

Welcome to [E-XFL.COM](https://www.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	32MHz
Connectivity	I <sup>2</sup> C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	12
Program Memory Size	28KB (16K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 11x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Through Hole
Package / Case	14-DIP (0.300", 7.62mm)
Supplier Device Package	14-PDIP
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/pic16f18326-e-p">https://www.e-xfl.com/product-detail/microchip-technology/pic16f18326-e-p</a>

**TABLE 4-4: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (CONTINUED)**

Address	Name	PIC16(L)F18326	PIC16(L)F18346	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 10-11													
CPU CORE REGISTERS; see Table 4-2 for specifics													
50Ch to 51Fh	—	—		Unimplemented								—	—
58Ch to 59Fh	—	—		Unimplemented								—	—
Bank 12													
60Ch	—	—		Unimplemented								—	—
60Dh	—	—		Unimplemented								—	—
60Eh	—	—		Unimplemented								—	—
60Fh	—	—		Unimplemented								—	—
610h	—	—		Unimplemented								—	—
611h	—	—		Unimplemented								—	—
612h	—	—		Unimplemented								—	—
613h	—	—		Unimplemented								—	—
614h	—	—		Unimplemented								—	—
615h	—	—		Unimplemented								—	—
616h	—	—		Unimplemented								—	—
617h	PWM5DCL			PWM5DC<1:0>		—	—	—	—	—	—	xx-- ----	uu-- ----
618h	PWM5DCH			PWM5DC<9:2>								xxxx xxxx	uuuu uuuu
619h	PWM5CON			PWM5EN	—	PWM5OUT	PWM5POL	—	—	—	—	0-00 ----	0-00 ----
61Ah	PWM6DCL			PWM6DC<1:0>		—	—	—	—	—	—	xx-- ----	uu-- ----
61Bh	PWM6DCH			PWM6DC<9:2>								xxxx xxxx	uuuu uuuu
61Ch	PWM6CON			PWM6EN	—	PWM6OUT	PWM6POL	—	—	—	—	0-00 ----	0-00 ----
61Dh to 61Eh	—	—		Unimplemented								—	—
61Fh	PWMTMRS			—	—	—	—	P6TSEL<1:0>		P5TSEL<1:0>		---- 0101	---- 0101

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

**Note 1:** Only on PIC16F18326/18346.

**Note 2:** Register accessible from both User and ICD Debugger.

## 6.1 Power-on Reset (POR)

The POR circuit holds the device in Reset until VDD has reached an acceptable level for minimum operation. Slow rising VDD, fast operating speeds or analog performance may require greater than minimum VDD. The PWRT, BOR or MCLR features can be used to extend the start-up period until all device operation conditions have been met.

## 6.2 Brown-out Reset (BOR)

The BOR circuit holds the device in Reset while VDD is below a selectable minimum level. Between the POR and BOR, complete voltage range coverage for execution protection can be implemented.

The Brown-out Reset module has four operating modes controlled by the BOREN<1:0> bits in Configuration Words. The four operating modes are:

- BOR is always on
- BOR is off when in Sleep
- BOR is controlled by software
- BOR is always off

Refer to [Table 6-1](#) for more information.

The Brown-out Reset voltage level is selectable by configuring the BORV bit in Configuration Words.

A VDD noise rejection filter prevents the BOR from triggering on small events. If VDD falls below VBOR for a duration greater than parameter TBORDC, the device will reset, and the BOR bit of the PCON0 register will be cleared, indicating that a Brown-out Reset condition occurred. See [Figure 6-2](#) for more information.

**TABLE 6-1: BOR OPERATING MODES**

BOREN<1:0>	SBOREN	Device Mode	BOR Mode	Instruction Execution upon: Release of POR or Wake-up from Sleep
11	X	X	Active	In these specific cases, “Release of POR” and “Wake-up from Sleep”, there is no delay in start-up. The BOR ready flag, (BORRDY = 1), will be set before the CPU is ready to execute instructions because the BOR circuit is forced on by the BOREN<1:0> bits.
10	X	Awake	Active	Waits for release of BOR (BORRDY = 1)
		Sleep	Disabled	BOR ignored when asleep
01	1	X	Active	In these specific cases, “Release of POR” and “Wake-up from Sleep”, there is no delay in start-up. The BOR ready flag, (BORRDY = 1), will be set before the CPU is ready to execute instructions because the BOR circuit is forced on by the BOREN<1:0> bits
	0	X	Disabled	Begins immediately (BORRDY = x)
00	X	X	Disabled	

## 6.3 Low-Power Brown-out Reset (LPBOR)

The Low-Power Brown-Out Reset (LPBOR) circuit provides alternative protection against Brown-out conditions. When  $V_{DD}$  falls below the LPBOR threshold, the device is held in Reset. When this occurs, the BOR bit of the PCON0 register is cleared to indicate that a Brown-out Reset occurred. The BOR bit will be cleared when either the BOR or the LPBOR circuitry detects a BOR condition. The LPBOR feature can be used with or without BOR enabled.

When used while BOR is enabled, the LPBOR can be used as a secondary protection circuit in case the BOR circuit fails to detect the BOR condition. Additionally, when BOR is enabled except while in Sleep ( $BOREN<1:0> = 10$ ), the LPBOR circuit will hold the device in Reset while  $V_{DD}$  is lower than the LPBOR threshold, and will also re-arm the POR. (see Figure 35-11 for LPBOR Reset voltage levels).

When used without BOR enabled, the LPBOR circuit provides a single Reset trip point with the benefit of reduced current consumption.

### 6.3.1 ENABLING LPBOR

The LPBOR is controlled by the  $\overline{LPBOR}$  bit of Configuration Words. When the device is erased, the LPBOR module defaults to disabled.

#### 6.3.1.1 LPBOR Module Output

The output of the LPBOR module is a signal indicating whether or not a Reset is to be asserted. This signal is OR'd together with the Reset signal of the BOR module to provide the generic  $\overline{BOR}$  signal, which goes to the PCON register and to the power control block.

## 6.4 $\overline{MCLR}$

The  $\overline{MCLR}$  is an optional external input that can reset the device. The  $\overline{MCLR}$  function is controlled by the MCLRE bit of Configuration Words and the LVP bit of Configuration Words (Table 6-2).

**TABLE 6-2:  $\overline{MCLR}$  CONFIGURATION**

MCLRE	LVP	$\overline{MCLR}$
0	0	Disabled
1	0	Enabled
x	1	Enabled

### 6.4.1 $\overline{MCLR}$ ENABLED

When  $\overline{MCLR}$  is enabled and the pin is held low, the device is held in Reset. The  $\overline{MCLR}$  pin is connected to  $V_{DD}$  through an internal weak pull-up.

The device has a noise filter in the  $\overline{MCLR}$  Reset path. The filter will detect and ignore small pulses.

**Note:** A Reset does not drive the  $\overline{MCLR}$  pin low.

### 6.4.2 $\overline{MCLR}$ DISABLED

When  $\overline{MCLR}$  is disabled, the pin functions as a general purpose input and the internal weak pull-up is under software control. See Section 12.2 “PORTA Registers” for more information.

## 6.5 Watchdog Timer (WDT) Reset

The Watchdog Timer generates a Reset if the firmware does not issue a  $\overline{CLRWDT}$  instruction within the time-out period. The  $\overline{TO}$  and PD bits in the STATUS register as well as the  $\overline{RWDT}$  bit in the PCON register, are changed to indicate the WDT Reset. See Section 10.0 “Watchdog Timer (WDT)” for more information.

## 6.6 RESET Instruction

A  $\overline{RESET}$  instruction will cause a device Reset. The  $\overline{RI}$  bit in the PCON register will be set to ‘0’. See Table 6-4 for default conditions after a  $\overline{RESET}$  instruction has occurred.

## 6.7 Stack Overflow/Underflow Reset

The device can reset when the Stack Overflows or Underflows. The STKOVF or STKUNF bits of the PCON register indicate the Reset condition. These Resets are enabled by setting the STVREN bit in Configuration Words. See Section 4.4 “Stack” for more information.

## 6.8 Programming Mode Exit

Upon exit of Programming mode, the device will behave as if a device Reset had just occurred.

## 6.9 Power-up Timer

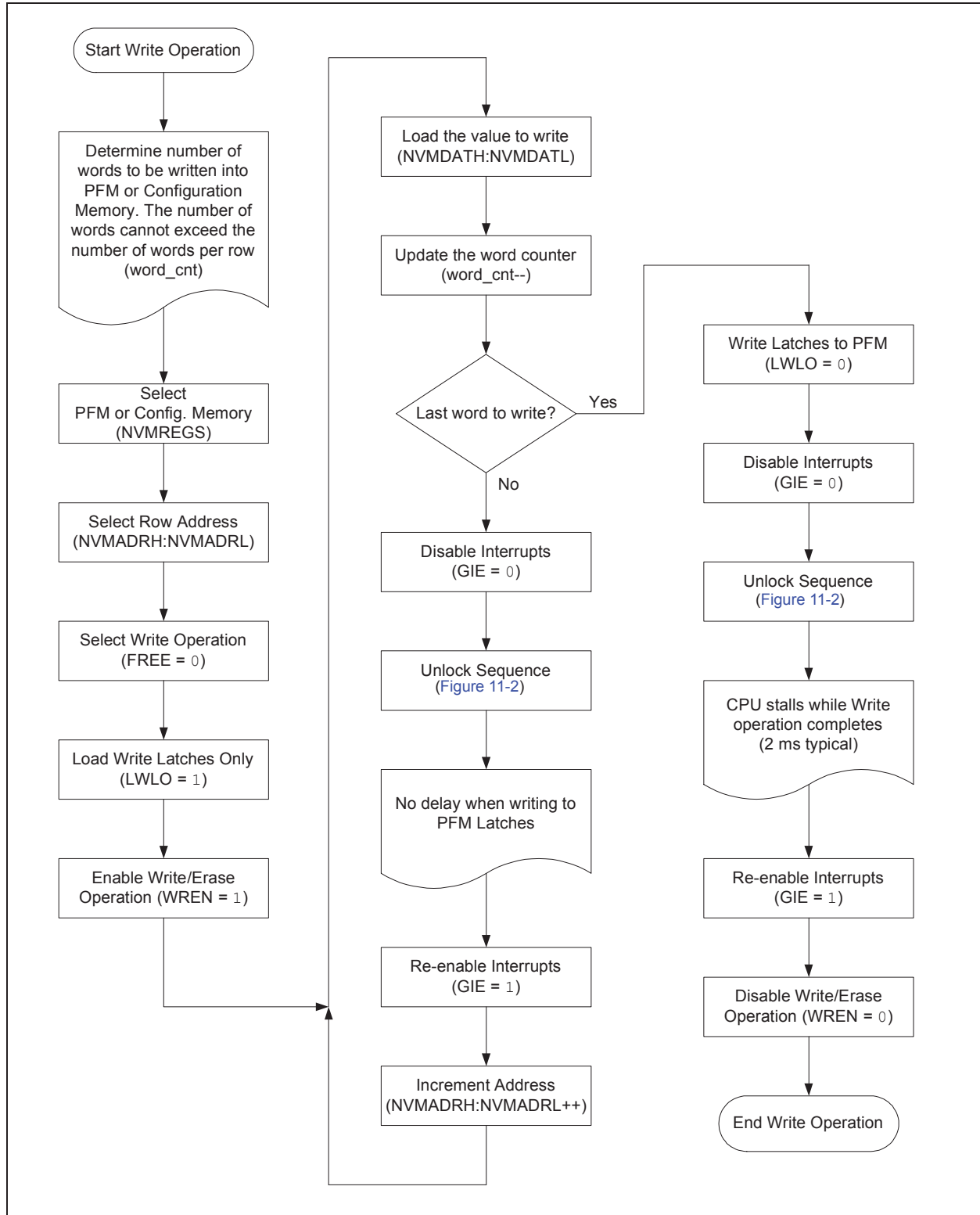
The Power-up Timer provides a nominal 64 ms time-out on POR or Brown-out Reset.

The device is held in Reset as long as PWRT is active. The PWRT delay allows additional time for the  $V_{DD}$  to rise to an acceptable level. The Power-up Timer is enabled by clearing the PWRT bit in Configuration Words.

The Power-up Timer starts after the release of the POR and BOR.

For additional information, refer to Application Note AN607, *Power-up Trouble Shooting* (DS00607).

**FIGURE 11-5: PROGRAM FLASH MEMORY WRITE FLOWCHART**





# PIC16(L)F18326/18346

**REGISTER 15-9: IOCCF: INTERRUPT-ON-CHANGE PORTC FLAG REGISTER**

R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0
IOCCF7 <sup>(1)</sup>	IOCCF6 <sup>(1)</sup>	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0
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	HS - Bit is set in hardware

bit 7-6 **IOCCF<7:6>:** Interrupt-on-Change PORTC Flag bits

- 1 = An enabled change was detected on the associated pin.  
Set when IOCCPx = 1 and a rising edge was detected on RCx, or when IOCCNx = 1 and a falling edge was detected on RCx.
- 0 = No change was detected, or the user cleared the detected change.

bit 5-0 **IOCCF<5:0>:** Interrupt-on-Change PORTC Flag bits

- 1 = An enabled change was detected on the associated pin.  
Set when IOCCPx = 1 and a rising edge was detected on RCx, or when IOCCNx = 1 and a falling edge was detected on RCx.
- 0 = No change was detected, or the user cleared the detected change.

**Note 1:** PIC16(L)F18346 only.

**TABLE 15-1: SUMMARY OF REGISTERS ASSOCIATED WITH INTERRUPT-ON-CHANGE**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELA	—	—	ANSA4	ANSA4	—	ANSA2	ANSA1	ANSA0	144
ANSELB <sup>(1)</sup>	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	150
ANSELC	ANSC7 <sup>(1)</sup>	ANSC6 <sup>(1)</sup>	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	157
TRISA	—	—	TRISA5	TRISA4	— <sup>(2)</sup>	TRISA2	TRISA1	TRISA0	143
TRISB <sup>(1)</sup>	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	149
TRISC	TRISC7 <sup>(1)</sup>	TRISC6 <sup>(1)</sup>	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	155
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	100
PIE0	—	—	TMR0IE	IOCIE	—	—	—	INTE	101
IOCAP	—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	174
IOCAN	—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	174
IOCAF	—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	175
IOCBP <sup>(1)</sup>	IOCBP7	IOCBP6	IOCBP5	IOCBP4	—	—	—	—	175
IOCBN <sup>(1)</sup>	IOCBN7	IOCBN6	IOCBN5	IOCBN4	—	—	—	—	176
IOCBF <sup>(1)</sup>	IOCBF7	IOCBF6	IOCBF5	IOCBF4	—	—	—	—	176
IOCCP	IOCCP7 <sup>(1)</sup>	IOCCP6 <sup>(1)</sup>	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	177
IOCCN	IOCCN7 <sup>(1)</sup>	IOCCN6 <sup>(1)</sup>	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	177
IOCCF	IOCCF7 <sup>(1)</sup>	IOCCF6 <sup>(1)</sup>	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	178

**Legend:** — = unimplemented location, read as '0'. Shaded cells are not used by interrupt-on-change.

**Note 1:** PIC16(L)F18346 only.

**2:** Unimplemented, read as '1'.

# PIC16(L)F18326/18346

## 18.0 COMPARATOR MODULE

Comparators are used to interface analog circuits to a digital circuit by comparing two analog voltages and providing a digital indication of their relative magnitudes. Comparators are very useful mixed signal building blocks because they provide analog functionality independent of program execution. The analog comparator module includes the following features:

- Programmable input selection
  - Selectable voltage reference
- Programmable output polarity
- Rising/falling output edge interrupts
- Wake-up from Sleep
- CWG Auto-shutdown source

### 18.1 Comparator Overview

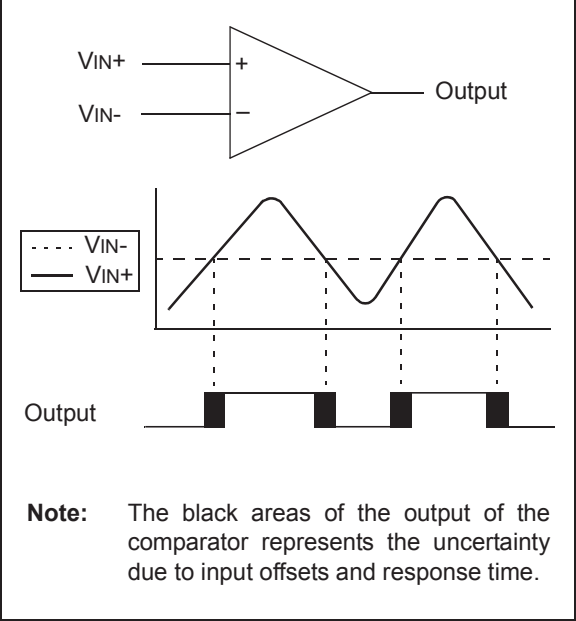
A single comparator is shown in [Figure 18-1](#) along with the relationship between the analog input levels and the digital output. When the analog voltage at  $V_{IN+}$  is less than the analog voltage at  $V_{IN-}$ , the output of the comparator is a digital low level. When the analog voltage at  $V_{IN+}$  is greater than the analog voltage at  $V_{IN-}$ , the output of the comparator is a digital high level.

The comparators available for this device are located in [Table 18-1](#).

TABLE 18-1: AVAILABLE COMPARATORS

Device	C1	C2
PIC16(L)F18326	•	•
PIC16(L)F18346	•	•

FIGURE 18-1: SINGLE COMPARATOR





# PIC16(L)F18326/18346

**TABLE 19-3: SUMMARY OF REGISTERS ASSOCIATED WITH PWMx**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
TRISA	—	—	TRISA5	TRISA4	— <sup>(2)</sup>	TRISA2	TRISA1	TRISA0	143
ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	144
TRISB <sup>(1)</sup>	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	149
ANSELB <sup>(1)</sup>	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	150
TRISC	TRISC7 <sup>(1)</sup>	TRISC6 <sup>(1)</sup>	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	155
ANSELC	ANSC7 <sup>(1)</sup>	ANSC6 <sup>(1)</sup>	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	157
PWM5CON	PWM5EN	—	PWM5OUT	PWM5POL	—	—	—	—	196
PWM5DCH	PWM5DC<9:2>								196
PWM5DCL	PWM5DC<1:0>		—	—	—	—	—	—	196
PWM6CON	PWM6EN	—	PWM6OUT	PWM6POL	—	—	—	—	196
PWM6DCH	PWM6DC<9:2>								196
PWM6DCL	PWM6DC<1:0>		—	—	—	—	—	—	196
PWMTMRS	—	—	—	—	P6TSEL<1:0>		P5TSEL<1:0>		197
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	100
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	BCL1IF	TMR2IF	TMR1IF	107
PIR2	TMR6IF	C2IF	C1IF	NVMIF	SSP2IF	BCL2IF	TMR4IF	NCO1IF	108
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	BCL1IE	TMR2IE	TMR1IE	102
PIE2	TMR6IE	C2IE	C1IE	NVMIE	SSP2IE	BCL2IE	TMR4IE	NCO1IE	103
T2CON	—	T2OUTPS<3:0>				TMR2ON	T2CKPS<1:0>		298
T4CON	—	T4OUTPS<3:0>				TMR4ON	T4CKPS<1:0>		292
T6CON	—	T6OUTPS<3:0>				TMR6ON	T6CKPS<1:0>		292
TMR2	TMR2<7:0>								299
TMR4	TMR4<7:0>								299
TMR6	TMR6<7:0>								299
PR2	PR2<7:0>								299
PR4	PR4<7:0>								299
PR6	PR6<7:0>								299
CWGxDAT	—	—	—	—	DAT<3:0>				215
CLCxSEly	—	—	LCxDyS<5:0>						229
MDSRC	—	—	—	—	MDMS<3:0>				272
MDCARH	—	MDCHPOL	MDCHSYNC	—	MDCH<3:0>				273
MDCARL	—	MDCLPOL	MDCLSYNC	—	MDCL<3:0>				274

**Legend:** — = Unimplemented locations, read as '0'. Shaded cells are not used by the PWM module.

**Note 1:** PIC16(L)F18346 only.

**2:** Unimplemented, read as '1'.

## REGISTER 20-9: CWGxDBF: CWGx FALLING DEAD-BAND COUNT REGISTER

U-0	U-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
—	—	DBF<5:0>					
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 7-6

**Unimplemented:** Read as '0'

bit 5-0

**DBF<5:0>:** CWG Falling Edge Triggered Dead-Band Count bits

11 1111 = 63-64 CWG clock periods

11 1110 = 62-63 CWG clock periods

.

.

.

00 0010 = 2-3 CWG clock periods

00 0001 = 1-2 CWG clock periods

00 0000 = 0 CWG clock periods. Dead-band generation is bypassed.

# PIC16(L)F18326/18346

## REGISTER 22-6: ADRESH: ADC RESULT REGISTER HIGH (ADRESH) 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<9:8>	
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 7-2 **Reserved:** Do not use.

bit 1-0 **ADRES<9:8>:** ADC Result Register bits  
Upper two bits of 10-bit conversion result

## REGISTER 22-7: ADRESL: ADC RESULT REGISTER LOW (ADRESL) 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

## 30.2.4 SPI SLAVE MODE

In Slave mode, the data is transmitted and received as external clock pulses appear on SCK. When the last bit is latched, the SSPxIF interrupt flag bit is set.

Before enabling the module in SPI Slave mode, the clock line must match the proper Idle state. The clock line can be observed by reading the SCK pin. The Idle state is determined by the CKP bit of the SSPxCON1 register.

While in Slave mode, the external clock is supplied by the external clock source on the SCK pin. This external clock must meet the minimum high and low times as specified in the electrical specifications.

While in Sleep mode, the slave can transmit/receive data. The shift register is clocked from the SCK pin input and when a byte is received, the device will generate an interrupt. If enabled, the device will wake-up from Sleep.

### 30.2.4.1 Daisy-Chain Configuration

The SPI bus can sometimes be connected in a daisy-chain configuration. The first slave output is connected to the second slave input, the second slave output is connected to the third slave input, and so on. The final slave output is connected to the master input. Each slave sends out, during a second group of clock pulses, an exact copy of what was received during the first group of clock pulses. The whole chain acts as one large communication shift register. The daisy-chain feature only requires a single Slave Select line from the master device.

Figure 30-7 shows the block diagram of a typical daisy-chain connection when operating in SPI mode.

In a daisy-chain configuration, only the most recent byte on the bus is required by the slave. Setting the BOEN bit of the SSPxCON3 register will enable writes to the SSPxBUF register, even if the previous byte has not been read. This allows the software to ignore data that may not apply to it.

## 30.2.5 SLAVE SELECT SYNCHRONIZATION

The Slave Select can also be used to synchronize communication. The Slave Select line is held high until the master device is ready to communicate. When the Slave Select line is pulled low, the slave knows that a new transmission is starting.

If the slave fails to receive the communication properly, it will be reset at the end of the transmission, when the Slave Select line returns to a high state. The slave is then ready to receive a new transmission when the Slave Select line is pulled low again. If the Slave Select line is not used, there is a risk that the slave will eventually become out of sync with the master. If the slave misses a bit, it will always be one bit off in future transmissions. Use of the Slave Select line allows the slave and master to align themselves at the beginning of each transmission.

The  $\overline{SS}$  pin allows a Synchronous Slave mode. The SPI must be in Slave mode with  $\overline{SS}$  pin control enabled ( $SSPxCON1<3:0> = 0100$ ).

When the  $\overline{SS}$  pin is low, transmission and reception are enabled and the SDO pin is driven.

When the  $\overline{SS}$  pin goes high, the SDO pin is no longer driven, even if in the middle of a transmitted byte and becomes a floating output. External pull-up/pull-down resistors may be desirable depending on the application.

- Note 1:** When the SPI is in Slave mode with  $\overline{SS}$  pin control enabled ( $SSPxCON1<3:0> = 0100$ ), the SPI module will reset if the  $\overline{SS}$  pin is set to VDD.

**2:** When the SPI is used in Slave mode with CKE set; the user must enable  $\overline{SS}$  pin control.

**3:** While operated in SPI Slave mode the SMP bit of the SSPxSTAT register must remain clear.

When the SPI module resets, the bit counter is forced to '0'. This can be done by either forcing the  $\overline{SS}$  pin to a high level or clearing the SSPEN bit.

## 30.2.6 SPI OPERATION IN SLEEP MODE

In SPI Master mode, when the Sleep mode is selected, all module clocks are halted and the transmission/reception will remain in that state until the device wakes. After the device returns to Run mode, the module will resume transmitting and receiving data.

In SPI Slave mode, the SPI Transmit/Receive Shift register operates asynchronously to the device. This allows the device to be placed in Sleep mode and data to be shifted into the SPI Transmit/Receive Shift register. When all eight bits have been received, the MSSP interrupt flag bit will be set and if enabled, will wake the device.

## 30.3 I<sup>2</sup>C Mode Overview

The Inter-Integrated Circuit (I<sup>2</sup>C) bus is a multi-master serial data communication bus. Devices communicate in a master/slave environment where the master devices initiate the communication. A slave device is controlled through addressing.

The I<sup>2</sup>C bus specifies two signal connections:

- Serial Clock (SCL)
- Serial Data (SDA)

Figure 30-11 shows the block diagram of the MSSPx module when operating in I<sup>2</sup>C mode.

Both the SCL and SDA connections are bidirectional open-drain lines, each requiring pull-up resistors for the supply voltage. Pulling the line to ground is considered a logical zero and letting the line float is considered a logical one.

Figure 30-2 and Figure 30-3 show a typical connection between two processors configured as master and slave devices.

The I<sup>2</sup>C bus can operate with one or more master devices and one or more slave devices.

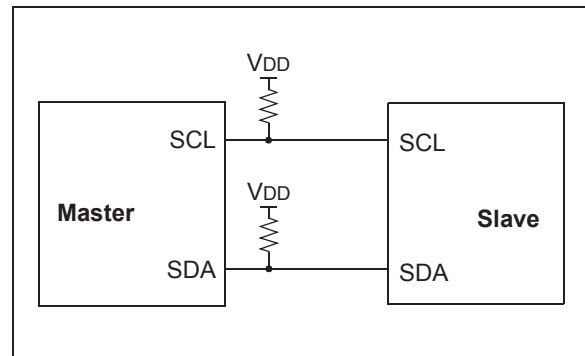
There are four potential modes of operation for a given device:

- Master Transmit mode  
(master is transmitting data to a slave)
- Master Receive mode  
(master is receiving data from a slave)
- Slave Transmit mode  
(slave is transmitting data to a master)
- Slave Receive mode  
(slave is receiving data from the master)

To begin communication, the master device sends out a Start condition followed by the address byte of the slave it intends to communicate with. This is followed by a single Read/Write bit, which determines whether the master intends to transmit to or receive data from the slave device.

If the requested slave exists on the bus, it will respond with an Acknowledge bit, otherwise known as an ACK. The master then continues in either transmit or receive data from the slave device.

**FIGURE 30-11: I<sup>2</sup>C MASTER/SLAVE CONNECTION**



On the last byte of data communicated, the master device may end the transmission by sending a Stop bit. If the master device is in Receive mode, it sends a NACK in place of the last ACK bit. A Stop bit is indicated by a low-to-high transition of the SDA line while the SCL line is held high.

In some cases, the master may want to maintain control of the bus and re-initiate another transmission. If so, the master device may send a Restart condition in place of the Stop bit or last ACK bit when it is in Receive mode.

The I<sup>2</sup>C bus specifies three message protocols;

- Single message where a master writes data to a slave.
- Single message where a master reads data from a slave.
- Combined message where a master initiates a minimum of two writes, or two reads, or a combination of writes and reads, to one or more slaves.

### 30.3.1 CLOCK STRETCHING

When a slave device has not completed processing data, it can delay the transfer of more data through the process of clock stretching. An addressed slave device may hold the SCL clock line low after receiving or sending a bit, indicating that it is not yet ready to continue. The master that is communicating with the slave will attempt to raise the SCL line in order to transfer the next bit, but will detect that the clock line has not yet been released. Because the SCL connection is open-drain, the slave has the ability to hold that line low until it is ready to continue communicating.

Clock stretching allows receivers that cannot keep up with a transmitter to control the flow of incoming data.

# PIC16(L)F18326/18346

## 30.4.5 START CONDITION

The I<sup>2</sup>C specification defines a Start condition as a transition of SDA from a high to a low state while SCL line is high. A Start condition is always generated by the master and signifies the transition of the bus from an Idle to an Active state. Figure 30-12 shows wave forms for Start and Stop conditions.

## 30.4.6 STOP CONDITION

A Stop condition is a transition of the SDA line from low-to-high state while the SCL line is high.

**Note:** At least one SCL low time must appear before a Stop is valid, therefore, if the SDA line goes low then high again while the SCL line stays high, only the Start condition is detected.

## 30.4.7 RESTART CONDITION

A Restart is valid any time that a Stop would be valid. A master can issue a Restart if it wishes to hold the bus after terminating the current transfer. A Restart has the same effect on the slave that a Start would, resetting all slave logic and preparing it to clock in an address. The master may want to address the same or another slave. Figure 30-13 shows the wave form for a Restart condition.

In 10-bit Addressing Slave mode a Restart is required for the master to clock data out of the addressed slave. Once a slave has been fully addressed, matching both high and low address bytes, the master can issue a Restart and the high address byte with the R/W bit set. The slave logic will then hold the clock and prepare to clock out data.

## 30.4.8 START/STOP CONDITION INTERRUPT MASKING

The SCIE and PCIE bits of the SSPxCON3 register can enable the generation of an interrupt in Slave modes that do not typically support this function. Slave modes where interrupt on Start and Stop detect are already enabled, these bits will have no effect.

FIGURE 30-12: I<sup>2</sup>C START AND STOP CONDITIONS

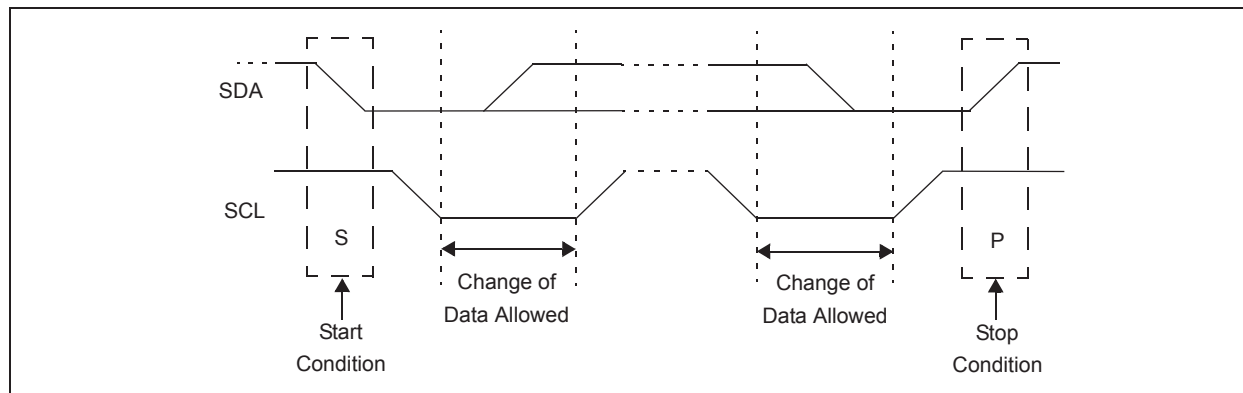
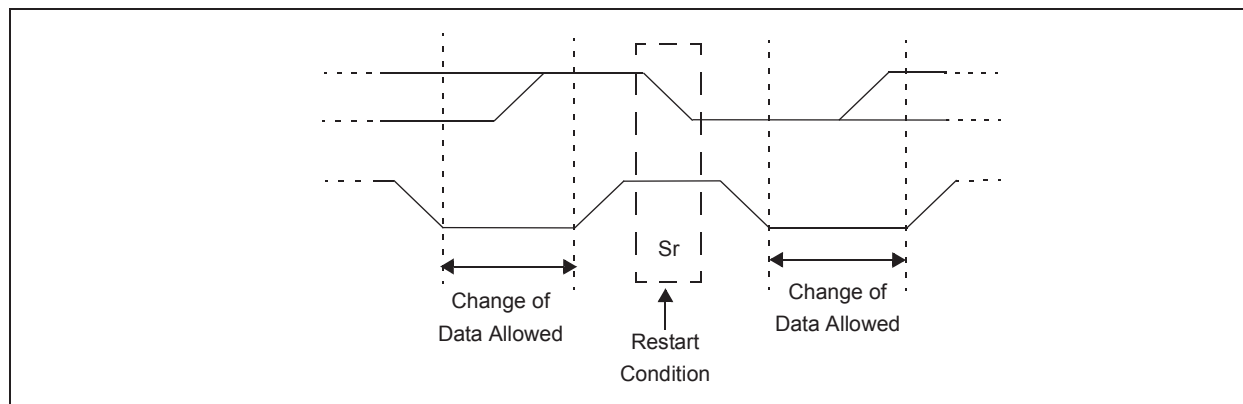
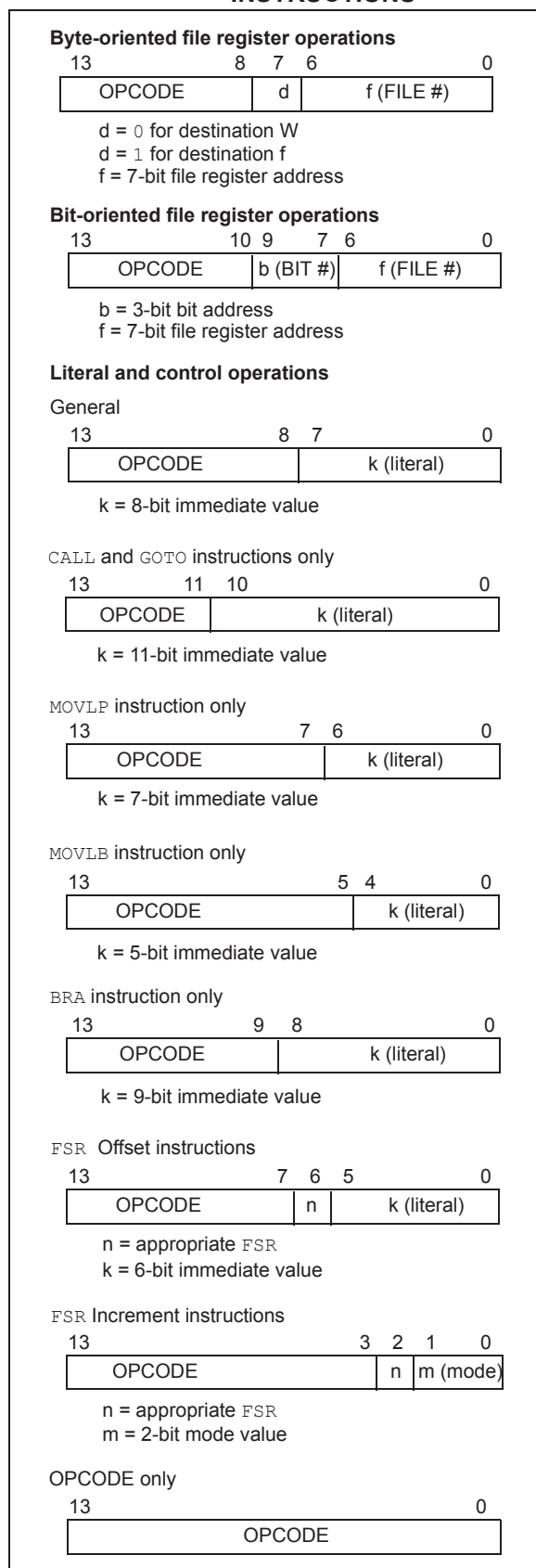


FIGURE 30-13: I<sup>2</sup>C RESTART CONDITION



**FIGURE 34-1: GENERAL FORMAT FOR INSTRUCTIONS**



# PIC16(L)F18326/18346

## CALL Call Subroutine

Syntax:	[ <i>label</i> ] CALL k
Operands:	$0 \leq k \leq 2047$
Operation:	(PC)+1 → TOS, k → PC<10:0>, (PCLATH<6:3>) → PC<14:11>
Status Affected:	None
Description:	Call Subroutine. First, return address (PC + 1) is pushed onto the stack. The 11-bit immediate address is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH. CALL is a 2-cycle instruction.

## CLRWDTClear Watchdog Timer

Syntax:	[ <i>label</i> ] CLRWDTClear Watchdog Timer
Operands:	None
Operation:	00h → WDT 0 → WDT prescaler, 1 → $\overline{TO}$ 1 → $\overline{PD}$
Status Affected:	$\overline{TO}$ , $\overline{PD}$
Description:	CLRWDTClear Watchdog Timer instruction resets the Watchdog Timer. It also resets the prescaler of the WDT. Status bits $\overline{TO}$ and $\overline{PD}$ are set.

## CALLW Subroutine Call With W

Syntax:	[ <i>label</i> ] CALLW
Operands:	None
Operation:	(PC)+1 → TOS, (W) → PC<7:0>, (PCLATH<6:0>) → PC<14:8>
Status Affected:	None
Description:	Subroutine call with W. First, the return address (PC + 1) is pushed onto the return stack. Then, the contents of W is loaded into PC<7:0>, and the contents of PCLATH into PC<14:8>. CALLW is a 2-cycle instruction.

## COMF Complement f

Syntax:	[ <i>label</i> ] COMF f,d
Operands:	$0 \leq f \leq 127$ d ∈ [0,1]
Operation:	( $\bar{f}$ ) → (destination)
Status Affected:	Z
Description:	The contents of register 'f' are complemented. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f'.

## CLRF Clear f

Syntax:	[ <i>label</i> ] CLRF f
Operands:	$0 \leq f \leq 127$
Operation:	00h → (f) 1 → Z
Status Affected:	Z
Description:	The contents of register 'f' are cleared and the Z bit is set.

## DECF Decrement f

Syntax:	[ <i>label</i> ] DECF f,d
Operands:	$0 \leq f \leq 127$ d ∈ [0,1]
Operation:	(f) - 1 → (destination)
Status Affected:	Z
Description:	Decrement register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

## CLRW Clear W

Syntax:	[ <i>label</i> ] CLRWClear W
Operands:	None
Operation:	00h → (W) 1 → Z
Status Affected:	Z
Description:	W register is cleared. Zero bit (Z) is set.



# PIC16(L)F18326/18346

## MOVWI Move W to INDFn

Syntax: [ *label* ] MOVWI ++FSRn  
[ *label* ] MOVWI --FSRn  
[ *label* ] MOVWI FSRn++  
[ *label* ] MOVWI FSRn--  
[ *label* ] MOVWI k[FSRn]

Operands:  $n \in [0,1]$   
 $mm \in [00,01, 10, 11]$   
 $-32 \leq k \leq 31$

Operation:  $W \rightarrow \text{INDFn}$   
Effective address is determined by

- FSR + 1 (preincrement)
- FSR - 1 (predecrement)
- FSR + k (relative offset)

After the Move, the FSR value will be either:

- FSR + 1 (all increments)
- FSR - 1 (all decrements)

Unchanged

Status Affected: None

Mode	Syntax	mm
Preincrement	++FSRn	00
Predecrement	--FSRn	01
Postincrement	FSRn++	10
Postdecrement	FSRn--	11

Description: This instruction is used to move data between W and one of the indirect registers (INDFn). Before/after this move, the pointer (FSRn) is updated by pre/post incrementing/decrementing it.

**Note:** The INDFn registers are not physical registers. Any instruction that accesses an INDFn register actually accesses the register at the address specified by the FSRn.

FSRn is limited to the range 0000h-FFFFh. Incrementing/decrementing it beyond these bounds will cause it to wrap-around.

The increment/decrement operation on FSRn WILL NOT affect any Status bits.

## NOP No Operation

Syntax: [ *label* ] NOP

Operands: None

Operation: No operation

Status Affected: None

Description: No operation.

Words: 1

Cycles: 1

Example: NOP

## RESET Software Reset

Syntax: [ *label* ] RESET

Operands: None

Operation: Execute a device Reset. Resets the RI flag of the PCON register.

Status Affected: None

Description: This instruction provides a way to execute a hardware Reset by software.

## RETFIE Return from Interrupt

Syntax: [ *label* ] RETFIE k

Operands: None

Operation:  $\text{TOS} \rightarrow \text{PC}$ ,  
 $1 \rightarrow \text{GIE}$

Status Affected: None

Description: Return from Interrupt. Stack is POPed and Top-of-Stack (TOS) is loaded in the PC. Interrupts are enabled by setting Global Interrupt Enable bit, GIE (INTCON<7>). This is a 2-cycle instruction.

Words: 1

Cycles: 2

Example: RETFIE

After Interrupt

PC	=	TOS
GIE	=	1

# PIC16(L)F18326/18346

TABLE 35-8: OSCILLATOR PARAMETERS<sup>(1)</sup>

Standard Operating Conditions (unless otherwise stated)							
Param. No.	Sym.	Characteristic	Min.	Typ.†	Max.	Units	Conditions
OS20	FHFOSC	Precision Calibrated HFINTOSC Frequency	3.92	4	4.08	MHz	25°C
OS20	FHFOSC	Precision Calibrated HFINTOSC Frequency	—	4 8 12 16 32	—	MHz	-40°C to 125°C <sup>(2)</sup>
OS21	FHFOSCLP	Low-Power Optimized HFINTOSC Frequency	0.93 1.86	1 2	1.07 2.14	MHz MHz	
OS23	LFOSC	Internal LFINTOSC Frequency	—	31	—	kHz	
OS24	THFOSCST	HFINTOSC Wake-up from Sleep Start-up Time	—	11 50	20 —	μs μs	VREGPM = 0 VREGPM = 1
OS26	TLFOSCST	LFINTOSC Wake-up from Sleep Start-up Time	—	0.2	—	ms	

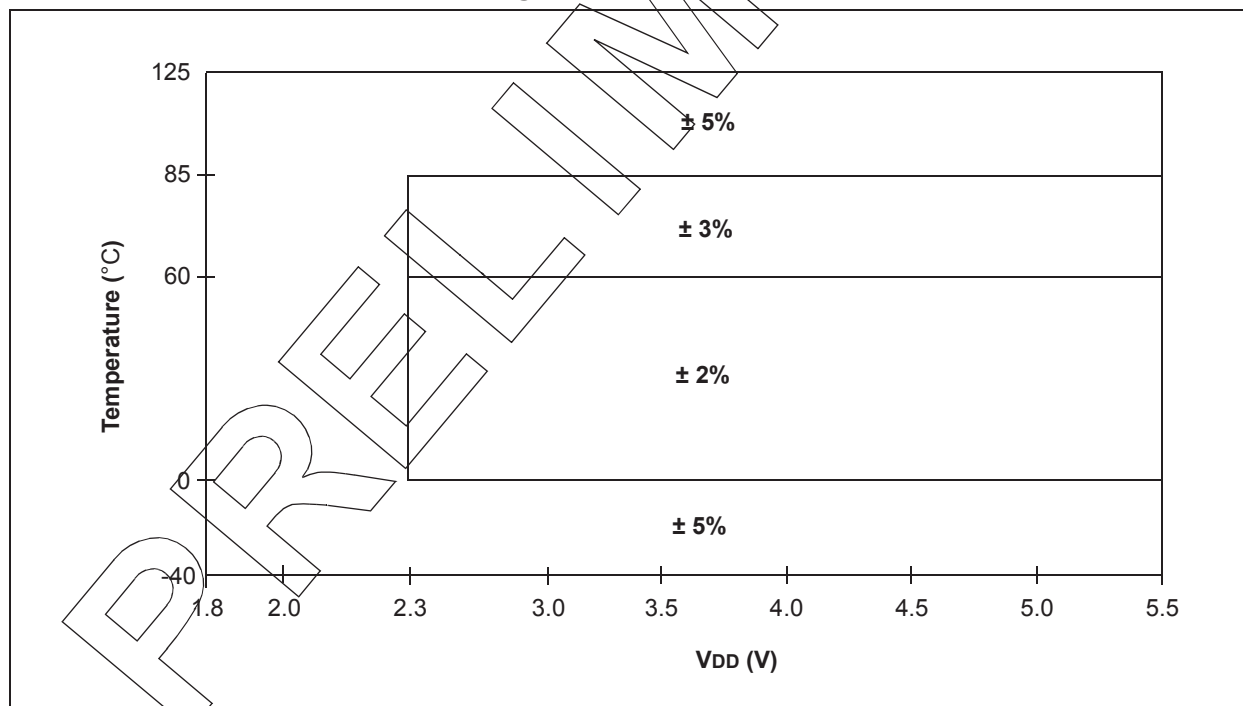
\* These parameters are characterized but not tested.

† Data in "Typ." column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** To ensure these oscillator frequency tolerances, VDD and VSS must be capacitively decoupled as close to the device as possible. 0.1 μF and 0.01 μF values in parallel are recommended.

**2:** See Figure 35-6.

FIGURE 35-6: PRECISION CALIBRATED HFINTOSC FREQUENCY ACCURACY OVER DEVICE VDD AND TEMPERATURE



**TABLE 35-12: ANALOG-TO-DIGITAL CONVERTER (ADC) CHARACTERISTICS<sup>(1,2)</sup>**

<b>Standard Operating Conditions</b> (unless otherwise stated) VDD = 3.0V, TA = 25°C							
Param. No.	Sym.	Characteristic	Min.	Typ.†	Max.	Units	Conditions
AD01	NR	Resolution	—	—	10	bit	
AD02	EIL	Integral Error	—	±0.1	±1.0	LSb	ADCREF+ = 3.0V, ADCREF- = 0V
AD03	EDL	Differential Error	—	±0.1	±1.0	LSb	ADCREF+ = 3.0V, ADCREF- = 0V
AD04	EOFF	Offset Error	—	0.5	2	LSb	ADCREF+ = 3.0V, ADCREF- = 0V
AD05	EGN	Gain Error	—	±0.2	±1.0	LSb	ADCREF+ = 3.0V, ADCREF- = 0V
AD06	VADREF	ADC Reference Voltage (ADREF+) <sup>(3)</sup>	1.8	—	VDD	V	
AD07	VAIN	Full-Scale Range	VSS	—	ADREF+	V	
AD06	VADREF	ADC Reference Voltage (ADREF+ - ADREF-) <sup>(3)</sup>	1.8	—	VDD	V	
AD07	VAIN	Full-Scale Range	ADREF-	—	ADREF+	V	
AD08	ZAIN	Recommended Impedance of Analog Voltage Source	—	10	—	kΩ	
AD09	RVREF	ADC Voltage Reference Ladder Impedance	—	—	—	kΩ	

\* These parameters are characterized but not tested.

† Data in "Typ." column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** Total Absolute Error is the sum of the offset, gain and integral non-linearity (INL) errors.

**2:** The ADC conversion result never decreases with an increase in the input and has no missing codes.

## 37.2 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

## 37.3 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

## 37.4 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

## 37.5 MPLAB Assembler, Linker and Librarian for Various Device Families

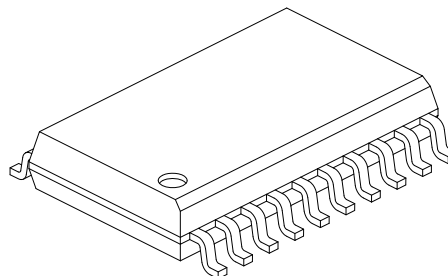
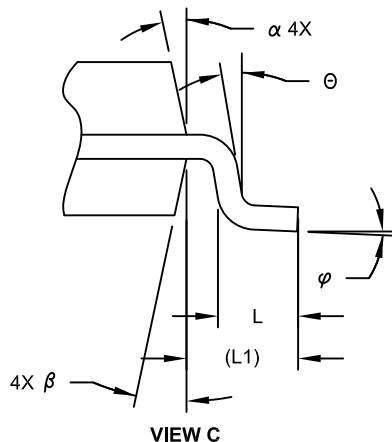
MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC DSC devices. MPLAB XC Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object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

# PIC16(L)F18326/18346

## 20-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	20		
Pitch	e	1.27 BSC		
Overall Height	A	-	-	2.65
Molded Package Thickness	A2	2.05	-	-
Standoff §	A1	0.10	-	0.30
Overall Width	E	10.30 BSC		
Molded Package Width	E1	7.50 BSC		
Overall Length	D	12.80 BSC		
Chamfer (Optional)	h	0.25	-	0.75
Foot Length	L	0.40	-	1.27
Footprint	L1	1.40 REF		
Lead Angle	$\theta$	0°	-	-
Foot Angle	$\phi$	0°	-	8°
Lead Thickness	c	0.20	-	0.33
Lead Width	b	0.31	-	0.51
Mold Draft Angle Top	$\alpha$	5°	-	15°
Mold Draft Angle Bottom	$\beta$	5°	-	15°

### Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- § Significant Characteristic
- Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M
  - BSC: Basic Dimension. Theoretically exact value shown without tolerances.
  - REF: Reference Dimension, usually without tolerance, for information purposes only.
- Datums A & B to be determined at Datum H.

Microchip Technology Drawing No. C04-094C Sheet 2 of 2