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	32MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	25
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 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/pic16f18856-i-ss

PIC16(L)F188XX Family Types

Device	Data Sheet Index	Program Flash Memory (Words)	Program Flash Memory (KB)	EEPROM (bytes)	Data SRAM (bytes)	I/O Pins[1]	10-Bit ADC[2] (ch)	5-Bit DAC	Comparator	8-Bit (with HLT)/ 16-Bit Timers	SMT	Windowed Watchdog Timer	CRC and Memory Scan	CCP/10-Bit PWM	Zero-Cross Detect	CWG	NCO	CLC	DSM	EUSART/I²C/SPI	Peripheral Pin Select	Peripheral Module Disable
PIC16(L)F18854	(1)	4096	7	256	512	25	24	1	2	3/4	2	Y	Y	5/2	Y	3	1	4	1	1/2	Y	Y
PIC16(L)F18855	(2)	8192	14	256	1024	25	24	1	2	3/4	2	Y	Y	5/2	Y	3	1	4	1	1/2	Y	Y
PIC16(L)F18856	(3)	16384	28	256	2048	25	24	1	2	3/4	2	Y	Y	5/2	Y	3	1	4	1	1/2	Y	Y
PIC16(L)F18857	(4)	32768	56	256	4096	25	24	1	2	3/4	2	Y	Y	5/2	Y	3	1	4	1	1/2	Y	Y
PIC16(L)F18875	(2)	8192	14	256	1024	36	35	1	2	3/4	2	Y	Y	5/2	Y	3	1	4	1	1/2	Y	Y
PIC16(L)F18876	(3)	16384	28	256	2048	36	35	1	2	3/4	2	Y	Y	5/2	Y	3	1	4	1	1/2	Y	Y
PIC16(L)F18877	(4)	32768	56	256	4096	36	35	1	2	3/4	2	Y	Y	5/2	Y	3	1	4	1	1/2	Y	Y

Note 1: One pin is input-only.

Data Sheet Index: (Unshaded devices are described in this document)

1: DS40001826 PIC16(L)F18854 Data Sheet, 28-Pin, Full-Featured 8-bit Microcontrollers
2: DS40001802 PIC16(L)F18855/75 Data Sheet, 28/40-Pin, Full-Featured 8-bit Microcontrollers
3: DS40001824 PIC16(L)F18856/76 Data Sheet, 28/40-Pin, Full-Featured 8-bit Microcontrollers
4: DS40001825 PIC16(L)F18857/77 Data Sheet, 28/40-Pin, Full-Featured 8-bit Microcontrollers

Note: For other small form-factor package availability and marking information, please visit **http://www.microchip.com/packaging** or contact your local sales office.

TABLE 1-3: PIC16F18876 PINOUT DESCRIPTION (CONTINUED)

Name	Function	Input Type	Output Type	Description
RB2/ANB2/SDA2(3,4)/SDI2(1)/ CWG3IN(1)/IOCB2	RB2	TTL/ST	CMOS/OD	General purpose I/O.
	ANB2	AN	—	ADC Channel B2 input.
	SDA2(3,4)	I^2C/SMBus	OD	MSSP2 I^2C serial data input/output.
	SDI2(1)	TTL/ST	—	MSSP2 SPI serial data input.
	CWG3IN(1)	TTL/ST	—	Complementary Waveform Generator 3 input.
	IOCB2	TTL/ST	—	Interrupt-on-change input.
RB3/ANB3/C1IN2-/C2IN2-/IOCB3	RB3	TTL/ST	CMOS/OD	General purpose I/O.
	ANB3	AN	—	ADC Channel B3 input.
	C1IN2-	AN	—	Comparator negative input.
	C2IN2-	AN	—	Comparator negative input.
	IOCB3	TTL/ST	—	Interrupt-on-change input.
RB4/ANB4/ADCACT(1)/T5G(1)/ SMTWIN2(1)/IOCB4	RB4	TTL/ST	CMOS/OD	General purpose I/O.
	ANB4	AN	—	ADC Channel B4 input.
	ADCACT(1)	TTL/ST	—	ADC Auto-Conversion Trigger input.
	T5G(1)	TTL/ST	—	Timer5 gate input.
	SMTWIN2(1)	TTL/ST	—	Signal Measurement Timer2 (SMT2) window input.
	IOCB4	TTL/ST	—	Interrupt-on-change input.
RB5/ANB5/T1G(1)/SMTSIG2(1)/ CCP3(1)/IOCB5	RB5	TTL/ST	CMOS/OD	General purpose I/O.
	ANB5	AN	—	ADC Channel B5 input.
	T1G(1)	TTL/ST	—	Timer1 gate input.
	SMTSIG2(1)	TTL/ST	—	Signal Measurement Timer2 (SMT2) signal input.
	CCP3(1)	TTL/ST	CMOS/OD	Capture/compare/PWM3 (default input location for capture function).
	IOCB5	TTL/ST	—	Interrupt-on-change input.
RB6/ANB6/CLCIN2(1)/IOCB6/ ICSPCLK	RB6	TTL/ST	CMOS/OD	General purpose I/O.
	ANB6	AN	—	ADC Channel B6 input.
	CLCIN2(1)	TTL/ST	—	Configurable Logic Cell source input.
	IOCB6	TTL/ST	—	Interrupt-on-change input.
	ICSPCLK	ST	—	In-Circuit Serial Programming™ and debugging clock input.
RB7/ANB7/DAC1OUT2/T6IN(1)/ CLCIN3(1)/IOCB7/ICSPDAT	RB7	TTL/ST	CMOS/OD	General purpose I/O.
	ANB7	AN	—	ADC Channel B7 input.
	DAC1OUT2	—	AN	Digital-to-Analog Converter output.
	T6IN(1)	TTL/ST	—	Timer6 external digital clock input.
	CLCIN3(1)	TTL/ST	—	Configurable Logic Cell source input.
	IOCB7	TTL/ST	—	Interrupt-on-change input.
	ICSPDAT	ST	CMOS	In-Circuit Serial Programming™ and debugging data input/ output.

Legend: AN = Analog input or output CMOS = CMOS compatible input or output OD = Open-Drain
TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels I^2C = Schmitt Trigger input with I^2CHV= High Voltage XTAL= Crystal levels

Note 1: This is a PPS remappable input signal. The input function may be moved from the default location shown to one of several other PORTx pins. Refer to Table 13-1 for details on which PORT pins may be used for this signal.

2: All output signals shown in this row are PPS remappable. These signals may be mapped to output onto one of several PORTx pin options as described in Table 13-3.

3: This is a bidirectional signal. For normal module operation, the firmware should map this signal to the same pin in both the PPS input and PPS output registers.

4: These pins are configured for I^2C logic levels. The SCLx/SDAx signals may be assigned to any of the RB1/RB2/RC3/RC4 pins. PPS assignments to the other pins (e.g., RA5) will operate, but input logic levels will be standard TTL/ST, as selected by the INLVL register, instead of the I^2C specific or SMBus input buffer thresholds.

TABLE 3-13: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (CONTINUED)

Address	Name	PIC16(L)F18856	PIC16(L)F18876	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 28 (Continued)													
E22h	CLC2GLS2			LC2G3D4T	LC2G3D4N	LC2G3D3T	LC2G3D3N	LC2G3D2T	LC2G3D2N	LC2G3D1T	LC2G3D1N	xxxx xxxx	uuuu uuuu
E23h	CLC2GLS3			LC2G4D4T	LC2G4D4N	LC2G4D3T	LC2G4D3N	LC2G4D2T	LC2G4D2N	LC2G4D1T	LC2G4D1N	xxxx xxxx	uuuu uuuu
E24h	CLC3CON			LC3EN	—	LC3OUT	LC3INTP	LC3INTN	LC3MODE<2:0>			0-x0 0000	0-x0 0000
E25h	CLC3POL			LC3POL	—	—	—	LC3G4POL	LC3G3POL	LC3G2POL	LC3G1POL	0--- xxxx	0--- uuuu
E26h	CLC3SEL0			—	—	LC3D1S<5:0>						--xx xxxx	--uu uuuu
E27h	CLC3SEL1			—	—	LC3D2S<5:0>						--xx xxxx	--uu uuuu
E28h	CLC3SEL2			—	—	LC3D3S<5:0>						--xx xxxx	--uu uuuu
E29h	CLC3SEL3			—	—	LC3D4S<5:0>						--xx xxxx	--uu uuuu
E2Ah	CLC3GLS0			LC3G1D4T	LC3G1D4N	LC3G1D3T	LC3G1D3N	LC3G1D2T	LC3G1D2N	LC3G1D1T	LC3G1D1N	xxxx xxxx	uuuu uuuu
E2Bh	CLC3GLS1			LC3G2D4T	LC3G2D4N	LC3G2D3T	LC3G2D3N	LC3G2D2T	LC3G2D2N	LC3G2D1T	LC3G2D1N	xxxx xxxx	uuuu uuuu
E2Ch	CLC3GLS2			LC3G3D4T	LC3G3D4N	LC3G3D3T	LC3G3D3N	LC3G3D2T	LC3G3D2N	LC3G3D1T	LC3G3D1N	xxxx xxxx	uuuu uuuu
E2Dh	CLC3GLS3			LC3G4D4T	LC3G4D4N	LC3G4D3T	LC3G4D3N	LC3G4D2T	LC3G4D2N	LC3G4D1T	LC3G4D1N	xxxx xxxx	uuuu uuuu
E2Eh	CLC4CON			LC4EN	—	LC4OUT	LC4INTP	LC4INTN	LC4MODE<2:0>			0-x0 0000	0-x0 0000
E2Fh	CLC4POL			LC4POL	—	—	—	LC4G4POL	LC4G3POL	LC4G2POL	LC4G1POL	0--- xxxx	0--- uuuu
E30h	CLC4SEL0			—	—	LC4D1S<5:0>						--xx xxxx	--uu uuuu
E31h	CLC4SEL1			—	—	LC4D2S<5:0>						--xx xxxx	--uu uuuu
E32h	CLC4SEL2			—	—	LC4D3S<5:0>						--xx xxxx	--uu uuuu
E33h	CLC4SEL3			—	—	LC4D4S<5:0>						--xx xxxx	--uu uuuu
E34h	CLC4GLS0			LC4G1D4T	LC4G1D4N	LC4G1D3T	LC4G1D3N	LC4G1D2T	LC4G1D2N	LC4G1D1T	LC4G1D1N	xxxx xxxx	uuuu uuuu
E35h	CLC4GLS1			LC4G2D4T	LC4G2D4N	LC4G2D3T	LC4G2D3N	LC4G2D2T	LC4G2D2N	LC4G2D1T	LC4G2D1N	xxxx xxxx	uuuu uuuu
E36h	CLC4GLS2			LC4G3D4T	LC4G3D4N	LC4G3D3T	LC4G3D3N	LC4G3D2T	LC4G3D2N	LC4G3D1T	LC4G3D1N	xxxx xxxx	uuuu uuuu
E37h	CLC4GLS3			LC4G4D4T	LC4G4D4N	LC4G4D3T	LC4G4D3N	LC4G4D2T	LC4G4D2N	LC4G4D1T	LC4G4D1N	xxxx xxxx	uuuu uuuu
E38h to E6Fh	—	—		Unimplemented								—	—

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

Note 1: Register present on PIC16F18855/75 devices only.

2: Unimplemented, read as '1'.

7.1 Operation

Interrupts are disabled upon any device Reset. They are enabled by setting the following bits:

- GIE bit of the INTCON register
- Interrupt Enable bit(s) for the specific interrupt event(s)
- PEIE bit of the INTCON register (if the Interrupt Enable bit of the interrupt event is contained in the PIEx registers)

The PIR1, PIR2, PIR3 and PIR4 registers record individual interrupts via interrupt flag bits. Interrupt flag bits will be set, regardless of the status of the GIE, PEIE and individual interrupt enable bits.

The following events happen when an interrupt event occurs while the GIE bit is set:

- Current prefetched instruction is flushed
- GIE bit is cleared
- Current Program Counter (PC) is pushed onto the stack
- Critical registers are automatically saved to the shadow registers (See **"Section 7.5 "Automatic Context Saving"**)
- PC is loaded with the interrupt vector 0004h

The firmware within the Interrupt Service Routine (ISR) should determine the source of the interrupt by polling the interrupt flag bits. The interrupt flag bits must be cleared before exiting the ISR to avoid repeated interrupts. Because the GIE bit is cleared, any interrupt that occurs while executing the ISR will be recorded through its interrupt flag, but will not cause the processor to redirect to the interrupt vector.

The `RETFIE` instruction exits the ISR by popping the previous address from the stack, restoring the saved context from the shadow registers and setting the GIE bit.

For additional information on a specific interrupt's operation, refer to its peripheral chapter.

Note 1:	Individual interrupt flag bits are set, regardless of the state of any other enable bits.
2:	All interrupts will be ignored while the GIE bit is cleared. Any interrupt occurring while the GIE bit is clear will be serviced when the GIE bit is set again.

7.2 Interrupt Latency

Interrupt latency is defined as the time from when the interrupt event occurs to the time code execution at the interrupt vector begins. The latency for synchronous interrupts is three or four instruction cycles. For asynchronous interrupts, the latency is three to five instruction cycles, depending on when the interrupt occurs. See Figure 7-2 and Figure 7-3 for more details.

REGISTER 7-9: PIE7: PERIPHERAL INTERRUPT ENABLE REGISTER 7

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
SCANIE	CRCIE	NVMIE	NCO1IE	—	CWG3IE	CWG2IE	CWG1IE
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 = Hardware set

bit 7 **SCANIE:** Scanner Interrupt Enable bit
1 = Enables the scanner interrupt
0 = Disables the scanner interrupt

bit 6 **CRCIE:** CRC Interrupt Enable bit
1 = Enables the CRC interrupt
0 = Disables the CRC interrupt

bit 5 **NVMIE:** NVM Interrupt Enable bit
1 = NVM task complete interrupt enabled
0 = NVM interrupt not enabled

bit 4 **NCO1IE:** NCO Interrupt Enable bit
1 = NCO rollover interrupt enabled
0 = NCO rollover interrupt disabled

bit 3 **Unimplemented:** Read as '0'.

bit 2 **CWG3IE:** Complementary Waveform Generator (CWG) 3 Interrupt Enable bit
1 = CWG3 interrupt enabled
0 = CWG3 interrupt disabled

bit 1 **CWG2IE:** Complementary Waveform Generator (CWG) 2 Interrupt Enable bit
1 = CWG2 interrupt is enabled
0 = CWG2 interrupt disabled

bit 0 **CWG1IE:** Complementary Waveform Generator (CWG) 2 Interrupt Enable bit
1 = CWG1 interrupt is enabled
0 = CWG1 interrupt disabled

Note: Bit PEIE of the INTCON register must be set to enable any peripheral interrupt controlled by registers PIE1-PIE8.

12.8.8 CURRENT-CONTROLLED DRIVE MODE CONTROL

The CCDPC and CCDNC registers (Register 12-30 and Register 12-31) control the Current-Controlled Drive mode for both the positive-going and negative-going drivers. When a CCDPC[y] or CCDNC[y] bit is set and the CCDEN bit of the CCDCON register is set, the Current-Controlled mode is enabled for the corresponding port pin. When the CCDPC[y] or CCDNC[y] bit is clear, the Current-Controlled mode for the corresponding port pin is disabled. If the CCDPC[y] or CCDNC[y] bit is set and the CCDEN bit is clear, operation of the port pin is undefined (see **Section 12.1.1 "Current-Controlled Drive"** for current-controlled use precautions).

12.8.9 PORTC FUNCTIONS AND OUTPUT PRIORITIES

Each pin defaults to the PORT latch data after Reset. Other output functions are selected with the peripheral pin select logic. See **Section 13.0 "Peripheral Pin Select (PPS) Module"** for more information.

Analog input functions, such as ADC and comparator inputs, are not shown in the peripheral pin select lists. Digital output functions may continue to control the pin when it is in Analog mode.

18.12 Register Definitions: Comparator Control

REGISTER 18-1: CMxCON0: COMPARATOR Cx CONTROL REGISTER 0

R/W-0/0	R-0/0	U-0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
ON	OUT	—	POL	—	—	HYS	SYNC
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 **ON:** Comparator Enable bit
1 = Comparator is enabled
0 = Comparator is disabled and consumes no active power

bit 6 **OUT:** Comparator Output bit
If CxPOL = 1 (inverted polarity):
1 = CxVP < CxVN
0 = CxVP > CxVN
If CxPOL = 0 (non-inverted polarity):
1 = CxVP > CxVN
0 = CxVP < CxVN

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

bit 4 **POL:** Comparator Output Polarity Select bit
1 = Comparator output is inverted
0 = Comparator output is not inverted

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

bit 1 **HYS:** Comparator Hysteresis Enable bit
1 = Comparator hysteresis enabled
0 = Comparator hysteresis disabled

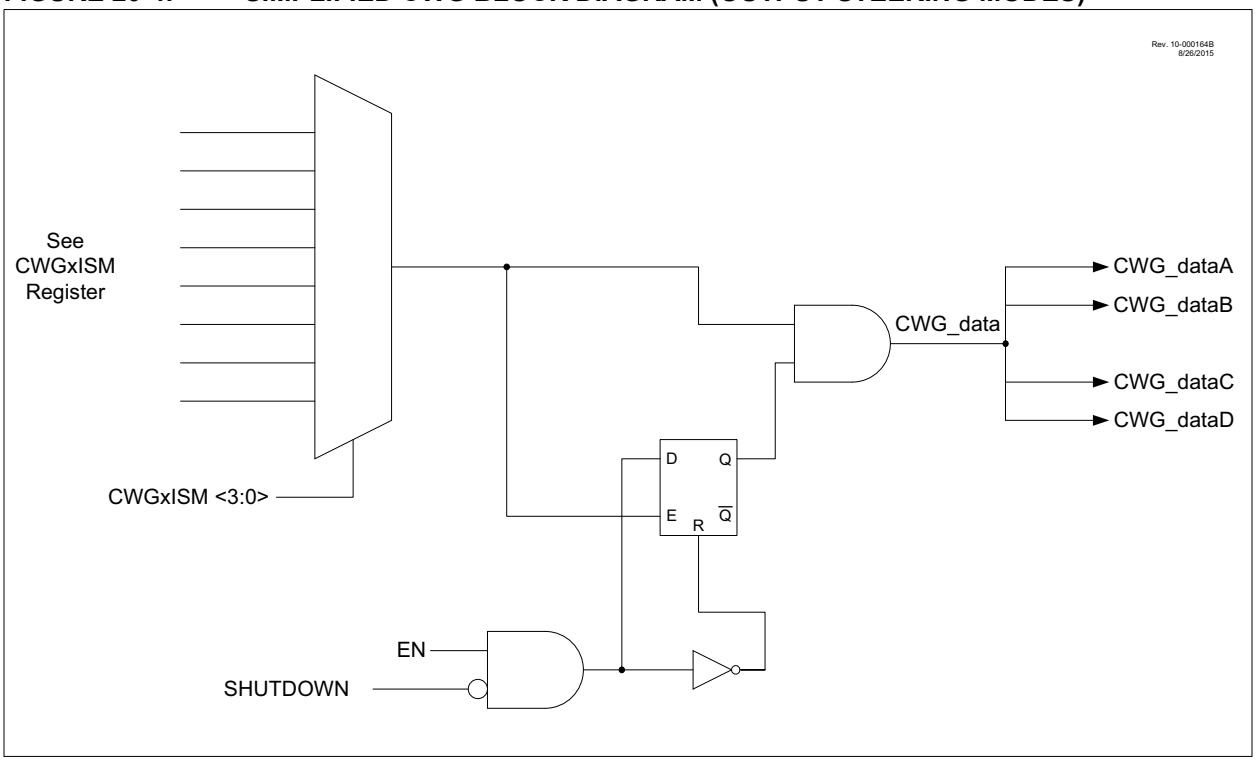
bit 0 **SYNC:** Comparator Output Synchronous Mode bit
1 = Comparator output to Timer1 and I/O pin is synchronous to changes on Timer1 clock source. Output updated on the falling edge of Timer1 clock source.
0 = Comparator output to Timer1 and I/O pin is asynchronous

20.1.4 STEERING MODES

In Steering modes, the data input can be steered to any or all of the four CWG output pins. In Synchronous Steering mode, changes to steering selection registers take effect on the next rising input.

In Non-Synchronous mode, steering takes effect on the next instruction cycle. Additional details are provided in **Section 20.9 "CWG Steering Mode"**.

FIGURE 20-4: SIMPLIFIED CWG BLOCK DIAGRAM (OUTPUT STEERING MODES)



20.2 Clock Source

The CWG module allows the following clock sources to be selected:

- Fosc (system clock)
- HFINTOSC (16 MHz only)

The clock sources are selected using the CS bit of the CWGxCLKCON register.

20.12 Configuring the CWG

The following steps illustrate how to properly configure the CWG.

1. Ensure that the TRIS control bits corresponding to the desired CWG pins for your application are set so that the pins are configured as inputs.
2. Clear the EN bit, if not already cleared.
3. Set desired mode of operation with the MODE bits.
4. Set desired dead-band times, if applicable to mode, with the CWGxDBR and CWGxDBF registers.
5. Setup the following controls in the CWGxAS0 and CWGxAS1 registers.
   a. Select the desired shutdown source.
   b. Select both output overrides to the desired levels (this is necessary even if not using auto-shutdown because start-up will be from a shutdown state).
   c. Set which pins will be affected by auto-shutdown with the CWGxAS1 register.
   d. Set the SHUTDOWN bit and clear the REN bit.
6. Select the desired input source using the CWGxISM register.
7. Configure the following controls.
   a. Select desired clock source using the CWGxCLKCON register.
   b. Select the desired output polarities using the CWGxCON1 register.
   c. Set the output enables for the desired outputs.
8. Set the EN bit.
9. Clear TRIS control bits corresponding to the desired output pins to configure these pins as outputs.
10. If auto-restart is to be used, set the REN bit and the SHUTDOWN bit will be cleared automatically. Otherwise, clear the SHUTDOWN bit to start the CWG.

20.12.1 PIN OVERRIDE LEVELS

The levels driven to the output pins, while the shutdown input is true, are controlled by the LSBD and LSAC bits of the CWGxAS0 register. LSBD<1:0> controls the CWGxB and D override levels and LSAC<1:0> controls the CWGxA and C override levels. The control bit logic level corresponds to the output logic drive level while in the shutdown state. The polarity control does not affect the override level.

20.12.2 AUTO-SHUTDOWN RESTART

After an auto-shutdown event has occurred, there are two ways to resume operation:

- Software controlled
- Auto-restart

The restart method is selected with the REN bit of the CWGxCON2 register. Waveforms of software controlled and automatic restarts are shown in Figure 20-13 and Figure 20-14.

20.12.2.1 Software Controlled Restart

When the REN bit of the CWGxAS0 register is cleared, the CWG must be restarted after an auto-shutdown event by software. Clearing the shutdown state requires all selected shutdown inputs to be low, otherwise the SHUTDOWN bit will remain set. The overrides will remain in effect until the first rising edge event after the SHUTDOWN bit is cleared. The CWG will then resume operation.

20.12.2.2 Auto-Restart

When the REN bit of the CWGxCON2 register is set, the CWG will restart from the auto-shutdown state automatically. The SHUTDOWN bit will clear automatically when all shutdown sources go low. The overrides will remain in effect until the first rising edge event after the SHUTDOWN bit is cleared. The CWG will then resume operation.

23.1 ADC Configuration

When configuring and using the ADC the following functions must be considered:

- Port configuration
- Channel selection
- ADC voltage reference selection
- ADC conversion clock source
- Interrupt control
- Result formatting
- Conversion Trigger Selection
- ADC Acquisition Time
- ADC Precharge Time
- Additional Sample and Hold Capacitor
- Single/Double Sample Conversion
- Guard Ring Outputs

23.1.1 PORT CONFIGURATION

The ADC can be used to convert both analog and digital signals. When converting analog signals, the I/O pin should be configured for analog by setting the associated TRIS and ANSEL bits. Refer to **Section 12.0 "I/O Ports"** for more information.

Note:	Analog voltages on any pin that is defined as a digital input may cause the input buffer to conduct excess current.

23.1.2 CHANNEL SELECTION

There are several channel selections available:

- Eight PORTA pins (RA<7:0>)
- Eight PORTB pins (RB<7:0>)
- Eight PORTC pins (RC<7:0>)
- Eight PORTD pins (RD<7:0>, PIC16(L)F18875 only)
- Three PORTE pins (RE<2:0>, PIC16(L)F18875 only)
- Temperature Indicator
- DAC output
- Fixed Voltage Reference (FVR)
- AVSS (ground)

The ADPCH register determines which channel is connected to the sample and hold circuit.

When changing channels, a delay is required before starting the next conversion. Refer to **Section 23.2 "ADC Operation"** for more information.

Note:	It is recommended that when switching from an ADC channel of a higher voltage to a channel of a lower voltage, the software selects the VSS channel before switching to the channel of the lower voltage. If the ADC does not have a dedicated VSS input channel, the VSS selection (DAC1R<4:0> = b'00000') through the DAC output channel can be used. If the DAC is in use, a free input channel can be connected to VSS, and can be used in place of the DAC.

23.1.3 ADC VOLTAGE REFERENCE

The ADPREF bits of the ADREF register provides control of the positive voltage reference. The positive voltage reference can be:

- VREF+ pin
- VDD
- FVR 1.024V
- FVR 2.048V
- FVR 4.096V

The ADNREF bit of the ADREF register provides control of the negative voltage reference. The negative voltage reference can be:

- VREF- pin
- VSS

See **Section 16.0 "Fixed Voltage Reference (FVR)"** for more details on the Fixed Voltage Reference.

23.1.4 CONVERSION CLOCK

The source of the conversion clock is software selectable via the ADCLK register and the ADCS bit of the ADCON0 register. There are two possible clock sources:

- FOSC/(2*(n+1)) (where n is from 0 to 63),
- FRC (dedicated RC oscillator)

The time to complete one bit conversion is defined as TAD. One full 10-bit conversion requires 11.5 TAD periods as shown in Figure 23-2.

For correct conversion, the appropriate TAD specification must be met. Refer to Table 37-13 for more information. Table 23-1 gives examples of appropriate ADC clock selections.

Note 1:	Unless using the FRC, any changes in the system clock frequency will change the ADC clock frequency, which may adversely affect the ADC result.
2:	The internal control logic of the ADC runs off of the clock selected by the ADCS bit of ADCON0. What this can mean is when the ADCS bit of ADCON0 is set to 1 (ADC runs on FRC), there may be unexpected delays in operation when setting ADC control bits.

REGISTER 26-4: MDCARH: MODULATION HIGH CARRIER CONTROL REGISTER

U-0	U-0	U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	—	—	MDCHS<3:0>(1)			
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 **MDCHS<3:0>** Modulator Data High Carrier Selection bits (1)

1111 = LC4_out
1110 = LC3_out
1101 = LC2_out
1100 = LC1_out
1011 = NCO output
1010 = PWM7_out
1001 = PWM6_out
1000 = CCP5 output (PWM Output mode only)
0111 = CCP4 output (PWM Output mode only)
0110 = CCP3 output (PWM Output mode only)
0101 = CCP2 output (PWM Output mode only)
0100 = CCP1 output (PWM Output mode only)
0011 = Reference clock module signal (CLKR)
0010 = HFINTOSC
0001 = FOSC
0000 = Pin selected by MDCARHPPS

Note 1: Narrowed carrier pulse widths or spurs may occur in the signal stream if the carrier is not synchronized.

29.5.6 EDGE-TRIGGERED ONE-SHOT MODE

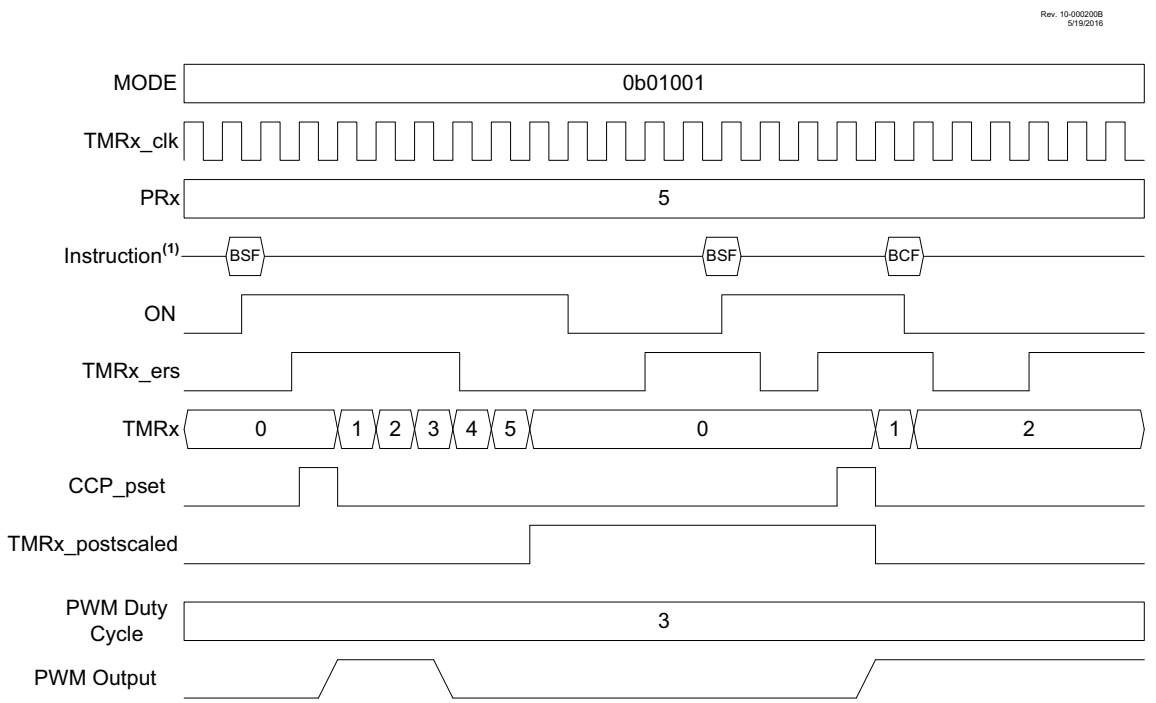
The Edge-Triggered One-Shot modes start the timer on an edge from the external signal input, after the ON bit is set, and clear the ON bit when the timer matches the PRx period value. The following edges will start the timer:

- Rising edge (MODE<4:0> = 01001)
- Falling edge (MODE<4:0> = 01010)
- Rising or Falling edge (MODE<4:0> = 01011)

If the timer is halted by clearing the ON bit then another TMRx_ers edge is required after the ON bit is set to resume counting. Figure 29-9 illustrates operation in the rising edge One-Shot mode.

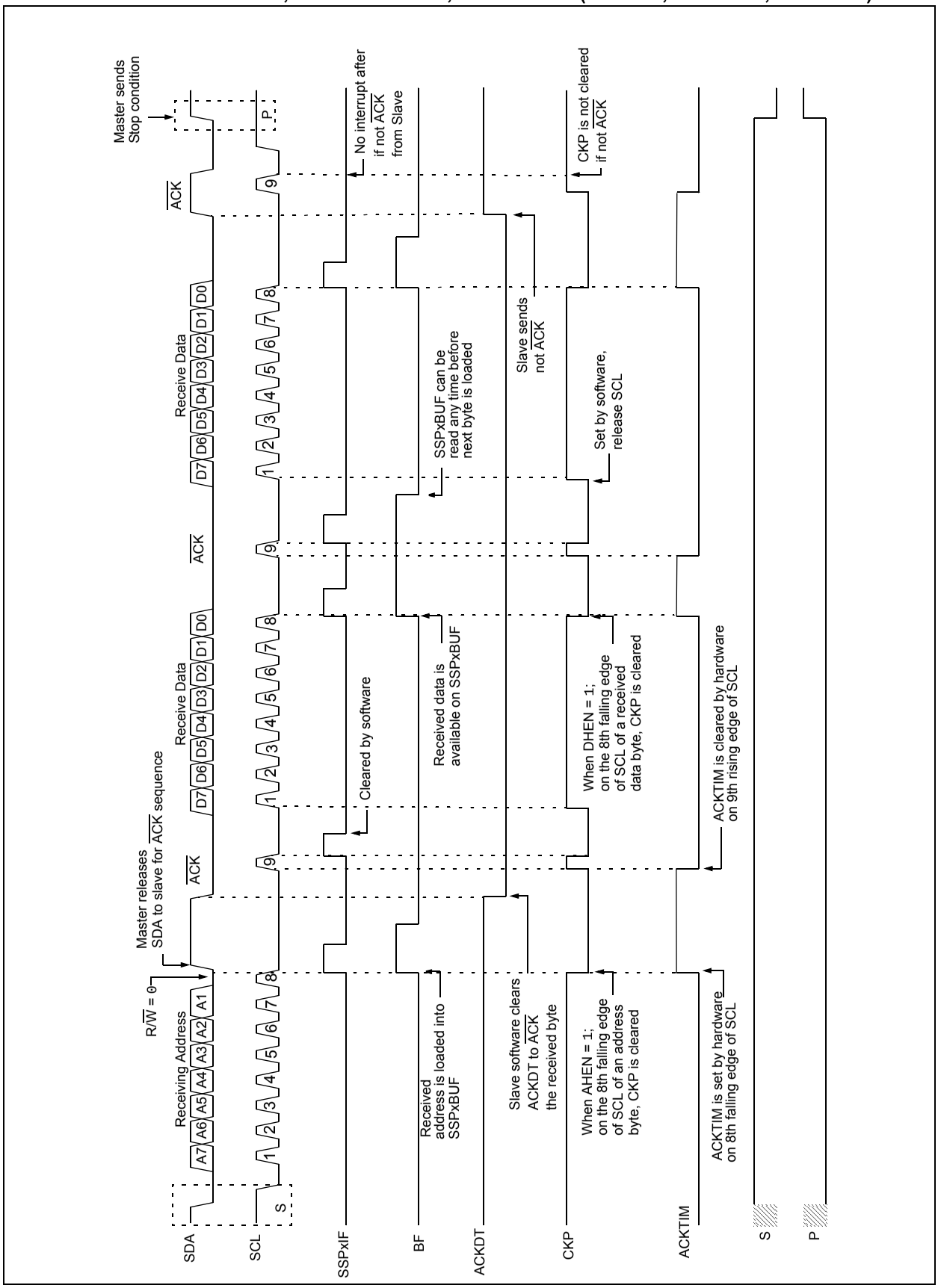
When Edge-Triggered One-Shot mode is used in conjunction with the CCP then the edge-trigger will activate the PWM drive and the PWM drive will deactivate when the timer matches the CCPRx pulse width value and stay deactivated when the timer halts at the PRx period count match.

FIGURE 29-9: EDGE-TRIGGERED ONE-SHOT MODE TIMING DIAGRAM (MODE = 01001)



Note 1: BSF and BCF represent Bit-Set File and Bit-Clear File instructions executed by the CPU to set or clear the ON bit of TxCON. CPU execution is asynchronous to the timer clock input.

FIGURE 31-17: I²C SLAVE, 7-BIT ADDRESS, RECEPTION (SEN = 1, AHEN = 1, DHEN = 1)



31.5.3.3 7-bit Transmission with Address Hold Enabled

Setting the AHEN bit of the SSPxCON3 register enables additional clock stretching and interrupt generation after the eighth falling edge of a received matching address. Once a matching address has been clocked in, CKP is cleared and the SSPxIF interrupt is set.

Figure 31-19 displays a standard waveform of a 7-bit address slave transmission with AHEN enabled.

1. Bus starts Idle.
2. Master sends Start condition; the S bit of SSPxSTAT is set; SSPxIF is set if interrupt on Start detect is enabled.
3. Master sends matching address with R/$\overline{W}$ bit set. After the eighth falling edge of the SCL line the CKP bit is cleared and SSPxIF interrupt is generated.
4. Slave software clears SSPxIF.
5. Slave software reads ACKTIM bit of SSPxCON3 register, and R/$\overline{W}$ and D/$\overline{A}$ of the SSPxSTAT register to determine the source of the interrupt.
6. Slave reads the address value from the SSPxBUF register clearing the BF bit.
7. Slave software decides from this information if it wishes to $\overline{ACK}$ or not $\overline{ACK}$ and sets the ACKDT bit of the SSPxCON2 register accordingly.
8. Slave sets the CKP bit releasing SCL.
9. Master clocks in the $\overline{ACK}$ value from the slave.
10. Slave hardware automatically clears the CKP bit and sets SSPxIF after the $\overline{ACK}$ if the R/$\overline{W}$ bit is set.
11. Slave software clears SSPxIF.
12. Slave loads value to transmit to the master into SSPxBUF setting the BF bit.

Note: SSPxBUF cannot be loaded until after the $\overline{ACK}$.

13. Slave sets the CKP bit releasing the clock.
14. Master clocks out the data from the slave and sends an $\overline{ACK}$ value on the ninth SCL pulse.
15. Slave hardware copies the $\overline{ACK}$ value into the ACKSTAT bit of the SSPxCON2 register.
16. Steps 10-15 are repeated for each byte transmitted to the master from the slave.
17. If the master sends a not $\overline{ACK}$ the slave releases the bus allowing the master to send a Stop and end the communication.

Note: Master must send a not $\overline{ACK}$ on the last byte to ensure that the slave releases the SCL line to receive a Stop.

FIGURE 32-1: SMT BLOCK DIAGRAM

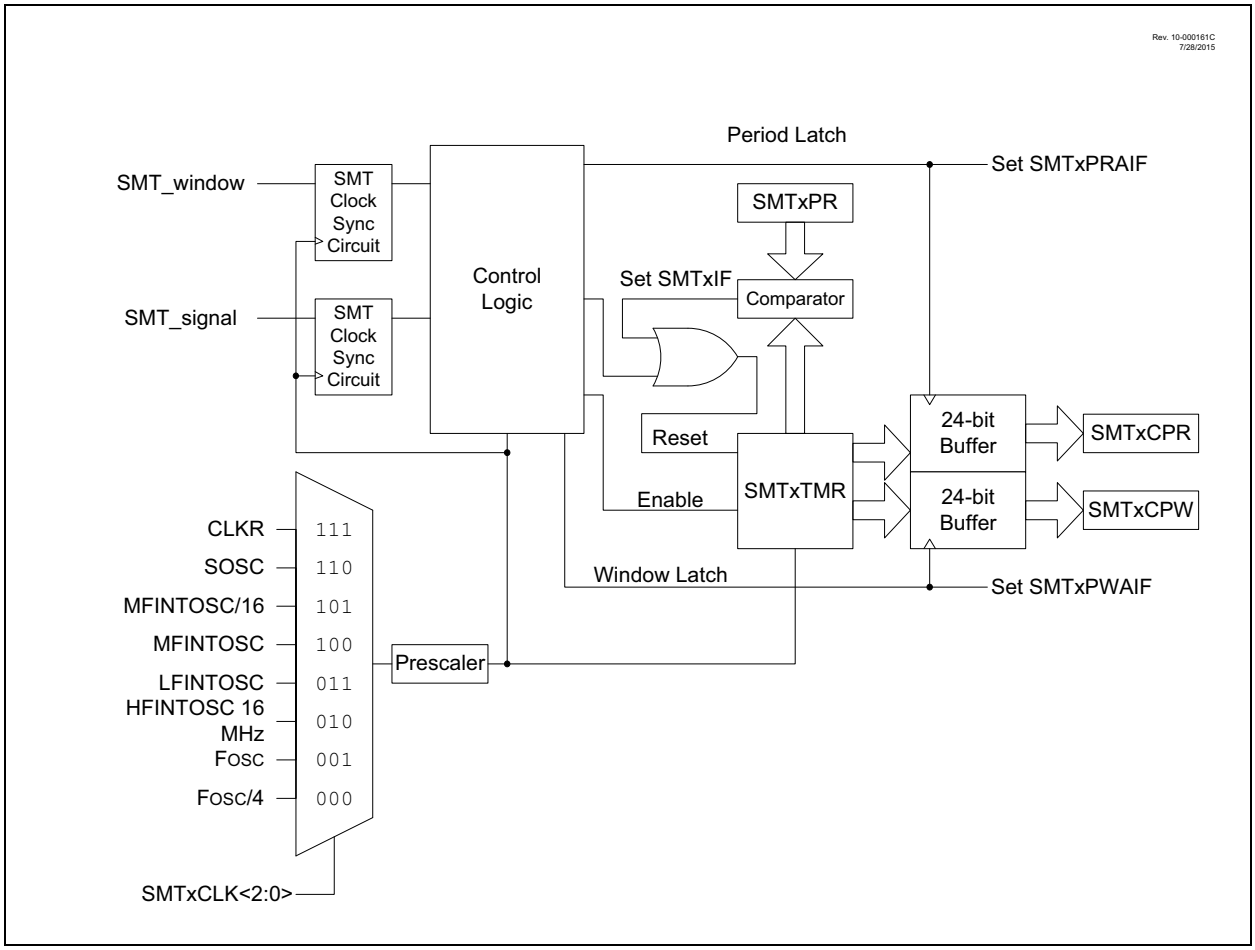
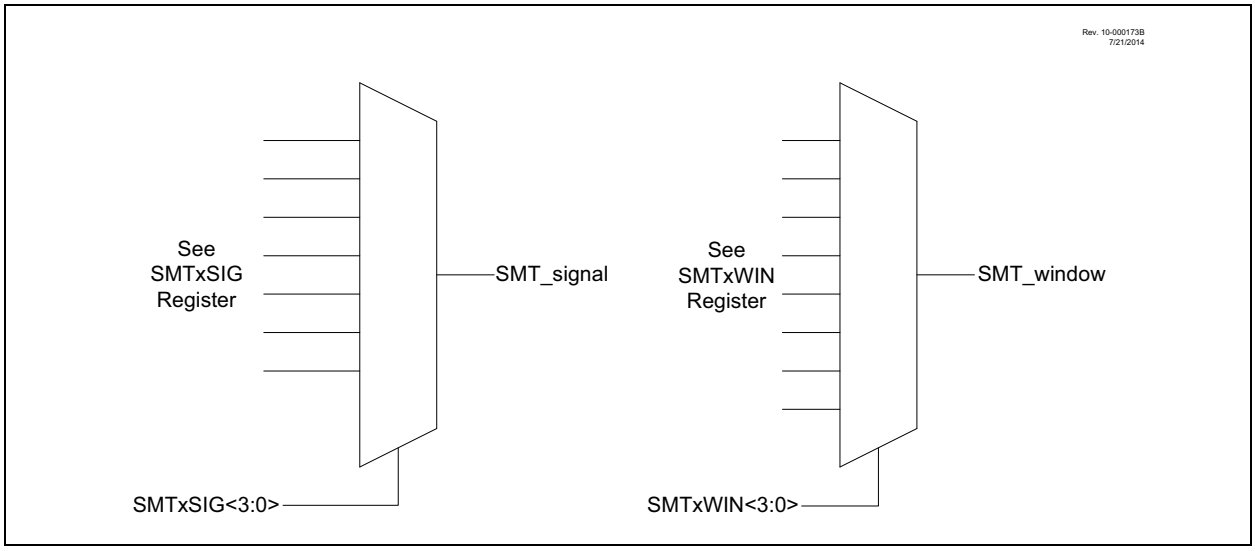


FIGURE 32-2: SMT SIGNAL AND WINDOW BLOCK DIAGRAM



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: $W \rightarrow INDFn$
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 $\overline{RI}$ 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: $TOS \rightarrow PC$,
$1 \rightarrow GIE$

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

XORLW	Exclusive OR literal with W
Syntax:	[*label*] XORLW k
Operands:	$0 \le k \le 255$
Operation:	(W) .XOR. k $\rightarrow$ (W)
Status Affected:	Z
Description:	The contents of the W register are XOR'ed with the 8-bit literal 'k'. The result is placed in the W register.

XORWF	Exclusive OR W with f
Syntax:	[*label*] XORWF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(W) .XOR. (f) $\rightarrow$ (destination)
Status Affected:	Z
Description:	Exclusive OR the contents of the W register with 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'.

Note: Unless otherwise noted, VIN = 5V, FOSC = 300 kHz, CIN = 0.1 µF, TA = 25°C.

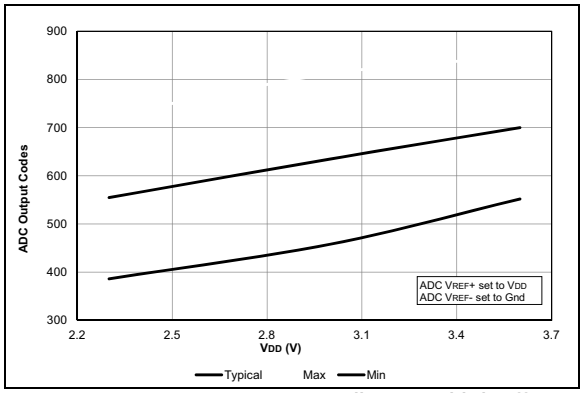


FIGURE 38-43: *Temp. Indicator Initial Offset, Low Range, Temp. = 20°C, PIC16LF18856/76 Only.*

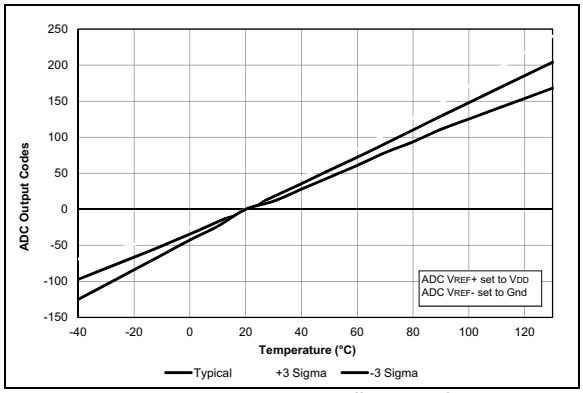


FIGURE 38-46: *Temp. Indicator Slope Normalized to 20°C, High Range, VDD = 3.0V.*

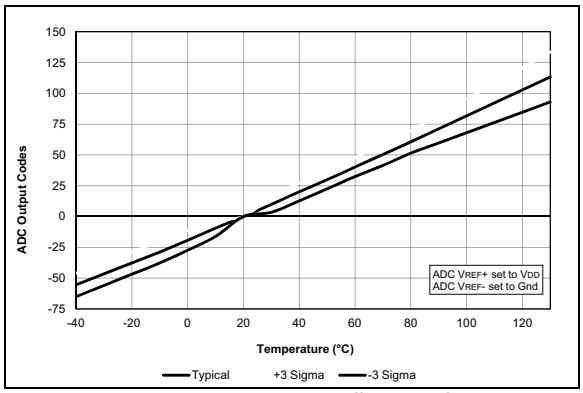


FIGURE 38-44: *Temp. Indicator Slope Normalized to 20°C, High Range, VDD = 5.5V, PIC16F18856/76 Only.*

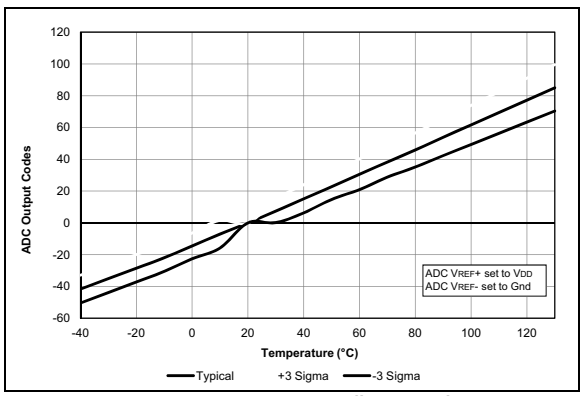


FIGURE 38-47: *Temp. Indicator Slope Normalized to 20°C, Low Range, VDD = 3.6V.*

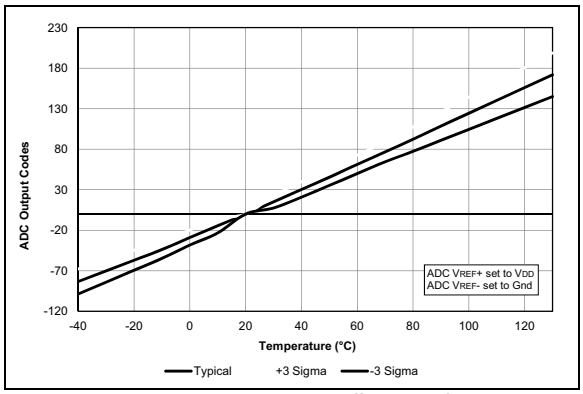


FIGURE 38-45: *Temp. Indicator Slope Normalized to 20°C, High Range, VDD = 3.6V.*

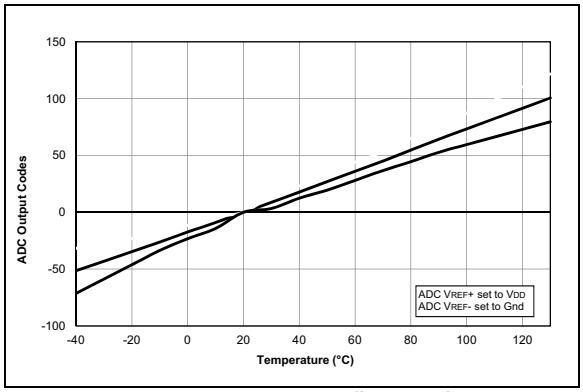


FIGURE 38-48: *Temp. Indicator Slope Normalized to 20°C, Low Range, VDD = 3.0V.*

39.6 MPLAB X SIM Software Simulator

The MPLAB X SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB X SIM Software Simulator fully supports symbolic debugging using the MPLAB XC Compilers, and the MPASM and MPLAB Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

39.7 MPLAB REAL ICE In-Circuit Emulator System

The MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs all 8, 16 and 32-bit MCU, and DSC devices with the easy-to-use, powerful graphical user interface of the MPLAB X IDE.

The emulator is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ-11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradeable through future firmware downloads in MPLAB X IDE. MPLAB REAL ICE offers significant advantages over competitive emulators including full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, logic probes, a ruggedized probe interface and long (up to three meters) interconnection cables.

39.8 MPLAB ICD 3 In-Circuit Debugger System

The MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost-effective, high-speed hardware debugger/programmer for Microchip Flash DSC and MCU devices. It debugs and programs PIC Flash microcontrollers and dsPIC DSCs with the powerful, yet easy-to-use graphical user interface of the MPLAB IDE.

The MPLAB ICD 3 In-Circuit Debugger probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with a connector compatible with the MPLAB ICD 2 or MPLAB REAL ICE systems (RJ-11). MPLAB ICD 3 supports all MPLAB ICD 2 headers.

39.9 PICkit 3 In-Circuit Debugger/ Programmer

The MPLAB PICkit 3 allows debugging and programming of PIC and dsPIC Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB IDE. The MPLAB PICkit 3 is connected to the design engineer's PC using a full-speed USB interface and can be connected to the target via a Microchip debug (RJ-11) connector (compatible with MPLAB ICD 3 and MPLAB REAL ICE). The connector uses two device I/O pins and the Reset line to implement in-circuit debugging and In-Circuit Serial Programming™ (ICSP™).

39.10 MPLAB PM3 Device Programmer

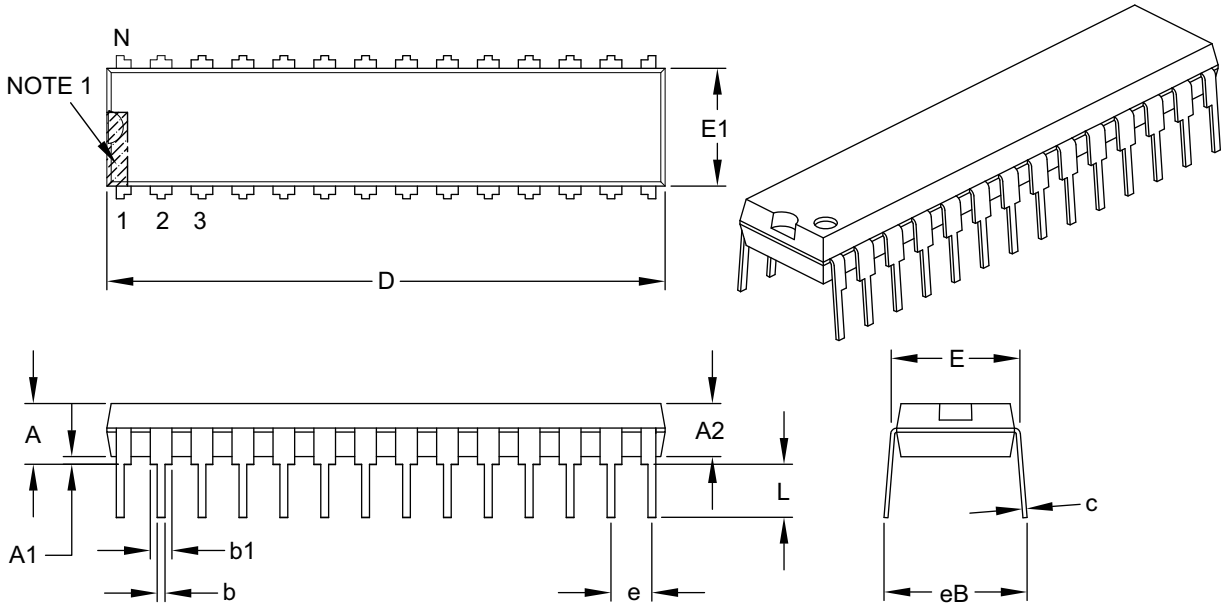
The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at V_{DDMIN} and V_{DDMAX} for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages, and a modular, detachable socket assembly to support various package types. The ICSP cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices, and incorporates an MMC card for file storage and data applications.

40.2 Package Details

The following sections give the technical details of the packages.

28-Lead Skinny Plastic Dual In-Line (SP) – 300 mil Body [SPDIP]

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



Units		INCHES		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		28	
Pitch	e		.100 BSC	
Top to Seating Plane	A	–	–	.200
Molded Package Thickness	A2	.120	.135	.150
Base to Seating Plane	A1	.015	–	–
Shoulder to Shoulder Width	E	.290	.310	.335
Molded Package Width	E1	.240	.285	.295
Overall Length	D	1.345	1.365	1.400
Tip to Seating Plane	L	.110	.130	.150
Lead Thickness	c	.008	.010	.015
Upper Lead Width	b1	.040	.050	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	–	–	.430

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-070B