FEELING TECHNOLOGY

FM8PE581M

OTP-Based 8-Bit Microcontroller

Devices Included in this Data Sheet:

- FM8PE581MA: 14-pin OTP device
- FM8PE581MB: 16-pin OTP device
- FM8PE581MC: 18-pin OTP device

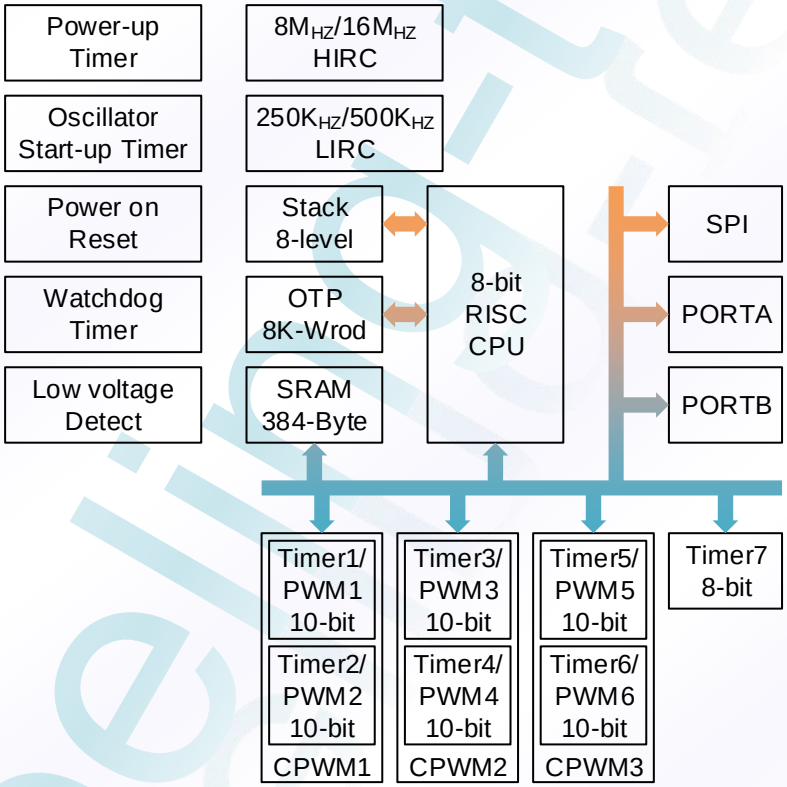
FEATURES

- 8K word on chip OTP-ROM and 56 bytes on chip special purpose registers and 384 bytes on chip general purpose registers (SRAM).
- 8-level deep hardware stack.
- 7 real time down-count Timer/Counter with 3-bit programmable pre-scaler:
  - TMR1: 10-bit (8+2), PWM1 & CPWM1 Period & Timer1.
  - TMR2: 10-bit (8+2), PWM2 & CPWM1 Duty & Timer2.
  - TMR3: 10-bit (8+2), PWM3 & CPWM2 Period & Timer3.
  - TMR4: 10-bit (8+2), PWM4 & CPWM2 Duty & Timer4.
  - TMR5: 10-bit (8+2), PWM5 & CPWM3 Period & Timer5.
  - TMR6: 10-bit (8+2), PWM6 & CPWM3 Duty & Timer6.
  - TMR7: 8-bit, Timer7.
- Serial Peripheral Interface (SPI); Three-wire (or Two wire) synchronous communication.
- Clock output: F_{OSC}/2, /4, /8, /16, /32, /64 clock output, 3/4 duty clock output.
- Built-in 5 levels Low Voltage Detector (LVDT) for Brown-out Reset (BOR).
- Power-up Reset Timer (PWRT).
- On chip Watchdog Timer (WDT) with internal oscillator for reliable operation.
- Two I/O ports PORTA and PORTB with independent direction control:
  - 15 Bi-direction I/O port (Programmable Pull-up enable in Input mode).
  - One open-drain port (IOB3/RSTB).
- Five kinds of interrupt source: 7 Timers/Counters, 6 external interrupt sources: IOA5~IOA7, IOB3~IOB5, SPI interface, Internal watchdog timer (i_WDT) wakeup.
- Wake-up from SLEEP:
  - PORTA (IO5~IOA7) and PORTB (IOB3~IOB5) pin change wakeup.
  - WDT overflow.
  - i_WDT overflow.
- HALT function reduce power consumption.
- Power saving SLEEP mode.
- Configurable CPU clock per instruction cycle: F_{OSC}/4 and F_{OSC}/2.
- All instructions are single cycle except for program branches which are two-cycles.
- Programmable Code Protection.
- Six selectable oscillator options:
  - ERC, XT, LF, 16M$_{HZ}$/8M$_{HZ}$ HIRC, 500K$_{HZ}$/250K$_{HZ}$ LIRC.
- Wide-operating voltage range:
  - OTP: 3.3V to 5.5V @ 8MIPS, 2.0V to 5.5V @ 4MIPS.

GENERAL DESCRIPTION

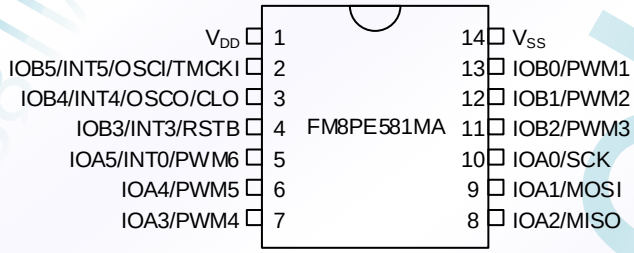
The FM8PE581M is a high noise immunity, OTP-based 8-bit CMOS microcontrollers. It employs a RISC architecture with 54 instructions. All instructions are single cycle except for program branches which take two cycles. The easy to use and easy to remember instruction set reduces development time significantly.
The FM8PE581M consists of Power-on Reset (POR), Brown-out Reset (BOR), Power-up Reset Timer (PWRT), Watchdog Timer, OTP, SRAM, tristate I/O port, I/O pull-high control, Power saving SLEEP mode, 7 real time programmable clock/counter, Interrupt, Wake-up from SLEEP mode, and Code Protection for OTP products. There are eight oscillator configurations to be chosen from, including the power-saving LF (Low Frequency) oscillator and cost saving internal RC oscillator.
The FM8PE581M address 8K of program memory.
The FM8PE581M can directly or indirectly address its register files and data memory. All special function registers including the program counter are mapped in the data memory.
The FM8PE581M provides up to 6 channel PWM output.

BLOCK DIAGRAM

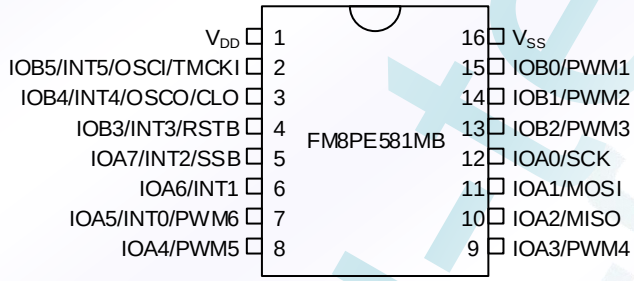


PIN CONNECTION

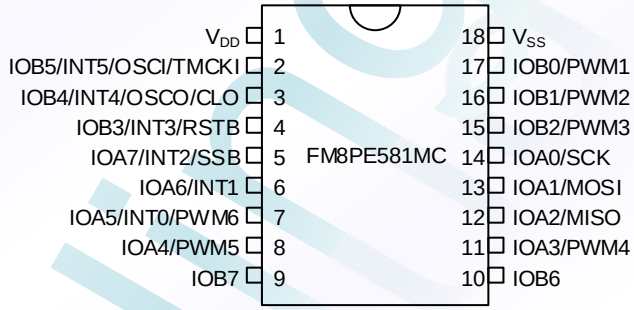
PDIP14, SOP14



PDIP16, SOP16



PDIP18, SOP18



PIN DESCRIPTIONS

Name	I/O	Description
IOA0/SCK	I/O	• Bi-direction I/O port (programmable Pull-high in Input mode). • SPI clock.
IOA1/MOSI	I/O	• Bi-direction I/O port (programmable Pull-high in Input mode). • SPI Master output / Slave input.
IOA2/MISO	I/O	• Bi-direction I/O port (programmable Pull-high in Input mode). • SPI Master input / Slave output.
IOA3/PWM4	I/O	• Bi-direction I/O port (programmable Pull-high in Input mode). • PWM4 output.
IOA4/PWM5	I/O	• Bi-direction I/O port (programmable Pull-high in Input mode). • PWM5 output.
IOA5/INT0/PWM6	I/O	• Bi-direction I/O port with system wake-up/pin change interrupt function (programmable Pull-high in Input mode). • PWM6 output.
IOA6/INT1	I/O	• Bi-direction I/O port (programmable Pull-high in Input mode).
IOA7/INT2/SSB	I/O	• Bi-direction I/O port (programmable Pull-high in Input mode). • SPI Slave select.
IOB0/PWM1	I/O	• Bi-direction I/O port (programmable Pull-high in Input mode). • PWM1 output.
IOB1/PWM2	I/O	• Bi-direction I/O port (programmable Pull-high in Input mode). • PWM2 output.
IOB2/PWM3	I/O	• Bi-direction I/O port (programmable Pull-high in Input mode). • PWM3 output.
IOB3/INT3/RSTB	I/O	• Input pin only with system wake-up/pin change interrupt function; **voltage on this pin must not exceed V_{DD}.** • System clear (RESET) input. This pin is an active low RESET to the device • Open-Drain output.
IOB4/INT4/CLO /OCSO	I/O	• Bi-direction I/O port with system wake-up/pin change interrupt function (programmable Pull-high in Input mode). • Software controlled pull-high. • Clock output with pre-scaler shared with IOB4. • Oscillator output (XT, LF, ERC mode).
IOB5/INT5/OSCI /TMCKI	I/O	• Bi-direction I/O port with system wake-up/pin change interrupt function (programmable Pull-high in Input mode). • Software controlled pull-high. • Oscillator input (XT, LF, ERC mode). • Timer external clock input shared with IOB5.
IOB6, IOB7	I/O	• Bi-direction I/O port (programmable Pull-high in Input mode).
V_{DD}	-	Positive supply.
V_{SS}	-	Ground.

Legend: I=input, O=output, I/O=input/output
Note: Please refer to 2.2 for detail IO type description.

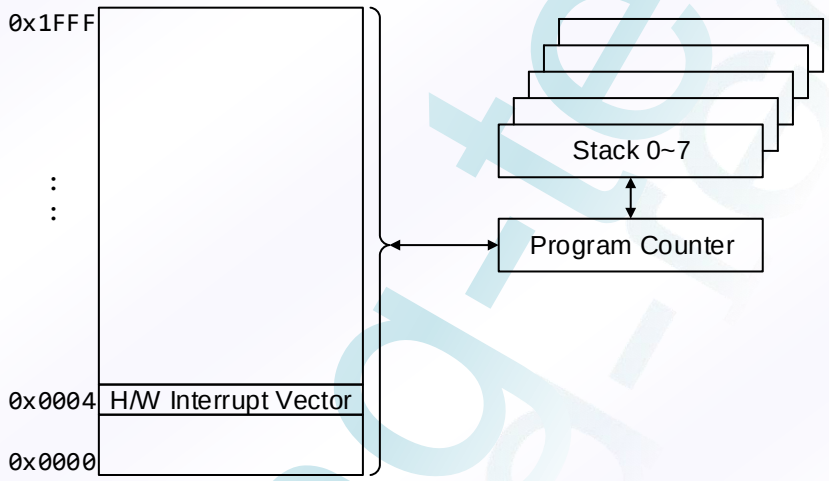
1.0 MEMORY ORGANIZATION

FM8PE581M memory is organized into program memory and data memory.

1.1 Program Memory Organization

The FM8PE581M has a 13-bit Program Counter capable of addressing an 8K program memory space.
The RESET vector for the FM8PE581M is at 0x000.
The H/W interrupt vector is at 0x004.
User can use “CALL (far call) / GOTO (far goto)” instructions to program user's code within entire program area.

Figure 1.1: Program Memory Map and STACK



1.2 Data Memory Organization

Data memory is composed of 56 bytes Special Function Registers and 384 bytes General Purpose Registers. The data memory can be accessed either directly or indirectly through the FSRH and FSRL registers.

Table 1.1: Registers File Map for FM8PE581M

Address	Description
0x000 : 0x034	Special Purpose Register
0x080 : 0x1FF	General Purpose Register

Table 1.2: Special Purpose Registers Map

Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
System									
0x000 (r/w)	INDF	Uses contents of FSRH and FSRL to address data memory (not a physical register)							
0x002 (r/w)	PCL	Low order 8 bits of PC							
0x003 (r/w)	STATUS	-	-	-	$\overline{TO}$	$\overline{PD}$	Z	DC	C
0x004 (r/w)	FSRL	Indirect data memory address pointer low byte							
0x005 (r/w)	FSRH	-	-	-	-	-	-	-	B8
0x006 (r/w)	PCHBUF	0	0	0	Upper 5 MSBs Buffer of PC				
IO PAD & CONTROL									
0x007 (r/w)	IOSTA	IOSTA7	IOSTA6	IOSTA5	IOSTA4	IOSTA3	IOSTA2	IOSTA1	IOSTA0
0x008 (r/w)	PORTA	IOA7	IOA6	IOA5	IOA4	IOA3	IOA2	IOA1	IOA0
0x009 (r/w)	IOSTB	IOSTB7	IOSTB6	IOSTB5	IOSTB4	IOSTB3	IOSTB2	IOSTB1	IOSTB0
0x00A (r/w)	PORTB	IOB7	IOB6	IOB5	IOB4	IOB3	IOB2	IOB1	IOB0
Timer1: 10-bit timer & PWM1									
0x00B (r/w)	T1CON	T1EN	-	T1CS1	T1CS0	T1EDG	T1PS2	T1PS1	T1PS0
0x00C (r/w)	P1CON	T1MOD	P1INV	EPWM1	CPWM1	PIR13	PIR12	PIR11	PIR10
0x00D (r/w)	T1RLLB	10-bit real-time timer/counter reload buffer low byte							
0x00E (r/w)	T1RLHB	-	-	-	-	-	-	D9	D8
Timer2: 10-bit Timer & PWM2									
0x00F (r/w)	T2CON	T2EN	-	T2CS1	T2CS0	T2EDG	T2PS2	T2PS1	T2PS0
0x010 (r/w)	P2CON	T2MOD	P2INV	EPWM2	-	PIR23	PIR22	PIR21	PIR20
0x011 (r/w)	T2RLLB	10-bit real-time timer/counter reload buffer low byte							
0x012 (r/w)	T2RLHB	-	-	-	-	-	-	D9	D8
Timer3: 10-bit Timer & PWM3									
0x013 (r/w)	T3CON	T3EN	-	T3CS1	T3CS0	T3EDG	T3PS2	T3PS1	T3PS0
0x014 (r/w)	P3CON	T3MOD	P3INV	EPWM3	CPWM2	PIR33	PIR32	PIR31	PIR30
0x015 (r/w)	T3RLLB	10-bit real-time timer/counter reload buffer low byte							
0x016 (r/w)	T3RLHB	-	-	-	-	-	-	D9	D8
Timer4: 10-bit Timer & PWM4									
0x017 (r/w)	T4CON	T4EN	-	T4CS1	T4CS0	T4EDG	T4PS2	T4PS1	T4PS0
0x018 (r/w)	P4CON	T4MOD	P4INV	EPWM4	-	PIR43	PIR42	PIR41	PIR40
0x019 (r/w)	T4RLLB	10-bit real-time timer/counter reload buffer low byte							
0x01A (r/w)	T4RLHB	-	-	-	-	-	-	D9	D8

Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
Timer5: 10-bit Timer & PWM5									
0x01B (r/w)	T5CON	T5EN	-	T5CS1	T5CS0	T5EDG	T5PS2	T5PS1	T5PS0
0x01C (r/w)	P5CON	T5MOD	P5INV	EPWM5	CPWM3	PIR53	PIR52	PIR51	PIR50
0x01D (r/w)	T5RLLB	10-bit real-time timer/counter reload buffer low byte							
0x01E (r/w)	T5RLHB	-	-	-	-	-	-	D9	D8
Timer6: 10-bit Timer & PWM6									
0x01F (r/w)	T6CON	T6EN	-	T6CS1	T6CS0	T6EDG	T6PS2	T6PS1	T6PS0
0x020 (r/w)	P6CON	T6MOD	P6INV	EPWM6	-	PIR63	PIR62	PIR61	PIR60
0x021 (r/w)	T6RLLB	10-bit real-time timer/counter reload buffer low byte							
0x022 (r/w)	T6RLHB	-	-	-	-	-	-	D9	D8
Timer7: 8-bit Timer									
0x023 (r/w)	T7CON	T7EN	-	T7CS1	T7CS0	T7EDG	T7PS2	T7PS1	T7PS0
0x024 (r/w)	T7RL	8-bit real-time timer/counter reload buffer							
0x025 (r)	T7CNT	8-bit real-time timer/counter Count							
SPI									
0x026 (r/w)	SPICON1	SSE	CPOL	CPHA	SWAP	DORD	SPIMOD	SPIPS1	SPIPS0
0x027 (r/w)	SPICON2	SPIEN	TXOV	RXOV	SPISTS	-	MOSIST	MISOST	SSBEN
0x028 (r/w)	SPITXB	SPI Transmitter buffer							
0x029 (r)	SPIRXB	SPI Receive buffer							
Interrupt									
0x02A (r/w)	INTEN	GIE	-	PIE	-	-	-	-	SPIIE
0x02B (r/w)	INTEN1	-	T7IE	T6P6IE	T5P5IE	T4P4IE	T3P3IE	T2P2IE	T1P1IE
0x02C (r/w)	INTFLAG	-	-	PIF	-	-	-	-	SPIIF
0x02D (r/w)	INTFLAG1	-	T7IF	T6P6IF	T5P5IF	T4P4IF	T3P3IF	T2P2IF	T1P1IF
Others									
0x02E (r/w)	APHCON	PHA7	PHA6	PHA5	PHA4	PHA3	PHA2	PHA1	PHA0
0x02F (r/w)	BPHCON	PHB7	PHB6	PHB5	PHB4	-	PHB2	PHB1	PHB0
0x030 (r/w)	PLCON	-	-	PLA5	PLA4	PLA3	PLB2	PLB1	PLB0
0x031 (r/w)	CLOCON	CLOEN	SPICS	-	DINV	DUTY	CLOPS2	CLOPS1	CLOPS0
0x032 (r/w)	WDTCON	WDTEN	I_WDT	I_TWDT	EXCLK	-	WDTPS2	WDTPS1	WDTPS0
0x033 (r/w)	INTPAB	-	-	PB5IEN	PB4IEN	PB3IEN	PA7IEN	PA6IEN	PA5IEN
0x034 (r/w)	OSCCON	CLKSW	-	T16F	HCS	HRT	-	IRCPD	ECLKPD

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

2.0 FUNCTIONAL DESCRIPTIONS

2.1 Operational Registers

2.1.1 INDF (Indirect Addressing Register)

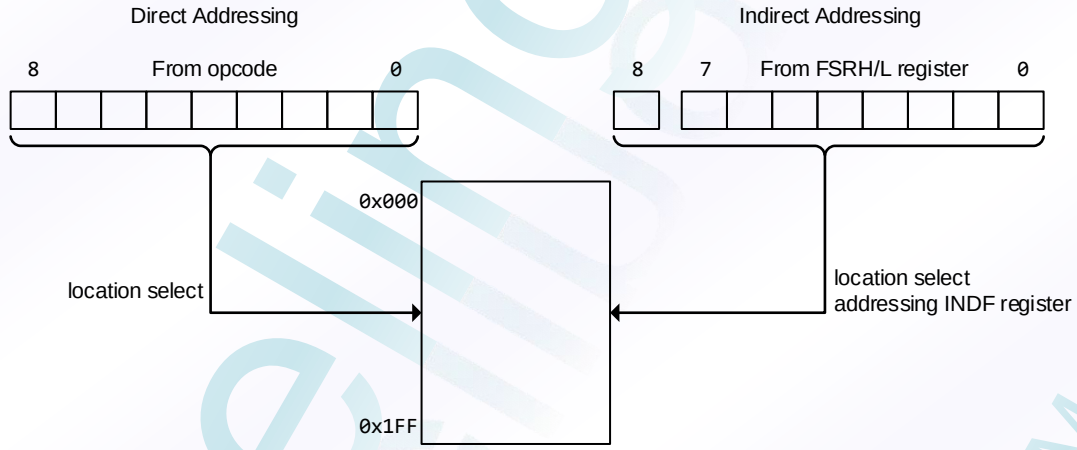
Read/Write-POR		R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x000	INDF	Uses contents of FSRH and FSRL to address data memory (not a physical register)							

Legend: x = unknown, more bits default state, please refer to Table 2.8.

The INDF Register is not a physical register. Any instruction accessing the INDF register can actually access the register pointed by FSRH and FSRL Register. Reading the INDF register itself indirectly (FSRH and FSRL="0x00") will read 0x00. Writing to the INDF register indirectly results in a no-operation (although status bits may be affected).

Example 2.1: INDIRECT ADDRESSING
Register file 0x148 contains the value 0x10
Register file 0x149 contains the value 0x0A
Load the value 0x148 into the FSRH and FSRL Register
A read of the INDF Register will return the value of 0x10
Increment the value of the FSRL Register by one (@FSRH and FSRL=0x149)
A read of the INDF register now will return the value of 0x0A.

Figure 2.1: Direct/Indirect Addressing for FM8PE581M



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
0x002	PCL	Low order 8 bits of PC							

Note: more bits default state, please refer to Table 2.8.

FM8PE581M device has 13-bit wide Program Counter (PC) and eight-level deep 13-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<12:8> bits and is not directly readable or writable. All updates to the PCH register go through the PCHBUF register. 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<12:0> is provided by the GOTO instruction word. The PCL register is mapped to PC<7:0>, and the PCHBUF register is not updated.

For a CALL instruction, the PC<12:0> 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>, and the PCHBUF register is not updated.

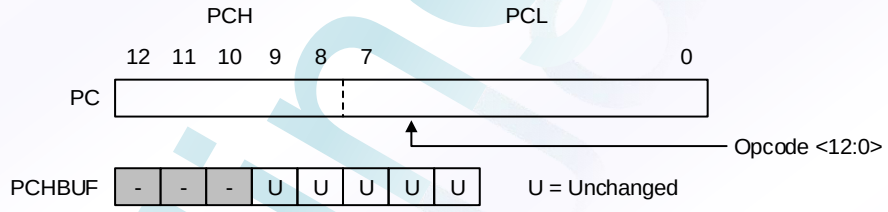
For a RETURN, RETFIE or RETIA instruction, the PC are updated (POPed) from the top of STACK. The PCL register is mapped to PC<7:0>, and the PCHBUF register is not updated.

For any instruction where the PCL is the destination, the PC<7:0> is provided by the instruction word or ALU result. However, the PC<12:8> will come from the PCHBUF<4:0> bits (PCHBUF → PCH).

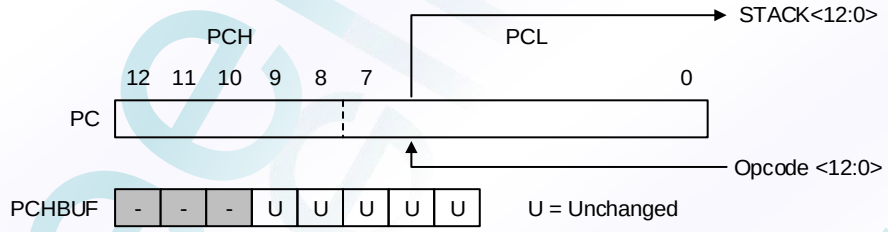
PCHBUF only when the PCL is written, will be updated to the PCH.

Figure 2.2: Loading of PC in Different Situations

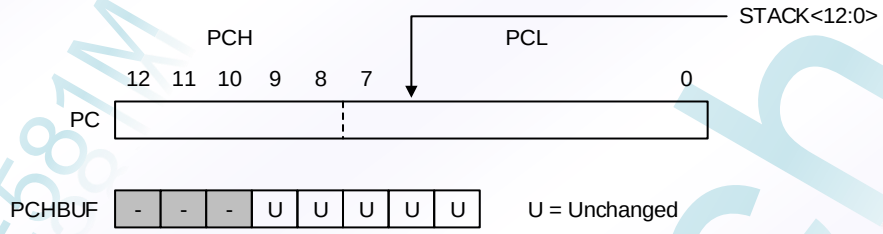
Situation 1: GOTO Instruction



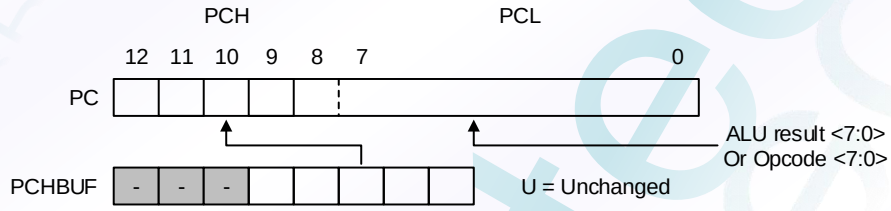
Situation 2: CALL Instruction



Situation 3: RETURN or RETFIE or RETIA Instruction



Situation 4: Instruction with PCL as destination



2.1.3 STATUS (Status Register)

Read/Write-POR		-	-	-	R-#	R-#	R/W-x	R/W-x	R/W-x
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x003	STATUS	-	-	-	$\overline{TO}$	$\overline{PD}$	Z	DC	C

Legend: - = unimplemented, read as '0', x = unknown, # refer Table 2.9 for detail description, more bits default state, please refer to Table 2.8.

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, ADCAR, ADDIA:
= 0, No Carry occurred.
= 1, Carry occurred.
SUBAR, SBCAR, SUBIA:
= 0, Borrow occurred.
= 1, No 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, ADCAR, ADDIA:
= 0, No Carry from the 4th low order bit of the result occurred.
= 1, Carry from the 4th low order bit of the result occurred.
SUBAR, SBCAR, SUBIA:
= 0, Borrow from the 4th low order bit of the result occurred.
= 1, No Borrow from the 4th low order bit of the result occurred.

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

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

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

2.1.4 FSRH and FSRL (High and Low Bytes Indirect Data Memory Address Pointer)

Read/Write-POR		R/W-x	R/W-x	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
0x004	FSRL	Indirect data memory address pointer low byte							

Read/Write-POR		-	-	-	-	-	-	-	R/W-x
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x005	FSRH	-	-	-	-	-	-	-	B8

Legend: - = unimplemented, read as '0', x = unknown, more bits default state, please refer to Table 2.8.

Bit8:Bit0: Select registers address in the indirect addressing mode. See 2.1.1 for detail description.

2.1.5 PCHBUF (High Byte Buffer of Program Counter)

Read/Write-POR		-	-	-	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
0x006	PCHBUF	0	0	0	Upper 5 MSBs Buffer of PC				

Note: 0 = Not used, must fixed to '0' for future, more bits default state, please refer to Table 2.8.

Bit4:Bit0: See 2.1.2 for detail description.

2.1.6 PORTA, PORTB, IOSTA and IOSTB (Port Data Registers and Port Direction Control Registers)

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
0x007	IOSTA	IOSTA7	IOSTA6	IOSTA5	IOSTA4	IOSTA3	IOSTA2	IOSTA1	IOSTA0

Read/Write-POR		R/W-x	R/W-x	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
0x008	PORTA	IOA7	IOA6	IOA5	IOA4	IOA3	IOA2	IOA1	IOA0

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
0x009	IOSTB	IOSTB7	IOSTB6	IOSTB5	IOSTB4	IOSTB3	IOSTB2	IOSTB1	IOSTB0

Read/Write-POR		R/W-x	R/W-x	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
0x00A	PORTB	IOB7	IOB6	IOB5	IOB4	IOB3*	IOB2	IOB1	IOB0

Legend: x = unknown, more bits default state, please refer to Table 2.8.

The registers (IOSTA and IOSTB) are used to define the input or output of each port.

= 0, Output.

= 1, Input.

Reading the port (PORTA and PORTB 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. Please refer to 2.2 for detail I/O Port description.

Note: IOB3 is open-drain output only. If IOSTB3 ='0' and IOB3 bit is set to '1', the IOB3 pin will be float. Please refer to 2.2 for detail I/O Port description.

2.1.7 Timer1: 10-bit Timer & PWM1 duty & CPWM1 period

In timer mode, the Timer1 is a 10-bit down count timer/counter which includes reload buffer (T1RLLB and T1RLHB). Please refer to 2.3 for detail Timer description.
In Normal or Extension PWM mode, the Timer1 as PWM1 duty-cycle, PWM function controlled by the register P1CON. In cascade mode, the Timer1 as CPWM1 Period-cycle, Timer2 as CPWM1 duty-cycle, PWM function and Brake source controlled by the register P1CON. Please refer to 2.4 for detail PWM description.

2.1.7.1 T1CON (Timer1 Control 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
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x00B	T1CON	T1EN	-	T1CS1	T1CS0	T1EDG	T1PS2	T1PS1	T1PS0

Legend: x = unknown, more bits default state, please refer to Table 2.8.

T1PS2:T1PS0: Timer1 Pre-scaler select bits.

T1PS2:T1PS0			Timer1 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

T1EDG: Timer1 clock edge select bit. This bit works only when external clock source TMCKI (IOB5) selected.
= 0, Timer1 decreased while external clock L→H (Rising edge).
= 1, Timer1 decreased while external clock H→L (Falling edge).

T1CS1:T1CS0: Timer1 clock source select bits.

T1CS1:T1CS0		Timer1 clock source
0	0	TMCKI(IOB5)
0	1	OSC or LIRC
1	0	HIRC or ERC
1	1	HIRCx2

T1EN: Timer1 Enable/Disable bit.
= 0, Timer1 (PWM1) Disable.
= 1, Timer1 (PWM1) Enable.

2.1.7.2 P1CON (PWM1 Control 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
0x00C	P1CON	T1MOD	P1INV	EPWM1	CPWM1	PIR13	PIR12	PIR11	PIR10

Note: more bits default state, please refer to Table 2.8.

PIR13:PIR10: Interrupt Event Rate of PWM1.
"1:N" means interrupt occurred after "N" PWM1 pulses.

PIR13:PIR10				PWM1 Interrupt rate
0	0	0	0	1:1
0	0	0	1	1:2
0	0	1	0	1:3
0	0	1	1	1:4
		|		|
1	1	0	1	1:14
1	1	1	0	1:15
1	1	1	1	1:16

CPWM1: Cascade mode of PWM1 enable/disable bit (T1MOD bit = 1).
= 0, Disable Cascade PWM mode, PWM1 and PWM2 are fixed period-cycle of PWM.
= 1, Enable Cascade PWM mode, PWM1 for period-cycle, PWM2 for duty-cycle.

EPWM1: Extension mode of PWM1 enable/disable bit.
= 0, Disable PWM1 Extension mode.
= 1, Enable PWM1 Extension mode.

P1INV: PWM1 output Invert select bit.
= 0, Set the initial state to H, change to L when duty overflow.
= 1, Set the initial state to L, change to H when duty overflow.

T1MOD: Timer1 operation mode select bit.
= 0, Timer mode.
= 1, PWM mode.

2.1.7.3 T1RLHB & T1RLLB (Timer1 Reload High & Low byte Register)

Read/Write-POR		R/W-x	R/W-x	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
0x00D	T1RLLB	10-bit real-time timer/counter reload buffer low byte							

Read/Write-POR		-	-	-	-	-	-	R/W-x	R/W-x
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x00E	T1RLHB	-	-	-	-	-	-	D9	D8

Legend: - = unimplemented, read as '0', x = unknown, more bits default state, please refer to Table 2.8.

T1RLLB and T1RLHB are Timer1 reload buffer, when the underflow occurs, this value will be loaded into T1CNT (Can't be accessed) continues to counting, see 2.3 for detail description.
Please note: When the update these reload buffers, low byte (register T1RLLB) value must be written first.

2.1.8 Timer2: 10-bit Timer & PWM2 duty & CPWM1 duty

In timer mode, the Timer2 is a 10-bit down count timer/counter which includes reload buffer (T2RLLB, T2RLHB). Please refer to 2.3 for detail Timer description.
In Normal or Extension PWM mode, the Timer2 as PWM2 duty-cycle, PWM function controlled by the register P2CON. In cascade mode, the Timer1 as CPWM1 period-cycle, Timer2 as CPWM1 duty-cycle, PWM duty-cycle time-base controlled by the register T2CON. P2CON register must be keep 0x00. Please refer to 2.4 for detail PWM description.

2.1.8.1 T2CON (Timer2 Control 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
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x00F	T2CON	T2EN	-	T2CS1	T2CS0	T2EDG	T2PS2	T2PS1	T2PS0

Legend: - = unimplemented, read as '0', more bits default state, please refer to Table 2.8.

T2PS2:T2PS0: Timer2 Pre-scaler select bits.

T2PS2:T2PS0			Timer2 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

T2EDG: Timer2 clock edge select bit. This bit works only when external clock source TMCKI (IOB5) selected.
= 0, Timer2 decreased while external clock L→H (Rising edge).
= 1, Timer2 decreased while external clock H→L (Falling edge).

T2CS1:T2CS0: Timer2 clock source select bits.

T2CS1:T2CS0		Timer2 clock source
0	0	TMCKI(IOB5)
0	1	OSC or LIRC
1	0	HIRC or ERC
1	1	HIRCx2

T2EN: Timer2 Enable/Disable bit.
= 0, Timer2 Disable.
= 1, Timer2 Enable.
Note: In CPWM mode, this bit controlled by T1EN bit, and can be keep 0.

2.1.8.2 P2CON (PWM2 Control 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
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x010	P2CON	T2MOD	P2INV	EPWM2	-	PIR23	PIR22	PIR21	PIR20

Legend: - = unimplemented, read as '0', more bits default state, please refer to Table 2.8.

PIR23:PIR20: Interrupt Event Rate of PWM2.
"1:N" means interrupt occurred after "N" PWM2 pulses.

PIR23:PIR20	PWM2 Interrupt rate
0 0 0 0	1:1
0 0 0 1	1:2
0 0 1 0	1:3
0 0 1 1	1:4
|	|
1 1 0 1	1:14
1 1 1 0	1:15
1 1 1 1	1:16

EPWM2: Extension mode of PWM2 enable/disable bit.
= 0, Disable PWM2 Extension mode.
= 1, Enable PWM2 Extension mode.

P2INV: PWM2 output Invert select bit.
= 0, Set the initial state to H, change to L when duty overflow.
= 1, Set the initial state to L, change to H when duty overflow.

T2MOD: Timer2 operation mode select bit.
= 0, Timer mode.
= 1, PWM mode.

2.1.8.3 T2RLHB & T2RLLB (Timer2 Reload High & Low byte Register)

Read/Write-POR		R/W-x	R/W-x	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
0x011	T2RLLB	10-bit real-time timer/counter reload buffer low byte							

Read/Write-POR		-	-	-	-	-	-	R/W-x	R/W-x
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x012	T2RLHB	-	-	-	-	-	-	D9	D8

Legend: - = unimplemented, read as '0', x = unknown, more bits default state, please refer to Table 2.8.

T2RLLB and T2RLHB are Timer2 reload buffer, when the underflow occurs, this value will be loaded into T2CNT (Can't be accessed) continues to counting, see 2.3 for detail description.
Please note: When the update these reload buffers, low byte (register T2RLLB) value must be written first.

2.1.9 Timer3: 10-bit Timer & PWM3 duty & CPWM2 period

In timer mode, the Timer3 is a 10-bit down count timer/counter which includes reload buffer (T3RLLB and T3RLHB). Please refer to 2.3 for detail Timer description.
In Normal or Extension PWM mode, the Timer3 as PWM3 duty-cycle, PWM function controlled by the register P3CON. In cascade mode, the Timer3 as CPWM2 Period-cycle, Timer4 as CPWM2 duty-cycle, PWM function and Brake source controlled by the register P3CON. Please refer to 2.4 for detail PWM description.

2.1.9.1 T3CON (Timer3 Control 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
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x013	T3CON	T3EN	-	T3CS1	T3CS0	T3EDG	T3PS2	T3PS1	T3PS0

Legend: - = unimplemented, read as '0', more bits default state, please refer to Table 2.8.

T3PS2:T3PS0: Timer3 Pre-scaler select bits.

T3PS2:T3PS0			Timer3 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

T3EDG: Timer3 clock edge select bit. This bit works only when external clock source TMCKI (IOB5) selected.
= 0, Timer3 decreased while external clock L→H (Rising edge).
= 1, Timer3 decreased while external clock H→L (Falling edge).

T3CS1:T3CS0: Timer3 clock source select bits.

T3CS1:T3CS0		Timer3 clock source
0	0	TMCKI(IOB5)
0	1	OSC or LIRC
1	0	HIRC or ERC
1	1	HIRCx2

T3EN: Timer3 Enable/Disable bit.
= 0, Timer3 (PWM3) Disable.
= 1, Timer3 (PWM3) Enable.

2.1.9.2 P3CON (PWM3 Control 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
0x014	P3CON	T3MOD	P3INV	EPWM3	CPWM2	PIR33	PIR32	PIR31	PIR30

Note: more bits default state, please refer to Table 2.8.

PIR33:PIR30: Interrupt Event Rate of PWM3.
"1:N" means interrupt occurred after "N" PWM3 pulses.

PIR33:PIR30	PWM3 Interrupt rate
0 0 0 0	1:1
0 0 0 1	1:2
0 0 1 0	1:3
0 0 1 1	1:4
|	|
1 1 0 1	1:14
1 1 1 0	1:15
1 1 1 1	1:16

CPWM2: Cascade mode of PWM3 enable/disable bit (T3MOD bit = 1).
= 0, Disable Cascade PWM mode, PWM3 and PWM4 are fixed period-cycle of PWM.
= 1, Enable Cascade PWM mode, PWM3 for period-cycle, PWM4 for duty-cycle.

EPWM2: Extension mode of PWM3 enable/disable bit.
= 0, Disable PWM3 Extension mode.
= 1, Enable PWM3 Extension mode.

P3INV: PWM3 output Invert select bit.
= 0, Set the initial state to H, change to L when duty overflow.
= 1, Set the initial state to L, change to H when duty overflow.

T3MOD: Timer3 operation mode select bit.
= 0, Timer mode.
= 1, PWM mode.

2.1.9.3 T3RLHB & T3RLLB (Timer3 Reload High & Low byte Register)

Read/Write-POR		R/W-x	R/W-x	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
0x015	T3RLLB	10-bit real-time timer/counter reload buffer low byte							

Read/Write-POR		-	-	-	-	-	-	R/W-x	R/W-x
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x016	T3RLHB	-	-	-	-	-	-	D9	D8

Legend: - = unimplemented, read as '0', x = unknown, more bits default state, please refer to Table 2.8.

T3RLLB and T3RLHB are Timer3 reload buffer, when the underflow occurs, this value will be loaded into T3CNT (Can't be accessed) continues to counting, see 2.3 for detail description.
Please note: When the update these reload buffers, low byte (register T3RLLB) value must be written first.

2.1.10 Timer4: 10-bit Timer & PWM4 duty & CPWM2 duty

In timer mode, the Timer4 is a 10-bit down count timer/counter which includes reload buffer (T4RLLB, T4RLHB). Please refer to 2.3 for detail Timer description.
In Normal or Extension PWM mode, the Timer4 as PWM4 duty-cycle, PWM function controlled by the register P4CON. In cascade mode, the Timer3 as CPWM2 period-cycle, Timer4 as CPWM2 duty-cycle, PWM duty-cycle time-base controlled by the register T4CON. P4CON register must be keep 0x00. Please refer to 2.4 for detail PWM description.

2.1.10.1 T4CON (Timer4 Control 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
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x017	T4CON	T4EN	-	T4CS1	T4CS0	T4EDG	T4PS2	T4PS1	T4PS0

Legend: - = unimplemented, read as '0', more bits default state, please refer to Table 2.8.

T4PS2:T4PS0: Timer4 Pre-scaler select bits.

T4PS2:T4PS0			Timer4 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

T4EDG: Timer4 clock edge select bit. This bit works only when external clock source TMCKI (IOB5) selected.
= 0, Timer4 decreased while external clock L→H (Rising edge).
= 1, Timer4 decreased while external clock H→L (Falling edge).

T4CS1:T4CS0: Timer4 clock source select bits.

T4CS1:T4CS0		Timer4 clock source
0	0	TMCKI(IOB5)
0	1	OSC or LIRC
1	0	HIRC or ERC
1	1	HIRCx2

T4EN: Timer4 Enable/Disable bit.
= 0, Timer4 Disable.
= 1, Timer4 Enable.
Note: In CPWM mode, this bit controlled by T3EN bit, and can be keep 0.

2.1.10.2 P4CON (PWM4 Control 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
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x018	P4CON	T4MOD	P4INV	EPWM4	-	PIR43	PIR42	PIR41	PIR40

Legend: - = unimplemented, read as '0', more bits default state, please refer to Table 2.8.

PIR43:PIR40: Interrupt Event Rate of PWM4.
"1:N" means interrupt occurred after "N" PWM4 pulses.

PIR43:PIR40				PWM4 Interrupt rate
0	0	0	0	1:1
0	0	0	1	1:2
0	0	1	0	1:3
0	0	1	1	1:4
		|		|
1	1	0	1	1:14
1	1	1	0	1:15
1	1	1	1	1:16

EPWM4: Extension mode of PWM4 enable/disable bit.
= 0, Disable PWM4 Extension mode.
= 1, Enable PWM4 Extension mode.

P4INV: PWM4 output Invert select bit.
= 0, Set the initial state to H, change to L when duty overflow.
= 1, Set the initial state to L, change to H when duty overflow.

T4MOD: Timer4 operation mode select bit.
= 0, Timer mode.
= 1, PWM mode.

2.1.10.3 T4RLHB & T4RLLB (Timer4 Reload High & Low byte Register)

Read/Write-POR		R/W-x	R/W-x	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
0x019	T4RLLB	10-bit real-time timer/counter reload buffer low byte							

Read/Write-POR		-	-	-	-	-	-	R/W-x	R/W-x
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x01A	T4RLHB	-	-	-	-	-	-	D9	D8

Legend: - = unimplemented, read as '0', x = unknown, more bits default state, please refer to Table 2.8.

T4RLLB and T4RLHB are Timer4 reload buffer, when the underflow occurs, this value will be loaded into T4CNT (Can't be accessed) continues to counting, see 2.3 for detail description.
Please note: When the update these reload buffers, low byte (register T4RLLB) value must be written first.

2.1.11 Timer5: 10-bit Timer & PWM5 & CPWM3 Period

In timer mode, the Timer1 is a 10-bit down count timer/counter which includes reload buffer (T5RLLB and T5RLHB). Please refer to 2.3 for detail Timer description.
In Normal or Extension PWM mode, the Timer5 as PWM5 duty-cycle, PWM function controlled by the register P5CON. In cascade mode, the Timer5 as CPWM3 Period-cycle, Timer6 as CPWM3 duty-cycle, PWM function and Brake source controlled by the register P5CON. Please refer to 2.4 for detail PWM description.

2.1.11.1 T5CON (Timer5 Control 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
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x01B	T5CON	T5EN	-	T5CS1	T5CS0	T5EDG	T5PS2	T5PS1	T5PS0

Legend: - = unimplemented, read as '0', more bits default state, please refer to Table 2.8.

T5PS2:T5PS0: Timer5 Pre-scaler select bits.

T5PS2:T5PS0			Timer5 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

T5EDG: Timer5 clock edge select bit. This bit works only when external clock source TMCKI (IOB5) selected.
= 0, Timer5 decreased while external clock L→H (Rising edge).
= 1, Timer5 decreased while external clock H→L (Falling edge).

T5CS1:T5CS0: Timer5 clock source select bits.

T5CS1:T5CS0		Timer5 clock source
0	0	TMCKI(IOB5)
0	1	OSC or LIRC
1	0	HIRC or ERC
1	1	HIRCx2

T5EN: Timer5 Enable/Disable bit.
= 0, Timer5 (PWM5) Disable.
= 1, Timer5 (PWM5) Enable.

2.1.11.2 P5CON (PWM5 Control 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
0x01C	P5CON	T5MOD	P5INV	EPWM5	CPWM3	PIR53	PIR52	PIR51	PIR50

Note: more bits default state, please refer to Table 2.8.

PIR53:PIR50: Interrupt Event Rate of PWM5.
"1:N" means interrupt occurred after "N" PWM5 pulses.

PIR53:PIR50				PWM5 Interrupt rate
0	0	0	0	1:1
0	0	0	1	1:2
0	0	1	0	1:3
0	0	1	1	1:4
		|		|
1	1	0	1	1:14
1	1	1	0	1:15
1	1	1	1	1:16

CPWM3: Cascade mode of PWM3 enable/disable bit (T5MOD bit = 1).
= 0, Disable Cascade PWM mode, PWM5 and PWM6 are fixed period-cycle of PWM.
= 1, Enable Cascade PWM mode, PWM5 for period-cycle, PWM6 for duty-cycle.

EPWM5: Extension mode of PWM5 enable/disable bit.
= 0, Disable PWM5 Extension mode.
= 1, Enable PWM5 Extension mode.

P5INV: PWM5 output Invert select bit.
= 0, Set the initial state to H, change to L when duty overflow.
= 1, Set the initial state to L, change to H when duty overflow.

T5MOD: Timer5 operation mode select bit.
= 0, Timer mode.
= 1, PWM mode.

2.1.11.3 T5RLHB & T5RLLB (Timer5 Reload High & Low byte Register)

Read/Write-POR		R/W-x	R/W-x	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
0x01D	T5RLLB	12/10-bit real-time timer/counter reload buffer low byte							

Read/Write-POR		-	-	-	-	-	-	R/W-x	R/W-x
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x01E	T5RLHB	-	-	-	-	-	-	D9	D8

Legend: - = unimplemented, read as '0', x = unknown, more bits default state, please refer to Table 2.8.

T5RLLB and T5RLHB are Timer5 reload buffer, when the underflow occurs, this value will be loaded into T5CNT (Can't be accessed) continues to counting, see 2.3 for detail description.
Please note: When the update these reload buffers, low byte (register T5RLLB) value must be written first.

2.1.12 Timer6: 10-bit Timer & PWM6 duty & CPWM3 duty

In timer mode, the Timer6 is a 10-bit down count timer/counter which includes reload buffer (T6RLLB, T6RLHB). Please refer to 2.3 for detail Timer description.
In Normal or Extension PWM mode, the Timer6 as PWM6 duty-cycle, PWM function controlled by the register P6CON. In cascade mode, the Timer5 as CPWM3 period-cycle, Timer6 as CPWM3 duty-cycle, PWM duty-cycle time-base controlled by the register T6CON. P6CON register must be keep 0x00. Please refer to 2.4 for detail PWM description.

2.1.12.1 T6CON (Timer6 Control 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
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x01F	T6CON	T6EN	-	T6CS1	T6CS0	T6EDG	T6PS2	T6PS1	T6PS0

Legend: - = unimplemented, read as '0', more bits default state, please refer to Table 2.8.

T6PS2:T6PS0: Timer6 Pre-scaler select bits.

T6PS2:T6PS0			Timer6 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

T6EDG: Timer6 clock edge select bit. This bit works only when external clock source TMCKI (IOB5) selected.
= 0, Timer6 decreased while external clock L→H (Rising edge).
= 1, Timer6 decreased while external clock H→L (Falling edge).

T6CS1:T6CS0: Timer6 clock source select bits.

T6CS1:T6CS0		Timer6 clock source
0	0	TMCKI(IOB5)
0	1	OSC or LIRC
1	0	HIRC or ERC
1	1	HIRCx2

T6EN: Timer6 Enable/Disable bit.
= 0, Timer6 Disable.
= 1, Timer6 Enable.
Note: In CPWM mode, this bit controlled by T5EN bit, and can be ignore.

2.1.12.2 P6CON (PWM6 Control 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
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x020	P6CON	T6MOD	P6INV	EPWM6	-	PIR63	PIR62	PIR61	PIR60

Legend: - = unimplemented, read as '0', more bits default state, please refer to Table 2.8.

PIR63:PIR60: Interrupt Event Rate of PWM6.
"1:N" means interrupt occurred after "N" PWM6 pulses.

PIR63:PIR60				PWM6 Interrupt rate
0	0	0	0	1:1
0	0	0	1	1:2
0	0	1	0	1:3
0	0	1	1	1:4
		|		|
1	1	0	1	1:14
1	1	1	0	1:15
1	1	1	1	1:16

EPWM6: Extension mode of PWM6 enable/disable bit.
= 0, Disable PWM6 Extension mode.
= 1, Enable PWM6 Extension mode.

P6INV: PWM6 output Invert select bit.
= 0, Set the initial state to H, change to L when duty overflow.
= 1, Set the initial state to L, change to H when duty overflow.

T6MOD: Timer6 operation mode select bit.
= 0, Timer mode.
= 1, PWM mode.

2.1.12.3 T6RLHB & T6RLLB (Timer6 Reload High & Low byte Register)

Read/Write-POR		R/W-x	R/W-x	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
0x021	T6RLLB	10-bit real-time timer/counter reload buffer low byte							

Read/Write-POR		-	-	-	-	-	-	R/W-x	R/W-x
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x022	T6RLHB	-	-	-	-	-	-	D9	D8

Legend: - = unimplemented, read as '0', x = unknown, more bits default state, please refer to Table 2.8.

T6RLLB and T6RLHB are Timer6 reload buffer, when the underflow occurs, this value will be loaded into T6CNT (Can't be accessed) continues to counting, see 2.3 for detail description.
Please note: When the update these reload buffers, low byte (register T6RLLB) value must be written first.

2.1.13 Timer7: 8-bit Timer

The Timer7 is an 8-bit down count timer/counter which includes reload buffer (T7RL). Please refer to 2.3 for detail Timer description.

2.1.13.1 T7CON (Timer7 Control 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
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x023	T7CON	T7EN	-	T7CS1	T7CS0	T7EDG	T7PS2	T7PS1	T7PS0

Legend: - = unimplemented, read as '0', more bits default state, please refer to Table 2.8.

T7PS2:T7PS0: Timer7 Pre-scaler select bits

T7PS2:T7PS0			Timer7 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

T7EDG: Timer7 clock edge select bit. This bit works only when external clock source TMCKI (IOB5) selected.
= 0, Timer7 decreased while external clock L→H (Rising edge).
= 1, Timer7 decreased while external clock H→L (Falling edge).

T7CS1:T7CS0: Timer7 clock source select bits.

T7CS1:T7CS0		Timer7 clock source
0	0	TMCKI(IOB5)
0	1	OSC or LIRC
1	0	HIRC or ERC
1	1	No function, do not use.

T7EN: Timer7 Enable/Disable bit.
= 0, Timer7 Disable.
= 1, Timer7 Enable.

2.1.13.2 T7RL & T7CNT (Timer7 Reload buffer & Counter 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
0x024	T7RL	8-bit real-time timer/counter reload buffer							

Read/Write-POR		R-1	R-1	R-1	R-1	R-1	R-1	R-1	R-1
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x025	T7CNT	8-bit real-time timer/counter Count							

Note: more bits default state, please refer to Table 2.8.

T7RL is Timer7 reload buffer, when the underflow occurs, this value will be loaded into T7CNT continues to counting, see 2.3 for detail description.

2.1.14 SPI Control Registers

2.1.14.1 SPICON1 (SPI 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
0x026	SPICON1	SSE	CPOL	CPHA	SWAP	DORD	SPIMOD	SPIPS1	SPIPS0

Note: more bits default state, please refer to Table 2.8.

SPIPS1:SPIPS0: SPI clock output pre-scaler select bits (only for master mode).

SPIPS1:SPIPS0		SPI clock pre-scaler ratio
0	0	System clock/4
0	1	System clock/8
1	0	System clock/16
1	1	No function, don't use.

Note: SPI clock source defined by SPICS bit (CLOCON<6>).

SPIMOD: SPI master/slave mode select bit.
= 0, Slave mode.
= 1, Master mode.

DORD: SPI data transmission order.
= 0, Data shift in/out MSB first.
= 1, Data shift in/out LSB first.

SWAP: Swap function of MOSI and MISO pin.
= 0, Disable swap function.
= 1, Enable swap function.
Note: See Table 2.4 and Table 2.5 for detail description.

CPOL:CPHA: Clock phase and polarity select bits.

CPOL:CPHA		SPI SCK phase, polarity and data sample, setup edge
0	0	SCK is low when idle. Data are captured on the SCK rising edge and data is output on the falling edge.
0	1	SCK is low when idle. Data are captured on the SCK falling edge and data is output on the rising edge.
1	0	SCK is high when idle. Data are captured on the SCK falling edge and data is output on the rising edge.
1	1	SCK is high when idle. Data are captured on the SCK rising edge and data is output on the falling edge.

Note: See Figure 2.11 and Figure 2.12 for detail description.

SSE: SPI shift enable bit (only in master mode).
= 0, Reset by hardware as soon as the shifting is complete.
= 1, Start to transmit/receive, and keep on "1" while the current byte is still being transmitted/received.

2.1.14.2 SPICON2 (SPI Control Register 2)

Read/Write-POR		R/W-0	R/W-0	R/W-0	R-x	-	R/W-0	R/W-0	R/W-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x027	SPICON2	SPIEN	TXOV	RXOV	SPISTS	-	MOSIST	MISOST	SSBEN

Legend: - = unimplemented, read as '0', x = unknown, more bits default state, please refer to Table 2.8.

SSBEN: SSB pin enable bit (only in slave mode).
= 0, SSB pin is disable, IOA7 is normal I/O pin.
= 1, SSB pin is enable, IOA7 is SSB control pin.

MISOST: MISO pin status select bit.
= 0, MISO pin input/output state is defined by IOSTA2.
= 1, MISO pin input/output state is defined by SPI module.
Note: See Table 2.4 and Table 2.5 for detail description.

MOSIST: MOSI pin status select bit.
= 0, MOSI pin input/output state is defined by IOSTA1.
= 1, MOSI pin input/output state is defined by SPI module.
Note: See Table 2.4 and Table 2.5 for detail description.

SPISTS: SPI transmitter/receiver Status bit.
= 0, SPI transmitter/receiver is in progress.
= 1, SPI transmitter/receiver is complete.

RXOV: SPI receive buffer overflow bit (only in slave mode), Set when overflow occur, reset by software.

TXOV: SPI transmitter buffer overwrite bit, Set when overwrite occur, reset by software.

SPIEN: SPI module enable/disable bit.
= 0, Disable SPI module.
= 1, Enable SPI module.

2.1.14.3 SPITXB (SPI Transmit Buffer 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
0x028	SPITXB	SPI Transmitter buffer							

Note: more bits default state, please refer to Table 2.8.

SPI transmits data buffer. Once the first valid clock pulse appears on SCK pin, the data in SPITXB will be loaded into SPISR and start to shift in/out.
The new data must be written to SPITXB before the 8-bits data transmission is completed if needed.

2.1.14.4 SPIRXB (SPI Receive Buffer Register)

Read/Write-POR		R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x029	SPIRXB	SPI Receive buffer							

Note: more bits default state, please refer to Table 2.8.

SPI receives data buffer. Once the 8-bits data have been received, the data in SPI shift register (SPISR) will be moved to the SPIRXB register.
The data must be read out before the next 8-bits data reception is completed if needed.

2.1.15 Interrupt Control Registers

2.1.15.1 INTEN (Interrupt Mask Register)

Read/Write-POR		R/W-0	-	R/W-0	-	-	-	-	R/W-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x02A	INTEN	GIE	-	PIE	-	-	-	-	SPIIE

Legend: - = unimplemented, read as '0', more bits default state, please refer to Table 2.8.

SPIIE: SPI module interrupt enable bit.
= 0, Disable the SPI module interrupt.
= 1, Enable the SPI module interrupt.

PIE: PORTA and PORTB pin change and i-WDT wakeup interrupt enable bit
= 0, Disable interrupt.
= 1, Enable interrupt.

GIE: Global interrupt enable bit.
= 0, Disable all interrupts.
= 1, Enable all un-masked interrupts.
Note: When an interrupt event occurred with the GIE bit and its corresponding interrupt enable bits are set, the GIE bit will be cleared by hardware to disable any further interrupts. The RETFIE instruction will exit the interrupt routine and set the GIE bit to re-enable interrupt.

2.1.15.2 INTEN1 (Interrupt Mask 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
0x02B	INTEN1	-	T7IE	T6P6IE	T5P5IE	T4P4IE	T3P3IE	T2P2IE	T1P1IE

Legend: - = unimplemented, read as '0', more bits default state, please refer to Table 2.8.

T1P1IE: Timer1/PWM1 interrupt enable bit.
= 0, Disable interrupt.
= 1, Enable interrupt.

T2P2IE: Timer2/PWM2 interrupt enable bit.
= 0, Disable interrupt.
= 1, Enable interrupt.

T3P3IE: Timer3/PWM3 interrupt enable bit.
= 0, Disable interrupt.
= 1, Enable interrupt.

T4P4IE: Timer4/PWM4 interrupt enable bit.
= 0, Disable interrupt.
= 1, Enable interrupt.

T5P5IE: Timer5/PWM5 interrupt enable bit.
= 0, Disable interrupt.
= 1, Enable interrupt.

T6P6IE: Timer6/PWM6 interrupt enable bit.
= 0, Disable interrupt.
= 1, Enable interrupt.

T7IE: Timer7 interrupt enable bit.
= 0, Disable interrupt.
= 1, Enable interrupt.

2.1.15.3 INTFLAG (Interrupt Status Register)

Read/Write-POR		-	-	R/W-0	-	-	-	-	R/W-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x02C	INTFLAG	-	-	PIF	-	-	-	-	SPIIF

Legend: - = unimplemented, read as '0', more bits default state, please refer to Table 2.8.

SPIIF: SPI module interrupt flag. Set after one byte of SPI transmission is completed, reset by software.

PIF: PORTA and PORTB or i_WDT Interrupt flag. Set when pin changed on selected I/O by register INTPAB or i-WDT wakeup, and clear by software.

2.1.15.4 INTFLAG1 (Interrupt Status 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
0x02D	INTFLAG1	-	T7IF	T6P6IF	T5P5IF	T4P4IF	T3P3IF	T2P2IF	T1P1IF

Legend: - = unimplemented, read as '0', more bits default state, please refer to Table 2.8.

T1P1IF: Timer1 or PWM1 interrupt flag. Set when TMR1 underflow or PWM1 pulse counts to selected interrupt rate, and clear by software.

T2P2IF: Timer2 or PWM2 interrupt flag. Set when TMR2 underflow or PWM2 pulse counts to selected interrupt rate, and clear by software.

T3P3IF: Timer3 or PWM3 interrupt flag. Set when TMR3 underflow or PWM3 pulse counts to selected interrupt rate, and clear by software.

T4P4IF: Timer4 or PWM4 interrupt flag. Set when TMR4 underflow or PWM4 pulse counts to selected interrupt rate, and clear by software.

T5P5IF: Timer5 or PWM5 interrupt flag. Set when TMR5 underflow or PWM5 pulse counts to selected interrupt rate, and clear by software.

T6P6IF: Timer6 or PWM6 interrupt flag. Set when TMR6 underflow or PWM6 pulse counts to selected interrupt rate, and clear by software.

T7IF: Timer7 interrupt flag. Set when TMR7 underflow, and clear by software.

2.1.16 APHCON, BPHCON, PLCON (PORTA and PORTB Pull-high / Pull-low Control 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
0x02E	APHCON	PHA7	PHA6	PHA5	PHA4	PHA3	PHA2	PHA1	PHA0

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
0x02F	BPHCON	PHB7	PHB6	PHB5	PHB4	-	PHB2	PHB1	PHB0

Read/Write-POR		-	-	R/W-%	R/W-%	R/W-%	R/W-%	R/W-%	R/W-%
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x030	PLCON	-	-	PLA5	PLA4	PLA3	PLB2	PLB1	PLB0

Legend: - = unimplemented, read as '0', % = refer to the configuration bit "PPDS", more bits default state, please refer to Table 2.8.

These registers are used to setup pull-high or pull-low resistor enable/disable of each IO pins.

PHAx: = 0, Disabled PORTA corresponding pull-high resistor.
= 1, Enabled PORTA corresponding pull-high resistor.

PHBx: = 0, Disabled PORTB corresponding pull-high resistor.
= 1, Enabled PORTB corresponding pull-high resistor.

PLAx: = 0, Disabled PORTA corresponding pull-low resistor.
= 1, Enabled PORTA corresponding pull-low resistor.

PLBx: = 0, Disabled PORTB corresponding pull-low resistor.
= 1, Enabled PORTB corresponding pull-low resistor.

Note: To minimize power consumption, pull-up / pull-low resistors on the IOA5~IOA3 and IOB2~IOB0 must be carefully managed.

2.1.17 CLOCON (Clock output Control 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
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x031	CLOCON	CLOEN	SPICS	-	DINV	DUTY	CLOPS2	CLOPS1	CLOPS0

Legend: - = unimplemented, read as '0', more bits default state, please refer to Table 2.8.

CLOPS2:CLOPS0: Clock Output pre-scaler setting.

CLOPS2	CLOPS1	CLOPS0	Clock Output pre-scaler ratio
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
Other			No function, don't use.

DUTY: Clock Output Special Duty select bit.
= 0, 1/2 duty output.
= 1, 3/4 duty output.

DINV: Clock Output Special Duty Invert select bit (must be DUTY = 1).
= 0, 3/4 duty output
= 1, 1/4 duty output

SPICS: SPI module clock source select bit.
= 0, SPI clock source is from HIRC or ERC.
= 1, SPI clock source is from LIRC or Crystal.

CLOEN: Clock Output (IOB4) function select bit.
= 0, IOB4 is normal I/O.
= 1, IOB4 is System Clock Output.

2.1.18 WDTCON (Watchdog Timer Control Register)

Read/Write-POR		R/W-1	R/W-0	R/W-0/	R/W-0	-	R/W-1	R/W-1	R/W-1
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x032	WDTCON	WDTEN	I_WDT	I_TWDT	EXCLK	-	WDTPS2	WDTPS1	WDTPS0

Legend: - = unimplemented, read as '0', more bits default state, please refer to Table 2.8.

The FM8PE581M builds in a watchdog timer with two different modes, normal watchdog reset and internal watchdog wakeup. The watchdog timer is controlled by this register. Please refer to 2.8 for detail Watchdog Timer description.

WDTPS2:WDTPS0: Watchdog timer pre-scaler setting bits.

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

EXCLK: External clock (IOB5/TMCKI) function select bit.
= 0, IOB5 is normal I/O.
= 1, IOB5 is external clock input of timer.

I_TWDT: Watchdog Timer Stable time required when operating in I_WDT mode (I_WDT bit = 1).
= 0, 2.5ms.
= 1, 1.25ms.

I_WDT: Internal Watchdog Wakeup mode select bit.
= 0, Internal Watchdog Wakeup Disable.
= 1, Internal Watchdog Wakeup Enable.

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

2.1.19 INTPAB (PORTA and PORTB Interrupt / Wakeup 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
0x033	INTPAB	-	-	PB5IEN	PB4IEN	PB3IEN	PA7IEN	PA6IEN	PA5IEN

Legend: - = unimplemented, read as '0', more bits default state, please refer to Table 2.8.

This register is used to enable/disable the interrupt/wakeup function of PORTA and PORTB. Please refer to 2.6.1 for detail description of External Interrupt and Wake up function.

= 0, Disabled corresponding interrupt/wake-up.
= 1, Enabled corresponding interrupt/wake-up.

2.1.20 OSCCON (Oscillator and Clock 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
0x034	OSCCON	CLKSW	-	T16F	HCS	HRT	-	IRCPD	ECLKPD

Legend: - = unimplemented, read as '0', more bits default state, please refer to Table 2.8.

The FM8PE581M could be operated either dual or single clock system selected by configuration words. Please refer to 2.7 for detail configuration selection description. This register is used to control the switch between different system clocks and power-down function of those clocks.

ECLKPD: Crystal OSC (External clock) / LIRC Power down Control (only valid in dual clock mode)
= 0, Crystal OSC / LIRC Power ON.
= 1, Crystal OSC / LIRC Power Down.
Note: Make sure the system clock been switch to External RC / Internal HIRC before power down Crystal OSC.

IRCPD: Internal RC Power down Control (only valid in dual clock mode)
= 0, External RC / Internal HIRC Power ON.
= 1, External RC / Internal HIRC Power Down.
Note: Make sure the system clock been switch to Crystal OSC / LIRC before power down internal HIRC.

HRT: Wakeup time select in HALT mode (when system clock is stopped).
= 0, 64*F_{OSC}+160uS.
= 1, Power on reset time.

HCS: System clock stop/continue select in HALT mode.
= 0, System clock is continuing.
= 1, System clock is stopped.

T16F: Timer HIRC 16M_{HZ} Filter Enable/Disable bit.
= 0, Disable Filter.
= 1, Enable Timer 16M_{HZ} Filter.

CLKSW: System Clock Select bit (only valid in dual clock mode).
= 0, System Clock is External RC / Internal HIRC.
= 1, System Clock is Crystal OSC / Internal LIRC.
Note: In Dual IRC mode (HIRC/LIRC), the default system clock is LIRC and HIRC off. The other dual-clock mode, the default system clock is HIRC.

2.1.21 ACC (Accumulator)

Read/Write-POR		R/W-x	R/W-x	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
N/A	ACC	Accumulator							

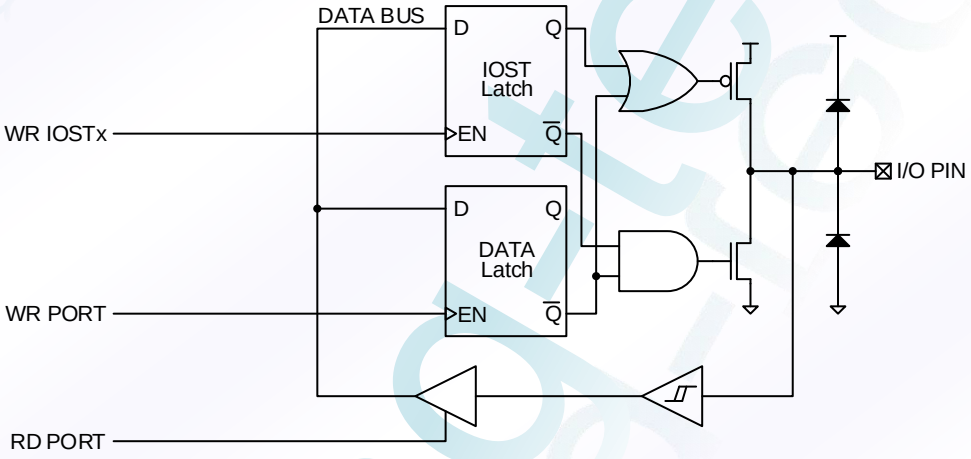
Legend: x = unknown, more bits default state, please refer to Table 2.8.

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

2.2 I/O Ports

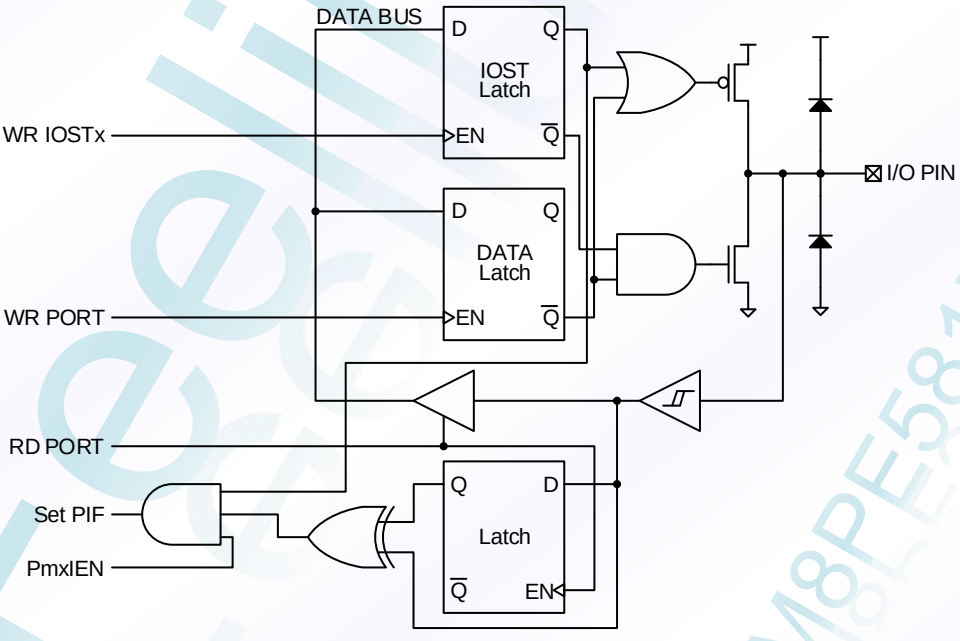
There are totally 16 bi-directional tristate I/O ports. All I/O pins (IOA<7:0> and IOB<7:0>) have specified data direction control registers (IOSTA and IOSTB) which can configure these pins as output or input. Please note that IOB3 is an input or open-drain output pin.
All the IO pins (without IOB3) can also enable or disable a weak internal pull-high by setting APHCON and BPHCON. This weak pull-high will be automatically turned off when the pin is configured as an output pin.
The Configuration Words can set IOB3 to reset functions. When acting as Reset functions the pins will read as "0" during port read.
Please note, IOB3 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
IOA4 ~ IOA0, IOB7, IOB6, IOB2, IOB1:



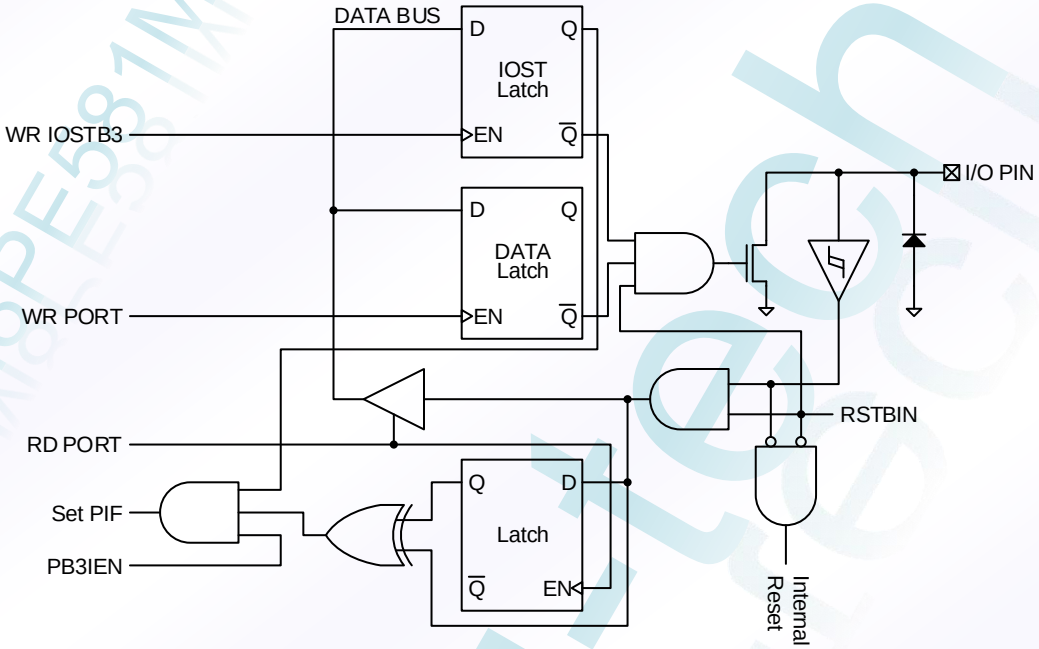
Pull-high/ADC/Comparator control block is not shown in this figure

IOA7 ~ IOA5, IOB5 ~ IOB4:



Pull-high/ADC/OSC/Comparator control block is not shown in this figure

IOB3:

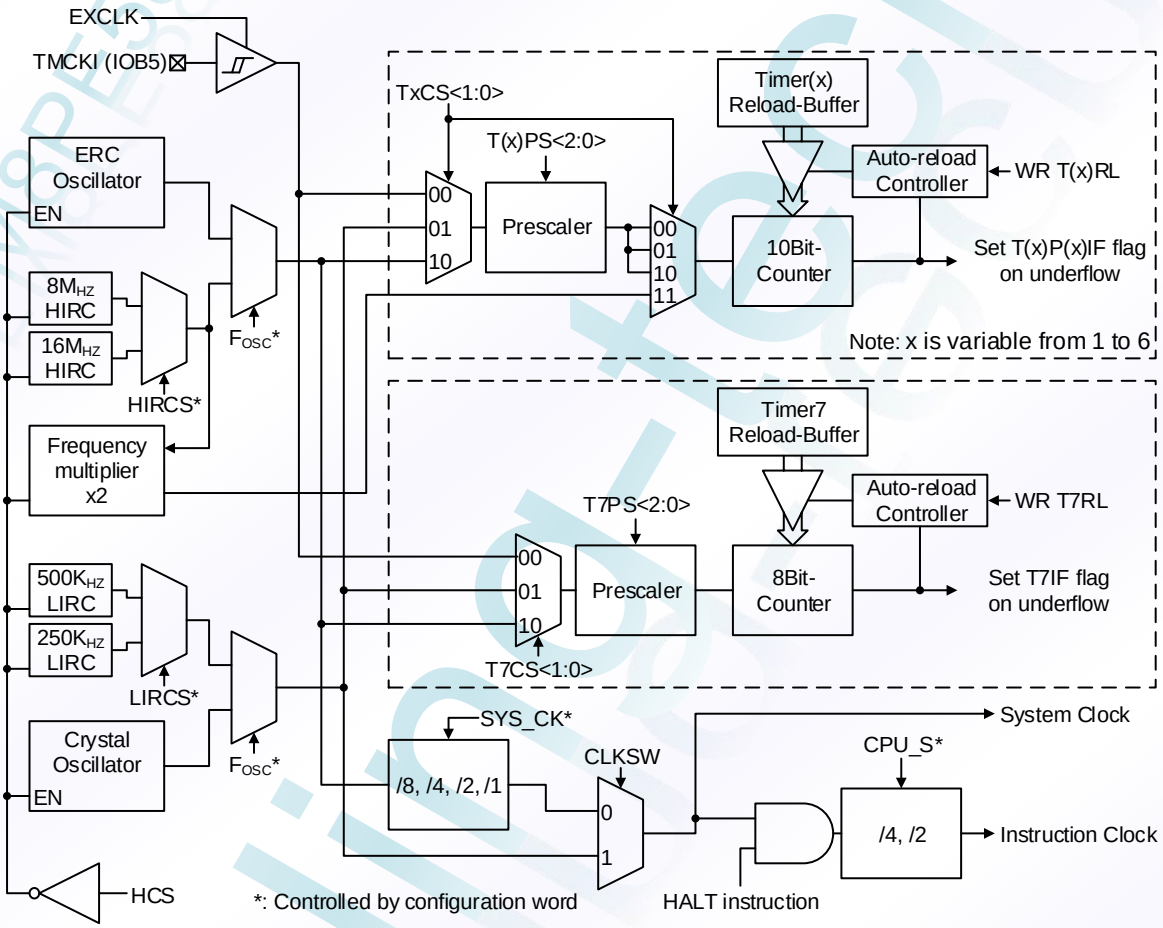


Voltage on this pin must not exceed V_{DD}.

2.3 Timer/Event Counter (Timer1 ~ 7)

The FM8PE581M contains six 10-bit down-count and one 8-bit down-count Timers. All these timers have auto reload function.

Figure 2.4: Simple Block Diagram of the Timer 1 ~ 7



2.3.1 Clock Source

2.3.1.1 TMCKI (IOB5)

This external clock source selected by setting the EXCLK bit (WDTCON <5>) and IOSTB5 bit (IOSTB <5>). In this mode, the timer will decrement every rising or falling edge of pin TMCKI. The decrementing edge is determined by the edge select bit T(x)EDG.

Note: x=1~7.

2.3.1.2 Oscillator (HIRC, LIRC, XT, LF or ERC Oscillator)

In these mode, the timer clock source from Internal RC, Crystal or External RC oscillator module. Oscillator module operating modes are defined by the FOSC bit in the configuration word.
Please note that, in this case, the clock input to the timer in two paths, and therefore will have the following composition:

Table 2.1: Selection of Timer 1 ~ 7 Clock source

F_{OSC} mode of Configuration word	Timer 1 ~ 7 Clock source
HIRC	Only HIRC ($16M_{HZ}$ or $8\ M_{HZ}$) can be selected
LIRC & HIRC	HIRC ($16M_{HZ}$ or $8M_{HZ}$) or LIRC ($500K_{HZ}$ or $250K_{HZ}$) can be selected
XT & HIRC or LF & HIRC	Crystal oscillator or HIRC ($16M_{HZ}$ or $8M_{HZ}$) can be selected
ERC	Only External RC oscillator can be selected
XT or LF	Only Crystal oscillator can be selected

Since the oscillator module is controlled by the F_{OSC} bit, if need a combination of multiple clock sources, the need to carefully choose the configuration word F_{OSC} operating mode.

2.3.1.3 Internal $16M_{HZ}$ or $8M_{HZ}$ *2

In this mode, the HIRC frequency is multiplied by 2, as the timer clock source, this clock source using the same Opportunity and Table 2.2.

Note: 1. In this mode, the frequency multiplier minimum operating voltage limits, please refer to electrical characteristics table item.
2. This mode only for Timer1 to Timer6.

2.3.2 Pre-scaler

Each timer contains a 3-bits pre-scaler which can scale the timer or counter from 1:1 to 1:128.

Table 2.2: Timer1~7 Pre-scaler ratio selection

TxPS2:TxPS0			Timer-x 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

2.4 Pulse Width Modulation (PWM)

2.4.1 Normal PWM

FM8PE581M provides six Normal PWM output shared with Timer1 to Timer6. When PWM1 to PWM6 one or more is selected, the corresponding timer becomes the duty-cycle of PWM. In this mode, PWM period-time is fixed and cannot be programmable.
The PWM outputs are on the IOB0/PWM1, IOB1/PWM2, IOB2/PWM3, IOA3/PWM4, IOA4/PWM5 and IOA5/PWM6 pins.
The PWM1 to PWM6 outputs has a maximum resolution of 10-bits, the duty cycle of the output can vary from 1% to 99%.

The user needs to set the T1MOD bit (P1CON<7>) and cleared CPWM1 bit (P1CON<4>) to enable the PWM1 output. Similarly, PWM3 and PWM5 is the same mode of operation.
The user needs to set the T2MOD bit (P2CON<7>) to enable PWM2 output. Similarly, PWM4 and PWM6 is the same mode of operation.
When T1MOD bit is set, the IOB0/PWM1 pin is configured as PWM1 output and forced as an output mode, irrespective of the data direct bit (IOSTB<0>). When the T1MOD is clear, the pin behaves as a I/O pin.
Similarly, T2MOD, T3MOD, T4MOD, T5MOD bit and T6MOD bit configurations corresponding IOB1/PWM2, IOB2/PWM3, IOA3/PWM4, IOA4/PWM5 and IOA5/PWM6 pin.
Please note, do not written 0x03FF to reload buffer. If written, PWM will generated wrong waveform.

The PWM1 period time can be calculated as follows:

$$\text{Period time of PWM1} = \frac{1024 * \text{Timer1 Pre-scaler rate}}{\text{Timer1 Clock source frequency}}$$

Similarly, this formula can be used directly on PWM2 to PWM6.

The PWM1 duty cycle time can be calculated as follows:

$$\text{Duty cycle time of PWM1} = \frac{(\text{T1RL}+1) * \text{Timer1 Pre-scaler rate}}{\text{Timer2 Clock source frequency}}$$

or

$$\text{T1RL} = \frac{\text{Duty cycle time} * \text{Timer1 Clock source frequency}}{\text{Timer1 Pre-scaler rate}} - 1$$

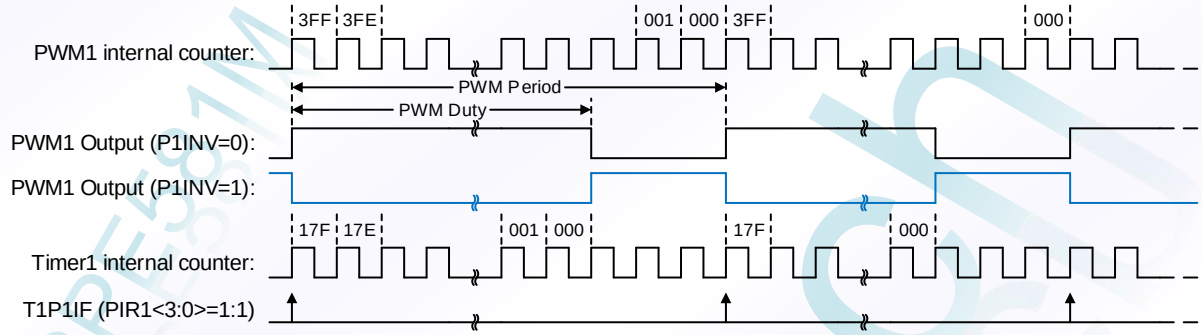
Similarly, this formula can be used directly on PWM2 to PWM6.

For example, F_{OSC} is selected HIRC $8M_{HZ}$, Pre-scaler rate is 1:1, duty time is 48uS PWM calculated as follows:

$$\text{Period-cycle time} = \frac{1024 * \text{Timer1 Pre-scaler rate}}{\text{Timer1 Clock source frequency}} => \frac{1024*1}{8M_{HZ}} = 128uS$$

$$\text{T1RL} = \frac{\text{Duty cycle time} * \text{Timer1 Clock source frequency}}{\text{Timer1 Pre-scaler rate}} - 1 => \frac{48uS * 8M_{HZ}}{1} - 1 => 0x17F$$

Figure 2.5 Normal PWM1 Output Waveform



Example 2.2: Normal PWM1 Setting (F_{OSC}=HIRC 8M$_{HZ}$)

ASM Language Code

```
#include    <8PE581M.ASH>
            …
            // Set PWM1
            MOVIA       0x20
            MOVAR       T1CON           ; CLK source is HIRC, Pre-scaler 1:1
            MOVIA       0x80
            MOVAR       P1CON           ; Normal PWM mode, interrupt rate 1:1
            MOVIA       0x7F
            MOVAR       T1RLLB          ; Low-byte must be written first
            MOVIA       0x01
            MOVAR       T1RLHB          ; Set Duty (0x17F down count to 0x000)
                                        ; Duty time = (0x17F+1)*1/8M_HZ = 48uS
            BSR         T1CON,T1EN_B    ; Start PWM1 (this bit must be last step)

            // Interrupt setting, not required
            BSR         INTEN1,T1P1IE_B ;Enable PWM1 interrupt
            MOVIA       0xFE
            MOVAR       INTFLAG1        ;Clear T1P1IF(PWM) flag (old record)
            BSR         INTEN,GIE_B     ;Enable global interrupt
```

C Language Code

```
#include    <8PE581M.H>
            …
            // Set PWM1
            T1CON=0x20;                 // CLK source is HIRC, Pre-scaler 1:1
            P1CON=0x80;                 // Normal PWM mode, interrupt rate 1:1

            T1RLLB=0x7F;                // Low-byte must be written first
            T1RLHB=0x01;                // Set Duty (0x17F down count to 0x000)
                                        // Duty time = (0x17F+1)*1/8M_HZ = 48uS
            T1CONbits.T1EN=1;           // Start PWM1 (this bit must be last step)

            // Interrupt setting, not required
            INTEN1bits.T1P1IE=1;        // Enable PWM1 interrupt
            INTFLAG1=0xFE;              // Clear T1P1IF(PWM) flag (old record)
            ENI();                      // Enable global interrupt
```

Note: 1. **BCR instruction is not recommended for Clear interrupt flag (INTFLAG and INTFLAG1 register).**
2. **INTFLAGxbits.xxx=0 syntax is not recommended for Clear interrupt flag (INTFLAG and INTFLAG1 register)**

2.4.1 Extension PWM

2.4.1.1 8+2 bits mode

In this extension PWM mode, a PWM cycle is divided into four modulation cycles (modulation cycle 0 to modulation cycle 3). This four modulation cycles, can be defined by M<1:0> (TxRLLB<1:0>, x=1~6), as shown in Figure 2.6. When the corresponding modulation mode is selected, PWM module will increase the duty cycle of the delay, as shown in Table 2.3.
This extension PWM can be applied PWM1 to PWM6, in this mode, duty cycle maximum resolution is 8-bits. Please note that, this extension PWM mode cannot be applied in Cascade PWM mode.

Figure 2.6: TxRL bits allocation in the Extension PWM mode

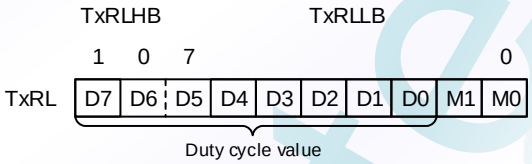
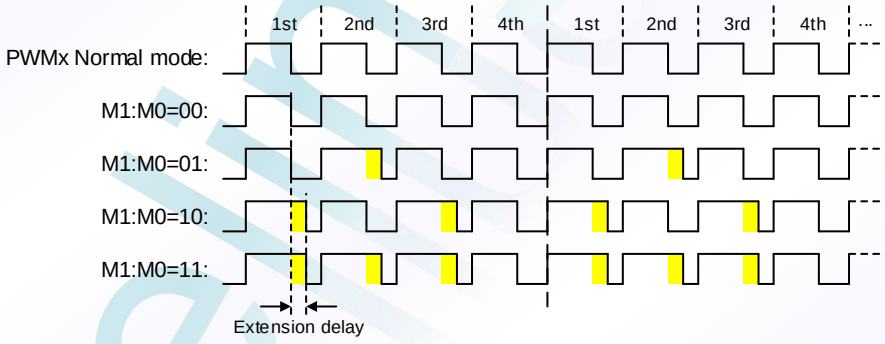


Table 2.3: Stretched cycle number of modulation cycle

M1:M0		Stretched cycle number
0	0	None (Same as Normal mode)
0	1	Only 2nd
1	0	1st and 3rd
1	1	1st, 2nd and 3rd

Figure 2.7: PWM1 to 6 Extension PWM mode Waveform



In this mode, period and duty cycle of the formula refer to the Normal mode.

Extension delay time is calculated as follows:

$$\text{Extension delay time} = \frac{\text{Pre-scaler rate}}{\text{Timer 1 Clock source frequency}}$$

Similarly, this formula can be used directly on PWM2 to PWM6.

2.4.2 Cascade PWM

FM8PE581M provides three CPWM output shared with Timer1&2, Timer3&4 and Timer5&6. When CPWM1, CPWM2 or CPWM3 selected, Timer1 becomes the period-cycle of CPWM1, Timer3 becomes the period-cycle of CPWM2 and Timer5 becomes the period-cycle of CPWM3. And Timer2 will be the duty-cycle of CPWM1, Timer4 will be the duty-cycle of CPWM2 and Timer6 will be the duty-cycle of CPWM3. In this mode, PWM period-time is can be programmable.
The PWM outputs are on the IOB0/CPWM1, IOB2/CPWM2 and IOA4/CPWM3 pins.
The CPWM1, CPWM2 and CPWM3 outputs has a maximum resolution of 10-bits, the duty cycle of the output can vary from 1% to 99%.

The user needs to set the T1MOD bit (P1CON<7>) and CPWM1 bit (P1CON<4>) to enable the CPWM1 output, set the T3MOD bit (P3CON<7>) and CPWM2 bit (P3CON<4>) to enable the CPWM2 output, set the T5MOD bit (P5CON<7>) and CPWM3 bit (P5CON<4>) to enable the PWM3 output.
When T1MOD bit is set, the IOB0/CPWM1 pin is configured as CPWM1 output and forced as an output mode, irrespective of the data direct bit (IOSTB<0>). When the T1MOD is clear, the pin behaves as a I/O pin.
Similarly, T3MOD bit and T5MOD bit configurations corresponding IOB2/CPWM2 and IOA4/CPWM3 pin.

The CPWM1 period time can be calculated as follows:

$$\text{Period time of CPWM1} = \frac{(\text{T1RL}+1) * \text{Timer1 Pre-scaler rate}}{\text{Timer1 Clock source frequency}}$$

or

$$\text{T1RL} = \frac{\text{Period time} * \text{Timer1 Clock source frequency}}{\text{Timer1 Pre-scaler rate}} - 1$$

The CPWM1 duty cycle time can be calculated as follows:

$$\text{Duty cycle time of CPWM1} = \frac{(\text{T2RL}+1) * \text{Timer2 Pre-scaler rate}}{\text{Timer2 Clock source frequency}}$$

or

$$\text{T2RL} = \frac{\text{Duty cycle time} * \text{Timer2 Clock source frequency}}{\text{Timer2 Pre-scaler rate}} - 1$$

Similarly, these formulas can be used directly on CPWM2 and CPWM3.

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

For example, $10K_{HZ}$ CPWM1 output, duty=40%, calculated as follows:

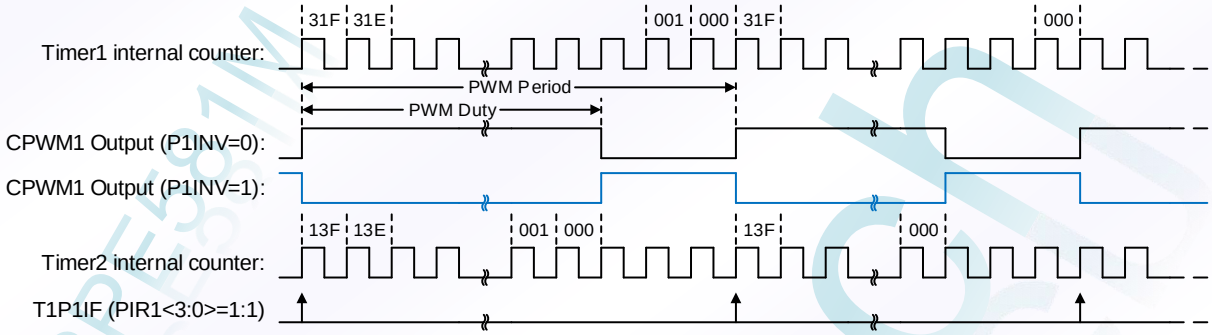
$$\text{Period-cycle time} = \frac{1}{10K_{HZ}} = 100uS, \text{ Duty-cycle time} = 100uS*40\% = 40uS$$

Given operating conditions: F_{OSC}=HIRC $8M_{HZ}$, pre-scaler rate is 1:1

$$\text{T1RL} = \frac{\text{Period time} * \text{Timer1 Clock source frequency}}{\text{Timer1 Pre-scaler rate}} - 1 \Rightarrow \frac{100uS * 8M_{HZ}}{1} - 1 = 0x31F$$

$$\text{T2RL} = \frac{\text{Period time} * \text{Timer2 Clock source frequency}}{\text{Timer2 Pre-scaler rate}} - 1 \Rightarrow \frac{40uS * 8M_{HZ}}{1} - 1 = 0x13F$$

Figure 2.8 CPWM1 Output Waveform



Example 2.3: CPWM1 Setting

```
ASM Language Code
#include    <8PE581M.ASH>
            …
            // Set CPWM1 Period
            MOVIA       0x20
            MOVAR       T1CON           ; CLK source is HIRC, Pre-scaler 1:1
            MOVIA       0x90
            MOVAR       P1CON           ; Cascade mode, interrupt rate 1:1
            MOVIA       0x1F
            MOVAR       T1RLLB          ; Low-byte must be written first
            MOVIA       0x03
            MOVAR       T1RLHB          ; Set period (0x31F down count to 0x000)
                                        ; Period time = (0x31F+1)*1/8MHZ = 100uS
            // Set CPWM1 Duty
            MOVIA       0x20
            MOVAR       T2CON           ; CLK source is HIRC, Pre-scaler 1:1
            MOVIA       0x00
            MOVAR       P2CON           ; Cascade mode, this byte must be keep 0x00
            MOVIA       0x3F
            MOVAR       T2RLLB          ; Low-byte must be written first
            MOVIA       0x01
            MOVAR       T2RLHB          ; Set Duty (0x13F down count to 0x000)
                                        ; Duty time = (0x13F+1)*1/8MHZ = 40uS
            BSR         T1CON,T1EN_B    ; Start CPWM1 (this bit must be last step)

            // Interrupt setting, not required
            BSR         INTEN1,T1P1IE_B ; Enable CPWM interrupt
            MOVIA       0xFE
            MOVAR       INTFLAG1        ; Clear T1P1IF(PWM) flag
            BSR         INTEN,GIE_B     ; Enable global interrupt
            …
```

Note: 1. The PWM duty-cycle time must be less than PWM period-cycle time.

2. **BCR instruction is not recommended for Clear interrupt flag (INTFLAG to INTFLAG2 register).**

C Language Code

```
#include    <8PE581M.H>
            …
            // Set CPWM1 Period
            T1CON=0x20;                 // CLK source is HIRC, Pre-scaler 1:1
            P1CON1=0x90;                // Cascade mode, interrupt rate 1:1

            T1RLLB=0x1F;                // Low-byte must be written first
            T1RLHB=0x03;                // Set period (0x31F down count to 0x000)
                                        // Period time = (0x31F+1)*1/8MHZ = 100uS

            // Set CPWM1 Duty
            T2CON=0x20;                 // CLK source is HIRC, Pre-scaler 1:1
            P2CON1=0x00;                // Cascade mode, this byte must be keep 0x00

            T2RLLB=0x3F;                // Low-byte must be written first
            T2RLHB=0x01;                // Set period (0x13F down count to 0x000)
                                        // Duty time = (0x13F+1)*1/8MHZ = 40uS

            T1CONbits.T1EN=0x1;         // Start CPWM1 (this bit must be last step)

            // Interrupt setting, not required
            INTEN1bits.T1P1IE=1;        // Enable CPWM interrupt
            INTFLAG1=0xFE;              // Clear T1P1IF(PWM) flag
            ENI();                      // Enable global interrupt
            …
```

Note: 1. The PWM duty-cycle time must be less than PWM period-cycle time.

2. **INTFLAGxbits.xxx=0 syntax is not recommended for Clear interrupt flag (INTFLAG to INTFLAG2 register)**

2.5 SPI (Serial Peripheral Interface) Module

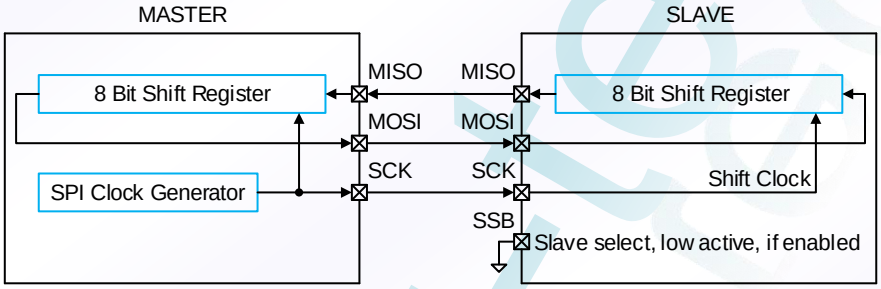
The Serial Peripheral Interface (SPI) Module is a serial interface communicating with other peripheral or microcontroller device.
The SPI mode allows 8-bit of data to be synchronously transmitted and received simultaneously. To accomplish communication, typically four pins are used:

1. Serial Clock (SCK)
2. Serial Data Input (SDI, Master is MISO, Slave is MOSI)
3. Serial Data Output (SDO, Master is MOSI, Slave is MISO)

Additionally, a fourth pin may be used when in a slave mode of operation:

4. Slave Select (SSB)

Figure 2.9: SPI Master-Slave Interconnection



The interconnection between the master and slave devices using SPI is shown in Figure 2.9. The system consists of two shift registers and a master clock generator.
When the SPI master initiates a communication cycle, it pulls down the required slave select pin (SSB pin, if SSB function is used).
Master and Slave prepare the data to be sent in their respective shift registers, and the Master generates the required clock pulses on the SCK line to interchange data.
The master always shifts the data to the slave on the MOSI pin, and the slave always shifts the data to the master on the MISO pin (if SWAP function is not used).
Before starting communication, the clock phase and polarity of the master and the slave, and MSB first or LSB first should be configured in the same mode.
The SPITXB register is not buffered. Writing to SPITXB will also write to SPISR. If write data to SPITXB during the transmission, it will get an incorrect result. In this case, the TXOV flag (SPICON2<6>) will be set to 1 by the hardware.
When the application software expects to receive valid data, it should read SPIRXB before the next byte transfer is complete.
In the slave mode, an overwrite occurs when the SPIRXB is not read after the next byte transfer is completed. In this case, the RXOV (SPICON2<5>) flag will be set to 1 by the hardware. In the master mode, since the SPI communication cycle is controlled by the user software, the RXOV flag will not be set by the hardware when overlay occurs.
When the SPI communication cycle is complete, the SPI interrupt flag SPIIF (INTFLAG<0>) will be set to 1 by the hardware. If the SPI interrupt is enabled, a hardware interrupt will be generated.

Figure 2.10: SPI Block Diagram

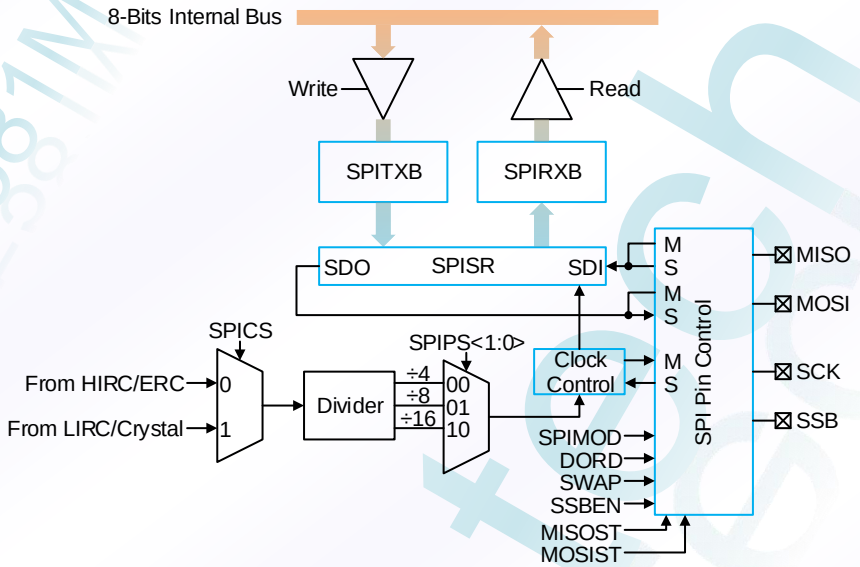


Table 2.4: SPI pin overrides (SWAP=0)

Pin	SSBEN	SPIEN=0 (undefined)		SPIEN=1, MASTER mode		SPIEN=1, SLAVE mode	
		SSB Pin=0	SSB Pin=1	SSB Pin=0	SSB Pin=1	SSB Pin=0	SSB Pin=1
IOA7/SSB	0	I/O=IOSTA7 FN=IOA7		I/O=IOSTA7 FN=IOA7		I/O=IOSTA7 FN=IOA7	
	1	I/O=Input FN=Ignore		X, Don't used		I/O=Input FN=SSB	

Pin	SSBEN	SPIEN=0 (undefined)		SPIEN=1, MASTER mode		SPIEN=1, SLAVE mode	
		MxSxST=0	MxSxST=1	MxSxST=0	MxSxST=1	MxSxST=0	MxSxST=1
IOA0/SCK (SSB Pin=0)	0	I/O=IOSTA0 FN=IOA0	Z	I/O=IOSTA0 FN=SCK	I/O=Output FN=SCK	I/O=IOSTA0 FN=SCK	I/O=Input FN=SCK
	1			X, Don't used			
IOA0/SCK (SSB Pin=1)	0	I/O=IOSTA0 FN=IOA0	Z	I/O=IOSTA0 FN=SCK	I/O=Output FN=SCK	I/O=IOSTA0 FN=SCK	I/O=Input FN=SCK
	1			X, Don't used		I/O=IOSTA0 FN=Ignore	I/O=Input FN=Ignore

Pin	SSBEN	SPIEN=0 (undefined)		SPIEN=1, MASTER mode		SPIEN=1, SLAVE mode	
		MOSIST=0	MOSIST=1	MOSIST=0	MOSIST=1	MOSIST=0	MOSIST=1
IOA1/MOSI (SSB Pin=0)	0	I/O=IOSTA1 FN=IOA1	Z	I/O=IOSTA1 FN=SDO	I/O=Output FN=SDO	I/O=IOSTA1 FN=SDI	I/O=Input FN=SDI
	1			X, Don't used			
IOA1/MOSI (SSB Pin=1)	0	I/O=IOSTA1 FN=IOA1	Z	I/O=IOSTA1 FN=SDO	I/O=Output FN=SDO	I/O=IOSTA1 FN=SDI	I/O=Input FN=SDI
	1			X, Don't used		I/O=IOSTA1 FN=Ignore	I/O=Input FN=Ignore

Legend: I/O = Pin input/output status. FN = Pin function. Z = Keep in input mode, waiting user defined SPI operating mode. **X = Unknown, in the master mode, don't enable SSB function.**

Table 2.4: SPI pin overrides (SWAP=0, continue)

Pin	SSBEN	SPIEN=0 (undefined)		SPIEN=1, MASTER mode		SPIEN=1, SLAVE mode	
		MISOST=0	MISOST=1	MISOST=0	MISOST=1	MISOST=0	MISOST=1
IOA2/MISO (SSB Pin=0)	0	I/O=IOSTA2 FN=IOA2	Z	I/O=IOSTA2 FN=SDI	I/O=Input FN=SDI	I/O=IOSTA2 FN=SDO	I/O=Output FN=SDO
	1			X, Don't used			
IOA2/MISO (SSB Pin=1)	0	I/O=IOSTA2 FN=IOA2	Z	I/O=IOSTA2 FN=SDI	I/O=Input FN=SDI	I/O=IOSTA2 FN=SDO	I/O=Output FN=SDO
	1			X, Don't used		I/O=IOSTA2 FN=Ignore	I/O=Input FN=Ignore

Table 2.5: SPI pin overrides (SWAP=1)

Pin	SSBEN	SPIEN=0 (undefined)		SPIEN=1, MASTER mode		SPIEN=1, SLAVE mode	
		MOSIST=0	MOSIST=1	MOSIST=0	MOSIST=1	MOSIST=0	MOSIST=1
IOA1/MOSI (SSB Pin=0)	0	I/O=IOSTA0 FN=IOA0	Z	I/O=IOSTA1 FN=SDI	I/O=Input FN=SDI	I/O=IOSTA1 FN=SDO	I/O=Output FN=SDO
	1			X, Don't used			
IOA1/MOSI (SSB Pin=1)	0	I/O=IOSTA0 FN=IOA0	Z	I/O=IOSTA1 FN=SDI	I/O=Input FN=SDI	I/O=IOSTA1 FN=SDO	I/O=Output FN=SDO
	1			X, Don't used		I/O=IOSTA1 FN=Ignore	I/O=Input FN=Ignore

Pin	SSBEN	SPIEN=0 (undefined)		SPIEN=1, MASTER mode		SPIEN=1, SLAVE mode	
		MISOST=0	MISOST=1	MISOST=0	MISOST=1	MISOST=0	MISOST=1
IOA2/MISO (SSB Pin=0)	0	I/O=IOSTA2 FN=IOA2	Z	I/O=IOSTA2 FN=SDO	I/O=Output FN=SDO	I/O=IOSTA2 FN=SDI	I/O=Input FN=SDI
	1			X, Don't used			
IOA2/MISO (SSB Pin=1)	0	I/O=IOSTA2 FN=IOA2	Z	I/O=IOSTA2 FN=SDO	I/O=Output FN=SDO	I/O=IOSTA2 FN=SDI	I/O=Input FN=SDI
	1			X, Don't used		I/O=IOSTA2 FN=Ignore	I/O=Input FN=Ignore

Legend: I/O = Pin input/output status. FN = Pin function. Z = Keep in input mode, waiting user defined SPI operating mode. **X = Unknown, in the master mode, don't enable SSB function.**

2.5.1 Data Transfer Mode

There are four combinations of SCK phase and polarity, relative to serial data, which are determined by control bits CPHA (SPICON1<5>) and CPOL (SPICON1<6>).
The SPI data transfer format is shown in Figure 2.11 and Figure 2.12 (Master mode). The data bits are shifted out and latched at the opposite edges of the SCK signal to ensure that the data signal has sufficient time to stabilize.
In the SPI data transfer, the MSB-first or LSB-first can be determined by the DORD (SPICON1<3>) control bit.

Figure 2.11: SPI Transfer Format with CPHA=0

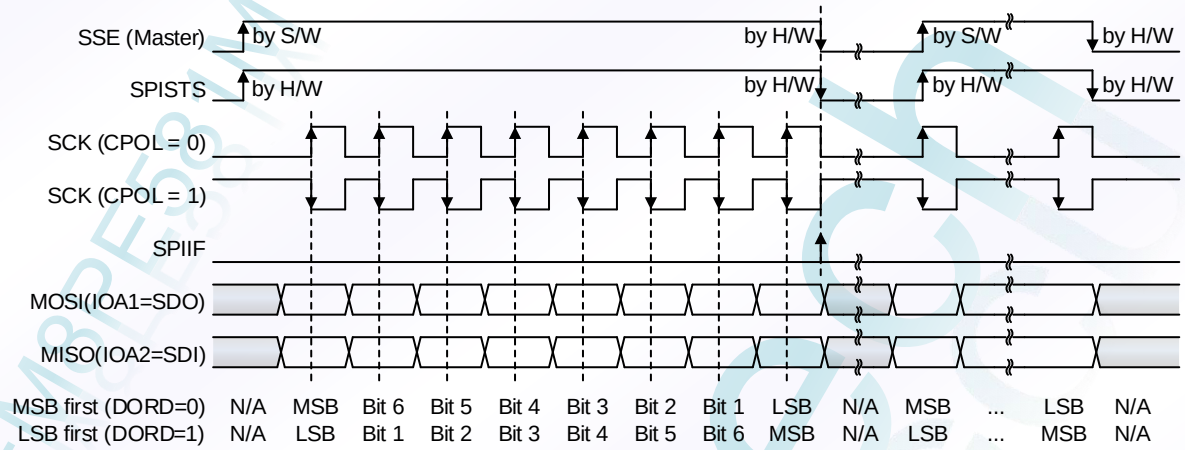
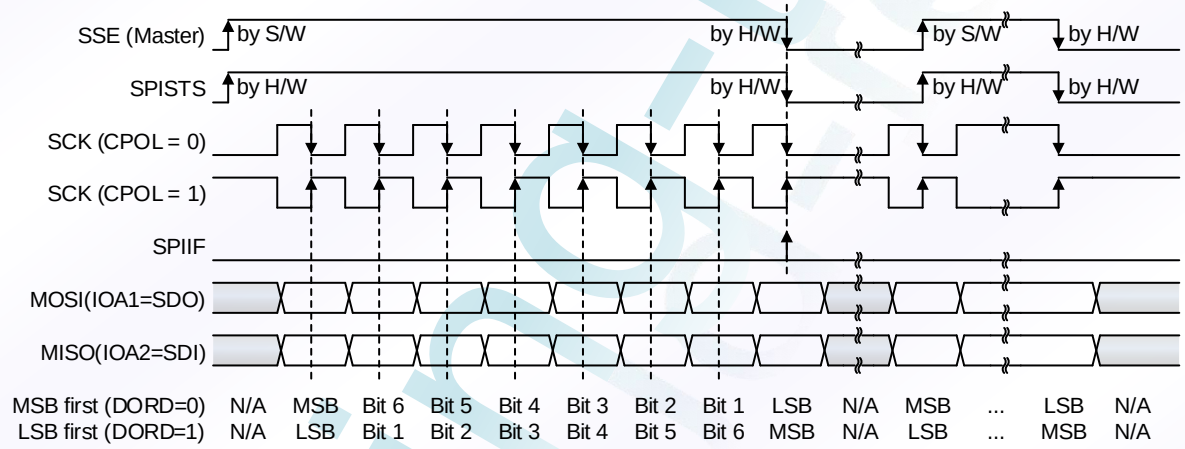


Figure 2.12: SPI Transfer Format with CPHA=1



2.5.2 Master Mode

In master mode, the data is transmitted / received as soon as the SPI shift register enable bit SSE (SPICON1<7>) bit is setting to “1” by S/W. The data in SPITXB will be loaded into SPISR at the same time and start to shift in/out. The SSE bit will be kept in “1” if the communication is still undergoing, and the SSE bit will be cleared by hardware while the shifting is completed. Once the 8-bits of data have been received, the data in SPISR will be moved to the SPIRXB register, interrupt flag bit (SPIIF, INTFLAG<0>) are set. And then user could read out the SPIRXB register before next 8-bit data transmission is completed if needed.

How to transmit/receive data in this master mode:

1. Enable SPI function by setting the SPIEN (SPICON2<7>) bit. And decide master mode by programming SPIMOD (SPICON1<2>) bit.
2. Decide the transmission rate and source by programming SPIPS<1:0> (SPICON1<1:0>) bits. Decide the transmission format by programming CPHA (SPICON1<5>), CPOL (SPICON1<6>) and DORD (SPICON1<3>) bits. Decide the SPI communication pins status by programming SWAP (SPICON1<4>), MISOST (SPICON2<1>), MOSIST (SPICON2<2>) bits.
3. Write the data that you want to transmit to SPITXB register if needed.
4. Set SSE (SPICON1<7>) bit to start transmit.
5. When the 8-bit data transmission is completed, the SSE bit will be reset to “0” by hardware. Therefore, if user wants to transmit/receive another 8-bit data, write next byte data to SPITXB register and set SSE bit to “1” again.

6. When the 8-bit data transmission is completed, the SPIIF (INTFLAG<0>) interrupt flag will set to 1. Besides, the bit is cleared by software.
7. Read out the SPIRXB register before next byte transmission being finished if needed.

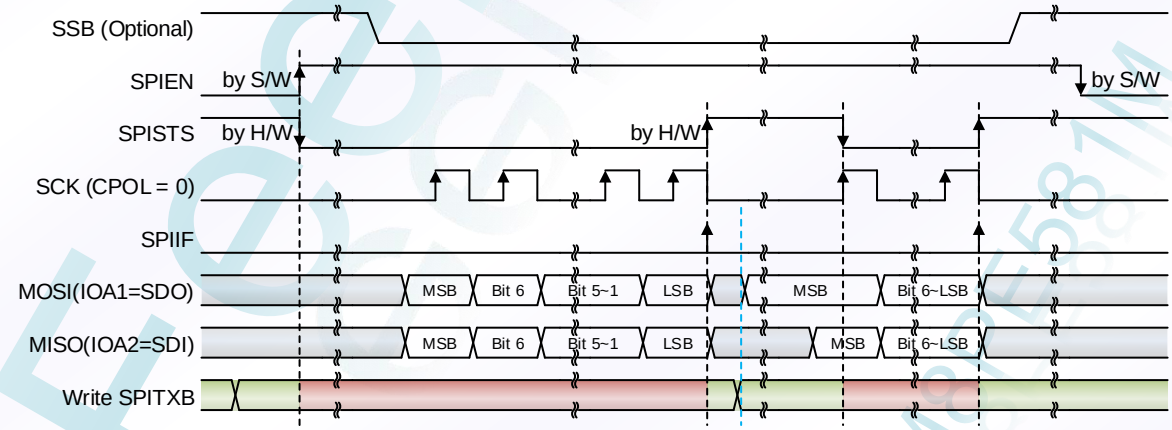
The SPI Master mode waveform, refer Figure 2.11 and Figure 2.12 for detail description.

2.5.3 Slave Mode

In slave mode, the data is transmitted and received as the external clock pulses appear on SCK pin. Once the write data to SPITXB, data will immediate loaded into SPISR and start to shift in/out. The SPISTS (SPICON2<4>) bit will be kept in “0” if the communication is still undergoing, and the SPISTS bit will be settled to "1" by hardware while the shifting is completed. Once the 8-bits of data have been received, the data in SPISR will be moved to the SPIRXB register, interrupt flag bit (SPIIF) are set. And then user could read out the SPIRXB register before next 8-bit data transmission is completed if needed.
The SSB pin allows a synchronous slave mode. The SPI must be in slave mode with SSB pin control enabled (SSBEN=1, SPICON2<0>). When the SSB pin is low, transmission and reception are enabled and the MISO pin is driven (if SWAP is not enabled). When the SSB pin goes high, the MISO pin is no longer driven, even if in the middle of transmitted byte, and becomes a floating output. External pull-up/pull-down resistors may be desirable, depending on the application.

How to transmit/receive data in this slave mode:
1. Enable/disable the SSB pin control by programming SSBEN (SPICON2<0>) bit.
2. Decide the transmission format by programming CPHA (SPICON1<5>), CPOL (SPICON1<6>) and DORD (SPICON1<3>) bits. Decide the SPI communication pins status by programming SWAP (SPICON1<4>), MISOST (SPICON2<1>), MOSIST (SPICON2<2>) bits. The clock phase and polarity of the slave, and MSB-first or LSB-first, must be configured to be the same as master the mode.
3. Write the data that you want to transmit to SPITXB register if needed.
4. SPIEN (SPICON2<7>) bit and wait the external clock pulses appear on SCK pin to start transmit.
5. **Do not write next new data to SPITXB register before this byte transmission being finished!!!**
6. When the 8-bit data transmission is completed, the SPISTS (SPICON2<4>) bit will be reset to “1” by hardware. Therefore, if user wants to transmit/receive another 8-bit data, user must write next byte data to SPITXB register (if needed) before next clock pulse appearing SCK pin.
7. When the 8-bit data transmission is completed, the SPIIF (INTFLAG<0>) interrupt flag will set to 1. Besides, the bit is cleared by software.
8. Read out the SPIRXB register before next byte transmission being finished if needed.

Figure 2.13: SPI Slave Mode Timing (Example CPHA=0, with SSB control enabled)



Note: Do not write next new data to SPITXB in red blocks.

2.6 Interrupt

The FM8PE581M has three kinds of interrupt sources:
1. 6 External IOB<5:3> and IOA<7:5> pin changed interrupt
2. 7 Timers underflow interrupt (or PWM interrupt)
3. SPI interrupt

INTFLAG and INTFLAG1 is the interrupt flag register that recodes the interrupt requests to 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 and INTEN1 register regardless of the status of the GIE bit.
When an interrupt event occurs 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 0x004. 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.
The flag bit in INTFLAG and INTFLAG1 register is set by interrupt event regardless of the status of its mask bit.
Please note that, In the Assembly language code, **BCR instruction is not recommended for Clear interrupt flag** (INTFLAG and INTFLAG1). In the C language code, **Clear bit syntax is not recommended for Clear interrupt flag**, see below example:

Example 2.4: Recommended Clear interrupt flag (Clear T1P1IF, T2P2IF)

ASM Language Code

```
#include    <8PE581M.ASH>
            …
            MOVIA       0xFE        ; To clear bit specified to 0, other remain 1.
            MOVAR       INTFLAG1    ; Clear T1P1IF (bit 0)
            …
            MOVIA       0xFD        ; To clear bit specified to 0, other remain 1.
            MOVAR       INTFLAG1    ; Clear T1P2IF (bit 1)
            …
```

C Language Code

```
#include    <8PE581M.H>
            …
            INTFLAG1=0xFE;          // To clear bit specified to 0, other remain 1.
                                    // Clear T1P1IF (bit 0)
            …
            INTFLAG1=0xFD;          // To clear bit specified to 0, other remain 1.
                                    // Clear T2P2IF (bit 1)
            …
```

2.6.1 PORTB<5:3> and PORTA<7:5> External Interrupt and Wakeup Function

The external interrupt on PORTB<5:3> and PORTA<7:5> are selected by INTPAB<5:0> and PIE (INTEN<5>). When the device is in normal mode and the specified IO status changed, the interrupt event will be triggered and the program will jump to 0x004.
When the device is in sleep mode, those interrupts can also be used as an external wakeup signal. The device will restart system clock and the program will jump to 0x004 after startup timer timeout.

Example 2.5: External IOB5 pin change interrupt

ASM Language Code

```
#include    <8PE581M.ASH>
            …
            … (Backup status code)                  ; 1. IOB5 pin change
            …                                       ; User PORTB pin change ISR code
            MOVIA       0xDF
            MOVAR       INTFLAG                     ; Clear PIF flag(Note1)
            MOVR        PORTB,R                     ; Update PORTB pin status
            … (Restore status code)
            RETFIE
; 2. Return from ISR
            MOVIA       0x20
            MOVAR       IOSTB                       ; Set IOB5 as input
            MOVIA       0xA0
            MOVAR       INTEN                       ; Enable global & PORTB interrupt
            MOVIA       0xDF
            MOVAR       INTFALG                     ; Clear PIF flag(Note1)
            MOVR        PORTB,R                     ; Update PORTB pin status
            MOVIA       0x20
            MOVAR       INTPAB                      ; Set IOB5 pin change
            …
            …
            …
```

Note: 1. BCR instruction is not recommended for Clear interrupt flag (INTFLAG and INTFLAG1 register).
2. Interrupt backup / restore status code are not shown in this example.

C Language Code

```
#include    <8PE581M.H>
void interrupt HW_isr(void) @ HWINT_BASE
{                                                   // 1. IOB5 pin change
            …                                       // User PORTB pin change ISR code
            INTFLAG=0xDF;                           // Clear PIF flag(Note)
            PORTB=PORTB;                            // Update PORTB pin status
}       // 2. Return from ISR

void main(void)
{
            IOSTB=0x20;                             // Set IOB5 as input
            INTEN=0xA0;                             // Enable global & PORTB interrupt
            INTFLAG=0xDF;                           // Clear PIF flag(Note)
            PORTB=PORTB;                            // Update PORTB pin status
            INTPAB=0x20;                            // Set IOB5 pin change
            …
            …
            …
}
```

Note: Clear bit syntax is not recommended for Clear interrupt flag (INTFLAG and INTFLAG1 register).

Example 2.6: External IOB5 pin change wakeup interrupt

ASM Language Code

```
#include    <8PE581M.ASH>
            …
            … (Backup status code)          1. IOB5 pin change
            …                               ; User PORTA & PORTB pin change wakeup ISR code
            MOVIA   0xDF
            MOVAR   INTFLAG                 ; Clear PIF flag(Note1)
            MOVR    PORTB,R                 ; Update PORTB pin status
            … (Restore status code)
            RETFIE
  2. Return from ISR
            MOVIA   0x20
            MOVAR   IOSTB                   ; Set IOB5 as input
            MOVIA   0xA0
            MOVAR   INTEN                   ; Enable global & PORTA & PORTB interrupt
            MOVIA   0xDF
            MOVAR   INTFALG                 ; Clear PIF flag(Note1)
            MOVR    PORTB,R                 ; Update PORTB pin status
            MOVIA   0x20
            MOVAR   INTPAB                  ; Set IOB5 pin change wakeup
            SLEEP
            NOP
            …
```

Note: 1. BCR instruction is not recommended for Clear interrupt flag (INTFLAG and INTFLAG1 register).
2. Interrupt backup / restore status code is not shown in this example.

C Language Code

```
#include    <8PE581M.H>
void interrupt HW_isr(void) @ HWINT_BASE
{                                   1. IOB5 pin change
            …                       // User PORTA & PORTB pin change ISR code
            INTFLAG=0xDF;           // Clear PIF flag(Note)
            PORTB=PORTB;            // Update PORTB pin status
}   2. Return from ISR

void main(void)
{
            IOSTB=0x20;             // Set IOB5 as input
            INTEN=0xA0;             // Enable global & PORTA & PORTB interrupt
            INTFLAG=0xDF;           // Clear PIF flag(Note)
            PORTB=PORTB;            // Update PORTB pin status
            INTPAB=0x20;            // Set IOB5 pin change wakeup
            SLEEP();
            NOP();
            …
}
```

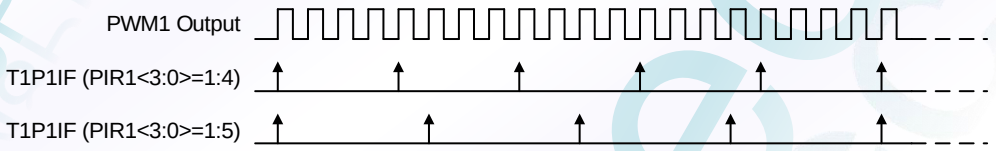
Note: Clear bit syntax is not recommended for Clear interrupt flag (INTFLAG and INTFLAG1 register).

2.6.2 Timer 1~7 Interrupt's

2.6.2.1 Timer 1 interrupt

At Timer mode, an underflow (0x000 → 0x3FF) in the Timer1 counter will set the flag bit T1P1IF (INTFLAG1<0>).
At Normal or Extension PWM or Cascade PWM mode, the end of each PWM period-cycle to generate an interrupt.
The interrupt rate can be adjusted by P1CON<3:0>.
The T1P1IF bit can be cleared by software. This interrupt can be disabled by clearing T1P1IE bit (INTEN1<0>).

Figure 2.14: PWM Interrupt Waveform



2.6.2.2 Timer 2 interrupt

At Timer mode, an underflow (0x000 → 0x3FF) in the Timer2 counter will set the flag bit T2P2IF (INTFLAG1<1>).
At Normal or Extension PWM mode, the end of each PWM period-cycle to generate an interrupt. The interrupt rate can be adjusted by P2CON<3:0>.
At Cascade PWM mode, this interrupt does not occur, only PWM (T1P1IF) interrupts occur.
The T2P2IF bit can be cleared by software. This interrupt can be disabled by clearing T2P2IE bit (INTEN1<1>).

2.6.2.3 Timer 3 interrupt

At Timer mode, an underflow (0x000 → 0x3FF) in the Timer3 counter will set the flag bit T3P3IF (INTFLAG1<2>).
At Normal or Extension PWM or Cascade PWM mode, the end of each PWM period-cycle to generate an interrupt.
The interrupt rate can be adjusted by P3CON<3:0>.
The T3P3IF bit can be cleared by software. This interrupt can be disabled by clearing T3P3IE bit (INTEN1<2>).

2.6.2.4 Timer 4 interrupt

At Timer mode, an underflow (0x000 → 0x3FF) in the Timer4 counter will set the flag bit T4P4IF (INTFLAG1<3>).
At Normal or Extension PWM mode, the end of each PWM period-cycle to generate an interrupt. The interrupt rate can be adjusted by P4CON<3:0>.
At Cascade PWM mode, this interrupt does not occur, only PWM (T3P3IF) interrupts occur.
The T2P2IF bit can be cleared by software. This interrupt can be disabled by clearing T4P4IE bit (INTEN1<3>).

2.6.2.5 Timer 5 interrupt

At Timer mode, an underflow (0x000 → 0x3FF) in the Timer5 counter will set the flag bit T5P5IF (INTFLAG1<4>).
At Normal or Extension PWM or Cascade PWM mode, the end of each PWM period-cycle to generate an interrupt.
The interrupt rate can be adjusted by P5CON<3:0>.
The T5P5IF bit can be cleared by software. This interrupt can be disabled by clearing T5P5IE bit (INTEN1<4>).

2.6.2.6 Timer 6 interrupt

At Timer mode, an underflow (0x000 → 0x3FF) in the Timer6 counter will set the flag bit T6P6IF (INTFLAG1<5>).
At Normal or Extension PWM mode, the end of each PWM period-cycle to generate an interrupt. The interrupt rate can be adjusted by P6CON<3:0>.
At Cascade PWM mode, this interrupt does not occur, only PWM (T5P5IF) interrupts occur.
The T6P6IF bit can be cleared by software. This interrupt can be disabled by clearing T6P6IE bit (INTEN1<5>).

2.6.2.7 Timer 7 interrupt

An underflow (0x00 → 0xFF) in the Timer7 counter will set the flag bit T7IF (INTFLAG1<6>). And the T7IF bit can be cleared by software. This interrupt can be disabled by clearing T7IE bit (INTEN1<6>).

2.6.3 SPI interrupt

When the transmitted complete, SPIIF bit (INTFLAG<0>) will be set. And the SPIIF bit can be cleared by software. This interrupt can be disabled by clearing SPIIE bit (INTEN<0>).

2.7 Dual Clock Function

The chip can be operated in three different dual clock function, users need to use it, and the configuration word must be set to one of following:

- LIRC & HIRC
- XT & HIRC
- LF & HIRC

If not in these states, will not be able to use dual clock function. By default, the system is the use of internal IRC frequency as the clock source, and the two oscillator circuit is in the enable state. If not used, turn off unused oscillator power (via control register OSCCON), can be reduce unnecessary current consumption.
When you want to switch clock source, recommend follow these steps:

1. Turn-on another oscillator power.
2. Wait oscillator to stable (XT and LF mode requires this step).
3. Set WDT pre-scaler to 1:128 and Clear Watch-dog (avoid watchdog overflow).
4. Set or Clear CLKSW bit (OSCCON<7>) to switch to another clock source.
5. Wait two NOP instruction (Required sequence).
6. Clear Watch-dog and set back to original settings.
7. If original oscillator not used, turn-off it.

Since the oscillator from the off state to the normal output clock oscillator needs some time to wait for a stable, at each oscillation mode, we recommend waiting time should be greater than the following table:

Table 2.6: Recommend typical wait time

Situation	Typical waiting time
Crystal → HIRC	10uS
HIRC → Crystal (XT, 4 to 20MHZ)	20mS
HIRC → Crystal (LF, 32KHZ)	5~370mS

Note: 1. This table is for reference only.
2. Quartz crystal characteristics vary according to type, package and manufacturer, the users must be carefully tested and verified.
3. RC oscillator mode will change depending on the operating voltage, the user must carefully tested and verified.

Example 2.7: Switching from HIRC to Crystal

ASM Language Code

```
#include    <8PE581M.ASH>
            …
            BCR     OSCCON,ECLKPD_B   ; Turn-on External oscillator
            CALL    Delay             ; Wait Crystal oscillator to stable
            MOVIA   0x87
            MOVAR   WDTCON            ; If Watch-dog enable, recommend set to 1:128
            CLRWDT                    ; If Watch-dog enable, clean it!
            BSR     OSCCON,CLKSW_B        ; Switching clock from HIRC to Crystal
            NOP                       } Required sequence
            NOP
            CLRWDT                    ; If Watch-dog enable, clean it!
            BSR     OSCCON,IRCPD_B    ; Turn-off HIRC oscillator (if unused)
            MOVIA   0xnn
            MOVAR   WDTCON            ; Set back original settings (if Watch-dog used)
            …
```

Similarly, switching clock from Crystal to HIRC also this procedure.

```
C Language Code
#include    <8PE581M.H>
            …
            OSCCONbits.ECLKPD=0;        // Turn-on External oscillator
            Delay ();                   // Wait Crystal oscillator to stable
            WDTCON=0x87;                // If Watch-dog enable, recommend set to 1:128
            CLRWDT ();                  // If Watch-dog enable, clean it!

            OSCCONbits.CLKSW=0;             // Switching clock from HIRC to Crystal
            NOP();                      } Required sequence
            NOP();
            CLRWDT ();                  // If Watch-dog enable, clean it!
            OSCCONbits.IRCPD=1;         // Turn-off HIRC oscillator (if unused)
            WDTCON=0xnn;                // Set back original settings (if Watch-dog used)
            …
```

Similarly, switching clock from Crystal to HIRC also this procedure.

2.8 Watch Dog Timer (WDT)

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 OSCI and OSCO pins is turned off, such as in SLEEP mode. The WDT can be disabled by clearing the control bit WDTEN (WDTCON <7>) to “0”.

The WDT has a typical time-out period of 20mS (without pre-scaler). This period of this timer may be variant slightly because of temperature, voltage, and process variation. 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 WDTCON register <2:0>. Thus, the longest time-out period is approximately 2.56 seconds.
The CLRWDT instruction clears the WDT and prevents it from timing out and generating a device reset.
The SLEEP instruction also resets the WDT. This gives the maximum SLEEP time before a WDT Wake-up Reset.

There are two type of watchdog timer mode could be selected by I_WDT (WDTCON <6>). When I_WDT bit disable, normal watchdog timer reset is selected. During normal operation or in SLEEP mode, a WDT time-out will cause the device reset and the $\overline{TO}$ bit (STATUS<4>) will be cleared.
If I_WDT bit enabled, the internal watchdog timer wakeup will be used. The system wakeups from sleep, then jumps into interrupt vector with external interrupt request PIF (PORTA and PORTB interrupt, INTFLAG<5>) and continues from next instruction instead of triggering a reset event. There is a stabilization time required for internal watchdog wakeup could be selected by I_ TWDT (WDTCON<5>). The default value of this stabilization timer is 2.5ms.

Example 2.8: Internal Watchdog Wakeup

ASM Language Code

```
#include    <8PE581M.ASH>
            …
            … (Backup status code)
            …                               ; User WDT Wakeup ISR code
            MOVIA       0xDF
            MOVAR       INTFLAG             ; Clear PIF flag(Note1)
            … (Restore status code)
            RETFIE
            MOVIA       0xA0
            MOVAR       INTEN               ; Enable global & PORTA & PORTB interrupt
            CLRWDT
            MOVIA       0xE7
            MOVAR       WDTCON              ; Sleep: 2.56S + Wakeup:1.25mS
            …
            …
            SLEEP
            NOP
            …
```

1. WDT Wakeup

2. Return from ISR

Note: 1. BCR instruction is not recommended for Clear interrupt flag (INTFLAG and INTFLAG1 register).
2. Interrupt backup / restore status code are not shown in this example.

C Language Code

```
#include    <8PE581M.H>
void interrupt HW_isr(void) @ HWINT_BASE
{
            …                               // User WDT Wakeup ISR code
            INTFLAG=0xDF;                   // Clear PIF flag(Note)
}

void main(void)
{
            …
            INTEN=0xA0;                     // Enable global & PORTA & PORTB interrupt
            CLRWDT();
            WDTCON=0xE7;                    // Sleep: 2.56S + Wakeup:1.25mS
            …
            …
            SLEEP();
            NOP();
            …
}
```

1. WDT Wakeup

2. Return from ISR

Note: Clear bit syntax is not recommended for Clear interrupt flag (INTFLAG and INTFLAG1 register).

Example 2.9: Typical Watchdog Reset

ASM Language Code

```
#include    <8PE581M.ASH>
            …
            …                                   WDT Reset
            CLRWDT
            MOVIA       0x87
            MOVAR       WDTCON          ; Sleep: 2.56S + Wakeup:20mS
            …
            …
            SLEEP
            NOP
            …
            …
```

C Language Code

```
#include    <8PE581M.H>
void main(void)
{
            …                                   WDT Reset
            …
            CLRWDT();
            WDTCON=0x87;                // Sleep: 2.56S + Wakeup:1.25mS
            …
            …
            SLEEP();
            NOP();
            …
}
```

2.9 Power-down Mode (SLEEP)

Power-down mode is entered by executing a SLEEP instruction.
When SLEEP instruction is executed, the $\overline{PD}$ bit (STATUS<3>) is cleared, the $\overline{TO}$ bit is set, the watchdog timer will be cleared and keeps running, and the oscillator driver is turned off.
All I/O pins maintain the status they had before the SLEEP instruction was executed.

2.9.1 Wake-up from SLEEP Mode

The device can wake-up from SLEEP mode through one of the following events:
1. RSTB reset.
2. WDT time-out reset (if enabled).
3. 6 External IOB<5:3> and IOA<7:5> pin changed interrupt.
4. SPI interrupt (Slave mode).

External RSTB reset and WDT time-out reset will cause a device reset. The $\overline{PD}$ and $\overline{TO}$ bits can be used to determine the cause of device reset. The $\overline{PD}$ bit is set on power-up and is cleared when SLEEP instruction is executed. The $\overline{TO}$ bit is cleared if a WDT time-out occurred.
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 not be wake-up. If the GIE bit is set, the device will branch to the interrupt address (0x004).
The system wake-up delay time, please refer to Table 2.7.

2.10 Hold Mode (HALT)

Hold mode is entered by executing a HALT instruction.
When HALT instruction is executed, the $\overline{PD}$ bit (STATUS<3>) is cleared, the $\overline{TO}$ bit is set, the watchdog timer will be cleared and keeps running, and the CPU oscillator driver is turned off.
When the HCS bit is cleared to "0", the system oscillator will not be turned off, peripheral modules will remain running.
All I/O pins maintain the status they had before the HALT instruction was executed (without peripheral module output pin).

2.10.1 Wake-up from HALT Mode

The device can wake-up from HALT mode through one of the following events:
1. RSTB reset.
2. WDT time-out reset (if enabled).
3. 6 External IOB<5:3> and IOA<7:5> pin changed interrupt.
4. 7 Timers underflow interrupt (or PWM interrupt).
5. SPI interrupt (Master/Slave mode).

External RSTB reset and WDT time-out reset will cause a device reset. The $\overline{PD}$ and $\overline{TO}$ bits can be used to determine the cause of device reset. The $\overline{PD}$ bit is set on power-up and is cleared when SLEEP instruction is executed. The $\overline{TO}$ bit is cleared if a WDT time-out occurred.
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 not be wake-up. If the GIE bit is set, the device will branch to the interrupt address (0x004).

Table 2.7: SLEEP & HALT Wake-up time

Instruction	CPU	Oscillator		Wake-up time		
		HCS=0	HCS=1	HRT=0	HRT=1	
					HCS=0	HCS=1
HALT	Stopped	Running	Stopped	160uS+64F_{OSC}	160uS	20ms, 5ms, 1ms or 160us (Note2)
SLEEP	Stopped	Stopped	Stopped	WDTEN=0, 20ms, 5ms, 1ms or 160uS (Note2) WDTEN=1, 20mS		

Note: 1. If the starting oscillator is a crystal oscillator, do not select too short a wake-up time, otherwise an accident may occur.
2. Set-up time 20ms, 5ms, 1ms or 160uS, defined by SUT bit of configuration word.

2.11 Reset

FM8PE581M device may be RESET in one of the following ways:
1. Power-on Reset (POR)
2. Brown-out Reset (BOR)
3. RSTB Pin Reset
4. WDT time-out Reset

Some registers are not affected in any RESET condition. Their status is unknown on Power-on Reset and unchanged in any other RESET. Most other registers are reset to a "reset state" on Power-on Reset, RSTB or WDT Reset.
A Power-on RESET pulse is generated on-chip when V_{DD} rise is detected. To use this feature, the user merely ties the RSTB pin to V_{DD}.
On-chip Low Voltage Detector (LVDT) places the device into reset when V_{DD} is below a fixed voltage. This ensures that the device does not continue program execution outside the valid operation V_{DD} range. Brown-out RESET is typically used in AC line or heavy loads switched applications.
A RSTB or WDT Wake-up from SLEEP also results in a device RESET, and not a continuation of operation before SLEEP.

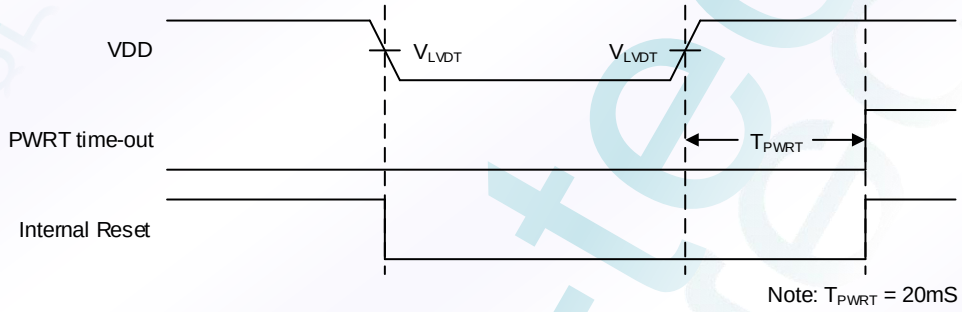
The $\overline{TO}$ and $\overline{PD}$ bits (STATUS<4:3>) are set or cleared depending on the different reset conditions.

2.11.1 Power-up Reset Timer (PWRT)

The Power-up Reset Timer provides a nominal 20ms 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. The PWDT delay will vary from device to device due to V_{DD}, temperature, and process variation.

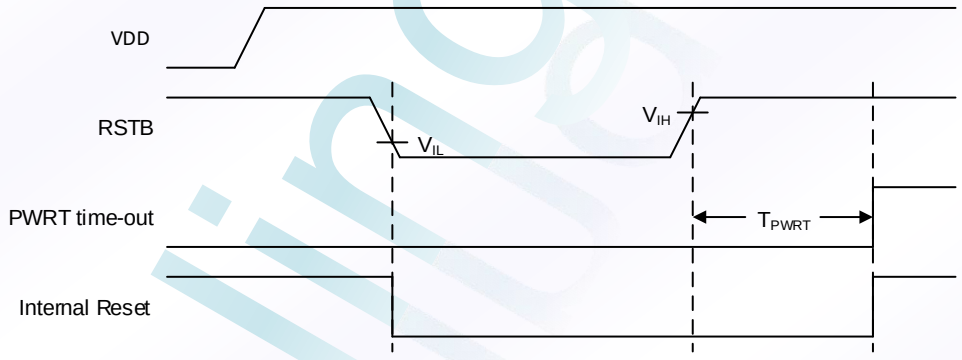
Figure 2.15: Reset Timing

Case1: LVDT ON, RSTB Disable



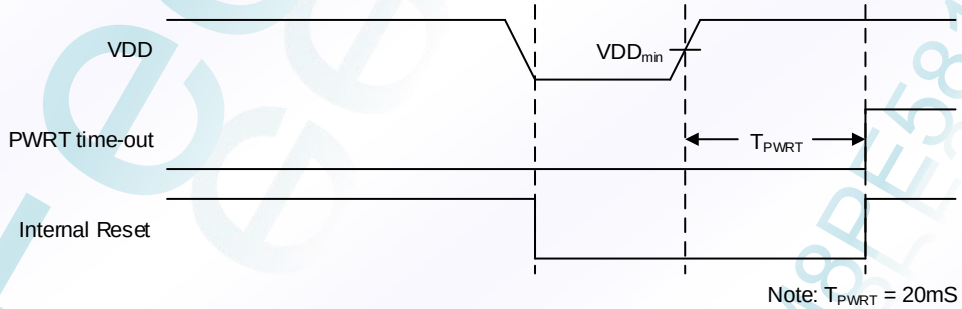
Note: T_{PWRT} = 20mS

Case2: LVDT OFF, RSTB Enable



Note: T_{PWRT} = 20mS

Case3: LVDT OFF, RSTB Disable



Note: T_{PWRT} = 20mS

Figure 2.16: Simplified Block Diagram of on-chip Reset Circuit

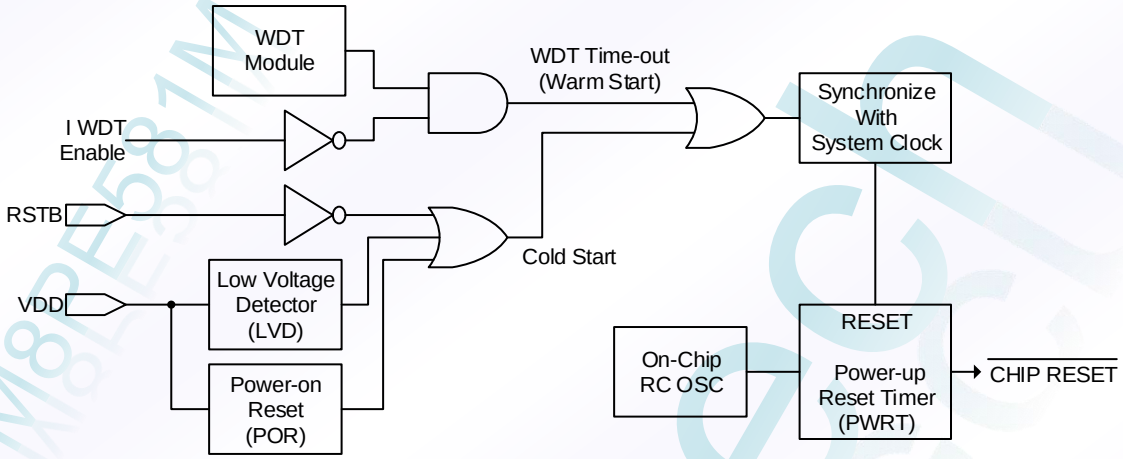


Table 2.8: Reset Conditions for Operational Registers

Register	Address	Power-on Reset Brown-out Reset	WDT Reset RSTB Reset
ACC	N/A	xxxx xxxx	uuuu uuuu
INDF	0x000	xxxx xxxx	uuuu uuuu
PCL	0x002	0000 0000	0000 0000
STATUS	0x003	---1 1xxx	---# #xxx
FSRL	0x004	xxxx xxxx	uuuu uuuu
FSRH	0x005	---- ---x	---- ---u
PCHBUF	0x006	---x xxxx	---u uuuu
IOSTA	0x007	1111 1111	1111 1111
PORTA	0x008	xxxx xxxx	uuuu uuuu
IOSTB	0x009	1111 1111	1111 1111
PORTB	0x00A	xxxx xxxx	uuuu uuuu
T1CON	0x00B	0-00 0000	0-00 0000
P1CON	0x00C	0000 0000	0000 0000
T1RLLB	0x00D	xxxx xxxx	uuuu uuuu
T1RLHB	0x00E	---- --xx	---- --uu
T2CON	0x00F	0-00 0000	0-00 0000
P2CON	0x010	000- 0000	000- 0000
T2RLLB	0x011	xxxx xxxx	uuuu uuuu
T2RLHB	0x012	---- --xx	---- --uu
T3CON	0x013	0-00 0000	0-00 0000
P3CON	0x014	0000 0000	0000 0000
T3RLLB	0x015	xxxx xxxx	uuuu uuuu
T3RLHB	0x016	---- --xx	---- --uu
T4CON	0x017	0-00 0000	0-00 0000
P4CON	0x018	000- 0000	000- 0000
T4RLLB	0x019	xxxx xxxx	uuuu uuuu
T4RLHB	0x01A	---- --xx	---- --uu
T5CON	0x01B	0-00 0000	0-00 0000

Register	Address	Power-on Reset Brown-out Reset	WDT Reset RSTB Reset
P5CON	0x01C	0000 0000	0000 0000
T5RLLB	0x01D	xxxx xxxx	uuuu uuuu
T5RLHB	0x01E	---- --xx	---- --uu
T6CON	0x01F	0-00 0000	0-00 0000
P6CON	0x020	000- 0000	000- 0000
T6RLLB	0x021	xxxx xxxx	uuuu uuuu
T6RLHB	0x022	---- --xx	---- --uu
T7CON	0x023	0-00 0000	0-00 0000
T7RL	0x024	1111 1111	1111 1111
T7CNT	0x025	1111 1111	1111 1111
SPICON1	0x026	0000 0000	0000 0000
SPICON2	0x027	000x -000	000u -000
SPITXB	0x028	0000 0000	0000 0000
SPIRXB	0x029	xxxx xxxx	uuuu uuuu
INTEN	0x02A	0-0- ---0	0-0- ---0
INTEN1	0x02B	-000 0000	-000 0000
INTFLAG	0x02C	--0- ---0	--0- ---0
INTFLAG1	0x02D	-000 0000	-000 0000
APHCON	0x02E	0000 0000	0000 0000
BPHCON	0x02F	0000 -000	0000 -000
PLCON	0x030	--%% %%%%	--%% %%%%
CLOCON	0x031	00-0 0000	00-0 0000
WDTCON	0x032	1000 -111	1000 -111
INTPAB	0x033	--00 0000	--00 0000
OSCCON	0x034	0-00 0-00	0-00 0-00
General Purpose Registers	0x080 ~ 0x1FF	xxxx xxxx	uuuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented, # = refer to the following table for possible values.
% = refer to the configuration bit "PPDS".

Table 2.9: $\overline{TO}$ and $\overline{PD}$ Status after Reset

$\overline{TO}$	$\overline{PD}$	RESET was caused by
0	0	WDT timer overflow from SLEEP / HALT mode
0	1	WDT timer overflow from normal mode
1	0	Set "low" at RESETB from SLEEP / HALT mode
1	1	Power on reset / Brown-out reset
u	u	Set "low" at RESETB from normal mode

Legend: u = unchanged.

Table 2.10: Events Affecting$\overline{TO}$ / $\overline{PD}$ Status Bits

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

Legend: u = unchanged.

2.12 Oscillator Configurations

FM8PE581M can be operated in eight different combinations of oscillator modes. Users can program configuration word (F_{OSC}) to select the appropriate modes. The five different system clock modes are combination of the following oscillators:

- LF: Low Frequency Crystal Oscillator.
- XT: Crystal/Resonator Oscillator.
- ERC: External Resistor/Voltage Controlled Oscillator.
- HIRC: High-speed Internal Resistor/Capacitor Oscillator.
- LIRC: Low-speed Internal Resistor/Capacitor Oscillator.

In LF or XT modes, a crystal or ceramic resonator in connected to the OSCI and OSCO pins to establish oscillation. When in LF or XT modes, the devices can have an external clock source drive the OSCI pin.
The ERC device option offers additional cost savings for timing insensitive applications. The RC oscillator frequency is a function of the supply voltage, the resistor (Rext) and capacitor (Cext), the operating temperature, and the process parameter.
The HIRC/LIRC option offers largest cost savings for timing insensitive applications.

Figure 2.17: XT or LF Oscillator Modes (Crystal Operation or Ceramic Resonator)

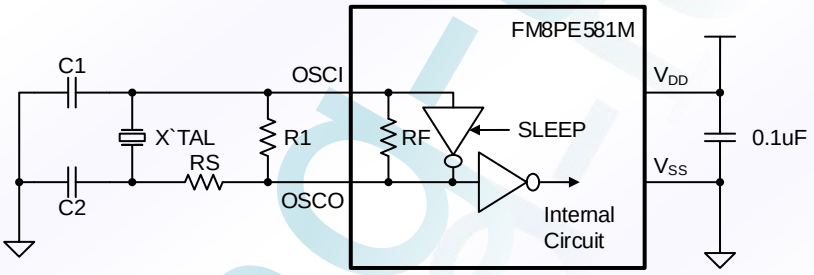


Figure 2.18: XT or LF Oscillator Modes (External Clock Input Operation)

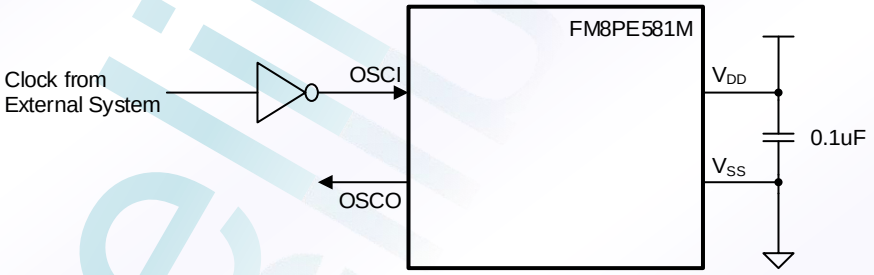
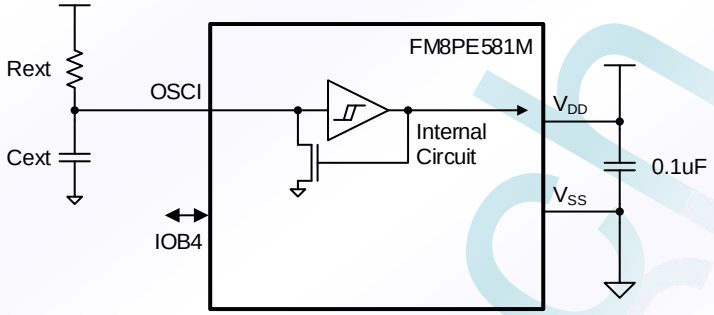


Figure 2.19: ERC Oscillator Mode (External RC Oscillator)

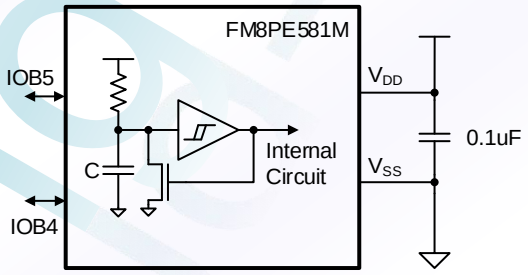


The typical oscillator frequency vs. external resistor is as following table
When Cext = 0.01uF (103)

Frequency	Rext @ 3V	Rext @ 5V
455K$_{HZ}$	984K	1465K
1M$_{HZ}$	700K	929K
4M$_{HZ}$	282K	314K
8M$_{HZ}$	154K	167K

Note: Values are provided for design reference only.

Figure 2.20: HIRC or LIRC Oscillator Mode (Internal R, Internal C Oscillator)



2.13 Configuration Words

Table 2.11: Configuration Words

Name	Description
F_{OSC}	Oscillator Selection Bit → HIRC mode (default) → LIRC & HIRC mode → XT crystal & HIRC mode → LF crystal & HIRC mode → ERC mode → LIRC mode → XT crystal mode → LF crystal mode
CPU_S	Instruction Period Selection Bit → four oscillator periods (4T) (default) → two oscillator periods (2T)
LVDT	Low Voltage Detector Selection Bit → LVDT = 3.0V (default) → LVDT = 2.3V → LVDT = 2.1V → LVDT = 2.0V → LVDT = 1.8V
RSTBIN	IOB3/RSTB Pin Selection Bit → RSTB pin is selected (default) → IOB3 pin is selected
SUT	Reset Set-up Time Selection Bit → 20mS (default) → 5mS → 1mS → 160uS
HIRCS	High-speed IRC Frequency Selection Bit → $16M_{HZ}$ → $8M_{HZ}$ (default)
SYS_CK	System Clock Selection Bit → HIRC/8 → HIRC/4 → HIRC/2 → HIRC (default)
LIRCS	Low-speed IRC Frequency Selection Bit → $500K_{HZ}$ → $250K_{HZ}$ (default)
WDTEN	Watchdog Timer Enable Bit → WDT enabled (default) → WDT disabled
OSCOUT	IOB4/OSCO Pin Selection Bit for ERC Mode → IOB4 pin is selected (default) → OSCO pin is selected
PPDS	IOA5~IOA3 & IOB2~IOB0 Pre-pull down Selection Bit → Enable → Disable (default)
PROTECT	Code Protection Bit → NO, OTP code protection off (default) → YES, OTP code protection on

Table 2.12: Selection of IOB5/OSCI and IOB4/OSCO Pin

Mode of oscillation	IOB5/OSCI	IOB4/OSCO
HIRC, LIRC, HIRC & LIRC	Force to IOB5	Force to IOB4
ERC	Force to OSCI	IOB4/OSCO selected by OSCOUT bit
XT, LF, XT & HIRC, LF & HIRC	Force to OSCI	Force to OSCO

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	00h → WDT, 00h → WDT pre-scaler	1	$\overline{TO}$,$\overline{PD}$
SLEEP		Go into power-down mode	0x00 → WDT, 0x00 → WDT pre-scaler	1	$\overline{TO}$,$\overline{PD}$
HALT		Go into hold 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	-
CLRA		Clear ACC	00h → ACC	1	Z
CLRR	**R**	Clear R	00h → 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
XORIA	**I**	Exclusive OR Immediate to ACC	ACC xor I → ACC	1	Z

Mnemonic, Operands		Description	Operation	Cycles	Status Affected
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<12:0>	2	-
GOTO	**I**	Unconditional branch	I → PC<12:0>	2	-

Note:1.2 cycles for skip, else 1 cycle.

2. bit: Bit address within an 8-bit register R
R: Register address (0x000 to 0x1FF)
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
PCHBUF: Program Counter High-byte buffer
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:	0≤R≤0x1FF 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:	0≤R≤0x1FF 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:	0≤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:	0≤R≤0x1FF 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:	0≤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:	0≤R≤0x1FF 0≤b≤7
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:	0≤R≤0x1FF 0≤b≤7
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:	0≤R≤0x1FF 0≤b≤7
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:	0≤R≤0x1FF 0≤b≤7
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:	0≤I≤0x1FFF
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 13-bit immediate address is loaded into PC bits <12:0>.
Cycles:	2

CLRA	Clear ACC
Syntax:	CLRA
Operands:	None
Operation:	00h → 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:	0≤R≤0x1FF
Operation:	00h → 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. The status bits $\overline{TO}$ and $\overline{PD}$ will be set.
Cycles:	1

COMR	Complement R
Syntax:	COMR R, d
Operands:	0≤R≤0x1FF 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:	0≤R≤0x1FF 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:	0≤R≤0x1FF 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:	0≤I≤0x1FFF
Operation:	I → PC<12:0>
Status Affected:	None
Description:	GOTO is an unconditional branch. The 13-bit immediate value is loaded into PC bits <12:0>.
Cycles:	2

HALT	Enter HALT Mode
Syntax:	HALT
Operands:	None
Operation:	0x00 → WDT; 0x00 → WDT pre-scaler; 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 and its pre-scaler cleared. The processor is put into HALT mode; system clock related peripherals still work.
Cycles:	1

INCR	Increment R
Syntax:	INCR R, d
Operands:	0≤R≤0x1FF 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:	0≤R≤0x1FF 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:	0≤R≤0x1FF 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:	0≤I≤0x1FF
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

MOVAR	Move ACC to R
Syntax:	MOVAR R
Operands:	0≤R≤0x1FF
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:	0≤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:	0≤R≤0x1FF 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:	0≤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:	0≤R≤0x1FF 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:	0≤R≤0x1FF 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 SLEEP Mode
Syntax:	SLEEP
Operands:	None
Operation:	0x00 → WDT; 0x00 → WDT pre-scaler; 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 and its pre-scaler cleared. The processor is put into SLEEP mode.
Cycles:	1

SBCAR	Subtract ACC from R with Carry
Syntax:	SBCAR R, d
Operands:	0≤R≤0x1FF 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:	0≤R≤0x1FF 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:	0≤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:	0≤R≤0x1FF 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

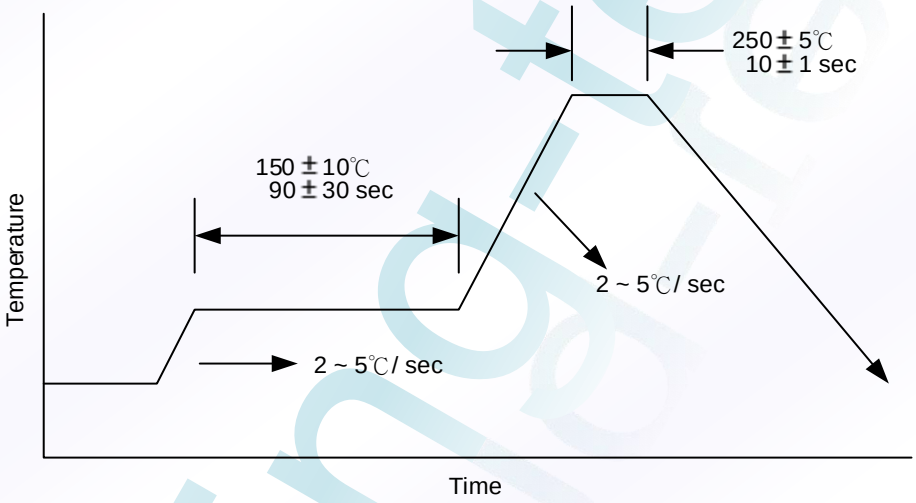
XORAR	Exclusive OR ACC with R
Syntax:	XORAR R, d
Operands:	0≤R≤0x1FF d∈[0,1]
Operation:	ACC xor R → dest
Status Affected:	Z
Description:	Exclusive OR the contents of the ACC register with register 'R'. If 'd' is 0 the result is stored in the ACC register. If 'd' is 1 the result is stored back in register 'R'.
Cycles:	1

XORIA	**Exclusive OR Immediate with ACC**
Syntax:	XORIA I
Operands:	0≤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	-	0	-	70	°C
-	Store Temperature	-	-65	-	150	°C
V_{DD}	DC Supply Voltage	-	0	-	6.0	V
-	Input Voltage with respect to Ground	-	-0.3	-	V_{DD}+0.3	V
-	ESD Susceptibility	HBM (Human Body Mode)	-	2.5	-	KV
		MM (Machine Mode)	-	200	-	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	-	2.2	-	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 speed Internal RC Oscillation range	3V	16M$_{HZ}$ HIRC, V_{DD}=High	-5%	16	+5%	M$_{HZ}$
		5V	16M$_{HZ}$ HIRC, V_{DD}=Low	-5%	16	+5%	
		3V	8M$_{HZ}$ HIRC, V_{DD}=High	-5%	8	+5%	
		5V	8M$_{HZ}$ HIRC, V_{DD}=Low	-5%	8	+5%	
F_{LIRC}	Low speed Internal RC Oscillation range	3V	500K$_{HZ}$ LIRC mode	-	460	-	K$_{HZ}$
		5V		-	468	-	
		3V	250K$_{HZ}$ LIRC mode	-	260	-	
		5V		-	259	-	
F_{ERC}	ERC Oscillation range	3V	ERC mode	0.455	-	8	M$_{HZ}$
		5V		0.455	-	16	
F_{XT}	Crystal Oscillation range	3V	XT mode	0.455	-	8	M$_{HZ}$
		5V		0.455	-	20	
F_{LF}	Low frequency Crystal Oscillation range	3V	LF mode	-	32	-	K$_{HZ}$
		5V		-	32	-	
T_{WDT}	WDT period time	3V	Pre-scaler rate=1:1	-	20.5	-	mS
		4V		-	18.3	-	
		5V		-	16.4	-	

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

Under Operating Conditions, at two clock instruction cycles and WDT is disable, I/O output float.

Symbol	Description	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{sys}	Operating voltage	-	F_{SYS}=16M$_{HZ}$ (8MIPS)	3.3	-	5.5	V
		-	F_{SYS}=8M$_{HZ}$ (4MIPS)	2.0	-	5.5	
		-	F_{SYS}=500K$_{HZ}$ (0.25MIPS)	1.9	-	5.5	
		-	F_{SYS}=250K$_{HZ}$ (0.125MIPS)	1.9	-	5.5	
V_{IH}	Input high voltage, I/O Ports	3V	-	-	1.74	V_{DD}	V
		5V		-	2.45	V_{DD}	
	Input high voltage, RSTB Pin	3V		2.0	1.68	V_{DD}	
		5V		2.7	2.46	V_{DD}	
V_{IL}	Input low voltage, I/O Ports	3V	-	V_{SS}	1.07	-	V
		5V		V_{SS}	1.45	-	
	Input low voltage, RSTB Pin	3V		V_{SS}	0.84	0.6	
		5V		V_{SS}	1.26	1.0	
V_{LVDT}	LVDT Voltage	-	LVDT=3.0V	2.7	-	3.03	V
		-	LVDT=2.3V	2.07	-	2.33	
		-	LVDT=2.1V	1.89	-	2.12	
		-	LVDT=2.0V	1.8	-	2.02	
		-	LVDT=1.8V	1.62	-	1.82	

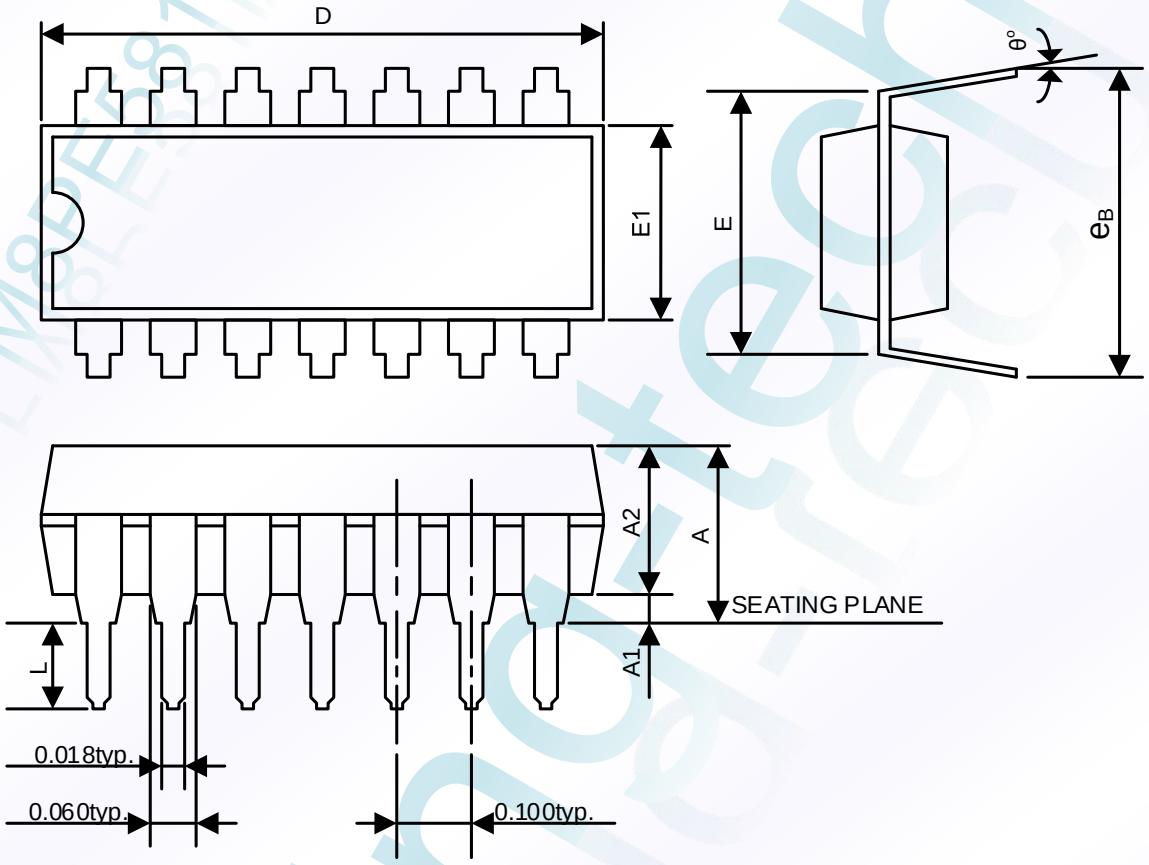
Symbol	Description	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
I_{OH}	IOB4, IOB5 Drive current	3V	V_{OH}=0.9V_{DD}	2.75	5.58	-	mA
		5V		-	13.94	-	
	IOA3, IOA4, IOA5, IOB0, IOB1, IOB2 Drive current	3V		3.5	7.4	-	
		5V		-	18.4	-	
	Other I/O Pins Drive current	3V		2	3.96	-	
		5V		-	10.21	-	
I_{OL}	IOB3, IOB4, IOB5 Drive current	3V	V_{OL}=0.1V_{DD}	7	14.21	-	mA
		5V		-	32.69	-	
	IOA3, IOA4, IOA5, IOB0, IOB1, IOB2 Drive current	3V		9	19.01	-	
		5V		-	42.84	-	
	Other I/O Pins Drive current	3V		5	10.03	-	
		5V		-	24.03	-	
I_{PH}	IOB4, IOB5 Pull-high current	3V	Input pin at V_{SS}	4	7.95	12	uA
		5V		13.5	27.36	40.5	
	Other I/O Pins Pull-high current	3V		8.5	17.24	25.5	
		5V		29	57.94	87	
I_{PL}	I/O Ports Pull-low current	3V	Input pin at V_{DD}	45	92.75	135	uA
		5V		150	297.21	450	
I_{LVDT}	LVDT current	5V	LVDT=3.0V	-	31.78	-	uA
		3V	LVDT=2.3V	-	24.22	-	
		5V		-	30.45	-	
		3V	LVDT=2.1V	-	22.04	-	
		5V		-	28.3	-	
		3V	LVDT=2.0V	-	3.2	-	
		5V		-	6.7	-	
		3V	LVDT=1.8V	-	24.85	-	
		5V		-	31.35	-	
I_{WDT}	WDT current	3V	Sleep mode, Pre-scaler rate=1:128	-	1	-	uA
		5V		-	5.53	-	
I_{SB}	Sleep mode (Power down) current	3V	LVDT=1.8V	-	24.85	-	uA
		5V		-	31.35	60	
I_{DD1}	Operating current	3V	HIRC16M, 8MIPS, LVDT=1.8V	-	1.4	-	mA
		5V		-	2.97	-	
I_{DD2}	Operating current	3V	HIRC8M, 4MIPS, LVDT=1.8V	-	0.8	-	mA
		5V		-	1.5	-	
I_{DD3}	Operating current	3V	LIRC500K, 0.25MIPS, LVDT=1.8V	-	75.7	-	uA
		5V		-	131.7	-	
I_{DD4}	Operating current	3V	LIRC250K, 0.125MIPS, LVDT=1.8V	-	55.4	-	uA
		5V		-	90.1	-	

6.3 ELECTRICAL CHARACTERISTICS Charts of FM8PE581M

To be define...

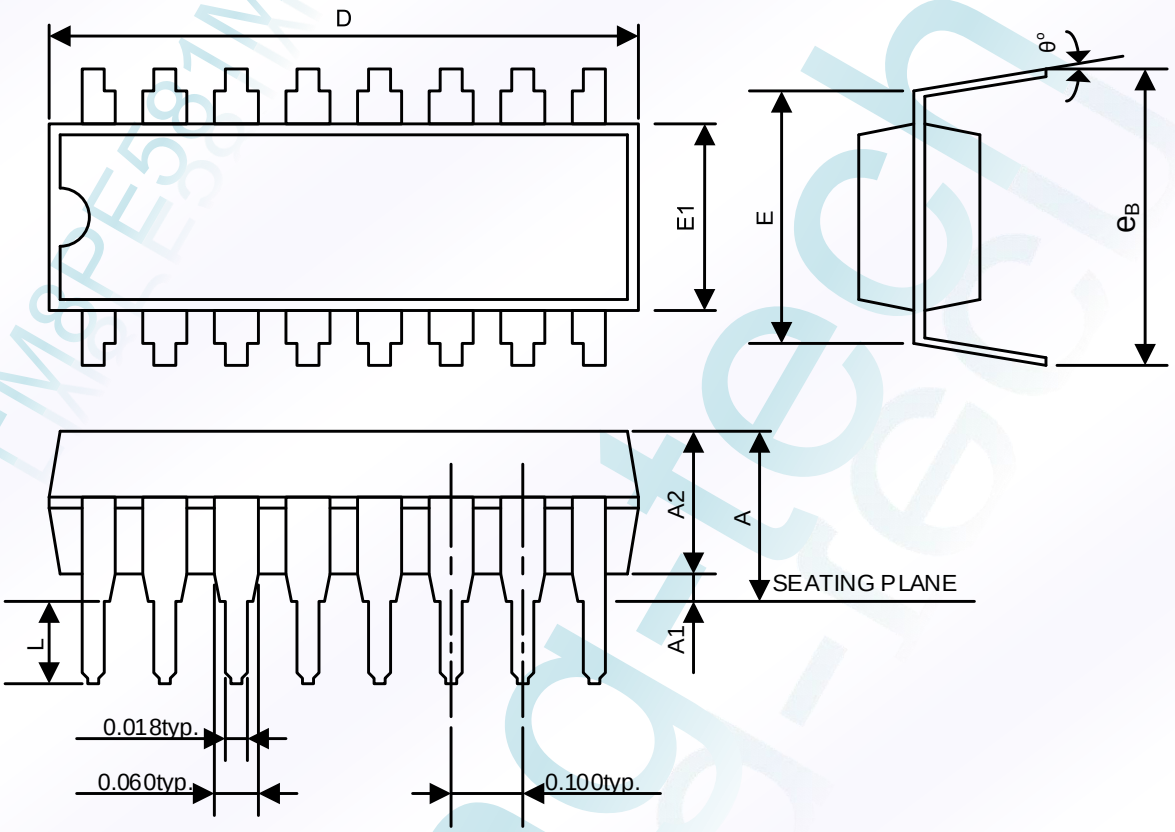
7.0 PACKAGE DIMENSION

7.1 14-PIN PDIP



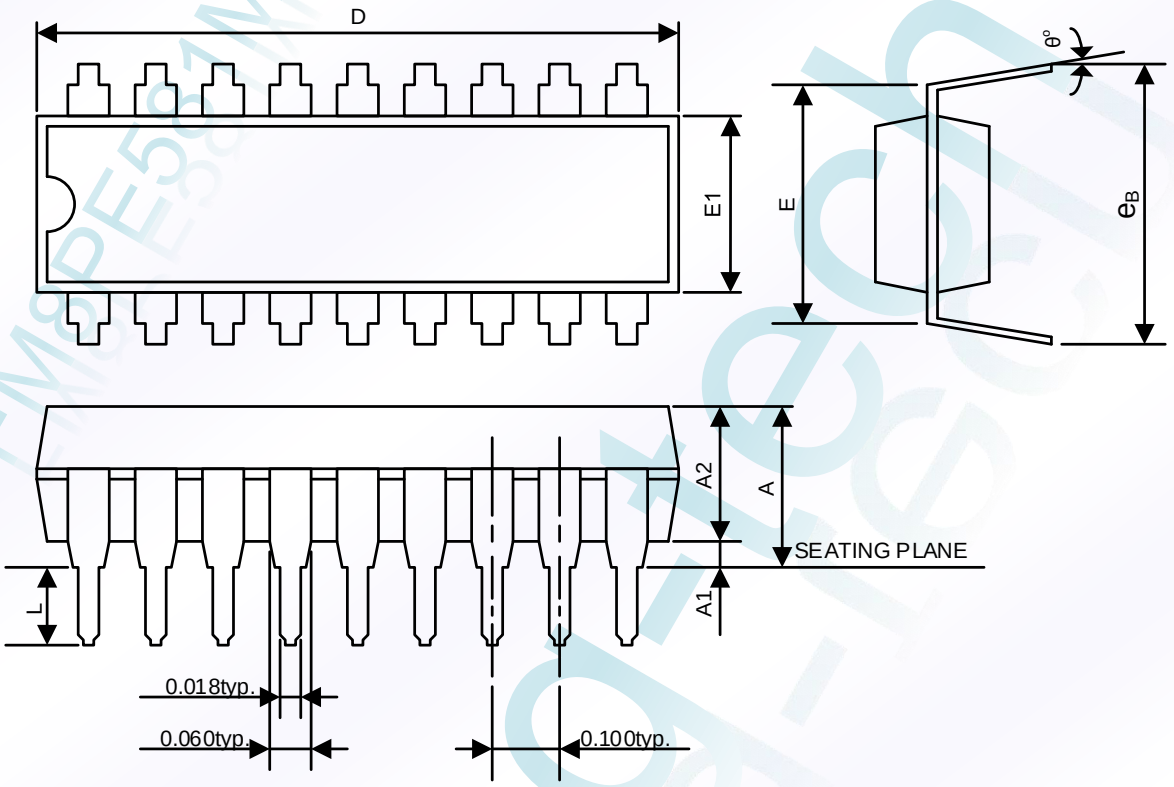
Symbols	Dimension In Inches		
	Min	Nom	Max
A	-	-	0.210
A1	0.015	-	-
A2	0.125	0.130	0.135
D	0.735	0.750	0.775
E		0.300 BSC.	
E1	0.245	0.250	0.255
L	0.115	0.130	0.150
eB	0.335	0.355	0.375
θ°	0°	7°	15°

7.2 16-PIN PDIP



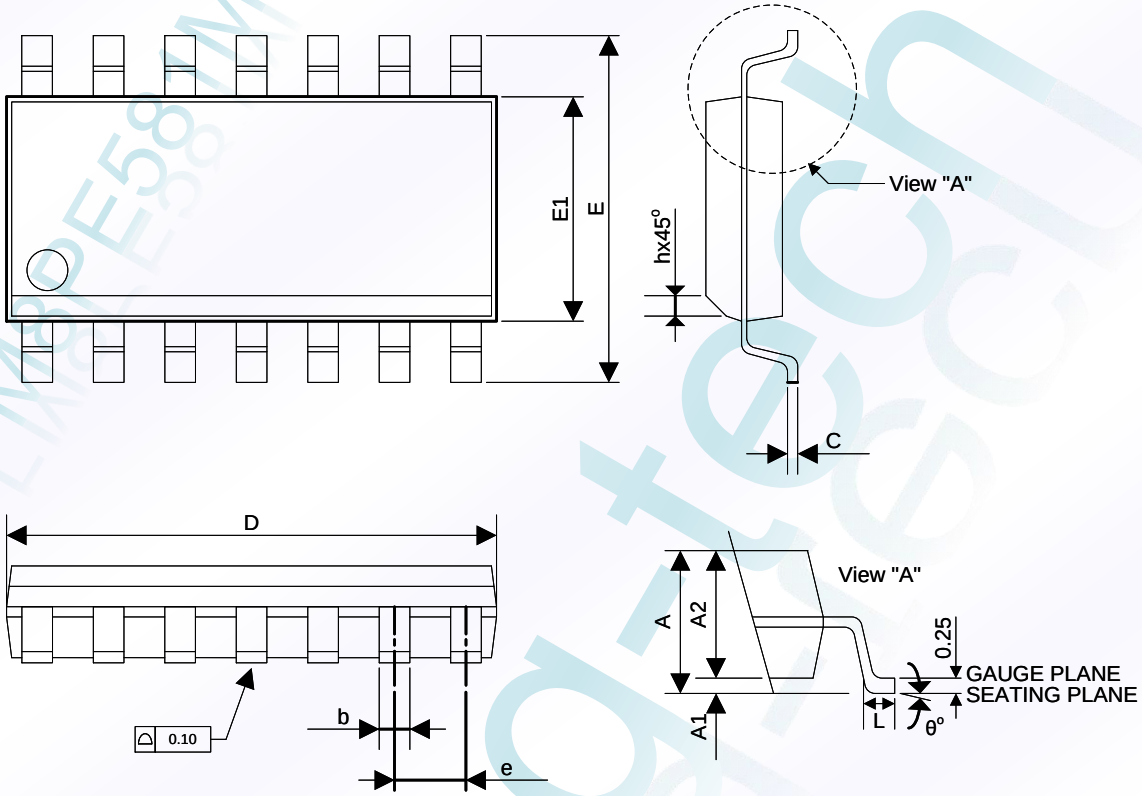
Symbols	Dimension In Inches		
	Min	Nom	Max
A	-	-	0.172
A1	0.015	-	0.038
A2	0.125	0.130	0.135
D	0.735	0.755	0.775
E		0.300 BSC.	
E1	0.245	0.250	0.255
L	0.115	0.130	0.150
eB	0.335	0.355	0.375
θ°	0°	7°	15°

7.3 18-PIN PDIP



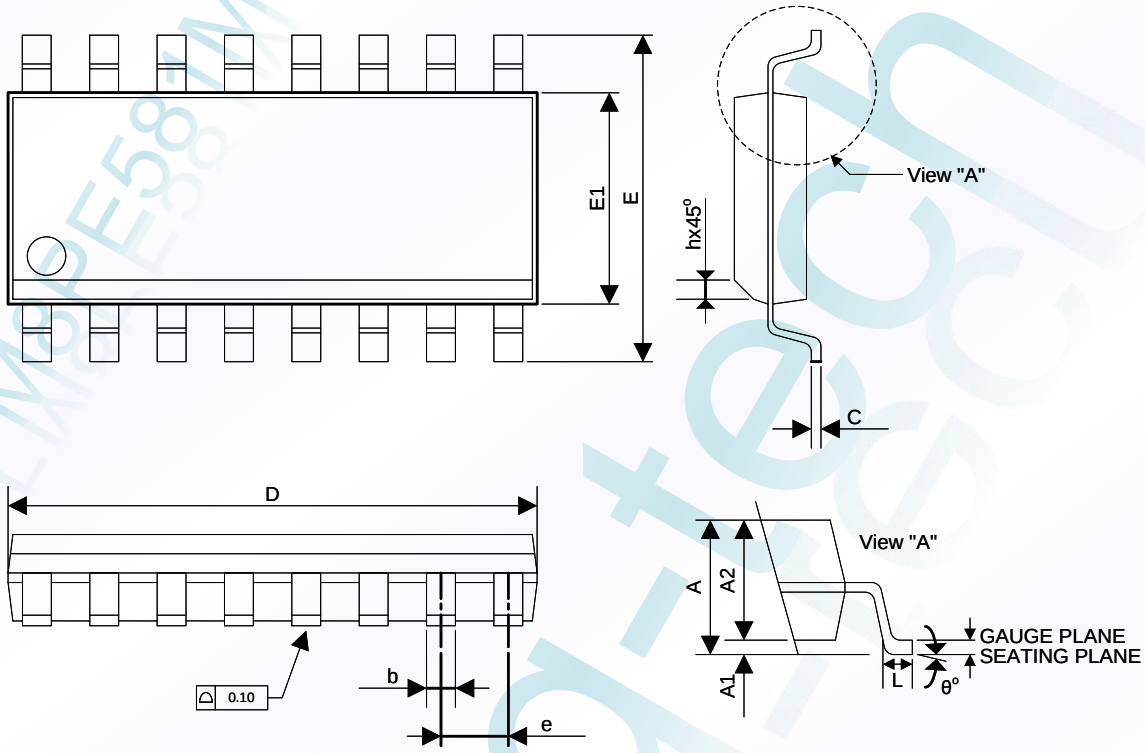
Symbols	Dimension In Inches		
	Min	Nom	Max
A	-	-	0.210
A1	0.015	-	-
A2	0.125	0.130	0.135
D	0.880	0.900	0.920
E		0.300 BSC.	
E1	0.245	0.250	0.255
L	0.115	0.130	0.150
eB	0.335	0.355	0.375
θ°	0°	7°	15°

7.4 14-PIN SOP 150mil



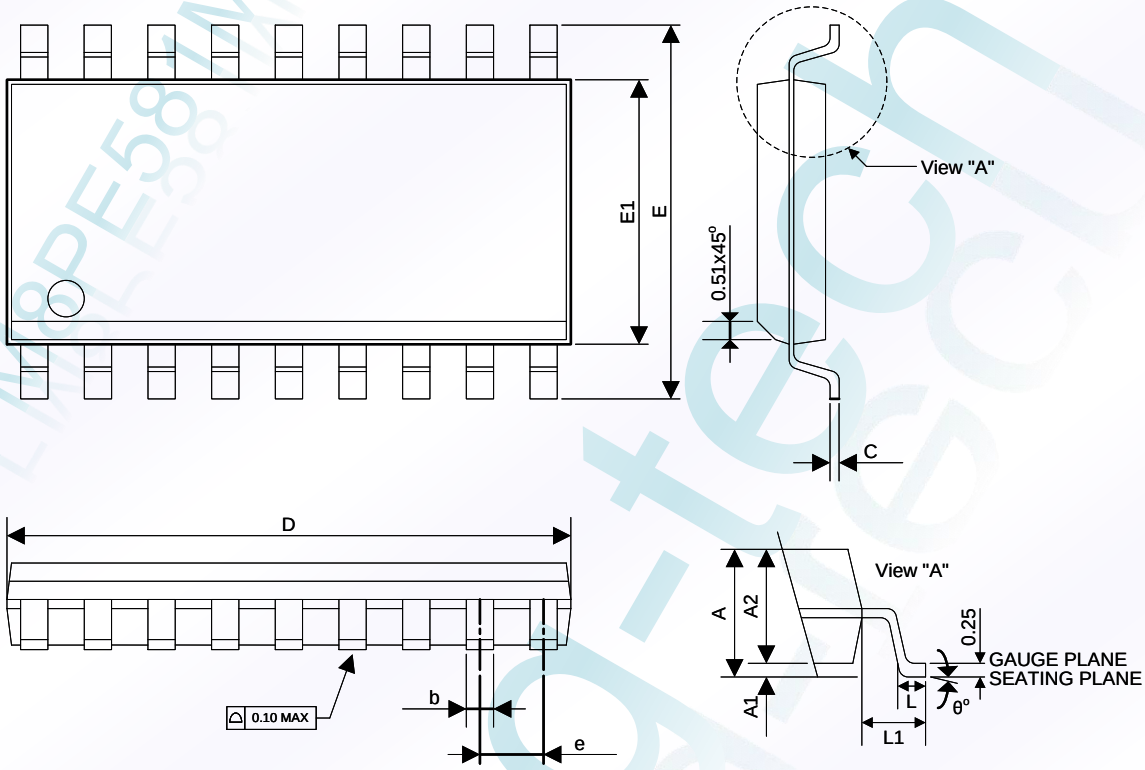
Symbols	Dimension In MM		
	Min	Nom	Max
A	-	-	1.75
A1	0.10	-	0.25
A2	1.25	-	-
b	0.31	-	0.51
C	0.10	-	0.25
D	8.65 BSC		
E	6.00 BSC		
E1	3.90 BSC		
e	1.27 BSC		
L	0.40	-	1.27
H	0.25		0.50
θ	0°	-	8°

7.5 16-PIN SOP 150mil



Symbols	Dimension In MM		
	Min	Nom	Max
A	-	-	1.75
A1	0.10	-	0.25
A2	1.25	-	-
b	0.31	-	0.51
c	0.10	-	0.25
D		9.90 BSC	
E		6.00 BSC	
E1		3.90 BSC	
e		1.27 BSC	
L	0.40	-	1.27
H	0.25	-	0.50
θ	0°	-	8°

7.6 18-PIN SOP 300mil



Symbols	Dimension In MM		
	Min	Nom	Max
A	-	-	2.65
A1	0.10	-	0.30
A2	2.05	-	-
b	0.31	-	0.51
c	0.20	-	0.33
D	11.55 BSC		
E	7.50 BSC		
E1	10.30 BSC		
e	1.27 BSC		
L1	1.40 REF		
L	0.40	-	1.27
θ	0°	-	8°

8.0 ORDERING INFORMATION

OTP Type MCU	Package Type	Pin Count	Package Size	MOQ	MSL	Sample Stock
FM8PE581MAP	PDIP	14	300mil	3,000EA/Tube	3	Call sales
FM8PE581MAD	SOP	14	150mil	3,000EA/Tube 3,000EA/Reel	3	Call sales
FM8PE581MBP	PDIP	16	300mil	3,000EA/Tube	3	Call sales
FM8PE581MBD	SOP	16	150mil	3,000EA/Tube 3,000EA/Reel	3	Call sales
FM8PE581MCP	PDIP	18	300mil	3,000EA/Tube	3	Available
FM8PE581MCD	SOP	18	300mil	3,000EA/Tube 1,000EA/Reel*3	3	Available

Please note: This product is only sold packaged, no longer sold wafers.