FEELING TECHNOLOGY

FM8PC713AM

OTP-Based 8-Bit Microcontroller with 10 bit ADC

Devices Included in this Data Sheet:

- FM8PC713AMAEx : 8-pin OTP device
- FM8PC713AMAEL : 6-pin OTP device

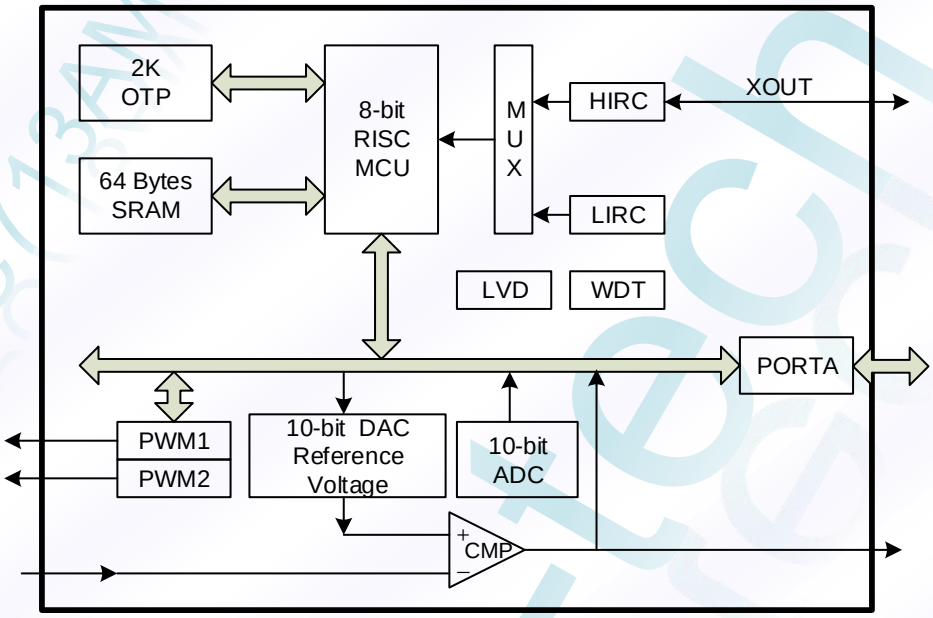
GENERAL DESCRIPTION

The FM8PC713AM is a low-cost, high speed, high noise immunity, OTP-based 8-bit CMOS microcontrollers. It employs a RISC architecture with only 39 instructions. All instructions are single cycle except for program branches which take two cycles. The easy to use and easy to remember instruction set reduces development time significantly.

FEATURES

- One clock per instruction cycle (1T, F_{CPU}=16M$_{HZ}$ @ VDD≥ 2.4V).
- 2K on chip OTP and 64 x 8 bits on chip general purpose registers (SRAM).
- 6-level deep hardware stack
- Optional F_{IRC} / (1, 2, 4, 8 or 16) MCU clock.
- Internal ±2% 16M$_{HZ}$ Oscillator.
- Internal oscillator 32 K$_{HZ}$ free run clock for Slow / Green mode.
- Three real-time Timer/Counter with 3-bit programmable pre-scaler
  - TM0: 8-bit Timer.
  - PWM0: 16-bit PWM/Buzzer/Timer, output pin can be programmable.
  - PWM1: 8-bit PWM/Buzzer/Timer, output pin can be programmable.
- Watch dog Timer.
- 6 I/O pins PORTA with independent direction control
  - 15mA drive/sink current on any I/O pins (PA1 excepted).
  - Soft-ware I/O pull-high or open-drain control.
- Total 5 external channel and 2 internal channel 10bit AD converter
  - Internal ±2% reference voltage 1.5V, 2.0V, 3.0V.
  - Internal 1/4V_{DD} and V_{SS} Measurement.
  - Optional Continues or Trigger mode to sample.
- Built-in voltage Comparator
  - Low input offset voltage.
  - Support 5 channel external input and internal 1/4V_{DD}, V_{SS} input comparison voltage.
- Five kinds of interrupt source: TM0, PWM0, PWM1 and LVDT, 6 external interrupt sources: PA5~PA0, Comparator and ADC.
- Built-in programmable 4 levels Low Voltage Detector (LVDT) (1.8V/2.0V/2.4V/3.0V).
- Built-in Fixed Low Voltage Detect (2.0V).

BLOCK DIAGRAM



PIN CONNECTION

Note that less than 8-pin MCU package types are not marketed in the following countries: USA, UK, Germany, The Netherlands, France and Italy.

SOT23-6

DIP/SOP8

PIN DESCRIPTIONS

Name	I/O	Description
PA0/AIN0	I/O	• Bi-direction I/O pin. • Software controlled pull-high. • A/D converter input. • Programming Comparator Negative input. • Programming PWM0/Buzzer0 or PWM1/Buzzer1 output.
PA1/RSTB	I/O	• Bi-direction I/O pin (Open-drain). • Software controlled pull-high. • System clear (RESET) input. Active low RESET to the device.
PA2/AIN1	I/O	• Bi-direction I/O pin. • Software controlled pull-high. • A/D converter input. • Programming Comparator Negative input. • Fixed Comparator Positive input. • Fixed PWM0/Buzzer0 differential output.
PA3/AIN2	I/O	• Bi-direction I/O pin. • Software controlled pull-high. • A/D converter input. • Programming Comparator Negative input. • Programming PWM0/Buzzer0 or PWM1/Buzzer1 output.
PA4/AIN3/ T0CK	I/O	• Bi-direction I/O pin. • Software controlled pull-high. • A/D converter input. • External Clock input. • Programming Comparator Negative input. • Programming PWM0/Buzzer0 or PWM1/Buzzer1 output.
PA5/AIN4/ CMPOP/XOUT	I/O	• Bi-direction I/O pin. • Software controlled pull-high. • A/D converter input. • Comparator output. • Up to 16M$_{HZ}$ F$_{IRC}$ external clock output. • Programming Comparator Negative input. • Programming PWM0/Buzzer0 or PWM1/Buzzer1 output. • Fixed PWM1/Buzzer1 differential output.
V_{DD}	-	Positive supply.
V_{SS}	-	Ground.

Legend: I=input, O=output, I/O=input/output.

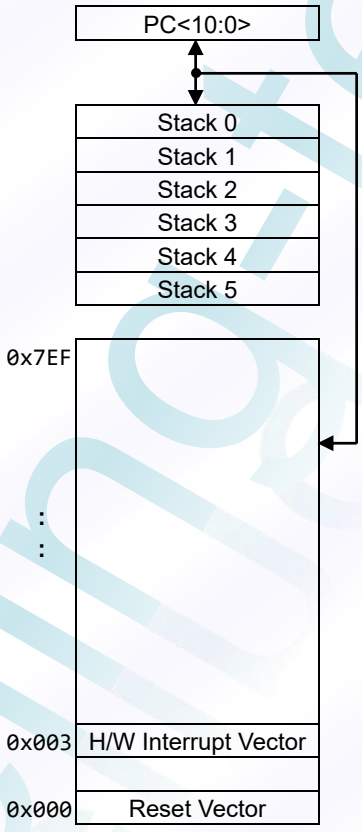
1.0 MEMORY ORGANIZATION

FM8PC713AM memory is organized into program memory and data memory.

1.1 Program Memory Organization

The FM8PC713AM have a 11-bit Program Counter capable of addressing a 2K program memory space.
The RESET vector for the FM8PC713AM is at 0x000.
The H/W interrupt vector is at 0x003.
FM8PC713AM supports all OTP area CALL/GOTO instructions without page.

Figure 1.1: Program Memory Map and STACK



FM8PC713AM

Note: Address 0x7F0 ~ 0x7FF Reserved for Configuration word.

1.2 Data Memory Organization

Data memory is composed of Special Function Registers and General Purpose Registers.
The General Purpose Registers are accessed either directly or indirectly through the FSR register.
The Special Function Registers are registers used by the CPU and peripheral functions to control the operation of the device.

Table 1.1: Registers File Map for FM8PC713AM

Address	Description
0x00	INDF
0x02	PCL
0x03	STATUS
0x04	FSR
0x05	PORTA
0x07	PAIE
0x09	PACON
0x0B	INTEN
0x0C	INTFLAG
0x0E	TM0
0x0F	TM0CON
0x10	T0RLD
0x11	WDT
0x12	PCON
0x13	CLKCFG
0x14	PWM0CON
0x15	PWM0CR
0x16	P0TMLB
0x17	P0TMDTLB
0x18	P0TMHB
0x19	P0TMDTHB
0x20	PWM1CON
0x21	PWM1CR
0x22	P1TM
0x23	P1TMDT
0x24	BZS
0x28	ADCON1
0x29	ADCON2
0x30	ADCHB
0x31	ADCLB
0x32	ADCON3
0x35	CMPCON1
0x36	DACR1HB
0x37	DACR1LB
0x38	CMPCON2
0x39	DACR2HB
0x3A	DACR2LB
0x40 ~ 0x7F	General Purpose Registers

0x08	PAMODE0
0x09	PAMODE1

0x17	P0TMPRLB

0x19	P0TMPRHB

0x23	P1TMPR

Table 1.2: The Registers Controlled by IOST or IOSTR Instructions

Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
I/O PAD Mode									
0x08 (r/w)	PAMODE0	-	-	PAMD05	PAMD04	PAMD03	PAMD02	PAMD01	PAMD00
0x09 (r/w)	PAMODE1	-	-	PAMD15	PAMD14	PAMD13	PAMD12	PAMD11	PAMD10
PWM									
0x17 (r/w)	P0TMPRLB	PWM0 Period cycle pre-set low-byte register							
0x19 (r/w)	P0TMPRHB	PWM0 Period cycle pre-set high-byte register							
0x23 (r/w)	P1TMPR	PWM1 Period cycle pre-set register							

Legend: - = unimplemented, read as '0'.

Table 1.3: Operational Registers Map

Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
Basic									
0x00 (r/w)	INDF	Uses contents of FSR to address data memory (not a physical register)							
0x02 (r/w)	PCL	Low order 8 bits of PC							
0x03 (r/w)	STATUS	-	-	FRP	$\overline{TO}$	$\overline{PD}$	Z	DC	C
0x04 (r/w)	FSR	-	Indirect data memory address pointer						
I/O PAD & I/O Interrupt Control									
0x05 (r/w)	PORTA	-	-	PA5	PA4	PA3	PA2	PA1	PA0
0x07 (r/w)	PAIE	-	-	PAIE5	PAIE4	PAIE3	PAIE2	PAIE1	PAIE0
0x09 (r/w)	PACON	-	-	PACON5	PACON4	PACON3	PACON2	PACON1	PACON0
Interrupt									
0x0B (r/w)	INTEN	GIE	-	-	LVDTSVIE	LVDT20IE	-	-	TM0IE
0x0C (r/w)	INTFLAG	-	PAIF	-	LVDTSVIF	LVDT20IF	-	-	TM0IF
Timer0									
0x0E (r)	TM0	8-bit real time clock/counter							
0x0F (r/w)	TM0CON	TM0EN	-	TM0PS2	TM0PS1	TM0PS0	TM0CKS1	TM0CKS0	-
0x10 (r/w)	T0RLD	TM0 Compare Pre-set register							
System									
0x11 (r/w)	WDT	WDTE	WDTSL	WDTPS2	WDTPS1	WDTPS0	LVRSL	-	-
0x12 (r/w)	PCON	-	GRN_MD	LVDIS	RSTSL1	RSTSL0	UGMD	LVDTSV	LVDT20
0x13 (r/w)	CLKCFG	SELCLK2	SELCLK1	SELCLK0	CLKSW	-	INCODS	EXOSEN	IRCEN
PWM0									
0x14 (r/w)	PWM0CON	P0INTSL	-	LPTS0	ENLP0	P0CKS1	P0CKS0	P0OSEL1	P0OSEL0
0x15 (r/w)	PWM0CR	PWM0EN	P0OUTS	P0TPS2	P0TPS1	P0TPS0	P0TMEN	P0TMIE	P0TMIF
0x16 (r)	P0TMLB	PWM0 real-time counter low-byte							
0x17 (r/w)	P0TMDTLB	PWM0 Duty cycle pre-set low-byte register							
0x18 (r)	P0TMHB	PWM0 real-time counter high-byte							
0x19 (r/w)	P0TMDTHB	PWM0 Duty cycle pre-set high-byte register							
PWM1									
0x20 (r/w)	PWM1CON	P1INTSL	-	LPTS1	ENLP1	P1CKS1	P1CKS0	P1OSEL1	P1OSEL0
0x21 (r/w)	PWM1CR	PWM1EN	P1OUTS	P1TPS2	P1TPS1	P1TPS0	P1TMEN	P1TMIE	P1TMIF
0x22 (r)	P1TM	PWM1 real-time counter							
0x23 (r/w)	P1TMDT	PWM1 Duty cycle pre-set register							
0x24 (r/w)	BZS	BZS1	BZS0	LVDTS1	LVDTS0	-	-	-	-
ADC/DAC/Comparator									
0x28 (r/w)	ADCON1	ADCEN	ADCST	CHSEL2	CHSEL1	CHSEL0	ADCSR2	ADCSR1	ADCSR0
0x29 (r/w)	ADCON2	ADCIE	ADCIF	-	ADCNT	ADCLS	ADCTCK	SELVER1	SELVER0
0x30 (r)	ADCHB	ADB9	ADB8	ADB7	ADB6	ADB5	ADB4	ADB3	ADB2
0x31 (r)	ADCLB	-	-	-	-	-	-	ADB1	ADB0
0x32 (r/w)	ADCON3	-	-	-	INEN4	INEN3	INEN2	INEN1	INEN0

Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x35 (r/w)	CMPCON1	CMPEN	COP1	CMPIE1	CMPRIF1	CMPFIF1	CMPCS2	CMPCS1	CMPCS0
0x36 (r/w)	DACR1HB	DA1B9	DA1B8	DA1B7	DA1B6	DA1B5	DA1B4	DA1B3	DA1B2
0x37 (r/w)	DACR1LB	-	-	-	-	-	-	DA1B1	DA1B0
0x38 (r/w)	CMPCON2	CMPINT1	COP2	CMPIE2	CMPRIF2	CMPFIF2	CMPINT2	TOGSEL1	TOGSEL0
0x39 (r/w)	DACR2HB	DA2B9	DA2B8	DA2B7	DA2B6	DA2B5	DA2B4	DA2B3	DA2B2
0x3A (r/w)	DACR2LB	-	-	-	-	-	-	DA2B1	DA2B0

Legend: - = unimplemented, read as '0'.

2.0 FUNCTIONAL DESCRIPTIONS

2.1 Operational Registers

2.1.1 INDF (Indirect Addressing Register)

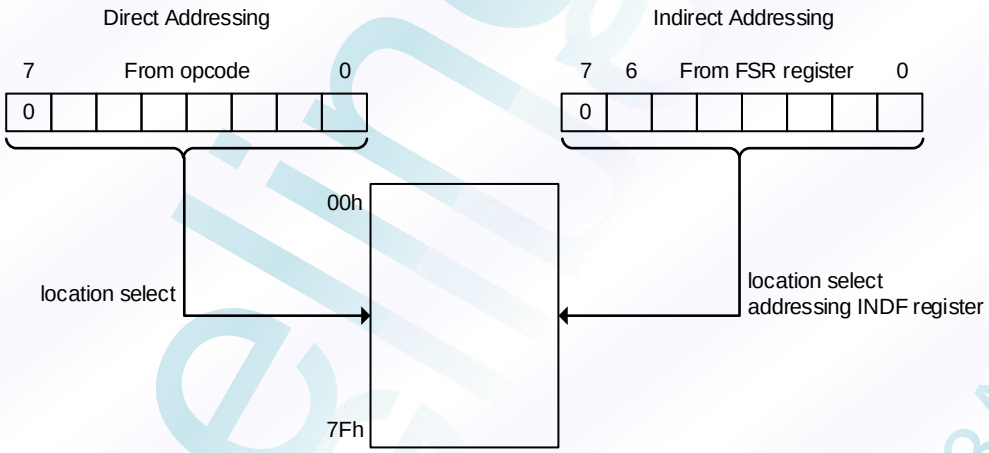
Read/Write-POR		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x00	INDF	Uses contents of FSR to address data memory (not a physical register)							

The INDF Register is not a physical register. Any instruction accessing the INDF register can actually access the register pointed by FSR Register. Reading the INDF register itself indirectly (FSR="0x00") will read 0x00. Writing to the INDF register indirectly results in a no-operation (although status bits may be affected).
The bits 6-0 of FSR register are used to select up to 128 registers (address: 0x00 ~ 0x7F).

Example 2.1: INDIRECT ADDRESSING

- Register file 0x48 contains the value 0x10
- Register file 0x49 contains the value 0x0A
- Load the value 0x48 into the FSR Register
- A read of the INDF Register will return the value of 0x10
- Increment the value of the FSR Register by one (@FSR=0x49)
- A read of the INDF register now will return the value of 0x0A.

Figure 2.1: Direct/Indirect Addressing



2.1.2 PCL (Low Byte of Program Counter) & Stack

Read/Write-POR		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x02	PCL	Low order 8 bits of PC							

FM8PC713AM device has a 11-bit wide Program Counter (PC) and six-level deep 11-bit hardware push/pop stack. The low byte of PC is called the PCL register. This register is readable and writable. The high byte of PC is called the PCH register. This register contains the PC<10:8> bits and is unable to readable or writable. All updates to the PCH register go through the CALL or GOTO instruction. As a program instruction is executed, the Program Counter will contain the address of the next program instruction to be executed. The PC value is increased by one, every instruction cycle, unless an instruction changes the PC.

For a GOTO instruction, the PC<10:0> is provided by the GOTO instruction word. The PCL register is mapped to PC<7:0>.

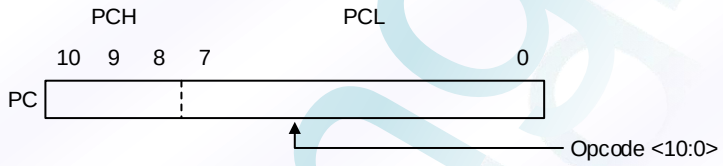
For a CALL instruction, the PC<10:0> is provided by the CALL instruction word. The next PC will be loaded (PUSHed) onto the top of STACK. The PCL register is mapped to PC<7:0>.

For a RETIA, RETFIE, or RETURN instruction, the PC are updated (POPed) from the top of STACK. The PCL register is mapped to PC<7:0>.

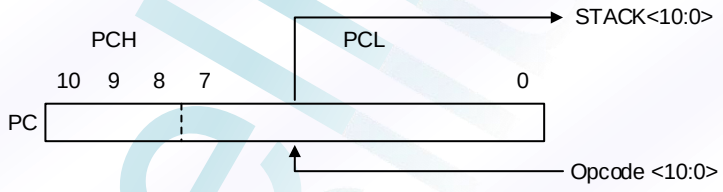
For any instruction where the PCL is the destination, the PC<7:0> is provided by the instruction word or ALU result.

Figure 2.2: Loading of PC in Different Situations

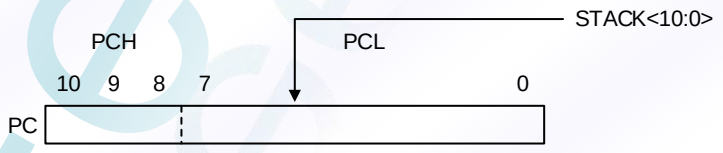
Situation 1: **GOTO** Instruction



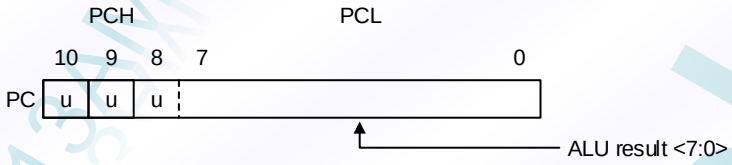
Situation 2: **CALL** Instruction



Situation 3: **RETIA**, **RETFIE**, or **RETURN** Instruction



Situation 4: Instruction with PCL as destination



2.1.3 STATUS (Status Register)

Read/Write-POR		-	-	R/W-0	R-#	R-#	R/W-0	R/W-0	R/W-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x03	STATUS	-	-	FRP	$\overline{TO}$	$\overline{PD}$	Z	DC	C

Legend: - = unimplemented, read as '0', # = refer Table 2.8 for detail description.

This register contains the arithmetic status of the ALU, the RESET status.
If the STATUS Register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the $\overline{TO}$ and $\overline{PD}$ bits are not writable. Therefore, the result of an instruction with the STATUS Register as destination may be different than intended. For example, CLRR STATUS will clear the upper three bits and set the Z bit. This leaves the STATUS Register as 000u u1uu (where u = unchanged).

C : Carry/borrow bit.
ADDAR, ADDIA
= 1, Carry occurred.
= 0, No Carry occurred.
SUBAR, SUBIA
= 1, No borrow occurred.
= 0, Borrow occurred.
Note : A subtraction is executed by adding the two's complement of the second operand. For rotate (RRR, RLR) instructions, this bit is loaded with either the high or low order bit of the source register.

DC : Half carry/half borrow bit.
ADDAR, ADDIA
= 1, Carry from the 4th low order bit of the result occurred.
= 0, No Carry from the 4th low order bit of the result occurred.
SUBAR, SUBIA
= 1, No Borrow from the 4th low order bit of the result occurred.
= 0, Borrow from the 4th low order bit of the result occurred.

Z : Zero bit.
= 1, The result of a logic operation is zero.
= 0, The result of a logic operation is not zero.

$\overline{PD}$: Power down flag bit.
= 1, after power-up or by the CLRWDT instruction.
= 0, by the SLEEP instruction.

$\overline{TO}$: Time overflow flag bit.
= 1, after power-up or by the CLRWDT or SLEEP instruction.
= 0, a watch-dog time overflow occurred.

FRP : From Register/Pin select bit.
= 1, When BSR/BCR instruction execute, the other bit of updated from pin.
= 0, When BSR/BCR instruction execute, the other bit of updated from register.

2.1.4 FSR (Indirect Data Memory Address Pointer)

Read/Write-POR		-	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x04	FSR	-	Indirect data memory address pointer						

Legend: - = unimplemented, read as '0'.

Bit6:Bit0 : Select registers address in the indirect addressing mode. See section 2.1.1 for detail description on page 8.

2.1.5 PORTA (Port Data Register)

Read/Write-POR		-	-	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x05	PORTA	-	-	PA5	PA4	PA3	PA2	PA1	PA0

Legend: - = unimplemented, read as '0'.

Reading the port (PORTA register) reads the status of the pins independent of the pin's input/output modes. Writing to these ports will write to the port data latch.
PORTA is 6-bit port data Registers.

PA5:PA0 : PORTA I/O pin.
= 1, Port pin is high level.
= 0, Port pin is low level.
Note: PA1 pin is open-drain output only. If PAMODE0<1> & PAMODE1<1> = "1,1", and PA1 bit is set to "1", the PA1 pin will be float. See section 2.2 block diagram for detail description on page 28.

2.1.6 PAIE (PORTA Interrupt Control Register)

Read/Write-POR		-	-	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x07	PAIE	-	-	PAIE5	PAIE4	PAIE3	PAIE2	PAIE1	PAIE0

Legend: - = unimplemented, read as '0'.

PAIE0 : = 1, Enable the input change interrupt function of PA0 pin.
= 0, Disable the input change interrupt function of PA0 pin.

PAIE1 : = 1, Enable the input change interrupt function of PA1 pin.
= 0, Disable the input change interrupt function of PA1 pin.

PAIE2 : = 1, Enable the input change interrupt function of PA2 pin.
= 0, Disable the input change interrupt function of PA2 pin.

PAIE3 : = 1, Enable the input change interrupt function of PA3 pin.
= 0, Disable the input change interrupt function of PA3 pin.

PAIE4 : = 1, Enable the input change interrupt function of PA4 pin.
= 0, Disable the input change interrupt function of PA4 pin.

PAIE5 : = 1, Enable the input change interrupt function of PA5 pin.
= 0, Disable the input change interrupt function of PA5 pin.

2.1.7 PACON (PORTA Interrupt pin-change/edge control Register)

Read/Write-POR		-	-	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x09	PACON	-	-	PACON5	PACON4	PACON3	PACON2	PACON1	PACON0

Legend: - = unimplemented, read as '0'.

PACON0 : = 1, Pin-change to interrupt of PA0 pin.
= 0, Falling edge to interrupt of PA0 pin.

PACON1 : = 1, Pin-change to interrupt of PA1 pin.
= 0, Falling edge to interrupt of PA1 pin.

PACON2 : = 1, Pin-change to interrupt of PA2 pin.
= 0, Falling edge to interrupt of PA2 pin.

PACON3 : = 1, Pin-change to interrupt of PA3 pin.
= 0, Falling edge to interrupt of PA3 pin.

PACON4 : = 1, Pin-change to interrupt of PA4 pin.
= 0, Falling edge to interrupt of PA4 pin.

PACON5 : = 1, Pin-change to interrupt of PA5 pin.
= 0, Falling edge to interrupt of PA5 pin.

2.1.8 INTEN (Interrupt Mask Register)

Read/Write-POR		R/W-0	-	-	R/W-0	R/W-0	-	-	R/W-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x0B	INTEN	GIE	-	-	LVDTSVIE	LVDT20IE	-	-	TM0IE

Legend: - = unimplemented, read as '0'.

TM0IE : Timer0 match interrupt enable bit.
= 1, Enable the Timer0 match interrupt.
= 0, Disable the Timer0 match interrupt.

LVDT20IE : LVDT2.0V interrupt enable bit.
= 1, Enable the LVDT 2.0V falling edge interrupt.
= 0, Disable the LVDT 2.0V falling edge interrupt.

LVDTSVIE : LVDTSV interrupt enable bit.
= 1, Enable the LVDTSV falling edge interrupt.
= 0, Disable the LVDTSV falling edge interrupt.

GIE : Global interrupt enable bit.
= 1, Enable all un-masked interrupts.
= 0, Disable all interrupts.

2.1.9 INTFLAG (Interrupt Status Register)

Read/Write-POR		-	R/W-0	-	R/W-0	R/W-0	-	-	R/W-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x0C	INTFLAG	-	PAIF	-	LVDTSVIF	LVDT20IF	-	-	TM0IF

Legend: - = unimplemented, read as '0'.

TM0IF : Timer1 match interrupt flag. Set when TM0 register matches to T0RLD register, reset by software.

LVDT20IF : LVDT2.0V falling edge interrupt flag, Set when LVDT2.0V falling edge, reset by software.

LVDTSVIF : LVDTSV falling edge interrupt flag, Set when LVDTSV falling edge, reset by software.

PAIF : Port A input change interrupt flag. Set when Port A input changes, reset by software.

2.1.10 TM0 (Timer0 Clock/Counter register)

Read/Write-POR		R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x0E	TM0	8-bit real time clock/counter							

The Timer0 is an 8-bit timer/counter. The clock source of Timer0 can come from the F_{CPU}, F_{IRC}, F_{SIRC}, or by an external clock source (T0CK pin) defined by TM0CKS<1:0> bits (TM0CON<2:1>). If T0CK pin is selected, the Timer0 is increased by T0CK signal rising edge.

2.1.11 TM0CON (Timer0 Control Register)

Read/Write-POR		R/W-0	-	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	-
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x0F	TM0CON	TM0EN	-	TM0PS2	TM0PS1	TM0PS0	TM0CKS1	TM0CKS0	-

Legend: - = unimplemented, read as '0'.

TM0CKS1:TM0CKS0 : Timer0 clock source selection bits.

TM0CKS1	TM0CKS0	Timer0 clock source
0	0	F_{CPU}
0	1	F_{IRC} ($16M_{HZ}$)
1	0	F_{SIRC}
1	1	T0CK (External clock, counter is increased on the rising edge)

TM0PS2:TM0PS0 : Timer0 Pre-scaler selection bits.

TM0PS2	TM0PS1	TM0PS0	Timer0 Pre-scaler rate
0	0	0	1:2
0	0	1	1:4
0	1	0	1:8
0	1	1	1:16
1	0	0	1:32
1	0	1	1:64
1	1	0	1:128
1	1	1	1:256

TM0EN : Timer0 Enable/Disable bit.
= 1, Timer0 Enable.
= 0, Timer0 Disable.

2.1.12 T0RLD (Timer0 compare Pre-set Register)

Read/Write-POR		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x10	T0RLD	TM0 Compare Pre-set register							

T0RLD is Timer0 compare value pre-set register, see section 2.3 for detail description on page 29.

2.1.13 WDT (Watch dog Control Register)

Read/Write-POR		R/W-1	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0	-	-
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x11	WDT	WDTE	WDTSL	WDTPS2	WDTPS1	WDTPS0	LVRSL	-	-

Legend: - = unimplemented, read as '0'.

LVRSL : LVR signal select bit.
If device in Green mode :
= 1, LVR signal will wake-up device to normal mode (PC = Next instruction).
= 0, LVR signal will Reset device.
else:
Ignore.

WDTPS2:WDTPS0 : Watchdog timer pre-scaler setting

WDTPS2 : WDTPS0			WDT pre-scaler rate
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

Note: Watchdog period time-base select by Configuration word.

WDTSL : Watchdog wakeup mode selection bit.
= 1, Disable WDT wakeup, when the WDT time-out, device will be reset.
= 0, Enable WDT wakeup, when the WDT time-out, device will be wakeup to normal mode, and execution next instruction

WDTE : Watchdog Timer Enable/ Disable.
= 1, WDT Enable.
= 0, WDT disable.

2.1.14 PCON (Power Control Register)

Read/Write-POR		-	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-#	R-#
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x12	PCON	-	GRN_MD	LVDIS	RSTSL1	RSTSL0	UGMD	LVDTSV	LVDT20

Legend: - = unimplemented, read as '0'.

LVDT20 : LVDT 2.0V detect signal.
= 1, VDD > 2.0V.
= 0, VDD ≤ 2.0V.

LVDTSV : LVDT software selection voltage detection signal.
= 1, VDD > (1.8V, 2.0V, 2.4V or 3.0V).
= 0, VDD ≤ (1.8V, 2.0V, 2.4V or 3.0V).
Note: This detect voltage define by BZS register.

UGMD : Ultra Green mode select bit.
= 1, SIRC is set to 3.2K$_{HZ}$.
= 0, SIRC set-back 32K$_{HZ}$.

RSTSL1:RSTSL0 : Reset function select bits.

RSTSL1 : 0		Reset function description
0	0	POR, PDR, PDRS and WDR signal can be reset device.
0	1	POR, PDR, PDRS, WDR and LVDTSV signal can be reset device.
1	0	POR, PDR, PDRS, WDR and LVDT20 signal can be reset device.
1	1	POR, PDR, PDRS, WDR and RSTB signal can be reset device.

Note : When the POR, PDR, PDRS event occurs, RSTSL<1:0> will be reset to default state. Other event (WDR, LVDTSV, LVDT20V or RSTB) occurs, RSTL<1:0> will remain the last setting.

GRN_MD:LVDIS : Operation mode select bits.

GRM_MD : LVDIS		Operation mode description
0	0	Normal Mode, and All LVDT20V, LVDTSV circuit are enable.
0	1	Slow Mode, and All LVDT20V, LVDTSV circuit are disable. F_{CPU} is enable (F_{CPU} = F_{SIRC}). Note: In this mode, Program must be first Switching F_{CPU} clock to F_{SIRC}.
1	0	Green Mode 0, and All LVDT20V, LVDTSV circuit are enable. F_{CPU} Disable.
1	1	Green Mode 1, and All LVDT20V, LVDTSV circuit are disable. F_{CPU} Disable.

Note : 1. Operation mode select bit recommended using MOVAR instruction to access.
2. Device wake-up from Green mode 0 or Green mode 1, Operation mode select bits will automatically revert to the previous operating mode.
3. See section 2.10 for detail description on page 49.

2.1.15 CLKCFG (Oscillator and Clock Control Register)

Read/Write-POR		R/W-0	R/W-0	R/W-0	R/W-0	-	R/W-1	R/W-0	R/W-1
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x13	CLKCFG	SELCLK2	SELCLK1	SELCLK0	CLKSW	-	INCODS	EXOSEN	IRCEN

Legend: - = unimplemented, read as '0'.

IRCEN : Internal RC Enable bit
= 1, Enable internal RC clock source.
= 0, Disable internal RC clock source (Power down F_{IRC}).
Note: Make sure the system clock (F_{CPU}) been switch to external clock source before power down internal RC.

EXOSEN : Internal clock / External clock Source Selection bit
= 1, The clock source is external clock (T0CK).
= 0, The clock source is internal clock (F_{IRC}).

INCODS : Internal clock output Enable bit.
= 1, Disable internal clock output.
= 0, Enable internal RC clock output, the IRC clock is output to XOUT pin.

CLKSW : System Clock (F_{CPU}) Selection bit
= 1, CPU Clock is F_{SEL}.
= 0, CPU Clock is F_{FIG}.

SELCK2:SELCK0 : Clock divider and SIRC selection bits.

SELCK2 : SELCK0			Description	
			EXOSEN = 0	EXOSEN = 1
0	0	0	F_{IRC}/4	T0CK/4
0	0	1	F_{IRC}/16	T0CK/16
0	1	0	F_{IRC}/8	T0CK/8
0	1	1	F_{IRC}/2	T0CK/2
1	0	0	F_{IRC}/1	T0CK/1
1	0	1	SIRC	
1	1	0	No function, don't use.	
1	1	1		

Note : 1. This control register prohibits direct write value, Need using BSR/BCR instruction.
2. Please refer section 2.12 for detail description on page 55.

2.1.16 PWM0CON (PWM0 Control Register 1)

Read/Write-POR		R/W-0	-	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x14	PWM0CON	P0INTSL	-	LPTS0	ENLP0	P0CKS1	P0CKS0	P0OSEL1	P0OSEL0

Legend: - = unimplemented, read as '0'.

P0OSEL1:P0OSEL0 : PWM0 / Buzzer0 output channel selection bits.

P0OSEL1 : P0OSEL0		PWM0 / Buzzer0 output Pin
0	0	PWM0 / Buzzer0 output on PA3 pin.
0	1	PWM0 / Buzzer0 output on PA0 pin.
1	0	PWM0 / Buzzer0 output on PA4 pin.
1	1	PWM0 / Buzzer0 output on PA5 pin.

Note: In differential mode, PA2 has been fixed for the differential output.

P0CKS1:P0CKS0 : PWM0 / Buzzer0 counter clock source selection bits.

P0CKS1 : P0CKS0		PWM0 / Buzzer0 clock source
0	0	F_{CPU}
0	1	F_{IRC} (16M$_{HZ}$)
1	0	F_{IRC} (16M$_{HZ}$), PA2 is differential output.*
1	1	T0CK (External clock, counter is increased on the rising edge)

Note: P0CKS <1:0> = '10' only for differential output mode.

ENLP0 : Non-overlap for PWM0 differential output.
If P0CKS<1:0> = 10:
= 1, Disable Non-overlap function.
= 0, Enable Non-overlap function.
If P0CKS<1:0> = Other:
Ignore.

LPTS0 : PWM0 Non-overlap Selection bit
If P0CKS<1:0> = 10:
= 1, Non-overlap timing > 125nS.
= 0, Non-overlap timing > 8nS.
If P0CKS<1:0> = Other:
Ignore.

P0INTSL : PWM0 interrupt event selection bit.
= 1, Interrupt at PWM0 counter matches to P0TMDTLB and P0TMDTHB register.
= 0, Interrupt at PWM0 counter matches to P0TMPRLB and P0TMPRHB register.
Note: Set to "1" only for Buzzer / Timer mode. See section 2.4 for detail description on page 29.

2.1.17 PWM0CR (PWM0 Control Register 2)

Read/Write-POR		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x15	PWM0CR	PWM0EN	P0OUTS	P0TPS2	P0TPS1	P0TPS0	P0TMEN	P0TMIE	P0TMIF

P0TMIF : PWM0 interrupt flag, Set when PWM0 compare match, reset by software. See section 2.4 for detail description on page 29.

P0TMIE : PWM0 matches interrupt enable bit.
= 1, Enable PWM0 interrupt.
= 0, Disable PWM0 interrupt.

P0TMEN : PWM0 Enable/Disable bit.
= 1, Enable PWM0.
= 0, Disable PWM0.

P0TPS2:P0TPS0 : PWM0 pre-scaler selection bits.

P0TPS2 : P0TPS0			PWM0 Pre-scaler rate
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

P0OUTS : PWM0 Output invert selection bit.
= 1, PWM0 Duty cycle pulse is low (Invert).
= 0, PWM0 Duty cycle pulse is high (Normal).

PWM0EN : PWM0 Operation mode selection bit.
= 1, PWM mode.
= 0, Timer / Buzzer mode.

2.1.18 P0TMLB and P0TMHB (Low byte and High byte of PWM0 Counter Register)

Read/Write-POR		R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x16	P0TMLB	PWM0 real-time counter low-byte							

Read/Write-POR		R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x18	P0TMHB	PWM0 real-time counter high-byte							

The P0TMLB and P0TMHB are PWM0 real-time counter, these register are only read. See section 2.4 for detail description on page 29.

2.1.19 P0TMDTLB and P0TMDTHB (Low byte and High byte of PWM0 Duty cycle Pre-set Register)

Read/Write-POR		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x17	P0TMDTLB	PWM0 Duty pre-set low-byte register							

Read/Write-POR		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x19	P0TMDTHB	PWM0 Duty pre-set high-byte register							

In PWM mode, these registers are pre-set duty cycle comparison value register, set these registers can be defined PWM0 duty cycle.
In Timer/Buzzer mode, P0TMLB and P0TMHB increase until the value matches to these register, the P0TMLB and P0TMHB will be reset to “0”. See section 2.4 for detail description on page 29.

2.1.20 PWM1CON (PWM1 Control Register 1)

Read/Write-POR		R/W-0	-	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x20	PWM1CON	P1INTSL	-	LPTS1	ENLP1	P1CKS1	P1CKS0	P1OSEL1	P1OSEL0

Legend: - = unimplemented, read as ‘0’.

P1OSEL1:P1OSEL0 : PWM1 / Buzzer1 output channel selection bits.

P1OSEL1 : P1OSEL0		PWM1 / Buzzer1 output Pin
0	0	PWM1 / Buzzer1 output on PA4 pin.
0	1	PWM1 / Buzzer1 output on PA0 pin.
1	0	PWM1 / Buzzer1 output on PA3 pin.
1	1	PWM1 / Buzzer1 output on PA5 pin*.

Note: In differential mode, PA5 has been fixed for the differential output, P1CKS<1:0> = 11 will not be available.

P1CKS1:P1CKS0 : PWM1 / Buzzer1 counter clock source selection bits.

P1CKS1 : P1CKS0		PWM1 / Buzzer1 clock source
0	0	F_{CPU}
0	1	F_{IRC} (16M$_{HZ}$)
1	0	F_{IRC} (16M$_{HZ}$), PA5 is differential output.*
1	1	T0CK (External clock, counter is increased on the rising edge)

Note: P1CKS <1:0> = ‘10’ only for differential output mode.

ENLP1 : Non-overlap for PWM1 differential output.
If P1CKS<1:0> = 10:
= 1, Disable Non-overlap function.
= 0, Enable Non-overlap function.
If P1CKS<1:0> = Other:
Ignore.

LPTS1 : PWM1 Non-overlap Selection bit
If P1CKS<1:0> = 10:
= 1, Non-overlap timing > 125nS.
= 0, Non-overlap timing > 8nS.
If P1CKS<1:0> = Other:
Ignore.

P1INTSL : PWM1 interrupt event selection bit.
= 1, Interrupt at PWM1 counter matches to P1TMDT register.
= 0, Interrupt at PWM1 counter matches to P1TMPR register.
Note: Set to "1" only for Buzzer / Timer mode. See section 2.5 for detail description on page 35.

2.1.21 PWM1CR (PWM1 Control Register 2)

Read/Write-POR		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x21	PWM1CR	PWM1EN	P1OUTS	P1TPS2	P1TPS1	P1TPS0	P1TMEN	P1TMIE	P1TMIF

P1TMIF : PWM1 interrupt flag, Set when PWM1 compare match, reset by software. See section 2.5 for detail description on page 35.

P1TMIE : PWM1 matches interrupt enable bit.
= 1, Enable PWM1 interrupt.
= 0, Disable PWM1 interrupt.

P1TMEN : PWM1 Enable/Disable bit.
= 1, Enable PWM1.
= 0, Disable PWM1.

P1TPS2:P1TPS0 : PWM1 pre-scaler selection bits.

P1TPS2 : P1TPS0			PWM1 Pre-scaler rate
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

P1OUTS : PWM1 Output invert selection bit.
= 1, PWM1 Duty cycle pulse is low (Invert).
= 0, PWM1 Duty cycle pulse is high (Normal).

PWM1EN : PWM1 Operation mode selection bit.
= 1, PWM mode.
= 0, Timer / Buzzer mode.

2.1.22 P1TM (PWM1 Counter Register)

Read/Write-POR		R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x22	P1TM	PWM1 real-time counter							

The P1TM is PWM1 real-time counter, this register is only read. See section 2.5 for detail description on page 35.

2.1.23 P1TMDT (PWM1 Duty cycle Pre-set Register)

Read/Write-POR		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x23	P1TMDT	PWM1 Duty pre-set register							

In PWM mode, this register is pre-set duty cycle comparison value register, set this register can be defined PWM1 duty cycle.
In Timer/Buzzer mode, P1TM increase until the value matches to this register, the P1TM will be reset to “0”. See section 2.5 for detail description on page 35.

2.1.24 BZS (Buzzer Control Register)

Read/Write-POR		R/W-0	R/W-0	R/W-0	R/W-0	-	-	-	-
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x24	BZS	BZS1	BZS0	LVDTS1	LVDTS0	-	-	-	-

Legend: - = unimplemented, read as ‘0’.

LVDTS1:LVDTS0 : LVDT level Programmable bit

LVDTS1	LVDTS0	LVDT level
0	0	$V_{DD} \geq 1.8V$, LVDTSV = 1 $V_{DD} \leq 1.8V$, LVDTSV = 0
0	1	$V_{DD} \geq 2.0V$, LVDTSV = 1 $V_{DD} \leq 2.0V$, LVDTSV = 0
1	0	$V_{DD} \geq 2.4V$, LVDTSV = 1 $V_{DD} \leq 2.4V$, LVDTSV = 0
1	1	$V_{DD} \geq 3.0V$, LVDTSV = 1 $V_{DD} \leq 3.0V$, LVDTSV = 0

BZS1 : Buzzer0 output select bit
= 1, Enable Buzzer output (PWM0EN bit must be “0” (PWM0CR<7>)).
= 0, Disable Buzzer output (Don’t care PWM0EN bit).

BZS1 : Buzzer1 output select bit
= 1, Enable Buzzer output (PWM1EN bit must be “0” (PWM1CR<7>)).
= 0, Disable Buzzer output (Don’t care PWM1EN bit).

2.1.25 ADCON1 (AD Converter Control Register 1)

Read/Write-POR		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x28	ADCON1	ADCEN	ADCST	CHSEL2	CHSEL1	CHSEL0	ADCSR2	ADCSR1	ADCSR0

ADCSR1:ADCSR0 : ADC Conversion clock source select bits.

ADCSR2	ADCSR1	ADCSR0	Conversion clock
0	0	0	$F_{CPU}/8$
0	0	1	$F_{CPU}/4$
0	1	0	$F_{CPU}/2$
0	1	1	$F_{CPU}/1$ (fastest result, lowest quality)
1	0	0	$F_{CPU}/16$
1	0	1	$F_{CPU}/32$
1	1	0	$F_{CPU}/64$
1	1	1	$F_{CPU}/128$ (slowest result, best quality)

Note : 1. This clock is used to control the conversion precision and speed. The precision will be dropped off if faster conversion rate been used. The lowest conversion rate would be recommended in order to acquire most accurate data.

2. **Very important, ADC when converting AIN4 ~ AIN0, the ADC clock should not exceed 4M_{HZ}!! When the Converting AIN5 (1/4 V_{DD}), the ADC clock should not exceed 1M_{HZ}!! See section 2.8.2 for detail description on page 41.**

CHS3:CHSEL0 : ADC input channel select bits.

CHSEL2	CHSEL1	CHSEL0	Input channel
0	0	0	AIN0 (PA0)
0	0	1	AIN1 (PA2)
0	1	0	AIN2 (PA3)
0	1	1	AIN3 (PA4)
1	0	0	AIN4 (PA5)
1	0	1	AIN5 ($V_{DD}/4$)
1	1	0	AIN6 (V_{SS})
1	1	1	No function, do not use.

ADCST : AD Conversion start bit.
If ADCNT bit = 1 :
Ignore, AD Conversion controlled by H/W.
If ADCNT bit = 0 :
= 1, Start AD conversion.
= 0, AD conversion completed.
Note : This bit should be set by software and would be reset by hardware after the ADC end of conversion (ADCNT bit = 0).

ADCEN : ADC module enable bit.
= 1, Enable ADC module.
= 0, Disable ADC module.

2.1.26 ADCON2 (AD Converter Control Register 2)

Read/Write-POR		R/W-0	R/W-0	-	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x29	ADCON2	ADCIE	ADCIF	-	ADCNT	ADCLS	ADCTCK	SELVER1	SELVER0

Legend: - = unimplemented, read as '0'.

SELVER1:SELVER0 : ADC Reference voltage select bits.

SELVER1	SELVER0	Internal V_{REF} voltage
0	0	V_{DD}
0	1	1.5V
1	0	3V
1	1	2V

ADCTCK : ADC Sample rate adjust select bit.
= 1, Sample Rate = AD CLK / 44
= 0, Sample Rate = AD CLK / 34
Note: AD CLK defined by ADCS<2:0>bits (ADCON1<2:0>).

ADCLS : ADC Low sample rate select bit.
= 1, ADC is low sample rate for High solution.
= 0, ADC is normal sample rate.

ADCNT : ADC sample mode select bit.
= 1, ADC continues mode.
= 0, Trigger mode, based on ADCST bit.

ADCIF : ADC Interrupt flag. Set when ADC conversion is completed, reset by software.

ADCIE : ADC conversion completed interrupt enable bit.
= 1, Enable interrupt.
= 0, Disable interrupt.

2.1.27 ADCHB and ADCLB (AD Conversion data high-byte and low-byte Register)

Read/Write-POR		R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x30	ADCHB	ADB9	ADB8	ADB7	ADB6	ADB5	ADB4	ADB3	ADB2

Read/Write-POR		-	-	-	-	-	-	R-0	R-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x31	ADCLB	-	-	-	-	-	-	ADB1	ADB0

Legend: - = unimplemented, read as '0'.

The ADCHB and ADCLB registers is ADC conversion result. When ADC conversion is completed, the result is loaded into ADCHB and ADCLB, the ADCST bit will be cleared, and the ADCIF bit will be set.

2.1.28 ADCON3 (AD Converter Control Register 3)

Read/Write-POR		-	-	-	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x32	ADCON3	-	-	-	INEN4	INEN3	INEN2	INEN1	INEN0

Legend: - = unimplemented, read as '0'.

INEN0 : = 1, PA0 is Analog input (AIN0).
= 0, PA0 is Digital input.

INEN1 : = 1, PA2 is Analog input (AIN1).
= 0, PA2 is Digital input.

INEN2 : = 1, PA3 is Analog input (AIN2).
= 0, PA3 is Digital input.

INEN3 : = 1, PA4 is Analog input (AIN3).
= 0, PA4 is Digital input.

INEN4 : = 1, PA5 is Analog input (AIN4).
= 0, PA5 is Digital input.

2.1.29 CMPCON1 (Comparator Control Register 1)

Read/Write-POR		R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x35	CMPCON1	CMPEN	COP1	CMPIE1	CMPRIF1	CMPFIF1	CMPCS2	CMPCS1	CMPCS0

CMPCS2:CMPCS0 : CN (Comparator Negative) input select bits.

CMPCS2	CMPCS1	CMPCS0	Input Channel
0	0	0	AIN0 (PA0)
0	0	1	AIN1 (PA2)
0	1	0	AIN2 (PA3)
0	1	1	AIN3 (PA4)
1	0	0	AIN4 (PA5)
1	0	1	AIN5 (V_{DD}/4)
1	1	0	AIN6 (V_{SS})
1	1	1	No function, do not use.

CMPFIF1 : Comparator output falling edge interrupt flag. Set when DAC1 voltage < CN (CP<CN), reset by software.

CMPRIF1 : Comparator output rising edge interrupt flag. Set when DAC1 voltage > CN (CP>CN), reset by software.

CMPIE1 : Compare DAC1 interrupt enable bit.
= 1, Enable Compare DAC1 interrupt.
= 0, Disable Compare DAC1 interrupt.

COP1 : Result of compare with DAC1.
= 1, DAC1 voltage > CN.
= 0, DAC1 voltage < CN.

CMPEN : Comparator Enable/Disable bit.
= 1, Enable comparator.
= 0, Disable comparator.

2.1.30 DACR1HB and DACR1LB (DA Conversion data high-byte and low-byte Register 1)

Read/Write-POR		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x36	DACR1HB	DAC1B9	DAC1B8	DAC1B7	DAC1B6	DAC1B5	DAC1B4	DAC1B3	DAC1B2

Read/Write-POR		-	-	-	-	-	-	R/W-0	R/W-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x37	DACR1LB	-	-	-	-	-	-	DAC1B1	DAC1B0

Legend: - = unimplemented, read as '0'.

These register settings DAC1 output voltage. See 2.9.2.2 for detail description on page 45.

2.1.31 CMPCON2 (Comparator Control Register 2)

Read/Write-POR		R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x38	CMPCON2	CMPINT1	COP2	CMPIE2	CMPRIF2	CMPFIF2	CMPINT2	TOGSEL1	TOGSEL0

Legend: - = unimplemented, read as '0'.

TOGSEL1:TOGSEL0 : ADC and CMP toggle operation mode selection bits. See 2.9 for detail description on page 43.

CMPINT2 : Compare DAC2 interrupt event select bit.
= 1, CMPRIF2 or CMPFIF2 set to 1 to generate interrupt (Comparator output change)
= 0, Only in CMPFIF2 is set to 1 interrupt (Comparator output Falling edge).

CMPFIF2 : Comparator output falling edge interrupt flag. Set when DAC2 voltage < CN (CP<CN), reset by software.

CMPRIF2 : Comparator output rising edge interrupt flag. Set when DAC2 voltage > CN (CP>CN), reset by software.

CMPIE2 : Compare DAC2 interrupt enable bit.
= 1, Enable Compare DAC2 interrupt.
= 0, Disable Compare DAC2 interrupt.

COP2 : Result of compare with DAC2.
= 1, DAC2 voltage > CN.
= 0, DAC2 voltage < CN.

CMPINT1 : Compare DAC1 interrupt event select bit.
= 1, CMPRIF1 or CMPFIF1 set to 1 to generate interrupt (Comparator output change)
= 0, Only in CMPFIF1 is set to 1 interrupt (Comparator output Falling edge).

2.1.32 DACR2HB and DACR2LB (DA Conversion data high-byte and low-byte Register 2)

Read/Write-POR		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-	R/W-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x39	DACR2HB	DAC2B9	DAC2B8	DAC2B7	DAC2B6	DAC2B5	DAC2B4	DAC2B3	DAC2B2

Read/Write-POR		-	-	-	-	-	-	R/W-0	R/W-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x3A	DACR2LB	-	-	-	-	-	-	DAC2B1	DAC2B0

Legend: - = unimplemented, read as '0'.

These register settings DAC2 output voltage. See 2.9.2.2 for detail description on page 45.

2.1.33 ACC (Accumulator)

Read/Write-POR		R/W-x							
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
N/A	ACC	Accumulator							

Accumulator is an internal data transfer, or instruction operand holding. It cannot be addressed.

2.1.34 PAMODE0 and PAMODE1 (Port A mode Control Register 0 & 1)

Read/Write-POR		-	-	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x08	PAMODE0	-	-	PAMD05	PAMD04	PAMD03	PAMD02	PAMD01	PAMD00

Read/Write-POR		-	-	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x09	PAMODE1	-	-	PAMD15	PAMD14	PAMD13	PAMD12	PAMD11	PAMD10

Legend: - = unimplemented, read as '0'.
These registers are accessed by the IOST / IOSTR instruction.

PAMODE0 and PAMODE1 is control PORTA pins input / output / driver / sink / pull-up function, each bit in these two registers will control the corresponding PORTA pin, functions is defined as follows:

Data register (PAx)	MODE1	MODE0	Output Drive Strength
1	0	0	HI-z(Input mode)
0			
1	0	1	Pull-High (input mode)
0			
1	1	0	Pull-High (input mode)
0			Sink (output mode)
1	1	1	Normal Drive
0			Sink

2.1.35 P0TMPRLB and P0TMPRHB (Low byte and High byte of PWM0 Period cycle Pre-set Register)

Read/Write-POR		R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x17	P0TMPRLB	PWM0 Period cycle pre-set low-byte register							

Read/Write-POR		R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x19	P0TMPRHB	PWM0 Period cycle pre-set high-byte register							

These registers are accessed by the IOST / IOSTR instruction.

In PWM mode, these registers are pre-set period cycle comparison value register, set these registers can be defined PWM0 period cycle. In Timer/Buzzer mode, these registers are not used, user must be set all bits to “1”. See section 2.4 for detail description on page 29.

2.1.36 P1TMPR (PWM1 Period cycle Pre-set Register)

Read/Write-POR		R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x23	P1TMPR	PWM1 Period cycle pre-set register							

Accessed by IOST / IOSTR instruction.

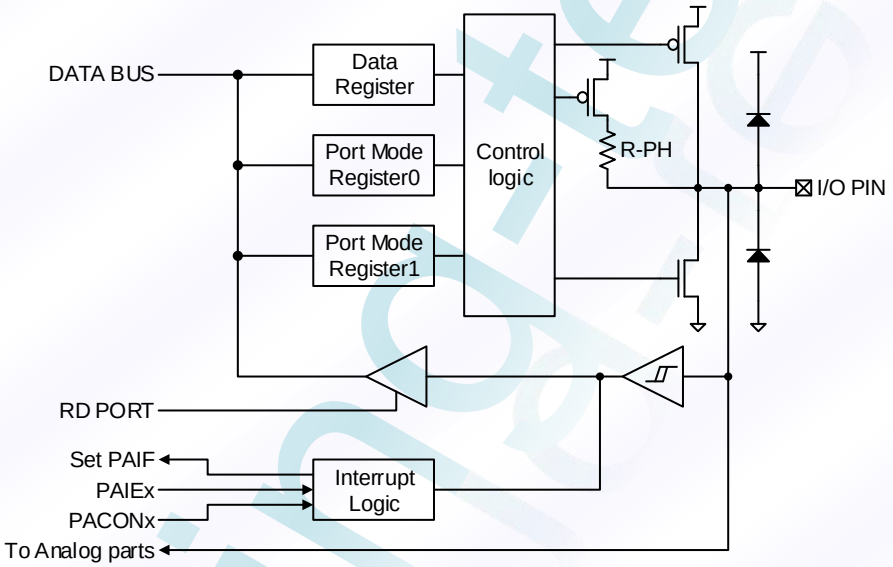
In PWM mode, these registers are pre-set period cycle comparison value register, set these registers can be defined PWM1 period cycle. In Timer/Buzzer mode, these registers are not used, user must be set all bits to “1”. See section 2.5 for detail description on page 35.

2.2 I/O Ports

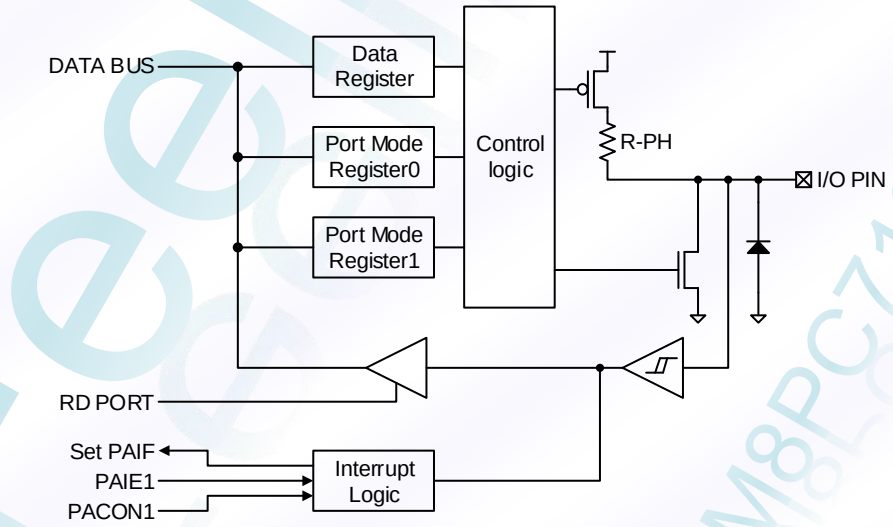
Port A is 6-pin bi-directional tristate I/O ports. Please note that PA1 is an input or open-drain output pin.
All I/O pins have corresponding function control registers (PAMODE0, PAMODE1) which can configure these pins as input, output, pull-high and open-drain with pull-high.
PORTA<5:0> also provides the input change interrupt/wake-up function. Each pin has its corresponding input change interrupt/wake-up enable bits (PAIE) to select the input change interrupt/wake-up source. The input change or falling edge function can be selected by PACONx bit (PACON register).
The control register ADCON3 can set several I/Os to analog function. When acting as analog function the pins will read as "0".
Please note, PA1 voltage on this pin must not exceed V_{DD}, otherwise it will cause the pin breakdown!!

Figure 2.3: Block Diagram of I/O Pins

PA5 ~ PA2, PA0:



PA1:

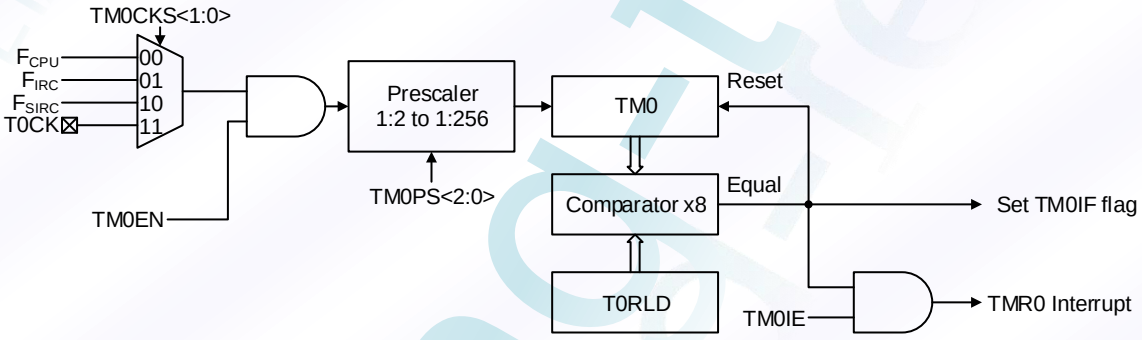


Voltage on this pin must not exceed V_{DD}.

2.3 Timer0

The Timer0 is an 8-bit clock counter with a programmable pre-scaler and an 8-bit compare pre-set register (T0RLD). The clock source of Timer0 comes from the F_{CPU}, F_{IRC}, F_{SIRC} or by an external clock source (T0CK pin) defined by TM0CKS<1:0> bits (TM0CON<2:1>).
The option of Timer0 pre-scaler (1:2 to 1:256) is defined by TM0PS2:TM0PS0 (TM0CON<5:3>) bits. **The pre-scaler is not be cleared when a value is written to TM0CON register.**
The timer increments from 0x00 until it equals the compare pre-set register (T0RLD). It then resets to 0x00 at the next increment cycle. The timer interrupt flag (TM0IF, INTFLAG<0>) is set when the timer rollover to 0x00. **Please note, do not written 0x00 to compare pre-set register (T0RLD). If written, timer will unable to count.**
The timer also has a corresponding interrupt enable bit (TM0IE, INTEN<0>). The timer interrupt can be enabled/disabled by setting/clearing this bit.
The timer is can be turned on and off under software control. When the timer on control bit (TM0EN, TM0CON<7>) is set, the timer is re-initial and increments from the clock source rising edge. When TM0EN is cleared, the timer is turned off.

Figure 2.4: Block Diagram of the Timer0



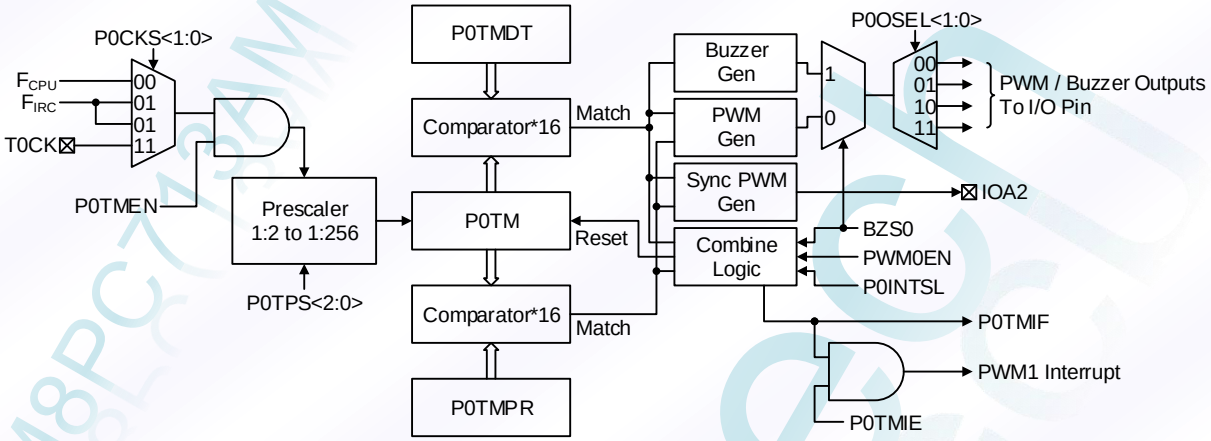
2.4 PWM0 / Buzzer0 / Timer1

The PWM0 module is a 16-bit clock counter with a programmable pre-scaler, and dual 16-bit compare pre-set register. The clock source comes from the F_{CPU}, F_{IRC} or by an external clock source (T0CK pin) defined by P0CKS<1:0> bits (PWM0CON<3:2>).
The option of PWM0 pre-scaler (1:2 to 1:256) is defined by P0TPS<2:0> (PWM0CR<5:3>) bits. **The pre-scaler is not be cleared when a value is written to PWM0CR register.**
The PWM0 / Buzzer0 output pin can be programmable to PA0, PA3, PA4 or PA5 defined by P0OSEL<1:0> (PWM0CR<5:3>) bits. When I/O is programmed as PWM0 / Buzzer0 output, the corresponding I/O must be set to output mode by program.
The PWM0 module also has a corresponding interrupt enable bit (P0TMIE, PWM0CR<2>). The PWM module interrupt can be enabled/disabled by setting/clearing this bit.
The PWM0 module is can be turned on and off under software control. When the PWM0 enable control bit (P0TMEN, PWM0CR<2>) is set, the PWM0 module is re-initial and increments from the clock source rising edge. When P0TMEN is cleared, the PWM0 module is turned off.
The PWM0 module has three modes of operation; PWM, Buzzer and timer mode, detailed description as follows:

Table 2.1: PWM0 and PWM1 module operating mode

BZSx	PWMxEN	PxINTSL	PWMx module Operating mode
0	0	1	Timer mode
0	1	x	PWM mode
1	0	1	Buzzer mode
Other			Do not select

Figure 2.5: Block Diagram of the PWM0



2.4.1 Normal PWM mode

In this mode, PWM0 duty-cycle defined by P0TMDTLB and P0TMDTHB register, Period-cycle defined by P0TMPRLB and P0TMPRHB. Meanwhile, BZS0, PWM0EN and P0INTSL bits must be set to the state specified of Table 2.1. **Please note, do not written 0x0000 to compare pre-set register (P0TMPRLB and P0TMPRHB). If written, PWM0 counter will unable to count.**

When changing PWM0 Period-cycle or Duty-cycle, the next cycle will load the new settings.

The PWM0 period-cycle time can be calculated as follows:

$$\text{Period-cycle time of PWM0} = \frac{(\text{P0TMPR}+1) * \text{PWM0 Pre-scaler rate}}{\text{PWM0 Clock source frequency}}$$

or

$$\text{P0TMPR} = \frac{\text{Period-cycle time} * \text{PWM0 Clock source frequency}}{\text{PWM0 Pre-scaler rate}} - 1$$

The PWM0 duty-cycle time can be calculated as follows:

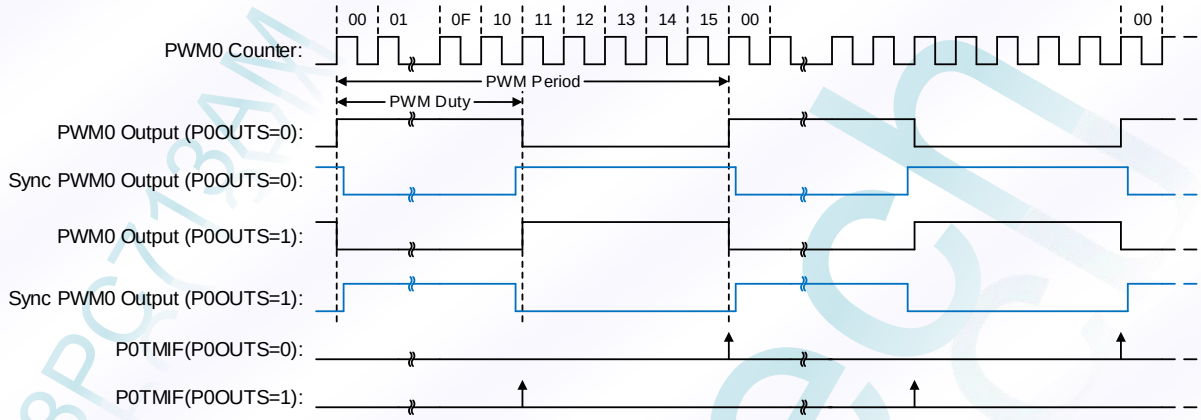
$$\text{Duty-cycle time of PWM0} = \frac{\text{P0TMDT} * \text{PWM0 Pre-scaler rate}}{\text{PWM0 Clock source frequency}}$$

or

$$\text{P0TMDT} = \frac{\text{Duty-cycle time} * \text{PWM0 Clock source frequency}}{\text{PWM0 Pre-scaler rate}}$$

Please note: The PWM duty-cycle time must be less than PWM period-cycle time.

Figure 2.6: PWM0 Output and Interrupt event Waveform



Example 2.2: PWM0 Setting (BZS0=0, PWM0EN=1, P0INTSL=x)

ASM Language Code

```
#include   <8PC713AM.ASH>
           …
           MOVIA    0x05          ; PWM interrupt occur at end of period-cycle
                                  ; (Occur at end of duty-cycle is 0x85)
           MOVAR    PWM0CON       ; Clock source=FIRC 16MHZ, Output Pin = PA0

           MOVIA    0xA2          ; PWM mode, No-Invert, Pre-scaler=1:16,
           MOVAR    PWM0CR        ; Enable interrupt (Non-interrupt is 0xA0)

           MOVIA    0x10
           MOVAR    P0TMDTLB
           MOVIA    0x00          ; Set Duty cycle Pre-set=0x0010
           MOVAR    P0TMDTHB      ; Duty time = 0x0010*16*1/16MHZ = 16uS

           MOVIA    0x15
           MOVAR    P0TMPRLB
           MOVIA    0x00          ; Set Period cycle Pre-set=0x0015
           MOVAR    P0TMPRHB      ; Duty time = (0x0015+1)*16*1/16MHZ = 21uS

           BSR      PWM0CR,P0TMEN_B ; Start PWM0
           //Interrupt setting, not required
           BSR      INTEN,GIE_B   ; Enable Global interrupt
           …
```

C Language Code

```
#include   <8PC713AM.h>
           …
                                  // PWM interrupt occur at end of period-cycle
                                  // (Occur at end of duty-cycle is 0x85)
           PWM0CON=0x05;          // Clock source=FIRC 16MHZ, Output Pin = PA0

           PWM0CR=0xA2;           // PWM mode, No-Invert, Pre-scaler=1:16,
                                  // Enable interrupt (Non-interrupt is 0xA0)

           P0TMDTLB=0x10;         // Set Duty cycle Pre-set=0x0010
           P0TMDTHB=0x00;         // Duty time = 0x0010*16*1/16MHZ) = 16uS

           P0TMPRLB=0x15;         // Set Period cycle Pre-set=0x0015
           P0TMPRHB=0x00;         // Duty time = (0x0015+1)*16*1/16MHZ) = 21uS
```

C Language Code (Continue)

```
PWM0CRbits.P0TMEN=1;                    // Start PWM0
//Interrupt setting, not required
INTENbits.GIE=1;                        // Enable Global interrupt
…
```

2.4.2 Synchronous PWM mode

In this mode, PWM0 differential output pin fixed at PA2. This mode can be enable by setting the Clock source select bits P0CKS<1:0> of the PWM0CON register.
Non-overlapping time between the differential output has >125nS and >8nS can be selected by the LPTS0 (PWM0CON<5>) bit.
The Non-overlapping time also has a corresponding enable bit (ENLP0, PWM0CON<4>). The Non-overlapping time function can be enabled/disabled by clearing/setting this bit.
Other details of the operation, same as the normal mode, see the previous section explained.

2.4.3 Normal Buzzer mode

In this mode, Buzzer0 duty-cycle defined by P0TMDTLB and P0TMDTHB register, Period-cycle time is fixed double duty-cycle time. Therefore, P0TMPRLB and P0TMPRHB register must be fixed at 0xFFFF. Meanwhile, BZS0, PWM0EN and P0INTSL bits must be set to the state specified of Table 2.1. **Please note, do not written 0x0000 to compare pre-set register (P0TMDTLB and P0TMDTHB). If written, PWM0 counter will unable to count.**
When changing Buzzer0 Duty-cycle, the next cycle will load the new settings.
The Buzzer0 duty-cycle time can be calculated as follows:

$$\text{Duty-cycle time of Buzzer0} = \frac{(\text{P0TMDT}+1) * \text{PWM0 Pre-scaler rate}}{\text{PWM0 Clock source frequency}}$$

or

$$\text{P0TMDT} = \frac{\text{Duty-cycle time} * \text{PWM0 Clock source frequency}}{\text{PWM0 Pre-scaler rate}} - 1$$

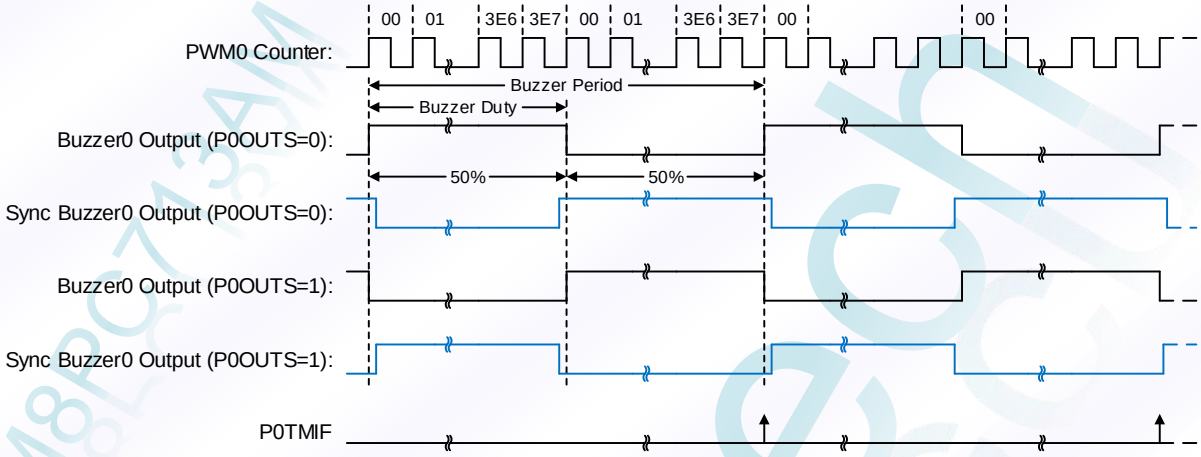
For example, 500_{HZ} Buzzer calculated as follows:

$$\text{Period-cycle of Buzzer0} = \frac{1}{500_{HZ}} = 2\text{mS}, \quad \text{Duty-cycle of Buzzer0} = \frac{2\text{mS}}{2} = 1\text{mS}$$

Given operating conditions: Clock source from F_{IRC} $16M_{HZ}$, pre-scaler rate is 1:16

$$\text{P0TMDT} = \frac{\text{Duty-cycle time} * \text{PWM0 Clock source frequency}}{\text{PWM0 Pre-scaler rate}} - 1 \Rightarrow \frac{1\text{mS} * 16M_{HZ}}{16} - 1 = 999 = 0\text{x}3\text{E}7$$

Figure 2.7: Buzzer0 Output and Interrupt event Waveform



Example 2.3: 500$_{HZ}$ Buzzer0 Setting (BZS0=1, PWM0EN=0, P0INTSL=1)

ASM Language Code

```
#include   <8PC713AM.ASH>
           …
           MOVIA    0x85          ; In Timer/Buzzer mode, P0INTSL must be set to 1
           MOVAR    PWM0CON       ; Clock source=F_IRC 16M_HZ, Output Pin = PA0

           MOVIA    0x22          ; Timer/Buzzer mode, No-Invert, Pre-scaler=1:16,
           MOVAR    PWM0CR        ; Enable interrupt (Non-interrupt is 0x20)

           MOVIA    0xE7
           MOVAR    P0TMDTLB      : Set Duty cycle Pre-set=0x03E7
           MOVIA    0x03          ; Duty time = (0x3E7+1)*16*1/16M_HZ) = 1mS
           MOVAR    P0TMDTHB      ; Buzzer Period time = Duty time *2 = 2mS

           BSR      BZS,BZS0_B    ; Configure to Buzzer mode
           BSR      PWM0CR,P0TMEN_B ; Start Buzzer0
           //Interrupt setting, not required
           BSR      INTEN,GIE_B   ; Enable Global interrupt
           …
```

C Language Code

```
#include   <8PC713AM.h>
           …
                                  // In Timer/Buzzer mode, P0INTSL must be set to 1
           PWM0CON=0x85;          // Clock source=F_IRC 16M_HZ, Output Pin = PA0

           PWM0CR=0x22;           // Timer/Buzzer mode, No-Invert, Pre-scaler=1:16,
                                  // Enable interrupt (Non-interrupt is 0x20)

           P0TMDTLB=0xE7;         // Set Duty cycle Pre-set=0x03E7
           P0TMDTHB=0x03;         // Duty time = (0x3E7+1)*16*1/16M_HZ) = 1mS
                                  // Buzzer Period time = Duty time *2 = 2mS

           BZSbits.BZS0=1;        // Configure to Buzzer mode
           PWM0CRbits.P0TMEN=1;   // Start Buzzer0
           //Interrupt setting, not required
           INTENbits.GIE=1;       // Enable Global interrupt
           …
```

2.4.4 Synchronous Buzzer mode

In this mode, Buzzer0 differential output pin fixed at PA2. This mode can be enable by setting the Clock source select bits P0CKS<1:0> of the PWM0CON register.
Non-overlapping time between the differential output has >125nS and >8nS can be selected by the LPTS0 (PWM0CON<5>) bit.
The Non-overlapping time also has a corresponding enable bit (ENLP0, PWM0CON<4>). The Non-overlapping time function can be enabled/disabled by clearing/setting this bit.
Other details of the operation, same as the normal mode, see the previous section explained.

2.4.5 Timer mode

In this mode, Timer1 Period count-time defined by P0TMDTLB and P0TMDTHB register, P0TMPRLB and P0TMPRHB register must be fixed at 0xFFFF. Meanwhile, BZS0, PWM0EN and P0INTSL bits must be set to the state specified of Table 2.1 on page 29. **Please note, do not written 0x0000 to compare pre-set register (P0TMDTLB and P0TMDTHB). If written, PWM0 counter will unable to count.**
The Timer1 count-time can be calculated as follows:

$$\text{Count-time of Timer1} = \frac{(\text{P0TMDT}+1) * \text{PWM0 Pre-scaler rate}}{\text{PWM0 Clock source frequency}}$$

or

$$\text{P0TMDT} = \frac{\text{Count-time} * \text{PWM0 Clock source frequency}}{\text{PWM0 Pre-scaler rate}} - 1$$

Example 2.4: 1mS Count-time of Timer1 Setting (BZS0=0, PWM0EN=0, P0INTSL=1)

ASM Language Code

```
#include    <8PC713AM.ASH>
            …
            MOVIA     0x85          ; In Timer/Buzzer mode, P0INTSL must be set to 1
            MOVAR     PWM0CON       ; Clock source=FIRC 16MHZ

            MOVIA     0x22          ; Timer mode, Pre-scaler=1:16,
            MOVAR     PWM0CR        ; Enable interrupt (Non-interrupt is 0x20)

            MOVIA     0xE7
            MOVAR     P0TMDTLB
            MOVIA     0x03          ; Set Count Pre-set=0x03E7
            MOVAR     P0TMDTHB      ; Count time = (0x3E7+1)*16*1/16MHZ) = 1mS

            BSR       PWM0CR,P0TMEN_B ; Start Timer1
            //Interrupt setting, not required
            BSR       INTEN,GIE_B   ; Enable Global interrupt
            …
```

C Language Code

```
#include    <8PC713AM.h>
            …
                                    // In Timer/Buzzer mode, P0INTSL must be set to 1
            PWM0CON=0x85;           // Clock source=FIRC 16MHZ

            PWM0CR=0x22;            // Timer mode, Pre-scaler=1:16,
                                    // Enable interrupt (Non-interrupt is 0x20)

            P0TMDTLB=0xE7;          // Set Count Pre-set=0x03E7
            P0TMDTHB=0x03;          // Count time = (0x3E7+1)*16*1/16MHZ) = 1mS

            PWM0CRbits.P0TMEN=1;    // Start Timer1
            //Interrupt setting, not required
            INTENbits.GIE=1;        // Enable Global interrupt
            …
```

2.5 PWM1 / Buzzer1 / Timer2

The PWM1 module is a 8-bit clock counter with a programmable pre-scaler, and dual 8-bit compare pre-set register. The clock source comes from the F_{CPU}, F_{IRC} or by an external clock source (T0CK pin) defined by P1CKS<1:0> bits (PWM1CON<3:2>).

The option of PWM1 pre-scaler (1:2 to 1:256) is defined by P1TPS<2:0> (PWM1CR<5:3>) bits. **The pre-scaler is not be cleared when a value is written to PWM1CR register.**

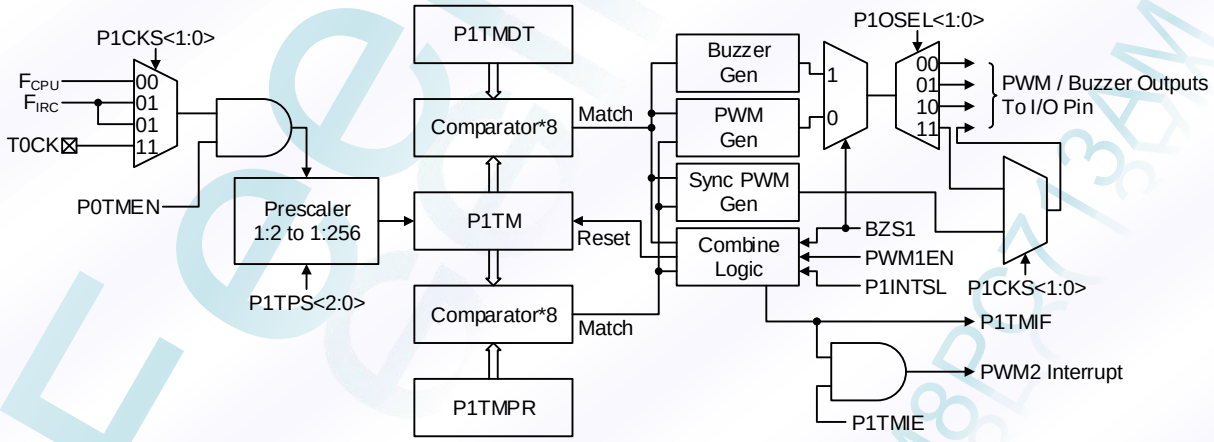
The PWM1 / Buzzer1 output pin can be programmable to PA0, PA3, PA4 or PA5 defined by P2OSEL<1:0> (PWM1CR<5:3>) bits. When I/O is programmed as PWM1 / Buzzer1 output, the corresponding I/O must be set to output mode by program.

The PWM1 module also has a corresponding interrupt enable bit (P1TMIE, PWM1CR<2>). The PWM module interrupt can be enabled/disabled by setting/clearing this bit.

The PWM1 module is can be turned on and off under software control. When the PWM1 enable control bit (P1TMEN, PWM1CR<2>) is set, the PWM1 module is re-initial and increments from the clock source rising edge. When P1TMEN is cleared, the PWM1 module is turned off.

The PWM1 module has three modes of operation; PWM, Buzzer and timer mode. Wherein, PWM and Buzzer mode with a synchronous output, detailed description as follows:

Figure 2.8: Block Diagram of the PWM1



2.5.1 Normal PWM mode

In this mode, PWM1 duty-cycle defined by P1TMDT register, Period-cycle defined by P1TMPR. Meanwhile, BZS1, PWM1EN and P1INTSL bits must be set to the state specified of Table 2.1 on page 29. **Please note, do not written 0x00 to compare pre-set register (P1TMPR). If written, PWM1 counter will unable to count.**
When changing PWM1 Period-cycle or Duty-cycle, the next cycle will load the new settings.
The PWM1 period-cycle time can be calculated as follows:

$$\text{Period-cycle time of PWM1} = \frac{(\text{P1TMPR}+1) * \text{PWM1 Pre-scaler rate}}{\text{PWM1 Clock source frequency}}$$

or

$$\text{P1TMPR} = \frac{\text{Period-cycle time} * \text{PWM1 Clock source frequency}}{\text{PWM1 Pre-scaler rate}} - 1$$

The PWM1 duty-cycle time can be calculated as follows:

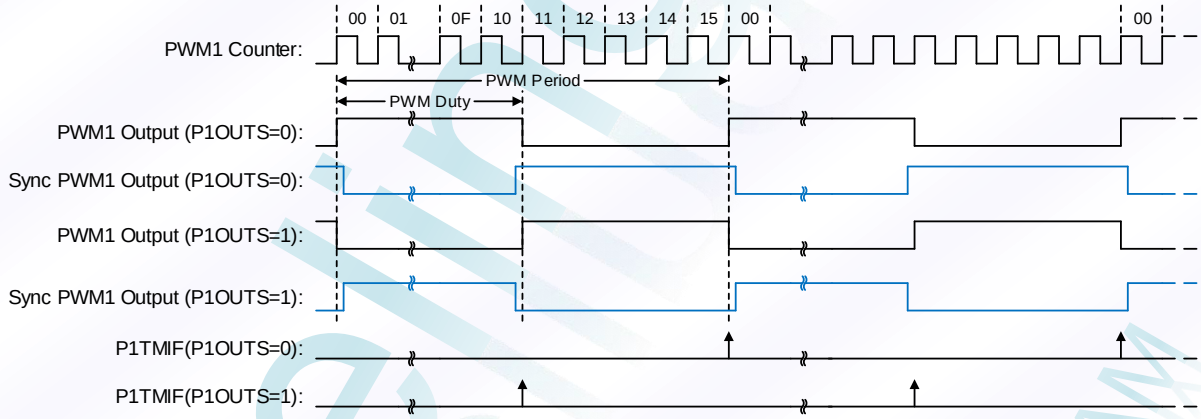
$$\text{Duty-cycle time of PWM1} = \frac{\text{P1TMDT} * \text{PWM1 Pre-scaler rate}}{\text{PWM1 Clock source frequency}}$$

or

$$\text{P1TMDT} = \frac{\text{Duty-cycle time} * \text{PWM1 Clock source frequency}}{\text{PWM1 Pre-scaler rate}}$$

Please note: The PWM duty-cycle time must be less than PWM period-cycle time.

Figure 2.9: PWM1 Output and Interrupt event Waveform



2.5.2 Synchronous PWM mode

In this mode, PWM1 differential output pin fixed at PA5. At PA5 Normal PWM output function will be forced to disable.
This mode can be enable by setting the Clock source select bits P1CKS<1:0> of the PWM1CON register.
Non-overlapping time between the differential output has >125nS and >8nS can be selected by the LPTS1 (PWM1CON<5>) bit.
The Non-overlapping time also has a corresponding enable bit (ENLP1, PWM1CON<4>). The Non-overlapping time function can be enabled/disabled by clearing/setting this bit.
Other details of the operation, same as the normal mode, see the previous section explained.

2.5.3 Normal Buzzer mode

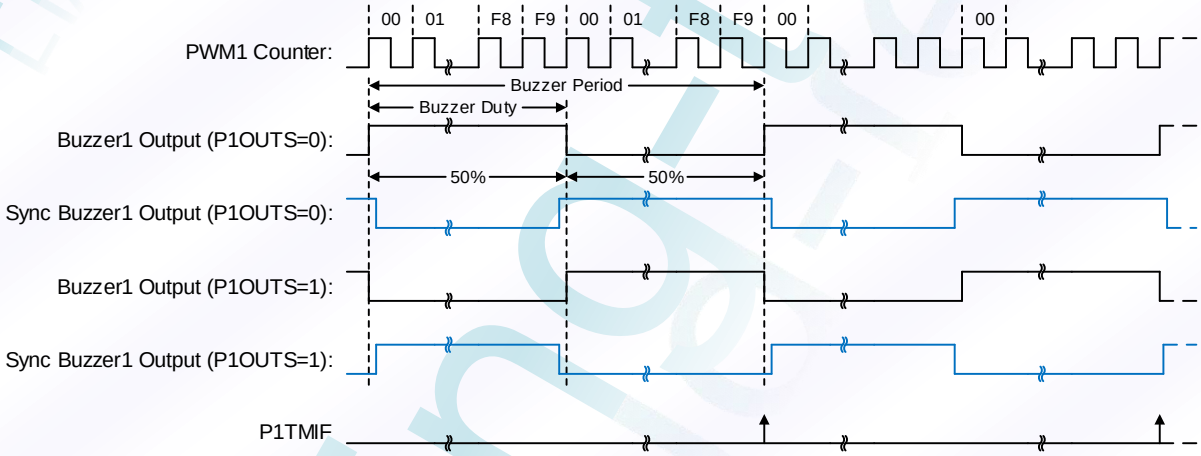
In this mode, Buzzer1 duty-cycle defined by P1TMDT register, Period-cycle time is fixed double duty-cycle time. Therefore, P1TMPR register must be fixed at 0xFF. Meanwhile, BZS1, PWM1EN and P1INTSL bits must be set to the state specified of Table 2.1 on page 29. **Please note, do not written 0x00 to compare pre-set register (P1TMDT). If written, PWM0 counter will unable to count.**
When changing Buzzer1 Duty-cycle, the next cycle will load the new settings.
The Buzzer1 duty-cycle time can be calculated as follows:

$$\text{Duty-cycle time of Buzzer1} = \frac{(\text{P1TMDT}+1) * \text{PWM1 Pre-scaler rate}}{\text{PWM1 Clock source frequency}}$$

or

$$\text{P1TMDT} = \frac{\text{Duty-cycle time} * \text{PWM1 Clock source frequency}}{\text{PWM1 Pre-scaler rate}} - 1$$

Figure 2.10: Buzzer1 Output and Interrupt event Waveform



2.5.4 Synchronous Buzzer mode

In this mode, Buzzer1 differential output pin fixed at PA5. At PA5 Normal Buzzer output function will be forced to disable. This mode can be enable by setting the Clock source select bits P1CKS<1:0> of the PWM1CON register.
Non-overlapping time between the differential output has >125nS and >8nS can be selected by the LPTS1 (PWM1CON<5>) bit.
The Non-overlapping time also has a corresponding enable bit (ENLP1, PWM1CON<4>). The Non-overlapping time function can be enabled/disabled by clearing/setting this bit.
Other details of the operation, same as the normal mode, see the previous section explained.

2.5.5 Timer mode

In this mode, Timer2 Period count-time defined by P1TMDT register, P1TMPR register must be fixed at 0xFF. Meanwhile, BZS1, PWM1EN and P1INTSL bits must be set to the state specified of Table 2.1 on page 29. **Please note, do not written 0x00 to compare pre-set register (P1TMDT). If written, PWM1 counter will unable to count.**

The Timer2 count-time can be calculated as follows:

$$\text{Count-time of Timer2} = \frac{(\text{P1TMDT}+1) * \text{PWM1 Pre-scaler rate}}{\text{PWM1 Clock source frequency}}$$

or

$$\text{P1TMDT} = \frac{\text{Count-time} * \text{PWM1 Clock source frequency}}{\text{PWM1 Pre-scaler rate}} - 1$$

2.6 WatchDog

The Watchdog Timer (WDT) is a free running on-chip RC oscillator which does not require any external components. So the WDT will still run even if the clock on the T0CK pin is turned off, such as in Green mode. Please note, in the Suspend mode, the watchdog will be forced to disable.

The WDT can be disabled by clearing the control bit WDTE (WDT <7>).

Typical WDT timeout can be selected by configuration word (without pre-scaler). If a longer time-out period is desired, a pre-scaler with a division ratio of up to 1:128 can be assigned to the WDT controlled by the WDT register <2:0>. Thus, the longest time-out period is approximately 65.536 seconds (WDTSEL=512mS, pre-scaler=1:128).

The CLRWDT instruction clears the WDT and prevents it from timing out and generating a device reset.

The watchdog has two different modes, reset the device or execution of the next instruction mode, this function can be selected by WDTSL bit (WDT<6>). If using Watchdog wake-up function, need executing CLRWDT instruction before entering the Green mode.

2.7 Interrupts

The FM8PC713AM has up to six sources of interrupt:

1. TM0 overflow interrupt.
2. Port A external interrupt (pins PA5:PA0).
3. PWM0/PWM1 (Buzzer0/Buzzer1) interrupt.
4. AD conversion completion interrupt.
5. Comparator interrupt.
6. Fixed 2.0V LVDT & Programmable voltage LVDT interrupt.

INTFLAG is the interrupt flag register that recodes the interrupt requests in the relative flags.

A global interrupt enable bit, GIE (INTEN<7>), enables (if set) all un-masked interrupts or disables (if cleared) all interrupts. Individual interrupts can be enabled/disabled through their corresponding enable bits in INTEN register regardless of the status of the GIE bit.

When an interrupt event occur with the GIE bit and its corresponding interrupt enable bit are all set, the GIE bit will be cleared by hardware to disable any further interrupts, and the next instruction will be fetched from address 0x003. The interrupt flag bits must be cleared by software before re-enabling GIE bit to avoid recursive interrupts.

The RETFIE instruction exits the interrupt routine and set the GIE bit to re-enable interrupt.

2.7.1 Timer0 Interrupt

A match condition (TM0 = T0RLD) in the TM0 register will set the flag bit TM0IF (INTFLAG<0>). And TM0IF bit can be cleared by software. This interrupt can be disabled by clearing TM0IE bit (INTEN<0>).

2.7.2 Port A external Interrupt

An input change or falling edge on PORTA<5:0> set flag bit PAIF (INTFLAG<6>). And PAIF bit can be cleared by software. This interrupt can be disabled by clearing PAIE register.
Any pin which corresponding control bit (PAIE<5:0>) is cleared to "0" will be excluded from this function. Please note that even if the I/O configured as an output mode, if they meet the interrupt condition, will still generate an interrupt.
Each pin has a corresponding bit (PACON<5:0>) of the interrupt event selection, can be selected as the input change interrupt or falling edge interrupt.
The Port A interrupt also can wake-up the system from suspend or green mode condition, if bit PAIE was set before going to Suspend or green mode. And GIE bit also decides whether or not the processor branches to the interrupt vector following wake-up. If GIE bit was set, the program will execute interrupt service routine after wake-up; or if GIE bit was cleared, the program will execute next PC after wake-up.

2.7.3 PWM0/PWM1 Interrupt

2.7.3.1 PWM0 interrupt

In the Timer/Buzzer mode, a match condition (P0TM=P0TMDT) in the P0TMLB and P0TMHB registers will set the flag bit P0TMIF (PWM0CR<0>). And P0TMIF bit can be cleared by software. This interrupt can be disabled by clearing P0TMIE bit (PWM0CR<1>).
In the PWM mode, a match condition (P0TM=P0TMDT, end of duty-cycle) or (P0TM=P0TMPR, end of period-cycle) in the P0TMLB and P0TMHB registers will set the flag bit P0TMIF (PWM0CR<0>), this interrupt event can be selected P0INTSL bit (PWM0CON<7>). And P0TMIF bit can be cleared by software. This interrupt can be disabled by clearing P0TMIE bit (PWM0CR<1>).

2.7.3.2 PWM1 interrupt

In the Timer/Buzzer mode, a match condition (P1TM=P1TMDT) in the P1TM register will set the flag bit P1TMIF (PWM1CR<0>). And P1TMIF bit can be cleared by software. This interrupt can be disabled by clearing P1TMIE bit (PWM1CR<1>).
In the PWM mode, a match condition (P1TM=P1TMDT, end of duty-cycle) or (P1TM=P1TMPR, end of period-cycle) in the P1TM register will set the flag bit P1TMIF (PWM1CR<0>), this interrupt event can be selected P1INTSL bit (PWM1CON<7>). And P1TMIF bit can be cleared by software. This interrupt can be disabled by clearing P1TMIE bit (PWM1CR<1>).

2.7.4 AD Conversion completion interrupt

When the A/D conversion is completed, the flag bit ADCIF (ADCON2<6>) will be set. And the ADCIF bit can be cleared by software. This interrupt can be disabled by clearing ADCIE bit (ADCON2<7>).

2.7.5 Comparator interrupt

When the comparator output falling edge occur, flag bit CMPFIF1 (CMPCON1<3>) will be set. When the comparator output rising edge occur, flag bit CMPRIF1 (CMPCON1<4>) will be set. These CMPFIF1 and CMPRIF1 bits can be cleared by software. Comparator interrupt event can be selected by CMPINT1 (CMPCON2<7>) bit. These interrupt can be disable by clearing CMPIE1 (CMPCON1<5>) bit.
Similarly, CMPFIF2 (CMPCON2<3>), CMPRIF2 (CMPCON2<4>), CMPINT2 (CMPCON2<2>) and CMPIE2 (CMPCON2<5>) operation same as CMPFIF1, CMPRIF1, CMPINT1 and CMPIE1.

2.7.6 Fixed 2.0V LVDT & Programmable voltage LVDT interrupt.

2.7.6.1 Fixed 2.0V LVDT interrupt

When the V_{DD} voltage drops below the 2.0V, flag bit LVDT20IF (INTFLAG<4>) will be set. And the LVDT20IF bit can be cleared by software. This interrupt can be disable by clearing LVDT20IE (INTEN<4>) bit.

2.7.6.2 Programmable voltage LVDT interrupt

When the V_{DD} voltage drops below programmed voltage (Selection by LVDTS<1:0> bits (BZS<5:4>)), flag bit LVDTSVIF (INTFLAG<3>) will be set. And the LVDTSVIF bit can be cleared by software. This interrupt can be disable by clearing LVDTSVIE (INTEN<3>) bit.

2.8 Analog to Digital Converter (ADC)

The analog to digital converter (ADC) with 7-input sources and allows conversion of an analog input signal to a 10-bit binary representation of that signal. Input source can be selected by CHSEL<2:0> (ADCON1<5:3>) to select analog signal input pin (AIN pin), internal V_{DD}/4 and V_{SS} voltage source. The converter generates a 10-bit binary result via successive approximation and stores the conversion result into the ADC result registers (ADCLB and ADCHB).

ADC has continue mode can be selected, if ADCNT (ADCON2<5>) bit is set, after the completion of a conversion cycle, it will automatically continue to convert next analog signals. If ADCLS (ADCON2<4>) bit is set, conversion will be smoother.

The ADC module also has a corresponding interrupt enable bit (ADCIE, ADCON2<7>). The timer interrupt can be enabled/disabled by setting/clearing this bit.

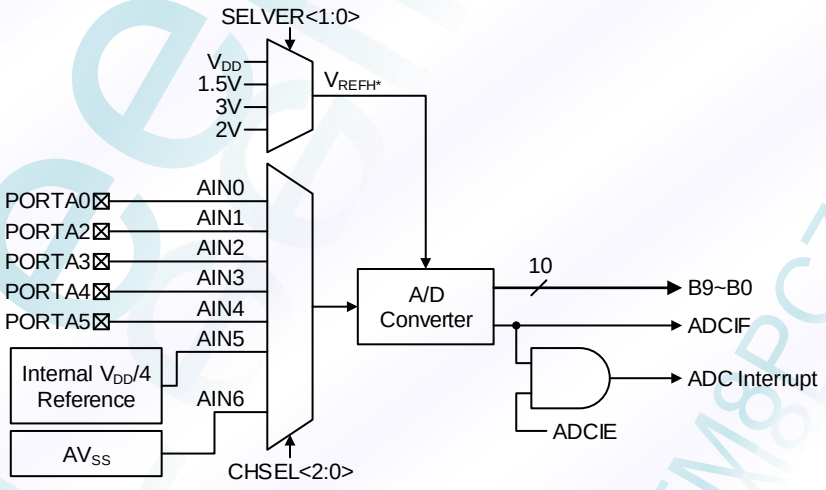
When changing channels, a conversion-time is required before starting the next conversion.

The ADC voltage reference is software selectable to either V_{DD} or three internal voltage (3V, 2V or 1.5V, selected by SELVER<1:0> bits (ADCON2<1:0>)).

2.8.1 Port configuration

ADC input pins are shared with digital I/O pins. Connect an analog signal to COMS digital input pin, especially, the analog signal level is about V_{DD}/2 will cause extra current leakage. In the any operation mode, above leakage current will cause unnecessary distress. Therefore, can be setting the corresponding INEN (ADCON3) bit configuration I/O pin as analog input pin.

Figure 2.11: Block Diagram of the AD Converter



Note: Some control signal are not show in this Block Diagram.

2.8.2 Converter clock source

ADC conversion clock source is from F_{CPU}, and provides 8 kinds of conversion source. The clock source can be selected by ADCSR<2:0> bits (ADCON1<2:0>).
The ADC has two sample rate can be selected by ADCTCK bit (ADCON2<4>).
The ADC conversion time can be calculated as follows:

$$\text{Conversion time} = \frac{\text{ADC Conversion clock source} * \text{ADC sample rate}}{F_{CPU}\ \text{frequency}}$$

If ADCTCK bit = 1

$$\text{Conversion time} = \frac{\text{ADC Conversion clock source} * 44}{F_{CPU}\ \text{frequency}}$$

If ADCTCK bit = 0

$$\text{Conversion time} = \frac{\text{ADC Conversion clock source} * 34}{F_{CPU}\ \text{frequency}}$$

For example, F_{CPU} = 8M$_{HZ}$, Clock source is F_{CPU}/4, ADCTCK = 1, conversion time calculated as follows:

$$\text{Conversion time} = \frac{4 * 44}{8M_{HZ}} = 22uS$$

Table 2.2: ADC conversion time (In Pure AD Conversion mode) table

ADCTCK = 1 (44 clocks)		F_{CPU} Frequency				
ADC clock source	ADCSR<2:0>	16M$_{HZ}$	8M$_{HZ}$	4M$_{HZ}$	2M$_{HZ}$	1M$_{HZ}$
F_{CPU}/1	011	-(1)	-(1)	11uS(2)	22uS(2)	44uS(3)
F_{CPU}/2	010	-(1)	11uS(2)	22uS(2)	44uS(3)	88uS(3)
F_{CPU}/4	001	11uS(2)	22uS(2)	44uS(3)	88uS(3)	176uS(3)
F_{CPU}/8	000	22uS(2)	44uS(3)	88uS(3)	176uS(3)	352uS(3)
F_{CPU}/16	100	44uS(3)	88uS(3)	176uS(3)	352uS(3)	704uS(3)
F_{CPU}/32	101	88uS(3)	176uS(3)	352uS(3)	704uS(3)	1.41mS(3)
F_{CPU}/64	110	176uS(3)	352uS(3)	704uS(3)	1.41mS(3)	2.82mS(3)
F_{CPU}/128	111	352uS(3)	704uS(3)	1.41mS(3)	2.82mS(3)	5.63mS(3)

ADCTCK = 0 (34 clocks)		F_{CPU} Frequency				
ADC clock source	ADCSR<2:0>	16M$_{HZ}$	8M$_{HZ}$	4M$_{HZ}$	2M$_{HZ}$	1M$_{HZ}$
F_{CPU}/1	011	-(1)	-(1)	8.5uS(2)	17uS(2)	34uS(2)
F_{CPU}/2	010	-(1)	8.5uS(2)	17uS(2)	34uS(2)	68uS(3)
F_{CPU}/4	001	8.5uS(2)	17uS(2)	34uS(2)	68uS(3)	135uS(3)
F_{CPU}/8	000	17uS(2)	34uS(2)	68uS((3)	135uS(3)	272uS(3)
F_{CPU}/16	100	34uS(2)	68uS(3)	135uS(3)	272uS(3)	544uS(3)
F_{CPU}/32	101	68uS(3)	135uS(3)	272uS(3)	544uS(3)	1.09mS(3)
F_{CPU}/64	110	135uS(3)	272uS(3)	544uS(3)	1.09mS(3)	2.18mS(3)
F_{CPU}/128	111	272uS(3)	544uS(3)	1.09mS(3)	2.18mS(3)	5.35mS(3)

Legend: Shaded cells are outside of recommended range.

Note : **1. AD Conversion time cannot be less than 8uS.**

2. If conversion source is from V_{DD}/4 (AIN5), Conversion time cannot be less than 34uS.

3. For faster conversion times, the selection of another clock source is recommended.

4. If conversion source is from V_{DD}/4 (AIN5), need once dummy conversion.

2.8.3 Continues conversion mode

ADC has continue mode can be selected, if ADCNT (ADCON2<5>) bit is set, after the completion of a conversion cycle, it will automatically continue to convert next analog signals. In continuous conversion mode, AD conversion operation controlled by hardware. Therefore, AD conversion start/stop by setting/clearing ADCNT bit, ADCST bit status will be ignored.
When ADCNT bit is cleared, AD conversion will stop immediately. In this case, the ADC result is wrong.

Figure 2.12: AD Conversion timing (ADCNT = 1)

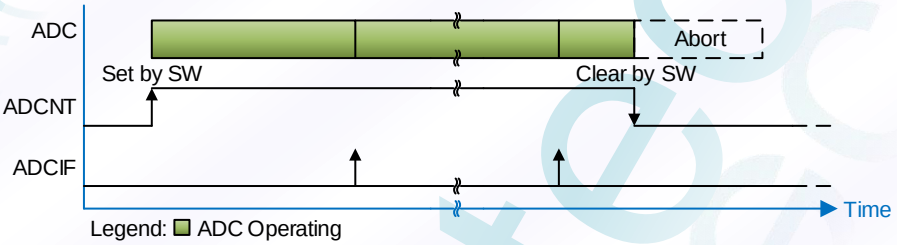
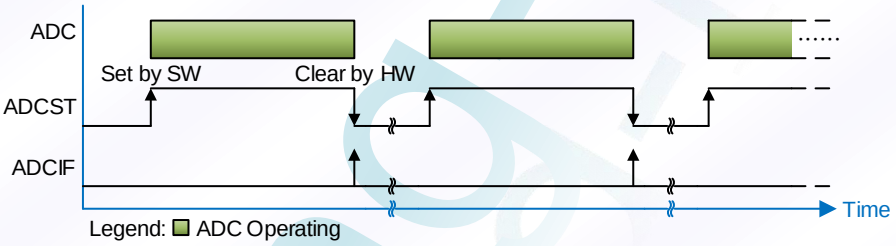


Figure 2.13: AD Conversion timing (ADCNT = 0)



2.9 Voltage Comparator

The comparator module has seven input sources, the positive input of the comparator is connected to the DAC output.
Input source can be selected by CMPCS<2:0> (CMPCON1<2:0>) to select analog signal input pin (AIN pin), internal V_{DD}/4 and V_{SS} voltage source.
Comparator module has 5 modes of operation, the comparator can be used alone, compared with the DAC output voltage, and interact with the ADC.

Figure 2.14: Block Diagram of the Voltage Comparator

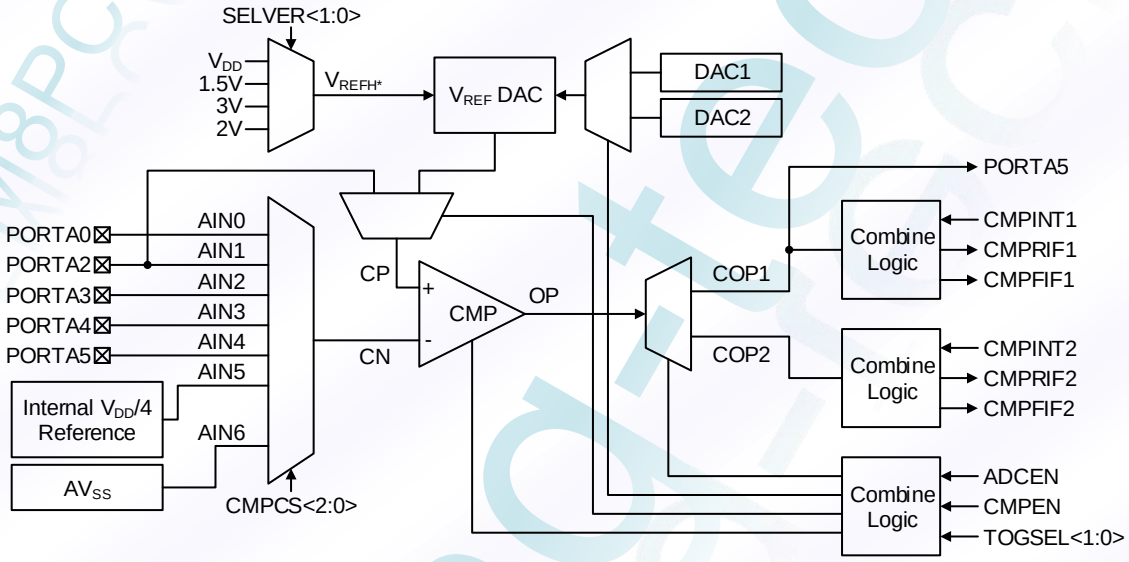


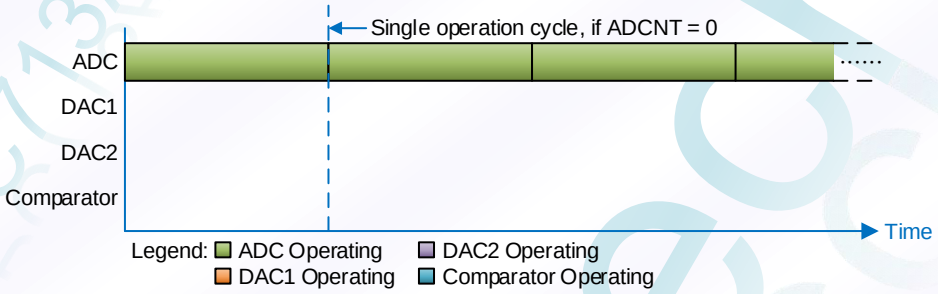
Table 2.3: Comparator operating mode

ADCEN	CMPEN	TOGSEL1 : TOGSEL0		Function Descript
1	0	0	0	Toggle mode 0. This mode is normal ADC convert mode.
1	1	0	0	Toggle mode 1. This mode is DAC1 and DAC2 interact comparison.
1	1	0	1	Toggle mode 2. This mode is DAC1, DAC2 and ADC interact compare and conversion.
1	1	1	0	Toggle mode 3. This mode is Pure Comparator mode. In this mode, ADC, DAC will stop.
1	1	1	1	Toggle mode 4. This mode same Toggle mode 1, PA5 is COP1 status output.

2.9.1 Toggle mode 0

This mode is pure AD Conversion mode. See section 2.8 for detail description on page 40.

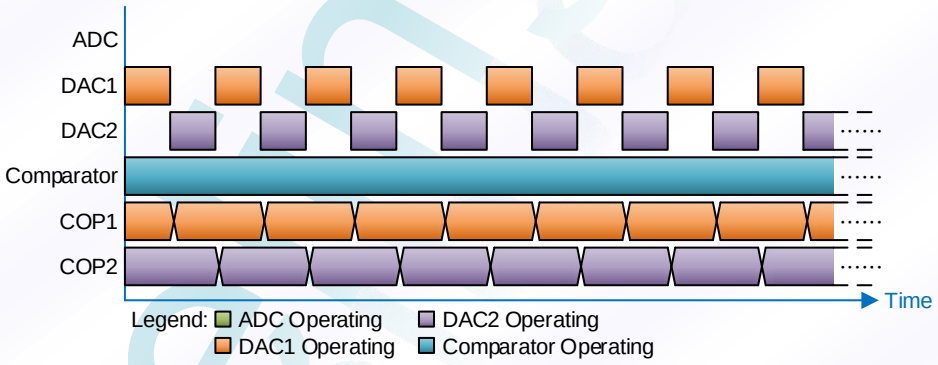
Figure 2.15: Operational timing of Toggle mode 0 (ADCNT = 1)



2.9.2 Toggle mode 1

In this mode device will enabled the Comparator, DAC1 and DAC2. The negative input of the comparator will turn sequentially compared with the DAC1, DAC2 (Polling, Operational timing refer to Figure 2.16 on page 44). ADC functionality will be forced to disabled.
Comparison of the results will be sequentially output to COP1 (CMPCON1<6>) and COP2 (CMPCON2<6>) bits, and set the associated rise/fall flag (CMPRIFx/CMPFIFx).
The results of the comparator is not real-time, it will only update the final results until the end of each of the Toggle stage.

Figure 2.16: Operational timing of Toggle mode 1



The Comparator outputs COP1 and COP2 have corresponding interrupt enable bits (CMPIE1 CMPCON1<5>, CMPIE2 CMPCON2<5>). Interrupt can be enabled/disabled by setting/clearing these bit. The comparator output-change/falling-edge interrupts can be selected by CMPINT1 (CMPCON2<7>) and CMPINT2 (CMPCON2<2>) bits. In this mode, the hardware will be fixed to the continuous mode, even if ADCNT bit is set to 0.

2.9.2.1 DAC clock source

DAC conversion clock source is from F_{CPU}, and provides 8 kinds of conversion source (Shared with ADC). The clock source can be selected by ADCSR<2:0> bits (ADCON1<2:0>).
The DAC has two conversion rate can be selected by ADCTCK bit (ADCON2<4>). In this mode, when ADCTCK bit is 1, DAC conversion time requires 9 clock. When ADCTCK bit is 0, DAC conversion time required 7 clock.
The DAC conversion time can be calculated as follows:

Continue to Next page.

$$\text{Conversion time} = \frac{\text{DAC Conversion clock source} * \text{DAC Conversion rate}}{F_{CPU}\ \text{frequency}}$$

If ADCTCK bit = 1

$$\text{Conversion time} = \frac{\text{DAC Conversion clock source} * 9}{F_{CPU}\ \text{frequency}}$$

If ADCTCK bit = 0

$$\text{Conversion time} = \frac{\text{DAC Conversion clock source} * 7}{F_{CPU}\ \text{frequency}}$$

For example, F_{CPU} = 8M$_{HZ}$, Clock source is F_{CPU}/4, ADCTCK = 1, conversion time calculated as follows:

$$\text{Conversion time} = \frac{4 * 9}{8M_{HZ}} = 4.5uS$$

Table 2.4: DAC conversion time table (In toggle mode)

ADCTCK = 1 (9 clocks)		F_{CPU} Frequency				
ADC clock source	ADCSR<2:0>	16M$_{HZ}$	8M$_{HZ}$	4M$_{HZ}$	2M$_{HZ}$	1M$_{HZ}$
F_{CPU}/1	011	-(1)	-(1)	2.25uS(2)	4.5uS(2)	9uS(3)
F_{CPU}/2	010	-(1)	2.25uS(2)	4.5uS(2)	9uS(3)	18uS(3)
F_{CPU}/4	001	2.25uS(2)	4.5uS(2)	9uS(3)	18uS(3)	36uS(3)
F_{CPU}/8	000	4.5uS(2)	9uS(3)	18uS(3)	36uS(3)	72uS(3)
F_{CPU}/16	100	9uS(3)	18uS(3)	36uS(3)	72uS(3)	144uS(3)
F_{CPU}/32	101	18uS(3)	36uS(3)	72uS(3)	144uS(3)	288uS(3)
F_{CPU}/64	110	36uS(3)	72uS(3)	144uS(3)	288uS(3)	576uS(3)
F_{CPU}/128	111	72uS(3)	144uS(3)	288uS(3)	576uS(3)	1.15mS(3)

ADCTCK = 0 (7 clocks)		F_{CPU} Frequency				
ADC clock source	ADCSR<2:0>	16M$_{HZ}$	8M$_{HZ}$	4M$_{HZ}$	2M$_{HZ}$	1M$_{HZ}$
F_{CPU}/1	011	-(1)	-(1)	1.75uS(2)	3.5uS(2)	7uS(2)
F_{CPU}/2	010	-(1)	1.75uS(2)	3.5uS(2)	7uS(2)	14uS(3)
F_{CPU}/4	001	1.75uS(2)	3.5uS(2)	7uS(2)	14uS(3)	28uS(3)
F_{CPU}/8	000	3.5uS(2)	7uS(2)	14uS(3)	28uS(3)	56uS(3)
F_{CPU}/16	100	7uS(2)	14uS(3)	28uS(3)	56uS(3)	112uS(3)
F_{CPU}/32	101	14uS(3)	28uS(3)	56uS(3)	112uS(3)	224uS(3)
F_{CPU}/64	110	28uS(3)	56uS(3)	112uS(3)	224uS(3)	448uS(3)
F_{CPU}/128	111	56uS(3)	112uS(3)	224uS(3)	448uS(3)	896uS(3)

Legend: Shaded cells are outside of recommended range.

Note : **1. DA Conversion time cannot be less than 1.75uS.**

2. For faster conversion times, the selection of another clock source is recommended.

2.9.2.2 DAC VREF

The DAC voltage reference is software selectable to either V_{DD} or three internal voltage (3V, 2V or 1.5V, selected by SELVER<1:0> bits (ADCON2<1:0>), Shared with ADC).

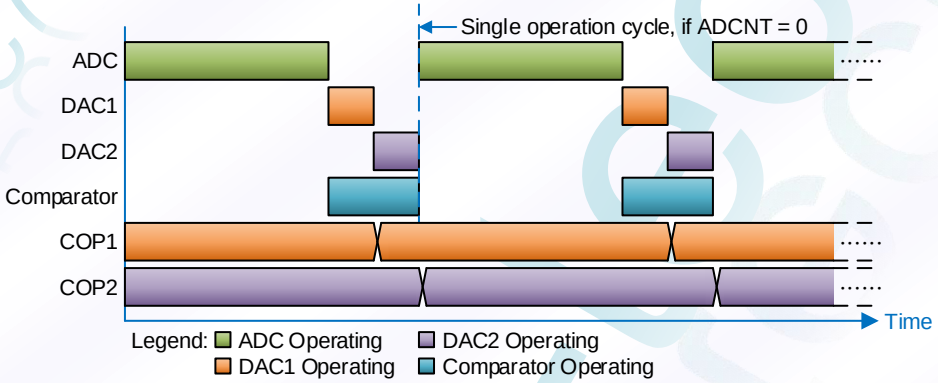
The DAC voltage output can be calculated as follows:

$$\text{DAC1 } V_{OUT} = \frac{V_{REFH} * [\text{DACR1HB} : \text{DACR1LB}]}{1024}, \quad \text{DAC2 } V_{OUT} = \frac{V_{REFH} * [\text{DACR2HB} : \text{DACR2LB}]}{1024}$$

2.9.3 Toggle mode 2

In this mode, all functions will be enabled (ADC, DAC and Comparator). The ADC, DAC1 and DAC2 will be sequential converted and compared. The comparator input channels and the ADC input channels independently, therefore, ADC and comparator can be specify different input source.
In this mode, in addition to different ADC conversion time, the other operating mode, please refer section 2.8 and 2.9.2 (on page 40 and 44).

Figure 2.17: Operational timing of Toggle mode 2 (ADCNT = 1)



2.9.3.1 AD Converter clock source

ADC conversion clock source is from F_{CPU}, and provides 8 kinds of conversion source. The clock source can be selected by ADCSR<2:0> bits (ADCON1<2:0>).
The ADC has two sample rate can be selected by ADCTCK bit (ADCON2<4>).
In this mode, **ADC and DAC sequential polling reason, when ADCTCK bit is 1, ADC conversion time requires 47 clock. When ADCTCK bit is 0, ADC conversion time required 36 clock.**
The ADC conversion time can be calculated as follows:

$$\text{Conversion time} = \frac{\text{ADC Conversion clock source} * \text{ADC sample rate}}{F_{CPU}\ \text{frequency}}$$

If ADCTCK bit = 1

$$\text{Conversion time} = \frac{\text{ADC Conversion clock source} * 47}{F_{CPU}\ \text{frequency}}$$

If ADCTCK bit = 0

$$\text{Conversion time} = \frac{\text{ADC Conversion clock source} * 36}{F_{CPU}\ \text{frequency}}$$

For example, F_{CPU} = 8MHZ, Clock source is F_{CPU}/4, ADCTCK = 1, conversion time calculated as follows:

$$\text{Conversion time} = \frac{4 * 47}{8M_{HZ}} = 23.5uS$$

Continue to Next page.

Table 2.5: ADC conversion time table (In toggle mode)

ADCTCK = 1 (47 clocks)		F_{CPU} Frequency				
ADC clock source	ADCSR<2:0>	16MHZ	8MHZ	4MHZ	2MHZ	1MHZ
F_{CPU}/1	011	-(1)	-(1)	11.75uS(2)	23.5uS(2)	47uS(3)
F_{CPU}/2	010	-(1)	11.75uS(2)	23.5uS(2)	47uS(3)	94uS(3)
F_{CPU}/4	001	11.75uS(2)	23.5uS(2)	47uS(3)	94uS(3)	188uS(3)
F_{CPU}/8	000	23.5uS(2)	47uS(3)	94uS(3)	188uS(3)	376uS(3)
F_{CPU}/16	100	47uS(3)	94uS(3)	188uS(3)	376uS(3)	752uS(3)
F_{CPU}/32	101	94uS(3)	188uS(3)	376uS(3)	752uS(3)	1.5mS(3)
F_{CPU}/64	110	188uS(3)	376uS(3)	752uS(3)	1.5mS(3)	3mS(3)
F_{CPU}/128	111	376uS(3)	752uS(3)	1.5mS(3)	3mS(3)	6.02mS(3)

ADCTCK = 0 (36 clocks)		F_{CPU} Frequency				
ADC clock source	ADCSR<2:0>	16MHZ	8MHZ	4MHZ	2MHZ	1MHZ
F_{CPU}/1	011	-(1)	-(1)	9uS(2)	18uS(2)	36uS(2)
F_{CPU}/2	010	-(1)	9uS(2)	18uS(2)	36uS(2)	72uS(3)
F_{CPU}/4	001	9uS(2)	18uS(2)	36uS(2)	72uS(3)	144uS(3)
F_{CPU}/8	000	18uS(2)	36uS(2)	72uS(3)	144uS(3)	288uS(3)
F_{CPU}/16	100	36uS(2)	72uS(3)	144uS(3)	288uS(3)	576uS(3)
F_{CPU}/32	101	72uS(3)	144uS(3)	288uS(3)	576uS(3)	1.15mS(3)
F_{CPU}/64	110	144uS(3)	288uS(3)	576uS(3)	1.15mS(3)	2.3mS(3)
F_{CPU}/128	111	288uS(3)	576uS(3)	1.15mS(3)	2.3mS(3)	4.61mS(3)

Legend: Shaded cells are outside of recommended range.

Note : **1. AD Conversion time cannot be less than 9uS.**

2. If conversion source is from V_{DD}/4 (AIN5), Conversion time cannot be less than 36uS.

3. DA Conversion timing, refer Table 2.4 for detail (on page 45).

4. For faster conversion times, the selection of another clock source is recommended.

5. If conversion source is from V_{DD}/4 (AIN5), need once dummy conversion.

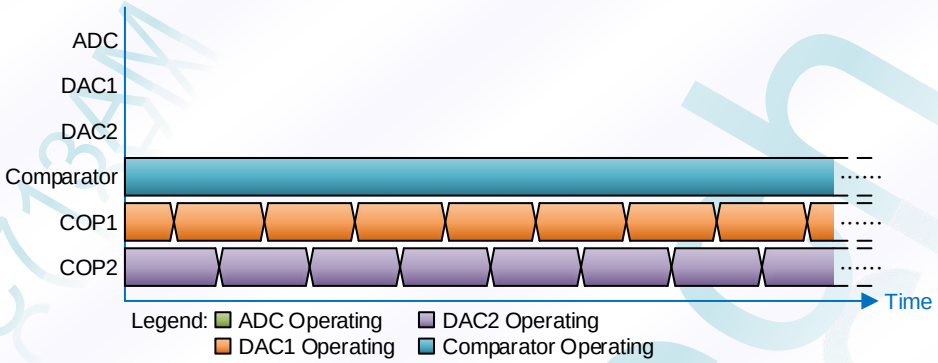
2.9.3.2 DA Converter clock source and VREF

This part same as Toggle mode 1. See section 2.9.2 for detail description on page 44.

2.9.4 Toggle mode 3

This mode is pure comparator mode, ADC and DAC functionality will be forced to disabled. The positive input is fixed at PA2 pin, negative input can be selected by CMPCS<2:0> bits (CMPCON1<2:0>).
Comparison of the results will be sequentially output to COP1 (CMPCON1<6>) and COP2 (CMPCON2<6>) bits, and set the associated rise/fall flag (CMPRIFx/CMPFIFx).
The results of the comparator is not real-time, it will only update the final results until the end of each of the Toggle stage.
The Comparator outputs COP1 and COP2 have corresponding interrupt enable bits (CMPIE1 CMPCON1<5>, CMPIE2 CMPCON2<5>). Interrupt can be enabled/disabled by setting/clearing these bit. The comparator output-change/falling-edge interrupts can be selected by CMPINT1 (CMPCON2<7>) and CMPINT2 (CMPCON2<2>) bits.
COP1 and COP2 alternating time was 7 or 9 clock time. Setting and calculation methods with the same section 2.9.2.1 (on page 44). In this mode, No shortest time limit.
In this mode, the hardware will be fixed to the continuous mode, even if ADCNT bit is set to 0.

Figure 2.18: Operational timing of Toggle mode 3



2.9.5 Toggle mode 4

This mode same Toggle mode-2, but the state of COP1 will output to PA5 pin. See section 2.9.3 for detail description on page 46.

2.10 Operating Mode

The device have four operating mode for difference clock rate and power saving reason. Mode follows:

- Normal mode
- Slow mode
- Green mode (Green mode0/1 and Ultra Green mode0/1)
- Suspend mode

Figure 2.19: Block Diagram Operation mode

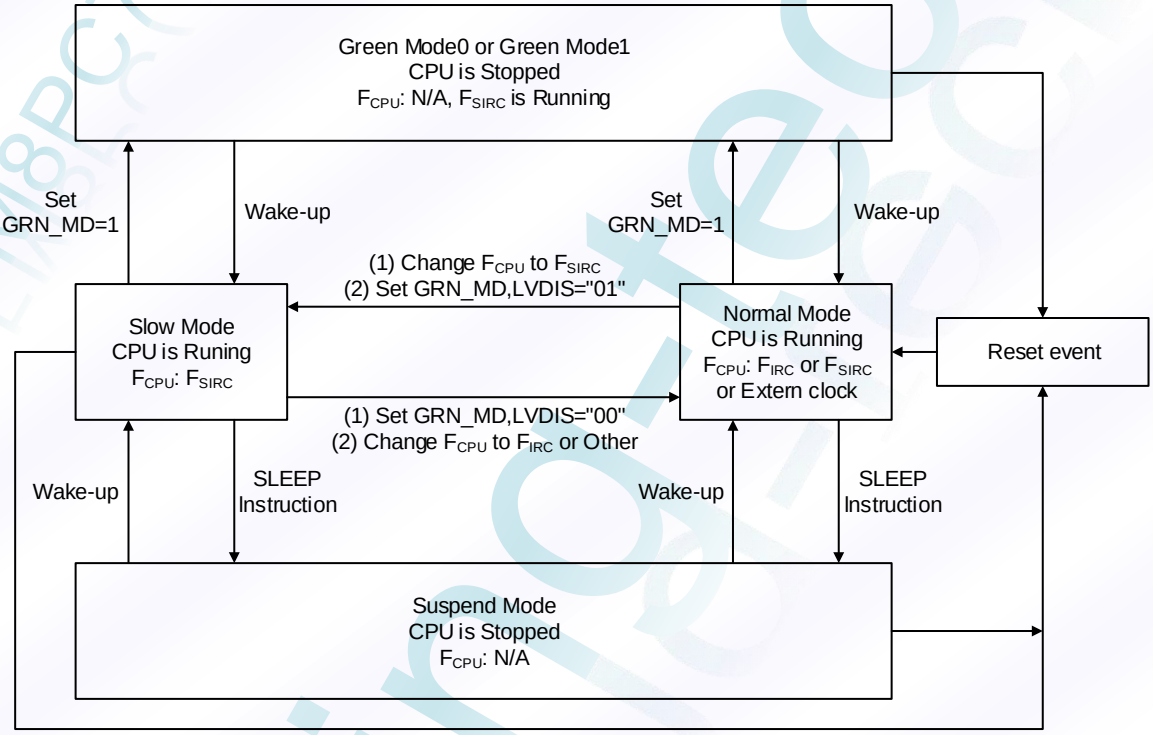


Table 2.6: Device status of Operation mode

MODE / Module	Normal mode	Slow mode	Green mode	Suspend mode
F_{IRC}	Running (IRCEN=1)	Stopped	Stopped	Stopped
F_{SIRC}	Running	Running	Running	Stopped
Watchdog	Running (WDTE=1)	Running (WDTE=1)	Running (WDTE=1)	Stopped
Internal interrupt	All enable	All enable	All enable	All enable
External interrupt	All enable	All enable	All enable	All enable
Wake-up	N/A	N/A	Port A, Timer0, WDT time-out, RSTB	Port A, RSTB
Sleep instruction	Non-required	Non-required	Non-required	Required
CPU	Work	Work	Stop	Stop
Clock source of CPU	F_{IRC} / F_{SIRC} / External clock	F_{SIRC}	N/A	N/A

2.10.1 Normal mode

In this mode, the device can operate at high-speed clock sources, and all functions are enabled. Similarly, the maximum power consumption in this mode.
This mode is the default mode, when after a reset event, will automatically enter this mode.
This mode can be enter Green mode or Suspend mode.

2.10.2 Slow mode

In slow mode, the device operate at a low-speed clock source, **high-speed clock source will be forcibly disabled. Therefore, before entering the slow mode, the system clock source must be first switched to low-speed source.** In this mode, in addition to LVDT function will be disabled, the other functions remains enabled. Similarly, this mode will be slightly lower than the normal-mode power consumption.
This mode can be enter Green mode or Suspend mode.
Note: Switch clock source detail, refer section 2.12 for detail description on page 55.

2.10.3 Green mode 0 and Green mode 1

Green Mode 0 will only keep F_{SIRC}, Watchdog, Timer 0 interrupt and PortA interrupt, other functions will be disabled.
Green Mode 1 will only keep F_{SIRC}, Watchdog, Timer 0 interrupt, PortA interruptions and LVDT, other functions will be disabled. In this mode, power consumption will be slightly larger than the Suspend mode.
This mode can be set
This mode is controlled by GRN_MD:LVDIS (PCON<6:5>) bits. When the GRN_MD:LVDIS bits is set to 10, the device will be enter Green mode 0, When the set to 11, the device will be enter mode 1.
If setting UGMD bit (PCON<2>) after enter green mode, the device will be entry the ultra-green mode, F_{SIRC} frequency will drop to 3.2K$_{HZ}$, will have a lower power consumption.
Before entering the green mode, wake-related flags or watchdog must be cleared. Otherwise, device may not be awakened.
The device can wake-up from Green mode through one of the following events:

1. PORTA external interrupt.
2. Timer0 interrupt
3. WDT time-out reset (if enabled).
4. LVDT interrupt or reset (only for Green mode 0)
5. RSTB reset (if enabled).

External RSTB reset and WDT time-out reset will cause a device reset. The $\overline{TO}$ bits can be used to determine the reset or wake-up by watchdog.
For the device to wake-up through an interrupt event, the corresponding interrupt enable bit must be set. Wake-up is regardless of the GIE bit. If GIE bit is cleared, the device will continue execution next instruction. If the GIE bit is set, the device will branch to the interrupt address (0x003).
When the wake-up event occurs, the device will return to the previous operating mode. If Reset event occurs, the device will be reset, and go to Normal mode.
Please note, this mode cannot directly enter Suspend mode. If need to switch to suspend mode, must be first switch to Normal mode or slow mode.

2.10.4 Suspend mode

The suspend mode is disable all function (exclude PortA interrupt) for lowest power consumption. Suspend mode is entered by executing a SLEEP instruction. **If need to enter Suspend mode, it must first be switch to the Normal or Slow mode before enter Suspend mode.**
When SLEEP instruction is executed, the $\overline{PD}$ bit (STATUS<3>) is cleared, the $\overline{TO}$ bit is set, the watchdog timer will be force to disable.
All I/O pins maintain the status they had before the SLEEP instruction was executed.
Before entering Suspend mode, PAIF (INTFLAG<6>) must be cleared. Otherwise, device may not be awakened.
The device can wake-up from Suspend mode through one of the following events:

1. PORTA external interrupt.
2. RSTB reset (if enabled).

When the wake-up event occurs, the device will return to the previous operating mode. If Reset event occurs, the device will be reset, and go to Normal mode.

2.11 Reset

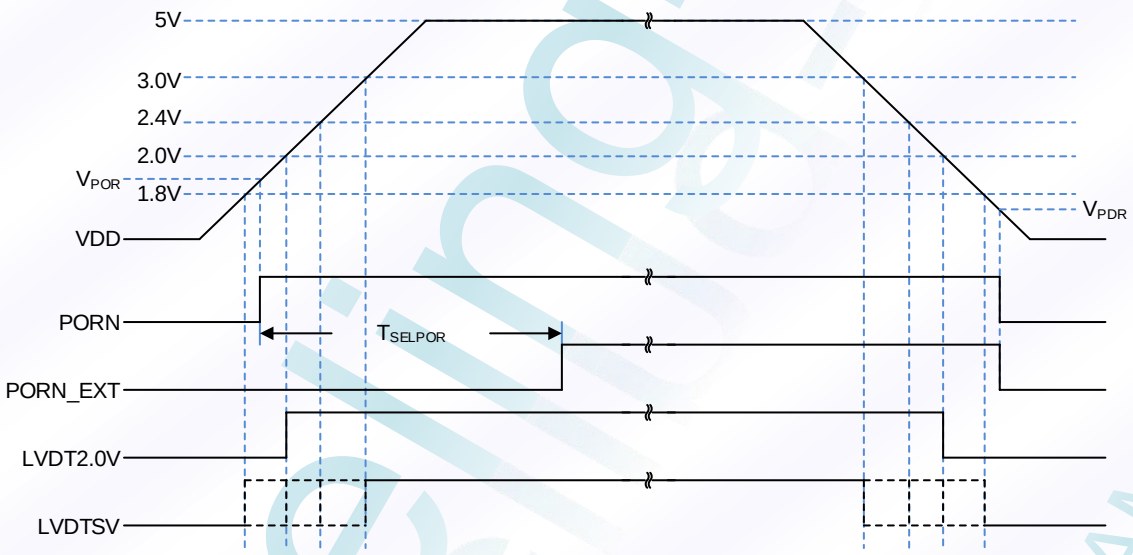
FM8PC713AM devices may be RESET in one of the following ways:

1. Power-on Reset (POR) and Power-down Reset (PDR)
2. Power-down Reset in Sleep mode (PDRS)
3. WDT time-out Reset (WDR)
4. Brown-out Reset (BOR)
5. RSTB Pin Reset

When the POR, PDR, PDRS and WDR Reset event occurs, the device will be into the reset state. BOR (LVDT20V or LVDTSV) and RSTB pin reset event can be enabled by select bits RSTL<1:0> (PCON<4:3>). When the device enters the reset state, all registers will be reset to default, some flags will not be affected, and generate the corresponding state. Register details, refer to Table 2.7 on page 53.

After the reset, the system clock source will be preset to F_{FIG}, clock frequency is $1M_{HZ}$.

Figure 2.20: Power-on Reset and LVDT20V, LVDTSV Timing

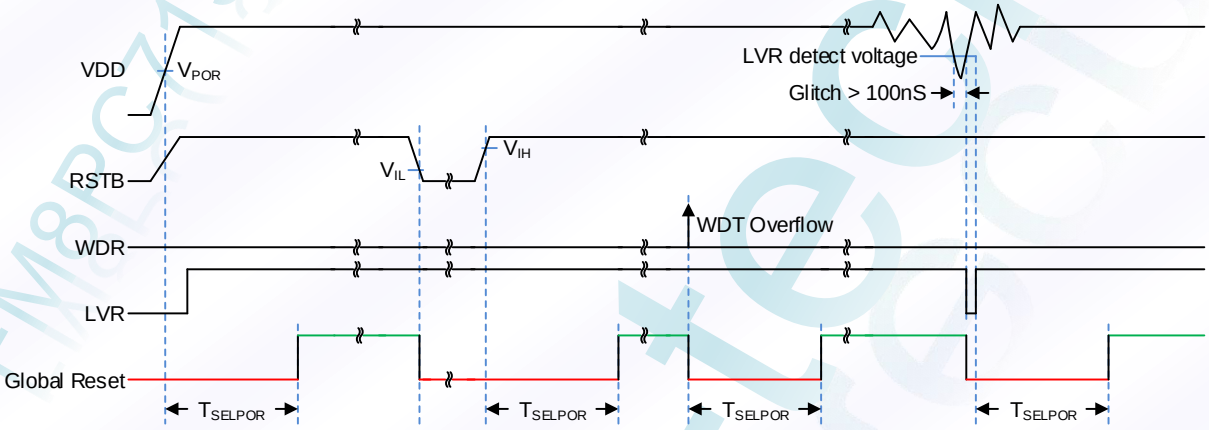


Note : 1. T_{SELPOR} select by configuration bit, refer Configuration word for detail.
2. Please refer DC characteristics for V_{POR} and V_{PDR} detail.

2.11.1 Power-up Reset Timer (PWRT)

The Power-up Reset Timer provides a delay after Power-on Reset (POR), Brown-out Reset (BOR), RSTB Reset or WDT time-out Reset. The device is kept in reset state as long as the PWRT is active. This delay time can be selected by SELPOR bits of configuration word.

Figure 2.21: Global Reset timing



Note : 1. T_{SELPOR} select by configuration bit, refer Configuration word for detail.
2. Please refer DC characteristics for V_{POR} and V_{PDR} detail.

Table 2.7: Reset Conditions for All Registers

Register	Address	Power-on Reset Brown-out Reset	RSTB Reset WDT Reset LVR Reset
ACC	N/A	xxxx xxxx	uuuu uuuu
PAMODE0	0x08	--00 0010	--00 0010
PAMODE1	0x09	--00 0000	--00 0000
P0TMPRLB	0x17	1111 1111	1111 1111
P0TMPRHB	0x19	1111 1111	1111 1111
P1TMPR	0x23	1111 1111	1111 1111
INDF	0x00	0000 0000	0000 0000
PCL	0x02	0000 0000	0000 0000
STATUS	0x03	--01 1000	000# #000
FSR	0x04	-000 0000	0000 0000
PORTA	0x05	--xx xxxx	--uu uuuu
PAIE	0x07	--00 0000	--00 0000
PACON	0x09	--00 0000	--00 0000
INTEN	0x0B	0--0 0--0	0--0 0--0
INTFLAG	0x0C	-0-0 0--0	-0-0 0--0
TM0	0x0E	0000 0000	0000 0000
TM0CON	0x0F	0-00 000-	0-00 000-
T0RLD	0x10	0000 0000	0000 0000
WDT	0x11	1100 00--	1100 00--
PCON	0x12	-000 00LL	-00u u0LL
CLKCFG	0x13	0000 -101	0000 -101
PWM0CON	0x14	0-00 0000	0-00 0000
PWM0CR	0x15	0000 0000	0000 0000
P0TMLB	0x16	0000 0000	0000 0000
P0TMDTLB	0x17	0000 0000	0000 0000
P0TMHB	0x18	0000 0000	0000 0000
P0TMDTHB	0x19	0000 0000	0000 0000
PWM1CON	0x20	0-00 0000	0-00 0000
PWM1CR	0x21	0000 0000	0000 0000
P1TM	0x22	0000 0000	0000 0000
P1TMDT	0x23	0000 0000	0000 0000
BZS	0x24	0000 ----	0000 ----
ADCON1	0x28	0000 0000	0000 0000
ADCON2	0x29	00-0 1000	00-0 1000
ADCHB	0x30	0000 0000	0000 0000
ADCLB	0x31	---- --00	---- --00
ADCON3	0x32	---0 0000	---0 0000
CMPCON1	0x35	0000 0000	0000 0000
DACR1HB	0x36	0000 0000	0000 0000
DACR1LB	0x37	---- --00	---- --00
CMPCON2	0x38	0000 0000	0000 0000

Register	Address	Power-on Reset Brown-out Reset	RSTB Reset WDT Reset LVR Reset
DACR2HB	0x39	0000 0000	0000 0000
DACR2LB	0x3A	---- --00	---- --00
General Purpose Registers	0x40 ~ 0x7F	xxxx xxxx	uuuu uuuu

Legend : u = unchanged, x = unknown, - = unimplemented, read as 0.
# = refer to the following table for possible values.
L = LVDT flags, this flag will be based on V_{DD} voltage different states.

Table 2.8: $\overline{TO}$ / $\overline{PD}$ Status after Reset or Wake-up

$\overline{TO}$	$\overline{PD}$	RESET was caused by
1	1	Power-on Reset
1	1	Brown-out reset
u	u	RSTB Reset during normal operation
1	0	RSTB Reset during SLEEP
0	1	WDT Reset (or Wake-up) during Normal, Slow or Green mode

Legend: u = unchanged

Table 2.9: Events Affecting$\overline{TO}$ / $\overline{PD}$Status Bits

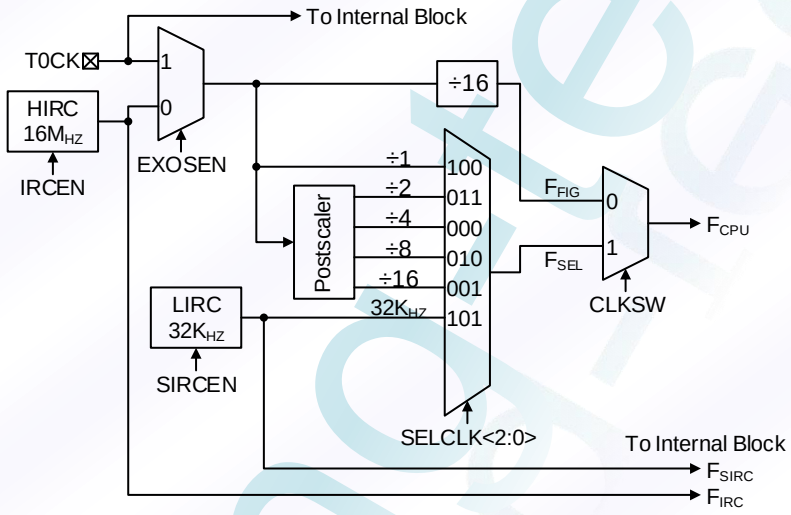
Event	$\overline{TO}$	$\overline{PD}$
Power-on	1	1
WDT Time-Out	0	u
SLEEP instruction	1	0
CLRWDT instruction	1	1

Legend: u = unchanged

2.12 Oscillator Configurations

The system clock sources can be selected from external clock or Internal Resistor-Capacitor (IRC) oscillator. Internal clock includes two internal oscillators: HIRC (16M$_{HZ}$) and LIRC (32K$_{HZ}$) oscillator. These oscillator frequency has been calibrated.
The system clock source has two paths, respectively F$_{FIG}$ and F$_{SEL}$. When the chip at power-up or reset, the clock source and the path will default to HIRC and F$_{FIG}$ path. Such as the need to change the system clock speed, see section 2.12.3 detail description on page 56.
The system clock (F$_{CPU}$) frequency and voltage operating range, please refer to the DC characteristics table VF$_{CPU}$ values.

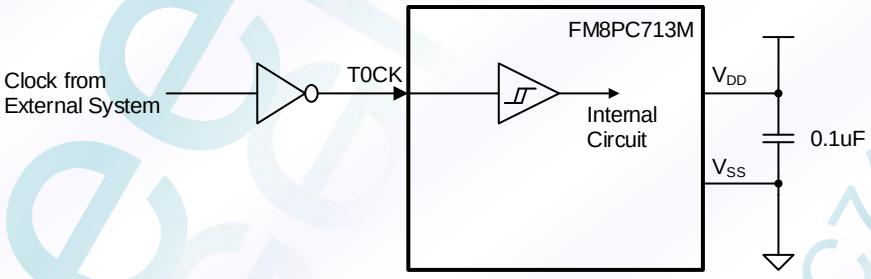
Figure 2.22: System Clock Source Block Diagram



2.12.1 External Clock Modes

The External Clock mode allows an externally generated logic level as the system clock source. In this mode, the PA4 must be configuration to input mode.

Figure 2.23: External Clock Mode



2.12.2 Internal Clock Modes

Internal clock sources are contained internally within the IRC module. The IRC module has two internal oscillators: the 16M$_{HZ}$ High-Frequency Internal Oscillator (HIRC) and the 32K$_{HZ}$ Low-Frequency Internal Oscillator (LIRC) that can be selected as the system clock source.

2.12.3 Change system clock speed

The system clock speed can be selected by setting the Clock Divider Select bits SELCLK<2:0> of the CLKCFG register.

When you want to switch clock source, Need follow these steps:

1. Switching the clock source path to F_{FIG} (CLKSW bit (CLKCFG<4>) set to 0).
2. Insert a NOP instruction.
3. Specify new clock speed** (see note).
4. Insert a NOP instruction.
5. Switching the clock source path to F_{SEL} (CLKSW bit (CLKCFG<4>) set to 1).

Example 2.5: Change clock speed (Change to $8M_{HZ}$)

ASM Language Code

```
#include   <8PC713AM.ASH>
           …
           BCR     CLKCFG,CLKSW_B      ; Switch system clock source to FFIG path, required.
           NOP
           BSR     CLKCFG,SELCLK0_B
           BSR     CLKCFG,SELCLK1_B
           BCR     CLKCFG,SELCLK2_B    ; 8MHZ SELCK<2:0> is 011
           NOP
           BSR     CLKCFG,CLKSW_B      ; Switch system clock source to FSEL path, required.
           …
```

C Language Code

```
#include   <8PC713AM.h>
           …
           CLKCFGbits.CLKSW=0;         // Switch system clock source to FFIG path, required.
           NOP();
           CLKCFGbits.SELCLK0=1;
           CLKCFGbits.SELCLK1=1;
           CLKCFGbits.SELCLK2=0;       // 8MHZ SELCK<2:0> is 011
           NOP();
           CLKCFGbits.CLKSW=1;         // Switch system clock source to FSEL path, required.
           …
```

Note : 1. CLKCFG register prohibits direct write value, Need using BSR/BCR instruction.

2. When switch to another speed, setting procedures must same as example.

3. Failure to follow set procedures could result in unexpected situation occurs.

2.12.4 Change system clock source

The system clock source can be selected by setting the Clock source Select bit EXOSEN of the CLKCFG register. When you want to switch clock source, Need follow these steps:

Situation 1: **Switching from the internal clock (HIRC) to an external clock**

1. Switching the clock source path to F_{FIG} (CLKSW bit (CLKCFG<4>) set to 0).
2. Insert a NOP instruction.
3. Switching the clock source to LIRC** (see note).
4. Insert a NOP instruction.
5. Switching the clock source path to F_{SEL} (CLKSW bit (CLKCFG<4>) set to 1).
6. Switching the clock source to external clock (EXOSEN bit (CLKCFG<1>) set to 1).
7. Wait for the end of the switching time (switching-time specified by the INSWDY bit of the Configuration word).
8. Switching the clock source path to F_{FIG} (CLKSW bit (CLKCFG<4>) set to 0).
9. Insert a NOP instruction.
10. Specify new clock speed** (see note).
11. Insert a NOP instruction.
12. Switching the clock source path to F_{SEL} (CLKSW bit (CLKCFG<4>) set to 1).
13. If HIRC clock sources no longer in use, can be set IRCEN bit (CLKCFG<0>) to 0, turn-off HIRC.

Situation 2: **Switching from the an external clock to internal clock (HIRC)**

1. The IRCEN bit (CLKCFG<0>) is set to 1, turn-on HIRC (If HIRC is turn-off).
2. Switching the clock source path to F_{FIG} (CLKSW bit (CLKCFG<4>) set to 0).
3. Insert a NOP instruction.
4. Switching the clock source to LIRC** (see note).
5. Insert a NOP instruction.
6. Switching the clock source path to F_{SEL} (CLKSW bit (CLKCFG<4>) set to 1).
7. Switching the clock source to internal clock (EXOSEN bit (CLKCFG<1>) set to 0).
8. Switching the clock source path to F_{FIG} (CLKSW bit (CLKCFG<4>) set to 0).
9. Insert a NOP instruction.
10. Specify new clock speed** (see note).
11. Insert a NOP instruction.
12. Switching the clock source path to F_{SEL} (CLKSW bit (CLKCFG<4>) set to 1).

Situation 3: **Switching from the an external clock to LIRC or Switching from the LIRC to an external clock**

Setting procedures are the same as 2.12.3 on page 56, please refer to the section setting step.

Note : 1. CLKCFG register prohibits direct write value, Need using BSR/BCR instruction.

2. Failure to follow set procedures could result in unexpected situation occurs.

2.13 Configuration Words

Table 2.10: Configuration Words

Name	Description
SELXOUT	Internal RC Output Frequency Selection Bit → XOUT= 16M$_{HZ}$ (default) → XOUT = 8M$_{HZ}$ → XOUT = 4M$_{HZ}$ → XOUT = 2M$_{HZ}$
INSWDY	Internal clock switch to External clock delay time Selection Bit → 128uS (default) → 4mS
ENWDT	Watchdog Timer Enable Bit → WDT enabled (default) → WDT disabled
WDTSEL	Watch dog time-out Selection Bit → 128mS → 512mS → 8mS → 32mS (default)
SELPOR	Power-on reset extend timing Selection Bit → 8mS → 64mS → 32mS → 16mS (default)
PROTECT	Code Protection Bit → NO, OTP code protection off (default) → YES, OTP code protection on
V$_{REFH}$	Reference voltage calibration center Selection Bit → 2V (default) → 3V → 1.5V

3.0 INSTRUCTION SET

Mnemonic, Operands		Description	Operation	Cycles	Status Affected
BCR	**R, bit**	Clear bit in R	0 → R<b>	1	-
BSR	**R, bit**	Set bit in R	1 → R<b>	1	-
BTRSC	**R, bit**	Test bit in R, Skip if Clear	Skip if R<b> = 0	1/2(1)	-
BTRSS	**R, bit**	Test bit in R, Skip if Set	Skip if R<b> = 1	1/2(1)	-
NOP		No Operation	No operation	1	-
CLRWDT		Clear Watchdog Timer	0x00 → WDT, 0x00 → WDT pre-scaler	1	$\overline{TO}$,$\overline{PD}$
SLEEP		Go into power-down mode	0x00 → WDT, 0x00 → WDT pre-scaler	1	$\overline{TO}$,$\overline{PD}$
RETURN		Return from subroutine	Top of Stack → PC	2	-
RETFIE		Return from interrupt, set GIE bit	Top of Stack → PC, 1 → GIE	2	-
IOST	**R**	Load IOST register	ACC → IOST register	1	-
IOSTR	**R**	Read IOST register	IOST register → ACC	1	-
CLRA		Clear ACC	0x00 → ACC	1	Z
CLRR	**R**	Clear R	0x00 → R	1	Z
MOVAR	**R**	Move ACC to R	ACC → R	1	-
MOVR	**R, d**	Move R	R → dest	1	Z
DECR	**R, d**	Decrement R	R - 1 → dest	1	Z
DECRSZ	**R, d**	Decrement R, Skip if 0	R - 1 → dest, Skip if result = 0	1/2(1)	-
INCR	**R, d**	Increment R	R + 1 → dest	1	Z
INCRSZ	**R, d**	Increment R, Skip if 0	R + 1 → dest, Skip if result = 0	1/2(1)	-
ADDAR	**R, d**	Add ACC and R	R + ACC → dest	1	C, DC, Z
SUBAR	**R, d**	Subtract ACC from R	R - ACC → dest	1	C, DC, Z
ADCAR	**R, d**	Add ACC and R with Carry	R + ACC + C → dest	1	C, DC, Z
SBCAR	**R, d**	Subtract ACC from R with Carry	R + $\overline{ACC}$ + C → dest	1	C, DC, Z
ANDAR	**R, d**	AND ACC with R	ACC and R → dest	1	Z
IORAR	**R, d**	Inclusive OR ACC with R	ACC or R → dest	1	Z
XORAR	**R, d**	Exclusive OR ACC with R	R xor ACC → dest	1	Z
COMR	**R, d**	Complement R	$\overline{R}$→ dest	1	Z
RLR	**R, d**	Rotate left R through Carry	R<7> → C, R<6:0> → dest<7:1>, C → dest<0>	1	C
RRR	**R, d**	Rotate right R through Carry	C → dest<7>, R<7:1> → dest<6:0>, R<0> → C	1	C
SWAPR	**R, d**	Swap R	R<3:0> → dest<7:4>, R<7:4> → dest<3:0>	1	-
MOVIA	**I**	Move Immediate to ACC	I → ACC	1	-
ADDIA	**I**	Add ACC and Immediate	I + ACC → ACC	1	C, DC, Z
SUBIA	**I**	Subtract ACC from Immediate	I - ACC → ACC	1	C, DC, Z
ANDIA	**I**	AND Immediate with ACC	ACC and I → ACC	1	Z
IORIA	**I**	OR Immediate with ACC	ACC or I → ACC	1	Z

Mnemonic, Operands		Description	Operation	Cycles	Status Affected
XORIA	**I**	Exclusive OR Immediate to ACC	ACC xor I → ACC	1	Z
RETIA	**I**	Return, place Immediate in ACC	I → ACC, Top of Stack → PC	2	-
CALL	**I**	Call subroutine	PC + 1 → Top of Stack, I → PC	2	-
GOTO	**I**	Unconditional branch	I → PC	2	-

Note: 1. 2 cycles for skip, else 1 cycle.
2. bit :Bit address within an 8-bit register R
R :Register address (0x00 to 0x7F)
I :Immediate data
ACC :Accumulator
d :Destination select;
=0 (store result in ACC)
=1 (store result in file register R)
dest :Destination
PC :Program Counter
WDT :Watchdog Timer Counter
GIE :Global interrupt enable bit
$\overline{TO}$:Time-out bit
$\overline{PD}$:Power-down bit
C :Carry bit
DC :Digital carry bit
Z :Zero bit

ADCAR	**Add ACC and R with Carry**
Syntax:	ADCAR R, d
Operands:	0x00≤R≤0x7F d∈[0,1]
Operation:	R + ACC + C → dest
Status Affected:	C, DC, Z
Description:	Add the contents of the ACC register and register 'R' with Carry. If 'd' is 0 the result is stored in the ACC register. If 'd' is '1' the result is stored back in register 'R'.
Cycles:	1

ADDAR	**Add ACC and R**
Syntax:	ADDAR R, d
Operands:	0x00≤R≤0x7F d∈[0,1]
Operation:	ACC + R → dest
Status Affected:	C, DC, Z
Description:	Add the contents of the ACC register and register 'R'. If 'd' is 0 the result is stored in the ACC register. If 'd' is '1' the result is stored back in register 'R'.
Cycles:	1

ADDIA	**Add ACC and Immediate**
Syntax:	ADDIA I
Operands:	0x00≤I≤0xFF
Operation:	ACC + I → ACC
Status Affected:	C, DC, Z
Description:	Add the contents of the ACC register with the 8-bit immediate 'I'. The result is placed in the ACC register.
Cycles:	1

ANDAR	**AND ACC and R**
Syntax:	ANDAR R, d
Operands:	0x00≤R≤0x7F d∈[0,1]
Operation:	ACC and R → dest
Status Affected:	Z
Description:	The contents of the ACC register are AND'ed with register 'R'. If 'd' is 0 the result is stored in the ACC register. If 'd' is '1' the result is stored back in register 'R'.
Cycles:	1

ANDIA	**AND Immediate with ACC**
Syntax:	ANDIA I
Operands:	0x00≤I≤0xFF
Operation:	ACC AND I → ACC
Status Affected:	Z
Description:	The contents of the ACC register are AND'ed with the 8-bit immediate 'I'. The result is placed in the ACC register.
Cycles:	1

BCR	**Clear Bit in R**
Syntax:	BCR R, b
Operands:	0x00≤R≤0x7F 0x0≤b≤0x7
Operation:	0 → R<b>
Status Affected:	None
Description:	Clear bit 'b' in register 'R'.
Cycles:	1

BSR	**Set Bit in R**
Syntax:	BSR R, b
Operands:	0x000≤R≤0x1FF 0x0≤b≤0x7
Operation:	1 → R<b>
Status Affected:	None
Description:	Set bit 'b' in register 'R'.
Cycles:	1

BTRSC	**Test Bit in R, Skip if Clear**
Syntax:	BTRSC R, b
Operands:	0x000≤R≤0x1FF 0x0≤b≤0x7
Operation:	Skip if R<b> = 0
Status Affected:	None
Description:	If bit 'b' in register 'R' is 0 then the next instruction is skipped. If bit 'b' is 0 then next instruction fetched during the current instruction execution is discarded, and a NOP is executed instead making this a 2-cycle instruction.
Cycles:	1/2

BTRSS	**Test Bit in R, Skip if Set**
Syntax:	BTRSS R, b
Operands:	0x000≤R≤0x1FF 0x0≤b≤0x7
Operation:	Skip if R<b> = 1
Status Affected:	None
Description:	If bit 'b' in register 'R' is '1' then the next instruction is skipped. If bit 'b' is '1', then the next instruction fetched during the current instruction execution, is discarded and a NOP is executed instead, making this a 2-cycle instruction.
Cycles:	1/2

CALL	**Subroutine Call**
Syntax:	CALL I
Operands:	0x000≤I≤0x7EF
Operation:	PC + 1 → Top of Stack, I → PC<12:0>
Status Affected:	None
Description:	Subroutine call. First, return address (PC+1) is pushed onto the stack. The 11-bit immediate address is loaded into PC bits <10:0>.
Cycles:	2

CLRA	Clear ACC
Syntax:	CLRA
Operands:	None
Operation:	0x00 → ACC; 1 → Z
Status Affected:	Z
Description:	The ACC register is cleared. Zero bit (Z) is set.
Cycles:	1

CLRR	Clear R
Syntax:	CLRR R
Operands:	0x00≤R≤0x7F
Operation:	0x00 → R; 1 → Z
Status Affected:	Z
Description:	The contents of register 'R' are cleared and the Z bit is set.
Cycles:	1

CLRWDT	Clear Watchdog Timer
Syntax:	CLRWDT
Operands:	None
Operation:	0x00 → WDT; 0x00 → WDT pre-scaler 1 → $\overline{TO}$; 1 → $\overline{PD}$
Status Affected:	$\overline{TO}$, $\overline{PD}$
Description:	The CLRWDT instruction resets the WDT. It also resets the pre-scaler and cleared internal WDT wake-up flag. Status bits $\overline{TO}$ and $\overline{PD}$ are set.
Cycles:	1

COMR	Complement R
Syntax:	COMR R, d
Operands:	0x00≤R≤0x7F d∈[0,1]
Operation:	$\overline{R}$→ dest
Status Affected:	Z
Description:	The contents of register 'R' are complemented. If 'd' is 0 the result is stored in the ACC register. If 'd' is 1 the result is stored back in register 'R'.
Cycles:	1

DECR	Decrement R
Syntax:	DECR R, d
Operands:	0x00≤R≤0x7F d∈[0,1]
Operation:	R - 1 → dest
Status Affected:	Z
Description:	Decrement of register 'R'. If 'd' is 0 the result is stored in the ACC register. If 'd' is 1 the result is stored back in register 'R'.
Cycles:	1

DECRSZ	Decrement R, Skip if 0
Syntax:	DECRSZ R, d
Operands:	0x00≤R≤0x7F d∈[0,1]
Operation:	R - 1 → dest; skip if result =0
Status Affected:	None
Description:	The contents of register 'R' are decrement. If 'd' is 0 the result is placed in the ACC register. If 'd' is 1 the result is stored back in register 'R'. If the result is 0, the next instruction, which is already fetched, is discarded and a NOP is executed instead and making it a two-cycle instruction.
Cycles:	1/2

GOTO	Unconditional Branch
Syntax:	GOTO I
Operands:	0x000≤I≤0x7EF
Operation:	I → PC<12:0>
Status Affected:	None
Description:	GOTO is an unconditional branch. The 11-bit immediate value is loaded into PC bits <10:0>.
Cycles:	2

INCR	Increment R
Syntax:	INCR R, d
Operands:	0x00≤R≤0x7F d∈[0,1]
Operation:	R + 1 → dest
Status Affected:	Z
Description:	The contents of register 'R' are increment. If 'd' is 0 the result is placed in the ACC register. If 'd' is 1 the result is stored back in register 'R'.
Cycles:	1

INCRSZ	Increment R, Skip if 0
Syntax:	INCRSZ R, d
Operands:	0x00≤R≤0x7F d∈ [0,1]
Operation:	R + 1 → dest, skip if result = 0
Status Affected:	None
Description:	The contents of register 'R' are increment. If 'd' is 0 the result is placed in the ACC register. If 'd' is the result is stored back in register 'R'. If the result is 0, then the next instruction, which is already fetched, is discarded and a NOP is executed instead and making it a two-cycle instruction.
Cycles:	1/2

IORAR	OR ACC with R
Syntax:	IORAR R, d
Operands:	0x00≤R≤0x7F d∈[0,1]
Operation:	ACC or R → dest
Status Affected:	Z
Description:	Inclusive OR the ACC register with register 'R'. If 'd' is 0 the result is placed in the ACC register. If 'd' is 1 the result is placed back in register 'R'.
Cycles:	1

IORIA	OR Immediate with ACC
Syntax:	IORIA I
Operands:	0x00≤I≤0x7F
Operation:	ACC or I → ACC
Status Affected:	Z
Description:	The contents of the ACC register are OR'ed with the 8-bit immediate 'I'. The result is placed in the ACC register.
Cycles:	1

IOST	Load IOST Register
Syntax:	IOST R
Operands:	R = 0x08, 0x09, 0x17, 0x19 or 0x23
Operation:	ACC → IOST register R
Status Affected:	None
Description:	IOST register 'R' (R= 0x08, 0x09, 0x17, 0x19 or 0x23) is loaded with the contents of the ACC register.
Cycles:	1

IOSTR	Read IOST Register
Syntax:	IOST R
Operands:	R = 0x08, 0x09, 0x17, 0x19 or 0x23
Operation:	IOST register R → ACC
Status Affected:	None
Description:	The ACC register is loaded with the contents of IOST register 'R' (0x08, 0x09, 0x17, 0x19 or 0x23).
Cycles:	1

MOVAR	Move ACC to R
Syntax:	MOVAR R
Operands:	0x00≤R≤0x7F
Operation:	ACC → R
Status Affected:	None
Description:	Move data from the ACC register to register 'R'.
Cycles:	1

MOVIA	Move Immediate to ACC
Syntax:	MOVIA I
Operands:	0x00≤I≤0xFF
Operation:	I → ACC
Status Affected:	None
Description:	The 8-bit immediate 'I' is loaded into the ACC register. The don't cares will assemble as 0s.
Cycles:	1

MOVR	Move R
Syntax:	MOVR R, d
Operands:	0x00≤R≤0x7F d∈[0,1]
Operation:	R → dest
Status Affected:	Z
Description:	The contents of register 'R' is moved to destination 'd'. If 'd' is 0, destination is the ACC register. If 'd' is 1, the destination is file register 'R'. 'd' is 1 is useful to test a file register since status flag Z is affected.
Cycles:	1

NOP	No Operation
Syntax:	NOP
Operands:	None
Operation:	No operation
Status Affected:	None
Description:	No operation.
Cycles:	1

RETFIE	Return from Interrupt, Set 'GIE' Bit
Syntax:	RETFIE
Operands:	None
Operation:	Top of Stack → PC 1 → GIE
Status Affected:	None
Description:	The program counter is loaded from the top of the stack (the return address). The 'GIE' bit is set to 1. This is a two-cycle instruction.
Cycles:	2

RETIA	Return with Immediate in ACC
Syntax:	RETIA I
Operands:	0x00≤I≤0xFF
Operation:	I → ACC; Top of Stack → PC
Status Affected:	None
Description:	The ACC register is loaded with the 8-bit immediate 'I'. The program counter is loaded from the top of the stack (the return address). This is a two-cycle instruction.
Cycles:	2

RETURN	Return from Subroutine
Syntax:	RETURN
Operands:	None
Operation:	Top of Stack → PC
Status Affected:	None
Description:	The program counter is loaded from the top of the stack (the return address). This is a two-cycle instruction.
Cycles:	2

RLR	Rotate Left R through Carry
Syntax:	RLR R, d
Operands:	0x00≤R≤0x7F d∈[0,1]
Operation:	R<7> → C; R<6:0> → dest<7:1>; C → dest<0>
Status Affected:	C
Description:	The contents of register 'R' are rotated left one bit to the left through the Carry Flag. If 'd' is 0 the result is placed in the ACC register. If 'd' is 1 the result is stored back in register 'R'.
Cycles:	1

RRR	**Rotate Right R through Carry**
Syntax:	RRR R, d
Operands:	0x00≤R≤0x7F d∈[0,1]
Operation:	C → dest<7>; R<7:1> → dest<6:0>; R<0> → C
Status Affected:	C
Description:	The contents of register 'R' are rotated one bit to the right through the Carry Flag. If 'd' is 0 the result is placed in the ACC register. If 'd' is 1 the result is placed back in register 'R'.
Cycles:	1

SLEEP	**Enter Suspend Mode**
Syntax:	SLEEP
Operands:	None
Operation:	0x00 → WDT; 0x00 → WDT prescaler; 1 → $\overline{TO}$; 0 → $\overline{PD}$
Status Affected:	$\overline{TO}$, $\overline{PD}$
Description:	Time-out status bit ($\overline{TO}$) is set. The power-down status bit ($\overline{PD}$) is cleared. The WDT is cleared. The processor is put into Suspend mode.
Cycles:	1

SBCAR	**Subtract ACC from R with Carry**
Syntax:	SBCAR R, d
Operands:	0x00≤R≤0x7F d∈[0,1]
Operation:	R + $\overline{ACC}$ + C → dest
Status Affected:	C, DC, Z
Description:	Add the 2's complement data of the ACC register from register 'R' with Carry. If 'd' is 0 the result is stored in the ACC register. If 'd' is 1 the result is stored back in register 'R'.
Cycles:	1

SUBAR	**Subtract ACC from R**
Syntax:	SUBAR R, d
Operands:	0x00≤R≤0x7F d∈[0,1]
Operation:	R - ACC → dest
Status Affected:	C, DC, Z
Description:	Subtract (2's complement method) the ACC register from register 'R'. If 'd' is 0 the result is stored in the ACC register. If 'd' is 1 the result is stored back in register 'R'.
Cycles:	1

SUBIA	**Subtract ACC from Immediate**
Syntax:	SUBIA I
Operands:	0x00≤I≤0xFF
Operation:	I - ACC → ACC
Status Affected:	C, DC, Z
Description:	Subtract (2's complement method) the ACC register from the 8-bit immediate 'I'. The result is placed in the ACC register.
Cycles:	1

SWAPR	**Swap nibbles in R**
Syntax:	SWAPR R, d
Operands:	0x00≤R≤0x7F d∈[0,1]
Operation:	R<3:0> → dest<7:4>; R<7:4> → dest<3:0>
Status Affected:	None
Description:	The upper and lower nibbles of register 'R' are exchanged. If 'd' is 0 the result is placed in ACC register. If 'd' is 1 the result in placed in register 'R'.
Cycles:	1

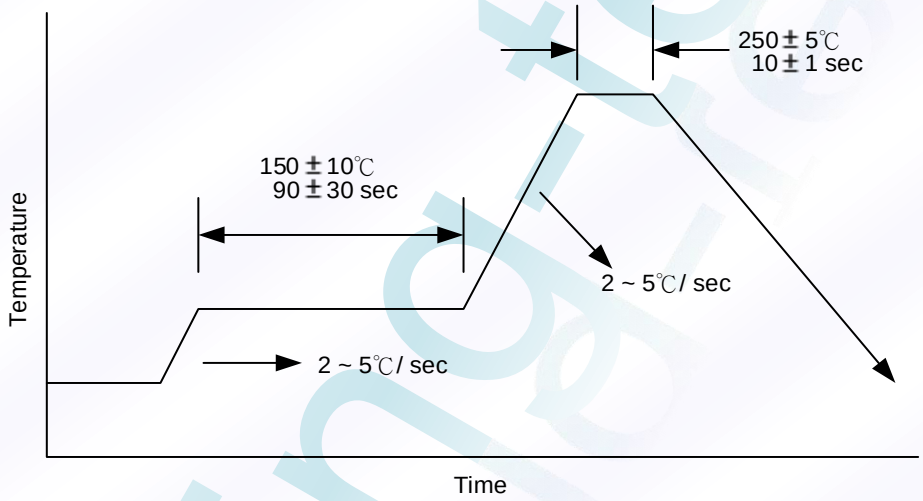
XORAR	Exclusive OR ACC with R
Syntax:	XORAR R, d
Operands:	0x00≤R≤0x7F d∈[0,1]
Operation:	ACC xor R → dest
Status Affected:	Z
Description:	Exclusive OR the contents of the ACC register with register 'R'. If 'd' is 0 the result is stored in the ACC register. If 'd' is 1 the result is stored back in register 'R'.
Cycles:	1

XORIA	**Exclusive OR Immediate with ACC**
Syntax:	XORIA I
Operands:	0x00≤I≤0xFF
Operation:	ACC xor I → ACC
Status Affected:	Z
Description:	The contents of the ACC register are XOR'ed with the 8-bit immediate 'I'. The result is placed in the ACC register.
Cycles:	1

4.0 ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
	Ambient Operating Temperature	-	-10	-	85	°C
	Store Temperature	-	-40	-	150	°C
V_{DD}	DC Supply Voltage	-	2.0	-	5.5	V
	Input Voltage with respect to Ground	-	-0.3	-	V_{DD}+0.3	V
	ESD Susceptibility (Standard)	HBM (Human Body Mode)	-	5	-	KV
		MM (Machine Mode)	-	350	-	V
	Lead Temperature	Soldering, 10 Sec	-	-	250	°C

4.1 PACKAGE IR Re-flow Soldering Curve



5.0 RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
V_{DD}	DC Supply Voltage	-	2.0	-	5.5	V
	Operating Temperature	-	0	-	70	°C

6.0 ELECTRICAL CHARACTERISTICS

6.1 AC Characteristics

Ta=25°C

Symbol	Description	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
F_{HIRC}	High Frequency IRC	-	V_{DD} = 2.2V ~ 5V	-2%	16	+2%	MHZ
F_{LIRC}	Low Frequency IRC	-	V_{DD} = 2.2V ~ 5V	-10%	32	+10%	KHZ
T_{POREXT}	Power-on extern time	3V	SELPOR = 16mS	13.5	16	20	mS
		5V		13.5	16	20	

Note: At any time, a 0.1μF decoupling capacitor should be connected between V_{DD} and V_{SS} and device as close as possible.

6.2 DC Characteristics

Ta=25°C

Symbol	Description	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{IH}	Input high voltage, I/O Ports (Without PA1 Pin)	3V	-	-	$0.7V_{DD}$	V_{DD}	V
		5V		-		V_{DD}	
	Input high voltage, PA1 Pin	3V	-	-	$0.68V_{DD}$	V_{DD}	
		5V		-		V_{DD}	
V_{IL}	Input low voltage, I/O Ports	3V	-	V_{SS}	$0.3V_{DD}$	-	V
		5V				-	
V_{LVDT}	Low voltage detect	-	LVDT30	-	3.0	-	V
		-	LVDT24	-	2.4	-	
		-	LVDT20	-	2.0	-	
		-	LVDT18	-	1.8	-	
V_{POR}	Power-on Reset voltage	-	Power from 0V to V_{DD}	1.6	-	1.8	V
V_{PDR}	Power-down Reset voltage	-	Power from V_{DD} to 0V	1.5	-	1.8	V
V_{PDRS}	Power-down Reset voltage in Suspend mode	-	Power from V_{DD} to 0V	-	1.3	-	V
V_{REFH}	Internal reference voltage	5V	V_{REFH} = 1.5V	-2%	1.5	+2%	V
		5V	V_{REFH} = 2V	-2%	2.0	+2%	
		5V	V_{REFH} = 3V	-2%	3.0	+2%	
VF_{CPU}	Operating voltage	-	F_{CPU} = 4MHZ	1.9	-	V_{DD}	V
		-	F_{CPU} = 8MHZ	2.0	-	V_{DD}	
		-	F_{CPU} = 16MHZ	2.4	-	V_{DD}	
I_{OH}	I/O Ports Drive current (Without PA1 Pin)	2V	V_{OH}=0.9V_{DD}	-	4.6	-	mA
		3V		-	11	-	
		5V		15	27.1	-	
I_{OL}	I/O Ports Sink current (Without PA1 Pin)	2V	V_{OL}=0.1V_{DD}	-	5.6	-	mA
		3V		-	13.1	-	
		5V		15	30.8	-	
	PA1 Pin Sink current	2V	V_{OL}=0.1V_{DD}	-	1.1	-	
		3V		-	2.2	-	
		5V		-	4.3	-	
I_{PH}	I/O Ports Pull-high current (Without PA1 Pin)	3V	Input pin at V_{SS}	-	32.9	-	uA
		5V		41.7	54.9	-	
	PA1 Pin Pull-high current	3V	Input pin at V_{SS}	-	20.2	-	
		5V		-	61.5	-	

Symbol	Description	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
I_{DD1}	Normal mode Operating current	2V	F_{CPU} = 1M$_{HZ}$, No load Disable ADC, WDT	-	0.5	-	mA
		5V		-	1.1	-	
		5V	F_{CPU} = 4M$_{HZ}$, No load Disable ADC, WDT	-	1.8	-	
		5V	F_{CPU} = 16M$_{HZ}$, No load Disable ADC, WDT	-	2.5	-	
I_{DD2}	Slow mode Operating current	3V	F_{CPU} = 32K$_{HZ}$, No load Disable ADC, LVDT	-	380	-	uA
		5V		-	550	-	
I_{DD3}	Green mode Operating current	5V	F_{CPU} = Stop, No load Disable ADC, LVDT WDT = 4096mS	-	45	-	uA
	Ultra Green mode Operating current	3V	F_{CPU} = Stop, No load Disable ADC, LVDT WDT = 256mS	-	8	-	
		5V		-	17	-	
I_{DD4}	Suspend (Power-down) mode current	2V	F_{CPU} = Stop, No load Disable ADC, LVDT, WDT	-	<1	-	uA
		5V		-	<1	<2	

6.3 A/D Characteristics

Ta=25°C

Symbol	Description	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{ADC}	ADC operation voltage	-	V_{REFH} = Int 1.5V/2V/3V	2.1	-	5.5	V
DNL	Differential Nonlinearity	5V	Resolution = 10 bit, V_{REFH} = 3V, F_{ADCSR} = 32K$_{HZ}$	-	-	-	LSB
INL	Integral Nonlinearity	5V	Resolution = 10 bit, V_{REFH} = 3V, F_{ADCSR} = 32K$_{HZ}$	-	-	-	LSB
GE	Gain Error	5V	Resolution = 10 bit, V_{REFH} = 3V, F_{ADCSR} = 32K$_{HZ}$	-	-	-	LSB
R_{AD}	Resolution	5V	V_{REFH} = Int 1.5V/2V/3V F_{ADCSR} = 32K$_{HZ}$	8	9	10	Bit
T_{ADC}	AD Conversion time AIN Pins (AIN1 ~ AIN4)	5V	Toggle mode 0	8	-	-	uS
			Toggle mode 2	9	-	-	
	AD Conversion time V_{DD}/4 (AIN5)	5V	Toggle mode 0	34	-	-	
			Toggle mode 2	36	-	-	

6.4 D/A Characteristics

Ta=25°C

Symbol	Description	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
T_{DAC}	DA Conversion time	5V	Toggle mode 1 & 3	1.75	-	-	uS

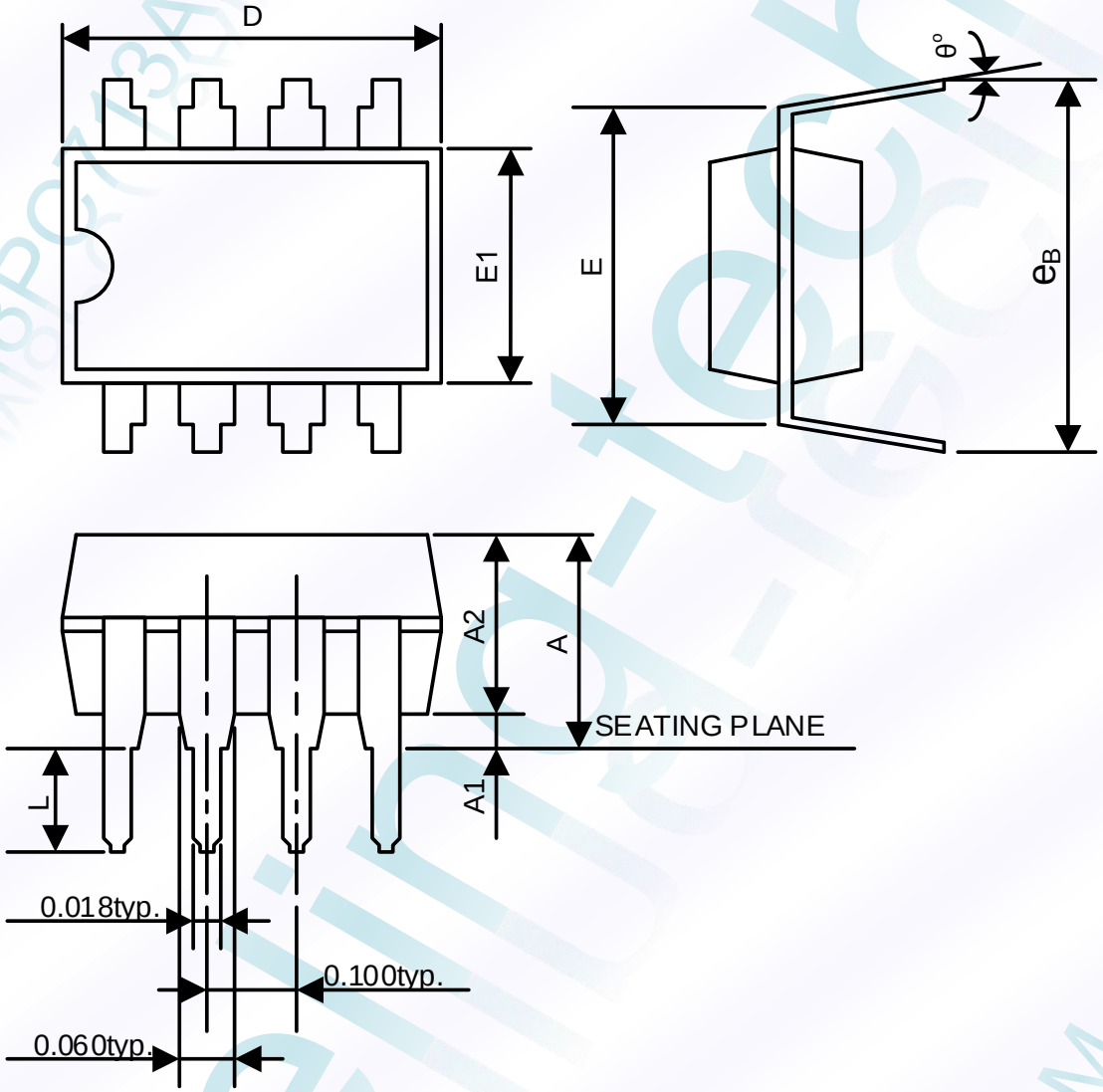
6.5 Comparator Characteristics

Ta=25°C

Symbol	Description	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{IO}	Comparator input offset voltage	-	V_{DD}=2.1V~5V	-	±15	-	mV
V_{ICM}	Comparator input common mode voltage	-	V_{DD}=2.1V~5V	0	-	V_{DD}	V

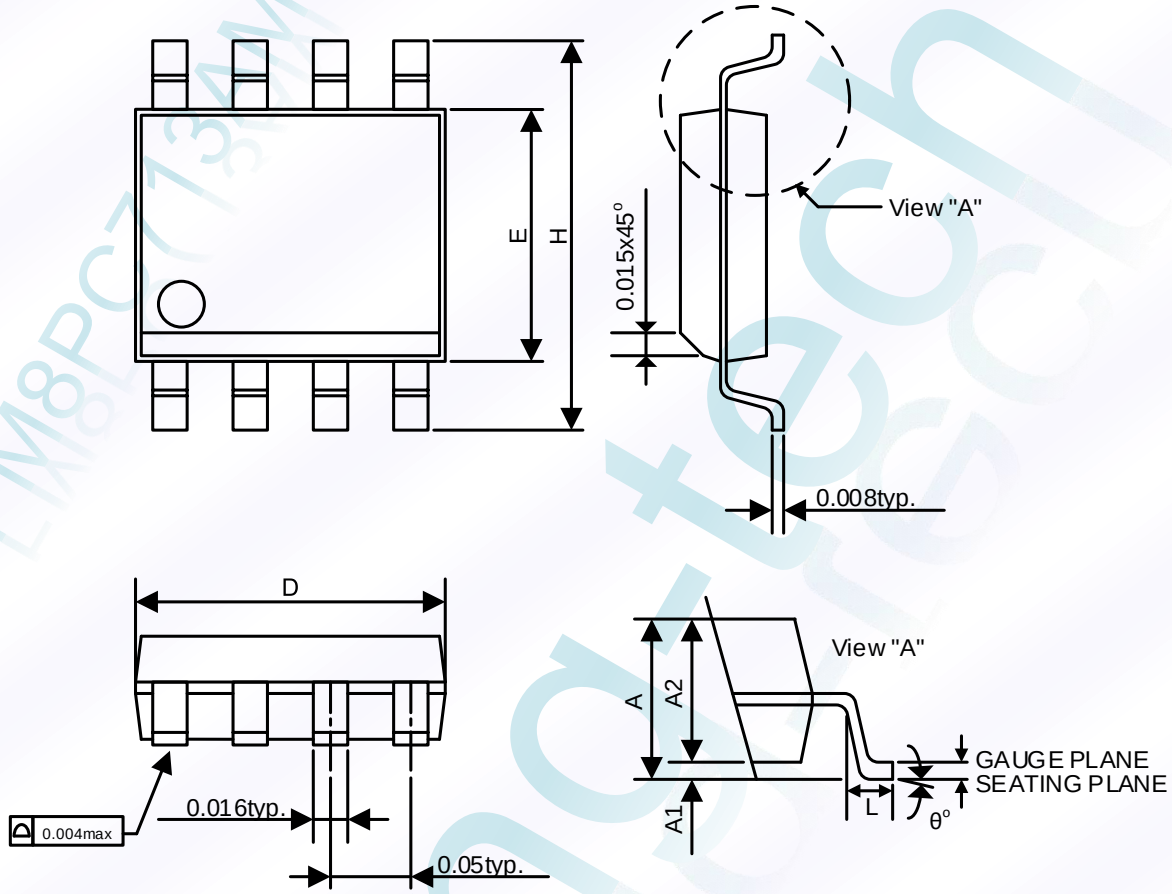
7.0 PACKAGE DIMENSION

7.1 8-PIN PDIP 300mil



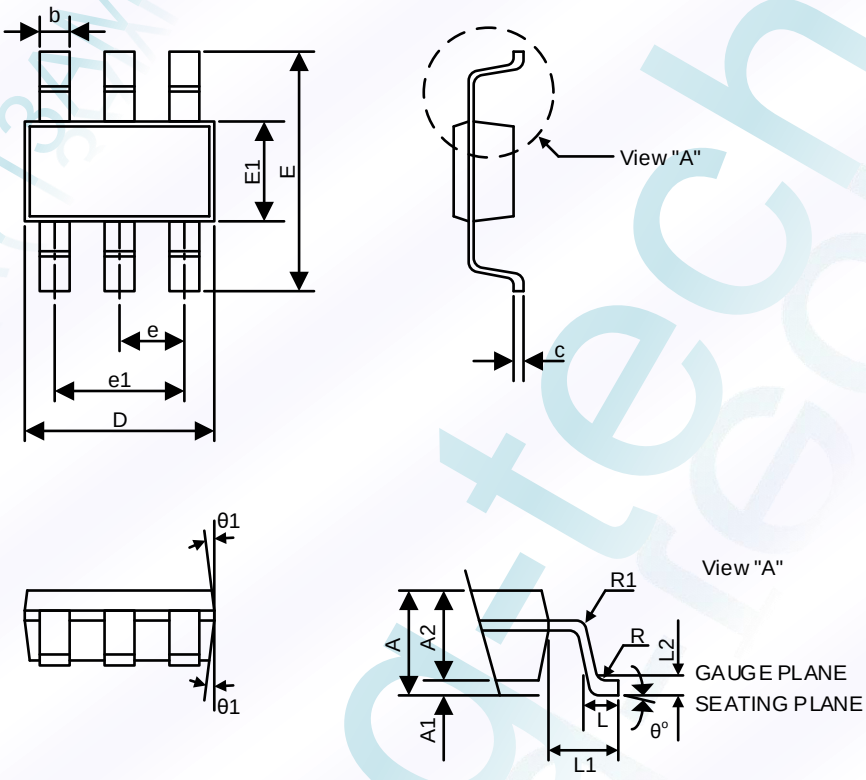
Symbols	Dimension In Inches		
	Min	Nom	Max
A	-	-	0.210
A1	0.015	-	-
A2	0.125	0.130	0.135
D	0.355	0.365	0.400
E		0.300 BSC.	
E1	0.245	0.250	0.255
L	0.115	0.130	0.150
eB	0.335	0.355	0.375
θ°	0°	7°	15°

7.2 8-PIN SOP 150mil



Symbols	Dimension In Inches		
	Min	Nom	Max
A	0.053	-	0.069
A1	0.004	-	0.010
A2	-	-	0.059
D	0.189	-	0.196
E	0.150	-	0.157
H	0.228	-	0.244
L	0.016	-	0.050
θ	0°	-	8°

7.3 6-PIN SOT23-6



Symbols	Dimension In MM		
	Min	Nom	Max
A	-	-	1.45
A1	-	-	0.15
A2	0.90	1.15	1.30
b	0.30	-	0.50
c	0.08	-	0.22
D		2.90 BSC.	
E		2.80 BSC.	
E1		1.60 BSC.	
e		0.95 BSC.	
e1		1.90 BSC.	
L	0.30	0.45	0.60
L1		0.60 REF.	
L2		0.25 BSC.	
R	0.10	-	-
R1	0.10	-	0.25
θ	0°	4°	8°
θ1	5°	10°	15°

8.0 ORDERING INFORMATION

Note that less than 8-pin MCU package types are not marketed in the following countries: USA, UK, Germany, The Netherlands, France and Italy.

OTP Type MCU	Package Type	Pin Count	Package Size	SAMPLE Stock
FM8PC713AMAEP	PDIP	8	300 mil	Available
FM8PC713AMAED	SOP	8	150 mil	Available
FM8PC713AMAEL	SOT23-6	6	-	Available