EPROM/ROM-Based 8-Bit Microcontroller Series

Devices Included in this Data Sheet:

- FM8P55E/57E : EPROM devices
- FM8P55/57 : Mask ROM devices

FEATURES

- Only 47 single word instructions
- All instructions are single cycle except for program branches which are two-cycle
- 13-bit wide instructions
- All ROM/EPROM area GOTO/FGOTO instruction
- All ROM/EPROM area subroutine CALL/FCALL instruction
- 8-bit wide data path
- 5-level deep hardware stack
- Operating speed: DC-20 MHz clock input
  DC-100 ns instruction cycle

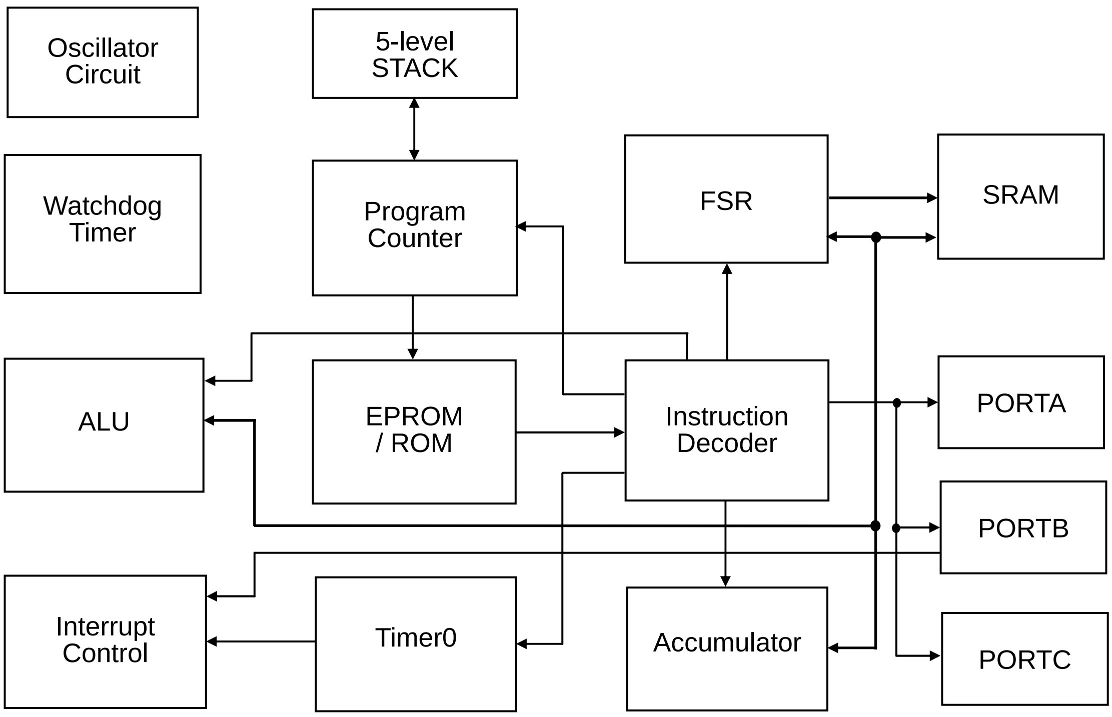
Device	Pins #	I/O #	EPROM/ROM (Byte)	RAM (Byte)
FM8P55/55E	28	20	512	48
FM8P57/57E	28	20	2K	96

- Direct, indirect addressing modes for data accessing
- 8-bit real time clock/counter (Timer0) with 8-bit programmable prescaler
- 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
- Three I/O ports IOA, IOB and IOC with independent direction control
- 16 soft-ware control pull-high pins: Port B/Port C
- 8 soft-ware control pull-down pins:IOA0~A3/IOB0~B3
- 2 soft-ware control open-drain pins: IOC6/IOC7
- One internal interrupt source: Timer0 overflow; One external interrupt source: INT pin
- Wake-up from SLEEP by Port B/IOC4/IOC5 input falling
- Power saving SLEEP mode
- Programmable Code Protection
- Selectable oscillator options:
  - ERC: External Resistor/Capacitor Oscillator
  - XT: Crystal/Resonator Oscillator
  - HF: High Frequency Crystal/Resonator Oscillator
  - LF: Low Frequency Crystal Oscillator
- Wide-operating voltage range:
  - EPROM : 2.3V to 5.5V
  - ROM : 2.3V to 5.5V

GENERAL DESCRIPTION

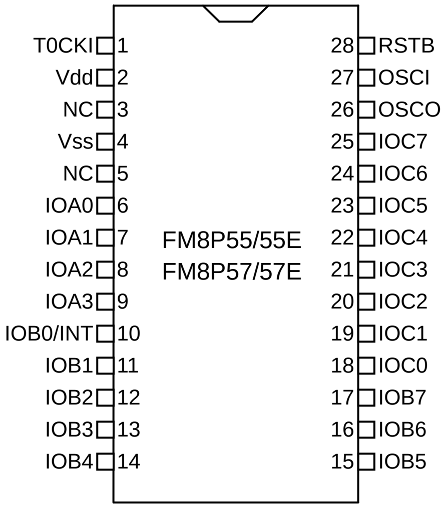
The FM8P55/57 series is a family of low-cost, high speed, high noise immunity, EPROM/ROM-based 8-bit CMOS microcontrollers. It employs a RISC architecture with only 47 instructions. All instructions are single cycle except for program branches which take two cycles. The easy to use and easy to remember instruction set reduces development time significantly.
The FM8P55/57 series consists of Power-on Reset (POR), Brown-out Reset (BOR), Power-up Reset Timer (PWRT), Oscillator Start-up Timer(OST), Watchdog Timer, EPROM/ROM, SRAM, tri-state I/O port, I/O pull-high/open-drain/pull-down control, Power saving SLEEP mode, real time programmable clock/counter, Interrupt, Wake-up from SLEEP mode, and Code Protection for EPROM products. There are four oscillator configurations to choose from, including the power-saving LP (Low Power) oscillator and cost saving RC oscillator.
The FM8P55/55E address 512×13 of program memory, and the FM8P57/57E address 2K×13 of program memory.
The FM8P55/57 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

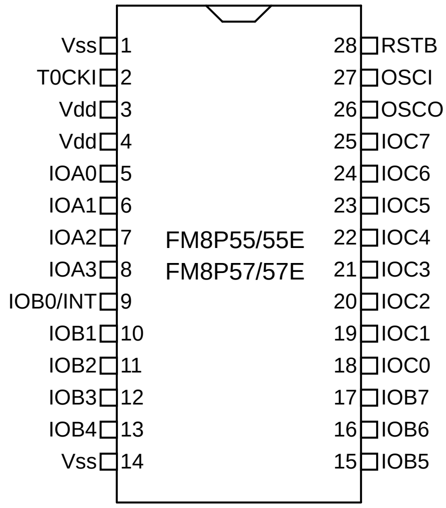


PIN CONNECTION

PDIP, SOP



SSOP



PIN DESCRIPTIONS

Name	I/O	Description
IOA0 ~ IOA3	I/O	IOA0 ~ IOA3 as bi-direction I/O port
IOB0/INT	I/O	Bi-direction I/O pin with system wake-up function / External interrupt input
IOB1 ~ IOB7	I/O	Bi-direction I/O port with system wake-up function
IOC0 ~ IOC7	I/O	Bi-direction I/O port
T0CKI	I	Clock input to Timer0. Must be tied to Vss or Vdd, if not in use, to reduce current consumption
RSTB	I	System clear (RESET) input. This pin is an active low RESET to the device.
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 with the instruction cycle rate
Vdd	-	Positive supply
Vss	-	Ground

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

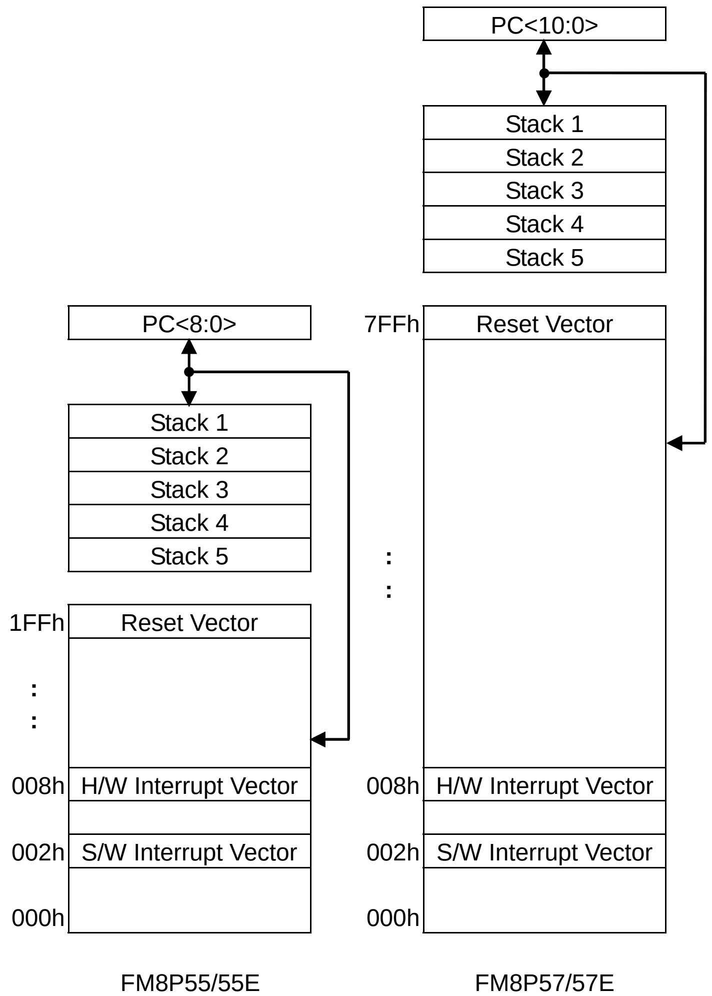
1.0 MEMORY ORGANIZATION

FM8P55/57 memory is organized into program memory and data memory.

1.1 Program Memory Organization

The FM8P55/55E have a 9-bit Program Counter (PC) capable of addressing a 512×13 program memory space.
The FM8P57/57E have an 11-bit Program Counter capable of addressing a 2K×13 program memory space.
The RESET vector for the FM8P55/55E is at 1FFh. The RESET vector for the FM8P57/57E is at 7FFh.
The H/W interrupt vector is at 008h. And the S/W interrupt vector is at 002h.
FM8P57/57E 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 2K program memory address range for FM8P57/57E, there is another one bit to specify the program memory page. This paging bit comes from the PCHBUF<2> bit. When doing a CALL or GOTO instruction, the user must ensure that page bit PCHBUF<2> are programmed so that the desired program memory page is addressed. When one of the return instructions is executed, the entire 11-bit PC is POPed from the stack. Therefore, manipulation of the PCHBUF <2> 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 to 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 FM8P57/57E, 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 FM8P57/57E Series

Address \ FSR<7:6>	0 0 Bank 0	0 1 Bank 1	1 0 Bank 2	1 1 Bank 3
00h	INDF	Memory back to address in Bank 0		
01h	TMR0			
02h	PCL			
03h	STATUS			
04h	FSR			
05h	PORTA			
06h	PORTB			
07h	PORTC			
08h	PCON			
09h	WUCON			
0Ah	PCHBUF			
0Bh	PDCON			
0Ch	BPHCON			
0Dh	CPHCON			
0Eh	INTEN			
0Fh	INTFLAG			
10h | 2Fh	General Purpose Registers			
30h | 3Fh	General Purpose Registers	General Purpose Registers	General Purpose Registers	General Purpose Registers

Address	Register
N/A	OPTION
05h	IOSTA
06h	IOSTB
07h	IOSTC

TABLE 1.2: Registers File Map for FM8P55/55E Series

Address	Description
00h	INDF
01h	TMR0
02h	PCL
03h	STATUS
04h	FSR
05h	PORTA
06h	PORTB
07h	PORTC
08h	PCON
09h	WUCON
0Ah	PCHBUF
0Bh	PDCON
0Ch	BPHCON
0Dh	CPHCON
0Eh	INTEN
0Fh	INTFLAG
10h ~ 3Fh	General Purpose Registers

N/A	OPTION
05h	IOSTA
06h	IOSTB
07h	IOSTC

TABLE 1.3: The Registers Controlled by OPTION or IOST Instructions

Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
N/A (w)	OPTION	-	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
05h (w)	IOSTA	Port A I/O Control Register							
06h (w)	IOSTB	Port B I/O Control Register							
07h (w)	IOSTC	Port C I/O Control Register							

TABLE 1.4: Operational Registers Map

Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
00h (r/w)	INDF	Uses contents of FSR to address data memory (not a physical register)							
01h (r/w)	TMR0	8-bit real-time clock/counter							
02h (r/w)	PCL	Low order 8 bits of PC							
03h (r/w)	STATUS	GP2	GP1	GP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
04h (r/w)	FSR	RP1 (2)	RP0 (2)	Indirect data memory address pointer					
05h (r/w)	PORTA	-	-	-	-	IOA3	IOA2	IOA1	IOA0
06h (r/w)	PORTB	IOB7	IOB6	IOB5	IOB4	IOB3	IOB2	IOB1	IOB0
07h (r/w)	PORTC	IOC7	IOC6	IOC5	IOC4	IOC3	IOC2	IOC1	IOC0
08h (r/w)	PCON	WDTE	EIS	LVDTE	ROC	-	-	ODC67	/WUC45
09h (r/w)	WUCON	/WUB7	/WUB6	/WUB5	/WUB4	/WUB3	/WUB2	/WUB1	/WUB0
0Ah (r/w)	PCHBUF (1)	-	-	-	-	-	Upper 3 bits Buffer of PC		
0Bh (r/w)	PDCON	/PDB3	/PDB2	/PDB1	/PDB0	/PDA3	/PDA2	/PDA1	/PDA0
0Ch (r/w)	BPHCON	/PHB7	/PHB6	/PHB5	/PHB4	/PHB3	/PHB2	/PHB1	/PHB0
0Dh (r/w)	CPHCON	/PHC7	/PHC6	/PHC5	/PHC4	/PHC3	/PHC2	/PHC1	/PHC0
0Eh (r/w)	INTEN	GIE	-	-	-	-	INTIE	-	T0IE
0Fh (r/w)	INTFLAG	-	-	-	-	-	INTIF	-	T0IF

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

Note 1 : There is only 1 bit in FM8P55/55E. And there are 3 bits in FM8P57/57E.

2 : For FM8P55/55E, these bits are not used, read as ‘1’

2.0 FUNCTIONAL DESCRIPTIONS

2.1 Operational Registers

2.1.1 INDF (Indirect Addressing Register)

Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
00h (r/w)	INDF	Uses contents of FSR to address data memory (not a physical register)							

The INDF Register is not a physical register. Any instruction accessing the INDF register can actually access the register pointed by FSR Register. Reading the INDF register itself indirectly (FSR="0") will read 00h. 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: 00h ~ 3Fh).
In FM8P57/57E, 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.

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

EXAMPLE 2.1: INDIRECT ADDRESSING

- Register file 38 contains the value 10h
- Register file 39 contains the value 0Ah
- Load the value 38 into the FSR Register
- A read of the INDF Register will return the value of 10h
- Increment the value of the FSR Register by one (@FSR=39h)
- A read of the INDR register now will return the value of 0Ah.

FIGURE 2.1: Direct/Indirect Addressing for FM8P55/55E

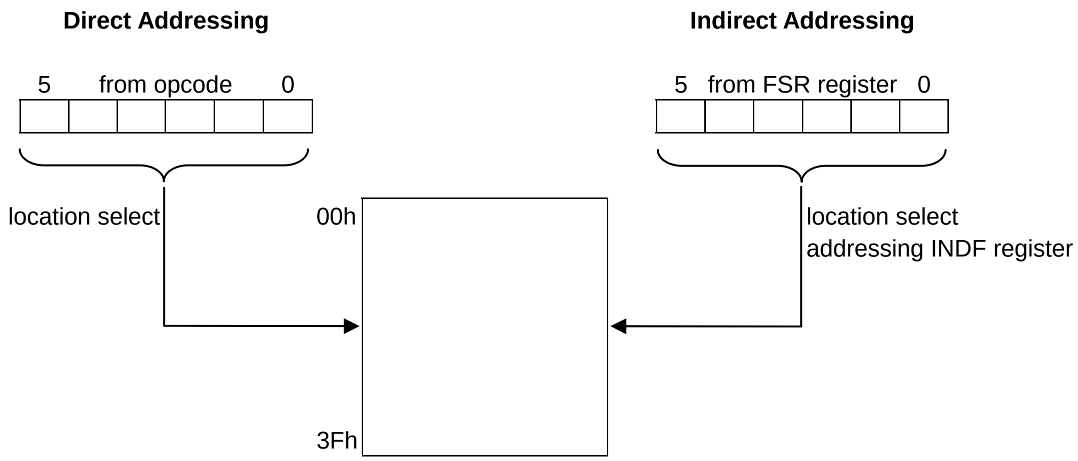
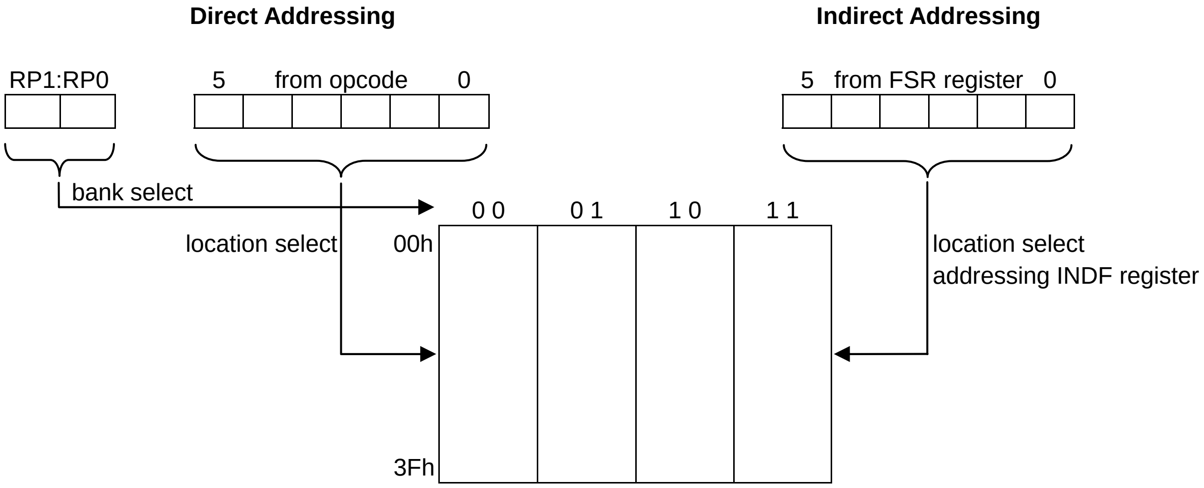


FIGURE 2.2: Direct/Indirect Addressing for FM8P57/57E



2.1.2 TMR0 (Time Clock/Counter register)

Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
01h (r/w)	TMR0	8-bit real-time clock/counter							

The Timer0 is a 8-bit timer/counter. The clock source of Timer0 can come from the instruction cycle clock or by an external clock source (T0CKI pin) defined by T0CS bit (OPTION<5>). If T0CKI pin is selected, the Timer0 is increased by T0CKI signal rising/falling edge (selected by T0SE bit (OPTION<4>)).
The prescaler is assigned to Timer0 by clearing the PSA bit (OPTION<3>). In this case, the prescaler will be cleared when TMR0 register is written with a value.

2.1.3 PCL (Low Bytes of Program Counter) & Stack

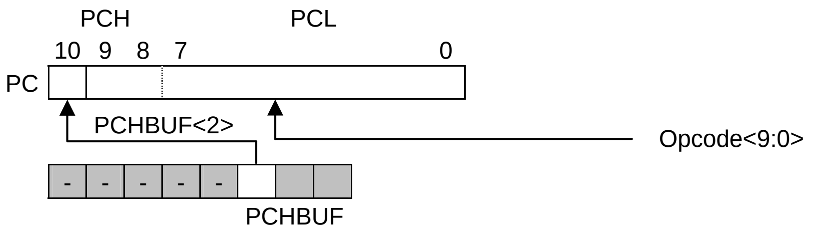
Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
02h (r/w)	PCL	Low order 8 bits of PC							

FM8P55/57 devices have a 9-bit (for FM8P55/55E) or 11-bit (for FM8P57/57E) wide Program Counter (PC) and five-level deep 9-bit (or 11-bit) hardware push/pop stack. The low byte of PC is called the PCL register. This register is readable and writable. The high byte of PC is called the PCH register. This register contains the PC<10:8> bits and is not directly readable or writable. All updates to the PCH register go through the PCHBUF register. As a program instruction is executed, the Program Counter will contain the address of the next program instruction to be executed. The PC value is increased by one, every instruction cycle, unless an instruction changes the PC.
For a GOTO instruction, the PC<9:0> is provided by the GOTO instruction word. The PC<10> is updated from the PCHBUF<2>. The PCL register is mapped to PC<7:0>, and the PCHBUF register is not updated.
For a CALL instruction, the PC<9:0> is provided by the CALL instruction word. The PC<10> is updated from the PCHBUF<2>. The next PC will be loaded (PUSHed) onto the top of STACK. The PCL register is mapped to PC<7:0>, and the PCHBUF register is not updated.
For a FGOTO instruction, the PC<10:0> is provided by the FGOTO instruction word. The PCL register is mapped to PC<7:0>, the PCHBUF<2> bit is also updated from the FGOTO instruction word, and the PCHBUF<1:0> bits are not updated.
For a FCALL instruction, the PC<10:0> 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 PCHBUF<2> bit is also updated from the FCALL instruction word, and the PCHBUF<1:0> bits are not updated.
For a RETIA, RETFIE, or RETURN instruction, the PC are updated (POPed) from the top of STACK. The PCL

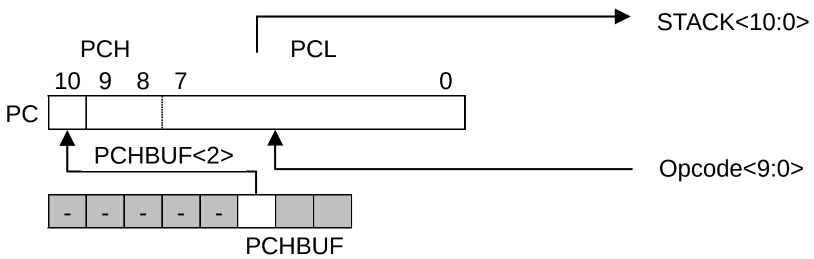
register is mapped to PC<7:0>, and the PCHBUF register is not updated.
For any instruction where the PCL is the destination (excluding TBL instruction), the PC<7:0> is provided by the instruction word or ALU result. However, the PC<10:8> will come from the PCHBUF<2:0> bits (PCHBUF → PCH).
For TBL instruction, the PC<7:0> is provided by the ALU result, and the PC<9:8> are not changed. The PC<10> will come from the PCH<2> bit.
PCHBUF register is never updated with the contents of PCH.

FIGURE 2.2: Loading of PC in Different Situations

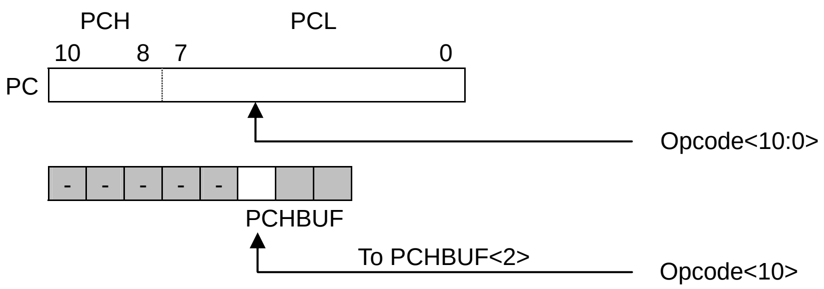
Situation 1: **GOTO** Instruction



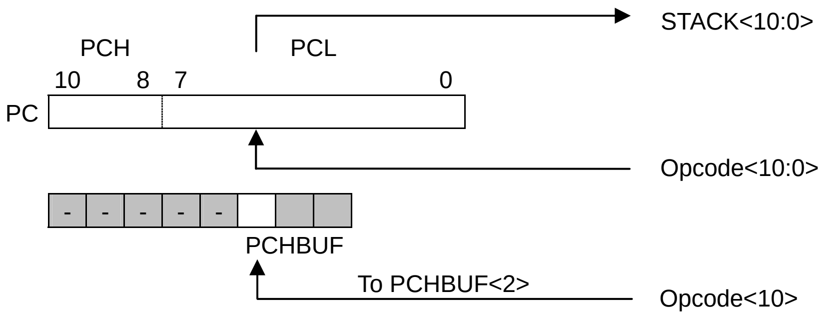
Situation 2: **CALL** Instruction



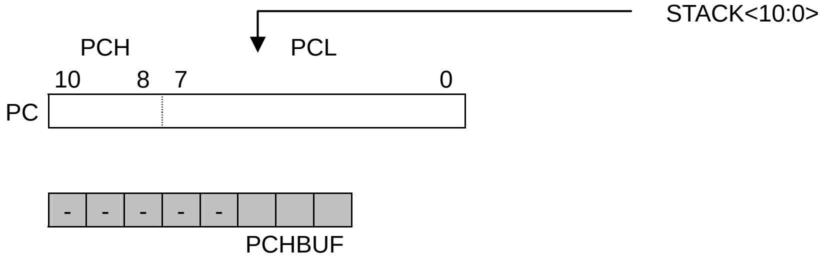
Situation 3: **FGOTO** Instruction



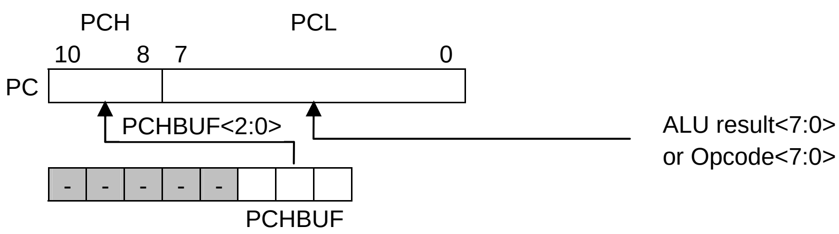
Situation 4: **FCALL** Instruction



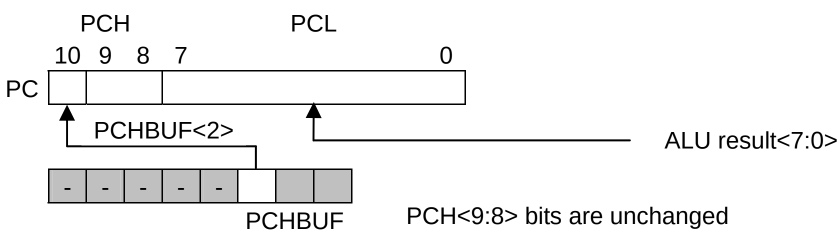
Situation 5: **RETIA**, **RETFIE**, or **RETURN** Instruction



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



Situation 7: **TBL** Instruction



Note: 1. Bits PC<10:9> and PCHBUF<2:1> are unimplemented for FM8P55/55E.
2. PCHBUF is used only for instruction with PCL as destination (except TBL instruction) for FM8P55/55E.
PCHBUF is used for instruction with PCL as destination, GOTO and CALL instructions for FM8P57/57E.

2.1.4 STATUS (Status Register)

Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
03h (r/w)	STATUS	GP2	GP1	GP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C

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

C : Carry/borrow bit.
ADDAR, ADDIA
= 1, a carry occurred.
= 0, a carry did not occur.
SUBAR, SUBIA
= 1, a borrow did not occur.
= 0, a borrow occurred.
Note : A subtraction is executed by adding the two's complement of the second operand. For rotate (RRR, RLR) instructions, this bit is loaded with either the high or low order bit of the source register.

DC : Half carry/half borrow bit.
ADDAR, ADDIA
= 1, a carry from the 4th low order bit of the result occurred.
= 0, a carry from the 4th low order bit of the result did not occur.
SUBAR, SUBIA
= 1, a borrow from the 4th low order bit of the result did not occur.
= 0, a borrow from the 4th low order bit of the result occurred.

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

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

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

GP2:GP0 : General purpose read/write bits.

2.1.5 FSR (Indirect Data Memory Address Pointer)

Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
04h (r/w)	FSR	RP1	RP0	Indirect data memory address pointer					

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

RP1:RP0 : For FM8P55/55E, these bits are not used. Read as "1"s.
For FM8P57/57E, these bits are used to switching the bank of four data memory banks. User can use "BANK" instruction to change bank. See 2.1.1 for detail description.

2.1.6 PORTA, PORTB & PORTC (Port Data Registers)

Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
05h (r/w)	PORTA	-	-	-	-	IOA3	IOA2	IOA1	IOA0
06h (r/w)	PORTB	IOB7	IOB6	IOB5	IOB4	IOB3	IOB2	IOB1	IOB0
07h (r/w)	PORTC	IOC7	IOC6	IOC5	IOC4	IOC3	IOC2	IOC1	IOC0

Reading the port (PORTA, PORTB, PORTC register) reads the status of the pins independent of the pin's input/output modes. Writing to these ports will write to the port data latch.
PORTA is a 4-bit port data Register. Only the low order 4 bits are used (PORTA<3:0>). Bits 7-4 are unimplemented and read as '0's.
PORTB and PORTC are 8-bit port data registers.

2.1.7 PCON (Power Control Register)

Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
08h (r/w)	PCON	WDTE	EIS	LVDTE	ROC	-	-	ODC67	/WUC45

WUC45 : = 0, Enable the input falling wake-up function of IOC4 and IOC5 pins.
= 1, Disable the input falling wake-up function of IOC4 and IOC5 pins.

ODC67 : = 0, Disable the internal open-drain of IOC6 and IOC7 pins.
= 1, Enable the internal open-drain of IOC6 and IOC7 pins.

Bit3:Bit2 : Not used. Read as "0"s.

ROC : R-option function of IOC0 and IOC1 pins enable bit.
= 0, Disable the R-option function.
= 1, Enable the R-option function. In this case, if a 430KΩ external resister is connected/disconnected to Vss, the status of IOC0 (IOC1) is read as "0"/"1".

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

EIS : Define the function of IOB0/INT pin.
= 0, IOB0 (bi-directional I/O pin) is selected. The path of INT is masked.
= 1, INT (external interrupt pin) is selected. In this case, the I/O control bit of IOB0 must be set to "1". The path of Port B input change of IOB0 pin is masked by hardware, the status of INT pin can also be read by way of reading PORTB.

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

2.1.8 WUCON (Port B Input Falling Wake-up Control Register)

Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
09h (r/w)	WUCON	/WUB7	/WUB6	/WUB5	/WUB4	/WUB3	/WUB2	/WUB1	/WUB0

/WUB0 : = 0, Enable the input falling wake-up function of IOB0 pin.
= 1, Disable the input falling wake-up function of IOB0 pin.

/WUB1 : = 0, Enable the input falling wake-up function of IOB1 pin.

= 1, Disable the input falling wake-up function of IOB1 pin.

/WUB2 : = 0, Enable the input falling wake-up function of IOB2 pin.
= 1, Disable the input falling wake-up function of IOB2 pin.

/WUB3 : = 0, Enable the input falling wake-up function of IOB3 pin.
= 1, Disable the input falling wake-up function of IOB3 pin.

/WUB4 : = 0, Enable the input falling wake-up function of IOB4 pin.
= 1, Disable the input falling wake-up function of IOB4 pin.

/WUB5 : = 0, Enable the input falling wake-up function of IOB5 pin.
= 1, Disable the input falling wake-up function of IOB5 pin.

/WUB6 : = 0, Enable the input falling wake-up function of IOB6 pin.
= 1, Disable the input falling wake-up function of IOB6 pin.

/WUB7 : = 0, Enable the input falling wake-up function of IOB7 pin.
= 1, Disable the input falling wake-up function of IOB7 pin.

2.1.9 PCHBUF (High Byte Buffer of Program Counter)

Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0Ah (r/w)	PCHBUF	-	-	-	-	-	Upper 3 bits Buffer of PC		

PCHBUF<2> : Program memory page selection bit.
= 0, Page 0.
= 1, Page 1.

User can use “PAGE” instruction to change memory page and maintains the program memory page. Otherwise, user can use “FGOTO” (far goto), or “FCALL” (far call) instructions to program user's code.
There is only 1 bit in FM8P55/55E. And there are 3 bits in FM8P57/57E.
See 2.1.3 for detail description.

2.1.10 PDCON (Pull-down Control Register)

Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0Bh (r/w)	PDCON	/PDB3	/PDB2	/PDB1	/PDB0	/PDA3	/PDA2	/PDA1	/PDA0

/PDA0 : = 0, Enable the internal pull-down of IOA0 pin.
= 1, Disable the internal pull-down of IOA0 pin.

/PDA1 : = 0, Enable the internal pull-down of IOA1 pin.
= 1, Disable the internal pull-down of IOA1 pin.

/PDA2 : = 0, Enable the internal pull-down of IOA2 pin.
= 1, Disable the internal pull-down of IOA2 pin.

/PDA3 : = 0, Enable the internal pull-down of IOA3 pin.
= 1, Disable the internal pull-down of IOA3 pin.

/PDB0 : = 0, Enable the internal pull-down of IOB0 pin.
= 1, Disable the internal pull-down of IOB0 pin.

/PDB1 : = 0, Enable the internal pull-down of IOB1 pin.
= 1, Disable the internal pull-down of IOB1 pin.

/PDB2 : = 0, Enable the internal pull-down of IOB2 pin.
= 1, Disable the internal pull-down of IOB2 pin.

/PDB3 : = 0, Enable the internal pull-down of IOB3 pin.
= 1, Disable the internal pull-down of IOB3 pin.

2.1.11 BPHCON (PortB Pull-high Control Register)

Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0Ch (r/w)	BPHCON	/PHB7	/PHB6	/PHB5	/PHB4	/PHB3	/PHB2	/PHB1	/PHB0

/PHB0 : = 0, Enable the internal pull-high of IOB0 pin.
= 1, Disable the internal pull-high of IOB0 pin.

/PHB1 : = 0, Enable the internal pull-high of IOB1 pin.
= 1, Disable the internal pull-high of IOB1 pin.

/PHB2 : = 0, Enable the internal pull-high of IOB2 pin.
= 1, Disable the internal pull-high of IOB2 pin.

/PHB3 : = 0, Enable the internal pull-high of IOB3 pin.
= 1, Disable the internal pull-high of IOB3 pin.

/PHB4 : = 0, Enable the internal pull-high of IOB4 pin.
= 1, Disable the internal pull-high of IOB4 pin.

/PHB5 : = 0, Enable the internal pull-high of IOB5 pin.
= 1, Disable the internal pull-high of IOB5 pin.

/PHB6 : = 0, Enable the internal pull-high of IOB6 pin.
= 1, Disable the internal pull-high of IOB6 pin.

/PHB7 : = 0, Enable the internal pull-high of IOB7 pin.
= 1, Disable the internal pull-high of IOB7 pin.

2.1.12 CPHCON (PortC Pull-high Control Register)

Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0Dh (r/w)	CPHCON	/PHC7	/PHC6	/PHC5	/PHC4	/PHC3	/PHC2	/PHC1	/PHC0

/PHC0 : = 0, Enable the internal pull-high of IOC0 pin.
= 1, Disable the internal pull-high of IOC0 pin.

/PHC1 : = 0, Enable the internal pull-high of IOC1 pin.
= 1, Disable the internal pull-high of IOC1 pin.

/PHC2 : = 0, Enable the internal pull-high of IOC2 pin.
= 1, Disable the internal pull-high of IOC2 pin.

/PHC3 : = 0, Enable the internal pull-high of IOC3 pin.
= 1, Disable the internal pull-high of IOC3 pin.

/PHC4 : = 0, Enable the internal pull-high of IOC4 pin.
= 1, Disable the internal pull-high of IOC4 pin.

/PHC5 : = 0, Enable the internal pull-high of IOC5 pin.
= 1, Disable the internal pull-high of IOC5 pin.

/PHC6 : = 0, Enable the internal pull-high of IOC6 pin.
= 1, Disable the internal pull-high of IOC6 pin.

/PHC7 : = 0, Enable the internal pull-high of IOC7 pin.
= 1, Disable the internal pull-high of IOC7 pin.

2.1.13 INTEN (Interrupt Mask Register)

Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0Eh (r/w)	INTEN	GIE	-	-	-	-	INTIE	-	T0IE

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

Bit1 : Not used. Read as “0”.

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

Bit6:BIT3 : Not used. Read as “0”s.

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

2.1.14 INTFLAG (Interrupt Status Register)

Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
0Fh (r/w)	INTFLAG	-	-	-	-	-	INTIF	-	T0IF

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

Bit1 : Not used. Read as “0”.

INTIF : External INT pin interrupt flag. Set by rising/falling (selected by INTEDG bit (OPTION<6>)) edge on INT pin, reset by software.

Bit7:BIT3 : Not used. Read as “0”s.

2.1.15 ACC (Accumulator)

Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
N/A (r/w)	ACC	Accumulator							

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

2.1.16 OPTION Register

Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
N/A (w)	OPTION	-	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

Accessed by OPTION instruction.

By executing the OPTION instruction, the contents of the ACC Register will be transferred to the OPTION Register. The OPTION Register is a 7-bit wide, write-only register which contains various control bits to configure the Timer0/WDT prescaler, Timer0, and the external INT interrupt.
The OPTION Register are “write-only” and are set all “1”s except INTEDG bit.

PS2:PS0 : Prescaler rate select 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 : Prescaler assign bit.
= 1, WDT (watch-dog timer).
= 0, TMR0 (Timer0).

T0SE : TMR0 source edge select bit.
= 1, Falling edge on T0CKI pin.
= 0, Rising edge on T0CKI pin.

T0CS : TMR0 clock source select bit.
= 1, External T0CKI pin.
= 0, internal instruction clock cycle.

INTEDG : Interrupt edge select bit.
= 1, interrupt on rising edge of INT pin.
= 0, interrupt on falling edge of INT pin.

Bit7 : Not used.

2.1.17 IOSTA, IOSTB & IOSTC (Port I/O Control Registers)

Address	Name	B7	B6	B5	B4	B3	B2	B1	B0
N/A (w)	IOSTA	Port A I/O Control Register							
N/A (w)	IOSTB	Port B I/O Control Register							
N/A (w)	IOSTC	Port C I/O Control Register							

Accessed by IOST instruction.

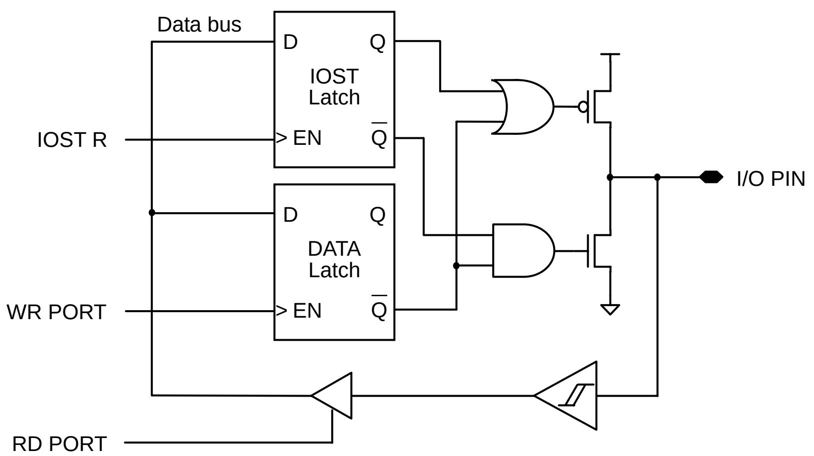
The Port I/O Control Registers are loaded with the contents of the ACC Register by executing the IOST R (05h~07h) instruction. A '1' from a IOST Register bit puts the corresponding output driver in hi-impedance state (input mode). A '0' enables the output buffer and puts the contents of the output data latch on the selected pins (output mode). The IOST Registers are "write-only" and are set (output drivers disabled) upon RESET.

2.2 I/O Ports

Port A, port B and port C are bi-directional tri-state I/O ports. Port A is a 4-pin I/O port. Port, port C are 8-pin I/O ports. All I/O pins (IOA<3:0>, IOB<7:0> and IOC<7:0>) have data direction control registers (IOSTA, IOSTB, IOSTC) which can configure these pins as output or input.
IOB and IOC have its corresponding pull-high control bits (BPHCON and CPHCON registers) to enable the weak internal pull-high. The weak pull-high is automatically turned off when the pin is configured as an output pin.
IOA<3:0> and IOB<3:0> have its corresponding pull-down control bits (PDCON register) to enable the weak internal pull-down. The weak pull-down is automatically turned off when the pin is configured as an output pin.
IOC<7:6> have its corresponding open-drain control bit (ODC67 bit (PCON<1>)) to enable the open-drain output when these pins are configured to be an output pin.
IOA0 and IOA1 are the R-option pins enabled by setting the ROC bit (PCON<4>). When the R-option function is used, it is recommended that IOA0 and IOA1 are used as output pins, and read the status of IOA0 and IOA1 before these pins are configured to be an output pin.
IOB and IOC<5:4> also provide the input falling wake-up function. Each pin has its corresponding input change wake-up enable bits (WUCON register and /WUC45 bit (PCON<0>)) to select the input falling wake-up source.
The IOB0 is also an external interrupt input signal by setting the EIS bit (PCON<6>). In this case, IOB0 input falling wake-up function will be disabled by hardware even if it is enabled by software.

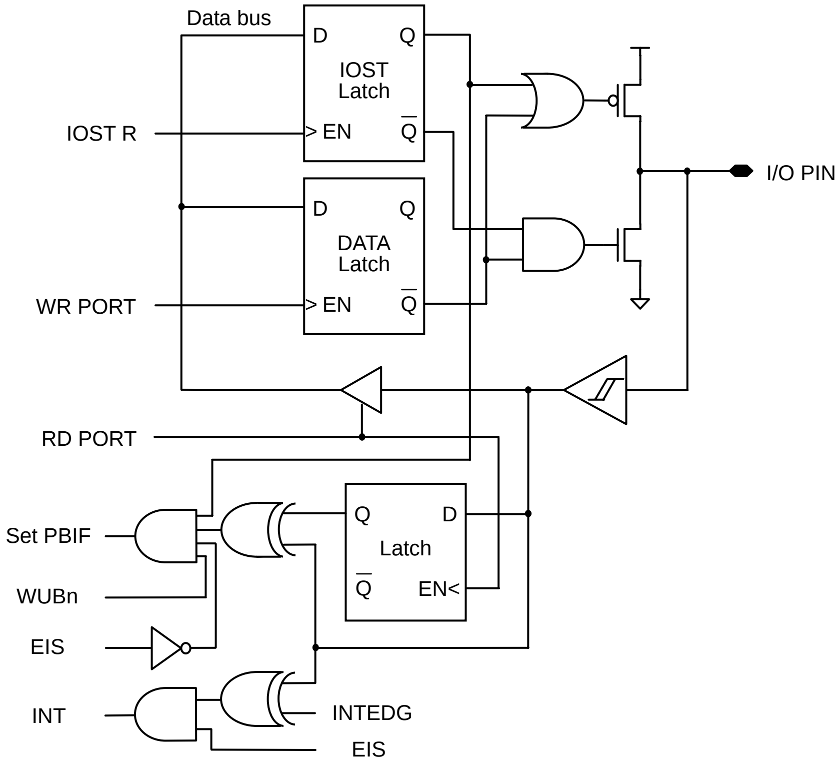
FIGURE 2.3: Block Diagram of I/O PINs

IOA3 ~ IOA0, IOC7 ~ IOC0 :



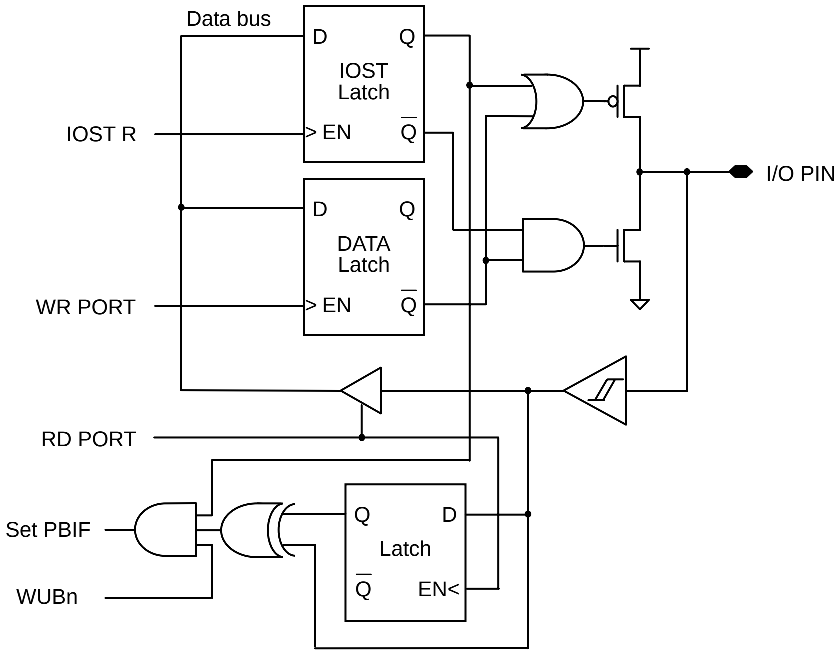
Pull-down is not shown in the figure

IOB0/INT :



Pull-high/pull-down and open-drain are not shown in the figure

IOB7 ~ IOB1 :



Pull-high/pull-down and open-drain are not shown in the figure

2.3 Timer0/WDT & Prescler

2.3.1 Timer0

The Timer0 is a 8-bit timer/counter. The clock source of Timer0 can come from the internal clock or by an external clock source (T0CKI pin).

2.3.1.1 Using Timer0 with an Internal Clock : Timer mode

Timer mode is selected by clearing the T0CS bit (OPTION<5>). In timer mode, the timer0 register (TMR0) will increment every instruction cycle (without prescaler). If TMR0 register is written, the increment is inhibited for the following two cycles.

2.3.1.2 Using Timer0 with an External Clock : Counter mode

Counter mode is selected by setting the T0CS bit (OPTON<5>). In this mode, Timer0 will increment either on every rising or falling edge of pin T0CKl. The incrementing edge is determined by the source edge select bit T0SE (OPTION<4>).
The external clock requirement is due to internal phase clock (Tosc) synchronization. Also, there is a delay in the actual incrementing of Timer0 after synchronization.
When no prescaler is used, the external clock input is the same as the prescaler output. The synchronization of T0CKI with the internal phase clocks is accomplished by sampling the prescaler output on the T2 and T4 cycles of the internal phase clocks. Therefore, it is necessary for T0CKI to be high for at least 2 T_{OSC} and low for at least 2 Tosc.
When a prescaler is used, the external clock input is divided by the asynchronous prescaler. For the external clock to meet the sampling requirement, the ripple counter must be taken into account. Therefore, it is necessary for T0CKI to have a period of at least 4Tosc divided by the prescaler value.

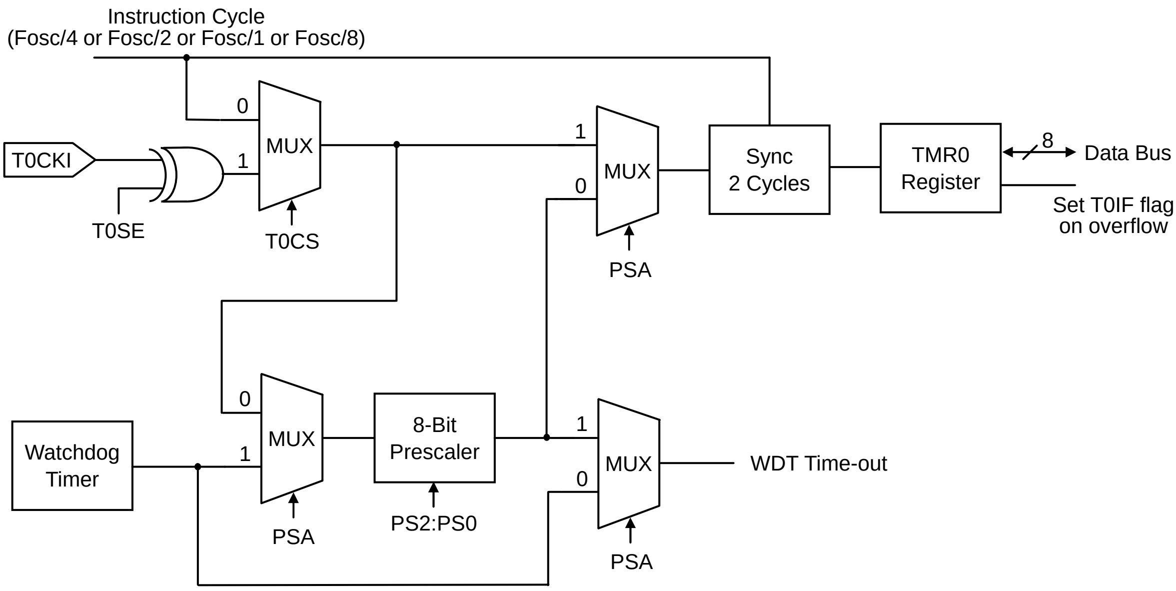
2.3.2 Watchdog Timer (WDT)

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 $\overline{TO}$ bit (STATUS<4>) will be cleared.
The WDT can be disabled by clearing the control bit WDTE (PCON<7>) to "0".
The WDT has a nominal time-out period of 18 ms (without prescaler). If a longer time-out period is desired, a prescaler 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 approxmately 2.3 seconds.
The CLRWDT instruction clears the WDT and the prescaler, if assigned to the WDT, and prevents it from timing out and generating a device reset.
The SLEEP instruction resets the WDT and the prescaler, if assigned to the WDT. This gives the maximum SLEEP time before a WDT Wake-up Reset.

2.3.3 Prescaler

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

FIGURE 2.4: Block Diagram of The Timer0/WDT Prescaler



2.4 Interrupts

The FM8P55/57 series has up to two sources of interrupt:

1. External interrupt INT pin.
2. TMR0 overflow interrupt.

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

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

When an interrupt event occur with the GIE bit and its corresponding interrupt enable bit are all set, the GIE bit will be cleared by hardware to disable any further interrupts, and the next instruction will be fetched from address 008h. The interrupt flag bits must be cleared by software before re-enabling GIE bit to avoid recursive interrupts.

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

The flag bit in INTFLAG 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 002h.

2.4.1 External INT Interrupt

External interrupt on INT pin is rising or falling edge triggered selected by INTEDG (OPTION<6>).

When a valid edge appears on the INT pin the flag bit INTIF (INTFLAG<2>) is set. This interrupt can be disabled by clearing INTIE bit (INTEN<2>).

2.4.2 Timer0 Interrupt

An overflow (FFh → 00h) in the TMR0 register will set the flag bit T0IF (INTFLAG<0>). This interrupt can be disabled by clearing T0IE bit (INTEN<0>).

2.5 Power-down Mode (SLEEP)

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

2.5.1 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. PORTB/IOC4/IOC5 input falling.

External RSTB reset and WDT time-out reset will cause a device reset. The $\overline{PD}$ and $\overline{TO}$ bits can be used to determine the cause of device reset. The $\overline{PD}$ bit is set on power-up and is cleared when SLEEP instruction is executed. The $\overline{TO}$ bit is cleared if a WDT time-out occurred.
For the device to wake-up through an PORTB/IOC4/IOC5 input falling event, and the program will execute next PC after wake-up. Any pin which corresponding /WUBn bit (WUCON<7:0>) or /WUC45 bit (PCON<0>) is set to “1” or configured as output will be excluded from this function.
The system wake-up delay time is 18ms plus 128 oscillator cycle time.

2.6 Reset

FM8P55/57 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 Vdd rise is detected. To use this feature, the user merely ties the RSTB pin to Vdd.
On-chip Low Voltage Detector (LVD) places the device into reset when Vdd is below a fixed voltage. This ensures that the device does not continue program execution outside the valid operation Vdd range. Brown-out RESET is typically used in AC line or heavy loads switched applications.
A RSTB or WDT Wake-up from SLEEP also results in a device RESET, and not a continuation of operation before SLEEP.
The $\overline{TO}$ and $\overline{PD}$ bits (STATUS<4:3>) are set or cleared depending on the different reset conditions.

2.6.1 Power-up Reset Timer(PWRT)

The Power-up Reset Timer provides a nominal 18ms 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 Vdd, temperature, and process variation.

2.6.2 Oscillator Start-up Timer(OST)

The OST timer provides a 128 oscillator cycle delay (from OSCI input) after the PWRT delay (18ms) 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.

2.6.3 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 18ms plus 128 oscillator cycle time.

FIGURE 2.5: Simplified Block Diagram of on-chip Reset Circuit

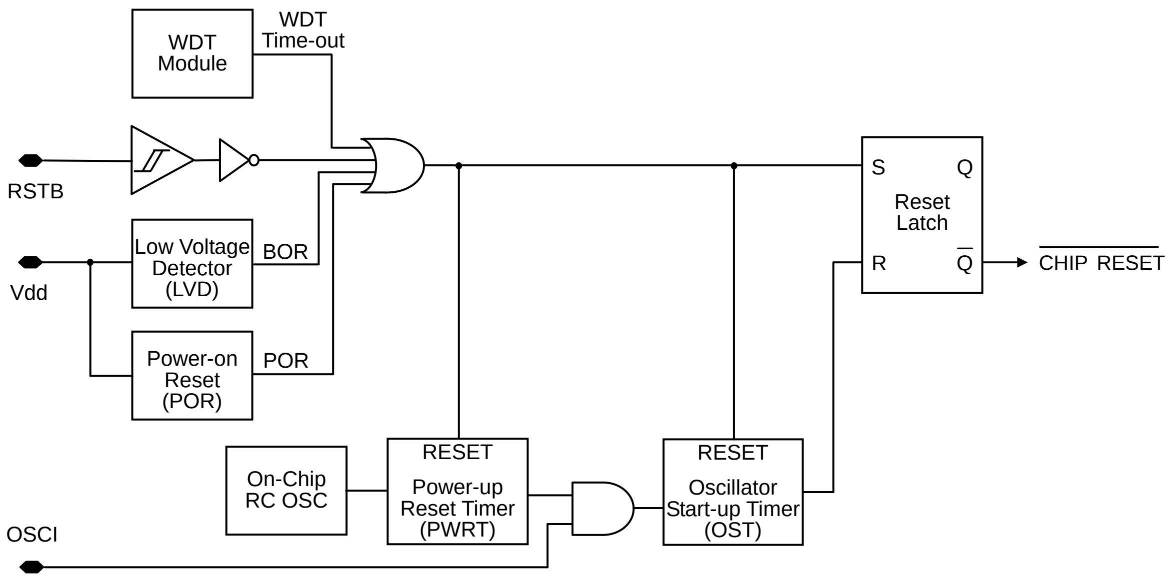
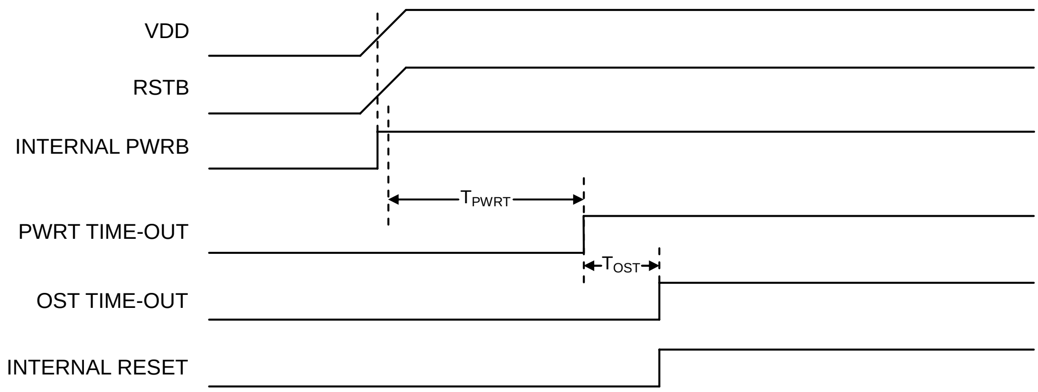
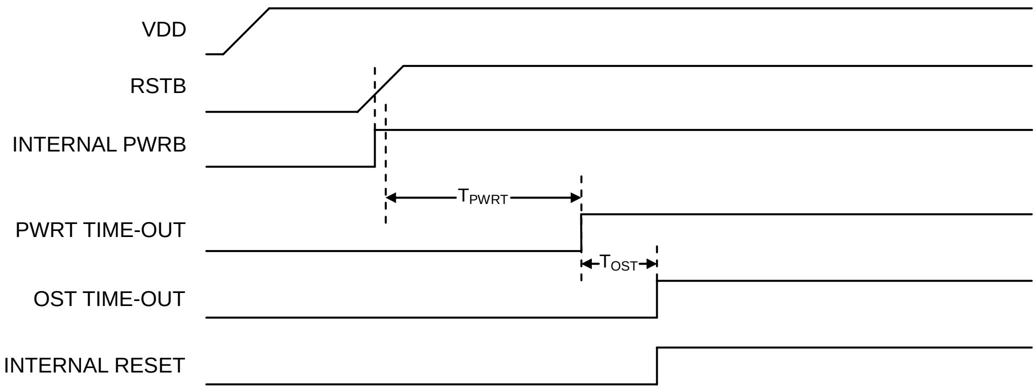


FIGURE 2.6: Time-out Sequence on Power-up (RSTB Pin Tied to Vdd)



Note: T_{PWRT} = 18 ms; T_{OST} = 128 oscillator cycle time

FIGURE 2.7: Time-out Sequence on Power-up (RSTB Pin Not Tied to Vdd)



Note: T_{PWRT} = 18 ms; T_{OST} = 128 oscillator cycle time

TABLE 2.1: Reset Conditions for All Registers

Register	Address	Power-on Reset Brown-out Reset	RSTB Reset WDT Reset
ACC	N/A	xxxx xxxx	uuuu uuuu
OPTION	N/A	-011 1111	-011 1111
IOSTA	05h	---- 1111	---- 1111
IOSTB	06h	1111 1111	1111 1111
IOSTC	07h	1111 1111	1111 1111
INDF	00h	xxxx xxxx	uuuu uuuu
TMR0	01h	xxxx xxxx	uuuu uuuu
PCL	02h	1111 1111	1111 1111
STATUS	03h	0001 1xxx	000# #uuu
FSR	04h	55: 11xx xxxx 57: xxxx xxxx	55: 11uu uuuu 57: uuuu uuuu
PORTA	05h	---- xxxx	---- uuuu
PORTB	06h	xxxx xxxx	uuuu uuuu
PORTC	07h	xxxx xxxx	uuuu uuuu
PCON	08h	1010 ----	1010 ----
WUCON	09h	0000 0000	0000 0000
PCHBUF	0Ah	55: ---- ---0 57: ---- -000	55: ---- ---0 57: ---- -000
PDCON	0Bh	1111 1111	1111 1111
BPHCON	0Ch	1111 1111	1111 1111
CPHCON	0Dh	1111 1111	1111 1111
INTEN	0Eh	0--- -0-0	0--- -0-0
INTFLAG	0Fh	---- -0-0	---- -0-0
General Purpose Registers	10 ~ 3Fh	xxxx xxxx	uuuu uuuu

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

TABLE 2.2: $\overline{TO}$ / $\overline{PD}$ Status after Reset

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

Legend: u = unchanged

TABLE 2.3: Events Affecting $\overline{TO}$ / $\overline{PD}$ Status Bits

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

Legend: u = unchanged

2.7 Hexadecimal Convert to Decimal (HCD)

Decimal format is another number format for FM8P55/57. 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

```
MOVIA   90h       ;Set immediate data = decimal format number "90" (ACC ← 90h)
MOVAR   30h       ;Load immediate data "90" to data memory address 30H
MOVIA   10h       ;Set immediate data = decimal format number "10" (ACC ← 10h)
ADDAR   30h, 0    ;Contents of the data memory address 30H and ACC are binary-added
                  ;the result loads to the ACC (ACC ← A0h, 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

```
MOVIA   10h       ;Set immediate data = decimal format number "10" (ACC ← 10h)
MOVAR   30h       ;Load immediate data "10" to data memory address 30H
MOVIA   20h       ;Set immediate data = decimal format number "20" (ACC ← 20h)
SUBAR   30h, 0    ;Contents of the data memory address 30H and ACC are binary-subtracted
                  ;the result loads to the ACC (ACC ← F0h, 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"
```

2.8 Oscillator Configurations

FM8P55/57 can be operated in four different oscillator modes. Users can program two configuration bits (Fosc<1:0>) to select the appropriate modes:

- LF: Low Frequency Crystal Oscillator
- XT: Crystal/Resonator Oscillator
- HF: High Frequency Crystal/Resonator Oscillator
- ERC: External Resistor/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 device option offers additional cost savings for timing insensitive applications. The RC oscillator frequency is a function of the supply voltage, the resistor (Rext) and capacitor (Cext), the operating temperature, and the process parameter.

FIGURE 2.8: HF, XT or LF Oscillator Modes (Crystal Operation or Ceramic Resonator)

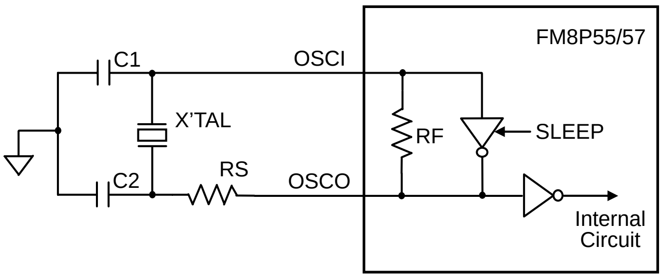


FIGURE 2.9: HF, XT or LF Oscillator Modes (External Clock Input Operation)

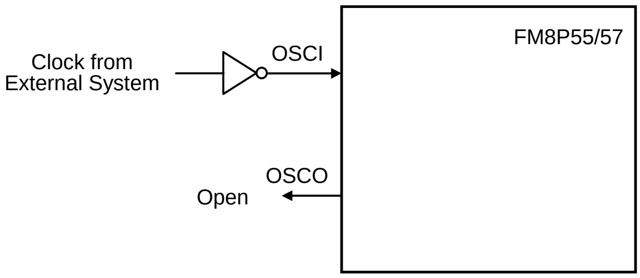
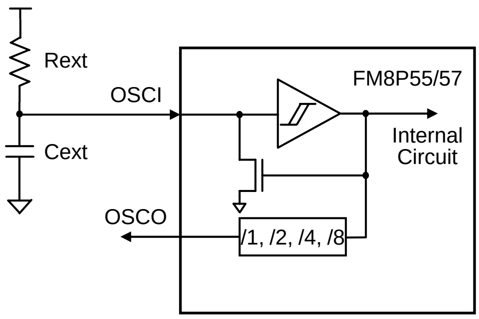


FIGURE 2.10: ERC Oscillator Mode



2.9 Configurations Word

TABLE 2.4: Configurations Word

bit	Name	Description
1, 0	Fosc<1:0>	Oscillator Selection Bits = 1, 1 → ERC mode (default) = 1, 0 → HF mode = 0, 1 → XT mode = 0, 0 → LF mode
2	WDTEN	Watchdog Timer Enable Bit = 1, WDT enabled (default) = 0, WDT disabled
3	PROTECT	Code Protection Bit = 1, EPROM code protection off (default) = 0, EPROM code protection on
5, 4	LVDT<1:0>	Low Voltage Detector Selection Bit = 1, 1 → disable (default) = 1, 0 → enable, LVDT voltage = 2.0V, controlled by SLEEP = 0, 1 → enable, LVDT voltage = 2.0V = 0, 0 → enable, LVDT voltage = 3.6V
7, 6	OSCD<1:0>	Instruction Period Selection Bits = 1, 1 → four oscillator periods (default) = 1, 0 → two oscillator periods = 0, 1 → one oscillator period = 0, 0 → eight oscillator periods
8	PMOD	Power Mode Selection Bit = 1, Non-power saving (default) = 0, Power saving
12 ~ 9	-	Unused

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	00h → WDT, 00h → WDT prescaler	1	$\overline{TO}$, $\overline{PD}$
OPTION	Load OPTION register	ACC → OPTION	1	-
SLEEP	Go into power-down mode	00h → WDT, 00h → WDT prescaler	1	$\overline{TO}$, $\overline{PD}$
TBL	Table look-up	PC<7:0> + ACC → PC<7:0> PC<9:8> unchanged PCHBUF<2> → PC<10>	1	C, DC, Z
INT	S/W interrupt	PC + 1 → Top of Stack, 002h → PC	2	-
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	-
RETURN	Return from subroutine	Top of Stack → PC	2	-
RETFIE	Return from interrupt, set GIE bit	Top of Stack → PC, 1 → GIE	2	-
CLRA	Clear ACC	00h → ACC	1	Z
IOST R	Load IOST register	ACC → IOST register	1	-
CLRR R	Clear R	00h → R	1	Z
MOVAR R	Move ACC to R	ACC → R	1	-
MOVR R, d	Move R	R → dest	1	Z
DECR R, d	Decrement R	R - 1 → dest	1	Z
DECRSZ R, d	Decrement R, Skip if 0	R - 1 → dest, Skip if result = 0	1/2/3 (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
RLR R, d	Rotate left f through Carry	R<7> → C, R<6:0> → dest<7:1>, C → dest<0>	1	C

RRR	**R, d**	Rotate right f 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 → PCHBUF<2>	1	-
CALL	**I**	Call subroutine	PC + 1 → Top of Stack, I → PC<9:0> PCHBUF<2> → PC<10>	2	-
GOTO	**I**	Unconditional branch	I → PC<9:0> PCHBUF<2> → PC<10>	2	-
FCALL	**I**	Call subroutine	PC + 1 → Top of Stack, I → PC<10:0> I<10> → PCHBUF<2>	3	-
FGOTO	**I**	Unconditional branch	I → PC<10:0> I<10> → PCHBUF<2>	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 (00h to 3Fh)
I : Immediate data
ACC : Accumulator
d : Destination select;
=0 (store result in ACC)
=1 (store result in file register R)
dest : Destination
PC : Program Counter
PCHBUF : High Byte Buffer of Program Counter
WDT : Watchdog Timer Counter
GIE : Global interrupt enable bit
$\overline{TO}$: Time-out bit
$\overline{PD}$: Power-down bit
C : Carry bit
DC : Digital carry bit
Z : Zero bit

ADCAR **Add ACC and R with Carry**

Syntax: ADCAR R, d
Operands: $0 \leq R \leq 63$
$d \in [0,1]$
Operation: R + ACC + C → dest
Status Affected: C, DC, Z
Description: Add the contents of the ACC register and register 'R' with Carry. If 'd' is 0 the result is stored in the ACC register. If 'd' is '1' the result is stored back in register 'R'.
Cycles: 1

ADDAR **Add ACC and R**

Syntax: ADDAR R, d
Operands: $0 \leq R \leq 63$
$d \in [0,1]$
Operation: ACC + R → dest
Status Affected: C, DC, Z
Description: Add the contents of the ACC register and register 'R'. If 'd' is 0 the result is stored in the ACC register. If 'd' is '1' the result is stored back in register 'R'.
Cycles: 1

ADDIA **Add ACC and Immediate**

Syntax: ADDIA I
Operands: $0 \leq I \leq 255$
Operation: ACC + I → ACC
Status Affected: C, DC, Z
Description: Add the contents of the ACC register with the 8-bit immediate 'I'. The result is placed in the ACC register.
Cycles: 1

ANDAR **AND ACC and R**

Syntax: ANDAR R, d
Operands: $0 \leq R \leq 63$
$d \in [0,1]$
Operation: ACC and R → dest
Status Affected: Z
Description: The contents of the ACC register are AND'ed with register 'R'. If 'd' is 0 the result is stored in the ACC register. If 'd' is '1' the result is stored back in register 'R'.
Cycles: 1

ANDIA **AND Immediate with ACC**

Syntax: ANDIA I
Operands: $0 \leq I \leq 255$
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:	0≤I≤3
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:	BCF R, b
Operands:	0≤R≤63 0≤b≤7
Operation:	0 → R<b>
Status Affected:	None
Description:	Clear bit 'b' in register 'R'.
Cycles:	1

BSR	**Set Bit in R**
Syntax:	BSR R, b
Operands:	0≤R≤63 0≤b≤7
Operation:	1 → R<b>
Status Affected:	None
Description:	Set bit 'b' in register 'R'.
Cycles:	1

BTRSC	**Test Bit in R, Skip if Clear**
Syntax:	BTRSC R, b
Operands:	0≤R≤63 0≤b≤7
Operation:	Skip if R<b> = 0
Status Affected:	None
Description:	If bit 'b' in register 'R' is 0 then the next instruction is skipped. If bit 'b' is 0 then next instruction fetched during the current instruction execution is discarded, and a NOP is executed instead making this a 2-cycle instruction..
Cycles:	1/2 (3 cycles if skip and followed by a 2-word instruction FCALL/FGOTO)

BTRSS	**Test Bit in R, Skip if Set**
Syntax:	BTRSS R, b
Operands:	0≤R≤63 0≤b≤7
Operation:	Skip if R<b> = 1
Status Affected:	None
Description:	If bit 'b' in register 'R' is '1' then the next instruction is skipped. If bit 'b' is '1', then the next instruction fetched during the current instruction execution, is discarded and a NOP is executed instead, making this a 2-cycle instruction.
Cycles:	1/2 (3 cycles if skip and followed by a 2-word instruction FCALL/FGOTO)

CALL	**Subroutine Call**
Syntax:	CALL I
Operands:	0 ≤ I ≤ 1023
Operation:	PC +1 → Top of Stack; I → PC<9:0> PCHBUF<2> → PC<10>
Status Affected:	None
Description:	Subroutine call. First, return address (PC+1) is pushed onto the stack. The 10-bit immediate address is loaded into PC bits <10:0>. CALL is a two-cycle instruction.
Cycles:	2

CLRA	**Clear ACC**
Syntax:	CLRA
Operands:	None
Operation:	00h → ACC; 1 → Z
Status Affected:	Z
Description:	The ACC register is cleared. Zero bit (Z) is set.
Cycles:	1

CLRR	**Clear R**
Syntax:	CLRR R
Operands:	0 ≤ R ≤ 63
Operation:	00h → R; 1 → Z
Status Affected:	Z
Description:	The contents of register 'R' are cleared and the Z bit is set.
Cycles:	1

CLRWDT	**Clear Watchdog Timer**
Syntax:	CLRWDT
Operands:	None
Operation:	00h → WDT; 00h → WDT prescaler (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 prescaler, if the prescaler 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:	0 ≤ R ≤ 63 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:	$0 \leq R \leq 63$ $d \in [0,1]$
Operation:	R - 1 → dest
Status Affected:	Z
Description:	Decrement register 'R'. If 'd' is 0 the result is stored in the ACC register. If 'd' is 1 the result is stored back in register 'R'.
Cycles:	1

DECRSZ **Decrement R, Skip if 0**

Syntax:	DECRSZ R, d
Operands:	$0 \leq R \leq 63$ $d \in [0,1]$
Operation:	R - 1 → dest; skip if result =0
Status Affected:	None
Description:	The contents of register 'R' are decremented. If 'd' is 0 the result is placed in the ACC register. If 'd' is 1 the result is placed back in register 'R'. If the result is 0, the next instruction, which is already fetched, is discarded and a NOP is executed instead making it a 2-cycle instruction.
Cycles:	1/2 (3 cycles if skip and followed by a 2-word instruction FCALL/FGOTO)

FCALL **Subroutine Call**

Syntax:	FCALL I
Operands:	$0 \leq I \leq 2047$
Operation:	PC +1 → Top of Stack; I → PC<10:0> I<10> → PCHBUF<2>
Status Affected:	None
Description:	Subroutine call. First, return address (PC+1) is pushed onto the stack. The 11-bit immediate address is loaded into PC bits <10:0>. FCALL is a two-word (three-cycle) instruction.
Cycles:	3

FGOTO	**Unconditional Branch**
Syntax:	FGOTO I
Operands:	0≤I≤2047
Operation:	I → PC<10:0> I<10> → PCHBUF<2>
Status Affected:	None
Description:	FGOTO is an unconditional branch. The 11-bit immediate value is loaded into PC bits <10:0>. FGOTO is a two-word (three-cycle) instruction.
Cycles:	3

GOTO	**Unconditional Branch**
Syntax:	GOTO I
Operands:	0≤I≤1023
Operation:	I → PC<9:0> PCHBUF<2> → PC<10>
Status Affected:	None
Description:	GOTO is an unconditional branch. The 10-bit immediate value is loaded into PC bits <9:0>. GOTO is a two-cycle instruction.
Cycles:	2

INCR	**Increment R**
Syntax:	INCR R, d
Operands:	0≤R≤63 d∈[0,1]
Operation:	R + 1 → dest
Status Affected:	Z
Description:	The contents of register 'R' are incremented. 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

INCRSZ	**Increment R, Skip if 0**
Syntax:	INCRSZ R, d
Operands:	0≤R≤63 d∈[0,1]
Operation:	R + 1 → dest, skip if result = 0
Status Affected:	None
Description:	The contents of register 'R' are incremented. If 'd' is 0 the result is placed in the ACC register. If 'd' is the result is placed 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 making it a 2-cycle instruction.
Cycles:	1/2 (3 cycles if skip and followed by a 2-word instruction FCALL/FGOTO)

INT	**S/W Interrupt**
Syntax:	INT
Operands:	None
Operation:	PC + 1 → Top of Stack, 002h → PC
Status Affected:	None
Description:	Interrupt subroutine call. First, return address (PC+1) is pushed onto the stack. The address 002h is loaded into PC bits <10:0>.
Cycles:	2

IORAR	**OR ACC with R**
Syntax:	IORAR R, d
Operands:	$0 \leq R \leq 63$ $d \in [0,1]$
Operation:	ACC or R → dest
Status Affected:	Z
Description:	Inclusive OR the ACC register with register ‘R’. If ‘d’ is 0 the result is placed in the ACC register. If ‘d’ is 1 the result is placed back in register ‘R’.
Cycles:	1

IORIA	**OR Immediate with ACC**
Syntax:	IORIA I
Operands:	$0 \leq I \leq 255$
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 = 5,6 or 7
Operation:	ACC → IOST register R
Status Affected:	None
Description:	IOST register ‘R’ (R= 5,6 or 7) is loaded with the contents of the ACC register.
Cycles:	1

MOVAR	**Move ACC to R**
Syntax:	MOVAR R
Operands:	$0 \leq R \leq 63$
Operation:	ACC → R
Status Affected:	None
Description:	Move data from the ACC register to register ‘R’.
Cycles:	1

MOVIA	**Move Immediate to ACC**
Syntax:	MOVIA I
Operands:	$0 \leq I \leq 255$
Operation:	I → ACC
Status Affected:	None
Description:	The 8-bit immediate ‘I’ is loaded into the ACC register. The don’t cares will assemble as 0s.
Cycles:	1

MOVR	**Move R**
Syntax:	MOVR R, d
Operands:	$0 \leq R \leq 63$ $d \in [0,1]$
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

PAGE Move Immediate to program page bits

Syntax:	PAGE I
Operands:	$0 \leq I \leq 1$
Operation:	I → PCHBUF<2>
Status Affected:	None
Description:	The program page bits are loaded with the 1-bit immediate 'I'.
Cycles:	1

RETFIE **Return from Interrupt, Set 'GIE' Bit**

Syntax:	RETFIE
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). The 'GIE' bit is set to 1. This is a two-cycle instruction.
Cycles:	2

RETIA **Return with Immediate in ACC**

Syntax:	RETIA I
Operands:	$0 \leq I \leq 255$
Operation:	I → ACC; Top of Stack → PC
Status Affected:	None
Description:	The ACC register is loaded with the 8-bit immediate 'I'. The program counter is loaded from the top of the stack (the return address). This is a two-cycle instruction.
Cycles:	2

RETURN **Return from Subroutine**

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

RLR	**Rotate Left f through Carry**
Syntax:	RLR R, d
Operands:	0≤R≤63 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 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 f through Carry**
Syntax:	RRR R, d
Operands:	0≤R≤63 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:	00h → WDT; 00h → WDT prescaler; 1 → $\overline{TO}$; 0 → $\overline{PD}$
Status Affected:	$\overline{TO}$, $\overline{PD}$
Description:	Time-out status bit ($\overline{TO}$) is set. The power-down status bit ($\overline{PD}$) is cleared. The WDT and its prescaler are cleared. The processor is put into SLEEP mode.
Cycles:	1

SBCAR	**Subtract ACC from R with Carry**
Syntax:	SBCAR R, d
Operands:	0≤R≤63 d∈[0,1]
Operation:	R + $\overline{ACC}$ + C → dest
Status Affected:	C, DC, Z
Description:	Add the 2's complement data of the ACC register from register 'R' with Carry. If 'd' is 0 the result is stored in the ACC register. If 'd' is 1 the result is stored back in register 'R'.
Cycles:	1

SUBAR	**Subtract ACC from R**
Syntax:	SUBAR R, d
Operands:	0 ≤ R ≤ 63 d ∈ [0,1]
Operation:	R - ACC → dest
Status Affected:	C, DC, Z
Description:	Subtract (2's complement method) the ACC register from register 'R'. If 'd' is 0 the result is stored in the ACC register. If 'd' is 1 the result is stored back in register 'R'.
Cycles:	1

SUBIA	**Subtract ACC from Immediate**
Syntax:	SUBIA I
Operands:	0 ≤ I ≤ 255
Operation:	I - ACC → ACC
Status Affected:	C, DC, Z
Description:	Subtract (2's complement method) the ACC register from the 8-bit immediate 'I'. The result is placed in the ACC register.
Cycles:	1

SWAPR	**Swap nibbles in R**
Syntax:	SWAPR R, d
Operands:	0 ≤ R ≤ 63 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 PCHBUF<2> → PC<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:	0 ≤ R ≤ 63 d ∈ [0,1]
Operation:	ACC xor R → dest
Status Affected:	Z
Description:	Exclusive OR the contents of the ACC register with register 'R'. If 'd' is 0 the result is stored in the ACC register. If 'd' is 1 the result is stored back in register 'R'.
Cycles:	1

XORIA	**Exclusive OR Immediate with ACC**
Syntax:	XORIA I
Operands:	$0 \leq I \leq 255$
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

Ambient Operating Temperature	0℃ to +70℃
Store Temperature	-65℃ to +150℃
DC Supply Voltage (Vdd)	0V to +6.0V
Input Voltage with respect to Ground (Vss)	-0.3V to (Vdd + 0.3)V

5.0 OPERATING CONDITIONS

DC Supply Voltage	+2.3V to +5.5V
Operating Temperature	0℃ to +70℃

6.0 ELECTRICAL CHARACTERISTICS

6.1 ELECTRICAL CHARACTERISTICS of FM8P55E/57E

Under Operating Conditions, at four clock instruction cycles and WDT & LVDT are disabled

Sym	Description	Conditions	Min.	Typ.	Max.	Unit
F_{HF}	X'tal oscillation range	HF mode, Vdd=5V	1		20	MHz
		HF mode, Vdd=3V	1		15	
F_{XT}	X'tal oscillation range	XT mode, Vdd=5V	0.5		10	MHz
		XT mode, Vdd=3V	0.5		10	
F_{LF}	X'tal oscillation range	LF mode, Vdd=5V	32		4000	KHZ
		LF mode, Vdd=3V	32		1000	
F_{ERC}	RC oscillation range	ERC mode, Vdd=5V	DC		15	MHz
		ERC mode, Vdd=3V	DC		7	
V_{IH}	Input high voltage	I/O ports, Vdd=5V	2.0			V
		RSTB, T0CKI pins, Vdd=5V	4.0			
		I/O ports, Vdd=3V	1.5			
		RSTB, T0CKI pins, Vdd=3V	2.4			
V_{IL}	Input low voltage	I/O ports, Vdd=5V			1.0	V
		RSTB, T0CKI pins, Vdd=5V			1.0	
		I/O ports, Vdd=3V			0.6	
		RSTB, T0CKI pins, Vdd=3V			0.6	
V_{OH}	Output high voltage	I_{OH}=-5.4mA, Vdd=5V	3.6			V
V_{OL}	Output low voltage	I_{OL}=8.7mA, Vdd=5V			0.6	V
I_{PH}	Pull-high current	Input pin at Vss, Vdd=5V		-45		uA
I_{PD}	Pull-down current	Input pin at Vdd, Vdd=5V		35		uA
I_{WDT}	WDT current	Vdd=5V		9	12	uA
		Vdd=3V		2	4	
T_{WDT}	WDT period	Vdd=3V		20.4		mS
		Vdd=4V		17.9		
		Vdd=5V		16.2		
I_{LVDT}	LVDT current	Vdd=5V LVDT = 3.6V		3	4	uA
		Vdd=5V LVDT = 2V		2	3	
		Vdd=3V LVDT = 2V		1.5	2.5	
I_{SB}	Power down current	Sleep mode, Vdd=5V, WDT enable		20		uA
		Sleep mode, Vdd=5V, WDT disable		3		
		Sleep mode, Vdd=3V, WDT enable		2.5		
		Sleep mode, Vdd=3V, WDT disable		1.1		
I_{DD}	Operating current	HF mode, Vdd=5V, 4 clock instruction				mA
		20MHz		2.04		
		15MHz		1.68		
		10MHz		1.28		
		4MHz		0.78		
		2MHz		0.62		

I_{DD}	Operating current	HF mode, Vdd=3V, 4 clock instruction				mA
		20MHz		0.92		
		15MHz		0.72		
		10MHz		0.54		
		4MHz		0.30		
		2MHz		0.19		
I_{DD}	Operating current	HF mode, Vdd=5V, 2 clock instruction				mA
		20MHz		2.94		
		15MHz		2.34		
		10MHz		1.74		
		4MHz		0.96		
		2MHz		0.68		
I_{DD}	Operating current	HF mode, Vdd=3V, 2 clock instruction				mA
		20MHz		1.38		
		15MHz		1.07		
		10MHz		0.77		
		4MHz		0.38		
		2MHz		0.24		
I_{DD}	Operating current	XT mode, Vdd=5V, 4 clock instruction				mA
		20MHz		1.69		
		15MHz		1.36		
		10MHz		1.04		
		4MHz		0.64		
		2MHz		0.49		
I_{DD}	Operating current	XT mode, Vdd=3V, 4 clock instruction				mA
		20MHz		0.78		
		15MHz		0.60		
		10MHz		0.44		
		4MHz		0.24		
		2MHz		0.17		
I_{DD}	Operating current	XT mode, Vdd=5V, 2 clock instruction				mA
		20MHz		2.81		
		15MHz		2.20		
		10MHz		1.60		
		4MHz		0.87		
		2MHz		0.61		
I_{DD}	Operating current	XT mode, Vdd=3V, 2 clock instruction				mA
		20MHz		1.36		
		15MHz		1.05		
		10MHz		0.73		
		4MHz		0.36		
		2MHz		0.23		

I_{DD}	Operating current	LF mode, Vdd=5V, 4 clock instruction						uA
		2MHz				290		
		1MHz				208		
		500KHz				167		
		100KHz				118		
		32KHz				101		
I_{DD}	Operating current	LF mode, Vdd=3V, 4 clock instruction						uA
		2MHz				105		
		1MHz				73		
		500KHz				54		
		100KHz				33		
		32KHz				26		
I_{DD}	Operating current	LF mode, Vdd=5V, 2 clock instruction						uA
		2MHz				371		
		1MHz				269		
		500KHz				194		
		100KHz				130		
		32KHz				108		
I_{DD}	Operating current	LF mode, Vdd=3V, 2 clock instruction						uA
		2MHz				158		
		1MHz				100		
		500KHz				67		
		100KHz				38		
		32KHz				29		
I_{DD}	Operating current	ERC mode, Vdd=5V, 4 clock instruction						mA
		C=3P	R=1Kohm	F=14.96MHz		4.572		
			R=3.3Kohm	F=11.06MHz		1.845		
			R=10Kohm	F=5.80MHz		0.761		
			R=100Kohm	F=808KHz		0.170		
			R=300Kohm	F=276KHz		0.119		
		C=20P	R=1Kohm	F=11.7MHz		4.226		
			R=3.3Kohm	F=6.35MHz		1.519		
			R=10Kohm	F=2.73MHz		0.613		
			R=100Kohm	F=320KHz		0.147		
			R=300Kohm	F=108KHz		0.109		
		C=100P	R=1Kohm	F=5.23MHz		3.429		
			R=3.3Kohm	F=2.05MHz		1.163		
			R=10Kohm	F=748KHz		0.454		
			R=100Kohm	F=80KHz		0.126		
			R=300Kohm	F=26.4KHz		0.100		
		C=300P	R=1Kohm	F=2.5MHz		3.024		

			R=3.3Kohm	F=900KHz		1.021		
			R=10Kohm	F=316KHz		0.403		
			R=100Kohm	F=32KHz		0.119		
			R=300Kohm	F=10.67KHz		0.098		
I_{DD}	Operating current	ERC mode, Vdd=3V, 4 clock instruction						mA
		C=3P	R=1Kohm	F=8.29MHz		2.280		
			R=3.3Kohm	F=7.2MHz		0.913		
			R=10Kohm	F=4.58MHz		0.396		
			R=100Kohm	F=900KHz		0.071		
			R=300Kohm	F=316KHz		0.040		
		C=20P	R=1Kohm	F=7MHz		2.214		
			R=3.3Kohm	F=5.1MHz		0.837		
			R=10Kohm	F=2.71MHz		0.327		
			R=100Kohm	F=374KHz		0.058		
			R=300Kohm	F=128KHz		0.035		
		C=100P	R=1Kohm	F=4.14MHz		2.060		
			R=3.3Kohm	F=2.11MHz		0.688		
			R=10Kohm	F=848KHz		0.253		
			R=100Kohm	F=96KHz		0.047		
			R=300Kohm	F=32KHz		0.030		
		C=300P	R=1Kohm	F=2.36MHz		1.890		
			R=3.3Kohm	F=972KHz		0.630		
			R=10Kohm	F=360KHz		0.226		
			R=100Kohm	F=38KHz		0.043		
			R=300Kohm	F=12.71KHz		0.028		
I_{DD}	Operating current	ERC mode, Vdd=5V, 2 clock instruction						mA
		C=3P	R=1Kohm	F=15.16MHz		5.435		
			R=3.3Kohm	F=11.27MHz		2.358		
			R=10Kohm	F=5.77MHz		986		
			R=100Kohm	F=826KHz		0.183		
			R=300Kohm	F=274KHz		0.108		
		C=20P	R=1Kohm	F=11.56MHz		4.835		
			R=3.3Kohm	F=6.12MHz		1.808		
			R=10Kohm	F=2.72MHz		0.701		
			R=100Kohm	F=308KHz		0.138		
			R=300Kohm	F=105KHz		0.092		
		C=100P	R=1Kohm	F=5.32MHz		3.680		
			R=3.3Kohm	F=1.99MHz		1.234		
			R=10Kohm	F=722KHz		0.479		
			R=100Kohm	F=77KHz		0.110		
			R=300Kohm	F=25.0KHz		0.081		
		C=300P	R=1Kohm	F=2.52MHz		3.107		
			R=3.3Kohm	F=892KHz		1.057		
			R=10Kohm	F=312KHz		0.398		
			R=100Kohm	F=32KHz		0.102		
			R=300Kohm	F=11KHz		0.077		

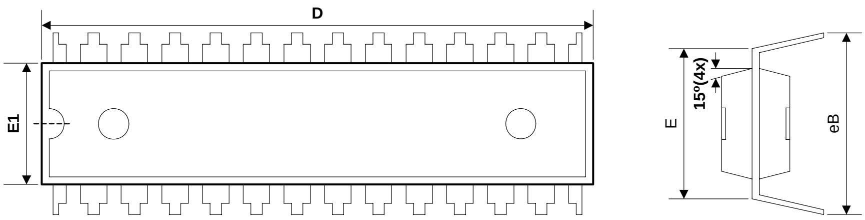
I_{DD}	Operating current	ERC mode, Vdd=3V, 2 clock instruction						mA
		C=3P	R=1Kohm	F=8.306MHz		2.552		
			R=3.3Kohm	F=7.29MHz		1.130		
			R=10Kohm	F=4.81MHz		0.518		
			R=100Kohm	F=904KHz		0.084		
			R=300Kohm	F=338KHz		0.039		
		C=20P	R=1Kohm	F=7.08MHz		2.445		
			R=3.3Kohm	F=5.07MHz		0.986		
			R=10Kohm	F=2.68MHz		0.393		
			R=100Kohm	F=362KHz		0.061		
			R=300Kohm	F=123KHz		0.031		
		C=100P	R=1Kohm	F=4.11MHz		2.197		
			R=3.3Kohm	F=2.03MHz		0.745		
			R=10Kohm	F=810KHz		0.270		
			R=100Kohm	F=91KHz		0.043		
			R=300Kohm	F=30KHz		0.025		
		C=300P	R=1Kohm	F=2.37MHz		1.953		
			R=3.3Kohm	F=964KHz		0.648		
			R=10Kohm	F=354KHz		0.231		
			R=100Kohm	F=38KHz		0.038		
			R=300Kohm	F=13KHz		0.022		

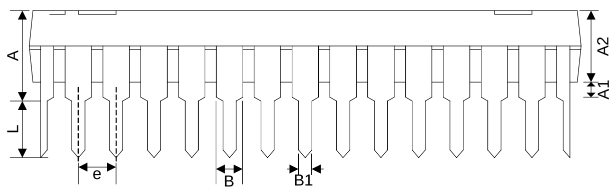
6.2 ELECTRICAL CHARACTERISTICS of FM8P55/57

To be defined

PACKAGE DIMENSION

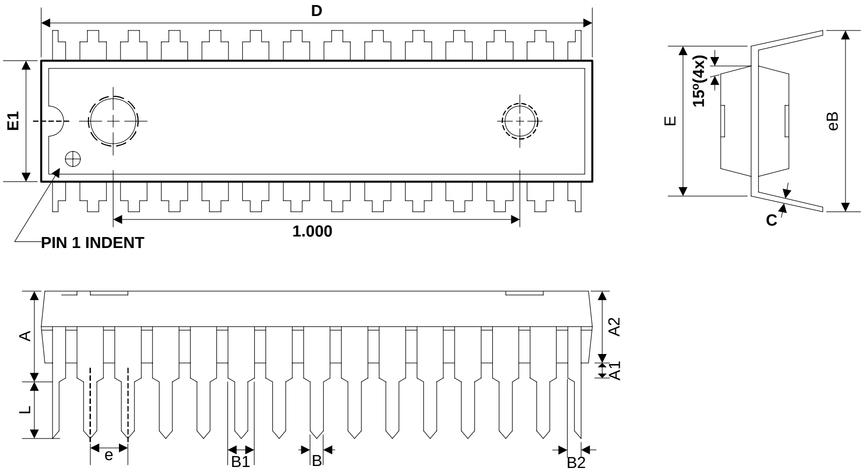
7.1 28-PIN PDIP 600mil





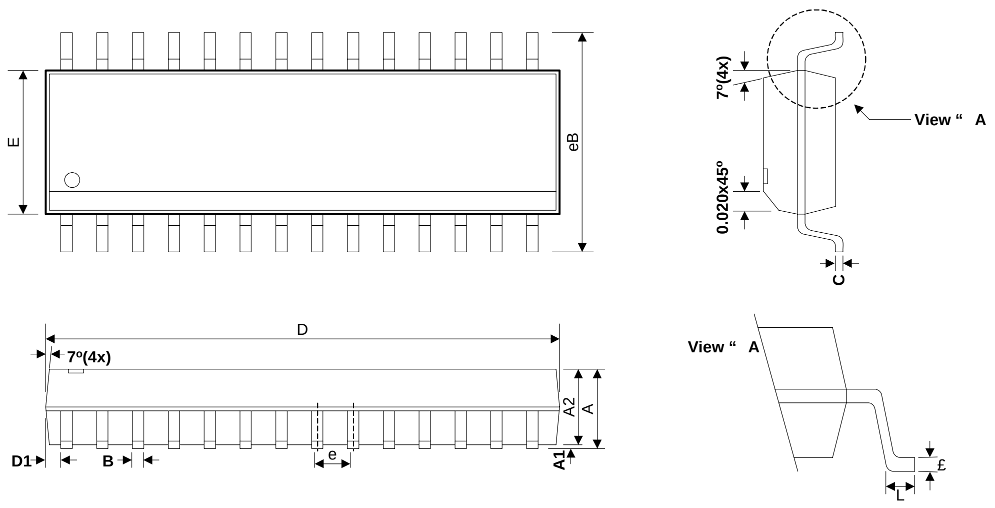
Symbols	Dimension In Millimeters			Dimension In Inches		
	Min	Nom	Max	Min	Nom	Max
A	-	-	5.59	-	-	0.220
A1	0.38	-	-	0.015	-	-
A2	3.81	3.94	4.06	0.150	0.155	0.160
B	-	1.52	-	-	0.06	-
B1	-	0.46	-	-	0.018	-
D	36.96	37.08	37.34	1.455	1.460	1.470
E	-	15.24	-	-	0.600	-
E1	13.72	13.84	13.97	0.540	0.545	0.550
e	-	2.54	-	-	0.100	-
L	3.18	-	-	0.125	-	-
eB	16.00	16.51	17.02	0.630	0.650	0.670

7.2 28-PIN Skinny PDIP 300mil



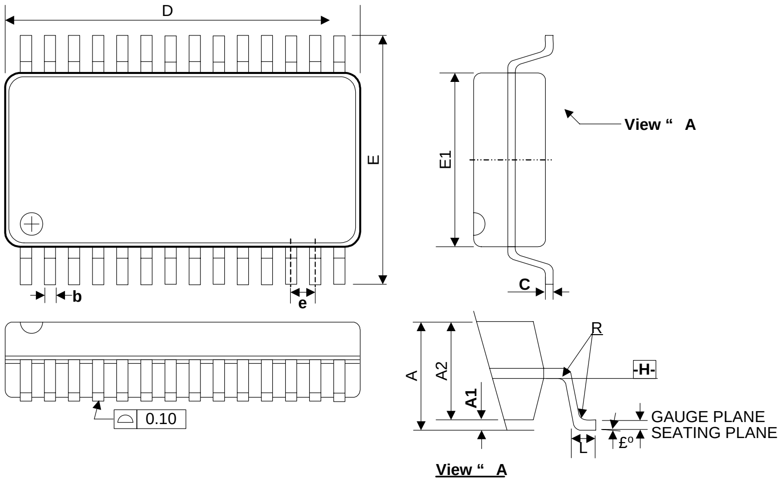
Symbols	Dimension In Millimeters			Dimension In Inches		
	Min	Nom	Max	Min	Nom	Max
A	-	-	4.57	-	-	0.180
A1	0.38	-	-	0.015	-	-
A2	-	3.30	3.56	-	0.130	0.140
B	1.02	-	1.65	0.0040	-	0.065
B1	0.41	-	0.58	0.016	-	0.023
B2	0.71	-	1.12	0.028	-	0.044
C	0.20	0.25	0.33	0.008	0.010	0.013
D	35.13	35.18	35.43	1.383	1.385	1.395
E	7.87	8.31	8.38	0.310	0.327	0.330
E1	7.26	7.32	7.52	0.284	0.288	0.296
e	-	2.54	-	-	0.100	-
L	3.18	-	-	0.125	-	-
eB	8.64	-	9.65	0.340	-	0.380

7.3 28-PIN SOP 300mil



Symbols	Dimension In Millimeters			Dimension In Inches		
	Min	Nom	Max	Min	Nom	Max
A	-	2.488	2.743	-	0.098	0.108
A1	0.152	-	-	0.006	-	-
A2	2.21	2.336	2.464	0.087	0.091	0.097
B	0.305	0.406	0.508	0.012	0.016	0.020
C	0.204	0.254	0.304	0.008	0.010	0.012
D	17.78	17.91	18.42	0.700	0.705	0.725
E	7.366	7.493	7.62	0.290	0.295	0.300
e	1.219	1.270	1.321	0.048	0.050	0.052
eB	10.26	10.42	10.57	0.404	0.410	0.416
L	0.635	-	-	0.025	-	-
θ	0°	4°	8°	0°	4°	8°
D1	0.356	0.508	-	0.014	0.020	-

7.4 28-PIN SSOP 209mil



Symbols	Dimension In Millimeters		
	Min	Nom	Max
A	-	-	2.00
A1	0.05	-	-
A2	1.62	1.75	1.85
b	0.22	-	0.38
c	0.09	-	0.25
D	9.90	10.20	10.50
E	7.40	7.80	8.20
E1	5.00	5.30	5.60
e	0.65 BSC	10.42	10.57
L	0.55	0.75	0.95
R	0.09	-	-
θ^o	0^o	4^o	8^o

7.0 ORDERING INFORMATION

OTP Type MCU	Package Type	Pin Count	Package Size
FM8P55EP	PDIP	28	600 mil
FM8P55EM	Skinny PDIP	28	300 mil
FM8P55ED	SOP	28	300 mil
FM8P55ER	SSOP	28	209 mil
FM8P57EP	PDIP	28	600 mil
FM8P57EM	Skinny PDIP	28	300 mil
FM8P57ED	SOP	28	300 mil
FM8P57ER	SSOP	28	209 mil

Mask Type MCU	Package Type	Pin Count	Package Size
FM8P55P	PDIP	28	600 mil
FM8P55M	Skinny PDIP	28	300 mil
FM8P55D	SOP	28	300 mil
FM8P55R	SSOP	28	209 mil
FM8P57P	PDIP	28	600 mil
FM8P57M	Skinny PDIP	28	300 mil
FM8P57D	SOP	28	300 mil
FM8P57R	SSOP	28	209 mil