Elan Microelectronics Crop.

ELAN RISC II™ series

EPG1280

June 20, 2001
Version 1.2 (Preliminary)

Specification Revision History		
Version	**Content**	**Date**
1.0	Initial version	Feb. 7,2001
1.2	a. NEW CPU series rename to ***ELAN RISC II™ series*** b. Update TEST pin type c. Modify LVD spec to Vdet +/- 0.1V, and add the Vdet hysteresis voltage description d. Modify the DC Specification temperature range to Ta=0~+40°C e. EPG-1280 rename to **EPG1280.**	June 20,2001

INDEX

I. General Description

ELAN RISC II™ series is a 8 bit uC based data processor IC for use on palm top devices. Integrated on the chip are large ROM size and large RAM size,8*8 hardware multiplier, one SPI and UART, melody timer, EL generator, low voltage detector and external memory parallel interface.

II. Feature

(1) Kernel Feature:

- Operating voltage:2.2~3.6V.
- Maximum operation Speed: 10MHz clock.
- One Instruction cycle time = 2* System clock time
- 8 bit CPU with 80 instructions.
- 128K words program ROM.
- 128 bytes unbanked RAM, 32K bytes banked RAM .
- 1088 bytes LCD RAM.
- RAM stack can achieve maximum 128 levels.
- Enhanced TABLE LOOK UP function.(1 or 2 cycles)
- High capacity memory access ability.
- Register to Register instruction.
- Compare and Branch in one instruction (2 cycles).
- Single Repeat function.
- 8*8 hardware multiplier.
- Decimal ADD and SUB instruction.
- Full range CALL and JUMP ability (2 cycles).

(2) Peripheral Feature:

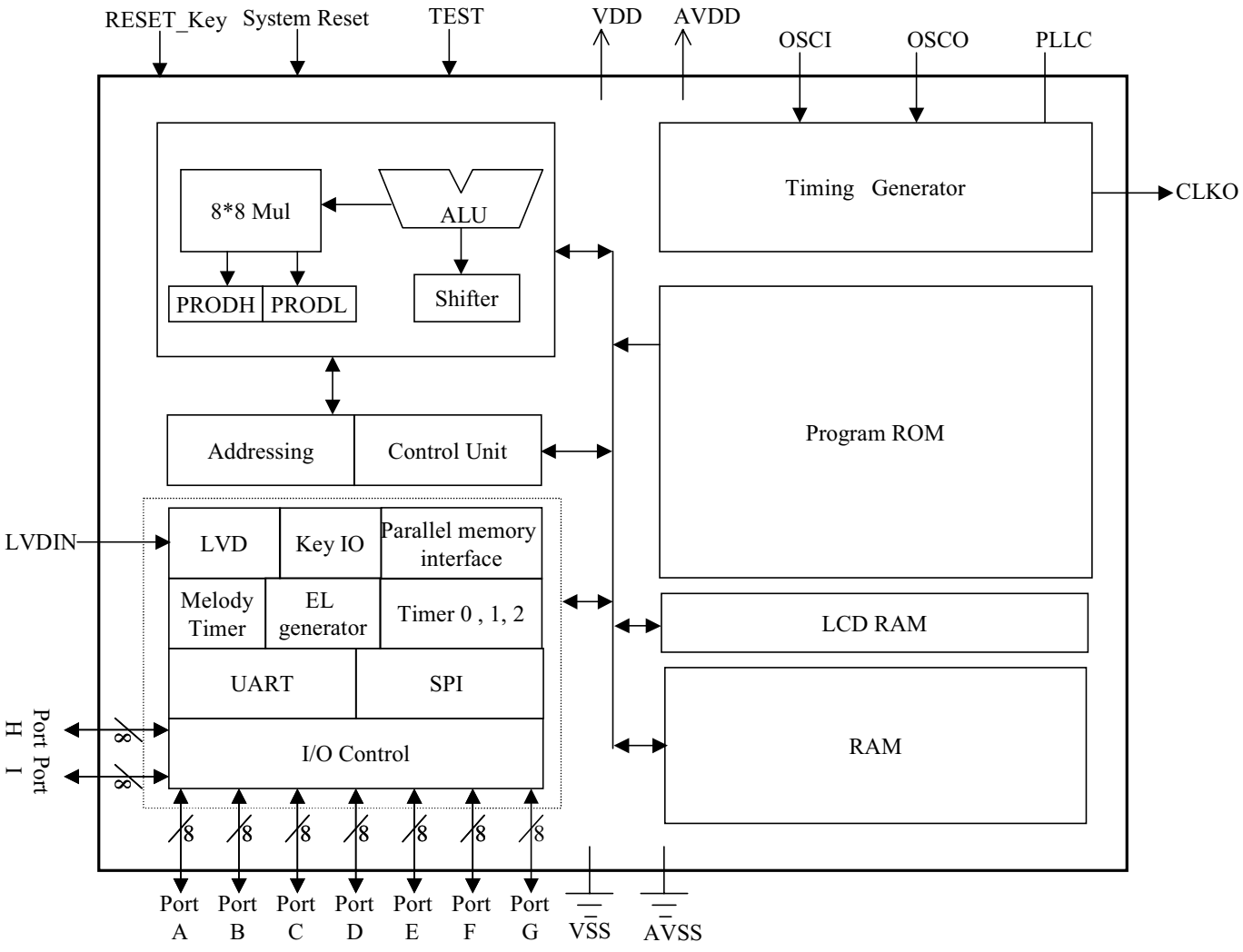
- 72 I/O pins (Port A ~ Port I)
- One 16 bits general purpose timer, two 8 bits general purpose timer, one melody timer
- EL generator.
- Low voltage detector.
- SPI (Serial Peripheral Interface)
- UART (Universal Asynchronous Receiver Transmitter)
- Clock output for external device(LCD driver…,etc.)

- Parallel External memory interface

(3) Special micro controller Feature:

- Watchdog Timer with its own on-chip RC oscillator
- CPU mode: SLEEP MODE, IDLE MODE, SLOW MODE, FAST MODE
- PLL frequency adjustable

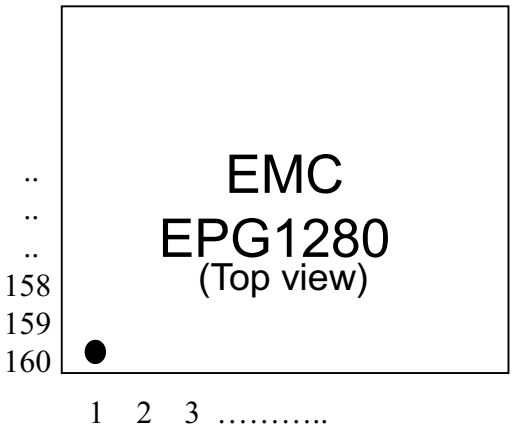
III. Block Diagram



IV. ELAN RISC II™ Series Part List

Part Number	PROM	RAM	LCD	Peripheral	I/O pins
EPG1280	**128k*16**	**32k*8**	**External**	**SPI, UART, Melody, EL , LVD, Parallel memory interface**	**72 (Port A~I)**
EPG6400	64k*16	8k*8	External	SPI, UART, Melody, EL , LVD, Parallel memory interface	72 (Port A~I)

V. Pin Assignment



No.	Pin NAME	No.	Pin NAME	No.	Pin NAME	No.	Pin NAME
1	NC	41	NC	81	NC	121	PG.3(D3)
2	NC	42	NC	82	NC	122	PG.4(D4)
3	LVDIN	43	NC	83	NC	123	PG.5(D5)
4	(Reserved)	44	NC	84	PC.6(UTXD)	124	PG.6(D6)
5	(Reserved)	45	NC	85	PC.7(URXD)	125	PG.7(D7)
6	(Reserved)	46	NC	86	*PD.7*	126	NC
7	(Reserved)	47	NC	87	*PD.6*	127	NC
8	(Reserved)	48	NC	88	*PD.5*	128	NC
9	AVDD	49	NC	89	*PD.4*	129	NC
10	PLLC	50	NC	90	*PD.3*	130	NC
11	VSS	51	NC	91	*PD.2*	131	NC
12	VSS	52	NC	92	*PD.1*	132	PH.0(Strobe 0)
13	OSCI	53	NC	93	*PD.0*	133	PH.1(Strobe 1)
14	OSCO	54	NC	94	*PF.7(A15)*	134	PH.2(Strobe 2)

15	TEST	55	NC	95	*PF.6(A14)*	135	PH.3(Strobe 3)
16	CLKO	56	NC	96	*PF.5(A13)*	136	PH.4(Strobe 4)
17	PA0	57	NC	97	*PF.4(A12)*	137	PH.5(Strobe 5)
18	PA.1	58	NC	98	*PF.3(A11)*	138	PH.6(Strobe 6)
19	PA.2	59	NC	99	*PF.2(A10)*	139	PH.7(Strobe 7)
20	PA.3	60	NC	100	*PF.1(A9)*	140	PI.0(Strobe 8)
21	PA.4	61	NC	101	*PF.0(A8)*	141	PI.1(Strobe 9)
22	PA.5	62	NC	102	*PE.7(A7)*	142	PI.2(Strobe 10)
23	PA.6	63	NC	103	*PE.6(A6)*	143	PI.3(Strobe 11)
24	PA.7(ON_key)	64	NC	104	*PE.5(A5)*	144	PI.4(Strobe 12)
25	PB.0(/SPISS)	65	NC	105	*PE.4(A4)*	145	PI.5(Strobe 13)
26	PB.1(SPISCK)	66	NC	106	*PE.3(A3)*	146	PI.6(Strobe 14)
27	PB.2(SPISDO)	67	NC	107	*PE.2(A2)*	147	PI.7(Strobe 15)
28	PB.3(SPISDI)	68	NC	108	*PE.1(A1)*	148	RESET KEY
29	PB.4(CK)	69	NC	109	*PE.0(A0)*	149	SYSTEM RESET
30	PB.5(CHOP)	70	NC	110	PG.0(D0)	150	VDD
31	PB.6(MDO2)	71	NC	111	PG.1(D1)	151	NC
32	PB.7(MDO1)	72	NC	112	PG.2(D2)	152	NC
33	PC.0	73	NC	113	PG.3(D3)	153	NC
34	PC.1	74	NC	114	PG.4(D4)	154	NC
35	PC.2	75	NC	115	PG.5(D5)	155	NC
36	PC.3	76	NC	116	PG.6(D6)	156	NC
37	PC.4	77	NC	117	PG.7(D7)	157	NC
38	PC.5(LATCH)	78	NC	118	NC	158	NC
39	NC	79	NC	119	NC	159	NC
40	NC	80	NC	120	NC	160	NC

(Note:1. Port D, Port E and Port F pin assignment is different from other Port.

2. Pad 4 ~ Pad 8 is for factory test only. Please do NOT bonding these pads on application circuit.)

VI. Pin Description

(1)System pin

Name	I/O type	Description	Note
VDD	-	Digital Power supply	
VSS	-	Digital Ground	
AVDD	-	Analog power supply	
AVSS	-	Analog Ground	
SYSTEM RESET	I	System reset pin	Int. pull-up
RESET_KEY	I	Low voltage related reset pin	Int. pull-up
TEST	I	Test mode select pin(High active)	Int. PULL down
PLLC	I	PLL capacitor connecting pin	Ext. C to VSS

OSCI	I	X'tal or RC oscillator connecting pin	Ext. R to VDD
OSCO	O	X'tal oscillator connecting pin	
CLKO	O	Clock output pin	
LVDIN	I	Low Voltage detection input pin	

(2) I/O Port

PORT	Bit	Function	I/O type	Description	Notes
Port A	**Bit7**	General Input	I	On key input	Int. Pull up controllable
		Interrupt and wake up	I	Input port interrupt and wake up pin	
	Bit6~0	General Input	I	Key input	Int. Pull up resistor controllable
		Interrupt and wake up	I	Input port interrupt and wake up pin	
Port B	**Bit 7**	MDO1	O	Melody positive output	-
		General I/O	I/O		Int. Pull up controllable
	Bit 6	MDO2	O	Melody negative output	-
		General I/O	I/O		Int. Pull up controllable
	Bit 5	CHOP	O	EL CHOP output pin	-
		General I/O	I/O		Int. Pull up controllable
	Bit 4	CK	O	EL CK output pin	-
		General I/O	I/O		Int. Pull up controllable
	Bit 3	SPISDI	I	SPI SDI pin	-
		General I/O	I/O		Int. Pull up controllable
	Bit 2	SPISDO	O	SPI SDO pin	-
		General I/O	I/O		Int. Pull up controllable
	Bit 1	SPISCK	I/O	SPI SCK pin	-
		General I/O	I/O		Int. Pull up controllable
	Bit 0	/SPISS	I	SPI /SS pin	-
		General I/O	I/O		Int. Pull up controllable
Port C	**Bit 7**	URXD	I	UART Rx pin	-
		General I/O	I/O		Int. Pull up controllable
	Bit 6	UTXD	O	UART Tx pin	-
		General I/O	I/O		Int. Pull up controllable
	Bit 5	LATCH	O	External memory LATCH output	-

		General I/O	I/O		Int. Pull up controllable
	Bit4~0	General I/O	I/O		Int. Pull up controllable
Port D	**Bit7~0**	General I/O	I/O	FLASH MEMORY BANK	
Port E	**Bit7~0**	General I/O	I/O	FLASH MEMORY ADDRESS(low byte)	Int. Pull up controllable
Port F	**Bit7~0**	General I/O	I/O	FLASH MEMORY ADDRESS(high byte)	Int. Pull up controllable
Port G	**Bit7~0**	General I/O	I/O	FLASH MEMORY I/O	Int. Pull up controllable
Port H	**Bit 7~0**	KEY STROBE 7~0	O	KEY STROBE	Int. Pull up controllable
		General I/O	I/O		
Port I	**Bit 7~0**	KEY STROBE 15~8	O	KEY STROBE	Int. Pull up controllable
		General I/O	I/O		

VII. Function Descriptions

(1) Kernel Part:

Feature:

1. Large ROM & RAM capacity, high expand ability.
2. Enhanced TABLE LOOK UP function.
3. High capacity external memory access ability(1 or 2 cycles).
4. Register to Register instruction:
   (i) Register to Register within one cycle.
   (ii) Port to Register or Register to Port within one cycle.
5. Compare and Branch in one instruction(2 cycles).
6. Single Repeat function.
7. Decimal ADD and SUB instruction.
8. 8*8 multiply instruction.
9. Full range CALL and JUMP ability(2 cycles)

File Register MAP:

RAM size:128 bytes+ 256 bank *128 bytes=32896 bytes

	Addr.	Unbanked		Addr.	Unbanked
SPECIAL REGISTER	00h	**INDF0**		10h	**LCDDATA**
	01h	**FSR0**		11h	**TRL2**
	02h	**PCL**		12h	**PRODL**
	03h	**PCM**		13h	**PRODH**
	04h	**PCH**		14h	**SPRL**
	05h	**BSR**		15h	**SPRM**
	06h	**STKPTR**		16h	**SPRH**
	07h	**BSR1**		17h	**UARTTX**
	08h	**INDF1**		18h	**UARTRX**
	09h	**FSR1**		19h	**PORTA**
	0Ah	**ACC**		1Ah	**PORTB**
	0Bh	**TABPTRL**		1Bh	**PORTC**
	0Ch	**TABPTRM**		1Ch	**PORTD**
	0Dh	**TABPTRH**		1Dh	**PORTE**
	0Eh	**CPUCON**		1Eh	**PORTF**
	0Fh	**STATUS**		1Fh	**PORTG**

CONTROL REGISTER	Addr.	Unbanked		Addr.	Unbanked
	20h	**PFS**		30h	**POST_ID**
	21h	**PACON**		31h	**TRL0L**
	22h	**STBCON**		32h	**TRL0H**
	23h	**DCRB**		33h	**TRL1**
	24h	**DCRC**		34h	**TRCON**
	25h	**DCRD**		35h	**MTRL**
	26h	**DCREF**		36h	**MWTCON**
	27h	**DCRG**		37h	**ELCON**
	28h	(reserved)		38h	**PAINTSTA**
	29h	**UARTCON**		39h	**PAINTEN**
	2Ah	**UARTSTA**		3Ah	**INTSTA**
	2Bh	**SPICON**		3Bh	**INTCON**
	2Ch	**SPISTA**		3Ch	**PORTH**
	2Dh	**PAWAKE**		3Dh	**PORTI**
	2Eh	**LCDARL**		3Eh	**DCRHI**
	2Fh	**LCDARH**		3Fh	**PBCON**

Addr	Unbanked
40h | 7fh	General purpose RAM

Banked Register(select by BSR)

Addr.	Bank 0	Bank 1	Bank 2	Bank 3	……	Bank 255
80h | FFh	General Purpose RAM	General Purpose RAM	General Purpose RAM	General Purpose RAM	……	General Purpose RAM

#Initial Top of STACK position (STKPTR) locates at 00.

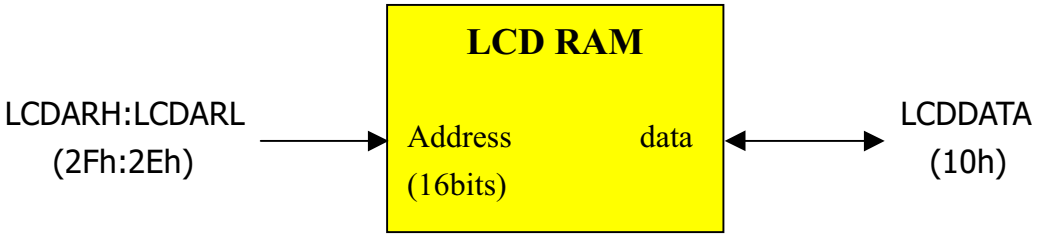
The stack level is bottom up(decrease) start from FFh of BANK 255

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

Bit 0~6 of STKPTR use to point address from 80h~FFh

Bit 7=1 use to select BANK 255 ,Bit 7=0 use to select BANK 254.

LCD RAM MAP(1088 bytes):



PROGRAM ROM Map:

8 K words* 16 Segments=128k words

Addr.	Segment		Addr.	Segment
0000h | 1FFFh	Segment 0		10000h | 11FFFh	Segment 8
2000h | 3FFFh	Segment 1		12000h | 13FFFh	Segment 9
4000h | 5FFFh	Segment 2		14000h | 15FFFh	Segment 10
6000h | 7FFFh	Segment 3		16000h | 17FFFh	Segment 11
8000h | 9FFFh	Segment 4		18000h | 19FFFh	Segment 12
A000h | BFFFh	Segment 5		1A000h | 1BFFFh	Segment 13
C000h | DFFFh	Segment 6		1C000h | 1DFFFh	Segment 14
E000h | FFFFh	Segment 7		1E000h | 1FFDFh	Segment 15
			1FFE0h-1FFFBh	TEST Program(28 Words)
			1FFFCh-1FFFFh	Code Option

Register Description

Remark:

R = Read only bit **W** = Write only bit **-** = Not implement

Addr.	Register name	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	Bit 0
0	**INDF0**	R/W							
		Indirect addressing pointer0.							
1	**FSR0**	R/W							
		File select register 0 for INDF0(R0)							
2	**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	**PCH**	-	-	-	-	-	-	-	R/W
		-	-	-	-	-	-	-	PC16
5	**BSR**	Bit7 ~ Bit 0: R/W							
		Bank select register(for INDF0 & general)							
6	**STKPTR**	R/W							
		Stack pointer							
7	**BSR1**	R/W							
		Bank select register(for INDF1)							
8	**INDF1**	R/W							
		Indirect addressing pointer1.							
9	**FSR1**	Bit 7: R (Fix at 1) , Bit6 ~ Bit 0: R/W							
		File select register 1 for INDF1(R8)							
A	**ACC**	R/W							
		Accumulator							
B	**TABPTRL**	R/W							
		Table pointer low							
C	**TABPTRM**	R/W							
		Table pointer middle							
D	**TABPTRH**	R/W							
		Table pointer high							
E	**CPUCON**	R/W	R/W	R/W	R	R/W	R	R/W	R/W
		CLKOEN	CKS	LVDEN	/LV	/GLINTD	PS	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	**LCDDATA**	R/W							
		Indirect register to LCD RAM							
11	**TRL2**	R/W							
		Timer2 reload register							
12	**PRODL**	R/W							
		Multiplier product low							
13	**PRODH**	R/W							
		Multiplier product high							
14	**SPRL**	R/W							
		Shift register low byte of SPI							
15	**SPRM**	R/W							

		Shift register middle byte of SPI						
16	**SPRH**	R/W						
		Shift register high byte of SPI						
17	**UARTTX**	W						
		UART transfer register						
18	**UARTRX**	R						
		UART receiver register						
19	**PORTA**	R	R	R	R	R	R	R	R
		A.7	A.6	A.5	A.4	A.3	A.2	A.1	A.0
1A	**PORTB**	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0
1B	**PORTC**	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		C.7	C.6	C.5	C.4	C.3	C.2	C.1	C.0
1C	**PORTD**	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		D.7	D.6	D.5	D.4	D.3	D.2	D.1	D.0
1D	**PORTE**	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		E.7	E.6	E.5	E.4	E.3	E.2	E.1	E.0
1E	**PORTF**	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		F.7	F.6	F.5	F.4	F.3	F.2	F.1	F.0
1F	**PORTG**	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
20	**PFS**	R/W							
		PLL frequency select							
21	**PACON**	R/W	R/W	R/W	R/W	R/W	R/W	-	-
		LVD2	LVD1	LVD0	LVDSEL	Bit7PU	/REN	-	-
22	**STBCON**	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
		UINVEN	-	BitST	ALL	STB3	STB2	STB1	STB0
23	**DCRB**	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		Bit7DC	Bit6DC	Bit5DC	Bit4DC	Bit3DC	Bit2DC	Bit1DC	Bit0DC
24	**DCRC**	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		Bit7DC	Bit6DC	Bit5DC	Bit4DC	Bit3DC	Bit2DC	Bit1DC	Bit0DC
25	**DCRD**	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		Bit7DC	Bit6DC	Bit5DC	Bit4DC	Bit3DC	Bit2DC	Bit1DC	Bit0DC
26	**DCREF**	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		FHNPU	FLNPU	FHNDC	FLNDC	EHNPU	ELNPU	EHNDC	ELNDC
27	**DCRG**	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		-	LAHEN	CHNPU	CLNPU	GHNPU	GLNPU	GHNDC	GLNDC
28	(Reserved)	-							
		-							
29	**UARTCON**	W	R/W	R/W	R/W	R/W	R/W	R	R/W
		TB8	UMODE1	UMODE0	BRATE2	BRATE1	BRATE0	UTBE	TXE
2A	**UARTSTA**	R	R/W	R/W	R	R	R	R	R/W
		RB8	EVEN	PRE	PRERR	OVERR	FMERR	URBF	RXE
2B	**SPICON**	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
		TLS1	TLS0	BRS2	BRS1	BRS0	EDS	DORD	SE
2C	**SPISTA**	-	-	-	-	R/W	R/W	R	R
		-	-	-	-	SPWKEN	SMP	DCOL	RBF
2D	**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
2E	**LCDARL**	R/W							

		Address low to LCD RAM							
2F	**LCDARH**	Bit 7~3: Not Implement, Bit 2 ~ Bit 0: R/W Address high to LCD RAM							
30	**POST_ID**	R/W EM_ID	R/W LCD_ID	R/W FSR1_ID	R/W FSR0_ID	R/W EMPE	R/W LCDPE	R/W FSR1PE	R/W FSR0PE
31	**TRL0L**	R/W Timer0 reload low byte							
32	**TRL0H**	R/W Timer0 reload high byte							
33	**TRL1**	R/W Timer1 reload register							
34	**TRCON**	R/W T1WKEN	R/W T1EN	R/W T1PSR1	R/W T1PSR0	R/W T0EN	R/W T0CS	R/W T0PSR1	R/W T0PSR0
35	**MTRL**	R/W Melody Timer reload register							
36	**MWTCON**	R/W T2EN	R/W T2PSR1	R/W T2PSR0	R/W WDTEN	R/W WDTPSR1	R/W WDTPSR0	R/W MTEN	R/W MDO2EN
37	**ELCON**	R/W ELTEN	R/W CKP	R/W ELTRL5 ~ ELTRL0					
38	**PAINTSTA**	R/W PA7I	R/W PA6I	R/W PA5I	R/W PA4I	R/W PA3I	R/W PA2I	R/W PA1I	R/W PA0I
39	**PAINTEN**	R/W PA7IE	R/W PA6IE	R/W PA5IE	R/W PA4IE	R/W PA3IE	R/W PA2IE	R/W PA1IE	R/W PA0IE
3A	**INTSTA**	R/W URXI	R/W UTXI	R/W UERRI	R/W SRBFI	R/W LVDI	R/W TMR2I	R/W TMR1I	R/W TMR0I
3B	**INTCON**	R/W URXIE	R/W UTXIE	R/W UERRIE	R/W SRBFIE	R/W LVDIE	R/W TMR2IE	R/W TMR1IE	R/W TMR0IE
3C	**PORTH**	R/W H.7	R/W H.6	R/W H.5	R/W H.4	R/W H.3	R/W H.2	R/W H.1	R/W H.0
3D	**PORTI**	R/W I.7	R/W I.6	R/W I.5	R/W I.4	R/W I.3	R/W I.2	R/W I.1	R/W I.0
3E	**DCRHI**	R/W IHNPU	R/W ILNPU	R/W IHNDC	R/W ILNDC	R/W HHNPU	R/W HLNPU	R/W HHNDC	R/W HLNDC
3F	**PBCON**	R/W Bit7PU	R/W Bit6PU	R/W Bit5PU	R/W Bit4PU	R/W Bit3PU	R/W Bit2PU	R/W Bit1PU	R/W Bit0PU

R0(INDF0)

*R0 is not a physically implemented register. It is useful as indirect addressing pointer0. Any instruction using R0 as register actually accesses data pointed by the File select register0(R1) and bank select register(R5).

R1(FSR0)

*R1 is an address register for indirect addressing pointer0. User can select up to 256 bytes register on working bank(selected by BSR).

*When enable FSR0 auto increase/decrease function by R30(Post_ID), FSR0 will ***NOT*** carry into or borrow from BSR register. On the contrary, FSR1 will carry into or borrow from BSR1.

R2~R4(PCL,PCM,PCH)

PC16	PC15 ~ PC8	PC7 ~ PC0
PCH	PCM	PCL

*Generates up to 128K*16 on chip ROM addresses to the relative programming instruction codes.

*"S0CALL" loads the low 12 bits of the PC(4K*16 ROM).

*"SCALL","SJUMP" loads the low 13 bits of the PC(8K*16 ROM).

*"LCALL","LJUMP" loads the full 17 bits of the PC(128K*16 ROM).

*"ADD R2,A" allows a relative address be added to the current PC. The carry bit of R2 will automatically carry into PCM , PCH.

R5(BSR)

*Determine which bank is active(working bank) among the 256 banks.

*INDF0 is pointed on the working bank determined by R5.

R6(STKPTR)

*The stack level is bottom up(decrease) start from FFh of BANK 255.

*Each INT/CALL will stack two bytes address, total have capacity of 128 levels.

*Stack locates on BANK 255 and 254 from address FFh~80h. And initial stack pointer is 00h.

*Bit 0~6 of STKPTR use to pointer address from 80h~FFh

*Bit 7=1 used to select BANK 255 ,Bit 7=0 used to select BANK 254.

R7(BSR1)

*R7 is a bank register for indirect addressing pointer 1(R8,R9) only. Can't determine the working bank for general register.

R8(INDF1)

*R8 is not a physically implemented register. It is useful as indirect addressing pointer 1. Any instruction using R8 as register actually accesses data pointed by the File select Register1(R9) and bank select register1(R7).

R9(FSR1)

*R9 is an address register for INDF1 only. User can select address 0x80 to 0xFF register on the bank pointed by R7.

*Bit 7 of FSR1 is fixed to 1.

*Linear addressing ability of INDF1 shown below.

File Register

	BANK 4	BANK 5
80h		BB
81h		CC
82h		DD
:		:
FDh	88	
FEh	99	
FFh	AA	

Auto Increase on FSR1
(Set FSR1PE=1,FSR1_ID=1)

Instruction	BSR1	FSR1	INDF1	ACC
MOV A,INDF1	04	FF	AA	00
↓	**05**	**80**	BB	AA
MOV A,INDF1	05	81	CC	BB
↓	05	82	DD	CC
MOV A,INDF1				
:				

(FSR1 will carry into BSR1)*
*(*Bit 7 of FSR1is fixed to 1)*

Auto Decrease on FSR1
(Set FSR1PE=1,FSR1_ID=0)

Instruction	BSR1	FSR1	INDF1	ACC
MOV A,INDF1	05	80	BB	00
↓	**04**	**FF**	AA	BB
MOV A,INDF1	04	FE	99	AA
↓	04	FD	88	99
MOV A,INDF1				
:				

(FSR1 will borrow from BSR1)*
*(*Bit 7 of FSR1is fixed to 1)*

RA(ACC)

*Accumulator. Internal data transfer , or instruction operand holding.

RB~RD(TABPTRL,TABPTRM,TABPTRH)

bit23	bit16	bit8	bit0
TABPTRH	TABPTRM	TABPTRL	

*Internal program ROM or external memory address register.

*Bit 23 used to select internal/external memory, **Bit 22~Bit 1** used to point the address of memory, **Bit 0**

used to select the low or high byte of pointed word. (See TBRD instruction).

*External memory table look up function is reserved for the future model.

RE(CPUCON)

bit 7 bit0

CLKOEN	CKS	LVDEN	/LV	/GLINTD	PS	MS1	MS0

*bit7(CLKOEN): clock output enable control bit.

CLKO pin is tri-state when clock output disable.**(CLKOEN=0)**

*bit6(CKS): clock select bit of clock output pin (CLKO).

0: CLKO pin output frequency at 32.768KHz(Fosc)

1: CLKO pin output frequency at 21.845KHz(Fosc/1.5).

*bit5(LVDEN): Enable Low Voltage Detector.

*bit4(/LV): Low voltage detected. This is a read only bits. When the voltage of **LVDIN** pin is lower than Vdet(selected by **LVD0~LVD2** bit of R21), this bit will be clear.

0: Low voltage is detected.

1: Low voltage is not detected or LVD is disable.

*bit3(/GLINTD): Global interrupt disable bit.

0:Disable all interrupt.

1:Enable all un-mask interrupt.

*bit2(PS): PLL stable flag. This bit will be cleared when PLL not turned on or PLL is turned on but not stable. And be set after PLL turned on and frequency stable.

0: PLL is not turned on or PLL frequency not yet stable .

1: PLL is turned on and frequency stable.

*bit1(MS1): Mode select bit 1. Used to enter SLEEP MODE or IDLE MODE after executing "**SLEP**" instruction.

0: SLEEP MODE

1: IDLE MODE

*bit0(MS0): Mode select bit 0. Used to enter SLOW MODE or FAST MODE.

0: SLOW MODE.

1: FAST MODE.

RF(STATUS)

bit 7 bit0

/TO	/PD	SGE	SLE	OV	Z	DC	C

*bit7(/TO): Reset to 0 when WDT time out reset. Set to 1 by "WDTC" instruction , enter SLEEP MODE , power on reset or Reset pin low condition.

*bit6(/PD) : Reset to 0 when enter sleep mode. Set to 1 by "WDTC" instruction , power on reset or Reset pin low condition.

*bit5(SGE): Computation result great than or equal to zero(Positive value) after signed arithmetic. Only affected by HEX arithmetic instruction.

*bit4(SLE): Computation result less than or equal to zero(Negative value) after signed arithmetic. Only affected by HEX arithmetic instruction.

*bit3(OV): Overflow flag. Use in signed operation when bit6 carry into or borrow from signed bit(bit7).

*Note: 1.When OV=1 after signed arithmetic , user can check **SGE** bit and **SLE** bit to know either overflow(carry into sign bit) or underflow (borrow from sign bit) happened.*

OV=1 and SGE=1 ➔ overflow happened

OV=1 and SLE=1 ➔ underflow happened

*2. When overflow happened, user should **clear** the MSB of Accumulator to get the correct value.*

*When underflow happened, user should **set** the MSB of accumulator to get the correct value.*

Ex1. ADD positive value with positive value, and ACC signed bit was affected

```
MOV     ACC,#60h    ;signed number +60h
ADD     ACC,#70h    ;+60h ADD with +70h
=====================================
```

after instruction:

ACC=d0h

SGE=1, means the result is great than or equal to 0(positive value)

OV=1, means result carry into signed bit(bit 7), overflow happened.

correct the signed bit:

ACC=50h (Clear the signed bit)

The actual result= +80h* 1(OV=1) + 50h = +d0h

Ex2. SUB positive value from negative value, and ACC signed bit was 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 result borrow from signed bit(bit 7), underflow happened.

correct the signed bit:

ACC=c0h (set the signed bit)

The actual result= -80h* 1(OV=1) - 40h(signed number of c0h) = -c0h

*bit2(Z) : Zero flag.

*bit1(DC) : Auxiliary carry flag.

*bit0(C) : Carry flag.

R10(LCDDATA)

*R10 is an indirect addressing pointer of LCD RAM.

Any instruction using R10 as register actually accesses LCD RAM pointed by LCDARH:LCDARL(R2F:R2E).

R11(TRL2)

*Timer2 reload register.

R12,R13(PRODL,PRODH)

*An unsigned 8*8 hardware multiplier is included in the micro controller. The result is stored into the 16 bits product register (PRODH:PRODL).

R14~R16(SPRL:SPRM:SPRH)

*SPI shift buffer for 24/16/8 bits length.

R17(UARTTx)

bit 7							bit0
TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0

*TB7~TB0 : UART Transmission data register.

R18(UARTRx)

bit 7							bit0
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0

*RB7~RB0 : UART receiving data register

R19~R1F(PORTA~PORTG)

*Port A ~ Port G are general I/O register.

R20(PFS)

*PLL frequency select.

R21(PACON)

bit 7							bit0
LVD2	LVD1	LVD0	LVDSEL	Bit7PU	/REN	-	-

*bit7~5(LVD2~0): LVD detection voltage select bits.

LVD2~0	Vdet(Vth-)	Vdet(Vth+)	Deviation	LVD2~0	Vdet(Vth-)	Vdet(Vth+)	Deviation
000	2.2V	2.3V	±0.1V	100	2.6V	2.7V	±0.1V
001	2.3V	2.4V		101	2.7V	2.8V	
010	2.4V	2.5V		110	2.8V	2.9V	
011	2.5V	2.6V		111	2.9V	3.0V	

The initial value of LVD2~0 after CPU reset is "000".

*bit4(LVDSEL): Built in LVD/External LVD select

0: External LVD selected.

1: Built-in LVD selected.

*bit3(Bit7PU): Enable PortA.7 pull up resistor.

*bit2(/REN): Enable PortA.0~PortA.6 pull up resistor.

0: Enable pull up resistor.

1: Disable pull up resistor.

*bit1~0: Reserved.

R22(STBCON)

bit 7							bit0
UINVEN	-	BitST	ALL	STB3	STB2	STB1	STB0

*bit7(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.

*bit6:Reserved.

*bit5(BitST) : Enable Port H and Port I as key strobe pin.

0 : Port H and Port I are general I/O port.

1 : Port H and Port I are key strobe pin. Strobe signal as **STB3~0** defined.

*bit4(ALL): Set All strobe.

0 : Bit strobe.

1 : All strobe.

*bit3~0(STB3~0): 16 to 1 multiplexing selector of key strobe pin(Port H ~ Port I)

R23(DCRB)

*bit7~0(Bit7DC~Bit0DC):Direction control of Port B .

0:output pin

1:input pin

R24(DCRC)

*bit7~0(Bit7DC~Bit0DC):Direction control of Port C.

0:output pin

1:input pin

R25(DCRD)

*bit7~0(Bit7DC~Bit0DC):Direction control of Port D.

0:output pin

1:input pin

R26(DCREF)

bit 7 bit0

FHNPU	FLNPU	FHNDC	FLNDC	EHNPU	ELNPU	EHNDC	ELNDC

*bit7(FHNPU):Enable Port F high nibble pull up resistor .

*bit6(FLNPU):Enable Port F Low nibble pull up resistor.

*bit5(FHNDC):Port F high nibble direction control.

0:output pin

1:input pin

*bit4(FLNDC):Port F low nibble direction control.

0:output pin

1:input pin

*bit3(EHNPU):Enable Port E high nibble pull up resistor .

*bit2(ELNPU): Enable Port E low nibble pull up resistor .

*bit1(EHNDC):Port E High nibble direction control.

0:output pin

1:input pin

*bit0(ELNDC):Port E Low nibble direction control.

0:output pin

1:input pin

R27(DCRG)

bit 7 bit0

-	LAHEN	CHNPU	CLNPU	GHNPU	GLNPU	GHNDC	GLNDC

*bit7:Reserved.

*bit6(LAHEN):Enable PortC.5 as external memory data output LATCH signal.

LATCH signal output is only available when write to PORTG register.

*bit5(CHNPU):Enable Port C high nibble pull up resistor .

*bit4(CLNPU):Enable Port C Low nibble pull up resistor.

*bit3(GHNPU):Enable Port G high nibble pull up resistor .

*bit2(GLNPU): Enable Port G low nibble pull up resistor .

*bit1(GHNDC):Port G High nibble direction control.

0:output pin

1:input pin

*bit0(GLNDC):Port G Low nibble direction control.

0:output pin

1:input pin

R28

*Reserved.

R29(UARTCON)

bit 7 bit0

TB8	UMODE1	UMODE0	BRATE2	BRATE1	BRATE0	UTBE	TXE

*bit7(TB8) : Transmission data bit8

*bit6~5(UMODE1~0) : UART mode

00 : Mode 1: 7-bit data

01 : Mode 2: 8-bit data

10 : Mode 3: 9-bit data

11: reserved

*bit4~2(BRATE2~0) : Baud rate selector

000 : connect to Timer0/32 signal

001 : 600 baud

010 : 1200 baud

011 : 2400 baud

100 : 4800 baud

101 : 9600 baud

110 : 19200 baud

111 : 38400 baud

*bit1(UTBE) : UART transfer buffer empty. Set to 1 when transfer buffer empty. Reset to 0 automatically when write into UARTTx register.
(**UTBE** bit is read-only)

*bit0(TXE) : Enable transmission.

R2A(UARTSTA)

bit 7 bit0

RB8	EVEN	PRE	PRERR	OVERR	FMERR	URBF	RXE

*bit7(RB8) :Receiving data bit8

*bit6(EVEN) : Select parity check
0 : Odd parity
1 : Even parity

*bit5(PRE) : Enable parity addition
0 : Disable
1 : Enable

*bit4(PRERR) : Parity error flag. Set to 1 when parity error happened, and clear to 0 by software.

*bit3(OVERR) : Over running error flag. Set to 1 when overrun error happened, and clear to 0 by software.

*bit2(FMERR) : Framing error flag. Set to 1 when framing error happened, and clear to 0 by software.

*bit1(URBF) : UART read buffer full flag. Set to 1 when one character is received . Reset to 0 automatically when read from UARTRx register.
(**URBF** bit is read-only)

*bit0(RXE) : Enable receiving function

R2B(SPICON)

bit 7 bit0

TLS1	TLS0	BRS2	BRS1	BRS0	EDS	DORD	SE

*Bit7~6(TLS1~TLS0):Shift buffer length select. Shift buffer length is programmable.
00: SPI disable
01: SPI shift buffer length = 24 bits
10: SPI shift buffer length = 16 bits
11: SPI shift buffer length = 8 bits

*bit5~3(BRS2~BRS0):Bit rate select. Programming 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

*bit2(EDS): Select the raising / falling edges latch by programming the EDS bit
0: Falling edge
1: Raising edge

*bit1(DORD):Data transmission order.
0:Shift left (MSB first)
1:Shift right (LSB first)

*bit0(SE): Shift Enable.
Set to 1 automatically when write data into SPRL register and begin to shift.
Reset to 0 when transfer buffer empty detected.
(**SE** bit is read-only)

R2C(SPISTA)

bit 7							bit0
-	-	-	-	SPWKEN	SMP	DCOL	RBF

*bit7~4:Reserved.

*bit3(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.

*bit2(SMP):SPI data input sample phase.
0:Input data sampled at middle of data output time
1:Input data sampled at the end of data output time
In slave mode, data input sample is fixed at middle of data output time.

*bit1(DCOL):SPI data collision.
0:Data collision didn't occurs
1:Data collision occurs. Should be cleared by software.

*bit0(RBF):Set to 1 by Buffer Full Detector, and clear to 0 automatically when read data from SPRL register.
(**RBF** bit is read-only)

R2D(PAWAKE)

bit 7 bit0

WKEN7	WKEN6	WKEN5	WKEN4	WKEN3	WKEN2	WKEN1	WKEN0

*bit7~0(WKEN7~WKEN0): Wake up enable control bit of Port A.

0: Disable Port A wake up function.

1: Enable Port A wake up function.

R2E~R2F(LCDARL,LCDARH)

* LCDARH:LCDARL is the address for LCD RAM.

*LCDARH: Page address for LCD RAM(valid from 00H to 07H).

*LCDARL: Column address for LCD RAM(valid from 00H to 87H).

LCD RAM MAP:

PAGE 00 (LCDARH=00H)

RAM address		COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7
	LCDARL	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
SEG0	00H								
SEG1	01H								
SEG2	02H								
:	:								
:	:								
SEG134	86H								
SEG135	87H								

PAGE 01 (LCDARH=01H)

RAM address		COM8	COM9	COM10	COM11	COM12	COM13	COM14	COM15
	LCDARL	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
SEG0	00H								
SEG1	01H								
SEG2	02H								
:	:								
:	:								
SEG134	86H								
SEG135	87H								

PAGE 02 (LCDARH=02H)

RAM address		COM16	COM17	COM18	COM19	COM20	COM21	COM22	COM23
	LCDARL	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
SEG0	00H								
SEG1	01H								
SEG2	02H								
:	:								
:	:								
SEG134	86H								

SEG135	87H								

……..

……..

PAGE 07 (LCDARH=07H)

RAM address		COM56	COM57	COM58	COM59	COM60	COM61	COM62	COM63
	LCDARL	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
SEG0	00H								
SEG1	01H								
SEG2	02H								
:	:								
:	:								
SEG134	86H								
SEG135	87H								

R30(POST_ID)

bit 7 bit0

EM_ID	LCD_ID	FSR1_ID	FSR0_ID	EMPE	LCDPE	FSR1PE	FSR0PE

*Post increase / decrease control register.

*For example , enable LCD post decrease function(bit2=1[enable],bit6=0[decrease]),
then LCDARL will automatic decrease after access(read or write) LCDDATA(R10).
(Note: LCDARL will NOT carry into /borrow from LCDARH.)

*bit 0(FSR0PE): Enable FSR0 post increase/decrease function.

*bit 1(FSR1PE): Enable FSR1 post increase/decrease function.

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

*bit 3(EMPE): Enable External memory address (PORTF:PORTE) post increase/decrease function.
when **read data from PORT G.**
(Only **read data from PORTG register** can active this function)

*bit 4(FSR0_ID): Set to 1 means auto_increase, reset to 0 means auto_decrease of FSR0.

*bit 5(FSR1_ID): Set to 1 means auto_increase, reset to 0 means auto_decrease of FSR1.

*bit 6(LCD_ID): Set to 1 means auto_increase, reset to 0 means auto_decrease of LCDARL.

*bit7(EM_ID): Set to 1 means PORTF:PORTE register auto increase, reset to 0 means auto decrease .

*The initial value of bit0~bit3 is 0, and bit 4~bit 7 is 1.

R31~R32(TRL0L~TRL0H)

*Timer0 reload low byte and high byte.

R33(TRL1)

*Timer1 reload register.

R34(TRCON)

bit 7 bit0

T1WKEN	T1EN	T1PSR1	T1PSR0	T0EN	T0CS	T0PSR1	T0PSR0

*Timer0 and timer1 control register.

*bit7(T1WKEN): Enable bit of Timer1 underflow wake up function in IDLE MODE.

0: Disable Timer1 wake up function .

1: Enable Timer1 wake up function.

*bit6(T1EN): Timer1 enable control bit

0: Disable Timer1(stop counting) .

1: Enable Timer1 .

*bit5~4(T1PSR1~T1PSR0): Timer1 Prescaler select bits.

T1PSR1:T1PSR0	Prescale value
00	1:4
01	1:16
10	1:64
11	1:256

*bit3(T0EN): Timer0 enable control bit

0: Disable Timer0(stop counting) .

1: Enable Timer0 .

*bit2(T0CS): Timer 0 clock source select bit.

0: Clock source is from Fosc.

1: Clock source is from half of the system clock.

*bit1~0(T0PSR1~T0PSR0): Timer0 Prescaler select bits.

T0PSR1:T0PSR0	Prescale value
00	1:1
01	1:4
10	1:16
11	1:64

R35(MTRL)

*Melody Timer reload register.

R36(MWTCON)

bit 7 bit0

T2EN	T2PSR1	T2PSR0	WDTEN	WDTPSR1	WDTPSR0	MTEN	MDO2EN

*Timer2 ,Melody timer and WDT control register.

*bit7(T2EN): Timer2 enable control bit.

0: Disable Timer2(stop counting) .

1: Enable Timer2 .

*bit6~5(T2PSR1~T2PSR0): Timer2 Prescaler select bits.

T2PSR1:T2PSR0	Prescale value
00	1:1
01	1:2
10	1:4
11	1:8

*bit4(WDTEN): Watch Dog Timer enable bit.

*bit3~2(WDTPSR1~WDTPSR0): Watch Dog timer Prescaler select bits.

WDTPSR1:WDTPSR0	Prescale value
00	1:4
01	1:16
10	1:64
11	1:128

*bit1(MTEN): Melody Timer enable control bit.

*bit0(MDO2EN): Melody negative output port enable control bit.

R37(ELCON)

bit 7 bit0

ELTEN	CKP	ELTRL5 ~ ELTRL0

*bit7(ELTEN): EL Timer enable control bit.

0: Disable EL Timer(stop counting) and recover CK and CHOP pin to general I/O pin.

1: Enable EL Timer and change Port B.4 and Port B.5 to CK and CHOP output pin.

*bit6(CKP): EL clock polarity select bit.

0: Idle state for CHOP pin is low level.

1: Idle state for CHOP pin is high level.

*bit5~0(ELTRL5~0): Used to store the auto reload value of EL timer. When underflow happens, ELTRL will automatic reload into 6 bits counter .

R38(PAINTSTA)

bit 7 bit0

PA7I	PA6I	PA5I	PA4I	PA3I	PA2I	PA1I	PA0I

*bit7~0(PA7I~PA0I): INT status of Port A interrupt. Set to 1 when pin **falling edge** detected. Clear to 0 by software.

R39(PAINTEN)

bit 7 bit0

PA7IE	PA6IE	PA5IE	PA4IE	PA3IE	PA2IE	PA1IE	PA0IE

*Port A interrupt enable control register

*bit7~0(PA7IE~PA0IE): Control bit of interrupt.

0: Disable Port A interrupt function.

1: Enable Port A interrupt function.

R3A(INTSTA)

bit 7 bit0

URXI	UTXI	UERRI	SRBFI	LVDI	TMR2I	TMR1I	TMR0I

*bit0(TMR0I): Set to 1 when TMR0 underflow happened. Clear to 0 by software or Timer0 disable.

*bit1(TMR1I): Set to 1 when TMR1 underflow happened. Clear to 0 by software or Timer1 disable.

*bit2(TMR2I): Set to 1 when TMR2 underflow happened. Clear to 0 by software or Timer2 disable.

*bit3(LVDI): Set to 1 when LVD low level detected(**Falling edge**). Clear to 0 by software or LVD disable.

*bit4(SRBFI): Set to 1 when SPI read buffer full happened. Clear to 0 by software or SPI disable.

*bit5(UERRI): Set to 1 when UART receiving error happened. Clear to 0 by software or UART disable.

*bit6(UTXI): Set to 1 when UART transfer buffer empty happened. Clear to 0 by software or UART TX disable(TXE=0)

*bit7(URXI):Set to 1 when UART receiver buffer full happened. Clear to 0 by software or UART RX disable(RXE=0)

R3B(INTCON)

bit 7 bit0

URXIE	UTXIE	UERRIE	SRBFIE	LVDIE	TMR2IE	TMR1IE	TMR0IE

* 0: Disable interrupt function.

1: Enable interrupt function.

*bit0(TMR0IE): Control bit of TIMER0 interrupt.

*bit1(TMR1IE): Control bit of TIMER1 interrupt.

*bit2(TMR2IE): Control bit of TIMER2 interrupt.

*bit3(LVDIE): Control bit of LVD interrupt.

*bit4(SRBFIE): Control bit of SPI read buffer full interrupt.

*bit5(UERRIE): Control bit of UART receiving error interrupt .

*bit6(UTXIE): Control bit of UART Transfer buffer empty interrupt.

*bit7(URXIE): Control bit of UART Receiver buffer full interrupt.

R3C~R3D(PORTH~PORTI)

*Port H and PORT I general I/O register.

R3E(DCRHI)

bit 7 bit0

IHNPU	ILNPU	IHNDC	ILNDC	HHNPU	HLNPU	HHNDC	HLNDC

*bit7(IHNPU):Enable Port I high nibble pull up resistor .

*bit6(ILNPU):Enable Port I Low nibble pull up resistor.

*bit5(IHNDC):Port I high nibble direction control.

0:output pin

1:input pin

*bit4(ILNDC):Port I low nibble direction control.

0:output pin

1:input pin

*bit3(HHNPU):Enable Port H high nibble pull up resistor .

*bit2(HLNPU): Enable Port H low nibble pull up resistor .

*bit1(HHNDC):Port H High nibble direction control.

0:output pin

1:input pin

*bit0(HLNDC):Port H Low nibble direction control.

0:output pin

1:input pin

R3F(PBCON)

*bit7~0(Bit7PU~Bit0PU):Pull up resistor control of Port B.

0:Disable pull up resistor

1:Enable pull up resistor

Code Option

(Locate on the final 4 words of Program ROM)

WORD0 (0x1FFFC):

bit 7							bit0
-	-	-	TBRD_3T	-	ELSEL	IM	OSCSEL

*bit0 (OSCSEL): RC oscillator or Crystal select bit

0:RC oscillator

1:Crystal oscillator

*bit1 (IM): Initial mode after Reset.

0: SLOW Mode

1: FAST Mode

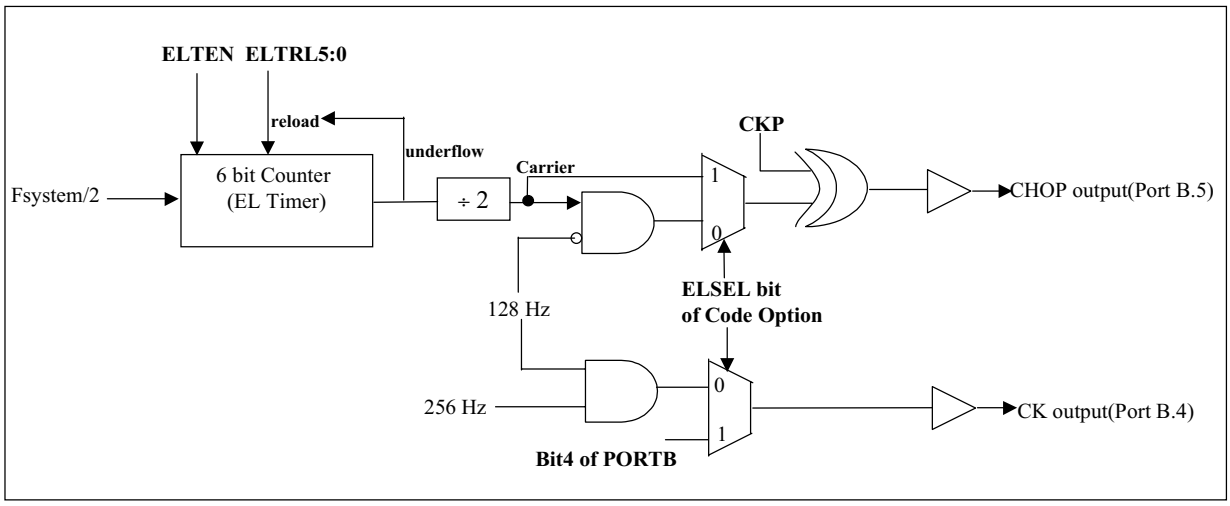
*bit2(ELSEL): EL output timing select bit

0: CHOP output is from carrier gating with 128Hz and CKP .

CK output is from 128Hz gating with 256Hz.

1: CHOP output is directly from carrier .

CK output is directly from Bit4 of PORTB register.



*bit3: Reserved.

*bit4(TBRD_3T): TBRD instruction three cycle enable bit.

0: "TBRD" is 2 cycles instruction.

1: "TBRD" is 3 cycles instruction. (for use at large DATA ROM version)

*bit5~15: Not use.

WORD1 (0x1FFFD):

bit 7							bit0
-	-	CSOP	PLLST_OPT	HFSEL	OPT1	OPT0	-

*bit0:Reserved.

*bit1(OPT0):Reserved.

*bit2(OPT1):Reserved.

*bit3(HFSEL): High frequency select bit. ***This function is only available in ROMless CPU.***

0: High frequency system clock is from PLL clock.

1: High frequency system clock is from HOSCO pin.

*bit4(PLLST_OPT):PLL stable accuracy select bit.

0: PS bit will be set when PLL frequency is within target +/- 10%.

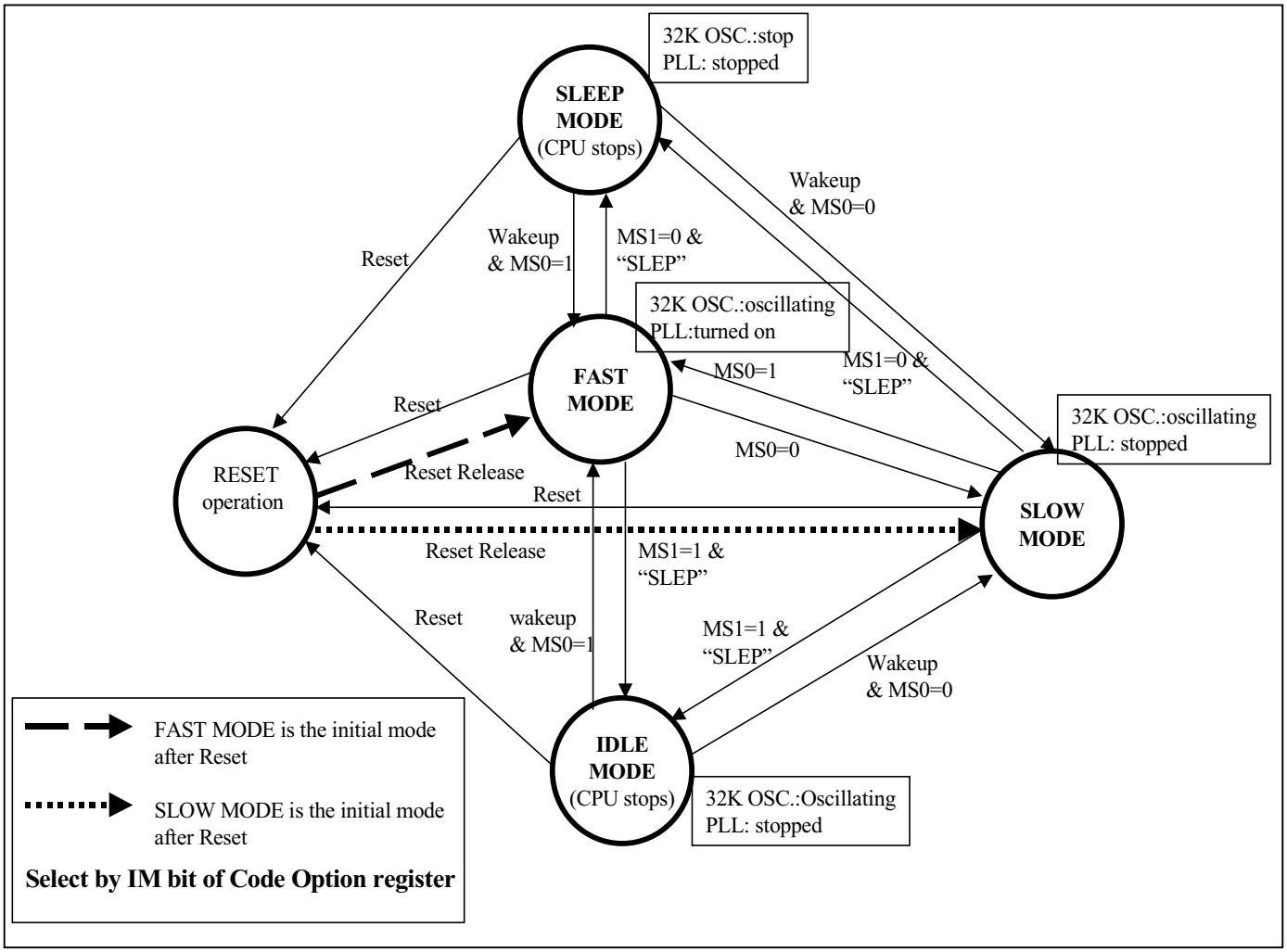
1: PS bit will be set when PLL frequency is within target +/- 3%.

*bit5(CSOP):Reserved.

*Bit6~15: Not use.

WORD2~WORD3 (0x1FFFE~0x1FFFF): Not use.

(2) CPU OPERATION MODE:



Mode / Device	SLEEP	IDLE	SLOW	FAST
Osc.(32768Hz)	X	O	O	O
PLL	X	X	X	O
CPU	X	X	O	O
LVD	O	O	O	O
CLOCK OUTPUT	X	O	O	O
Timer0,1,2, Melody Timer, EL Timer	X	O	O	O
INT	X(*2)	X(*2)	O	O
SPI	O(slave)	O(slave)	O	O
UART	X	X	X	O

I/O wake up	O	O	X	X
SPI wake up	O	O	X	X
Timer1 wake up	X	O	X	X

(Note): *1: O: Function available if enable.

X: Function NOT available.

*2: Interrupt flag will be recorded but not be executed until CPU wake up.

Function Description:

1. SLEEP MODE: When set **MS1** bit of CPUCON register to '0' and execute "**SLEP"** instruction, the CPU will enter SLEEP MODE.

The SLEEP MODE suspends all system operation and holds the internal status immediately before the suspension with low power consumption. This mode will be released by CPU reset , I/O pins or SPI receive full wake up.

The **/PD** bit of STATUS Register(RFh) is cleared when entering SLEEP MODE. This bit will be set to 1 by "WDTC" instruction , Power on RESET or Reset pin low condition.

All register and LCD RAM unchanged in SLEEP MODE.

2.IDLE MODE: When set **MS1** bit of CPUCON register to '1' and execute "**SLEP"** instruction, the CPU will enter IDLE MODE.

The IDLE MODE suspends all SLOW operations except for the Oscillator , clock output . It retains the internal status with low power consumption without stopping the clock function.

The IDLE MODE will be waken up and return to the either SLOW MODE(**MS0**=0) or FAST MODE(**MS0**=1) by the timer 1 wake up ,SPI receive full or I/O pins wake up (if enable) .

All register and LCD RAM unchanged in IDLE MODE.

3.SLOW MODE: When set **MS0** bit of CPUCO`N register to '0' , the CPU will enter SLOW MODE. Expect PLL and UART, every device can be turned on. System clock is at 32.768KHz. This feature allows all the internal operations to slow down and thus reduces power consumption.

4.FAST MODE: When set **MS0** bit of CPUCON register to '1' , the CPU will enter FAST MODE. After turning on PLL and wait 32 clocks from Oscillator, then system clock switched to

high frequency of PLL(adjustable by PFS register).
This mode allows all the internal operations at fast speed. But it will consumes the most power .

Register Description:

- **CPUCON Register:**

bit 7 bit0

CLKOEN	CKS	LVDEN	/LV	/GLINTD	PS	**MS1**	**MS0**

- **MS0**: Mode select bit 0. Used to enter SLOW MODE or FAST MODE.
  0: SLOW MODE.
  1: FAST MODE.
- **MS1:** Mode select bit 1. Used to enter SLEEP MODE or IDLE MODE after executing "**SLEP"** instruction.
  0: SLEEP MODE
  1: IDLE MODE

(3) Reset and WAKE UP:

RESET:

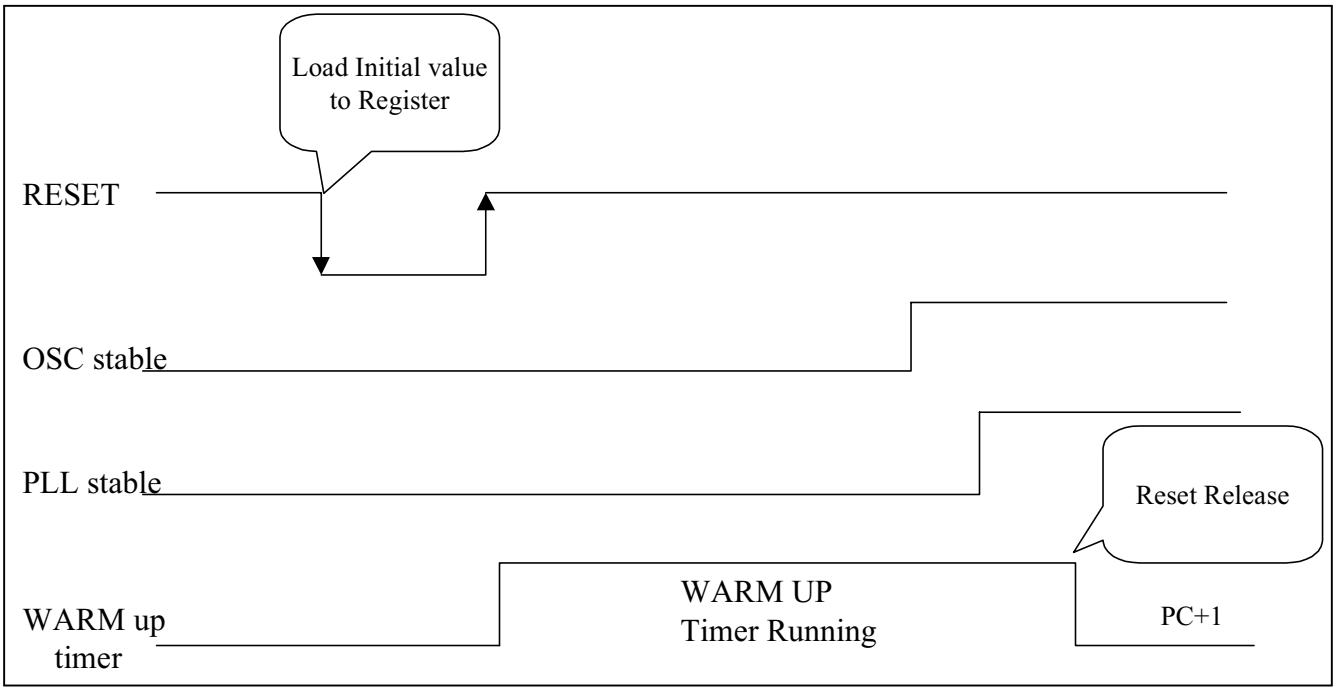
The reset can be caused by :

1. Power on reset.
2. Reset pin(SYSTEM RESET or Reset_key) is at low condition(level hold)
3. WDT time out(if WDT enable)

If the CPU reset is caused by WDT time out, the **/TO** bit of STATUS register will be cleared. Set to 1 during power up or by "WDTC" instruction or when entering SLEEP MODE.

If the CPU changed to SLEEP MODE, the **/PD** bit of STATUS register will be cleared. Set to 1 during power up or by "WDTC" instruction .

Once the RESET occurs, special function register will be reset to initial value except the **/TO, /PD** bit of STATUS register .



EVENT	/TO	/PD	REMARK
WDT time out reset from SLEEP MODE	0	0	
WDT time out reset (not SLEEP MODE)	0	1	
Wake up from SLEEP MODE by SPI or Input port	1	0	
Power up or RESET pin low condition	1	1	

Initialization after RESET occurs:

- The oscillator is running, or will be started.
- The Watchdog timer is cleared.
- If power on reset or RESET pin low condition, the **/TO** bit and **/PD** bit of RF(STATUS) are set to "1". If WDT time out reset, the **/TO** bit is cleared.
- The program counter(PCH:PCM:PCL) is clear to all "0".
- The other register initial value is as following.

Special Register:

Addr.	NAME	Initial value	Addr.	NAME	Initial value
00h	**INDF0**	---- ----(*1)	10h	**LCDDATA**	---- ----(*1)
01h	**FSR0**	0000 0000	11h	**TRL2**	uuuu uuuu
02h	**PCL**	0000 0000	12h	**PRODL**	uuuu uuuu
03h	**PCM**	0000 0000	13h	**PRODH**	uuuu uuuu
04h	**PCH**	---- ---0	14h	**SPRL**	xxxx xxxx
05h	**BSR**	0000 0000	15h	**SPRM**	xxxx xxxx
06h	**STKPTR**	0000 0000	16h	**SPRH**	xxxx xxxx
07h	**BSR1**	0000 0000	17h	**UARTTX**	xxxx xxxx
08h	**INDF1**	---- ----(*1)	18h	**UARTRX**	xxxx xxxx
09h	**FSR1**	1000 0000	19h	**PORTA**	xxxx xxxx
0Ah	**ACC**	xxxx xxxx	1Ah	**PORTB**	xxxx xxxx
0Bh	**TABPTRL**	0000 0000	1Bh	**PORTC**	xxxx xxxx
0Ch	**TABPTRM**	0000 0000	1Ch	**PORTD**	xxxx xxxx
0Dh	**TABPTRH**	0000 0000	1Dh	**PORTE**	xxxx xxxx
0Eh	**CPUCON**	000x 0x0c(*2)	1Eh	**PORTF**	xxxx xxxx
0Fh	**STATUS**	cuxx xxxx (*3)	1Fh	**PORTG**	xxxx xxxx

Control Register:

Addr.	NAME	Initial value	Addr.	NAME	Initial value
20h	**PFS**	0010 0000	30h	**POST_ID**	1111 0000
21h	**PACON**	0000 01--	31h	**TRL0L**	uuuu uuuu
22h	**STBCON**	0-00 0000	32h	**TRL0H**	uuuu uuuu
23h	**DCRB**	1111 1111	33h	**TRL1**	uuuu uuuu
24h	**DCRC**	1111 1111	34h	**TRCON**	0000 0000
25h	**DCRD**	1111 1111	35h	**MTRL**	uuuu uuuu
26h	**DCREF**	0011 0011	36h	**MWTCON**	0000 0000
27h	**DCRG**	-000 0011	37h	**ELCON**	00uu uuuu
28h	(reserved)	---- ----	38h	**PAINTSTA**	0000 0000
29h	**UARTCON**	0000 0010	39h	**PAINTEN**	0000 0000
2Ah	**UARTSTA**	0000 0000	3Ah	**INTSTA**	0000 0000
2Bh	**SPICON**	0000 0000	3Bh	**INTCON**	0000 0000

2Ch	**SPISTA**	---- 0000	3Ch	**PORTH**	xxxx xxxx
2Dh	**PAWAKE**	0000 0000	3Dh	**PORTI**	xxxx xxxx
2Eh	**LCDARL**	0000 0000	3Eh	**DCRHI**	0011 0011
2Fh	**LCDARH**	0000 0000	3Fh	**PBCON**	0000 0000

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

Note:

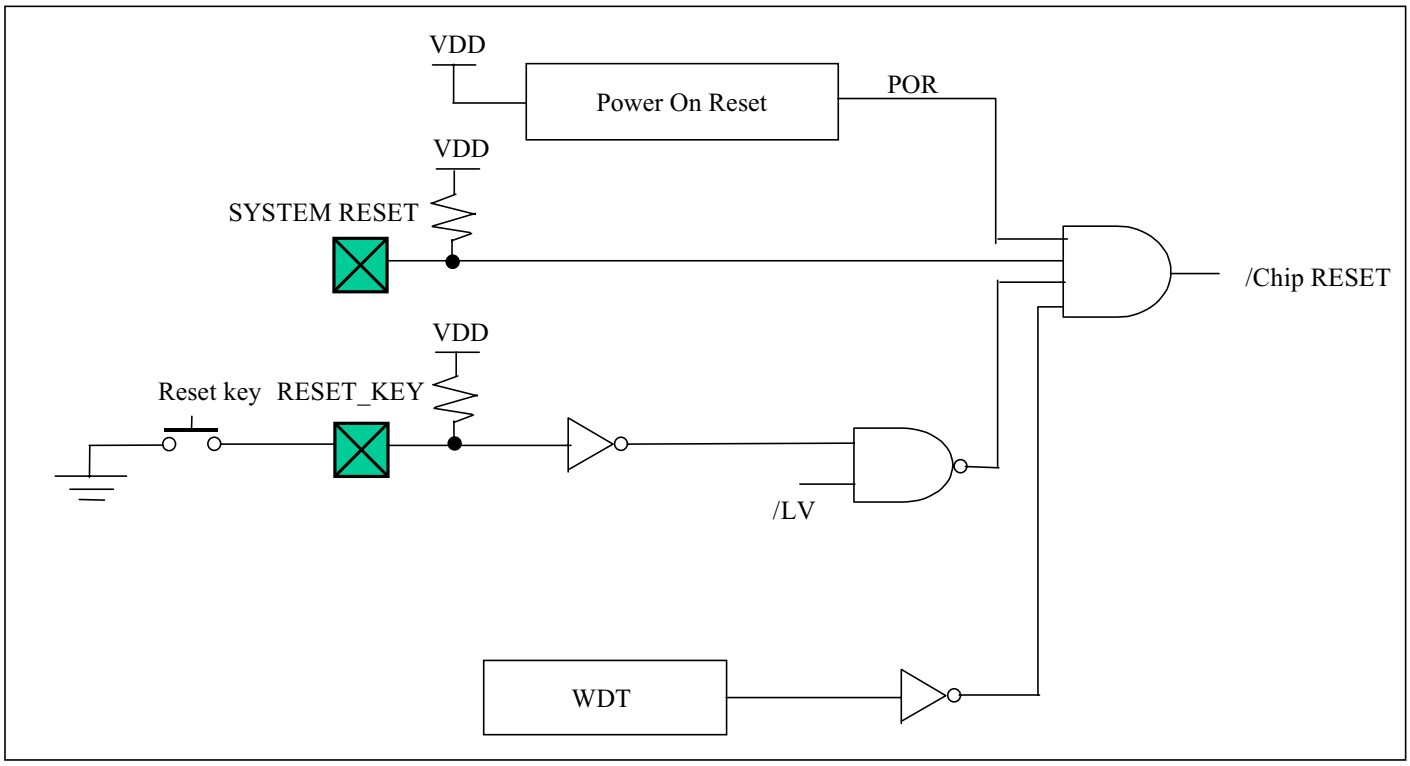
(*1)Not a physical register

(*2)Bit0(MS0) of RE(CPUCON) is reload from "**IM**" bit of code option when CPU reset.

(*3) If power on reset or RESET pin low condition, the **/TO** bit and **/PD** bit of RF(STATUS) are set to "1".

If WDT time out reset, the **/TO** bit is cleared and **/PD** bit unchanged.

Block Diagram of On-chip RESET circuit:



WAKE UP:

When SLEEP MODE , oscillator was off . CPU will be awaken by input port or SPI receive full then return to FAST MODE or SLOW MODE (determine by **MS0** bit of CPUCON register).

And when IDLE MODE, oscillator keeps running. CPU will be awaken by Timer1 , input port or SPI receive full then return to FAST MODE or SLOW MODE (determine by **MS0** bit of CPUCON register).

The **T1WKEN** bit of TRCON register can enable/disable Timer1 wake up function. The input port wake up function is enable/disable by PAWAKE register. And SPI wake function is enable/disable by **SPWKEN** bit of SPISTA register.

Register Description:

- **TRCON(Timer0 & Timer1 control register)**

bit 7							bit0
T1WKEN	T1EN	T1PSR1	T1PSR0	T0EN	T0CS	T0PSR1	T0PSR0

◆ **T1WKEN:** Enable bit of Timer1 underflow wake up function in IDLE MODE.
0: Disable Timer1 wake up function .
1: Enable Timer1 wake up function.

- **PAWAKE (Port A WAKE UP Control Register)**

bit 7							bit0
WKEN7	**WKEN6**	**WKEN5**	**WKEN4**	**WKEN3**	**WKEN2**	**WKEN1**	**WKEN0**

◆ **WKEN7~WKEN0**: Wake up enable control bit of Port A.
0: Disable Port A wake up function.
1: Enable Port A wake up function.

- **SPISTA(SPI status register):**

bit 7							bit0
-	-	-	-	**SPWKEN**	SMP	DCOL	RBF

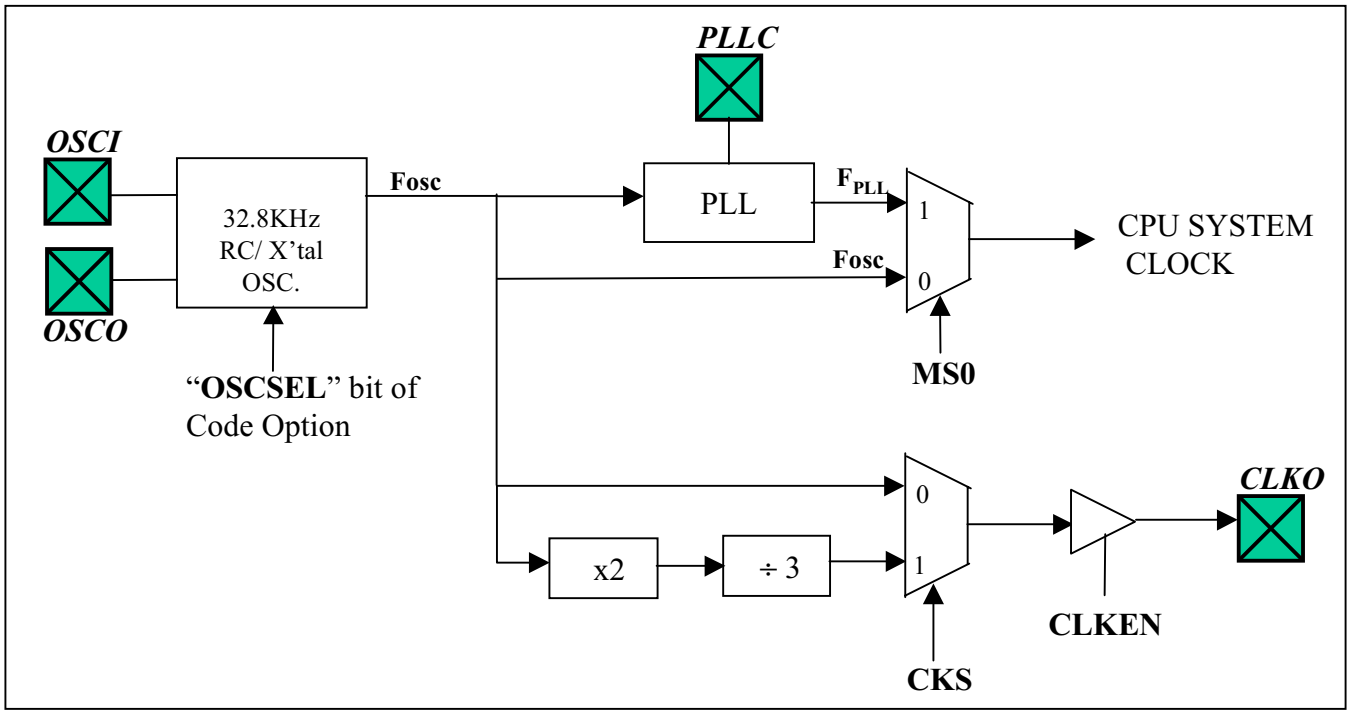
◆ **SPWKEN:** SPI wake up enable control bit.
0:Disable SPI (slave mode) read buffer full wakeup.
1: Enable SPI (slave mode) read buffer full wakeup.

(4) Oscillator System:

Feature:

- 32.8KHz RC OSC(Ext. R and Int. C) or 32768Hz Crystal.
- Built-in PLL, high frequency range can be selected from 1MHz to 10MHz by changing PFS register.
- Clock output with frequency selectable in IDLE MODE, SLOW MODE and FAST MODE.
- Operating voltage from VDD=2.2V~3.6V
- 32.8k RC OSC frequency deviation ±15% @Vdd=2.2~3.6V , Ta=25ºC.

Function Block Diagram:



Pin Description:

- **OSCI:** Input pin for 32.768KHz crystal oscillator or external Resistor- internal Capacitor 32.8KHz oscillator. Selected by Code Option.
- **OSCO:** Output pin for 32.768KHz crystal oscillator. If R-C oscillator is selected, this pin should be

floating.

- **CLKO:** Clock output pin for external device(LCD Driver or slave CPU ,etc.). **This pin is tri-state when clock output disable.(CLKOEN=0)**
- **PLLC:** Phase lock loop capacitor, connect a capacitor with AVSS.

Register Description:

- **PFS:** PLL frequency select register. System clock can be fine tuned from 1MHz to 10MHz. The initial value of PFS register after RESET will be set to "20h" (F_{PLL}=2.097 MHz)

F_{PLL} = 2 * PFS * Fosc

PFS reg.	F_{PLL}(MHz)	PFS reg.	F_{PLL}(MHz)
0~3	*N.A.(*1)*	92	6.029
4	0.262	107	7.012
15	0.983	122	7.995
31	2.032	137	8.978
46	3.015	*150*	*9.83 (*3)*
61	3.998	153	10.027
76	4.981	255	16.712

Note: 1. PFS=0 ~ 3 is not available.
2. The Maximum range of PLL is from 262.144 K ~ 16.712 MHz
3. When enable UART, system clock should be 9.83MHz
4. The table is base on 32.768KHz oscillator frequency.

- **CPUCON Register:**

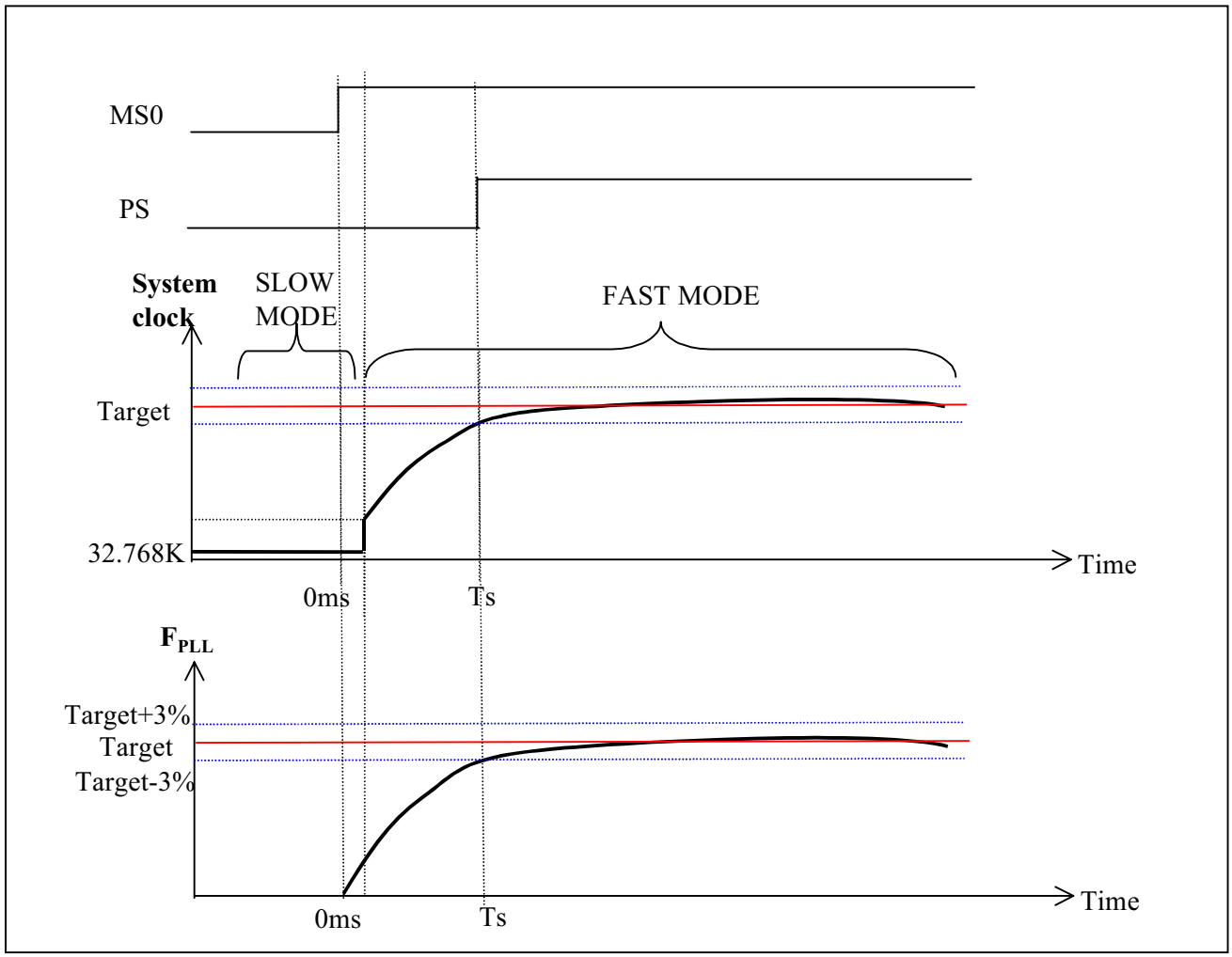
bit 7 bit0

CLKOEN	**CKS**	LVDEN	/LV	/GLINTD	**PS**	MS1	MS0

- **CLKOEN**: Clock output enable control bit.
  CLKO pin is tri-state when clock output disable.**(CLKOEN=0)**
- **CKS:** clock select bit of clock output pin (**CLKO**).
  0: CLKO pin output frequency at 32.768KHz(Fosc)
  1: CLKO pin output frequency at 21.845KHz(Fosc/1.5).

◆ **PS**: PLL stable flag. This bit will be cleared when PLL not turned on or PLL is turned on but not stable. And be set after PLL turned on and frequency stable.
PLL stable means actual frequency within target frequency +/- 3%.
0: PLL is not turned on or PLL frequency not yet stable .
1: PLL is turned on and frequency stable.

Timing Diagram:



Note: 1.Slow mode switch to Fast mode at Time=0ms
2.System clock will switch to F_{PLL} immediately,and system clock will be hundreds KHz then.
3. When PLLST_OPT code option =0, PLL frequency will be stable (+/-10%) less than 5ms.
And PLLST_OPT code option =1, PLL frequency will be stable (+/-3%) around 2 to 9ms.

(5) INTERRUPT:

There are 3 types of interrupt for the whole operation:

Level 1: Input port (Port A)interrupt .

Level 2: Timer interrupt. Include of Timer0(**TMR0I**), Timer1(**TMR1I**), Timer2 (**TMR2I**) interrupt.

Level 3: Peripheral interrupt. Include of Low voltage detected (**LVDI**), SPI read buffer full (**SRBFI**) , UART receiving error(**UERRI**) , UART transfer buffer empty (**UTXI**),UART receiver buffer full(**URXI**).

The **/GLINTD** bit of CPUCON register disable all interrupts, include of LEVEL 1~LEVEL 3. Set this bit to 1 can enable all un-mask interrupt.

Interrupt vector :

Interrupt level	Interrupt source	Start address	Remark
	RESET	0x00000	
Level 1	Input Port	0x00002	PAINT
Level 2	Timer	0x00004	Timer0,Timer1,Timer2
Level 3	Peripheral	0x00006	SRBFI,UERRI,UTXI,URXI,LVDI

Register Description:

- **CPUCON(CPU control register)**

bit 7 bit0

CLKOEN	CKS	LVDEN	/LV	**/GLINTD**	PS	MS1	MS0

◆ **/GLINTD:** Global interrupt disable bit.

0:Disable all interrupt.

1:Enable all un-mask interrupt.

- **PAINTEN (Port A INTERRUPT Enable Control Register)**

bit 7 bit0

PA7IE	**PA6IE**	**PA5IE**	**PA4IE**	**PA3IE**	**PA2IE**	**PA1IE**	**PA0IE**

◆ **PA7IE~PA0IE**: Control bit of port A interrupt .

0: Disable interrupt function.

1: Enable interrupt function.

- **PAINTSTA (Port A INTERRUPT STATUS Register)**

bit 7 bit0

PA7I	**PA6I**	**PA5I**	**PA4I**	**PA3I**	**PA2I**	**PA1I**	**PA0I**

  - **PA7I~PA0I**: INT status of Port A interrupt. Set to 1 when pin **falling edge** detected. Clear to 0 by software.

- **INTCON (INT Control Register)**

bit 7 bit0

URXIE	**UTXIE**	**UERRIE**	**SRBFIE**	**LVDIE**	**TMR2IE**	**TMR1IE**	**TMR0IE**

0: Disable interrupt function.

1: Enable interrupt function.

  - **TMR0IE**: Control bit of TIMER0 interrupt.
  - **TMR1IE**: Control bit of TIMER1 interrupt.
  - **TMR2IE**: Control bit of TIMER2 interrupt.
  - **LVDIE**: Control bit of LVD interrupt.
  - **SRBFIE**: Control bit of SPI read buffer full interrupt
  - **UERRIE**: Control bit of UART receiving error interrupt.
  - **UTXIE**: Control bit of UART Transfer buffer empty interrupt.
  - **URXIE**: Control bit of UART Receiver buffer full interrupt.

- **INTSTA (INT STATUS Register)**

bit 7 bit0

URXI	**UTXI**	**UERRI**	**SRBFI**	**LVDI**	**TMR2I**	**TMR1I**	**TMR0I**

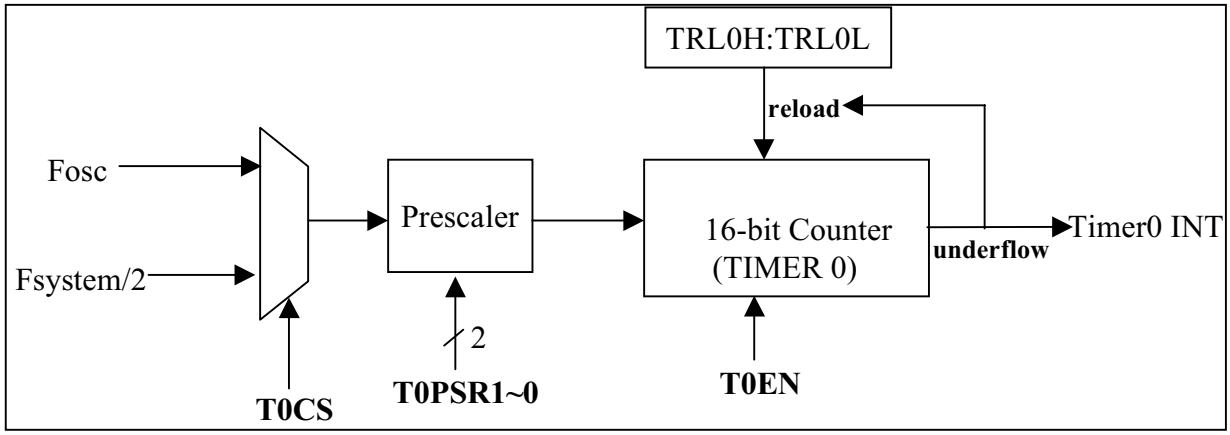
  - **TMR0I**: Set to 1 when TMR0 underflow happened. Clear to 0 by software or Timer0 disable.
  - **TMR1I**: Set to 1 when TMR1 underflow happened. Clear to 0 by software or Timer1 disable.
  - **TMR2I**: Set to 1 when TMR2 underflow happened. Clear to 0 by software or Timer2 disable.
  - **LVDI**: Set to 1 when LVD low level detected(**Falling edge**). Clear to 0 by software or LVD disable.
  - **SRBFI**: Set to 1 when SPI read buffer full happened. Clear to 0 by software or SPI disable.
  - **UERRI**: Set to 1 when UART receiving error happened. Clear to 0 by software or UART disable.
  - **UTXI**: Set to 1 when UART transfer buffer empty happened. Clear to 0 by software or UART TX disable(TXE=0)
  - **URXI**: Set to 1 when UART receiver buffer full happened. Clear to 0 by software or UART RX disable(RXE=0)

(6) PERIPHERAL :

A. Timer/Melody Timer/EL Timer/Watchdog Timer

1. Timer0(16 bits):

Function Block Diagram:



Function Description:

Timer0 is a general purpose 16 bits down counter for some applications required time counting. There is an interrupt available for user's application. The clock source is selectable from the oscillator clock or half of the system clock.

There is a prescaler for the timer. The **T0PSR1~T0PSR0** bit of TRCON register determine the prescale ratio and generate different clock rate as the clock source for the timer.

Counter value will be decreased by one(count down) according to the timer clock source. When underflow happens, the timer interrupt will be triggered if the global interrupt and timer0 interrupt are both enabled. At the same time,TRL0H:TRL0L will automatic reload into 16 bits counter .

The Timer0 frequency range is from 1/128 Hz(clock source is from Fosc, TRL0H:TRL0L=FFFFh, prescaler=1:64) to 5MHz(clock source is from Fsystem/2, system clock=10MHz, TRL0H:TRL0L=0h, prescaler=1:1).

$$T = \frac{1}{Freq.} * Prescale * (TRL0H:TRL0L+1)$$

Register Description:

- **TRL0H:TRL0L(Timer 0 reload register):**

  Used to store the auto reload value of TIMER0. When enabling Timer0 or underflow happens, TRL0H:TRL0L register will automatically reload into 16 bits counter .

- **TRCON(Timer0 & Timer1 control register)**

bit 7 bit0

T1WKEN	T1EN	T1PSR1	T1PSR0	**T0EN**	**T0CS**	**T0PSR1**	**T0PSR0**

  - **T0PSR1~T0PSR0**: Timer0 Prescaler select bits.

T0PSR1:T0PSR0	Prescale value
00	1:1
01	1:4
10	1:16
11	1:64

  - **T0CS:** Timer 0 clock source select bit.
    0: Clock source is from Fosc.
    1: Clock source is from half of the system clock.
  - **T0EN:** Timer0 enable control bit
    0: Disable Timer0(stop counting) .
    1: Enable Timer0 .

- **INTCON (INT Control Register)**

bit 7 bit0

URXIE	UTXIE	UERRIE	SRBFIE	LVDIE	TMR2IE	TMR1IE	**TMR0IE**

  - **TMR0IE**: Control bit of TIMER0 interrupt.
    0: Disable interrupt function.
    1: Enable interrupt function.

- **INTSTA (INT STATUS Register)**

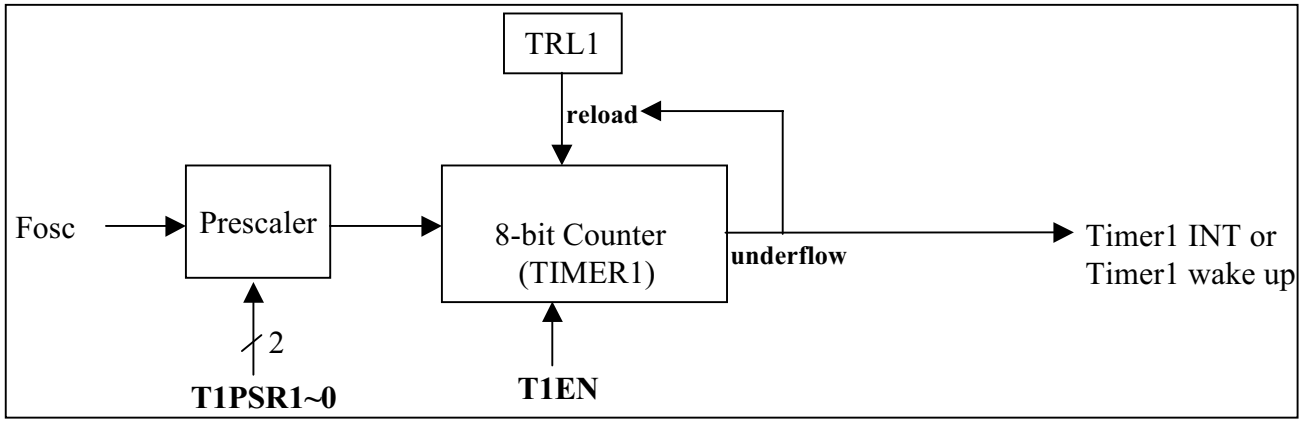
bit 7 bit0

URXI	UTXI	UERRI	SRBFI	LVDI	TMR2I	TMR1I	**TMR0I**

  - **TMR0I**: Set to 1 when TMR0 underflow happened. Clear to 0 by software or Timer0 disable.

2.Timer1(8 bits):

Function Block Diagram:



Function Description:

Timer1 is a general purpose 8 bits down counter for some applications required time counting. There are interrupt and wake up function available for user's application. The clock source is from the oscillator clock.

There is a prescaler for the timer. The **T1PSR1~T1PSR0** bit of TRCON register determine the prescale ratio and generate different clock rate as the clock source for the timer. Set **T1WKEN** bit of TRCON register to 1 will enable Timer 1 underflow wake up function in IDLE MODE.

Counting value will be decreased by one(count down) according to the timer clock source. When underflow happens, the timer interrupt will be triggered if the global interrupt and timer1 interrupt are both enabled. At the same time,TRL1 will automatic reload into 8 bits counter.

The Timer1 frequency range is from 0.5 Hz(TRL1=FFh, prescaler=1:256) to 8.192KHz(TRL1=0h, prescaler=1:4).

$$T = \frac{1}{Fosc} * Prescale * (TRL1+1)$$

Register Description:

- **TRL1(Timer 1 reload register):**
  Used to store the auto reload value of TIMER1. When enabling Timer1 or underflow happens, TRL1 register will automatically reload into 8 bits counter .

- **TRCON(Timer0 & Timer1 control register)**

bit 7 bit0

T1WKEN	**T1EN**	**T1PSR1**	**T1PSR0**	T0EN	T0CS	T0PSR1	T0PSR0

  - **T1PSR1~T1PSR0**: Timer1 Prescaler select bits.

T1PSR1:T1PSR0	Prescale value
00	1:4
01	1:16
10	1:64
11	1:256

  - **T1EN:** Timer1 enable control bit
    0: Disable Timer1(stop counting) .
    1: Enable Timer1 .
  - **T1WKEN:** Enable bit of Timer1 underflow wake up function in IDLE MODE.
    0: Disable Timer1 wake up function .
    1: Enable Timer1 wake up function.

- **INTCON (INT Control Register)**

bit 7 bit0

URXIE	UTXIE	UERRIE	SRBFIE	LVDIE	TMR2IE	**TMR1IE**	TMR0IE

  - **TMR1IE**: Control bit of TIMER1 interrupt.
    0: Disable interrupt function.
    1: Enable interrupt function.

- **INTSTA (INT STATUS Register)**

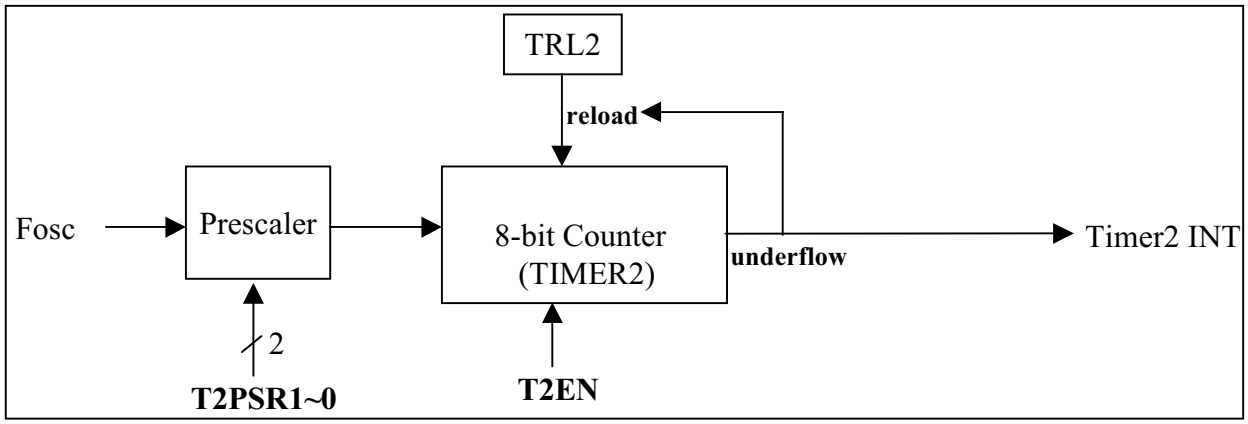
bit 7 bit0

URXI	UTXI	UERRI	SRBFI	LVDI	TMR2I	**TMR1I**	TMR0I

  - **TMR1I**: Set to 1 when TMR1 underflow happened. Clear to 0 by software or Timer1 disable.

3.Timer2(8 bits):

Function Block Diagram:



Function Description:

Timer2 is a general purpose 8 bits down counter for some applications required time counting. There is an interrupt available for user's application. The clock source is from the oscillator clock.

There is a prescaler for the timer. The **T2PSR1~T2PSR0** bit of MWTCON register determine the prescale ratio and generate different clock rate as the clock source for the timer.

Counting value will be decreased by one(count down) according to the timer clock source. When underflow happens, the timer interrupt will be triggered if the global interrupt and timer2 interrupt are both enabled. At the same time,TRL2 will automatic reload into 8 bits counter

The Timer2 frequency range is from 16 Hz(TRL2=FFh, prescaler=1:8) to 32.768KHz(TRL2=0h, prescaler =1:1).

$$T=\frac{1}{Fosc}*Prescale*(TRL2+1)$$

Register Description:

- **TRL2(Timer 2 reload register):**

  Used to store the auto reload value of TIMER2. When enabling Timer2 or underflow happens, TRL2

register will automatically reload into 8 bits counter .

- **MWTCON(Melody Timer , WDT and Timer 2 control register)**

bit 7 bit0

T2EN	**T2PSR1**	**T2PSR0**	WDTEN	WDTPSR1	WDTPSR0	MTEN	MDO2EN

◆ **T2PSR1~T2PSR0**: Timer2 Prescaler select bits.

T2PSR1:T2PSR0	Prescale value
00	1:1
01	1:2
10	1:4
11	1:8

◆ **T2EN:** Timer2 enable control bit

0: Disable Timer2(stop counting) .

1: Enable Timer2 .

- **INTCON (INT Control Register)**

bit 7 bit0

URXIE	UTXIE	UERRIE	SRBFIE	LVDIE	**TMR2IE**	TMR1IE	TMR0IE

◆ **TMR2IE**: Control bit of TIMER2 interrupt.

0: Disable interrupt function.

1: Enable interrupt function.

- **INTSTA (INT STATUS Register)**

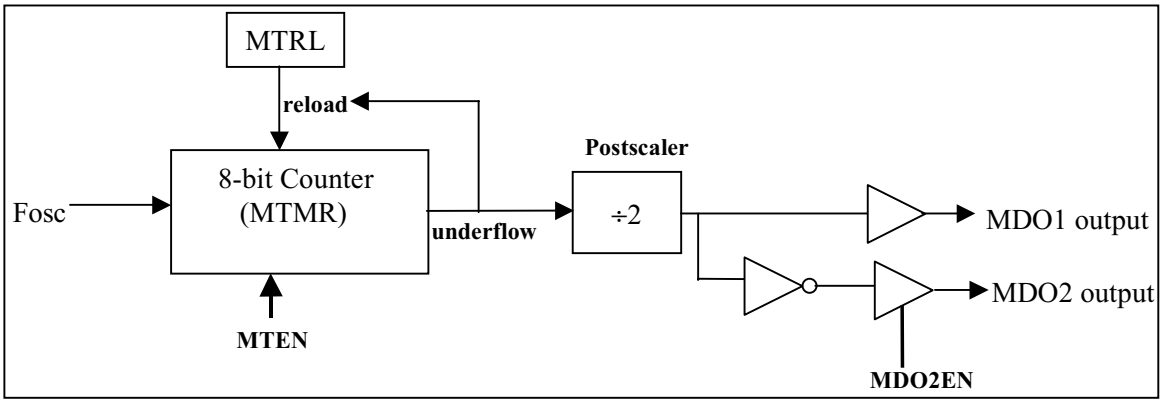
bit 7 bit0

URXI	UTXI	UERRI	SRBFI	LVDI	**TMR2I**	TMR1I	TMR0I

◆ **TMR2I**: Set to 1 when TMR2 underflow happened. Clear to 0 by software or Timer2 disable.

4.Melody Timer(8 bits):

Function Block Diagram:



Function Description:

Melody Timer is suitable for counting the tempo and rhythm of music. There are positive/negative output available for user's application. The clock source is from the oscillator clock(32.768KHz).

When set **MTEN to 1** will enable Melody Timer.

Counting value will be decreased by one(count down) according to the timer clock source. When enabling Melody Timer or underflow happens, MTRL will automatic reload into 8 bits counter. And **MDO1** and **MDO2** pin output level will toggle at the same time. **MDO2** pin output is the inverse of **MDO1** pin.

The frequency range of melody timer is from 64 Hz(MTRL=FFh, postscaler=1:2) to 16KHz(MTRL=0h, postscaler=1:2).

$$F = Fosc * \frac{1}{(MTRL+1)} * \frac{1}{2}$$

Pin Description:

- **MDO1(Melody positive output):** Melody positive output pin.
- **MDO2(Melody negative output):** Melody negative output pin.

Register Description:

- **MTRL(Melody Timer reload register):**
  Used to store the auto reload value of melody timer. When enabling Melody Timer or underflow happens, MTRL will automatically reload into 8 bits counter .

- **MWTCON(Melody Timer , WDT and Timer 2 control register)**

bit 7 bit0

T2EN	T2PSR1	T2PSR0	WDTEN	WDTPSR1	WDTPSR0	**MTEN**	**MDO2EN**

  - **MTEN**: Melody Timer enable control bit.
  - **MDO2EN**: Melody negative output port enable control bit.

Melody Table:

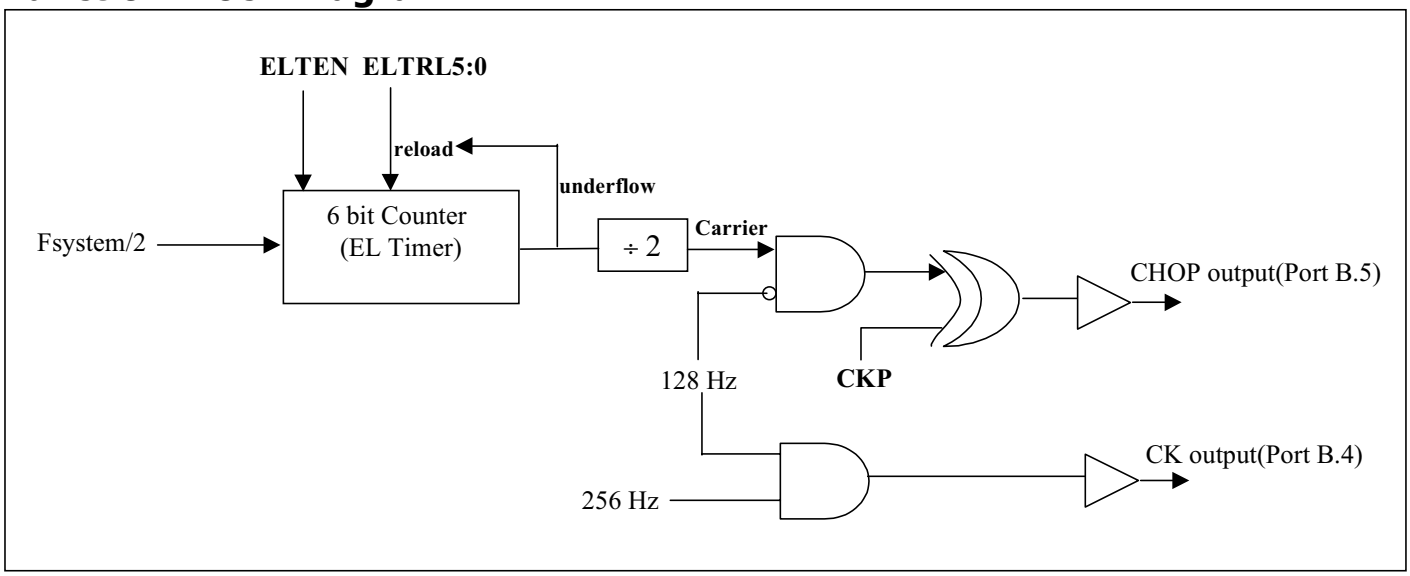
Clock source : 32768 Hz

Musical note	Standard Frequency(Hz)	MTRL	Postscaler	F_{MELODY}(Hz)	DEV(%)
L1	130.813	**124**	1:2	131.07	0.20%
L1#	138.591	**117**	1:2	138.85	0.18%
L2	146.832	**111**	1:2	146.29	-0.37%
L2#	155.563	**104**	1:2	156.04	0.31%
L3	164.814	**98**	1:2	165.49	0.41%
L4	174.614	**93**	1:2	174.30	-0.18%
L4#	184.997	**88**	1:2	184.09	-0.49%
L5	195.998	**83**	1:2	195.05	-0.48%
L5#	207.652	**78**	1:2	207.39	-0.13%
L6	220.000	**73**	1:2	221.41	0.64%
L6#	233.082	**69**	1:2	234.06	0.42%
L7	246.942	**65**	1:2	248.24	0.53%
M1	261.626	**62**	1:2	260.06	-0.60%
M1#	277.183	**58**	1:2	277.69	0.18%
M2	293.665	**55**	1:2	292.57	-0.37%
M2#	311.127	**52**	1:2	309.13	-0.64%
M3	329.628	**49**	1:2	327.68	-0.59%
M4	349.228	**46**	1:2	348.60	-0.18%
M4#	369.994	**43**	1:2	372.36	0.64%

M5	391.995	**41**	1:2	390.10	-0.48%
M5#	415.305	**38**	1:2	420.10	1.16%
M6	440.000	**36**	1:2	442.81	0.64%
M6#	466.164	**34**	1:2	468.11	0.42%
M7	493.883	**32**	1:2	496.48	0.53%
H1	523.251	**30**	1:2	528.52	1.01%
H1#	554.365	**29**	1:2	546.13	-1.48%
H2	587.330	**27**	1:2	585.14	-0.37%
H2#	622.254	**25**	1:2	630.15	1.27%
H3	659.255	**24**	1:2	655.36	-0.59%
H4	698.456	**22**	1:2	712.35	1.99%
H4#	739.989	**21**	1:2	744.73	0.64%
H5	783.991	**20**	1:2	780.19	-0.48%
H5#	830.609	**19**	1:2	819.20	-1.37%
H6	880.000	**18**	1:2	862.32	-2.01%
H6#	932.328	**17**	1:2	910.22	-2.37%
H7	987.767	**16**	1:2	963.76	-2.43%

5.EL Timer(6 bits):

Function Block Diagram:



Function Description:

EL Timer is a 6 bit down counter which is suitable for counting the CHOP signal output. The clock source is from 1/2 system clock(1MHz~10MHz).

Counting value will be decreased by one(count down) according to the timer clock source. When underflow happens, the CHOP output level will be toggled. When enabling EL timer or underflow happens , ELTRL will automatic reload into 6 bits counter.

ELTEN bit is used to enable EL timer and change Port B.5 and Port B.4 to CHOP and CK output pin. **CKP** bit is used to select clock polarity of CHOP output.(See timing diagram for detail)

The frequency range of CHOP carrier signal will be 128Hz(System clock at 32768Hz, ELTRL=3Fh) to 2.5MHz(System clock at 10MHz, ELTRL=0h).

$$Fchop = (Fsystem / 2) * \frac{1}{(ELTRL+1)} * \frac{1}{2}$$

Pin Description:

- **CHOP pin:** Chop output pin of EL driver.
- **CK pin:** CK output pin of EL driver.

Register Description:

- **ELCON(EL control register)**

bit 7 bit0

ELTEN	CKP	ELTRL5 ~ ELTRL0

◆ **ELTEN:** EL Timer enable control bit

0: Disable EL Timer(stop counting) and recover CK and CHOP pin to general I/O pin.

1: Enable EL Timer and change Port B.4 and Port B.5 to CK and CHOP output pin.

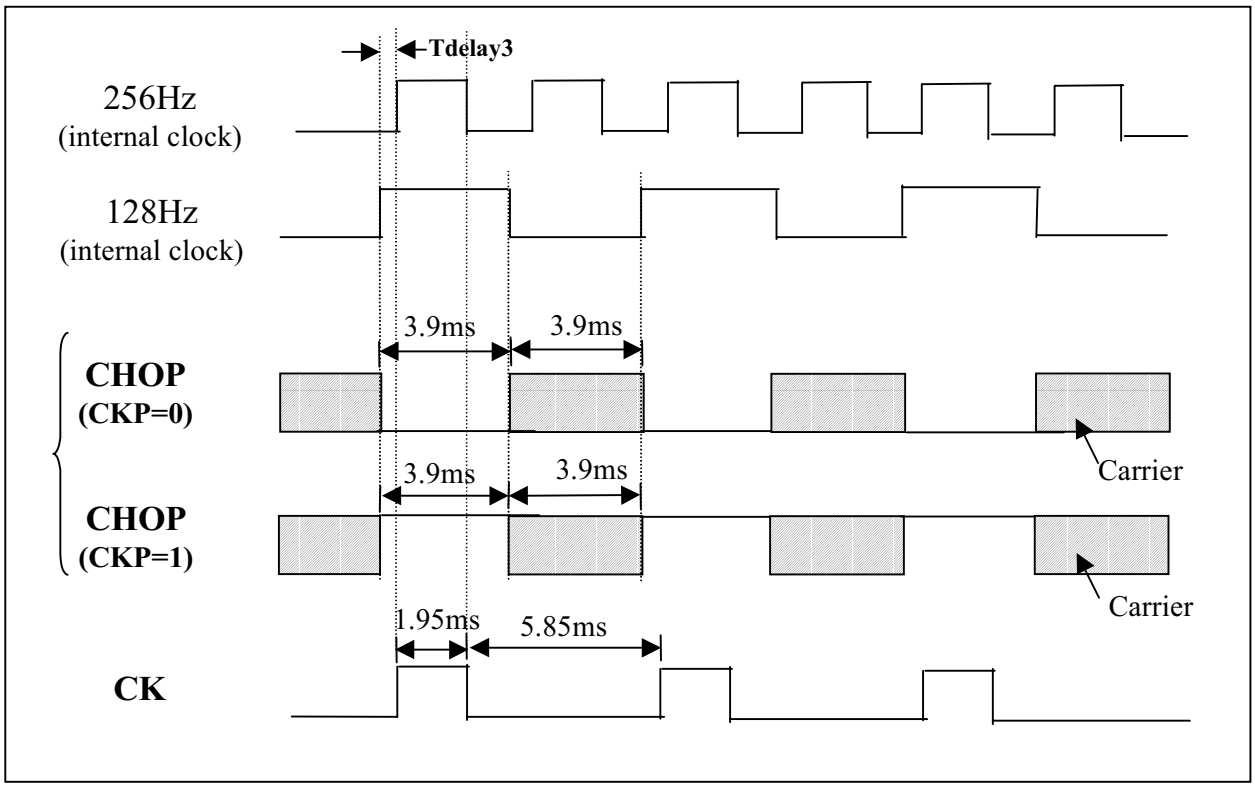
◆**CKP**: EL clock polarity select bit.

0: Idle state for CHOP pin is low level

1: Idle state for CHOP pin is high level

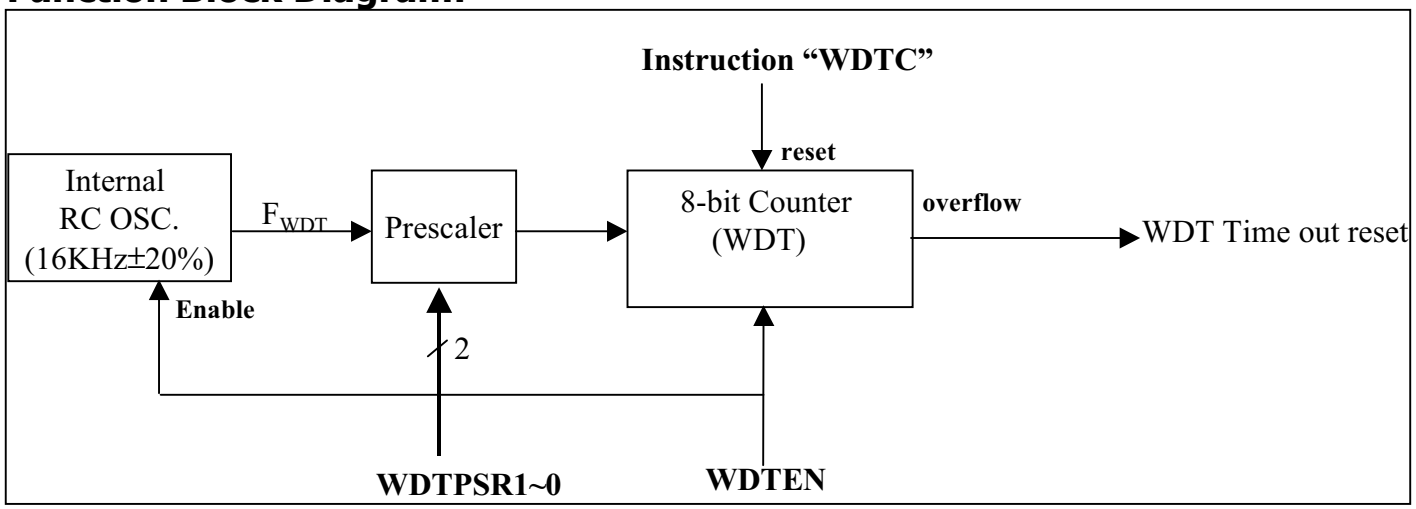
◆**ELTRL5~0:** Used to store the auto reload value of EL timer. When enabling EL timer or underflow happens, **ELTRL5~0** will automatically reload into 6 bits counter .

Timing Diagram:



6.Watch Dog Timer(WDT):

Function Block Diagram:



Function Description:

The clock source of watch dog timer(WDT) is from on-chip RC oscillator(16KHz±20%) . The WDT will keep running even the oscillator has been turned off (i.e. in SLEEP MODE).

WDT time-out will cause the CPU reset(if WDT enabled). To avoid the reset happens, user should clear the WDT value by using "WDTC" instruction before WDT time-out. Set **WDTEN** bit will enable WDT running. The initial state of WDT is disable.

There is also a prescaler to generate different clock rate for the clock source of WDT. The prescaler ratio is defined by **WDTPSR1 & WDTPSR0**.

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

Register Description:

- **MWTCON(Melody Timer , WDT and Timer 2 control register)**

bit 7							bit0
T2EN	T2PSR1	T2PSR0	**WDTEN**	**WDTPSR1**	**WDTPSR0**	MTEN	MDO2EN

  - **WDTPSR1~WDTPSR0**: Watch Dog timer Prescaler select bits.

WDTPSR1:WDTPSR0	Prescale value
00	1:4

01	1:16
10	1:64
11	1:128

- **WDTEN:** Watch Dog Timer enable bit

  0: Disable watch dog timer(stop running) .

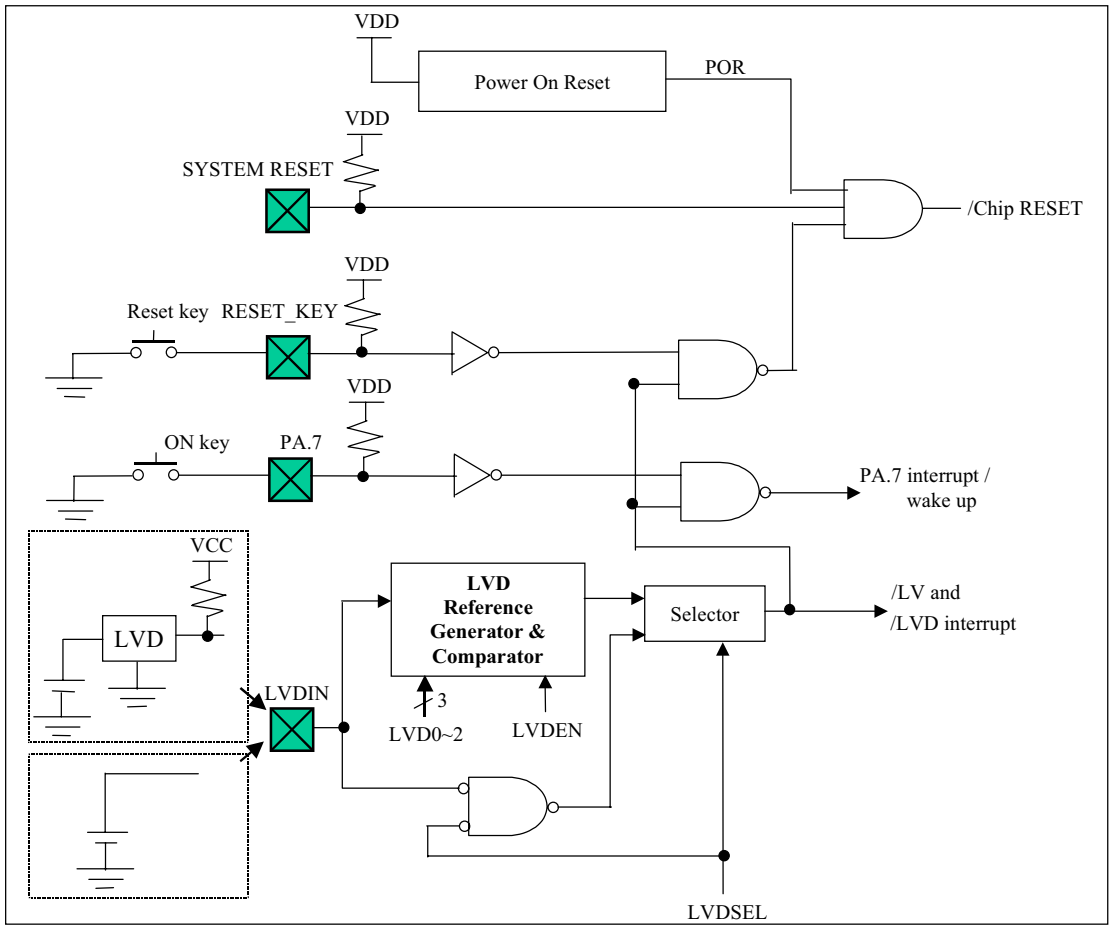
  1: Enable watch dog timer .

B. LOW VOLTAGE DETECTION(LVD)

Feature:

- Built-in LVD (Low Voltage Detector),
  $V_{det(Vth-)}$=2.2V ~ 2.9V ±0.1V. selectable.
- When the voltage of LVDIN pin is lower than selected detection voltage(/LV=0),
  PA.7 interrupt/Wake up is disabled.
  And the reset function of RESET_KEY pin is unavailable.
- Built-in LVD/ External LVD selectable.
- LVD interrupt optional.

Block Diagram:



Function Description:

Low Voltage Detection function is used to detect Battery Low. When enable **LVDEN** bit of CPUCON Register, **/LV** bit is the voltage detection result of **LVDIN** pin. When the voltage of **LVDIN** pin is lower than Vdet(Vth-), **/LV** bit will be clear.

Setup **LVDSEL** bit of STBCON register to select either external LVD device or internal built-in LVD.

Setup **LVD2~LVD0** bit of PACON register to determine the detection voltage.

User can enable Low Voltage Detected Interrupt function. If **/LV** bit becomes low, interrupt will happen at the same time.

When low level detected(/LV=0), the PA.7 interrupt/wake up is disable. And the reset caused by RESET_KEY pin is disable.

Pin Description:

- **LVDIN:** Input pin of Low Voltage Detection to examine the voltage level higher than Vdet(Vth+) or lower than Vdet(Vth-).

Register Description:

- **CPUCON Register:**

bit 7 bit0

CLKOEN	CKS	**LVDEN**	**/LV**	/GLINTD	PS	MS1	MS0

◆**/LV**: Low voltage detected. This is a read only bits. When the voltage **LVDIN** pin is lower than Vdet(Vth-), this bit will be clear. Otherwise if higher than Vdet(Vth+) ,this bit will be set.

0: Low voltage is detected.

1: Low voltage is not detected or LVD is disable.

/LV bit will be set to 1 when LVD disable.

◆**LVDEN:** Enable Low Voltage Detector.

LVDEN bit will be clear after CPU reset.

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

bit 7 bit0

LVD2	**LVD1**	**LVD0**	**LVDSEL**	Bit7PU	/REN	-	-

◆ **LVD2~0**: LVD detection voltage select bits.

The default value of LVD2~0 after CPU reset is "000".

LVD2~0	Vdet(Vth-)	Vdet(Vth+)	Deviation	LVD2~0	Vdet(Vth-)	Vdet(Vth+)	Deviation
000	2.2V	2.3V	±0.1V	100	2.6V	2.7V	±0.1V
001	2.3V	2.4V		101	2.7V	2.8V	
010	2.4V	2.5V		110	2.8V	2.9V	
011	2.5V	2.6V		111	2.9V	3.0V	

- **LVDSEL:** Built in LVD/External LVD select
  0: External LVD selected.
  1: Built-in LVD selected.

- **INTCON (INT Control Register)**

bit 7 bit0

URXIE	UTXIE	UERRIE	SRBFIE	**LVDIE**	TMR2IE	TMR1IE	TMR0IE

- **LVDIE**: Control bit of LVD interrupt.
  0: Disable interrupt function.
  1: Enable interrupt function.

- **INTSTA (INT STATUS Register)**

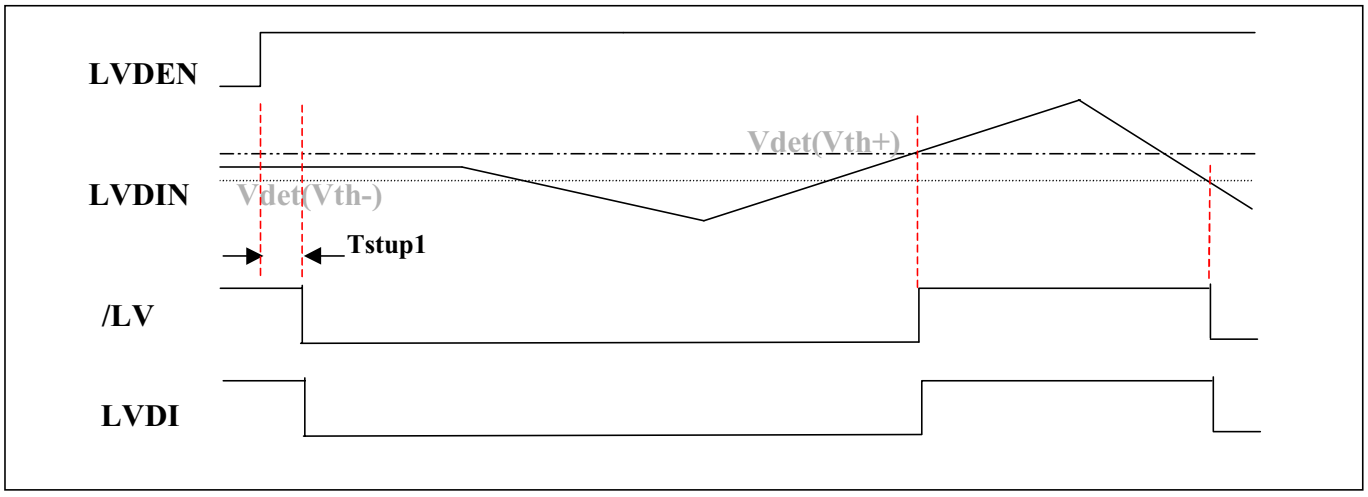
bit 7 bit0

URXI	UTXI	UERRI	SRBFI	**LVDI**	TMR2I	TMR1I	TMR0I

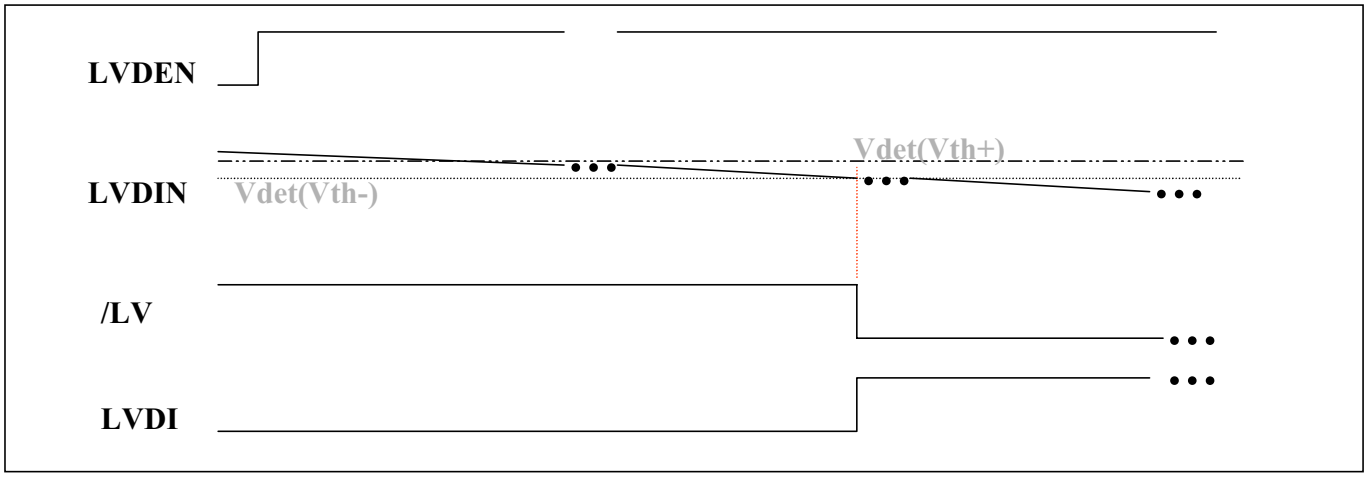
- **LVDI**: Set to 1 when LVD low level detected(**Falling edge**). Clear to 0 by software or LVD disable.

Timing diagram:

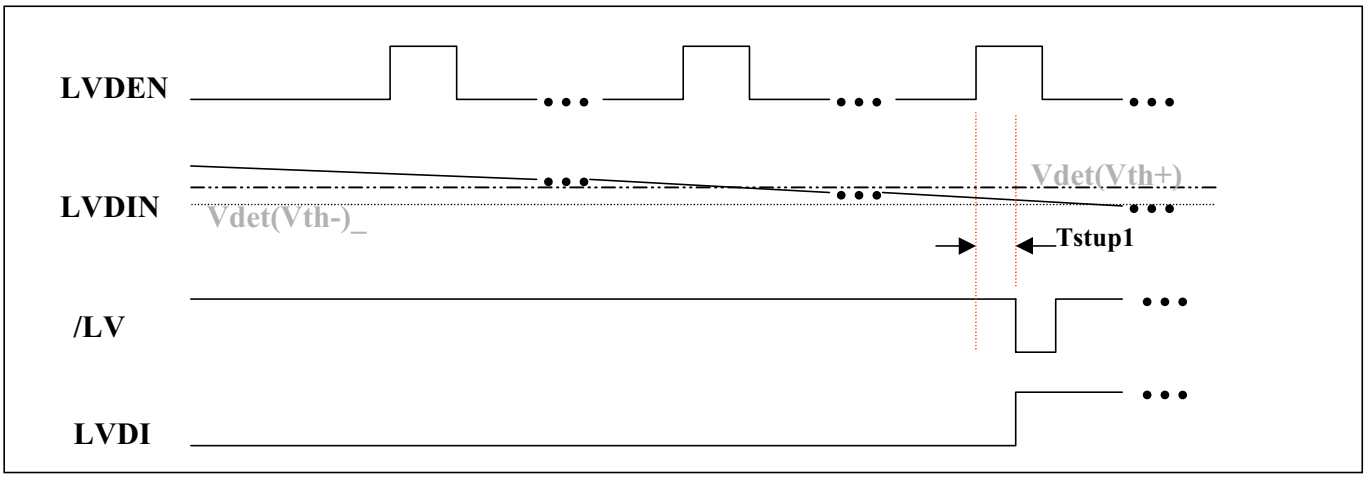
(1)LVDIN voltage is within Vdet threshold region when enabling LVD: *LOW voltage detected*.



(2)Continuous LVD Detection



(3) Power Saving Control of LVD



C. SERIAL PERIPHERAL INTERFACE (SPI)

Features:

- 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 bit rates of communication.
- Programmable clock polarity.
- Programmable shift direction.
- Programmable sample phase
- Interrupt flag available for the read buffer full.
- Up to 2.5MHz (system clock at 10MHz) bit frequency.

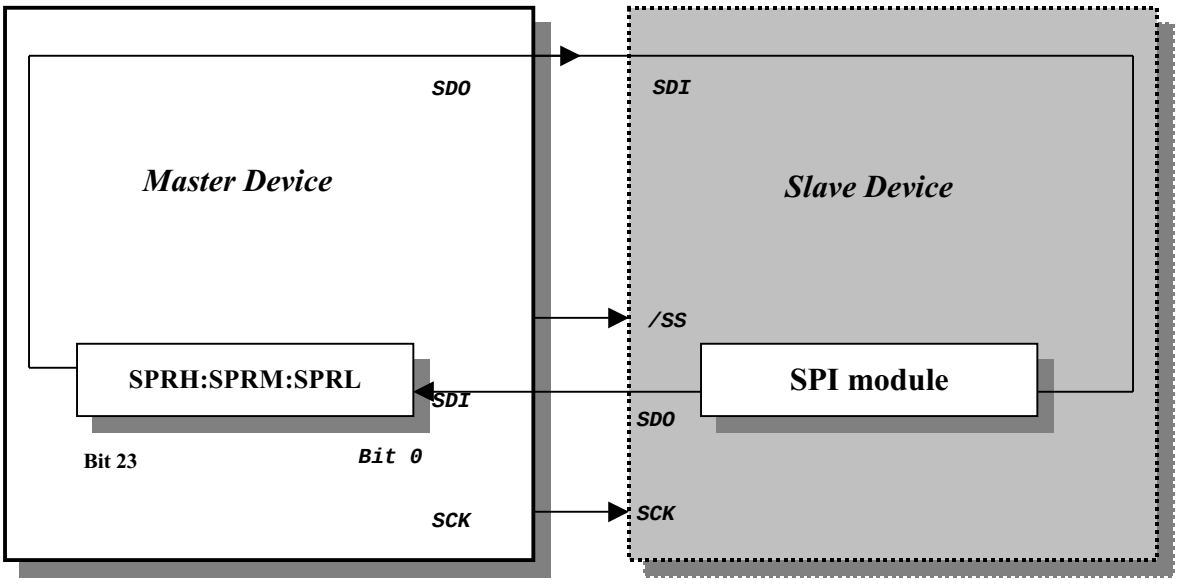


Fig 1 Single SPI Master / Slave Communication

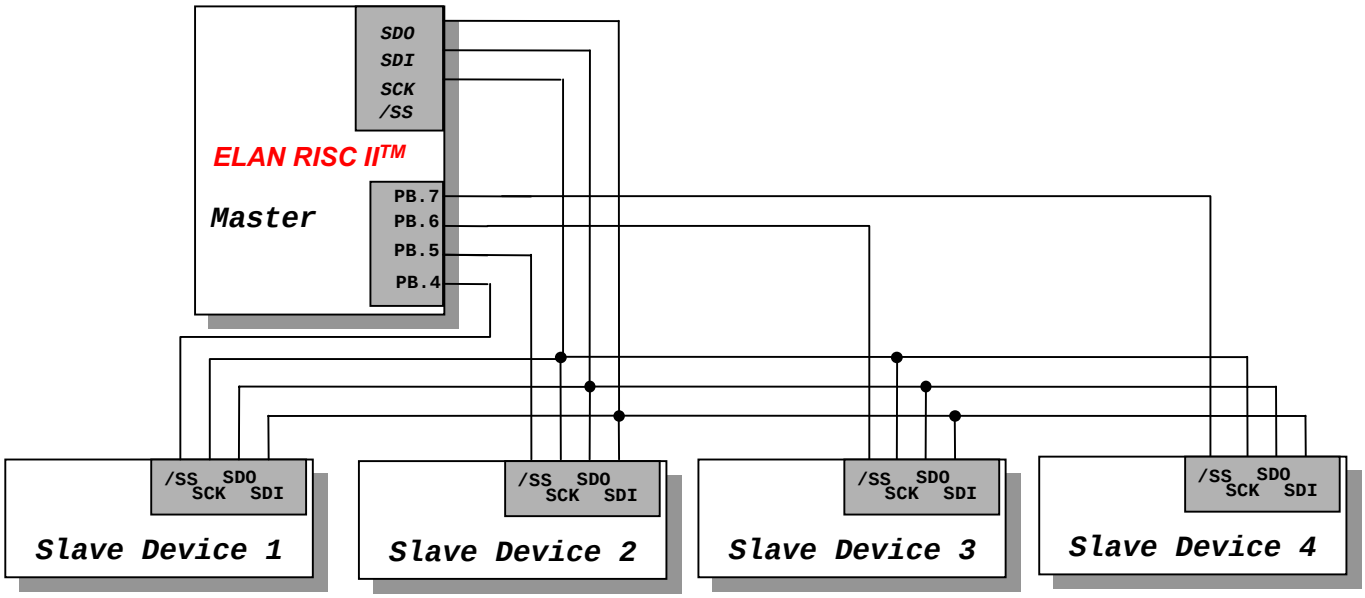


Fig 2 The SPI configuration example of Single-Master and Multi-Slaves

Function Block:

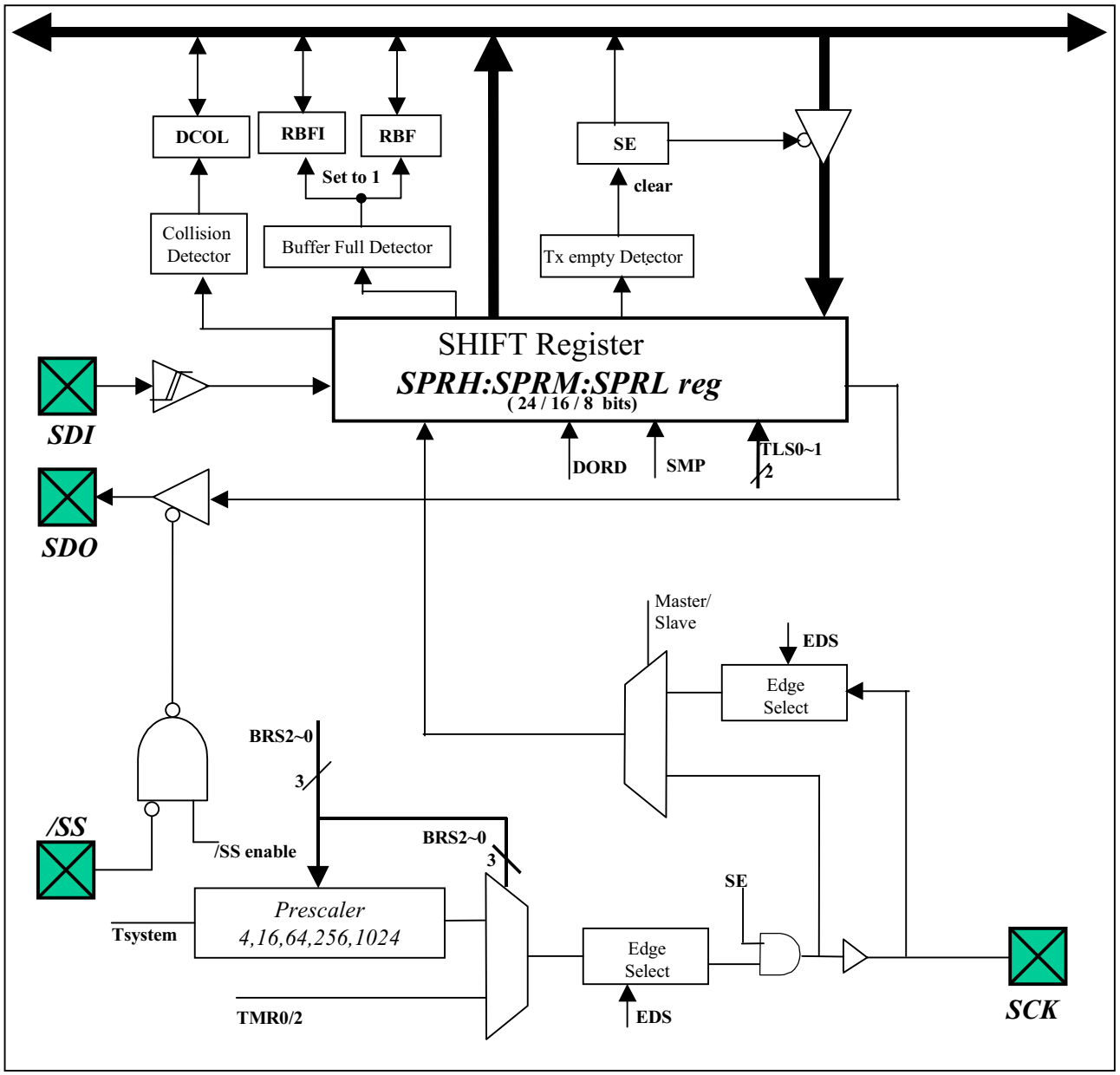


Fig 3 SPI block diagram

Function Description

Fig. 1 shows how MCU communicates with other devices by SPI module. If MCU is a master controller, it sends clock through the SCK pin. An 8-bit data is transmitted and received at the same time. If 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 the selected edge.

Setup **TLS1~TLS0** bit of SPICON register to select the shift register length of SPI and enable/disable SPI function. Setup **BRS2~BRS0** bit of SPICON register to select the SPI mode(master/slave) and Bit Rate. When master mode, the clock source can be selected from system clock or half of timer 0 interval. When slave mode, the /SS pin can be enable or disable. Set up the **DORD** bit of SPICON register can determine the shift direction. Set up the **EDS** bit of SPICON register can select raising edge or falling edge latch the data. Set up the **SMP** bit of SPISTA register can select the sample phase at middle or the end of data output time.

(1)MASTER MODE:

In master mode, SCK pin is as a clock output pin.

If select 24 bits shift register length, SPRH, SPRM and SPRL register is the high, middle and low byte of the shift register. (If select 8 bits shift register length, SPRL register is the content of the shift register.) During write data to SPRH, SPRM and SPRL register, after write data into SPRL register, **SE** bit of SPICON register will be set by hardware automatically and start shifting. After shift buffer empty, **SE** bit will be clear by hardware and stop clock output from **SCK** pin.

Receiver is active while SPI transfer. When receive buffer full, **RBF** flag will be set and interrupt happens(if enable). During read out the shift register content, after SPRL register has been read out, hardware will clear the **RBF** flag automatically. If SPRL register not been read out, **RBF** bit still remains set. Data collision will happens when next clock in.

(2)SLAVE MODE:

In slave mode, input clock is from MASTER device. SCK pin is as a clock input pin. The **SE** bit is not useful to control starting shift in this mode. But it is a Transfer buffer empty status bit.

The same as MASTER MODE, user can select shift register length. And write transfer data to SPRH, SPRM, SPRL register. After write data into SPRL register, **SE** bit of SPICON register will be set by hardware. But the start shifting is control by the MASTER device clock input.

While shift buffer empty the **SE** bit will be clear. At the same time, receive buffer was full, **RBF** flag will be set and interrupt happens(if enable). The received data is at SPRH, SPRM and SPRL register, user should read them out before next clock in. Otherwise the data was collision and **DCOL** bit of SPI STATUS register will be set.

Pin Description:

- **SDI(I):** Serial Data Input.
  Receive serially.
- **SDO(O):** Serial Data Output .
  Transmit serially.
  In slave mode, defined as high-impedance, if not selected.
- **SCK(I/O):** Serial Clock input/output . When Master mode, sends clock through the SCK pin. However, if defined as a slave, its SCK pin is programmed as an input pin .
- **/SS(I):** /Slave Select . This pin (/SS) will be active when enable /SS function (BRS=110),else /SS pin is a general purpose I/O.
  Master device remains low to /SS pin to signify the slave(s) for transmit/receive data. Ignores the data on the SDI and SDO pins while /SS is high, because the SDO is no longer driven.

Register Description:

- **SPRH:SPRM:SPRL:** SPI shift buffer for 24/16/8 bits length.
  The buffer will deny any write until the shifting is completed.
  If user select 24 bits shift buffer, SPRH:SPRM:SPRL register is the content of 24 bits shift buffer. Else if select 8 bits shift buffer, SPRL register is the content of the shift register.
  When write data into SPRL register, **SE** bit of SPICON will be set by hardware and start shifting.
  While shift buffer empty ,at the same time, receive buffer was full, the received data is shifted in SPRH, SPRM and SPRL register. After SPRL register has been read out, hardware will clear the **RBF** flag automatically.

- **SPICON(SPI Control register):**

bit 7							bit0
TLS1	**TLS0**	**BRS2**	**BRS1**	**BRS0**	**EDS**	**DORD**	**SE**

  - **TLS1~TLS0:** Shift buffer length select. 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

◆ **BRS2~BRS0:** Bit rate select. Programming 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

Prescaler		Fsystem		
BRS2:0	**Bit rate**	**10MHz**	**4MHz**	**32.768KHz**
001	Fsystem/4	2500000	1000000	8196
010	Fsystem/16	625000	250000	2048
011	Fsystem/64	156250	62500	512
100	Fsystem/256	39063	15625	128
101	Fsystem/1024	9766	3906	32

Unit: bits/sec

Table. SPI bit rate table

◆ **EDS:** Select the raising / falling edges latch by programming the EDS bit

0: Falling edge
1: Raising edge

◆ **DORD:** Data transmission order.

0:Shift left (MSB first)
1:Shift right (LSB first)

◆ **SE:** Shift enable.

Set to 1 automatically when write data into SPRL register and begin to shift.
Reset to 0 when transfer buffer empty detected.
SE bit will be clear by hardware when enabling SPI. And **SE** bit is read-only . Therefore, write SPRL register is necessary when want to start shifting.

- **SPISTA(SPI status register):**

bit 7 bit0

-	-	-	-	**SPWKEN**	**SMP**	**DCOL**	**RBF**

◆ **RBF:** Set to 1 by Buffer Full Detector, and clear to 0 automatically when read data from **SPRL** register.

RBF bit will be clear by hardware when enabling SPI. And **RBF** bit is read-only . Therefore, read SPRL register is necessary to avoid data collision happens(**DCOL**).

◆ **DCOL:** SPI Data collision.

0:Data collision didn't occurs

1:Data collision occurs. Should be cleared by software.

◆ **SMP:** SPI data input sample phase.

0:Input data sampled at middle of data output time

1:Input data sampled at the end of data output time

In slave mode, data input sample is fixed at middle of data output time.

◆ **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.

- **INTCON (INT Control Register)**

bit 7 bit0

URXIE	UTXIE	UERRIE	**SRBFIE**	LVDIE	TMR2IE	TMR1IE	TMR0IE

◆ **SRBFIE**: Control bit of SPI read buffer full interrupt

0: Disable interrupt function.

1: Enable interrupt function.

- **INTSTA (INT STATUS Register)**

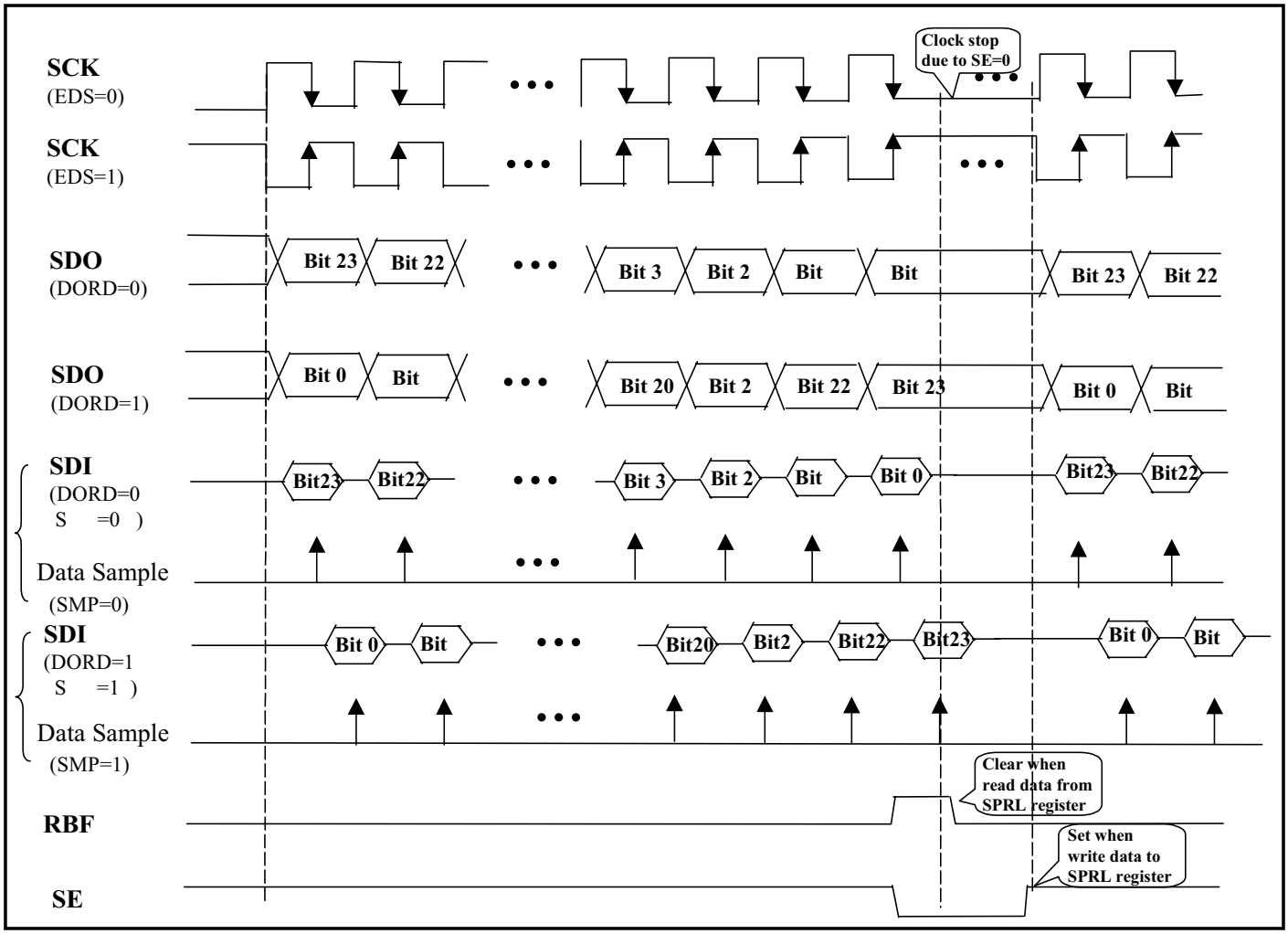
bit 7 bit0

URXI	UTXI	UERRI	**SRBFI**	LVDI	TMR2I	TMR1I	TMR0I

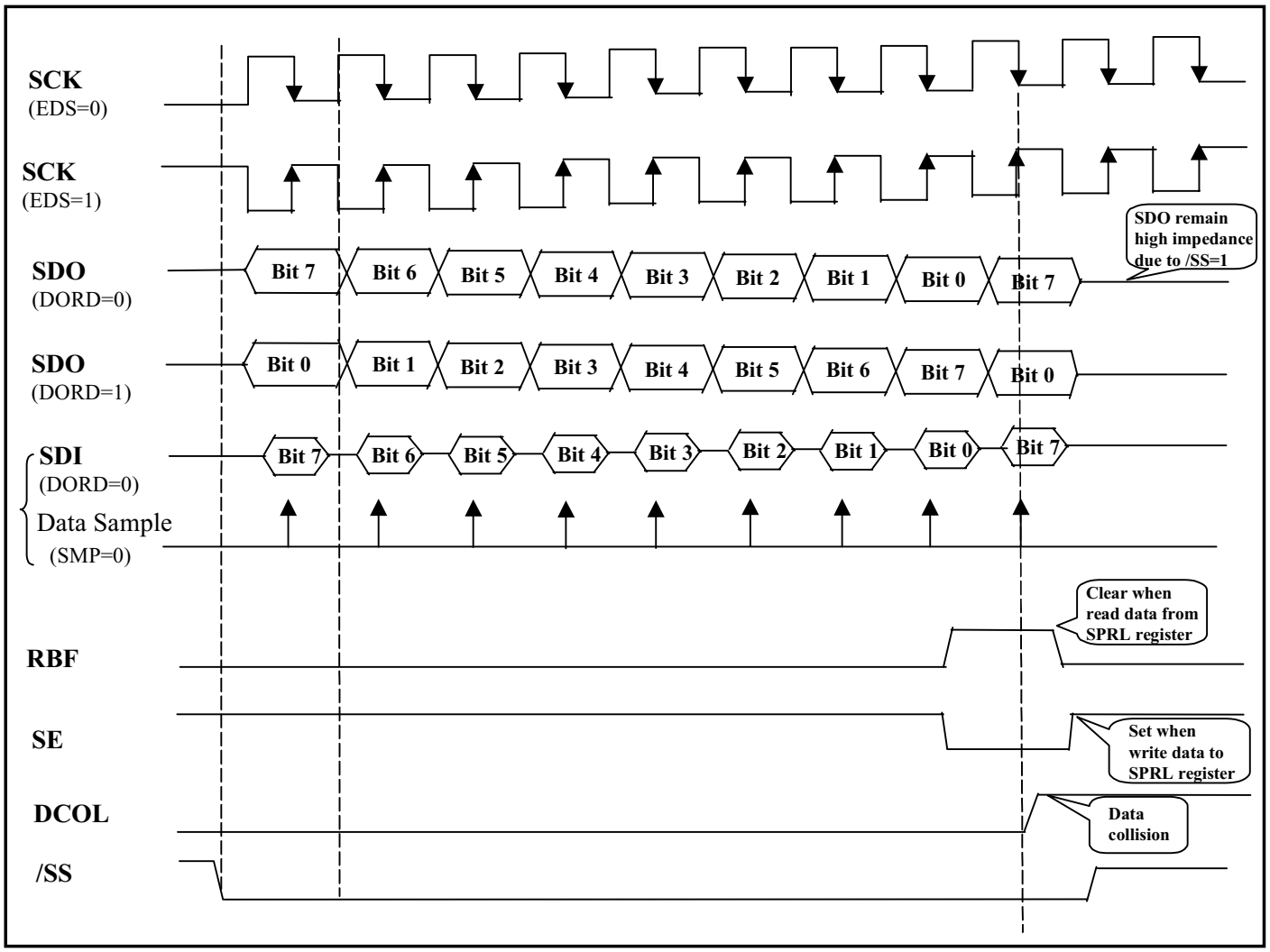
◆ **SRBFI**: Set to 1 when SPI read buffer full happened. Clear to 0 by software or SPI disable.

Timing Diagram:

(1)MASTER MODE: (Shift buffer Length = 24bits)



(2)SLAVE MODE: (Shift buffer Length = 8 bits , /SS enable)

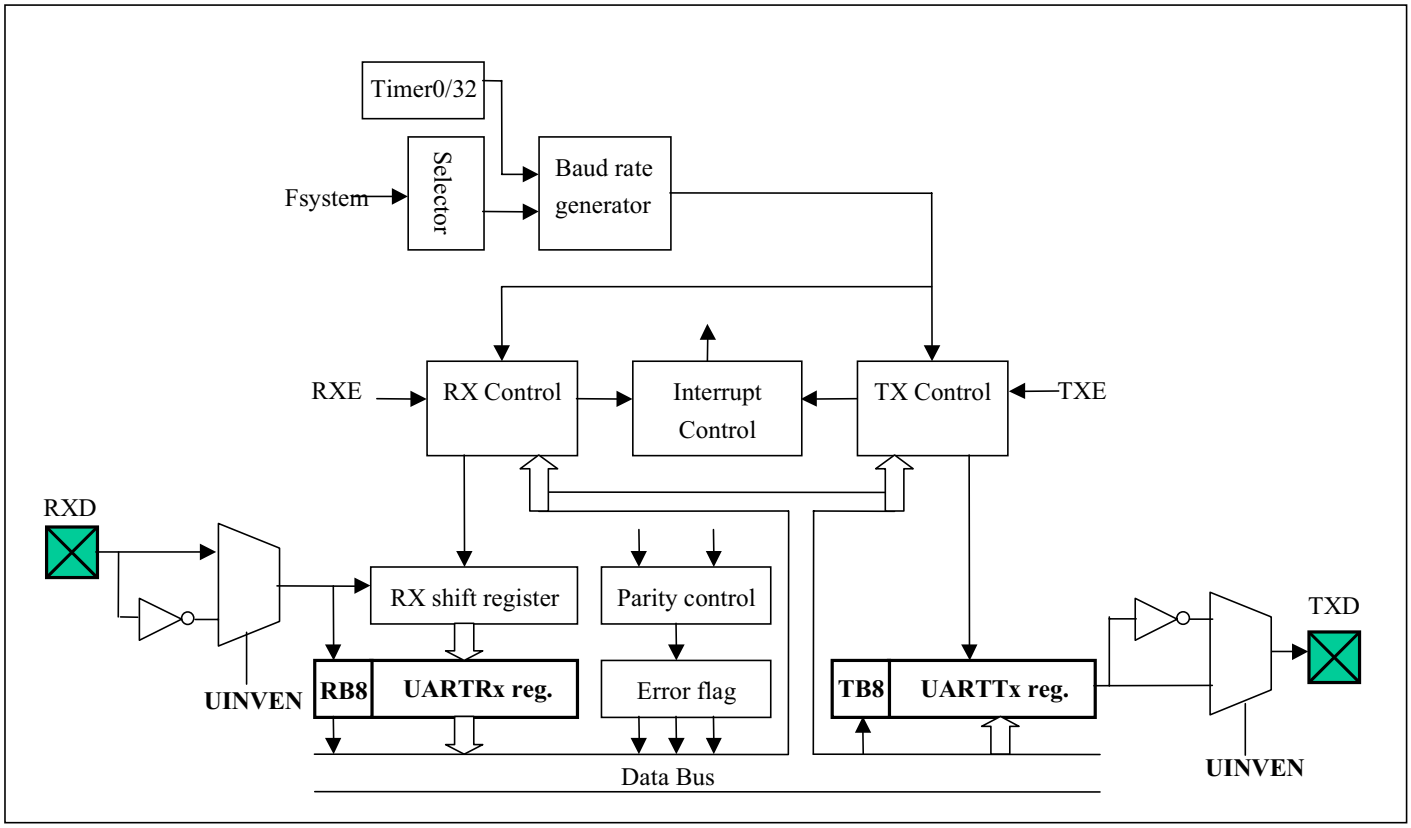


D. Universal Asynchronous Receiver Transmitter (UART)

Feature:

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

Function Block Diagram



Function Description:

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 section enable 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 mark 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) confirming the end of the frame.

In receiving, the UART synchronizes on the falling edge of the start bit. When two or more "0" are detected during 3 samples, it is recognized as normal start bit and the receiving operation is started.

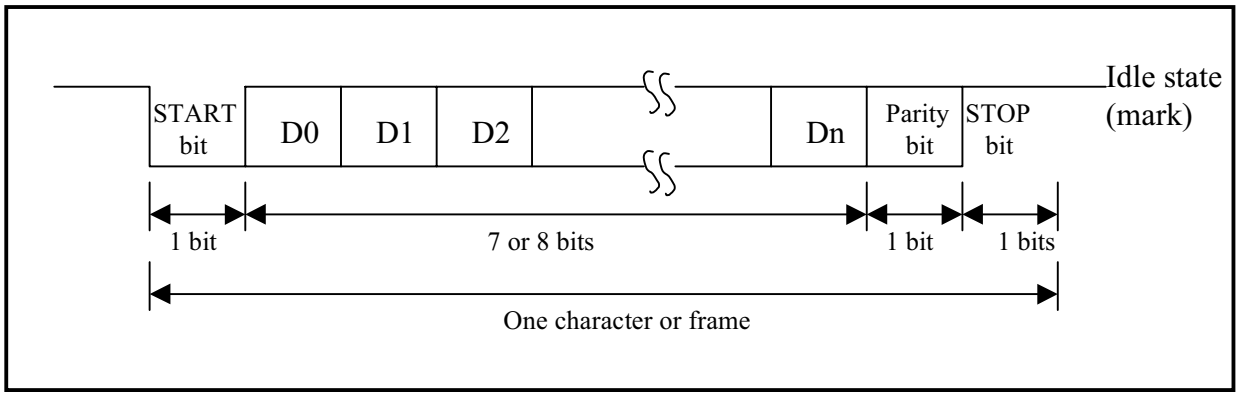


Figure. DATA Format in UART

(1)UART MODE:

There are 3 mode 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. Figure below shows the data format in each mode.

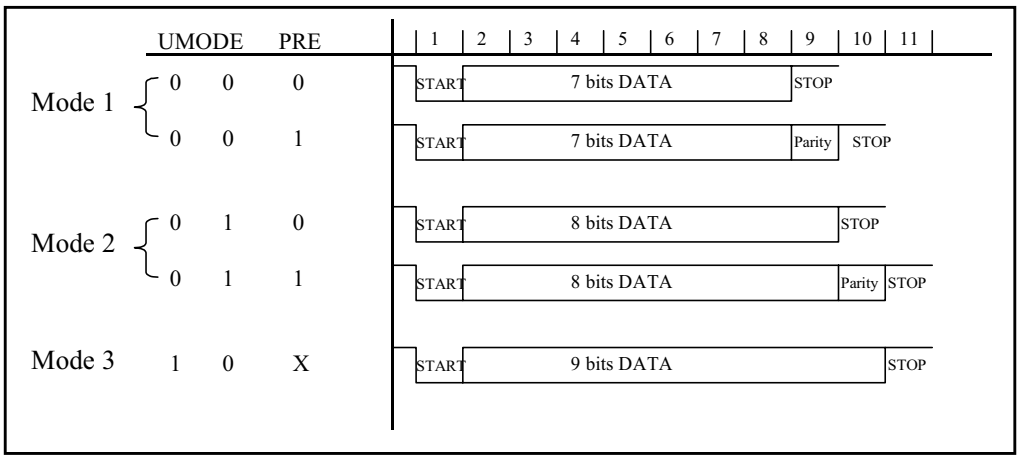


Figure. UART mode

(2)Transmission:

In transmitting serial data, the UART operates as follows.

1. Set **TXE** bit of UARTCON register to enable UART transmission function.
2. Write data into UARTTx register and **TBE** bit of UARTCON register will be set by hardware. Then start transmitting .
3. Serial transmit data are transmitted in the following order from 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 enable)

(3)Receiving:

In receiving, the UART operates as follows.

1. Set **RXE** bit of UARTCON register to enable UART receiving function.
   The UART monitors the RXD pin and synchronizes internally when it detects a start bit.
2. Receive data is shifted into UARTRx register in order from LSB to MSB.
3. The parity bit and the stop bit are received.
   After one character received, the UART generates a **URXI** interrupt (if enable). And **URBF** bit of UARTSTA register will be set to 1.
4. The UART makes the following checks:

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

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

(c)Overrun check: **URBF** bit of UARTCON register must be cleared(means UARTRx register should be read out) before next received data load into UARTRx register.

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

5.Read received data from UARTRx register. And **URBF** bit will be clear by hardware.

(4)Baud rate generator:

The baud rate generator comprises a circuit that generates a clock pulse to determine the transfer speed for transmission/receiving in the UART. **The input clock of baud rate generator is from system clock divides by 256 or Timer 0 divides by 32** . The system clock should be at 9.830MHz when UART enable.

The **BRATE2~BRATE0** bit of UARTCON register can determine the desired baud rate.

Register Description:

- **UARTCON (UART control register)**

bit 7 bit0

TB8	UMODE1	UMODE0	BRATE2	BRATE1	BRATE0	UTBE	TXE

- **TB8** : Transmission data bit8
- **UMODE1~0** : UART mode
  - 00 : Mode 1: 7-bit data
  - 01 : Mode 2: 8-bit data
  - 10 : Mode 3: 9-bit data
  - 11: reserved
- **BRATE2~0** : Baud rate selector
  - 000 : Timer0/32
  - 001 : 600 baud
  - 010 : 1200 baud
  - 011 : 2400 baud
  - 100 : 4800 baud

101 : 9600 baud
110 : 19200 baud
111 : 38400 baud

- **UTBE** : UART transfer buffer empty flag. Set to 1when transfer buffer empty. Reset to 0 automatically when write into UARTTx register.
  UTBE bit will be clear by hardware when enabling transmission. And **UTBE** bit is read-only . Therefore, write UARTTX register is necessary when want to start transmittinghifting.
- **TXE** : Enable transmission.

- **UARTSTA (UART STATUS register)**

bit 7 bit0

RB8	EVEN	PRE	PRERR	OVERR	FMERR	URBF	RXE

- **RB8** :Receiving data bit8
- **EVEN** : Select parity check
  0 : Odd parity
  1 : Even parity
- **PRE** : Enable parity addition
  0 : Disable
  1 : Enable
- **PRERR** : Parity error flag. Set to 1when parity error happened, and clear to 0 by software.
- **OVERR** : Over running error flag. Set to 1 when overrun error happened, and clear to 0 by software.
- **FMERR** : Framing error flag. Set to 1 when framing error happened, and clear to 0 by software.
- **URBF** : UART read buffer full flag. Set to 1 when one character is received . Reset to 0 automatically when read from UARTRx register.
  URBF will be clear by hardware when enabling receiving. And **URBF** bit is read-only . Therefore, read UARTRX register is necessary to avoid overrun error.
- **RXE** : Enable receiving function

- **UARTTX (UART Transfer register)**

bit 7 bit0

TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0

- **TB7~TB0** : Transmission data register
  UARTTX register is write-only.

- **UARTRX (UART Receiver register)**

bit 7 bit0

RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0

  - **RB7~RB0** : Receiving data register

    UARTRX register is read-only.

- **STBCON (Strobe output control Register)**

bit 7 bit0

UINVEN	SCAN	BitST	ALL	STB3	STB2	STB1	STB0

  - **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.

- **INTCON (INT Control Register)**

bit 7 bit0

URXIE	**UTXIE**	**UERRIE**	SRBFIE	LVDIE	TMR2IE	TMR1IE	TMR0IE

  0: Disable interrupt function.

  1: Enable interrupt function.

  - **UERRIE**: Control bit of UART receiving error interrupt.
  - **UTXIE**: Control bit of UART Transfer buffer empty interrupt.
  - **URXIE**: Control bit of UART Receiver buffer full interrupt.

- **INTSTA (INT STATUS Register)**

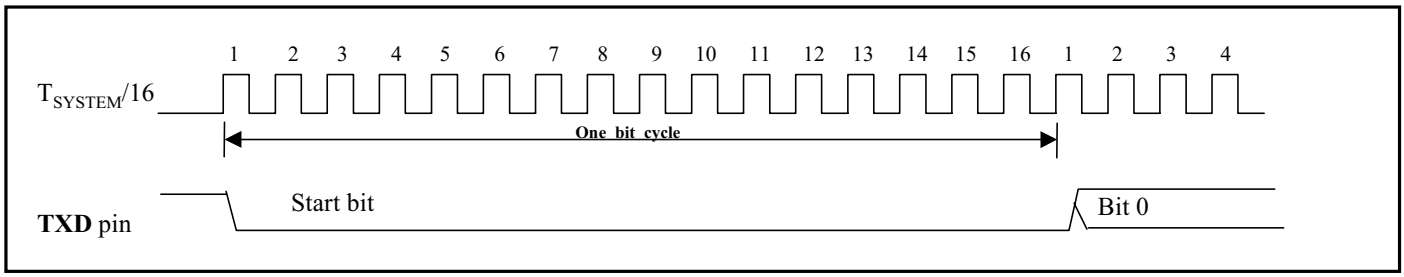
bit 7 bit0

URXI	**UTXI**	**UERRI**	SRBFI	LVDI	TMR2I	TMR1I	TMR0I

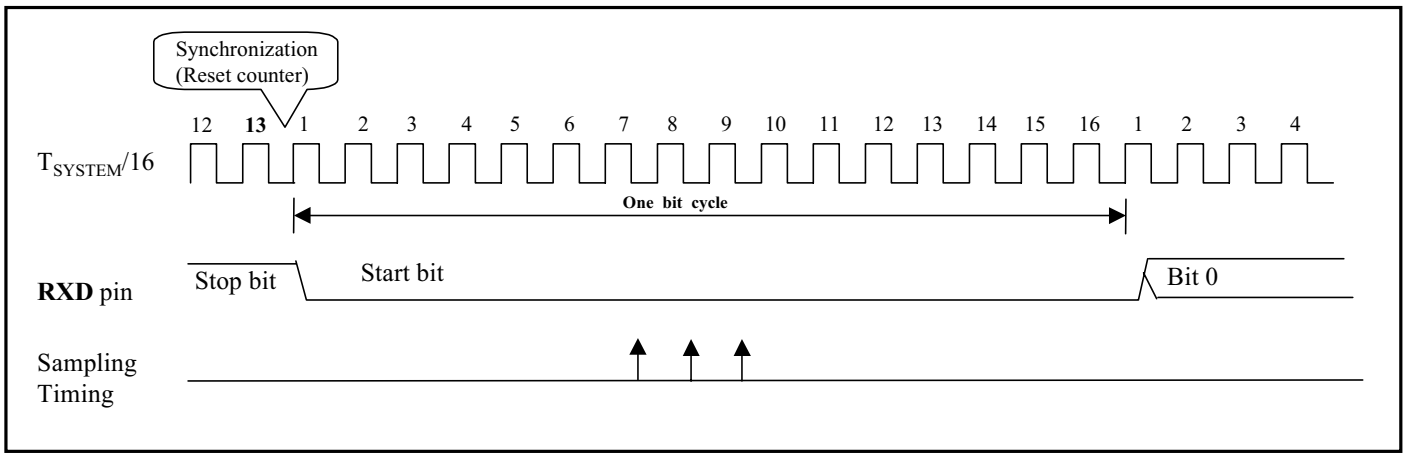
  - **UERRI**: Set to 1 when UART receiving error happened. Clear to 0 by software or UART disable.
  - **UTXI**: Set to 1 when UART transfer buffer empty happened. Clear to 0 by software or UART TX disable(TXE=0)
  - **URXI**: Set to 1 when UART receiver buffer full happened. Clear to 0 by software or UART RX disable(RXE=0)

UART Timing :

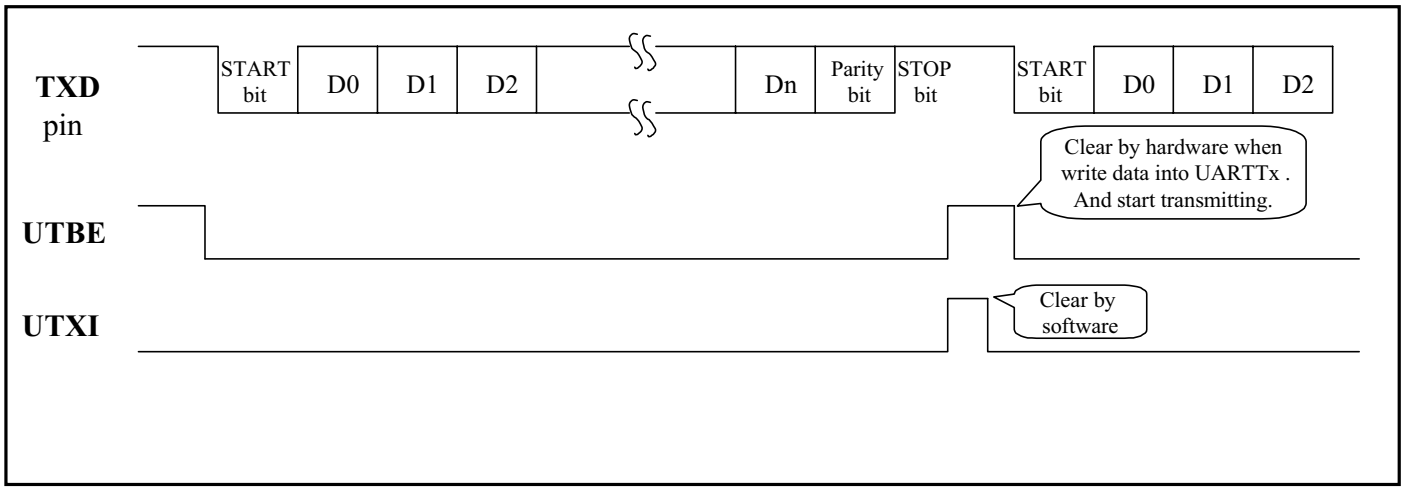
(1) Transmission counter timing:



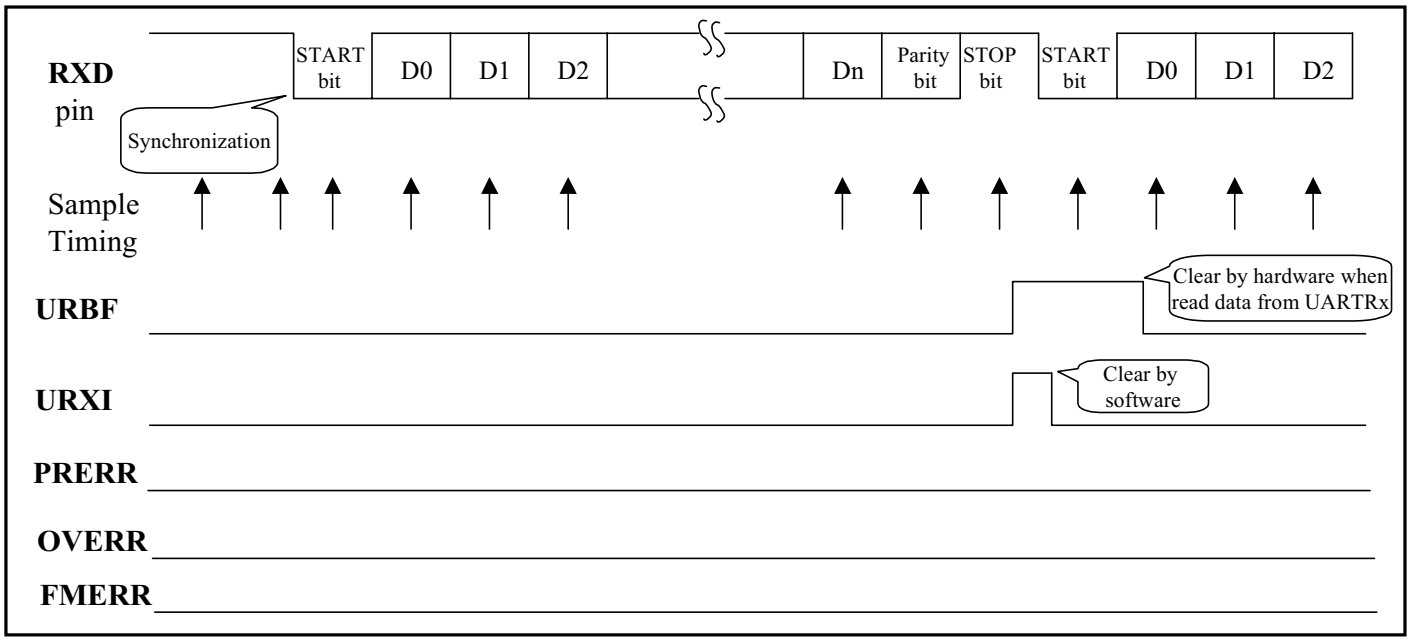
(2) Receiving counter timing:



(3)UART Transmit operation (8 bits data with parity bit):



(4)UART Receive operation (8 bits data with parity and stop bit):

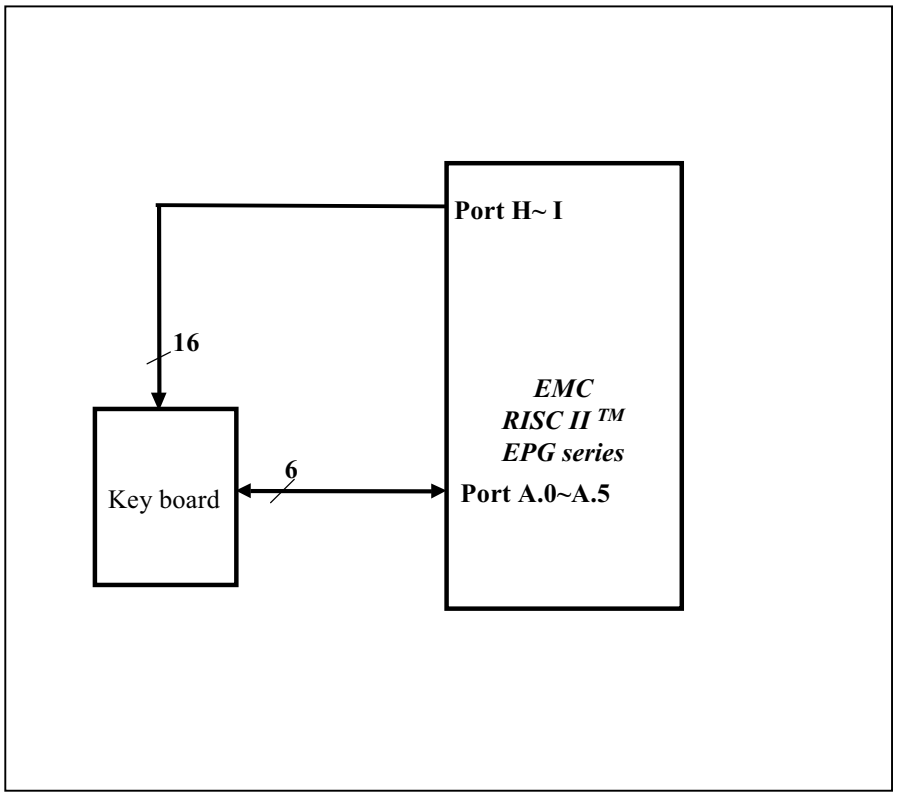


E. Key I/Os

Feature

- 7 pins key input(Port A) and 16 pins key strobe (Port H and Port I) can achieve maximum 112 keys matrix.
- 16 to 1 multiplexing of key strobe pin controlled by **STB3~STB0** bit of STBCON register.
- Interrupt available when Port A low level detected.
- Wakeup available when key input low level detected .

Function Block Diagram



Function Description

7 pins key input(Port A) and 16 pins key strobe (Port H and Port I) can achieve maximum 112 keys matrix.

Port A.0~.6 are input port with pull up resistor controllable.

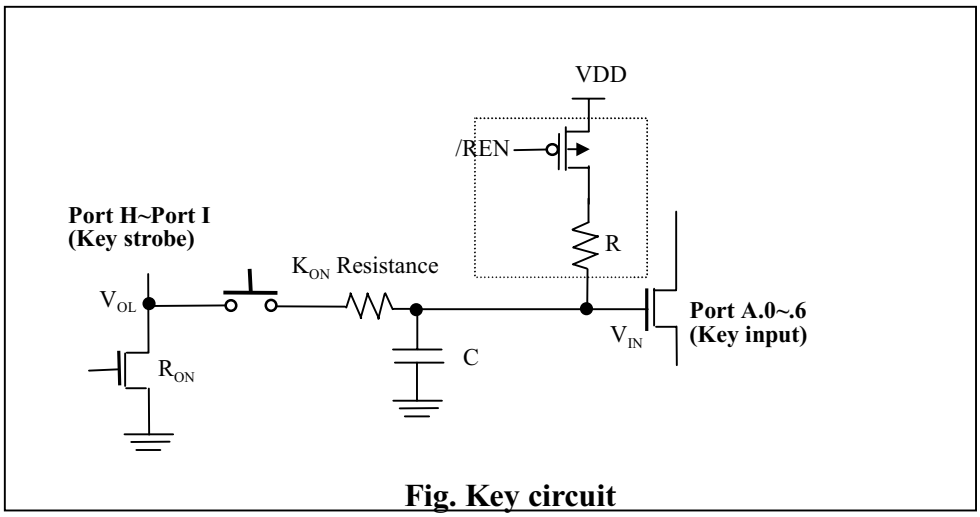


Fig. Key circuit

Port H ~ Port I are designed for 16 bits key strobe or general I/O. When set **BitST** of STBCON register, Port H and Port I become key strobe pin. 16 to 1 multiplexing of Port H and Port I is selected by **STB3~STB0** bit of STBCON register. Only selected pin will output low but the other 15 pins will remain high level. When set **ALL** bit of STBCON register, all strobe pin will output low.

The function table is as following figure.

STBCON			Key Strobe (Port H & Port I)															
BitST	**ALL**	**STB3~0**	PH.0	PH.1	PH.2	PH.3	PH.4	PH.5	PH.6	PH.7	PI.0	PI.1	PI.2	PI.3	PI.4	PI.5	PI.6	PI.7
0	x	x	As general I/O port															
1	0	0000	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0001	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		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	xxxx	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register Description

- **PortA register (Port A register)**

bit 7 bit0

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

  - **Bit6~Bit0** : Key input , input low level interrupt or wake up pin.

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

bit 7 bit0

LVD2	LVD1	LVD0	LVDSEL	Bit7PU	/REN	-	-

  - **/REN:** Port A.0~Port A.6 Pull up resistor control bit.
    0: Enable pull up resistor.
    1: Disable pull up resistor.

- **PAINTEN (Port A INTERRUPT Enable Control Register)**

bit 7 bit0

PA7IE	PA6IE	PA5IE	PA4IE	PA3IE	PA2IE	PA1IE	PA0IE

  - **PA7IE~PA0IE**: Control bit of interrupt .
    0: Disable interrupt function.
    1: Enable interrupt function.

- **PAINTSTA (Port A INTERRUPT STATUS Register)**

bit 7 bit0

PA7I	PA6I	PA5I	PA4I	PA3I	PA2I	PA1I	PA0I

  - **PA7I~PA0I**: INT status of Port A interrupt. Set to 1 when pin **falling edge** detected. Clear to 0 by software.

- **STBCON (Strobe output control Register)**

bit 7 bit0

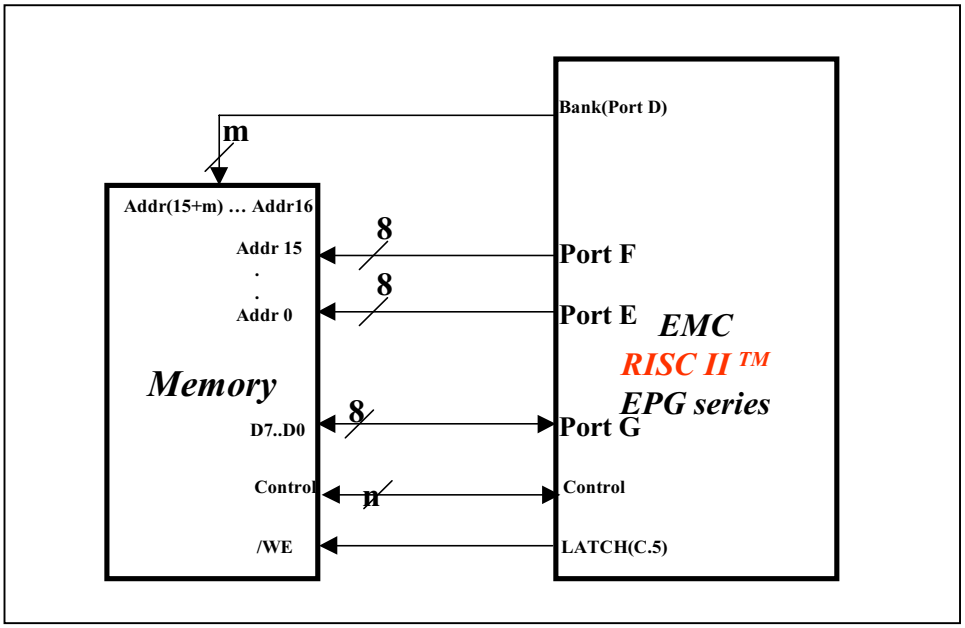
UINVEN	-	BitST	ALL	STB3	STB2	STB1	STB0

  - **STB0~3:** Strobe output selector bit.
  - **ALL:** Set All strobe.
    0 : Bit strobe.
    1 : All strobe.
  - **BitST** : Enable Bit strobe
    0 : Port H and Port I are general I/O port.
    1 : Port H and Port I are key strobe pin. Strobe signal as **STB3~0** defined.

F. External Memory Interface

ELAN RISC II™ series MCU provides parallel memory interface for external memory device. Such as Flash memory, MASK ROM, SRAM…etc. . There are 16 bits address bus and 8 bits data I/O bus . Providing the address auto increase/decrease function to achieve sequential memory access function.

Function Block Diagram



Function Description

Port E and PORT F are designed for external memory address bus, and Port G is for external memory data bus. Set **EMPE** bit of POST_ID register to 1 to enable external memory address post increase or post decrease function. Then set up **EM_ID** bit of POST_ID register to determine post increase or post decrease.

After read data from PORTG register , the PORTF:PORTE register will auto increase or decrease. This function can achieve sequential memory access without effort on changing the address data.

A negative pulse will be generated from PORT C.5(LATCH pin) when **write** to Port G register (refer to LATCH signal timing). This signal can be used as /WE for external memory.

Register Description

- **Post_ID register (Post increase/ Decrease register)**

bit 7 bit0

EM_ID	LCD_ID	FSR1_ID	FSR0_ID	**EMPE**	LCDPE	FSR1PE	FSR0PE

  - **EM_ID:** Set this bit means PORTF:PORTE register auto increase, reset this bit means auto decrease .
  - **EMPE:** Enable External memory address post increase/decrease function when **read data from PORT G.**
    (Only **read data from PORTG register** can active this function)

- **DCRG register(Port G Direction control register)**

bit 7 bit0

-	**LAHEN**	CHNPU	CLNPU	GHNPU	GLNPU	GHNDC	GLNDC

  - **LAHEN:** Enable PortC.5 as external memory data output LATCH signal. (See timing diagram)
    1: Enable.
    0: Disable.
    LATCH signal output is only available when write to PORTG register.

- **PORTE register (Port E register)**
  Low byte of external memory address bus.
  (PORTE register has ability to carry into/borrow from PORTF register.)

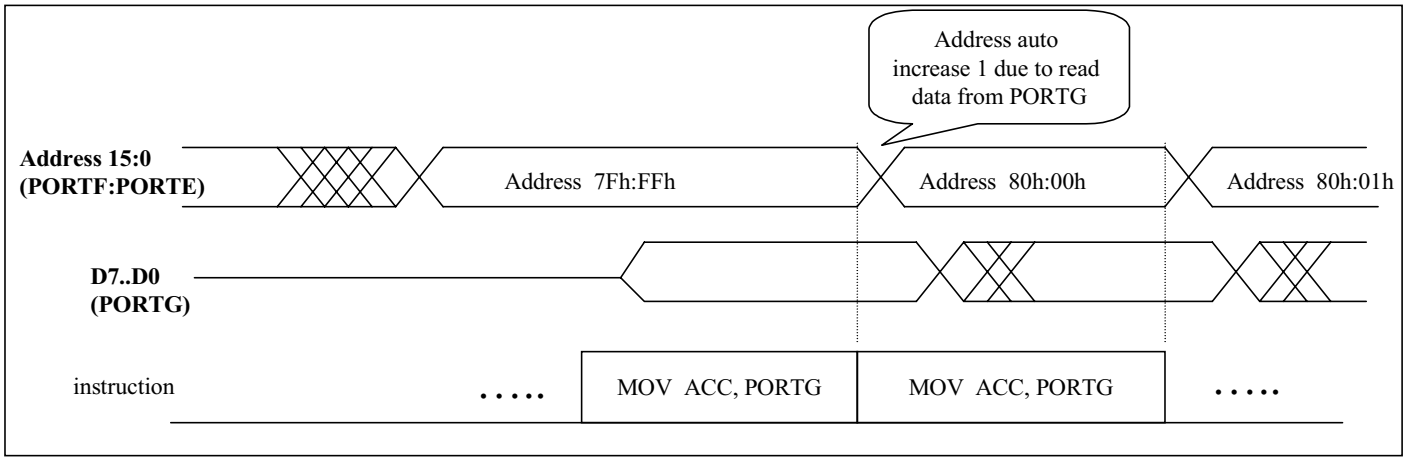
- **PORTF register (Port F register)**
  High byte of external memory address bus.

- **PORTG register (Port G register)**
  Register of external memory data bus.

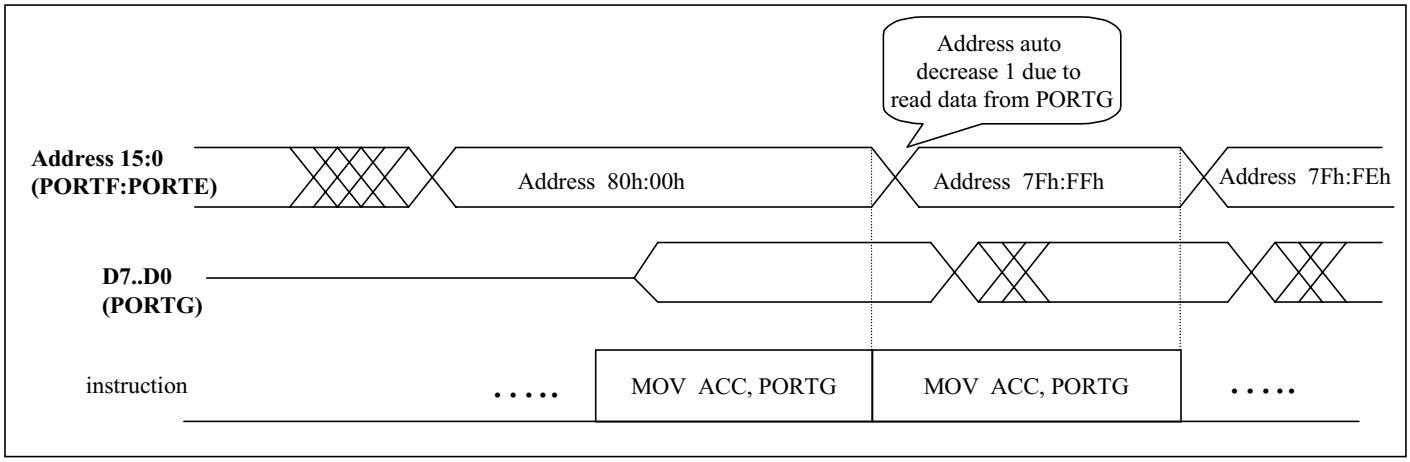
Timing Diagram:

(1)Read mode timing:

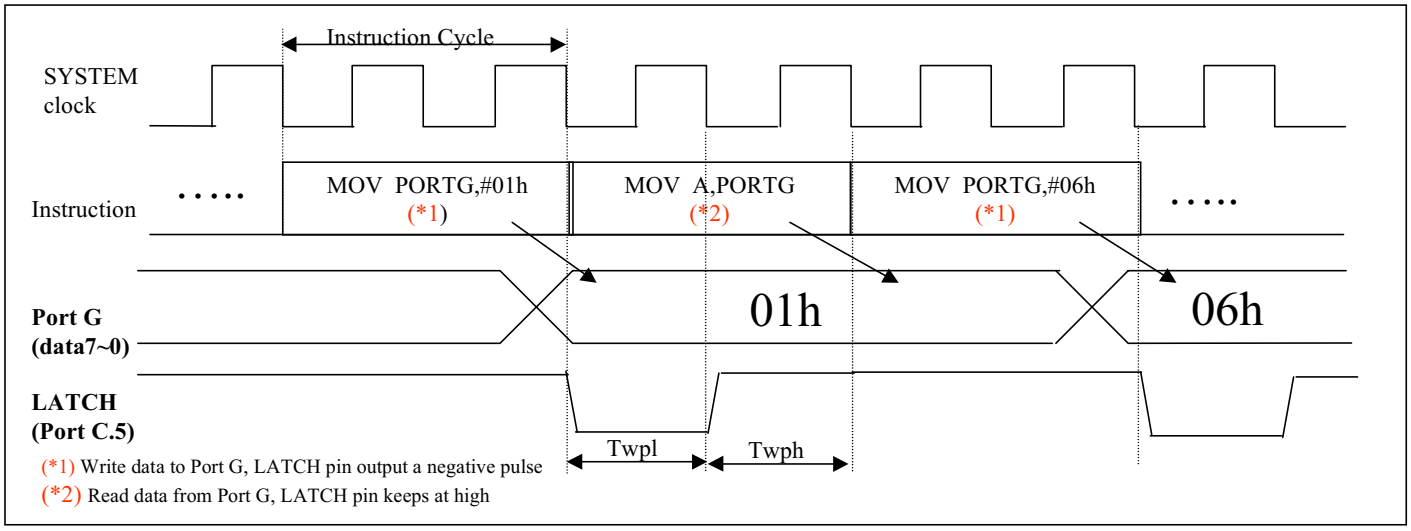
EMPE=1 (enable post increase/decrease function) , EM_ID=1 (set to auto increase)



EMPE=1 (enable post increase/decrease function) , EM_ID=0 (set to address auto decrease)



(2)LATCH signal timing:



(Note: LATCH signal output is only available when write to PORTG register.)

VIII. Electrical characteristic

ABSOLUTE MAXIMUM RATINGS

Items	Sym.	Condition	Limits	Unit
Supply voltage	VDD		-0.3 to +3.6	V
Input voltage	VIN		-0.5 to VDD +0.5	V
Operating temperature range	TOPR		-10 to +60	°C
Storage temperature range	TSTR		-55 to +125	°C

RECOMMENDED OPERATING CONDITIONS

Items	Sym.	Condition	Limits	Unit
Supply voltage	VDD		2.6 to 3.2	V
Input voltage	VIH		VDD*0.9 to VDD	V
	VIL		0 to VDD*0.1	V
LVDIN voltage	V_{LVD}	LVDSEL=1 (Internal LVD)	2.0V~3.6V	V
Operating temperature	TOPR		0 to +40	°C

DC ELECTRICAL CHARACTERISTICS (condition : Ta=0~40°C, VDD= 2.9 +/- 0.3V)

Parameter	Sym.	Condition		Min	Typ	Max	Unit
CLOCK	Fmain	Main-clock frequency		1	-	10	MHz
	Fsub	Sub-clock frequency	RC OSC.	24.6	32.8	41	KHz
			X'tal OSC	-	32.768	-	
Supply current	Idd1	SLEEP mode	VDD=3V, no load	-	-	1	µA
	Idd2	IDLE mode	VDD=3V, RC OSC., no load	-	8	12	
	Idd3		VDD=3V, X'tal OSC., no load	-	5	8	
	Idd4	SLOW mode	VDD=3V, RC/X'tal OSC ,no load	-	20	30	
	Idd5	FAST mode	VDD=3V, Fmain=4MHz, ,no load	-	750	900	
	Idd6		VDD=3V, Fmain=10MHz, ,no load	-	1400	1600	
Input voltage	VIH	PA[0:7],PB[0:7] ~ PI[0:7](as general input port)		VDD×0.7	-	VDD	V
	VIL			0	-	VDD×0.3	
Input threshold voltage (Schmitt)	VT+	Port A interrupt , SYSTEM RESET and RESET_KEY		0.5×VDD	-	0.75×VDD	V
	VT-			0.2×VDD	-	0.4×VDD	
Output current	IOH1	PB[0:3];PC[0:7] ~ PG[0:7]	VDD=3V , VOH=2.4V	-1	-2	-3	mA
	IOL1		VDD=3V , VOL=0.2V	+1	+2	+3	
	IOH2	PB[6:7]	VDD=3V , VOH=2.5V	-1	-2	-3	
	IOL2		VDD=3V , VOL=0.5V	+1	+2	+3	
	IOH3	PB[4:5]	**VDD=3V, VOH=1V**	-3	-6	-9	
	IOL3		**VDD=3V, VOL=0.5V**	+1.5	+3	+4.5	
	IOH4	PH[0:7] ~ PI[0:7]	VDD=3V , VOH=2.4V	-5	-10	-15	µA
	IOL4		VDD=3V , VOL=0.2V	+0.50	+1	+1.5	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[0:7],PB[0:7] ~ PI[0:7]	Vin=GND	500	1000	1500	KΩ
	RPU2	SYSTEM RESET , RESET_KEY	Vin=GND	200	400	800	
Small Pull up resistance	RPU3	PA[0:7],PB[0:7] ~ PI[0:7]	Vin=2V	50	100	200	KΩ
	RPU4	SYSTEM RESET , RESET_KEY	Vin=2V	50	100	200	
Large Pull down resistance	RPD1	TEST	Vin=VDD	250	500	750	KΩ
Small Pull down resistance	RPD2	TEST	Vin=1V	10	20	30	KΩ
Low battery detect voltage	Vdet	Ta=0 to 40°C		Vdet -4%	Vdet (*1)	Vdet +4%	V
LVD supply current	Ilvd	LVDIN=2V		-	3	5	µA
Data retention voltage	Vret			1.6	-	-	V
Power ON reset voltage	Vpor			1.4	1.5	1.6	V

*1 : Typical detected voltage is chosen by software from the following.

LVD2~0	Vdet(Vth-)	Vdet(Vth+)	Deviation	LVD2~0	Vdet(Vth-)	Vdet(Vth+)	Deviation
000	2.2V	2.3V	±0.1V	100	2.6V	2.7V	±0.1V
001	2.3V	2.4V		101	2.7V	2.8V	
010	2.4V	2.5V		110	2.8V	2.9V	
011	2.5V	2.6V		111	2.9V	3.0V	

AC ELECTRICAL CHARACTERISTICS (condition : Ta= 0~40 °C, VDD= 2.9 +/- 0.3V)

Parameter	Sym.	Condition	Min	Typ	Max	Unit
LVD setup time	Tstup1	LVDEN ↑ to /LV ↓ (LVDIN=2.0V)	-	30	100	µs
SDI data setup time	Tstup2	Setup time of SDI data input to SCK ↑ or SCK ↓	-	25	50	ns
SDI data hold time	Thold2	Hold time of SDI data input to SCK ↓ or SCK ↑	-	25	50	ns
SDO output valid time	Tvalid1	SCK ↑ or SCK ↓ to SDO data output	-	25	50	ns
SCK input high time	Tsckh	Slave mode(Fmain=10 MHz)	200	-	-	ns
SCK input low time	Tsckl	Slave mode(Fmain=10 MHz)	200	-	-	ns
Slave mode setup time	Tsetup3	/SS ↓ to SCK ↑ or SCK ↓ (Fmain=10 MHz)	400	-	-	ns
Slave mode unselect delay time	Tdelay1	/SS ↑ to SDO output hi-impedance delay time	-	25	50	ns
PF[7:0]:PE[7:0] auto increase/ decrease delay time	Tdelay2	Q4 end of read Port G instruction to Addr[15:0] auto increase/ decrease (Fmain=10 MHz)	-	50	100	ns
CK delay time	Tdelay3	CHOP carrier stop to CK ↑ delay time	22.9	30.5	38.2	µs
LATCH write pulse low time	Twpl	Q4 end of read/write Port G instruction to LATCH ↑ (Fmain=10 MHz)	-	100	-	ns
LATCH write pulse high time	Twph	LATCH ↑ to Q4 end of current instruction(Fmain=10 MHz)	-	100	-	ns
Instruction cycle time	Tcycle	Fmain=1MHz	-	2(*1)	-	µs
		Fmain=4MHz	-	0.5(*1)	-	
		Fmain=10MHz	-	0.2(*1)	-	

*1: Instruction cycle time= 2 * System clock time

Figure VIII-1: Low voltage detection timing

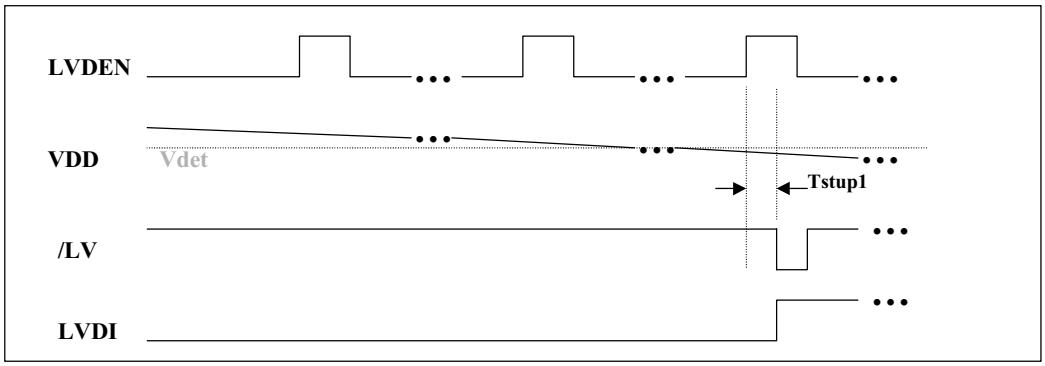


Figure VIII-2: EL timing

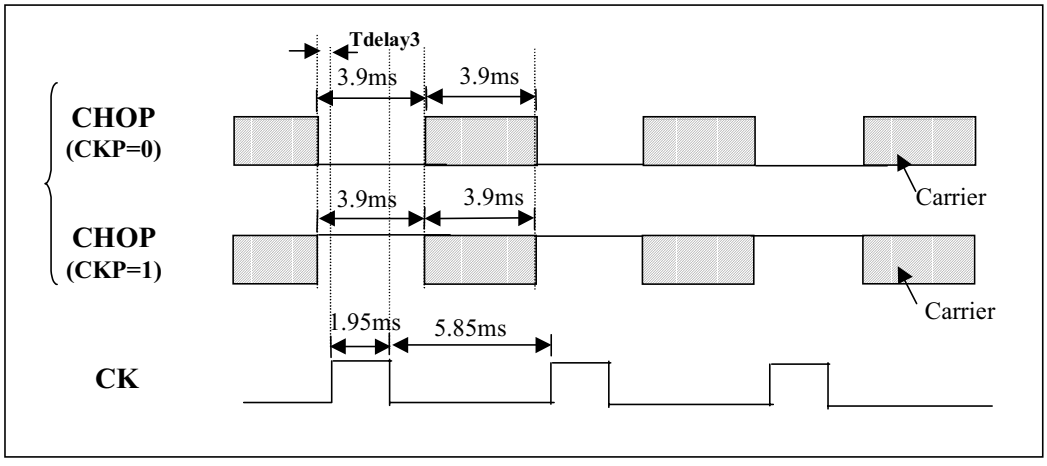


Figure VIII-3: SPI master mode timing

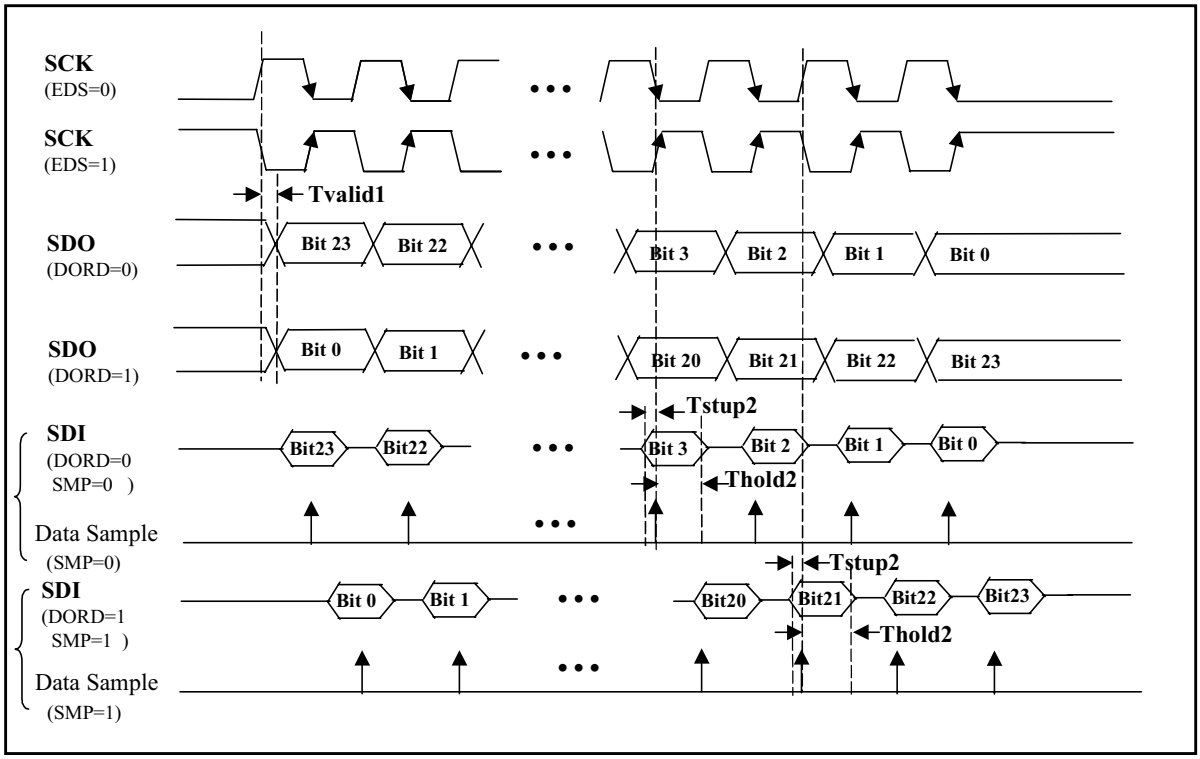


Figure VIII-4: SPI slave mode timing

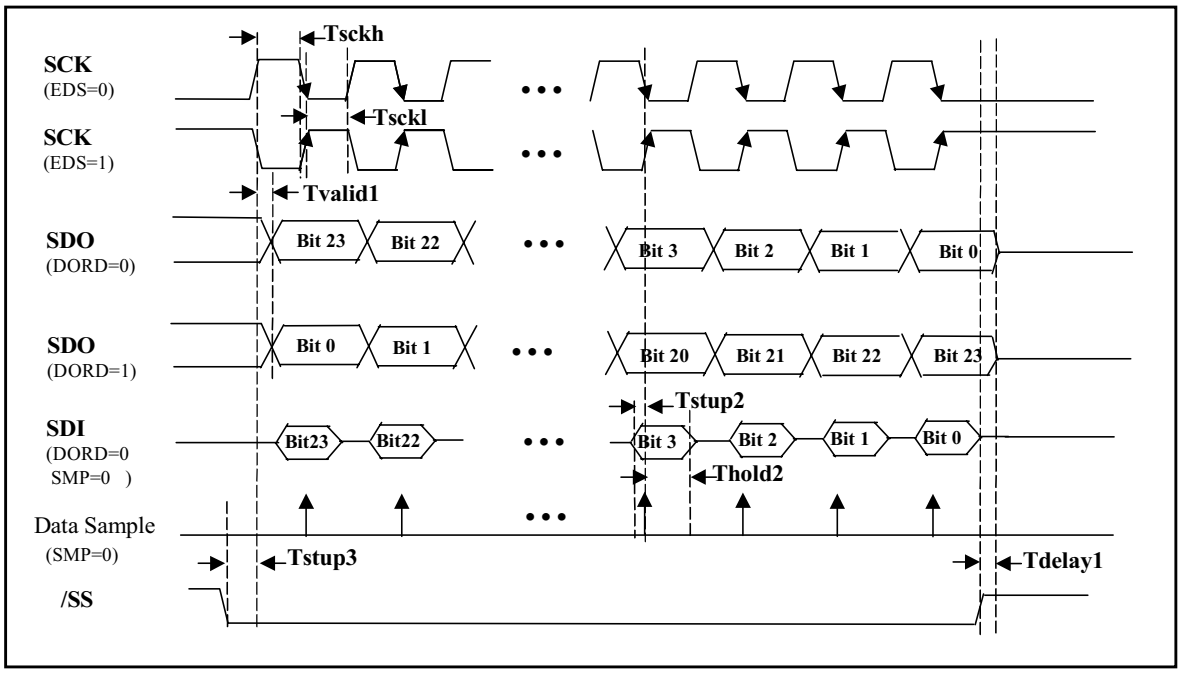


Figure VIII-5: Memory interface timing (Read mode)

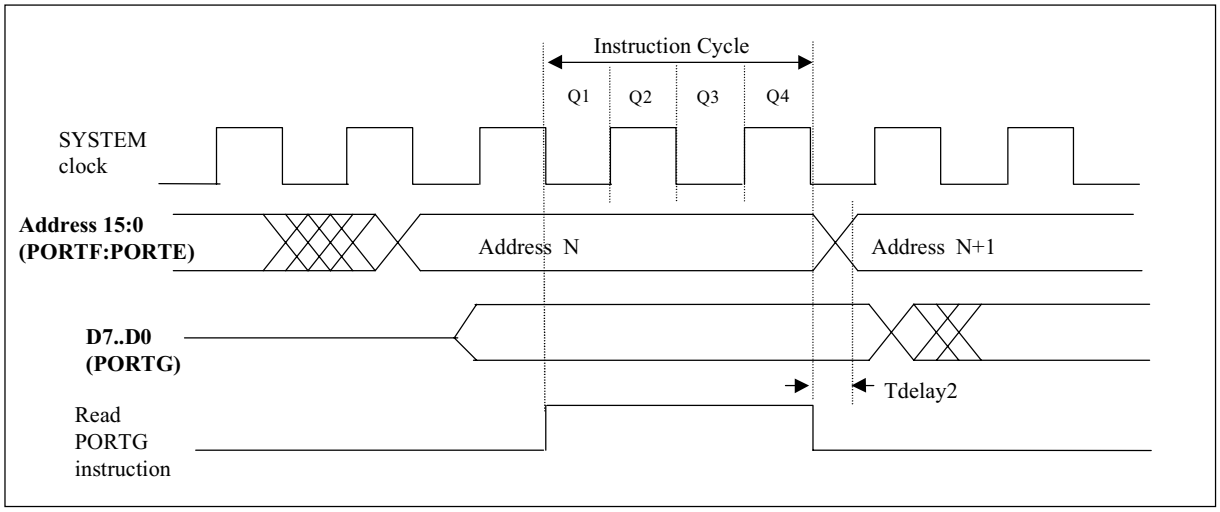
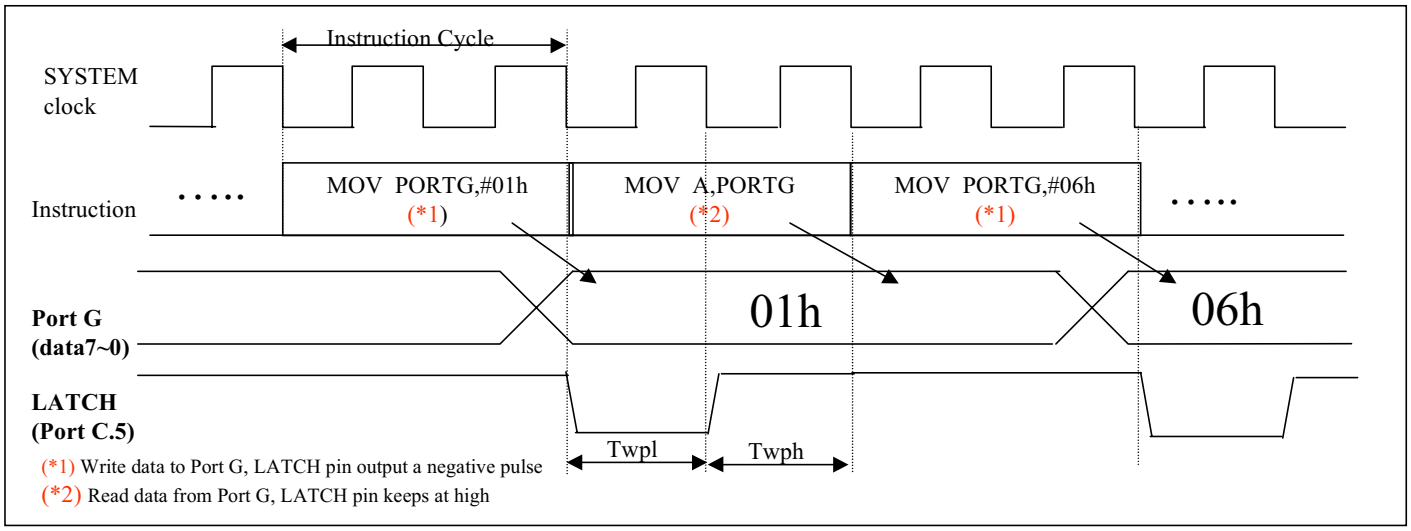
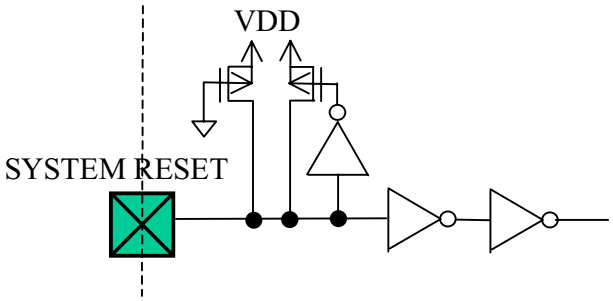


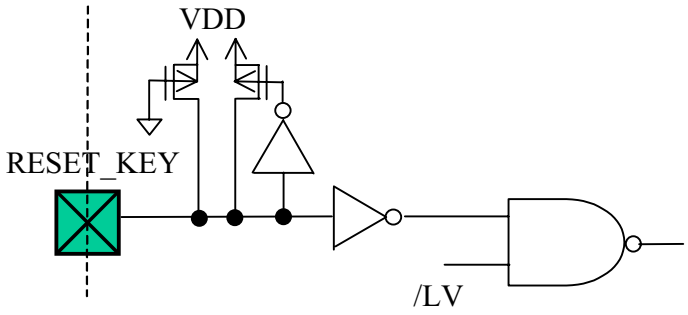
Figure VIII-6: LATCH signal timing



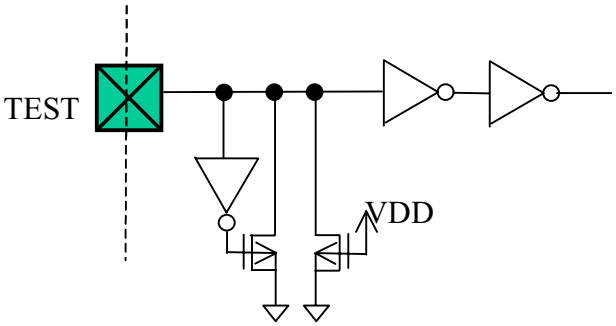
IX. Pin Type Circuit Diagrams

IX.1 Reset Pin Type

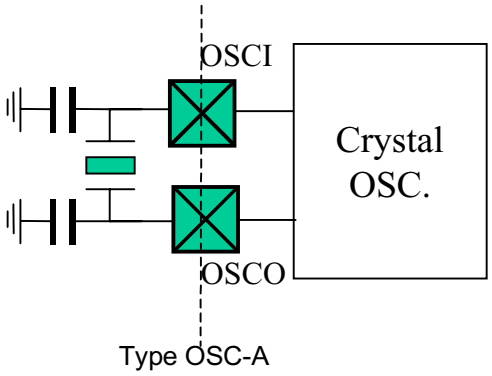


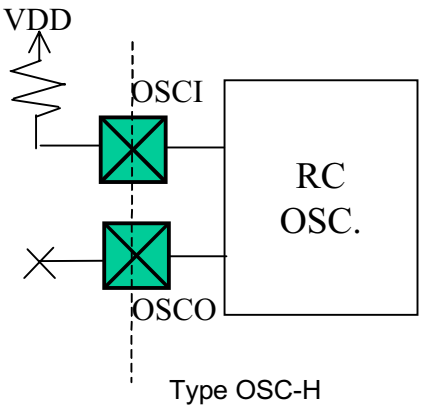


IX.2 TEST Pin Type

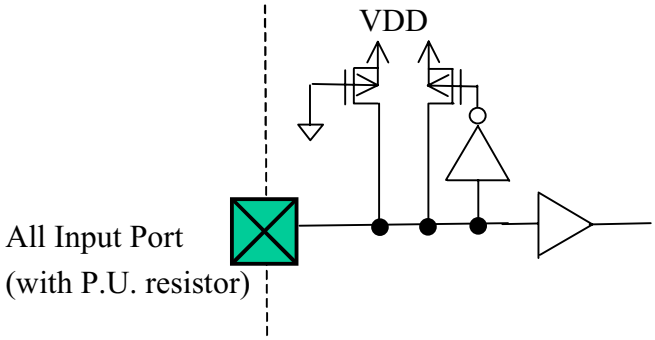


IX.3 Oscillator Pin Type

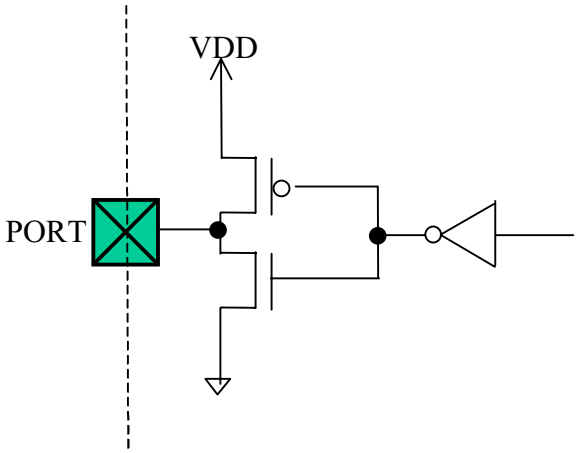




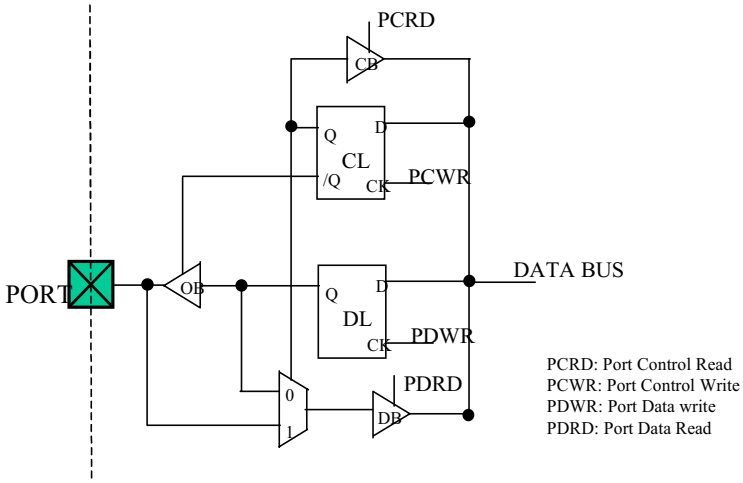
IX.4 Input Pin Type



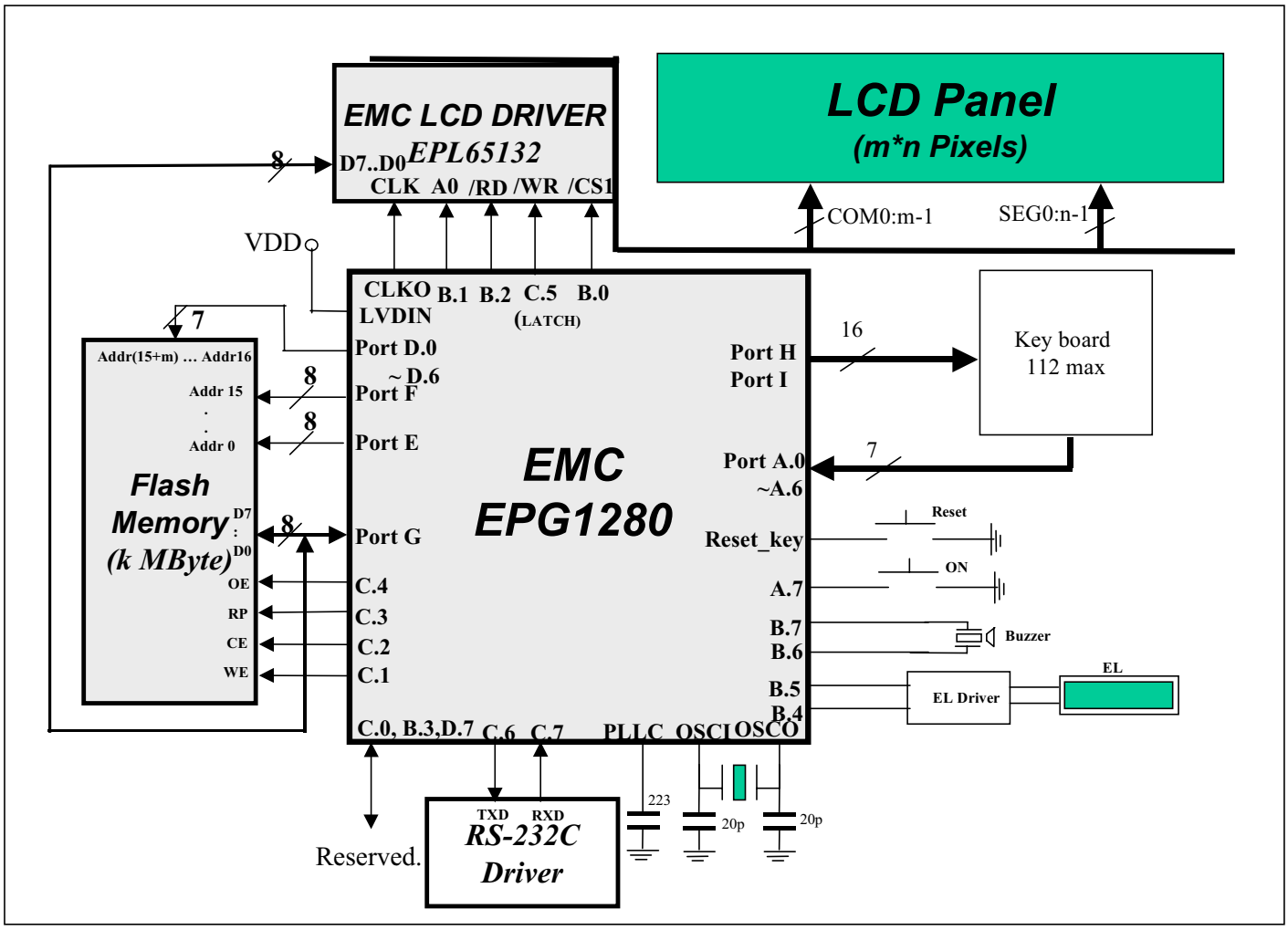
IX.5 Output Pin Type



IX.6 General I/O Pin Type



X. Application Circuit



XI. Instruction Set

Remark:

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

Type	Instruction Binary	Mnemonic	Operation	Status Affected	Cycles
SYSTEM CONTROL	0000 0000 0000 0000	NOP	No operation	None	1
	0000 0000 0000 0001	WDTC	WDT ← 0 /TO←1,/PD←1	None	1
	0010 1011 1111 1110	RET	PC ← (Top of Stack)	None	1
	0010 1011 1111 1111	RETI	PC ← (Top of Stack); Enable Interrupt	None	1
	0000 0000 0000 0010	SLEP	Enter IDLE MODE if MS1=1 Enter SLEEP MODE if MS1=0	None	1
	0100 0011 kkkk kkkk	BANK #k	BSR← k	None	1
TABLE LOOK UP	0100 0000 kkkk kkkk	TBPTL #k	TABPTRL ← k	None	1
	0100 0001 kkkk kkkk	TBPTM #k	TABPTRM ← k	None	1
	0100 0010 kkkk kkkk	TBPTH #k	TABPTRH ← k	None	1
	0010 11ii rrrr rrrr	TBRD i,r	r←ROM[(TABPTR)]; (*1)(*2)	None	1 or 2 (*3)
	0010 1111 rrrr rrrr	TBRD A,r	r←ROM[(TABPTR+ACC)]; (*2)	None	1 or 2 (*3)
LOGIC	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
	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

LOGIC	0000 1101 rrrr rrrr	RLC r	r(n+1) ← r(n); C ← r(7) ; r(0) ← C	C	1
	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
Compare Branch	0101 1bbb rrrr rrrr aaaa aaaa aaaa aaaa	JBC r,b,addr	If r(b)=0,jump to addr PC[15:0] ← addr (*4)	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 (*4)	None	2
	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 (*4)	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 (*4)	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 (*4)	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 (*4)	None	2
	0100 0111 kkkk kkkk aaaa aaaa aaaa aaaa	JGE A,#k,addr	Jump to addr if A≧k PC[15:0] ← addr (*4)	None	2
	0100 1000 kkkk kkkk aaaa aaaa aaaa aaaa	JLE A,#k,addr	Jump to addr if A≦k PC[15:0] ← addr (*4)	None	2
	0100 1001 kkkk kkkk aaaa aaaa aaaa aaaa	JE A,#k,addr	Jump to addr if A=k PC[15:0] ← addr (*4)	None	2
	0101 0101 rrrr rrrr aaaa aaaa aaaa aaaa	JGE A,r,addr	Jump to addr if A≧r PC[15:0] ← addr (*4)	None	2
	0101 0110 rrrr rrrr aaaa aaaa aaaa aaaa	JLE A,r,addr	Jump to addr if A≦r PC[15:0] ← addr (*4)	None	2
	0101 0111 rrrr rrrr aaaa aaaa aaaa aaaa	JE A,r,addr	Jump to addr if A=r PC[15:0] ← addr (*4)	None	2
PROCESS	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
	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
	0010 0100 rrrr rrrr	CLR r	r ← 0	Z	1
	0010 0101 rrrr rrrr	TEST r	Z ← 0 if r<>0; Z ← 1 if r=0	Z	1

PROCESS	0010 0111 rrrr rrrr	RPT r	Single repeat *(r) times on next instruction *(r) is the content of register r	None	1
ARITH-METIC	0001 0000 rrrr rrrr	ADD A,r	A ← A+r	C,DC,Z, OV,SGE, SLE	1
	0001 0001 rrrr rrrr	ADD r,A	r ← r+A (*5)	C,DC,Z, OV,SGE, SLE	1
	0100 1010 kkkk kkkk	ADD A,#k	A ← A+k	C,DC,Z, OV,SGE, SLE	1
	0001 0010 rrrr rrrr	ADC A,r	A ← A+r+C	C,DC,Z, OV,SGE, SLE	1
	0001 0011 rrrr rrrr	ADC r,A	r ← r+A+C	C,DC,Z, OV,SGE, SLE	1
	0100 1011 kkkk kkkk	ADC A,#k	A ← A+k+C	C,DC,Z, OV,SGE, SLE	1
	0001 0110 rrrr rrrr	SUB A,r	A ← r-A	C,DC,Z, OV,SGE, SLE	1
	0001 0111 rrrr rrrr	SUB r,A	r ← r-A	C,DC,Z, OV,SGE, SLE	1
	0100 1100 kkkk kkkk	SUB A,#k	A ← k-A	C,DC,Z, OV,SGE, SLE	1
	0001 1000 rrrr rrrr	SUBB A,r	A ← r-A-/C	C,DC,Z, OV,SGE, SLE	1
	0001 1001 rrrr rrrr	SUBB r,A	r ← r-A-/C	C,DC,Z, OV,SGE, SLE	1
	0100 1101 kkkk kkkk	SUBB A,#k	A ← k-A-/C	C,DC,Z, OV,SGE, SLE	1
	0001 0100 rrrr rrrr	ADDDC A,r	A ← (Decimal ADD) A+r+C	C,DC,Z	1
	0001 0101 rrrr rrrr	ADDDC r,A	r ← (Decimal ADD) r+A+C	C,DC,Z	1
	0001 1010 rrrr rrrr	SUBDB A,r	A ← (Decimal SUB) r-A-/C	C,DC,Z	1
	0001 1011 rrrr rrrr	SUBDB r,A	r ← (Decimal SUB) r-A-/C	C,DC,Z	1

	0010 0110 rrrr rrrr	MUL A,r	PRODH:PRODL ← A*r	None	1
ARITH-METIC	0100 1111 kkkk kkkk	MUL A,#k	PRODH:PRODL ← A*k	None	1
	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 1110 rrrr rrrr	DECA r	A ← r-1	C,Z	1
	0001 1111 rrrr rrrr	DEC r	r← r-1	C,Z	1
Move	0010 0000 rrrr rrrr	MOV A,r	A←r	Z	1
	0010 0001 rrrr rrrr	MOV r,A	r←A	None	1
	100p pppp rrrr rrrr	MOVRP p,r	Register p ← Register r	None	1
	101p pppp rrrr rrrr	MOVPR r,p	Register r ← Register p	None	1
	0100 1110 kkkk kkkk	MOV A,#k	A← k	None	1
Branch	110a aaaa aaaa aaaa	SJMP addr	PC ← addr PC[13..16] unchange	None	1
	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] unchange	None	1
	0000 0000 0010 aaaa aaaa aaaa aaaa aaaa	LJMP addr (2 words)	PC ← addr	None	2
	0000 0000 0011 aaaa aaaa aaaa aaaa aaaa	LCALL addr (2 words)	[Top of Stack]←PC+1 PC ← addr	None	2

(*1) TBRD i,r :

r← ROM[(TABPTR)];

i=00: TABPTR not change

i=01: TABPTR←TABPTR+1

i=10: TABPTR←TABPTR-1

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

*Bit 7 of TABPTRH use to select internal ROM space or external memory space.

Bit7=0: internal ROM space

Bit7=1:external memory space.

*Bit 0 of TABPTRL use to select low byte or high byte of pointed ROM data.

Bit0=0: Low byte of pointed ROM data

Bit0=1:High byte of pointed ROM data.

*The maximum table look up space is internal 8Mbytes and external 8Mbytes.

(*3) **TBRD** instruction will take 2 cycles at first time, and then 1 cycle at following **TBRD** instructions.

Ex1. TBRD 1,reg1 ➔ **Take 2 cycle**
TBRD 1,reg2 ➔ Take 1 cycle
TBRD 1,reg3 ➔ Take 1 cycle
……..
TBRD 1,reg99 ➔ Take 1 cycle
NOP
TBRD 1,reg100 ➔ **Take 2 cycles**
TBRD 1,reg101 ➔ Take 1 cycle

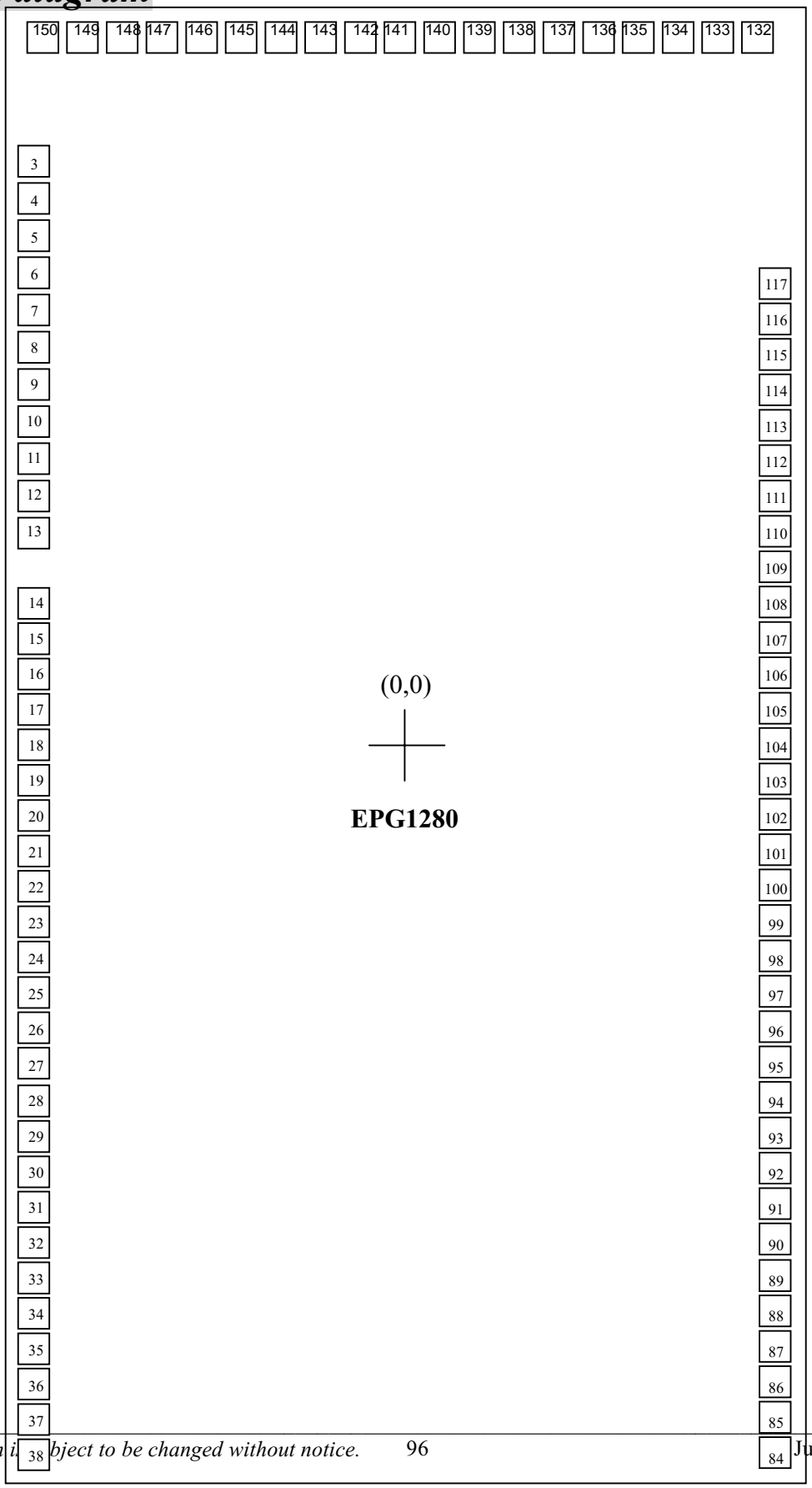
Ex2. MOV A,#10
RPT ACC
TBRD 1,INDF1 ➔ Repeat 10 times, total take 11 cycles

(*4)The maximum jump range is 64K absolute address, means only can jump within the same 64K range, i.e. 0~64K or 64K~128K.

(*5)Carry bit of ADD PCL,A or ADD TABPTRL,A will automatic carry into PCH or TABPTR. The Instruction cycle of write to PC(program counter) takes TWO cycle.

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

XII. Pad diagram



Pin NO.	Symbol	X	Y	Pin NO.	Symbol	X	Y
1	NC			41	NC		
2	NC			42	NC		
3	LVDIN	-1785.0	1793.1	43	NC		
4	(reserved)	-1785.0	1661.1	44	NC		
5	(reserved)	-1785.0	1541.1	45	NC		
6	(reserved)	-1785.0	1421.1	46	NC		
7	(reserved)	-1785.0	1301.1	47	NC		
8	(reserved)	-1785.0	1181.1	48	NC		
9	AVDD	-1785.0	1039.1	49	NC		
10	PLLC	-1785.0	919.1	50	NC		
11	VSS	-1785.0	799.1	51	NC		
12	VSS	-1785.0	667.1	52	NC		
13	OSCI	-1785.0	535.1	53	NC		
14	OSCO	-1785.0	224.2	54	NC		
15	TEST	-1785.0	104.2	55	NC		
16	CLKO	-1785.0	-15.8	56	NC		
17	PA0	-1785.0	-135.8	57	NC		
18	PA1	-1785.0	-255.8	58	NC		
19	PA2	-1785.0	-375.8	59	NC		
20	PA3	-1785.0	-495.8	60	NC		
21	PA4	-1785.0	-615.8	61	NC		
22	PA5	-1785.0	-735.8	62	NC		
23	PA6	-1785.0	-855.8	63	NC		
24	PA7	-1785.0	-975.8	64	NC		
25	PB0	-1785.0	-1095.8	65	NC		
26	PB1	-1785.0	-1215.8	66	NC		
27	PB2	-1785.0	-1335.8	67	NC		
28	PB3	-1785.0	-1455.8	68	NC		
29	PB4	-1785.0	-1575.8	69	NC		
30	PB5	-1785.0	-1695.8	70	NC		
31	PB6	-1785.0	-1815.8	71	NC		
32	PB7	-1785.0	-1935.8	72	NC		
33	PC0	-1785.0	-2055.8	73	NC		
34	PC1	-1785.0	-2175.8	74	NC		
35	PC2	-1785.0	-2295.8	75	NC		
36	PC3	-1785.0	-2415.8	76	NC		
37	PC4	-1785.0	-2535.8	77	NC		
38	PC5	-1785.0	-2655.8	78	NC		
39	NC			79	NC		
40	NC			80	NC		

Pin NO.	Symbol	X	Y	Pin NO.	Symbol	X	Y
81	NC			121	NC		
82	NC			122	NC		
83	NC			123	NC		
84	PC6	1778.5	-2655.8	124	NC		
85	PC7	1778.5	-2535.9	125	NC		
86	PD7	1778.5	-2415.9	126	NC		
87	PD6	1778.5	-2295.8	127	NC		
88	PD5	1778.5	-2175.8	128	NC		
89	PD4	1778.5	-2055.8	129	NC		
90	PD3	1778.5	-1935.9	130	NC		
91	PD2	1778.5	-1815.9	131	NC		
92	PD1	1778.5	-1695.8	132	PH0	1749.3	2683.1
93	PD0	1778.5	-1575.9	133	PH1	1551.0	2683.1
94	PF7	1778.5	-1455.9	134	PH2	1362.4	2683.1
95	PF6	1778.5	-1335.8	135	PH3	1164.1	2683.1
96	PF5	1778.5	-1215.8	136	PH4	975.3	2683.1
97	PF4	1778.5	-1095.8	137	PH5	777.1	2683.1
98	PF3	1778.5	-975.9	138	PH6	588.4	2683.1
99	PF2	1778.5	-855.8	139	PH7	390.1	2683.1
100	PF1	1778.5	-735.9	140	PI0	200.2	2683.1
101	PF0	1778.5	-615.8	141	PI1	1.9	2683.1
102	PE7	1778.5	-495.8	142	PI2	-186.7	2683.1
103	PE6	1778.5	-375.9	143	PI3	-385.1	2683.1
104	PE5	1778.5	-255.8	144	PI4	-573.8	2683.1
105	PE4	1778.5	-135.9	145	PI5	-772.0	2683.1
106	PE3	1778.5	-15.8	146	PI6	-960.8	2683.1
107	PE2	1778.5	104.2	147	PI7	-1159.0	2683.1
108	PE1	1778.5	224.1	148	RESET KEY	-1347.7	2683.1
109	PE0	1778.5	344.2	149	SYSTEM RESET	-1546.0	2683.1
110	PG0	1778.5	464.1	150	VDD	-1746.3	2683.1
111	PG1	1778.5	584.1	151	NC		
112	PG2	1778.5	704.2	152	NC		
113	PG3	1778.5	824.2	153	NC		
114	PG4	1778.5	944.1	154	NC		
115	PG5	1778.5	1064.2	155	NC		
116	PG6	1778.5	1184.1	156	NC		
117	PG7	1778.5	1304.2	157	NC		
118	NC			158	NC		
119	NC			159	NC		
120	NC			160	NC		

Note: Pad 4 ~ Pad 8 is for factory test only. Please do NOT bonding these pads on

application circuit.

Chip size : 3860 * 5660 um

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