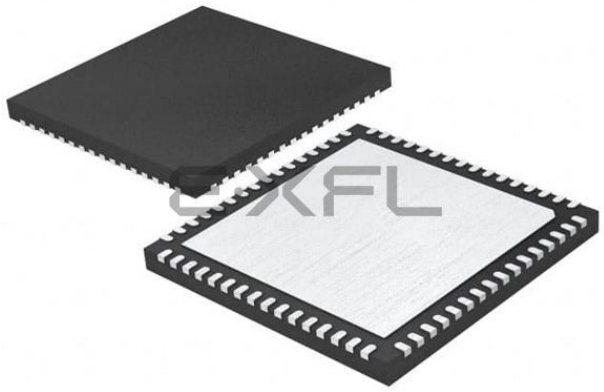


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	64MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LCD, POR, PWM, WDT
Number of I/O	53
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 16x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	64-VFQFN Exposed Pad
Supplier Device Package	64-VQFN (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f66k90-e-mr

Special Microcontroller Features:

- Operating Voltage Range: 1.8V to 5.5V
- On-Chip 3.3V Regulator
- Operating Speed up to 64 MHz
- Up to 128 Kbytes On-Chip Flash Program Memory
- Data EEPROM of 1,024 Bytes
- 4K x 8 General Purpose Registers (SRAM)
- 10,000 Erase/Write Cycle Flash Program Memory, Minimum
- 1,000,000 Erase/write Cycle Data EEPROM Memory, Typical
- Flash Retention 40 Years, Minimum
- Three Internal Oscillators: LF-INTRC (31 kHz), MF-INTOSC (500 kHz) and HF-INTOSC (16 MHz)
- Self-Programmable under Software Control
- Priority Levels for Interrupts
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
  - Programmable period from 4 ms to 4,194s (about 70 minutes)
- In-Circuit Serial Programming™ (ICSP™) via Two Pins
- In-Circuit Debug via Two Pins
- Programmable:
  - BOR
  - LVD
- Two Enhanced Addressable USART modules:
  - LIN/J2602 support
  - Auto-Baud Detect (ABD)
- 12-Bit A/D Converter with up to 24 Channels:
  - Auto-acquisition and Sleep operation
  - Differential Input mode of operation

TABLE 3-2: CAPACITOR SELECTION FOR CERAMIC RESONATORS

Typical Capacitor Values Used:			
Mode	**Freq.**	**OSC1**	**OSC2**
HS	8.0 MHz 16.0 MHz	27 pF 22 pF	27 pF 22 pF

Capacitor values are for design guidance only.

Different capacitor values may be required to produce acceptable oscillator operation. The user should test the performance of the oscillator over the expected VDD and temperature range for the application. Refer to the following application notes for oscillator-specific information:

- AN588, *"PIC® Microcontroller Oscillator Design Guide"*
- AN826, *"Crystal Oscillator Basics and Crystal Selection for rfPIC® and PIC® Devices"*
- AN849, *"Basic PIC® Oscillator Design"*
- AN943, *"Practical PIC® Oscillator Analysis and Design"*
- AN949, *"Making Your Oscillator Work"*

See the notes following Table 3-3 for additional information.

TABLE 3-3: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR

Osc Type	Crystal Freq.	Typical Capacitor Values Tested:	
		C1	**C2**
HS	4 MHz	27 pF	27 pF
	8 MHz	22 pF	22 pF
	20 MHz	15 pF	15 pF

Capacitor values are for design guidance only.

Different capacitor values may be required to produce acceptable oscillator operation. The user should test the performance of the oscillator over the expected VDD and temperature range for the application.

Refer to the Microchip application notes cited in Table 3-2 for oscillator-specific information. Also see the notes following this table for additional information.

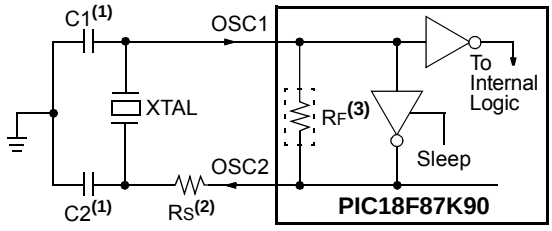
Note 1: Higher capacitance increases the stability of the oscillator but also increases the start-up time.

2: Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components.

3: Rs may be required to avoid overdriving crystals with low drive level specification.

4: Always verify oscillator performance over the VDD and temperature range that is expected for the application.

FIGURE 3-2: CRYSTAL/CERAMIC RESONATOR OPERATION (HS OR HSPLL CONFIGURATION)



Note 1: See Table 3-2 and Table 3-3 for initial values of C1 and C2.

2: A series resistor (RS) may be required for AT strip cut crystals.

3: RF varies with the oscillator mode chosen.

3.5 RC Oscillator

For timing-insensitive applications, the RC and RCIO Oscillator modes offer additional cost savings. The actual oscillator frequency is a function of several factors:

- Supply voltage
- Values of the external resistor (REXT) and capacitor (CEXT)
- Operating temperature – Given the same device, operating voltage and temperature and component values, there will also be unit-to-unit frequency variations. These are due to factors, such as:
  - Normal manufacturing variation
  - Difference in lead frame capacitance between package types (especially for low CEXT values)
  - Variations within the tolerance of limits of REXT and CEXT

3.6.2 INTPLL MODES

The 4x Phase Locked Loop (PLL) can be used with the HF-INTOSC to produce faster device clock speeds than are normally possible with the internal oscillator sources. When enabled, the PLL produces a clock speed of 32 MHz or 64 MHz.

PLL operation is controlled through software. The control bit, PLLEN (OSCTUNE<6>) is used to enable or disable its operation. Additionally, the PLL will only function when the selected HF-INTOSC frequency is either 8 MHz or 16 MHz (OSCCON<6:4> = 111 or 110).

Like the INTIO modes, there are two distinct INTPLL modes available:

- In INTPLL1 mode, the OSC2 pin outputs FOSC/4, while OSC1 functions as RA7 for digital input and output. Externally, this is identical in appearance to INTIO1 (Figure 3-8).
- In INTPLL2 mode, OSC1 functions as RA7 and OSC2 functions as RA6, both for digital input and output. Externally, this is identical to INTIO2 (Figure 3-9).

3.6.3 INTERNAL OSCILLATOR OUTPUT FREQUENCY AND TUNING

The internal oscillator block is calibrated at the factory to produce an INTOSC output frequency of 16 MHz. It can be adjusted in the user's application by writing to TUN<5:0> (OSCTUNE<5:0>) in the OSCTUNE register (Register 3-3).

When the OSCTUNE register is modified, the INTOSC (HF-INTOSC and MF-INTOSC) frequency will begin shifting to the new frequency. The oscillator will require some time to stabilize. Code execution continues during this shift and there is no indication that the shift has occurred.

The LF-INTOSC oscillator operates independently of the HF-INTOSC or the MF-INTOSC source. Any changes in the HF-INTOSC or the MF-INTOSC source, across voltage and temperature, are not necessarily reflected by changes in LF-INTOSC or vice versa. The frequency of LF-INTOSC is not affected by OSCTUNE.

3.6.4 INTOSC FREQUENCY DRIFT

The INTOSC frequency may drift as VDD or temperature changes and can affect the controller operation in a variety of ways. It is possible to adjust the INTOSC frequency by modifying the value in the OSCTUNE register. Depending on the device, this may have no effect on the LF-INTOSC clock source frequency.

Tuning INTOSC requires knowing when to make the adjustment, in which direction it should be made, and in some cases, how large a change is needed. Three compensation techniques are shown here.

3.6.4.1 Compensating with the EUSART

An adjustment may be required when the EUSART begins to generate framing errors or receives data with errors while in Asynchronous mode. Framing errors indicate that the device clock frequency is too high. To adjust for this, decrement the value in OSCTUNE to reduce the clock frequency. On the other hand, errors in data may suggest that the clock speed is too low. To compensate, increment OSCTUNE to increase the clock frequency.

3.6.4.2 Compensating with the Timers

This technique compares device clock speed to some reference clock. Two timers may be used; one timer is clocked by the peripheral clock, while the other is clocked by a fixed reference source, such as the SOSC oscillator.

Both timers are cleared, but the timer clocked by the reference source generates interrupts. When an interrupt occurs, the internally clocked timer is read and both timers are cleared. If the internally clocked timer value is much greater than expected, then the internal oscillator block is running too fast. To adjust for this, decrement the OSCTUNE register.

3.6.4.3 Compensating with the CCP Module in Capture Mode

A CCP module can use free-running Timer1 (or Timer3), clocked by the internal oscillator block and an external event with a known period (i.e., AC power frequency). The time of the first event is captured in the CCPRxH:CCPRxL registers and is recorded for use later. When the second event causes a capture, the time of the first event is subtracted from the time of the second event. Since the period of the external event is known, the time difference between events can be calculated.

If the measured time is much greater than the calculated time, the internal oscillator block is running too fast. To compensate, decrement the OSCTUNE register. If the measured time is much less than the calculated time, the internal oscillator block is running too slow. To compensate, increment the OSCTUNE register.

TABLE 6-2: PIC18F87K90 FAMILY REGISTER FILE SUMMARY (CONTINUED)

Address	File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR
F81h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx
F82h	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx
F83h	PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	xxxx xxxx
F84h	PORTE	RE7	RE6	RE5	RE4	RE3	RE2	RE1	RE0	xxxx xxxx
F85h	PORTF	RF7	RF6	RF5	RF4	RF3	RF2	RF1	—	xxxx xxx-
F86h	PORTG	—	—	RG5(1)	RG4	RG3	RG2	RG1	RG0	--xx xxxx
F87h	PORTH(2)	RH7	RH6	RH5	RH4	RH3	RH2	RH1	RH0	xxxx xxxx
F88h	PORTJ(2)	RJ7	RJ6	RJ5	RJ4	RJ3	RJ2	RJ1	RJ0	xxxx xxxx
F89h	LATA	LATA7	LATA6	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0	xxxx xxxx
F8Ah	LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	xxxx xxxx
F8Bh	LATC	LATC7	LATC6	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	xxxx xxxx
F8Ch	LATD	LATD7	LATD6	LATD5	LATD4	LATD3	LATD2	LATD1	LATD0	xxxx xxxx
F8Dh	LATE	LATE7	LATE6	LATE5	LATE4	LATE3	LATE2	LATE1	LATE0	xxxx xxxx
F8Eh	LATF	LATF7	LATF6	LATF5	LATF4	LATF3	LATF2	LATF1	—	xxxx xxx-
F8Fh	LATG	—	—	—	LATG4	LATG3	LATG2	LATG1	LATG0	---x xxxx
F90h	LATH(2)	LATH7	LATH6	LATH5	LATH4	LATH3	LATH2	LATH1	LATH0	xxxx xxxx
F91h	LATJ(2)	LATJ7	LATJ6	LATJ5	LATJ4	LATJ3	LATJ2	LATJ1	LATJ0	xxxx xxxx
F92h	TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	1111 1111
F93h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111
F94h	TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111
F95h	TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	1111 1111
F96h	TRISE	TRISE7	TRISE6	TRISE5	TRISE4	TRISE3	TRISE2	TRISE1	TRISE0	1111 1111
F97h	TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	—	1111 111-
F98h	TRISG	—	—	—	TRISG4	TRISG3	TRISG2	TRISG1	TRISG0	---1 1111
F99h	TRISH(2)	TRISH7	TRISH6	TRISH5	TRISH4	TRISH3	TRISH2	TRISH1	TRISH0	1111 1111
F9Ah	TRISJ(2)	TRISJ7	TRISJ6	TRISJ5	TRISJ4	TRISJ3	TRISJ2	TRISJ1	TRISJ0	1111 1111
F9Bh	OSCTUNE	INTSRC	PLLEN	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0	0000 0000
F9Ch	PSTR1CON	CMPL1	CMPL0	—	STRSYNC	STRD	STRC	STRB	STRA	00-0 0001
F9Dh	PIE1	—	ADIE	RC1IE	TX1IE	SSP1IE	TMR1GIE	TMR2IE	TMR1IE	-000 0000
F9Eh	PIR1	—	ADIF	RC1IF	TX1IF	SSP1IF	TMR1GIF	TMR2IF	TMR1IF	-000 0000
F9Fh	IPR1	—	ADIP	RC1IP	TX1IP	SSP1IP	TMR1GIP	TMR2IP	TMR1IP	-111 1111
FA0h	PIE2	OSCFIE	—	SSP2IE	BCL2IE	BCL1IE	HLVDIE	TMR3IE	TMR3GIE	0-10 0000
FA1h	PIR2	OSCFIF	—	SSP2IF	BCL2IF	BCL1IF	HLVDIF	TMR3IF	TMR3GIF	0-10 0000
FA2h	IPR2	OSCFIP	—	SSP2IP	BCL2IP	BCL1IP	HLVDIP	TMR3IP	TMR3GIP	1-00 1110
FA3h	PIE3	TMR5GIE	LCDIE	RC2IE	TX2IE	CTMUIE	CCP2IE	CCP1IE	RTCCIE	0000 0000
FA4h	PIR3	TMR5GIF	LCDIF	RC2IF	TX2IF	CTMUIF	CCP2IF	CCP1IF	RTCCIF	0000 0000
FA5h	IPR3	TMR5GIP	LCDIP	RC2IP	TX2IP	CTMUIP	CCP2IP	CCP1IP	RTCCIP	1111 1111
FA6h	PIR6	—	—	—	EEIF	—	CMP3IF	CMP2IF	CMP1IF	---0 -000
FA7h	—	—	—	—	—	—	—	—	—	---- ----
FA8h	HLVDCON	VDIRMAG	BGVST	IRVST	HLVDEN	HLVDL3	HLVDL2	HLVDL1	HLVDL0	0000 0000
FA9h	IPR6	—	—	—	EEIP	—	CMP3IP	CMP2IP	CMP1IP	---1 -111
FAAh	T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/$\overline{\text{T1DONE}}$	T1GVAL	T1GSS1	T1GSS0	0000 0x00
FABh	RCSTA1	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x
FACh	TXSTA1	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010
FADh	TXREG1	USART1 Transmit Register								xxxx xxxx
FAEh	RCREG1	USART1 Receive Register								0000 0000
FAFh	SPBRG1	USART1 Baud Rate Generator								0000 0000

Note 1: This bit is available when Master Clear is disabled (MCLRE = 0). When MCLRE is set, the bit is unimplemented.
2: Unimplemented in 64-pin devices (PIC18F6XK90).
3: Unimplemented in devices with a program memory of 32 Kbytes (PIC18FX5K90).

6.4 Data Addressing Modes

Note: The execution of some instructions in the core PIC18 instruction set are changed when the PIC18 extended instruction set is enabled. For more information, see **Section 6.6 "Data Memory and the Extended Instruction Set"**.

While the program memory can be addressed in only one way, through the Program Counter, information in the data memory space can be addressed in several ways. For most instructions, the addressing mode is fixed. Other instructions may use up to three modes, depending on which operands are used and whether or not the extended instruction set is enabled.

The addressing modes are:

- Inherent
- Literal
- Direct
- Indirect

An additional addressing mode, Indexed Literal Offset, is available when the extended instruction set is enabled (XINST Configuration bit = 1). For details on this mode's operation, see **Section 6.6.1 "Indexed Addressing with Literal Offset"**.

6.4.1 INHERENT AND LITERAL ADDRESSING

Many PIC18 control instructions do not need any argument at all. They either perform an operation that globally affects the device or they operate implicitly on one register. This addressing mode is known as Inherent Addressing. Examples of this mode include SLEEP, RESET and DAW.

Other instructions work in a similar way, but require an additional explicit argument in the opcode. This method is known as the Literal Addressing mode because the instructions require some literal value as an argument. Examples of this include ADDLW and MOVLW, which respectively, add or move a literal value to the W register. Other examples include CALL and GOTO, which include a 20-bit program memory address.

6.4.2 DIRECT ADDRESSING

Direct Addressing specifies all or part of the source and/or destination address of the operation within the opcode itself. The options are specified by the arguments accompanying the instruction.

In the core PIC18 instruction set, bit-oriented and byte-oriented instructions use some version of Direct Addressing by default. All of these instructions include some 8-bit literal address as their Least Significant Byte. This address specifies the instruction's data source as either a register address in one of the banks of data RAM (see **Section 6.3.3 "General Purpose Register File"**) or a location in the Access Bank (see **Section 6.3.2 "Access Bank"**).

The Access RAM bit 'a' determines how the address is interpreted. When 'a' is '1', the contents of the BSR (**Section 6.3.1 "Bank Select Register"**) are used with the address to determine the complete 12-bit address of the register. When 'a' is '0', the address is interpreted as being a register in the Access Bank. Addressing that uses the Access RAM is sometimes also known as Direct Forced Addressing mode.

A few instructions, such as MOVFF, include the entire 12-bit address (either source or destination) in their opcodes. In these cases, the BSR is ignored entirely.

The destination of the operation's results is determined by the destination bit, 'd'. When 'd' is '1', the results are stored back in the source register, overwriting its original contents. When 'd' is '0', the results are stored in the W register. Instructions without the 'd' argument have a destination that is implicit in the instruction, either the target register is being operated on or the W register.

6.4.3 INDIRECT ADDRESSING

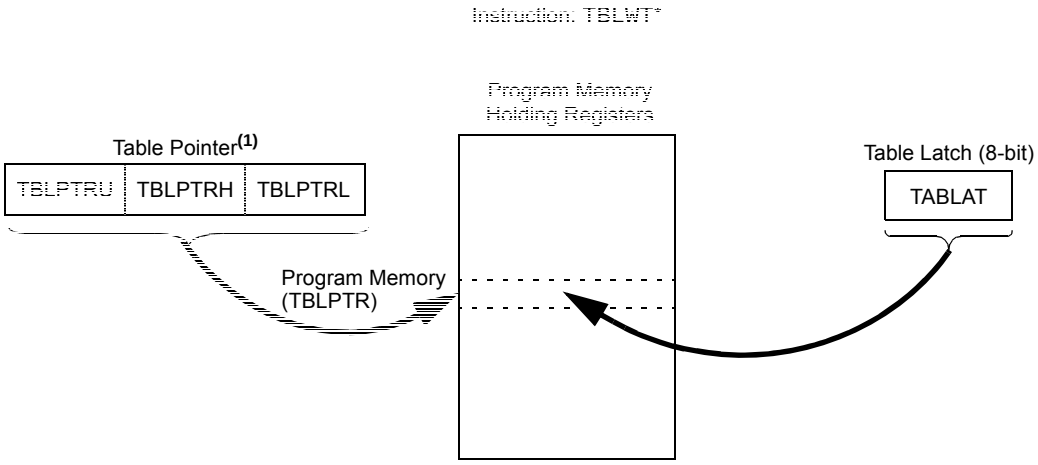
Indirect Addressing allows the user to access a location in data memory without giving a fixed address in the instruction. This is done by using File Select Registers (FSRs) as pointers to the locations to be read or written to. Since the FSRs are themselves located in RAM as Special Function Registers, they can also be directly manipulated under program control. This makes FSRs very useful in implementing data structures such as tables and arrays in data memory.

The registers for Indirect Addressing are also implemented with Indirect File Operands (INDFs) that permit automatic manipulation of the pointer value with auto-incrementing, auto-decrementing or offsetting with another value. This allows for efficient code using loops, such as the example of clearing an entire RAM bank in Example 6-5. It also enables users to perform Indexed Addressing and other Stack Pointer operations for program memory in data memory.

EXAMPLE 6-5: HOW TO CLEAR RAM (BANK 1) USING INDIRECT ADDRESSING

```
        LFSR    FSR0, 100h ;
NEXT    CLRF    POSTINC0   ; Clear INDF
                           ; register then
                           ; inc pointer
        BTFSS   FSR0H, 1   ; All done with
                           ; Bank1?
        BRA     NEXT       ; NO, clear next
CONTINUE                   ; YES, continue
```

FIGURE 7-2: TABLE WRITE OPERATION



Note 1: The Table Pointer actually points to one of 64 holding registers; the address of which is determined by TBLPTRL<5:0>. The process for physically writing data to the program memory array is discussed in Section 7.5 "Writing to Flash Program Memory".

7.2 Control Registers

Several control registers are used in conjunction with the TBLRD and TBLWT instructions. These include the:

- EECON1 register
- EECON2 register
- TABLAT register
- TBLPTR registers

7.2.1 EECON1 AND EECON2 REGISTERS

The EECON1 register (Register 7-1) is the control register for memory accesses. The EECON2 register, not a physical register, is used exclusively in the memory write and erase sequences. Reading EECON2 will read all '0's.

The EEPGD control bit determines if the access is a program or data EEPROM memory access. When clear, any subsequent operations operate on the data EEPROM memory. When set, any subsequent operations operate on the program memory.

The CFGS control bit determines if the access is to the Configuration/Calibration registers or to program memory/data EEPROM memory. When set, subsequent operations operate on Configuration registers regardless of EEPGD (see **Section 28.0 "Special Features of the CPU"**). When clear, memory selection access is determined by EEPGD.

The FREE bit, when set, allows a program memory erase operation. When FREE is set, the erase operation is initiated on the next WR command. When FREE is clear, only writes are enabled.

The WREN bit, when set, allows a write operation. On power-up, the WREN bit is clear. The WRERR bit is set in hardware when the WR bit is set and cleared when the internal programming timer expires and the write operation is complete.

Note:	During normal operation, the WRERR is read as '1'. This can indicate that a write operation was prematurely terminated by a Reset, or a write operation was attempted improperly.

The WR control bit initiates write operations. The bit cannot be cleared, only set, in software. It is cleared in hardware at the completion of the write operation.

Note:	The EEIF interrupt flag bit (PIR6<4>) is set when the write is complete. It must be cleared in software.

7.4 Erasing Flash Program Memory

The erase block is 32 words or 64 bytes for the PIC18FX5K90 and PIC18FX6K90 devices, and 64 words or 128 bytes for the PIC18FX7K90 devices. Word erase in the Flash array is not supported.

When initiating an erase sequence from the microcontroller itself, a block of 64 or 128 bytes of program memory is erased. The Most Significant 16 bits of the TBLPTR<21:6> point to the block being erased. The TBLPTR<5:0> bits are ignored.

The EECON1 register commands the erase operation. The EEPGD bit must be set to point to the Flash program memory. The WREN bit must be set to enable write operations. The FREE bit is set to select an erase operation.

For protection, the write initiate sequence for EECON2 must be used.

A long write is necessary for erasing the internal Flash. Instruction execution is halted while in a long write cycle. The long write will be terminated by the internal programming timer.

7.4.1 FLASH PROGRAM MEMORY ERASE SEQUENCE

The sequence of events for erasing a block of internal program memory location is:

1. Load the Table Pointer register with the address of the row to be erased.
2. Set the EECON1 register for the erase operation:
   - Set the EEPGD bit to point to program memory
   - Clear the CFGS bit to access program memory
   - Set the WREN bit to enable writes
   - Set the FREE bit to enable the erase
3. Disable the interrupts.
4. Write 0x55 to EECON2.
5. Write 0xAA to EECON2.
6. Set the WR bit.

   This begins the row erase cycle.

   The CPU will stall for the duration of the erase for TIW. (See Parameter D133A.)
7. Re-enable interrupts.

EXAMPLE 7-2: ERASING A FLASH PROGRAM MEMORY ROW

```
                MOVLW   CODE_ADDR_UPPER         ; load TBLPTR with the base
                MOVWF   TBLPTRU                 ; address of the memory block
                MOVLW   CODE_ADDR_HIGH
                MOVWF   TBLPTRH
                MOVLW   CODE_ADDR_LOW
                MOVWF   TBLPTRL
    ERASE_ROW
                BSF     EECON1, EEPGD           ; point to Flash program memory
                BCF     EECON1, CFGS            ; access Flash program memory
                BSF     EECON1, WREN            ; enable write to memory
                BSF     EECON1, FREE            ; enable Row Erase operation
                BCF     INTCON, GIE             ; disable interrupts
Required        MOVLW   0x55
Sequence        MOVWF   EECON2                  ; write 55h
                MOVLW   0xAA
                MOVWF   EECON2                  ; write 0AAh
                BSF     EECON1, WR              ; start erase (CPU stall)
                BSF     INTCON, GIE             ; re-enable interrupts
```

TABLE 11-14: PORTG FUNCTIONS (CONTINUED)

Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RG2/RX2/DT2/ AN18/C3INA	RG2	0	O	DIG	LATG<2> data output.
		1	I	ST	PORTG<2> data input.
	RX2	1	I	ST	Asynchronous serial receive data input (EUSART module).
	DT2	1	O	DIG	Synchronous serial data output (EUSART module); takes priority over port data.
		1	I	ST	Synchronous serial data input (EUSART module); user must configure as an input.
	AN18	1	I	ANA	A/D Input Channel 18. Default input configuration on POR; does not affect digital output.
	C3INA	x	I	ANA	Comparator 3 Input A.
RG3/CCP4/AN17/ P3D/C3INB	RG3	0	O	DIG	LATG<3> data output.
		1	I	ST	PORTG<3> data input.
	CCP4	0	O	DIG	CCP4 compare/PWM output; takes priority over port data.
		1	I	ST	CCP4 capture input.
	AN17	1	I	ANA	A/D Input Channel 17. Default input configuration on PR; does not affect digital output.
	C3INB	x	I	ANA	Comparator 3 Input B.
	P3D	0	O	—	ECCP3 PWM Output D. May be configured for tri-state during Enhanced PWM.
RG4/SEG26/ RTCC/T7CKI/ T5G/CCP5/ AN16/P1D/ C3INC	RG4	0	O	DIG	LATG<4> data output.
		1	I	ST	PORTG<4> data input.
	SEG26	1	O	ANA	LCD Segment 26 output; disables all other pin functions.
	RTCC	x	O	DIG	RTCC output.
	T7CKI	x	I	ST	Timer7 clock input.
	T5G	x	I	ST	Timer5 external clock gate input.
	CCP5	0	O	DIG	CCP5 compare/PWM output; takes priority over port data.
		1	I	ST	CCP5 capture input.
	AN16	1	I	ANA	A/D Input Channel 17. Default input configuration on POR; does not affect digital output.
	C3INC	x	I	ANA	Comparator 3 Input C.
	P1D	0	O	—	ECCP1 PWM Output D. May be configured for tri-state during Enhanced PWM.
RG5			I	ST	See the $\overline{\text{MCLR}}$/RG5 pin.

Legend: O = Output, I = Input, ANA = Analog Signal, DIG = Digital Output, ST = Schmitt Trigger Buffer Input, x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

TABLE 11-15: SUMMARY OF REGISTERS ASSOCIATED WITH PORTG

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
PORTG	—	—	RG5(1)	RG4	RG3	RG2	RG1	RG0	78
TRISG	—	—	—	TRISG4	TRISG3	TRISG2	TRISG1	TRISG0	78
LCDSE3	SE31	SE30	SE29	SE28	SE27	SE26	SE25	SE24	83
ANCON2	ANSEL23	ANSEL22	ANSEL21	ANSEL20	ANSEL19	ANSEL18	ANSEL17	ANSEL16	81
ODCON1	SSP1OD	CCP2OD	CCP1OD	—	—	—	—	SSP2OD	81
ODCON2	CCP10OD(2)	CCP9OD(2)	CCP8OD	CCP7OD	CCP6OD	CCP5OD	CCP4OD	CCP3OD	81

Legend: — = unimplemented, read as '0'. Shaded cells are not used by PORTG.

Note 1: This bit is available when Master Clear is disabled (MCLRE = 0). When MCLRE is set, the bit is unimplemented.

2: Unimplemented in devices with a program memory of 32 Kbytes (PIC18FX5K90).

TABLE 11-18: PORTJ FUNCTIONS

Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RJ0	RJ0	0	O	DIG	LATJ<0> data output.
		1	I	ST	PORTJ<0> data input.
RJ1/SEG33	RJ1	0	O	DIG	LATJ<1> data output.
		1	I	ST	PORTJ<1> data input.
	SEG33	1	O	ANA	LCD Segment 33 output; disables all other pin functions.
RJ2/SEG34	RJ2	0	O	DIG	LATJ<2> data output.
		1	I	ST	PORTJ<2> data input.
	SEG34	1	O	ANA	LCD Segment 34 output; disables all other pin functions.
RJ3/SEG35	RJ3	0	O	DIG	LATJ<3> data output.
		1	I	ST	PORTJ<3> data input.
	SEG35	1	O	ANA	LCD Segment 35 output; disables all other pin functions.
RJ4/SEG39	RJ4	0	O	DIG	LATJ<4> data output.
		1	I	ST	PORTJ<4> data input.
	SEG39	1	O	ANA	LCD Segment 39 output; disables all other pin functions.
RJ5/SEG38	RJ5	0	O	DIG	LATJ<5> data output.
		1	I	ST	PORTJ<5> data input.
	SEG38	1	O	ANA	LCD Segment 38 output; disables all other pin functions.
RJ6/SEG37	RJ6	0	O	DIG	LATJ<6> data output.
		1	I	ST	PORTJ<6> data input.
	SEG37	1	O	ANA	LCD Segment 37 output; disables all other pin functions.
RJ7/SEG36	RJ7	0	O	DIG	LATJ<7> data output.
		1	I	ST	PORTJ<7> data input.
	SEG36	1	O	ANA	LCD Segment 36 output; disables all other pin functions.

Legend: O = Output, I = Input, ANA = Analog Signal, DIG = Digital Output, ST = Schmitt Trigger Buffer Input, x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

TABLE 11-19: SUMMARY OF REGISTERS ASSOCIATED WITH PORTJ

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
PORTJ	RJ7	RJ6	RJ5	RJ4	RJ3	RJ2	RJ1	RJ0	78
LATJ	LATJ7	LATJ6	LATJ5	LATJ4	LATJ3	LATJ2	LATJ1	LATJ0	78
TRISJ	TRISJ7	TRISJ6	TRISJ5	TRISJ4	TRISJ3	TRISJ2	TRISJ1	TRISJ0	78
LCDSE4	SE39	SE38	SE37	SE36	SE35	SE34	SE33	SE32	83
PADCFG1	RDPU	REPU	RJPU(1)	—	—	RTSECSEL1	RTSECSEL0	—	80

Legend: Shaded cells are not used by PORTJ.

Note 1: Unimplemented in PIC18F6XK90 devices, read as '0'.

NOTES:

REGISTER 19-1: CCPxCON: ENHANCED CAPTURE/COMPARE/PWM x CONTROL

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PxM1	PxM0	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-6 **PxM<1:0>:** Enhanced PWM Output Configuration bits

If CCPxM<3:2> = 00, 01, 10:

xx = PxA is assigned as a capture/compare input/output; PxB, PxC and PxD are assigned as PORT pins

If CCPxM<3:2> = 11:

00 = Single output: PxA, PxB, PxC and PxD are controlled by steering (see **Section 19.4.7 "Pulse Steering Mode"**)

01 = Full-bridge output forward: PxD is modulated; PxA is active; PxB, PxC are inactive

10 = Half-bridge output: PxA, PxB are modulated with dead-band control; PxC and PxD are assigned as PORT pins

11 = Full-bridge output reverse: PxB is modulated; PxC is active; PxA and PxD are inactive

bit 5-4 **DCxB<1:0>**: PWM Duty Cycle Bit 1 and Bit 0

Capture mode:

Unused.

Compare mode:

Unused.

PWM mode:

These bits are the two LSbs of the 10-bit PWM duty cycle. The eight MSbs of the duty cycle are found in CCPRxL.

bit 3-0 **CCPxM<3:0>**: ECCPx Mode Select bits

0000 = Capture/Compare/PWM off (resets ECCPx module)

0001 = Reserved

0010 = Compare mode: toggle output on match

0011 = Capture mode

0100 = Capture mode: every falling edge

0101 = Capture mode: every rising edge

0110 = Capture mode: every fourth rising edge

0111 = Capture mode: every 16th rising edge

1000 = Compare mode: initialize the ECCPx pin low; set the output on a compare match (set CCPxIF)

1001 = Compare mode: initialize the ECCPx pin high; clear the output on a compare match (set CCPxIF)

1010 = Compare mode: generate a software interrupt only; ECCPx pin reverts to an I/O state

1011 = Compare mode: trigger special event (ECCPx resets TMR1 or TMR3, starts A/D conversion, sets CCxIF bit)

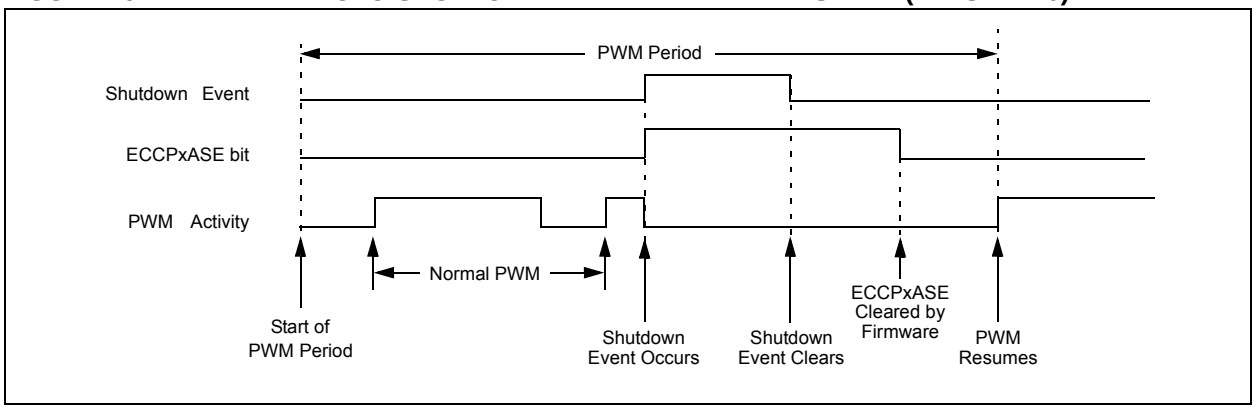
1100 = PWM mode: PxA and PxC are active-high; PxB and PxD are active-high

1101 = PWM mode: PxA and PxC are active-high; PxB and PxD are active-low

1110 = PWM mode: PxA and PxC are active-low; PxB and PxD are active-high

1111 = PWM mode: PxA and PxC are active-low; PxB and PxD are active-low

FIGURE 19-12: PWM AUTO-SHUTDOWN WITH FIRMWARE RESTART (PxRSEN = 0)



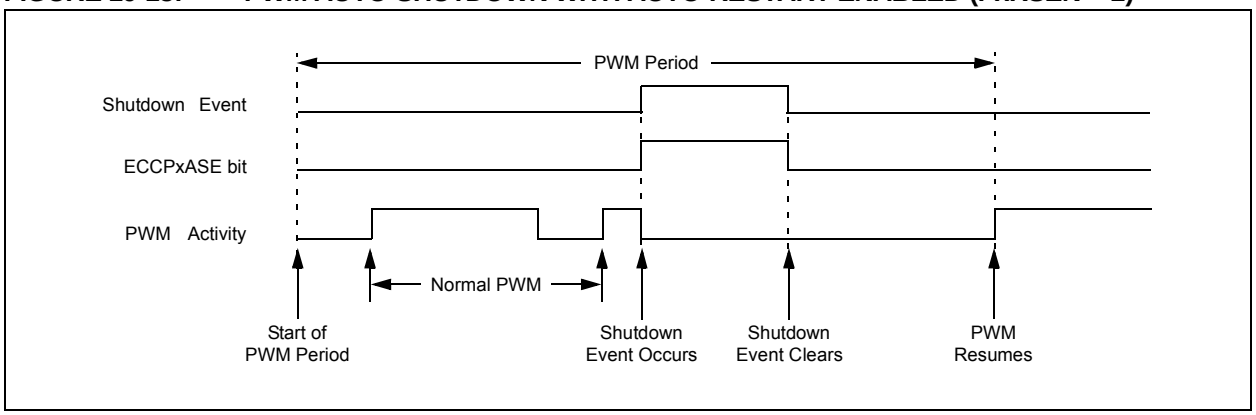
19.4.5 AUTO-RESTART MODE

The Enhanced PWM can be configured to automatically restart the PWM signal once the auto-shutdown condition has been removed. Auto-restart is enabled by setting the PxRSEN bit (ECCPxDEL<7>).

If auto-restart is enabled, the ECCPxASE bit will remain set as long as the auto-shutdown condition is active. When the auto-shutdown condition is removed, the ECCPxASE bit will be cleared via hardware and normal operation will resume.

The module will wait until the next PWM period begins, however, before re-enabling the output pin. This behavior allows the auto-shutdown with auto-restart features to be used in applications based on current mode of PWM control.

FIGURE 19-13: PWM AUTO-SHUTDOWN WITH AUTO-RESTART ENABLED (PxRSEN = 1)



20.3.2 INTERNAL RESISTOR BIASING

This mode does not use external resistors, but rather internal resistor ladders that are configured to generate the bias voltage.

The internal reference ladder actually consists of three separate ladders. Disabling the internal reference ladder disconnects all of the ladders, allowing external voltages to be supplied.

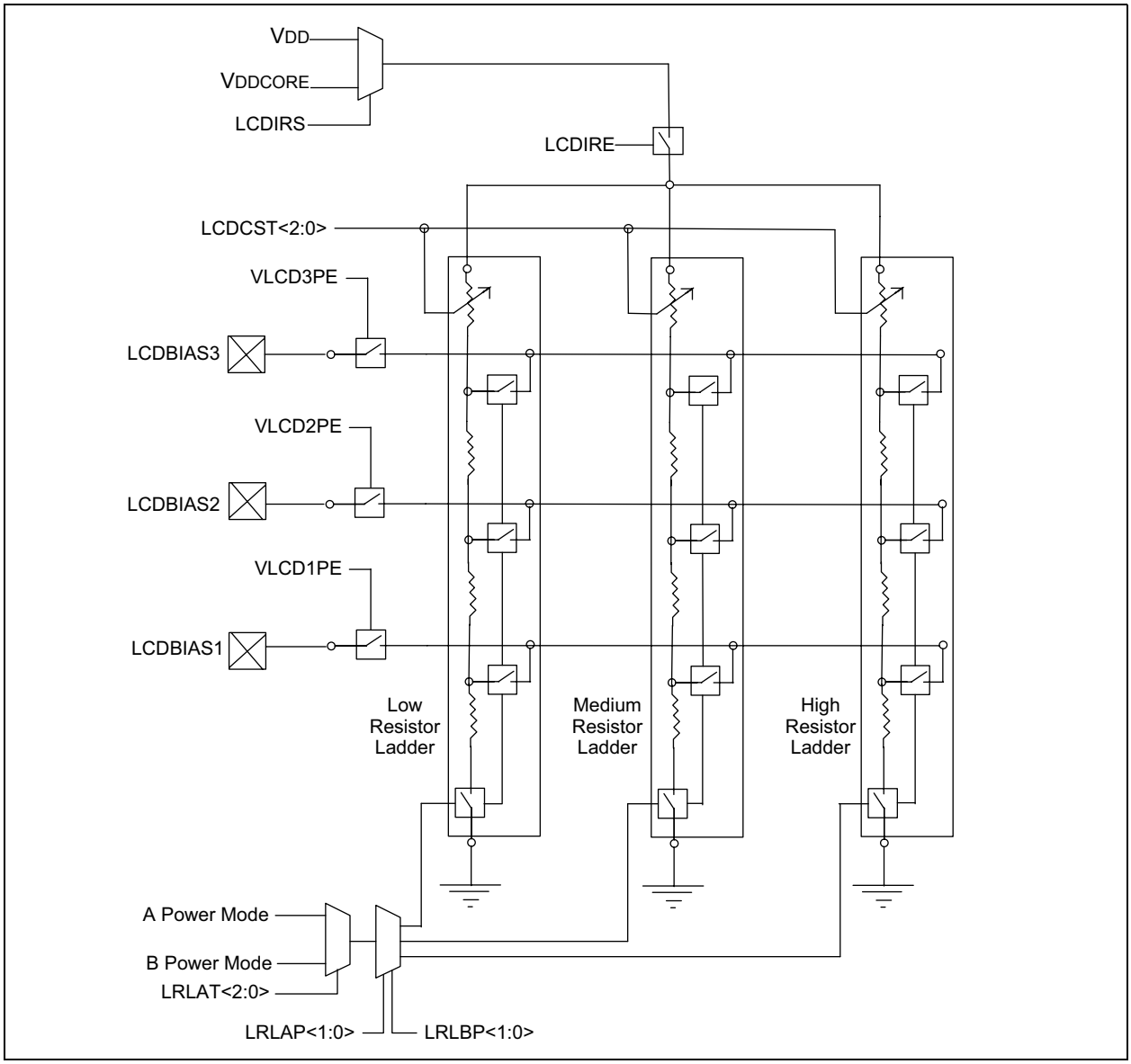
Depending on the total resistance of the resistor ladders, the biasing can be classified as low, medium or high power.

Table 20-3 shows the total resistance of each of the ladders. Figure 20-4 shows the internal resister ladder connections. When the internal resistor ladder is selected, the bias voltage can either be from VDD or from VDDCORE, depending on the LCDIRS setting.

TABLE 20-3: INTERNAL RESISTANCE LADDER POWER MODES

Power Mode	Nominal Resistance of Entire Ladder	IDD
Low	3 MΩ	1 μA
Medium	300 kΩ	10 μA
High	30 kΩ	100 μA

FIGURE 20-4: LCD BIAS INTERNAL RESISTOR LADDER CONNECTION DIAGRAM



RETURN **Return from Subroutine**

Syntax: RETURN {s}

Operands: s ∈ [0,1]

Operation: (TOS) → PC;
if s = 1,
(WS) → W,
(STATUSS) → STATUS,
(BSRS) → BSR,
PCLATU, PCLATH are unchanged

Status Affected: None

Encoding:

0000	0000	0001	001s

Description: Return from subroutine. The stack is popped and the top of the stack (TOS) is loaded into the Program Counter. If 's'= 1, the contents of the shadow registers WS, STATUSS and BSRS are loaded into their corresponding registers W, STATUS and BSR. If 's' = 0, no update of these registers occurs.

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	Process Data	POP PC from stack
No operation	No operation	No operation	No operation

Example: RETURN

After Instruction:
PC = TOS

RLCF **Rotate Left f through Carry**

Syntax: RLCF f {,d {,a}}

Operands: 0 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

Operation: (f<n>) → dest<n + 1>,
(f<7>) → C,
(C) → dest<0>

Status Affected: C, N, Z

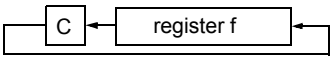
Encoding:

0011	01da	ffff	ffff

Description: The contents of register 'f' are rotated one bit to the left through the Carry flag. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f'.

If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank.

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See **Section 29.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.



Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

Example: RLCF REG, 0, 0

Before Instruction
REG = 1110 0110
C = 0
After Instruction
REG = 1110 0110
W = 1100 1100
C = 1

XORWF **Exclusive OR W with f**

Syntax: XORWF f {,d {,a}}

Operands: $0 \le f \le 255$
$d \in [0,1]$
$a \in [0,1]$

Operation: (W) .XOR. (f) → dest

Status Affected: N, Z

Encoding:

0001	10da	ffff	ffff

Description: Exclusive OR the contents of W with register 'f'. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in the register 'f'.

If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank.

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \le 95$ (5Fh). See **Section 29.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

Example: `XORWF REG, 1, 0`

Before Instruction
REG = AFh
W = B5h
After Instruction
REG = 1Ah
W = B5h

MOVSS **Move Indexed to Indexed**

Syntax: MOVSS [z_s], [z_d]

Operands: $0 \le z_s \le 127$
$0 \le z_d \le 127$

Operation: ((FSR2) + z_s) → ((FSR2) + z_d)

Status Affected: None

Encoding:

1st word (source)	1110	1011	1zzz	$zzzz_s$
2nd word (dest.)	1111	xxxx	xzzz	$zzzz_d$

Description: The contents of the source register are moved to the destination register. The addresses of the source and destination registers are determined by adding the 7-bit literal offsets, 'z_s' or 'z_d', respectively, to the value of FSR2. Both registers can be located anywhere in the 4096-byte data memory space (000h to FFFh).

The MOVSS instruction cannot use the PCL, TOSU, TOSH or TOSL as the destination register.

If the resultant source address points to an Indirect Addressing register, the value returned will be 00h. If the resultant destination address points to an Indirect Addressing register, the instruction will execute as a NOP.

Words: 2

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Determine source addr	Determine source addr	Read source reg
Decode	Determine dest addr	Determine dest addr	Write to dest reg

Example: MOVSS [05h], [06h]

Before Instruction
FSR2 = 80h
Contents of 85h = 33h
Contents of 86h = 11h
After Instruction
FSR2 = 80h
Contents of 85h = 33h
Contents of 86h = 33h

PUSHL **Store Literal at FSR2, Decrement FSR2**

Syntax: PUSHL k

Operands: $0 \le k \le 255$

Operation: k → (FSR2),
FSR2 – 1 → FSR2

Status Affected: None

Encoding:

1110	1010	kkkk	kkkk

Description: The 8-bit literal 'k' is written to the data memory address specified by FSR2. FSR2 is decremented by 1 after the operation.

This instruction allows users to push values onto a software stack.

Words: 1

Cycles: 1

Q Cycle Activity:

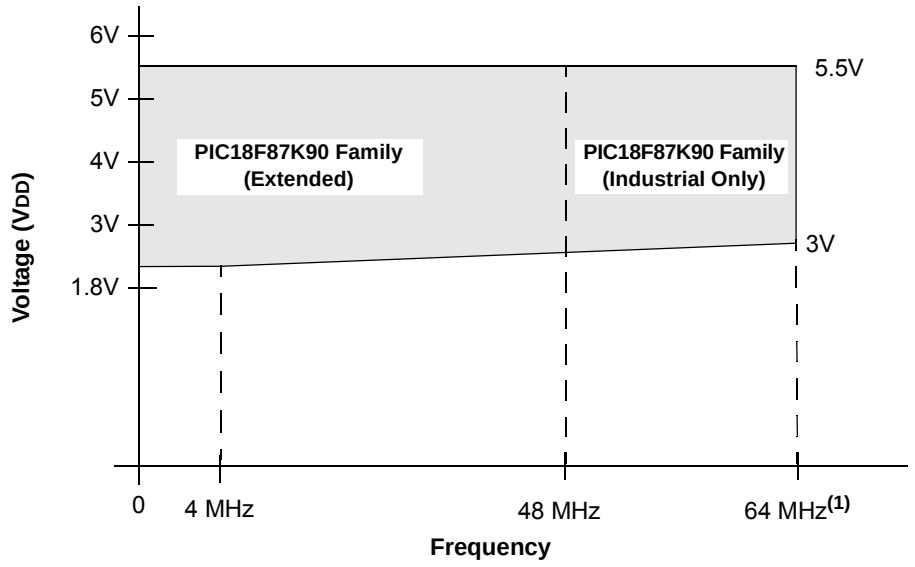
Q1	Q2	Q3	Q4
Decode	Read 'k'	Process data	Write to destination

Example: PUSHL 08h

Before Instruction
FSR2H:FSR2L = 01ECh
Memory (01ECh) = 00h

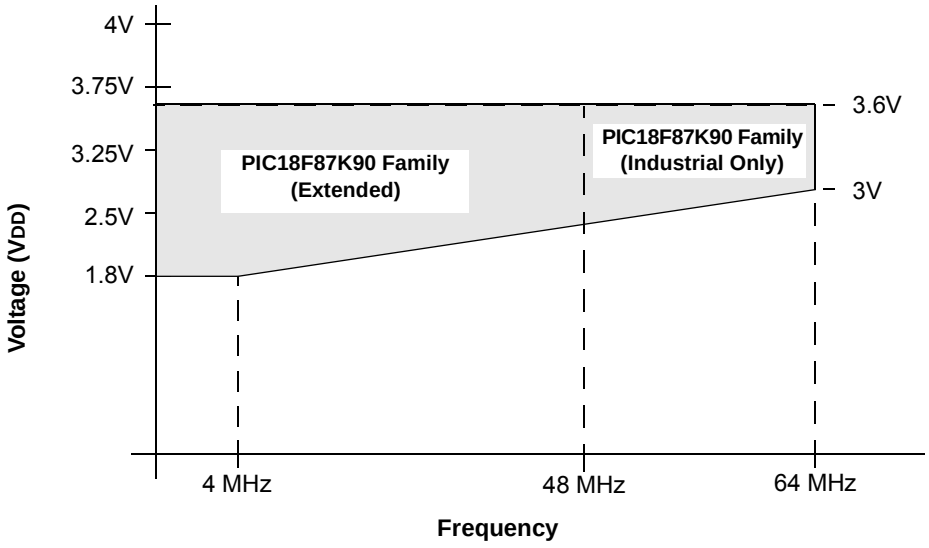
After Instruction
FSR2H:FSR2L = 01EBh
Memory (01ECh) = 08h

FIGURE 31-1: VOLTAGE-FREQUENCY GRAPH, REGULATOR ENABLED (INDUSTRIAL/EXTENDED)(1)



Note 1: FMAX = 64 MHz in all other modes. For VDD values, 1.8V to 3V, FMAX = (VDD – 1.72)/0.02 MHz.

FIGURE 31-2: VOLTAGE-FREQUENCY GRAPH, REGULATOR DISABLED (INDUSTRIAL/EXTENDED)(1,2)



Note 1: When the on-chip voltage regulator is disabled, VDD must be maintained so that VDD ≤ 3.6V.

2: For VDD values, 1.8V to 3V, FMAX = (VDD – 1.72)/0.02 MHz.

31.2 DC Characteristics: Power-Down and Supply Current PIC18F87K90 Family (Industrial/Extended) (Continued)

PIC18F87K90 Family		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended					
Param No.	**Device**	**Typ**	**Max**	**Units**	**Conditions**		
	Supply Current (IDD)(2,3)						
	All devices	5.3	10	µA	-40°C	VDD = 1.8V(4) Regulator Disabled	FOSC = 31 kHz (**RC_RUN** mode, LF-INTOSC)
		5.5	10	µA	+25°C		
		5.5	10	µA	+85°C		
		12	24	µA	+125°C		
	All devices	10	15	µA	-40°C	VDD = 3.3V(4) Regulator Disabled	
		10	16	µA	+25°C		
		11	17	µA	+85°C		
		15	35	µA	+125°C		
	All devices	70	180	µA	-40°C	VDD = 5V(5) Regulator Enabled	
		80	185	µA	+25°C		
		90	190	µA	+85°C		
		200	500	µA	+125°C		
	All devices	410	850	µA	-40°C	VDD = 1.8V(4) Regulator Disabled	FOSC = 1 MHz (**RC_RUN** mode, HF-INTOSC)
		410	800	µA	+25°C		
		410	830	µA	+85°C		
		700	1500	µA	+125°C		
	All devices	680	990	µA	-40°C	VDD = 3.3V(4) Regulator Disabled	
		680	960	µA	+25°C		
		670	950	µA	+85°C		
		800	1700	µA	+125°C		
	All devices	760	1400	µA	-40°C	VDD = 5V(5) Regulator Enabled	
		780	1400	µA	+25°C		
		800	1500	µA	+85°C		
		1200	2400	µA	+125°C		

Note 1: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in a high-impedance state and tied to VDD or VSS, and all features that add delta current are disabled (such as WDT, SOSC oscillator, BOR, etc.).

2: The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD;
$\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

3: Standard, low-cost 32 kHz crystals have an operating temperature range of -10°C to +70°C. Extended temperature crystals are available at a much higher cost.

4: Voltage regulator disabled (ENVREG = 0, tied to VSS, $\overline{RETEN}$ (CONFIG1L<0>) = 1).

5: Voltage regulator enabled (ENVREG = 1, tied to VDD, SRETEN (WDTCON<4>) = 1 and $\overline{RETEN}$ (CONFIG1L<0>) = 0).

6: LCD glass is not connected; resistor current is not included.

7: 48 MHz maximum frequency at 125°C.

INDEX

A

B