



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	I ² C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	25
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	3.6K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 24x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SSOP (0.209", 5.30mm Width)
Supplier Device Package	28-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f26k40t-i-ss

3.2 Register Definitions: Configuration Words

REGISTER 3-1: Configuration Word 1L (30 0000h): Oscillators

U-1	R/W-1	R/W-1	R/W-1	U-1	R/W-1	R/W-1	R/W-1
—	RSTOSC<2:0>			—	FEXTOSC<2:0>		
bit 7				bit 0			

Legend:

R = Readable bit W = Writable bit U = Unimplemented bit, read as '1'
 -n = Value for blank device '1' = Bit is set '0' = Bit is cleared x = Bit is unknown

bit 7 **Unimplemented:** Read as '1'

bit 6-4 **RSTOSC<2:0>:** Power-up Default Value for COSC bits
 This value is the Reset default value for COSC and selects the oscillator first used by user software. Refer to COSC operation.
 111 = EXTOSC operating per FEXTOSC bits (device manufacturing default)
 110 = HFINTOSC with HFFRQ = 4 MHz (Register 4-5) and CDIV = 4:1 (Register 4-2)
 101 = LFINTOSC
 100 = SOSC
 011 = Reserved
 010 = EXTOSC with 4x PLL, with EXTOSC operating per FEXTOSC bits
 001 = Reserved
 000 = HFINTOSC with HFFRQ = 64 MHz (Register 4-5) and CDIV = 1:1 (Register 4-2). Resets COSC/NOSC to 3'b110.

bit 3 **Unimplemented:** Read as '1'

bit 2-0 **FEXTOSC<2:0>:** FEXTOSC External Oscillator Mode Selection bits
 111 = EC (external clock) above 8 MHz; PFM set to high power (device manufacturing default)
 110 = EC (external clock) for 500 kHz to 8 MHz; PFM set to medium power
 101 = EC (external clock) below 500 kHz; PFM set to low power
 100 = Oscillator not enabled
 011 = Reserved (do not use)
 010 = HS (crystal oscillator) above 8 MHz; PFM set to high power
 001 = XT (crystal oscillator) above 500 kHz, below 8 MHz; PFM set to medium power
 000 = LP (crystal oscillator) optimized for 32.768 kHz; PFM set to low power

PIC18(L)F26/45/46K40

REGISTER 3-5: CONFIGURATION WORD 3L (30 0004h): WINDOWED WATCHDOG TIMER

U-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	WDTE<1:0>		WDTCPs<4:0>				
bit 7							bit 0

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '1'

-n = Value for blank device

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7

Unimplemented: Read as '1'

bit 6-5

WDTE<1:0>: WDT Operating Mode bits

11 = WDT enabled regardless of Sleep; SEN bit in WDTCON0 is ignored

10 = WDT enabled while Sleep = 0, suspended when Sleep = 1; SEN bit in WDTCON0 is ignored

01 = WDT enabled/disabled by SEN bit in WDTCON0

00 = WDT disabled, SEN bit in WDTCON0 is ignored

bit 4-0

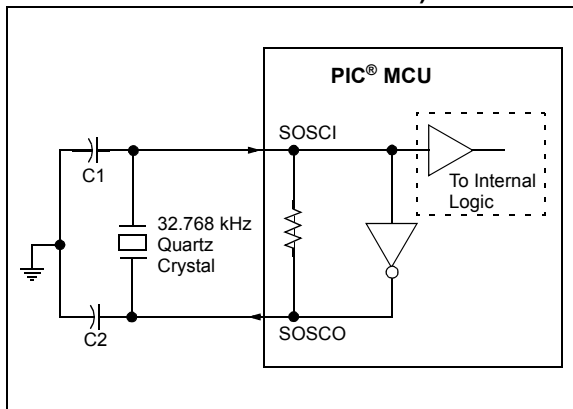
WDTCPs<4:0>: WDT Period Select bits

WDTCPs	WDTPS at POR				Software Control of WDTPS?
	Value	Divider Ratio		Typical Time Out (F _{IN} = 31 kHz)	
11111	01011	1:65536	2 ¹⁶	2s	Yes
10011	10011	1:32	2 ⁵	1 ms	No
...	...				
11110	11110				
10010	10010	1:8388608	2 ²³	256s	No
10001	10001	1:4194304	2 ²²	128s	
10000	10000	1:2097152	2 ²¹	64s	
01111	01111	1:1048576	2 ²⁰	32s	
01110	01110	1:524299	2 ¹⁹	16s	
01101	01101	1:262144	2 ¹⁸	8s	
01100	01100	1:131072	2 ¹⁷	4s	
01011	01011	1:65536	2 ¹⁶	2s	
01010	01010	1:32768	2 ¹⁵	1s	
01001	01001	1:16384	2 ¹⁴	512 ms	
01000	01000	1:8192	2 ¹³	256 ms	
00111	00111	1:4096	2 ¹²	128 ms	
00110	00110	1:2048	2 ¹¹	64 ms	
00101	00101	1:1024	2 ¹⁰	32 ms	
00100	00100	1:512	2 ⁹	16 ms	
00011	00011	1:256	2 ⁸	8 ms	
00010	00010	1:128	2 ⁷	4 ms	
00001	00001	1:64	2 ⁶	2 ms	
00000	00000	1:32	2 ⁵	1 ms	

4.3.1.5 Secondary Oscillator

The secondary oscillator is a separate oscillator block that can be used as an alternate system clock source. The secondary oscillator is optimized for 32.768 kHz, and can be used with an external crystal oscillator connected to the SOSC1 and SOSCO device pins, or an external clock source connected to the SOSCIN pin. The secondary oscillator can be selected during run-time using clock switching. Refer to **Section 4.4 “Clock Switching”** for more information.

FIGURE 4-5: QUARTZ CRYSTAL OPERATION (SECONDARY OSCILLATOR)



Note 1: Quartz crystal characteristics vary according to type, package and manufacturer. The user should consult the manufacturer data sheets for specifications and recommended application.

- 2: Always verify oscillator performance over the VDD and temperature range that is expected for the application.
- 3: For oscillator design assistance, reference the following Microchip Application Notes:

- AN826, “Crystal Oscillator Basics and Crystal Selection for PIC® and PIC® Devices” (DS00826)
- AN849, “Basic PIC® Oscillator Design” (DS00849)
- AN943, “Practical PIC® Oscillator Analysis and Design” (DS00943)
- AN949, “Making Your Oscillator Work” (DS00949)
- TB097, “Interfacing a Micro Crystal MS1V-T1K 32.768 kHz Tuning Fork Crystal to a PIC16F690/SS” (DS91097)
- AN1288, “Design Practices for Low-Power External Oscillators” (DS01288)

4.3.2 INTERNAL CLOCK SOURCES

The device may be configured to use the internal oscillator block as the system clock by performing one of the following actions:

- Program the RSTOSC<2:0> bits in Configuration Words to select the INTOSC clock source, which will be used as the default system clock upon a device Reset.
- Write the NOSC<2:0> bits in the OSCCON1 register to switch the system clock source to the internal oscillator during run-time. See **Section 4.4 “Clock Switching”** for more information.

In INTOSC mode, OSC1/CLKIN is available for general purpose I/O. OSC2/CLKOUT is available for general purpose I/O or CLKOUT.

The function of the OSC2/CLKOUT pin is determined by the CLKOUTEN bit in Configuration Words.

The internal oscillator block has two independent oscillators that can produce two internal system clock sources.

1. The **HFINTOSC** (High-Frequency Internal Oscillator) is factory-calibrated and operates from 1 to 64 MHz. The frequency of HFINTOSC can be selected through the OSCFRQ Frequency Selection register, and fine-tuning can be done via the OSCTUNE register.
2. The **LFINTOSC** (Low-Frequency Internal Oscillator) is factory-calibrated and operates at 31 kHz.

8.12 Start-up Sequence

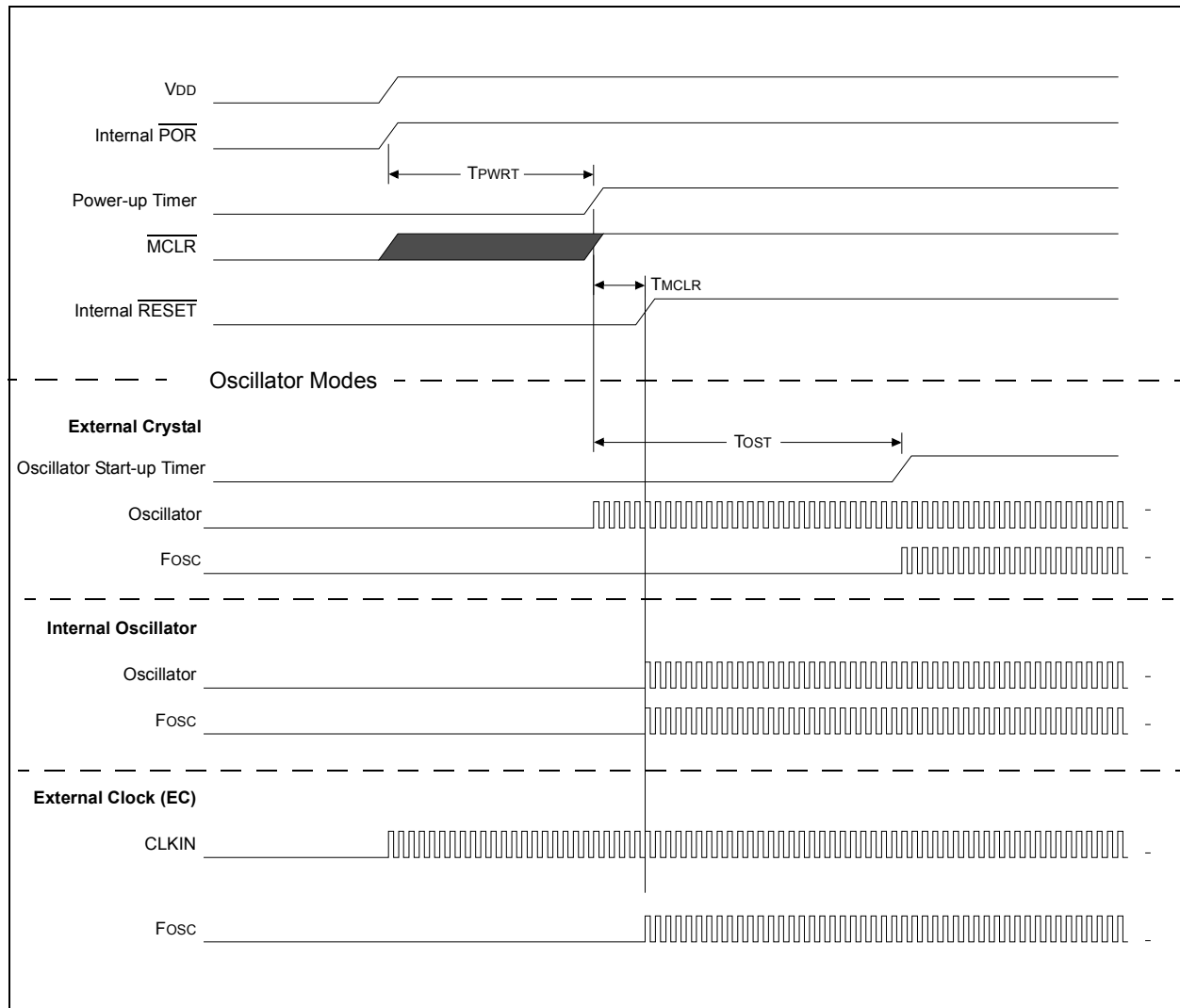
Upon the release of a POR or BOR, the following must occur before the device will begin executing:

1. Power-up Timer runs to completion (if enabled).
2. Oscillator start-up timer runs to completion (if required for selected oscillator source).
3. $\overline{\text{MCLR}}$ must be released (if enabled).

The total time out will vary based on oscillator configuration and Power-up Timer configuration. See **Section 4.0 “Oscillator Module (with Fail-Safe Clock Monitor)”** for more information.

The Power-up Timer and oscillator start-up timer run independently of $\overline{\text{MCLR}}$ Reset. If $\overline{\text{MCLR}}$ is kept low long enough, the Power-up Timer and oscillator Start-up Timer will expire. Upon bringing $\overline{\text{MCLR}}$ high, the device will begin execution after 10 F_{OSC} cycles (see Figure 8-4). This is useful for testing purposes or to synchronize more than one device operating in parallel.

FIGURE 8-4: RESET START-UP SEQUENCE



11.1.1 TABLE READS AND TABLE WRITES

In order to read and write program memory, there are two operations that allow the processor to move bytes between the program memory space and the data RAM:

- Table Read (TBLRD)
- Table Write (TBLWT)

The program memory space is 16 bits wide, while the data RAM space is eight bits wide. Table reads and table writes move data between these two memory spaces through an 8-bit register (TABLAT).

The table read operation retrieves one byte of data directly from program memory and places it into the TABLAT register. Figure 11-1 shows the operation of a table read.

The table write operation stores one byte of data from the TABLAT register into a write block holding register. The procedure to write the contents of the holding registers into program memory is detailed in **Section 11.1.6 “Writing to Program Flash Memory”**. Figure 11-2 shows the operation of a table write with program memory and data RAM.

Table operations work with byte entities. Tables containing data, rather than program instructions, are not required to be word aligned. Therefore, a table can start and end at any byte address. If a table write is being used to write executable code into program memory, program instructions will need to be word aligned.

FIGURE 11-1: TABLE READ OPERATION

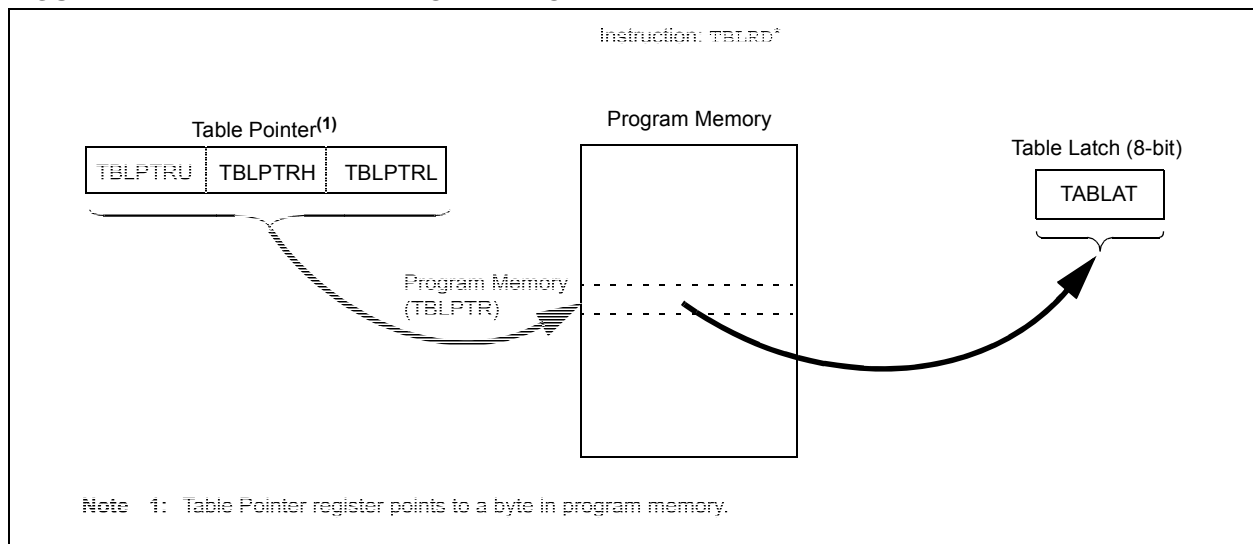
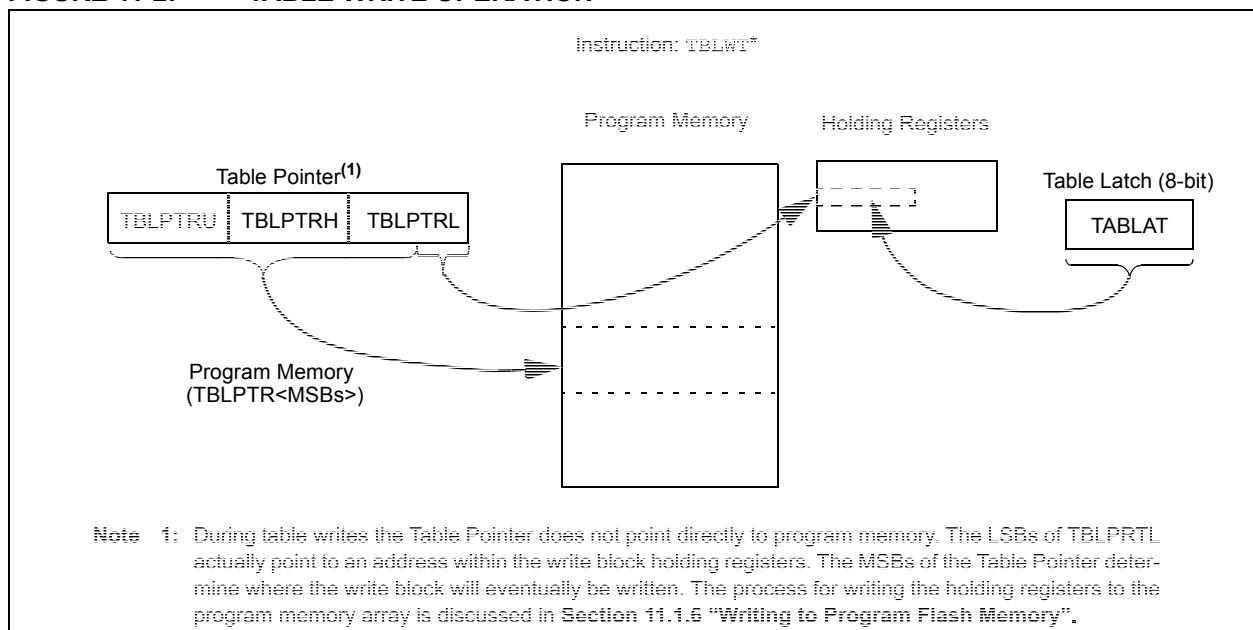


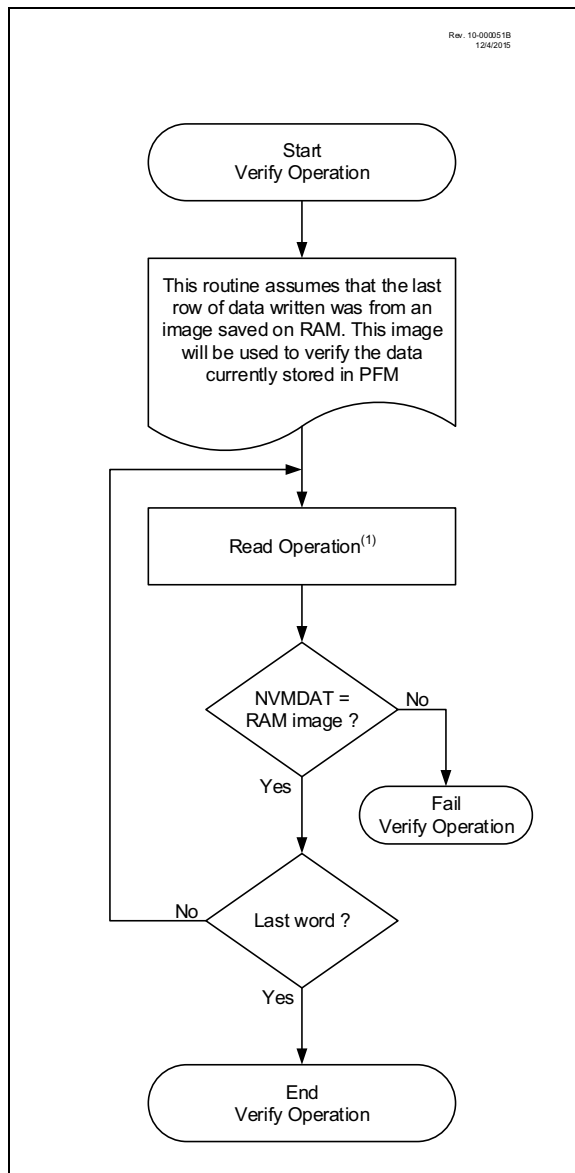
FIGURE 11-2: TABLE WRITE OPERATION



11.1.6.2 Write Verify

Depending on the application, good programming practice may dictate that the value written to the memory should be verified against the original value. This should be used in applications where excessive writes can stress bits near the specification limit. Since program memory is stored as a full page, the stored program memory contents are compared with the intended data stored in RAM after the last write is complete.

FIGURE 11-10: PROGRAM FLASH MEMORY VERIFY FLOWCHART



11.1.6.3 Unexpected Termination of Write Operation

If a write is terminated by an unplanned event, such as loss of power or an unexpected Reset, the memory location just programmed should be verified and reprogrammed if needed. If the write operation is interrupted by a MCLR Reset or a WDT Time-out Reset during normal operation, the WRERR bit will be set which the user can check to decide whether a rewrite of the location(s) is needed.

11.1.6.4 Protection Against Spurious Writes

A write sequence is valid only when both the following conditions are met, this prevents spurious writes which might lead to data corruption.

1. The WR bit is gated through the WREN bit. It is suggested to have the WREN bit cleared at all times except during memory writes. This prevents memory writes if the WR bit gets set accidentally.
2. The NVM unlock sequence must be performed each time before a write operation.

11.2 User ID, Device ID and Configuration Word Access

When NVMREG<1:0> = 0x01 or 0x11 in the NVMCON1 register, the User ID's, Device ID/ Revision ID and Configuration Words can be accessed. Different access may exist for reads and writes (see Table 11-3).

11.2.1 Reading Access

The user can read from these blocks by setting the NVMREG bits to 0x01 or 0x11. The user needs to load the address into the TBLPTR registers. Executing a TBLRD after that moves the byte pointed to the TABLAT register. The CPU operation is suspended during the read and resumes after. When read access is initiated on an address outside the parameters listed in Table 11-3, the TABLAT register is cleared, reading back '0's.

11.2.2 Writing Access

The WREN bit in NVMCON1 must be set to enable writes. This prevents accidental writes to the CONFIG words due to errant (unexpected) code execution. The WREN bit should be kept clear at all times, except when updating the CONFIG words. The WREN bit is not cleared by hardware. The WR bit will be inhibited from being set unless the WREN bit is set.

12.0 8x8 HARDWARE MULTIPLIER

12.1 Introduction

All PIC18 devices include an 8x8 hardware multiplier as part of the ALU. The multiplier performs an unsigned operation and yields a 16-bit result that is stored in the product register pair, PRODH:PRODL. The multiplier's operation does not affect any flags in the STATUS register.

Making multiplication a hardware operation allows it to be completed in a single instruction cycle. This has the advantages of higher computational throughput and reduced code size for multiplication algorithms and allows the PIC18 devices to be used in many applications previously reserved for digital signal processors. A comparison of various hardware and software multiply operations, along with the savings in memory and execution time, is shown in Table 12-1.

12.2 Operation

Example 12-1 shows the instruction sequence for an 8x8 unsigned multiplication. Only one instruction is required when one of the arguments is already loaded in the WREG register.

Example 12-2 shows the sequence to do an 8x8 signed multiplication. To account for the sign bits of the arguments, each argument's Most Significant bit (MSb) is tested and the appropriate subtractions are done.

EXAMPLE 12-1: 8x8 UNSIGNED MULTIPLY ROUTINE

```
MOVF    ARG1, W    ;
MULWF   ARG2        ; ARG1 * ARG2 ->
                        ; PRODH:PRODL
```

EXAMPLE 12-2: 8x8 SIGNED MULTIPLY ROUTINE

```
MOVF    ARG1, W    ;
MULWF   ARG2        ; ARG1 * ARG2 ->
                        ; PRODH:PRODL

BTFSC   ARG2, SB    ; Test Sign Bit
SUBWF   PRODH, F    ; PRODH = PRODH
                        ;      - ARG1

MOVF    ARG2, W    ;
BTFSC   ARG1, SB    ; Test Sign Bit
SUBWF   PRODH, F    ; PRODH = PRODH
                        ;      - ARG2
```

TABLE 12-1: PERFORMANCE COMPARISON FOR VARIOUS MULTIPLY OPERATIONS

Routine	Multiply Method	Program Memory (Words)	Cycles (Max)	Time			
				@ 64 MHz	@ 40 MHz	@ 10 MHz	@ 4 MHz
8x8 unsigned	Without hardware multiply	13	69	4.3 µs	6.9 µs	27.6 µs	69 µs
	Hardware multiply	1	1	62.5 ns	100 ns	400 ns	1 µs
8x8 signed	Without hardware multiply	33	91	5.7 µs	9.1 µs	36.4 µs	91 µs
	Hardware multiply	6	6	375 ns	600 ns	2.4 µs	6 µs
16x16 unsigned	Without hardware multiply	21	242	15.1 µs	24.2 µs	96.8 µs	242 µs
	Hardware multiply	28	28	1.8 µs	2.8 µs	11.2 µs	28 µs
16x16 signed	Without hardware multiply	52	254	15.9 µs	25.4 µs	102.6 µs	254 µs
	Hardware multiply	35	40	2.5 µs	4.0 µs	16.0 µs	40 µs

REGISTER 13-11: SCANCON0: SCANNER ACCESS CONTROL REGISTER 0

R/W-0/0	R/W/HC-0/0	R-0	R-1	R/W-0/0	U-0	R/W-0/0	R/W-0/0
SCANEN	SCANGO	BUSY	INVALID	INTM	—	MODE<1:0>	
bit 7							bit 0

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	HC = Bit is cleared by hardware

bit 7	SCANEN: Scanner Enable bit ⁽¹⁾ 1 = Scanner is enabled 0 = Scanner is disabled, internal states are reset
bit 6	SCANGO: Scanner GO bit ^(2, 3) 1 = When the CRC sends a ready signal, NVM will be accessed according to MDx and data passed to the client peripheral. 0 = Scanner operations will not occur
bit 5	BUSY: Scanner Busy Indicator bit ⁽⁴⁾ 1 = Scanner cycle is in process 0 = Scanner cycle is complete (or never started)
bit 4	INVALID: Scanner Abort Signal bit 1 = SCANLADRL/H/U has incremented to an invalid address ⁽⁶⁾ or the scanner was not setup correctly ⁽⁷⁾ 0 = SCANLADRL/H/U points to a valid address
bit 3	INTM: NVM Scanner Interrupt Management Mode Select bit <u>If MODE = 10:</u> This bit is ignored <u>If MODE = 01 (CPU is stalled until all data is transferred):</u> 1 = SCANGO is overridden (to zero) during interrupt operation; scanner resumes after returning from interrupt 0 = SCANGO is not affected by interrupts, the interrupt response will be affected <u>If MODE = 00 or 11:</u> 1 = SCANGO is overridden (to zero) during interrupt operation; scan operations resume after returning from interrupt 0 = Interrupts do not prevent NVM access
bit 2	Unimplemented: Read as '0'
bit 1-0	MODE<1:0>: Memory Access Mode bits ⁽⁵⁾ 11 = Triggered mode 10 = Peek mode 01 = Burst mode 00 = Concurrent mode

- Note 1:** Setting SCANEN = 0 (SCANCON0 register) does not affect any other register content.
- 2:** This bit is cleared when LADR > HADR (and a data cycle is not occurring).
- 3:** If INTM = 1, this bit is overridden (to zero, but not cleared) during an interrupt response.
- 4:** BUSY = 1 when the NVM is being accessed, or when the CRC sends a ready signal.
- 5:** See Table 13-2 for more detailed information.
- 6:** An invalid address can occur when the entire range of PFM is scanned and the value of LADR rolls over. An invalid address can also occur if the value in the Scan Low address registers points to a location that is not mapped in the memory map of the device.
- 7:** CRCEN and CRCGO bits must be set before setting SCANGO bit. Refer to **Section 13.9 “Program Memory Scan Configuration”**.

REGISTER 14-24: IPR6: PERIPHERAL INTERRUPT PRIORITY REGISTER 6

U-0	U-0	U-0	U-0	U-0	U-0	R/W-1/1	R/W-1/1
—	—	—	—	—	—	CCP2IP	CCP1IP
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 7-2 **Unimplemented:** Read as '0'

bit 1 **CCP2IP:** ECCP2 Interrupt Priority bit
 1 = High priority
 0 = Low priority

bit 0 **CCP1IP:** ECCP1 Interrupt Priority bit
 1 = High priority
 0 = Low priority

PIC18LF26/45/46K40

REGISTER 15-6: ODCONx: OPEN-DRAIN CONTROL REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
ODCx7	ODCx6	ODCx5	ODCx4	ODCx3	ODCx2	ODCx1	ODCx0
bit 7							bit 0

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

-n/n = Value at POR and BOR/Value at all other Resets

bit 7-0

ODCx<7:0>: Open-Drain Configuration on Pins Rx<7:0>

1 = Output drives only low-going signals (sink current only)

0 = Output drives both high-going and low-going signals (source and sink current)

TABLE 15-7: OPEN-DRAIN CONTROL REGISTERS

Name	Device		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	28 Pins	40/44 Pins								
ODCONA	X	X	ODCA7	ODCA6	ODCA5	ODCA4	ODCA3	ODCA2	ODCA1	ODCA0
ODCONB	X	X	ODCB7	ODCB6	ODCB5	ODCB4	ODCB3	ODCB2	ODCB1	ODCB0
ODCONC	X	X	ODCC7	ODCC6	ODCC5	ODCC4	ODCC3	ODCC2	ODCC1	ODCC0
ODCOND	X		—	—	—	—	—	—	—	—
		X	ODCD7	ODCD6	ODCD5	ODCD4	ODCD3	ODCD2	ODCD1	ODCD0
ODCONE	X		—	—	—	—	—	—	—	—
		X	—	—	—	—	—	ODCE2	ODCE1	ODCE0

17.8 Register Definitions: PPS Input Selection

REGISTER 17-1: xxxPPS: PERIPHERAL xxx INPUT SELECTION

U-0	U-0	U-0	R/W-m/u ⁽¹⁾	R/W-m/u ⁽¹⁾	R/W-m/u ⁽¹⁾	R/W-m/u ⁽¹⁾	R/W-m/u ⁽¹⁾
—	—	—	xxxPPS<4:0>				
bit 7			bit 0				

Legend:

R = Readable bit	W = Writable bit	-n/n = Value at POR and BOR/Value at all other Resets
u = Bit is unchanged	x = Bit is unknown	q = value depends on peripheral
'1' = Bit is set	U = Unimplemented bit, read as '0'	m = value depends on default location for that input
'0' = Bit is cleared		

bit 7-5 **Unimplemented:** Read as '0'

bit 4-3 **xxxPPS<4:3>:** Peripheral xxx Input PORTx Pin Selection bits
See Table 17-1 for the list of available ports and default pin locations.
11 = PORTD (PIC18(L)F4xK40 only)
10 = PORTC
01 = PORTB
00 = PORTA

bit 2-0 **xxxPPS<2:0>:** Peripheral xxx Input PORTx Pin Selection bits
111 = Peripheral input is from PORTx Pin 7 (Rx7)
110 = Peripheral input is from PORTx Pin 6 (Rx6)
101 = Peripheral input is from PORTx Pin 5 (Rx5)
100 = Peripheral input is from PORTx Pin 4 (Rx4)
011 = Peripheral input is from PORTx Pin 3 (Rx3)
010 = Peripheral input is from PORTx Pin 2 (Rx2)
001 = Peripheral input is from PORTx Pin 1 (Rx1)
000 = Peripheral input is from PORTx Pin 0 (Rx0)

Note 1: The Reset value 'm' of this register is determined by device default locations for that input.

19.4 Timer1/3/5 Prescaler

Timer1/3/5 has four prescaler options allowing 1, 2, 4 or 8 divisions of the clock input. The CKPS bits of the TxCON register control the prescale counter. The prescale counter is not directly readable or writable; however, the prescaler counter is cleared upon a write to TMRxH or TMRxL.

19.5 Secondary Oscillator

A secondary low-power 32.768 kHz oscillator circuit is built-in between pins SOSCI (input) and SOSCO (amplifier output). This internal circuit is to be used in conjunction with an external 32.768 kHz crystal. The secondary oscillator is not dedicated only to Timer1/3/5; it can also be used by other modules.

The oscillator circuit is enabled by setting the SOSSEN bit of the OSCEN register (Register 4-7). This can be used as the clock source to the Timer using the TMRxCLK bits. The oscillator will continue to run during Sleep.

Note: The oscillator requires a start-up and stabilization time before use. Thus, the SOSSEN bit of the OSCEN register should be set and a suitable delay observed prior to enabling Timer1/3/5. A software check can be performed to confirm if the secondary oscillator is enabled and ready to use. This is done by polling the SOR bit of the OSCSTAT (Register 4-4).

19.6 Timer1/3/5 Operation in Asynchronous Counter Mode

If control bit $\overline{\text{SYNC}}$ of the TxCON register is set, the external clock input is not synchronized. The timer increments asynchronously to the internal phase clocks. If external clock source is selected then the timer will continue to run during Sleep and can generate an interrupt on overflow, which will wake-up the processor. However, special precautions in software are needed to read/write the timer (see **Section 19.6.1 “Reading and Writing Timer1/3/5 in Asynchronous Counter Mode”**).

Note: When switching from synchronous to asynchronous operation, it is possible to skip an increment. When switching from asynchronous to synchronous operation, it is possible to produce an additional increment.

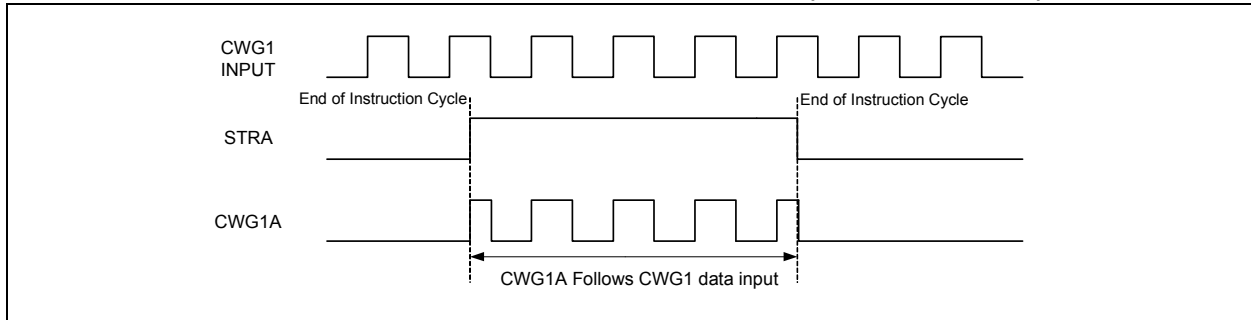
19.6.1 READING AND WRITING TIMER1/3/5 IN ASYNCHRONOUS COUNTER MODE

Reading TMRxH or TMRxL while the timer is running from an external asynchronous clock will ensure a valid read (taken care of in hardware). However, the user should keep in mind that reading the 16-bit timer in two 8-bit values itself, poses certain problems, since the timer may overflow between the reads. For writes, it is recommended that the user simply stop the timer and write the desired values. A write contention may occur by writing to the timer registers, while the register is incrementing. This may produce an unpredictable value in the TMRxH:TMRxL register pair.

24.2.4.2 Asynchronous Steering Mode

In Asynchronous mode (MODE<2:0> bits = 000, Register 24-1), steering takes effect at the end of the instruction cycle that writes to STR. In Asynchronous Steering mode, the output signal may be an incomplete waveform (Figure 24-10). This operation may be useful when the user firmware needs to immediately remove a signal from the output pin.

FIGURE 24-10: EXAMPLE OF ASYNCHRONOUS STEERING (MODE<2:0>= 000)



24.2.4.3 Start-up Considerations

The application hardware must use the proper external pull-up and/or pull-down resistors on the CWG output pins. This is required because all I/O pins are forced to high-impedance at Reset.

The POLy bits (Register 24-2) allow the user to choose whether the output signals are active-high or active-low.

24.3 Clock Source

The clock source is used to drive the dead-band timing circuits. The CWG module allows the following clock sources to be selected:

- FOSC (system clock)
- HFINTOSC

When the HFINTOSC is selected, the HFINTOSC will be kept running during Sleep. Therefore, CWG modes requiring dead band can operate in Sleep, provided that the CWG data input is also active during Sleep. The clock sources are selected using the CS bit of the CWG1CLKCON register (Register 24-3). The system clock FOSC, is disabled in Sleep and thus dead-band control cannot be used.

24.4 Selectable Input Sources

The CWG generates the output waveforms from the input sources in Table 24-1.

TABLE 24-1: SELECTABLE INPUT SOURCES

Source Peripheral	Signal Name	ISM<2:0>
CWG1PPS	Pin selected by CWG1PPS	000
CCP1	CCP1 Output	001
CCP2	CCP2 Output	010
PWM3	PWM3 Output	011
PWM4	PWM4 Output	100
CMP1	Comparator 1 Output	101
CMP2	Comparator 2 Output	110
DSM	Data signal modulator output	111

The input sources are selected using the ISM<2:0> bits in the CWG1ISM register (Register 24-4).

24.5 Output Control

24.5.1 CWG OUTPUTS

Each CWG output can be routed to a Peripheral Pin Select (PPS) output via the RxyPPS register (see **Section 17.0 “Peripheral Pin Select (PPS) Module”**).

24.5.2 POLARITY CONTROL

The polarity of each CWG output can be selected independently. When the output polarity bit is set, the corresponding output is active-high. Clearing the output polarity bit configures the corresponding output as active-low. However, polarity does not affect the override levels. Output polarity is selected with the POLY bits of the CWG1CON1. Auto-shutdown and steering options are unaffected by polarity.

24.6 Dead-Band Control

The dead-band control provides non-overlapping PWM signals to prevent shoot-through current in PWM switches. Dead-band operation is employed for Half-Bridge and Full-Bridge modes. The CWG contains two 6-bit dead-band counters. One is used for the rising edge of the input source control in Half-Bridge mode or for reverse dead-band Full-Bridge mode. The other is used for the falling edge of the input source control in Half-Bridge mode or for forward dead band in Full-Bridge mode.

Dead band is timed by counting CWG clock periods from zero up to the value in the rising or falling dead-band counter registers. See CWG1DBR and CWG1DBF registers, respectively.

24.6.1 DEAD-BAND FUNCTIONALITY IN HALF-BRIDGE MODE

In Half-Bridge mode, the dead-band counters dictate the delay between the falling edge of the normal output and the rising edge of the inverted output. This can be seen in Figure 24-2.

24.6.2 DEAD-BAND FUNCTIONALITY IN FULL-BRIDGE MODE

In Full-Bridge mode, the dead-band counters are used when undergoing a direction change. The MODE<0> bit of the CWG1CON0 register can be set or cleared while the CWG is running, allowing for changes from Forward to Reverse mode. The CWG1A and CWG1C signals will change immediately upon the first rising input edge following a direction change, but the modulated signals (CWG1B or CWG1D, depending on the direction of the change) will experience a delay dictated by the dead-band counters.

PIC18(L)F26/45/46K40

REGISTER 30-2: DAC1CON1: DAC DATA REGISTER

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	DAC1R<4:0>				
bit 7							bit 0

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

u = Bit is unchanged

x = Bit is unknown

-n/n = Value at POR and BOR/Value at all other Resets

'1' = Bit is set

'0' = Bit is cleared

bit 7-5 **Unimplemented:** Read as '0'

bit 4-0 **DAC1R<4:0>:** Data Input Register for DAC bits

TABLE 30-2: SUMMARY OF REGISTERS ASSOCIATED WITH THE DAC MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
DAC1CON0	EN	—	OE1	OE2	PSS<1:0>		—	NSS	428
DAC1CON1	—	—	—	DAC1R<4:0>					429
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		423

Legend: — = Unimplemented location, read as '0'. Shaded cells are not used with the DAC module.

PIC18(L)F26/45/46K40

TABLE 31-1: ADC CLOCK PERIOD (T_{AD}) Vs. DEVICE OPERATING FREQUENCIES^(1,4)

ADC Clock Period (T _{AD})		Device Frequency (F _{osc})						
ADC Clock Source	ADCS<5:0>	64 MHz	32 MHz	20 MHz	16 MHz	8 MHz	4 MHz	1 MHz
FOSC/2	000000	31.25 ns ⁽²⁾	62.5 ns ⁽²⁾	100 ns ⁽²⁾	125 ns ⁽²⁾	250 ns ⁽²⁾	500 ns ⁽²⁾	2.0 μs
FOSC/4	000001	62.5 ns ⁽²⁾	125 ns ⁽²⁾	200 ns ⁽²⁾	250 ns ⁽²⁾	500 ns ⁽²⁾	1.0 μs	4.0 μs
FOSC/6	000010	125 ns ⁽²⁾	187.5 ns ⁽²⁾	300 ns ⁽²⁾	375 ns ⁽²⁾	750 ns ⁽²⁾	1.5 μs	6.0 μs
FOSC/8	000011	187.5 ns ⁽²⁾	250 ns ⁽²⁾	400 ns ⁽²⁾	500 ns ⁽²⁾	1.0 μs	2.0 μs	8.0 μs ⁽³⁾
...
FOSC/16	000100	250 ns ⁽²⁾	500 ns ⁽²⁾	800 ns ⁽²⁾	1.0 μs	2.0 μs	4.0 μs	16.0 μs ⁽³⁾
...
FOSC/128	111111	2.0 μs	4.0 μs	6.4 μs	8.0 μs	16.0 μs ⁽³⁾	32.0 μs ⁽²⁾	128.0 μs ⁽²⁾
FRC	ADCS(ADCON0<4>) = 1	1.0-6.0 μs	1.0-6.0 μs	1.0-6.0 μs	1.0-6.0 μs	1.0-6.0 μs	1.0-6.0 μs	1.0-6.0 μs

Legend: Shaded cells are outside of recommended range.

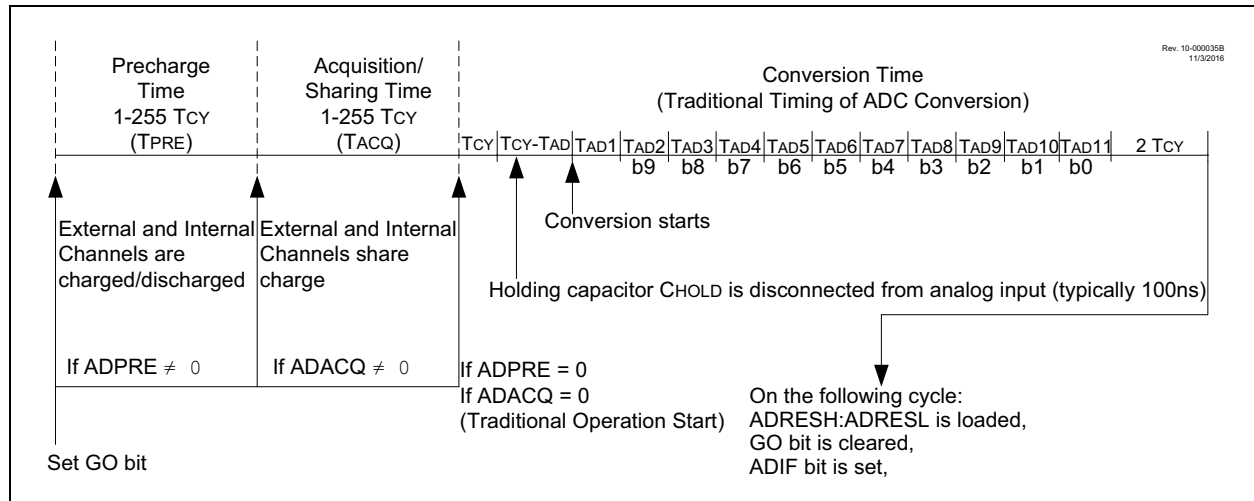
Note 1: See T_{AD} parameter for FRC source typical T_{AD} value.

2: These values violate the required T_{AD} time.

3: Outside the recommended T_{AD} time.

4: The ADC clock period (T_{AD}) and total ADC conversion time can be minimized when the ADC clock is derived from the system clock FOSC. However, the FRC oscillator source must be used when conversions are to be performed with the device in Sleep mode.

FIGURE 31-2: ANALOG-TO-DIGITAL CONVERSION T_{AD} CYCLES



REGISTER 31-19: ADRESL: ADC RESULT REGISTER LOW, ADFM = 1

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
ADRES<7:0>							
bit 7				bit 0			

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-0 **ADRES<7:0>**: ADC Result Register bits. Lower eight bits of 10-bit conversion result.

REGISTER 31-20: ADPREVH: ADC PREVIOUS RESULT REGISTER

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
ADPREV<15:8>							
bit 7				bit 0			

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-0 **ADPREV<15:8>**: Previous ADC Results bits
 If ADPSIS = 1:
 Upper byte of ADFLTR at the start of current ADC conversion
 If ADPSIS = 0:
 Upper bits of ADRES at the start of current ADC conversion⁽¹⁾

Note 1: If ADPSIS = 0, ADPREVH and ADPREVL are formatted the same way as ADRES is, depending on the ADFM bit.

PIC18(L)F26/45/46K40

FIGURE 34-2: PICKIT™ PROGRAMMER STYLE CONNECTOR INTERFACE

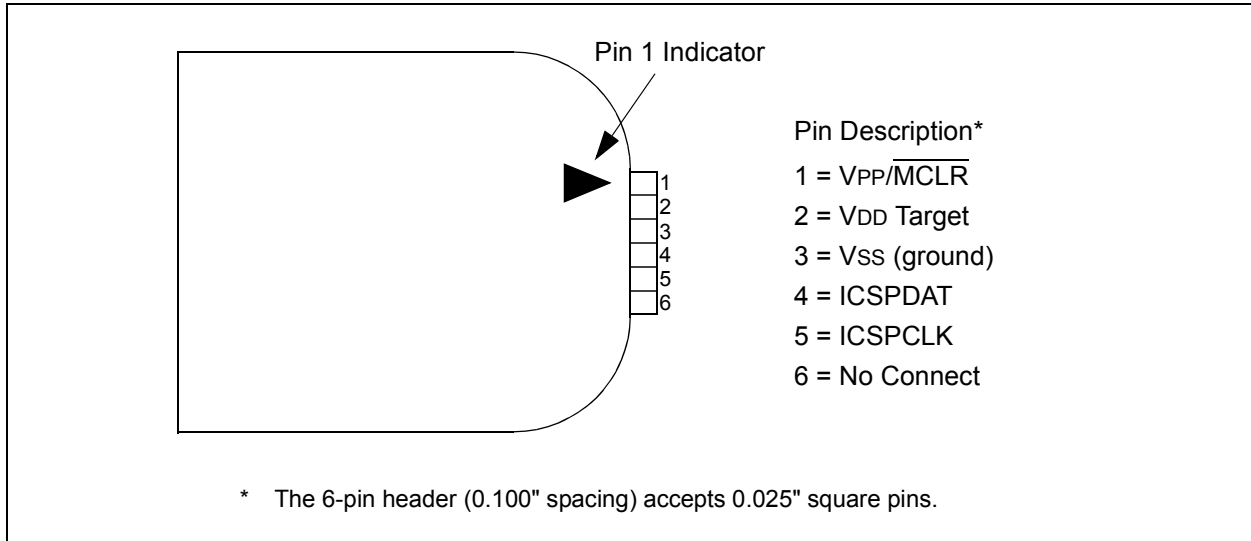
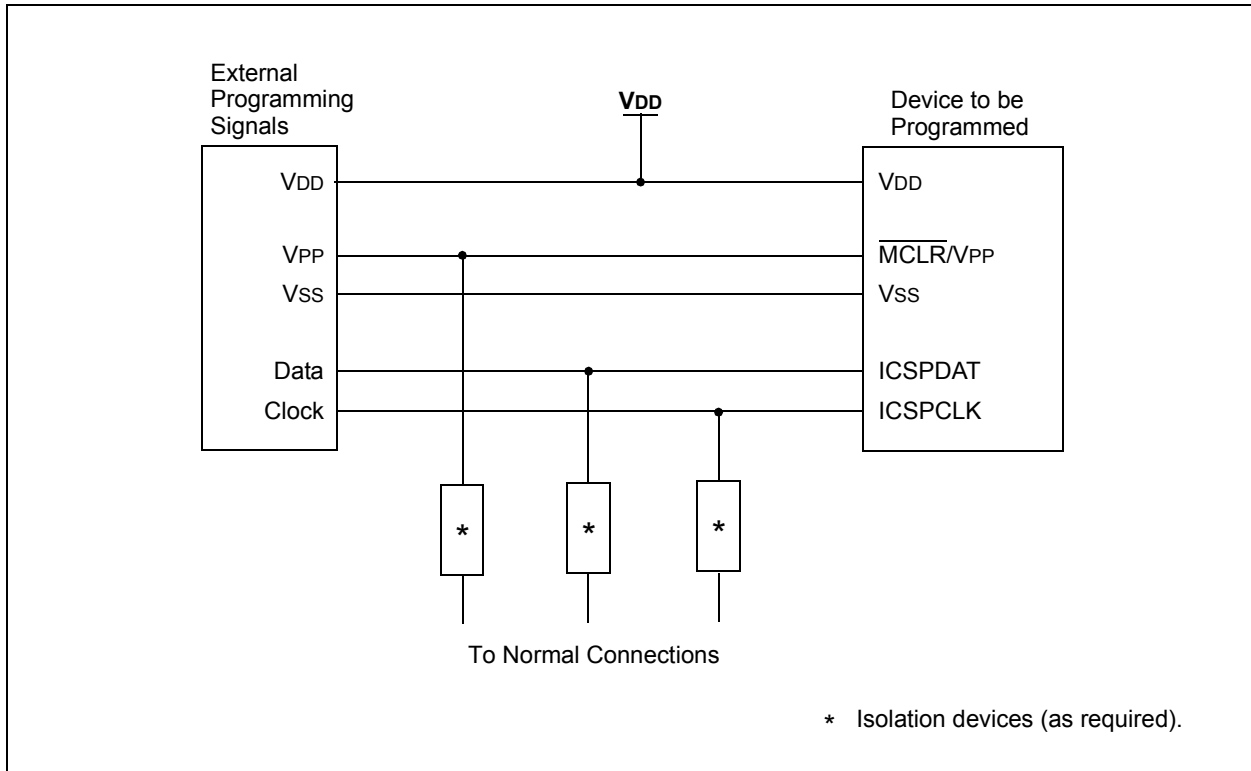


FIGURE 34-3: TYPICAL CONNECTION FOR ICSP™ PROGRAMMING



PIC18(L)F26/45/46K40

LFSR Load FSR

Syntax: LFSR f, k

Operands: $0 \leq f \leq 2$
 $0 \leq k \leq 4095$

Operation: $k \rightarrow \text{FSRf}$

Status Affected: None

Encoding:

1110	1110	00ff	$k_{11}kkk$
1111	0000	k_7kkk	$kkkk$

Description: The 12-bit literal 'k' is loaded into the File Select Register pointed to by 'f'.

Words: 2

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k' MSB	Process Data	Write literal 'k' MSB to FSRfH
Decode	Read literal 'k' LSB	Process Data	Write literal 'k' to FSRfL

Example: LFSR 2, 3ABh

After Instruction

FSR2H = 03h
 FSR2L = ABh

MOVf Move f

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

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

Operation: $f \rightarrow \text{dest}$

Status Affected: N, Z

Encoding:

0101	00da	ffff	ffff
------	------	------	------

Description: The contents of register 'f' are moved to a destination dependent upon the status of 'd'. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default). Location 'f' can be anywhere in the 256-byte bank.

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 $f \leq 95$ (5Fh). See **Section 35.2.3 "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 W

Example: MOVF REG, 0, 0

Before Instruction

REG = 22h
 W = FFh

After Instruction

REG = 22h
 W = 22h

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELoQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
== ISO/TS 16949 ==

Trademarks

The Microchip name and logo, the Microchip logo, AnyRate, AVR, AVR logo, AVR Freaks, BeaconThings, BitCloud, chipKIT, chipKIT logo, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, Helder, JukeBlox, KEELoQ, KEELoQ logo, Kleer, LANCheck, LINK MD, maXStylus, maXTouch, MediaLB, megaAVR, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, Prochip Designer, QTouch, RightTouch, SAM-BA, SpyNIC, SST, SST Logo, SuperFlash, tinyAVR, UNI/O, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

ClockWorks, The Embedded Control Solutions Company, EtherSynch, Hyper Speed Control, HyperLight Load, IntelliMOS, mTouch, Precision Edge, and Quiet-Wire are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BodyCom, CodeGuard, CryptoAuthentication, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, Inter-Chip Connectivity, JitterBlocker, KleerNet, KleerNet logo, Mindi, MiWi, motorBench, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PureSilicon, QMatrix, RightTouch logo, REAL ICE, Ripple Blocker, SAM-ICE, Serial Quad I/O, SMART-I.S., SQI, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© -2017, Microchip Technology Incorporated, All Rights Reserved.

ISBN: 978-1-5224-1639-5