off
101	Normal display mode	On	On	On	2x or 3x VDD	Off
Others	Reserved					

Bit 5 (LCDON): LCD display control bit. (All COM and SEG pins are tied to ground when LCD display is off)
"**0**": LCD display off
"**1**": LCD display on

NOTE
Before entering Sleep mode, the LCD function of LCDCONB control must be set:
1. The LCDON bit of LCDCONB set to 0 (LCD off).
2. The LCDPM[2:0] bit of LCDCONB set to [000] (LCD power off mode).

Bit 7 (REV): LCD panel display status control. The REV is used to invert the display status on the LCD panel without rewriting the contents of the display data RAM.
"**0**": Normal: Display data "1" turns on the LCD
"**1**": Inverse: Display data "0" turns on the LCD

■ **LCDCONC (R52h):** LCD Control Register C

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	DRSEL2	DRSEL1	DRSEL0	BOOST	Fix 0	LCDARH1	LCDARH0

Bit 1 ~ Bit 0 (LCDARH1 ~ LCDARH0): Page address for the LCD RAM

Bit 3 (BOOST): Set the number of boosting steps
"**0**": 2 times
"**1**": 3 times

NOTE
When the BOOST is set to "1", the clamping circuit is enabled to clamp the Vx at 2.4V.

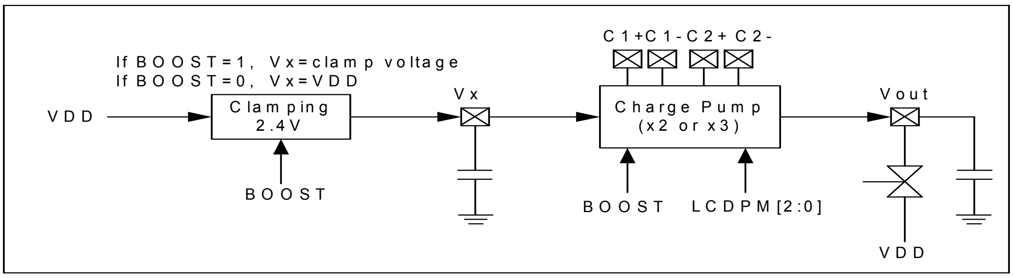


Figure 8-25a Clamping Circuitry

Bit 6 ~ Bit 4 (DRSEL2 ~ DRSEL0): LCD duty select. Eight LCD duty ratio options are offered with which you can select and change into different operating conditions. Operating in normal or partial display mode can reduce power consumption.

DRSEL2	DRSEL1	DRSEL0	Duty Ratio	Max. Display Size
0	0	0	1/4	64×4
0	0	1	1/8	64×8
0	1	0	1/9	56×9
0	1	1	1/11	56×11
1	0	0	1/16	56×16
1	0	1	1/24	48×24
1	1	0	1/32	40×32

- **LCDARL (R53h):** LCD RAM Column Address

- **LCDDATA (R54h):** LCDDATA register is an indirect addressing pointer of the LCD RAM. Any instruction using LCDDATA as register actually accesses the LCD RAM pointed to by LCDARH**:** LCDARL.

Figure 8-25b LCDDATA Accessing the LCDARH: LCDARL Pointed LCD RAM

- **POST_ID (R2Bh):** Post increase/decrease the control register. After accessing (read or write) the LCD RAM, the **LCDARL** register can be automatically increased or decreased by setting the POST_ID register.

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

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

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

■ LCD RAM MAP

NOTE
LCDARL=18h~2Fh and 40h~4Fh are NOT Used

Page 00 (LCDARH[2:0]=000)

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
00H (SEG0)								
⇣								
17H (SEG23)								
18h~2Fh	**Not Used**							
30H (SEG48)								
⇣								
3FH (SEG63)								
40h~4Fh	**Not Used**							
50H (SEG80)								
⇣								
67H (SEG103)								

Page 01 (LCDARH[2:0]=001)

RAM Address LCDARL	COM8	COM9	COM10	COM11	COM12	COM13	COM14	COM15
	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
00H (SEG0)								
⇣								
17H (SEG23)								
18h~2Fh	**Not Used**							
30H (SEG48)								
⇣								
3FH (SEG63)								
40h~4Fh	**Not Used**							
50H (SEG80)								
⇣								
67H (SEG103)								

Page 02 (LCDARH[2:0]=010)

RAM Address LCDARL	COM16	COM17	COM18	COM19	COM20	COM21	COM22	COM23
	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
00H (SEG0)								
⇣								
17H (SEG23)								
18h~2Fh	**Not Used**							
30H (SEG48)								
⇣								
3FH (SEG63)								
40h~4Fh	**Not Used**							
50H (SEG80)								
⇣								
67H (SEG103)								

Page 03 (LCDARH[2:0]=011)

RAM Address LCDARL	COM24 Bit 0	COM25 Bit 1	COM26 Bit 2	COM27 Bit 3	COM28 Bit 4	COM29 Bit 5	COM30 Bit 6	COM31 Bit 7
00H (SEG0)								
↓								
17H (SEG23)								
18h~2Fh	**Not Used**							
30H (SEG48)								
↓								
3FH (SEG63)								
40h~4Fh	**Not Used**							
50H (SEG80)								
↓								
67H (SEG103)								

■ **LCD Waveform:**

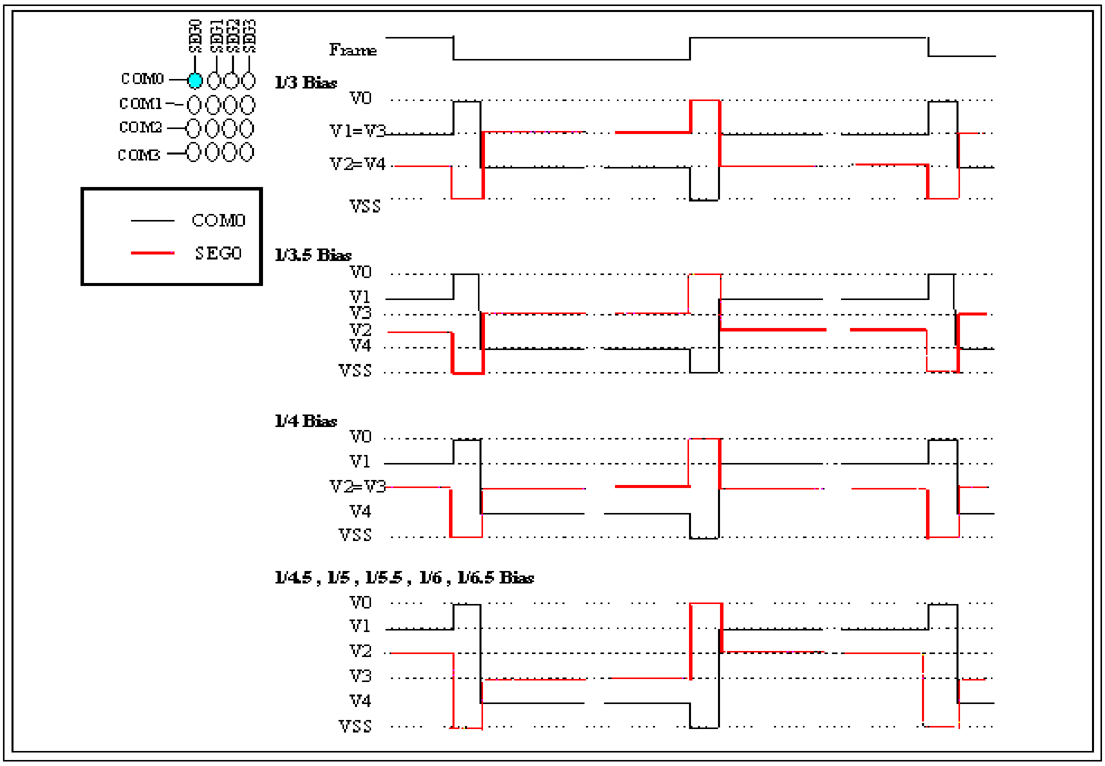


Figure 8-25c LCD Waveform

■ **Code Example:**

```
; === 1/32 Duty & 1/6.5 Bias
L32Duty:
; --- /6.5 Bias
    MOV      A,#11110000B
    MOV      LCDCONA,A
; --- 1/32 Duty, BOOST=1
    MOV      A,#01101000B
    MOV      LCDCONC,A
; --- LCD Off, Normal Display Mode
    MOV      A,#00010111B
    MOV      LCDCONB,A
    SCALL    DspRAMdot
; --- LCD turn-on
    BS       LCDCONB,LCDON
    LCALL    Delay1sec
     :
DspLoop:
; --- Partial display mode
    BC       LCDCONB,LCDPM0
    LCALL    Delay1sec
; --- Inverse display
    BS       LCDCONB,LCDPM0
    BS       LCDCONB,REV
    LCALL    Delay1sec
; --- Normal display
    BC       LCDCOMB,REV
    LCALL    Delay1sec
     :
    SJMP     DspLoop
```

```
; *** Display LCD RAM is data 55 & AA
DspRAMdot:
; --- LCD increase enable
    MOV      A,#01000100B
    MOV      POST_ID,A
; --- CD page 00
    MOV      A,#11111000B
    AND      LCDCONC,A
DspRAMd1:
    CLR      LCDARL
    MOV      A,#0X20
    MOV      CNT_HI,A
    SCALL    WrLRAMd
    MOV      A,#030H
    MOV      LCDARL,A
    MOV      A,#0X10
    MOV      CNT_HI,A
    SCALL    WrLRAMd
    INC      LCDCONC
    MOV      A,#00000011B
    AND      A,LCDCONC
    JLE      A,#00000011B,DspRAMd1
    RET
; === Write LCD RAM is dot matrix
WrLRAMd:
    MOV      A,#0X55
    MOV      LCDDATA,A
    MOV      A,#0XAA
    MOV      LCDDATA,A
    JDNZ     CNT_HI,WrLRAMd
    RET
```

8.10 Serial Peripheral Interface (SPI)

- Operation in either Master mode or Slave mode
- Three-wire or Four-wire full duplex synchronous communication
- Programmable Shift Register Length (24/16/8 bits)
- Programmable communication bit rates
- Programmable clock polarity
- Programmable shift direction
- Programmable sample phase
- Interrupt flag available for the read buffer full
- Up to 4MHz (system clock at 16MHz) bit frequency

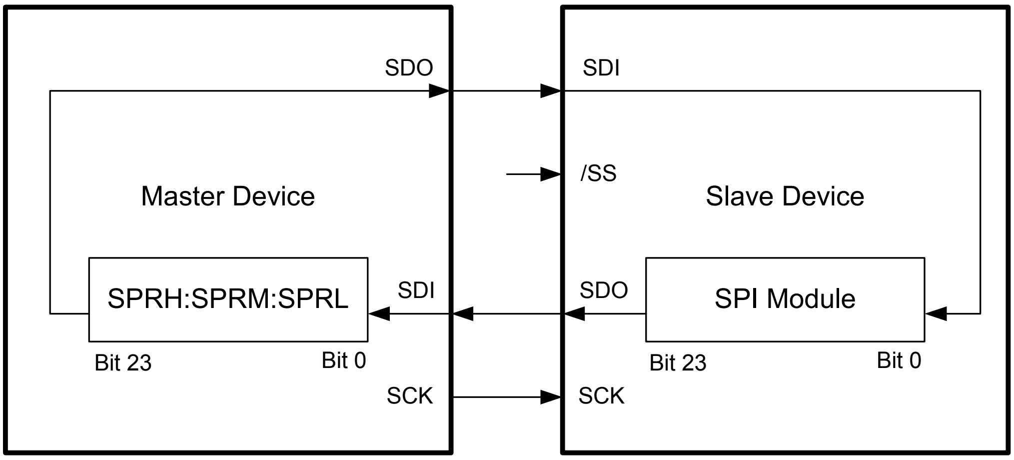


Figure 8-26a Single SPI Master/Slave Communication

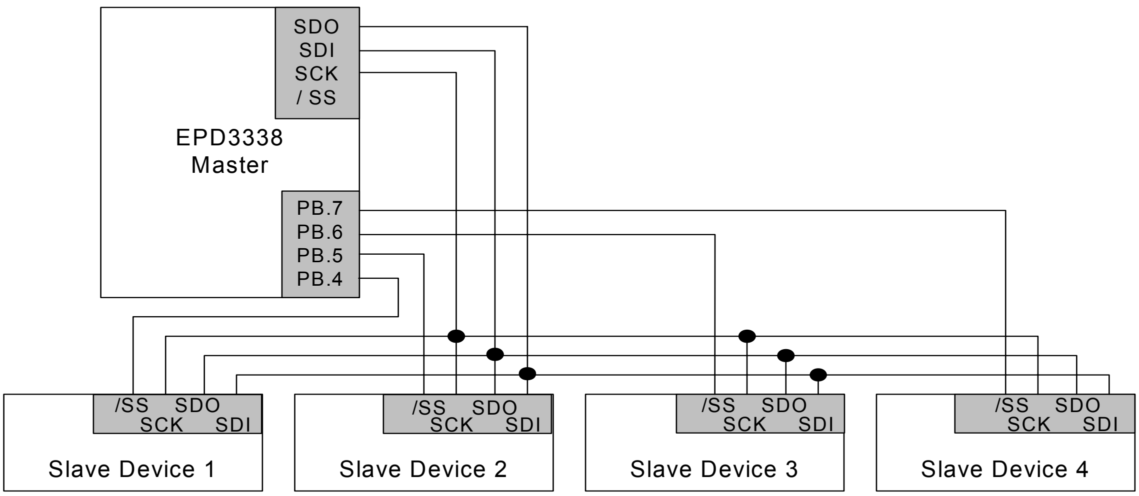


Figure 8-26b SPI Configuration Example of Single-Master and Multi-Slaves

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

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

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

■ Master Mode

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

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

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

■ Slave Mode

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

As with the Master Mode, you can also select the shift register length under Slave mode. Transfer data are written to SPRH, SPRM, SPRL registers. After writing data into the SPRL register, the **SE** bit of SPICON register is then set by hardware. However, the shifting start is controlled by the Master device clock input.

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

8.10.1 SPI Pin Description

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

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

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

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

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

8.10.2 SPI Applicable Registers

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

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

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

■ **SPICON (R3Fh):** SPI Control Register

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

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

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

Bit 1 (DORD): Data transmission order
"**0**"**:** Shift left (MSB first)
"**1**"**:** Shift right (LSB first)

Bit 2 (EDS): Select the rising / falling edge latch by programming the EDS bit
"**0**"**:** Falling edge
"**1**"**:** Rising edge

Bit 5 ~ Bit 3 (BRS2 ~ BRS0): Bit rate select. Programs the clock frequency/rates and sources.
000: Master, TMR0/2
001: Master, Fsystem/4
010: Master, Fsystem/16
011: Master, Fsystem/64
100: Master, Fsystem/256
101: Master, Fsystem/1024
110: Slave, /SS enable
111: Slave, /SS disable

■ **SPI Bit Rate Table:**

Prescaler		Fsystem			
BRS2:0	**Bit Rate**	**16MHz**	**10MHz**	**4MHz**	**32.768kHz**
001	(Fpll/2)/4	4000000	2500000	1000000	8196
010	(Fpll/2)/16	1000000	625000	250000	2048
011	(Fpll/2)/64	250000	156250	62500	512
100	(Fpll/2)/256	62500	39063	15625	128
101	(Fpll/2)/1024	15625	9766	3096	32

Bit 7 ~ Bit 6 (TLS1 ~ TLS0): Shift buffer length select. The Shift buffer length is programmable.
00**:** SPI disable
01**:** Enable SPI and shift buffer length = 24 bits
10**:** Enable SPI and shift buffer length = 16 bits
11**:** Enable SPI and shift buffer length = 8 bits

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

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WEN	-	SRBFIE	SRBFI	SPWKEN	SMP	DCOL	RBF

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

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

Bit 1 (DCOL): SPI Data collision

Bit 2 (SMP): SPI data input sample phase

“**0**”**:** Input data sampled at the middle of data output time
“**1**”**:** Input data sampled at the end of data output time

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

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

“**0**”**:** Disable SPI (Slave mode) read buffer full wakeup
“**1**”**:** Enable SPI (Slave mode) read buffer full wakeup

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

“**0**”**:** Data collision does not occur
“**1**”**:** Data collision occurs. Should be cleared by software

Bit 5 (SRBFIE): Control bit of SPI read buffer full interrupt

“**0**”**:** Disable interrupt function
“**1**”**:** Enable interrupt function

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

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

Bit 2 (GLINT): Global interrupt control bit

“**0**”**:** Disable all interrupts
“**1**”**:** Enable all un-mask interrupts

8.10.3 SPI Timing Diagrams

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

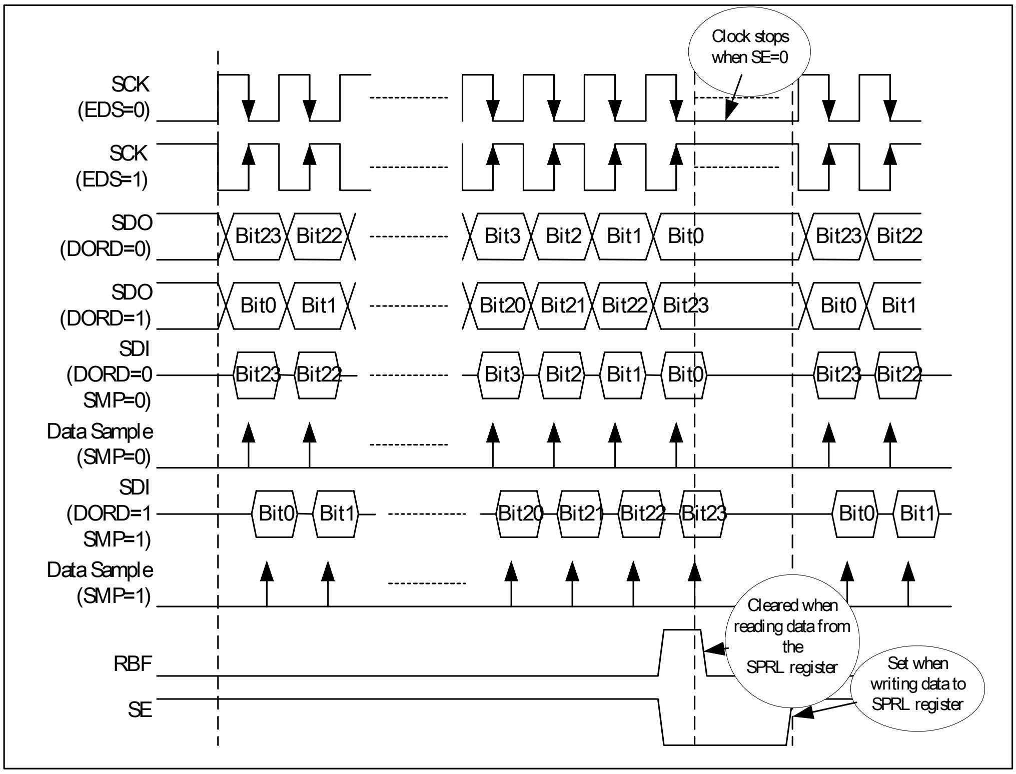


Figure 8-27a SPI Master Mode Timing Diagram

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

```
;*** Interrupt SPI
PERIPH:
    PUSH
    COMA     DATACNT
;--- SPI read buffer full
    JBC      SPISTA,SRBFI,Q_SPINT
    BC       SPISTA,SRBFI
    BS       INTFLAG,F_SPI
;--- SPI Data collision
    JBC      SPISTA,DCOL,Q_SPINT
    MOV      A,#0XFF
Q_SPINT:
    MOV      DATACNT,A
    POP
    RETI
;=== 8MHz/4 = 2000000 bit rate
SPIM_SR:
       :
    System setting 8MHz
    Port G setting output port
       :
;--- 8bit, Fsystem/4, Rising edge & MSB
    MOV      A,#11001100B
    MOV      SPICON,A
```

```
;--- SPI full interrupt
    MOV      A,#00100000B
    MOV      SPISTA,A
;--- Global interrupt
    BS       CPUCON,GLINT
;--- SPI data output => 55
    MOV      A,#0X55
    MOV      DATACNT,A
SPI8LOOP:
    MOV      A,DATACNT
    MOV      SPRL,A
;--- SPI Data collision
    JBC      SPISTA,DCOL,SPI8LP1
    BC       SPISTA,DCOL
;--- SPI data output resend => 55
    MOV      A,#0X55
    MOV      DATACNT,A
SPI8LP1:
    JBC      INTFLAG,F_SPI,SPI8LP1
SPI8LP2:
    BC       INTFLAG,F_SPI
    MOVRP    PORTG,SPRL
    SJMP     SPI8LOOP
```

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

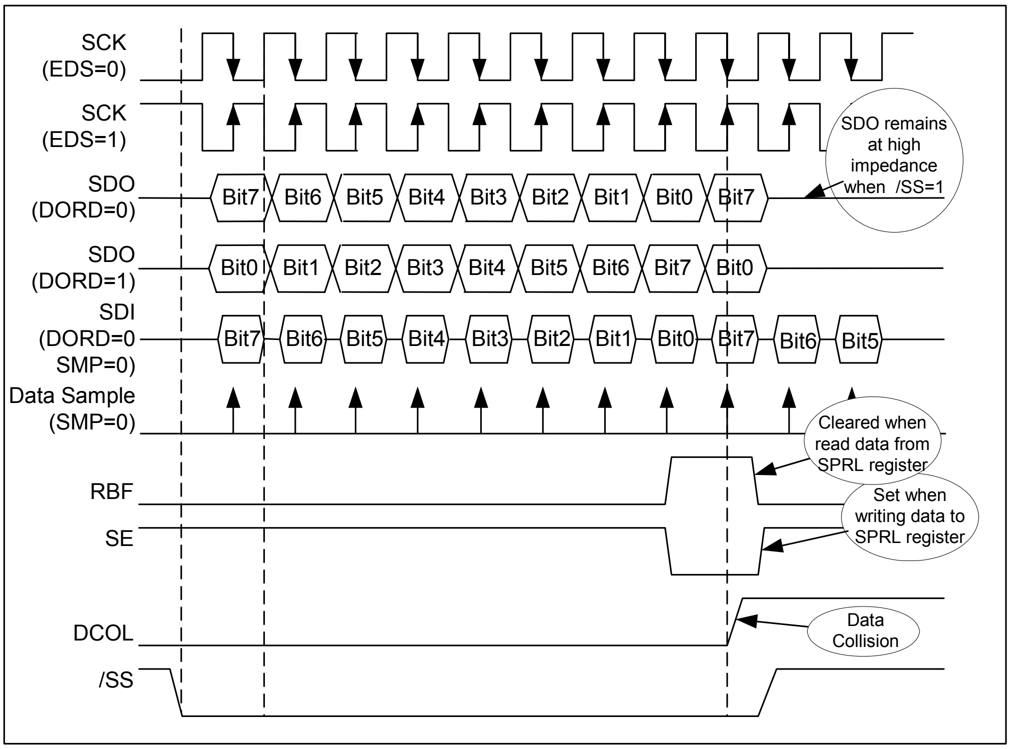


Figure 8-27b SPI Slave Mode Timing Diagram

Code Example: Slave Mode (8bit)

```
; *** Interrupt SPI
PERIPH:
      PUSH
      JBC    SPISTA,SRBFI,Q_SPINT
      BC     SPISTA,SRBFI
      BS     INTFLAG,F_SPI
Q_SPINT:
      POP
      RETI
; *** SPI slave mode
        :
      System setting 8MHz
      Port G setting output port
        :
; === SPI 8bit & Sleep mode
SPIS_SR:
; --- 8bit, Slave /SS enable, Rising edge & LSB
      MOV    A,#11110100B
      MOV    SPICON,A
```

```
; --- SPI Wakeup & SPI full interrupt
     MOV       A,#00101000B
     MOV       SPISTA,A
; --- Global interrupt
     BS        CPUCON,GLINT
; --- Sleep mode
     BC        CPUCON,MS1
SPIS8Lp:
     SLEP
     NOP
     MOVRP     PORTG,SPRL
     BC        INTFLAG,F_SPI
; --- SPI Data collision
     JBC       SPISTA,DCOL,SPIS8Lp
     MOV       A,#0XFF
     MOV       SPRL,A
     SJMP      SPIS8Lp
```

8.11 Melody/Speech Synthesizer

The EPD3338 MCU provides four channels for melody/speech function. Channels 1~3 are destined for melody channel, and Channel 4 can be either a melody or a speech channel as determined by SPHSB bit (Bit 2 of R44). Channels 1 ~ 4 are controlled by R45 ~ R4A of the corresponding control register Banks 0 ~ 3. Bits 0 ~ 2 of R44 are used to select the current control register bank.

	Melody Channel 1	Melody Channel 2	Melody Cannel 3	Melody Channel 4/ Speech Channel
R44h	×××× ×000	×××× ×001	×××× ×010	×××× ×011 / ×××× ×1××
R45h	ADDL	ADDL	ADDL	ADDL / -
R46h	ADDM	ADDM	ADDM	ADDM / -
R47h	ADDH	ADDH	ADDH	ADDH / -
R48h	ENV	ENV	ENV	ENV / SPHDR
R49h	MTCON	MTCON	MTCON	MTCON / SPHTCON
R4Ah	MTRL	MTRL	MTRL	MTRL / SRHTRL

■ **SFCR (R44h):** Special Function Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AGMD2	AGMD1	AGMD0	WDTPSR1	WDTPSR0	SPHSB	CSB1	CSB0

Bits 0 ~ 1 (CSB0 ~ CSB1): Channel select bits

Bit 2 (SPHSB): Speech Channel/Melody Channel 4 select bit

"**0**": Melody Channel 4 enabled, Speech channel disabled

"**1**": Melody Channel 4 disabled, Speech channel enabled

SFCR[2:0]	Channel Selection	Control Register Bank
000	Melody Channel 1	Bank 0
001	Melody Channel 2	Bank 1
010	Melody Channel 3	Bank 2
011	Melody Channel 4	Bank 3
1××	Speech Channel	Bank 3

8.11.1 Melody Function

The MCU melody function can effectively manage the instrument waveform address setting, instrument synthesis frequency control, and envelope control. It is embedded with four melody channels and with built-in large data ROM size for melody waveform data storage. To synthesize the instrument melody, you should write the starting address of the waveform to R45 ~ R47, setup the envelope value, and then enable the melody timer. The control registers are listed as follows**:**

- **ADDH, ADDM, ADDL (R47h ~ R45h):** Address Registers (Write-only registers)

These registers, i.e., ADDL, ADDM, and ADDH are treated as instrument waveform address. Each melody channel has its own waveform data address pointer that points to the waveform start address in the data ROM. The address values are written by the program and its total length is 24 bits.

- **ENV (R48h):** Envelope Register

The envelope register stores the envelope value for the current melody channel. The user's program should calculate the proper envelope value to obtain a suitable ADSR (Attack-Decay-Sustain-Release) for different instruments. The tone generator will automatically process the waveform data with the envelope and then synthesize the final instrument melody to the mixer of the PWM and D/A converter.

The data written to the envelope register should be a 7-bit unsigned value and located in Bits 0~6 (the corresponding envelope value must be 0 to 127), which means the envelope resolution is of 128 steps. The reset initial value is "**0**."

- **MTRL (R4Ah**): Melody Timer Auto-reload Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MTRL7	MTRL6	MTRL5	MTRL4	MTRL3	MTRL2	MTRL1	MTRL0

Melody timer is an 11-bit down counter for melody applications. The frequency generated by the melody timer is determined by the value of 11-bit melody timer auto-reload register (including MTRL and MTRLH0~2 of MTCON). When the counter value underflows, the timer will auto-reload. To obtain the correct frequency, consult a frequency reference table and fetch the correct value for MTRL and MTRLH0~2 of MTCON.

- **MTCON (R49h):** Melody Timer Control Register

The MTCON is used to determine the three MSB's of the 11-bit auto-reload register and to enable/disable the melody timer of the current melody channel. Once the melody timer is enabled, it will fetch the waveform data (pointed to by the address registers) from the data ROM, process the data with the envelope, and then automatically feed the data to the DAC or PWM mixer. . The reset initial value of the MTCON is "×××× 0000".

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	-	-	MTEN	MTRLH2	MTRLH1	MTRLH0

Bits 0 ~ 2 (MTRLH0 ~ MTRLH2): Bit 8 ~10 of the melody timer auto-reload register

Bit 3 (MTEN): Melody Timer Enable Control Bit

MTEN	Melody Timer Enable or Disable
0	Melody Timer Disable
1	Melody Timer Enable

8.11.2 Speech Function

The 11-bit speech timer is shared with a melody timer (MT4) for Channel 4. The clock source for the speech timer is from $F_{PLL}/2$. When R44 [2:0] = “1××,” the control register bank will change to speech channel. An interrupt function is available for your application. The control registers are listed as follow:

$$Sampling\_rate = \frac{F_{PLL}/2}{SPHTRL[10:0]+1}$$

■ **SPHDR (R48h):** Speech Data Register

In speech function control, SPHDR acts as an output window to the PWM and D/A converter mixer. The program should write the synthesized data to SPHDR, and the data is fed into the mixer at the next speech timer underflow. For correct mixing operation, the value to be written to SPHDR must be an 8-bit signed data. The reset initial value is “0”.

■ **SPHTRL (R4Ah):** Low byte of Speech Timer Auto-reload Register

The Speech timer is an 11-bit down counter for speech applications. The frequency generated by the speech timer is determined by the value of the 11-bit auto-reload register, including SPHTRL and SPHTRLH0 ~ SPHTRLH2 of SPHTCON. When the counter value underflows, the timer interrupt will occur and auto-reload from the 11-bit auto-reload register.

■ **SPHTCON (R49h):** Speech Timer Control Register

SPHTCON is used to determine the three MSB of the 11-bit auto-reload register and enable/disable the speech timer. The reset initial value of SPHTCON is “××00 0000”.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	SPHTI	SPHTIE	SPHTEN	SPHTRLH2	SPHTRLH1	SPHTRLH0

Bits 0~2 (SPHTRLH0~ SPHTRLH2): Bits 8 ~ 10 of the 11-bit auto-reload register

Bit 3 (SPHTEN): Speech Timer Enable Control Bits

SPHTEN	Speech Timer Enable or Disable
0	Speech Timer Disabled
1	Speech Timer Enabled

Bit 4 (SPHTIE): Speech Timer interrupt control bit
"**0**"**:** Disable interrupt function
"**1**"**:** Enable interrupt function

Bit 5 (SPHTI): Speech timer interrupt flag. Set to "**1**" when the speech timer interrupt occurs. Clear to "**0**" by software or disable the speech timer.

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

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

Bit 2 (GLINT): Global interrupt control bit
"**0**"**:** Disable all interrupts
"**1**"**:** Enable all un-masked interrupts

8.12 PWM / DAC Function

The EPD3338 is embedded with two choices of melody/speech outputs, i.e., PWM and D/A converter.

When the PWM function is enabled, the voice output uses PWM to drive the speaker directly. The 8-bit PWM function block diagram is shown in the figure below. The PWD register is double buffered for glitch free operation.

When Bit 7 of PWD is "**1**" and the PWM timer counter equals to PWM value (Bits 0 ~ 6 of PWD), the VO1 will transfer to low until the PWM timer is reset or overflowed. The VO2 is always kept at "**0**" in this case.

When the Bit 7 of PWD is "**0**" and the PWM timer counter equals to the inverse of Bits 0 ~ 6 of PWD, the VO2 will transfers to low until the PWM timer is reset or overflowed. The VO1 is always kept at "0" in this case.

$$T_{period} = \frac{128}{F_{PLL}} \times Prescaler; \quad T_{duty} = \frac{1}{F_{PLL}} \times Prescaler \times (PWD + 1)$$

8.12.1 PWM Functional Block Diagram

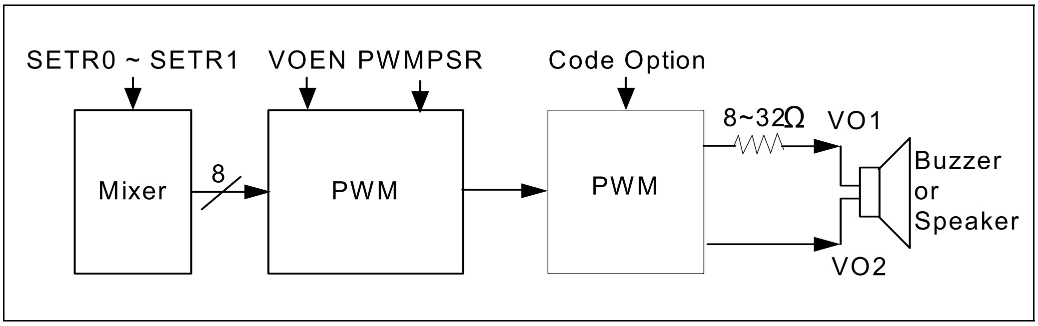


Figure 8-28 PWM Function Block Diagram

8.12.2 DAC Functional Block Diagram

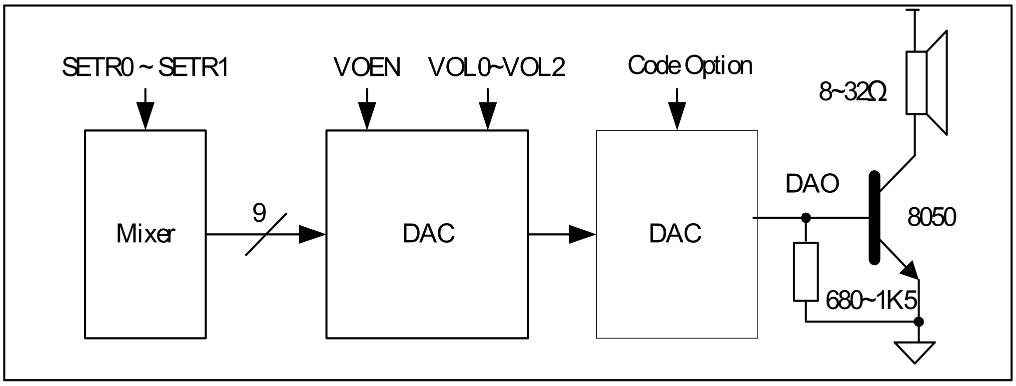


Figure 8-29 DAC Function Block Diagram

If both SPHSB and VOEN bits are set to "**1**" and SPHTEN bit is cleared to "**0**", the data of the speech data register will output immediately through the D/A converter or PWM when the register changes.

8.12.3 PWM / DAC Function Registers

■ **VOCON (R4Bh):** Voice Output Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
VOEN	-	SETR1	SETR0	PWMPSR	VOL2	VOL1	VOL0

Bits 0 ~ 2 (VOL0 ~ VOL2): Volume control of DAC

VOL2 ~ VOL0	Volume
000	1 (min.)
001	2
010	3
011	4
100	5
101	6
110	7
111	8 (max.)

Bit 3 (PWMPSR): PWM Timer prescaler select bit

"**0**": Prescaler 1:1

"**1**": Prescaler 1:2

Bits 5 ~ 4 (SETR1 ~ SETR0): Set dynamic range

While mixing, the mixer accumulated result may have a large dynamic range (up to 11-bit), while DAC has only 9-bit resolution and PWM has only 8-bit. You can define a suitable output data range to prevent the saturation condition from occurring.

SETR1~SETR0	Output Data fed to PWM/DAC
10	Take Bits 3~10 of mixer accumulation result for PWM Take Bits 2~10 of mixer accumulation result for DAC
01	Take Bits 2~9 of mixer accumulation result for PWM Take Bits 1~9 of mixer accumulation result for DAC
00 or 11	Take Bits 1~8 of mixer accumulation result for PWM Take Bits 0~8 of mixer accumulation result for DAC

Bit 7 (VOEN): Voice output control bit

"**0**"**:** DAC/PWM disabled

"**1**"**:** DAC/PWM enabled

■ **Code Example:** Refer to Melody & Speech Application Notes

9 Electrical Characteristics

9.1 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
Power Dissipation (Topr=70ºC)	PD	-	300	mW
Operating temperature range	TOPR	-	-10 to +70	°C
Storage temperature range	TSTR	-	-55 to +125	°C

9.2 Recommended Operating Conditions

Items	Sym.	Condition	Limits	Unit
Supply voltage	VDD	-	2.2 to 3.6	V
	AVDD		2.4 to 3.6	
Input voltage	VIH	-	VDD × 0.9 to VDD	V
	VIL	-	0 to VDD × 0.1	V
A/D full-Scale input span	ADRG	Positive input-negative input	0 to VREX	V
Operating temperature	TOPR	-	-10 to +70	°C

9.3 DC Electrical Characteristics

Condition: Ta=-10ºC ~ +70ºC, VDD= 3.0V ± 0.3V

Parameter	Sym.	Condition		Min.	Typ.	Max.	Unit
Clock	Fmain	Main-clock frequency		1	-	16	MHz
	Fsub	Sub-clock frequency	RC OSC	24.6	32.8	41	kHz
			Crystal OSC	-	32.768	-	
Supply Current	Idd1	Sleep mode	VDD = 3V, No load	-	-	1	µA
	Idd2	Idle mode	VDD = 3V RC OSC, LCD disable	-	8	12	
	Idd3		VDD = 3V, Crystal OSC, LCD disable	-	5	8	
	Idd4		Normal current: VDD = 3V, Crystal / RC OSC, LCD enable, Vout = 2VDD, code option: V0 & V1~V4 OP on (No load)	-	80	100	
			Small current: VDD = 3V, Crystal / RC OSC, LCD enable (Partial display mode, Vout = VDD) code option: V0 OP off (=<Vdd, external voltage), & V1~ V4 OP on (No load)	-	40	60	
	Idd5	Slow mode	VDD = 3V, LCD disable, RC/Crystal OSC, No load	-	20	30	
	Idd6	Fast mode	VDD = 3V, Fmain = 4MHz, No load	-	1200	1600	
	Idd7		VDD = 3V, Fmain = 10MHz, No load	-	2200	2900	
			VDD = 3V, Fmain = 15MHz, No load	-	3500	4800	
Input Voltage	VIH1	PA[0:7], PB[0:2,5:7], PC[2:7], PD[4:7], PG[0:7] PH[0:7] (as general input port)		VDD×0.7	-	VDD	V
	VIL1			0	-	VDD×0.3	
Input Threshold Voltage (Schmitt)	VT+	RSTB, PB.5 (as EVIN or CPIN)		0.5×VDD	-	0.75×VDD	V
	VT-			0.2×VDD	-	0.4×VDD	
Output Current	IOH1	PB[7:6],PB[5],PB[1:0],PC[2:7], PD[7:4] (as general output port),	VDD = 3V, VOH = 2.4V	-1.1	-2.2	-3.3	mA
	IOL1		VDD = 3V, VOL = 0.2V	+1.1	+2.2	+3.3	
	IOH2	PB[1] (as D/A output)	VDD = 3.0V, VOH = 0.7V	-2.5	-3.5	-4.5	
	IOH3	PB[1:0] (as PWM output)	VDD = 3.0V, VOH = 1.5V	-200	-300	-400	
	IOL3		VDD = 3.0V, VOL = 1.5V	+200	+300	+400	
	IOH5	PG[7:0]~PH[0:7]	VDD = 3.0V, VOH = 2.4V	-1.6	-2.6	-3.6	mA
	IOL5		VDD = 3.0V, VOL = 0.2V	+1.1	+2.2	+3.3	mA
	IOH6	PB[2] (as IR output pin)	VDD = 3.0V, VOH = 2.1V	-5	-10	-15	mA
	IOL6		VDD = 3.0V, VOL = 0.9V	+4	+8	+12	mA
Input Leakage Current	IIL	All Input port (without pull up/down resistor) Vin = VDD or GND		-	-	+/-1	µA
Large Pull-up Resistance	RPU1	PA[6:0]	Key high resistance, pulled up by R2, LCD enable (BOOST = 1, normal display mode), Vin = GND, VDD = 3V	100	200	300	KΩ
	RPU3	PA[7], PB[0:2,5:7], PC[2:7], PD[4:7]	Vin = GND, VDD = 3V	300	800	1300	
		PG[7:0]~PH[7:0]	Vin = GND, VDD = 3V	50	150	250	
	RPU5	RSTB	Vin = GND, VDD = 3V	250	500	750	

(Continuation)

Parameter	Sym.	Condition		Min.	Typ.	Max.	Unit
Small Pull-up Resistance	RPU2	PA[6:0]	Key low resistance, pulled up by R1//R2, LCD enable (BOOST = 0, normal display), Vin = GND, VDD = 3V	10	25	40	KΩ
			Key low resistance, pulled up by R1//R2, LCD enable (BOOST = 1, normal display), Vin = GND, VDD = 3V	15	30	45	
	RPU4	PA[7], PB[0:2,5:7], PC[2:7], PD[4:7]	Vin = 2V, VDD = 3V	50	100	200	
	RPU6	RSTB	Vin = 2V, VDD = 3V	50	100	200	
Large Pull-down Resistance	RPD1	TEST	Vin = VDD, VDD = 3V	250	500	750	KΩ
Small Pull-down Resistance	RPD2	TEST	Vin = 1V, VDD = 3V	1.1	2.2	3.3	KΩ
Touch Panel Pull-down Resistance	RPD3	DET = 1, Xn pin	Vin = VDD, VDD = 3V	25	50	100	KΩ
Data Retention Voltage	Vret	-		1.6	–	–	V
Power-on Reset Voltage	Vpor	-		1.4	1.5	1.6	V
A/D Conversion (VDD = 3.0V, AVDD = 3.0V, Ta = -10 ~ +70°C, Fclk = 12*Fsample)							
Analog Input							
Mux Leakage Current	Imux	On/off leakage current, Vin = 0 or VDD		–	0.1	1	μA
System Performance							
Resolution				–	10	–	Bits
Integral Non-Linearity	INL			-2	–	+2	LSB
Differential Non-Linearity	DNL			-2	–	+2	LSB
Offset Error	OErr			-4	–	+4	LSB
Gain Error	GErr			-4	–	+4	LSB
Missing Code	MC			No missing code			Bit
AVDD Supply Current	Ivdd3	AVDD = 3.0V, VDD = 3.0V, Fsample = 20kHz, ADEN = 1, VRS = 1		–	0.5	0.7	mA
	Ivdd4	ADEN = 0, VRS = 1		–	–	1	uA
Driver Current	IOH	Xp, Yp (VDD = 2.9 ± 0.3V) (Voh = VDD - 0.2V)		-20	-30	-45	mA
Sink Current	IOL	Xn, Yn (VDD = 2.9 ± 0.3V) (Vol = 0.2V)		+20	+30	+45	mA

(Continuation)

Parameter	Sym.	Condition		Min.	Typ.	Max.	Unit
Reference Voltage							
Internal Reference Voltage	VRIN	AVDD = 3.0 ± 0.3V		1.8	2.0	2.2	V
Internal Reference Supply Current	Ivrin	VDD = 3.0V, AVDD = 3.0V, VRS = 0, VOH = 0.2V		400	500	–	µA
VREX input current	Iref1	ADEN = 1, VRS = 1		–	300	500	µA
	Iref2	ADEN = 0, VRS = 1		–	–	1	µA
LCD Driver							
Reference Voltage	Vref1	Ta = 20°C [1]		2.035	2.12	2.205	V
	Vref2	Ta = 0°C [1]		2.169	2.26	2.351	V
	Vref3	Ta = 40°C [1]		1.900	1.98	2.060	V
Charge Pump Output	Vout	2 times pumping (charge pump capacitor C1: 0.1µF)		2*Vdd-5%	2*Vdd	–	V
		3 times pumping (charge pump capacitor C1 and C2: 0.1µF)		3*Vx-5%	3*Vx	–	V
Clamping Voltage	Vx	BOOST = 0, 2 times pumping		–	Vdd	–	V
		BOOST = 1, 3 times pumping		2.3	2.4	2.5	
Regulated Voltage	V0	VDD = 2.3V~3.3V, Ta = 25°C		V0-10%	V0 [2]	V0+10%	V
LCD Display Output ON-Resistance	ROC	Com[0:31]	VOH = V0 ± 0.2V	1	2	3	KΩ
			VOM = V1 ± 0.2V				
			VOM = V4 ± 0.2V				
			VOL = 0.2V				
	ROS	Seg [0 : 63, 80:111]	VOH = V0 ± 0.2V	1	2	3	KΩ
			VOM = V2 ± 0.2V				
			VOM = V3 ± 0.2V				
			VOL = 0.2V				
Strobe Output ON-Resistance	ROP	Seg [0:15] (as key strobe)	V = VDD - 0.2V	45	70	100	KΩ
	RON		V = 0.2V	0.7	1.0	1.5	
Display Frame Frequency	Frame	Sub-Clock : RC OSC		48	–	100.5	Hz
		Sub-Clock : Crystal OSC		64	–	80.4	

(Continuation)

Parameter	Sym.	Condition		Min.	Typ.	Max.	Unit
Op. Amp Voltage Output of LCD Power Supply	Vout0	V0	No load	-[3]	V0 [2]	-[3]	mV
	Vout1	V1		-[3]	V1 [2]	-[3]	
	Vout2	V2		-[3]	V2 [2]	-[3]	
	Vout3	V3		-[3]	V3 [2]	-[3]	
	Vout4	V4		-[3]	V4 [2]	-[3]	

[1] *Typical regulated voltage for V0 is selected by software from the table shown below.*
[2] *V0~V4 are Theoretical values*
[3] *The Target value of V0~V4 is Theoretical value ± 50mV*

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

9.4 AC Electrical Characteristics

Condition: Ta=-10~+70°C, VDD= 3.0 ± 0.3V

Parameter	Sym.	Condition	Min	Typ	Max	Unit
Instruction Cycle Time	Tcycle	Fmain = 1MHz	-	2*	-	µs
		Fmain = 4MHz	-	0.5*	-	
		Fmain = 10MHz	-	0.2*	-	
		Fmain = 15MHz	-	0.13*	-	
A/D Conversion (VDD = 3.0V, AVDD = 3.0V, Ta = -10~+70°C)						
Throughput Rate		VDD = 3.0V, AVDD = 3.0V	-	-	80	ksps
		VDD = 2.4V, AVDD = 2.4V	-	-	60	
Power Supply Rejection Ratio	PSRR1+	Power noise: 1kHz, 100mV	37	40	-	dB
	PSRR1-	Power noise: 1kHz, 100mV	43	46	-	
Signal to Noise Ratio	SNR		51	54	-	dB

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

10 Application Circuits

■ **EPD3338 Full-Scale Applications**

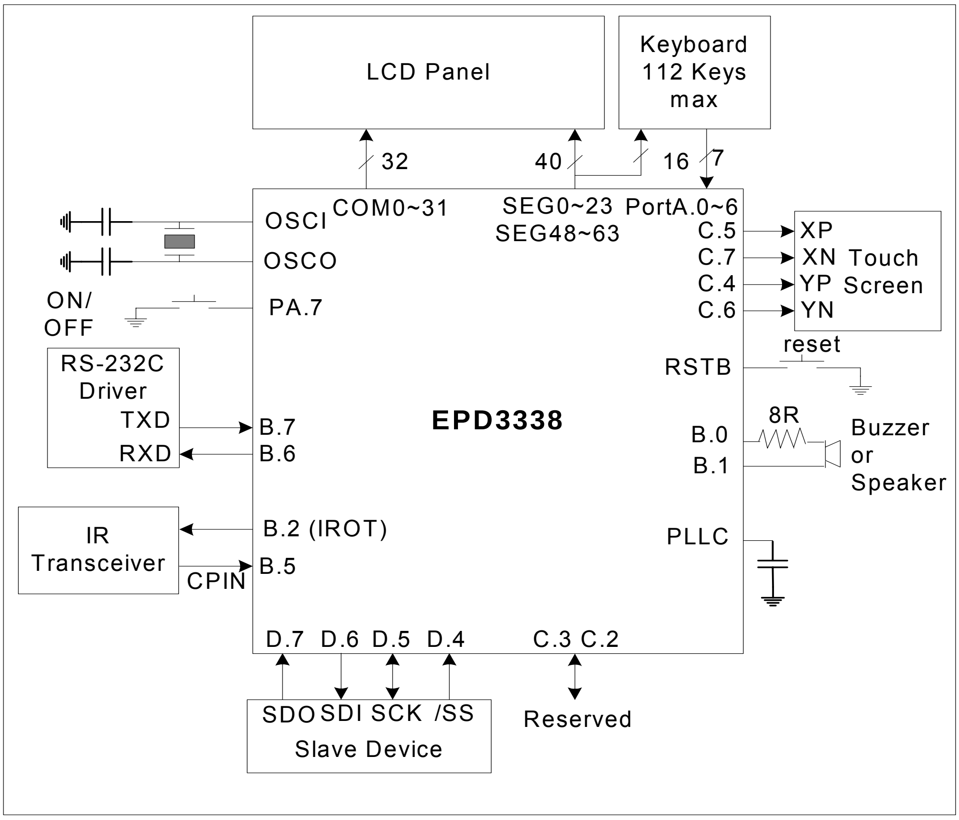


Figure 10-1a Full-Scale Application Circuit Diagram

■ **32×40 pixels driving application circuits** ("Single-chip" using internal oscillator)

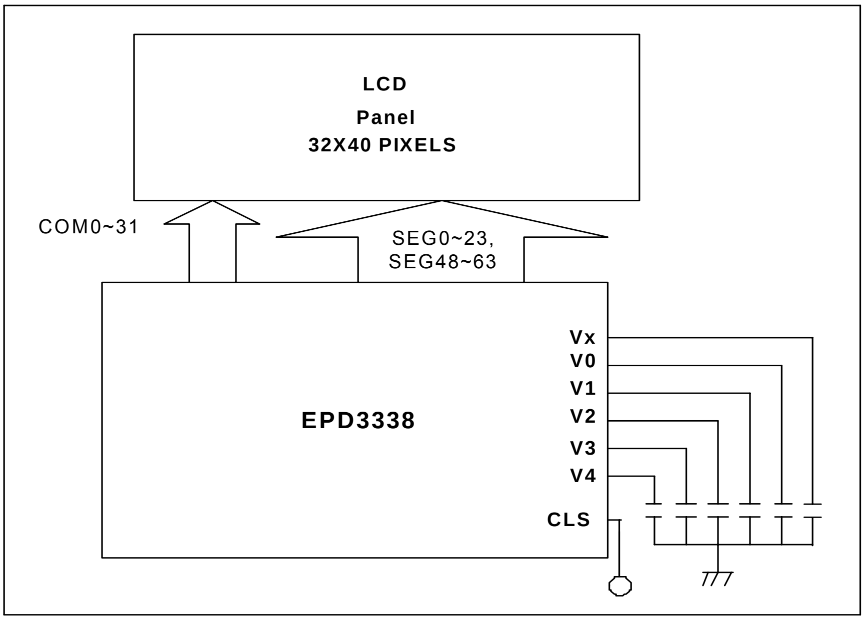


Figure 10-1b Driving 32×40 Pixels LCD Panel Application Circuit Diagram

11 Instruction Set

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

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

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

(Continuation)

Type	Binary Instruction	Mnemonic	Operation	Status Affected	Cycles
Arithmetic Operation	0010 0110 rrrr rrrr	MUL A,r	PRODH:PRODL ← A r	None	1
	0100 1111 kkkk kkkk	MUL A,#k	PRODH:PRODL ← A k	None	1
	0001 0100 rrrr rrrr	ADDDC A,r	A ← (Decimal ADD) A+r+C	C, DC, Z	1
	0001 0101 rrrr rrrr	ADDDC r,A	r ← (Decimal ADD) r+A+C	C, DC, Z	1
	0001 1010 rrrr rrrr	SUBDB A,r	A ← (Decimal SUB) r-A-/C	C, DC, Z	1
	0001 1011 rrrr rrrr	SUBDB r,A	r ← (Decimal SUB) r-A-/C	C, DC, Z	1
Logic Operation	0000 0010 rrrr rrrr	OR A,r	A ← A .or. r	Z	1
	0000 0011 rrrr rrrr	OR r,A	r ← r .or. A	Z	1
	0100 0100 kkkk kkkk	OR A,#k	A ← A .or. k	Z	1
	0000 0100 rrrr rrrr	AND A,r	A ← A .and. r	Z	1
	0000 0101 rrrr rrrr	AND r,A	r ← r .and. A	Z	1
	0100 0101 kkkk kkkk	AND A,#k	A ← A .and. k	Z	1
	0000 0110 rrrr rrrr	XOR A,r	A ← A .xor. r	Z	1
	0000 0111 rrrr rrrr	XOR r,A	r ← r .xor. A	Z	1
	0100 0110 kkkk kkkk	XOR A,#k	A ← A .xor. k	Z	1
	0000 1000 rrrr rrrr	COMA r	A ← /r.	Z	1
	0000 1001 rrrr rrrr	COM r	r ← /r.	Z	1
Rotate	0000 1010 rrrr rrrr	RRCA r	A(n-1) ←r(n);C ←r(0);A(7) ←C	C	1
	0000 1011 rrrr rrrr	RRC r	r(n-1) ←r(n);C ←r(0);r(7) ←C	C	1
	0000 1100 rrrr rrrr	RLCA r	A(n+1) ←r(n);C ←r(7);A(0) ←C	C	1
	0000 1101 rrrr rrrr	RLC r	r(n+1) ←r(n);C ←r(7);r(0) ←C	C	1
Shift	0010 0010 rrrr rrrr	SHRA r	A(n-1)←r(n); A(7)←C	None	1
	0010 0011 rrrr rrrr	SHLA r	A(n+1)←r(n); A(0)←C	None	1
Bit Compare & Jump	0101 1bbb rrrr rrrr aaaa aaaa aaaa aaaa	JBC r,b,addr	If r(b)=0,jump to addr; PC[15:0] ← addr.[5]	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.[5]	None	2
Compare	0010 0101 rrrr rrrr	TEST r	Z ← 0 if r<>0;Z ← 1 if r=0	Z	1
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.[5]	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.[5]	None	2
	0101 0010 rrrr rrrr aaaa aaaa aaaa aaaa	JINZ A,r,addr	A ← r+1,jump to addr if not zero; PC[15:0] ← addr.[5]	None	2
	0101 0011 rrrr rrrr aaaa aaaa aaaa aaaa	JINZ r,addr	r ← r+1,jump to addr if not zero; PC[15:0] ← addr.[5]	None	2
	0100 0111 kkkk kkkk aaaa aaaa aaaa aaaa	JGE A,#k,addr	Jump to addr if A≧k; PC [15:0] ← addr.[5]	None	2
	0100 1000 kkkk kkkk aaaa aaaa aaaa aaaa	JLE A,#k,addr	Jump to addr if A≦k; PC [15:0] ← addr.[5]	None	2
	0100 1001 kkkk kkkk aaaa aaaa aaaa aaaa	JE A,#k,addr	Jump to addr if A=k; PC[15:0] ← addr.[5]	None	2

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

(Continuation)

Type	Binary Instruction	Mnemonic	Operation	Status Affected	Cycles
Compare & Jump	0101 0101 rrrr rrrr aaaa aaaa aaaa aaaa	JGE A,r,addr	Jump to addr if A□r; PC[15:0] ← addr.[5]	None	2
	0101 0110 rrrr rrrr aaaa aaaa aaaa aaaa	JLE A,r,addr	Jump to addr if A□r; PC[15:0] ← addr.[5]	None	2
	0101 0111 rrrr rrrr aaaa aaaa aaaa aaaa	JE A,r,addr	Jump to addr if A=r; PC[15:0] ← addr.[5]	None	2
Jump	110a aaaa aaaa aaaa	SJMP addr	PC ← addr; PC [13..16] unchanged	None	1
	0000 0000 0010 aaaa aaaa aaaa aaaa aaaa	LJMP addr (2 words)	PC ← addr.	None	2
Subroutine	0011 aaaa aaaa aaaa	S0CALL addr	Top of Stack] ← PC+1; PC [11:0] ← addr; PC [12:16] ← 00000 [6]	None	1
	111a aaaa aaaa aaaa	SCALL addr	[Top of Stack] ← PC+1; PC [12:0] ← addr; PC [13:16] unchanged.	None	1
	0000 0000 0011 aaaa aaaa aaaa aaaa aaaa	LCALL addr (2 words)	[Top of Stack] ← PC+1; PC ← addr	None	2
	0010 1011 1111 1110	RET	PC ← (Top of Stack)	None	1
	0010 1011 1111 1111	RETI	PC ← (Top of Stack); Enable Interrupt	None	1

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

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

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

[4] *When in SUB operation, borrow flag is indicated by the inverse of carry bit, i.e., B = /C.*

[5] *The maximum jump range is 64K absolute address, which means it can only jump within the same 64K range.*

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

12 Pad Diagram

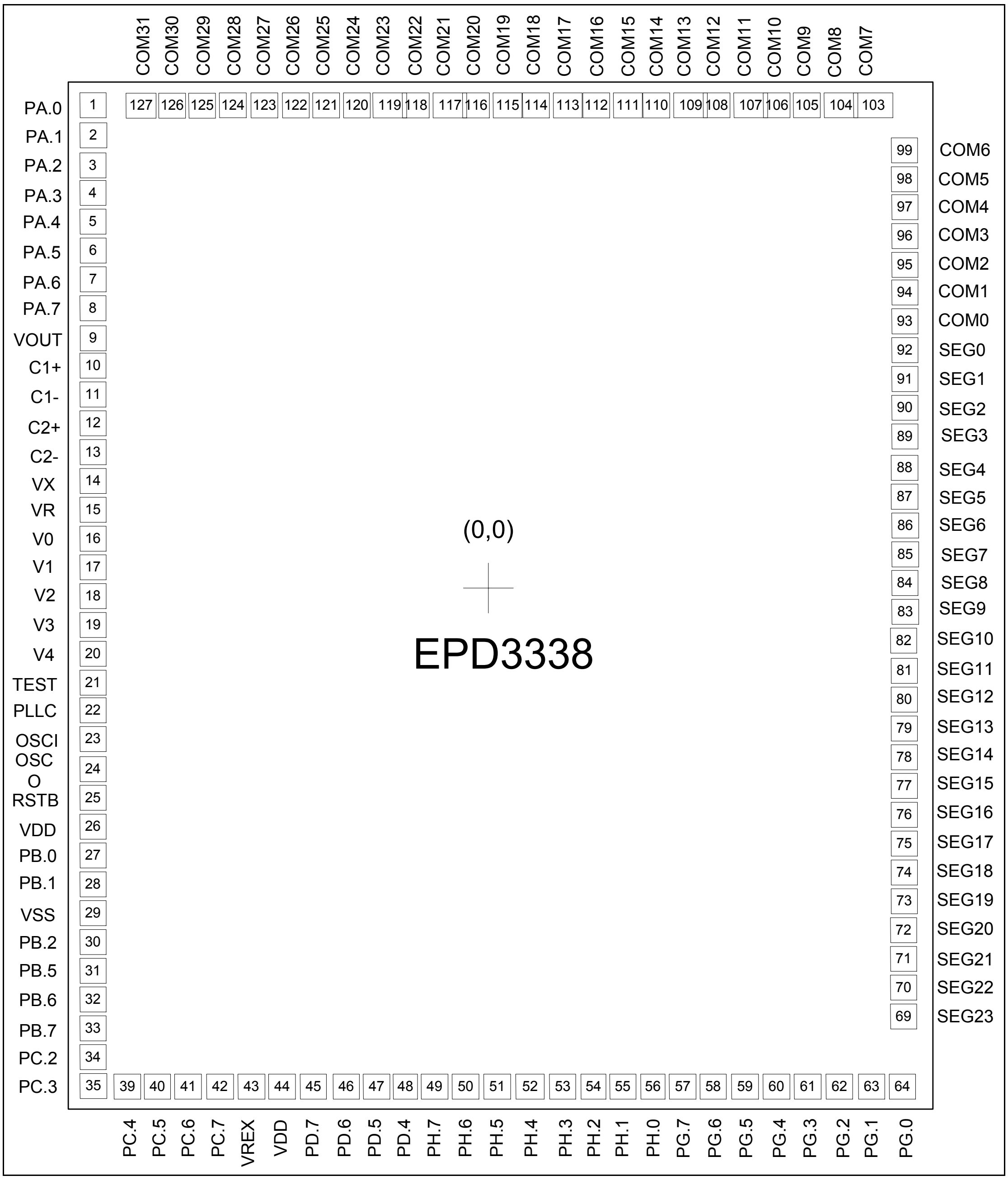


Figure 12-1 EPD3338 Pad Locations

Chip size: 3140 * 3890 μm^2

■ Pad Coordinates

Pin No.	Symbol	X	Y	Pin No.	Symbol	X	Y
1	PA_0	-1451.1	1806.65	65	NC		
2	PA_1	-1451.1	1696.65	66	NC		
3	PA_2	-1451.1	1586.65	67	NC		
4	PA_3	-1451.1	1476.65	68	NC		
5	PA_4	-1451.1	1361.65	69	SEG_23	1449.1	-1527
6	PA_5	-1451.1	1251.75	70	SEG_22	1449.1	-1422
7	PA_6	-1451.1	1141.65	71	SEG_21	1449.1	-1317
8	PA_7	-1451.1	1031.65	72	SEG_20	1449.1	-1212
9	VOUT	-1443	904.9	73	SEG_19	1449.1	-1107
10	C1P	-1443	799.15	74	SEG_18	1449.1	-1002
11	C1N	-1443	696.65	75	SEG_17	1449.1	-897
12	C2P	-1443	594.15	76	SEG_16	1449.1	-792
13	C2N	-1443	491.65	77	SEG_15	1449.1	-687
14	VX	-1443	389.15	78	SEG_14	1449.1	-582
15	VR	-1443	286.65	79	SEG_13	1449.1	-477
16	V0	-1443	184.15	80	SEG_12	1449.1	-372
17	V1	-1443	81.65	81	SEG_11	1449.1	-267
18	V2	-1443	-20.85	82	SEG_10	1449.1	-162
19	V3	-1443	-123.35	83	SEG_9	1449.1	-57
20	V4	-1443	-225.85	84	SEG_8	1449.1	48
21	TEST	-1443	-334.55	85	SEG_7	1449.1	153
22	PLLC	-1443	-439.55	86	SEG_6	1449.1	258
23	OSCI	-1443	-544.55	87	SEG_5	1449.1	363
24	OSCO	-1443	-649.55	88	SEG_4	1449.1	468
25	RSTB	-1443	-754.55	89	SEG_3	1449.1	573
26	VDD	-1443	-859.55	90	SEG_2	1449.1	678
27	PB_0	-1443	-964.55	91	SEG_1	1449.1	783
28	PB_1	-1443	-1069.55	92	SEG_0	1449.1	888
29	GND	-1442.95	-1175.7	93	COM_0	1449.1	993
30	PB_2	-1442.95	-1281.7	94	COM_1	1449.1	1098
31	PB_5	-1442.95	-1387.7	95	COM_2	1449.1	1203
32	PB_6	-1442.95	-1493.7	96	COM_3	1449.1	1308

Pin No.	Symbol	X	Y	Pin No.	Symbol	X	Y
33	PB_7	-1442.95	-1600.7	97	COM_4	1449.1	1413
34	PC_2	-1442.95	-1707.7	98	COM_5	1449.1	1518
35	PC_3	-1442.95	-1818	99	COM_6	1449.1	1623
36	NC	-	-	100	NC	-	-
37	NC	-	-	101	NC	-	-
38	NC	-	-	102	NC	-	-
39	PC_4	-1320.45	-1818	103	COM_7	1287.1	1822.15
40	PC_5	-1205.45	-1818	104	COM_8	1166.1	1822.15
41	PC_6	-1090.45	-1818	105	COM_9	1061.1	1822.15
42	PC_7	-977.4	-1818	106	COM_10	956.1	1822.15
43	VREX	-867.4	-1818	107	COM_11	851.1	1822.15
44	VDD	-757.4	-1818	108	COM_12	746.1	1822.15
45	PD_7	-652.4	-1818	109	COM_13	641.1	1822.15
46	PD_6	-547.4	-1818	110	COM_14	536.1	1822.15
47	PD_5	-442.3	-1818	111	COM_15	431.1	1822.15
48	PD_4	-337.3	-1818	112	COM_16	310.1	1822.15
49	PH_7	-230.4	-1817.85	113	COM_17	205.1	1822.15
50	PH_6	-125.4	-1817.85	114	COM_18	100.1	1822.15
51	PH_5	-18.4	-1817.85	115	COM_19	-4.9	1822.15
52	PH_4	88.6	-1817.85	116	COM_20	-109.9	1822.15
53	PH_3	195.6	-1817.85	117	COM_21	-214.9	1822.15
54	PH_2	302.6	-1817.85	118	COM_22	-319.9	1822.15
55	PH_1	412.6	-1817.85	119	COM_23	-424.9	1822.15
56	PH_0	522.6	-1817.85	120	COM_24	-545.9	1822.15
57	PG_7	632.6	-1817.85	121	COM_25	-650.9	1822.15
58	PG_6	742.6	-1817.85	122	COM_26	-755.9	1822.15
59	PG_5	852.6	-1817.85	123	COM_27	-860.9	1822.15
60	PG_4	962.6	-1817.85	124	COM_28	-965.9	1822.15
61	PG_3	1076	-1817.85	125	COM_29	-1070.9	1822.15
62	PG_2	1191	-1817.85	126	COM_30	-1175.9	1822.15
63	PG_1	1311	-1817.85	127	COM_31	-1280.9	1822.15
64	PG_0	1431	-1817.85	128	NC		

NOTE

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