FEELING TECHNOLOGY

FM8PE51M

OTP-Based 8-Bit Microcontroller

Devices Included in this Data Sheet:

- FM8PE51M: OTP device

FEATURES

- Only 51 single word instructions.
- All instructions are single cycle except for program branches which are two-cycle.
- 8-bit wide data path.
- 5-level deep hardware stack.
- 4K-Word on chip OTP, and 140-Byte on chip general purpose registers (SRAM).
- Direct, indirect addressing modes for data accessing.
- One 8-bit real time clock/counter (Timer0) with 8-bit programmable pre-scaler.
- Three 8-bit real time clock/counter (Timer1, Timer2, and Timer3) with period setting.
- Internal Power-on Reset (POR).
- Built-in Low Voltage Detector (LVD) for Brown-out Reset (BOR).
- Power-up Reset Timer (PWRT) and Oscillator Start-up Timer(OST).
- On chip Watchdog Timer (WDT) with internal oscillator for reliable operation and soft-ware watch-dog enable/disable control.
- Five I/O ports (38 I/O pins) with independent direction control.
- 32 soft-ware control pull-high input pins.
- 12 wake-up pins.
- 2 open-drain pins.
- 2 R-option pins.
- Clock output with high driving ability.
- 19 channels of 15-bit resolution Resistor to Frequency Converter (RFC) output.
- 2 channels of maximum 10-bit resolution Pulse Width Modulation (PWM) output.
- Built-in 8-bit data comparator.
- Seven internal interrupt source: Timer0 overflow, Timer1 match, Timer2 match, Timer3 match, SPI module (Receive & Transmit), RFC module; One external interrupt source: INT pin change.
- Wake-up from SLEEP by port input change.
- Serial Peripheral Interface (SPI).
- Programmable Code Protection.
- Power saving SLEEP mode.
- Built-in RC oscillator with external resistor.
- Built-in 8M$_{HZ}$, 4M$_{HZ}$, 1M$_{HZ}$, and 455K$_{HZ}$ internal RC oscillator.
- Selectable oscillator options:
  - ERC: External Resistor/Capacitor Oscillator.
  - HF: High Frequency Crystal/Resonator Oscillator.
  - XT: Crystal/Resonator Oscillator.
  - LF: Low Frequency Crystal Oscillator.
  - IRC: Internal Resistor/Capacitor Oscillator.
  - ERIC: External Resistor/Internal Capacitor Oscillator.
- Wide-operating voltage range: 2.0V to 5.5V.

This datasheet contains new product information. Feeling Technology reserves the rights to modify the product specification without notice. No liability is assumed as a result of the use of this product. No rights under any patent accompany the sales of the product.

GENERAL DESCRIPTION

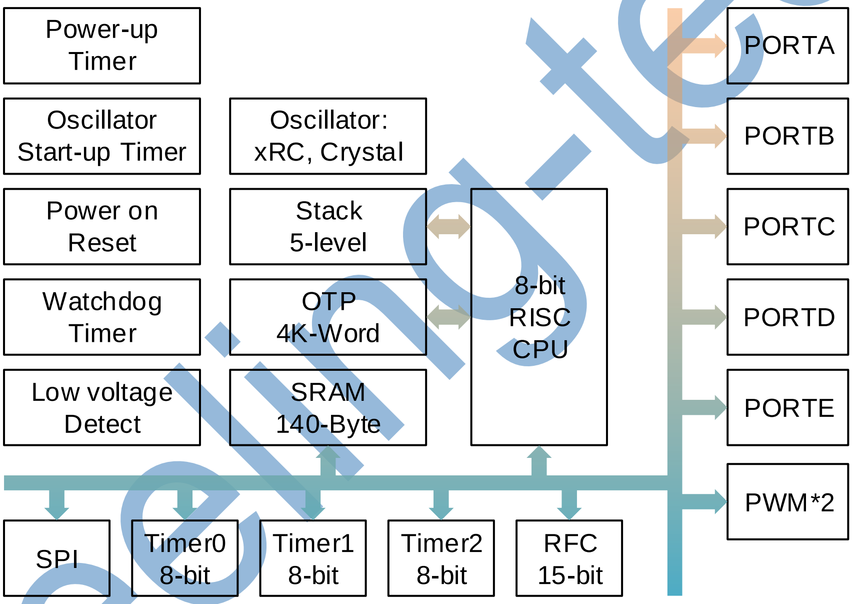
The FM8PE51M is a low-cost, high speed, high noise immunity, OTP-based 8-bit CMOS microcontrollers. It employs a RISC architecture with only 51 instructions. All instructions are single cycle except for program branches which take two cycles. The easy to use and easy to remember instruction set reduces development time significantly.

The FM8PE51M consists of Power-on Reset (POR), Brown-out Reset (BOR), Power-up Reset Timer (PWRT), Oscillator Start-up Timer(OST), Watchdog Timer, OTP, SRAM, tristate I/O port, I/O pull-high/open-drain control, Power saving SLEEP mode, 4 real time programmable clock/counter, Interrupt, SPI, RFC, data compare, Wake-up from SLEEP mode, and Code Protection for OTP products. There are six oscillator configurations to choose from, including the power-saving LF (Low Frequency) oscillator and cost saving RC oscillator.

The FM8PE51M address 4K-word of program memory.

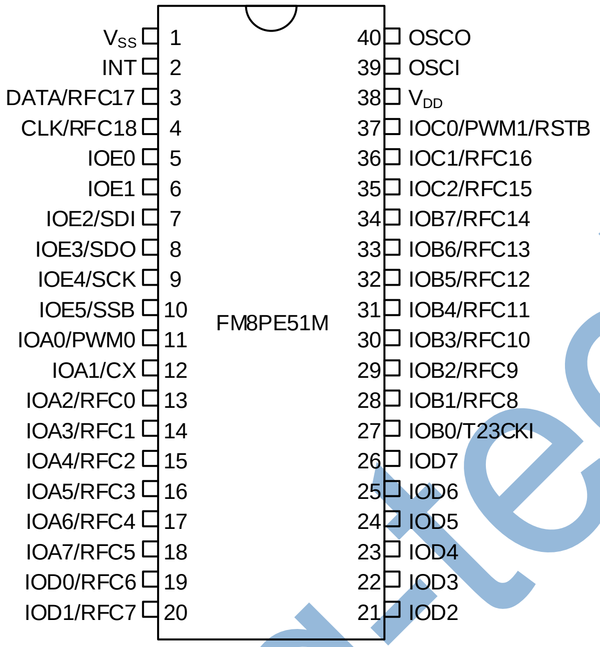
The FM8PE51M can directly or indirectly address its register files and data memory. All special function registers including the program counter are mapped in the data memory.

BLOCK DIAGRAM

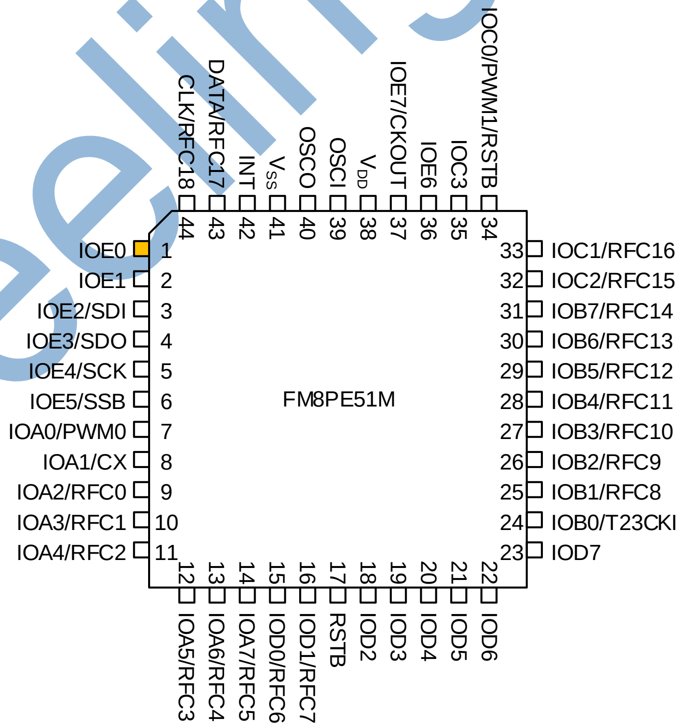


PIN CONNECTION

PDIP



QFP44



PIN DESCRIPTIONS

FM8PE51M

Name	I/O	Description
IOA0 ~ IOA7	I/O	• Bi-direction I/O port. • Software controlled pull-high for all pins.
IOB0 ~ IOB7	I/O	• Bi-direction I/O port with system wake-up function. • Software controlled pull-high for all pins.
IOC0 ~ IOC2	I/O	Bi-direction I/O pins with driving ability selection
IOC3	I/O	Bi-direction I/O pin
CLK	I/O	• By connecting IOC4 and IOC6 together. • IOC4 is a bi-direction I/O pin with system wake-up and software controlled pull-high function. • IOC6 is a bi-direction I/O pin with software controlled open-drain output. Note: Both IOC4 and IOC6 should not be defined as output pins at the same time.
DATA	I/O	• By connecting IOC5 and IOC7 together. • IOC4 is a bi-direction I/O pin with system wake-up and software controlled pull-high function. • IOC6 is a bi-direction I/O pin with software controlled open-drain output. Note: Both IOC4 and IOC6 should not be defined as output pins at the same time.
IOD0 ~ IOD7	I/O	• Bi-direction I/O port. • Software controlled pull-high for all pins. • IOD0 and IOD1 are R-option pins.
IOE0 ~ IOE1	I/O	• Bi-direction I/O pins with system wake-up function. • Software controlled pull-high for all pins.
IOE2 ~ IOE5	I/O	• Bi-direction I/O pins. • All pins can be pulled-high by software.
IOE6 ~ IOE7	I/O	Bi-direction I/O pins.
INT	I	• External interrupt input triggered by falling edge. • **Voltage on this pin must not exceed V_{DD},** See INT Pin diagram for detail description.
SDI	I	Serial data in for SPI.
SDO	O	Serial data out for SPI.
SCK	I/O	Serial clock for SPI.
SSB	I	Slave select (active low) for SPI.
PWM0 ~ PWM1	O	PWM output pins.
RFC0 ~ RFC18	O	The RC oscillator network output of RFC module.
CX	I	The RC oscillator network input of RFC module.
RSTB	I	System clear (RESET) input. This pin is an active low RESET to the device. Internal weak pull-high.
OSCI	I	• X'tal type: Oscillator crystal input. • RC type: Clock input of RC oscillator.
OSCO	O	• X'tal type: Oscillator crystal output. • RC mode: Outputs the instruction cycle rate.
CKOUT	O	Oscillator frequency output with high driving ability and output phase selection.
V_{DD}	-	Positive supply.
V_{SS}	-	Ground.

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

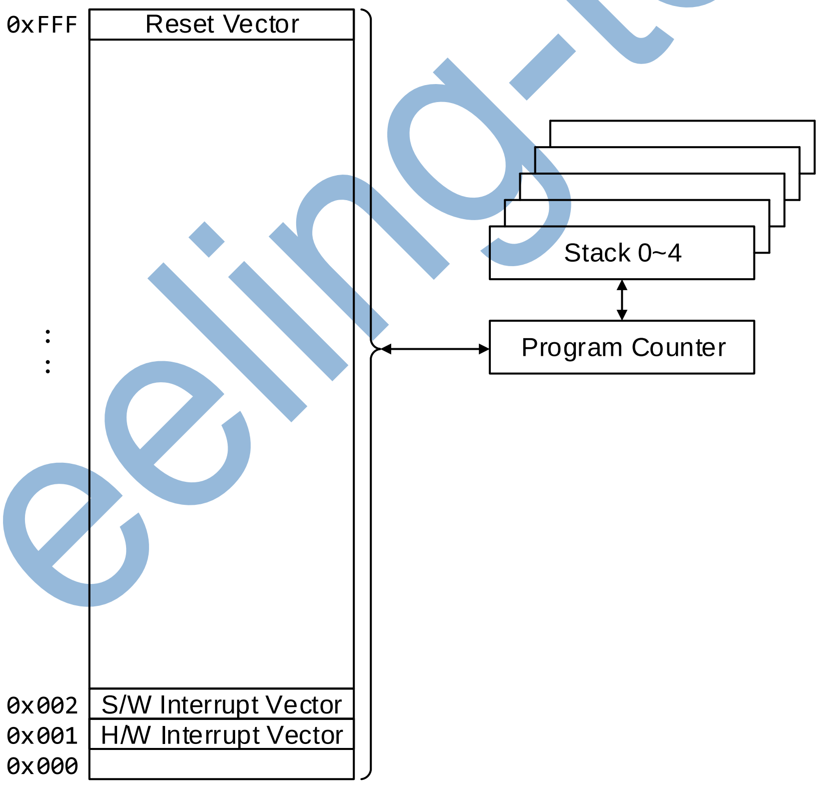
1.0 MEMORY ORGANIZATION

FM8PE51M memory is organized into program memory and data memory.

1.1 Program Memory Organization

The FM8PE51M has a 12-bit Program Counter capable of addressing a 4K program memory space.
The RESET vector for the FM8PE51M is at 0xFFF.
The H/W interrupt vector is at 0x001. And the S/W interrupt vector is at 0x002.
FM8PE51M has program memory size greater than 1K words, but the CALL and GOTO instructions only have a 10-bit address range. This 10-bit address range allows a branch within a 1K program memory page size. To allow CALL and GOTO instructions to address the entire 4K program memory address range for FM8PE51M, there is another two bits to specify the program memory page. This paging bit comes from the PG<1:0> bits (STATUS<6:5>). When doing a CALL or GOTO instruction, the user must ensure that page bit PG<1:0> are programmed so that the desired program memory page is addressed. When one of the return instructions is executed, the entire 12-bit PC is Popped from the stack. Therefore, manipulation of the PG <1:0> is not required for the return instructions. User can use "PAGE" instruction to change memory page and maintains the program memory page. Otherwise, user can use "FCALL (far call)/FGOTO (far goto)"instructions program user's code.

Figure 1.1: Program Memory Map and STACK



1.2 Data Memory Organization

Data memory is composed of Special Function Registers and General Purpose Registers.

The General Purpose Registers are accessed either directly or indirectly through the FSR register.

The Special Function Registers are registers used by the CPU and peripheral functions to control the operation of the device.

In FM8PE51M, the data memory is partitioned into four banks. Switching between these banks requires the RP1 and RP0 bits in the FSR register to be configured for the desired bank. User can use “BANK” instruction to change the data memory bank.

Table 1.1: Registers File Map for FM8PE51M

FSR<7:6> Address	Description			
	0 0 Bank 0	0 1 Bank 1	1 0 Bank 2	1 1 Bank 3
0x000	INDF	Memory back to address in Bank 0		
0x001	TMR0			
0x002	PCL			
0x003	STATUS			
0x004	FSR			
0x005	PORTA			
0x006	PORTB			
0x007	PORTC			
0x008	PORTD			
0x009	PORTE			
0x00A	SPIRXB	T23CON	PWMCON	RFCCON
0x00B	SPITXB	TMR2	PW0DCL	RFCDLB
0x00C	SPISTAT	PR2	PW0DCH	RFCDHB
0x00D	SPICON	TMR3	PW1DCL	CMPDX
0x00E	TMR1	PR3	PW1DCH	CMPDY
0x00F	PR1	-	-	CMPSTAT
0x010 | 0x01F	General Purpose Registers	Memory back to address in Bank 0		
0x020 | 0x03E	General Purpose Registers	General Purpose Registers	General Purpose Registers	General Purpose Registers
0x03F	INTFLAG	Memory back to address in Bank 0		

N/A	OPTION
0x005	IOSTA
0x006	IOSTB
0x007	IOSTC
0x008	IOSTD
0x009	IOSTE
0x00C	T1CON
0x00D	PHCON
0x00E	PCON
0x00F	INTEN

Table 1.2: The Registers Controlled by OPTION / OPTIONR or IOST / IOSTR Instructions

Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
N/A (r/w)	OPTION	/PHBCE	GIE	1	1	PSA	PS2	PS1	PS0
0x005 (r/w)	IOSTA	PORTA I/O Control Register							
0x006 (r/w)	IOSTB	PORTB I/O Control Register							
0x007 (r/w)	IOSTC	PORTC I/O Control Register							
0x008 (r/w)	IOSTD	PORTD I/O Control Register							
0x009 (r/w)	IOSTE	PORTE I/O Control Register							
0x00C (r/w)	T1CON	GP	GP	GP	GP	GP	T1ON	T1P1	T1P0
0x00D (r/w)	PHCON	HDC	1	1	1	/PHE	/PHD	/PHB	/PHA
0x00E (r/w)	PCON	LVDTE	ODE	WDTE	*	ROC	-	-	/WUE
0x00F (r/w)	INTEN	SPITXIE	RFCIE	T3IE	T2IE	T1IE	SPIRXIE	INTIE	T0IE

Legend: - = unimplemented, read as ‘0’, * = unimplemented, read as ‘1’, 1 = Not used, must fix to ‘1’.

Table 1.3: Operational Registers Map

Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
Unbanked									
0x000 (r/w)	INDF	Uses contents of FSR to address data memory (not a physical register)							
0x001 (r/w)	TMR0	8-bit real-time clock/counter							
0x002 (r/w)	PCL	Low order 8 bits of PC							
0x003 (r/w)	STATUS	GP	PG1	PG0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C
0x004 (r/w)	FSR	RP1	RP0	Indirect data memory address pointer					
0x005 (r/w)	PORTA	IOA7	IOA6	IOA5	IOA4	IOA3	IOA2	IOA1	IOA0
0x006 (r/w)	PORTB	IOB7	IOB6	IOB5	IOB4	IOB3	IOB2	IOB1	IOB0
0x007 (r/w)	PORTC	IOC7	IOC6	IOC5	IOC4	IOC3	IOC2	IOC1	IOC0
0x008 (r/w)	PORTD	IOD7	IOD6	IOD5	IOD4	IOD3	IOD2	IOD1	IOD0
0x009 (r/w)	PORTE	IOE7	IOE6	IOE5	IOE4	IOE3	IOE2	IOE1	IOE0
Bank 0									
0x00A (r)	SPIRXB	SPI Receive buffer							
0x00B (r/w)	SPITXB	SPI Transmitter buffer							
0x00C (r/w)	SPISTAT	GP	TXBFIF	TXBF	TM1IF	SDOOD	SCKOD	RXBFIF	RXBF
0x00D (r/w)	SPICON	CKEDG	SPION	RXOV	SSE	SSEMOD	SPIM2	SPIM1	SPIM0
0x00E (r/w)	TMR1	8-bit real-time clock/counter							
0x00F (r/w)	PR1	Timer1 period register							
Bank 1									
0x00A (r/w)	T23CON	-	-	-	T16	T3ON	T3CS	T2ON	T2CS
0x00B (r/w)	TMR2	8-bit real-time clock/counter							
0x00C (r/w)	PR2	Timer2 period register							
0x00D (r/w)	TMR3	8-bit real-time clock/counter							
0x00E (r/w)	PR3	Timer3 period register							
0x00F (r)	-	Unimplemented, read as “0”s							
Bank 2									
0x00A (r/w)	PWMCON	-	-	-	-	-	PW1T3	PW1ON	PW0ON
0x00B (r/w)	PW0DCL	DC1	DC0	-	-	-	-	-	-
0x00C (r/w)	PW0DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2
0x00D (r/w)	PW1DCL	DC1	DC0	-	-	-	-	-	-
0x00E (r/w)	PW1DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2
0x00F (r)	-	Unimplemented, read as “0”s							
Bank 3									
0x00A (r/w)	RFCCON	RFCON	START	-	RFCS4	RFCS3	RFCS2	RFCS1	RFCS0
0x00B (r)	RFCDLB	Low byte of 15 bit RFC conversion result							
0x00C (r)	RFCDHB	RFCOV	RFCD14	RFCD13	RFCD12	RFCD11	RFCD10	RFCD9	RFCD8
0x00D (r/w)	CMPDX	DX7	DX6	DX5	DX4	DX3	DX2	DX1	DX0
0x00E (r/w)	CMPDY	DY7	DY6	DY5	DY4	DY3	DY2	DY1	DY0
0x00F (r)	CMPSTAT	-	-	-	-	CMPF3	CMPF2	CMPF1	CMPF0
Unbanked									
0x03F (r/w)	INTFLAG	SPITXIF	RFCIF	T3IF	T2IF	T1IF	SPIRXIF	INTIF	T0IF

Legend: - = unimplemented, read as ‘0’, * = unimplemented, read as ‘1’.

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 FSR to address data memory (not a physical register)							

Legend: x = unknown, more bits' default state, please refer to Table 2.6.

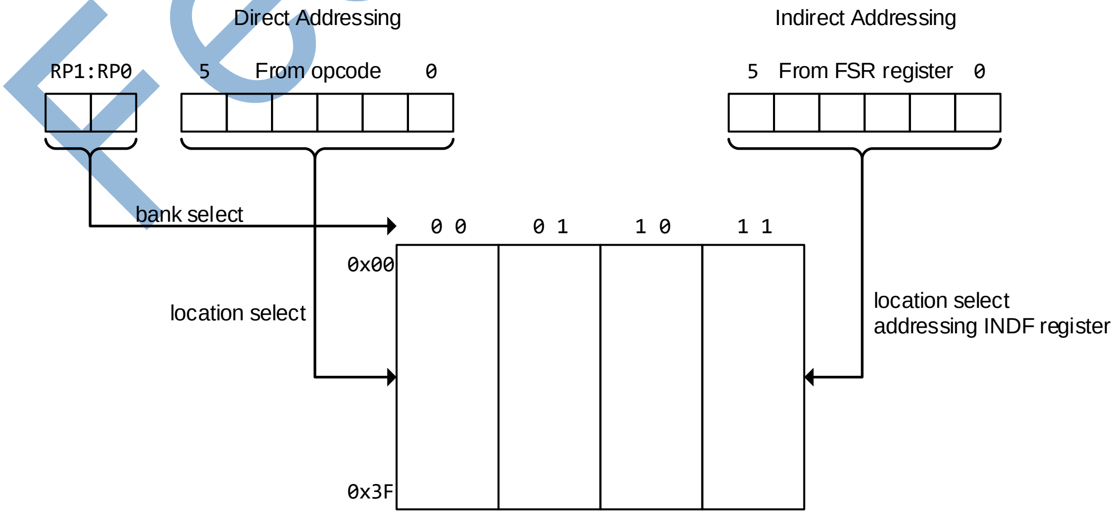
The INDF Register is not a physical register. Any instruction accessing the INDF register can actually access the register pointed by FSR Register. Reading the INDF register itself indirectly (FSR="0x00") will read 0x00. Writing to the INDF register indirectly results in a no-operation (although status bits may be affected).
The bits 5-0 of FSR register are used to select up to 64 registers (address: 0x00 ~ 0x3F).
In FM8PE51M, the data memory is partitioned into four banks. Switching between these banks requires the RP1 and RP0 bits in the FSR register to be configured for the desired bank. The lower locations of each bank are reserved for the Special Function Registers. Above the Special Function Registers are General Purpose Registers. All Special Function Registers and some of General Purpose Registers from other banks are mirrored in bank 0 for code reduction and quicker access.

RP1:RP0		Accessed Bank
0	0	0
0	1	1
1	0	2
1	1	3

Example 2.1: INDIRECT ADDRESSING

- Register file 0x38 contains the value 0x10
- Register file 0x39 contains the value 0x0A
- Load the value 0x38 into the FSR Register
- A read of the INDF Register will return the value of 0x10
- Increment the value of the FSR Register by one (@FSR=0x39)
- A read of the INDF register now will return the value of 0x0A.

Figure 2.1: Direct/Indirect Addressing for FM8PE51M



2.1.2 TMR0 (Time Clock/Counter 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
0x001	TMR0	8-bit real-time clock/counter							

Legend: x = unknown, more bits' default state, please refer to Table 2.6.

The Timer0 is an 8-bit timer/counter. The clock source of Timer0 can come from the instruction cycle clock.
The pre-scaler is assigned to Timer0 by clearing the PSA bit (OPTION<3>). In this case, the pre-scaler will be cleared when TMR0 register is written with a value.

2.1.3 PCL (Low Bytes of Program Counter) & Stack

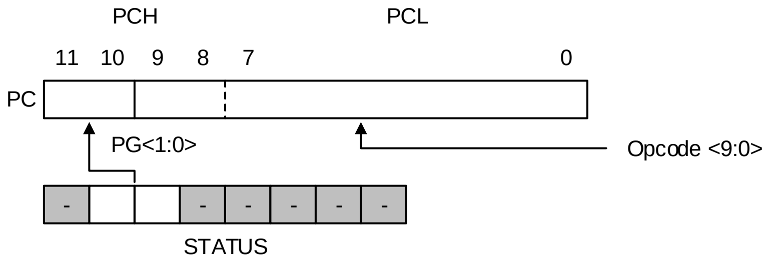
Read/Write-POR		R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x002	PCL	Low order 8 bits of PC							

Legend: x = unknown, more bits' default state, please refer to Table 2.6.

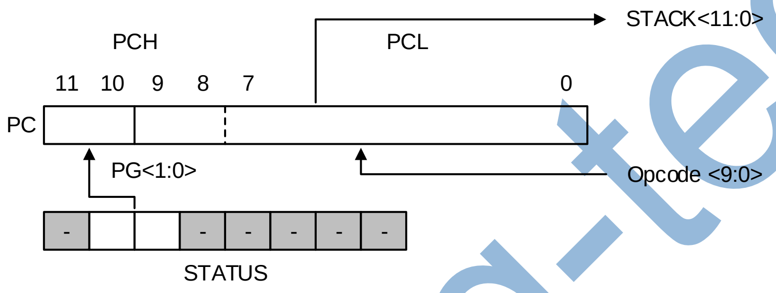
FM8PE51M devices have a 12-bit wide Program Counter (PC) and five-level deep 12-bit hardware push/pop stack. The low byte of PC is called the PCL register. This register is readable and writable. The high byte of PC is called the PCH register. This register contains the PC<11:8> bits and is not directly readable or writable. All updates to the PCH register go through the PG<1:0> bits (STATUS<6:5>). 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<9:0> is provided by the GOTO instruction word. The PC<11:10> is updated from the PG<1:0> bits (STATUS<6:5>). The PCL register is mapped to PC<7:0>.
For a CALL instruction, the PC<9:0> is provided by the CALL instruction word. The PC<11:10> is updated from the PG<1:0> bits (STATUS<6:5>). The next PC will be loaded (Pushed) onto the top of STACK. The PCL register is mapped to PC<7:0>.
For a FGOTO instruction, the PC<11:0> is provided by the FGOTO instruction word. The PCL register is mapped to PC<7:0>, the PG<1:0> bits (STATUS<6:5>) is also updated from the FGOTO instruction word, and the PG<1:0> bits are not updated.
For a FCALL instruction, the PC<11:0> is provided by the FCALL instruction word. The next PC will be loaded (PUSHed) onto the top of STACK. The PCL register is mapped to PC<7:0>, the PG<1:0> bits (STATUS<6:5>) is also updated from the FCALL instruction word, and the PG<1:0> bits are not updated.
For a RETIA, RETFIE, or RETURN instruction, the PC are updated (Popped) from the top of STACK. The PCL register is mapped to PC<7:0>.
For any instruction where the PCL is the destination (excluding TBL instruction), the PC<7:0> is provided by the instruction word or ALU result, and the PC<9:8> will be cleared. The PC<11:10> will come from the PG<1:0> bits (STATUS<6:5>).
For TBL instruction, the PC<7:0> is provided by the ALU result, and the PC<9:8> are not changed. The PC<11:10> will come from the PG<1:0> bits (STATUS<6:5>).

Figure 2.2: Loading of PC in Different Situations

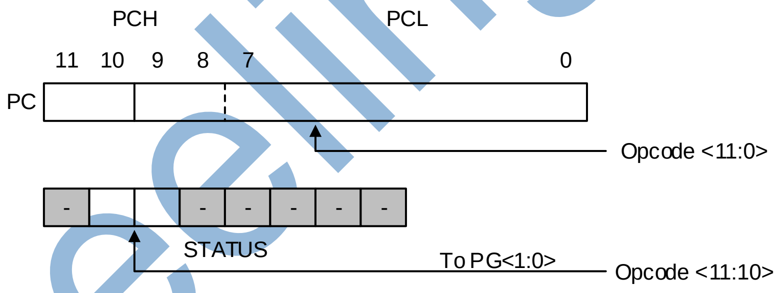
Situation 1: **GOTO** Instruction



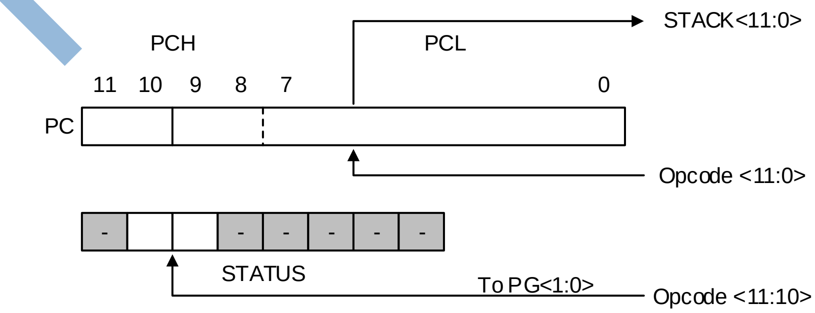
Situation 2: **CALL** Instruction



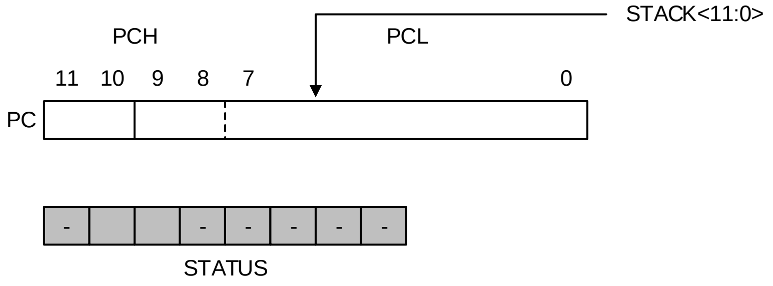
Situation 3: **FGOTO** Instruction



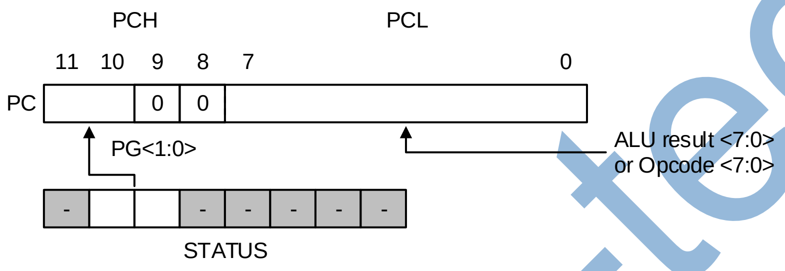
Situation 4: **FCALL** Instruction



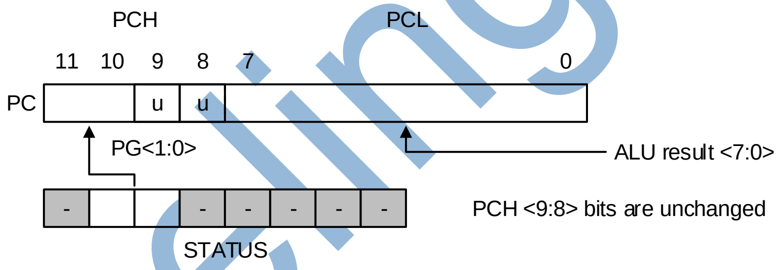
Situation 5: **RETIA**, **RETFIE**, or **RETURN** Instruction



Situation 6: Instruction with PCL as destination (except TBL instruction)



Situation 7: **TBL** instruction



2.1.4 STATUS (Status Register)

Read/Write-POR		R/W-0	R/W-0	R/W-0	R-#	R-#	R/W-x	R/W-x	R/W-x
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x003	STATUS	GP	PG1	PG0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C

Note: # = refer Table 2.7 for detail description, more bits' default state, please refer to Table 2.6.

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 $\overline{\text{TO}}$ and $\overline{\text{PD}}$ bits are not writable. Therefore, the result of an instruction with the STATUS Register as destination may be different than intended. For example, CLRR STATUS will clear the upper three bits and set the Z bit. This leaves the STATUS Register as 000u u1uu (where u = unchanged).

C: Carry/borrow bit.
ADDAR, ADDIA, ADCAR:
= 0, No Carry occurred.
= 1, Carry occurred.
SUBAR, SUBIA, SBCAR:
= 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, ADDIA, ADCAR:
= 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, SUBIA, SBCAR:
= 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.

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

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

PG1:PG0: Program memory page select bits. Used for GOTO, CALL, or any instruction with PCL as destination.

PG1:PG0		Program Memory Page [Address]
0	0	Page 0 [0x000~0x3FF]
0	1	Page 1 [0x400~0x7FF]
1	0	Page 2 [0x800~0xBFF]
1	1	Page 3 [0xC00~0xFFF]

GP: General purpose read/write bits.

2.1.5 FSR (Indirect Data Memory Address Pointer)

Read/Write-POR		R/W-0	R/W-0	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	FSR	RP1	RP0	Indirect data memory address pointer					

Legend: x = unknown, more bits' default state, please refer to Table 2.6.

Bit5:Bit0: Select registers address in the indirect addressing mode. See 2.1.1 for detail description.

RP1:RP0: These bits are used to switching the bank of four data memory banks. See 2.1.1 for detail description.

2.1.6 PORTA, PORTB, PORTC, PORTD, PORTE (Port Data Registers)

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
0x005	PORTA	IOA7	IOA6	IOA5	IOA4	IOA3	IOA2	IOA1	IOA0

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
0x006	PORTB	IOB7	IOB6	IOB5	IOB4	IOB3	IOB2	IOB1	IOB0

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
0x007	PORTC	IOC7	IOC6	IOC5	IOC4	IOC3	IOC2	IOC1	IOC0

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	PORTD	IOD7	IOD6	IOD5	IOD4	IOD3	IOD2	IOD1	IOD0

Read/Write-POR		R/W-%	R/W-%	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
0x009	PORTE	IOE7	IOE6	IOE5	IOE4	IOE3	IOE2	IOE1	IOE0

Legend: x = unknown, % = define by TYPE of the configuration word, detail and more bits' default state, please refer to Table 2.6.

Reading the port (PORTA, PORTB, PORTC, PORTD and PORTE 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.

2.1.7 SPIRXB (SPI Receive Buffer Register) (Bank 0)

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
0x0A	SPIRXB	SPI Receive buffer							

Legend: x = unknown, more bits' default state, please refer to Table 2.6.

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.

The RXBF flag is set when the data in SPISR is moved to the SPIRXB register, and cleared as the SPIRXB register reads.

2.1.8 SPITXB (SPI Transmit Buffer Register) (Bank 0)

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
0x0B	SPITXB	SPI Transmitter buffer							

Legend: x = unknown, more bits' default state, please refer to Table 2.6.

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.

The TXBF flag is set when the data in SPITXB is moved to the SPISR register, and cleared as the SPITXB register writes.

2.1.9 SPISTAT (SPI Status Register) (Bank 0)

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	SPISTAT	GP	TXBFIF	TXBF	TM1IF	SDOOD	SCKOD	RXBFIF	RXBF

Note: more bits' default state, please refer to Table 2.6.

RXBF: SPI receive buffer full flag. Set when the data in SPISR is moved to the SPIRXB register, reset by software or by reading SPIRXB register.
= 0, Receive not complete, SPIRXB is empty.
= 1, Receive complete, SPIRXB is full.

RXBFIF: SPI receive buffer full interrupt flag. Set when the data in SPISR is moved to the SPIRXB register, reset by software.
= 0, Receive not complete, SPIRXB is empty.
= 1, Receive complete, SPIRXB is full.

SCKOD: Open-drain control bit for SCK pin output.
= 0, Open-drain disable.
= 1, Open-drain enable.

SDOOD: Open-drain control bit for SDO pin output.
= 0, Open-drain disable.
= 1, Open-drain enable.

TM1IF: SPI receive complete interrupt flag in Timer 1 mode. Set when receiving complete, reset by software.
= 0, In Timer 1 mode, receiving not complete yet.
= 1, In Timer 1 mode, receiving complete.

TXBF: SPI transmit buffer empty flag. Set when the data in SPITXB is moved to the SPISR register, reset by software or by writing SPITXB register.
= 0, SPITXB is full.
= 1, Transmit start, SPITXB is empty.

TXBFIF: SPI transmit buffer empty interrupt flag. Set when the data in SPITXB is moved to the SPISR register, reset by software.
= 0, SPITXB is full.
= 1, Transmit start, SPITXB is empty.

GP: General purpose read/write bits.

2.1.10 SPICON (SPI Control Register) (Bank 0)

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
0x00D	SPICON	CKEDG	SPION	RXOV	SSE	SSEMOD	SPIM2	SPIM1	SPIM0

Note: more bits' default state, please refer to Table 2.6.

SPIM2:SPIM0: SPI mode select bits.

SPIM2:SPIM0			SPI Mode
0	0	0	SPI master mode, clock = $F_{OSC}/2$
0	0	1	SPI master mode, clock = $F_{OSC}/4$
0	1	0	SPI master mode, clock = $F_{OSC}/8$
0	1	1	SPI master mode, clock = $F_{OSC}/16$
1	0	0	SPI master mode, clock = $F_{OSC}/32$
1	0	1	SPI slave mode, clock = SCK pin, SSB pin control enabled
1	1	0	SPI slave mode, clock = SCK pin, SSB pin control disabled
1	1	1	SPI master mode, clock = Timer1 output/2

SSEMOD: SSE bit control enable bit.
= 0, Enable the SSE bit control. It means the SCK input/output will be inhibited if SSE = 0.
= 1, Disable the SSE bit control. It means the SCK input/output directly.

SSE: SPI shift register enable bit.
= 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.
Note: This bit is a "don't care" if SSEMOD = 1.

RXOV: SPI receive buffer overflow bit (only in slave mode).
=0, Not overflow.
=1, A new byte is received while the SPIRXB register is still holding the previous data. In this case, the data in SPISR register will be ignored and lost.

SPION: SPI module enable bit
= 0, Disable SPI module.
= 1, Enable SPI module.

CKEDG: Clock edge select bit
= 0, Data shifts out on rising edge of SCK, and shifts in on falling edge of SCK.
= 1, Data shifts out on falling edge of SCK, and shifts in on rising edge of SCK.

2.1.11 TMR1 (Timer 1 Register) (Bank 0)

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
0x00E	TMR1	8-bit real-time clock/counter							

Note: more bits' default state, please refer to Table 2.6.

Bit7:Bit0: Timer1 register and increase until the value matches to PR1 register, and then reset to "0".

2.1.12 PR1 (Timer 1 Pulse-width Register) (Bank 0)

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
0x00F	PR1	Timer1 period register							

Note: more bits' default state, please refer to Table 2.6.

Bit7:Bit0: Timer1 period register.

2.1.13 T23CON (Timer 2&3 Control Register) (Bank 1)

Read/Write-POR		-	-	-	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x00A	T23CON	-	-	-	T16	T3ON	T3CS	T2ON	T2CS

Legend: - = unimplemented, read as '0'; more bits' default state, please refer to Table 2.6.

T2CS: Timer 2 clock source selection bit
= 0, Internal clock F_{OSC}/4.
= 1, External clock input T23CKI pin.

T2ON: Timer 2 module enable bit
= 0, Disable Timer 2 module.
= 1, Enable Timer 2 module.

T3CS: Timer 3 clock source selection bit
= 0, Internal clock F_{OSC}/4.
= 1, External clock input T23CKI pin.

T3ON: Timer 3 module enable bit
= 0, Disable Timer 3 module.
= 1, Enable Timer 3 module.

T16: 8-bit or 16-bit timer selection bit for Timer2 & Timer3
= 0, Timer2 & Timer3 are two 8-bit incrementing timers.
= 1, Timer2 & Timer3 are concatenated to form a 16-bit timer.

2.1.14 TMR2 (Timer 2 Register) (Bank 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
0x00B	TMR2	8-bit real-time clock/counter							

Note: more bits' default state, please refer to Table 2.6.

Bit7:Bit0: Timer2 register and increase until the value matches to PR2 register, and then reset to "0".

2.1.15 PR2 (Timer 2 Pulse-width Register) (Bank 1)

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
0x00C	PR2	Timer2 period register							

Note: more bits' default state, please refer to Table 2.6.

Bit7:Bit0: Timer2 period register.

2.1.16 TMR3 (Timer 3 Register) (Bank 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
0x00D	TMR3	8-bit real-time clock/counter							

Note: more bits' default state, please refer to Table 2.6.

Bit7:Bit0: Timer 3 register and increase until the value matches to PR3 register, and then reset to "0".

2.1.17 PR3 (Timer 3 Pulse-width Register) (Bank 1)

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
0x00E	PR3	Timer3 period register							

Note: more bits' default state, please refer to Table 2.6.

Bit7:Bit0: Timer3 period register.

2.1.18 PWMCON (PWM Control Register) (Bank 2)

Read/Write-POR		-	-	-	-	-	R/W-0	R/W-0	R/W-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x00A	PWMCON	-	-	-	-	-	PW1T3	PW1ON	PW0ON

Legend: - = unimplemented, read as '0'; more bits' default state, please refer to Table 2.6.

PW0ON: PWM0 module enable bit
= 0, Disable PWM0 output, IOA0/PWM0 pin is configured to IOA0 pin.
= 1, Enable PWM0 output, IOA0/PWM0 pin is configured to PWM0 pin.

PW1ON: PWM1 module enable bit
= 0, Disable PWM1 output, IOC0/PWM1 pin is configured to IOC0 pin.
= 1, Enable PWM1 output, IOC0/PWM1 pin is configured to PWM1 pin.

PW1T3: PWM1 time base selection bit
= 0, the PWM1 time base is determined by TMR2 and PR2.
= 1, the PWM1 time base is determined by TMR3 and PR3.

2.1.19 PW0DCL & PW0DCH (PWM0 Duty Cycle Register Low Byte) (Bank 2)

Read/Write-POR		R/W-0	R/W-0	-	-	-	-	-	-
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x00B	PW0DCL	DC1	DC0	-	-	-	-	-	-

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	PW0DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2

Legend: - = unimplemented, read as '0'; more bits' default state, please refer to Table 2.6.

DC9:DC0: PWM0 duty cycle.

2.1.20 PW1DCL & PW1DCH (PWM1 Duty Cycle Register Byte) (Bank 2)

Read/Write-POR		R/W-0	R/W-0	-	-	-	-	-	-
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x00D	PW1DCL	DC1	DC0	-	-	-	-	-	-

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
0x00E	PW0DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2

Legend: - = unimplemented, read as '0'; more bits' default state, please refer to Table 2.6.

DC9:DC0: PWM1 duty cycle.

2.1.21 RFCCON (RFC Control Register) (Bank 3)

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
0x00A	RFCCON	RFCON	START	-	RFCS4	RFCS3	RFCS2	RFCS1	RFCS0

Legend: - = unimplemented, read as '0'; more bits' default state, please refer to Table 2.6.

RFCS4:RFCS0: Select one the RFC oscillation network of RFCx (x = 0 to 18). The selected IOxx/RFCx pin will be configured as RFCx output pin if RFCON = 1. Other IOxx/RFCx pins will still behave as port pins. If RFCON = 0, all IOxx/RFCx pins will still behave as port pins.

RFCS4:RFCS0					RFC channel
0	0	0	0	0	IOA2/RFC0 pin is configured to RFC0 pin.
0	0	0	0	1	IOA3/RFC1 pin is configured to RFC1 pin.
0	0	0	1	0	IOA4/RFC2 pin is configured to RFC2 pin.
0	0	0	1	1	IOA5/RFC3 pin is configured to RFC3 pin.
0	0	1	0	0	IOA6/RFC4 pin is configured to RFC4 pin.
0	0	1	0	1	IOA7/RFC5 pin is configured to RFC5 pin.
0	0	1	1	0	IOD0/RFC6 pin is configured to RFC6 pin.
0	0	1	1	1	IOD1/RFC7 pin is configured to RFC7 pin.
0	1	0	0	0	IOB1/RFC8 pin is configured to RFC8 pin.
0	1	0	0	1	IOB2/RFC9 pin is configured to RFC9 pin.
0	1	0	1	0	IOB3/RFC10 pin is configured to RFC10 pin.
0	1	0	1	1	IOB4/RFC11 pin is configured to RFC11 pin.
0	1	1	0	0	IOB5/RFC12 pin is configured to RFC12 pin.
0	1	1	0	1	IOB6/RFC13 pin is configured to RFC13 pin.
0	1	1	1	0	IOB7/RFC14 pin is configured to RFC14 pin.
0	1	1	1	1	IOC2/RFC15 pin is configured to RFC15 pin.
1	0	0	0	0	IOC1/RFC16 pin is configured to RFC16 pin.
1	0	0	0	1	DATA/RFC17 pin is configured to RFC17 pin.
1	0	0	1	0	CLK/RFC18 pin is configured to RFC18 pin.
Other					No function, don't use.

START: RFC counter enable bit.
= 0, Stop the RFC conversion, reset by hardware when conversion is finished or by software.
= 1, RFC counter start to convert.

RFCON: RFC module enable bit.
= 0, Disable RFC module, all the IOxx/RFCx pins are configured to IOXX pin, and the IOA1/CX pin is configured to IOA1 pin.
= 1, Enable RFC module, the selected IOxx/RFCx pin is configured to RFCx pin, and the IOA1/CX pin is configured to CX pin.

2.1.22 RFCDLB and RFCDHB (RFC Data Register Low Byte and High Byte) (Bank 3)

Read/Write-POR		R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x00B	RFCDLB	Low byte of 15 bit RFC conversion result							

Read/Write-POR		R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x00C	RFCDHB	RFCOV	RFCD14	RFCD13	RFCD12	RFCD11	RFCD10	RFCD9	RFCD8

Note: more bits' default state, please refer to Table 2.6.

RFCD14:RFCD0: RFC conversion result.

RFCOV: RFC counter overflow flag. Set when RFC counter overflow, reset by RFC counter reset.
= 0, Not overflow.
= 1, Overflow.

2.1.23 CMPDX (Data X Register Data Comparator) (Bank 3)

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
0x00D	CMPDX	DX7	DX6	DX5	DX4	DX3	DX2	DX1	DX0

Note: more bits' default state, please refer to Table 2.6.

DX7:DX0: The data X of data comparator, this data will be compared to CMPDY.

2.1.24 CMPDY (Data Y Register Data Comparator) (Bank 3)

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
0x00E	CMPDY	DY7	DY6	DY5	DY4	DY3	DY2	DY1	DY0

Note: more bits' default state, please refer to Table 2.6.

DY7:DY0: The data Y of data comparator, this data will be compared to CMPDX.

2.1.25 CMPSTAT (Data Comparator Status Register) (Bank 3)

Read/Write-POR		-	-	-	-	R-0	R-0	R-0	R-0
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x00F	CMPSTAT	-	-	-	-	CMPF3	CMPF2	CMPF1	CMPF0

Legend: - = unimplemented, read as '0'; more bits' default state, please refer to Table 2.6.

CMPF4:CMPF0: the error number of the compared result (0 ~ 8) of data comparator.

2.1.26 INTFLAG (Interrupt Status 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
0x03F	INTFLAG	SPITXIF	RFCIF	T3IF	T2IF	T1IF	SPIRXIF	INTIF	T0IF

Note: more bits' default state, please refer to Table 2.6.

T0IF: Timer0 overflow interrupt flag. Set when Timer0 overflows, reset by software.

INTIF: External INT pin interrupt flag. Set by falling edge on INT pin, reset by software.

SPIRXIF: SPI receive module interrupt flag. Set when SPI receiver buffer is full (SPI data transmission complete), reset by software.

T1IF: Timer1 match interrupt flag. Set when TMR1 register matches to PR1 register, reset by software.

T2IF: Timer2 match interrupt flag. Set when TMR2 register matches to PR2 register, reset by software.

T3IF: Timer3 match interrupt flag. Set when TMR3 register matches to PR3 register, reset by software.

RFCIF: RFC module interrupt flag. Set when RFC conversion is completed, reset by software.

SPITXIF: SPI transmit module interrupt flag. Set when SPI transmit buffer is empty (SPI data transmission start), reset by software.

2.1.27 ACC (Accumulator)

Read/Write-POR		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 2.6.

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

2.1.28 OPTION Register

Read/Write-POR		R/W-1	R/W-0	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
N/A	OPTION	/PHBCE	GIE	1	1	PSA	PS2	PS1	PS0

Accessed by OPTION / OPTIONR instruction.

Legend: 1 = Not used, must fix to “1”; more bits’ default state, please refer to Table 2.6.

By executing the OPTION instruction, the contents of the ACC Register will be transferred to the OPTION Register.
By executing the OPTIONR instruction, user can read this register into ACC.
The OPTION Register contains various control bits to configure the Timer0/WDT pre-scaler, Timer0, pull-high, and interrupt.

PS2:PS0: Pre-scaler rate selects bits.

PS2:PS0			Timer0 Rate	WDT Rate
0	0	0	1:2	1:1
0	0	1	1:4	1:2
0	1	0	1:8	1:4
0	1	1	1:16	1:8
1	0	0	1:32	1:16
1	0	1	1:64	1:32
1	1	0	1:128	1:64
1	1	1	1:256	1:128

PSA: Pre-scaler assign bit.
= 0, TMR0 (Timer0).
= 1, WDT (watch-dog timer).

GIE: Global interrupt enable bit.

= 0, Disable all interrupts. For wake-up from SLEEP mode through an interrupt event, the device will continue execution at the instruction after the SLEEP instruction.

= 1, Enable all un-masked interrupts. For wake-up from SLEEP mode through an interrupt event, the device will branch to the interrupt address (0x001).

Note: 1. The GIE bit is not writable bit. This bit is only set by ENI or RETFIE instructions, and cleared by DISI instruction or entering into interrupt subroutine.

2. 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. The RETFIE instruction will exit the interrupt routine and set the GIE bit to re-enable interrupt.

/PHBCE: Pull-high enable bit of the IOB0~IOB7, IOC4~IOC5 and IOE0~IOE5 pins.
= 0, Enable the internal pull-high of IOB0~ IOB7, IOC4~IOC5, and IOE0~IOE5 pins.
= 1, Disable the internal pull-high of IOB0~ IOB7, IOC4~IOC5, and IOE0~IOE5 pins.
Note: /PHB, /PHE are “AND” gating with /PHBCE, that is each one written “0” will enable pull-high.

2.1.29 IOSTA, IOSTB, IOSTC, IOSTD & IOSTE (Port I/O 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
0x005	IOSTA	IOSTA7	IOSTA6	IOSTA5	IOSTA4	IOSTA3	IOSTA2	IOSTA1	IOSTA0

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
0x006	IOSTB	IOSTB7	IOSTB6	IOSTB5	IOSTB4	IOSTB3	IOSTB2	IOSTB1	IOSTB0

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	IOSTC	IOSTC7	IOSTC6	IOSTC5	IOSTC4	IOSTC3	IOSTC2	IOSTC1	IOSTC0

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
0x008	IOSTD	IOSTD7	IOSTD6	IOSTD5	IOSTD4	IOSTD3	IOSTD2	IOSTD1	IOSTD0

Read/Write-POR		R/W-%	R/W-%	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	IOSTE	IOSTE7	IOSTE6	IOSTE5	IOSTE4	IOSTE3	IOSTE2	IOSTE1	IOSTE0

Accessed by IOST / IOSTR instruction.
Legend: % = define by TYPE of the configuration word, detail and more bits' default state, please refer to Table 2.6.

The Port I/O Control Registers are loaded with the contents of the ACC Register by executing the IOST R (0x05~0x09) instruction. By executing the IOSTR instruction, user can read these registers into ACC.
A '1' from a IOST Register bit puts the corresponding output driver in hi-impedance state (input mode). A '0' enables the output buffer and puts the contents of the output data latch on the selected pins (output mode).

2.1.30 T1CON (Timer 1 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	T1CON	GP	GP	GP	GP	GP	T1ON	T1P1	T1P0

Accessed by IOST / IOSTR instruction.
Note: more bits' default state, please refer to Table 2.6.

T1PS1:T1PS0: Timer1 pre-scaler rate select bits.

T1PS1:T1PS0		Pre-scaler Rate
0	0	1:1
0	1	1:4
1	0	1:8
1	1	1:16

T1ON: Timer 1 module enable bit.
= 0, Disable the Timer 1 module.
= 1, Enable the Timer 1 module.

GP: General purpose read/write bits.

2.1.31 PHCON (Pull-high Control 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
0x00D	PHCON	HDC	1	1	1	/PHE	/PHD	/PHB	/PHA

Accessed by IOST / IOSTR instruction.

Legend: 1 = Not used, must fix to "1"; more bits' default state, please refer to Table 2.6.

/PHA: = 0, Enable the internal pull-high of IOA0~ IOA7 pins.
= 1, Disable the internal pull-high of IOA0~ IOA7 pins.

/PHB: = 0, Enable the internal pull-high of IOB0~ IOB7 pins.
= 1, Disable the internal pull-high of IOB0~ IOB7 pins.
Note: /PHB is "AND" gating with /PHBCE, that is each one written "0" will enable pull-high.

/PHD: = 0, Enable the internal pull-high of IOD0~ IOD7 pins.
= 1, Disable the internal pull-high of IOD0~ IOD7 pins.

/PHE: = 0, Enable the internal pull-high of IOE0~ IOE5 pins.
= 1, Disable the internal pull-high of IOE0~ IOE5 pins.
Note: /PHE is "AND" gating with /PHBCE, that is each one written "0" will enable pull-high.

HDC: Driving ability enable bit of IOC0~IOC2 pins.
= 0, Normal driving ability of IOC0~IOC2 pins.
= 1, Reduce the driving ability of IOC0~IOC2 pins.

2.1.32 PCON (Power Control Register)

Read/Write-POR		R/W-1	R/W-0	R/W-1	-	R/W-0	-	-	R/W-1
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0x00E	PCON	LVDTE	ODE	WDTE	*	ROC	-	-	/WUE

Accessed by IOST / IOSTR instruction.

Legend: - = unimplemented, read as '0'; * = unimplemented, read as '1'; more bits' default state, please refer to Table 2.6.

/WUE: Input change wake-up function of IOB0~IOB7, IOC4~IOC5, and IOE0~IOE1 pins enable bit.
= 0, Enable the input change wake-up function.
= 1, Disable the input change wake-up function.

ROC: R-option function of IOD0 and IOD1 pins enable bit.
= 0, Disable the R-option function.
= 1, Enable the R-option function. In this case, if a 430KΩ external resistor is connected/disconnected to V_{SS}, the status of IOD0 (IOD1) is read as "0"/"1".

WDTE: WDT (watch-dog timer) enable bit.
= 0, Disable WDT.
= 1, Enable WDT.

ODE: Open-drain function of IOC6 and IOC7 pins enable bit.
= 0, Disable the open-drain function for both IOC6 and IOC7 pins.
= 1, Enable the open-drain function for both IOC6 and IOC7 pins.

LVDTE: LVDT (low voltage detector) enable bit.
= 0, Disable LVDT.
= 1, Enable LVDT.

2.1.33 INTEN (Interrupt Mask 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
0x00F	INTEN	SPITXIE	RFCIE	T3IE	T2IE	T1IE	SPIRXIE	INTIE	T0IE

Accessed by IOST / IOSTR instruction.
Note: more bits' default state, please refer to Table 2.6.

T0IE: Timer0 overflow interrupt enable bit.
= 0, Disable the Timer0 overflow interrupt.
= 1, Enable the Timer0 overflow interrupt.

INTIE: External INT pin interrupt enable bit.
= 0, Disable the External INT pin interrupt.
= 1, Enable the External INT pin interrupt.

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

T1IE: Timer1 match interrupt enable bit.
= 0, Disable the Timer1 match interrupt.
= 1, Enable the Timer1 match interrupt.

T2IE: Timer2 match interrupt enable bit.
= 0, Disable the Timer2 match interrupt.
= 1, Enable the Timer2 match interrupt.

T3IE: Timer3 match interrupt enable bit.
= 0, Disable the Timer3 match interrupt.
= 1, Enable the Timer3 match interrupt.

RFCIE: RFC module interrupt enable bit.
= 0, Disable the RFC module interrupt.
= 1, Enable the RFC module interrupt.

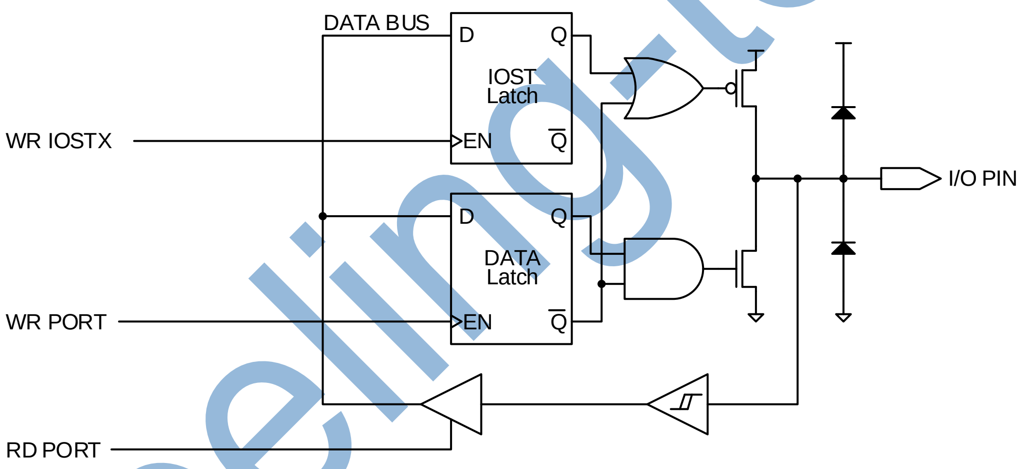
SPITXIE: SPI transmit module interrupt enable bit.
= 0, Disable the SPI transmit module interrupt.
= 1, Enable the SPI transmit module interrupt.

2.2 I/O Ports

PORTA, PORTB, PORTC, PORTD and PORTE are bi-directional tristate I/O ports.
All I/O pins (IOA, IOB, IOC, IOD and IOE) have data direction control registers (IOSTA, IOSTB, IOSTC, IOSTD, and IOSTE) which can configure these pins as output or input.
IOA<7:0>, IOB<7:0>, IOC<5:4>, IOD<7:0> and IOE<5:0> have corresponding pull-high control bits (/PHBCE, /PHA, /PHB, /PHD, and /PHE bits) to enable the weak internal pull-high. The weak pull-high is automatically turned off when the pin is configured as an output pin.
IOC6 and IOC7 have an open-drain control bit (ODE, PCON<6>) to enable the open-drain output when these pins are configured to be an output pin.
IOD0 and IOD1 are the R-option pins enabled by setting the ROC bit (PCON<3>). When the R-option function is used, it is recommended that IOD0 and IOD1 are used as output pins, and read the status of IOD0 and IOD1 before these pins are configured to be an output pin.
IOB0~IOB7, IOC4~IOC5, and IOE0~IOE1 also provide the input change interrupt/wake-up function which are enabled by clear /WUE bit (PCON<0>). The input change interrupt/wake-up function is automatically turned off when the pin is configured as an output pin.

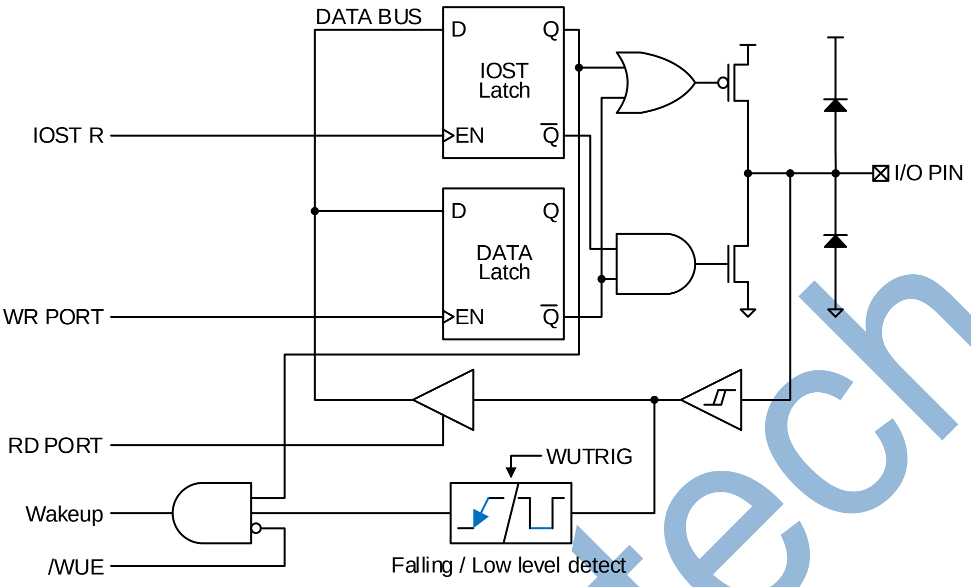
Figure 2.3: Block Diagram of I/O Pins

IOA7 ~ IOA0, IOC3 ~ IOC0, IOD7 ~ IOD0, IOE7 ~ IOE2 Pins:



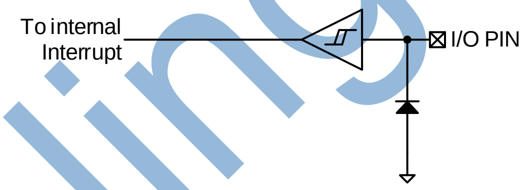
Pull-high/R-option are not shown in the figure

IOB7 ~ IOB0, IOE1, IOE0 Pins:



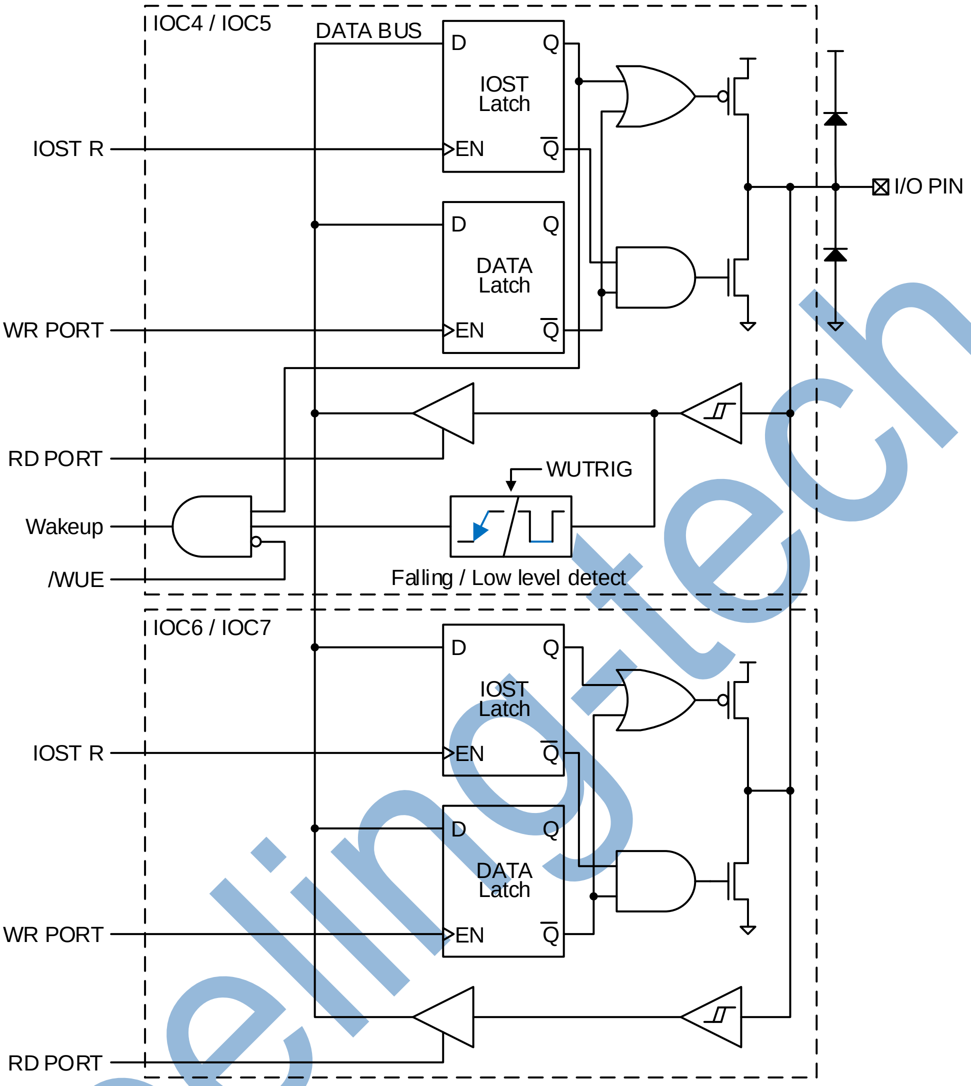
Pull-high are not shown in the figure

INT Pin:



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

IOC7 ~ IOC4 Pins:



Pull-high are not shown in the figure

Note: CLK pin is connecting IOC4 and IOC6 together.
DATA pin is connecting IOC5 and IOC7 together.

2.3 Timer0/WDT & Pre-scler

2.3.1 Timer0

The Timer0 is an 8-bit timer/counter. The clock source of Timer0 comes from the internal clock.
The timer0 register (TMR0) will increment every instruction cycle (without pre-scaler). If TMR0 register is written, the increment is inhibited for the following two cycles.

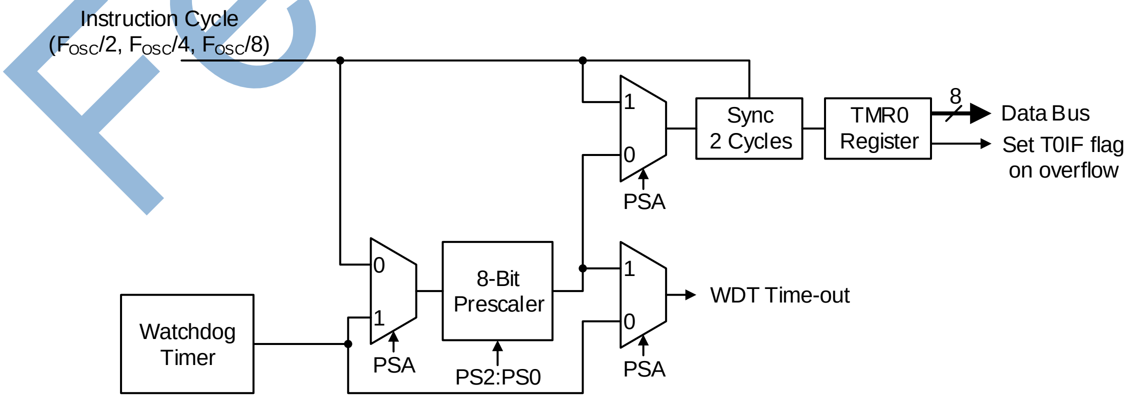
2.3.2 Watchdog Timer (WDT)

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. During normal operation or in SLEEP mode, a WDT time-out will cause the device reset and the $\overline{\text{TO}}$ bit (STATUS<4>) will be cleared.
The WDT can be disabled by clearing the control bit WDTE (PCON<5>) to "0".
The WDT has a nominal time-out period of 18 mS (without pre-scaler). If a longer time-out period is desired, a pre-scaler with a division ratio of up to 1:128 can be assigned to the WDT controlled by the OPTION register. Thus, the longest time-out period is approximately 2.3 seconds if reset delay time is set to 18ms.
The CLRWDT instruction clears the WDT and the pre-scaler, if assigned to the WDT, and prevents it from timing out and generating a device reset.
The SLEEP instruction resets the WDT and the pre-scaler, if assigned to the WDT. This gives the maximum SLEEP time before a WDT Wake-up Reset.

2.3.3 Pre-scaler

An 8-bit counter (down counter) is available as a pre-scaler for the Timer0, or as a post-scaler for the Watchdog Timer (WDT). Note that the pre-scaler may be used by either the Timer0 module or the WDT, but not both. Thus, a pre-scaler assignment for the Timer0 means that there is no pre-scaler for the WDT, and vice-versa.
The PSA bit (OPTION<3>) determines pre-scaler assignment. The PS<2:0> bits (OPTION<2:0>) determine pre-scaler ratio.
When the pre-scaler is assigned to the Timer0 module, all instructions writing to the TMR0 register will clear the pre-scaler. When it is assigned to WDT, a CLRWDT instruction will clear the pre-scaler along with the WDT.
The pre-scaler is neither readable nor writable. On a RESET, the pre-scaler contains all '1's.
To avoid an unintended device reset, CLRWDT or CLRR TMR0 instructions must be executed when changing the pre-scaler assignment from Timer0 to the WDT, and vice-versa.

Figure 2.4: Block Diagram of the Timer0/WDT Pre-scaler



2.4 Timer1

The Timer1 is an 8-bit clock counter with a programmable pre-scaler and an 8-bit period register (PR1). It also can be as a baud rate clock generator for the SPI module. The clock source of Timer1 comes from the internal clock (F_{OSC}/4). The option of Timer1 pre-scaler (1:1, 1:4, 1:8, 1:16) is defined by T1P1:T1P0 (T1CON<1:0>) bits. **The pre-scaler is cleared when a value is written to TMR1 or T1CON register, and during any kind of reset as well.**

The timer increments from 0x00 until it equals the period register (PR1). It then resets to 00h at the next increment cycle. The timer interrupt flag (T1IF) is set when the timer rollover to 0x00.

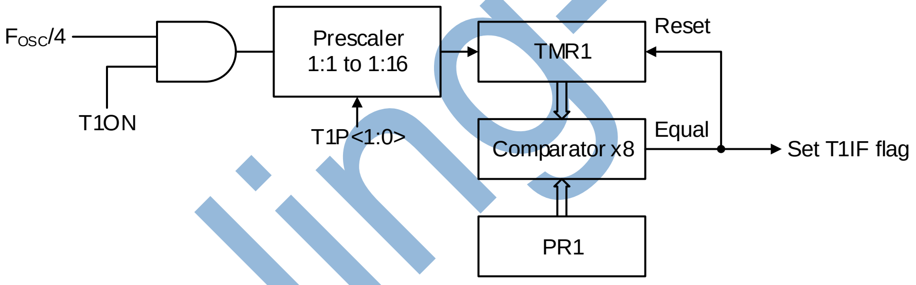
The timer also has a corresponding interrupt enable bit (T1IE). The timer interrupt can be enabled/disabled by setting/clearing this bit.

The timer s can be turned on and off under software control. When the timer on control bit (T1ON, T1CON<2>) is set, the timer increments from the clock source. When T1ON is cleared, the timer is turned off and cannot cause the timer interrupt flag to be set.

Table 2.1: Timer 1 Pre-scaler Rate

T1PS1:T1PS0		Pre-scaler Rate
0	0	1:1
0	1	1:4
1	0	1:8
1	1	1:16

Figure 2.5: Block Diagram of the Timer1



2.5 Timer2 & Timer3

Timer2 and Timer3 are two 8-bit incrementing timers, each with an 8-bit period register (PR2 and PR3, respectively) and separate overflow interrupt flags. Timer2 and Timer3 can operate either as timers (increment on internal clock, F_{OSC}/4), or as counters (increment on falling edge of external clock on pin IOB0/T23CKI). They are also software configurable to operate as a single 16-bit timer/counter. These timers are also used as the time base for the PWM (Pulse Width Modulation) modules.

2.5.1 Timer2 & Timer3 in 8-Bit Mode

Both Timer2 and Timer3 will operate in 8-bit mode when the T16 (T23CON<4>) bit is clear. These two timers can be independently configured to increment from the internal clock (F_{OSC}/4), or from an external clock source on IOB0/T23CKI pin. The timer clock source in configured by the TxCS bit (x = 2 for Timer2 (T23CON<0>), x = 3 for Timer3 (T23CON<2>)). When TxCS is set, the clock source is the IOB0/T23CKI pin and the counters will increment on every falling edge of the IOB0/T23CKI pin.

The timer increments from 0x00 until it equals the period register (PRx, x = 2 for Timer2, x = 3 for Timer3). It then resets to 0x00 at the next increment cycle. The timer interrupt flag is set when the timer rollover to 0x00. Timer2 and Timer3 have individual interrupt flag bits (T2IF and T3IF).

Each timer also has a corresponding interrupt enable bit (T2IE and T3IE). The timer interrupt can be enabled/disabled by setting/clearing this bit.

The timer s can be turned on and off under software control. When the timer on control bit (TxON, T23CON<1> and T23CON<3>) is set, the timer increments from the clock source. When TxON is cleared, the timer is turned off and cannot cause the timer interrupt flag to be set.

When TxCS is set, the clock source is the IOB0/T23CKI pin, and the counter will increment on every falling edge on the IOB0/T23CKI pin. The T23CKI input is synchronized with internal phase clocks. This cause a delay from the time a falling edge appears on T23CKI to the time TMR2 or TMR3 is actually incremented.

2.5.2 Timer2 & Timer3 in 16-bit Mode

To select 16-bit mode, set the T16 bit. In this mode, TMR2 and TMR3 are concatenated to from a 16-bit timer (TMR3:TMR2). The 16-bit timer increments until it matches the 16-bit period register (PR3:PR2). On the following timer clock, the timer value is reset to 0x0000, and the T2IF bit is set.

When selecting the clock source for the 16-bit timer, the T2CS bit control the entire 16-bit timer and T3CS is a “don’t care”. When T2CS is clear, the timer increments from the internal clock (F_{OSC}/4). When T2CS is set, the timer increments on every falling edge of the IOB0/T23CKI pin.

For the 16-bit timer to increment, both T2ON and T3ON bits must be set.

Table 2.2: Turning on 16-Bit Timer

T16	T3ON	T2ON	Result
1	1	1	16-bit timer (TMR3:TMR2) ON
1	0	1	16-bit timer ON, only TMR2 increments
1	x	0	16-bit timer OFF
0	1	1	Timers in 8-bit mode

When T2CS is set, the 16-bit TMR3:TMR2 increments on the falling edge of clock input T23CKI. The T23CKI input is synchronized with internal phase clocks. This cause a delay from the time a falling edge appears on T23CKI to the time TMR3:TMR2 is actually incremented.

Figure 2.6: Timer2 and Timer3 in Two 8-bit Timer/Counter Mode

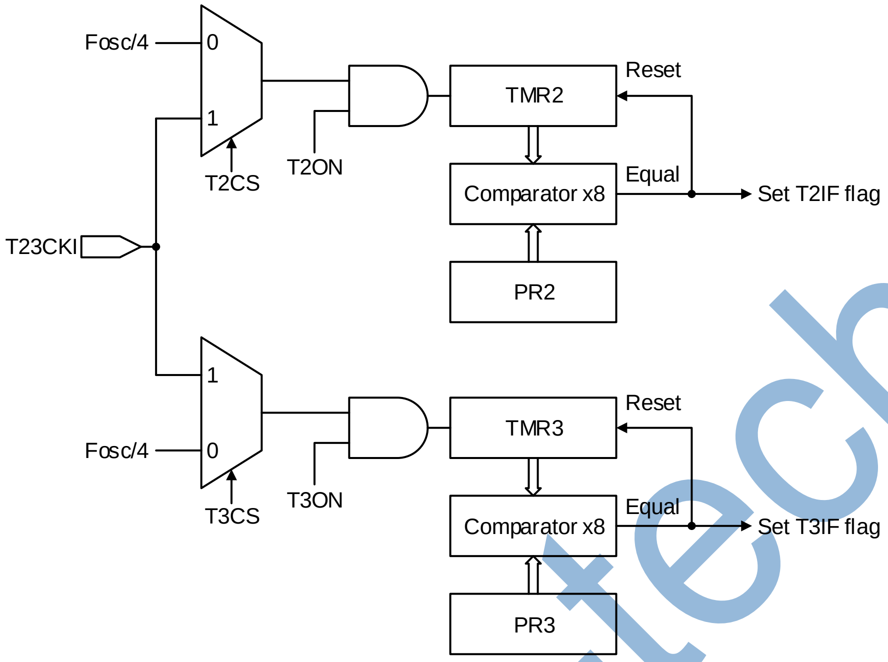
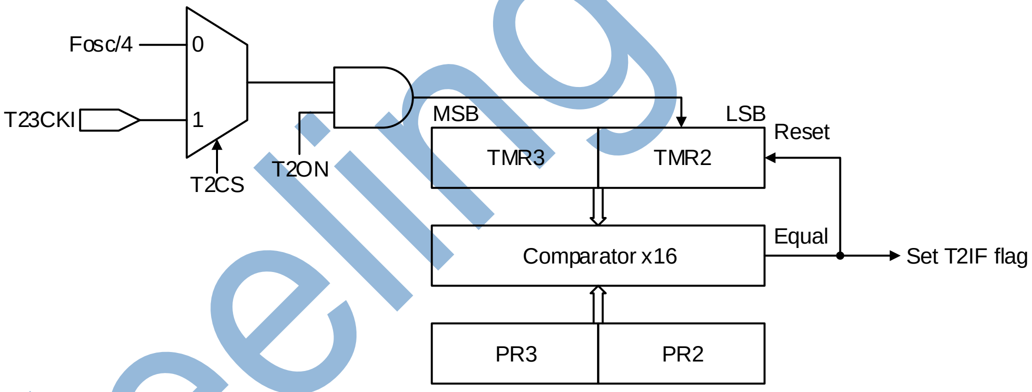


Figure 2.7: Timer2 and Timer3 in 16-bit Timer/Counter Mode



2.6 PWM (Pulse Width Modulation) Module

Two high speed pulse width modulation (PWM) outputs are provided. The PWM0 output uses Timer2 as its time base, the PWM1 may be configured to use Timer2 or Timer3 as the time base. The PWM outputs are on the IOA0/PWM0, and IOC0/PWM1 pins.

Each PWM output has a maximum resolution of 10-bits. The duty cycle of the output can vary from 0% to 100%.

The user needs to set the PW0ON bit (PWMCON<0>) to enable the PWM output. When PW0ON bit is set, the IOA0/PWM0 pin is configured as PWM0 output and forced as an output, irrespective of the data direct bit (IOSTA<0>). When the PW0ON is clear, the pin behaves as a port pin.

Similarly, the PW1ON bit (PWMCON<1>) controls the configuration of the IOC0/PWM1 pin.

2.6.1 PWM Periods

The period of PWM0 output is determined by Timer2 and its period register (PR2). The period of the PWM1 output can be software configured to use either Timer2 or Timer3 as the time base. For PWM1, when PW1T3 bit (PWMCON<2>) is clear, the time base is determined by TMR2 and PR2, and when PW1T3 bit is set, the time base is determined by TMR3 and PR3.

Running two different PWM outputs on two different timers allows different periods. Running all PWMs from Timer2 allows the best use of resources by freeing Timer3 to operate as an 8-bit timer. Timer2 and Timer3 cannot be used as a 16-bit timer if any PWM is being used.

The PWM periods can be calculated as follows:

Period of PWM0 = [(PR2) + 1] x 4Tosc

Period of PWM1 = [(PR2) + 1] x 4Tosc or
[(PR3) + 1] x 4Tosc

The duty cycle of PWMx is determined by the 10-bit value DCx<9:0>. The upper 8-bits are from register PWxDCH and lower2-bits are from PWxDCL<7:6> (PWxDCH:PWxDCL<7:6>).

The PWMx duty cycle is as follows:

PWMx Duty Cycle = (DCx) x Tosc
Where DCx represents the 10-bit value from PWxDCH:PWxDCL.

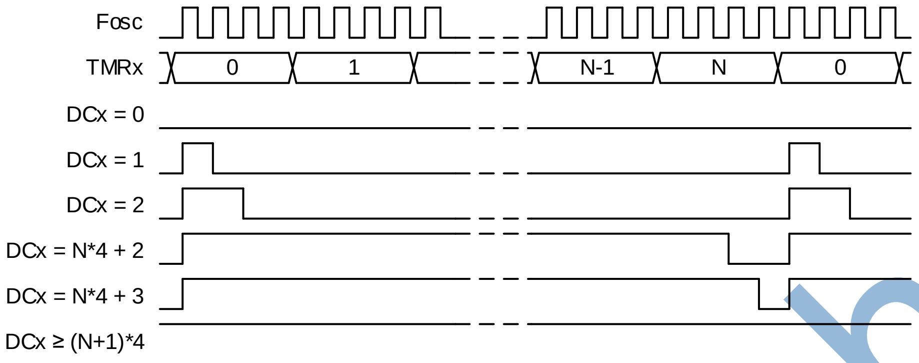
If DCx = 0, then the duty cycle is zero. If PRx = PWxDCH, then the PWM output will be low for one to four T_{OSC} (depending on the state of the PWxDCL<7:6> bits). For a duty cycle to be 100%, the PWxDCH value must be greater than the PRx value.

The duty cycle registers for both PWM outputs are double buffered. When the user writes to these registers, they are stored in master latches. When TMR2 (or TMR3) overflows and a new PWM period beings, the master latch values are transferred to the slave latches.

2.6.2 PWM Interrupts

The PWM modules make use of the TMR2 and/or TMR3 interrupts. A timer interrupt is generated when TMR2 or TMR3 equals its period register and on the following increment is cleared to zero. This interrupt also marks the beginning of a PWM cycle. The user can write new duty cycle values before the timer rollover. The TMR2 interrupt is latched into the T2IF bit and the TMR3 interrupt is latched into the T3IF bit. These flags must be cleared in software.

Figure 2.8: The output Waveform of PWM Driver (PRx = N)



2.7 SPI (Serial Peripheral Interface) Module

The Serial Port Interface (SPI) Module is a serial interface useful 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 three pins are used:

1. Serial Clock (SCK)
2. Serial Data In (SDI)
3. Serial Data Output (SDO)

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

4. Slave Select (SSB)

The SPI consists of a transmit/receive shift register (SPISR), a receive buffer register (SPIRXB), and a transmit buffer register (SPITXB). The SPISR shifts the data in and out of the device, MSB first. Once the first valid clock pulse appears on SCK pin (controlled by SSE (SPICON<4>) and SSEMOD (SPICON<3>) bits), data in SPITXB will be loaded into SPISR and start to shift in/out, then transmit buffer empty detect bit TXBF (SPISTAT<5>), and interrupt flag bits SPITXIF (INTFLAG<7>) and TXBFIF (SPISTAT<6>) are set. Once the 8-bits of data have been received, the data in SPISR will be moved to the SPIRXB register, then receive buffer full detect bit RXBF (SPISTAT<0>), and interrupt flag bits SPIRXIF (INTFLAG<2>) and RXBFIF (SPISTAT<1>) are set.

If FM8PE51M is a master controller, it sends clock through the SCK pin. A couple of 8-bit data are transmitted and received at the same time. And if FM8PE51M is defined as a slave, its SCK pin could be programmed as an input pin. Data will continue to be shifted based on both the clock rate and the selected edge.

When the application S/W is expecting to transmit valid data, the SPITXB should be written before the SSE bit is set (SSEMOD = 0) or the next data packet SCK pulse appears (SSEMOD = 1). Buffer empty bit TXBF indicates when SPISR has been loaded with the data of SPITXB (reception/transmission start). The TXBF bit is cleared by software or by writing SPITXB register. And the TXBF bit may be ignored if the SPI is only a receiver.

Also when the application S/W is expecting to receive valid data, the SPIRXB should be read before the next byte of data have been received completely. Buffer full bit RXBF indicates when SPIRXB has been loaded with the received data (reception/transmission is complete). The RXBF bit is cleared by software or by reading SPIRXB register. And the RXBF bit may be ignored if the SPI is only a transmitter.

Generally, the SPI interrupt is used to determine when the transmission/reception has started/completed, the SPIRXB/SPITXB must be read and/or written. If the interrupt method is not going to be used, then S/W polling RXBF and/or TXBF bits is needed.

If the SPI is only going to receive, the TXBF flag could be ignored.

Figure 2.9: SPI Block Diagram

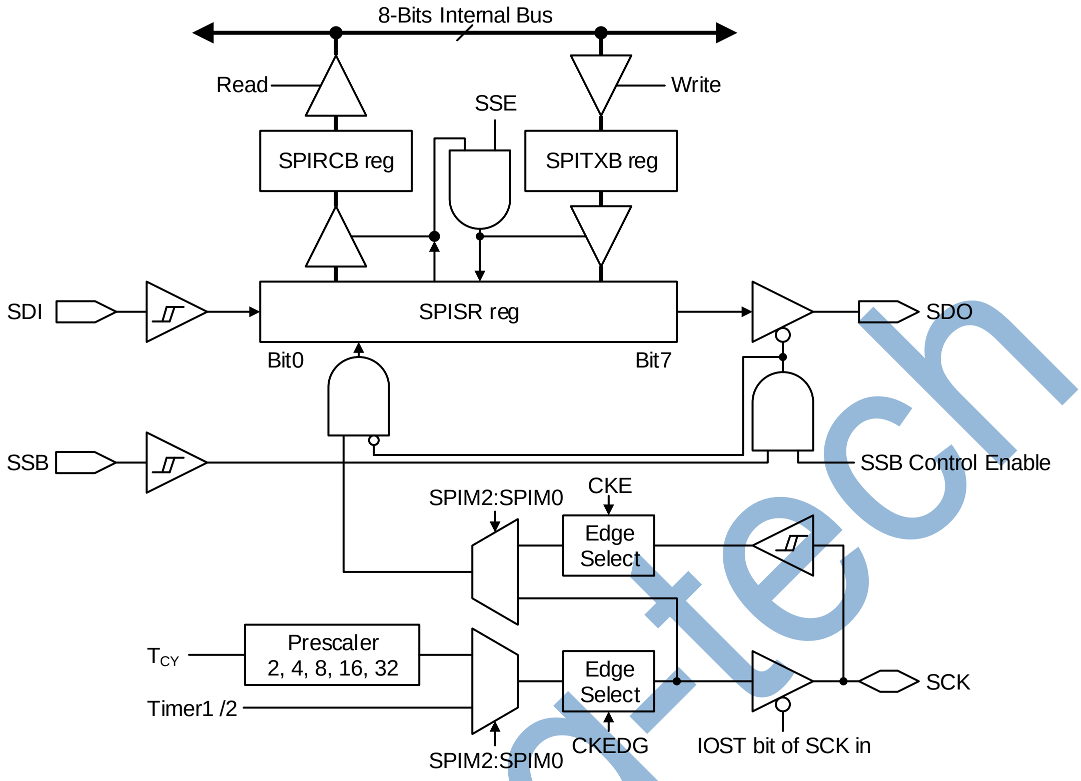


Table 2.3: SPI Mode Setting

SPIM2:SPIM0			SPI Mode
0	0	0	SPI master mode, clock = $F_{OSC}/2$
0	0	1	SPI master mode, clock = $F_{OSC}/4$
0	1	0	SPI master mode, clock = $F_{OSC}/8$
0	1	1	SPI master mode, clock = $F_{OSC}/16$
1	0	0	SPI master mode, clock = $F_{OSC}/32$
1	0	1	SPI slave mode, clock = SCK pin, SSB pin control enabled
1	1	0	SPI slave mode, clock = SCK pin, SSB pin control disabled
1	1	1	SPI master mode, clock = Timer1 output/2

Table 2.4: The Description SPI SCK Control Bit

CKEDG:	= 0, Data shifts out on rising edge of SCK, and shifts in on falling edge of SCK. = 1, Data shifts out on falling edge of SCK, and shifts in on rising edge of SCK.

2.7.1 Master Mode

2.7.1.1 Master Mode with SSE Control (SSEMOD = 0)

In this master mode, the data is transmitted/received as soon as the SPI shift register enable bit SSE (SPICON<4>) 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, then transmit buffer empty detect bit (TXBF), and interrupt flag bits (SPITXIF, TXBFIF) are set. And then user could write the next byte data to SPITXB register before the SSE bit is set if next byte transmission is needed. 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, then buffer full detect bit (RXBF), interrupt flag bits (SPIRXIF, RXBFIF) are set. And then user could read out the SPIRXB register before next 8-bit data transmission is completed if needed.

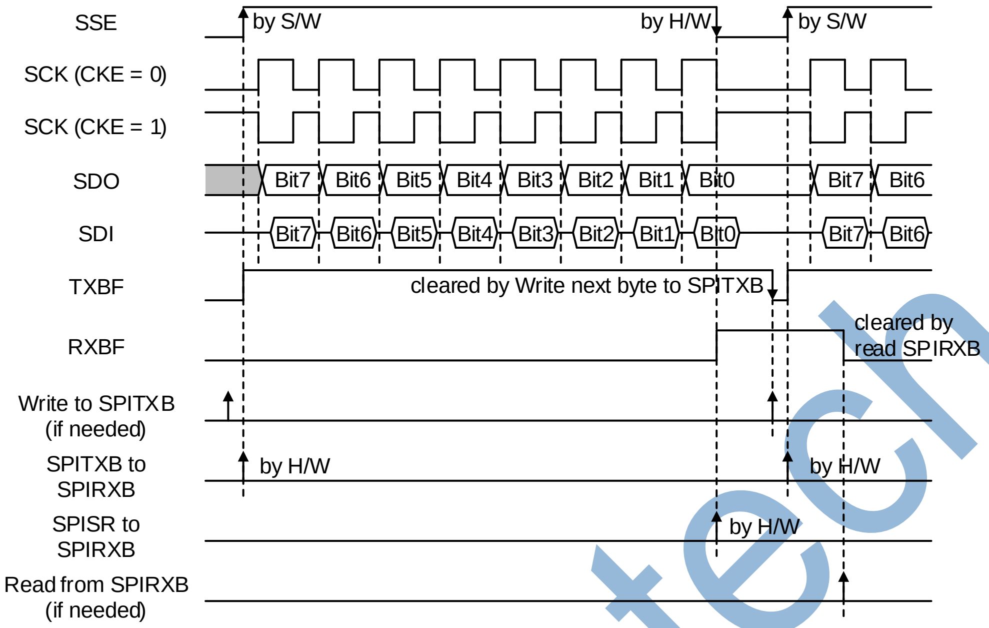
How to transmit/receive data in this master mode:
1. Enable SPI function by setting the SPION (SPICON<6>) bit.
2. Decide the transmission rate and source by programming SPIM2:SPIM0 (SPICON<2:0>) bits.
3. Write the data that you want to transmit to SPITXB register if needed.
4. Set SSE (SPICON<4>) bit to start transmit.
5. When the 8-bit data transmission starts, both of the SPITXIF and TXBFIF interrupt flags will set to 1. Besides, both of these bits are cleared by software. The TXBF flag also will be set to "1", cleared by software or by writing data to SPITXB register.
6. After 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.
7. Also after the 8-bit data transmission is completed, both of the SPIRXIF and RXBFIF interrupt flags will set to 1. Besides, both of these bits are cleared by software. The RXBF flag also will be set to "1", cleared by software or by reading out SPIRXB register.
8. Read out the SPIRXB register before next byte transmission being finished if needed.

2.7.1.2 Master Mode without SSE Control (SSEMOD = 1)

In this master mode, the data is transmitted/received as soon as write data to SPITXB register by S/W. The data in SPITXB will be loaded into SPISR at the same time and start to shift in/out, then transmit buffer empty detect bit (TXBF), and interrupt flag bits (SPITXIF, TXBFIF) are set. And then user could write the next byte data to SPITXB register if next byte transmission is needed. If the next byte data is not written into SPITXB, the communication will be stopped after the 8-bit data transmission is completed. Once the 8-bits of data have been received, the data in SPISR will be moved to the SPIRXB register, then buffer full detect bit (RXBF), interrupt flag bits (SPIRXIF, RXBFIF) 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 SPION (SPICON<6>) bit.
2. Decide the transmission rate and source by programming SPIM2:SPIM0 (SPICON<2:0>) bits.
3. Write the data that you want to transmit to SPITXB register to start transmit.
4. When the 8-bit data transmission starts, both of the SPITXIF and TXBFIF interrupt flags will set to 1. Besides, both of these bits are cleared by software. The TXBF flag also will be set to “1”, cleared by software or by writing data to SPITXB register.
5. After the 8-bit data transmission is completed, if user wants to transmit/receive another 8-bit data, write next byte data to SPITXB register before this byte transmission being finished.
6. Also after the 8-bit data transmission is completed, both of the SPIRXIF and RXBFIF interrupt flags will set to 1. Besides, both of these bits are cleared by software. The RXBF flag also will be set to “1”, cleared by software or by reading out SPIRXB register.
7. Read out the SPIRXB register before next byte transmission being finished if needed.

Figure 2.10: SPI Mode Timing (Master Mode)



2.7.2 Slave Mode

2.7.2.1 Slave Mode with SSE Control (SSEMOD = 0)

In this slave mode, the data is transmitted and received as the external clock pulses appear on SCK pin. Once the SPI shift register enable bit SSE (SPICON<4>) has been set to “1”, data in SPITXB will be loaded into SPISR and start to shift in/out, then transmit buffer empty detect bit (TXBF), and interrupt flag bits (SPITXIF, TXBFIF) are set. And then user could write the next byte data to SPITXB register before the SSE bit is set if next byte transmission is needed. Once the 8-bits of data have been received, the data in SPISR will be moved to the SPIRXB register, then buffer full detect bit (RXBF), interrupt flag bits (SPIRXIF, RXBFIF) 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 (SPICON<2:0> = 101). When the SSB pin is low, transmission and reception are enabled and the SDO pin is driven. When the SSB pin goes high, the SDO 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.

When the SPI module resets, the bit counter is forced to 0. This can be done by forcing the SSB pin to high level or clearing the SPION bit (SPICON<6>).

How to transmit/receive data in this slave mode:

1. Enable SPI function by setting the SPION (SPICON<6>) bit.
2. Enable/disable the SSB pin control by programming SPIM2:SPIM0 (SPICON<2:0>) bits.
3. Write the data that you want to transmit to SPITXB register if needed.
4. Set SSE (SPICON<4>) bit and wait the external clock pulses appear on SCK pin to start transmit.
5. When the 8-bit data transmission starts, both of the SPITXIF and TXBFIF interrupt flags will set to 1. Besides, both of these bits are cleared by software. The TXBF flag also will be set to "1", cleared by software or by writing data to SPITXB register.
6. After 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 before next clock pulse appearing SCK pin.

7. Also after the 8-bit data transmission is completed, both of the SPIRXIF and RXBFIF interrupt flags will set to “1”. Besides, both of these bits are cleared by software. The RXBF flag also will be set to "1", cleared by software or by reading out SPIRXB register.
8. Read out the SPIRXB register before next byte transmission being finished if needed.

2.7.2.2 Slave Mode without SSE Control (SSEMOD = 1)

In this slave mode, the data is transmitted and received as the external clock pulses appear on SCK pin. Once the clock pulse appears on SCK pin, data in SPITXB will be loaded into SPISR and start to shift in/out, then transmit buffer empty detect bit (TXBF), and interrupt flag bits (SPITXIF, TXBFIF) are set. And then user could write the next byte data to SPITXB register before the next data packet SCK pulse appears if next byte transmission is needed. Once the 8-bits of data have been received, the data in SPISR will be moved to the SPIRXB register, then buffer full detect bit (RXBF), interrupt flag bits (SPIRXIF, RXBFIF) 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 (SPICON<2:0> = 101). When the SSB pin is low, transmission and reception are enabled and the SDO pin is driven. When the SSB pin goes high, the SDO 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.
When the SPI module resets, the bit counter is forced to 0. This can be done by forcing the SSB pin to high level or clearing the SPION bit (SPICON<6>).

How to transmit/receive data in this slave mode:
1. Enable SPI function by setting the SPION (SPICON<6>) bit.
2. Enable/disable the SSB pin control by programming SPIM2:SPIM0 (SPICON<2:0>) bits.
3. Write the data that you want to transmit to SPITXB register.
4. Wait the external clock pulses appear on SCK pin to start transmit.
5. When the 8-bit data transmission starts, both of the SPITXIF and TXBFIF interrupt flags will set to 1. Besides, both of these bits are cleared by software. The TXBF flag also will be set to “1”, cleared by software or by writing data to SPITXB register.
6. After the 8-bit data transmission is completed, if user wants to transmit/receive another 8-bit data, write next byte data to SPITXB register before next clock pulse appearing SCK pin.
7. Also after the 8-bit data transmission is completed, both of the SPIRXIF and RXBFIF interrupt flags will set to 1. Besides, both of these bits are cleared by software. The RXBF flag also will be set to “1”, cleared by software or by reading out SPIRXB register.
8. Read out the SPIRXB register before next byte transmission being finished if needed.

Figure 2.11: SPI Mode Timing (Slave Mode, with SSB control enabled)

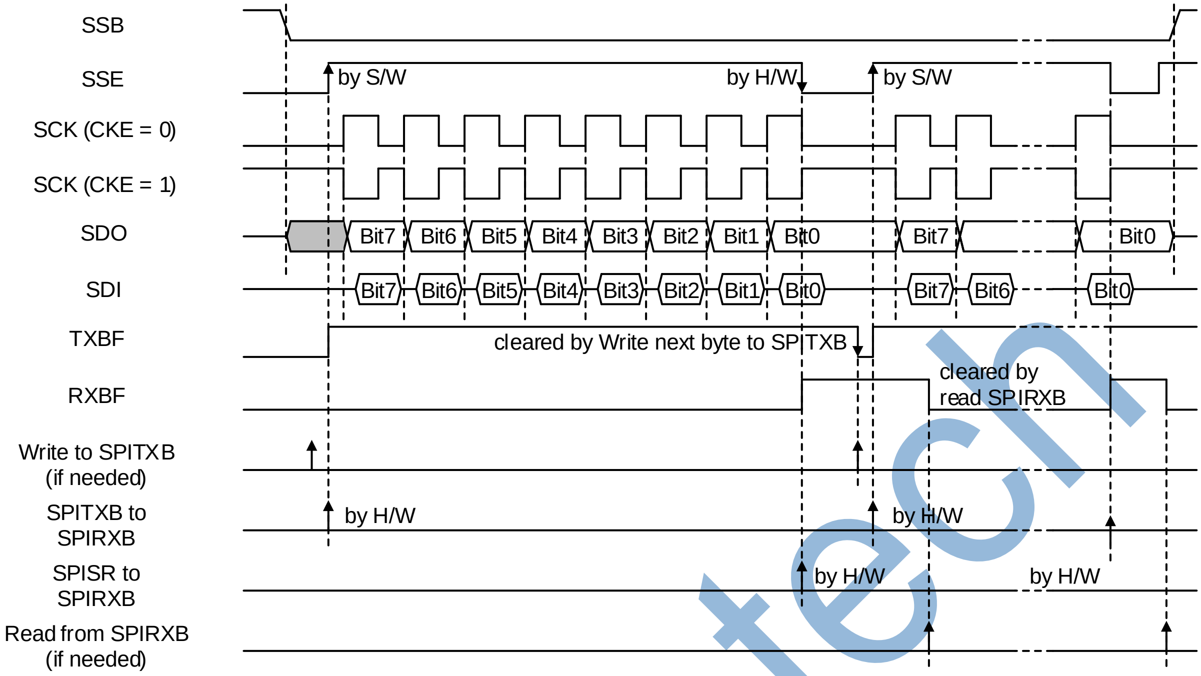
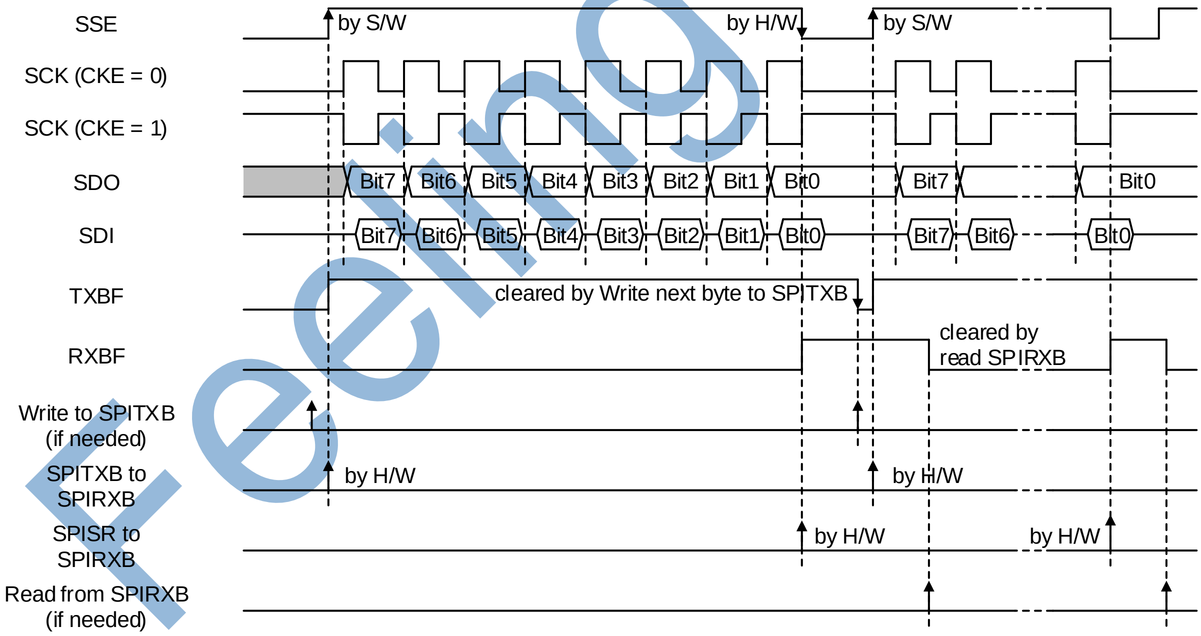


Figure 2.12: SPI Mode Timing (Slave Mode, with SSB control disabled)



2.8 Interrupts

The FM8PE51M has up to eight sources of interrupt:

1. External interrupt INT pin.
2. TMR0 overflow interrupt.
3. TMR1 match interrupt.
4. TMR2 match interrupt.
5. TMR3 match interrupt.
6. SPI receive module interrupt.
7. SPI transmit module interrupt.
8. RFC module interrupt.

INTFLAG is the interrupt flag register that recodes the interrupt requests in the relative flags.
A global interrupt enable bit, GIE (OPTION<6>), enables (if set) all un-masked interrupts or disables (if cleared) all interrupts. Individual interrupts can be enabled/disabled through their corresponding enable bits in INTEN register regardless of the status of the GIE bit.
When an interrupt event 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 0x001.
The interrupt flag bits must be cleared by software before re-enabling GIE bit to avoid recursive interrupts.
Executing the ENI instruction will set the GIE bit, and executing the DISI instruction will clear the GIE bit.
The RETFIE instruction exits the interrupt routine and set the GIE bit to re-enable interrupt.
The flag bit in INTFLAG register is set by interrupt event regardless of the status of its mask bit. Reading the INTFLAG register will be the logic AND of INTFLAG and INTEN.
When an interrupt is generated by the INT instruction, the next instruction will be fetched from address 0x002.

2.8.1 External INT Interrupt

External interrupt on INT pin is falling edge triggered.
When a falling edge appears on the INT pin the flag bit INTIF (INTFLAG<1>) is set. This interrupt can be disables by clearing INTIE bit (INTEN<1>).

2.8.2 Timer0 Interrupt

An overflow (0xFF → 0x00) in the TMR0 register will set the flag bit T0IF (INTFLAG<0>).
This interrupt can be disabled by clearing T0IE bit (INTEN<0>).

2.8.3 Timer1 Interrupt

A match condition (TMR1 = PR1) in the TMR1 register will set the flag bits T1IF (INTFLAG<3>) and TM1IF (SPISTAT<4>).
If user wants to clear the T1IF bit, user needs to clear the TM1IF bit first, and then the T1IF bit can be cleared by software.
This interrupt can be disabled by clearing T1IE bit (INTEN<3>).

2.8.4 Timer2 Interrupt

In 8-bit mode, a match condition (TMR2 = PR2) in the TMR2 register will set the flag bit T2IF (INTFLAG<4>).
In 16-bit mode, a match condition (TMR3:TMR2 = PR3:PR2) in the TMR3 and TMR2 register will set the flag bit T2IF (INTFLAG<4>).
This interrupt can be disabled by clearing T2IE bit (INTEN<4>).

2.8.5 Timer3 Interrupt

In 8-bit mode, a match condition (TMR3 = PR3) in the TMR1 register will set the flag bit T3IF (INTFLAG<5>).
In 16-bit mode, the Timer3 interrupt will be disabled.
This interrupt can be disabled by clearing T3IE bit (INTEN<5>).

2.8.6 SPI Receive Module Interrupt

After one byte of SPI transmission is completed, both the flag bit SPIRXIF (INTFLAG<2>) and RXBFIF (SPISTAT<1>) will be set.
If user wants to clear the SPIRXIF bit, user needs to clear the RXBFIF bit first, and then the SPIRXIF bit can be cleared by software.
This interrupt can be disabled by clearing SPIRXIE bit (INTEN<2>).

2.8.7 SPI Transmit Module Interrupt

After one byte of SPI transmission start, both the flag bit SPITXIF (INTFLAG<7>) and TXBFIF (SPISTAT<6>) will be set.
If user wants to clear the SPITXIF bit, user needs to clear the TXBFIF bit first, and then the SPITXIF bit can be cleared by software.
This interrupt can be disabled by clearing SPITXIE bit (INTEN<7>).

2.8.8 RFC Module Interrupt

After RFC conversion is finished, the RFCIF flag (INTFLAG<6>) will be set.
This interrupt can be disabled by clearing RFCIE bit (INTEN<6>).

2.9 Resistor to Frequency Converter (RFC)

The Resistor to Frequency Converter (RFC) can compare nineteen different sensors with the reference resistor separately.

This RFC contains nineteen external pins:

CX: the oscillation Schmitt-trigger input (IOA1/CX pin).

RFC0 ~ RFC18: the resistor/sensor output pin 0 ~ 18 (IOA2/RFC0, IOA3/RFC1, IOA4/RFC2, IOA5/RFC3, IOA6/RFC4, IOA7/RFC5, IOD0/RFC6, IOD1/RFC7, IOB1/RFC8, IOB2/RFC9, IOB3/RFC10, IOB4/RFC11, IOB5/RFC12, IOB6/RFC13, IOB7/RFC14, IOC2/RFC15, IOC1/RFC16, DATA/RFC17, and CLK/RFC18 pins)

Figure 2.13: The Block Diagram of RFC

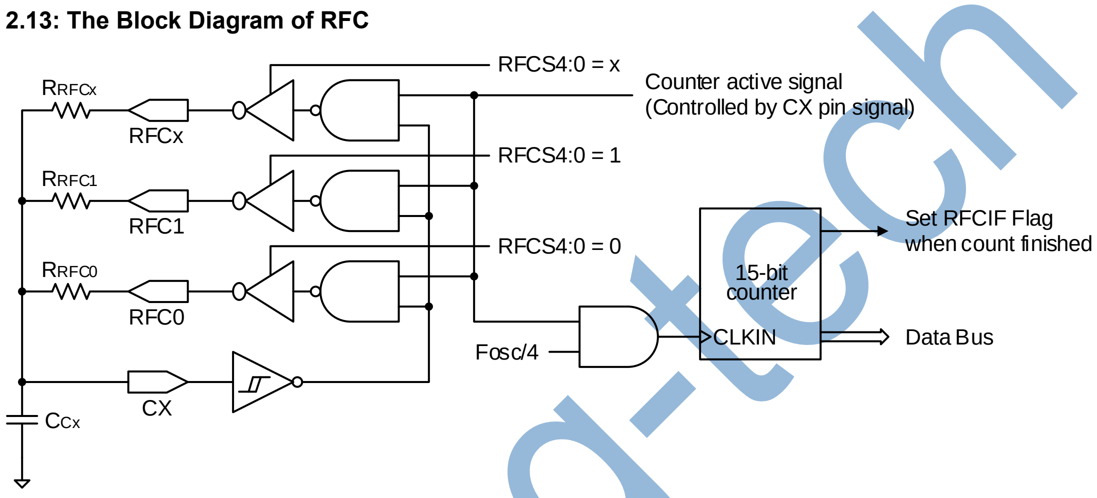


Table 2.5: The Description of RFC Control Bits

RFCS4:RFCS0	Select one the RFC oscillation network of RFCx (x = 0 to 18). The selected IOxx/RFCx pin will be configured as RFCx output pin if RFCON = 1. Other IOxx/RFCx pins will still behave as port pins. If RFCON = 0, all IOxx/RFCx pins will still behave as port pins.
START	= 0, Stop the RFC conversion = 1, RFC counter start to convert. Reset by hardware after conversion is finished.
RFCON	= 0, Disable RFC module, all the IOxx/RFCx pins are configured to IOxx pin, and the IOA1/CX pin is configured to IOA1 pin. = 1, Enable RFC module, the selected IOXX/RFCx pin is configured to RFCx pin, and the IOA1/CX pin is configured to CX pin.

2.9.1 RC Oscillator Network

The RFC circuitry may build up 19 RC oscillation networks through RFC0 to RFC18 and CX pins with external resistors. Only one RC oscillation network may be active at a time. When the oscillation network is built up, the count active pulse will be generated by the oscillation network and transferred to the 15-bit counter through the CX pin. It will then enable or disable the 15-bit counter in order to count the oscillation clock. **The 15-bit RFC counter is cleared when a value is written to RFCCON register, RFCON bit is cleared, and during any kind of reset as well.**

How to build the RC oscillation network:

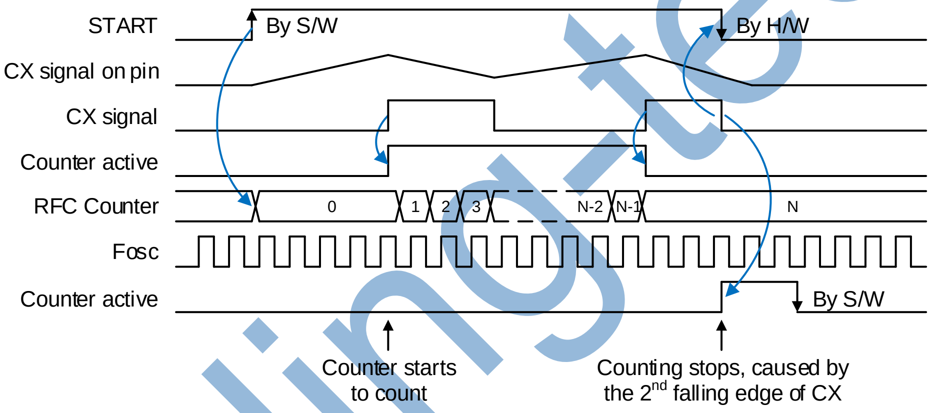
1. Connect the resistor and capacitor on RFCx (x = 0 to 18, if needed) and CX pins.
2. Switch all of the needed RFCx and CX pins to input mode.
3. Enable the RFC module by set the RFCON bit.

4. Select one of RFCx pins by RFCS4:RFCS0 bits to enable the output pin for RC networks respectively. The selected RFCx will output low at this time. Other RFCx pins will become of a tristate type.
5. Set START bit to enable the RC oscillation network and 15-bit counter. The RC oscillation network will not operate if this bit has not been set. After conversion is finished, the START bit will be cleared by hardware and the RFCIF flag will be set (if enable).

2.9.2 Enable/Disable the Counter by CX Signal

In this module, CX pin is the signal to control the counter period and the clock source of the counter comes from the internal system clock (F_{OSC}).
The counter will start to count after the first rising edge signal applied on the CX pin after the RFCON bit (RFCCON<7>) is set. Once the second rising edge is applied to the CX pin after the counter is enabled, the counter will stop counting. And after the second falling edge is applied to the CX pin, the RFC block will clear the START bit and set the RFC interrupt flag RFCIF bit (INTFLAG<6>) if RFCIE bit is set.
User also can be polling the RFCON or RFCIF bit to check if the conversion is finished.

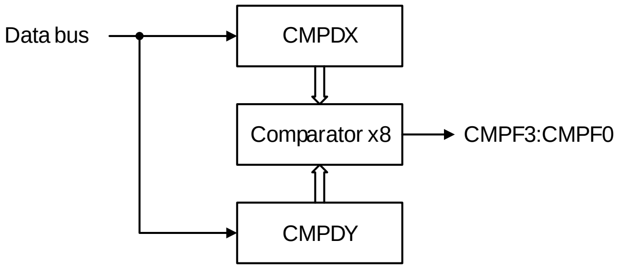
Figure 2.14: The Sample of the RFC Counter Controlled by the CX Pin



2.10 Data Comparator

The data comparator can compare two different data in compared data register (CMPDX, CMPDY), and output the number of different/error bits between CMPDX and CMPDY into CMPF3:CMPF0 bits (CMPSTAT<3:0>).
The data comparator will auto-compare the data of CMPDX to the data of CMPDY whenever anyone of (CMPDX, CMPDY) is changed.
If user write 0x00 to CMPDX, and write one 8-bit data to CMPDY, then CMPF3:CMPF0 means the number of “1” of the 8-bit data. Similarly, if user write 0xFF to CMPDX, and write one 8-bit data to CMPDY, then CMPF3:CMPF0 means the number of “0” of the 8-bit data.

Figure 2.15: The Block Diagram of Data Comparator



2.11 Power-down Mode (SLEEP)

Power-down mode is entered by executing a SLEEP instruction.
When SLEEP instruction is executed, the $\overline{\text{PD}}$ bit (STATUS<3>) is cleared, the $\overline{\text{TO}}$ bit (STATUS<4>) 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.

2.11.1 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. Input change wake-up

External RSTB reset and WDT time-out reset will cause a device reset. The $\overline{\text{PD}}$ and $\overline{\text{TO}}$ bits can be used to determine the cause of device reset. The $\overline{\text{PD}}$ bit is set on power-up and is cleared when SLEEP instruction is executed. The $\overline{\text{TO}}$ bit is cleared if a WDT time-out occurred.
Any one of the wake-up pins is set to “0”, the device will wake-up and continue execution at the instruction after the SLEEP instruction. In this case, before entering SLEEP mode, the wake-up function of trigger sources (IOB0~IOB7, IOC4~IOC5, and IOE0~IOE1) should be selected (e.g. input pin) and enabled (e.g. pull-high, wake-up control (/WUE bit, PCON<0>)).
The system wake-up delay time is 18ms plus 16 oscillator cycle time.

2.12 Reset

FM8PE51M devices 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 (LVD) 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 $\overline{\text{TO}}$ and $\overline{\text{PD}}$ bits (STATUS<4:3>) are set or cleared depending on the different reset conditions.

2.12.1 Power-up Reset Timer (PWRT)

The Power-up Reset Timer provides a nominal 18/4.5/288/72mS 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.

2.12.2 Oscillator Start-up Timer (OST)

The OST timer provides a 16 oscillator cycle delay (from OSCI input) after the PWRT delay (18/4.5/288/72mS) is over. This delay ensures that the X'tal oscillator or resonator has started and stabilized. The device is kept in reset state as long as the OST is active.

This counter only starts incrementing after the amplitude of the OSCI signal reaches the oscillator input thresholds.

2.12.3 Reset Sequence

When Power-on Reset (POR), Brown-out Reset (BOR), RSTB Reset or WDT time-out Reset is detected, the reset sequence is as follows:

1. The reset latch is set and the PWRT & OST are cleared.
2. When the internal POR, BOR, RSTB Reset or WDT time-out Reset pulse is finished, then the PWRT begins counting.
3. After the PWRT time-out, the OST is activated.
4. And after the OST delay is over, the reset latch will be cleared and thus end the on-chip reset signal.

The totally system reset delay time is 18/4.5/288/72mS plus 16 oscillator cycle time.

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

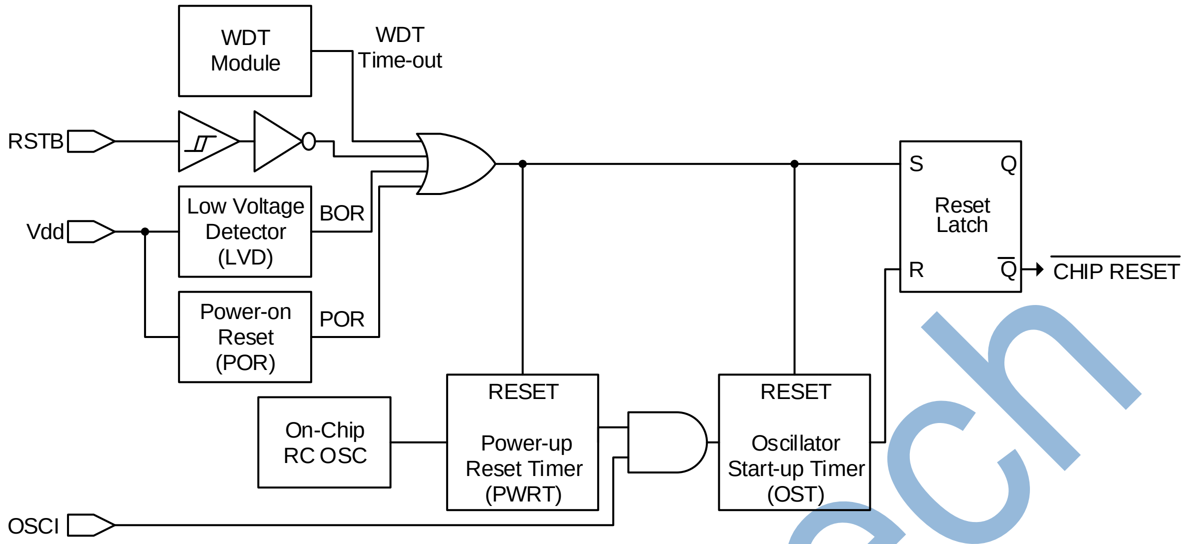


Table 2.6: Reset Conditions for All Registers

Register	Address	Power-on Reset Brown-out Reset	RSTB Reset	WDT Reset
ACC	N/A	xxxx xxxx	uuuu uuuu	uuuu uuuu
OPTION	N/A	1011 1111	1011 1111	1u11 1111
IOSTA	0x05	1111 1111	1111 1111	1111 1111
IOSTB	0x06	1111 1111	1111 1111	1111 1111
IOSTC	0x07	1111 1111	1111 1111	1111 1111
IOSTD	0x08	1111 1111	1111 1111	1111 1111
IOSTE	0x09	TYPE A: 1111 1111 TYPE B: 0011 1111	TYPE A: 1111 1111 TYPE B: 0011 1111	TYPE A: 1111 1111 TYPE B: 0011 1111
T1CON	0x0C	0000 0000	0000 0000	0000 0000
PHCON	0x0D	1111 1111	1111 1111	1111 1111
PCON	0x0E	101- 0--1	101- 0–1	101- 0--1
INTEN	0x0F	0000 0000	0000 0000	0000 0000
INDF	0x00, unbanked	xxxx xxxx	xxxx xxxx	xxxx xxxx
TMR0	0x01, unbanked	xxxx xxxx	xxxx xxxx	xxxx xxxx
PCL	0x02, unbanked	1111 1111	1111 1111	1111 1111
STATUS	0x03, unbanked	0001 1xxx	000# #uuu	000# #uuu
FSR	0x04, unbanked	00xx xxxx	00uu uuuu	00uu uuuu
PORTA	0x05, unbanked	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTB	0x06, unbanked	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTC	0x07, unbanked	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTD	0x08, unbanked	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTE	0x09, unbanked	TYPE A: xxxx xxxx TYPE B: 00xx xxxx	TYPE A: uuuu uuuu TYPE B: 00uu uuuu	TYPE A: uuuu uuuu TYPE B: 00uu uuuu
SPIRXB	0x0A, bank 0	xxxx xxxx	uuuu uuuu	uuuu uuuu
SPITXB	0x0B, bank 0	xxxx xxxx	uuuu uuuu	uuuu uuuu

Register	Address	Power-on Reset Brown-out Reset	RSTB Reset	WDT Reset
SPISTAT	0x0C, bank 0	0000 0000	0000 0000	0000 0000
SPICON	0x0D, bank 0	0000 0000	0000 0000	0000 0000
TMR1	0x0E, bank 0	0000 0000	0000 0000	0000 0000
PR1	0x0F, bank 0	1111 1111	1111 1111	1111 1111
T23CON	0x0A, bank 1	---0 0000	---0 0000	---0 0000
TMR2	0x0B, bank 1	0000 0000	0000 0000	0000 0000
PR2	0x0C, bank 1	1111 1111	1111 1111	1111 1111
TMR3	0x0D, bank 1	0000 0000	0000 0000	0000 0000
PR3	0x0E, bank 1	1111 1111	1111 1111	1111 1111
PWMCON	0x0A, bank 2	---- -000	---- -000	---- -000
PW0DCL	0x0B, bank 2	00-- ----	00-- ----	00-- ----
PW0DCH	0x0C, bank 2	0000 0000	0000 0000	0000 0000
PW1DCL	0x0D, bank 2	00-- ----	00-- ----	00-- ----
PW1DCH	0x0E, bank 2	0000 0000	0000 0000	0000 0000
RFCCON	0x0A, bank 3	000- 0000	000- 0000	000- 0000
RFCDLB	0x0B, bank 3	0000 0000	0000 0000	0000 0000
RFCDHB	0x0C, bank 3	0000 0000	0000 0000	0000 0000
CMPDX	0x0D, bank 3	0000 0000	0000 0000	0000 0000
CMPDY	0x0E, bank 3	0000 0000	0000 0000	0000 0000
CMPSTAT	0x0F, bank 3	---- 0000	---- 0000	---- 0000
INTFLAG	0x3F, unbanked	0000 0000	0000 0000	0000 0000
General Purpose Registers	0x10 ~ 0x3E	xxxx xxxx	uuuu uuuu	uuuu uuuu

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

Table 2.7: $\overline{\text{TO}}$ / $\overline{\text{PD}}$ Status after Reset or Wake-up

$\overline{\text{TO}}$	$\overline{\text{PD}}$	RESET was caused by
1	1	Power-on Reset
1	1	Brown-out reset
u	u	RSTB Reset during normal operation
1	0	RSTB Reset during SLEEP
0	1	WDT Reset during normal operation
0	0	WDT Wake-up during SLEEP

Legend: u = unchanged

Table 2.8: Events Affecting $\overline{\text{TO}}$ / $\overline{\text{PD}}$ Status Bits

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

Legend: u = unchanged

2.13 Hexadecimal Convert to Decimal (HCD)

Decimal format is another number format for FM8PE51M. When the content of the data memory has been assigned as decimal format, it is necessary to convert the results to decimal format after the execution of ALU instructions. When the decimal converting operation is processing, all of the operand data (including the contents of the data memory (RAM), accumulator (ACC), immediate data, and look-up table) should be in the decimal format, or the results of conversion will be incorrect.

Instruction DAA can convert the ACC data from hexadecimal to decimal format after any addition operation and restored to ACC.

The conversion operation is illustrated in Example 2.2.

Example 2.2: DAA CONVERSION

Code

```
#include    <8PE51M.ASH>
            …
            MOVIA     0x90        ; Set immediate data = decimal format number "90" (ACC ← 0x90)
            MOVAR     0x30        ; Load immediate data "90" to data memory address 0x30
            MOVIA     0x10        ; Set immediate data = decimal format number "10" (ACC ← 0x10)
            ADDAR     0x30,A      ; Contents of the data memory address 0x30 and ACC are binary-added
                                  ; the result loads to the ACC (ACC ← 0xA0, C ← 0)
            DAA                   ; Convert the content of ACC to decimal format, and restored to ACC
                                  ; The result in the ACC is "00" and the carry bit C is "1". This represents the
                                  ; decimal number "100"
            …
```

Instruction DAS can convert the ACC data from hexadecimal to decimal format after any subtraction operation and restored to ACC.

The conversion operation is illustrated in Example 2.3.

Example 2.3: DAS CONVERSION

Code

```
#include    <8PE51M.ASH>
            …
            MOVIA     0x10        ; Set immediate data = decimal format number "10" (ACC ← 0x10)
            MOVAR     0x30        ; Load immediate data "90" to data memory address 0x30
            MOVIA     0x20        ; Set immediate data = decimal format number "20" (ACC ← 0x20)
            SUBAR     0x30,A      ; Contents of the data memory address 0x30 and ACC are binary-subtracted
                                  ; the result loads to the ACC (ACC ← 0xF0, C ← 0)
            DAS                   ; Convert the content of ACC to decimal format, and restored to ACC
                                  ; The result in the ACC is "90" and the carry bit C is "0". This represents the
                                  ; decimal number " -10"
            …
```

2.14 Oscillator Configurations

FM8PE51M can be operated in five different oscillator modes. Users can program F_{OSC} configuration bit to select the appropriate modes:

- ERC: External Resistor/Capacitor Oscillator
- HF: High Frequency Crystal/Resonator Oscillator
- XT: Crystal/Resonator Oscillator
- LF: Low Frequency Crystal Oscillator
- IRC: Internal Resistor/Capacitor Oscillator
- ERIC: External Resistor/Internal Capacitor Oscillator

In LF, XT, or HF modes, a crystal or ceramic resonator in connected to the OSCI and OSCO pins to establish oscillation. When in LF, XT, or HF modes, the devices can have an external clock source drive the OSCI pin.
The ERC/ERIC device option offers additional cost savings for timing insensitive applications. The RC oscillator frequency is a function of the supply voltage, the resistor (R_{EXT}) and capacitor (C_{EXT}), the operating temperature, and the process parameter.
The IRC/ERIC device option offers largest cost savings for timing insensitive applications. These devices offer 4 different internal RC oscillator frequency, 8 M_{HZ}, 4 M_{HZ}, 1 M_{HZ}, and 455 K_{HZ}, which is selected by configuration bits (F_{OSC}). Or user can change the oscillator frequency with external resistor. The ERIC oscillator frequency is a function of the resistor (R_{EXT}), the operating temperature, and the process parameter.

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

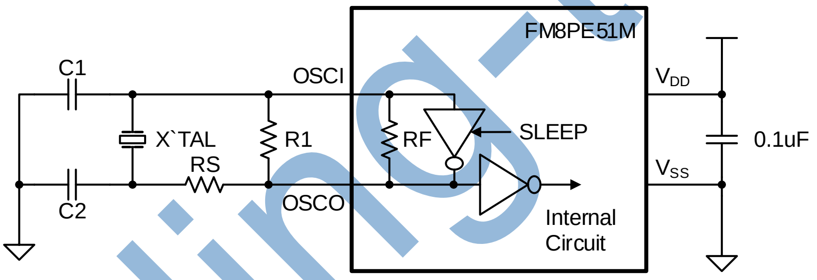


Figure 2.18: HF, XT or LF Oscillator Modes (External Clock Input Operation)

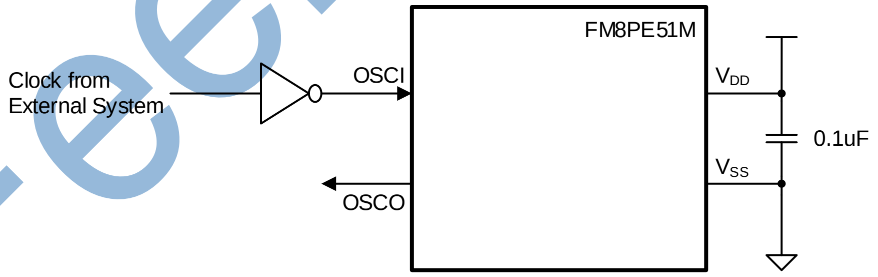


Figure 2.19: ERC Oscillator Mode (External RC Oscillator)

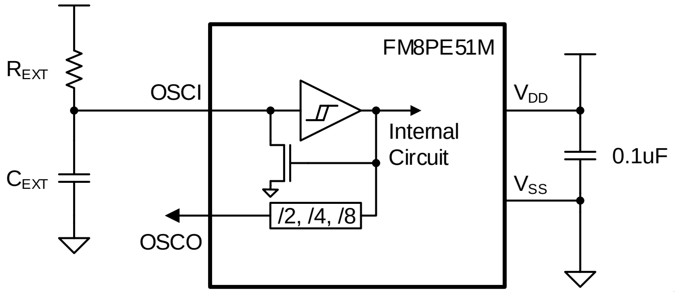
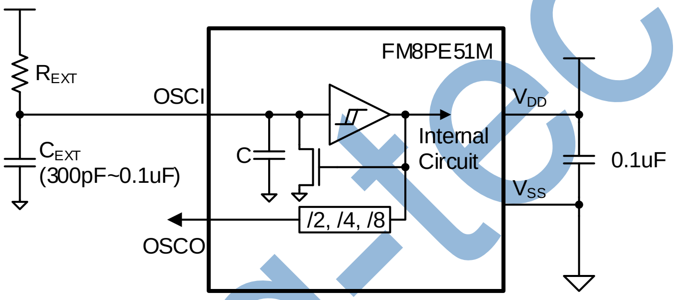


Figure 2.20: ERIC Oscillator Mode (External R, Internal C Oscillator)

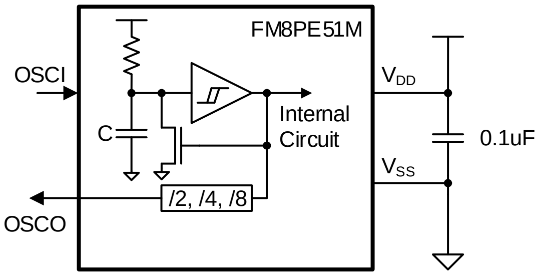


The typical oscillator frequency vs. external resistor is as following table

Frequency	R_{EXT} @ 3V	R_{EXT} @ 5V
455K_{HZ}	TBD	TBD
1M_{HZ}	TBD	TBD
4M_{HZ}	TBD	TBD
8M_{HZ}	TBD	TBD
16M_{HZ}	TBD	TBD

Note: Values are provided for design reference only.

Figure 2.21: IRC Oscillator Mode (Internal R, Internal C Oscillator)



2.15 Configuration Words

Table 2.9: Configuration Words

Name	Description
F_{OSC}	Oscillator Selection Bit → ERC mode (external R & C) (default) → HF mode → XT mode → LF mode → $4M_{HZ}$ IRC mode (internal R & C) → $8M_{HZ}$ IRC mode (internal R & C) → $1M_{HZ}$ IRC mode (internal R & C) → $455K_{HZ}$ IRC mode (internal R & C) → ERIC mode (external R & internal C)
LVDT	Low Voltage Detector Selection Bit → Disable (default) → Enable, LVDT voltage = 1.8V → Enable, LVDT voltage = 2.0V → Enable, LVDT voltage = 2.0V, controlled by SLEEP → Enable, LVDT voltage = 2.2V → Enable, LVDT voltage = 2.4V → Enable, LVDT voltage = 2.6V → Enable, LVDT voltage = 3.6V
SUT	Reset delay time selection bits → 18mS (default) → 4.5mS → 288mS → 72mS
RSTBIN	IOC0/PWM1/RSTB Pin Selection Bit → IOC0/PWM1 pin is selected → RSTB pin is selected
WDTEN	Watchdog Timer Enable Bit → WDT enabled (default) → WDT disabled
PROTECT	Code Protection Bit → OTP code protection off (default) → OTP code protection on
OSCD	Instruction Period Selection Bits → Four oscillator periods (default) → Two oscillator periods → Eight oscillator periods
PMOD	Power Mode Selection Bit → Non-power saving (default) → Power saving
RDPORT	Read Port Control Bit for Output Pins → From registers (default) → From pins
WUTRIG	Wake-up Trigger Control Bit → Falling edge trigger (default) → Low level trigger
SDOS	SDO Output Status Control Bit while SSB = 1 for Slave Mode with SSB Control → Disable, the SDO will be floating (default) → Enable, the SDO will keep last bit status

Name	Description
CKOUT	IOE7/CKOUT Selection Bit → Enable IOE7 (default) → Enable CKOUT, same phase of OSCI → Enable CKOUT, opposite phase of OSCI
DEL	SPI Input Delay Time Selection Bit → 0nS (default) → 50nS → 100nS
TYPE	Type Selection Bit → A type (44-pin) is selected (default) → B type (40-pin) is selected
RCT	18mS/140uS Reset/Wake-up time Selection for RC mode → 18mS (default) → 140uS

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	1/2/3(1)	-
BTRSS R, bit	Test bit in R, Skip if Set	Skip if R<b> = 1	1/2/3(1)	-
NOP	No Operation	No operation	1	-
CLRWDT	Clear Watchdog Timer	0x00 → WDT, 0x00 → WDT pre-scaler	1	$\overline{TO}$,$\overline{PD}$
SLEEP	Go into power-down mode	0x00 → WDT, 0x00 → WDT pre-scaler	1	$\overline{TO}$,$\overline{PD}$
OPTION	Load OPTION register	ACC → OPTION	1	-
OPTIONR	Read OPTION register	OPTION → ACC	1	-
DAA	Adjust ACC's data format from HEX to DEC after any addition operation	ACC(hex) → ACC (dec)	1	C
DAS	Adjust ACC's data format from HEX to DEC after any subtraction operation	ACC(hex) → ACC (dec)	1	-
ENI	Set GIE bit	1 → GIE	1	-
DISI	Clear GIE bit	0 → GIE	1	-
RETURN	Return from subroutine	Top of Stack → PC	2	-
RETFIE	Return from interrupt, set GIE bit	Top of Stack → PC, 1 → GIE	2	-
INT	S/W interrupt	PC + 1 → Top of Stack 0x002 → PC	2	-
IOST R	Load IOST register	ACC → IOST register	1	-
IOSTR R	Read IOST register	IOST register → ACC	1	-
TBL	Table look-up	PC<7:0> + ACC → PC<7:0> PC<9:8> unchanged PCHBUF<3:2> → PC<11:10>	1	C, DC, Z
CLRA	Clear ACC	0x00 → ACC	1	Z
CLRR R	Clear R	0x00 → R	1	Z
MOVAR R	Move ACC to R	ACC → R	1	-
MOVR R, d	Move R	R → dest	1	Z
DECR R, d	Decrement R	R - 1 → dest	1	Z
DECRSZ R, d	Decrement R, Skip if 0	R - 1 → dest, Skip if result = 0	1/2/3(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	1/2/3(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

Mnemonic, Operands	Description	Operation	Cycles	Status Affected
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
RETIA I	Return, place Immediate in ACC	I → ACC, Top of Stack → PC	2	-
BANK I	Move Immediate to memory bank bits	I → RP<1:0>	1	-
PAGE I	Move Immediate to program page bits	I → PG<1:0>	1	-
CALL I	Call subroutine	PC + 1 → Top of Stack, I → PC<9:0> PG<1:0> → PC<11:10>	2	-
GOTO I	Unconditional branch	I → PC<9:0> PG<1:0> → PC<11:10>	2	-
FCALL I	Call subroutine	PC + 1 → Top of Stack, I → PC<11:0> I<11:10> → PG<1:0>	3	-
FGOTO I	Unconditional branch	I → PC<11:0> I<11:10> → PG<1:0>	3	-

Note: 1. 2 cycles for skip, else 1 cycle. (3 cycles if skip and followed by a 2-word instruction FCALL/FGOTO)
2. bit: Bit address within an 8-bit register R
R: Register address (0x00 to 0x3F)
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
RP: RAM Page(Bank) Select Bits
PG: Program Memory Page Select Bits
WDT: Watchdog Timer Counter
GIE: Global interrupt enable bit
$\overline{TO}$: Time-out bit
$\overline{PD}$: Power-down bit
C: Carry bit
DC: Half carry bit
Z: Zero bit

ADCAR	**Add ACC and R with Carry**
Syntax:	ADCAR R, d
Operands:	0x000≤R≤0x03F 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:	0x000≤R≤0x03F d∈[0,1]
Operation:	ACC + R → dest
Status Affected:	C, DC, Z
Description:	Add the contents of the ACC register and register ‘R’. If ‘d’ is ‘0’, the result is stored in the ACC register. If ‘d’ is ‘1’, the result is stored back in register ‘R’.
Cycles:	1

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

ANDAR	**AND ACC and R**
Syntax:	ANDAR R, d
Operands:	0x000≤R≤0x03F d∈[0,1]
Operation:	ACC and R → dest
Status Affected:	Z
Description:	The contents of the ACC register are AND’ed with register ‘R’. If ‘d’ is ‘0’, the result is stored in the ACC register. If ‘d’ is ‘1’, the result is stored back in register ‘R’.
Cycles:	1

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

BANK	**Move Immediate to memory bank bits**
Syntax:	BANK I
Operands:	0x0≤I≤0x3
Operation:	I → RP<1:0>
Status Affected:	None
Description:	The memory bank bits are loaded with the 2-bit immediate ‘I’.
Cycles:	1

BCR	**Clear Bit in R**
Syntax:	BCR R, b
Operands:	0x000≤R≤0x03F 0x0≤b≤0x7
Operation:	0 → R<b>
Status Affected:	None
Description:	Clear bit ‘b’ in register ‘R’.
Cycles:	1

BSR	**Set Bit in R**
Syntax:	BSR R, b
Operands:	0x000≤R≤0x03F 0x0≤b≤0x7
Operation:	1 → R<b>
Status Affected:	None
Description:	Set bit ‘b’ in register ‘R’.
Cycles:	1

BTRSC	**Test Bit in R, Skip if Clear**
Syntax:	BTRSC R, b
Operands:	0x000≤R≤0x03F 0x0≤b≤0x7
Operation:	Skip if R<b> = 0
Status Affected:	None
Description:	If bit ‘b’ in register ‘R’ is ‘0’ then the next instruction is skipped. If bit ‘b’ is ‘0’, then 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

BTRSS	**Test Bit in R, Skip if Set**
Syntax:	BTRSS R, b
Operands:	0x000≤R≤0x03F 0x0≤b≤0x7
Operation:	Skip if R<b> = 1
Status Affected:	None
Description:	If bit ‘b’ in register ‘R’ is ‘1’ then the next instruction is skipped. If bit ‘b’ is ‘1’, then the next instruction fetched during the current instruction execution is discarded, and a NOP is executed instead, making this a 2-cycle instruction.
Cycles:	1/2

CALL	**Subroutine Call**
Syntax:	CALL I
Operands:	0x0000≤I≤0x03FF
Operation:	PC + 1 → Top of Stack, I → PC<9:0> PG<1:0> → PC<11:10>
Status Affected:	None
Description:	Subroutine call. First, return address (PC+1) is pushed onto the stack. The 12-bit immediate address is loaded into PC bits <11:0>.
Cycles:	2

CLRA	**Clear ACC**
Syntax:	CLRA
Operands:	None
Operation:	0x00 → ACC; 1 → Z
Status Affected:	Z
Description:	The ACC register is cleared and the 'Z' bit is set.
Cycles:	1

CLRR	**Clear R**
Syntax:	CLRR R
Operands:	0x000≤R≤0x03F
Operation:	0x00 → R; 1 → Z
Status Affected:	Z
Description:	The contents of register 'R' are cleared and the 'Z' bit is set.
Cycles:	1

CLRWDT	**Clear Watchdog Timer**
Syntax:	CLRWDT
Operands:	None

Operation:	0x00 → WDT;
	0x00 → WDT pre-scaler (if assigned);
	1 → $\overline{TO}$;
	1 → $\overline{PD}$

Status Affected:	$\overline{TO}$, $\overline{PD}$
Description:	The CLRWDT instruction resets the WDT. It also resets the pre-scaler, if the pre-scaler is assigned to the WDT and not Timer0. Status bits $\overline{TO}$ and $\overline{PD}$ are set.
Cycles:	1

COMR	**Complement R**
Syntax:	COMR R, d
Operands:	0x000≤R≤0x03F 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

DAA	Adjust ACC’s data format from HEX to DEC
Syntax:	DAA
Operands:	None
Operation:	ACC(hex) → ACC(dec)
Status Affected:	C
Description:	Convert the ACC data from hexadecimal to decimal format after any addition operation and restored to ACC.
Cycles:	1

DAS	Adjust ACC’s data format from HEX to DEC
Syntax:	DAS
Operands:	None
Operation:	ACC(hex) → ACC(dec)
Status Affected:	None
Description:	Convert the ACC data from hexadecimal to decimal format after any subtraction operation and restored to ACC.
Cycles:	1

DECR	Decrement R
Syntax:	DECR R, d
Operands:	0x000≤R≤0x03F 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:	0x000≤R≤0x03F 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 2-cycle instruction.
Cycles:	1/2

DISI	Clear GIE bit
Syntax:	DISI
Operands:	None
Operation:	0 → GIE
Status Affected:	None
Description:	Disable interrupt
Cycles:	1

ENI	**Set GIE bit**
Syntax:	ENI
Operands:	None
Operation:	1 → GIE
Status Affected:	None
Description:	Enable interrupt
Cycles:	1

FCALL	**Subroutine Call**
Syntax:	FCALL I
Operands:	0x0000≤I≤0x0FFF
Operation:	PC +1 → Top of Stack I → PC<11:0> I<11:10> → PG<1:0>
Status Affected:	None
Description:	Subroutine call. First, return address (PC+1) is pushed onto the stack. The 12-bit immediate address is loaded into PC bits <11:0>. FCALL is a two-word (three-cycle) instruction..
Cycles:	3

FGOTO	**Unconditional Branch**
Syntax:	FGOTO I
Operands:	0x0000≤I≤0x0FFF
Operation:	I → PC<11:0> I<11:10> → PG<1:0>
Status Affected:	None
Description:	FGOTO is an unconditional branch. The 12-bit immediate value is loaded into PC bits <11:0>. FGOTO is a two-word (three-cycle) instruction.
Cycles:	3

GOTO	**Unconditional Branch**
Syntax:	GOTO I
Operands:	0x000≤I≤0x3FF
Operation:	I → PC<9:0> PG<1:0> → PC<11:10>
Status Affected:	None
Description:	GOTO is an unconditional branch. The 12-bit immediate value is loaded into PC bits <11:0>.
Cycles:	2

INCR	**Increment R**
Syntax:	INCR R, d
Operands:	0x000≤R≤0x03F 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: 0x000≤R≤0x03F
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 ‘1’, 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 2-cycle instruction.

Cycles: 1/2

INT S/W Interrupt

Syntax: INT

Operands: None

Operation: PC + 1 → Top of Stack,
0x002 → PC

Status Affected: None

Description: Interrupt subroutine call. First, return address (PC+1) is pushed onto the stack. The address 0x002 is loaded into PC bits <9:0>.

Cycles: 2

IORAR OR ACC with R

Syntax: IORAR R, d

Operands: 0x000≤R≤0x03F
d∈[0,1]

Operation: ACC or R → dest

Status Affected: Z

Description: Inclusive OR the ACC register with register ‘R’. If ‘d’ is ‘0’, the result is placed in the ACC register. If ‘d’ is ‘1’, the result is placed back in register ‘R’.

Cycles: 1

IORIA OR Immediate with ACC

Syntax: IORIA I

Operands: 0x00≤I≤0xFF

Operation: ACC or I → ACC

Status Affected: Z

Description: The contents of the ACC register are OR’ed with the 8-bit immediate ‘I’. The result is placed in the ACC register.

Cycles: 1

IOST Load IOST Register

Syntax: IOST R

Operands: R = 0x05~ 0x09 or 0x0C ~ 0x0F

Operation: ACC → IOST register R

Status Affected: None

Description: IOST register ‘R’ (R= 0x05~ 0x09 or 0x0C ~ 0x0F) is loaded with the contents of the ACC register.

Cycles: 1

IOSTR	**Read IOST Register**
Syntax:	IOST R
Operands:	R = 0x05~ 0x09 or 0x0C ~ 0x0F
Operation:	IOST register R → ACC
Status Affected:	None
Description:	The ACC register is loaded with the contents of IOST register ‘R’ (0x05~ 0x09 or 0x0C ~ 0x0F).
Cycles:	1

MOVAR	**Move ACC to R**
Syntax:	MOVAR R
Operands:	0x000≤R≤0x03F
Operation:	ACC → R
Status Affected:	None
Description:	Move data from the ACC register to register ‘R’.
Cycles:	1

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

MOVR	**Move R**
Syntax:	MOVR R, d
Operands:	0x000≤R≤0x03F 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

OPTION	**Load OPTION Register**
Syntax:	OPTION
Operands:	None
Operation:	ACC → OPTION
Status Affected:	None
Description:	The content of the ACC register is loaded into the OPTION register.
Cycles:	1

OPTIONR	**Read OPTION Register**
Syntax:	OPTION
Operands:	None
Operation:	OPTION → ACC
Status Affected:	None
Description:	The content of the OPTION register is loaded into the ACC register.
Cycles:	1

PAGE	**Move Immediate to program page bits**
Syntax:	PAGE I
Operands:	0x0≤I≤0x3
Operation:	I → PG<1:0>
Status Affected:	None
Description:	The program page bits are loaded with the 2-bit immediate 'I'.
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'.
Cycles:	2

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

RLR	Rotate Left R through Carry
Syntax:	RLR R, d
Operands:	0x000≤R≤0x03F 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:	0x000≤R≤0x03F 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:	0x000≤R≤0x03F 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:	0x000≤R≤0x03F d∈[0,1]
Operation:	R - ACC → dest
Status Affected:	C, DC, Z
Description:	Subtract (2's complement method) the ACC register from register 'R'. If 'd' is '0', the result is stored in the ACC register. If 'd' is '1', the result is stored back in register 'R'.
Cycles:	1

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

SWAPR	**Swap nibbles in R**
Syntax:	SWAPR R, d
Operands:	0x000≤R≤0x03F 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

TBL	**Table Look-up**
Syntax:	TBL
Operands:	None
Operation:	PC<7:0> + ACC → PC<7:0> PC<9:8> unchanged PG<1:0> → PC<11:10>
Status Affected:	C, DC, Z
Description:	Operate with RETIA to look-up table.
Cycles:	1

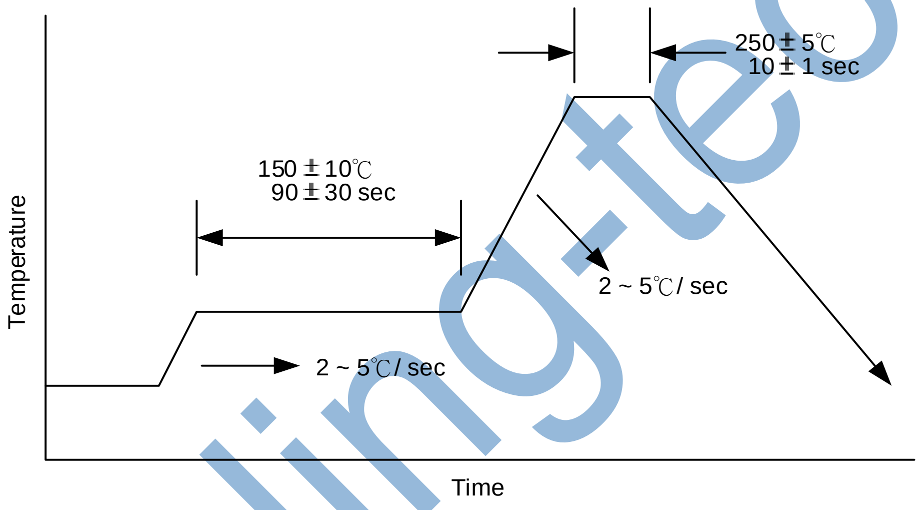
XORAR	**Exclusive OR ACC with R**
Syntax:	XORAR R, d
Operands:	0x000≤R≤0x03F d∈[0,1]
Operation:	ACC xor R → dest
Status Affected:	Z
Description:	Exclusive OR the contents of the ACC register with register 'R'. If 'd' is '0', the result is stored in the ACC register. If 'd' is '1', the result is stored back in register 'R'.
Cycles:	1

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

4.0 ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
	Ambient Operating Temperature	-	0	-	70	°C
	Store Temperature	-	-65	-	150	°C
V_{DD}	DC Supply Voltage	-	0	-	6	V
	Input Voltage with respect to Ground	-	-0.3	-	V_{DD}+0.3	V
	ESD Susceptibility	HBM (Human Body Mode)	-	TBD	-	KV
		MM (Machine Mode)	-	TBD	-	V
	Lead Temperature	Soldering, 10 Sec	-	-	250	°C

4.1 PACKAGE IR Re-flow Soldering Curve



5.0 OPERATING CONDITIONS

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
V_{DD}	DC Supply Voltage	-	2.3	-	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_{HF}	HF Oscillation range	3V	HF mode		TBD		M_{HZ}
		5V			TBD		
F_{XT}	XT Oscillation range	3V	XT mode		TBD		M_{HZ}
		5V			TBD		
F_{LF}	LF oscillation range	3V	LF mode		TBD		K_{HZ}
		5V			TBD		
F_{ERC}	ERC Oscillation range	3V	ERC mode	DC	-	TBD	M_{HZ}
		5V		DC	-	TBD	
F_{ERIC}	ERIC Oscillation range	3V	ERIC mode	DC	-	TBD	M_{HZ}
		5V		DC	-	TBD	
F_{IRC}	Internal RC Oscillation range	3V	455K_{HZ} IRC mode	-3%	455	+3%	M_{HZ}
		5V		-3%	455	+3%	
		3V	1M_{HZ} IRC mode	-3%	1	+3%	
		5V		-3%	1	+3%	
		3V	4M_{HZ} IRC mode	-3%	4	+3%	
		5V		-3%	4	+3%	
		3V	8M_{HZ} IRC mode	-3%	8	+3%	
		5V		-3%	8	+3%	
T_{WDT}	WDT period time	3V	Pre-scaler rate=1:1	-	TBD	-	mS
		4V		-	TBD	-	
		5V		-	TBD	-	

Note:
1. In the ERIC mode, to maintain the accuracy of the internal RC oscillator frequency, a 300pF ~ 0.1uF decoupling capacitor should be connected between OSCI and V_{SS} and located as close to the device as possible.
2. 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 & LVDT are disable, I/O output float.

Symbol	Description	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{IH1}	Input high voltage, I/O Ports	3V	With Schmitt-trigger	-	1.28	V_{DD}	V
		5V		1.9	-	V_{DD}	
	Input high voltage, RSTB, T0CKI Pins	3V	With Schmitt-trigger	-	1.79	V_{DD}	
		5V		-	3.31	V_{DD}	
V_{IH2}	Input high voltage, I/O Ports	3V	Without Schmitt-trigger	-	1.19	V_{DD}	V
		5V		-	1.58	V_{DD}	
	Input high voltage, RSTB, T0CKI Pins	3V	Without Schmitt-trigger	-	1.79	V_{DD}	
		5V		-	3.31	V_{DD}	
V_{IL1}	Input low voltage, I/O Ports	3V	With Schmitt-trigger	V_{SS}	1.0	-	V
		5V		V_{SS}	-	1.0	
	Input low voltage, RSTB, T0CKI Pins	3V	With Schmitt-trigger	V_{SS}	1.15	-	
		5V		V_{SS}	1.49	-	

Symbol	Description	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{IL2}	Input low voltage, I/O Ports	3V	Without Schmitt-trigger	V_{SS}	1.09	-	V
		5V		V_{SS}	1.5	-	
	Input low voltage, RSTB, T0CKI Pins	3V	Without Schmitt-trigger	V_{SS}	1.15	-	
		5V		V_{SS}	1.5	-	
V_{LVDT}	LVDT voltage	-	LVDT=3.6V	3.06	3.6	4.14	V
		-	LVDT=2.6V	2.21	2.6	2.99	
		-	LVDT=2.4V	2.04	2.4	2.76	
		-	LVDT=2.2V	1.87	2.2	2.53	
		-	LVDT=2.0V	1.7	2.0	2.3	
		-	LVDT=1.8V	1.6	1.8	2.07	
I_{OH}	I/O Ports Drive current (without A type IOA5)	3V	V_{OH}=0.9V_{DD}	-	1.74	-	mA
		5V		1.5	4.54	-	
	A type IOB0, B type IOA4 IR-out mode Drive current	3V	V_{OH}=0.9V_{DD}	-	3.38	-	
		5V		-	8.55	-	
I_{OL}	I/O Ports Sink current	3V	V_{OL}=0.1V_{DD}	-	9.69	-	mA
		5V		10	22.74	-	
	A type IOB0, B type IOA4 IR-out mode Sink current	3V	V_{OL}=0.1V_{DD}	-	18.67	-	
		5V		-	42.74	-	
I_{PH}	I/O Ports Pull-high current	3V	Input pin at V_{SS}	-	19.46	-	uA
		5V		55	65.03	85	
I_{PL}	I/O Ports Pull-low current	3V	Input pin at V_{DD}	-	12.64	-	uA
		5V		30	41.33	60	
I_{ROC}	IOC0 & IOC1 ROC mode Pull-high current	3V	Input pin at V_{SS}	-	0.31	-	uA
		5V		-	1.05	-	
I_{LVDT}	LVDT current	5V	LVDT=3.6V	-	1.09	-	uA
		3V	LVDT=2.6V	-	0.4	-	
		5V		-	1.4	-	
		3V	LVDT=2.4V	-	0.5	-	
		5V		-	1.5	-	
		3V	LVDT=2.2V	-	0.5	-	
		5V		-	1.6	-	
		3V	LVDT=2.0V	-	0.6	-	
		5V		-	1.7	-	
		3V	LVDT=1.8V	-	0.6	-	
		5V		-	1.9	-	
I_{WDT}	WDT current	3V	Sleep mode, Pre-scaler rate=1:256	-	0.5	-	uA
		5V		-	3.8	8	
I_{SB}	Sleep mode (Power down) current	3V	-	-	<1	-	uA
		5V		-	<1	1	
I_{DD1}	HF Operating current	3V	Freq=20M_{HZ}, 2T	-	1.78	-	mA
		5V		-	4.1	-	
I_{DD2}	XT Operating current	3V	Freq=16M_{HZ}, 2T	-	0.96	-	mA
		5V		-	2.27	-	
		5V	Freq=20M_{HZ}, 2T	-	3.04	-	
I_{DD3}	LF Operating current	3V	Freq=32K_{HZ}, 2T	-	45.82	-	uA
		5V		-	129.83	-	
I_{DD4}	ERC Operating current R_{EXT}=3.3K, C_{EXT}=3pF	3V	Freq=13.05M_{HZ}, 2T	-	1.8	-	mA
		5V	Freq=15.31M_{HZ}, 2T	-	3.5	-	
I_{DD5}	ERIC Operating current	3V	Freq=8M_{HZ}, 2T	-	0.79	-	mA
		5V		-	1.51	-	

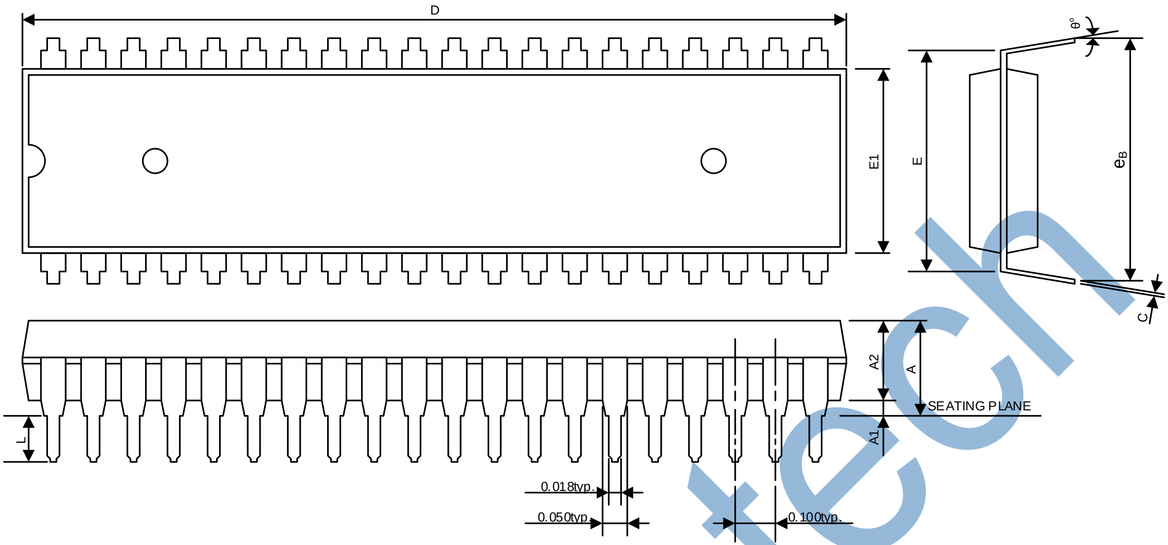
Symbol	Description	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
		5V	Freq=16M_{HZ}, 2T	-	2.87	-	
I_{DD6}	Operating current	3V	IRC 8M_{HZ}, 2T	-	0.79	-	mA
		5V		-	1.52	-	
I_{DD7}	Operating current	3V	IRC 4M_{HZ}, 2T	-	0.42	-	mA
		5V		-	0.8	-	
I_{DD8}	Operating current	3V	IRC 1M_{HZ}, 2T	-	141	-	uA
		5V		-	297.2	-	
I_{DD9}	Operating current	3V	IRC 455K_{HZ}, 2T	-	88.6	-	uA
		5V		-	202.8	-	

6.3 ELECTRICAL CHARACTERISTICS Charts of FM8PE51M

To be define…

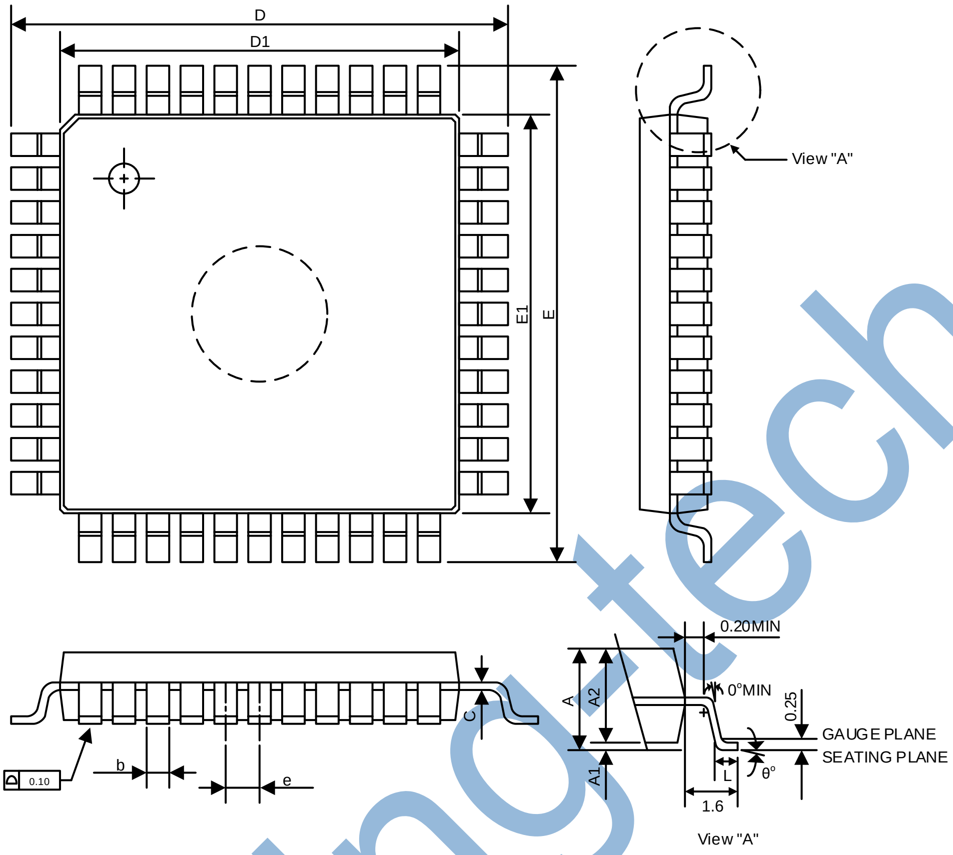
7.0 PACKAGE DIMENSION

7.1 40-PIN PDIP 600mil



Symbols	Dimension in Inches		
	Min	Nom	Max
A	-	-	0.190
A1	0.015	-	-
A2	0.150	0.155	0.160
C	0.008	-	0.015
D	2.055	2.060	2.070
E	0.600 BSC.		
E1	0.540	0.545	0.550
L	0.120	0.130	0.140
eB	0.630	0.650	0.670
θ°	0°	7°	15°

7.2 44-PIN QFP



Symbols	Dimension in MM		
	Min	Nom	Max
A	-	-	2.70
A1	0.25	-	0.50
A2	1.80	2.00	2.20
b	0.25	0.30	0.35
D	13.00	13.20	13.40
D1	9.9	10.00	10.10
E	13.00	13.20	13.40
E1	9.9	10.00	10.10
L	0.73	0.88	0.93
e	0.80 BSC.		
θ°	0	-	7
C	0.1	0.15	0.2

8.0 ORDERING INFORMATION

OTP Type MCU	Package Type	Pin Count	Package Size	MOQ	MSL	Sample Stock
FM8PE51MP	PDIP	40	600mil	3,000EA/Tube	3	Call sales
FM8PE51MF	QFP	44	10mm x 10mm		3	Call sales