



芯睿科技股份有限公司  
mikkon technology limited

Total page

158

Doc No

MK9A160P

Rev.

1.08

文件名稱 MK9A160P 8 Bit Microcontroller Data Sheet

版次	生效日	ECN No.	制修訂者	修訂內容概要
1.00	2009.11.20		李崑旭	新頒。
1.01	2009.11.24		Jemmy	P8. modify PIN assignment
1.02	2009.1.7		Jemmy	P116~119 : modify STROBE & example P109 & P116 : PAD_CTL6 & PAD_CTL7 P25 PC_CTL[4:3] → PC_CTL[7:0] P8,P11,P117 COM16/PF[7]/CP2- → COM16/PF[7]/CP3- COM15/PF[6]/CP2- → COM15/PF[7]/CP3+ COM14/PF[5] → COM14/PF[7]/CP2- COM13/PF[4] → COM13/PF[7]/CP2+
1.03	2009.2.29		Jemmy	P120. Add OPx2
1.04	2010.3.16		Jemmy	P116. Add Port B
1.05	2010.8.1		Jemmy	P143 Add 1/4 bias LCD
1.06	2010.8.23		Jemmy	P155 2.40 < LVD < 2.75
1.07	2010.10.13		Jemmy	P6 LVR is fixed 2.1V P modify IRQF
1.08	2011.1.5		Jemmy	P38 Bank1~16 → Bank0~15 P36 modify Fig1.9.1 P155 2.40<LVD<2.80

# ***MK9A160P***

***(Low power 8bit Microcontroller)***

## ***USER'S MANUAL***

## Table of Content

<b>1. Product Overview</b> .....	5
<b>1.1 Feature</b> .....	5
<b>1.2 Block Diagram</b> .....	7
<b>1.3 Pin Assignment ( Pin assignment for COB )</b> .....	8
<b>1.4 Pin Description</b> .....	10
<b>2. System Architecture</b> .....	15
<b>2.1 System Clock</b> .....	15
<b>2.1.1 Fast Clock (FCLK) Connection</b> .....	17
<b>2.1.2 Slow Clock (SCLK) Connection</b> .....	18
<b>2.1.3 FCLK &amp; SCLK Switch</b> .....	20
<b>2.2 Program Memory (ROM)</b> .....	21
<b>2.3 Data Memory (RAM)</b> .....	22
<b>2.4 Configuration Register</b> .....	23
<b>2.5 Special Purpose Register</b> .....	26
<b>2.7 SLEEP Function</b> .....	33
<b>2.8 Table Look-up Function</b> .....	34
<b>2.9 FSR: Bank Select Register</b> .....	35
<b>2.10 WBANK : RAM Bank control Register</b> .....	37
<b>2.11 Status Register</b> .....	38
<b>2.12 PCH &amp; PCL:</b> .....	40
<b>2.13 Reset</b> .....	43
<b>3. Timer and Capture</b> .....	48
<b>3.1 16 bit Pre-divider</b> .....	48
<b>3.2 Timer 0 (TM0)</b> .....	48
<b>3.3 FREQ and Tone</b> .....	50
<b>3.4 Timer 2&amp;3 (TM2 &amp; TM3)</b> .....	58
<b>3.4.1 Timer 2</b> .....	58
<b>3.4.2 Timer 3 (TM3)</b> .....	81
<b>3.5 Timer 4&amp;5 (TM4 &amp; TM5)</b> .....	91
<b>3.5.1 Timer 4</b> .....	91
<b>3.5.2 Timer 5 (TM5)</b> .....	95
<b>3.6 Watchdog Timer (WDT)</b> .....	98
<b>4. I/O Port and Other Control Function</b> .....	99
<b>4.1 /O port</b> .....	99
<b>4.1.1. Port A</b> .....	100

---

4.1.2	Port C.....	103
4.1.3	Port D.....	107
4.1.4	Port E.....	109
4.1.5	Port F.....	110
4.1.6	Port B.....	111
4.2	Define Shared Pin.....	112
4.3	Key Strobe Function.....	120
4.4	Interrupt & halt release .....	124
4.5	External Interrupt Pin -- PA[0~7] ,PC[0~7] & PD[5].....	128
4.6	Resister to Frequency Converter (RFC).....	129
4.6.1	Timer 2 act as 8 bit Capture .....	130
4.6.2	Timer 2 act as RFC counter.....	130
4.7	EL Panel Driver .....	132
4.8	Low Voltage Reset (LVR).....	133
4.9	Low Voltage Detect (LVD).....	133
4.10	SPI.....	134
4.11	Other Register .....	138
5.	LCD Driver .....	142
5.1	LCD Pad Connections (1/3 bias ) .....	142
5.2	LCD Attribute Setting .....	143
5.2.1	Bias Setting.....	143
5.2.2	Duty (COM) and Frame Frequency Setting .....	143
5.3	LCD Display RAM Mapping.....	147
6.	Typical Application Circuit.....	149
7.	Instruction Table .....	150
8.	Electrical Characteristis .....	154
8.1	Absolute Maximum Ratings .....	154
8.2	DC Characteristics .....	154
8.3	AC Characteristics .....	157
8.4	External RC Table.....	157

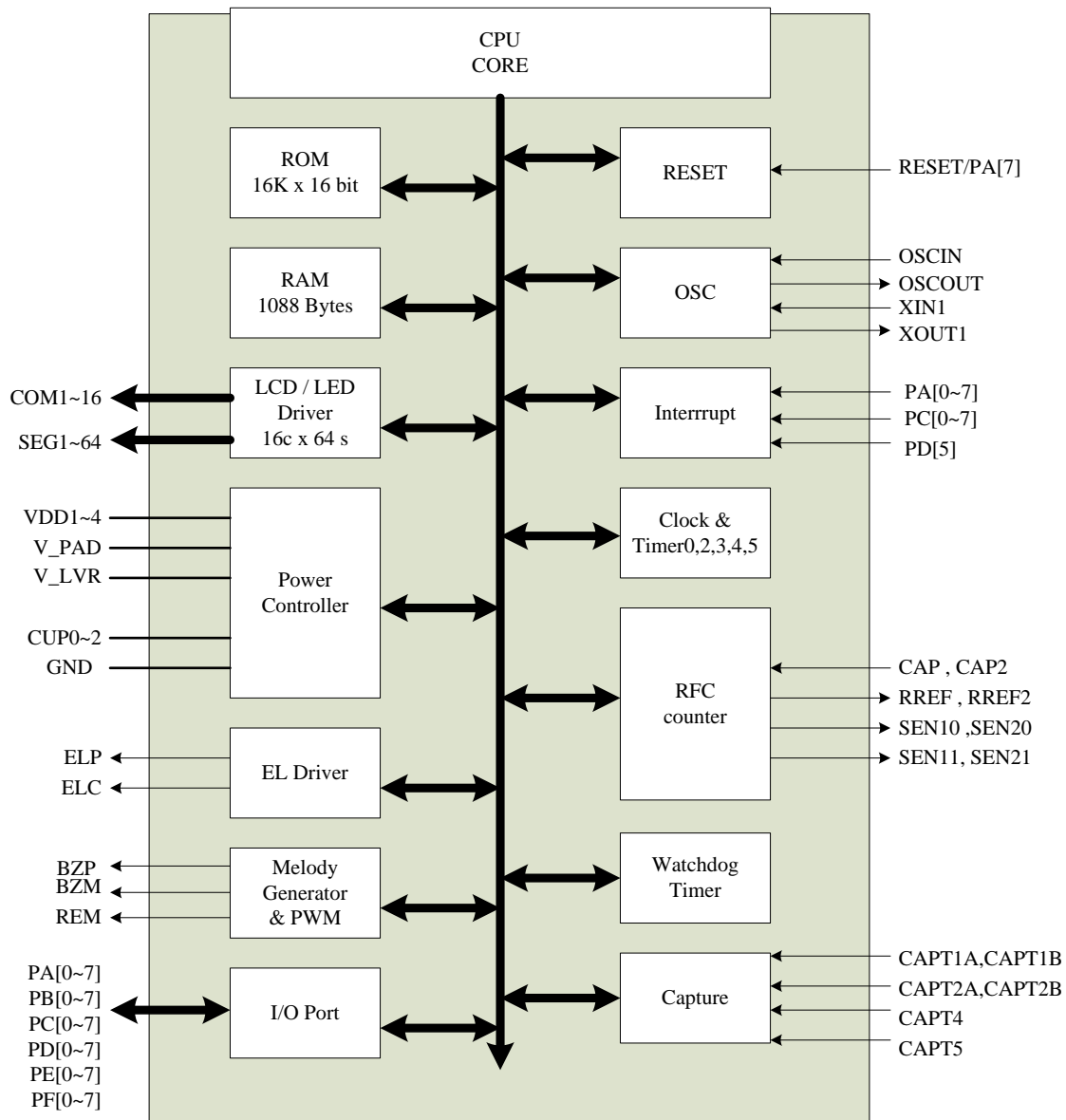
# 1. Product Overview

## 1.1 Feature

- ROM size: 16K \* 16 bits
- RAM size: 1088\* 8 bits
- Stack: 8 layers
- LCD/LED Driver: 16com \* 64seg
  - Duty: 1/2,1/3,1/4,1/5,1/6 ,1/7, 1/8 , 1/10,1/12,1/16 can be selected by register
  - 1/2 ,1/3 & 1/4 bias charge pump , 2 LED modes.
  - COM5~16 can be selected as I/O Port.
- I/O port: 47 bi-direction I/O port, 1 input port.
  - . PA[0~6] can be set to pull-up, pull-down,normal output, pmos open-drain or nmos open-drain
  - . PA7 input only with pull-down.
  - . PB[0~7] can be set to pull-down ,normal output or pmos open-drain
  - . PC[0~7] can be set to pull-down ,normal output or pmos open-drain
  - . PD[0~7] can be set to pull-down ,normal output or pmos open-drain
  - . PE[0~7] can be set to pull-down ,normal output or pmos open-drain
  - . PF[0~7] can be set to pull-down ,normal output or pmos open-drain
- Pin-edge interrupt
  - Global pins: PC[0~7] & PD[0~7]
  - Single pin : PD[5]
- Key strobe function – normal mode use only.
  - Polling mode : PC[0~7] & PD[0~7]
- System Clock: Dual clock operation
  - Low speed -> External 32KHz crystal, external R oscillator or internal slow RC oscillator,
  - High speed -> External 4MHz crystal, external R oscillator or (700Khz or 1.5 MHz) internal RC oscillator by configuration option
- TIMER0 (TM0):
  - One 8 bit general purpose timer
  - Remote output (include REM carrier )
  - TM2 & 3 RFC timer base input
- TIMER2 & TIMER3 (TM2 & TM3)
  - Two 8-bit Tmier : TM0,TM2 & TM3
  - One 16-bit Tmier : TM2 +TM3
  - Two 8-bit Capture/RFC : TM2 & TM3
  - One 16-bit Capture/RFC : TM2 +TM3
  - Three 8-bit PWM : TM2 , TM3 , (TM2 + TM3)
- TIMER4 & TIMER5 (TM4 & TM5)

- Two 8-bit Tmier : TM4 & TM5
- One 16-bit Tmier : TM4 +TM5
- Two 8-bit Capture/RFC : TM4 & TM5
- One 16-bit Capture/RFC : TM4 +TM5
- Three 8-bit PWM : TM4 , TM5 , (TM4 + TM5)
- **TIMER4 & TIMER5 (TM4 & TM5)**
  - Two 8-bit Tmier : TM4 & TM5
  - One 16-bit Tmier : TM4 +TM5
  - Two 8-bit Capture/RFC : TM4 & TM5
  - One 16-bit Capture/RFC : TM4 +TM5
  - Three 8-bit PWM : TM4 , TM5 , (TM4 + TM5)
- **Other TIME base sources**
  - PH\_IRQ
  - PH\_CLK
  - PH2\_CLK
  - 2HZ
  - One 16 bit pre-divider
  - Watchdog timer
- **Watchdog timer & 4'key reset function**
  - CONFIG WDTE=0 : 4'key reset enable & wdtdog timer disable
  - CONFIG WDTE=1 : 4'key reset disable & wdtdog timer enable
- **Built-in two RFC channel – (CAP1, REF1, SEN10 & SEN11) and (CAP2,REF2,SEN20 & SEN21)**
- **Built-in two PWM output – PWM2, PWM3,PWM4 & PWM5**
- **Built-in four Capture channel – CAPT1A, CAPT1B, CAPT2A ,CAPT2B , CAPT4 & CAPT5 .**
- **One comparator (three sets input )**
- **Two OP**
- **Spi mode**
- **IRQ source : 10**
- **Built-in EL driver circuit**
- **Built-in programmable FREQ, Tone & REM output.**
- **Multi-function (BZ,BZM) output – Tone , FREQ, 1Khz, 2Khz, 4Khz , PWM2 ,PWM3,PWM4 or PWM5 output**
- **Built-in Low Voltage Reset (LVR) – 2.1V**
- **Built-in Low Battery Detect – 2.68V, 2.56V & 2.42V**
- **REM – Remote output.**
- **HALT and SLEEP operation mode**
- **Fast instruction cycle time: 61us@32KHz operating**
- **Low power consumption (XIN1,XOUT1): 1.5 uA (@ 32KHz Halt mode , LCD on, No load )**

## 1.2 Block Diagram

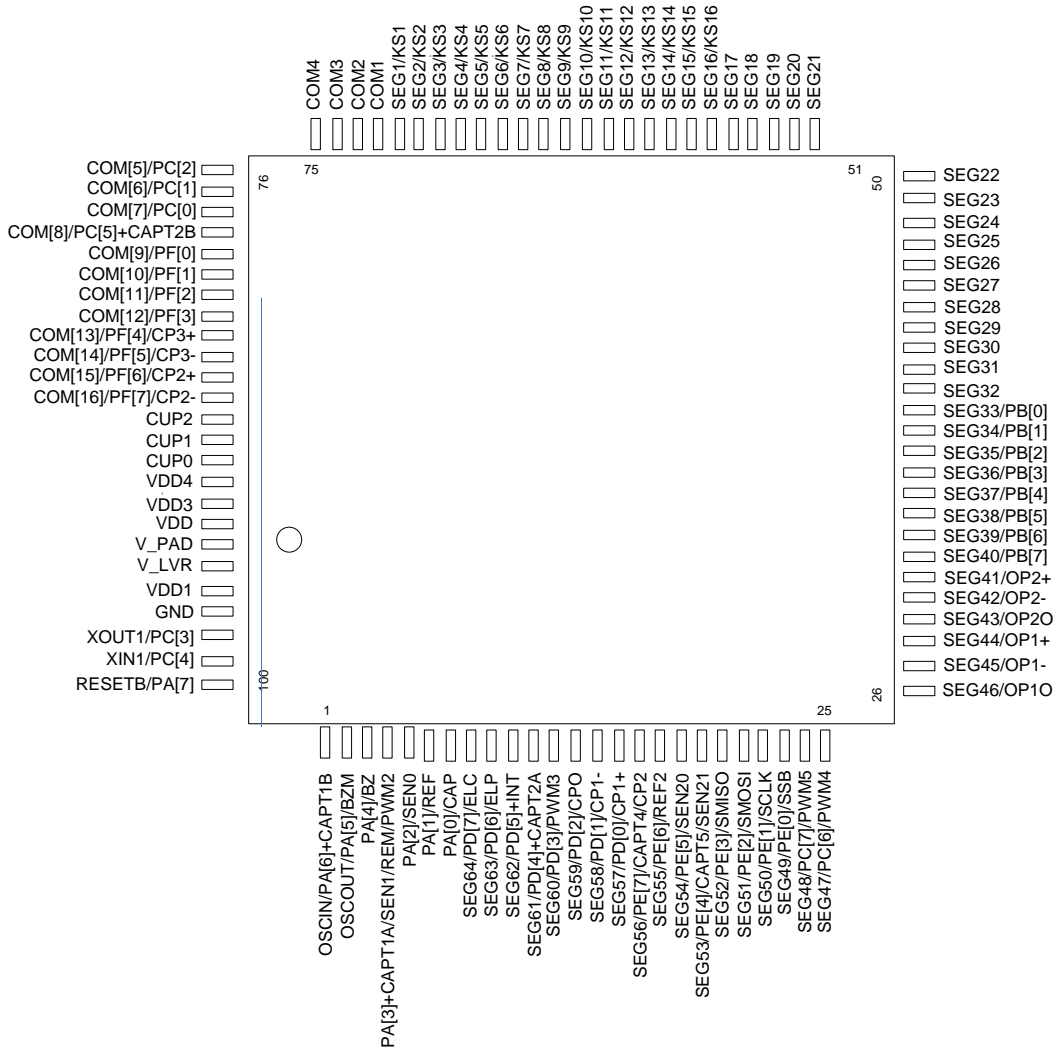


### 1.3 Pin Assignment ( Pin assignment for COB )

No	Name	No	Name
1	COM16 /PF[7]/CP3-	51	SEG35 / PB[2]
2	COM15 /PF[6]/CP3+	52	SEG36 / PB[3]
3	COM14 /PF[5]/CP2-	53	SEG37 / PB[4]
4	COM13/PF[4]/CP2+	54	SEG38 / PB[5]
5	COM12/PF[3]	55	SEG39 / PB[6]
6	COM11/PF[2]	56	SEG40 / PB[7]
7	COM10/PF[1]	57	SEG41/ OP2+
8	COM9/PF[0]	58	SEG42/ OP2-
9	COM8/PC[5] +CAPT2B	59	SEG43/ OP2O
10	COM7 /PC[0]	60	SEG44/ OP1+
11	COM6 /PC[1]	61	SEG45/ OP1-
12	COM5 /PC[2]	62	SEG46/ OP1O
13	COM4	63	SEG47 /PC[6] /PWM4
14	COM3	64	SEG48 /PC[7] /PWM5
15	COM2	65	SEG49 /PE[0] /SSB
16	COM1	66	SEG50 /PE[1] /SCLK
17	SEG1 /KS1	67	SEG51 /PE[2] /SMOSI
18	SEG2 /KS2	68	SEG52 /PE[3] / SMISO
19	SEG3 /KS3	69	SEG53 /PE[4]+CAPT5 /SEN21
20	SEG4 /KS4	70	SEG54 /PE[5] /SEN20
21	SEG5 /KS5	71	SEG55 /PE[6] /REF2
22	SEG6 /KS6	72	SEG56 /PE[7] +CAPT4 /CAP2
23	SEG7 /KS7	73	SEG57 /PD[0] /CP1+
24	SEG8 /KS8	74	SEG58 /PD[1] / CP1-
25	SEG9 /KS9	75	SEG59 /PD[2] / CPO
26	SEG10 /KS10	76	SEG60 /PD[3] / PWM3
27	SEG11 /KS11	77	SEG61 /PD[4] +CAPT2A
28	SEG12 /KS12	78	SEG62 /PD[5]+INT
29	SEG13 /KS13	79	SEG63 /PD[6] /ELP
30	SEG14 /KS14	80	SEG64 /PD[7] /ELC
31	SEG15 /KS15	81	PA[0] /CAP
32	SEG16 /KS16	82	PA[1] /REF
33	SEG17	83	PA[2] /SEN0
34	SEG18	84	PA[3+CAPT1A/SEN1/REM/PWM2
35	SEG19	85	PA[4] /BZ
36	SEG20	86	OSCOUT /PA[5]/BZM
37	SEG21	87	OSCIN /PA[6]+CAPT1B
38	SEG22	88	RESETB /PA[7]
39	SEG23	89	XIN/PC[4]
40	SEG24	90	XOUT/PC[3]
41	SEG25	91	GND
42	SEG26	92	VDD1
43	SEG27	93	V_LVR
44	SEG28	94	V_PAD
45	SEG29	95	VDD2
46	SEG30	96	VDD3
47	SEG31	97	VDD4
48	SEG32	98	CUP0
49	SEG33/ PB[0]	99	CUP1
50	SEG34/ PB[1]	100	CUP2



LQFP 100-PIN



## 1.4 Pin Description

Pin name	I/O	Description
PA[0] /CAP	I,I/O	1. Key input pin 2. Input port with pull up, pull down or pin wake up 3. Output port (Normal output, P open drain or N open drain) 4. CAP input
PA[1] /REF	I,I/O	1. Input port with pull up, pull down or pin wake up 2. Output port (Normal output, P open drain or N open drain) 3. REF output
PA[2] /SEN0	I,I/O	1. Input port with pull up, pull down or pin wake up 2. Output port (Normal output, P open drain or N open drain) 3. SEN0 output
PA[3]+CAPT1A/SEN1/ REM /PWM2	I,I/O	1. Input port with pull up, pull down or pin wake up 2. Output port (Normal output, P open drain or N open drain) 3. CAPT1A input 4. REM output 5. PWM2 (TM2) output 6. SEN1 output
PA[4] /BZ	I,I/O	1. Input port with pull up, pull down or pin wake up 2. Output port (Normal output, P open drain or N open drain) 3. BZM output
OSCOUT/ PA[5]+BZM	I,I/O	1. Crystal out 2. I/O port (pull-high, pull-down, open-drain, pin wake up)t 3. BZM output
OSCIN /PA[6]+CAP1B	O,I/O	1. Crystal in 2. External RC input 3. I/O port (pull-high, pull-down, open-drain, pin wake up) 4. CAP1B input
RESETB/PA[7]/INT	I,I	1. System reset input (floating) 2. Input port with pull-low 3. Wake –up on pin change 4. Pin interrupt
XIN1 /PC[4]	I,O	1. Crystal input (32K) 2. External RC input 3. Input port with pull-down 4. Output port (Normal output or PMOS open-drain)
XOUT1 /PC[3]	O,O	1. Crystal out 2. Input port with pull-down 3. Output port(Normal output or PMOS open-drain)
GND	P	System Ground
VPAD	P	System PAD power
VDD1	P	LCD power supply
VDD2	P	LCD power supply
VDD3	P	LCD power supply
VDD4	P	LCD power supply
V_PAD	P	IC PAD power supply
V_LVR	P	LVR/LVR power supply
CUP0	P	Voltage step up capacitor
CUP1	P	Voltage step up capacitor

CUP2	P	Voltage step up capacitor
COM1~4	O	LCD COM output
COM5/PC[2]	O,I	1. LCD COM output 2. Input port with pull-down 3. Output port (Normal output or PMOS open-drain)
COM6/PC[1]	O,I	1. LCD COM output 2. Input port with pull-down 3. Output port (Normal output or PMOS open-drain)
COM7/PC[0]	O,I	1. LCD COM output 2. Input port with pull-down 3. Output port (Normal output or PMOS open-drain)
COM8/PC[5]+ CAPT2B	O,I	1. LCD COM output 2. Input port with pull-down 3. Output port (Normal output or PMOS open-drain) 4. Capture input
COM9/PF[0]	O,I	1. LCD COM output 2. Input port with pull-down 3. Output port (Normal output or PMOS open-drain)
COM10/PF[1]	O,I	1. LCD COM output 2. Input port with pull-down 3. Output port (Normal output or PMOS open-drain)
COM11/PF[2]	O,I	1. LCD COM output 2. Input port with pull-down 3. Output port (Normal output or PMOS open-drain)
COM12/PF[3]	O,I	1. LCD COM output 2. Input port with pull-down 3. Output port (Normal output or PMOS open-drain)
COM13/PF[4] /CP2+	O,I	1. LCD COM output 2. Input port with pull-down 3. Output port (Normal output or PMOS open-drain) 4. Comparator V+ input
COM14/PF[5] /CP2-	O,I	1. LCD COM output 2. Input port with pull-down 3. Output port (Normal output or PMOS open-drain) 4. Comparator V- input
COM15/PF[6]/CP3+	O,I	1. LCD COM output 2. Input port with pull-down 3. Output port (Normal output or PMOS open-drain) 4. Comparator V+ input
COM16/PF[7]/CP3-	O,I	1. LCD COM output 2. Input port with pull-down 3. Output port (Normal output or PMOS open-drain) 4. Comparator V- input
SEG1~16 /KS1~16	O, O	1. LCD segment output 2. Key Strobe Output
SEG17~32	O	LCD segment output
SEG33 /PB[0]	O,I/O	1. LCD COM output 2. Input port with pull-down 3. Output port (Normal output or PMOS open-drain)
SEG34 /PB[1]	O,I/O	1. LCD COM output 2. Input port with pull-down 3. Output port (Normal output or PMOS open-drain)
SEG35 /PB[2]	O,I/O	1. LCD COM output 2. Input port with pull-down

		3. Output port (Normal output or PMOS open-drain)
SEG36 /PB[3]	O,I/O	1. LCD COM output 2. Input port with pull-down 3. Output port (Normal output or PMOS open-drain)
SEG37 /PB[4]	O,I/O	1. LCD COM output 2. Input port with pull-down 3. Output port (Normal output or PMOS open-drain)
SEG38 /PB[5]	O,I/O	1. LCD COM output 2. Input port with pull-down 3. Output port (Normal output or PMOS open-drain)
SEG39 /PB[6]	O,I/O	1. LCD COM output 2. Input port with pull-down 3. Output port (Normal output or PMOS open-drain)
SEG40 /PB[7]	O,I/O	1. LCD COM output 2. Input port with pull-down 3. Output port (Normal output or PMOS open-drain)
SEG41/ OP2+	O,I	1. LCD segment output 2. OP2+ input
SEG42/ OP2-	O,I	3. LCD segment output 4. OP2- input
SEG43/ OP2O	O,O	5. LCD segment output 6. OP2O output
SEG44/ OP1+	O,I	7. LCD segment output 8. OP1+ input
SEG45/ OP1-	O,I	9. LCD segment output 10. OP1- input
SEG46/ OP1O	O,O	11. LCD segment output 12. OP1 output
SEG47 /PC[6] /PWM4	O,I/O	1. LCD segment output 2. PC[6] input port with pull-down 3. PC[6] output port (Normal output or PMOS open-drain) 4. PWM4 (TM4) output
SEG48 /PC[7] /PWM5	O,I/O	1. LCD segment output 2. PC[7] input port with pull-down 3. PC[7] output port (Normal output or PMOS open-drain) 4. PWM5 (TM5) output
SEG49 /PE[0] /SSB	O,I/O	1. LCD segment output 2. PE[0] input port with pull-down 3. PE[0] output port (Normal output or PMOS open-drain) 4. SPI mode SSB input
SEG50 /PE[1] /SCLK	O,I/O	1. LCD segment output 2. PE[1] input port with pull-down 3. PE[1] output port (Normal output or PMOS open-drain)

		4. SPI mode SCLK input
SEG51 /PE[2] /SMOSI	O,I/O	1. LCD segment output 2. PE[2] input port with pull-down 3. PE[2] output port (Normal output or PMOS open-drain) 4. SPI mode SMOSI control
SEG52 /PE[3] /SMISO	O,I/O	1. LCD segment output 2. PE[3] input port with pull-down 3. PE[3] output port (Normal output or PMOS open-drain) 4. SPI mode SMISO I/O
SEG53 /PE[4] +CAPT4 /SEN21	O,I/O	1. LCD segment output 2. PE[4] input port with pull-down 3. PE[4] output port (Normal output or PMOS open-drain) 4. CAPT5 input (capture input ) 5. RFC mode SEN21 output
SEG54 /PE[5] /SEN20	O,I/O	1. LCD segment output 2. PE[5] input port with pull-down 3. PE[5] output port (Normal output or PMOS open-drain) 4. RFC mode SEN20 output
SEG55 /PE[6] /REF2	O,I/O	1. LCD segment output 2. PE[6] input port with pull-down 3. PE[6] output port (Normal output or PMOS open-drain) 4. RFC mode REF2 output
SEG56 /PE[7] +CAPT4 /CAP2	O,I/O	1. LCD segment output 2. PE[7] input port with pull-down 3. PE[7] output port (Normal output or PMOS open-drain) 4. CAPT4 input (capture input ) 5. RFC mode CAP2 input
SEG57 /PD[0] / CP+	O,I/O	1. LCD segment output 2. PD[0] input port with pull-down 3. PD[0] output port (Normal output or PMOS open-drain) 4. Comparator V+ input
SEG58 /PD[1] / CP-	O,I/O	1. LCD segment output 2. PD[1] input port with pull-down 3. PD[1] output port (Normal output or PMOS open-drain) 4. Comparator V- input
SEG59 /PD[2] /CPO	O,I/O	1. LCD segment output 2. PD[2] input port with pull-down 3. PD[2] output port (Normal output or PMOS open-drain) 4. Comparator output
SEG60 /PD[3] /PWM3	O,O	1. LCD segment output 2. PD[3] input port with pull-down

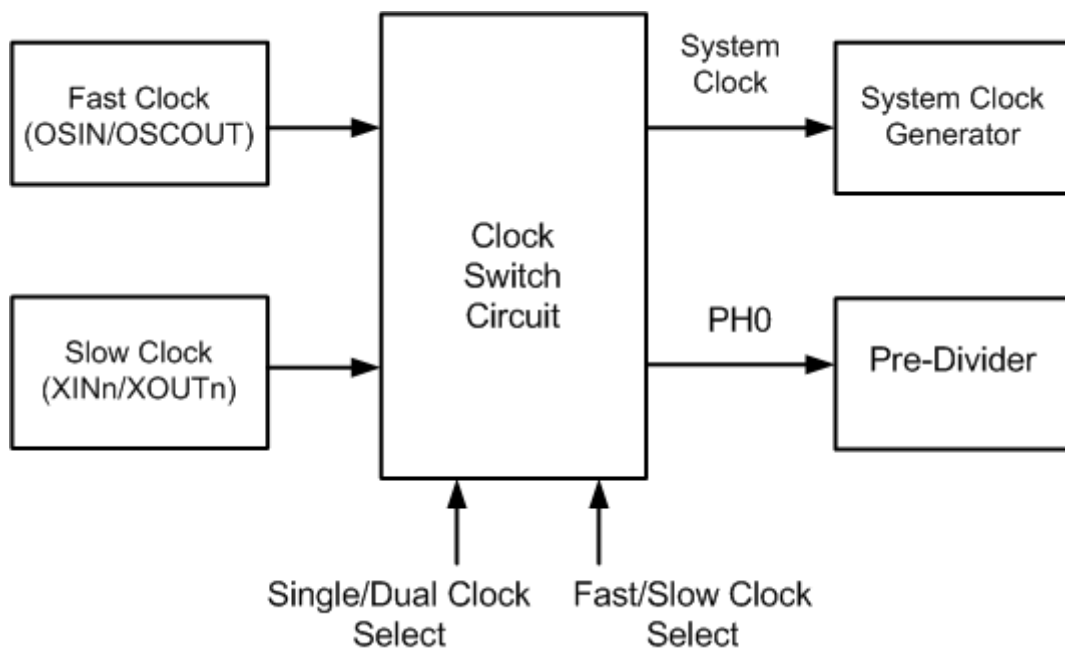
		<ul style="list-style-type: none"> <li>3. PD[3] output port (Normal output or PMOS open-drain)</li> <li>4. PWM3 (TM3) output</li> </ul>
SEG61 /PD[4]+CAPT2A	O,O	<ul style="list-style-type: none"> <li>1. LCD segment output</li> <li>2. PD[4] input port with pull-down</li> <li>3. PD[4] output port (Normal output or PMOS open-drain)</li> <li>4. CAPT2A input (capture input)</li> </ul>
SEG62 /PD[5]+INT	O,O	<ul style="list-style-type: none"> <li>1. LCD segment output</li> <li>2. PD[5] input port with pull-down</li> <li>3. PD[5] output port (Normal output or PMOS open-drain)</li> <li>4. PIN interrupt input</li> </ul>
SEG63 /PD[6]/ELP	O,O	<ul style="list-style-type: none"> <li>1. LCD segment output</li> <li>2. PD[6] input port with pull-down</li> <li>3. PD[6] output port (Normal output or PMOS open-drain)</li> <li>4. ELP output</li> </ul>
SEG64 /PD[7]/ELC	O,O	<ul style="list-style-type: none"> <li>1. LCD segment output</li> <li>2. PD[7] input port with pull-down</li> <li>3. PD[7] output port (Normal output or PMOS open-drain)</li> <li>4. ELC output</li> </ul>

## 2. System Architecture

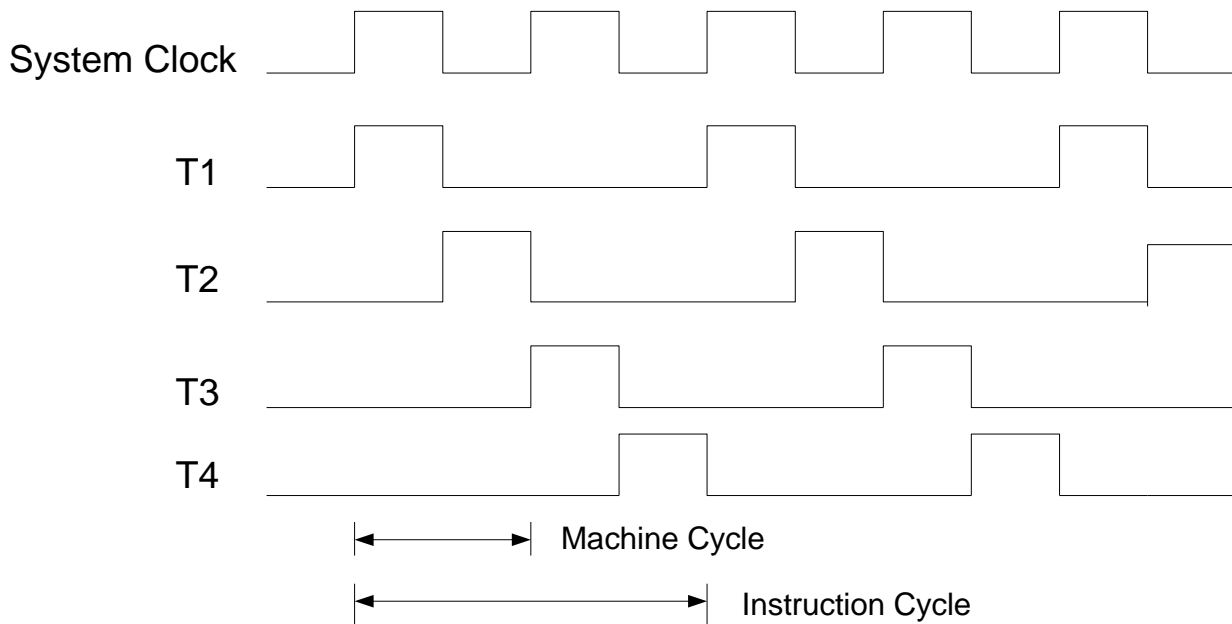
### 2.1 System Clock

The MK9A160P has dual clock operation mode and user can set bit0~3 of configuration register to be their request. OSCIN/OSCOU, which we call fast clock, can be used to connect external 4MHz crystal, external R oscillator or internal (700K,1.5MHz) RC oscillator by configuration option. When user select internal 700KHz RC oscillation mode, these two pins can be used as KI or I/O ports (PA) that is more flexibility. XIN1/XOUT1, which we call slow clock, has the same situation. They can be used to connect external 32KHz crystal, external R oscillator or internal 50KHz RC oscillator. When user select internal 50KHz RC oscillator, these two pin can be used as output ports (PB). The clock mode can be selected by configuration bit. Once dual clock mode is selected, user can switch between fast and slow clock by setting bit 7 of SYS\_CTL (\$3Eh). Or turn on/off these clock source individually by setting bit0~1.

The clock oscillation block diagram is as below which is composed of fast clock and slow clock.



**Fig.2.1.1** Clock switch circuit & system clock generator



**Fig.2.1.2 Machine Cycle & Instruction Cycle**

The below table shows the clock source status of system clock and pre-divider in different mode:

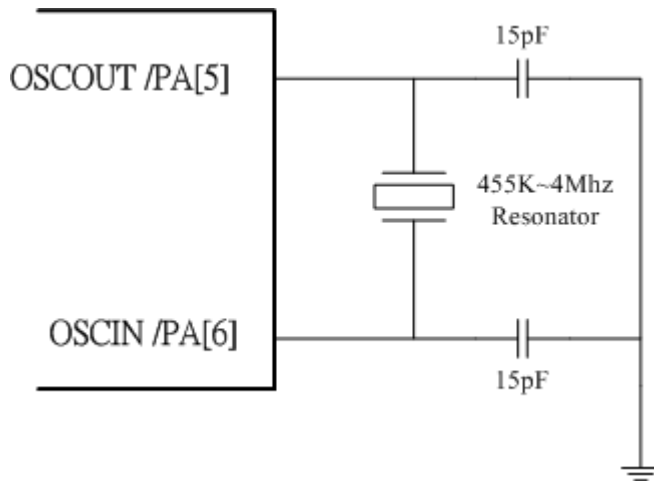
Clock Mode	System Clock	PH0
Slow Clock Only	SCLK (Slow Clock)	SCLK
Fast Clock Only	FCLK (Fast Clock)	FCLK
Initial Stage (Dual Clock Mode)	SCLK	SCLK
HALT Stage (Dual Clock Mode)	SCLK	SCLK
Slow Clock Active (Dual Clock Mode)	SCLK	SCLK
Fast Clock Active (Dual Clock Mode)	FCLK	SCLK



### 2.1.1 Fast Clock (FCLK) Connection

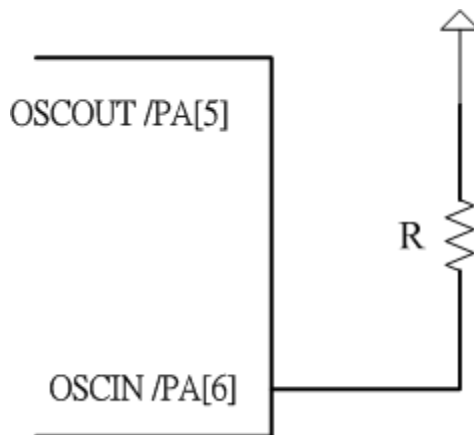
There are 3 connection circuits of fast clock which are external max. 4MHz crystal or resonator, external R oscillator and internal (700KHz,1.5Mhz) RC oscillator. User can select the operation mode by setting configuration register bit 2~3. The connection are as below drawing:

甲、 Connect external 3.58MHz Resonator, (FOSC1,FOSC0)=(0,0)



乙、 Connect external R oscillator, (FOSC1,FOSC0)=(1,0)

When set to this mode, OSCOUT pin can be used as I/O port (PA5).



丙、 Fast clock is set to internal RC oscillator or No, (FOSC1,FOSC0)=(1,1) or (0,1)

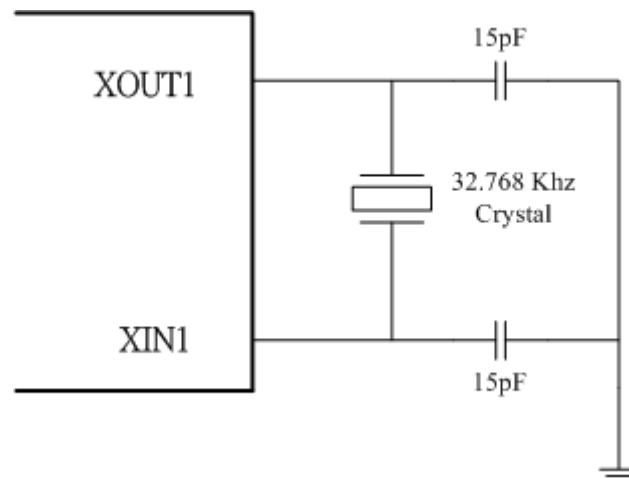
When set to this mode, OSCIN and OSCOUT can be used as I/O port (PA5 and PA6)



### 2.1.2 Slow Clock (SCLK) Connection

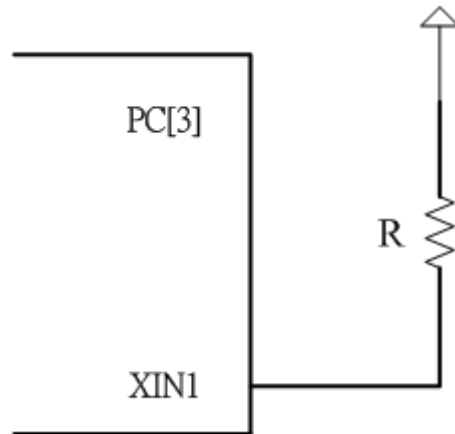
There are 3 connection circuits of fast clock which are external 32KHz crystal, external slow oscillator and internal slow RC oscillator. User can select the operation mode by setting configuration register bit 0~1. The connection are as below drawing:

- (a) Connect external 32.768KHz Crystal, (SOSC1,SOSC0)=(0,0)



- (b) Connect external slow oscillator, (SOSC1,SOSC0)=(1,0)

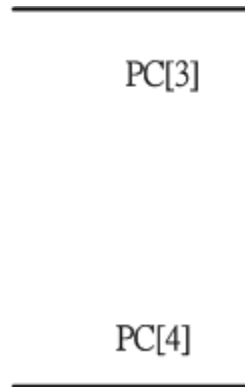
When set to this mode, XOUT1 pin can be used as i/o port (PC[3]).



(d) Slow clock is set to internal slow oscillator or No, (SOSC1,SOSC0)=(1,1) or (0,1)

When set to this mode, XIN1 can be used as i/o port (PC[4]) and

XOUT1 can be used as i/o ports (PC[3]).



## 2.1.3 FCLK & SCLK Switch

### 2.1.3-1 Cpu clock switch from SCLK to FCLK

```
;; cpu clock = SCLK
BC          SYS_CTL,1  ;; enable fast clock
NOP        ;; fast clock stabile time
NOP        ;; need !
NOP        ;; need !
BS          SYS_CTL,7  ;; cpu clock = FCLK
```

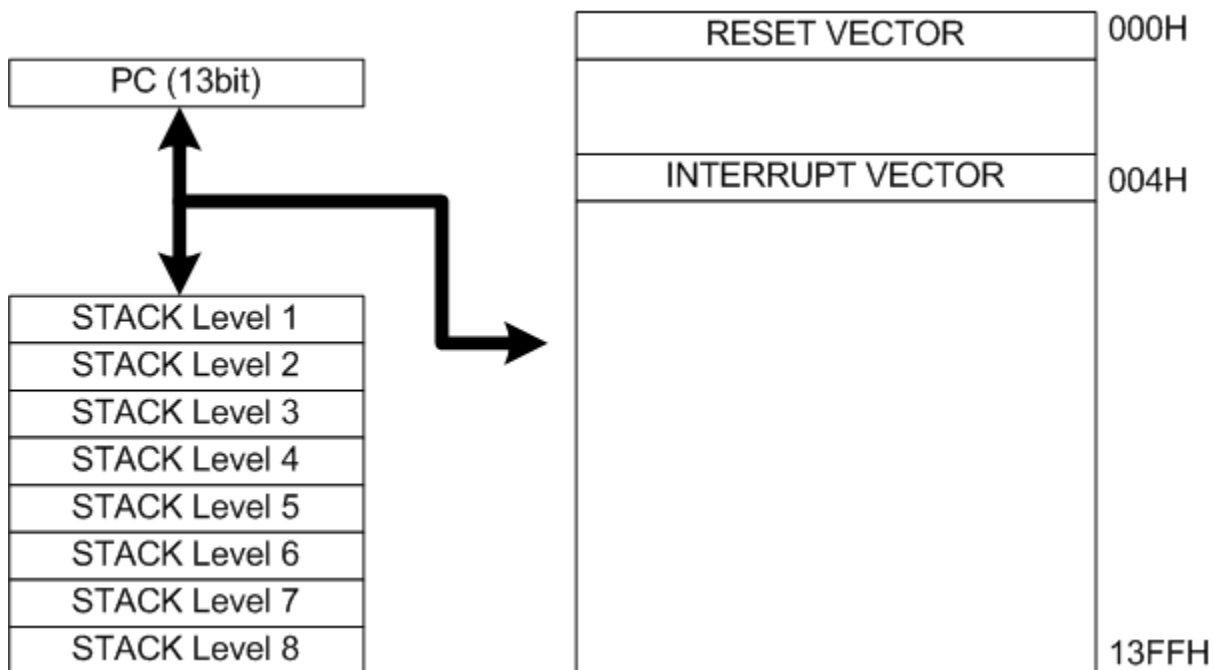
### 2.1.3-2 Cpu clock switch from FCLK to SCLK

```
;; FCLK & SCLK are all ON , cpu clock = FCLK
BC          SYS_CTL,7  ;; cpu clock = SCLK
NOP        ;; need !
NOP        ;; need !
BS          SYS_CTL,1  ;; Stop FCLK
```

## 2.2 Program Memory (ROM)

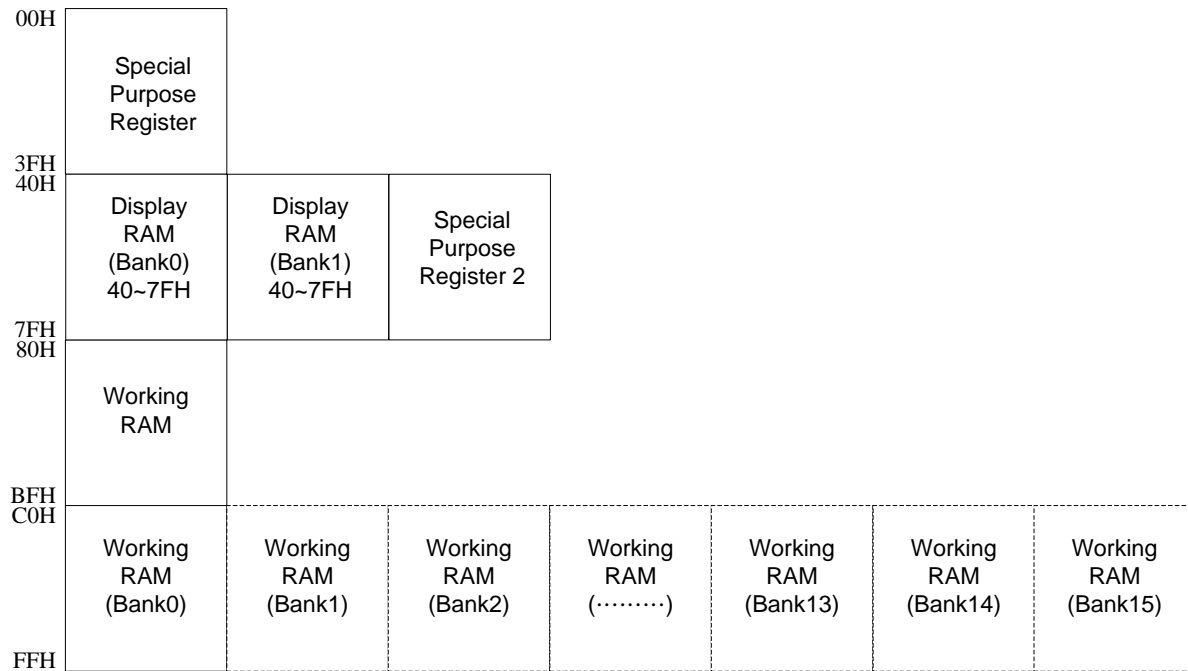
Instruction and table are stored at this area. There is only one interrupt vector existed which means all the interrupt occurred would jump to the same vector. Programmer should use interrupt flag to judge what kind of interrupt is occurred. The program counter (PC) is 13 bit which can directly address all the 8K x 16 location. Look-up table can be put at anywhere of ROM.

The RESET vector is located at 000H and Interrupt vector is at 004H. The map is as below



### 2.3 Data Memory (RAM)

The total RAM volume are 1213 x 8bits which includes three kinds of register group. One is 1088 x 8bits working RAM, another is special purpose register that are 75 x 8bits and display RAM are 50 x 8bits. The data memory map is as below:



**Fig.2.3.1 Memory mapping**

User can set WBANK register to switch different bank of data RAM.

## 2.4 Configuration Register

This register store set up option of chip which includes reset pin definition, timer clock source select, LVR detect voltage select and WDT control. The content of register can not be changed by software and will be fixed by Writer. It is like mask option when use mask ROM type MCU.

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CONFIG_L	LV1	LV0	WDTE	CPT	FOSC1	FOSC0	SOSC1	SOSC0
-	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
CONFIG_H	--	BYPASS	--	POWER	--	RSTE2	RSTE1	RST_DEF

- Bit 14 : Bypass select
  - 1 : Noise control ON (EMS is better ,but increase power consumption )
  - 0 : Noise control OFF .
- Bit 12 : Low power control
  - 1 : ON
  - 0 : OFF
- Bit 10~9 (RSTE2.1): PA reset key number control (Only PA work as I/O mode )

When press PA keys over 2 sec, Reset occurred

- 1 1: max. 4-key reset (4 key depress simultaneously) PA7,PA6,PA5,PA3
- 1 0: max. 4-key Reset (4 key depress simultaneously) PA7,PA5,PA4,PA3
- 0 1: max. 4-key Reset (4 key depress simultaneously) PA6,PA5,PA4,PA3
- 0 0: max. No key reset.

IO	WDTE.RSTE2~1	2'S reset RESET pins function	RESET pins				
			PA3	PA4	PA5	PA6	PA7
IO	1 X X	No key reset function	X	X	X	X	X
IO	0 0 0	No key reset function	X	X	X	X	X
IO	0 0 1	PA7,PA6,PA5,PA3	ON	X	ON	ON	ON
IO	0 1 0	PA7,PA5,PA4,PA3	ON	ON	ON	X	ON
IO	0 1 1	PA6,PA5,PA4,PA3	ON	ON	ON	ON	X

**<Note> To use this key reset function, PA should be set as I/O port at first in PAD\_CTL2(\$14) register and set PA as input pull-high in normal operation mode. When use this function and enter SLEEP mode, the specific PA port would automatically set as pull high. For example, if user define PA[0]~PA[1] to be key reset. When system enter SLEEP mode, PA[0] and PA[1] would be set as pull-high. At this moment, user should be careful don't hold these keys down, otherwise that will cause the power consumption.**

- Bit8 (RST\_DEF): RESETB pin function define  
 0: RESETB used as normal Input pin  
 1: RESETB used as system reset pin
- Bit7~6 (LV1~0): Low voltage reset function voltage selection bits

Bit7	Bit6	Detect voltage
LV1	LV0	
0	0	1.5V
0	1	1.7v
1	0	2V
1	1	Don't use

- Bit5 (WDTE): Watchdog timer enable/disable control  
 0: WDT disable  
 1: WDT enable
- Bit4 (CPT): Code Protection bit  
 0: ON  
 1: OFF
- Bit3~2 (FOSC2~1): OSCIN/OSCOUT frequency assignment bit.

Bit3	Bit2	OSC Type	Resonance Frequency
FOSC1	FOSC0		
0	0	NT (Normal speed)	455KHz~10Mhz resonator or crystal
0	1	No	OSCIN & OSCOUT used as I/O port or KI
1	0	External R	1. OSCIN connect to R (455KHz~4MHz) 2. OSCOUT can be used as I/O port
1	1	Internal RC	1. Internal (700KHz,1.5Mhz) RC oscillator 2. OSCIN & OSCOUT can be used as I/O port



- Bit1~0 (SOSC1~0): XIN1/XOUT1 frequency assignment bit.

Bit1	Bit0	OSC Type	Resonance Frequency
SOSC1	SOSC0		
0	0	LP (low power, Low speed)	XIN1 & XOUT1 work as CRYSTAL
0	1	No	XIN1 & XOUT1 work as I/O .
1	0	External R	1. XIN1 connect to R (around 50KHz) 1. XOUT1 can be used as PC3 .
1	1	Internal RC	3. Internal 50KHz RC oscillator 4. XIN1 & XOUT1 can be used as I/O port

<Note> Below table shows system clock status during Reset stage and CLKS bit setting:

Bit3	Bit2	Bit1	Bit0	System clock			PH0
FOSC1	FOSC0	SOSC1	SOSC0	Reset	CLKS=0	CLKS=1	
0	0	0	0	Slow clock	Slow clock	Fast clock	Slow clock
0	0	0	1	Slow clock only, Can't write "1" to CLKS			Slow clock
0	0	1	X	Slow clock	Slow clock	Fast clock	Slow clock
0	1	0	0	Fast clock only, Can't write "0" to CLKS			Slow clock
0	1	0	1	Don't care			XX
0	1	1	X	Fast clock only, Can't write "0" to CLKS			Fast clock
1	0	0	0	Slow clock	Slow clock	Fast clock	Slow clock
1	0	0	1	Slow clock only, Can't write "1" to CLKS			Slow clock
1	0	1	X	Slow clock	Slow clock	Fast clock	Slow clock
1	1	0	0	Slow clock	Slow clock	Fast clock	Slow clock
1	1	0	1	Slow clock only, Can't write "1" to CLKS			Slow clock
1	1	1	X	Slow clock	Slow clock	Fast clock	Slow clock

## 2.5 Special Purpose Register

The listed register table is as below, we will describe them in detail at the specific chapter.

Name	Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INDF	\$00	A7	A6	A5	A4	A3	A2	A1	A0
PCL	\$01	A7	A6	A5	A4	A3	A2	A1	A0
PCH	\$02	--	--	--	A12	A11	A10	A9	A8
STATUS	\$03	--	CORE_VDD	VOLT	$\overline{TO}$	$\overline{PD}$	Z	DC	C
FSR	\$04	BANK1	BANK0	D5	D4	D3	D2	D1	D0
<b>I/O PAD &amp; Control</b>									
Name	Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PA_DIR	\$05	--	DA6	DA5	DA4	DA3	DA2	DA1	DA0
PA_WAKE_UP	\$07	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
PA_EDGE	\$2D	EDGE7	EDGE6	EDGE5	EDGE4	EDGE3	EDGE2	EDGE1	EDGE0
PA_PUD1	\$08	A3-2	A3-1	A2-2	A2-1	A1-2	A1-1	A0-2	A0-1
PA_PUD2	\$09	--	A7-1	A6-2	A6-1	A5-2	A5-1	A4-2	A4-1
PA_DAT	\$0A	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
PC_WAKE_UP	\$1D	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
PC_EDGE	\$1E	EDGE7	EDGE6	EDGE5	EDGE4	EDGE3	EDGE2	EDGE1	EDGE0
PC_CTL	\$0B	KI7/IN7	KI6/IN6	KI5/IN5	KI4/IN4	KI3/IN3	KI2/IN2	KI1/IN1	KI0/IN0
PC_DIR	\$0C	DC7	DC6	DC5	DC4	DC3	DC2	DC1	DC0
PC_PUD	\$0D	UC7	UC6	UC5	UC4	UC3	UC2	UC1	UC0
PC_DAT	\$0E	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
PD_DIR	\$0F	DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0
PD_PUD	\$10	UD7	UD6	UD5	UD4	UD3	UD2	UD1	UD0
PD_CTL	\$11	KI7/IN7	KI6/IN6	KI5/IN5	KI4/IN4	KI3/IN3	KI2/IN2	KI1/IN1	KI0/IN0
PD_DAT	\$12	PD7	PD6	PD5	PD4	PD3	PE2	PE1	PD0
PE_DIR	\$1A	DE7	DE6	DE5	DE4	DE3	DE2	DE1	DE0
PE_PUD	\$1B	UE7	UE6	UE5	UE4	UE3	UE2	UE1	UE0
PE_DAT	\$1C	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
PF_DIR	\$2C	DF7	DF6	DF5	DF4	DF3	DF2	DF1	DF0
PF_PUD	\$73	UF7	UF6	UF5	UF4	UF3	UF2	UF1	UF0
PF_DAT	\$74	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
PAD_CTL1	\$13	SEG64/ PD[7]	SEG63/ PD[6]	SEG62/ PD[5]	SEG61/ PD[4]	SEG60/ PD[3]	SEG59/ PD[2]	SEG58/ PD[1]	SEG57/ PD[0]
PAD_CTL2	\$14	--	C6	C5	C4	C3	C2	C1	C0
PAD_CTL3	\$15	EDGE	--	SEN21_ON	SEN20_ON	REF2_ON	SEN1_ON	SEN0_ON	REF_ON

<b>PAD_CTL4</b>	\$16	SEG56/ PE[7]	SEG55/ PE[6]	SEG54/ PE[5]	SEG53/ PE[4]	SEG52/ PE[3]	SEG51/ PE[2]	SEG50/ PE[1]	SEG49/ PE[0]
<b>PAD_CTL5</b>	\$28	CAP2/ PE[7]	REF2/ PE[6]	SEN20/ PE[5]	SEN21/ PE[4]	SMISO/ PE[3]	SMOSI/ PE[2]	SCLK/ PE[1]	SSB/ PE[0]
<b>PAD_CTL6</b>	\$29	CP_EN	CP_OUT	CPO_EN	CP_S1	CP_S0	PF3C	PF1C	PWM3/ PD[3]
<b>PAD_CTL8</b>	\$06	OP2_SL	OP2_EN	OP1_SL	OP1_EN	SEG48/ PC[7]	PWM4/ PC[7]	SEG47/ PC[6]	PWM5/ PC[6]

**TM0: 8-bit Timer (TONE & FREQ out)**

Name	Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>TM0_CTL</b>	\$17	EN	WR_CNT	DATA	IRQ_S	SUR1	SUR0	DUTY1	DUTY0
<b>TM0_LA</b>	\$18	D7	D6	D5	D4	D3	D2	D1	D0
<b>TM0_CNT</b>	\$19	D7	D6	D5	D4	D3	D2	D1	D0
<b>TONE_CTL1</b>	\$39	EN	PH15E	PH14E	PH13E	PH12E	PH11E	PAT1	INV12
<b>TONE_CTL2</b>	\$3A						CRY2	CRY1	CRY0

**TM23 : 8-bit Timer x2 , 8-bit capture x2**

Name	Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>TM2_CTL1</b>	\$1F	EN	WR_CNT	BIT	MOD1	MOD0	EDGE	SUR1	SUR0
<b>TM2_CTL2</b>	\$20	ENC	CLR_CNT		CAPIN1/ RFC_T1	CAPIN0/ RFC_T0	INT_S	PWM_OS	OV
<b>TM2_LA</b>	\$21	D7	D6	D5	D4	D3	D2	D1	D0
<b>TM2_CNT</b>	\$22	D7	D6	D5	D4	D3	D2	D1	D0
<b>TM3_CTL1</b>	\$23	EN	WR_CNT		MOD1	MOD0	EDGE	SUR1	SUR0
<b>TM3_CTL2</b>	\$24	ENC	CLR_CNT		CAPIN1/ RFC_T1	CAPIN0/ RFC_T0	INT_S	PWM_OS	OV
<b>TM3_LA</b>	\$25	D7	D6	D5	D4	D3	D2	D1	D0
<b>TM3_CNT</b>	\$26	D7	D6	D5	D4	D3	D2	D1	D0

**Interrupt Control**

Name	Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>IRQM_CTL</b>	\$2F	INTM				SPIM	TM5M/ PWM5M/ CAPT5M/ RFC5M	TM4M/ PWM4M/ CAPT4M/ RFC4M	CMPM
<b>IRQM</b>	\$31	--	PACM	PINTM	2HZM	PHM	TM3M/ PWM3M/ CAPT3M/	TM2M/ PWM2M/ CAPT2M/	TM0M/ TONEM

							RFC3M	RFC2M	
CPU_RESUME	\$30	--	PACR	PINTR	2HZR	PHR	TM3R/ PWM3R/ CAPTR/ RFC3R	TM2R/ PWM2R/ CAPTR/ RFC2R	TM0R/ TONER
CPU_RESUME2	\$2A					SPIR	TM5R/ PWM5R/ CAPT5R/ RFC5R	TM4R/ PWM4R/ CAPT4R/ RFC4R	CMPR
IRQF	\$32	--	PACF	PINTF	2HZF	PHF	TM3F/ PWM3F/ CAPT3F RFC3F	TM2F/ PWM2F/ CAPT2F/ RFC2F	TM0F/ TONEF
IRQF2	\$2B	--				SPIF	TM5F/ PWM5F/ CAPT5F RFC5F	TM4F/ PWM4F/ CAPT4F/ RFC4F	CMPF
Other									
LBASDT	\$33	LCD1	LCD0	FRAM1	FRAM0	--	DUTY2	DUTY1	DUTY0
STROBE	\$34	FRAME	EN	KOAEN	KOEN	KO3	KO2	KO1	KO0
LCD_CTL	\$35	PUMP1	PUMP0	POW1	POW0	OVP1	OVP0	LCDM1	LCDM0
PH_CTL	\$36	ELON	EL_SEL	EL_P	CLR	PH_I3	PH_I2	PH_S3	PH_S2
PH2_CTL	\$27	--	--	--	--	--	--	PH_S5	PH_S4
PH_OUT	\$37	PH15	PH14	PH13	PH12	PH11	PH10	PH9	PH8
PH_OUT1	\$38	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0
WDT_CTL	\$3B	WDTEN	--	--	--	--	PRE2	PRE1	PRE0
WBANK	\$2E	WKMB3	WKMB2	WKMB1	WKMB0	--	SPEC	DP1	DP0
TAB_BNK	\$3D	--	--	TBA5	TBA4	TBA3	TBA2	TBA1	TBA0
SYS_CTL	\$3E	CLKS	HALT	IRC	LVD1	LVD0	LV	STP1	STP0
ACC	\$3F	D7	D6	D5	D4	D3	D2	D1	D0

## 2.5.1 Special Purpose Register 2

The listed register table is as below, we will describe them in detail at the specific chapter.

Refer to WBANK register. (WBANK bit2, SPEC=1 )

Name	Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>TM45 : 8-bit Timer x2 , 8-bit capture x2</b>									
Name	Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<b>TM4_CTL1</b>	<b>\$78</b>	EN	WR_CNT	BIT	MOD1	MOD0	EDGE	SUR1	SUR0
<b>TM4_CTL2</b>	<b>\$79</b>	ENC	CLR_CNT		CAPIN1/ RFC_T1	CAPIN0/ RFC_T0	INT_S	PWM_OS	OV
<b>TM4_LA</b>	<b>\$7A</b>	D7	D6	D5	D4	D3	D2	D1	D0
<b>TM4_CNT</b>	<b>\$7B</b>	D7	D6	D5	D4	D3	D2	D1	D0
<b>TM5_CTL1</b>	<b>\$7C</b>	EN	WR_CNT		MOD1	MOD0	EDGE	SUR1	SUR0
<b>TM5_CTL2</b>	<b>\$7D</b>	ENC	CLR_CNT		CAPIN1/ RFC_T1	CAPIN0/ RFC_T0	INT_S	PWM_OS	OV
<b>TM5_LA</b>	<b>\$7E</b>	D7	D6	D5	D4	D3	D2	D1	D0
<b>TM5_CNT</b>	<b>\$7F</b>	D7	D6	D5	D4	D3	D2	D1	D0
<b>SPI control</b>									
<b>SPI_CTL</b>	<b>\$75</b>	EN	CPOL	CPHA	SWAP	LSBF	MODE	CLK1	CLK0
<b>SPI_TX</b>	<b>\$76</b>	D7	D6	D5	D4	D3	D2	D1	D0
<b>SPI_RX</b>	<b>\$77</b>	D7	D6	D5	D4	D3	D2	D1	D0
<b>LCD control</b>									
<b>LBASDT2</b>	<b>\$72</b>	<b>LCD2</b>							
<b>Port B control</b>									
<b>PB_DIR</b>	<b>\$70</b>	DF7	DF6	DF5	DF4	DF3	DF2	DF1	DF0
<b>PB_PUD</b>	<b>\$6F</b>	UF7	UF6	UF5	UF4	UF3	UF2	UF1	UF0
<b>PB_DAT</b>	<b>\$6E</b>	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
<b>PAD_CTL9</b>	<b>\$71</b>	SEG40/ PB[7]	SEG39/ PB[6]	SEG38/ PB[5]	SEG37/ PB[4]	SEG36/ PB[3]	SEG35/ PB[2]	SEG34/ PB[1]	SEG33/ PB[0]
<b>PF_PUD</b>	<b>\$73</b>	UF7	UF6	UF5	UF4	UF3	UF2	UF1	UF0
<b>PF_DAT</b>	<b>\$74</b>	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0

## 2.6 HALT mode Function

HALT function is used to minimize the power consumption of CPU in standby. User can set register \$3Eh bit 6 to enter HALT mode in below table. During this stage, CPU operation is off which means the program memory is not in working. Only slow clock, timer and LCD driver blocks are in operation.

### SYS\_CTL (\$3Eh)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SYS_CTL	CLKS	HALT	IRC	LVD1	LVD0	LV	STP1	STP0

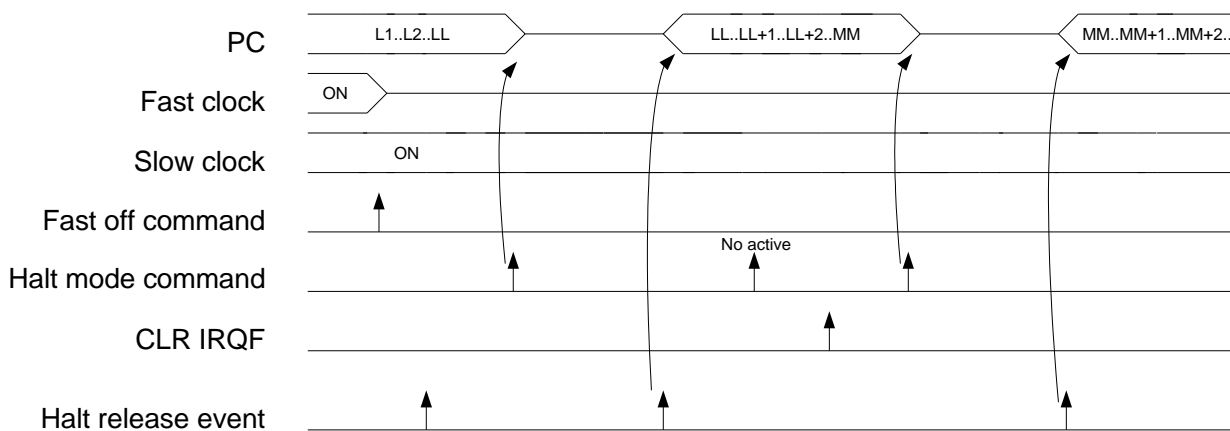
Bit	Symbol	Description	
7	CLKS	Clock select 0: Slow clock 1: Fast clock	
6	HALT	CPU on/off control 0: ON 1: CPU OFF	
5	IRC	Internal fast clock 0: 700 Khz (default) 1: 1.5 Mhz	
4~3	LVD1~0	Low voltage detector	
		1 1	ON (2.56)
		1 0	ON (2.42)
		0 1	ON (2.68)
		0 0	Function OFF
2	LV	Low power output 0: Power voltage > 2.56V (or 2.68V,2.42V) 1: Power voltage < 2.56V (or 2.68V,2.42V)	
1	STP1	Fast clock control 0: ON 1: OFF	
0	STP0	Slow clock control 0: ON 1: OFF	

There are several events can release the HALT mode which are:

- (1) Pin change wake up (External interrupt pin PA7~0 ,PC7~0 , PD5 )
- (2) Capture mode (CAPT1A ,CAPT1B,CAPT2A ,CAPT2B,CAPT4 & CAPT5 )
- (3) Timer interrupt ( PH, 2Hz, TMR0 ,TMR2 ,TMR3, TMR4 & TMR5 )
- (4) Watchdog timer
- (5) Reset

The diagram is as below:

**Fig.2.6.1 HALT Release**



**Fig.2.6.2 Halt mode & Halt Release waveform**

**2.6.1 halt mode example.**

```

#include "MK9A160P.INC" ; halt mode ,2hz wake-up
ORG 0x00
    LGOTO INITIAL

    ORG 004
    MOVLA B'01101111'
    MOVAM IRQF ;Clear PH 2Hz interrupt flag
    INC PD_DAT,m
    INC PA_DAT,m
    IRETI

    ORG 0x100
INITIAL
    CLR PA_DAT ; Clear floating
    CLR PD_DAT ; Clear floating
    
```

```

MOVLA    0x00          ; Set PA is output pin
MOVAM    PA_DIR
MOVLA    0xFF
MOVAM    PA_PUD1      ; Set Pa is normal output pin
MOVAM    PA_PUD2
MOVAM    PD_PUD1      ; Set PD is normal output pin
MOVAM    PD_PUD2
MOVLA    B'00000111'   ;frame=42hz, com1~10
MOVAM    LBASDT
MOVLA    B'00110010'   ;b5.4=11,low power ,lcd ON
MOVAM    LCD_CTL
CLR      IRQF          ; Clear interrupt flag
MOVLA    B'00010000'   ; Setup 2HZM interrupt
MOVAM    IRQM
BS       IRQM_CTL,7    ; Enable interrupt
BC       SYS_CTL,1     ;; FCLK ON
NOP
NOP
BS       SYS_CTL,7     ;; CPU CLK=FCLK
.....
BC       SYS_CTL,7     ;; CPU CLK= SCLK
LOOP
.....
BS       SYS_CTL,1     ;; FCLK turn OFF
                        ;; decrease power consumption
BS       SYS_CTL,6     ;; Setup HALT ; OSC is active but cpu be turned off.
                        ;; If wake-up ,only the system clock would be turn on
                        ;; no loading & 32k only , Idd=2.5uA (halt mode)
.....
LGOTO   LOOP
END

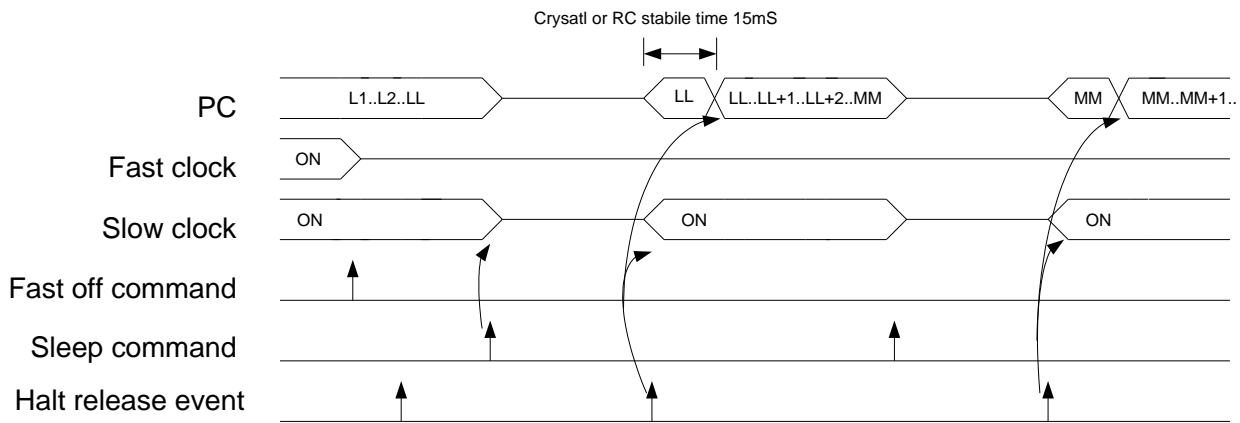
```



## 2.7 SLEEP Function

When system enter the sleep mode by using the instruction SLEEP, all the clocks and circuits (include LCD ) will stop operation except watchdog timer and pin change wake up circuit. During this mode, the current consumption is almost zero. Only two events can wake up from SLEEP mode, that are:

- (1) Pin change wake up (External interrupt pin PA7~0 ,PC7~0 ,PD5 )
- (2) Capture mode (CAPT1A ,CAPT1B,CAPT2A,CAPT2B,CAPT4 & CAPT5 )
- (3) Watchdog timer
- (4) Reset



**Fig.2.7.1 Sleep & Halt Release waveform**

## 2.8 Table Look-up Function

The MK9A160P provide table look-up function. The look-up tables can be placed at any location in the ROM space. The instruction of TABRDL is to read low byte of ROM table. And The TABRDH is to read high byte. The register of TAB\_BNK and PC7~0 are used to define starting address of table.

### TAB\_BNK (\$3Dh)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TBH	--	--	--	TBA4	TBA3	TBA2	TBA1	TBA0

- Bit4~0 (TBA4~0): High byte table location bits

### <Example>

The program ROM address \$700h~\$71F OTP data are moved to working RAM (64 bytes).

**Fig.2.8.1 Table read example**

```

#include "MK9A160P.INC"           ; move table data to working ram (80h~BFh)
#define RAM_INDEX C0H
#define RAM_DATA C1H
ORG 0x00
    LGOTO    INITIAL
ORG 0x04
    CLR     IRQF           ; Clear interrupt flag
    IRETI
    ORG    0x20
INITIAL
    CLR     RAM_INDEX
    CLR     RAM_DATA
    MOVLA  0Ah
    MOVAM  TAB_BNK       ; counter
    MOVLA  080h
    MOVAM  FSR           ; working RAM 80
TAB:    TABRDH  RAM_INDEX
        NOP
        MOVAM  IAR
        MOV    IAR,a
        INC    FSR,m
        TABRDL RAM_INDEX
        NOP
        MOVAM  IAR
        MOV    IAR,a
        INC    RAM_INDEX,m

```

```

INC      FSR,m
BTSS    RAM_INDx,6  ;; A0 ~
LGOTO   TAB
NOP
sleep
ORG   0A00h
DW      0001h
DW      0203h
DW      0405h
DW      0607h
DW      0809h
DW      0A0Bh
DW      0C0Dh
DW      .....
    
```

## 2.9 FSR: Bank Select Register

This register will be used with INDF register for indirect addressing data memory.

### FSR (\$04h)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FSR	BANK1	BANK0	D5	D4	D3	D2	D1	D0

Bit	Symbol	Description	
7~6	BANK1~0	RAM bank select	
		0 0	Special purpose register
		0 1	Display RAM
		1 0	Direct access working RAM (80h~BFh)
		1 1	Direct access working RAM (C0h~DFh)

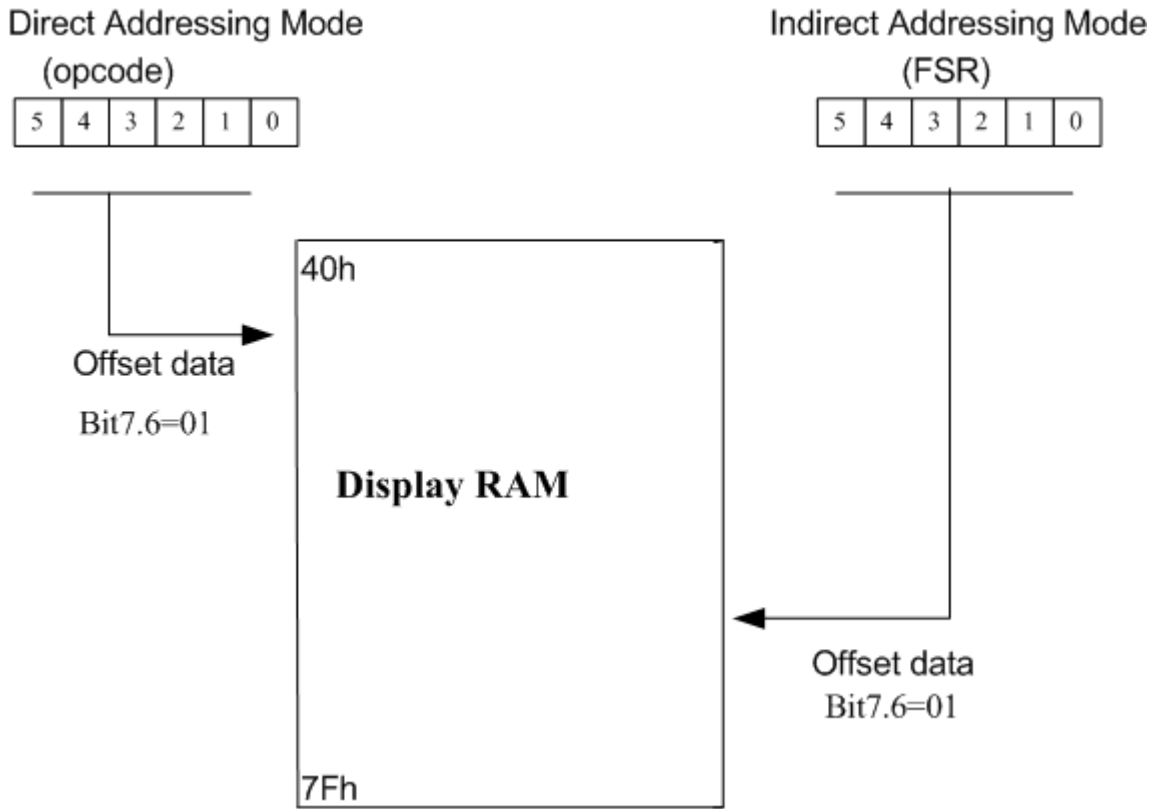


Fig.1.9.1 Display RAM

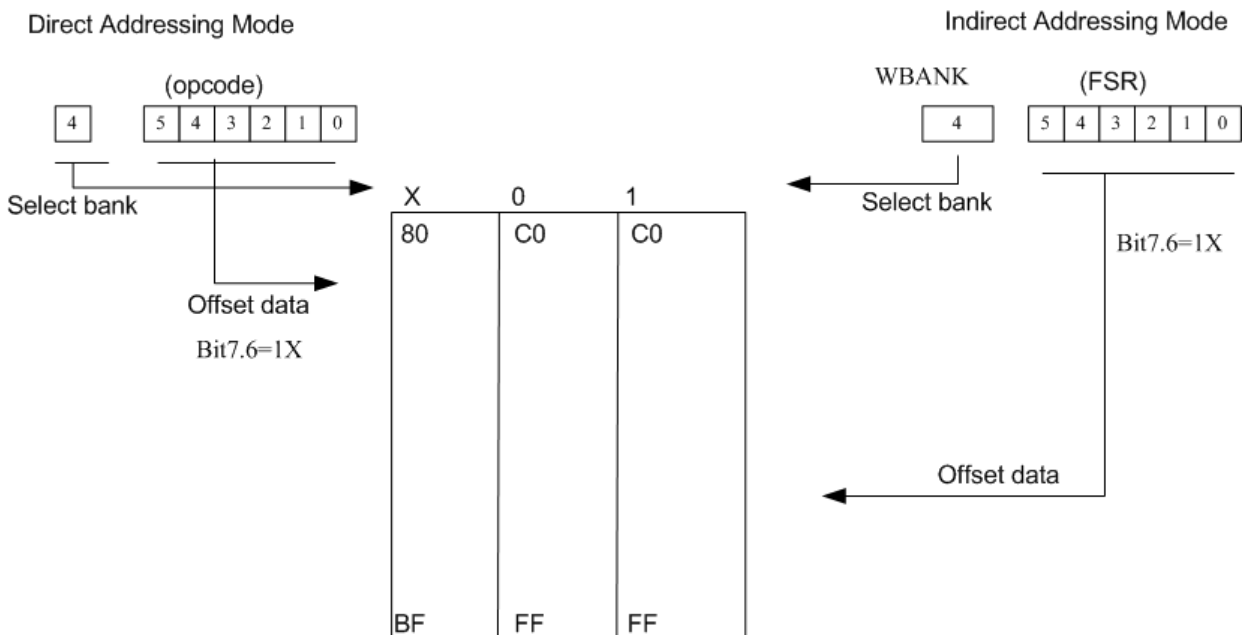


Fig.1.9.2 Working RAM

## 2.10 WBANK : RAM Bank control Register

This register will be used with INDF register for indirect addressing data memory.

### WBANK (\$2Eh)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WBANK	WKMB3	WKMB2	WKMB1	WKMB0	--	SPEC	DP1	DP0

- Bit4 (WKMB0): Working RAM bank select

Bit2~0	Bank RAM
0 0 0	Display RAM BANK 1 (64 bytes)
0 0 1	Display RAM BANK 2 (64 bytes)
0 1 0	Reserved
0 1 1	Reserved
1 X X	Special register (2)

- Bit7~4 (WKMB3~0): Working RAM bank select

Bit4		Bank RAM
BANK1-0	WKMB3-0	
1 1	0 0 0 0	BANK 0 (64 bytes)
1 1	0 0 0 1	BANK 1 (64 bytes)
1 1	0 0 1 0	BANK 2 (64 bytes)
1 1	0 0 1 1	BANK 3 (64 bytes)
1 1	0 1 0 0	BANK 4 (64 bytes)
1 1	0 1 0 1	BANK 5 (64 bytes)
1 1	0 1 1 0	BANK 6 (64 bytes)
1 1	0 1 1 1	BANK 7 (64 bytes)
1 1	1 0 0 0	BANK 8 (64 bytes)
1 1	1 0 0 1	BANK 9 (64 bytes)
1 1	1 0 1 0	BANK 10 (64 bytes)
1 1	1 0 1 1	BANK 11 (64 bytes)
1 1	1 1 0 0	BANK 12 (64 bytes)
1 1	1 1 0 1	BANK 13 (64 bytes)
1 1	1 1 1 0	BANK 14 (64 bytes)
1 1	1 1 1 1	BANK 15 (64 bytes)

## 2.11 Status Register

The STATUS register is an 8-bit register that contains the zero flag (Z), carry flag (C), Nibble carry flag (DC), power down flag ( $\overline{PD}$ ), and watchdog timer overflow flag ( $\overline{TO}$ ). It records the status information.

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
STATUS	--	CPU_VDD	VOLT	$\overline{TO}$	$\overline{PD}$	Z	DC	C

- Bit4 ( $\overline{TO}$ ): Timer overflow flag bit
- Bit3 ( $\overline{PD}$ ): Power down flag bit

$\overline{TO}$	$\overline{PD}$	Description
0	0	WDT timer overflow (or 4'key reset) from sleep mode
0	1	WDT timer overflow (or 4'key reset) from normal mode
1	0	Input a "low" at RESETB from sleep mode Sleep instruction
1	1	Power on reset CLRWDT instruction
Unchanged	Unchanged	Input a "low" at RESETB from normal mode

- Bit2 (Z): zero flag bit
  - 0: the result of a logic operation is not zero
  - 1: the result of a logic operation is zero
- Bit1 (DC): Nibble Carry and Nibble  $\overline{Borrow}$  flag bit
  - ADD instruction:
    - 0: no carry
    - 1: a carry from the low nibble bits of the result occurred
  - SUB instruction
    - 0: a borrow from the low nibble bits of the result occurred
    - 1: no borrow
- Bit0 (C): Carry and  $\overline{Borrow}$  flag bit
  - ADD instruction:
    - 0: no carry
    - 1: a carry occurred from the MSB
  - SUB instruction
    - 0: a borrow occurred from the MSB
    - 1: no borrow

Bit	Symbol	Description	
6	CPU_VDD	CPU_VDD: Internal CORE voltage switch ( 3V mode ) 1: 1.5V 0: 3V	
5	VOLT	VOLT: I/O PAD voltage ( read only ) 1: 3V (fixed ) 0: 1.5V	
4	$\overline{TO}$	Time out flag bit: 1: after power-on or by the CLRWDT or SLEEP instruction 0: Occur WDT time-overflow	
3	$\overline{PD}$	Power down flag bit: <sup>(Note2)</sup> 1: after power-on or by the CLRWDT instruction 0: execute SLEEP instruction	
2	Z	Zero bit: 1: the result of a logic operation is zero 0: the result of a logic operation is not zero	
1	DC	Nibble Carry and Nibble $\overline{Borrow}$ bit	
		ADD instruction	SUB instruction
		1: a carry from the low nibble bits of the result occurred 0: no carry	1: no borrow 0: a borrow from the low nibble bits of the result occurred
0	C	Carry and $\overline{Borrow}$ bit:	
		ADD instruction	SUB instruction
		1: a carry occurred from the MSB 0: no carry	1: no borrow <sup>(Note1)</sup> 0: a borrow occurred from the MSB

## 2.12 PCH & PCL:

### PCH (\$02h)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCH	--	--	A13	A12	A11	A10	A9	A8

### PCL (\$01h)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCL	A7	A6	A5	A4	A3	A2	A1	A0

The MK9A160P has an 12-bits program counter (PC) that includes PCL (8-bits) and PCH (4-bits). PC is stored the routing of program. If user changes the value of PCL, then program will jump to the indicated location.

Ex1: PCH=01H, PCL=02H+10H=12H, the program will jump to PC=112H.

Ex2: PCH= 01H, PCL=F0H+30H=20H with carry 1, the program will jump to PC=220H but PCH still be 01H.

<Note>

- (a) When execute IRET and IRETI, PCH data would not be updated
- (b) When execute RETLW, LGOTO, LCALL and RET, PCH would be updated .
- (c) PCH would be updated after mathematic operation and PC[8] is changed.

<Example>.

#### Fig.2.11.1 PCL & PCL control example

The below program is show how PCL and PCH working with direct mathematic.

```
#DEFINE PCL 01H ; Define address 01H of RAM named PCL
#DEFINE PCH 02H ; Define address 02H of RAM named PCH

ORG 00
LGOTO START
ORG 1C0h
START: MOVLA 02h
MOVAM PCH
MOVLA 33H
MOVAM PCL
NOP
NOP
NOP
NOP
ORG 233h
LGOTO A1
```



---

	NOP	
A1:	ORG	2FCh
	MOVLA	04h
	MOVAM	PCH
	MOVLA	88H
	MOVAM	PCL
	NOP	
	NOP	
	MOVLA	80h
	ADD	PCL,m
	NOP	
	NOP	
	MOVLA	A0h
	SUB	PCL,m
	NOP	
	NOP	
	MOVLA	020h
	NOP	
	ORG	36Ch
	NOP	
	MOVLA	022h
	NOP	
	ORG	370
	NOP	
	MOVLA	024h
	NOP	
	NOP	
	NOP	
	ORG	388h
	NOP	
	MOVLA	026h
	NOP	
	MOVLA	80h
	ADD	PCL,m
	NOP	
	NOP	
	ORG	40Dh
	NOP	

```

MOVLA 028h
NOP
MOVLA 0A0h
SUB PCL,m
NOP
MOVLA 02Ah
NOP
end

```

Fig.2.11.2 Fig2.11.1 program flow

The below program is show how PCL and PCH working with direct mathematic.

Current PC	ORG	1C0h	PC address after instruction is executed
1C0	MOVLA	02h	; PC=1C1H, PCL =C1H, PCH=00H
1C1	MOVAM	PCH	; PC=1C2H, PCL =C2H, <b>PCH=02H.</b>
1C2	MOVLA	33H	; PC=1C3H, PCL =C3H, PCH=02H.
1C3	MOVAM	PCL	; PC=233H, PCL =33H, PCH=02H.
			; The program will jump to <b>PC=233H</b>
233	LGOTO	A1	; PC=2FCH, PCL =FCH, PCH=02H.
2FC	A1: MOVLA	04h	; PC=2FDH, PCL =FDH, PCH=02H
2FD	MOVAM	PCH	; PC=2FEH, PCL =FEH, <b>PCH=04H.</b>
	MOVLA	88H	
2FE			; PC=300H, PCL =C0H, PCH=04H. when PCH=FFh → 00h ,PCH ← PC[11:8] ;; ERROR !!
2FF	MOVAM	PCL	; PC=388H, PCL =88H, <b>PCH=03H.</b> ( PCH ← PC[11:8]+1 ) ; The program will jump to PC=388H
388	NOP		; PC=389H, PCL =89H, PCH=03H.
389	MOVLA	26h	; PC=38AH, PCL =8AH, PCH=03H.
38A	NOP		; PC=38BH, PCL =8BH, PCH=03H.
38B	MOVLA	80h	; PC=38CH, PCL =8CH, PCH=03H.
38C	ADD	PCL	; PC=40DH, <b>PCL =0DH, PCH=04H.</b>
40D	NOP		; PC=40EH, PCL =0EH, PCH=04H.
40E	MOVLA	28h	; PC=40FH, PCL =0FH, PCH=04H.
40F	NOP		; PC=410H, PCL =10H, PCH=04H.
410	MOVLA	A0h	; PC=411H, PCL =11H, PCH=04H.
411	SUB	PCL	; PC=372H, <b>PCL =72H, PCH=03H.</b>
372	NOP		; PC=373H, PCL =73H, PCH=03H.
373	NOP		

## 2.13 Reset

There are 4 events will cause reset which is listed as below. The power-down event will cause MK9A160P reset and the detected voltage is according to the bit7~bit6 in the CONFIG register. This condition is used to protect chip in deficient power environment. The last two cases are called warm reset. The different reset events will affect registers and RAM. The  $\overline{TO}$  and  $\overline{PD}$  bits can be used to determine the type of reset.

- (1) Power-on reset. (Cold reset )
- (2) Low voltage reset (LVR) (Cold reset ) .
- (3) RESETB pin reset (input a negative pulse). (Warm reset )
- (4) 4'S KEY reset (Warm reset )
- (5) WDT timer overflow reset . (Warm reset )

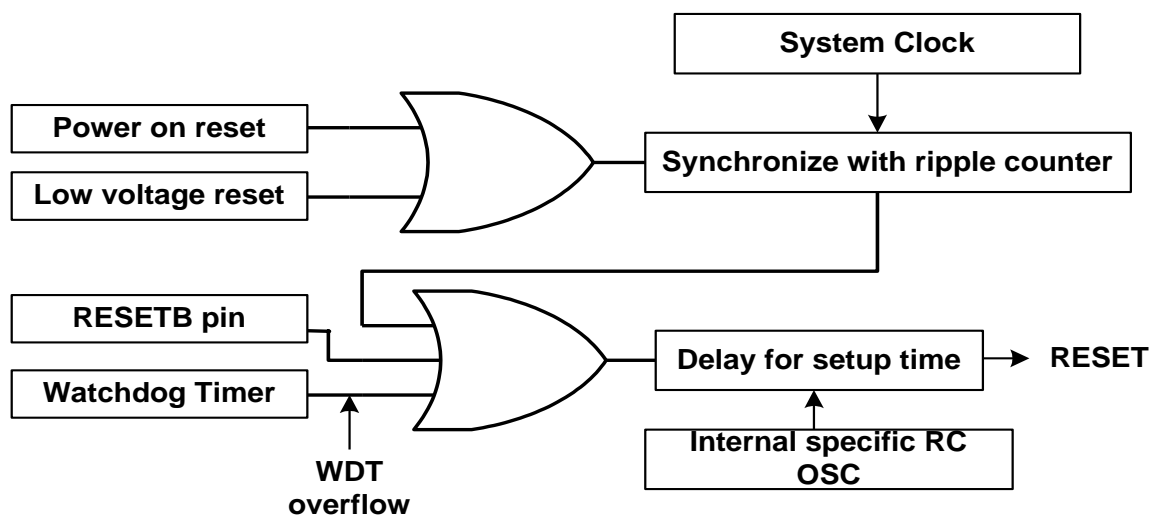


Fig.2.12.1 Reset Diagram

<Note>: the watchdog setup time is approximately 20ms that will has some tolerance due to power voltage, process and temperature variations. Setup time

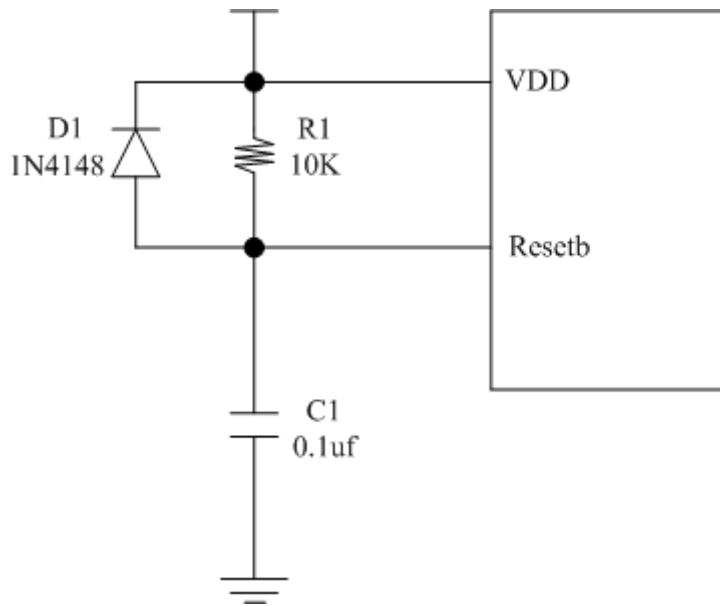


Fig.2.12.2 Reset Circuit

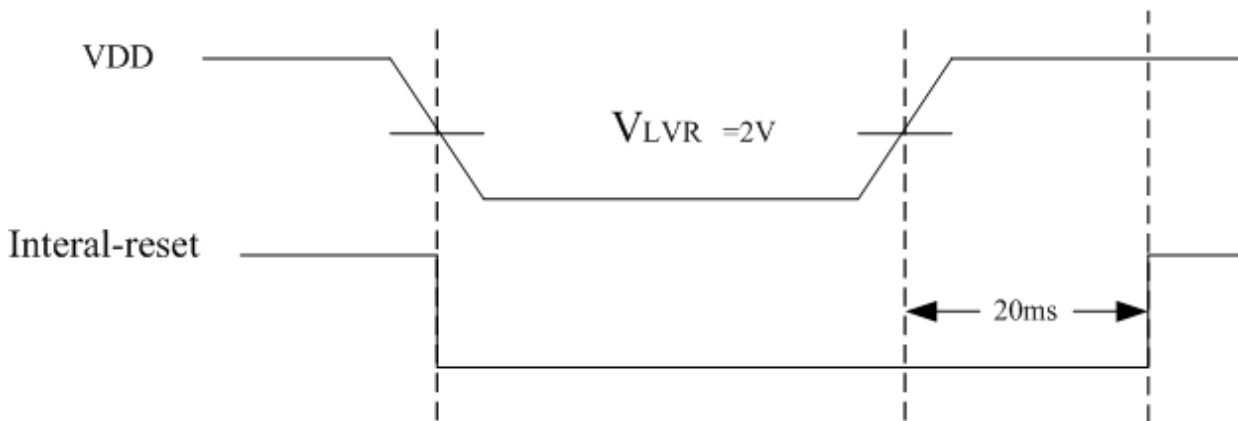


Fig.2.12.3 LVR ON

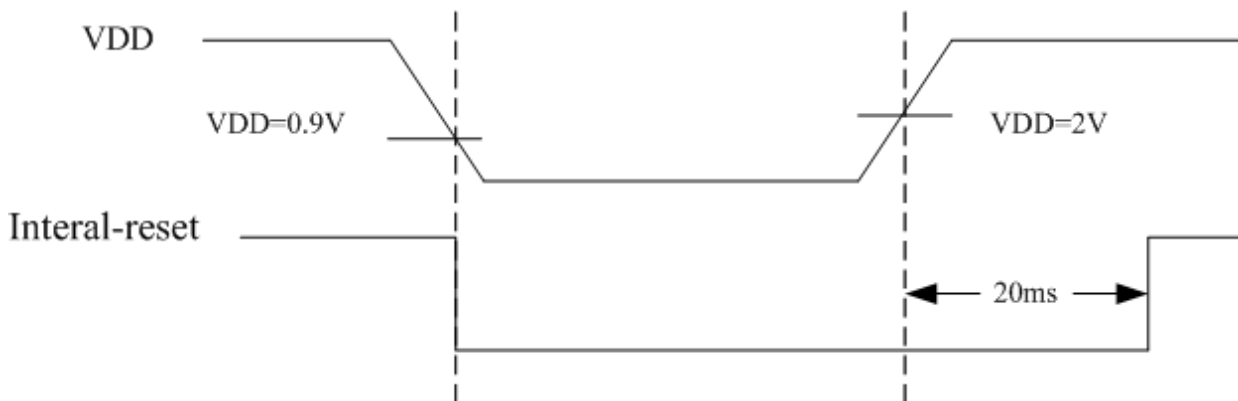


Fig.2.12.4 LVR OFF

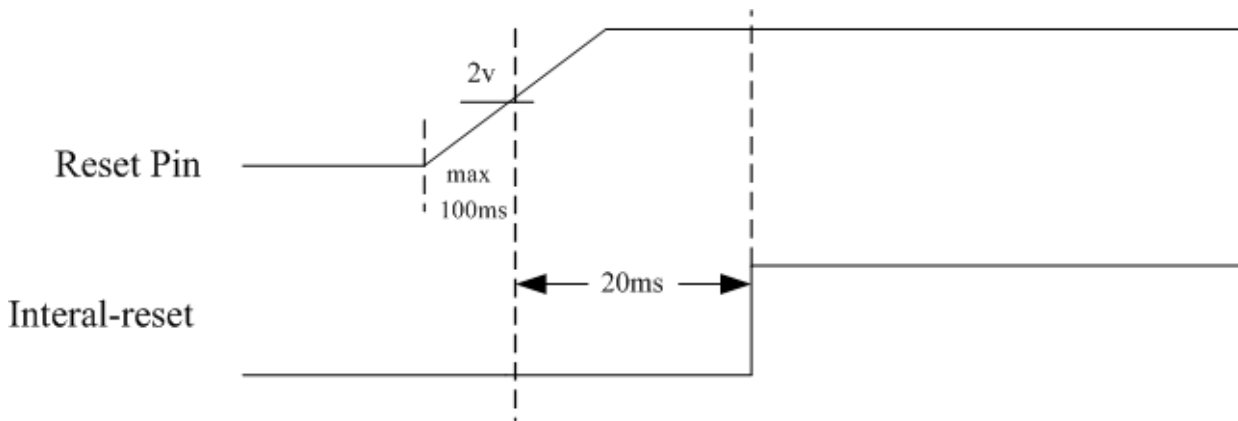


Fig.2.12.5 External Reset -- RESETB PIN

The default value during different reset condition

Address	Name	Cold Reset	Warm Reset
00H	INDF	0000 0000	0000 0000
01H	PCL	0000 0000	0000 0000
02H	PCH	--- 0 0000	--- 0 0000
03H	STATUS	0001 1XXX	0001 1PPP
04H	FSR	0000 0000	0000 0000
05H	PA_DIR	0111 1111	0111 1111
07H	PA_WAKE_UP	0000 0000	0000 0000
2DH	PA_EDGE	0000 0000	0000 0000
08H	PA_PUD1	0000 0000	0000 0000
09H	PA_PUD2	0000 0000	0000 0000
0AH	PA_DAT	XXXX XXXX	PPPP PPPP
1DH	PC_WAKE_UP	0000 0000	0000 0000
1EH	PC_EDGE	0000 0000	0000 0000
0BH	PC_CTL	0000 0000	0000 0000
0CH	PC_DIR	1111 1111	1111 1111
0DH	PC_PUD	0000 0000	0000 0000
0EH	PC_DAT	0000 0000	0000 0000
0FH	PD_DIR	1111 1111	1111 1111
10H	PD_PUD	0000 0000	0000 0000
11H	PD_CTL	0000 0000	0000 0000
12H	PD_DAT	0000 0000	0000 0000
1AH	PE_DIR	1111 1111	1111 1111

1BH	PE_PUD	0000 0000	0000 0000
1CH	PE_DAT	0000 0000	0000 0000
2CH	PF_DIR	1111 1111	1111 1111
73H	PF_PUD	0000 0000	0000 0000
74H	PF_DAT	0000 0000	0000 0000
13H	PAD_CTL1	0000 0000	0000 0000
14H	PAD_CTL2	0000 0000	0000 0000
15H	PAD_CTL3	0000 0000	0000 0000
16H	PAD_CTL4	0000 0000	0000 0000
28H	PAD_CTL5	0000 0000	0000 0000
29H	PAD_CTL6	0000 0000	0000 0000
06H	PAD_CTL8	0000 0000	0000 0000
17H	TM0_CTL	0000 0000	0000 0000
18H	TM0_LA	0000 0000	0000 0000
19H	TM0_CNT	0000 0000	0000 0000
39H	TONE_CTL1	0000 0000	0000 0000
3AH	TONE_CTL2	0000 0000	0000 0000
1FH	TM2_CTL1	0000 0000	0000 0000
20H	TM2_CTL2	0000 0000	0000 0000
21H	TM2_LA	0000 0000	0000 0000
22H	TM2_CNT	0000 0000	0000 0000
23H	TM3_CTL1	0000 0000	0000 0000
24H	TM3_CTL2	0000 0000	0000 0000
25H	TM3_LA	0000 0000	0000 0000
26H	TM3_CNT	0000 0000	0000 0000
78H	TM4_CTL1	0000 0000	0000 0000
79H	TM4_CTL2	0000 0000	0000 0000
7AH	TM4_LA	0000 0000	0000 0000
7BH	TM4_CNT	0000 0000	0000 0000
7CH	TM5_CTL1	0000 0000	0000 0000
7DH	TM5_CTL2	0000 0000	0000 0000
7EH	TM5_LA	0000 0000	0000 0000
7FH	TM5_CNT	0000 0000	0000 0000
2EH	WBANK	0000 0000	0000 0000
2FH	IRQM_CTL	0000 0000	0000 0000
30H	CPU_RESUME	0000 0000	0000 0000

2AH	CPU_RESUME2	0000 0000	0000 0000
31H	IRQM	0000 0000	0000 0000
32H	IRQF	0000 0000	0000 0000
2BH	IRQF2	0000 0000	0000 0000
33H	LBASDT	0101 0010	0101 0010
34H	STROBE	0000 0000	0000 0000
35H	LCD_CTL	0000 0000	0000 0000
36H	PH_CTL	0000 0000	0000 0000
27H	PH2_CTL	0000 0000	0000 0000
37H	PH_OUT	XXXX XXXX	PPPP PPPP
38H	PH_OUT1	XXXX XXXX	PPPP PPPP
3BH	WDT_CTL	1000 0111	1000 0111
3DH	TAB_BNK	0000 0000	0000 0000
3EH	SYS_CTL	0000 0010	0000 0010
3FH	ACC (Accumulator)	XXXX XXXX	PPPP PPPP
75H	SPI_CTL	0000 0000	0000 0000
76H	SPI_TX	0000 0000	0000 0000
77H	SPI_RX	0000 0000	0000 0000

X: unknown;

?: value depends on condition ;

P: previous data;

-:unimplemented and read as"0".

### 3. Timer and Capture

#### 3.1 16 bit Pre-divider

This 16 bit pre-divider can generate 2Hz timer interrupt which can provide an accurate 0.5sec timer base. By the way, some of this pre-divider timer would become time base of many other blocks, i.e. LCD driver, timer and capture. There are two registers which used to indicate each stage status of flip-flop of 16bit divider from PH which can be a time base for synchronization. These two registers can be read out.

**PH\_OUT (\$37h) & PH\_OUT1(\$38h): (Read only)**

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PH_OUT	PH15	PH14	PH13	PH12	PH11	PH10	PH9	PH8
PH_OUT1	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PH_OUT	1hz	2hz	4hz	8hz	16hz	32hz	64hz	128hz
PH_OUT1	256hz	512hz	1K	2K	4K	8K	16K	32K

#### 3.2 Timer 0 (TM0)

Timer 0 has two functions in this chip, one is for general purpose 8 bit timer, the other is to be the time base of FREQ or Tone . TM0 has two buffers, TM0\_LA and TM0\_CNT. Before timer 0 starting to count, user should write counter value to TM0\_LA. If WR\_CNT (Bit 6 of TM0\_CTL) is set to "1", then the data will automatically download to TM0\_CNT. This setting is need in the first time. After timer is counting and overflow is occurred, then it will generate interrupt and auto reload function will reload the TM0\_LA data to TM0\_CNT. The block is like below : (Fig.3.2.1)

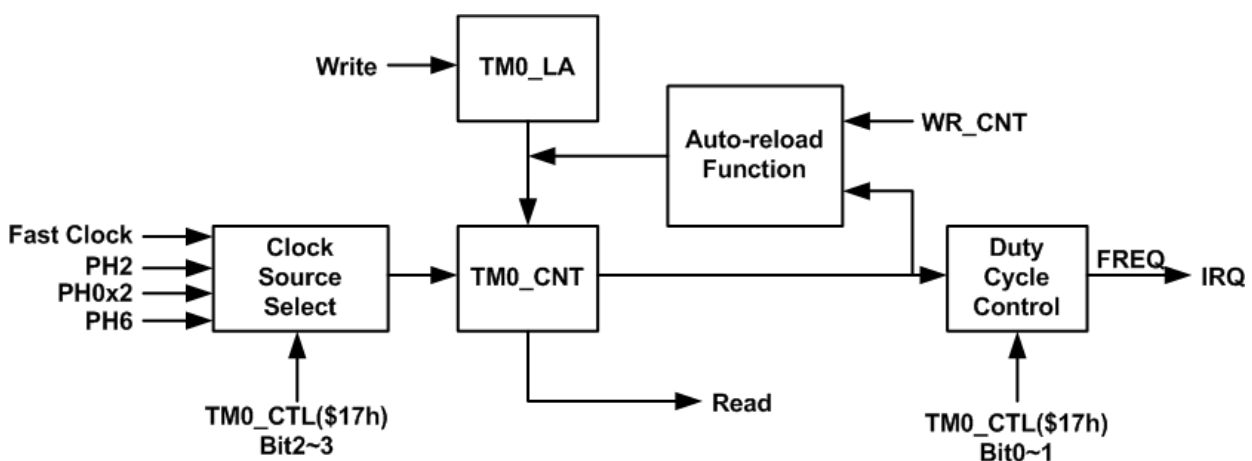
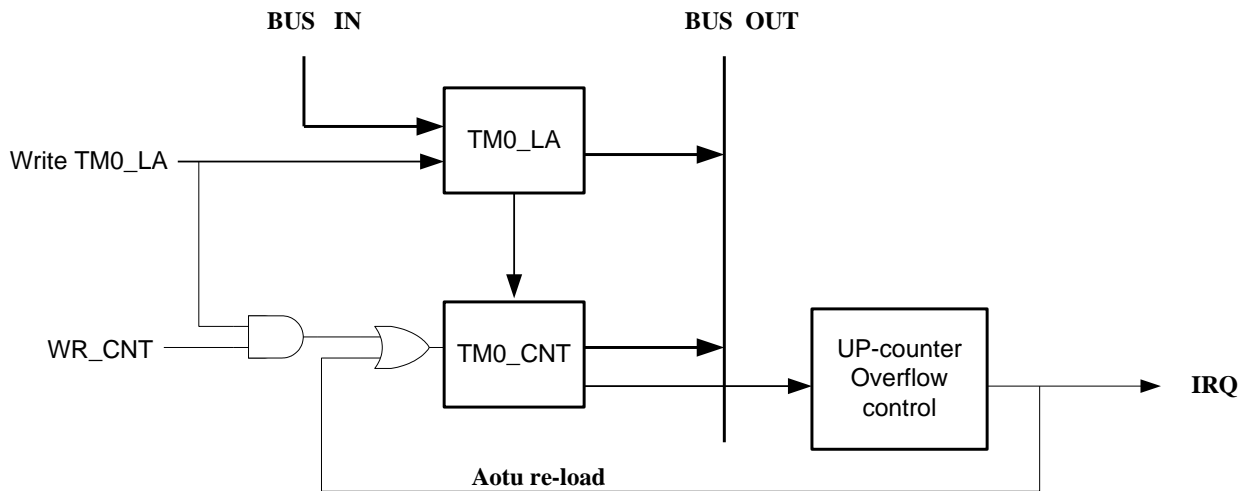


Fig.3.2.1 Timer 0 (TM0) Block Diagram



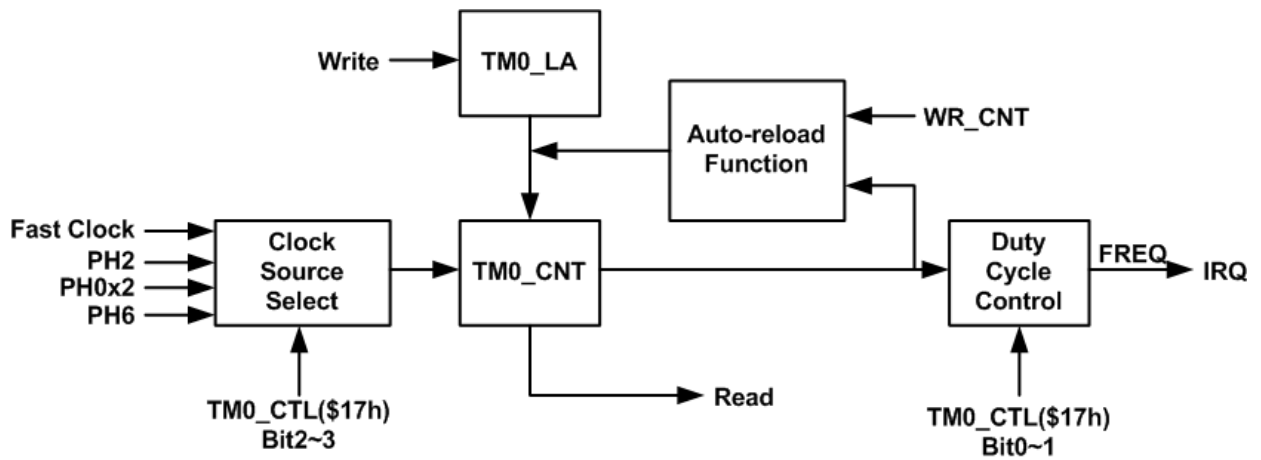


**Fig.3.2.2 Timer 0 WR\_CNT & auto-reload control**

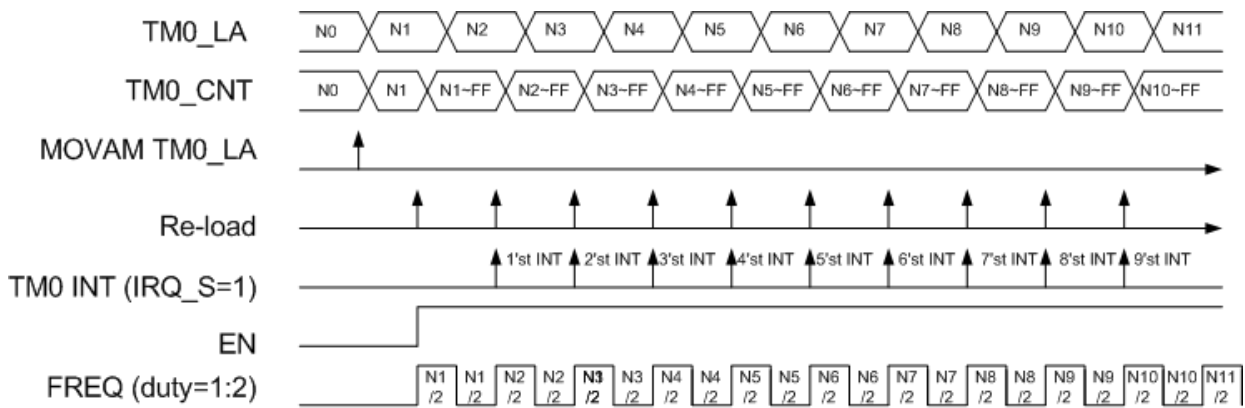
If TM0 is used as general purpose 8 bit timer, user can select the clock source by setting TMO\_CTL(\$17h) register bit 2~3. The default value of duty cycle control block is (0,0) which means duty is 1:1. Because TM0 would be used to generate FREQ or Tone , the duty cycle may change to another value. Once user will switch among these functions, please take care these two bit data when use TM0 as general timer again.

### 3.3 *FREQ and Tone*

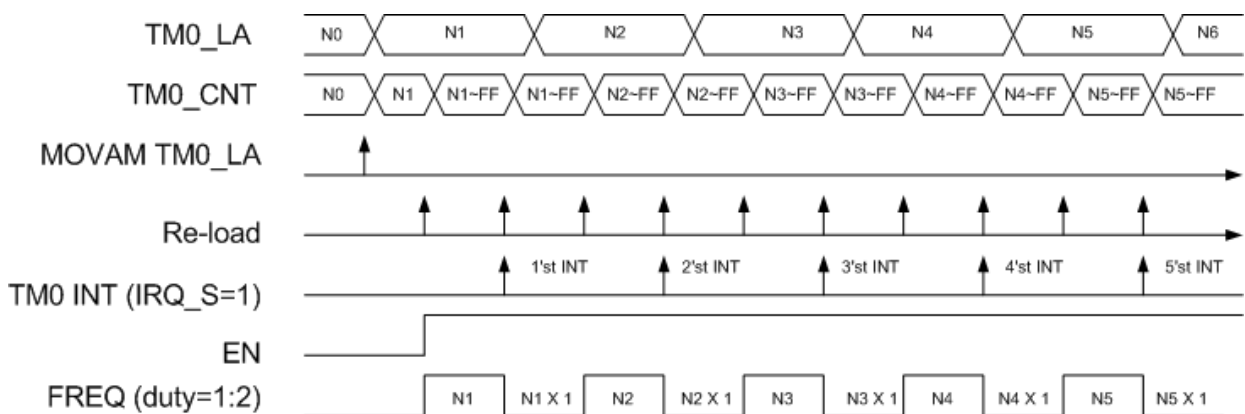
TM0 also can be used to generate FREQ and Tone. The block diagram is extended from TM0 block diagram. Please refer to below:



**Fig.3.3.1 FREQ Block Diagram**



**Fig.3.3.2 FREQ 1:1 & INT waveform**



**Fig.3.3.3 FREQ 1:2 & INT waveform**

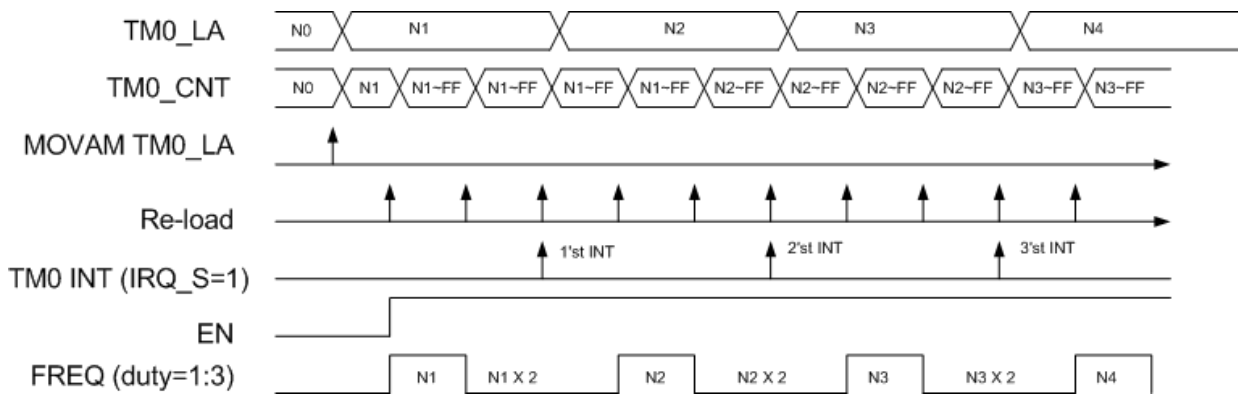


Fig.3.3.4 FREQ 1:3 & INT waveform

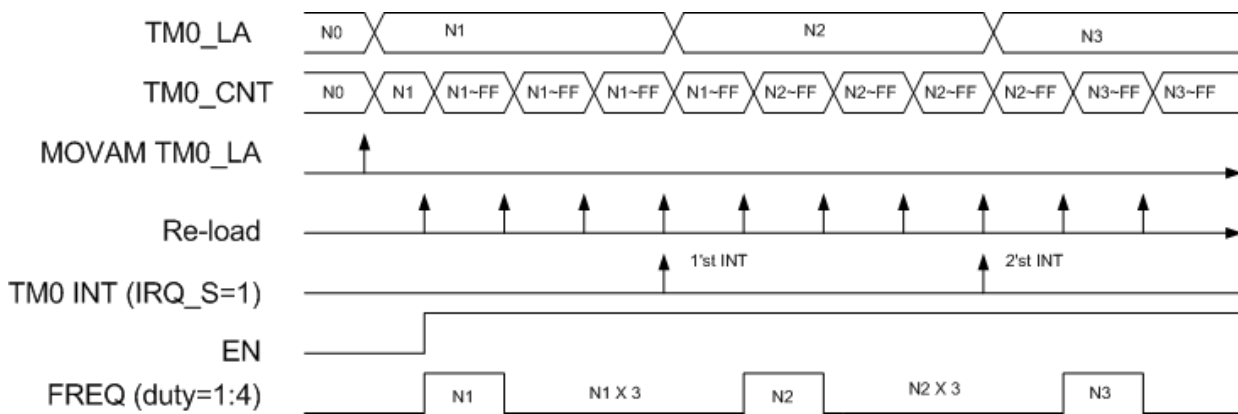


Fig.3.3.5 FREQ 1:4 & INT waveform

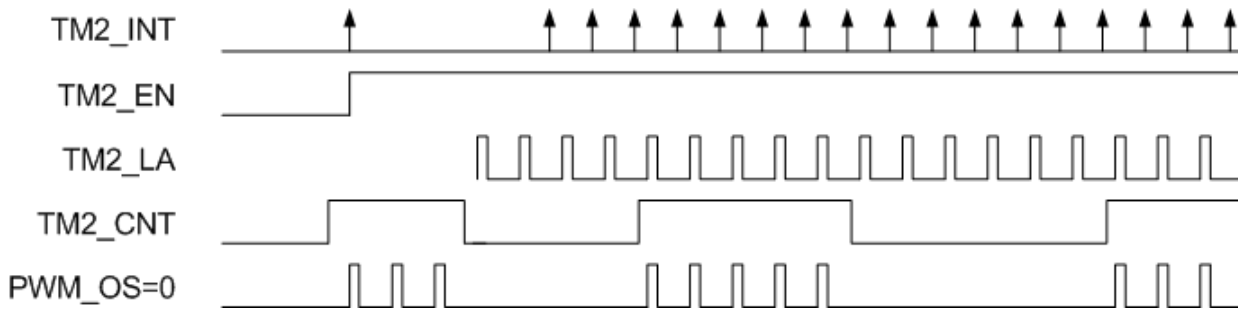


Fig.3.3.6 Remote & FREQ waveform

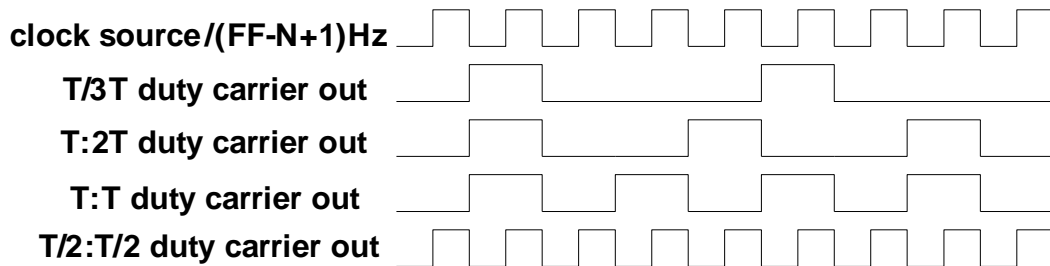
The frequency of FREQOUT pin is the same as FREQ. Please refer to below related register definition and setting flow.

The BZ output pin source can be FREQ or Tone by setting some register as below table. Because FREQ will also be the time base of some blocks, so it will always exist when TM0 is active.

**TM0\_CTL (\$17h): TM0 Control**

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM0_CTL	EN	WR_CNT	DATA	IRQ_S	SUR1	SUR0	DUTY1	DUTY0

Bit	Symbol	Description	
7	EN	TM0 enable /disable 0: Disable 1: Enable	
6	WR_CNT	TM0_CNT would be set through writing data to TM0_LA 0: Disable 1: Enable	
5	DATA	Remote mode , data output . (TM0 works as Carrier )	
4	IRQ_S	Tm0 interrupt output 0 : Tm0 overflow interrupt (RFC mode use ) 1 : FREQ cycle interrupt (Remote & FREQ use )	
3~2	SUR1~0	SUR1~0 : TM0 clock source select	
		0 0	Fast clock
		0 1	PH2
		1 0	PH0x2
1~0	DUTY1~0	DUTY1~0 : duty	
		0 0	1 : 1 H pulse:L pulse=T/2 : T/2
		0 1	1 : 2 H pulse:L pulse=T : T
		1 0	1 : 3 H pulse:L pulse=T : 2T
		1 1	1 : 4 H pulse:L pulse=T : 3T



<Note> If clock source/(FF-N+1) = (Odd number) Hz  
 The duty high = ((FF-N+1)+1)/2    The duty low = ((FF-N+1)-1)/2  
 Example : (FF-N+1)=3, H=2, L=1

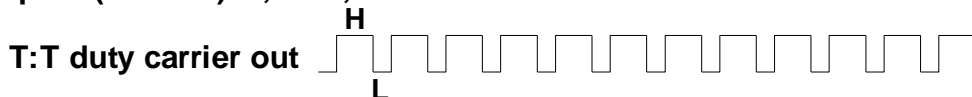


Fig.3.3.7 Timing Chart of Duty Setting

TM0\_LA (\$18h): TM0 data (R/W)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM0_LA	D7	D6	D5	D4	D3	D2	D1	D0

- Bit7~0: Timer 0 latch data (Data  $\neq$  FFh)

TM0\_CNT(\$19h): TM0 counter (R) (up counter)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM0_CNT	D7	D6	D5	D4	D3	D2	D1	D0

- Bit7~0: Timer 0 counter data

<Note> 1. This register is read only.

2. TM0 is up-count timer

5. Has auto reload function

Fig.3.3.5 Remote output example

```

INC      'MK9A160P.inc'    ;; Remote control
#DEFINE  RAM_80  80H
ORG      00                ;;
LGOTO    START

INT:     ORG      004
MOVLA   0x7E              ;; clear TM0 irq
MOVAM   IRQF
INC     RAM_*0,m          ;; Tone carrier control
BTSS    RAM_80,3
LGOTO   DATA_HI
BC      TM0_CTL,5         ;; Remote low output
LGOTO   DATA_END
DATA_HI BS      TM0_CTL,5  ;; Remote high output
        CLR     RAM_80
DATA_END NOP
        IRETI
    
```

```

START:  ORG      100h
        CLR     STATUS
        BC      SYS_CTL,b1      ;; Fast clock turn ON
        NOP
        BS      SYS_CTL,7       ;; Cpu clock = FCLK
        CLR     PAD_CTL1       ;; PD7~0 are selected
        CLR     PD_DIR         ;; PD7~0 output
        MOVLA   B'0101111'     ;; ra3-remote out
        PAD_CTL2
        MOVLA   B'01010010'    ;; fast clock , H:L=T:2T=1/3
        MOVAM   TM0_CTL
        MOVLA   B'11101001'    ;;4Mhz/[(FF-E9+1)x4]=4096K/72=56.9K
        MOVAM   TM0_LA
        MOVLA   0x01           ;; TM0 IRQ mask
        MOVAM   IRQM
        BC      TM0_CTL,6      ;; disable write through
        BS      TM0_CTL,5
        BS      IRQM_CTL,7
        BS      TM0_CTL,7
        LGOTO   $
    
```

**TONE\_CTL1 (\$39h): (W)**

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TONE_CTL1	EN	PH15E	PH14E	PH13E	PH12E	PH11E	PAT1	INV12

- Bit7: Tone enable signal  
0 : Tone disable  
1 : Tone enable
- Bit6: PH15 enable/disable  
0: Disable  
1: Enable
- Bit5: PH14 enable/disable  
0: Disable  
1: Enable
- Bit4: PH13 enable/disable  
0: Disable  
1: Enable
- Bit3: PH12 enable/disable

0: Disable

1: Enable

- Bit2: PH11 enable/disable

0: Disable

1: Enable

- Bit1: (PH14 and PH13) enable/disable

0: Disable

1: Enable

- Bit0: Delay 1/16sec enable/disable

0: Disable

1: Enable

**TONE\_CTL2 (\$3Ah): (W)**

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TONE_CTL2						CRY2	CRY1	CRY0

- Bit2-0: carrier select signal

000 : FREQOUT carrier

001 : 1Khz carrier

010 : 2Khz carrier

011 : 4Khz carrier

100 : PWM2 (TM2 pwm output)

101 : PWM3 (TM3 pwm output)

110 : PWM4 (TM4 pwm output)

111 : PWM5 (TM5 pwm output)

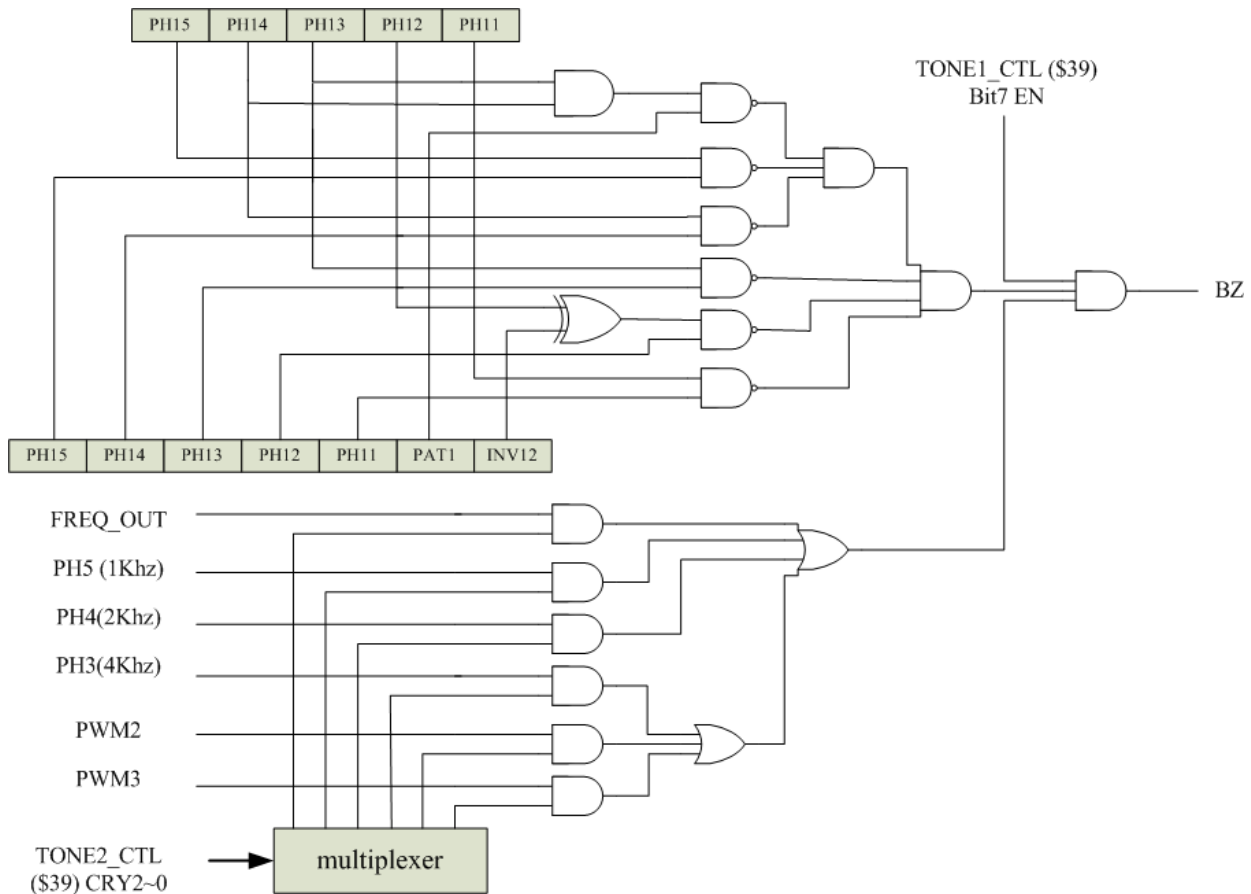


FIG. 3.3.8 Timing Chart of Duty Setting

Fig.3.3.9 Buzzer output example

```

INC      'MK9A160P.inc'    ;; Buzzer test include TONE & six carrier.
#DEFINE  RAM_80            80H
ORG      00                ;;
LGOTO    START

INT:     ORG              004
         MOVLA            0x00    ;; clear tm2
         MOVAM            IRQF
         INC              RAM_*0,m  ;; Tone carrier control
         BTSS             RAM_80,5
         LGOTO            DATA_LOW
DATA_INC INC              TONE_CTL2,m  ;; Tone carrier select signal
DATA_LOW CLR              RAM_80
         NOP
         IRETI

         ORG              100h
START:   CLR              STATUS
         MOVLA            02h
         MOVAM            LBASDT    ;; Com5~7 work as I/O
         BC               SYS_CTL,b1  ;; Fast clock turn ON
    
```



```

NOP
BS      SYS_CTL,7      ;; Cpu clock = FCLK
NOP
CLR     RAM_80
MOVLA  0x0              ;; PD output
MOVAM  PD_DIR
MOVLA  B'01011111'
MOVAM  PAD_CTL2
MOVLA  B'11000111'    ;; ph0 input , T:3T
MOVAM  TM0_CTL
MOVLA  B'1100000'
TM0_LA
MOVLA  B'0001000'
MOVAM  TONE_CTL1
CLR    TONE_CTL2
NOP
MOVLA  B'11011001'    ;; PH4,tm2 pwm mode
MOVAM  TM2_CTL!
MOVLA  B'11011001'    ;; PH5,tm3 pwm mode
MOVAM  TM3_CTL1
MOVLA  B'11101111'
MOVAM  TM2_LA
MOVAM  B'1111000'
MOVAM  TM3_LA
NOP
MOVLA  B'00001000'
MOVAM  IRQM
CLR    IRQF
BS     IRQM_CTL,7
BS     TONE_CTL1,7
LGOTO  $

```

### 3.4 Timer 2&3 (TM2 & TM3)

This two timers are multifunctional which can be set as independently 8 bit timer. The second operation mode is used to be event counters of capture to count external event from CAPT1A and CAPT1B pins. They can be used as two 8 bit counters independently . All the functions are setting by below registers and the block diagram are as below:

#### 3.4.1 Timer 2

TM2\_CTL1(\$1Fh)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM2_CTL1	EN	WR_CNT	BIT	MOD1	MOD0	EDGE	SUR1	SUR0

Bit	Symbol	Description
7	EN	TM2 enable /disable 0: Disable 1: Enable
6	WR_CNT	TM2_CNT would be set through writing data to TM2_LA (Timer ,capture ,pwm & RFC mode ) 0: Disable 1: Enable
5	BIT	BIT : 16-bit/8-bit control
		0      8-bit mode.
		1      16-bit mode , TM2+TM3.
4~3	MOD1~0	MOD1~0: TM2 operation mode selected
		0 0      Timer mode
		0 1      Capture mode
		1 0      RFC mode
		1 1      PWM mode
2	EDGE	Capture signal edge control bit 1:increment when H→L transition on external clock 0:increment when L→H transition on external clock
1~0	SUR1~0	Clock Source <b>(8-bit pwm mode ,PWM duty clock source comes from PH0X2 )</b>
		TIMER,CAPTURE      PWM mod,BIT=0      PWM mod,BIT=1
		Period      Period

	0 0	FCLK (Fast clock)	PH3	FCLK (Fast clock)
	0 1	PH0 X 2	PH4	PH0 X 2
	1 0	PH4	PH5	PH4
	1 1	PH_CLK	PH_CLK	PH_CLK

TM2\_CTL2(\$20h):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM2_CTL2	ENC	CLR_CNT		CAPIN1/ RFC_T1	CAPIN0/ RFC_T0	INT_S	PWM_OS	OV

Bit	Symbol	Description
7	ENC	Capture & RFC mode: counter auto clear (When overflow) 0: Auto clear counter (Hardware mode ) 1: Clear counter by software.
6	CLR_CNT	Capture & RFC mode: Clear counter (When ENC=1 and work in capture mode or RFC mode) 0: No clear 1: Clear counter and auto clear CLR_CNT
4~3	CAPIN1~0/ RFC_T1~0	1. Signal Source Select (Work in capture mode only ) 2. IRQ source select (Work in RFC mode only )
		mode      Capture mode      RFC mode
		00      CAPT1A input      TMR0 IRQ
		01      CAPT1B input      PH IRQ
		10      CAPT2A input      TMR3 IRQ
11      CAPT2B input      PH9		
2	INT_S	Signal Source Select (Work in capture or RFC mode )
		INT_S      Capture mode      RFC mode      RFC mode RFC=0      RFC=1
		0 <b>Capture IRQ</b> <b>No IRQ</b> <b>RFC IRQ</b>
1	PWM_OS	1 <b>Capture overflow IRQ</b> <b>RFC overflow IRQ</b> <b>RFC overflow IRQ</b>
		PWM_OS: Output state of PWM select bit. 0      The initial output state is <b>L</b> , this will change to <b>H</b> when timer overflow.

		1	The initial output state is <b>H</b> , this will change to <b>L</b> when timer overflow.
0	OV	Overflow bit ( capture & RFC mode, user should clear this bit after reading) 0: No overflow 1: Overflow	

RFC mode IRQ function

RFC	INT_S	TM3 IRQ ( IRQF bit2 )
0	0	No IRQ
0	1	RFC overflow IRQ
1	0	RFC IRQ
1	1	RFC overflow IRQ

TM2\_LA (\$21h): TM2 data (R/W)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM2_LA	D7	D6	D5	D4	D3	D2	D1	D0

TM2\_CNT(\$22h): TM2 counter (R/W) (up counter)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM2_CNT	D7	D6	D5	D4	D3	D2	D1	D0

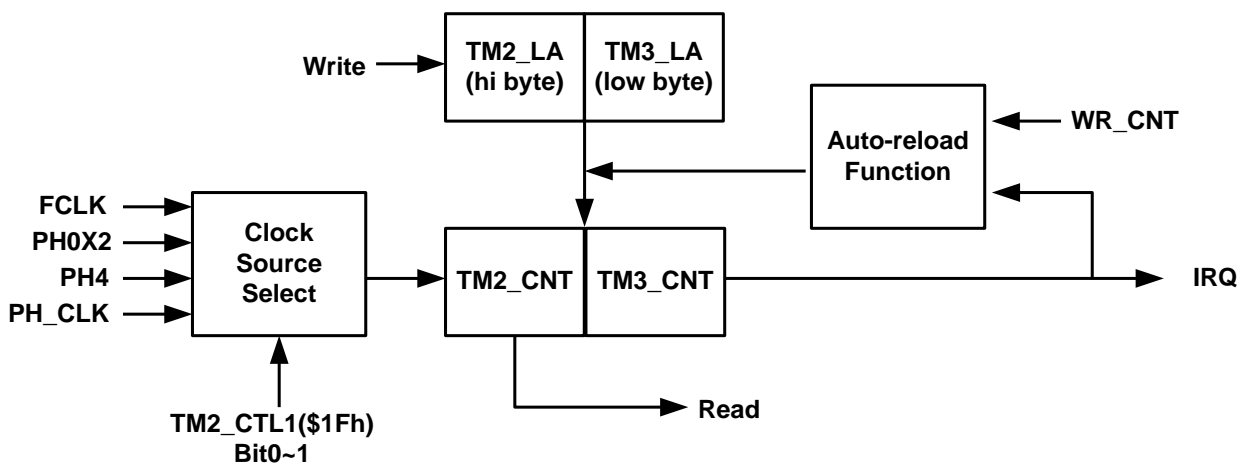


Fig.3.4.1 Block Diagram of TM2+TM3 as Timer (16-bit mode )

Fig.2.8.1 TM2+TM3 16-bit Timer mode example

```

INC      'MK9A160P.inc'  ;; TM2 Interrupt
#DEFINE  RAM_80          80H
          ORG            00          ;;
          LGOTO          START

INT:     ORG            004
          MOVLA          0x7D        ;; clear tm2
          MOVAM          IRQF
          INC            PD_DAT,m    ;; check TM2 IRQ
          MOVLA          0x010      ;; change tm2 data
          ADD            TM2_LA
          NOP
          IRETI

          ORG            100h
START:   CLR            STATUS
          MOVLA          02h
          MOVAM          LBASDT      ;; Com5~7 work as I/O
          BC             SYS_CTL,b1  ;; Fast clock turn ON
          NOP
          BS             SYS_CTL,7   ;; Cpu clock = FCLK
          NOP
          CLR            RAM_80
          CLR            PA_DAT
          CLR            PD_DAT
          CLR            PA_DIR      ;; PA output
          CLR            PAD_CTL1    ;; PD work as I/O
          CLR            PD_DIR      ;; PD output
          MOVLA          B'00110000'' ;; PA3 = PWM2 output
          MOVAM          PAD_CTL2
          MOVLA          B'01000001'  ;; 8-bit timer mode,
          MOVAM          TM2_CTL1
          MOVLA          B'00000001'
          MOVAM          TM2_CTL2
          BS             TM2_CTL1,6   ;; write TM2_CNT enable
          BS             TM3_CTL1,6   ;; write TM3_CNT enable
          MOVLA          0x0          ;; TM2 up-counter 00 → FF
          MOVAM          TM2_LA
          MOVLA          0x80        ;; pwm duty counter
          MOVAM          TM3_LA
          BC             TM2_CTL1,6

```

```

BC      TM3_CTL1,6
MOVLA  B'00000010'    ;; set TM2 irq mask
MOVAM  IRQM
CLR     IRQF
BS     IRQM_CTL,7
BS     TM2_CTL1,7
LGOTO  $
    
```

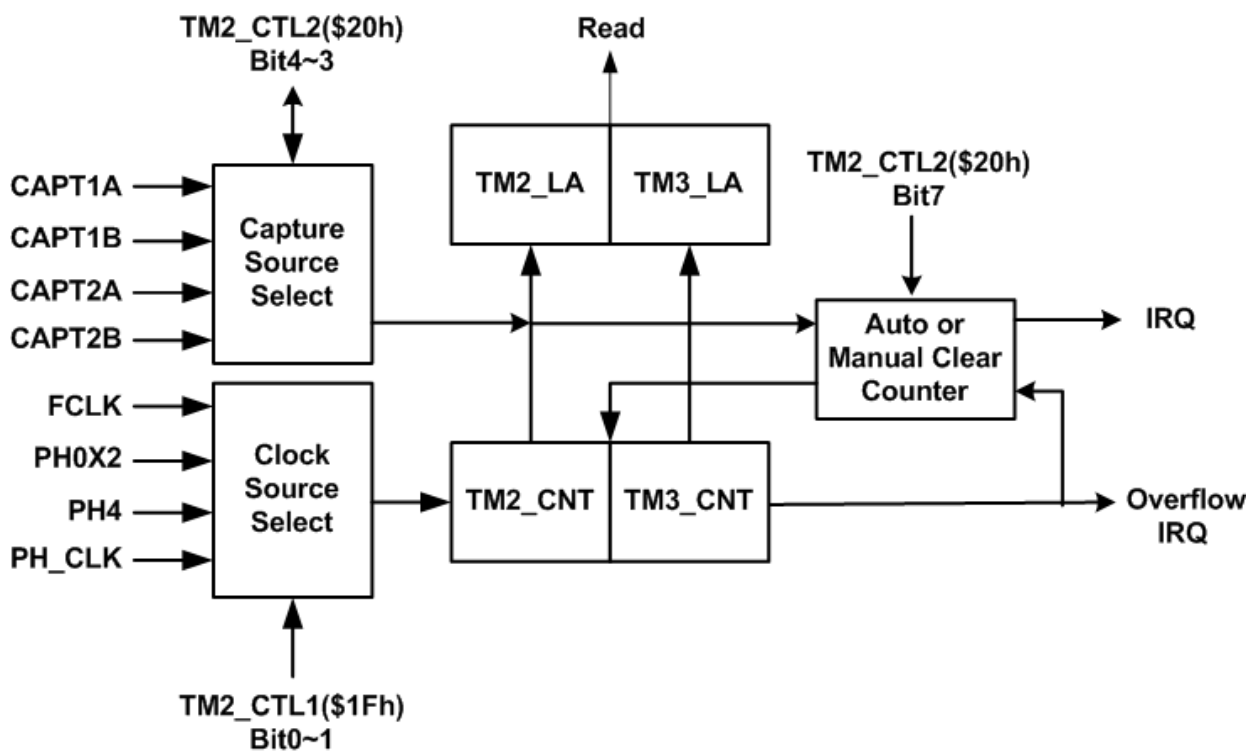


Fig.3.4.2 Block Diagram of TM2+TM3 as Capture (16-bit mode)

Fig.2.8.1 TM2+TM3 16-bit Capture mode example

```

INC      'MK9A160P.inc'    ;; TM2 Interrupt
#DEFINE  RAM_80  80H
ORG      00              ;;
LGOTO    START

INT:
ORG      004
MOVLA   0x7D             ;; clear tm2
MOVAM   IRQF
MOV     TM2_LA           ;; check Capture high byte data
MOVAM   PD_DAT
MOV     TM3_LA           ;; check Capture low byte data
MOVAM   PC_DAT
IRETI

ORG      100h
    
```

```
START:  CLR      STATUS
        MOVLA   02h
        MOVAM   LBASDT      ;; Com5~7 work as I/O
        BC      SYS_CTL,b1  ;; Fast clock turn ON
        NOP
        BS      SYS_CTL,7   ;; Cpu clock = FCLK
        NOP
        CLR     RAM_80
        CLR     PA_DAT
        CLR     PD_DAT
        MOVLA   0xFF
        MOVAM   PA_DIR      ;; PA input
        CLR     PAD_CTL1    ;; PD work as I/O
        CLR     PD_DIR      ;; PD output
        CLR     PC_DIR      ;; PC output
        MOVLA   B'00101001' ;; 16-bit capture,couter ph0x2
        MOVAM   TM2_CTL1
        BC      TM2_CTL2,3  ;; capture input = PA3
        BS      TM2_CTL2,3  ;; capture input = PA6
        MOVLA   B'00000010' ;; set TM2 irq mask
        MOVAM   IRQM
        CLR     IRQF
        BS      IRQM_CTL,7
        BS      TM2_CTL1,7
        LGOTO   $
```

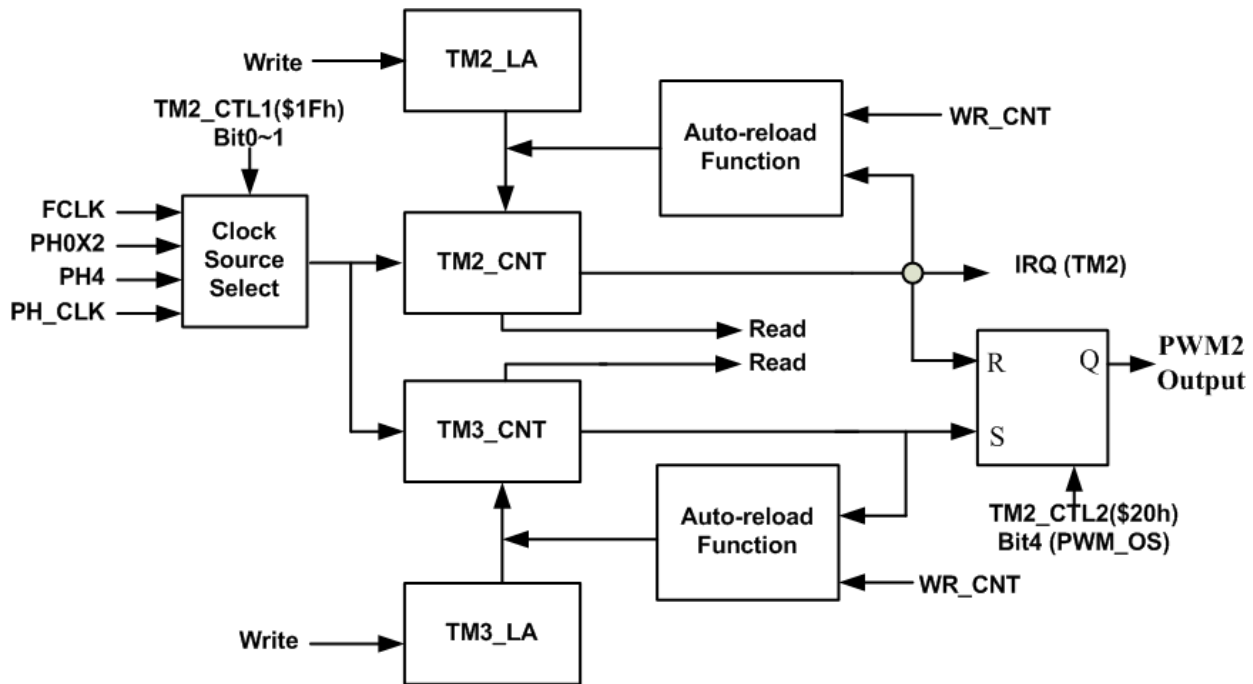


Fig.3.4.3 Block Diagram of TM2+TM3 as Pwm (BIT=1)

Fig.2.8.1 PWM2 (TM2+TM3 PWM) output example

```

INC      'MK9A160P.inc'    ;; Buzzer test include TONE & six carrier.
#DEFINE  RAM_80           80H
ORG      00                ;;
LGOTO    START

INT:
ORG      004
MOVLA   0x7D                ;; clear tm2
MOVAM   IRQF
INC     PD_DAT,m            ;; check TM2 IRQ
MOVLA   0x010               ;; change PWM period
ADD     TM2_LA
MOVLA   0x010               ;; change PWM duty
ADD     TM3_LA
NOP
IRETI

ORG      100h
START:
CLR     STATUS
MOVLA   02h
MOVAM   LBASDT              ;; Com5~7 work as I/O
BC      SYS_CTL,b1         ;; Fast clock turn ON
NOP
BS      SYS_CTL,7          ;; Cpu clock = FCLK
NOP
CLR     RAM_80
CLR     PA_DAT

```



```

CLR      PD_DAT
CLR      PA_DIR      ;; PA output
CLR      PAD_CTL1    ;; PD work as I/O
CLR      PD_DIR      ;; PD output
MOVLA    B'00110000'' ;; PA3 = PWM2 output
MOVAM    PAD_CTL2
MOVLA    B'00111000'' ;; pwm mode, TM2 work as period, TM3 work as duty
MOVAM    TM2_CTL1
MOVLA    B'00000001''
MOVAM    TM2_CTL2
BS       TM2_CTL1,6  ;; write TM2_CNT enable
BS       TM3_CTL1,6  ;; write TM3_CNT enable
MOVLA    0x20        ;; pwm period counter
MOVAM    TM2_LA
MOVLA    0x80        ;; pwm duty counter
MOVAM    TM3_LA
BC       TM2_CTL1,6
BC       TM3_CTL1,6
MOVLA    B'00000010'' ;; set TM2 irq mask
MOVAM    IRQM
CLR      IRQF
BS       IRQM_CTL,7
BS       TM2_CTL1,7
LGOTO    $

```

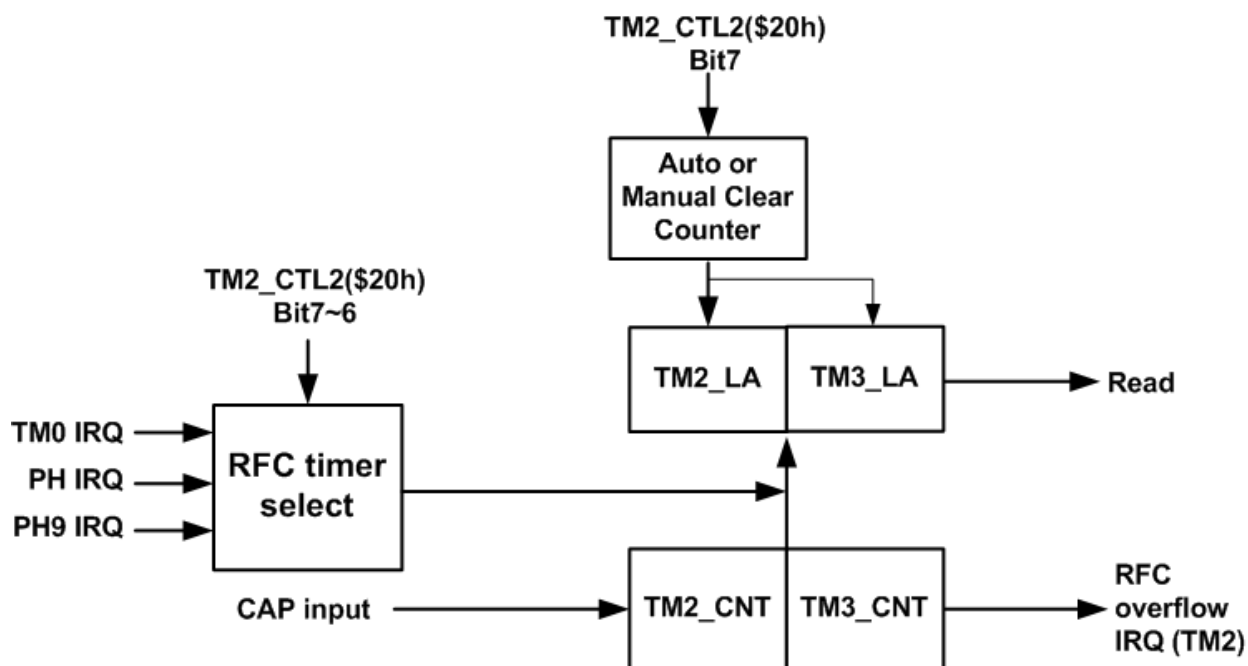


Fig.3.4.4 Block Diagram of TM2+TM3 work as RFC (16-bit mode )

Fig.2.8.1 TM2+TM3 16-bit RFC mode example

```

INC      'MK9A160P.inc'    ;; TM2 Interrupt
#DEFINE  RAM_80           80H
ORG      00                ;;
LGOTO   START

INT:     ORG      004
         MOVLA   0x7E        ;; clear tm0
         MOVAM   IRQF
         MOV     TM2_LA      ;; check RFC high byte data
         MOVAM   PD_DAT
         MOV     TM3_LA      ;; check RFC low byte data
         MOVAM   PC_DAT
         IRETI

START:   ORG      100h
         CLR     STATUS
         MOVLA   02h
         MOVAM   LBASDT      ;; Com5~7 work as I/O
         BC     SYS_CTL,b1   ;; Fast clock turn ON
         NOP
         BS     SYS_CTL,7    ;; Cpu clock = FCLK
         NOP
         CLR     RAM_80
         CLR     PA_DAT
         CLR     PD_DAT

```

```

CLR      PAD_CTL1      ;; PD work as I/O
CLR      PD_DIR        ;; PD output
CLR      PC_DIR        ;; PC output

MOVLA    0xFF
MOVAM    PA_DIR        ;; PA input
MOVLA    B'11101111'   ;; RFC,BZ & BZM out
MOVAM    PAD_CTL2
BS       PAD_CTL3,0    ;; RREF ON
;BS      PAD_CTL3,1    ;; SEN0 ON
;BS      PAD_CTL3,2    ;; SEN1 ON
MOVAM
MOVLA    B'00111000'   ;; 16-bit RFC
MOVAM    TM2_CTL1      ;; RFC IRQ come from TM0
CLR      TM2_CTL2
MOVLA    B'01000100'   ;; rfc,ph0x2 , T/2:T/2
MOVAM    TM0_CTL
MOVLA    B'00111111'
MOVAM    TM0_LA
BC       TM0_CTL,6

MOVLA    B'00000001'   ;; set TM0 irq mask
MOVAM    IRQM
BS       IRQM_CTL,7
BS       TM2_CTL1,7
BS       TM0_CTL1,7
CLR      IRQF
LGOTO   $

```

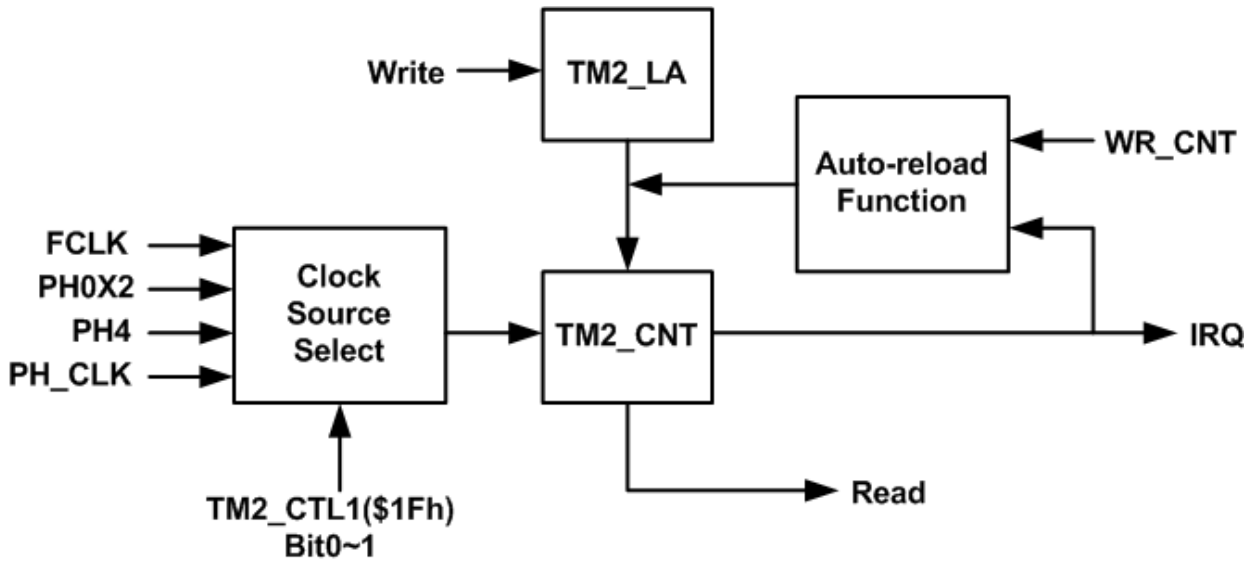


Fig.3.4.5 Block Diagram of TM2 works as Timer (8-bit mode )

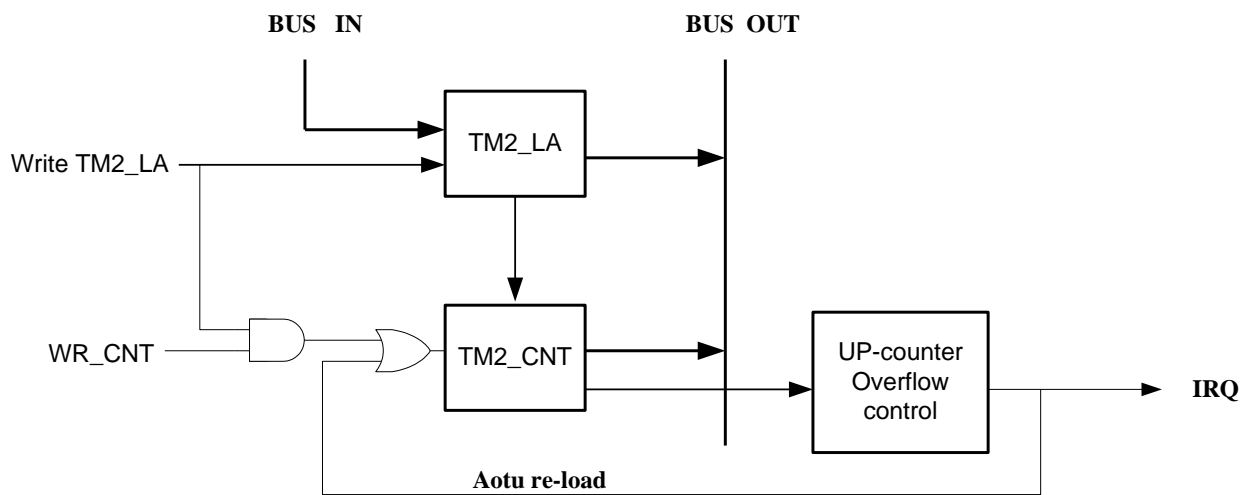


Fig.3.4.6 Block Diagram of TM2 works as Timer (8-bit mode )

Fig.3.4.6-1 TM2 8-bit Timer mode example

```

INC      'MK9A160P.inc'  ;; TM2 8-bit timer mode
#DEFINE  RAM_80         80H
ORG      00             ;;
LGOTO   START

INT:     ORG      004
         MOVLA   0x7D      ;; clear tm2
         MOVAM   IRQF
         INC     PD_DAT,m  ;; check TM2 IRQ
         MOVLA   0x010     ;; change tm2 data
         ADD     TM2_LA
         NOP
         IRETI
    
```

```

START:  ORG      100h
        CLR      STATUS
        MOVLA    02h
        MOVAM    LBASDT      ;; Com5~7 work as I/O
        BC       SYS_CTL,b1  ;; Fast clock turn ON
        NOP
        BS       SYS_CTL,7    ;; Cpu clock = FCLK
        NOP
        CLR      RAM_80
        CLR      PA_DAT
        CLR      PD_DAT
        CLR      PA_DIR      ;; PA output
        CLR      PAD_CTL1    ;; PD work as I/O
        CLR      PD_DIR      ;; PD output
        MOVLA    B'00110000'' ;; PA3 = PWM2 output
        MOVAM    PAD_CTL2
        MOVLA    B'01000001'  ;; 8-bit timer mode,
        MOVAM    TM2_CTL1
        MOVLA    B'00000001'
        MOVAM    TM2_CTL2
        BS       TM2_CTL1,6    ;; write TM2_CNT enable
        MOVLA    0x0          ;; TM2 up-counter 00 → FF
        MOVAM    TM2_LA
        BC       TM2_CTL1,6
        MOVLA    B'00000010'  ;; set TM2 irq mask
        MOVAM    IRQM
        CLR      IRQF
        BS       IRQM_CTL,7
        BS       TM2_CTL1,7
        LGOTO    $

```

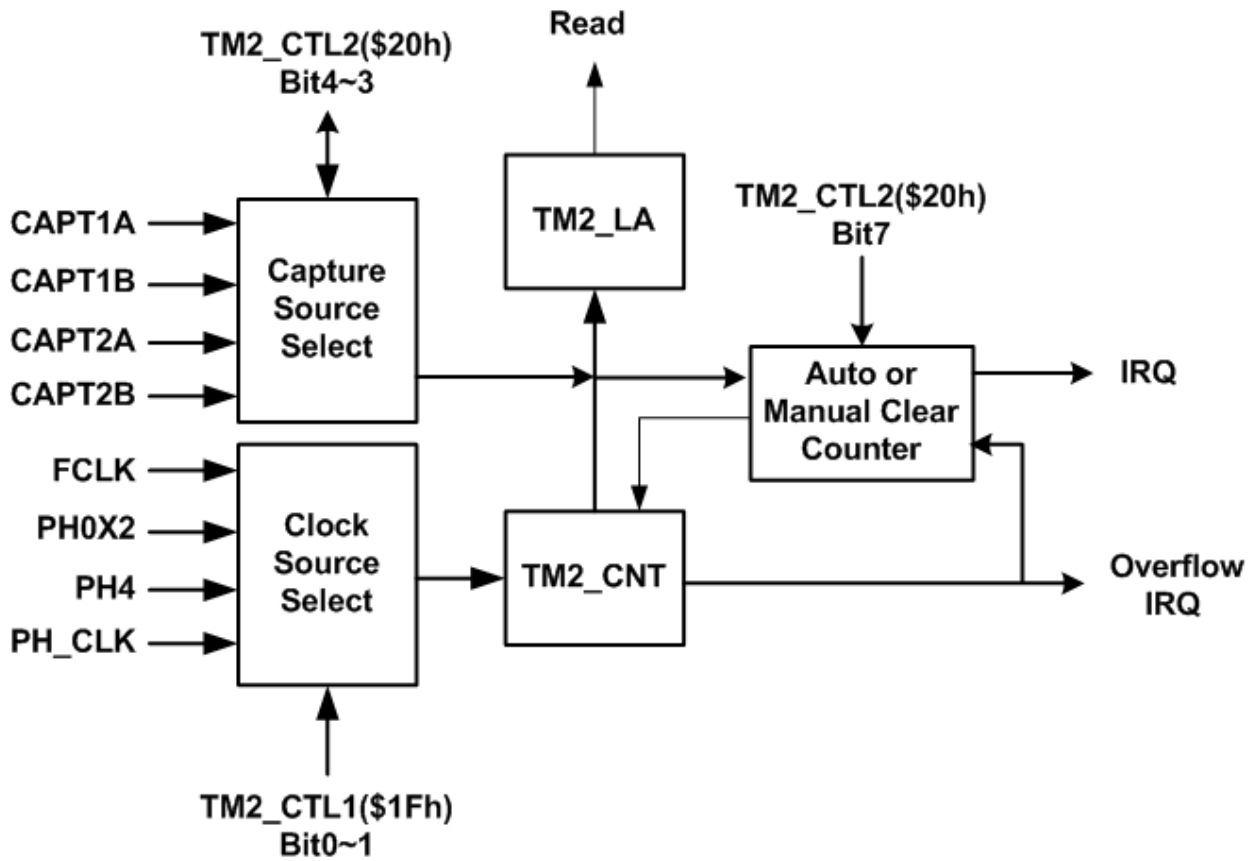


Fig.3.4.7 Block Diagram of TM2 as capture (8-bit mode )

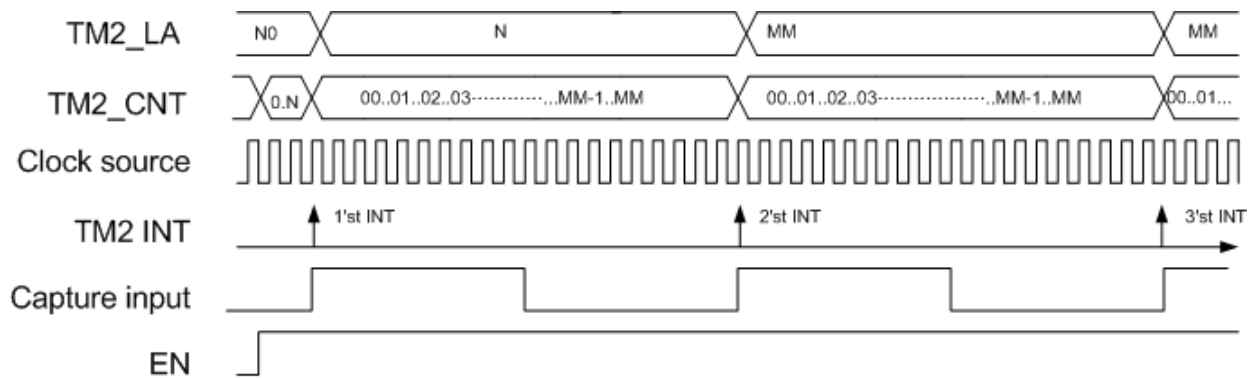


Fig.3.4.8 Block Diagram of TM2 as capture (Cycle , EDGE=0) (8-bit mode )

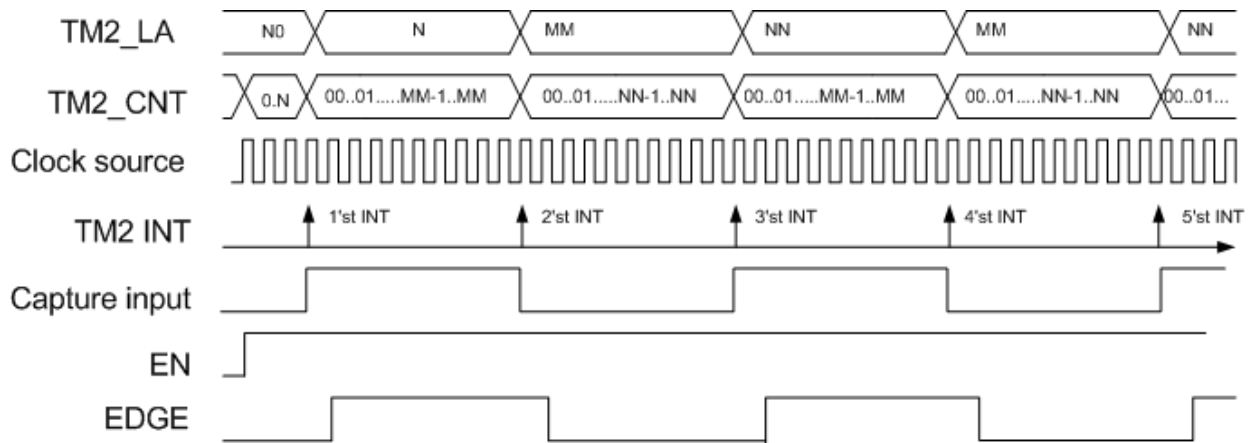


Fig.3.4.9 Block Diagram of TM2 as capture ( pulse ) ( 8-bit mode )

Fig.3.4.9-1 TM2 8-bit Capture mode example

```

INC      'MK9A160P.inc'    ;; TM2 8-bit capture mode
#DEFINE  RAM_80  80H
ORG      00                ;;
LGOTO    START

INT:     ORG      004
         MOVL    0x7D        ;; clear tm2
         MOVAM   IRQF
         MOV     TM2_LA      ;; check RFC high byte data
         MOVAM   PD_DAT
         IRETI

START:   ORG      100h
         CLR     STATUS
         MOVL    02h
         MOVAM   LBASDT      ;; Com5~7 work as I/O
         BC      SYS_CTL,b1  ;; Fast clock turn ON
         NOP
         BS      SYS_CTL,7   ;; Cpu clock = FCLK
         NOP
         CLR     RAM_80
         CLR     PA_DAT
         CLR     PD_DAT
         MOVL    0xFF
         MOVAM   PA_DIR      ;; PA input
         CLR     PAD_CTL1    ;; PD work as I/O
         CLR     PD_DIR      ;; PD output
         CLR     PC_DIR      ;; PC output
         MOVL    B'00001001' ;; 8-bit capture,couter ph0x2
         MOVAM   TM2_CTL1
    
```

```

BC      TM2_CTL2,3    ;; capture input = PA3
;; BS   TM2_CTL2,3    ;; capture input = PA6
MOVLA   B'00000010'  ;; set TM2 irq mask
MOVAM   IRQM
CLR     IRQF
BS      IRQM_CTL,7
BS      TM2_CTL1,7
LGOTO   $
    
```

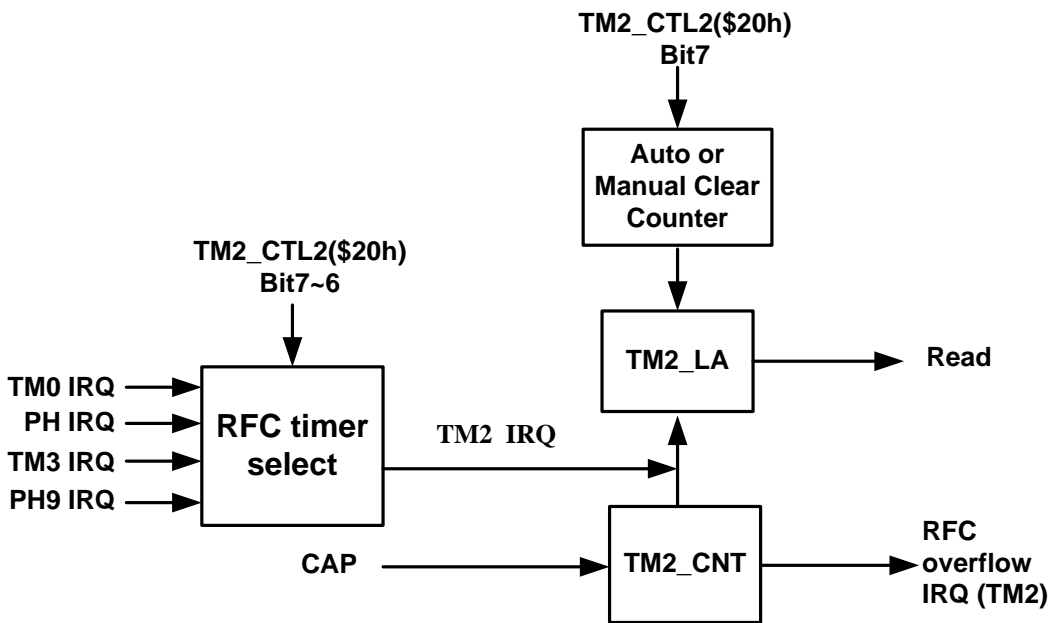


Fig.3.4.10 Block Diagram of TM2 as rfc (8-bit mode )



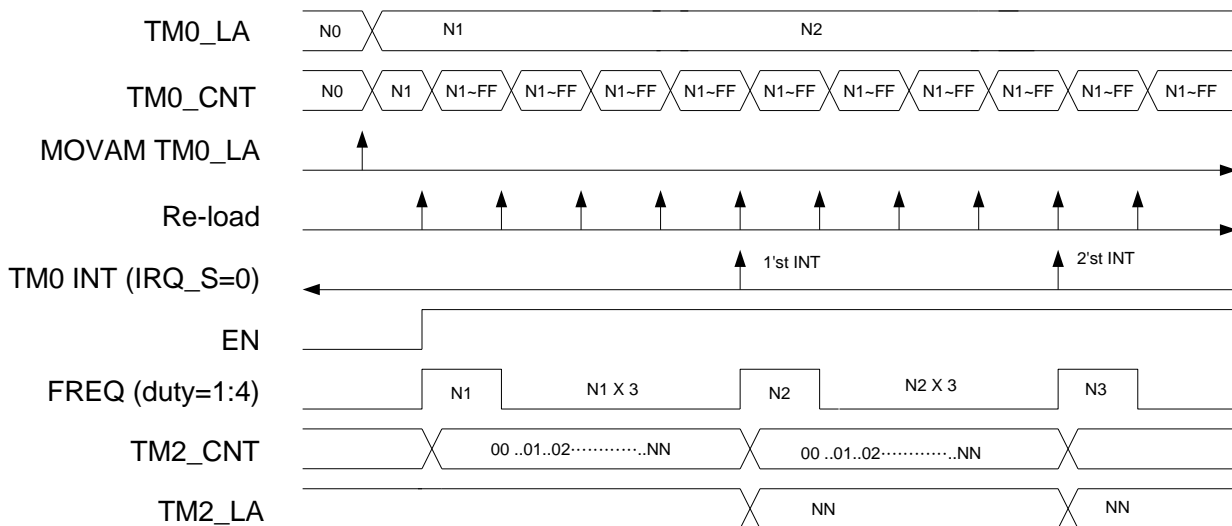


Fig.3.4.11 Block Diagram of TM2 as rfc (software mode , ENC=1) (8-bit mode )

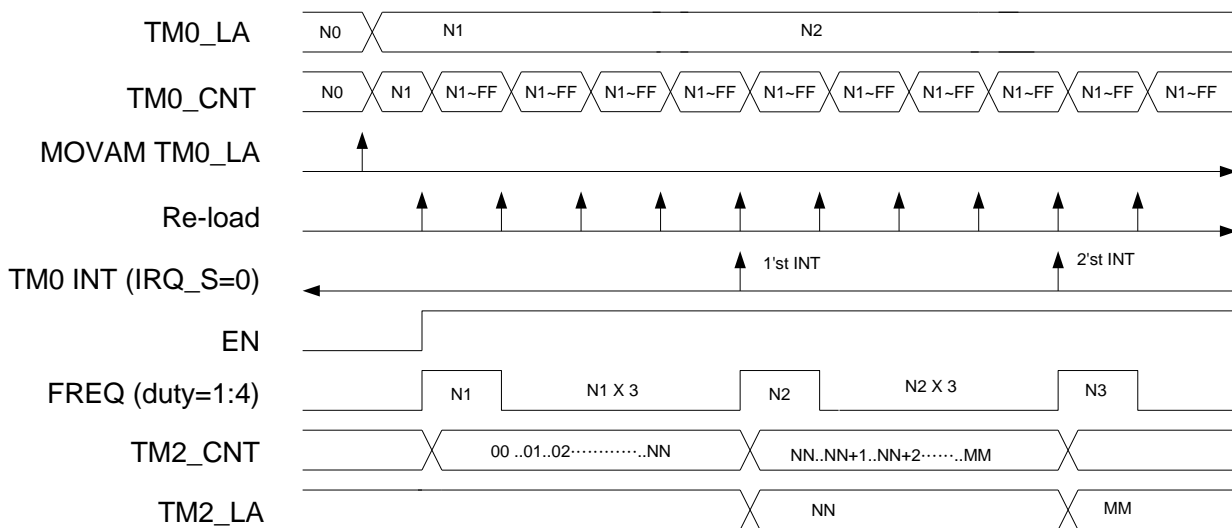


Fig.3.4.12 Block Diagram of TM2 as rfc (hardware mode , ENC=0) (8-bit mode )

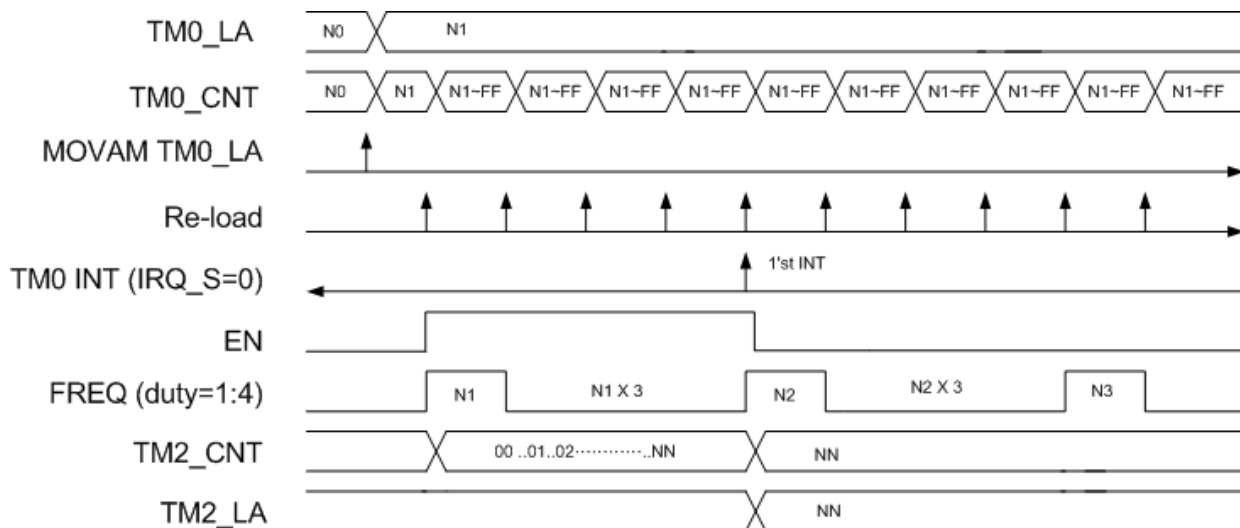


Fig.3.4.13 Block Diagram of TM2 as rfc (hardware mode , TM2\_CTL RFC=bit5=1 ) (8-bit mode )

Fig.3.4.12-1 TM2 8-bit RFC mode example

```

INC      'MK9A160P.inc'    ;; TM2 8-bit RFC mode
#DEFINE  RAM_80           80H
ORG      00                ;;
LGOTO    START

INT:     ORG              004
MOVLA   0x7E              ;; clear tm0
MOVAM   IRQF
MOV     TM2_LA            ;; check RFC data
MOVAM   PD_DAT
IRETI

ORG      100h
START:   CLR              STATUS
MOVLA   02h
MOVAM   LBASDT           ;; Com5~7 work as I/O
BC      SYS_CTL,b1       ;; Fast clock turn ON
NOP
BS      SYS_CTL,7        ;; Cpu clock = FCLK
NOP
CLR     RAM_80
CLR     PA_DAT
CLR     PD_DAT
CLR     PAD_CTL1         ;; PD work as I/O
CLR     PD_DIR           ;; PD output
CLR     PC_DIR           ;; PC output
    
```

```

MOVLA    0xFF
MOVAM    PA_DIR        ;; PA input
MOVLA    B'11101111'   ;; RFC,BZ & BZM out
MOVAM    PAD_CTL2
BS       PAD_CTL3,0    ;; RREF ON
;BS      PAD_CTL3,1    ;; SEN0 ON
;BS      PAD_CTL3,2    ;; SEN1 ON
MOVAM
MOVLA    B'00011000'   ;; TM2 work as 8-bit RFC
MOVAM    TM2_CTL1      ;; RFC IRQ come from TM0
CLR      TM2_CTL2
MOVLA    B'01000100'   ;; rfc,ph0x2 , T/2:T/2
MOVAM    TM0_CTL
MOVLA    B'00111111'
MOVAM    TM0_LA
BC       TM0_CTL,6

MOVLA    B'00000001'   ;; set TM0 irq mask
MOVAM    IRQM
BS       IRQM_CTL,7
BS       TM2_CTL1,7
BS       TM0_CTL1,7
CLR      IRQF
LGOTO   $

```

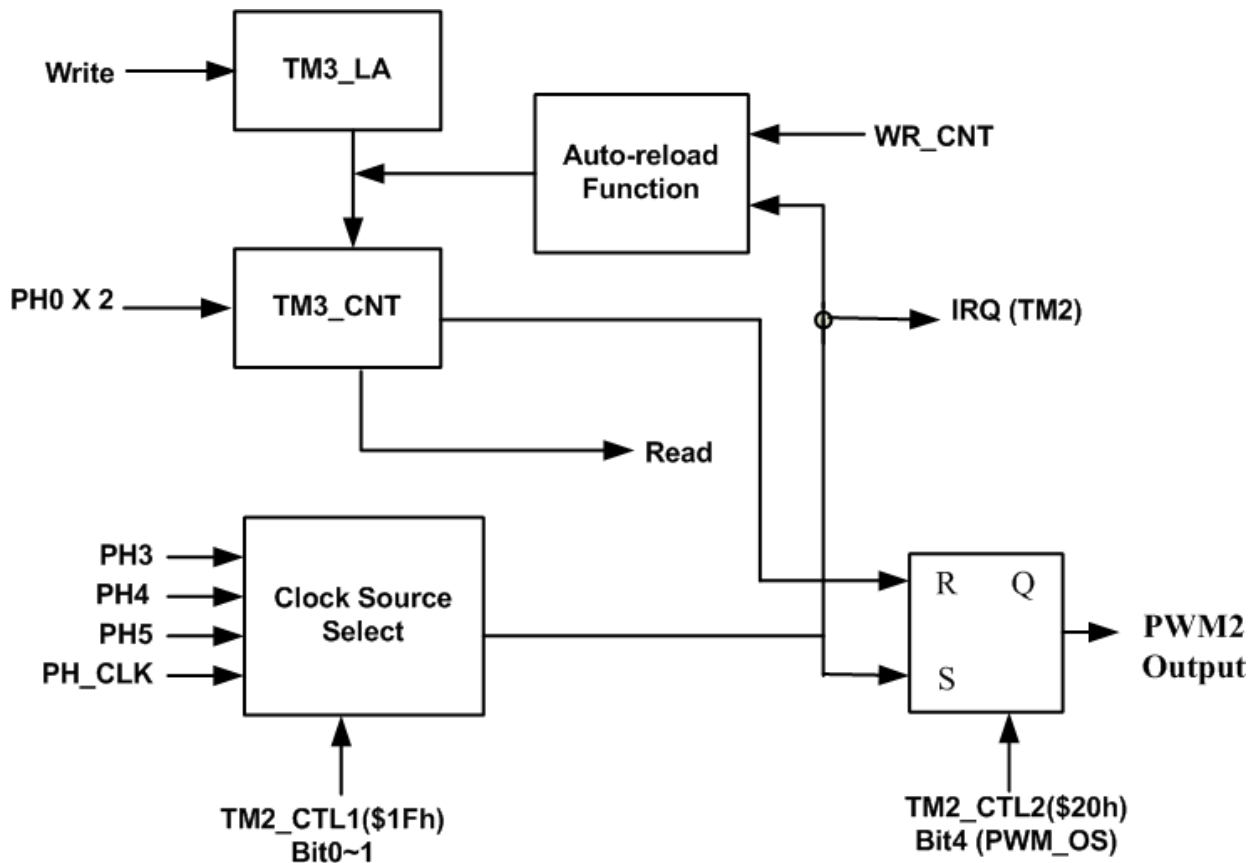


Fig.3.4.13 Block Diagram of TM2 as pwm (8-bit mode )

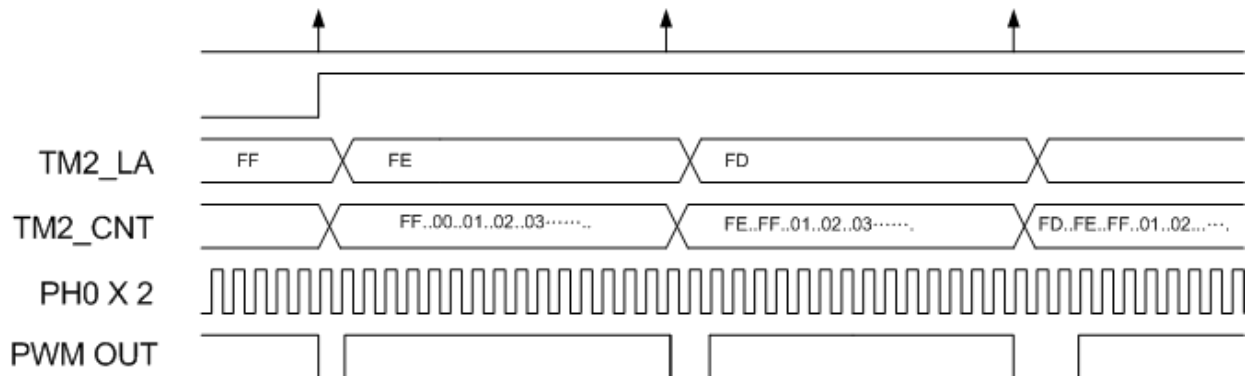
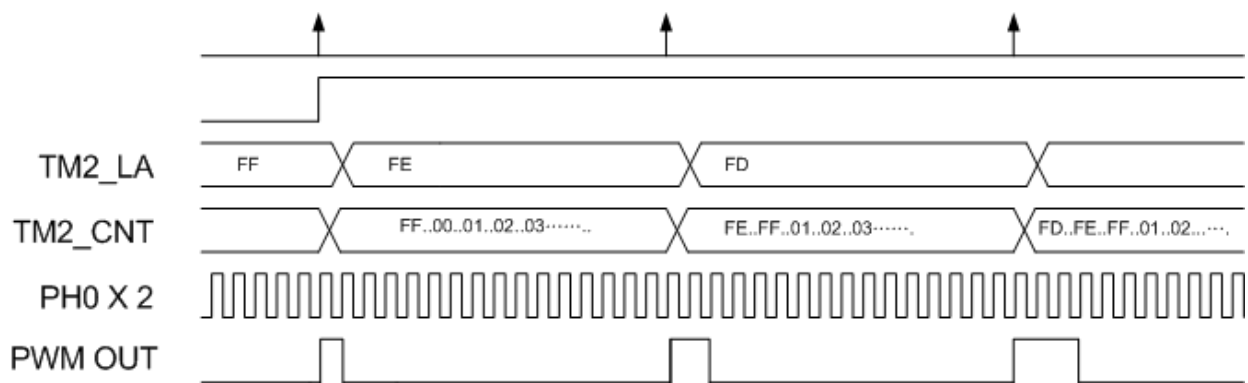


Fig.3.4.14 Pwm waveform (PWM\_OS=0 , Period=PH3) (8-bit mode )



**Fig.3.4.15 Pwm waveform (PWM\_OS=1 , Period=PH3) (8-bit mode )**

TM2_LA	PWM2 (L:Hi)	TM2_LA	PWM2 (L:Hi)
FF	1 : 15	F7	9:7
FE	2 : 14	F6	10:6
FD	3 : 13	F5	11:5
FC	4 : 12	F4	12:4
FB	5 : 11	F3	13:3
FA	6 : 10	F2	14:2
F9	7 : 9	F1	15:1
F8	8 : 8	<b>TM2_LA &lt; F1</b>	Always "0"

**Fig.3.4.16 Pwm table 1 (PWM\_OS=0 , Period=PH3) (8-bit mode )**

TM2_LA	PWM2 ( Hi : L )	TM2_LA	PWM2 ( Hi : L )
FF	1 : 15	F7	9:7
FE	2 : 14	F6	10:6
FD	3 : 13	F5	11:5
FC	4 : 12	F4	12:4
FB	5 : 11	F3	13:3
FA	6 : 10	F2	14:2
F9	7 : 9	F1	15:1
F8	8 : 8	<b>TM2_LA &lt; F1</b>	Always "1"

**Fig.3.4.17 Pwm table 2 (PWM\_OS=1 , Period=PH3) (8-bit mode )**

TM2_LA	PWM2 (L:Hi)	TM2_LA	PWM2 (L:Hi)
FF	1 : 31	F7	9 : 23
FE	2 : 30	F6	10 : 22
FD	3 : 29	F5	11 : 21
FC	4 : 28	F4	12 : 20
FB	5 : 27	F3	13 : 19
FA	6 : 26	F2	14 : 18
F9	7 : 25	F1	15 : 17
F8	8 : 24	<b>TM2_LA &lt; E1</b>	Always "0"

**Fig.3.4.18 Pwm table 3 (PWM\_OS=0 , Period=PH4) (8-bit mode )**

TM2_LA	PWM2 ( Hi : L )	TM2_LA	PWM2 ( Hi : L )
FF	1 : 31	F7	9 : 23
FE	2 : 30	F6	10 : 22
FD	3 : 29	F5	11 : 21
FC	4 : 28	F4	12 : 20
FB	5 : 27	F3	13 : 19
FA	6 : 26	F2	14 : 18
F9	7 : 25	F1	15 : 17
F8	8 : 24	<b>TM2_LA &lt; E1</b>	Always "1"

**Fig.3.4.19 Pwm table 4 (PWM\_OS=1 , Period=PH4) (8-bit mode )**

TM2_LA	PWM2 (L:Hi)	TM2_LA	PWM2 (L:Hi)
FF	1 : 63	F7	9 : 55
FE	2 : 62	F6	10 : 54
FD	3 : 61	F5	11 : 53
FC	4 : 60	F4	12 : 52
FB	5 : 59	F3	13 : 51
FA	6 : 58	F2	14 : 50
F9	7 : 57	F1	15 : 49
F8	8 : 56	<b>TM2_LA &lt; C1</b>	Always "0"

**Fig.3.4.20 Pwm table 5 (PWM\_OS=0 , Period=PH5) (8-bit mode )**

TM2_LA	PWM2 ( Hi : L )	TM2_LA	PWM2 ( Hi : L )
FF	1 : 63	F7	9 : 55
FE	2 : 62	F6	10 : 54
FD	3 : 61	F5	11 : 53
FC	4 : 60	F4	12 : 52
FB	5 : 59	F3	13 : 51
FA	6 : 58	F2	14 : 50
F9	7 : 57	F1	15 : 49
F8	8 : 56	<b>TM2_LA &lt; C1</b>	Always "1"

**Fig.3.4.21 Pwm table 6 (PWM\_OS=1 , Period=PH4) (8-bit mode )**

TM2_LA	PWM2 ( L : Hi )	TM2_LA	PWM2 ( L : Hi )
FF	1 : 255	F7	9 : 247
FE	2 : 254	F6	10 : 246
FD	3 : 253	F5	11 : 245
FC	4 : 252	F4	12 : 244
FB	5 : 251	F3	13 : 243
FA	6 : 250	F2	14 : 242
F9	7 : 249	MM	<b>(FF-MM+1) : MM</b>
F8	8 : 248	0	Always "0"

Fig.3.4.22 Pwm table 7 (PWM\_OS=0 , Period=PH7) (8-bit mode )

TM2_LA	PWM2 ( Hi : L )	TM2_LA	PWM2 ( Hi : L )
FF	1 : 255	F7	9 : 247
FE	2 : 254	F6	10 : 246
FD	3 : 253	F5	11 : 245
FC	4 : 252	F4	12 : 244
FB	5 : 251	F3	13 : 243
FA	6 : 250	F2	14 : 242
F9	7 : 249	MM	<b>(FF-MM+1) : MM</b>
F8	8 : 248	0	Always "1"

Fig.3.4.23 Pwm table 8 (PWM\_OS=1 , Period=PH7) (8-bit mode )

Fig.3.4.14-1 PWM2 output (TM2 8-bit PWM output) example

```

INC      'MK9A160P.inc'    ;; TM2 pwm mode
                                ;; Period=PH7 , Duty cycle come from TM2
#DEFINE  RAM_80  80H
          ORG    00          ;;
          LGOTO  START

INT:     ORG    004
          MOVLA 0x7D          ;; clear tm2
          MOVAM IRQF
          INC   PD_DAT,m     ;; check TM2 IRQ
          MOVLA 0x01
          ADD   TM2_LA       ;; change PWM duty
          NOP
          IRETI

          ORG    100h
START:   CLR    STATUS
          MOVLA 02h
          MOVAM LBASDT       ;; Com5~7 work as I/O
    
```

```

BC      SYS_CTL,b1      ;; Fast clock turn ON
NOP
BS      SYS_CTL,7      ;; Cpu clock = FCLK
NOP
CLR     RAM_80
CLR     PA_DIR          ;; PA output
CLR     PAD_CTL1       ;; PD work as I/O
CLR     PD_DIR         ;; PD output
MOVLA  B'00110000''    ;; PA3 = PWM2 output
MOVAM  PAD_CTL2
MOVLA  B'01011000'     ;; pwm mode , pwm duty=ph3
MOVAM  TM2_CTL1
MOVLA  B'00000001'
MOVAM  TM2_CTL2
MOVLA  0xE0            ;; pwm duty counter
MOVAM  TM2_LA
BC      TM2_CTL1,6
NOP
MOVLA  B'00000010'     ;; set TM2 irq mask
MOVAM  IRQM
CLR     IRQF
BS      IRQM_CTL,7
BS      TM2_CTL1,7
LGOTO  $

```



### 3.4.2 Timer 3 (TM3)

TM3\_CTL1(\$23h)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM3_CTL1	EN	WR_CNT	--	MOD1	MOD0	EDGE	SUR1	SUR0

Bit	Symbol	Description			
7	EN	TM3 enable /disable 0: Disable 1: Enable			
6	WR_CNT	TM3_CNT would be set through writing data to TM3_LA (Timer ,capture ,pwm & RFC mode ) 0: Disable 1: Enable			
4~3	MOD1~0	MOD1~0: TM3 operation mode selected			
		0 0	Timer mode		
		0 1	Capture mode		
		1 0	RFC mode		
1 1	PWM mode ( TM3 comes from PH0X2 input )				
2	EDGE	Capture signal edge control bit 1:increment when H→L transition on external clock 0:increment when L→H transition on external clock			
1~0	SUR1~0	Clock Source <b>(8-bit pwm mode ,PWM duty clock source comes from PH0X2 )</b>			
				PWM mod,BIT=0	PWM mod,BIT=1
			TIMER,CAPTURE	Period	Duty
		0 0	FCLK (Fast clock)	PH3	FCLK (Fast clock)
		0 1	PH0 X 2	PH4	PH0 X 2
		1 0	PH4	PH5	PH4
1 1	PH_CLK	PH_CLK	PH_CLK		

## TM3\_CTL2(\$24h):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM3_CTL2	ENC	CLR_CNT		CAPIN1/ RFC_T1	CAPIN0/ RFC_T0	INT_S	PWM_OS	OV

Bit	Symbol	Description			
7	ENC	Capture & RFC mode: counter auto clear (When overflow) 0: Auto clear counter (Hardware mode ) 1: Clear counter by software.			
6	CLR_CNT	Capture & RFC mode: Clear counter (When ENC=1 and work in capture mode or RFC mode) 0: No clear 1: Clear counter and auto clear CLR_CNT			
4~3	CAPIN1~0/ RFC_T1~0	1. Signal Source Select (Work in capture mode only ) 2. IRQ source select (Work in RFC mode only )			
		mode	Capture mode	RFC mode	
		00	CAPT1A input	TMR0 IRQ	
		01	CAPT1B input	PH IRQ	
		10	CAPT2A input	TMR2 IRQ	
2	INT_S	Signal Source Select (Work in capture or RFC mode )			
		INT_S	Capture mode	RFC mode RFC=0	RFC mode RFC=1
		0	<b>Capture IRQ</b>	<b>No IRQ</b>	<b>RFC IRQ</b>
		1	<b>Capture overflow IRQ</b>	<b>RFC overflow IRQ</b>	<b>RFC overflow IRQ</b>
1	PWM_OS	PWM_OS: Output state of PWM select bit.			
		0	The initial output state is L, this will change to H when timer overflow.		
		1	The initial output state is H, this will change to L when timer overflow.		
0	OV	Overflow bit ( capture & RFC mode only, user should clear this bit after reading) 0: No overflow 1: Overflow			

RFC mode IRQ function

RFC	INT_S	TM3 IRQ ( IRQF bit2 )
0	0	No IRQ
0	1	RFC overflow IRQ
1	0	RFC IRQ
1	1	RFC overflow IRQ

TM3\_LA (\$25h): TM3 data (R/W)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM3_LA	D7	D6	D5	D4	D3	D2	D1	D0

TM3\_CNT(\$26h): TM3 counter (R/W) (up counter)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM3_CNT	D7	D6	D5	D4	D3	D2	D1	D0

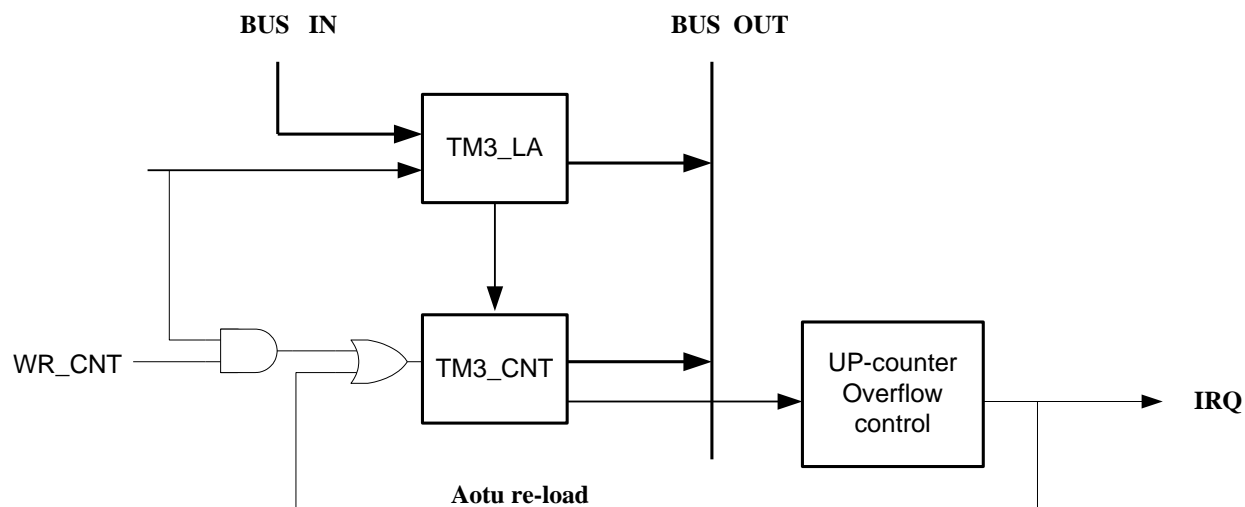


Fig.3.4.24 Block Diagram of TM3 auto-load function

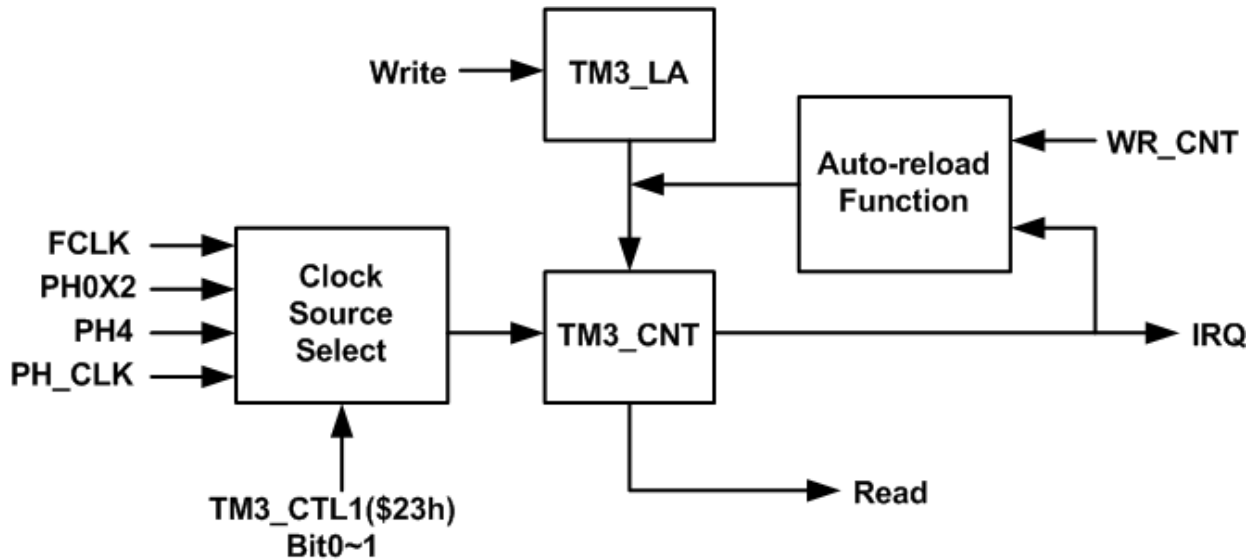


Fig.3.4.25 Block Diagram of TM3 as Timer

Fig.3.4.25-1 TM3 8-bit timer mode example

```

INC      'MK9A160P.inc'    ;; TM3 8-bit timer mode
#DEFINE  RAM_80           80H
ORG      00                ;;
LGOTO    START

INT:     ORG      004
         MOVLA   0x7B      ;; clear tm3
         MOVAM   IRQF
         INC     PD_DAT,m  ;; check TM3 IRQ
         MOVLA   0x010     ;; change tm3 data
         ADD     TM3_LA
         NOP
         IRETI

START:   ORG      100h
         CLR     STATUS
         MOVLA   02h
         MOVAM   LBASDT    ;; Com5~7 work as I/O
         BC     SYS_CTL,b1 ;; Fast clock turn ON
         NOP
         BS     SYS_CTL,7  ;; Cpu clock = FCLK
         NOP
         CLR     RAM_80
         CLR     PA_DAT
         CLR     PD_DAT
         CLR     PA_DIR    ;; PA output
         CLR     PAD_CTL1  ;; PD work as I/O

```

```

CLR    PD_DIR        ;; PD output
MOVLA  B'01000001'   ;; 8-bit timer mode,
MOVAM  TM3_CTL1
MOVLA  B'00000001'
MOVAM  TM3_CTL2
BS     TM3_CTL1,6    ;; write TM3_CNT enable
MOVLA  0x0           ;; TM3 up-counter, 00 → FF
MOVAM  TM3_LA
BC     TM3_CTL1,6
MOVLA  B'00000100'   ;; set TM3 irq mask
MOVAM  IRQM
CLR    IRQF
BS     IRQM_CTL,7
BS     TM3_CTL1,7
LGOTO  $
    
```

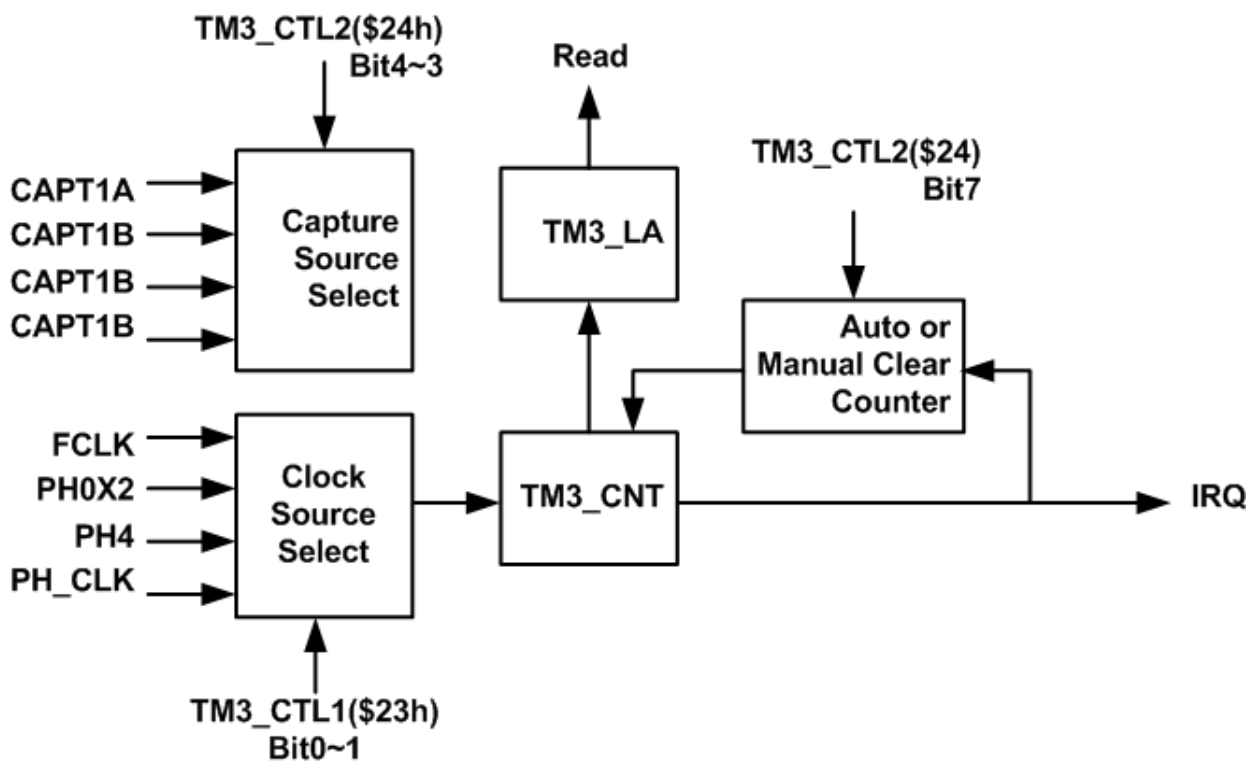


Fig.3.4.26 Block Diagram of TM3 as capture

Fig.3.4.26-1 TM3 8-bit Capture mode example

```

INC     'MK9A160P.inc' ;; TM2 capture
#DEFINE RAM_80 80H
    
```

```

        ORG      00          ;;
        LGOTO   START

INT:    ORG      004
        MOVLA  0x7B        ;; clear tm3
        MOVAM  IRQF
        MOV    TM3_LA      ;; check RFC high byte data
        MOVAM  PD_DAT
        IRETI

        ORG      100h

START:  CLR     STATUS
        MOVLA  02h
        MOVAM  LBASDT      ;; Com5~7 work as I/O
        BC     SYS_CTL,b1  ;; Fast clock turn ON
        NOP
        BS     SYS_CTL,7   ;; Cpu clock = FCLK
        NOP
        CLR    RAM_80
        CLR    PA_DAT
        CLR    PD_DAT
        MOVLA  0xFF
        MOVAM  PA_DIR      ;; PA input
        CLR    PAD_CTL1    ;; PD work as I/O
        CLR    PD_DIR      ;; PD output
        CLR    PC_DIR      ;; PC output
        BC     TM2_CTL1,5  ;; 80bit mode control
        MOVLA  B'00001001' ;; 8-bit capture,couter ph0x2
        MOVAM  TM3_CTL1
        BC     TM3_CTL2,3  ;; capture input = PA3
        ;; BS    TM3_CTL2,3 ;; capture input = PA6
        MOVLA  B'00000100' ;; set TM3 irq mask
        MOVAM  IRQM
        CLR    IRQF
        BS     IRQM_CTL,7
        BS     TM3_CTL1,7
        LGOTO  $

```

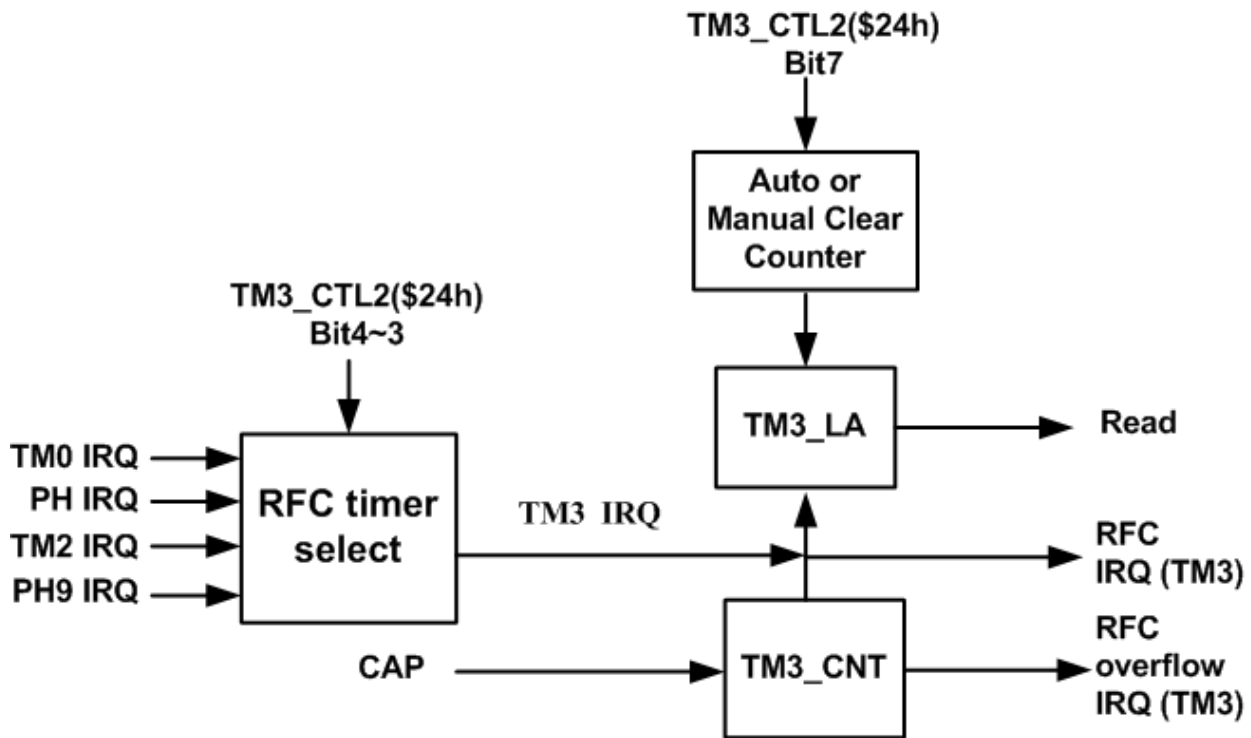


Fig.3.4.27 Block Diagram of TM3 as rfc

Fig.3.4.27-1 TM2 8-bit RFC mode example

```

INC      'MK9A160P.inc'  ;; TM3 RFC
#DEFINE  RAM_80         80H
ORG      00              ;;
LGOTO    START

INT:     ORG            004
         MOVLA         0x7E      ;; clear tm0
         MOVAM         IRQF
         MOV           TM3_LA     ;; check RFC data
         MOVAM         PD_DAT
         IRETI

START:   ORG            100h
         CLR           STATUS
         MOVLA         02h
         MOVAM         LBASDT     ;; Com5~7 work as I/O
         BC           SYS_CTL,b1  ;; Fast clock turn ON
         NOP
         BS           SYS_CTL,7   ;; Cpu clock = FCLK
         NOP
         CLR           RAM_80
         CLR           PA_DAT
         CLR           PD_DAT

```

```

CLR      PAD_CTL1      ;; PD work as I/O
CLR      PD_DIR        ;; PD output
CLR      PC_DIR        ;; PC output

MOVLA    0xFF
MOVAM    PA_DIR        ;; PA input
MOVLA    B'11101111'   ;; RFC,BZ & BZM out
MOVAM    PAD_CTL2
BS       PAD_CTL3,0    ;; RREF ON
;BS      PAD_CTL3,1    ;; SEN0 ON
;BS      PAD_CTL3,2    ;; SEN1 ON
BC       TM2_CTL1,5    ;; 8-bit mode control
MOVLA    B'00011000'   ;; TM2 work as 8-bit RFC
MOVAM    TM3_CTL1     ;; RFC IRQ come from TM0
CLR      TM3_CTL2
MOVLA    B'01000100'   ;; rfc,ph0x2 , T/2:T/2
MOVAM    TM0_CTL
MOVLA    B'00111111'
MOVAM    TM0_LA
BC       TM0_CTL,6

MOVLA    B'00000001'   ;; set TM0 irq mask
MOVAM    IRQM
BS       IRQM_CTL,7
BS       TM3_CTL1,7
BS       TM0_CTL1,7
CLR      IRQF
LGOTO    $

```



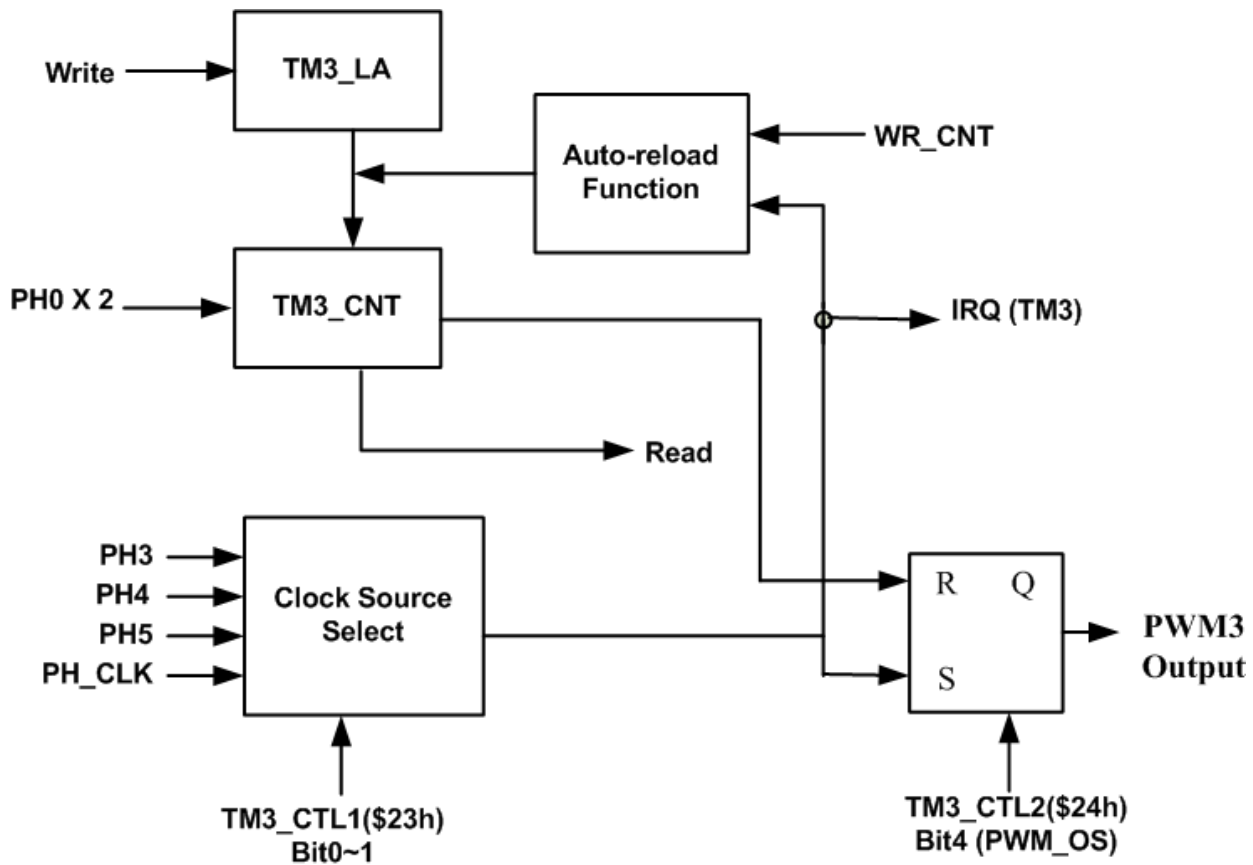


Fig.3.4.28 Block Diagram of TM3 as pwm

Fig.3.4.28-1 PWM2 (TM3 PWM) output example

```

INC      'MK9A160P.inc'    ;; TM3 pwm mode
                                ;; Period=PH7 , Duty cycle come from TM3

#DEFINE  RAM_80  80H
ORG      00
LGOTO   START

INT:
ORG      004
MOVLA   0x7B              ;; clear tm3
MOVAM   IRQF
INC      PD_DAT,m         ;; check TM3 IRQ
MOVLA   0x01
ADD      TM3_LA           ;; change TM3 PWM duty
NOP
IRETI

ORG      100h
START:
CLR      STATUS
MOVLA   02h
MOVAM   LBASDT           ;; Com5~7 work as I/O
BC      SYS_CTL,b1       ;; Fast clock turn ON
NOP
BS      SYS_CTL,7        ;; Cpu clock = FCLK

```

```

NOP
CLR    RAM_80
CLR    PA_DIR        ;; PA output
CLR    PAD_CTL1      ;; PD work as I/O
CLR    PD_DIR        ;; PD output
;;MOVLA B'00110000''  ;; PA3 = PWM2 output
;;MOVAM PAD_CTL2
MOVLA  B'01011000'   ;; pwm mode , pwm duty=ph3
MOVAM  TM3_CTL1
MOVLA  B'00000001'
MOVAM  TM3_CTL2
MOVLA  0xE0          ;; pwm duty counter
MOVAM  TM3_LA
BC     TM3_CTL1,6
NOP
MOVLA  B'00000100'   ;; set TM3 irq mask
MOVAM  IRQM
CLR    IRQF
BS     IRQM_CTL,7
BS     TM3_CTL1,7
LGOTO  $
```

### 3.5 Timer 4&5 (TM4 & TM5)

This two timers are multifunctional which can be set as independently 8 bit timer. The second operation mode is used to be event counters of capture to count external event from CAPT2B,CAPT1B,CAPT4 or CAPT5 pins. They can be used as two 8 bit counters independently . All the functions are setting by below registers and the block diagram are as below:

#### 3.5.1 Timer 4

TM4\_CTL1(\$78h)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM4_CTL1	EN	WR_CNT	BIT	MOD1	MOD0	EDGE	SUR1	SUR0

Bit	Symbol	Description
7	EN	TM4 enable /disable 0: Disable 1: Enable
6	WR_CNT	TM4_CNT would be set through writing data to TM4_LA (Timer ,capture ,pwm & RFC mode ) 0: Disable 1: Enable
5	BIT	BIT : 16-bit/8-bit control
		0     8-bit mode.
		1     16-bit mode , TM4+TM5.
4~3	MOD1~0	MOD1~0: TM4 operation mode selected
		0 0     Timer mode
		0 1     Capture mode
		1 0     RFC mode
		1 1     PWM mode
2	EDGE	Capture signal edge control bit 1:increment when H→L transition on external clock 0:increment when L→H transition on external clock
1~0	SUR1~0	Clock Source <b>(8-bit pwm mode ,PWM duty clock source comes from PH0X2 )</b>
		TIMER,CAPTURE     PWM mod,BIT=0     PWM mod,BIT=1
		Period     Period

	0 0	FCLK (Fast clock)	PH3	FCLK (Fast clock)
	0 1	PH0 X 2	PH4	PH0 X 2
	1 0	PH4	PH5	PH4
	1 1	PH2_CLK	PH2_CLK	PH2_CLK

TM4\_CTL2(\$79h):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM4_CTL2	ENC	CLR_CNT		CAPIN1/ RFC_T1	CAPIN0/ RFC_T0	INT_S	PWM_OS	OV

Bit	Symbol	Description			
7	ENC	Capture & RFC mode: counter auto clear (When overflow) 0: Auto clear counter (Hardware mode ) 1: Clear counter by software.			
6	CLR_CNT	Capture & RFC mode: Clear counter (When ENC=1 and work in capture mode or RFC mode) 0: No clear 1: Clear counter and auto clear CLR_CNT			
4-3	CAPIN1~0/ RFC_T1~0	1. Signal Source Select (Work in capture mode only ) 2. IRQ source select (Work in RFC mode only )			
		mode	Capture mode	RFC mode	
		00	CAPT4 input	TMR0 IRQ	
		01	CAPT1B input	PH IRQ	
		10	CAPT5 input	TMR3 IRQ	
		11	CAPT2B input	PH9	
2	INT_S	Signal Source Select (Work in capture or RFC mode )			
		INT_S	Capture mode	RFC mode RFC=0	RFC mode RFC=1
		0	Capture IRQ	No IRQ	RFC IRQ
		1	Capture overflow IRQ	RFC overflow IRQ	RFC overflow IRQ
1	PWM_OS	PWM_OS: Output state of PWM select bit.			
		0	The initial output state is L, this will change to H when timer overflow.		

		1	The initial output state is H, this will change to L when timer overflow.
0	OV	Overflow bit ( capture & RFC mode, user should clear this bit after reading) 0: No overflow 1: Overflow	

**TM4\_LA (\$7Ah): TM2 data (R/W)**

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM4_LA	D7	D6	D5	D4	D3	D2	D1	D0

**TM4\_CNT(\$7Bh): TM2 counter (R/W) (up counter)**

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM4_CNT	D7	D6	D5	D4	D3	D2	D1	D0



### 3.5.2 Timer 5 (TM5)

TM5\_CTL1(\$7Ch)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM5_CTL1	EN	WR_CNT	--	MOD1	MOD0	EDGE	SUR1	SUR0

Bit	Symbol	Description			
7	EN	TM3 enable /disable 0: Disable 1: Enable			
6	WR_CNT	TM3_CNT would be set through writing data to TM3_LA (Timer ,capture ,pwm & RFC mode ) 0: Disable 1: Enable			
4~3	MOD1~0	MOD1~0: TM3 operation mode selected			
		0 0	Timer mode		
		0 1	Capture mode		
		1 0	RFC mode		
1 1	PWM mode ( TM3 comes from PH0X2 input )				
2	EDGE	Capture signal edge control bit 1:increment when H→L transition on external clock 0:increment when L→H transition on external clock			
1~0	SUR1~0	Clock Source <b>(8-bit pwm mode ,PWM duty clock source comes from PH0X2 )</b>			
				PWM mod,BIT=0	PWM mod,BIT=1
			TIMER,CAPTURE	Period	Duty
		0 0	FCLK (Fast clock)	PH3	FCLK (Fast clock)
		0 1	PH0 X 2	PH4	PH0 X 2
		1 0	PH4	PH5	PH4
		1 1	PH2_CLK	PH2_CLK	PH2_CLK

TM5\_CTL2(\$7Dh):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM5_CTL2	ENC	CLR_CNT		CAPIN1/ RFC_T1	CAPIN0/ RFC_T0	INT_S	PWM_OS	OV

Bit	Symbol	Description
7	ENC	Capture & RFC mode: counter auto clear (When overflow) 0: Auto clear counter (Hardware mode ) 1: Clear counter by software.
6	CLR_CNT	Capture & RFC mode: Clear counter (When ENC=1 and work in capture mode or RFC mode) 0: No clear 1: Clear counter and auto clear CLR_CNT
4~3	CAPIN1~0/ RFC_T1~0	1. Signal Source Select (Work in capture mode only ) 2. IRQ source select (Work in RFC mode only )
		mode      Capture mode      RFC mode
		00      CAPT4 input      TMR0 IRQ
		01      CAPT1B input      PH IRQ
		10      CAPT5input      TMR2 IRQ
11      CAPT2B input      PH9		
2	INT_S	Signal Source Select (Work in capture or RFC mode )
		INT_S      Capture mode      RFC mode      RFC mode
		0 <b>Capture IRQ</b> <b>No IRQ</b> <b>RFC IRQ</b>
		1 <b>Capture overflow IRQ</b> <b>RFC overflow IRQ</b> <b>RFC overflow IRQ</b>
1	PWM_OS	PWM_OS: Output state of PWM select bit.
		0      The initial output state is L, this will change to H when timer overflow.
		1      The initial output state is H, this will change to L when timer overflow.
0	OV	Overflow bit ( capture & RFC mode only, user should clear this bit after reading) 0: No overflow 1: Overflow



**TM5\_LA (\$7Eh): TM3 data (R/W)**

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM5_LA	D7	D6	D5	D4	D3	D2	D1	D0

**TM5\_CNT(\$27F): TM3 counter (R/W) (up counter)**

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM5_CNT	D7	D6	D5	D4	D3	D2	D1	D0

### 3.6 Watchdog Timer (WDT)

WDT is a timer to prevent software from malfunction or jumping to an unknown location with unpredictable result. The source clock of WDT is slow clock. User should enable or disable watchdog timer by configuration register bit 5 (WDTE) at first.

#### WDT\_CTL (\$3Bh):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WDT_CTL	WDTEN	--	--	--	--	PRE 2	PRE 1	PRE0

- Bit7 (WDTEN): Watchdog timer enable/disable bit  
0: WDT disable.  
1: WDT enable.

<Note> Bit6~5 (TEST1/TEST0) are reserved bit for testing

- Bit2~0 (PRE2~0): WDT prescaler assignment bit.

Bit2	Bit1	Bit0	WDT Prescaler rate
PRE2	PRE1	PRE0	
0	0	0	Twdt
0	0	1	Twdt X 2
0	1	0	Twdt X 4
0	1	1	Twdt X 8
1	0	0	Twdt X 16
1	0	1	Twdt X 32
1	1	0	Twdt X 64
1	1	1	Twdt X128 (2'S key reset mode)

CONFIG		OSC Type	Twdt
SOSC1	SOSC0		
0	0	LP (low speed)	Twdt = Tsystem clock X 512
0	1	NO	Twdt = 15.6 mS
1	0	External RC	Twdt = Tsystem clock X 512
1	1	Internal RC	Twdt = 15.6 mS

Table 3.5.1 The relation between slow clock type and WDT

## 4. I/O Port and Other Control Function

### 4.1 I/O port

There are 4 I/O port groups PA, PC, PD and PE to input or output data, each port has different definition and most of them share the pin with other functions. Port A (PA) is bi-direction I/O port which has pull-up, pull-down, open-drain and pin change wake up functions by setting. Port C (PC) is bi-direction I/O port which has pull-down, open-drain and pin change wake up functions by setting. Port D (PD) and Port E (PE) is I/O port with pull-down, open-drain.

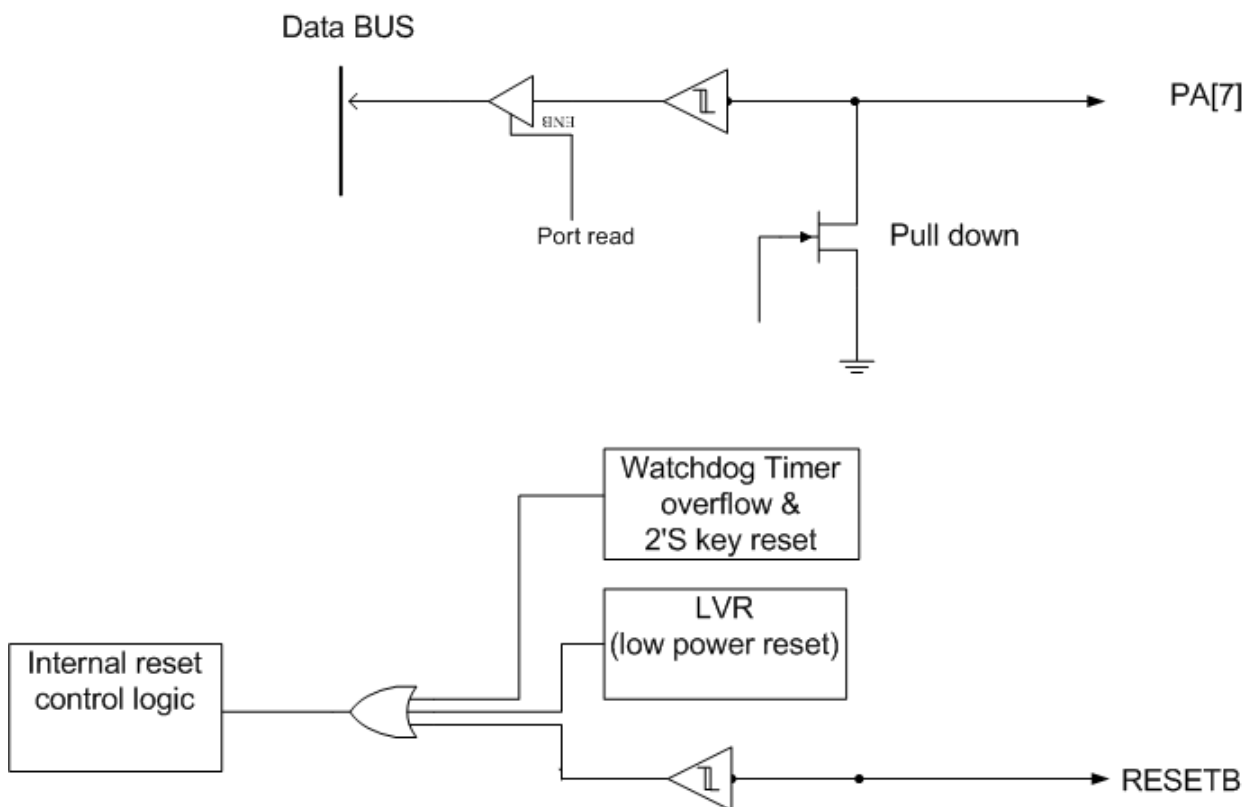
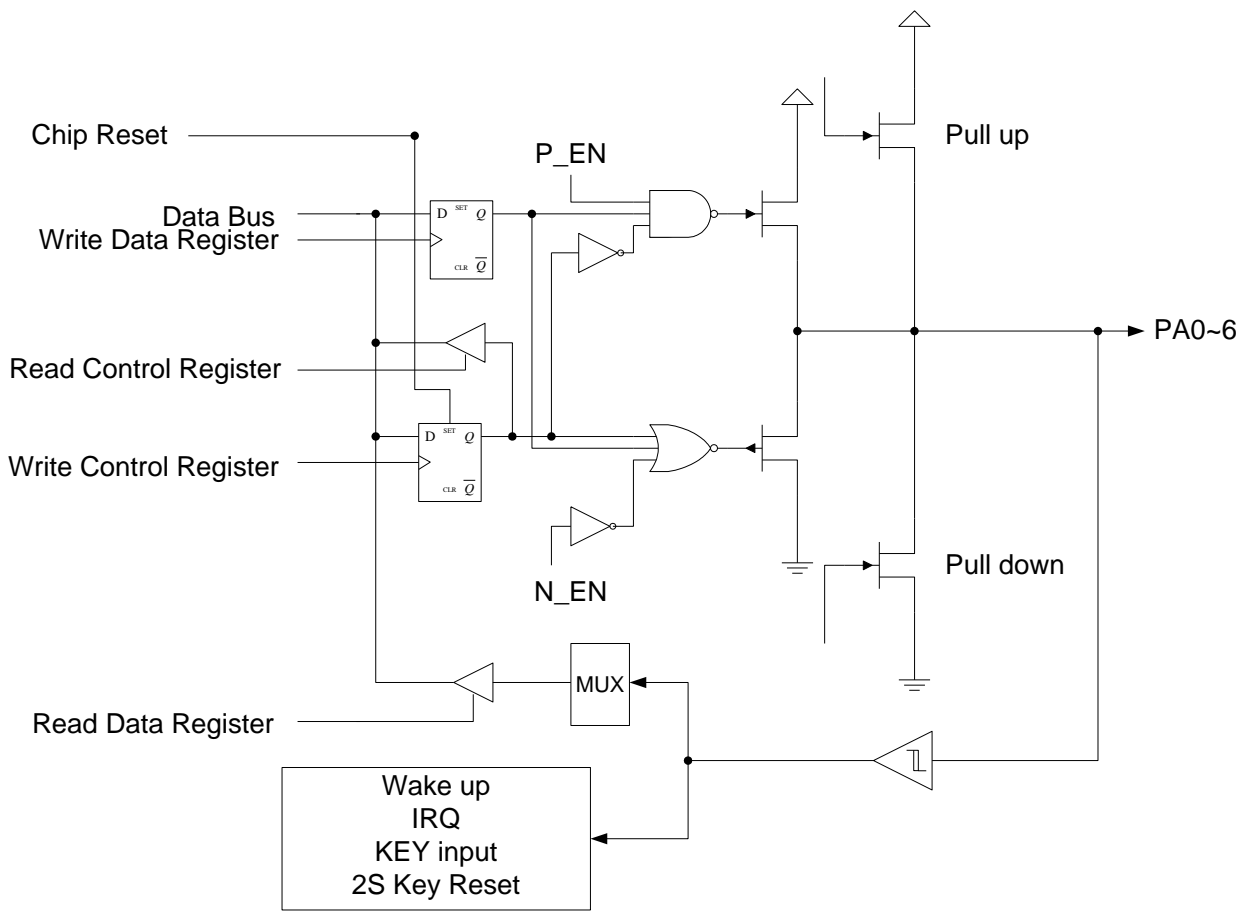


Fig.4.1.1-1 RESETB (PA7) STRUCTURE



**Fig.4.1.1-2 PORT A STRUCTURE**

### 4.1.1. Port A

Port A is the only one bidirectional I/O port in this chip. It has pull up, pull down, open-drain and pin wake up function. It also can replace RESET pin to act as power on reset if user set configuration bit. There are 6 registers to set the 8 I/O ports which are PA\_DIR, PA\_DAT, WAKE\_UP, PA\_PUD1 and PA\_PUD2. The registers are defined as below:

#### PA\_DIR (\$05h):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PA_DIR		CA6	CA5	CA4	CA3	CA2	CA1	CA0

- Bit6~0 (CA6~0): Set PA as input port or output port. (PA[7] only can be input port only)
  - 0: Output port
  - 1: Input port

#### PA\_EDGE (\$2Dh)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PA_EDGE	EDGE7	EDGE6	EDGE5	EDG4	EDG3	EDGE2	EDGE1	EDGE0

- Bit7~0 (EDGE~0): Pin PA7~0 rising/falling wake-up control bits.
  - 1: Falling edge wake-up
  - 0: Rising edge wake-up

### PA\_WAKE\_UP (\$07h)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PA_WAKE_UP	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0

- Bit7~0 (PA7~0): Pin PA7~0 pin change wake-up control bits.  
 0: Wake up disable  
 1: Wake up enable (Only active when PA was set as input at PAD\_CTL1 (\$14h) )

CAn (PA_DIR)	ENn (WAKE_UP)	KI (Pin-edge wake-up function)
1	1	ON
X	0	OFF
0	1	OFF

### A\_PUD1 (\$08h) & PA\_PUD2 (\$09h)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PA_PUD1	A3-2	A3-1	A2-2	A2-1	A1-2	A1-1	A0-2	A0-1
PA_PUD2	--	A7-1	A6-2	A6-1	A5-2	A5-1	A4-2	A4-1

- When set PA as I/O port (PAD\_CTL2) and direction is input (PA\_DIR), then  
 An-1=1 -> PAn pull down  
 An-2=1 -> PAn pull up
- When set PA as I/O port (PAD\_CTL2) and direction is output (PA\_DIR), then  
 An-1=1 -> PMOS open drain on  
 An-2=1 -> NMOS open drain on

These two registers are used to set PA to have pull-up or pull-down resistor. But this only active with PAD\_CTL2 was set as I/O port and PA\_DIR was set as input. The relation between them are as below table.

CAn (PA_DIR)	An-2	An-1	PULL-UP	PULL-DOWN	PMOS OPEN-DRAIN	NMOS OPEN-DRAIN	Description PAn (n=0~6)
1	0	0	OFF	OFF	OFF	OFF	PAn is input port
1	0	1	OFF	ON	OFF	OFF	PAn is input port
1	1	X	ON	OFF	OFF	OFF	PAn is input port
0	0	0	OFF	OFF	ON	ON	PAn is normal output
0	0	1	OFF	OFF	ON	OFF	PAn is nmos open -drain output port
0	1	X	OFF	OFF	OFF	ON	PAn is pmos open -drain output port

<Note> (1) PA[7] is shared the pin with RESETB which can be used as input port only. So, there is only pull down function can be set as below table.

(2) Others# include (ELP,ELC,PWM,REM,BZ,BZM)

(3) I/O# include (PA I/O mode,CAPT1A & CAPT1B)

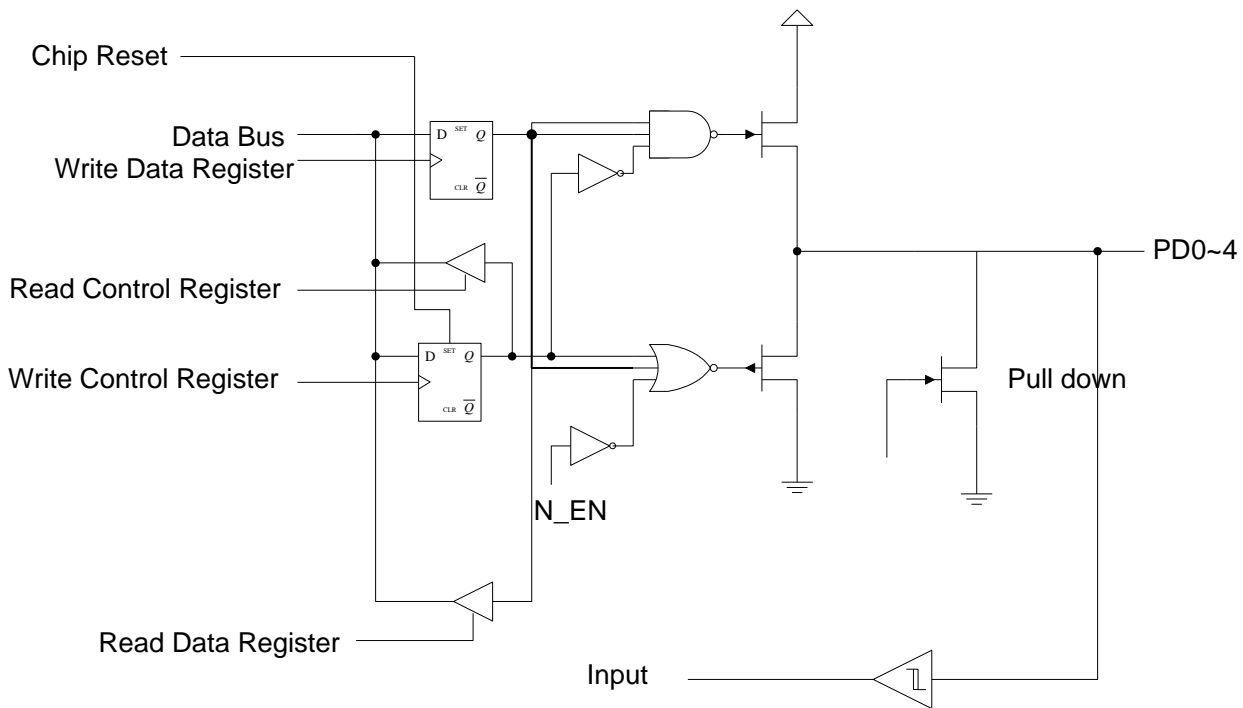
A7-2	A7-1	PULL-DOWN	PMOS OPEN-DRAIN	NMOS OPEN-DRAIN	PA7
0	0	OFF	OFF	OFF	PA7 is input port
0	1	ON	OFF	OFF	PA7 is input port
1	X	OFF	OFF	OFF	PA7 is input port

PA\_DAT(\$0Ah):

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

- Bit6~0 (PA6~0): PA input or output data bits.
- Bit7: Input data only.

<Note> PA[5]~PA[7] share the pin with OSCOUT, OSCIN and RESETB. If user want to use these pins as PA, then the Configuration bit 3~2 (FOSC1 and FOSC0) must set to (0,1). And bit 8 (RST\_DEF) must set to 0 as normal input port.



**Fig.4.1.2 PORT C STRUCTURE**

### 4.1.2 Port C

There are 2 registers to set the attribute of Port C which are PC\_CTL and PC\_DAT. Port C is input port only when it was used as I/O port.

#### PC\_EDGE (\$1Eh)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PC_EDGE	EDGE7	EDGE6	EDGE5	EDG4	EDG3	EDGE2	EDGE1	EDGE0

- Bit7~0 (EDGE~0): Pin PC7~0 rising/falling wake-up control bits.
  - 1: Falling edge wake-up
  - 0: Rising edge wake-up

#### PC\_WAKE\_UP (\$1Dh)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PC_WAKE_UP	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0

- Bit7~0 (EN7~0): Pin PC7~0 pin change wake-up control bits.
  - 0: Wake up disable
  - 1: Wake up enable (Only active when PA was set as input at PAD\_CTL1 (\$14h) )

CAn (PC_DIR)	ENn (WAKE_UP)	KI (Pin-edge wake-up function)
1	1	ON
X	0	OFF
0	1	OFF

**PC\_CTL (\$0Bh): (R/W) (default =0000000b )**

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PC_CTL	KI7/IO7	KI6/IO6	KI5/IO5	KI4/IO4	KI3/IO3	KI2/IO2	KI1/IO1	KI0/IO0

- Bit1~0 (KI<sub>n</sub>/IO<sub>n</sub>): PC7~0 work as KI or I/O input mode (input only )  
 0: PA as I/O input mode ,use pin-edge IRQ.(normal , halt mode and 2'S key reset function )  
 1: PA as KI input mode ,use KEY wake-up IRQ.( normal or halt mode )

PC_CTL	PC_DIR	WAKE_UP	FUNCTION
KI <sub>n</sub> /IO <sub>n</sub>	CAn	ENn	
1	X	X	Key input mode
0	1	1	Input pin-edge wake up mode

**PC\_DIR (\$0Ch): (R/W) (default =0000000b )**

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PC_DIR	DC7	DC6	DC5	DC4	DC3	DC2	DC1	DC0

- Bit4~0 (DC7~0): Set PC as input port or output port.  
 0: Output port  
 1: Input port

**PC\_PUD (\$0Dh) : (R/W) (default =0000000b )**

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PC_PUD	CD7	CD6	CD5	CD4	CD3	CD2	CD1	CD0

<Note> When these shared pins are set as PC, there are PMOS open drain or NMOS open drain can be selected. The setting method is as below table:

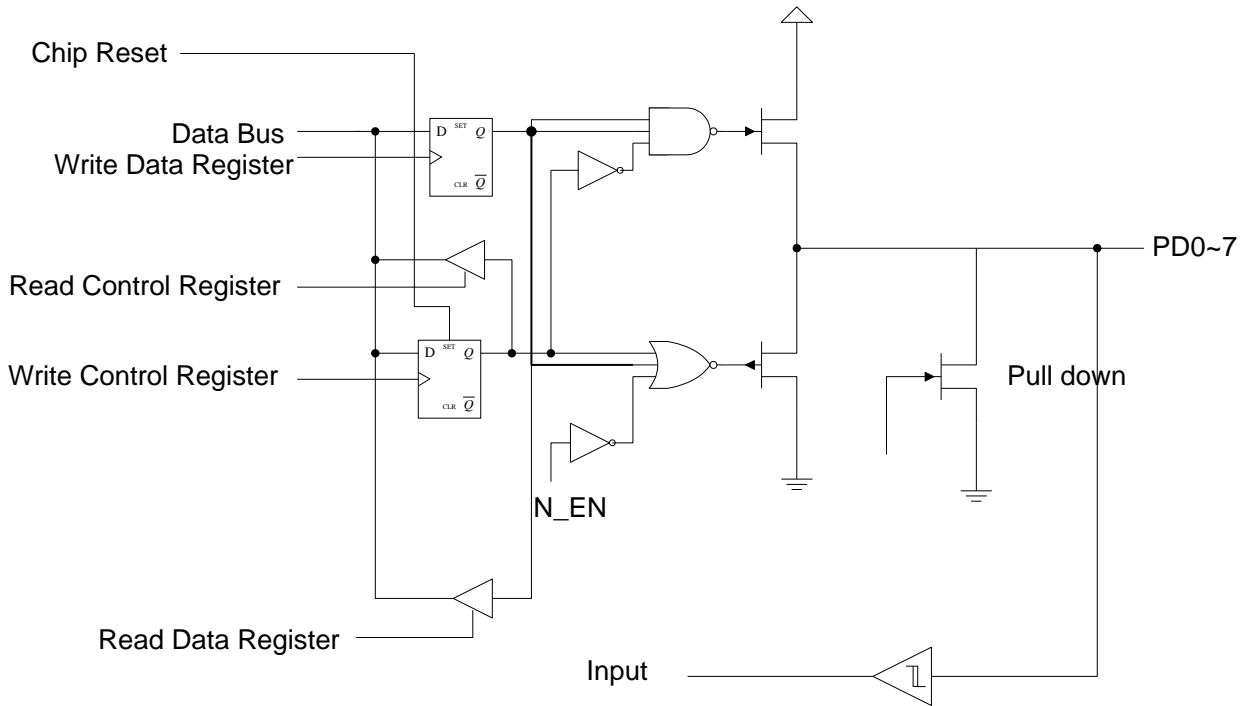
DDn (PC_DIR)	Dn-1	PULL-DOWN	PMOS OPEN-DRAIN	Description PCn (n=0~7)
1	0	OFF	OFF	PCn is input port
1	1	ON	OFF	PCn is input port
0	0	ON	ON	PCn is normal output
0	1	OFF	ON	PCn is pmos open drain output port



**PC\_DAT(\$0Eh): (R/W) (default =0000000b )**

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PC_DAT	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0

- Bit7~0 (PC7~0): PC input data bits.



**Fig.4.1.3 PORT D STRUCTURE**

**PC\_CTL (\$11h): (R/W) (default =0000000b )**

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PC_CTL	KI7/IO7	KI6/IO6	KI5/IO5	--	--	KI2/IO2	KI1/IO1	KI0/IO0

- Bit4~3 (DC7~0): PD work as KEY or I/O input mode (input only )  
 0: I/O input port  
 1: Key Input port

PORT	KIn/ IOn	PAD_CTL8	STROBE Bit6 EN	PC_DIR	PC_PUD	condition	PULL- DOWN	OUT	Description PDn (n=0~6)
PC0	KI	--	1	X	X	COM7 is off	ON	OFF	PC0 is avail key input
			0				OFF		Key function is off. No power consumption.
PC1	KI	--	1	X	X	COM6 is off	ON	OFF	PC1 is avail key input
			0				OFF		Key function is off. No power consumption.
PC2	KI	--	1	X	X	COM5 is off	ON	OFF	PC2 is avail key input
			0				OFF		Key function is off. No power consumption.
PC5	KI	--	1	X	X	COM8 is off	ON	OFF	PC5 is avail data input
			0				OFF		Key function is off. No power consumption.
PC6	KI	Bit1~0=00	1	X	X	SEG33 is off PWM4 is off	ON	OFF	PC6 is avail data input
			0				OFF		Key function is off. No power consumption.
PC7	KI	Bit3~2=00	1	X	X	SEG34 is off PWM5 is off	ON	OFF	PC7 is avail data input
			0				OFF		Key function is off. No power consumption.

### 4.1.3 Port D

There are 3 registers to set the attribute of Port D which are PD\_DIR, PD\_PUD and PD\_DAT.

Port D is output port only when it was used as I/O port.

#### PD\_DIR (\$0Fh): (R/W) (default =1111111b )

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PD_DIR	DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0

- Bit7~0 (DD7~0): Set PD as input port or output port.  
 0: Output port  
 1: Input port

#### PD\_PUD (\$10h) : (R/W) (default =00000000b )

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PD_PUD	CD7	CD6	CD5	CD4	CD3	CD2	CD1	CD0

<Note> When these shared pins are set as PD, there are PMOS open drain or NMOS open drain can be selected. The setting method is as below table:

DDn (PD_DIR)	Dn-1	PULL- DOWN	PMOS OPEN- DRAIN	Description PDn (n=0~5)
1	0	OFF	OFF	PDn is input port
1	1	ON	OFF	PDn is input port
0	0	ON	ON	PDn is normal output
0	1	OFF	ON	PDn is pmos open drain output port

#### PD\_CTL (\$11h): (R/W) (default =00000000b )

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PD_CTL	KI7/IO7	KI6/IO6	KI5/IO5	KI4/IO4	KI3/IO3	KI2/IO2	KI1/IO1	KI0/IO0

- Bit4~3 (DC7~0): PD work as KEY or I/O input mode (input only )  
 0: I/O input port  
 1: Key Input port

PORT	Kn/ IO <sub>n</sub>	PAD_CTL1 BIT <sub>n</sub>	STROBE Bit6 EN	PD_DIR	PD_PUD	condition	PULL- DOWN	OUT	Description PD <sub>n</sub> (n=0~6)
PD0	KI	0 SEG OFF	1	X	X	PAD_CTL6 Bit4 CP_EN=0	ON	OFF	PD0 is avail key input
			0				OFF		Key function is off. No power consumption.
PD1	KI	0 SEG OFF	1	X	X	PAD_CTL6 Bit4 CP_EN=0	ON	OFF	PD1 is avail key input
			0				OFF		Key function is off. No power consumption.
PD2	KI	0 SEG OFF	1	X	X	PAD_CTL6 Bit5 CPO_EN=0	ON	OFF	PD2 is avail key input
			0				OFF		Key function is off. No power consumption.
PD3	KI	0 SEG OFF	1	X	X	PAD_CTL6 Bit0=0	ON	OFF	PD3 is avail key input
			0				OFF		Key function is off. No power consumption.
PD4	KI	0 SEG OFF	1	X	X	--	ON	OFF	PD4 is avail key input
			0				OFF		Key function is off. No power consumption.
PD5	KI	0 SEG OFF	1	X	X	--	ON	OFF	PD5 is avail data input
			0				OFF		Key function is off. No power consumption.
PD6	KI	0 SEG OFF	1	X	X	PH_CTL Bit5 EL_P=0	ON	OFF	PD6 is avail data input
			0				OFF		Key function is off. No power consumption.
PD7	KI	0 SEG OFF	1	X	X	PH_CTL Bit5 EL_P=0	ON	OFF	PD7 is avail data input
			0				OFF		Key function is off. No power consumption.

**PD\_DAT(\$12h):**

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PD_DAT	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0

- Bit7~0 (PD7~0): PD output data bits.

#### 4.1.4 Port E

There are 3 registers to set the attribute of Port E which are PE\_DIR, PE\_PUD and PE\_DAT.

Port E is output port only when it was used as I/O port.

##### PE\_DIR (\$1Ah): (R/W) (default =1111111b )

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PE_DIR	DE7	DE6	DE5	DE4	DE3	DE2	DE1	DE0

5. Bit7~0 (DE7~0): Set PD as input port or output port.  
 0: Output port  
 1: Input port

##### PE\_PUD (\$1Bh) : (R/W) (default =00000000b )

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PE_PUD	CE7	CE6	CE5	CE4	CE3	CE2	CE1	CE0

<Note> When these shared pins are set as PD, there are PMOS open drain or NMOS open drain can be selected. The setting method is as below table:

D <sub>En</sub> (PE_DIR)	D <sub>n-1</sub>	PULL-DOWN	PMOS OPEN-DRAIN	Description PE <sub>n</sub> (n=0~7)
1	0	OFF	OFF	PE <sub>n</sub> is input port
1	1	ON	OFF	PE <sub>n</sub> is input port
0	0	ON	ON	PE <sub>n</sub> is normal output
0	1	OFF	ON	PE <sub>n</sub> is pmos open drain output port

##### PE\_DAT(\$1Ch): (R/W) (default =00000000b )

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PE_DAT	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0

6. Bit7~0 (PE7~0): PE output data bits.

#### 4.1.5 Port F

There are 3 registers to set the attribute of Port E which are PE\_DIR, PE\_PUD and PE\_DAT.

Port E is output port only when it was used as I/O port.

##### PF\_DIR (\$1Ah): (R/W) (default =1111111b )

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PF_DIR	DF7	DF6	DF5	DF4	DF3	DF2	DF1	DF0

7. Bit7~0 (DF7~0): Set PD as input port or output port.  
 0: Output port  
 1: Input port

##### PF\_PUD (\$1Bh) : (R/W) (default =00000000b )

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PF_PUD	CF7	CF6	CF5	CF4	CF3	CF2	CF1	CF0

<Note> When these shared pins are set as PD, there are PMOS open drain or NMOS open drain can be selected. The setting method is as below table:

D <sub>En</sub> (PF_DIR)	D <sub>n-1</sub>	PULL-DOWN	PMOS OPEN-DRAIN	Description PE <sub>Dn</sub> (n=0~7)
1	0	OFF	OFF	PF <sub>n</sub> is input port
1	1	ON	OFF	PF <sub>n</sub> is input port
0	0	ON	ON	PF <sub>n</sub> is normal output
0	1	OFF	ON	PF <sub>n</sub> is pmos open drain output port

##### PF\_DAT(\$1Ch): (R/W) (default =00000000b )

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PF_DAT	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0

8. Bit7~0 (PF7~0): PE output data bits.

#### 4.1.6 Port B

There are 3 registers to set the attribute of Port B which are PB\_DIR, PB\_PUD and PB\_DAT.

Port E is output port only when it was used as I/O port.

##### PB\_DIR (\$71h): (R/W) (default =1111111b) (Special Purpose Register 2)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PB_DIR	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0

- Bit7~0 (DB7~0): Set PB as input port or output port.  
 0: Output port  
 1: Input port

##### PB\_PUD (\$72h) : (R/W) (default =0000000b) (Special Purpose Register 2)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PB_PUD	CB7	CB6	CB5	CB4	CB3	CB2	CB1	CB0

<Note> When these shared pins are set as PD, there are PMOS open drain or NMOS open drain can be selected. The setting method is as below table:

D <sub>En</sub> (PB_DIR)	D <sub>n-1</sub>	PULL-DOWN	PMOS OPEN-DRAIN	Description PB <sub>n</sub> (n=0~7)
1	0	OFF	OFF	PB <sub>n</sub> is input port
1	1	ON	OFF	PB <sub>n</sub> is input port
0	0	ON	ON	PB <sub>n</sub> is normal output
0	1	OFF	ON	PB <sub>n</sub> is pmos open drain output port

##### PB\_DAT(\$73h): (R/W) (default =0000000b) (Special Purpose Register 2)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PB_DAT	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0

- Bit7~0 (PB7~0): PB output data bits.

## 4.2 Define Shared Pin

PAD\_CTL1 (\$13h): (R/W) (default =0000000b )

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PAD_CTL1	SEG64/ PD[7]	SEG63/ PD[6]	SEG62/ PD[5]	SEG61/ PD[4]	SEG60/ PD[3]	SEG59/ PD[2]	SEG58/ PD[1]	SG57/ PD[0]

<Note> This register is used to set the shared pin function which can be set as below table.

Bit	Bitn=1	Bitn=0
0	SEG57	PD[0] / CP1+
1	SEG58	PD[1] / CP1-
2	SEG59	PD[2] / CPO
3	SEG60	PD[3] /PWM3
4	SEG61	PD[4] + CAPT2A
5	SEG62	PD[5] + INT
6	SEG63	PD[6] / ELP
7	SEG64	PD[7] / ELC

Bit \value	10	X1	00
(PH_CTL.EL_P),B6	ELP	SEG63	PD6
(PH_CTL.EL_P),B7	ELC	SEG64	PD7

PAD\_CTL6

(\$29h): (R/W) (default =0000000b )

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PAD_CTL6	CP_EN	CP_OUT	CPO_EN	CP_S1	CP_S0	PF3C	PF1C	PWM3/ PD[3]

<Note> This register is used to set the shared pin function which can be set as below table.

Bit	Bitn=1	Bitn=0
1	PF[1] output from /PF[0] input	PF[1] normal output
2	PF[3] output from /PF[2] input	PF[3] normal output
0	PWM3	PD[3]
5	PD2 work as CPO_OUT	PD2 work as I/O
6	CPO_OUT (Comparator output)	
7	enable	disable



Bit \value	11	10	01	00
CP_S1.CP_S0	CP3+ CP3-	CP2+ CP2-	CP1+ CP1-	No comparator

**PAD\_CTL2 (\$14h): (R/W) (default =00000000b )**

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PAD_CTL2		C6	C5	C4	C3	C2	C1	C0

<Note> This register is used to set the shared pin function which can be set as below table.

Bit \value	0	1
C0	PA0	CAP
C1	PA1	REF
C2	PA2	SEN0
C3	PA4	BZ
C6	PA5	BZM

Bit \value	11	10	01	00
C5-C4	PWM2	SEN1	REM	PA3+CAPT1A

#### 4.2-1 PD[5] wake-up. ( IRQ )

```
#INCLUDE "MK9A160P.INC"           ;; PD[5] pin edge wake-up.
                                     ;; sleep mode or halt mode

                                     ORG      0x00
                                     LGOTO   INITIAL
INT:    ORG      0x04
```

#### PAD\_CTL3 (\$15h): (R/W) (default =0000000b )

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PAD_CTL3	EDGE	--	SEN21_ON	SEN20_ON	REF2_ON	SEN1_ON	SEN0_ON	REF_ON

<Note> This register is used to set the shared pin function which can be set as below table.

- Bit7 (EDGE): PD[5] pin-interrupt control  
0: Rising edge  
1: Falling edge
- Bit5 (SEN21\_ON): Set SEN21 pin output control (RFC2 mode only)  
0: Input disable  
1: input enable
- Bit4 (SEN20\_ON): Set SEN20 pin output control (RFC2 mode only)  
0: Input disable  
1: input enable
- Bit3 (REF2\_ON): Set REF2 pin output control (RFC2 mode only)  
0: Input disable  
1: input enable
- Bit2 (SEN1\_ON): Set SEN1 pin output control (RFC mode only)  
0: Input disable  
1: input enable
- Bit1 (SEN0\_ON): Set SEN0 pin output control (RFC mode only)  
0: Input disable  
1: input enable
- Bit0 (REF\_ON): Set REF pin output control (RFC mode only)  
0: Input disable  
1: input enable

**PAD\_CTL4 (\$16h): (R/W) (default =0000000b )**

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PAD_CTL4	SEG56/ PE[7]	SEG55/ PE[6]	SEG54/ PE[5]	SEG53/ PE[4]	SEG52/ PE[3]	SEG51/ PE[2]	SEG50/ PE[1]	SG49/ PE[0]

Bit	Bitn=1	Bitn=0
0	SEG49	PE[0] / SSB
1	SEG50	PE[1] / SCLK
2	SEG51	PE[2] / SMOSI
3	SEG52	PE[3] / SMISO
4	SEG53	PE[4] + CAPT5 / SEN21
5	SEG54	PE[5] / SEN20
6	SEG55	PE[6] / REF2
7	SEG56	PE[7] +CAPT4 / CAP2

**PAD\_CTL5 (\$28h): (R/W) (default =0000000b )**

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PAD_CTL5	CAP2/ PE[7]	REF2/ PE[6]	SEN20/ PE[5]	SEN21/ PE[4]	SMISO/ PE[3]	SMOSI/ PE[2]	SCLK/ PE[1]	SSB/ PE[0]

<Note> This register is used to set the shared pin function which can be set as below table.

Bit	Bitn=1	Bitn=0
0	SSB	PE[0]
1	SCLK	PE[1]
2	SMOSI	PE[2]
3	SMISO	PE[3]
4	SEN21	PE[4] + CAPT5
5	SEN20	PE[5]
6	REF2	PE[6]
7	CAP2	PE[7] + CAPT4

**PAD\_CTL6 (\$29h): (R/W) (default =0000000b )**

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PAD_CTL6	CP_EN	CP_OUT	CPO_EN	CP_S1	CP_S0	PF3C	PF1C	PWM3/ PD[3]

<Note> This register is used to set the shared pin function which can be set as below table.

Bit	Bitn=1	Bitn=0
1	PF[1] output from PF[0] input	PF[1] normal output
2	PF[3] output from /PF[2] input	PF[3] normal output
0	PWM3	PD[3]
5	PD2 work as CPO_OUT	PD2 work as I/O
6	CPO_OUT (Comparator output)	
7	enable	disable

Bit \value	11	10	01	00
CP_S1.CP_S0	CP3+ CP3-	CP2+ CP2-	CP1+ CP1-	No comparator

PORT	KIn/ IOIn	PAD_CTL1 BITn	STROBE Bit6 EN	PD_DIR	PD_PUD	condition	PULL- DOWN	OUT	Description PDn (n=0~6)
PD0	KI	0 SEG OFF	1	X	X	PAD_CTL6 Bit4~3 CP_S1~0/=01	ON	OFF	PD0 is avail key input
			0				OFF		Key function is off. No power consumption.
PD1	KI	0 SEG OFF	1	X	X	PAD_CTL6 Bit4~3 CP_S1~0/=01	ON	OFF	PD1 is avail key input
			0				OFF		Key function is off. No power consumption.
PD2	KI	0 SEG OFF	1	X	X	PAD_CTL6 Bit5 CPO_EN=0	ON	OFF	PD2 is avail key input
			0				OFF		Key function is off. No power consumption.
PD3	KI	0 SEG OFF	1	X	X	PAD_CTL6 Bit0=0	ON	OFF	PD3 is avail key input
			0				OFF		Key function is off. No power consumption.
PD4	KI	0	1	X	X	--	ON	OFF	PD4 is avail key input

		SEG OFF	0				OFF		Key function is off. No power consumption.
PD5	KI	0 SEG OFF	1	X	X	--	ON	OFF	PD5 is avail data input
			0				OFF		Key function is off. No power consumption.
PD6	KI	0 SEG OFF	1	X	X	PH_CTL Bit5 EL_P=0	ON	OFF	PD6 is avail data input
			0				OFF		Key function is off. No power consumption.
PD7	KI	0 SEG OFF	1	X	X	PH_CTL Bit5 EL_P=0	ON	OFF	PD7 is avail data input
			0				OFF		Key function is off. No power consumption.

PORT	KIn/ IOn	PAD_CTL8	STROBE Bit6 EN	PC_DIR	PC_PUD	condition	PULL- DOWN	OUT	Description PDn (n=0~6)
PC0	KI	--	1	X	X	COM7 is off	ON	OFF	PC0 is avail key input
			0				OFF		Key function is off. No power consumption.
PC1	KI	--	1	X	X	COM6 is off	ON	OFF	PC1 is avail key input
			0				OFF		Key function is off. No power consumption.
PC2	KI	--	1	X	X	COM5 is off	ON	OFF	PC2 is avail key input
			0				OFF		Key function is off. No power consumption.
PC5	KI	--	1	X	X	COM8 is off	ON	OFF	PC5 is avail data input
			0				OFF		Key function is off. No power consumption.
PC6	KI	Bit1~0=00	1	X	X	SEG33 is off PWM4 is off	ON	OFF	PC6 is avail data input
			0				OFF		Key function is off. No power consumption.
PC7	KI	Bit3~2=00	1	X	X	SEG34 is off PWM5 is off	ON	OFF	PC7 is avail data input
			0				OFF		Key function is off.

									No power consumption.
--	--	--	--	--	--	--	--	--	-----------------------

**PAD\_CTL8 (\$06h): (R/W) (default =0000000b )**

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PAD_CTL8	OP2_SL	OP2_EN	OP1_SL	OP1_EN	SEG48/ PC[7]	PWM5/ PC[7]	SEG47/ PC[6]	PWM4/ PC[6]

<Note> This register is used to set the shared pin function which can be set as below table.

Bit	00	01	10	11
1.0	PC[6]	PWM4	SEG47	SEG47
3.2	PC[7]	PWM5	SEG48	SEG48

Bit	00	01	10	11
5.4	OP1 off	OP1 enable	SEG44,45,46	SEG44,45,46
7.6	OP2 off	OP2 enable	SEG41,42,43	SEG41,42,43

**PAD\_CTL9 (\$70h): (R/W) (default =0000000b ) ( *Special Purpose Register 2* )**

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PAD_CTL9	SEG40/ PB[7]	SEG39/ PB[6]	SEG38/ PB[5]	SEG37/ PB[4]	SEG36/ PB[3]	SEG35/ PB[2]	SEG34/ PB[1]	SEG33/ PB[0]

<Note> This register is used to set the shared pin function which can be set as below table.

Bit	Bitn=1	Bitn=0
0	SEG33	PB[0]
1	SEG34	PB[1]
2	SEG35	PB[2]
3	SEG36	PB[3]
4	SEG37	PB[4]
5	SEG38	PB[5]
6	SEG39	PB[6]
7	SEG40	PB[7]

### 4.3 Key Strobe Function

The key scanning function used partial of segment frame frequency cycle to output scanning timing. This function be limited to normal mode or halt mode.

The register bits are defined as below:

#### STROBE(\$34h): Strobe Control (R/W) (default =000x0000b )

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
STROBE	FRAME	EN	KOAEN	KOEN	KO3	KO2	KO1	KO0

MODE/FUNCTION	SEG1~16	PORT PC1~0,PD0~7
EN	X	Pull down enable
KOAEN	SEG1~16 : Hi output	X
KOEN	SEGN : Hi output Others : Floating	X

- **Bit7 (FRAME): FRAME**
  - 0: Key input data is invalid
  - 1: Key input data is valid
- **Bit6 (EN): Key function enable**
  - 0: disable
  - 1: enable
- **Bit5 (KOAEN): Segment out control**
  - 0: SEG1~16 : LCD output
  - 1: SEG1~16 : Hi output
- **Bit4 (KOEN): Segment out control**
  - 0: SEG1~16 : LCD output
  - 1: SEGN : Hi output ,other's SEG : floating output

KIn/ PCn	CAn	An-2	An-1	STBEN	HSCNE	PULL- UP	PULL- DOWN	PMOS OPEN- DRAIN	NMOS OPEN- DRAIN	Description PAn (n=0~6)
I , KI	1	0	0	1	X	OFF	ON	OFF	OFF	PAn is KI pull-down input
I , KI	1	0	0	0	1	OFF	ON	OFF	OFF	PAn is KI pull-down input
I , KI	1	0	0	0	0	OFF	OFF	OFF	OFF	PAn is floating

PA\_CTL : PA KI / INPUT (I/O) control register

PA\_dir : PA IN / OUT control register



- Bit3~0(KO3~0): Select which strobe pin to output signal as the table. (f – floating )

Bit3~0	KS1	KS2	KS3	KS4	KS5	KS6	KS7	KS8	KS9	KS10	KS11	KS12	KS13	KS14	KS15	KS16
0000	HI	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f
0001	f	HI	f	f	f	f	f	f	f	f	f	f	f	f	f	f
0010	f	f	HI	f	f	f	f	f	f	f	f	f	f	f	f	f
0011	f	f	f	HI	f	f	f	f	f	f	f	f	f	f	f	f
0100	f	f	f	f	HI	f	f	f	f	f	f	f	f	f	f	f
0101	f	f	f	f	f	HI	f	f	f	f	f	f	f	f	f	f
0110	f	f	f	f	f	f	HI	f	f	f	f	f	f	f	f	f
0111	f	f	f	f	f	f	f	HI	f	f	f	f	f	f	f	f
1000	f	f	f	f	f	f	f	f	HI	f	f	f	f	f	f	f
1001	f	f	f	f	f	f	f	f	f	HI	f	f	f	f	f	f
1010	f	f	f	f	f	f	f	f	f	f	HI	f	f	f	f	f
1011	f	f	f	f	f	f	f	f	f	f	f	HI	f	f	f	f
1100	f	f	f	f	f	f	f	f	f	f	f	f	HI	f	f	f
1101	f	f	f	f	f	f	f	f	f	f	f	f	f	HI	f	f
1110	f	f	f	f	f	f	f	f	f	f	f	f	f	f	HI	f
1111	f	f	F	f	f	f	f	f	f	f	f	f	f	f	f	HI

**Software mode example. (PD[4:3]-input ,SEG[16:1] – output )**

```
#INCLUDE "MK9A160P.INC"                ;; normal only
                                        ;; PD[4:3] input with pull-down (KI mode)
                                        ;; SEG[16:1] open drain output

#DEFINE KEY_DATA      80h
#DEFINE AVAIL_KEY     81h                ;; avail key
#DEFINE KEY_NUM       82h                ;; KEY scan loop number
      ORG              0x00
      LGOTO            INITIAL

ORG              0x04
      .....
      IRET

KEY      MOVLA        B'01100000'      ;;
      MOVAM           STROBE           ;; SEG[16:1]=hi output , others are floating
                                        ;; PD[4:3] input with pull-down
```

```

;; PD[7:5] & PD[2:0] input floating.
MOV      PD_DAT,a
MOVAM    KEY_DAT
MOV      AVAIL_KEY,a
AND      KEY_DAT
TMSC     KEY_DAT
LCALL    KEY_SCAN
LGOTO    KEY_END    ;; no key input
KEY_SCAN
MOVLA    B'01010000' ;;
MOVAM    STROBE     ;; SEG[0]=hi output , others are floating
;; key matrix
KEY_LOOP ;; NOP      ;; wait for key loading
MOV      PD_DAT,a   ;; read PD[4:3] input
MOVAM    KEY_DAT,m
MOV      AVAIL_KEY,a
AND      KEY_DAT,m  ;; avail key data
TMSC     KEY_DAT
LGOTO    KEY_IN     ;; valid key
INC      STROBE,m
BTSS     STROBE,b6  ;; scan key over
LGOTO    KEY_LOOP
KEY_IN   .....
KEY_END  .....
MOVLA    B'00000000' ;; enable hardware key scan
MOVAM    STROBE     ;; PD[4:3] pull-down & input buffer turn OFF
;; Press KEY has no power consumption
RET
ORG      100h      ;; main program,press KEY has no power
                    consumption
INITIAL  CLR        STATUS
MOVLA    B'00011111' ;; KEY number =16 (SEG16~1)
MOVAM    KEY_NUM
MOVLA    B'00011000' ;; PD[4:3]
MOVAM    AVAIL_KEY
MOVLA    B'00000011' ;; 5 X COM , PC[1]=COM[6] , PC[0]=COM[7]

```

```

MOVAM    LBASDT
MOVLA    B'11100111'    ;; PD[3]=SEG[23],PD[4]=SEG[24]
MOVAM    PAD_CTL1
MOVLA    B'11111111'    ;; PD work as input mode
MOVAM    PD_DIR
MOVLA    B'00011000'    ;; PD[4:3] work as KI input , low power
MOVAM    PD_CTL
NOP
LOOP     .....        ;; main loop,PD[4:3] are input floating &
        .....        ;; input buffer off , No power consumption.
FRAME:   BTSS          STROBE,b7    ;; wait for valid key frame
        LGOTO          FRAME
        MOVLA          B'00000000'  ;; main loop,PD[4:3] are input floating &
        MOVAM          STROBE      ;; input buffer off , No power consumption.
        .....
        .....
        LGOTO          LOOP

```

## 4.4 Interrupt & halt release

The MK9A160P provides 11 interrupt event.

IRQM, IRQM\_CTL, CPU\_RESUME, CPU\_RESUME2, IRQF and IRQF2 registers are used to control or declare request state of all interrupts. The external interrupt is triggered by a high to low transition signal of {PA0~7, PC0~7} and the related interrupt request flag (PACF; bit7 of IRQF) will be set. IRQM is used to enable/disable interrupt and IRQF is used to indicate which interrupt is occurred. If the specific IRQM doesn't enable then the hardware interrupt would not occurred. But the IRQF will response the status no matter how IRQM enable or not. For example, user enable TM1 to start counting. If IRQM bit 1 is enabled, the hardware interrupt would generate when timer overflow and IRQF bit 1 will be set. At the same time, program will jump to interrupt vector. User should clear IRQF in interrupt service routine, otherwise the interrupt would not work properly. Another condition is if IRQM bit 1 is disabled, the interrupt would not generate when timer overflow, but IRQF bit 1 still will be set. Program would not jump to interrupt vector.

### IRQM\_CTL (\$2Fh)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IRQM_CTL	INTM	--	--	--	SPIM	TM5M/ PWM5M/ CAPT5M/ RFC5M	TM4M/ PWM4M/ CAPT4M/ RFC4M	CMPM

- Bit7 (INTM): Global enable/disable bit.  
0: Disable. All interrupts are mask.  
1: Enable. All interrupt are unmask

**<Note> When interrupt is serving, the INTM will reset to "0" to prevent the other interrupt happen. After served, the RETI instruction will set INTM as '1'.**

### IRQM (\$31h)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IRQM	--	PACM	PINTM	2HZM	PHM	TM3M/ PWM3F/ CAPT3M/ RFC3M	TM2M/ PWM2F/ CAPT2M/ RFC2M	TM0M/ TONEM

- Bit7 (INTM): Global enable/disable bit.  
0: Disable. All interrupts are mask.  
1: Enable. All interrupt are unmask

**CPU\_RESUME (\$30h)**

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CPU_RESUME	--	PACR	PINTR	2HZR	PHR	TM3R/ PWM3R/ CAPT3R/ RFC3R	TM2R/ PWM2R/ CAPT2R/ RFC2R	TM0R/ TONER

- Bit7~0 : halt release mode control 1  
0: Disable.  
1: Enable.

CPU_RESUME	IRQM	Interrupt	Interrupt flag	Normal mode	HALT mode	SLEEP mode
1	X	Disable	V	Next command	1.wake-up system clock 2.Next instruction	Can't use
0	1	V	V	Jump 004h	1.wake-up system clock 2.LCALL 004h (Enter IRQ)	1.wake-up system clock 2.LCALL 004h (Enter IRQ)
0	0	X	V	X	No IRQ & No wake-up function	No IRQ & No wake-up function

**CPU\_RESUME2 (\$2Ah)**

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CPU_RESUME2	--	--	--	--	SPIR	TM5R/ PWM5R/ CAPT5R/ RFC5R	TM4R/ PWM4R/ CAPT4R/ RFC4R	CMPR

- Bit7~0 : halt release mode control 1  
0: Disable.  
1: Enable.

**IRQF (\$32h)**

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IRQF	--	PACF	INTF	2HZF	PHR	TM3F/ PWM3F/ CAPTF/ RFC3F	TM2F/ PWM2F/ CAPT2F/ RFC2F	TM0F/ TONEF

- Bit6 (PACF): PA0~7 and PC7~0 Interrupt request flag:  
0: PA and PC interrupt request off  
1: PA and PC interrupt request on
- Bit5 (INTF): INT pin interrupt request flag:  
0: INT pin (PD5) interrupt request off  
1: INT pin (PD5) interrupt request on
- Bit4 (2HZF): 2HZ Interrupt request flag  
0: 2HZF overflow interrupt request off  
1: 2HZF overflow interrupt request on
- Bit3 (PHF): TM4 Interrupt request flag  
0: PH overflow interrupt request off  
1: PH overflow interrupt request on
- Bit2 (TM3F/CAPTF/PWMF/RFCF): TM3 interrupt request flag  
0: TM2 overflow interrupt request off  
1: TM2 overflow interrupt request on (Timer overflow, Capture , PWM & RFC overflow IRQ )
- Bit1 (TM2F/CAPTF/ PWMF/RFCF): TM2 interrupt request flag  
0: TM2 overflow interrupt request off  
1: TM2 overflow interrupt request on (Timer overflow, Capture , PWM & RFC overflow IRQ )
- Bit0 (TM0F/TONEF): TM0/Capture Interrupt flag  
0: TM0 overflow or Tone interrupt request off  
1: TM0 overflow or Tone interrupt request on

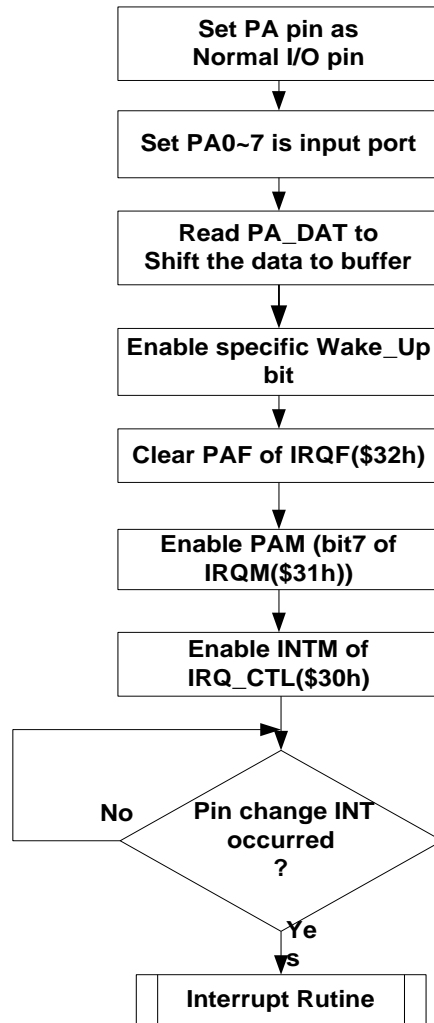
IRQF2 (\$2Bh)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IRQF2	--	--	--	--	SPIF	TM5F/ PWM5F/ CAPT5F/ RFC5F	TM4F/ PWM4F/ CAPT4F/ RFC4F	CMPF

- Bit3 (SPIF): SPI Interrupt request flag  
0: SPI interrupt request off  
1: SPI interrupt request on
- Bit2 (TM5F/CAPT5F/PWM5F/RFC5F): TM5 interrupt request flag  
0: TM5 overflow interrupt request off  
1: TM5 overflow interrupt request on (Timer overflow, Capture , PWM & RFC overflow IRQ )
- Bit1 (TM4F/CAPT4F/ PWM4F/RFC4F): TM4 interrupt request flag  
0: TM2 overflow interrupt request off  
1: TM2 overflow interrupt request on (Timer overflow, Capture , PWM & RFC overflow IRQ )
- Bit0 (CMPF): Comparator Interrupt flag  
0: CMPF interrupt request off  
1: CMPF interrupt request on

#### 4.5 External Interrupt Pin -- PA[0~7], PC[0~7] & PD[5]

Port A (PA) and port C (PC) provide external interrupt and wake up function. When device is not in sleep mode, the PA input single will serve as external interrupt. When external interrupt is occurred, program will jump to 004H (Interrupt vector). If device is in sleep mode, the PA input single will serve as wake up function. When wake up single input, device will let system clock work at first. Then wait for wake up timer (set by WDT\_CTL register \$3Bh) overflow. After that, program will jump to 004H. The below flow chart describe how to set port A to work as external interrupt or wake up function.





## 4.6 Resister to Frequency Converter (RFC)

RFC is a kind of single slope integral circuit which can be used like low speed of 16 bit ADC to detect some resistor type sensor. MK9A160P has two sets of RFC channels which have many different connections by setting register. In

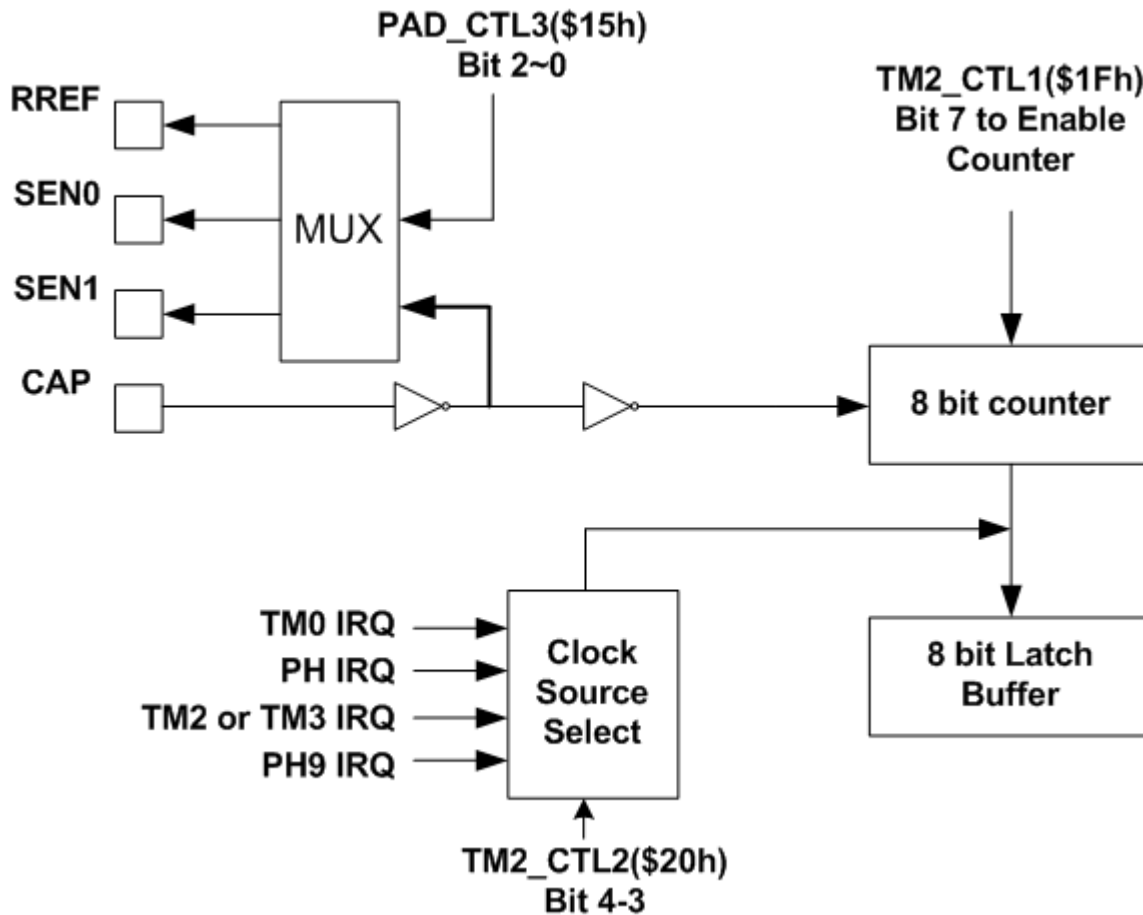


Fig.4.6.1 RFC application Block Diagram

One set of RFC block was composed of four external pins which are:

- 甲、 RREF: Reference resistor output pin
- 乙、 SEN0: Sensor 1 output pin
- 丙、 SEN1: Sensor 2 output pin
- 丁、 CAP: Oscillation input pin

Because all the RFC pins are shared with COM7~10 and (PB[2:4],PC[2]), user should set the specific register at first as below:

- (a) Set bit2~0 of LBASDT(\$33h) as (1,0,0). It means to set as 1/6 Duty, then COM7~10 will be I/O ports.
- (b) Set all bit3~0 of PAD\_CTL1(\$13h) as "0" which means to set all the ports as RFC function.

### 4.6.1 Timer 2 act as 8 bit Capture

This is one of the RFC 16 bit timer function. About setting flow, please refer to Fig.3.5.3. The source of 16 bit counter can be FCLK, PH0X2,PH4 or PH\_CLK by setting RFC\_CTL2 (\$28h) bit2~1. CAPn input clock will be the event to latch the counter to latch buffer.

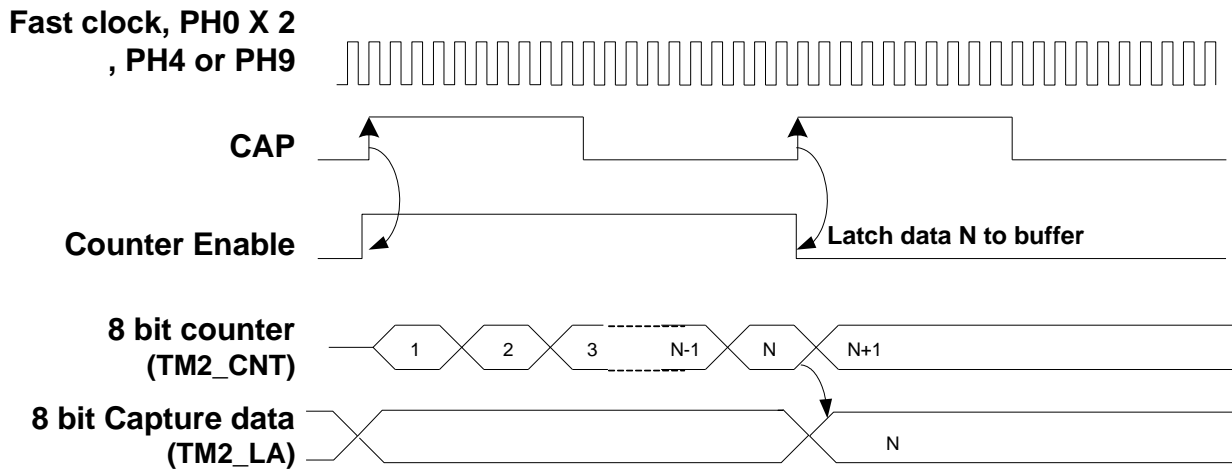


Fig.4.5.1 Timing Chart of RFC timer act as 8 bit Capture

### 4.6.2 Timer 2 act as RFC counter

When RFC timer act as 16 bit event counter to count the frequency of external RC oscillation, TM1 will be the time base and trigger signal to latch the data to buffer when timer overflow. The timing chart and setting flow are as below:

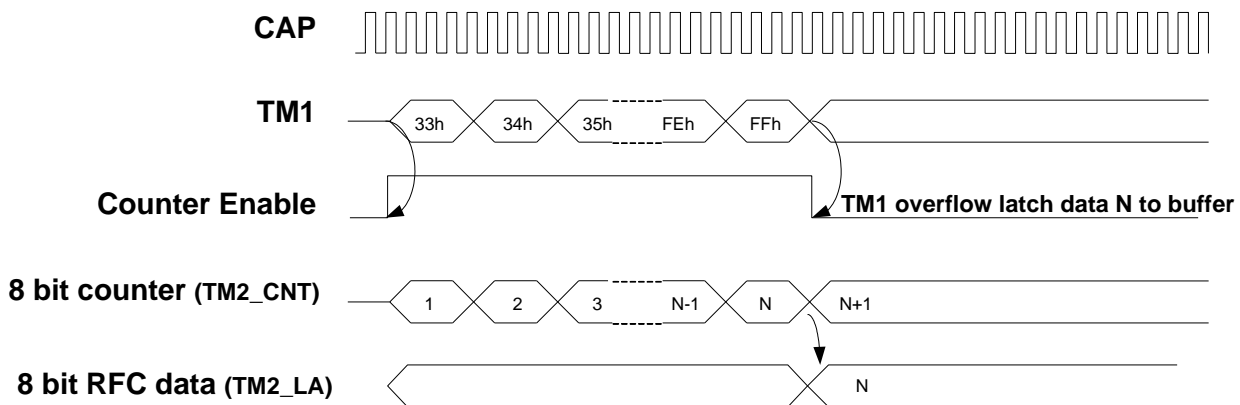


Fig.4.5.2 Timing Chart RFC timer act as RFC counter

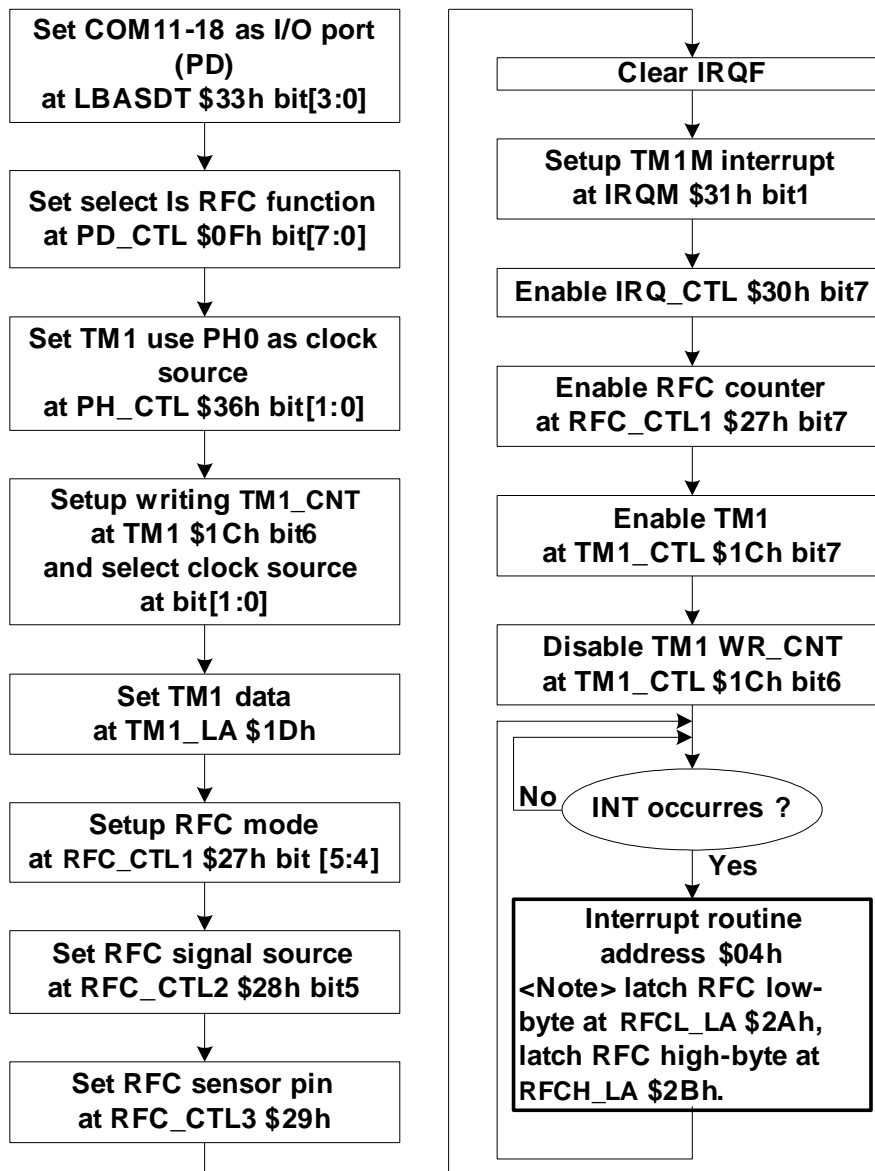
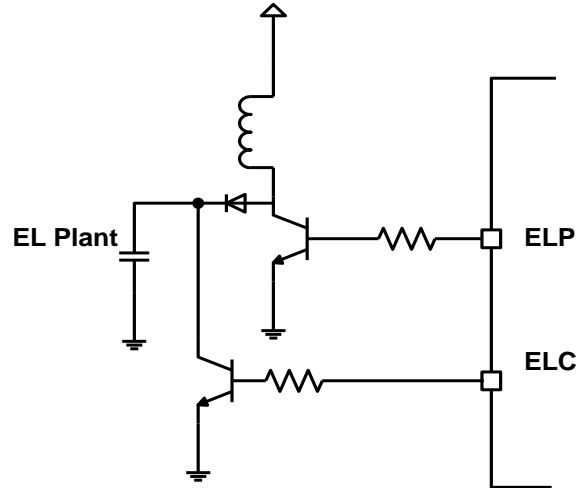


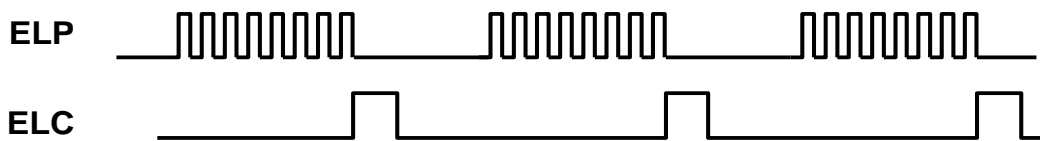
Fig.4.5.3 Setting Flow of RFC timer act as RFC counter

## 4.7 EL Panel Driver

MK9A160P provide EL panel driver for LCD backlight. The application circuit is as Fig.4.6.1. Because the ELP/ELC pin are shared with SEG39~40, user should select the pin definition by setting PAD\_CTL1 (\$13h) at first.



**Fig.4.7.1 Application circuit of EL plant connection**



**Fig.4.7.2 Timing Chart of EL Driver**

PH\_CTL (\$36h) bit6~7 are used to control EL driver. Bit6 is used to set ELP/ELC pump frequency and duty as below table. Bit7 is used to turn on/off EL driver circuit.

Bit	Symbol	Description	
6	EL_SEL	1: Fast	ELP,ELC : High speed mode If Fast clock=500Khz: ELP: 15.6KHz , 15/16 duty ELC: 488Hz, 1/4 duty (L:H=3:1)
		0: Slow	ELP,ELC : Low speed mode if Slow clock=32Khz: ELP: 16KHz , 3/4 duty ELC: 512Hz, 1/4 duty (L:H=3:1)

#### 4.8 Low Voltage Reset (LVR)

LVR function is used to prevent system from malfunction when the power drop to cause all the logics in unknown status. User can set different voltage or don't use this function by setting configuration register. The voltage in the table would have a little tolerance in different lot of chip and environment.

Bit7	Bit6	Detect voltage
LV1	LV0	
0	0	Don't use
0	1	Don't use
1	0	2.1V
1	1	Don't use

#### 4.9 Low Voltage Detect (LVD)

LVD function is used to detect battery low and prompt for user to change battery. User can set the detected voltage and status by SYS\_CTL (\$3Eh) bit4~3. This voltage in the table would have a little tolerance in different lot of chip and environment.

SYS\_CTL (\$3Eh)

Bit	Symbol	Description	
4~3	LVD1~0	Low voltage detector	
		1 1	ON (2.56V)
		1 0	ON (2.42V)
		0 1	ON (2.68V)
		0 0	Function OFF
2	LV	Low voltage detect output (read only)	
		1	VDD < 2.56V (or 2.42V,2.68V)
		0	VDD > 2.56V (or 2.42V,2.68V)

## 4.10 SPI

- SPI serial communication
- Master or slave operation
- Supports half duplex single data line mode for optical sensors
- SPI clock will be PH1 ,PH2 , FCLK/2 or FCLK/4 by option
- 8 bits transmit buffer, 8 bits receive buffer and 8 bits shift register
- Shift out complete interrupt happen

**PAD\_CTL4 (\$16h): (R/W) (default =0000000b )**

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PAD_CTL4	SEG56/ PE[7]	SEG55/ PE[6]	SEG54/ PE[5]	SEG53/ PE[4]	SEG52/ PE[3]	SEG51/ PE[2]	SEG50/ PE[1]	SG49/ PE[0]

Bit	Bitn=1	Bitn=0
0	SEG49	PE[0] / SSB
1	SEG50	PE[1] / SCLK
2	SEG51	PE[2] / SMOSI
3	SEG52	PE[3] / SMISO
4	SEG53	PE[4] + CAPT5 / SEN21
5	SEG54	PE[5] / SEN20
6	SEG55	PE[6] / REF2
7	SEG56	PE[7] +CAPT4 / CAP2

**PAD\_CTL5 (\$29h): (R/W) (default =0000000b )**

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PAD_CTL5	CAP2/ PE[7]	REF2/ PE[6]	SEN20/ PE[5]	SEN21/ PE[4]	SMISO/ PE[3]	SMOSI/ PE[2]	SCLK/ PE[1]	SSB/ PE[0]

<Note> This register is used to set the shared pin function which can be set as below table.

Bit	Bitn=1	Bitn=0
0	SSB	PE[0]
1	SCLK	PE[1]
2	SMOSI	PE[2]
3	SMISO	PE[3]
4	SEN21	PE[4] + CAPT5
5	SEN20	PE[5]
6	REF2	PE[6]
7	CAP2	PE[7] + CAPT4

**SPI\_CTL (\$06h): (R/W) (default =0000000b )**

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SPI_CTL	EN	CPOL	CPHA	SWAP	LSBF	MODE	CLK1	CLK0

Bit	Symbol	Description	
7	EN	SPI start	
		1	Start enable
		0	disable
6	CPOL	SPI clock polarity	
5	CPHA	SPI clock phase	
4	SWAP	Data swap	
		0	Swap function disabled
		1	SPI swap its use of SMOSI and SMISO. This is useful in Single wire SPI communications
3	LSBF	LSB first	
		0	The SPI transmits and receives the MSB first
		1	The SPI transmits and receives the LSB first
2	MODE	Master mode	
		0	SPI slave mode
		1	SPI master mode
1~0	CLK1~0	SPI clock output select	
		0 0	PH1
		0 1	PH5
		1 0	FCLK /2
		1 1	FREQ/2 (TM0)

PAD_CTL4 Bit0	PAD_CTL5 Bit0	SG35/ PE[0]/SSB	SPI_CTL MODE	Master/Slave
0	1	0	X	Slave
		1	X	Master
0	0	X	0	Slave
		X	1	Master
1	X	X	0	Slave
		X	1	Master

**SPI\_TX (\$29h): (R/W) (default =0000000b )**

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SPI_TX	D7	D6	D5	D4	D3	D2	D1	D0

SPI\_CTL : Write only

**SPI\_RX (\$2Ch): (R/W) (default =0000000b )**

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PAD_RX	D7	D6	D5	D4	D3	D2	D1	D0

SPI\_CTL : Read only

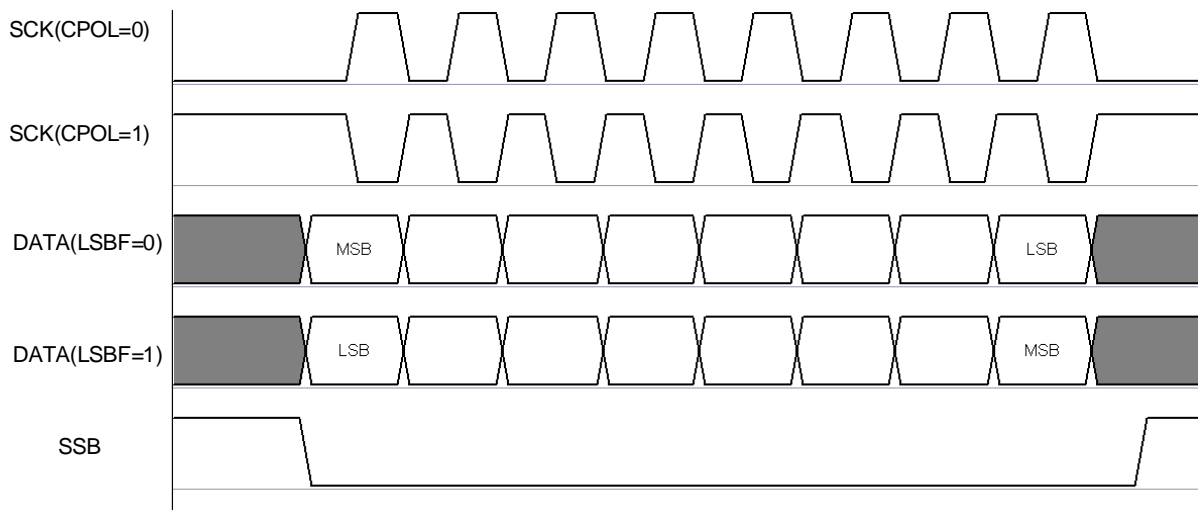


Figure: When SPI CPHA=0

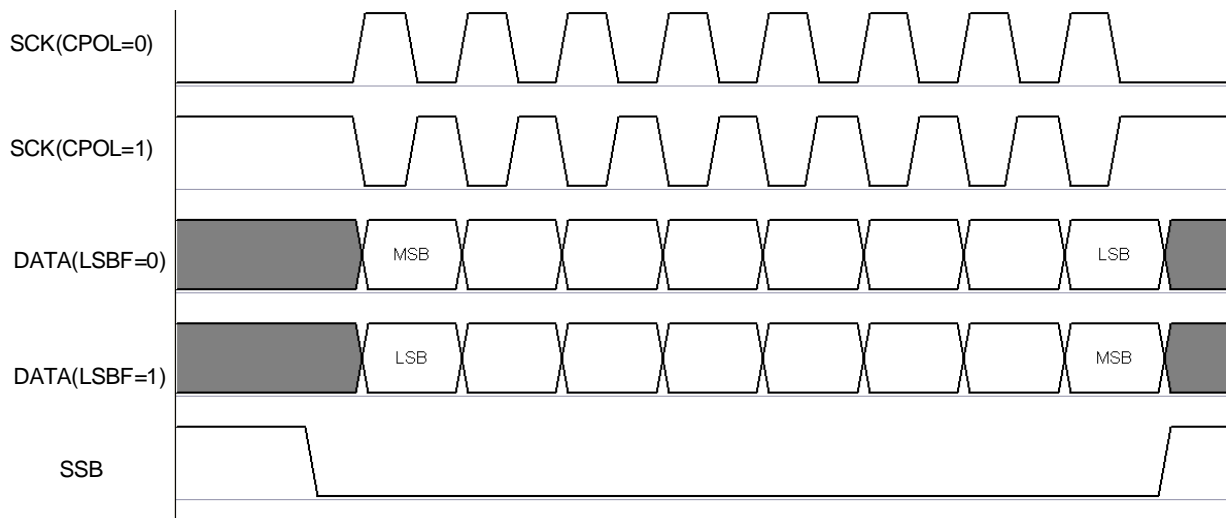
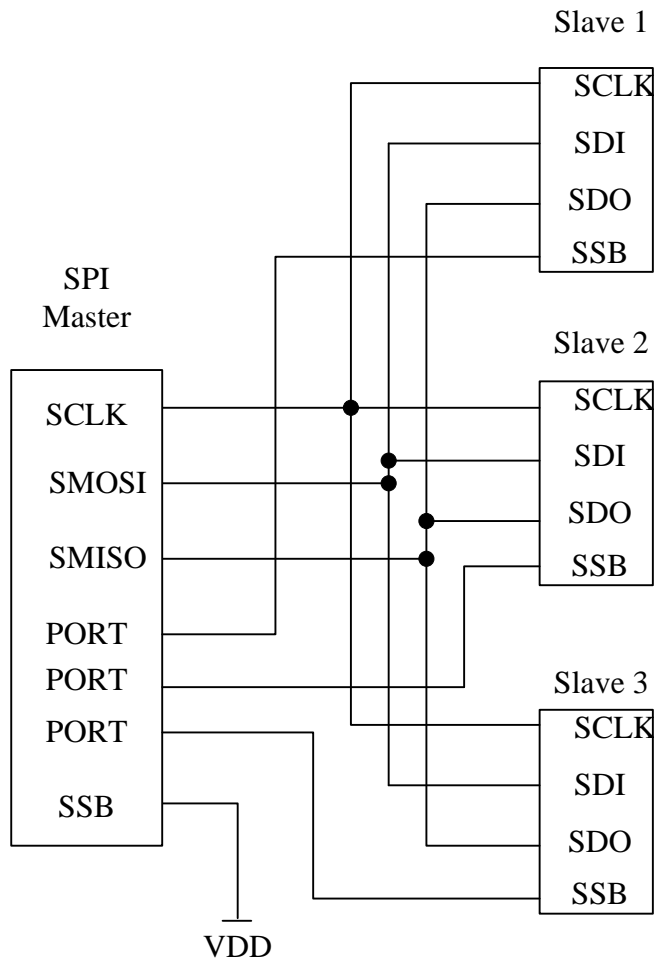


Figure: When SPI CPHA=1



4.10.1 When SPI is Master:



## 4.11 Other Register

PH\_CTL (\$36h):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PH_CTL	ELON	EL_SEL	EL_P	CLR	PH_I1	PH_I0	PH_S3	PH_S2

- Bit7: This bit is to turn on/off EL light charge pump wave form  
0: ELP/ELC ON  
1: ELP/ELC OFF, the output signal will be low.
- Bit6: EL light charge pump wave form control

Bit	Symbol	Description	
6	EL_SEL	1: Fast	ELP,ELC : High speed mode If Fast clock=500Khz: ELP: 15.6KHz , 15/16 duty ELC: 488Hz, 1/4 duty (L:H=3:1)
		0: Slow	ELP,ELC : Low speed mode If Slow clock=32Khz: ELP: 16KHz , 3/4 duty ELC: 512Hz, 1/4 duty (L:H=3:1)

- Bit5: EL pad control

Bit \ value	10	X1	00
(PH_CTL.EL_P),B6	ELP	SEG49	PD6
(PH_CTL.EL_P),B7	ELC	SEG50	PD7

1 : Clear PH11~PH15 and auto clear BIT5 (CLR) to "0".

- Bit4: Clear divider PH11~PH15  
0 : No clear  
1 : Clear PH11~PH15 and auto clear BIT5 (CLR) to "0".
- Bit3~2: PH IRQ (PHF) source select

Bit	Symbol	Description	
3~2	PH_I1~0	PH_I1~0	PH IRQ source select
		0 0	PHR <= PH10 (32hz)
		0 1	PHR <= PH11 (16hz)
		1 0	PHR <= PH12 (8hz)
		1 1	PHR <= PH13 (4hz)

- Bit1~0: (PH\_CLK) PH source select

Bit	Symbol	Description	
1~0	PH_S3~2	PH_S3~2	PH23_CLK , <b>TM2~3 use</b>
		0 0	PH23_CLK <= PH9 (64hz)
		0 1	PH23_CLK <= PH7 (256hz)
		1 0	PH23_CLK <= PH8 (128hz)
		1 1	PH23_CLK <= PH10 (32hz)

PH2\_CTL (\$27h):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PH2_CTL	--	--	--	--	--	--	PH_S5	PH_S4

- Bit1~0: (PH\_CLK) PH source select

Bit	Symbol	Description	
1~0	PH_S5~4	PH_S1~0	PH45_CLK . TM4~5 use
		0 0	PH45_CLK <= PH9 (64hz)
		0 1	PH45_CLK <= PH7 (256hz)
		1 0	PH45_CLK <= PH8 (128hz)
		1 1	PH45_CLK <= PH10 (32hz)

## 4.11.1 PH IRQ &amp; halt mode example.

```

#include "MK9A160P.INC" ; halt mode ,PH=PH10 wake-up
ORG 0x00
    LGOTO INITIAL

    ORG 004
    MOVLA '01110111'
    MOVAM IRQF ;Clear PH interrupt flag
    .....
    IRETI

INITIAL
    ORG 0x20

    CLR PA_DAT ; Clear floating
    CLR PD_DAT ; Clear floating
    MOVLA 0x00 ; Set PA is output pin
    MOVAM PA_DIR
    MOVLA 0xFF
    MOVAM PA_PUD1 ; Set Pa is normal output pin
    MOVAM PA_PUD2
    MOVAM PD_PUD1 ; Set PD is normal output pin
    MOVAM PD_PUD2
    MOVLA B'01000111' ;frame=42hz, com1~10
    MOVAM LBASDT
    MOVLA B'00110010' ;b5.4=11,low power ,lcd ON
    MOVAM LCD_CTL
    CLR IRQF ; Clear interrupt flag
    MOVLA B'00001000' ; Setup 2HZM interrupt
    MOVAM IRQM
    BS IRQM,7 ; Enable interrupt

LOOP
    BC LCD_CTL,1 ; turn off LCD
    BS SYS_CTL,6 ; Setup HALT ; OSC is active but cpu be turned off.
    ; If wake-up ,only the system clock would be turn on
    BS LCD_CTL,1 ; turn on LCD
    .....
    LGOTO LOOP

END

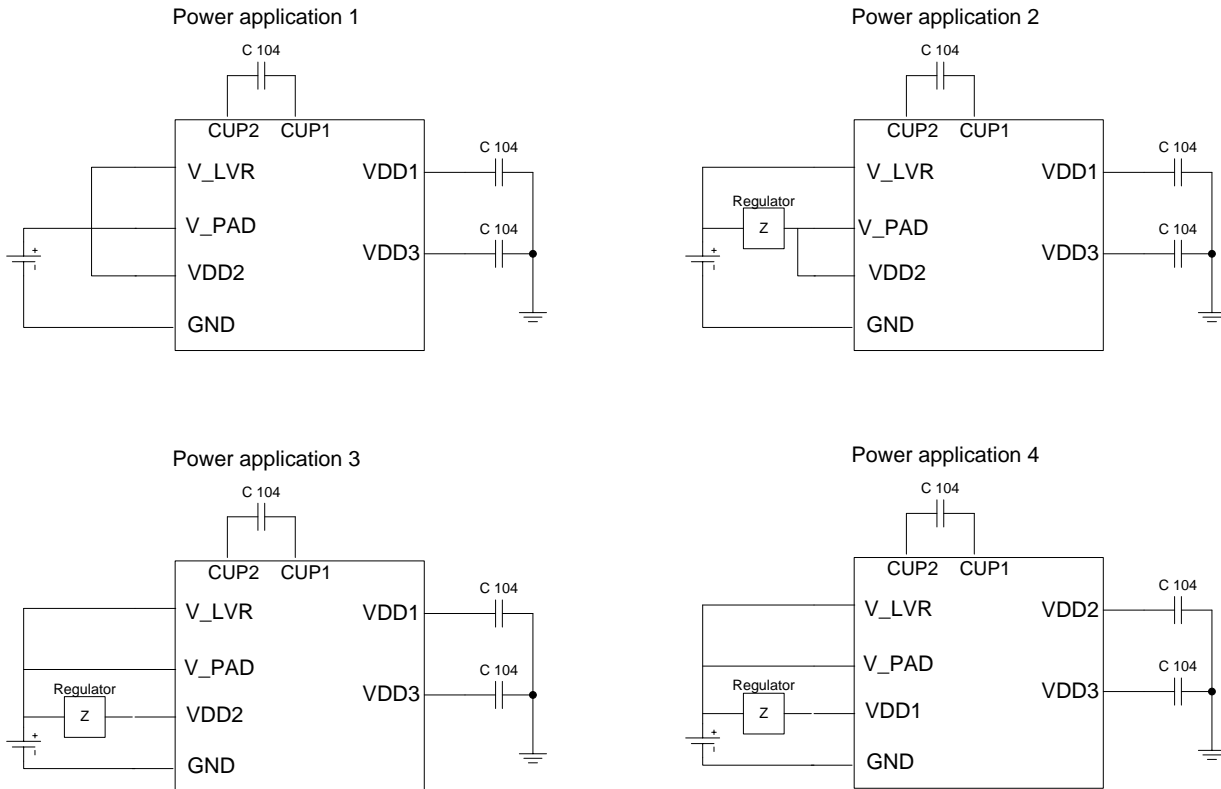
```

## 5. LCD Driver

MK9A160P has maximum 900 segment (10com \* 42seg). There are 1/4, 1/5, 1/6, 1/7, 1/8 and 1/10 1/18 Duty can be selected. There are two power modes, Li and EXT. Many COM pins are shared with I/O port or some special functions which let it more flexibility. If use less COM pin then the rest COM pin can be set to other function.

### 5.1 LCD Pad Connections (1/3 bias )

MK9A160P has different kinds of power mode, duty, bias composition. The pad connection of different bias is not the same which is as below diagram:



g.5.1.1 LCD Pad & power Connection

## 5.2 LCD Attribute Setting

There are many registers to set LCD driver attribute that is listed as below:

### 5.2.1 Bias Setting

Bias setting is in configuration register stage. Once it was fixed, user can not change by software. There is only two bias can be selected: 1/2 or 1/3.

### 5.2.2 Duty (COM) and Frame Frequency Setting

LBASDT (\$33h) is used to set duty and LCD frame frequency. Once the duty is set which means the selected pin will be used as LCD COM output. The rest COM pin will automatically been a I/O ports. The register definition is as below:

#### LBASDT2(\$74h): LCD DUTY SELECT 2(R/W) (Special Purpose Register 2)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LBASDT	LCD2	--	--	--	--	--	--	--

#### LBASDT(\$33h): LCD DUTY SELECT (R/W)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LBASDT	LCD1	LCD0	FRAM1	FRAM0	LDUTY3	LDUTY2	LDUTY1	LDTUY0

- Bit 7~6 : LCD0 mode select

LCD2.LCD1.LCD0

0 1 1 : LED mode 1 – COMn is high active , SEGn is low active (output voltage – 0 & VDD)

0 1 0 : LED mode 2 – COMn is low active , SEGn is low active (output voltage – 0 & VDD )

0 0 1 : LCD mode 2– **1/2 bias charge pump** . SEGn is low active

(output voltage – 0, VDD/2 , VDD3=VDD4=VDD)

0 0 0 : LCD mode 1– **1/3 bias charge pump** .

(output voltage – 0, VDD/2 , VDD3=VDD4=3\*VDD/2)

**1 0 0 : LCD mode 3– 1/4 bias charge pump** .

(output voltage – 0, VDD/2 , VDD3=3\*VDD/2,VDD4=2\*VDD)

DUTY	LCD ON		LCD OFF	
	COMn	SEGn	COMn	SEGn
LCD1	4.5V / 0V	0V / 4.5V	0	0
LCD2	3.0V / 0V	0V / 3.0V	0	0
LCD3	6.0V / 0V	0V / 6.0V	0	0
LED1	3V	0	0	0
LED2	0	3V	0	0

	LCD OFF	ALL ON	LCD ON	ALL OFF
LCD1	V	V	V	V
LCD2	V	V	V	V

LED1	V	X	V	X
LED2	V	X	V	X

- Bit5~4: LCD frame control (Write only)
  - 0 0 : FRAME 1
  - 0 1 : FRAME 2
  - 1 0 : FRAME 3
  - 1 1 : FRAME 4

DUTY	LCD FRAME (Hz)			
	FRAME 1	FRAME 2	FRAME 3	FRAME 4
1/2	65.36	43.57	<b>87.15</b>	130.72
1/3	64.05	42.67	85.33	128.10
1/4	64.05	42.67	85.33	128.10
1/5	64.68	43.12	86.23	129.36
1/6	64.05	42.67	85.33	128.10
1/7	62.04	41.36	82.72	124.08
1/8	65.02	43.35	86.70	130.04
1/10	65.02	43.35	86.70	130.4
1/12	64.00	42.67	85.34	128.01
1/16	85.29	56.86	113.72	170.58

DUTY	LED FRAME (Hz)			
	FRAME 1	FRAME 2	FRAME 3	FRAME 4
1/2	130.72	87.14	174.30	261.44
1/3	128.10	85.33	170.66	256.20
1/4	128.10	85.33	170.66	256.20
1/5	129.36	86.24	172.46	258.72
1/6	128.10	85.33	170.66	256.20
1/7	124.08	82.72	165.44	248.16
1/8	130.04	86.70	173.40	260.10
1/10	130.04	86.70	173.40	260.10
1/12	128.01	85.34	170.68	256.02
1/16	170.58	113.72	227.44	341.16



<Note> This frequency table base on 32KHz source clock. If user choose fast only clock mode and the clock is not 32KHz, the frequency would not be as above.

- Bit3~0: LCD driver duty setting (R/W)

LDUTY3~0	DUTY	COM 1~6			
		COM is for LCD	COM to be I/O	Port C	Port F
0 0 0 0	1/2	1~4	5~16	PC[0,1,2,5]	PF[0~7]
0 0 0 1	1/3	1~4	5~16	PC[0,1,2,5]	PF[0~7]
0 0 1 0	1/4	1~4	5~16	PC[0,1,2,5]	PF[0~7]
0 0 1 1	1/5	1~5	6~16	PC[0,1,5]	PF[0~7]
0 1 0 0	1/6	1~6	7~16	PC[0,,5]	PF[0~7]
0 1 0 1	1/7	1~7	8~16	PC[5]	PF[0~7]
0 1 1 0	1/8	1~8	9~16	XX	PF[0~7]
0 1 1 1	1/10	1~10	11~16	XX	PF[2~7]
1 0 0 0	1/12	1~12	13~16	XX	PF[4~7]
1 0 0 1	1/16	1~16	X	XX	XX
Others	Reserved				

### 5.2.3 LCD Pump Frequency and ON/OFF Control

#### LCD\_CTL(\$35h): LCD Control (R/W) (default = 0000xx00b)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LCD_CTL	PUMP1	PUMP0	POW1	POW0	OVP1	OVP0	LCDM1	LCDM0

- Bit7~6 (PUMP1~0): LCD charge pump clock select.
  - 00: Original pump clock
  - 01: Original pump clock x 2
  - 10: don't use
  - 11: Original pump clock x 4
- Bit5~4 (POW1~0): Low speed power saving control (halt mode)
  - 00: default
  - 11: low power ( suggest ,low power for halt mode )
- Bit3~2 (OVP1~0): LCD waveform control
  - 00: overlap
  - 01:  $T_{NON-OVERLAP}$  ( 15uS)
  - 10:  $T_{NON-OVERLAP} \times 2$  ( 30uS)
  - 11:  $T_{NON-OVERLAP} \times 4$  ( 60uS)
- Bit1~0 (LCDM1~0): LCD/LED mode select.(default : ALL OFF Mode )

Bit1	Bit0	Mode	Status
0	0	Normal Mode	LCD power OFF (LED OFF)
0	1	ALL ON Mode	When ALL ON is set to "0", all segment drivers are turned on waveform
1	0	Normal Mode	1.The display data in the register is output to the segment drivers LCD ON 2. LED ON
1	1	ALL OFF Mode	When ALL OFF is set to "1", all segment drivers are turned off waveform

### 5.3 LCD Display RAM Mapping

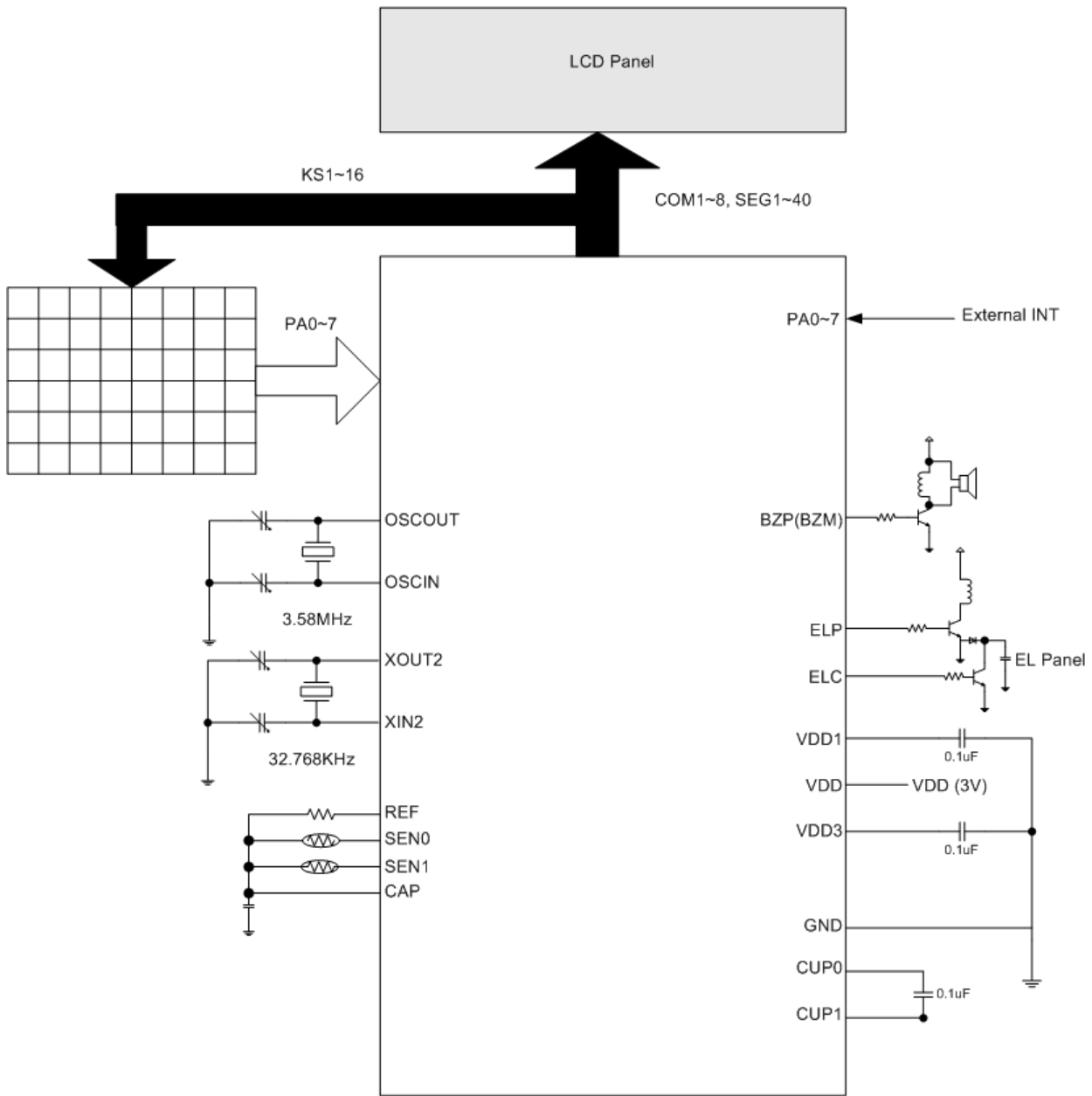
The LCD driver has a LCD RAM. Each bit RAM map to specific COM/Segment as below table. If LCD RAM doesn't use, they can be used as working RAM to store data.

\$68~69 work as working RAM only .

	Bit0~Bit7	Bit0~Bit7		Bit0~Bit7	Bit0~Bit7
	C1~C8	C9~C16		C1~C8	C9~C16
<b>SEG1</b>	B0/\$40	B1/\$40	<b>SEG33</b>	B0/\$60	B1/\$60
<b>SEG2</b>	B0/\$41	B1/\$41	<b>SEG34</b>	B0/\$61	B1/\$61
<b>SEG3</b>	B0/\$42	B1/\$42	<b>SEG35</b>	B0/\$62	B1/\$62
<b>SEG4</b>	B0/\$43	B1/\$43	<b>SEG36</b>	B0/\$63	B1/\$63
<b>SEG5</b>	B0/\$44	B1/\$44	<b>SEG37</b>	B0/\$64	B1/\$64
<b>SEG6</b>	B0/\$45	B1/\$45	<b>SEG38</b>	B0/\$65	B1/\$65
<b>SEG7</b>	B0/\$46	B1/\$46	<b>SEG39</b>	B0/\$66	B1/\$66
<b>SEG8</b>	B0/\$47	B1/\$47	<b>SEG40</b>	B0/\$67	B1/\$67
<b>SEG9</b>	B0/\$48	B1/\$48	<b>SEG41</b>	B0/\$68	B1/\$68
<b>SEG10</b>	B0/\$49	B1/\$49	<b>SEG42</b>	B0/\$69	B1/\$69
<b>SEG11</b>	B0/\$4A	B1/\$4A	<b>SEG43</b>	B0/\$6A	B1/\$6A
<b>SEG12</b>	B0/\$4B	B1/\$4B	<b>SEG44</b>	B0/\$6B	B1/\$6B
<b>SEG13</b>	B0/\$4C	B1/\$4C	<b>SEG45</b>	B0/\$6C	B1/\$6C
<b>SEG14</b>	B0/\$4D	B1/\$4D	<b>SEG46</b>	B0/\$6D	B1/\$6D
<b>SEG15</b>	B0/\$4E	B1/\$4E	<b>SEG47</b>	B0/\$6E	B1/\$6E
<b>SEG16</b>	B0/\$4F	B1/\$4F	<b>SEG48</b>	B0/\$6F	B1/\$6F
<b>SEG17</b>	B0/\$50	B1/\$50	<b>SEG49</b>	B0/\$70	B1/\$70
<b>SEG18</b>	B0/\$51	B1/\$51	<b>SEG50</b>	B0/\$71	B1/\$71
<b>SEG19</b>	B0/\$52	B1/\$52	<b>SEG51</b>	B0/\$72	B1/\$72
<b>SEG20</b>	B0/\$53	B1/\$53	<b>SEG52</b>	B0/\$73	B1/\$73
<b>SEG21</b>	B0/\$54	B1/\$54	<b>SEG53</b>	B0/\$74	B1/\$74
<b>SEG22</b>	B0/\$55	B1/\$55	<b>SEG54</b>	B0/\$75	B1/\$75
<b>SEG23</b>	B0/\$56	B1/\$56	<b>SEG55</b>	B0/\$76	B1/\$76
<b>SEG24</b>	B0/\$57	B1/\$57	<b>SEG56</b>	B0/\$77	B1/\$77
<b>SEG25</b>	B0/\$58	B1/\$58	<b>SEG57</b>	B0/\$78	B1/\$78
<b>SEG26</b>	B0/\$59	B1/\$59	<b>SEG58</b>	B0/\$79	B1/\$79
<b>SEG27</b>	B0/\$5A	B1/\$5A	<b>SEG59</b>	B0/\$7A	B1/\$7A
<b>SEG28</b>	B0/\$5B	B1/\$5B	<b>SEG60</b>	B0/\$7B	B1/\$7B
<b>SEG29</b>	B0/\$5C	B1/\$5C	<b>SEG61</b>	B0/\$7C	B1/\$7C

<b>SEG30</b>	B0/\$5D	B1/\$5D	<b>SEG62</b>	B0/\$7D	B1/\$7D
<b>SEG31</b>	B0/\$5E	B1/\$5E	<b>SEG63</b>	B0/\$7E	B1/\$7E
<b>SEG32</b>	B0/\$5F	B1/\$5F	<b>SEG64</b>	B0/\$7F	B1/\$7F

## 6. Typical Application Circuit



Application circuit

## 7. Instruction Table

JUMP INSTRUCTION				
LCALL I	Call subroutine. However, LCALL can addressing 16K address	2	None	01ii iiiiii iiiiii
LGOTO I	Go branch to any address	2	None	00ii iiiiii iiiiii
LOGIC				
AND M, a	$(M) \cdot (\text{acc}) \rightarrow (\text{acc})$	1	Z	1010 1000 MMMM MMMM
AND M, m	$(M) \cdot (\text{acc}) \rightarrow (M)$	1	Z	1010 1001 MMMM MMMM
ANDLA I	Immediate $\cdot (\text{acc}) \rightarrow (\text{acc})$	1	Z	1111 1000 iiiiii iiiiii
COM M, a	$\sim(M) \rightarrow (\text{acc})$	1	Z	1010 0100 MMMM MMMM
COM M, m	$\sim(M) \rightarrow (M)$	1	Z	1010 0101 MMMM MMMM
IOR M, a	$(M) \text{ or } (\text{acc}) \rightarrow (\text{acc})$	1	Z	1011 1110 MMMM MMMM
IOR M, m	$(M) \text{ or } (\text{acc}) \rightarrow (M)$	1	Z	1011 1111 MMMM MMMM
IORLA I	Immediate or $(\text{acc}) \rightarrow (\text{acc})$	1	Z	1111 0010 iiiiii iiiiii
RL M, a	Rotate left from m to acc $m[6:0] \rightarrow \text{acc}[7:1]$ $m[7] \rightarrow \text{acc}[0]$	1	None	1110 0000 MMMM MMMM
RL M, m	Rotate left from m to itself $m[6:0] \rightarrow m[7:1]$ $m[7] \rightarrow m[0]$	1	None	1110 0001 MMMM MMMM
RLC M, a	Rotate left from m to acc $m[7] \rightarrow c$ $m[6:0] \rightarrow \text{acc}[7:1]$ $c \rightarrow \text{acc}[0]$	1	C	1110 0010 MMMM MMMM
RLC M, m	Rotate left from m to itself $m[7] \rightarrow c$ $m[6:0] \rightarrow m[7:1]$ $c \rightarrow m[0]$	1	C	1110 0011 MMMM MMMM
SL0 M, a	Shift left from m to acc $m[6:0] \rightarrow \text{acc}[7:1]$ $0 \rightarrow \text{acc}[0]$	1	None	1110 0100 MMMM MMMM
SL0 M, m	Rotate left from m to itself $m[6:0] \rightarrow m[7:1]$ $0 \rightarrow m[0]$	1	None	1110 0101 MMMM MMMM
SL1 M, a	Shift left from m to acc $m[6:0] \rightarrow \text{acc}[7:1] \ \& \ 1 \rightarrow \text{acc}[0]$	1	None	1110 0110 MMMM MMMM
SL1 M, m	Rotate left from m to itself $m[6:0] \rightarrow m[7:1] \ \& \ 1 \rightarrow m[0]$	1	None	1110 0111 MMMM MMMM

RR M, a	Rotate right from m to acc 0→acc[7] m[7:1]→acc[6:0]	1	None	1110 1000 MMMM MMMM
RR M, m	Rotate right from m to itself M[0]→m[7] m[7:1]→m[6:0]	1	None	1110 1001 MMMM MMMM
RRC M, a	Rotate right from m to acc m[0]→c , c→acc[7] m[7:1]→acc[6:0]	1	C	1110 1010 MMMM MMMM
RRC M, m	Rotate right from m to itself m[0]→c , c→m[7] m[7:1]→m[6:0]	1	C	1110 1011 MMMM MMMM
SR0 M, a	Rotate right from m to acc 0→acc[7] m[7:1]→acc[6:0]	1	None	1110 1100 MMMM MMMM
SR0 M, m	Rotate right from m to itself 0→m[7] m[7:1]→m[6:0]	1	None	1110 1101 MMMM MMMM
SR1 M, a	Rotate right from m to acc 1→acc[7] m[7:1]→acc[6:0]	1	None	1110 1110 MMMM MMMM
SR1 M, m	Rotate right from m to itself 1→m[7] m[7:1]→m[6:0]	1	None	1110 1111 MMMM MMMM
SWAP M, a	m[7:4] → acc[3:0] m[3:0] → acc[7:4]	1	None	1011 1100 MMMM MMMM
SWAP M, m	m[7:4] ↔ m[3:0]	1	None	1011 1101 MMMM MMMM
XOR M, a	(M) xor (acc) → (acc)	1	Z	1011 0110 MMMM MMMM
XOR M, m	(M) xor (acc) → (M)	1	Z	1011 0111 MMMM MMMM
XORLA I	Immediate xor (acc) → (acc)	1	Z	1111 1001 iiiii iiiii
MATHEMATICS				
ADD M, a	(M)+(acc) → (acc)	1	C, DC, Z	1010 1010 MMMM MMMM
ADD M, m	(M)+(acc) → (M)	1	C, DC, Z	1010 1011 MMMM MMMM
ADDC M,a	(M)+(acc) + (carry) → (acc)	1	C, DC, Z	1011 1010 MMMM MMMM
ADDC M,m	(M)+(acc) + (carry) → (M)	1	C, DC, Z	1011 1011 MMMM MMMM
ADDLA I	Immediate + (acc) → (acc)	1	C, DC, Z	1111 1010 MMMM MMMM
BC M, bn	Clear bit n of (M)	1	None	1001 1bbb MMMM MMMM
BS M, bn	Set bit n of (M)	1	None	1001 0bbb MMMM MMMM
CLRA	Clear accumulator	1	Z	1010 0010 0000 0000
CLR M	Clear memory M	1	Z	1010 0011 MMMM MMMM
TABRDL M	Read low byte ROM table to (acc) ROM table address={TB_BNK,index	2	None	1101 1000 MMMM MMMM

	of M }			
TABRDH M	Read high byte ROM table to (acc) ROM table address={TB_BNK,index of M }	2	None	1101 1001 MMMM MMMM
SUBC M, a	(M)+(/acc)+ (carry) → (acc)	1	C, DC, Z	1101 0100 MMMM MMMM
SUBC M, m	(M)+(/acc) + (carry)→ (M)	1	C, DC, Z	1101 0101 MMMM MMMM
DAA M, a	Decimal Adjust M to ACC If ACC[3:0] > 9 or DC=1 Then ACC[3:0]←ACC[3:0]+6, DC1=~DC else ACC[3:0] ←ACC[3:0], DC1=0 If ACC[7:4]+DC1 > 9 or C=1 Then ACC[7:4]←ACC[7:4]+6+DC1, C=1 else ACC[7:4] ←ACC[7:4]+DC1, C=C	1	C	1101 0110 MMMM MMMM
DAA M, m	Decimal Adjust M to memory If ACC[3:0] > 9 or DC=1 Then M[3:0]←ACC[3:0]+6, DC1=~DC else M[3:0] ←ACC[3:0], DC1=0 If ACC[7:4]+DC1 > 9 or C=1 Then M[7:4]←ACC[7:4]+6+DC1, C=1 else M[7:4] ←ACC[7:4]+DC1, C=C	1	C	1101 0111 MMMM MMMM
DAS M, a	Decimal Adjust M to ACC If ACC[3:0] > 9 or DC=0 Then ACC[3:0]←ACC[3:0]-6, DC1=~DC Else ACC[3:0]←ACC[3:0], DC1=1 If ACC[7:4] -DC1 > 9 or C=0 Then ACC[7:4]←ACC[7:4]-6-DC1, C=0 else ACC[7:4] ←ACC[7:4]-DC1, C=~C	1	C	1101 1110 MMMM MMMM
DAS M, m	Decimal Adjust M to memory If ACC[3:0] > 9 or DC=0 Then M[3:0]←ACC[3:0]-6, DC1=~DC else M[3:0] ←ACC[3:0], DC1=1 If ACC[7:4]-DC1 > 9 or C=0	1	C	1101 1111 MMMM MMMM



	Then M[7:4]←ACC[7:4]-6-DC1, C=1 else M[7:4]←ACC[7:4]-DC1, C=C			
DEC M, a	(M) - 1 → (acc)	1	Z	1010 1100 MMMM MMMM
DEC M, m	(M) - 1 → (M)	1	Z	1010 1101 MMMM MMMM
INC M, a	(M) + 1 → (acc)	1	Z	1011 0000 MMMM MMMM
INC M, m	(M) + 1 → (M)	1	Z	1011 0001 MMMM MMMM
MOVAM m	(acc) → (M)	1	None	1010 0001 MMMM MMMM
MOV M, a	(M) → (acc)	1	Z	1010 0110 MMMM MMMM
MOV M, m	(M) → (M)	1	Z	1010 0111 MMMM MMMM
MOV2 M, a	(M) → (acc)	1	None	1111 0110 MMMM MMMM
MOV2 M, m	(M) → (M)	1	None	1111 0111 MMMM MMMM
MOVLA I	Immediate data → acc	1	None	1111 0000 iiiii iiiii
SUBLA I	(immediate data)-(Acc)→(Acc)	1	C, DC, Z	1111 0100 iiiii iiiii
SUB M, m	(M)-(acc) → (M)	1	C, DC, Z	1011 0101 MMMM MMMM
SUB M, a	(M)-(acc) → (acc)	1	C, DC, Z	1011 0100 MMMM MMMM
OTHER OPERATION				
NOP	No operation	1	None	1111 1111 1111 1111
CLRWDI	Clear watch-dog register	1	$\overline{TO}, \overline{PD}$	1111 1111 1111 0000
RET	Return (for lcall instruction )	2	None	1111 1111 1111 0001
I RETI	Return and enable INTM( for IRQ )	2	None	1111 1111 1111 0010
I RET	Return ( for IRQ )	2	None	1111 1111 1111 0011
RETLA	Return & Immediate data → acc	2	None	1101 1100 iiiii iiiii
SLEEP	Enter sleep (saving) mode	1	$\overline{TO}, \overline{PD}$	1111 1111 1111 0100
CONDITION OPERATION				
BTSC M, bn	If (bit n of (M))=0, skip next instruction	1 or 2	None	1000 1bbb MMMM MMMM
BTSS M, bn	If (bit n of (M))=1, skip next instruction	1 or 2	None	1000 0bbb MMMM MMMM
DECSZ M, a	(M) - 1 →(acc), skip if (acc) = 0	1 or 2	None	1010 1110 MMMM MMMM
DECSZ M, m	(M) - 1 → (M), skip if (M) = 0	1 or 2	None	1010 1111 MMMM MMMM
INCSZ M, a	(M) + 1 →(acc), skip if (acc) = 0	1 or 2	None	1011 0010 MMMM MMMM
INCSZ M, m	(M) + 1 → (M), skip if (M) = 0	1 or 2	None	1011 0011 MMMM MMMM
TMSS	If (acc) =0, skip next instruction	1 or 2	None	1011 1000 XXXX XXXX
TMSC M	If (M) = 0, skip next instruction	1 or 2	None	1011 1001 MMMM MMMM
TMSNC M	If (M) =\= 0, skip next instruction	1 or 2	None	1101 1011 MMMM MMMM
TMSNS	If (acc) =\=0, skip next instruction	1 or 2	None	1101 1010 XXXX XXXX
TMCOMPE M	If (acc) =(M), skip next instruction	1 or 2	None	1010 0000 MMMM MMMM
TMCOMPEB M	If (acc) =\=(M), skip next instruction	1 or 2	None	1101 1101 MMMM MMMM

## 8. Electrical Characteristics

### 8.1 Absolute Maximum Ratings

Supply Voltage ....  $V_{SS}-0.3V$  to  $V_{SS}+5.5V$       Storage Temperature .....  $-50^{\circ}C$  to  $125^{\circ}C$

Input Voltage .....  $V_{SS}-0.3V$  to  $V_{DD}+0.3V$       Operating Temperature...  $- 0^{\circ}C$  to  $70^{\circ}C$

<Note> : These are stress ratings only. Stresses exceeding the range specified under "Absolute Maximum Ratings" may cause substantial damage to the device. Functional operation of this device at other conditions beyond those listed in the specification is not implied and prolonged exposure to extreme conditions may affect device reliability.

### 8.2 DC Characteristics

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		VDD	Conditions				
VDD	Operating Voltage	---		2.2		3.6	V
$I_{DD1}$	Operating Current (32K Crystal )	3V	sleep mode, watchdog is off			1	$\mu A$
$I_{DD2}$	Operating Current (32K Crystal )	3V	sleep mode, watchdog is on		2.5		$\mu A$
$I_{DD3}$	Operating Current (32K Crystal )	3V	Halt mode,LCD is off, watchdog is off		2.5		$\mu A$
$I_{DD4}$	Operating Current (32K Crystal )	3V	Halt mode,LCD is off, watchdog is on		2.5		$\mu A$
$I_{DD5}$	Operating Current (32K Crystal )	3V	Halt mode,LCD is on, watchdog is off		2.5		$\mu A$
$I_{DD6}$	Operating Current (32K Crystal )	3V	Halt mode,LCD is on, watchdog is on		2.5		$\mu A$
$V_{IH1}$	Port PA0~6	3V	Input Low to High Voltage		1.7		V
$V_{IL1}$	Port PA0~6	3V	Input high to Low Voltage		1.2		V
$V_{IH2}$	RESETB	3V	Input Low to High Voltage		1.5		V
$V_{IL2}$	RESETB	3V	Input high to Low Voltage		1		V

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		VDD	Conditions				
$V_{IH3}$	Port PC0~5,CAP1 and CAP2	3V	input Low to High Voltage		2		V
$V_{IL3}$	Port PC0~5,CAP1 and CAP2	3V	Input high to Low Voltage		1.1		V
$V_{IL4}$	CAPT1A/CAPT1B input Low Voltage	3V	input Low to High Voltage		1		V
$V_{IH4}$	PA0~6 input High Voltage	3.3V	Input high to Low Voltage		1.8		V
$I_{IL}$	Input Leakage Current	3V	$V_{in}=V_{DD}, V_{SS}$			1	$\mu A$
$R_{PH1}$	Pull-high Resistance	3V	Port PA0~6		95		Kohm
$R_{PL1}$	Pull-down Resistance	3V	Port PA0~6,PC0~4, PD0~7		100		Kohm
$R_{PL2}$	Pull-down Resistance	3V	RESETB		150		Kohm
$I_{LVR1}$	LVR Current	3V	Low Voltage Detector		0.3		$\mu A$
		5V	Config bit7.bit6=10		1		
$I_{LVD1}$	LVD Current	3V	Low Voltage Detector		0.3		$\mu A$
		5V	Config bit7.bit6=10		1		
$V_{LVR3}$	LVR voltage	3V	Low Voltage Detector Config bit7.bit6=10		2.1		V
$V_{LVD1}$	LVD voltage	3V	SYS_CTL bit4.3=01		2.68		V
$V_{LVD2}$	LVD voltage	3V	SYS_CTL bit4.3=10		2.42		V
$V_{LVD3}$	LVD voltage	3V	SYS_CTL bit4.3=11	2.40	2.56	2.80	V

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		VDD	Conditions				
$I_{OH}$	PA0~3 output port , PB0~7 output port,, Driving Current	5V	Voh=4.5V		9		mA
			Voh=4.0V		18		mA
			Voh=3.5V		22		mA
			Voh=3.0V		26		mA
			Voh=2.5V		29		mA
		3V	Voh=2.7V		4		mA
			Voh=2.4V		7		mA
			Voh=2.1V		9		mA
			Voh=2.1V		10		mA
			Voh=1.8V		12		mA
$I_{OL}$	PA0~3 output port , PB0~7 output port, Sink Current	5V	Voh=0.5V		20		mA
			Voh=1.0V		37		mA
			Voh=1.5V		48		mA
			Voh=2.0V		55		mA
			Voh=2.5V		59		mA
		3V	Voh=0.3V		8		mA
			Voh=0.6V		16		mA
			Voh=0.9V		21		mA
			Voh=1.2V		25		mA
			Voh=1.5V		26		mA

### 8.3 AC Characteristics

Symbol	Parameter	Test Conditions		Min	Typ	Max	Unit
		Conditions	VDD				
$f_{sys1}$	System Clock	LP Crystal mode	3.3V		32		Khz
$f_{sys2}$	System Clock	NT Crystal mode	3.3V	0.455		10	Mhz
$f_{sys3}$	System Clock	Internal slow RC	3.3V			64	Khz
$f_{sys4}$	System Clock	Internal fast RC	3.3V			6	Mhz
$T_{wdt}$	1 : 1 Watchdog Timer	SOSC1.0=00	3.3V	C <sub>ystal</sub> x 512			mS
		SOSC1.0=01		24			
		SOSC1.0=10		External RC x 512			
		SOSC1.0=11		24			
$T_{rht}$	Reset Hold Time		3.3V		24		mS

### 8.4 External RC Table

#### Low speed external RC Table

R value	C value	RC frequency	R connect to (VDD,XIN1)
196K	0.1u (suggest)	80 Khz	The capacitor is for stabile frequency
295K	0.1u (suggest)	54 Khz	
384K	0.1u (suggest)	42.56 Khz	
500K	0.1u (suggest)	33.3 Khz	
670K	0.1u (suggest)	25.3 Khz	

#### High speed external RC Table

R value	C value	RC frequency	R connect to (VDD,OSCIN)
383K	0.1u (suggest)	203 Khz	The capacitor is for stabile frequency
196K	0.1u (suggest)	418 Khz	
147K	0.1u (suggest)	570 Khz	
98K	0.1u (suggest)	0.90 Mhz	
50.4K	0.1u (suggest)	1.89 Mhz	
38.4K	0.1u (suggest)	2.58 Mhz	
29.5K	0.1u (suggest)	3.46 Mhz	
19.7K	0.1u (suggest)	5.46 Mhz	