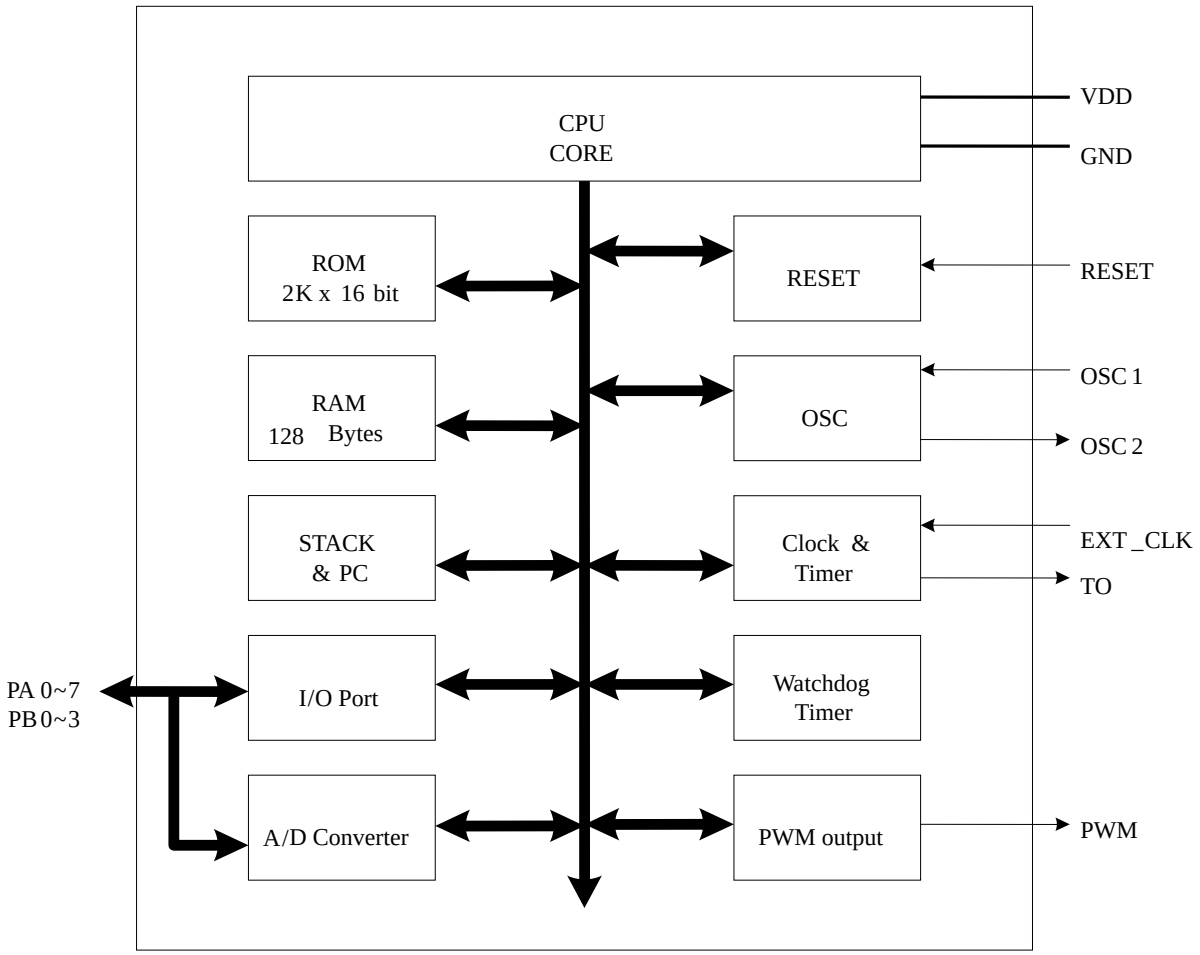
General Description

The MK7A25P is an 8 bit RISC high performance microcontroller with 10bit A/D converter. It is equipped with 2K word OTP(One Time Programmable) ROM, 128 Bytes RAM, Timer/Counter, Capture, Interrupt, LVR(Low Voltage Reset), I/O ports and PWM output in a single chip.

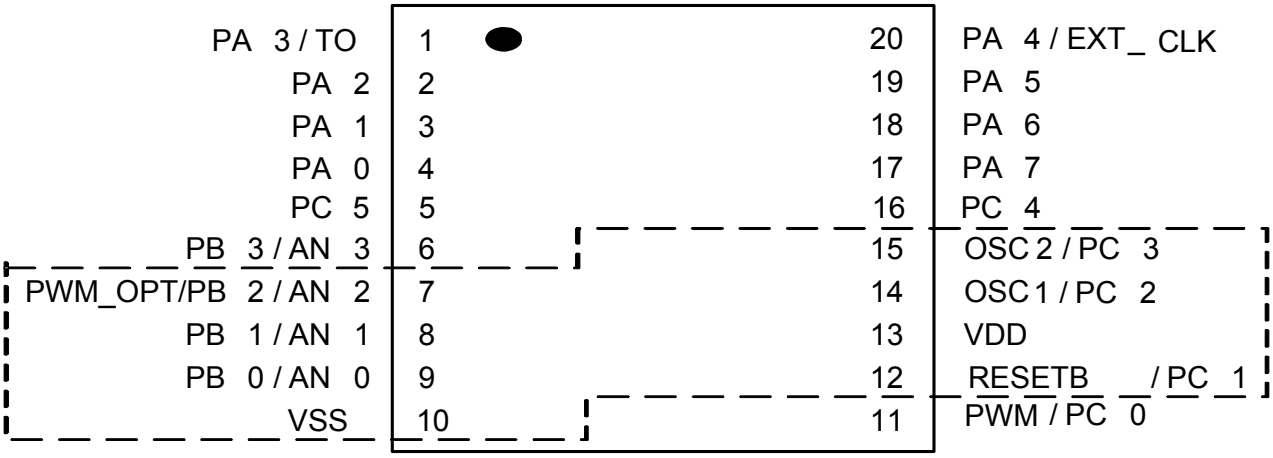
1. Feature

- ROM size: 2K x 16 bits
- RAM: 37 x 8 bits(Special Purpose Register) + 128 x 8 (General Purpose Register)
- STACK: 8 Levels
- One instruction build by two systems clock.
- Reset mode:
  (a) Power-On reset
  (b) Low voltage reset
  (c) RESETB/PC1 (if be set as reset pin) input a negative pulse.
  (d) Watchdog timer count overflow reset
- Dual Clock Mode
  - External RC or Crystal oscillator
  - Internal 4MHz RC oscillator
- Timer/counter: 3 units.
  - TM1: 16-bit, capture & Timer.
  - TM2: 8-bit, PWM (period) & Timer.
  - TM3: 8-bit, PWM (duty) & Timer.
  - TO : TM2(PWM) clock out
- Watchdog Timer: On chip WDT is based on an internal RC oscillator (for WDT used only). Have 8 period can be selected. User can extend the WDT overflow period by using prescaler.
- Interrupt events:
  (a) External interrupt (PA7~PA0).
  (b) Internal timer/event counter interrupt (TM1~TM3).
  (c) ADC end of conversion interrupt
- I/O port: 16 pins
- PWM: one channel
- ADC: max 10(6+4)-bit and 4 channels, at least 10-bit resolution. It can use at converter mode or compare mode
- Wake-up mode:
  1. Port A (PA7~PA0) pin change wakeup
  2. i_WDT wakeup
- Different Package Type:
  MK7A25PD20C: 20 pin DIP
  MK7A25PS20C: 20 pin SOP
  MK7A25PD18C: 18 pin DIP
  MK7A25PS18C: 18 pin SOP
  MK7A25PD14C: 14 pin DIP
  MK7A25PS14C: 14 pin SOP

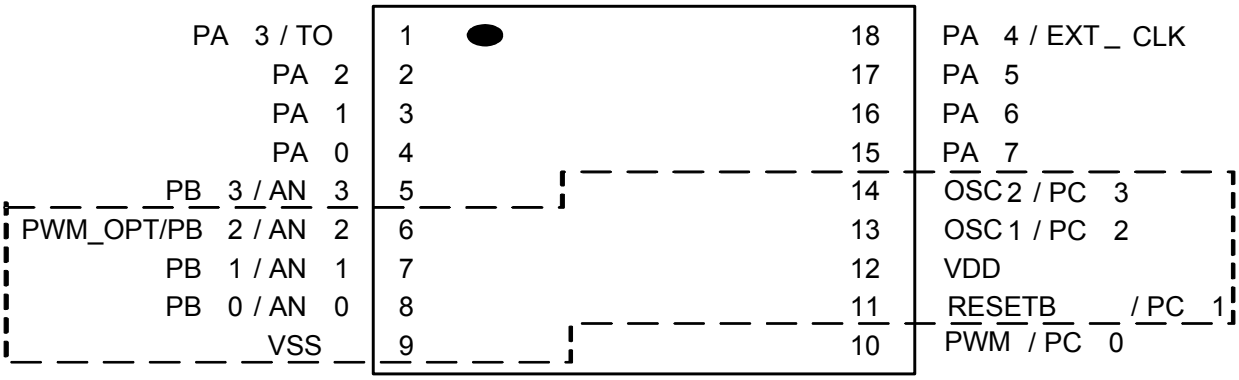
2. Block Diagram



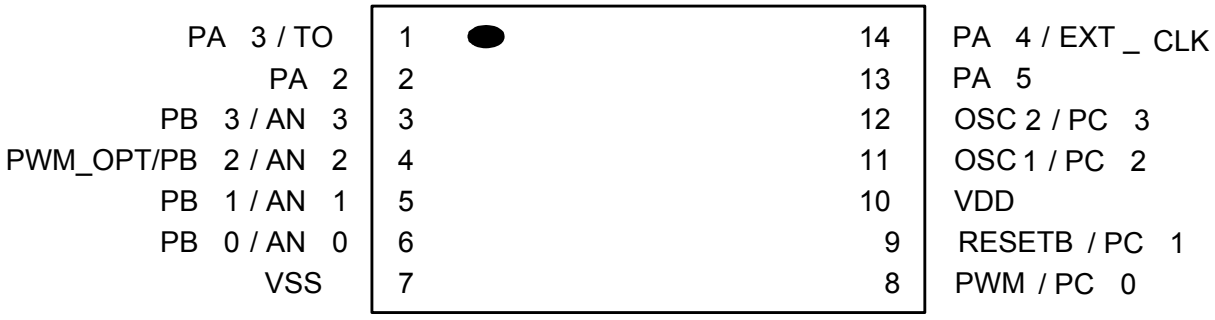
3. Pin Definition & Pad Assignment



PDIP 20 or SOP20



PDIP 18 or SOP18

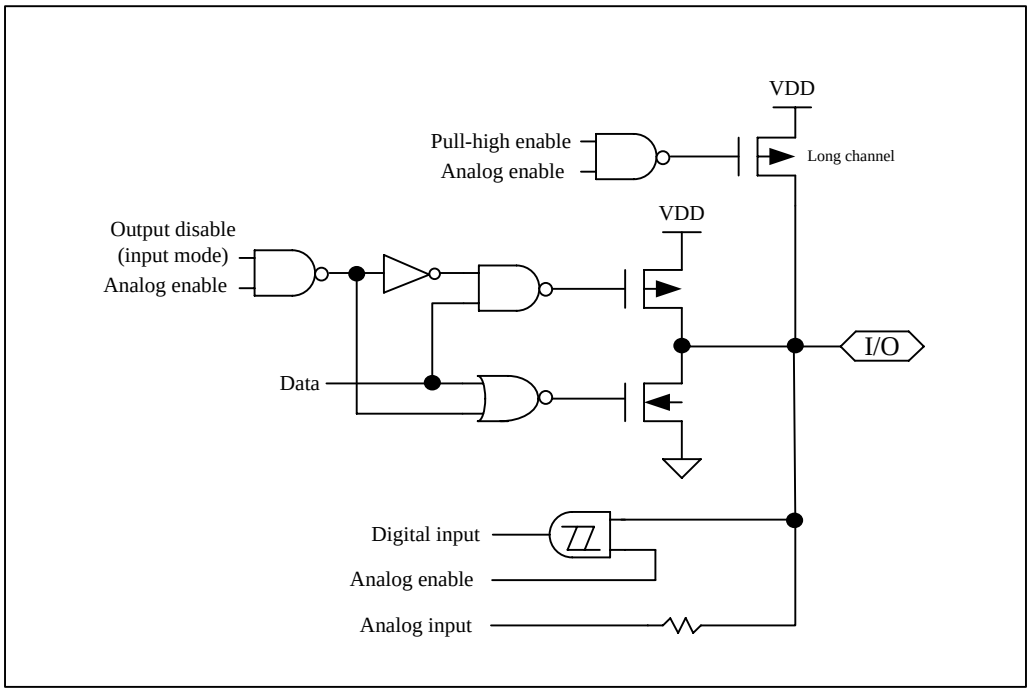


PDIP 14 or SOP14

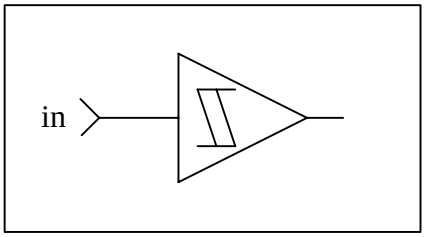
4. PIN Description

Pin name	I/O	Pin type	Description
PA0~2 PA5~7	I/O	D	1. I/O port (Pull-up option with Input mode) 2. Wake-up on pin change (option) 3. External interrupt input (option)
PA3/TO	I/O	D	1. I/O port (Pull-up option with Input mode) 2. Wake -up on pin change (option) 3. External interrupt input (option) 4. TO clock out
PA4/EXT_CLK	I/O	D	1. I/O port (Pull-up option with Input mode) 2. EXT_CLK clock input (or capture input) 3. Wake -up on pin change (option) 4. External interrupt input (option)
PB3/AN3	I/O	A	1. I/O port (Pull-up option with Input mode) 2. Analog input
PB1/AN1	I/O	A	
PB0/AN0	I/O	A	
PWM_OPT/PB2/AN2	I/O	A	1. I/O port (Pull-up option with Input mode) 2. Analog input 3. PWM option output, when PWM_OPT($1DH)set to 1.
PWM/PC0	I/O	D	4. I/O port (Pull-up option with Input mode) 5. PWM output , when PWM_OPT($1DH)reset to 0(init)
RESETB/PC1	I	B	1. Reset pin 2. Input port
OSC1/PC2	I, I/O	C	1. Oscillator input 2. I/O port (Pull-up option with Input mode)
OSC2/PC3	O, I/O	C	1. Oscillator output 2. I/O port (Pull-up option with Input mode)
PC4~PC5	I/O	D	1. I/O port (Pull-up option with Input mode) 2. The initial state pull_up(UC5,UC4) is enable.
VDD	P		Power input
VSS	P		Ground input

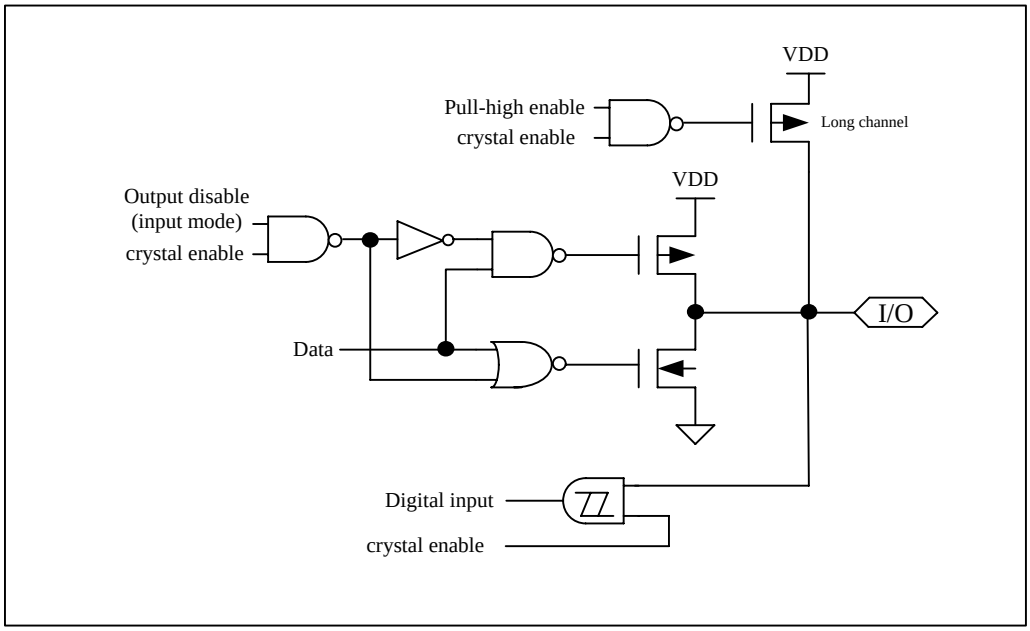
4.1 PIN Circuit



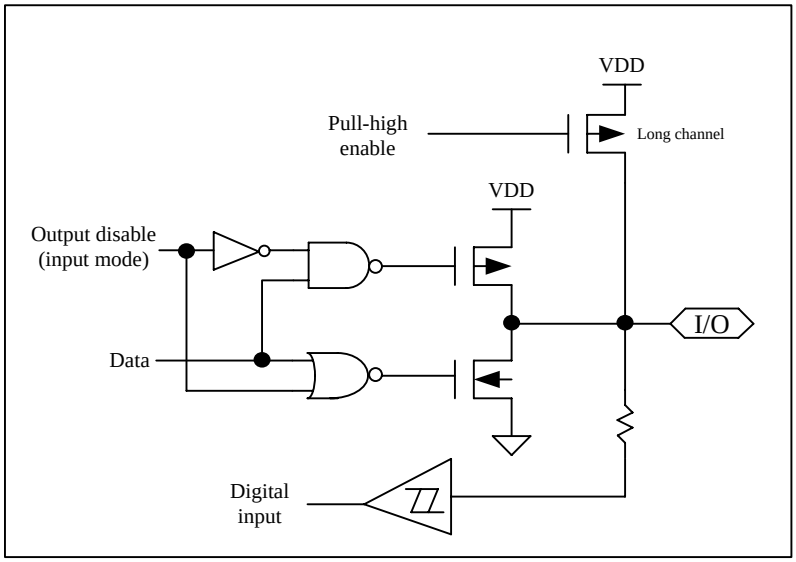
Pin circuit Type A



Pin circuit Type B



Pin circuit Type C



Pin circuit Type D

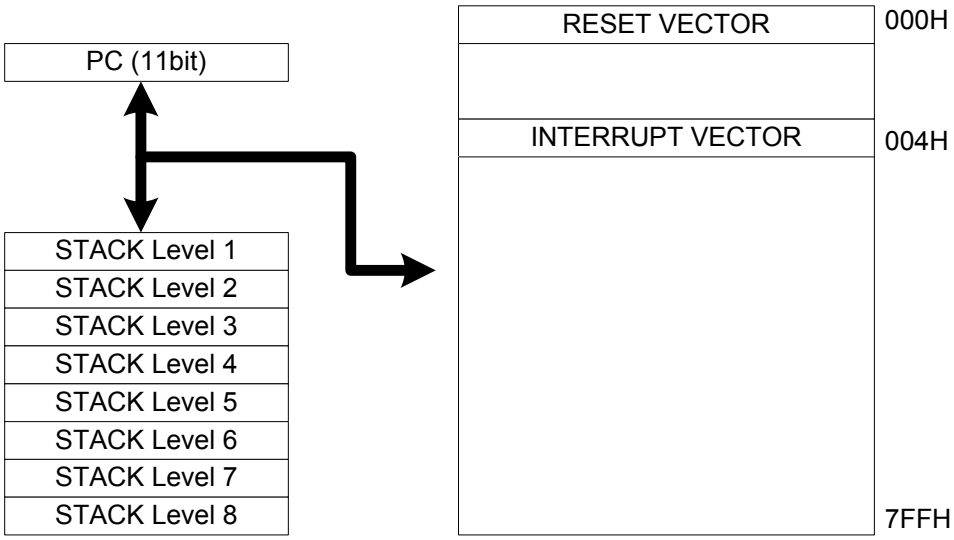
5. Memory Map

The MK7A25P have two kinds of memory which are ROM (program memory) and RAM (data memory). The ROM is used to store the program, table and interrupt vectors. It is continuous 2048 X 16bits and don't need to switch bank. The RAM is 165(37+128) X 8 bits that include special function register and general-purpose RAM.

5.1 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 11 bit which can directly address all the 2048 x 16 location. Look-up table can be put at anywhere of ROM.

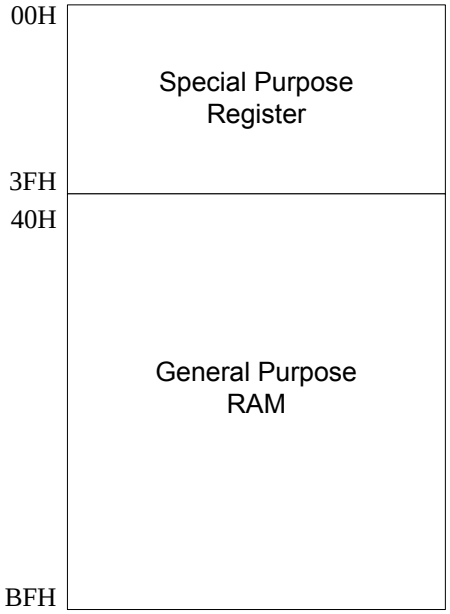
The RESET vector is located at 000H and Interrupt vector is at 004H. The map is as below:



5.2 Data Memory (RAM)

The total RAM volume are 165x8bits which includes two kinds of register group. One is 128×8 bits general purpose RAM, the other is special purpose register that are 37×8 bits. Every byte of special purpose register stored control's data or operation's data.

The data memory map is as below:



5.2.1 Special Purpose Register

Name	Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CONFIG_L		RST_DEF	--	--	WDTE	CPRT	INRC	FOSC1	FOSC0
CONFIG_H		ADJ6	ADJ5	ADJ4	ADJ3	ADJ2	ADJ1	ADJ0	RTCEN
INDF	$00	A7	A6	A5	A4	A3	A2	A1	A0
PCL	$01	A7	A6	A5	A4	A3	A2	A1	A0
PCH	$02	--	--	--	--	--	A10	A9	A8
STATUS	$03	--	--	--	$\overline{TO}$	$\overline{PD}$	Z	DC	C
FSR	$04	D7	D6	D5	D4	D3	D2	D1	D0
I/O PAD & Control									
Name	Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PA_DIR	$05	IOA7	IOA6	IOA5	IOA4	IOA3	CA2	IOA1	IOA0
PA_DAT	$06	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
PB_DIR	$07	--	--	--	--	IOB3	IOB2	IOB1	IOB0
PB_DAT	$08	--	--	--	--	DB3	DB2	DB1	DB0
PC_DIR	$09	--	--	IOC5	IOC4	IOC3	IOC2	--	IOC0
PC_DAT	$0A	--	--	DC5	DC4	DC3	DC2	DC1	DC0
Timer 1: 16-bit (Timer & capture)									
Name	Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM1_CTL1	$13	TM1_EN	WR_CNT	SUR1	SUR0	EDGE	PRE2	PRE1	PRE0
TM1_CTL2	$1F	E_CLR	--	--	--	--	--	--	--
CLR_CNT	$21	CLR_CNT							

TM1L_LA	$14	D7	D6	D5	D4	D3	D2	D1	D0
TM1H_LA	$15	D7	D6	D5	D4	D3	D2	D1	D0
TM1L_CNT	$16	D7	D6	D5	D4	D3	D2	D1	D0
TM1H_CNT	$17	D7	D6	D5	D4	D3	D2	D1	D0

Timer 2: 8-bit, PWM (period) & Timer

Name	Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM2_CTL1	$18	TM2_EN	WR_CNT	SUR1	SUR0	EDGE	PRE2	PRE1	PRE0
TM2_CTL2	$19	MOD	PWM_OS	TO_E	--	POS3	POS2	POS1	POS0
TM2_LA	$1A	D7	D6	D5	D4	D3	D2	D1	D0
TM2_CNT	$1C	D7	D6	D5	D4	D3	D2	D1	D0

Timer 3: 8-bit, PWM (duty) & Timer

Name	Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM_OPT	$1D	PWM_OPT	--	--	--	--	--	--	--
TM3_CTL	$1E	TM3_EN	WR_CNT	SUR1	SUR0	EDGE	PRE2	PRE1	PRE0
TM3_LA	$20	D7	D6	D5	D4	D3	D2	D1	D0
TM3_CNT	$22	D7	D6	D5	D4	D3	D2	D1	D0

IRQ

Name	Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IRQM	$25	INTM	ADCM	--	PAM	TM3M	TM2M/PWM	TM1M/CPT	--
IRQF	$26	--	ADCF	--	PAF	TM3F	TM2F/PWM	TM1F/CPT	--

ADC control

Name	Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AD_CTL1	$29	EN	--	MODE	--	--	--	CHSEL1	CHSEL0
AD_CTL2	$2A	RSUT	--	--	--	--	--	CKSEL1	CKSEL0
AD_CTL3	$2B	--	--	--	--	--	PBSEL2	PBSEL1	PBSEL0
AD_DAT_L	$2C	D1	D0	--	--	--	--	--	--
AD_DAT_H	$2D	D9	D8	D7	D6	D5	D4	D3	D2

Other

Name	Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PA_PLU	$31	UA7	UA6	UA5	UA4	UA3	UA2	UA1	UA0
PB_PLU	$33	--	--	--	--	UB3	UB2	UB1	UB0
PC_PLU	$35	--	--	UC5	UC4	UC3	UC2	--	UC0
WAKEUP	$3A	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
WDT_CTL	$3D	WDTEN	i_WDT	i_STAB	--	--	PRE2	PRE1	PRE0

TAB_BNK	**$3E**	**--**	**--**	**--**	**--**	**--**	**BNK2**	**BNK1**	**BNK0**
SYS_CTL	**$3F**	**CLKS**	**--**	**--**	**--**	**--**	**--**	**STP1**	**STP0**

<Note> "—" : mean no use.

5.2.2 Configure Register

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CONFIG_L	**RST_DEF**	**--**	**--**	**WDTE**	**CPRT**	**INRC**	**FOSC1**	**FOSC0**
-	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
CONFIG_H	**ADJ6**	**ADJ5**	**ADJ4**	**ADJ3**	**ADJ2**	**ADJ1**	**ADJ0**	**EXT_CLK**

- Bit15~9 (ADJ6~0): Used to calibrated internal RC oscillator.
- Bit8 (EXT_CLK): EXT_CLK input

  0: EXT_CLK (PA4) pin is normal I/O pin

  1: EXT_CLK (PA4) pin is timer source input & PA4 input
- Bit7 (RST_DEF): RESETB pin define

  0: RESETB is normal input pin

  1: RESETB is system reset pin
- Bit4 (WDTE): Watchdog timer enable/disable

  0: WDT disable

  1: WDT enable
- Bit3 (CPRT): ROM Code Protection bit

  0: ON

  1: OFF

- Bit2~0 (INRC, FOSC1~0): OSC type and system clock select

Bit2	Bit1	Bit0	OSC Type	Resonance Frequency
INRC	FOSC1	FOSC0		
0	0	0	LS (low speed)	System clock=32~200KHz
0	0	1	NS (Normal speed)	System clock=200K~10MHz
0	1	0	HS (high speed)	System clock=10~20MHz
0	1	1	External RC	System clock=32K ~ 10MHz
1	0	0	LS &Internal RC	1. Dual clock mode LS & 4MHz 2. Initial system clock=4MHz
1	0	1	NS &Internal RC	1. Dual clock mode NS & 4MHz 2. Initial system clock=4MHz
1	1	0	HS & Internal RC	1. Dual clock mode HS & 4MHz 2. Initial system clock=4MHz
1	1	1	Internal RC	1. System clock=4MHz 2. OSC1 & OSC2 work as I/O ports

6. Function Descriptions

This device provide many functions that are Timer, WDT, PWM, ADC, Capture, Interrupt, Table location, Reset, Program Counter and STATUS register. We would like to describe in detail.

6.1 I/O Port

There are 3 I/O ports to input or output data, each port has different function. The port A can be external interrupt, EXT_CLK clock input or capture input by register option. The port B can be analog input of ADC function. The port C can be external RC oscillator input, system reset input (reset function) or output of PWM function.

6.1.1 Port A

There are 3 registers to set the 8 I/O ports which are PA_DIR, PA_DAT, PA_PLU. Each pin of Port A can be external interrupt input or normal I/O. To know how to set these pins as external interrupt, please refer to Chapter 6.7. Pin PA4 has multiple functions. User should define it at Configure Register bit 8 at first.

A. PA_DIR($05H):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PA_DIR	IOA7	IOA6	IOA5	IOA4	IOA3	IOA2	IOA1	IOA0

- Bit7~0 (IOA7~0): To define each pin is input port or output port

  0: Output.

  1: Input.

B. PA_DAT($06H):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PA_DAT	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0

- Bit7~0 (DA7~0): Data buffer

C. PA_PLU($31H):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PA_PLU	UA7	UA6	UA5	UA4	UA3	UA2	UA1	UA0

- Bit7~0 (UA7~0): Pull up enables/disable

  0: Pull-up disable.

  1: Pull-up enable.

6.1.2 Port B

There are 3 registers to set the 4 I/O ports which are PB_DIR, PB_DAT, PB_PLU. Each pin of Port B can be set as ADC analog single input or normal I/O. PB2/ PWM_OPT can be set as PWM output(the PC0 is normal I/O).To know how to use ADC function, please refer to Chapter 6.8. User can set Pull up while Port B is set as input mode.

A. PB_DIR($07H):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PB_DIR	--	--	--	--	IOB3	IOB2	IOB1	IOB0

- Bit3~0 (IOB3~0): To define each pin is input port or output port

  0: Output.

  1: Input.

B. PB_DAT($08H):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PB_DAT	--	--	--	--	DB3	DB2	DB1	DB0

- Bit3~0 (DB3~0): Data buffer

C. PB_PLU($33H):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PB_PLU	--	--	--	--	UB3	UB2	UB1	UB0

- Bit3~0 (UB3~0): Pull up enable/disable.

  0: Pull-up disable.

  1: Pull-up enable.

6.1.3 Port C

There are 3 registers to set the 6 I/O ports which are PC_DIR, PC_DAT, PC_PLU. PC0 (PWM) can be set as PWM output(the PB2 is normal I/O). To know how to use PWM, please refer to Chapter 6.3. User can set Pull up while PC3, PC2, PC0 are set as input mode. PC1 (RESETB) can be set as system-reset(Low level voltage reset) single input pin by CONFIG register. Normally, PC2 (OSC1) and PC3 (OSC2) is external oscillator pin,

only when internal RC mode is selected, PC2 and PC3 can be normal I/O pin.

A. PC_DIR($09H):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PC_DIR	--	--	IOC5	IOC4	IOC3	IOC2	--	IOC0

- Bit5, 4, 3, 2, 0 (IOC5, 4, 3, 2, 0): To define each pin is input port or output port

  0: Output.

  1: Input.

<Note> IOC1 is input only.

B. PC_DAT($0AH):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PC_DAT	--	--	DC5	DC4	DC3	DC2	DC1	DC0

- Bit5, 4, 3, 2, 0 (DC5, 4, 3, 2, 0): Data buffer

<Note> 1. DC1 is input data only

C. PC_PLU($35H):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PC_PLU	--	--	UC5	UC4	UC3	UC2	--	UC0

- Bit5, 4, 3, 2, 0 (UC5, 4, 3, 2, 0): Pull up enable/disable.

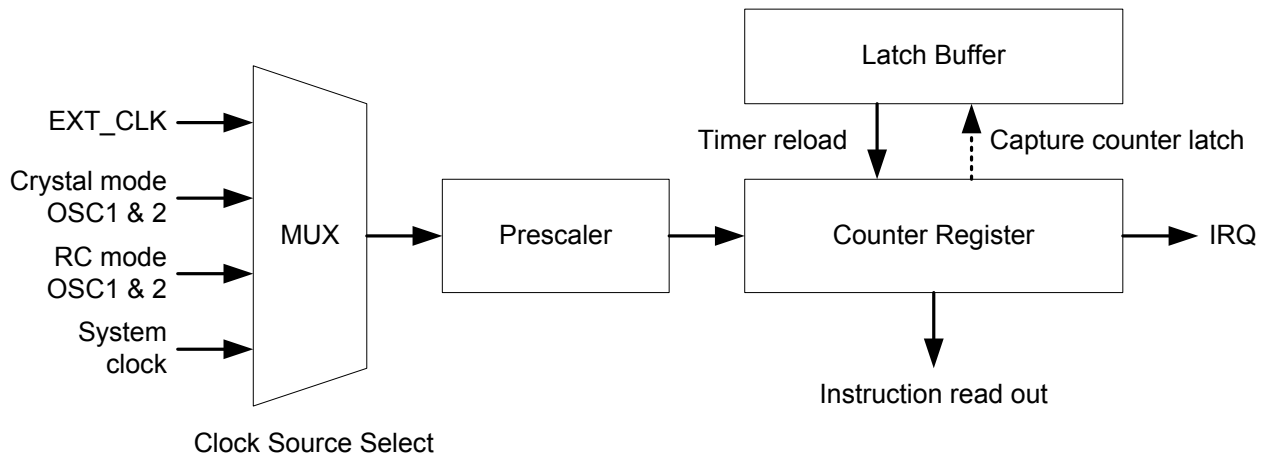
  0: Pull-up disable.

  1: Pull-up enable.

- Bit5, 4 (UC5, 4): Pull up is enable in the initial state.

6.2 Timer/Event Counter (TM1, TM2, TM3)

The MK7A25P provide 3 countdown timers/counters and 1 watchdog timer. Clock source of counters can be system clock or external clock by setting each timer control register. TM1 is a 16 bits counter, TM2 and TM3 are 8 bit counter. All these timers have auto reload function, TM1 has capture function and TM2/TM3 can be cascaded to do PWM function. The detailed registers setting and block diagram are as below.



6.2.1 TM1

TM1 is a 16 bit timer/counter. There are 5 register to set its attributes. To know how to use TM1 Capture function, please refer to Chapter 6.4.

A.TM1_CTL1 ($13H):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM1_CTL1	TM1_EN	WR_CNT	SUR1	SUR0	EDGE	PRE2	PRE1	PRE0

- Bit7 (TM1_EN): Timer1(TM1) enable/disable.
  0: TM1 disable.
  1: TM1 enable.
- Bit6 (WR_CNT): Latch buffer data write to counter register enable/disable.
  0: Latch buffer data write to counter register disable.
  1: Latch buffer data write to counter register enable

<Note> This bit is set only in the initial state of new timer/counter data to let latch buffer data write to counter register. When timer is overflow, the latch buffer data would auto reload into counter register. User doesn't need set again. It has no relation with this bit status.

- Bit5~4 (SUR1~0): TM1 clock source selection bits

Bit5	Bit4	TM1 clock source
SUR1	SUR0	
0	0	**EXT_CLK (PA4)**
0	1	**Crystal mode OSC1**
1	0	**RC mode (Ext. & Internal RC) OSC1**
1	1	**Clock source is system clock and capture input is PA4**

<Note> SUR1~0 define TM1 clock source. If TM1 is used in capture function, then SUR1~0 must set to (1,1) which can record the count from PA4 pin and the clock source is system clock. When capture is performed, the counter data of TM1 will be latched at TM1L_LA and TM1H_LA

- Bit3 (EDGE): TM1 clock source edge control bit

When TM1 use in Timer mode:
  0: increment when L→H on clock
  1: increment when H→L on clock

When TM1 use in Capture mode:
  0: To save TM1 counter register to latch buffer when EXT_CLK (PA4) L→H
  1: To save TM1 counter register to latch buffer when EXT_CLK (PA4) H→L

- Bit2~0 (PRE2~0): Set TM1 prescaler rate (Timer & capture)

Bit2	Bit1	Bit0	TMR1 Prescaler rate
PRE2	PRE1	PRE0	
0	0	0	1:1
0	0	1	1:2
0	1	0	1:4
0	1	1	1:8
1	0	0	1:16
1	0	1	1:32
1	1	0	1:64
1	1	1	1:128

B. TM1_CTL2 ($1FH):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM1_CTL2	ENC	--	--	--	--	--	--	--

- Bit7 (TM1_CTL1): Capture counter auto clear.

  0: Auto clear counter.(hardware)

  1: Clear counter by software.(to set CLR_CNT)

C. CLR_CNT ($21H):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CLR_CNT	CLR_CNT	--	--	--	--	--	--	--

- Write this register to clear capture counter. This is edge trigger, so write 0 or 1 are the same.

<Note> If TM1_CTL2 Bit 7 is set to 1 and CLR_CNT doesn't write data, then the counter will be kept instead of clear to 0.

D. TM1L_LA/TM1H_LA and TM1L_CNT/TM1H_CNT Register ($14H, 15H, 16H, 17H)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM1L_LA	D7	D6	D5	D4	D3	D2	D1	D0
TM1H_LA	D7	D6	D5	D4	D3	D2	D1	D0
TM1L_CNT	D7	D6	D5	D4	D3	D2	D1	D0
TM1H_CNT	D7	D6	D5	D4	D3	D2	D1	D0

<Note> TM1L_CNT & TM1H_CNT two register are read only

6.2.2 TM2 (or PWM period)

TM2 is an 8-bit timer/counter. There are 4 registers to set its attribute. The TM2 can be used as PWM period with TM3 to do PWM waveform.

A.TM2_CTL1 ($18H):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM2_CTL1	TM2_EN	WR_CNT	SUR1	SUR0	EDGE	PRE2	PRE1	PRE0

- Bit7 (TM2_EN): Timer2(TM2) enable/disable.

  0: TM2 disable.

  1: TM2 enable.

- Bit6 (WR_CNT): Latch buffer data write to counter register enable/disable.

  0: Latch buffer data write to counter register disable.

  1: Latch buffer data write to counter register enable

<Note> This bit is set only in the initial state of new timer/counter data to let latch buffer data write to counter register. When timer is overflow, the latch buffer data would auto reload into counter register. User doesn't need set again. It has no relation with this bit status.

- Bit5~4 (SUR1~0): TM2 clock source selection bits

Bit5	Bit4	TM2 clock source
SUR1	SUR0	
0	0	EXT_CLK (PA4)
0	1	Crystal mode OSC1
1	0	RC mode (Ext. & Internal RC) OSC1
1	1	Don't use

- Bit3 (EDGE): TM2 clock source edge control bit

  0: increment when L→H on clock

  1: increment when H→L on clock

- Bit2~0 (PRE2~0): Set TM2 prescaler rate

Bit2	Bit1	Bit0	TM2 Prescaler rate
PRE2	PRE1	PRE0	
0	0	0	1:1
0	0	1	1:2
0	1	0	1:4
0	1	1	1:8
1	0	0	1:16
1	0	1	1:32
1	1	0	1:64
1	1	1	1:128

B.TM2_CTL2 ($19H):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM2_CTL2	MOD	PWM_OS	TO_E	--	POS3	POS2	POS1	POS0

- Bit7 (MOD): Mode select bit.
  0: TM2 work in Timer mode.
  1: TM2 work in PWM mode
- Bit6 (PWM_OS): Output state of PWM select bit.
  0: The initial output state is H, this will change to L when TM3 timer overflow.
  1: The initial output state is L, this will change to H when TM3 timer overflow.
- Bit5(TO_E) : The Timer out(TO) enable/disable (pin shared with PA3)
  0: Set this pin is PA3 normal I/O pin
  1: Set this pin is TO (Timer output pin, frequency is TM2(PWM) counter frequency /2)

<Note> Before TO signal output, the PA3 must set as output port.

- Bit3~0 (POS3~0): PWM Poscaler selection bits (Only active in PWM mode)

Bit3	Bit2	Bit1	Bit0	PWM Poscaler rate
POS3	POS2	POS1	POS0	
0	0	0	0	1:1
0	0	0	1	1:2
0	0	1	0	1:3
.	.	.	.	.
.	.	.	.	.
1	1	1	0	1:15
1	1	1	1	1:16

These bits control how many times of PWM waveform output will service a PWM interrupt.

C. TM2_LA & TM2_CNT ($1AH, 1CH)

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	D7	D6	D5	D4	D3	D2	D1	D0

<Note> TM2_CNT register is read only

6.2.3 TM3 (or PWM duty)

TMR3 is an 8-bits timer/counter. There are 4 registers to set its attribute.. TMR3 can be used as PWM duty control with TM2 to do PWM waveform.

A.PWM_OPT ($1DH):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM_OPT	PWM_OPT	--	--	--	--	--	--	--

- Bit7 (PWM_OPT): PWM output pin select .
  0 : pwm output from PC0 (init) .
  1 : pwm output from PB2 .

B.TM3_CTL ($1EH):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM3_CTL	TMR3_EN	WR_CNT	SUR1	SUR0	EDGE	PRE2	PRE1	PRE0

- Bit7 (TM3_EN): TMR3 enable bit.
  0: TM3 disable.
  1: TM3 enable.

<Note> When TM2_CTL2 is set to PWM mode, this bit will be inhibited. The TM3 became duty counter of PWM waveform.

- Bit6 (WR_CNT): Latch buffer data write to counter register enable/disable.
  0: Latch buffer data write to counter register disable.
  1: Latch buffer data write to counter register enable

<Note> This bit is set only in the initial state of new timer/counter data to let latch buffer data write to counter register. When timer is overflow, the latch buffer data would auto reload into counter register. User doesn't need set again. It has no relation with this bit status.

- Bit5~4 (SUR1~0): TM3 clock source selection bits

Bit5	Bit4	TM3 clock source
SUR1	SUR0	
0	0	EXT_CLK (PA4)
0	1	Crystal mode OSC1
1	0	RC mode (Ext. & Internal RC) OSC1
1	1	Don't use

- Bit3 (EDGE): TM3 clock source edge control bit
  0: increment when L→H on clock
  1: increment when H→L on clock

- Bit2~0 (PRE2~0): Prescaler assignment bit.

Bit2	Bit1	Bit0	TM3 Prescaler rate
PRE2	PRE1	PRE0	
0	0	0	1:1
0	0	1	1:2
0	1	0	1:4
0	1	1	1:8
1	0	0	1:16
1	0	1	1:32
1	1	0	1:64
1	1	1	1:128

C. TM3_LA & TM3_CNT ($20H,22H)

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	D7	D6	D5	D4	D3	D2	D1	D0

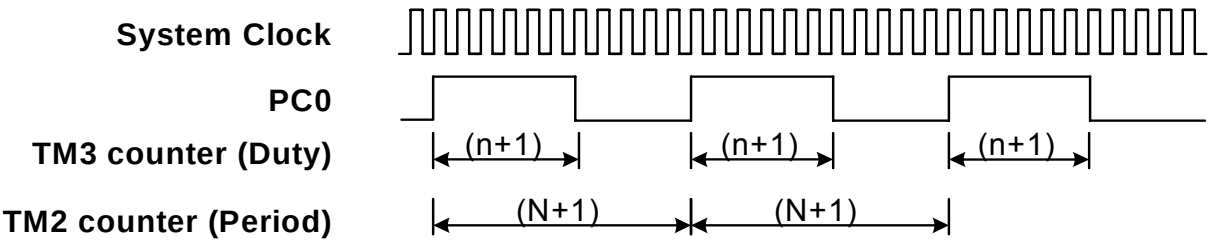
<Note> TM3_CNT register is read only

6.3 PWM (Pulse Width Modulation)

PWM waveform is composed of TM2 (period) and TM3(duty). These two timers can be used as general purpose time or PWM waveform counter by setting register. The setting flow is as following example.

Example:

```
ORG     000H
LGOTO   MAIN                ; Jump to main program.
ORG     004H                ; Interrupt program start vector.
BC      IRQF,b2             ; Clear PWM interrupt flag
                            ; User's Interrupt program
        RETI
ORG     050H
MAIN:
MOVLA   B'XXXXXXX0'         ; PC0 (PWM) has set to output.
MOVAM   PC_DIR              ; Set PC out
MOVLA   B'01010000'
MOVAM   TM2_CTL1            ; Set TM2 to Auto write and clock source is Crystal mode
                            OSC1.
MOVLA   B'10000000'
MOVAM   TM2_CTL2            ; Set TM2 to PWM mode and POS is1:1.
MOVLA   0AH                 ;The period is 0AH
MOVAM   TM2_LA              ; Set TM2_LA =0AH, WR_CNT bit=1, data will write to
                            TM2_CNT
MOVLA   B'01010000'
MOVAM   TM3_CTL             ; Set TM3 WR_CNT=1 and clock source is Crystal mode
                            OSC1.
MOVLA   05H                 ;The duty is 05H
MOVAM   TM3_LA              ; Set TM3_LA =05H, WR_CNT bit=1, data will write to
                            TM3_CNT
CLR     IRQF                ; Before enable IRQM, have to clear IRQF.
MOVLA   B'10000100'
MOVAM   IRQM                ; Enable INTM & TM2M interrupt.
BC      TM2_CTL1,b6         ; TM2 WR_CNT=0
BC      TM3_CTL ,b6         ; TM3 WR_CNT=0
BS      TM2_CTL1,b7         ; Enable PWM, PC0 start output.
```



PWM waveform

6.4 Capture

The capture function provides waveform measurement. The setting flow is as blow:

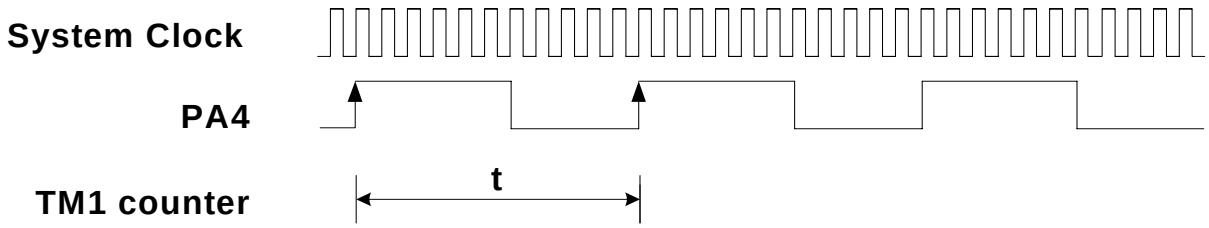
(a) User should set CONFIG register at first, setting EXT_CLK (bit 8) to “1”, let PA4 as capture input pin.

(b) Set TM1 prescaler rate and capture condition (polarity and edge). Then enable capture interrupt.

(c) Capture interrupt is occurred. The length of captured waveform will be latched at TM1L_LA and TM1H_LA.

Example1: Capture a cycle (two rising edge) and use interrupt (ENC=0, auto clear counter)

```
ORG     000H
LGOTO   MAIN            ; Jump to main program.
ORG     004H            ; Interrupt program start vector.
BC      IRQF,b1         ; Clear TM1 interrupt flag
COM     TM1L_LA,a       ;Read low-byte data (FFH - low register data)
MOVAM   40H             ;Save data to RAM
COM     TM1H_LA,a       ;Read high-byte data (FFH – high register data)
MOVAM   41H             ;Save data to RAM
MOV     TM1H_LA,a

        RETI
ORG     050H
MAIN:
MOVLA   B'XXX1XXXX'
MOVAM   PA_DIR          ;set PA4 INPUT
MOVLA   B'00110001'
MOVAM   TM1_CTL1        ; Capture mode enable, TM1 Prescaler rate 1:2.
CLR     IRQF            ; Before enable IRQM, have to clear IRQF.
MOVLA   B'10000010'
MOVAM   IRQM            ; Interrupt function enable, TM1 Interrupt enable.
BS      TM1_CTL1,b7     ; Start Capture function.
```



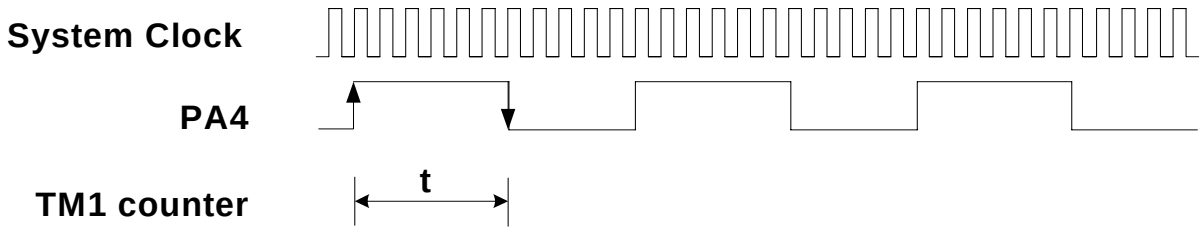
Example2: Capture a cycle (two rising edge) without using interrupt (ENC=0, auto clear counter)

```
MOVLA     B'XXX1XXXX'
MOVAM     PA_DIR              ;set PA4 INPUT
CLR       IRQF
MOVLA     B'10110000'
MOVAM     TM1_CTL1            ; Capture mode enable, TM1 Prescaler rate 1:1.

Loop:                         ; A waiting for Capture Flag enable loop.
BTSC      IRQF,b1             ; Consider this bit to know Flag is set or not.
LGOTO     Cap_part            ; If Flag is “1” then jump to read preload register
LGOTO     Loop                ; If Flag is “0” then jump to Loop
Cap_part:                     ; Read preload register after Captured
CLR       IRQF
COM       TM1L_LA,a           ; Read low-byte data (FFH - low register data)
MOVAM     40H                 ; Save data to RAM
COM       TM1H_LA,a           ; Read high-byte data (FFH - high register data)
MOVAM     41H                 ; Save data to RAM
LGOTO     Loop
```

Example3: Capture a pulse width and use interrupt (ENC=0, auto clear counter)

```
ORG       000H
LGOTO     MAIN
ORG       004H
BC        IRQF,b1             ; Clear TM1 interrupt flag
BTSC      TM1_CTL1,b3         ; Consider this bit to know which edge is happen.
LGOTO     Fall_edge           ; If now is rise edge interrupt, jump to Fall_edge part to set
                              control register to measure rise-fall edge length. If now is fall
                              edge interrupt, read data.
BS        TM1_CTL1,b3

RETI
Fall_edge:
BC        TM1_CTL1,b3         ; Set control register to capture interrupt happen at PA4
```

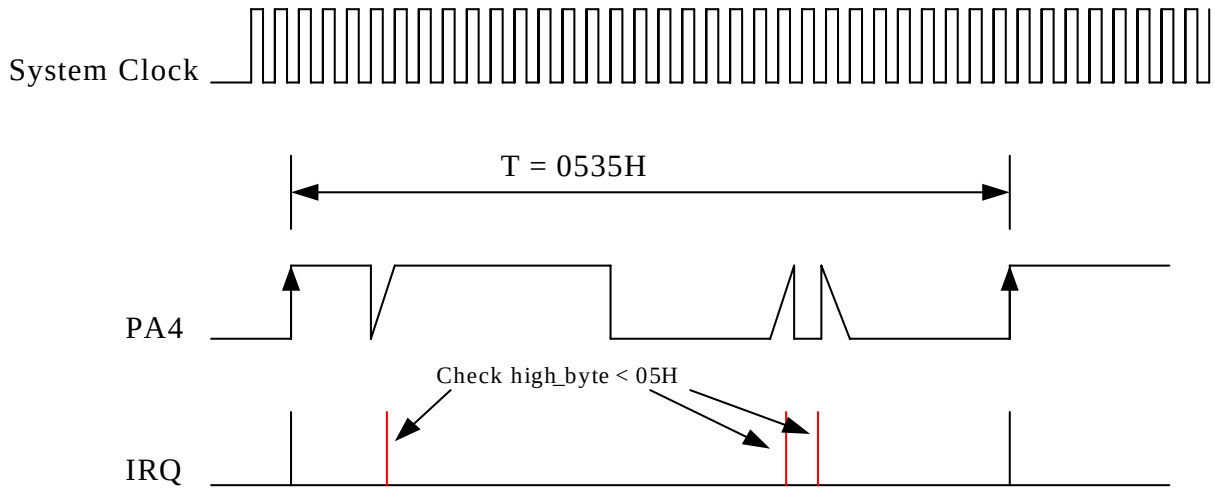
```
                                rise-edge.
COM         TM1L_LA,a           ; Read low-byte data (FFH - low register data)
MOVAM       40H                 ; Save data to RAM
COM         TM1H_LA,a           ; Read high-byte data (FFH - high register data)
MOVAM       41H                 ; Save data to RAM
RETI
ORG         050H
MAIN:
MOVLA       B'XXX1XXXX'
MOVAM       PA_DIR              ;set PA4 INPUT
MOVLA       B'00110000'
MOVAM       TM1_CTL1            ; Capture mode enable. Capture interrupt service at PA4 rise
                                edge. TM1 Prescaler rate 1:1.
CLR         IRQF                ; Before enable IRQM, have to clear IRQF.
MOVLA       B'10000010'
MOVAM       IRQM                ; Interrupt Function enable, TM1 Interrupt enable
BS          TM1_CTL1,b7
                                ; User's program
```



Example1: Capture a cycle (one rising edge) and use interrupt (ENC=1, clear counter by software)

```
ORG         000H
LGOTO       MAIN                ; Jump to main program.

ORG         004H                ; Interrupt program start vector.
BC          IRQF,b1             ; Clear TM1 interrupt flag
MOV         TM1H_LA,a           ;Read high-byte data
SUB         OLD_HDATA,a         ;Old_hdata – New_hdata(a)
MOVAM       SUM_H               ;Save high byte length
SUB         LENGTH,a            ;Check high_byte length =set value ?(check bounce)
BTSS        STATUS,b2           ;Yes : complete , No= continue counter
LGOTO       INTR
MOV         TM1L_LA,a           ;Read low-byte data
SUB         OLD_LDATA,a         ;Old_ldata – New_ldata(a)
```

```
MOVAM   SUM_L               ;Save low byte length
BTSS    STATUS,b0           ;Check haven't borrow (C=1?)
DEC     SUM_H,m             ; If C=0 (borrow), SUM_H – 1
INTR:
        RETI

ORG     050H
MAIN:
MOVLA   B'XXX1XXXX'
MOVAM   PA_DIR              ;set PA4 INPUT
MOVLA   B'00110000'
MOVAM   TM1_CTL1            ; Capture mode enable, TM1 Prescaler rate 1:1.
CLR     IRQF                ; Before enable IRQM, have to clear IRQF.
MOVLA   B'10000010'
MOVAM   IRQM                ; Interrupt function enable, TM1 Interrupt enable.
MOV     CLR_CNT,m           ; Clear counter
MOV     TM1L_LA,a
MOVAM   OLD_LDATA           ;Save low-byte data to register
MOV     TM1H_LA,a
MOVAM   OLD_HDATA           ;Save high-byte data to register
MOVLA   05H                 ;Set capture length =05
MOVAM   LENGTH              ;Save length to register
BS      TM1_CTL1,b7         ; Start Capture function.
```



6.5 WDT (Watchdog Timer)

WDT is a timer to prevent software from malfunction or jumping to an unknown location with unpredictable result. The source clock of WDT is an independent internal RC oscillator. This timer would be affected by temperature, voltage and different production lot.

A.WDT_CTL ($3DH):

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

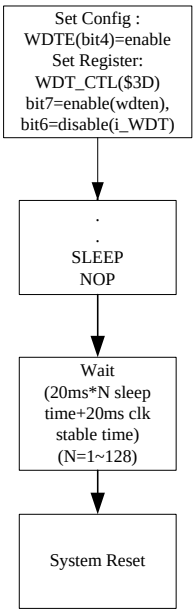
- Bit7 (WDTEN): Watchdog timer Enable bit
  0: WDT disable.
  1: WDT enable.
- Bit6 (i_WDT): i_WDT wakeup enable bit ,in this mode , pin change wakeup(PA7)disable .
  0: i_WDT disable.
  1: i_WDT enable.
- Bit5 (i_STAB): i_STAB wakeup times(in i_WDT mode)set bit .
  0: wakeup times=1.25ms(init) .
  1: wakeup times=625us

- Bit2~0 (PRE2~0): Set Prescaler rate. All the data are not accurate because it is RC OSC.

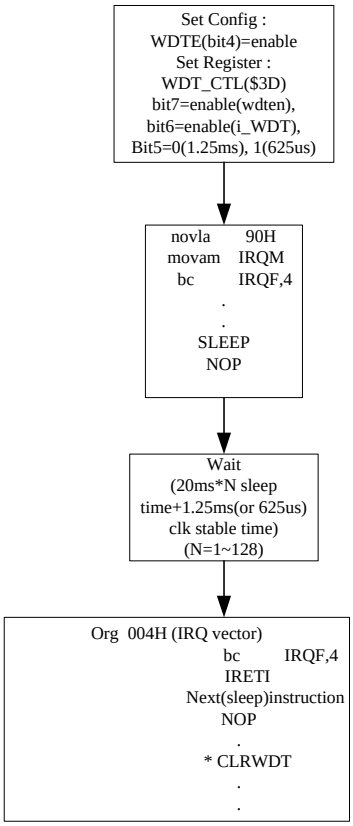
Bit2	Bit1	Bit0	WDT Prescaler rate
PRE2	PRE1	PRE0	
0	0	0	20mS
0	0	1	40mS
0	1	0	80mS
0	1	1	160mS
1	0	0	320mS
1	0	1	640mS
1	1	0	1.28S
1	1	1	2.56S

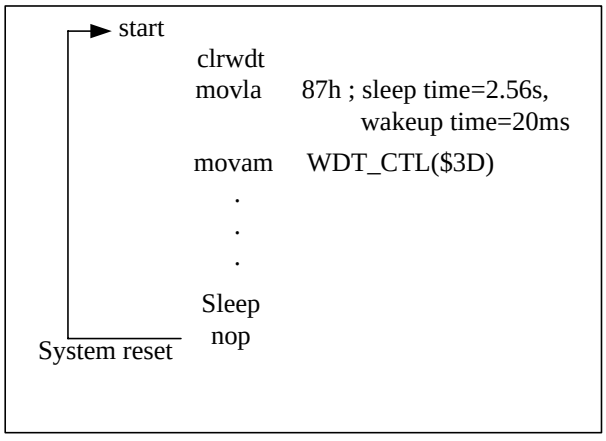
6.51 i_WDT wakeup flow

WDT-Wakeup :
(Watch-dog timer wake-up)

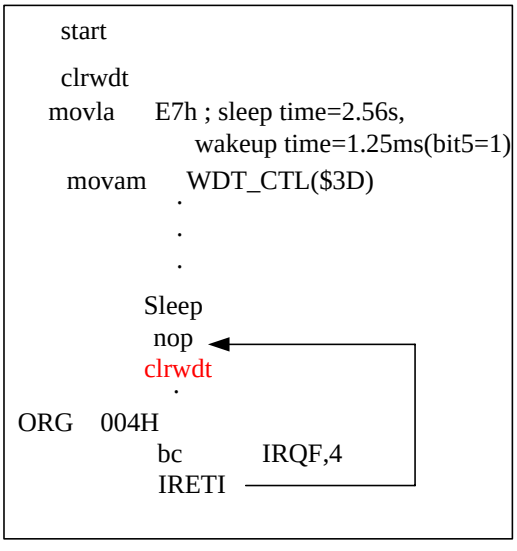


i_WDT-Wakeup :
(Internal watch-dog timer
wake-up)





*When wakeup must
CLRWDT, otherwise watch-
dog timer will keep operation



6.6 Reset

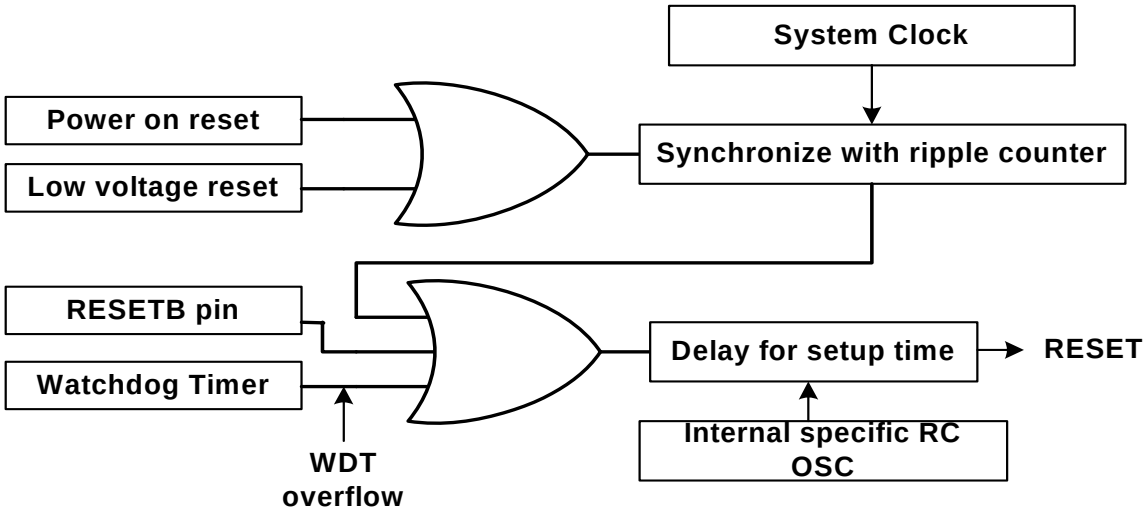
There are 4 events will cause reset which is listed as below. The power-down event will cause MK7A25P reset and the detected voltage is according to the bit6~bit5 in the CONFIG register. This condition is used to protect chip in deficient power environment. The last two cases are called warm reset. The 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.

(2) Low voltage reset (LVR).

(3) RESETB pin reset (input a negative pulse).

(4) WDT timer overflow reset.



<Note>: the watchdog setup time is approximately 20ms that will has some tolerance due to power voltage, process and ten Figure: System Reset Block

The default value during different reset condition

Address	Name	Cold Reset	Warm Reset
N/A	Accumulator	xxxx xxxx	pppp pppp
00H	INDF	0000 0000	0000 0000
01H	PCL	0000 0000	0000 0000
02H	PCH	- - - - -000	- - - - -000
03H	STATUS	0001 1xxx	0001 1xxx
04H	FSR	xxxx xxxx	pppp pppp
05H	PA_DIR	1111 1111	1111 1111
06H	PA_DAT	xxxx xxxx	pppp pppp
07H	PB_DIR	xxxx 1111	xxxx 1111

08H	**PB_DAT**	**xxxx xxxx**	**xxxx pppp**
09H	**PC_DIR**	**xx11 11x1**	**xx11 11x1**
0AH	**PC_DAT**	**xxxx xxxx**	**xxpp ppxp**
13H	**TM1_CTL1**	**0000 0000**	**0000 0000**
14H	**TM1L_LA**	**1111 1111**	**1111 1111**
15H	**TM1H_LA**	**1111 1111**	**1111 1111**
16H	**TM1L_DAT**	**1111 1111**	**1111 1111**
17H	**TM1H_ DAT**	**1111 1111**	**1111 1111**
18H	**TM2_CTL1**	**0000 0000**	**0000 0000**
19H	**TM2_CTL2**	**0000 0000**	**0000 0000**
1AH	**TM2_LA**	**1111 1111**	**1111 1111**
1CH	**TM2_DAT**	**1111 1111**	**1111 1111**
1DH	**PWM_OPT**	**0xxx xxxx**	**0xxx xxxx**
1EH	**TM3_CTL**	**0000 0000**	**0000 0000**
1FH	**TM1_CTL2**	**0000 0000**	**0000 0000**
20H	**TM3_LA**	**1111 1111**	**1111 1111**
22H	**TM3_DAT**	**1111 1111**	**1111 1111**
25H	**IRQM**	**00x0 000x**	**00x0 000x** [2]
26H	**IRQF**	**x0x0 000x**	**x0x0 000x**
29H	**AD_CTL1**	**0x0x xx00**	**0x0x xx00** [2]
2AH	**AD_CTL2**	**0xxx xx00**	**0xxx xx00** [2]
2BH	**AD_CTL3**	**xxxx x000**	**xxxx x000** [2]
2CH	**AD_DAT_L**	**00xx xxxx**	**00xx xxxx**
2DH	**AD_DAT_H**	**0000 0000**	**0000 0000**
31H	**PA_PLU**	**0000 0000**	**0000 0000**
33H	**PB_PLU**	**0000 0000**	**0000 0000**
35H	**PC_PLU**	**0011 0000**	**0011 0000**
3AH	**WAKEUP**	**0000 0000**	**0000 0000**
3DH	**WDT_CTL**	**100x x111**	**100x x111**
3EH	**TAB_BNK**	**xxxx x000**	**xxxx x000**
3FH	**SYS_CTL**	**0xxx xx00**	**0xxx xx00**

X: unknown; **?: value depends on condition ;**

P: previous data; **-:unimplemented and read as"0".**

6.7 Interrupt

The MK7A25P provides 8 external interrupt (PA0~7), three internal timer/event counter interrupt and an A/D converter interrupts. IRQM and IRQF registers are used to control or declare request state of all interrupts. The external interrupt is triggered by signal toggle of PA0~7 and the related interrupt request flag (PAF; bit4 of IRQF) will be set. The A/D converter interrupt is initialized by setting the A/D converter request flag (ADCF; bit 6 of IRQF) ,interrupt is occurred by end of A/D conversion.

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.

A. IRQM ($25H)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IRQM	INTM	ADCM	--	PAM	TM3M	TM2M/PWM	TM1M/CAPT	--

- Bit7 (INTM): Global enable bit.
  0: Disable. All interrupts are mask.
  1: Enable. All interrupt are unmask
  When interrupt is serving, the INTM will reset to "0" to prevent the other interrupt happen. After served, the IRETI instruction will set INTM as '1'.
- Bit6 (ADCM): ADC end of conversion (EOC) interrupt enable:
  0: Disable Interrupt
  1: Enable Interrupt
- Bit4 (PAM): PA interrupt enable.
  0: Disable Interrupt
  1: Enable Interrupt
- Bit3 (TM3M): TM3 interrupt enable
  0: Disable Interrupt
  1: Enable Interrupt
- Bit2 (TM2M/PWM): TM2/PWM interrupt enable
  0: Disable Interrupt
  1: Enable Interrupt
- Bit1 (TM1M/CAPT): TM1/Capture interrupt enable
  0: Disable Interrupt
  1: Enable Interrupt

B. IRQF ($26H)

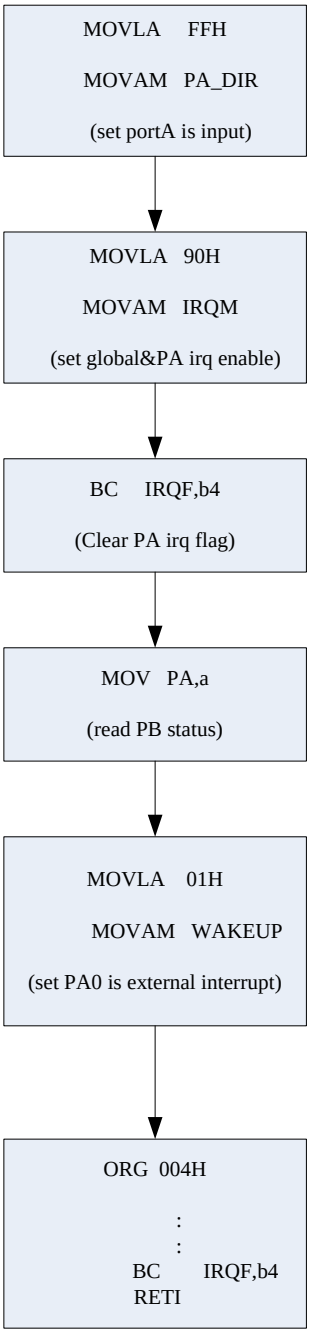
Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IRQF	--	ADCF	--	PAF	TM3F	TM2F/PWM	TM1F/CAPT	--

- Bit6 (ADCF): ADC end of conversion interrupt request flag:
  0: End of conversion interrupt request off
  1: End of conversion interrupt request on
- Bit4 (PAF): PA0~7 Interrupt request flag:
  0: PA interrupt request off
  1: PA interrupt request on
- Bit3 (TM3F): TM3 Interrupt request flag
  0: TM3 overflow interrupt request off
  1: TM3 overflow interrupt request on
- Bit2 (TM2F/PWM): TM2/PWM interrupt request flag
  0: TM2 overflow interrupt request off
  1: TM2 overflow interrupt request on
- Bit1 (TM1F/CAPT): TM1/Capture Interrupt flag
  0: TM1 overflow or Capture interrupt request off
  1: TM1 overflow or Capture interrupt request on

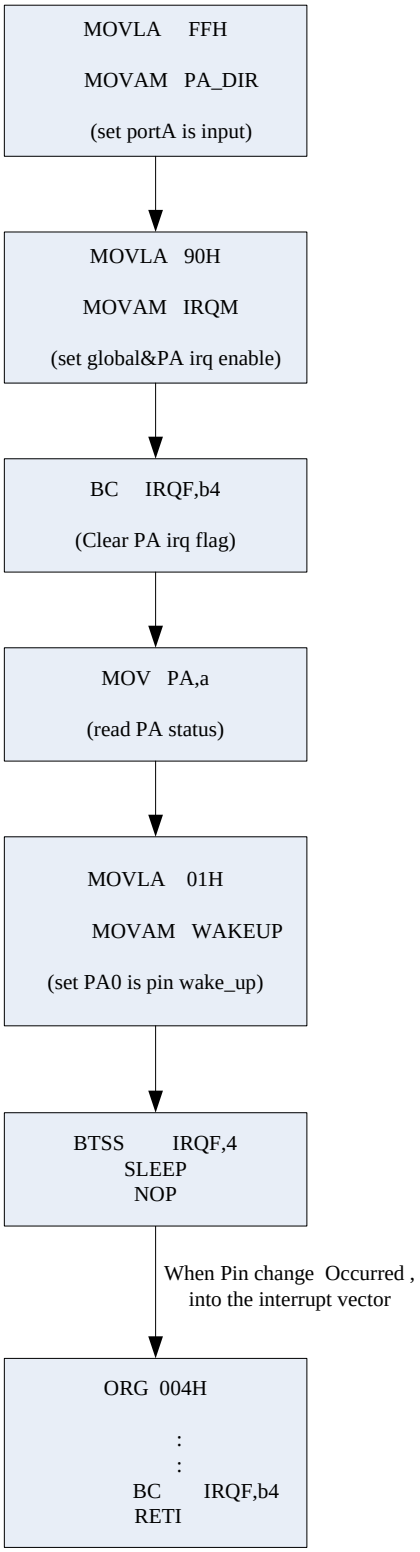
6.7.1 External interrupt / Wake up function

Port A (PA) 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 time 20ms. 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.

PortA External interrupt :



PortA Wake_up :



6.8 ADC

The MK7A25P provide 4 channels and 10-bit resolution A/D converter. The A/D converter contains 5 registers which are AD_CTL1 (29H), AD_CTL2 (2AH), AD_CTL3 (2BH), AD_DAT_L(2CH)and AD_DAT_H (2DH).

A. AD CTL1 ($29H)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AD_CTL1	EN	--	MODE	--	--	--	CHSEL1	CHSEL0

- Bit7 (EN): ADC enable bit

  0: ADC disable.

  1: ADC enable.

  <Note> When end of conversion, this bit will reset to “0”.

- Bit5 (MODE): ADC mode select bit

  0: ADC channels work as A/D conversion.

  1: ADC channels work as comparator

<Note> (a) If this bit is “1”, Vin data will compare with AD_DAT. The results are stored at the AD_CTL2 Bit7. If this bit is “0”, the Vin was converted into 8-bit digital data and saved in AD_DAT register.

(b) Vin: Input voltage from ADC channel

- Bit1~0 (CHSEL1~0): ADC input channel select bits

Bit1	Bit0	Input channel
CHSEL1	**CHSEL0**	
0	**0**	**Channel 0, PB0**
0	**1**	**Channel 1, PB1**
1	**0**	**Channel 2, PB2**
1	**1**	**Channel 3, PB3**

B. AD_CTL2 ($2AH)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AD_CTL2	RSUT	--	--	--	--	--	CKSEL1	CKSEL0

- Bit7 (RSUT): Compare mode result bit.

  0: Vin $<$ AD_DAT.

  1: Vin $\geqq$ AD_DAT

- Bit1~0 (CKSEL1~0) : ADC Conversion clock source select bits

Bit1	Bit0	Conversion clock
CKSEL1	CKSEL0	
0	0	System clock X2
0	1	System clock X8
1	0	System clock X32
1	1	System clock X128

<Note> The conversion clocks decide the conversion rate and precision. If fast conversion clock is selected, that will drop-off the precision. If user want to get more accurate A/D data, use slow speed is recommended.

C. AD_CTL3 ($2BH)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AD_CTL3	--	--	--	--	--	PBSEL2	PBSEL1	PBSEL0

- Bit2~0 (PBSEL2~0): ADC Channel input mode select bits

Bit2	Bit1	Bit0	PB3~PB0 configurations
PBSEL2	PBSEL1	PBSEL0	
0	0	0	PB3, PB2, PB1, PB0
0	0	1	PB3, PB2, PB1, AN0
0	1	0	PB3, PB2, AN1, AN0
0	1	1	PB3, AN2, AN1, AN0
1	X	X	AN3, AN2, AN1, AN0

<Note> To minimize power consumption, all the I/O pins should be carefully managed before entering sleep mode.

D. AD_DAT_L (2CH)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AD_DAT_L	D1	D0	--	--	--	--	--	--

E. AD_DAT_H (2DH)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AD_DAT_H	D9	D8	D7	D6	D5	D4	D3	D2

<Note> This register has two different usage. If work in compare mode, this data will compare with input voltage from ADC channel. In ADC mode, the registers stored the ADC conversion data.

Example :

```
        LOOP1
                BTSC        AD_CTL1,b7 ; Check A/D complete?
                BC          IRQF,b6 ; Clear A/D Flag ;
                MOVLA       03H
                MOVAM       AD_CTL1 ; set PB3 is A/D input ;
                MOVAM       AD_CTL2 ; set A/D clk=sys_clk/128;
                MOVLA       04H
                MOVAM       AD_CTL3 ; set PB0~PB3 is analog input ;
                BS          AD_CTL1,b7 ; Start A/D Transfer ;
      CHECK_FLAG
                BTSS        IRQF,b6 ; Check A/D flag=1 ? ;
                LGOTO       CHECK_FLAG
                MOVLA       AD_DAT,a
                MOVAM       40H ; Read A/D data to RAM
```

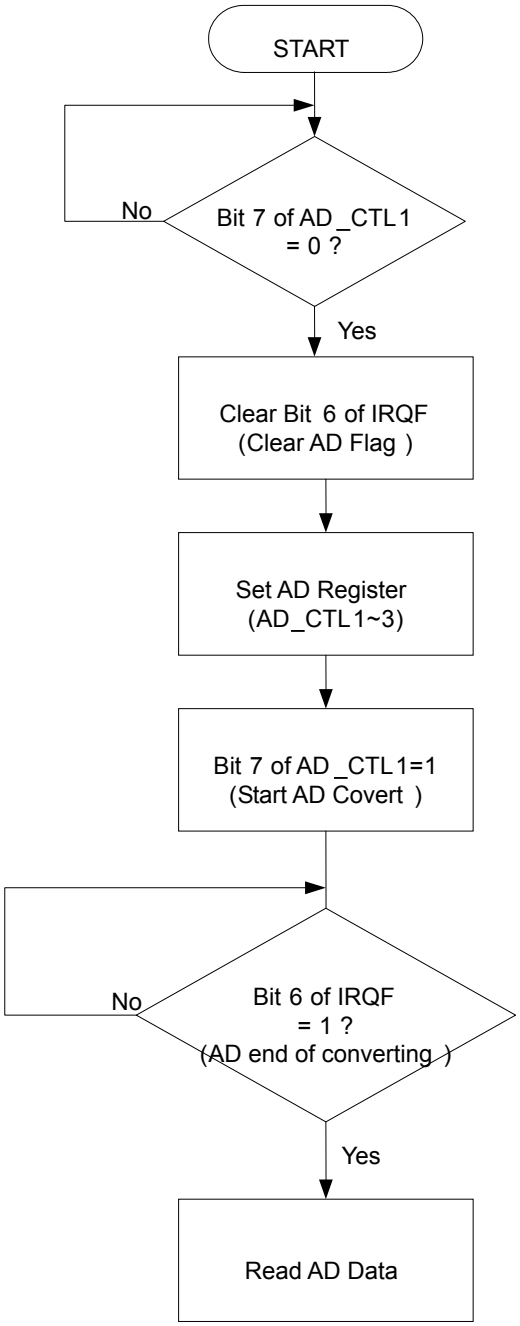


Fig. ADC Setting Flow

6.9 Table Look-up Function

The MK7A25P 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 is used to define address of high bit (MSB) of table location (3+8=11bits-address bit, 2^{11}=2Kbytes-data byte).

6.9.1 TAB_BNK ($3EH)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TAB_BNK	--	--	--	--	--	BANK2	BANK1	BANK0

- Bit2~0 (BANK2~0): High byte Table location select bits

Bit2	Bit1	Bit0	BANK select
BANK2	BANK1	BANK0	
0	**0**	**0**	**000 XXXX XXXX Table location**
0	**0**	**1**	**001 XXXX XXXX Table location**
0	**1**	**0**	**010 XXXX XXXX Table location**
.	**.**	**.**	**.**
1	**1**	**1**	**111 XXXX XXXX Table location**

6.9.2 Table look up Example program

The main function of below program is table looking-up. The results of TABRDL and TABRDH will be 55H and AAH (Address is 0704H).

```
#DEFINE     TAB_BNK   3EH       ; Define address 3EH of RAM named TAB_BNK

 BUFA       EQU       43H       ; Define address 43H of RAM named BUFA

(address)   ORG       0700H     ; Program start from 0700H of ROM
 0700H      MOVLA     00H       ; Save 00H to A register
 0701H      DW        1122H     ; Store 1122H at 0701H of ROM
 0702H      DW        3344H     ; Store 3344H at 0702H of ROM
 0703H      DW        6677H     ; Store 6677H at 0703H of ROM
 0704H      DW        55AAH     ; Store 55AAH at 0704H of ROM


            MOVLA     04H       ; Save 04H to A register (low bit address)
            MOVAM     BUFA      ; Save A register's value to BUFA
            MOVLA     0FH       ; Save 0FH to A register (high bit address)
            MOVAM     TAB_BNK   ; Save A register's value to TAB_BNK
            TABRDL    BUFA      ; Looking-up the Low-byte value of TAB_BNK and BUFA
                                  pointed address, saved it to A register.

            TABRDH    BUFA      ; Looking-up the High-byte value of TAB_BNK and BUFA
                                  pointed address, saved it to A register.
```

6.10 System Control

The MK7A25P provide Auto-Bank function and dual clock operation mode. When Sequence administers jumping, system will auto save the high byte of PC to prevent carry over error of PC that is Auto-Bank means. The dual clock mode has internal RC and external crystal clock source. User can use both of two in same time. For example, internal RC (4MHz) used to be system clock source, external crystal (32KHz) to be counter clock source. Because of external crystal is very accurate, User can get a very exact timer.

6.10.1 SYS_CTL ($3FH)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SYS_CTL	CLKS	--	--	--	--	--	STP1	STP0

- Bit7 (CLKS): System clock source select bit. Only active in dual clock mode.

  Dual clock mode:

  0: System clock is Internal RC

  1: System clock is OSC (crystal).

  Single clock mode:

  Don't use.

- Bit1 (STP1): Internal RC oscillator control bit.

  Dual clock mode:

  0: RC oscillation on.

  1: RC oscillation off.

  The other mode:

  Don't use.

<Note> Before setting RC oscillation off, CLKS should be switched to OSC oscillation at first.

- Bit0 (STP0): OSC (crystal) oscillator control bit

  Dual clock mode:

  0: OSC oscillation on.

  1: OSC oscillation off.

  The other mode:

  Don't use.

<Note> Before setting OSC oscillation off, CLKS should be switched to RC oscillation at first.

6.11 Program Counter – PC

The MK7A25P has an 11-bits program counter (PC) that includes PCL (8-bits) and PCH (3bits). 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 RET and RETI, RET_INT, PCH data would not be updated

(b) When execute LGOTO, LCALL and RET, PCH would be updated after mathematic operation

Example 1:

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
#DEFINE     ADMIN   41H     ; Adminicle for PCL operation.

  (address)
    1C0H    MOVLA   HIGH P1 ; Save P1 (1C6H) High-byte address to A register
                            ; PC=1C0H, PCL=C0H, PCH=00H
    1C1H    MOVAM   PCH     ; Save A register to PCH (To avoid jumping error at PCL
                            operation, store the real jumping high-byte address to PCH
                            at first.)
                            ; PC=1C1H, PCL =C1H, PCH=01H.
    1C2H    MOVLA   4BH     ; Save 4BH to A register (Address of ADDAM PCL,M is
                            1C5H, prepare jump to 210H, PCL have to add 210H-1C5H
                            =4BH)
                            ; PC=1C2H, PCL =C2H, PCH=01H.
    1C3H    MOVAM   ADMIN   ; Ready for PCL operation.
                            ; PC=1C3H, PCL =C3H, PCH=01H.
    1C4H    DEC     ADMIN, a ; ADMIN –1 (The real jumping happen at 1C6H, not 1C5H.
                            So 1C6H+(4BH-1H) =210H)
                            ; PC=1C4H, PCL =C4H, PCH=01H.
    1C5H    ADD     PCL,M   ; PCL add with A register, result store at PCL
                            ; PC=1C5H, PCL =C5H, PCH=01H
  P1: 1C6H  NOP             ; Jump to 0210H.
                            ; PC=1C6H, PCL=C6+4AH=10H with carry in 1,the carry
                            will count with PCH, PCH=01H, the purpose PC high
                            byte address will be  PCH + PCL's carry=02H. The
                            program will jump to PC=210H
    210H    MOVLA   00H     ; Purpose function part.
                            ; PC=210H, PCL =10H, PCH=01H
```

Example 2:

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

```
(address)
   1C0H   MOVLA   03H       ; Save 03H to A register
                            ; PC=1C0H, PCL=C0H, PCH=00H
   1C1H   MOVAM   PCH       ; Save A register to PCH
                            ; PC=1C1H, PCL =C1H, PCH=03H.
   1C2H   MOVLA   4BH       ; Save 4BH to A register
                            ; PC=1C2H, PCL =C2H, PCH=03H.
   1C3H   MOVAM   ADMIN     ; Ready for PCL operation.
                            ; PC=1C3H, PCL =C3H, PCH=01H.
   1C4H   DEC     ADMIN, a  ; ADMIN –1 (The real jumping happen at 1C6H, not 1C5H.
                            So 1C6H+(4BH-1H) =210H)
                            ; PC=1C4H, PCL =C4H, PCH=03H.
   1C5H   ADD     PCL,M     ; PCL add with A register, result store at PCL
                            ; PC=1C5H, PCL =C5H, PCH=03H
   1C6H   NOP               ; Jump to 410H.
                            ; PC=1C6H, PCL=C6+4AH=10H with carry in 1,the carry
                            will count with PCH, PCH=03H, the purpose PC high
                            byte address will be  PCH + PCL's carry=04H. The
                            program will jump to PC=410H
   410H   MOVLA   00H       ; Purpose function part.
                            ; PC=410H, PCL =10H, PCH=03H
```

Example3

The below program is show how PCL and PCH working through A register

```
(address)
   018H   MOVLA   02H       ; Save 02H to A register
   019H   MOVAM   PCH       ; Save A register to PCH(The purpose address is 200H, so
                            store "02H" to PCH)
   01AH   MOVLA   00H       ; Save 00H to A register
   01CH   MOVAM   PCL       ; Save A register to PCL(The purpose address is 200H, so
                            store "00H" to PCL)
   01DH   NOP               ; Jump to 200H.

   200H   MOVLA   00H       ; Purpose function part.
```

6.12 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	--	--	--	$\overline{TO}$	$\overline{PD}$	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 from sleep mode
0	1	WDT timer overflow from normal mode
1	0	Input a ‘low” at RESETB from sleep mode
1	1	Power on reset
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

7. Instruction

<Note> Instruction cycle is system clock/2

JUMP INSTRUCTION				
LCALL I	Call subroutine. However, LCALL can addressing 2K address	2	None	011i iiii iiii iiii
LGOTO I	Go branch to any address	2	None	010i iiii iiii iiii
LOGIC				
AND M, a	(M) · (acc) → (acc)	1	Z	1010 1000 MMMM MMMM
AND M, m	(M) · (acc) → (M)	1	Z	1010 1001 MMMM MMMM
ANDLA I	Immediate · (acc) → (acc)	1	Z	1111 1000 iiii iiii
COM M, a	~(M) → (acc)	1	Z	1010 0100 MMMM MMMM
COM M, m	~(M) → (M)	1	Z	1010 0101 MMMM MMMM
IOR M, a	(M) or (acc) → (acc)	1	Z	1011 1110 MMMM MMMM
IOR M, m	(M) or (acc) → (M)	1	Z	1011 1111 MMMM MMMM
IORLA I	Immediate or (acc) → (acc)	1	Z	1111 0010 iiii iiii
RL M, a	Rotate left from m to acc m[6:0]→acc[7:1] m[7]→ acc[0]	1	None	1110 0000 MMMM MMMM
RL M, m	Rotate left from m to itself m[6:0]→m[7:1] m[7]→ m[0]	1	None	1110 0001 MMMM MMMM
RLC M, a	Rotate left from m to acc m[7]→c m[6:0]→acc[7:1] c→acc[0]	1	C	1110 0010 MMMM MMMM
RLC M, m	Rotate left from m to itself m[7]→c m[6:0]→m[7:1] c→m[0]	1	C	1110 0011 MMMM MMMM
RR M, a	Rotate right from m to acc m[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	1	C	1110 1011 MMMM MMMM

	m[0]→c, c→m[7] m[7:1]→m[6:0]			
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 iiii iiii
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 (ROM bank)	2	None	1101 1000 MMMM MMMM
TABRDH M	Read high byte ROM table (ROM bank)	2	None	1101 1001 MMMM MMMM
DA 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
DA 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
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
MOVLA I	Immediate data → acc	1	None	1111 0000 iiii iiii
SUBLA I	(immediate data)-(Acc)→(Acc)	1	C, DC, Z	1111 0100 iiii iiii
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
CLRWDT	Clear watch-dog register	1	$\overline{TO}$, $\overline{PD}$	1111 1111 1111 0000
RET	Return (for lcall instruction)	2	None	1111 1111 1111 0001
RETI	Return and enable INTM (for IRQ)	2	None	1111 1111 1111 0010
RET_INT	Return (for IRQ)	2	None	1111 1111 1111 0011
SLEEP	Enter sleep (saving) mode	1	$\overline{TO}$, $\overline{PD}$	1111 1111 1111 100
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 A	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

<Note> After SLEEP instruction, please add NOP behind it to perform transient

8. Electrical Characteristics

8.1 Absolute Maximum Ratings

Supply Voltage …. Vss-0.3V to Vss+5.5V Storage Temperature ……. –40℃ to 125℃

Input Voltage …… Vss-0.3V to VDD+0.3V Operating Temperature… – 40℃ to 80℃

<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

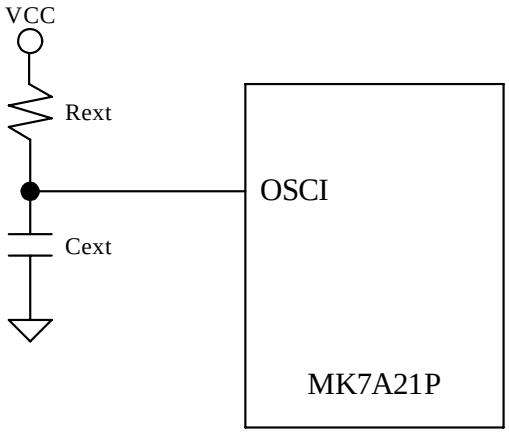
Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		VDD	Conditions				
VDD	Operating Voltage	---	f_{sys} =4MHZ	2.0		5.5	V
			f_{sys} =8MHZ	2.5		5.5	
			f_{sys} =12MHZ	3.3		5.5	
I_{DD1}	Operating Current (Crystal OSC)	3.3V	No Load , f_{sys} =4MHZ ADC disable		0.8		mA
		5V			1.6		
I_{DD2}	Operating Current (RC OSC)	3.3V	No Load , f_{sys} =4MHZ ADC disable		0.7		mA
		5V			1.2		
I_{DD3}	Operating Current (Crystal OSC)	5V	No Load , f_{sys} =8MHZ ADC disable		2.6		mA
I_{DD4}	Operating Current (Crystal OSC)	5V	No Load , f_{sys} =12MHZ ADC disable		3.6		mA
V_{IH}	Input High Voltage	5V	I/O Port	2		VDD	V
V_{IL}	Input Low Voltage	5V	I/O Port			0.8	V
I_{STB}	Standby Current	5V	WDT disable		3.5	5	μA
			WDT enable		12	15	
		3.3V	WDT disable		0.8	1.6	
			WDT enable		3	4	

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
I_{IL}	Input Leakage Current	5V	Vin=VDD, VSS			1	μA
I_{OH}	I/O Port Driving Current	5.5V	Voh=5V			9.9	mA
			Voh=4.5V			17.6	
			Voh=4V			24.8	
I_{OL}	I/O Port Sink Current	5.5V	Vol=0.5V			24.5	mA
			Vol=0.75V			35.3	
			Vol=1V			43.8	
R_{PH}	Pull-high Resistance	5V		70	85	100	KΩ
		3.3V		120	150	180	KΩ
V_{AD}	A/D input Voltage			0		VDD	V
E_{AD}	A/D Conversion Error					2	LSB
I_{ADC}	A/D Operating current	3.3V			100		uA
		5V			500		

8.3 AC Characteristics

Symbol	Parameter	Test Conditions		Min	Typ.	Max	Unit
		Conditions	VDD				
f_{sys1}	System Clock	LS Crystal mode	5V	32		200	Khz
			3V	32		200	
f_{sys2}	System Clock	NS Crystal mode	5V	0.2		10	Mhz
			3V	0.2		10	
f_{sys3}	System Clock	HS Crystal mode	5V	10		20	Mhz
f_{sys4}	System Clock	RC mode	5V	3.6	4	4.4	Mhz
			3V	3.6	4	4.4	
T_{wdt}	Watchdog Timer		5V		20		mS
			3V				
T_{rht}	Reset Hold Time		5V		20		mS
			3V				
T_{AD}	A/D clock period			3			us
T_{ADC}	A/D Conversion Time				30		t_{AD}
T_{ADCS}	A/D Sampling Time				8		t_{AD}
I_{OH}	Output high driving current	5V	Voh=0.9V_{DD}		-10		mA
		3V	Voh=0.9V_{DD}		-4		mA
I_{OL}	output low sink current	5V	Voh=0.1V_{DD}		20		mA
		3V	Voh=0.1V_{DD}		8		mA

8.4 EXT_RC Oscillator Frequencies

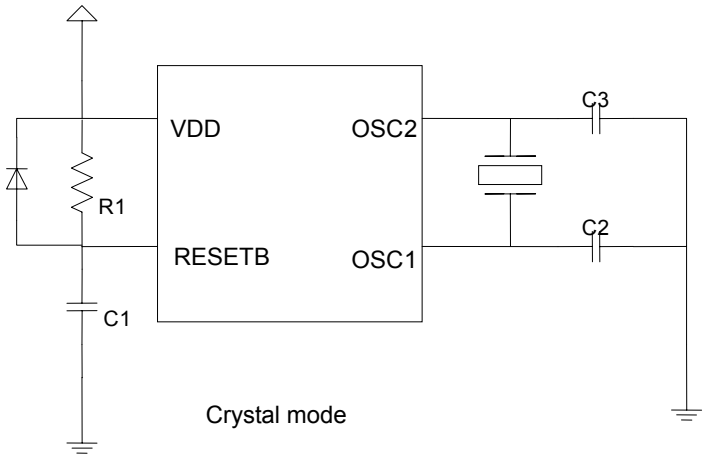


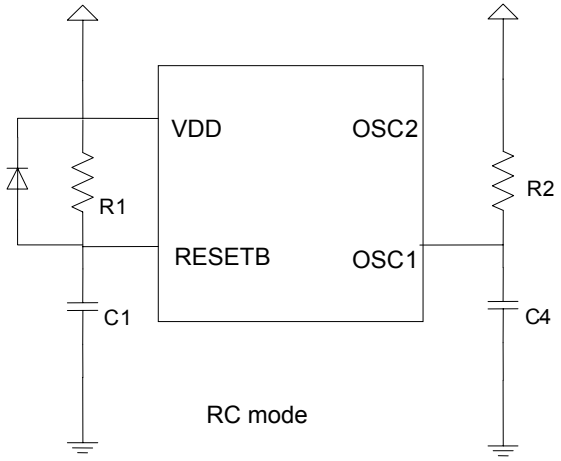
The typical external RC oscillation frequency is as below table

Cext = 0.1uf (104)

Rext	5V	3V
6M	32 KHZ	30KHZ
300K	460 KHZ	450 KHZ
140K	0.97 MHZ	0.90 MHZ
68K	2.0 MHZ	1.97 MHZ
37K	3.9 MHZ	3.85 MHZ
25K	6 MHZ	6 MHZ
15K	10.3 MHZ	10 MHZ

8.5 Oscillator circuit in different mode



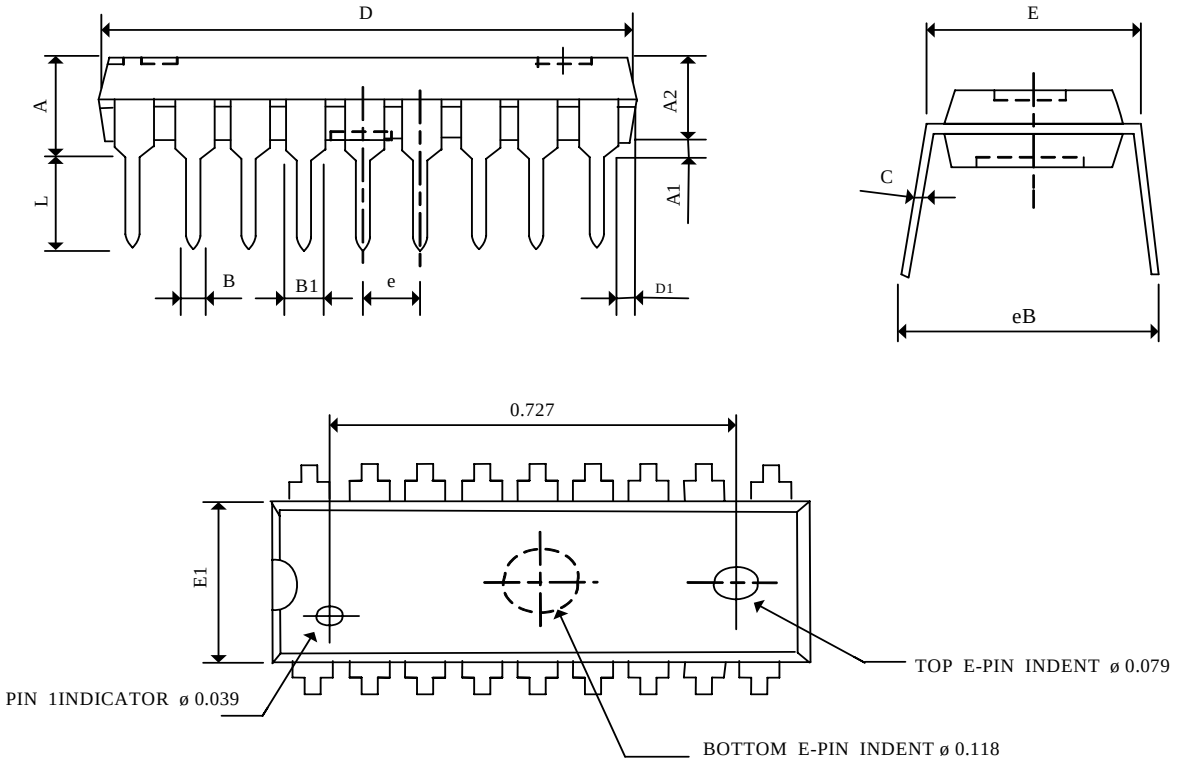


RC mode

Crystal mode	Crystal	C2	C3
LP mode	32 Khz	20~100P	20~100P
	455 Khz	100P	100P
NT mode	455 Khz	100P	100P
	1 Mhz	20~100P	20~100P
	2 Mhz	20~100P	20~100P
	4 Mhz	20~100P	20~100P
	8 Mhz	20~50P	20~50P
	10 Mhz	20~50P	20~50P
HS mode	12 Mhz	20~50P	20~50P
	16 Mhz	20~50P	20~50P
	20 Mhz	20~50P	20~50P

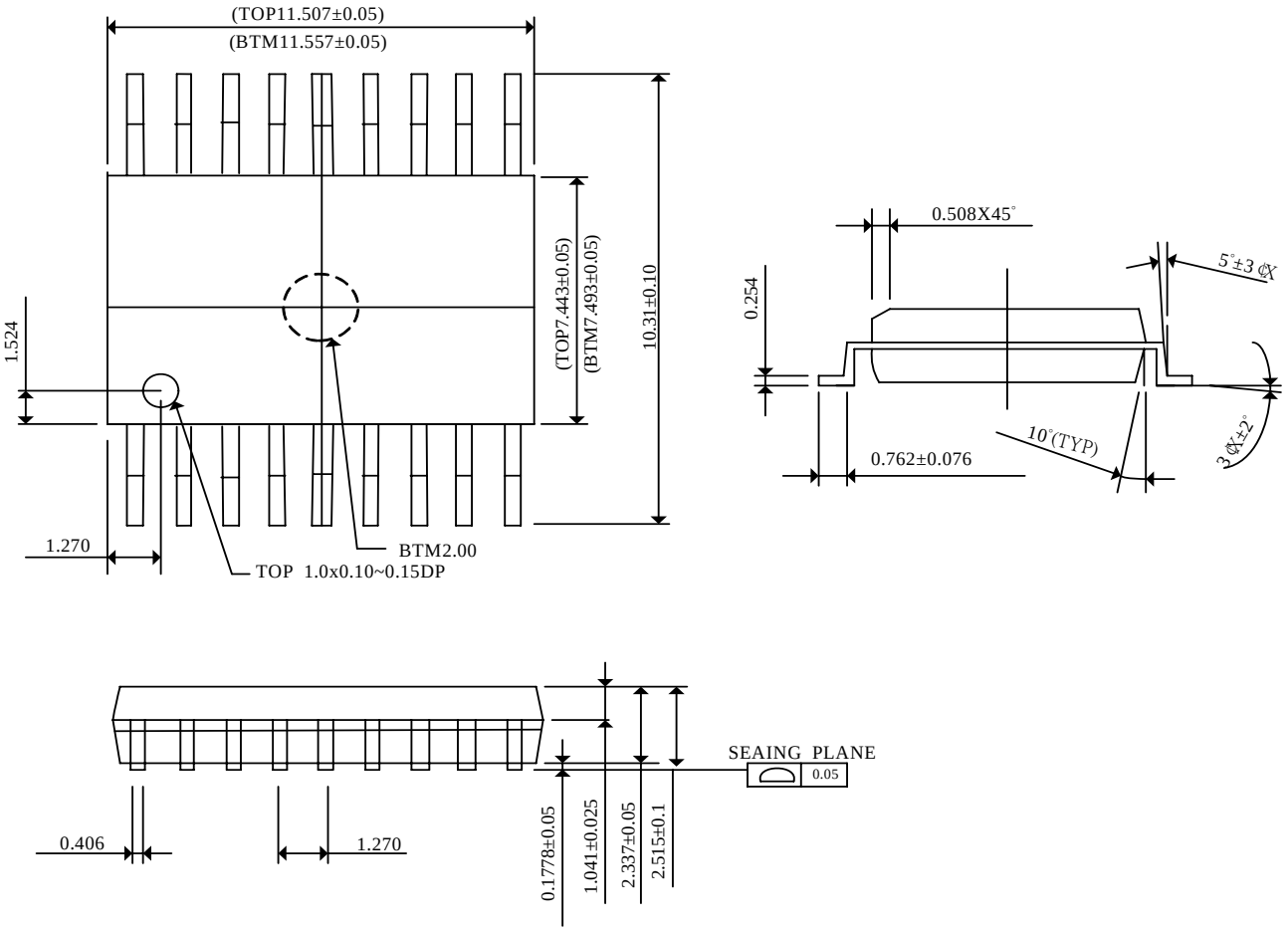
9. Package Dimension

(a) 18 Pin DIP

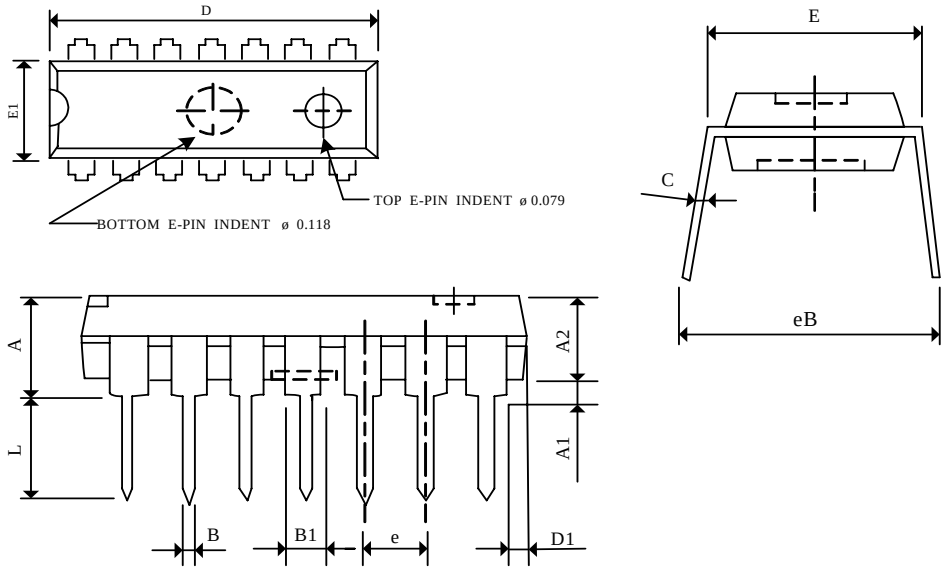


SYMBOLS	DIMENSIONS IN MILLIMETERS			DIMENSIONS IN INCHES		
	MIN	NOM	MAX	MIN	NOM	MAX
A	—	—	4.57	—	—	0.180
A1	0.38	—	—	0.015	—	—
A2	—	3.30	3.56	—	0.130	0.140
B	0.36	0.46	0.56	0.014	0.018	0.022
B1	1.27	1.52	1.78	0.050	0.060	0.070
C	0.20	0.25	0.33	0.008	0.010	0.013
D	22.71	22.96	23.11	0.894	0.904	0.910
D1	0.43	0.56	0.69	0.017	0.022	0.027
E	7.62	—	8.26	0.300	—	0.325
E1	6.40	6.50	6.65	0.252	0.256	0.262
e	—	2.54	—	—	0.100	—
L	3.18	—	—	0.125	—	—
eB	8.38	—	9.65	0.330	—	0.380

(b) 18 Pin SOP

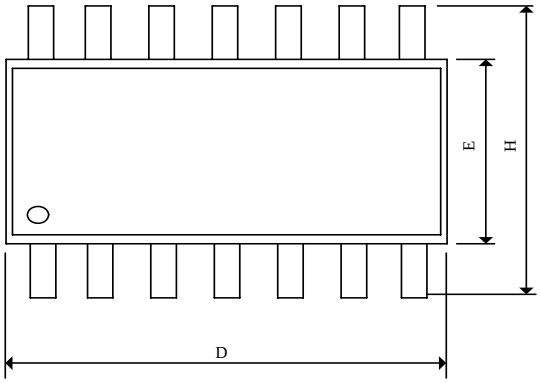

(TOP11.507±0.05)
(BTM11.557±0.05)
1.524
(TOP7.443±0.05)
(BTM7.493±0.05)
10.31±0.10
1.270
BTM2.00
TOP 1.0x0.10~0.15DP
0.508X45°
0.254
5°±3
0.762±0.076
10°(TYP)
3
SEAING PLANE
0.05
0.406
1.270
0.1778±0.05
1.041±0.025
2.337±0.05
2.515±0.1

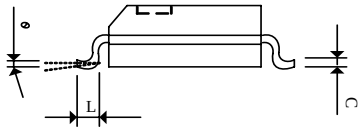
(c) 14 Pin DIP

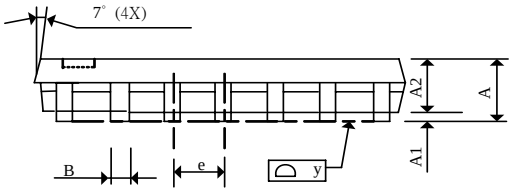


SYMBOLS	DIMENSIONS IN MILLIMETERS			DIMENSIONS IN INCHES		
	MIN	NOM	MAX	MIN	NOM	MAX
A	—	—	4.57	—	—	0.180
A1	0.38	—	—	0.015	—	—
A2	3.25	3.30	3.45	0.128	0.130	0.136
B	0.36	0.46	0.56	0.014	0.018	0.022
B1	1.27	1.52	1.78	0.050	0.060	0.070
C	0.20	0.25	0.33	0.008	0.010	0.013
D	18.90	19.15	19.30	0.744	0.754	0.760
D1	1.07	1.19	1.32	0.042	0.047	0.052
E	7.62	—	8.26	0.300	—	0.325
E1	6.35	6.50	6.65	0.250	0.256	0.262
e	—	2.54	—	—	0.100	—
L	3.18	—	—	0.125	—	—
eB	8.64	—	9.65	0.340	—	0.380

(d) 14 Pin SOP







SYMBOLS	DIMENSIONS IN MILLIMETERS			DIMENSIONS IN INCHES		
	MIN	NOM	MAX	MIN	NOM	MAX
A	1.35	1.60	1.75	0.053	0.063	0.069
A1	0.10	—	0.25	0.004	—	0.010
A2	—	1.45	—	—	0.057	—
B	0.33	—	0.51	0.013	—	0.020
C	0.19	—	0.25	0.007	—	0.010
D	8.55	—	8.75	0.337	—	0.344
E	3.80	—	4.00	0.150	—	0.157
e	—	1.27	—	—	0.050	—
H	5.80	—	6.20	0.228	—	0.244
L	0.40	—	1.27	0.016	—	0.050
Y	—	—	0.10	—	—	0.004
θ	0°	—	8°	0°	—	8°

<Important Notice>

Mikkon products are not designed, intended, authorized or warranted for use as components in systems, or equipment intended for surgical implantation, atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instrument, combustion control instruments, or for other applications intended to support or sustain life. Further more, Mikkon products are not intended for applications wherein failure of Mikkon products could result or lead to a situation wherein personal injury, death or severe property or environmental damage could occur.

Mikkon customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Mikkon for any damages resulting from such improper use or sales.