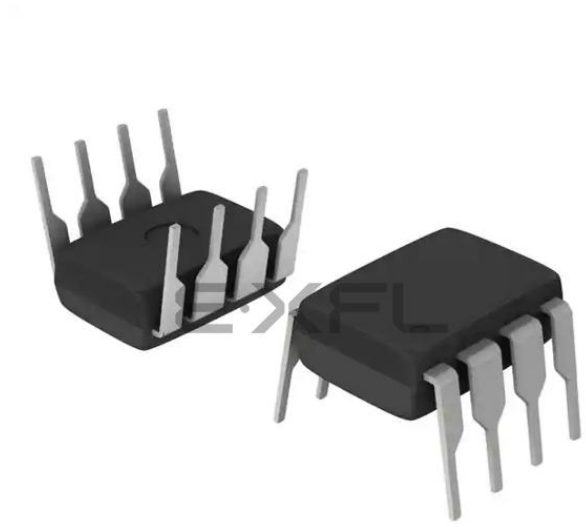


Welcome to E-XFL.COM

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	16MHz
Connectivity	-
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	3
Program Memory Size	448B (256 x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	64 x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 3x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	8-DIP (0.300", 7.62mm)
Supplier Device Package	8-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic10f320-i-p

TABLE 2-3: SPECIAL FUNCTION REGISTER SUMMARY (BANK 0)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other resets
Bank 0											
00h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx
01h	TMR0	Timer0 Module Register								xxxx xxxx	uuuu uuuu
02h	PCL	Program Counter (PC) Least Significant Byte								0000 0000	0000 0000
03h	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu
04h	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	uuuu uuuu
05h	PORTA	—	—	—	—	RA3	RA2	RA1	RA0	---- xxxx	---- uuuu
06h	TRISA	—	—	—	—	—(1)	TRISA2	TRISA1	TRISA0	---- 1111	---- 1111
07h	LATA	—	—	—	—	—	LATA2	LATA1	LATA0	---- -xxx	---- -uuu
08h	ANSELA	—	—	—	—	—	ANSA2	ANSA1	ANSA0	---- -111	---- -111
09h	WPUA	—	—	—	—	WPUA3	WPUA2	WPUA1	WPUA0	---- 1111	---- 1111
0Ah	PCLATH	—	—	—	—	—	—	—	PCLH0	---- ---0	---- ---0
0Bh	INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	0000 0000	0000 000u
0Ch	PIR1	—	ADIF	—	NCO1IF	CLC1IF	—	TMR2IF	—	-0-0 0-0-	-0-0 0-0-
0Dh	PIE1	—	ADIE	—	NCO1IE	CLC1IE	—	TMR2IE	—	-0-0 0-0-	-0-0 0-0-
0Eh	OPTION_REG	$\overline{WPUEN}$	INTEDG	T0CS	T0SE	PSA	PS<2:0>			1111 1111	uuuu uuuu
0Fh	PCON	—	—	—	—	—	—	$\overline{POR}$	$\overline{BOR}$	---- --qq	---- --uu
10h	OSCCON	—	IRCF<2:0>			HFIOFR	—	LFIOFR	HFIOFS	-110 0-00	-110 0-00
11h	TMR2	Timer2 Module Register								0000 0000	0000 0000
12h	PR2	Timer2 Period Register								1111 1111	1111 1111
13h	T2CON	—	TOUTPS<3:0>				TMR2ON	T2CKPS<1:0>		-000 0000	-000 0000
14h	PWM1DCL	PWM1DCL<1:0>		—	—	—	—	—	—	xx-- ----	uu-- ----
15h	PWM1DCH	PWM1DCH<7:0>								xxxx xxxx	uuuu uuuu
16h	PWM1CON	PWM1EN	PWM1OE	PWM1OUT	PWM1POL	—	—	—	—	0000 ----	0000 ----
17h	PWM2DCL	PWM2DCL<1:0>		—	—	—	—	—	—	xx-- ----	uu-- ----
18h	PWM2DCH	PWM2DCH<7:0>								xxxx xxxx	uuuu uuuu
19h	PWM2CON	PWM2EN	PWM2OE	PWM2OUT	PWM2POL	—	—	—	—	0000 ----	0000 ----
1Ah	IOCAP	—	—	—	—	IOCAP3	IOCAP2	IOCAP1	IOCAP0	---- 0000	---- 0000
1Bh	IOCAN	—	—	—	—	IOCAN3	IOCAN2	IOCAN1	IOCAN0	---- 0000	---- 0000
1Ch	IOCAF	—	—	—	—	IOCAF3	IOCAF2	IOCAF1	IOCAF0	---- 0000	---- 0000
1Dh	FVRCON	FVREN	FVRRDY	TSEN	TSRNG	—	—	ADFVR<1:0>		0x00 --00	0x00 --00
1Eh	ADRES	A/D Result Register								xxxx xxxx	uuuu uuuu
1Fh	ADCON	ADCS<2:0>			CHS<2:0>			GO/$\overline{DONE}$	ADON	0000 0000	0000 0000

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations are unimplemented, read as '0'.

Note 1: Unimplemented, read as '1'.

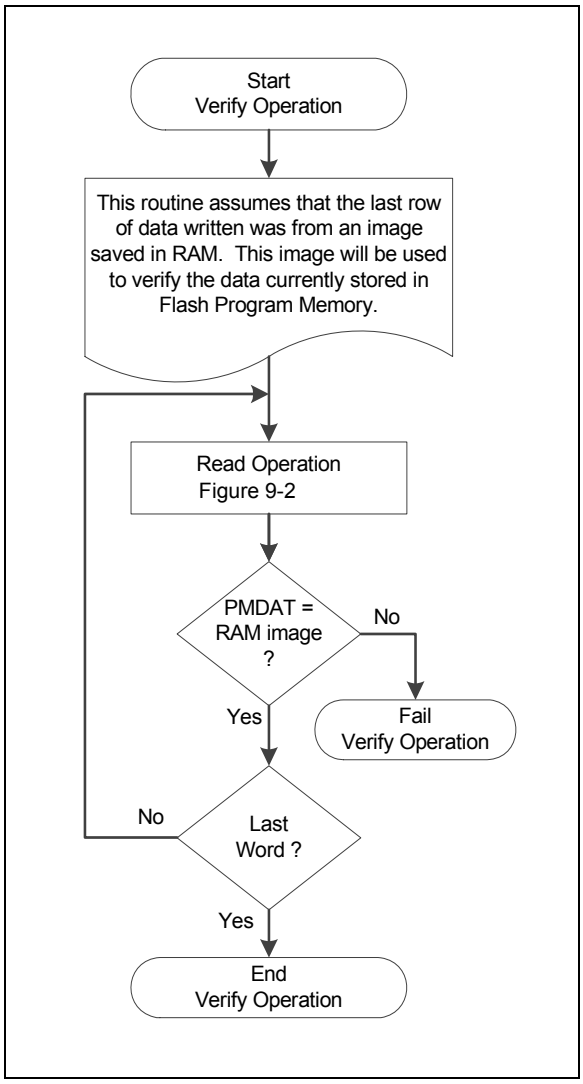
EXAMPLE 9-3: WRITING TO FLASH PROGRAM MEMORY

```
    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
    ; This write routine assumes the following:
    ;        A valid starting address (the least significant bits = '00')
    ;        is loaded in ADDRH:ADDRL
    ;        ADDRH, ADDRL and DATADDR are all located in data memory
    ;
   BANKSEL         PMADRH
   MOVF    ADDRH,W      ;Load initial address
   MOVWF   PMADRH       ;
   MOVF    ADDRL,W      ;
   MOVWF   PMADRL       ;
   MOVF    DATAADDR,W   ;Load initial data address
   MOVWF   FSR          ;
LOOP MOVF  INDF,W       ;Load first data byte into lower
   MOVWF   PMDATL       ;
   INCF    FSR,F        ;Next byte
   MOVF    INDF,W       ;Load second data byte into upper
   MOVWF   PMDATH       ;
   INCF    FSR,F        ;
   BANKSEL PMCON1
   BSF     PMCON1,WREN  ;Enable writes
   BCF     INTCON,GIE   ;Disable interrupts (if using)
   BTFSC   INTCON,GIE   ;See AN576
   GOTO    $-2
   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
   ;       Required Sequence
   MOVLW   55h          ;Start of required write sequence:
   MOVWF   PMCON2       ;Write 55h
   MOVLW   0AAh         ;
   MOVWF   PMCON2       ;Write 0AAh
   BSF     PMCON1,WR    ;Set WR bit to begin write
   NOP                  ;Required to transfer data to the buffer
   NOP                  ;registers
   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
   BCF     PMCON1,WREN  ;Disable writes
   BSF     INTCON,GIE   ;Enable interrupts (comment out if not using interrupts)
   BANKSEL PMADRL
   MOVF    PMADRL, W
   INCF    PMADRL,F     ;Increment address
   ANDLW   0x03         ;Indicates when sixteen words have been programmed
   SUBLW   0x03         ;Change value for different size write blocks
                        ;0x0F = 16 words
                        ;0x0B = 12 words
                        ;0x07 = 8 words
                        ;0x03 = 4 words
   BTFSS   STATUS,Z     ;Exit on a match,
   GOTO    LOOP         ;Continue if more data needs to be written
```

9.5 Write Verify

It is considered good programming practice to verify that program memory writes agree with the intended value. Since program memory is stored as a full page then the stored program memory contents are compared with the intended data stored in RAM after the last write is complete.

FIGURE 9-8: FLASH PROGRAM MEMORY VERIFY FLOWCHART



REGISTER 9-5: PMCON1: PROGRAM MEMORY CONTROL 1 REGISTER

U-1[1]	R/W-0/0	R/W-0/0	R/W/HC-0/0	R/W/HC-0/q[2]	R/W-0/0	R/S/HC-0/0	R/S/HC-0/0
—	CFGS	LWLO	FREE	WRERR	WREN	WR	RD
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
S = Bit can only be set	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 **Unimplemented:** Read as '1'

bit 6 **CFGS:** Configuration Select bit
- 1 = Access Configuration, User ID and Device ID Registers
- 0 = Access Flash program memory

bit 5 **LWLO:** Load Write Latches Only bit[3]
- 1 = Only the addressed program memory write latch is loaded/updated on the next WR command
- 0 = The addressed program memory write latch is loaded/updated and a write of all program memory write latches will be initiated on the next WR command

bit 4 **FREE:** Program Flash Erase Enable bit
- 1 = Performs an erase operation on the next WR command (hardware cleared upon completion)
- 0 = Performs an write operation on the next WR command

bit 3 **WRERR:** Program/Erase Error Flag bit
- 1 = Condition indicates an improper program or erase sequence attempt or termination (bit is set automatically on any set attempt (write '1') of the WR bit).
- 0 = The program or erase operation completed normally.

bit 2 **WREN:** Program/Erase Enable bit
- 1 = Allows program/erase cycles
- 0 = Inhibits programming/erasing of program Flash

bit 1 **WR:** Write Control bit
- 1 = Initiates a program Flash program/erase operation.
  The operation is self-timed and the bit is cleared by hardware once operation is complete.
  The WR bit can only be set (not cleared) in software.
- 0 = Program/erase operation to the Flash is complete and inactive.

bit 0 **RD:** Read Control bit
- 1 = Initiates a program Flash read. Read takes one cycle. RD is cleared in hardware. The RD bit can only be set (not cleared) in software.
- 0 = Does not initiate a program Flash read.

Note 1: Unimplemented bit, read as '1'.
2: The WRERR bit is automatically set by hardware when a program memory write or erase operation is started (WR = 1).
3: The LWLO bit is ignored during a program memory erase operation (FREE = 1).

REGISTER 9-6: PMCON2: PROGRAM MEMORY CONTROL 2 REGISTER

W-0/0	W-0/0	W-0/0	W-0/0	W-0/0	W-0/0	W-0/0	W-0/0
Program Memory Control Register 2							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
S = Bit can only be set	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 **Flash Memory Unlock Pattern bits**

To unlock writes, a 55h must be written first, followed by an AAh, before setting the WR bit of the PMCON1 register. The value written to this register is used to unlock the writes. There are specific timing requirements on these writes.

TABLE 9-3: SUMMARY OF REGISTERS ASSOCIATED WITH FLASH PROGRAM MEMORY

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	40
PMCON1	—	CFGS	LWLO	FREE	WRERR	WREN	WR	RD	65
PMCON2	Program Memory Control Register 2								66
PMADRL	PMADR<7:0>								64
PMADRH	—	—	—	—	—	—	—	PMADR8	64
PMDATL	PMDAT<7:0>								63
PMDATH	—	—	PMDAT<13:8>						63

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by Flash program memory module.

TABLE 9-4: SUMMARY OF CONFIGURATION WORD WITH FLASH PROGRAM MEMORY

Name	Bits	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	Register on Page
CONFIG	13:8	—	—	—	WRT<1:0>		BORV	LPBOR	LVP	20
	7:0	$\overline{CP}$	$\overline{MCLR}$	$\overline{PWRTE}$	WDTE<1:0>		BOREN<1:0>		FOSC	

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by Flash program memory.

10.2 Register Definitions: PORTA

REGISTER 10-1: PORTA: PORTA REGISTER

U-0	U-0	U-0	U-0	R-x/x	R/W-x/x	R/W-x/x	R/W-x/x
—	—	—	—	RA3	RA2	RA1	RA0
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 7-4 **Unimplemented**: Read as ‘0’

bit 3-0 **RA<3:0>:** PORTA I/O Value bits (RA3 is read-only)

Note 1: Writes to PORTx are actually written to the corresponding LATx register. Reads from PORTx register return actual I/O pin values.

REGISTER 10-2: TRISA: PORTA TRI-STATE REGISTER

U-0	U-0	U-0	U-0	U-1	R/W-1/1	R/W-1/1	R/W-1/1
—	—	—	—	—(1)	TRISA2	TRISA1	TRISA0
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 7-4 **Unimplemented**: Read as ‘0’.

bit 3 **Unimplemented**: Read as ‘1’.

bit 2-0 **TRISA<2:0>:** RA<2:0> Port I/O Tri-State Control bits

1 = Port output driver is disabled
0 = Port output driver is enabled

Note 1: Unimplemented, read as ‘1’.

REGISTER 10-3: LATA: PORTA DATA LATCH REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u
—	—	—	—	—	LATA2	LATA1	LATA0
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 7-3 **Unimplemented**: Read as ‘0’.

bit 2-0 **LATA<2:0>:** RA<2:0> Output Latch Value bits

Note 1: Writes to PORTx are actually written to the corresponding LATx register. Reads from LATx register return register values, not I/O pin values.

REGISTER 10-4: ANSELA: PORTA ANALOG SELECT REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-1/1	R/W-1/1	R/W-1/1
—	—	—	—	—	ANSA2	ANSA1	ANSA0
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 7-3 **Unimplemented:** Read as ‘0’.

bit 2-0 **ANSA<2:0>**: Analog Select between Analog or Digital Function on Pins RA<2:0>, respectively

1 = Analog input. Pin is assigned as analog input(1). Digital Input buffer disabled.

0 = Digital I/O. Pin is assigned to port or Digital special function.

Note 1: Setting a pin to an analog input automatically disables the digital input circuitry. Weak pull-ups, if available, are unaffected. The corresponding TRIS bit must be set to Input mode by the user in order to allow external control of the voltage on the pin.

11.0 INTERRUPT-ON-CHANGE

The PORTA pins can be configured to operate as Interrupt-On-Change (IOC) pins. An interrupt can be generated by detecting a signal that has either a rising edge or a falling edge. Any individual PORTA pin, or combination of PORTA pins, can be configured to generate an interrupt. The Interrupt-on-change module has the following features:

- Interrupt-on-Change enable (Master Switch)
- Individual pin configuration
- Rising and falling edge detection
- Individual pin interrupt flags

Figure 11-1 is a block diagram of the IOC module.

11.1 Enabling the Module

To allow individual PORTA pins to generate an interrupt, the IOCIE bit of the INTCON register must be set. If the IOCIE bit is disabled, the edge detection on the pin will still occur, but an interrupt will not be generated.

11.2 Individual Pin Configuration

For each PORTA pin, a rising edge detector and a falling edge detector are present. To enable a pin to detect a rising edge, the associated IOCAPx bit of the IOCAP register is set. To enable a pin to detect a falling edge, the associated IOCANx bit of the IOCAN register is set.

A pin can be configured to detect rising and falling edges simultaneously by setting both the IOCAPx bit and the IOCANx bit of the IOCAP and IOCAN registers, respectively.

11.3 Interrupt Flags

The IOCAFx bits located in the IOCAF register are status flags that correspond to the interrupt-on-change pins of PORTA. If an expected edge is detected on an appropriately enabled pin, then the status flag for that pin will be set, and an interrupt will be generated if the IOCIE bit is set. The IOCIF bit of the INTCON register reflects the status of all IOCAFx bits.

11.4 Clearing Interrupt Flags

The individual status flags, (IOCAFx bits), can be cleared by resetting them to zero. If another edge is detected during this clearing operation, the associated status flag will be set at the end of the sequence, regardless of the value actually being written.

In order to ensure that no detected edge is lost while clearing flags, only AND operations masking out known changed bits should be performed. The following sequence is an example of what should be performed.

EXAMPLE 11-1: CLEARING INTERRUPT FLAGS

```
MOVLW  0xff
XORWF  IOCAF, W
ANDWF  IOCAF, F
```

11.5 Operation in Sleep

The interrupt-on-change interrupt sequence will wake the device from Sleep mode, if the IOCIE bit is set.

If an edge is detected while in Sleep mode, the IOCAF register will be updated prior to the first instruction executed out of Sleep.

12.3 FVR Control Registers

REGISTER 12-1: FVRCON: FIXED VOLTAGE REFERENCE CONTROL REGISTER

R/W-0/0	R-q/q	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
FVREN	FVRRDY(1)	TSEN(3)	TSRNG(3)	—	—	ADFVR<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	q = Value depends on condition

bit 7 **FVREN:** Fixed Voltage Reference Enable bit
1 = Fixed Voltage Reference is enabled
0 = Fixed Voltage Reference is disabled

bit 6 **FVRRDY:** Fixed Voltage Reference Ready Flag bit(1)
1 = Fixed Voltage Reference output is ready for use
0 = Fixed Voltage Reference output is not ready or not enabled

bit 5 **TSEN:** Temperature Indicator Enable bit(3)
1 = Temperature Indicator is enabled
0 = Temperature Indicator is disabled

bit 4 **TSRNG:** Temperature Indicator Range Selection bit(3)
1 = $V_{OUT} = V_{DD} - 4V_T$ (High Range)
0 = $V_{OUT} = V_{DD} - 2V_T$ (Low Range)

bit 3-2 **Unimplemented**: Read as '0'

bit 1-0 **ADFVR<1:0>:** ADC Fixed Voltage Reference Selection bit
11 = ADC Fixed Voltage Reference Peripheral output is 4x (4.096V)(2)
10 = ADC Fixed Voltage Reference Peripheral output is 2x (2.048V)(2)
01 = ADC Fixed Voltage Reference Peripheral output is 1x (1.024V)
00 = ADC Fixed Voltage Reference Peripheral output is off.

Note 1: FVRRDY indicates the true state of the FVR.
2: Fixed Voltage Reference output cannot exceed V_{DD}.
3: See **Section 14.0 "Temperature Indicator Module"** for additional information.

TABLE 12-2: SUMMARY OF REGISTERS ASSOCIATED WITH FIXED VOLTAGE REFERENCE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	—	—	ADFVR<1:0>		78

Legend: Shaded cells are not used with the Fixed Voltage Reference.

15.2.5 A/D CONVERSION PROCEDURE

This is an example procedure for using the ADC to perform an Analog-to-Digital conversion:

1. Configure Port:
   - Disable pin output driver (Refer to the TRIS register)
   - Configure pin as analog (Refer to the ANSEL register)
   - Disable weak pull-ups either globally (Refer to the OPTION_REG register) or individually (Refer to the appropriate WPUX register)
2. Configure the ADC module:
   - Select ADC conversion clock
   - Select ADC input channel
   - Turn on ADC module
3. Configure ADC interrupt (optional):
   - Clear ADC interrupt flag
   - Enable ADC interrupt
   - Enable peripheral interrupt
   - Enable global interrupt**(1)**
4. Wait the required acquisition time**(2)**.
5. Start conversion by setting the GO/$\overline{\text{DONE}}$ bit.
6. Wait for ADC conversion to complete by one of the following:
   - Polling the GO/$\overline{\text{DONE}}$ bit
   - Waiting for the ADC interrupt (interrupts enabled)
7. Read ADC Result.
8. Clear the ADC interrupt flag (required if interrupt is enabled).

Note 1: The global interrupt can be disabled if the user is attempting to wake-up from Sleep and resume in-line code execution.

2: Refer to **Section 15.4 "A/D Acquisition Requirements"**.

15.3 ADC Register Definitions

The following registers are used to control the operation of the ADC.

REGISTER 15-1: ADCON: A/D CONTROL REGISTER 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	R/W-0/0
ADCS<2:0>			CHS<2:0>			GO/$\overline{\text{DONE}}$	ADON
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 **ADCS<2:0>:** A/D Conversion Clock Select bits
111 = FRC
110 = FOSC/64
101 = FOSC/16
100 = FOSC/4
011 = FRC
010 = FOSC/32
001 = FOSC/8
000 = FOSC/2

bit 4-2 **CHS<2:0>:** Analog Channel Select bits
111 = FVR (Fixed Voltage Reference) Buffer Output(2)
110 = Temperature Indicator(1)
101 = Reserved. No channel connected.
100 = Reserved. No channel connected.
011 = Reserved. No channel connected.
010 = AN2
001 = AN1
000 = AN0

bit 1 **GO/$\overline{\text{DONE}}$:** A/D Conversion Status bit

If ADON = 1:
1 = A/D conversion in progress (Setting this bit starts the A/D conversion)
0 = A/D conversion not in progress (This bit is automatically cleared by hardware when the A/D conversion is complete.)

If this bit is cleared while a conversion is in progress, the conversion will stop and the results of the conversion up to this point will be transferred to the result registers, but the ADIF interrupt flag bit will not be set.

If ADON = 0:
0 = A/D conversion not in progress

bit 0 **ADON:** ADC Enable bit
1 = ADC is enabled
0 = ADC is disabled and consumes no operating current

Note 1: See **Section 14.0 "Temperature Indicator Module"** for more information.
2: See **Section 12.0 "Fixed Voltage Reference (FVR)"** for more information.

18.0 PULSE-WIDTH MODULATION (PWM) MODULE

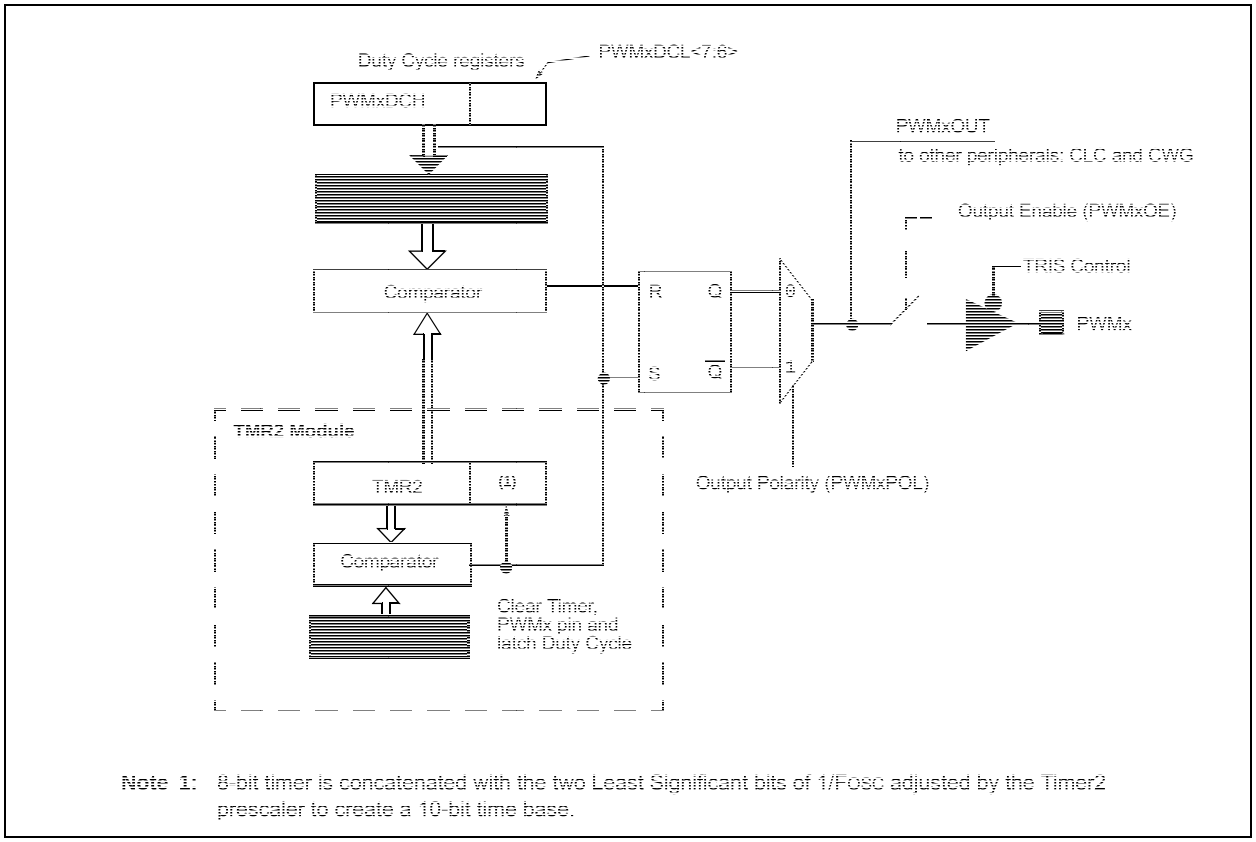
The PWM module generates a Pulse-Width Modulated signal determined by the duty cycle, period, and resolution that are configured by the following registers:

- PR2
- T2CON
- PWMxDCH
- PWMxDCL
- PWMxCON

Figure 18-1 shows a simplified block diagram of PWM operation.

Figure 18-2 shows a typical waveform of the PWM signal.

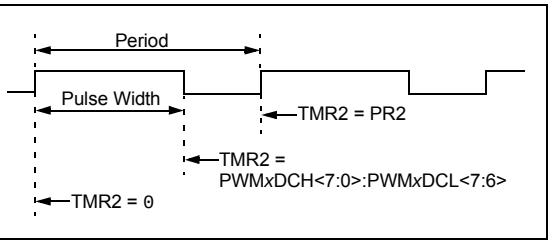
FIGURE 18-1: SIMPLIFIED PWM BLOCK DIAGRAM



Note 1: 8-bit timer is concatenated with the two Least Significant bits of 1/FOSC adjusted by the Timer2 prescaler to create a 10-bit time base.

For a step-by-step procedure on how to set up this module for PWM operation, refer to **Section 18.1.9 "Setup for PWM Operation using PWMx Pins"**.

FIGURE 18-2: PWM OUTPUT



18.1.9 SETUP FOR PWM OPERATION USING PWMx PINS

The following steps should be taken when configuring the module for PWM operation using the PWMx pins:

1. Disable the PWMx pin output driver(s) by setting the associated TRIS bit(s).
2. Clear the PWMxCON register.
3. Load the PR2 register with the PWM period value.
4. Clear the PWMxDCH register and bits <7:6> of the PWMxDCL register.
5. Configure and start Timer2:
   - Clear the TMR2IF interrupt flag bit of the PIR1 register. See Note below.
   - Configure the T2CKPS bits of the T2CON register with the Timer2 prescale value.
   - Enable Timer2 by setting the TMR2ON bit of the T2CON register.
6. Enable PWM output pin and wait until Timer2 overflows, TMR2IF bit of the PIR1 register is set. See Note below.
7. Enable the PWMx pin output driver(s) by clearing the associated TRIS bit(s) and setting the PWMxOE bit of the PWMxCON register.
8. Configure the PWM module by loading the PWMxCON register with the appropriate values.

Note 1: In order to send a complete duty cycle and period on the first PWM output, the above steps must be followed in the order given. If it is not critical to start with a complete PWM signal, then move Step 8 to replace Step 4.

2: For operation with other peripherals only, disable PWMx pin outputs.

FIGURE 21-2: TYPICAL CWG OPERATION WITH PWM1 (NO AUTO-SHUTDOWN)

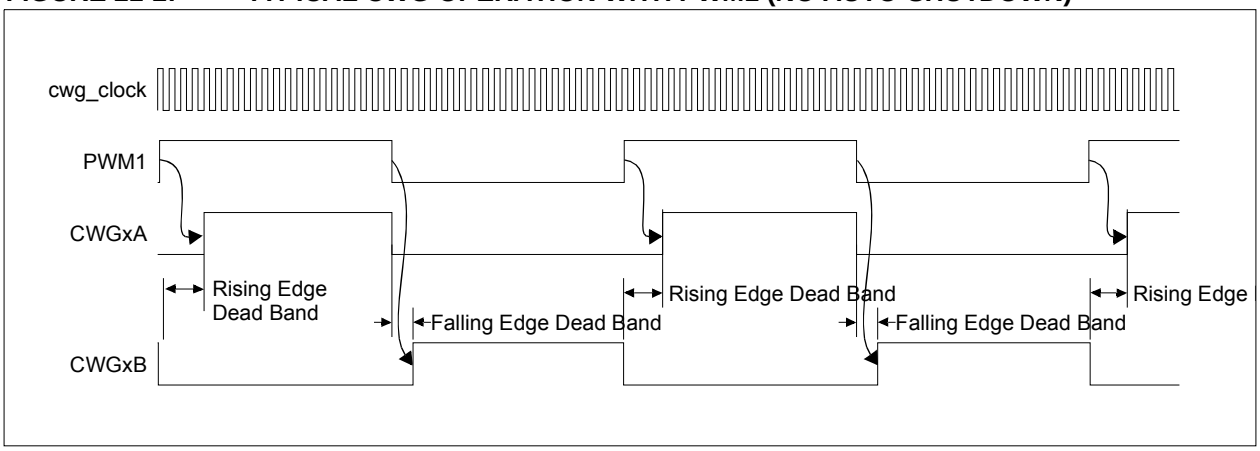


FIGURE 24-7: CLKR AND I/O TIMING

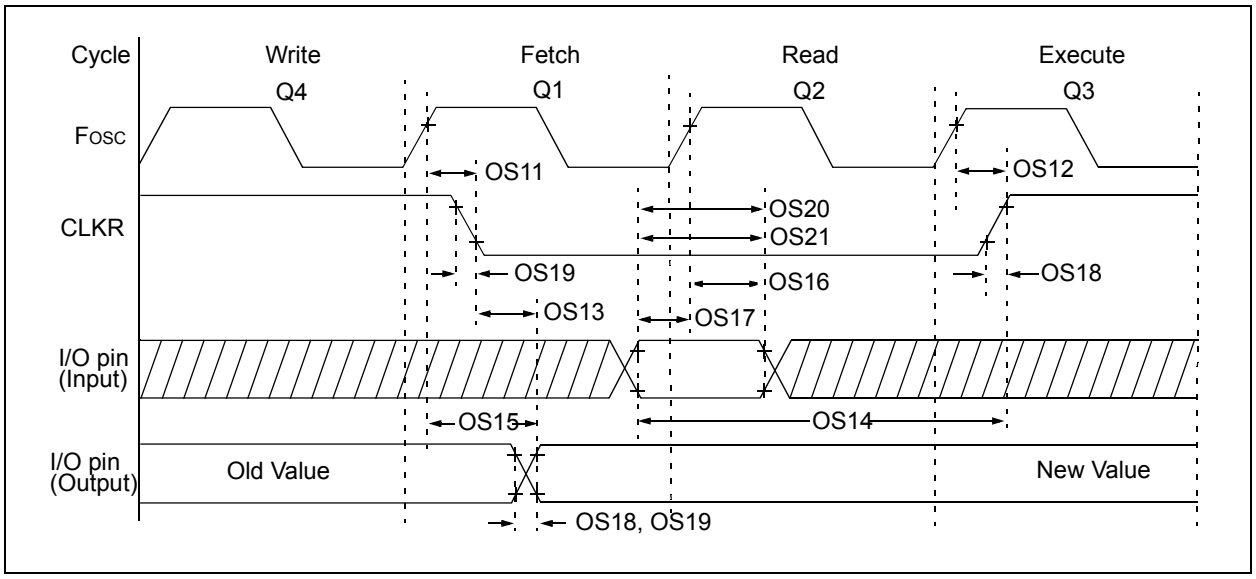


TABLE 24-8: CLKR AND I/O TIMING PARAMETERS

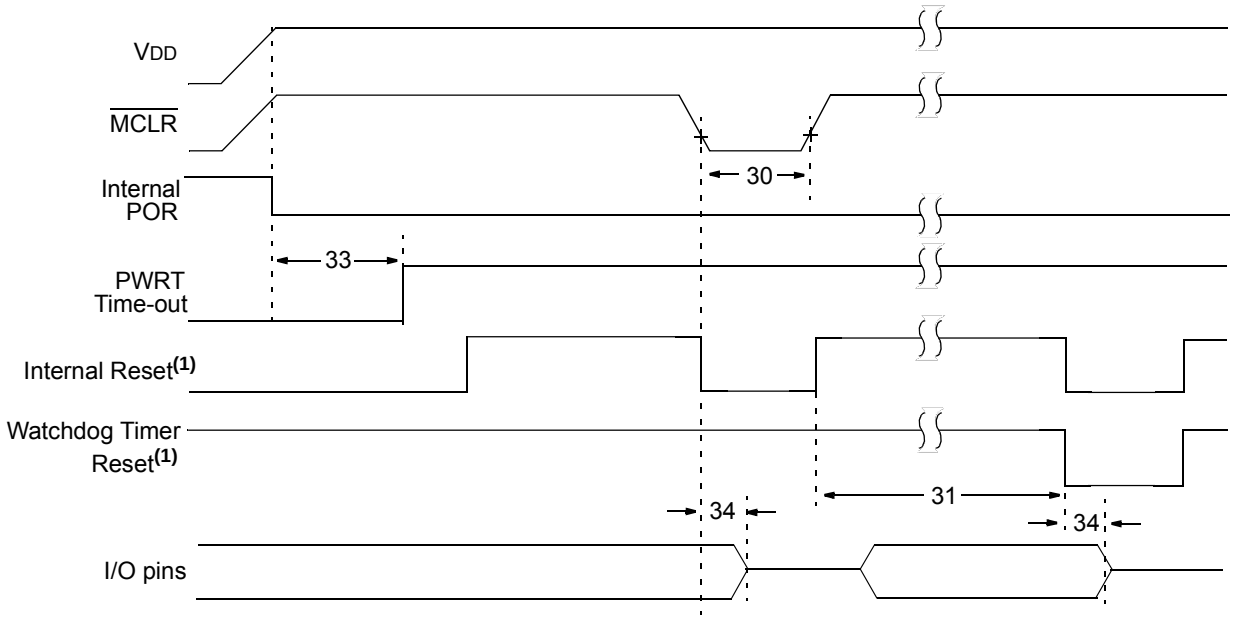
Standard Operating Conditions (unless otherwise stated)							
Param. No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
OS11	TosH2ckL	FOSC↑ to CLKOUT↓(1)	—	—	70	ns	3.3V ≤ VDD ≤ 5.0V
OS12	TosH2ckH	FOSC↑ to CLKOUT↑(1)	—	—	72	ns	3.3V ≤ VDD ≤ 5.0V
OS13	TckL2ioV	CLKOUT↓ to Port out valid(1)	—	—	20	ns	
OS14	TioV2ckH	Port input valid before CLKOUT↑(1)	TOSC + 200 ns	—	—	ns	
OS15	TosH2ioV	Fosc↑ (Q1 cycle) to Port out valid	—	50	70*	ns	3.3V ≤ VDD ≤ 5.0V
OS16	TosH2ioI	Fosc↑ (Q2 cycle) to Port input invalid (I/O in setup time)	50	—	—	ns	3.3V ≤ VDD ≤ 5.0V
OS17	TioV2osH	Port input valid to Fosc↑ (Q2 cycle) (I/O in setup time)	20	—	—	ns	
OS18*	TioR	Port output rise time	— —	40 15	72 32	ns	VDD = 1.8V 3.3V ≤ VDD ≤ 5.0V
OS19*	TioF	Port output fall time	— —	28 15	55 30	ns	VDD = 1.8V 3.3V ≤ VDD ≤ 5.0V
OS20*	Tinp	INT pin input high or low time	25	—	—	ns	
OS21*	Tioc	Interrupt-on-change new input level time	25	—	—	ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated.

Note 1: Measurements are taken in EXTRC mode where CLKOUT output is 4 x TOSC.

FIGURE 24-8: RESET, WATCHDOG TIMER, AND POWER-UP TIMER TIMING



Note 1: Asserted low.

FIGURE 24-9: BROWN-OUT RESET TIMING AND CHARACTERISTICS

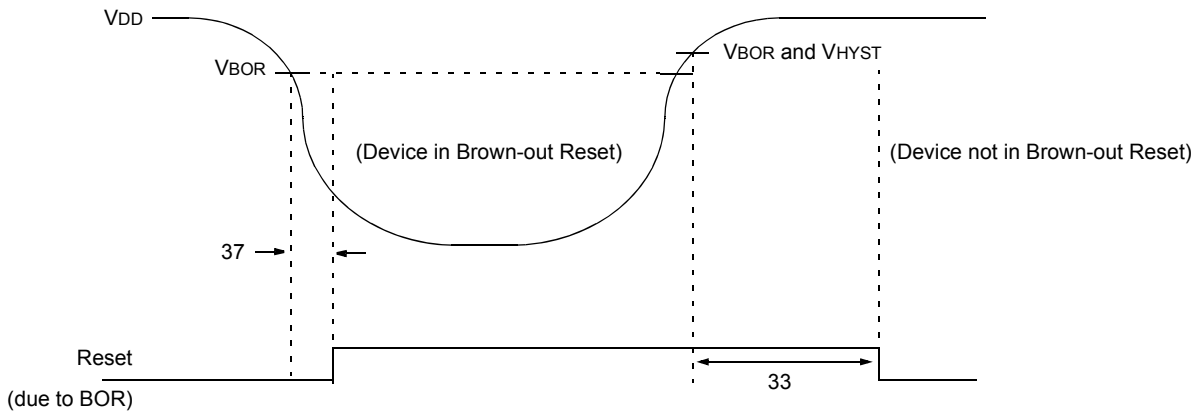
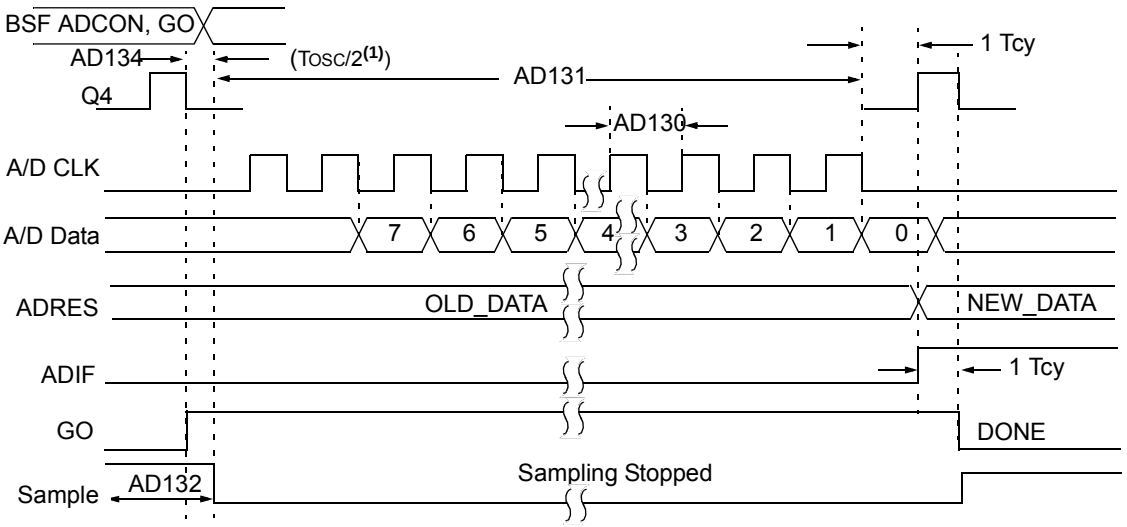
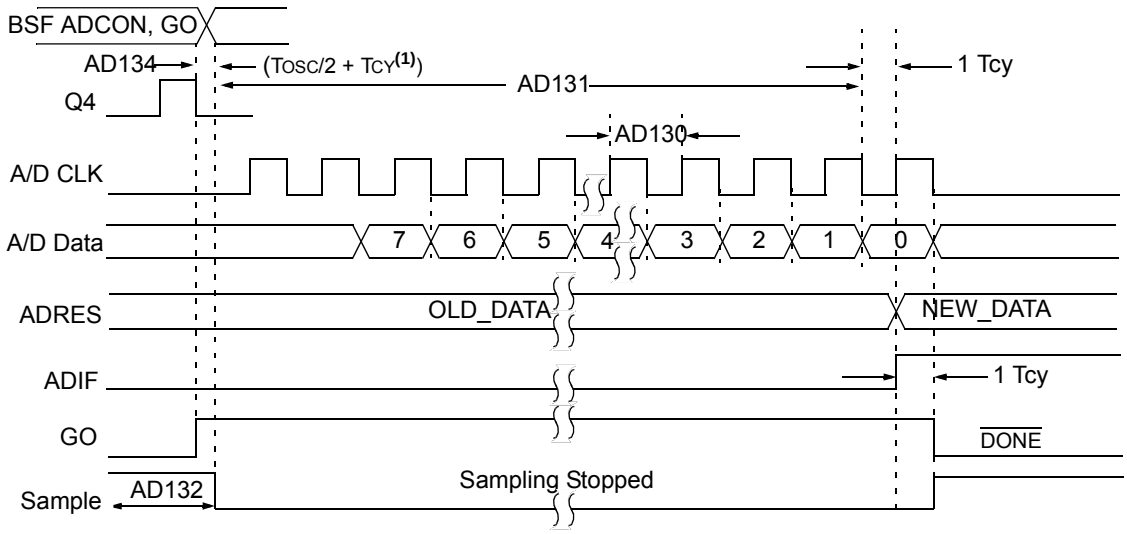


FIGURE 24-12: A/D CONVERSION TIMING (NORMAL MODE)



Note 1: If the A/D clock source is selected as FRC, a time of TCY is added before the A/D clock starts. This allows the SLEEP instruction to be executed.

FIGURE 24-13: A/D CONVERSION TIMING (SLEEP MODE)



Note 1: If the A/D clock source is selected as FRC, a time of TCY is added before the A/D clock starts. This allows the SLEEP instruction to be executed.

26.2 MPLAB XC Compilers

The MPLAB XC Compilers are complete ANSI C compilers for all of Microchip's 8, 16, and 32-bit MCU and DSC devices. These compilers provide powerful integration capabilities, superior code optimization and ease of use. MPLAB XC Compilers run on Windows, Linux or MAC OS X.

For easy source level debugging, the compilers provide debug information that is optimized to the MPLAB X IDE.

The free MPLAB XC Compiler editions support all devices and commands, with no time or memory restrictions, and offer sufficient code optimization for most applications.

MPLAB XC Compilers include an assembler, linker and utilities. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. MPLAB XC Compiler uses the assembler to produce its object file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

26.3 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code, and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB X IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multipurpose source files
- Directives that allow complete control over the assembly process

26.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

26.5 MPLAB Assembler, Linker and Librarian for Various Device Families

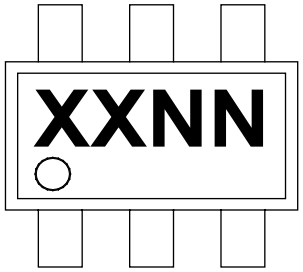
MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC DSC devices. MPLAB XC Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

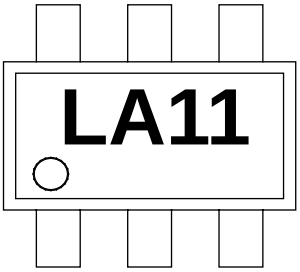
27.0 PACKAGING INFORMATION

27.1 Package Marking Information

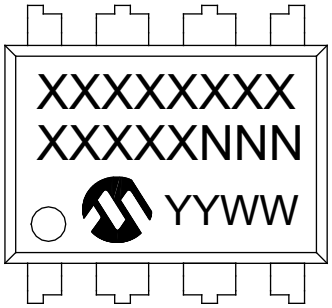
6-Lead SOT-23



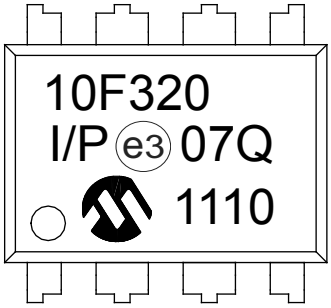
Example



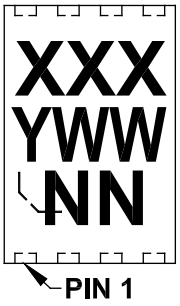
8-Lead PDIP (300 mil)



Example



8-Lead DFN (2x3x0.9 mm)



Example

Legend:	XX...X	Product-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week ‘01’)
	NNN	Alphanumeric traceability code
	(e3)	Pb-free JEDEC® designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator ((e3)) can be found on the outer packaging for this package.

Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

THE MICROCHIP WEBSITE

Microchip provides online support via our website at www.microchip.com. This website is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the website contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip website at www.microchip.com. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the website at: http://www.microchip.com/support