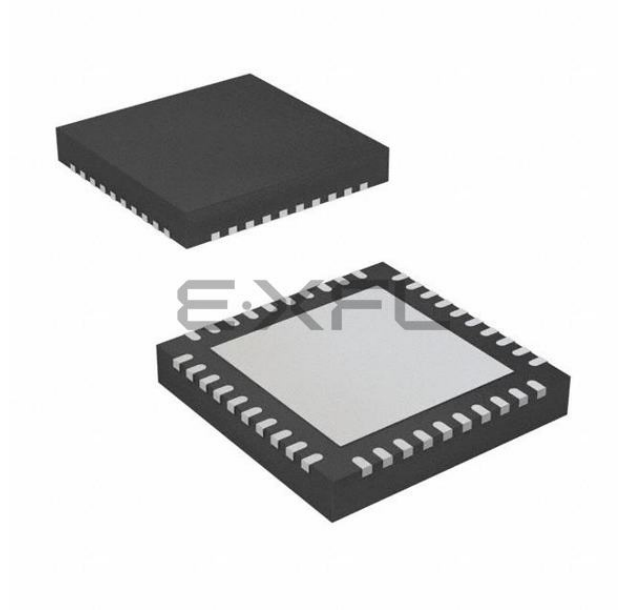


Welcome to E-XFL.COM

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	35
Program Memory Size	28KB (16K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 28x10b; D/A 1x5b, 1x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	40-UFQFN Exposed Pad
Supplier Device Package	40-UQFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f1719t-i-mv

TABLE 3-12: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Addr.	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 29											
E8Ch — E8Fh	—	Unimplemented								—	—
E90h	RA0PPS	—	—	—	RA0PPS<4:0>					---0 0000	---u uuuu
E91h	RA1PPS	—	—	—	RA1PPS<4:0>					---0 0000	---u uuuu
E92h	RA2PPS	—	—	—	RA2PPS<4:0>					---0 0000	---u uuuu
E93h	RA3PPS	—	—	—	RA3PPS4:0>					---0 0000	---u uuuu
E94h	RA4PPS	—	—	—	RA4PPS<4:0>					---0 0000	---u uuuu
E95h	RA5PPS	—	—	—	RA5PPS<4:0>					---0 0000	---u uuuu
E96h	RA6PPS	—	—	—	RA6PPS<4:0>					---0 0000	---u uuuu
E97h	RA7PPS	—	—	—	RA7PPS<4:0>					---0 0000	---u uuuu
E98h	RB0PPS	—	—	—	RB0PPS<4:0>					---0 0000	---u uuuu
E99h	RB1PPS	—	—	—	RB1PPS<4:0>					---0 0000	---u uuuu
E9Ah	RB2PPS	—	—	—	RB2PPS<4:0>					---0 0000	---u uuuu
E9Bh	RB3PPS	—	—	—	RB3PPS<4:0>					---0 0000	---u uuuu
E9Ch	RB4PPS	—	—	—	RB4PPS<4:0>					---0 0000	---u uuuu
E9Dh	RB5PPS	—	—	—	RB5PPS<4:0>					---0 0000	---u uuuu
E9Eh	RB6PPS	—	—	—	RB6PPS<4:0>					---0 0000	---u uuuu
E9Fh	RB7PPS	—	—	—	RB7PPS<4:0>					---0 0000	---u uuuu
EA0h	RC0PPS	—	—	—	RC0PPS<4:0>					---0 0000	---u uuuu
EA1h	RC1PPS	—	—	—	RC1PPS<4:0>					---0 0000	---u uuuu
EA2h	RC2PPS	—	—	—	RC2PPS<4:0>					---0 0000	---u uuuu
EA3h	RC3PPS	—	—	—	RC3PPS<4:0>					---0 0000	---u uuuu
EA4h	RC4PPS	—	—	—	RC4PPS<4:0>					---0 0000	---u uuuu
EA5h	RC5PPS	—	—	—	RC5PPS<4:0>					---0 0000	---u uuuu
EA6h	RC6PPS	—	—	—	RC6PPS<4:0>					---0 0000	---u uuuu
EA7h	RC7PPS	—	—	—	RC7PPS<4:0>					---0 0000	---u uuuu
EA8h	RD0PPS(1)	—	—	—	RD0PPS<4:0>					---0 0000	---u uuuu
EA9h	RD1PPS(1)	—	—	—	RD1PPS<4:0>					---0 0000	---u uuuu
EAAh	RD2PPS(1)	—	—	—	RD2PPS<4:0>					---0 0000	---u uuuu
EABh	RD3PPS(1)	—	—	—	RD3PPS<4:0>					---0 0000	---u uuuu
EACh	RD4PPS(1)	—	—	—	RD4PPS<4:0>					---0 0000	---u uuuu
EADh	RD5PPS(1)	—	—	—	RD5PPS<4:0>					---0 0000	---u uuuu
EAEh	RD6PPS(1)	—	—	—	RD6PPS<4:0>					---0 0000	---u uuuu
EAFh	RD7PPS(1)	—	—	—	RD7PPS<4:0>					---0 0000	---u uuuu
EB0h	RE0PPS(1)	—	—	—	RE0PPS<4:0>					---0 0000	---u uuuu
EB1h	RE1PPS(1)	—	—	—	RE1PPS<4:0>					---0 0000	---u uuuu
EB2h	RE2PPS(1)	—	—	—	RE2PPS<4:0>					---0 0000	---u uuuu
EB3h — EEFh	—	Unimplemented								—	—

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 on PIC16(L)F1718.

2: Unimplemented on PIC16LF1717/8/9

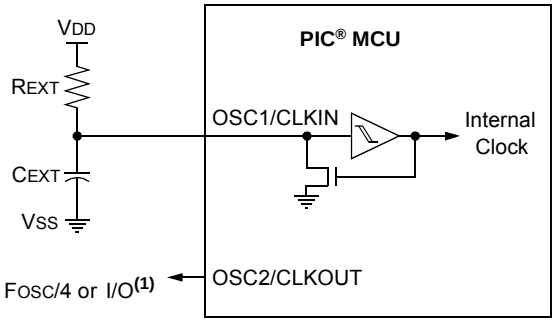
6.2.1.6 External RC Mode

The external Resistor-Capacitor (EXTRC) mode supports the use of an external RC circuit. This allows the designer maximum flexibility in frequency choice while keeping costs to a minimum when clock accuracy is not required.

The RC circuit connects to OSC1. OSC2/CLKOUT is available for general purpose I/O or CLKOUT. The function of the OSC2/CLKOUT pin is determined by the $\overline{\text{CLKOUTEN}}$ bit in Configuration Words.

Figure 6-6 shows the external RC mode connections.

FIGURE 6-6: EXTERNAL RC MODES



Recommended values: 10 kΩ ≤ REXT ≤ 100 kΩ, <3V
3 kΩ ≤ REXT ≤ 100 kΩ, 3-5V
CEXT > 20 pF, 2-5V

Note 1: Output depends upon $\overline{\text{CLKOUTEN}}$ bit of the Configuration Words.

The RC oscillator frequency is a function of the supply voltage, the resistor (REXT) and capacitor (CEXT) values and the operating temperature. Other factors affecting the oscillator frequency are:

- Threshold voltage variation
- Component tolerances
- Packaging variations in capacitance

The user also needs to take into account variation due to tolerance of external RC components used.

6.2.2 INTERNAL CLOCK SOURCES

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

- Program the FOSC<2:0> bits in Configuration Words to select the INTOSC clock source, which will be used as the default system clock upon a device Reset.
- Write the SCS<1:0> bits in the OSCCON register to switch the system clock source to the internal oscillator during run-time. See **Section 6.3 "Clock Switching"** for more information.

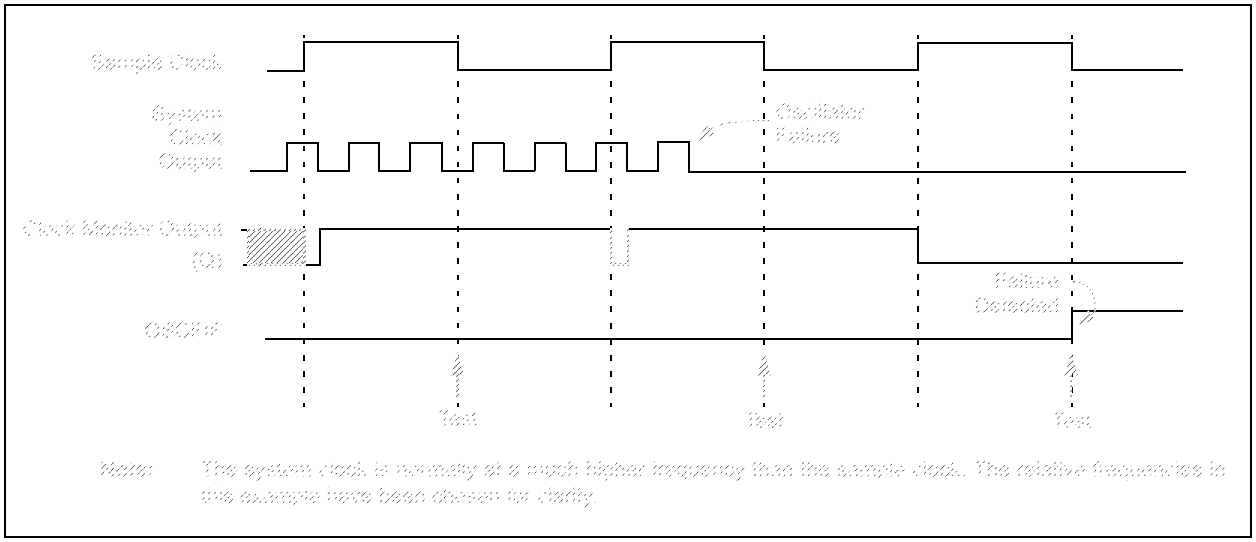
In **INTOSC** mode, OSC1/CLKIN is available for general purpose I/O. OSC2/CLKOUT is available for general purpose I/O or CLKOUT.

The function of the OSC2/CLKOUT pin is determined by the $\overline{\text{CLKOUTEN}}$ bit in Configuration Words.

The internal oscillator block has two independent oscillators and a dedicated Phase-Lock Loop, HFPLL that can produce one of three internal system clock sources.

1. The **HFINTOSC** (High-Frequency Internal Oscillator) is factory calibrated and operates at 16 MHz. The HFINTOSC source is generated from the 500 kHz MFINTOSC source and the dedicated Phase-Lock Loop, HFPLL. The frequency of the HFINTOSC can be user-adjusted via software using the OSCTUNE register (Register 6-3).
2. The **MFINTOSC** (Medium Frequency Internal Oscillator) is factory calibrated and operates at 500 kHz. The frequency of the MFINTOSC can be user-adjusted via software using the OSCTUNE register (Register 6-3).
3. The **LFINTOSC** (Low-Frequency Internal Oscillator) is uncalibrated and operates at 31 kHz.

FIGURE 6-10: FSCM TIMING DIAGRAM



Note: The system clock is normally at a much higher frequency than the sample clock. The relative frequencies in this example have been chosen for clarity.

REGISTER 11-28: ANSELD: PORTD ANALOG SELECT REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
ANSD7	ANSD6	ANSD5	ANSD4	ANSD3	ANSD2	ANSD1	ANSD0
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 **ANSD<7:0>**: Analog Select between Analog or Digital Function on Pins RD<7:0>, respectively(1)
0 = Digital I/O. Pin is assigned to port or digital special function.
1 = Analog input. Pin is assigned as analog input(1). Digital input buffer disabled.

Note 1: When setting a pin to an analog input, the corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.

REGISTER 11-29: WPUD: WEAK PULL-UP PORTD REGISTER(1,2)

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
WPUD7	WPUD6	WPUD5	WPUD4	WPUD3	WPUD2	WPUD1	WPUD0
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 **WPUD<7:0>**: Weak Pull-up Register bits
1 = Pull-up enabled
0 = Pull-up disabled

Note 1: Global $\overline{\text{WPUEN}}$ bit of the OPTION_REG register must be cleared for individual pull-ups to be enabled.
2: The weak pull-up device is automatically disabled if the pin is configured as an output.

18.0 COMPLEMENTARY OUTPUT GENERATOR (COG) MODULE

The primary purpose of the Complementary Output Generator (COG) is to convert a single output PWM signal into a two output complementary PWM signal. The COG can also convert two separate input events into a single or complementary PWM output.

The COG PWM frequency and duty cycle are determined by a rising event input and a falling event input. The rising event and falling event may be the same source. Sources may be synchronous or asynchronous to the COG_clock.

The rate at which the rising event occurs determines the PWM frequency. The time from the rising event input to the falling event input determines the duty cycle.

A selectable clock input is used to generate the phase delay, blanking, and dead-band times. Dead-band time can also be generated with a programmable time delay, which is independent from all clock sources.

Simplified block diagrams of the various COG modes are shown in Figure 18-2 through Figure 18-6.

The COG module has the following features:

- Six modes of operation:
  - Steered PWM mode
  - Synchronous Steered PWM mode
  - Forward Full-Bridge mode
  - Reverse Full-Bridge mode
  - Half-Bridge mode
  - Push-Pull mode
- Selectable COG_clock clock source
- Independently selectable rising event sources
- Independently selectable falling event sources
- Independently selectable edge or level event sensitivity
- Independent output polarity selection
- Phase delay with independent rising and falling delay times
- Dead-band control with:
  - independent rising and falling event dead-band times
  - Synchronous and asynchronous timing
- Blanking control with independent rising and falling event blanking times
- Auto-shutdown control with:
  - Independently selectable shutdown sources
  - Auto-restart enable
  - Auto-shutdown pin override control (high, low, off, and High-Z)

18.1 Fundamental Operation

18.1.1 STEERED PWM MODES

In steered PWM mode, the PWM signal derived from the input event sources is output as a single phase PWM which can be steered to any combination of the four COG outputs. Outputs are selected by setting the GxSTRA through GxSTRD bits of the COGxSTR register (Register 18-9). When the steering bits are cleared, then the output data is the static level determined by the GxSDATA through GxSDATD bits of the COGxSTR register. Output steering takes effect on the instruction cycle following the write to the COGxSTR register.

Synchronous steered PWM mode is identical to the steered PWM mode except that changes to the output steering take effect on the first rising event after the COGxSTR register write. Static output data is not synchronized.

Steering mode configurations are shown in Figure 18-2 and Figure 18-3.

Steered PWM and synchronous steered PWM modes are selected by setting the GxMD bits of the COGxCON0 register (Register 18-1) to '000' and '001' respectively.

18.1.2 FULL-BRIDGE MODES

In both Forward and Reverse Full-Bridge modes, two of the four COG outputs are active and the other two are inactive. Of the two active outputs, one is modulated by the PWM input signal and the other is on at 100% duty cycle. When the direction is changed, the dead-band time is inserted to delay the modulated output. This gives the unmodulated driver time to shut down, thereby, preventing shoot-through current in the series connected power devices.

In Forward Full-Bridge mode, the PWM input modulates the COGxD output and drives the COGA output at 100%.

In Reverse Full-Bridge mode, the PWM input modulates the COGxB output and drives the COGxC output at 100%.

The full-bridge configuration is shown in Figure 18-4. Typical full-bridge waveforms are shown in Figure 18-12 and Figure 18-13.

Full-Bridge Forward and Full-Bridge Reverse modes are selected by setting the GxMD bits of the COGxCON0 register to '010' and '011', respectively.

REGISTER 19-9: CLCxGLS2: GATE 3 LOGIC SELECT REGISTER

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
LCxG3D4T	LCxG3D4N	LCxG3D3T	LCxG3D3N	LCxG3D2T	LCxG3D2N	LCxG3D1T	LCxG3D1N
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 **LCxG3D4T:** Gate 3 Data 4 True (non-inverted) bit
1 = LCxD4T is gated into LCxG3
0 = LCxD4T is not gated into LCxG3

bit 6 **LCxG3D4N:** Gate 3 Data 4 Negated (inverted) bit
1 = LCxD4N is gated into LCxG3
0 = LCxD4N is not gated into LCxG3

bit 5 **LCxG3D3T:** Gate 3 Data 3 True (non-inverted) bit
1 = LCxD3T is gated into LCxG3
0 = LCxD3T is not gated into LCxG3

bit 4 **LCxG3D3N:** Gate 3 Data 3 Negated (inverted) bit
1 = LCxD3N is gated into LCxG3
0 = LCxD3N is not gated into LCxG3

bit 3 **LCxG3D2T:** Gate 3 Data 2 True (non-inverted) bit
1 = LCxD2T is gated into LCxG3
0 = LCxD2T is not gated into LCxG3

bit 2 **LCxG3D2N:** Gate 3 Data 2 Negated (inverted) bit
1 = LCxD2N is gated into LCxG3
0 = LCxD2N is not gated into LCxG3

bit 1 **LCxG3D1T:** Gate 3 Data 1 True (non-inverted) bit
1 = LCxD1T is gated into LCxG3
0 = LCxD1T is not gated into LCxG3

bit 0 **LCxG3D1N:** Gate 3 Data 1 Negated (inverted) bit
1 = LCxD1N is gated into LCxG3
0 = LCxD1N is not gated into LCxG3

REGISTER 19-10: CLCxGLS3: GATE 4 LOGIC SELECT REGISTER

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
LCxG4D4T	LCxG4D4N	LCxG4D3T	LCxG4D3N	LCxG4D2T	LCxG4D2N	LCxG4D1T	LCxG4D1N
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 **LCxG4D4T:** Gate 4 Data 4 True (non-inverted) bit
1 = LCxD4T is gated into LCxG4
0 = LCxD4T is not gated into LCxG4

bit 6 **LCxG4D4N:** Gate 4 Data 4 Negated (inverted) bit
1 = LCxD4N is gated into LCxG4
0 = LCxD4N is not gated into LCxG4

bit 5 **LCxG4D3T:** Gate 4 Data 3 True (non-inverted) bit
1 = LCxD3T is gated into LCxG4
0 = LCxD3T is not gated into LCxG4

bit 4 **LCxG4D3N:** Gate 4 Data 3 Negated (inverted) bit
1 = LCxD3N is gated into LCxG4
0 = LCxD3N is not gated into LCxG4

bit 3 **LCxG4D2T:** Gate 4 Data 2 True (non-inverted) bit
1 = LCxD2T is gated into LCxG4
0 = LCxD2T is not gated into LCxG4

bit 2 **LCxG4D2N:** Gate 4 Data 2 Negated (inverted) bit
1 = LCxD2N is gated into LCxG4
0 = LCxD2N is not gated into LCxG4

bit 1 **LCxG4D1T:** Gate 4 Data 1 True (non-inverted) bit
1 = LCxD1T is gated into LCxG4
0 = LCxD1T is not gated into LCxG4

bit 0 **LCxG4D1N:** Gate 4 Data 1 Negated (inverted) bit
1 = LCxD1N is gated into LCxG4
0 = LCxD1N is not gated into LCxG4

21.3 Register Definitions: ADC Control

REGISTER 21-1: ADCON0: ADC CONTROL REGISTER 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	R/W-0/0
—	CHS<4:0>					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 **Unimplemented:** Read as '0'

bit 6-2 **CHS<4:0>:** Analog Channel Select bits
- 11111 = FVR (Fixed Voltage Reference) Buffer 1 Output(2)
- 11110 = DAC1_output(1)
- 11101 = Temperature Indicator(3)
- 11100 = DAC2_output(5)
- 11011 = AN27(4)
- 11010 = AN26(4)
- 11001 = AN25(4)
- 11000 = AN24(4)
- 10111 = AN23(4)
- 10110 = AN22(4)
- 10101 = AN21(4)
- 10100 = AN20(4)
- 10011 = AN19
- 10010 = AN18
- 10001 = AN17
- 10000 = AN16
- 01111 = AN15
- 01110 = AN14
- 01101 = AN13
- 01100 = AN12
- 01011 = AN11
- 01010 = AN10
- 01001 = AN9
- 01000 = AN8
- 00111 = AN7(4)
- 00110 = AN6(4)
- 00101 = AN5(4)
- 00100 = AN4
- 00011 = AN3
- 00010 = AN2
- 00001 = AN1
- 00000 = AN0

bit 1 **GO/$\overline{\text{DONE}}$:** ADC Conversion Status bit
- 1 = ADC conversion cycle in progress. Setting this bit starts an ADC conversion cycle. This bit is automatically cleared by hardware when the ADC conversion has completed.
- 0 = ADC conversion completed/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 23.0 "8-Bit Digital-to-Analog Converter (DAC1) Module"** for more information.
2: See **Section 14.0 "Fixed Voltage Reference (FVR)"** for more information.
3: See **Section 15.0 "Temperature Indicator Module"** for more information.
4: PIC16(L)F1717/9 only.
5: See **Section 24.0 "5-Bit Digital-to-Analog Converter (DAC2) Module"** for more information.

REGISTER 21-5: ADRESL: ADC RESULT REGISTER LOW (ADRESL) ADFM = 0

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<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	

bit 7-6 **ADRES<1:0>**: ADC Result Register bits
Lower two bits of 10-bit conversion result

bit 5-0 **Reserved**: Do not use.

REGISTER 21-6: 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 21-7: 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

21.4 ADC Acquisition Requirements

For the ADC to meet its specified accuracy, the charge holding capacitor (CHOLD) must be allowed to fully charge to the input channel voltage level. The Analog Input model is shown in Figure 21-4. The source impedance (RS) and the internal sampling switch (RSS) impedance directly affect the time required to charge the capacitor CHOLD. The sampling switch (RSS) impedance varies over the device voltage (VDD), refer to Figure 21-4. **The maximum recommended impedance for analog sources is 10 kΩ.** As the source impedance is decreased, the acquisition time may be decreased. After the analog input channel is selected (or changed), an ADC acquisition must be done before the conversion can be started. To calculate the minimum acquisition time, Equation 21-1 may be used. This equation assumes that 1/2 LSb error is used (1,024 steps for the ADC). The 1/2 LSb error is the maximum error allowed for the ADC to meet its specified resolution.

EQUATION 21-1: ACQUISITION TIME EXAMPLE

Assumptions: $Temperature = 50°C$ *and external impedance of* $10k\Omega$ *5.0V* V_{DD}

$$T_{ACQ} = \text{Amplifier Settling Time} + \text{Hold Capacitor Charging Time} + \text{Temperature Coefficient}$$
$$= T_{AMP} + T_C + T_{COFF}$$
$$= 2\mu s + T_C + [(Temperature - 25°C)(0.05\mu s/°C)]$$

The value for T_C *can be approximated with the following equations:*

$$V_{APPLIED}\left(1 - \frac{1}{(2^{n+1}) - 1}\right) = V_{CHOLD}$$;[1] V_{CHOLD} *charged to within 1/2 lsb*

$$V_{APPLIED}\left(1 - e^{\frac{-TC}{RC}}\right) = V_{CHOLD}$$;[2] V_{CHOLD} *charge response to* $V_{APPLIED}$

$$V_{APPLIED}\left(1 - e^{\frac{-T_C}{RC}}\right) = V_{APPLIED}\left(1 - \frac{1}{(2^{n+1}) - 1}\right)$$;combining [1] and [2]

Note: Where n = number of bits of the ADC.

Solving for T_C*:*

$$T_C = -CHOLD(R_{IC} + R_{SS} + R_S)\ ln(1/2047)$$
$$= -10pF(1k\Omega + 7k\Omega + 10k\Omega)\ ln(0.0004885)$$
$$= 1.37\mu s$$

Therefore:

$$T_{ACQ} = 2\mu s + 892ns + [(50°C - 25°C)(0.05\mu s/°C)]$$
$$= 4.62\mu s$$

Note 1: The reference voltage (VREF) has no effect on the equation, since it cancels itself out.

2: The charge holding capacitor (CHOLD) is not discharged after each conversion.

3: The maximum recommended impedance for analog sources is 10 kΩ. This is required to meet the pin leakage specification.

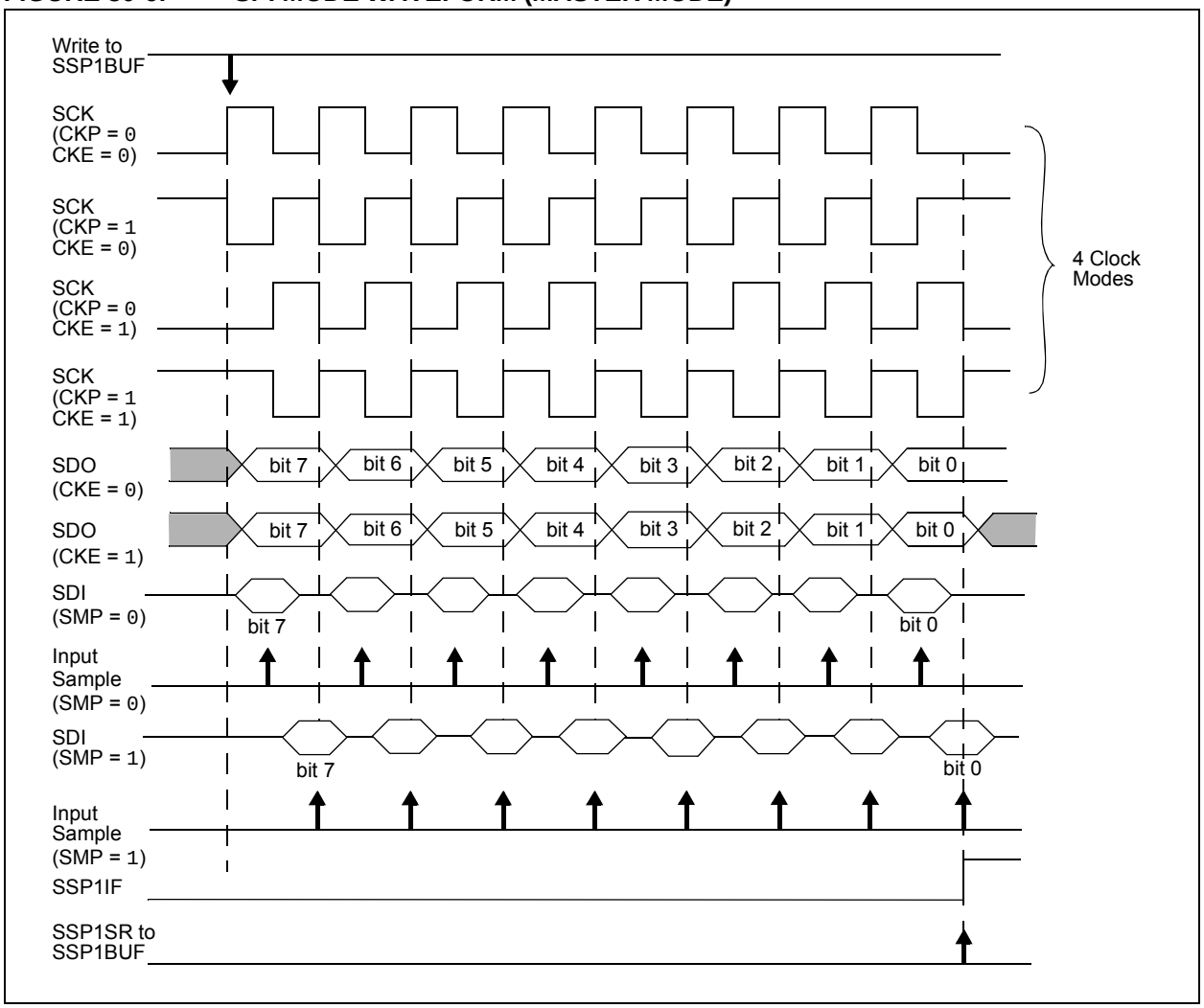
TABLE 28-1: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER2

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CCP2CON	—	—	DC2B<1:0>		CCP2M<3:0>				294
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	90
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	91
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	94
PR2	Timer2 Module Period Register								282*
T2CON	—	T2OUTPS<3:0>				TMR2ON	T2CKPS<1:0>		284
TMR2	Holding Register for the 8-bit TMR2 Register								282*

Legend: — = unimplemented location, read as '0'. Shaded cells are not used for Timer2 module.

* Page provides register information.

FIGURE 30-6: SPI MODE WAVEFORM (MASTER MODE)



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 SSP1IF 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 SSP1CON1 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 SSP1CON3 register will enable writes to the SSP1BUF register, even if the previous byte has not been read. This allows the software to ignore data that may not apply to it.

30.4.5 START CONDITION

The I^2C 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.

A bus collision can occur on a Start condition if the module samples the SDA line low before asserting it low. This does not conform to the I^2C Specification that states no bus collision can occur on a Start.

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/$\overline{W}$ bit set. The slave logic will then hold the clock and prepare to clock out data.

After a full match with R/$\overline{W}$ clear in 10-bit mode, a prior match flag is set and maintained until a Stop condition, a high address with R/$\overline{W}$ clear, or high address match fails.

30.4.8 START/STOP CONDITION INTERRUPT MASKING

The SCIE and PCIE bits of the SSP1CON3 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^2C START AND STOP CONDITIONS

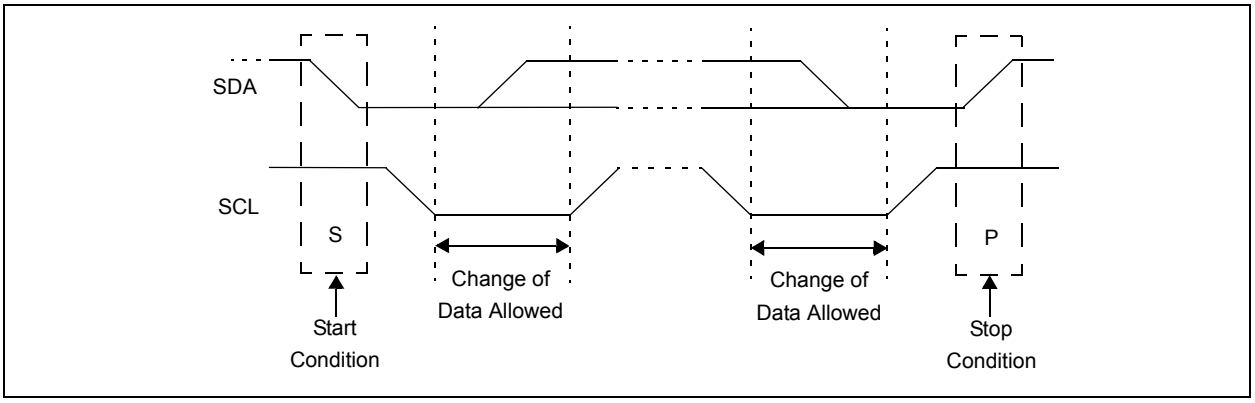
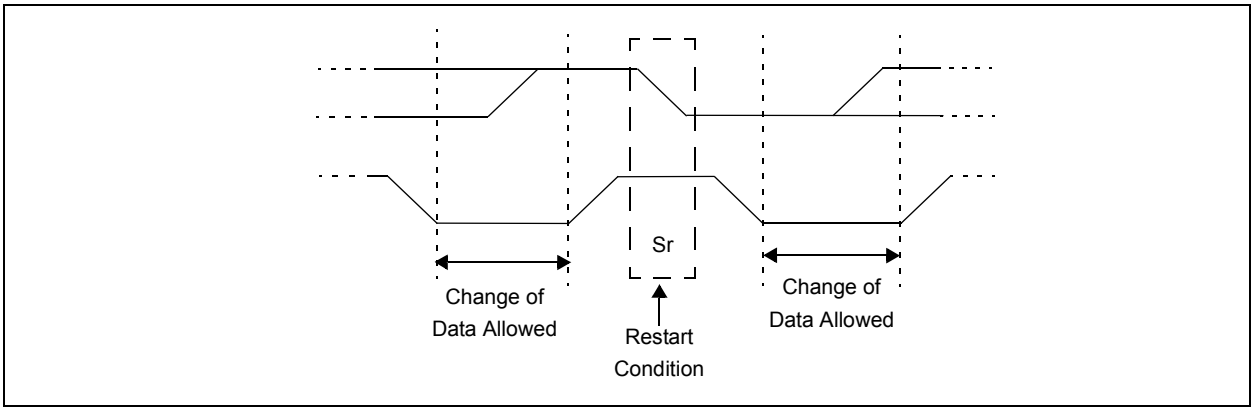


FIGURE 30-13: I^2C RESTART CONDITION



30.6 I²C Master Mode

Master mode is enabled by setting and clearing the appropriate SSPM bits in the SSP1CON1 register and by setting the SSPEN bit. In Master mode, the SDA and SCK pins must be configured as inputs. The MSSP peripheral hardware will override the output driver TRIS controls when necessary to drive the pins low.

Master mode of operation is supported by interrupt generation on the detection of the Start and Stop conditions. The Stop (P) and Start (S) bits are cleared from a Reset or when the MSSP module is disabled. Control of the I²C bus may be taken when the P bit is set, or the bus is Idle.

In Firmware Controlled Master mode, user code conducts all I²C bus operations based on Start and Stop bit condition detection. Start and Stop condition detection is the only active circuitry in this mode. All other communication is done by the user software directly manipulating the SDA and SCL lines.

The following events will cause the SSP Interrupt Flag bit, SSP1IF, to be set (SSP interrupt, if enabled):

- Start condition detected
- Stop condition detected
- Data transfer byte transmitted/received
- Acknowledge transmitted/received
- Repeated Start generated

Note 1: The MSSP module, when configured in I²C Master mode, does not allow queuing of events. For instance, the user is not allowed to initiate a Start condition and immediately write the SSP1BUF register to initiate transmission before the Start condition is complete. In this case, the SSP1BUF will not be written to and the WCOL bit will be set, indicating that a write to the SSP1BUF did not occur

2: Master mode suspends Start/Stop detection when sending the Start/Stop condition by means of the SEN/PEN control bits. The SSPxIF bit is set at the end of the Start/Stop generation when hardware clears the control bit.

30.6.1 I²C MASTER MODE OPERATION

The master device generates all of the serial clock pulses and the Start and Stop conditions. A transfer is ended with a Stop condition or with a Repeated Start condition. Since the Repeated Start condition is also the beginning of the next serial transfer, the I²C bus will not be released.

In Master Transmitter mode, serial data is output through SDA, while SCL outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the Read/Write (R/$\overline{W}$) bit. In this case, the R/$\overline{W}$ bit will be logic '0'. Serial data is transmitted eight bits at a time. After each byte is transmitted, an Acknowledge bit is received. Start and Stop conditions are output to indicate the beginning and the end of a serial transfer.

In Master Receive mode, the first byte transmitted contains the slave address of the transmitting device (7 bits) and the R/$\overline{W}$ bit. In this case, the R/$\overline{W}$ bit will be logic '1'. Thus, the first byte transmitted is a 7-bit slave address followed by a '1' to indicate the receive bit. Serial data is received via SDA, while SCL outputs the serial clock. Serial data is received eight bits at a time. After each byte is received, an Acknowledge bit is transmitted. Start and Stop conditions indicate the beginning and end of transmission.

A Baud Rate Generator is used to set the clock frequency output on SCL. See **Section 30.7 "Baud Rate Generator"** for more detail.

31.5.2 SYNCHRONOUS SLAVE MODE

The following bits are used to configure the EUSART for synchronous slave operation:

- SYNC = 1
- CSRC = 0
- SREN = 0 (for transmit); SREN = 1 (for receive)
- CREN = 0 (for transmit); CREN = 1 (for receive)
- SPEN = 1

Setting the SYNC bit of the TX1STA register configures the device for synchronous operation. Clearing the CSRC bit of the TX1STA register configures the device as a slave. Clearing the SREN and CREN bits of the RC1STA register ensures that the device is in the Transmit mode, otherwise the device will be configured to receive. Setting the SPEN bit of the RC1STA register enables the EUSART.

31.5.2.1 EUSART Synchronous Slave Transmit

The operation of the Synchronous Master and Slave modes are identical (see **Section 31.5.1.3 "Synchronous Master Transmission")**, except in the case of the Sleep mode.

If two words are written to the TX1REG and then the SLEEP instruction is executed, the following will occur:

1. The first character will immediately transfer to the TSR register and transmit.
2. The second word will remain in the TX1REG register.
3. The TXIF bit will not be set.
4. After the first character has been shifted out of TSR, the TX1REG register will transfer the second character to the TSR and the TXIF bit will now be set.
5. If the PEIE and TXIE bits are set, the interrupt will wake the device from Sleep and execute the next instruction. If the GIE bit is also set, the program will call the Interrupt Service Routine.

31.5.2.2 Synchronous Slave Transmission Setup

1. Set the SYNC and SPEN bits and clear the CSRC bit.
2. Clear the ANSEL bit for the CK pin (if applicable).
3. Clear the CREN and SREN bits.
4. If interrupts are desired, set the TXIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
5. If 9-bit transmission is desired, set the TX9 bit.
6. Enable transmission by setting the TXEN bit.
7. If 9-bit transmission is selected, insert the Most Significant bit into the TX9D bit.
8. Start transmission by writing the Least Significant eight bits to the TX1REG register.

33.0 INSTRUCTION SET SUMMARY

Each instruction is a 14-bit word containing the operation code (opcode) and all required operands. The opcodes are broken into three broad categories.

- Byte Oriented
- Bit Oriented
- Literal and Control

The literal and control category contains the most varied instruction word format.

Table 33-3 lists the instructions recognized by the MPASM™ assembler.

All instructions are executed within a single instruction cycle, with the following exceptions, which may take two or three cycles:

- Subroutine takes two cycles (CALL, CALLW)
- Returns from interrupts or subroutines take two cycles (RETURN, RETLW, RETFIE)
- Program branching takes two cycles (GOTO, BRA, BRW, BTFSS, BTFSC, DECFSZ, INCSFZ)
- One additional instruction cycle will be used when any instruction references an indirect file register and the file select register is pointing to program memory.

One instruction cycle consists of four oscillator cycles; for an oscillator frequency of 4 MHz, this gives a nominal instruction execution rate of 1 MHz.

All instruction examples use the format '0xhh' to represent a hexadecimal number, where 'h' signifies a hexadecimal digit.

33.1 Read-Modify-Write Operations

Any instruction that specifies a file register as part of the instruction performs a Read-Modify-Write (R-M-W) operation. The register is read, the data is modified, and the result is stored according to either the instruction, or the destination designator 'd'. A read operation is performed on a register even if the instruction writes to that register.

TABLE 33-1: OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1). The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1.
n	FSR or INDF number. (0-1)
mm	Pre-post increment-decrement mode selection

TABLE 33-2: ABBREVIATION DESCRIPTIONS

Field	Description
PC	Program Counter
$\overline{TO}$	Time-Out bit
C	Carry bit
DC	Digit Carry bit
Z	Zero bit
PD	Power-Down bit

CALL **Call Subroutine**

Syntax:	[*label*] CALL k
Operands:	0 ≤ k ≤ 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.

CALLW **Subroutine Call With W**

Syntax:	[*label*] 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.

CLRF **Clear f**

Syntax:	[*label*] CLRF f
Operands:	0 ≤ f ≤ 127
Operation:	00h → (f) 1 → Z
Status Affected:	Z
Description:	The contents of register ‘f’ are cleared and the Z bit is set.

CLRW **Clear W**

Syntax:	[*label*] CLRW
Operands:	None
Operation:	00h → (W) 1 → Z
Status Affected:	Z
Description:	W register is cleared. Zero bit (Z) is set.

CLRWDT **Clear Watchdog Timer**

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

COMF **Complement f**

Syntax:	[*label*] COMF f,d
Operands:	0 ≤ f ≤ 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’.

DECF **Decrement f**

Syntax:	[*label*] DECF f,d
Operands:	0 ≤ f ≤ 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’.

Note: Unless otherwise noted, V_{IN} = 5V, F_{OSC} = 500 kHz, C_{IN} = 0.1 µF, T_A = 25°C.

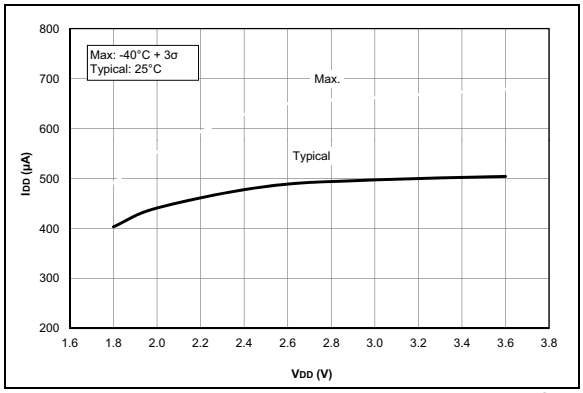


FIGURE 35-49: *IPD, Comparator, NP Mode (CxSP = 1), PIC16LF1717/8/9 Only.*

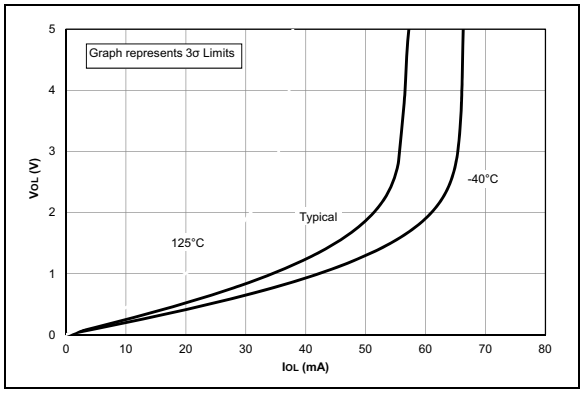


FIGURE 35-52: *VOL vs. IOL Over Temperature, VDD = 5.0V, PIC16F1717/8/9 Only.*

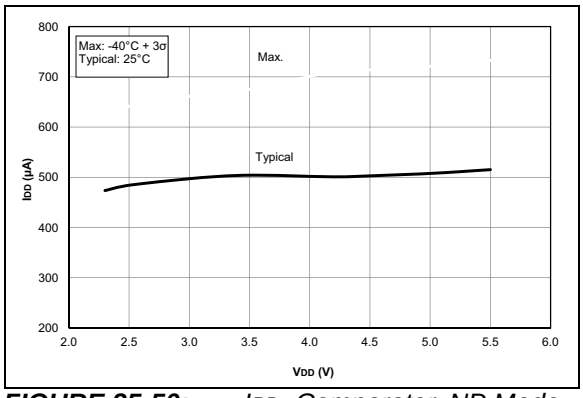


FIGURE 35-50: *IPD, Comparator, NP Mode (CxSP = 1), PIC16F1717/8/9 Only.*

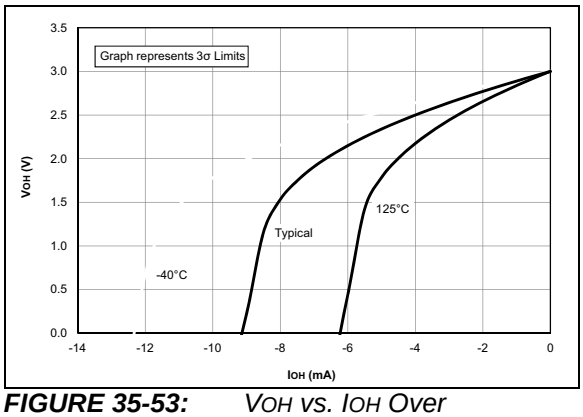


FIGURE 35-53: *VOH vs. IOH Over Temperature, VDD = 3.0V.*

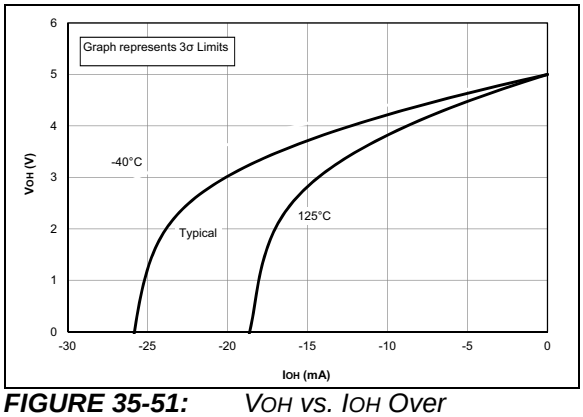


FIGURE 35-51: *VOH vs. IOH Over Temperature, VDD = 5.0V, PIC16F1717/8/9 Only.*

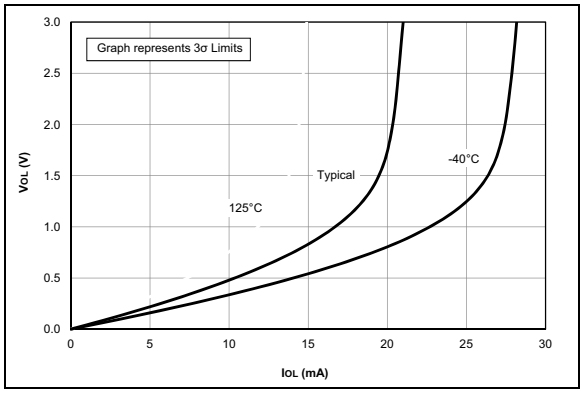


FIGURE 35-54: *VOL vs. IOL Over Temperature, VDD = 3.0V.*

Note: Unless otherwise noted, V_{IN} = 5V, F_{OSC} = 500 kHz, C_{IN} = 0.1 µF, T_A = 25°C.

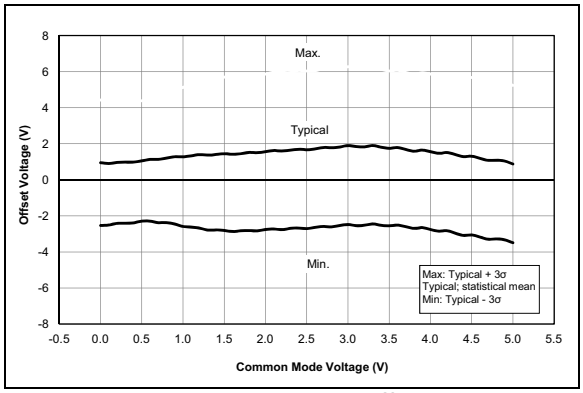


FIGURE 35-97: *Op Amp, Offset Over Common Mode Voltage, V_{DD} = 5.0V, Temp. = 25°C, PIC16F1717/8/9 Only.*

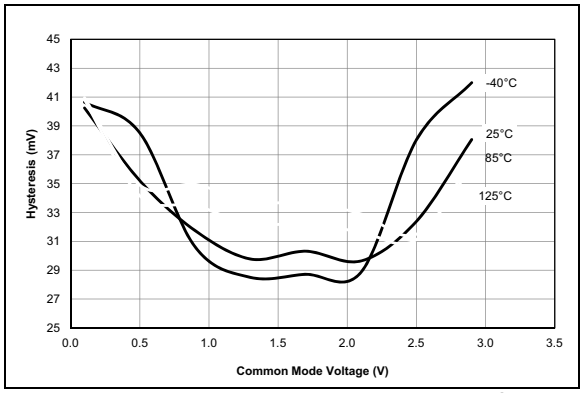


FIGURE 35-100: *Comparator Hysteresis, NP Mode (CxSP = 1), V_{DD} = 3.0V, Typical Measured Values.*

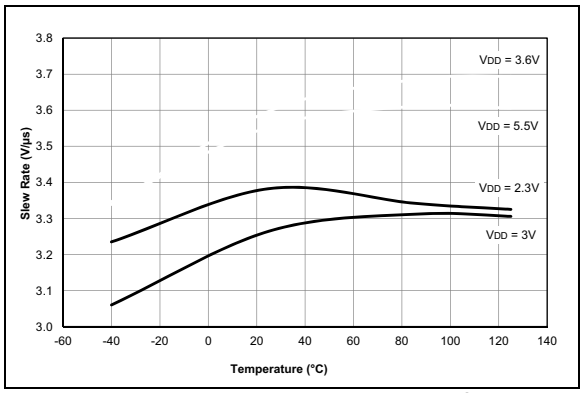


FIGURE 35-98: *Op Amp, Output Slew Rate, Rising Edge, PIC16F1717/8/9 Only.*

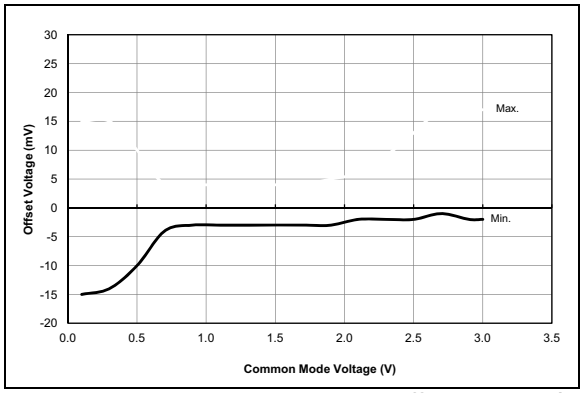


FIGURE 35-101: *Comparator Offset, NP Mode (CxSP = 1), V_{DD} = 3.0V, Typical Measured Values at 25°C.*

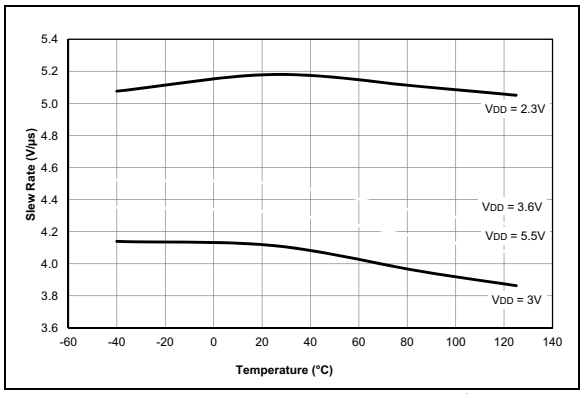


FIGURE 35-99: *Op Amp, Output Slew Rate, Falling Edge, PIC16F1717/8/9 Only.*

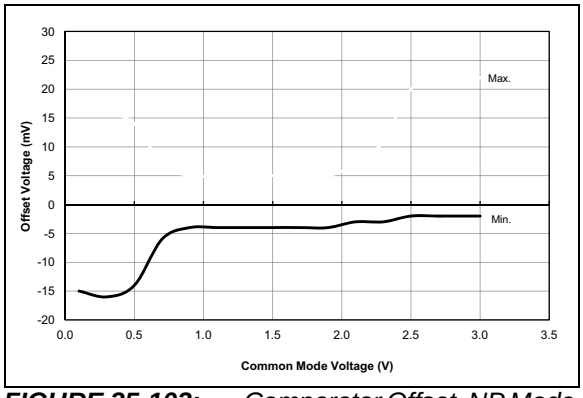
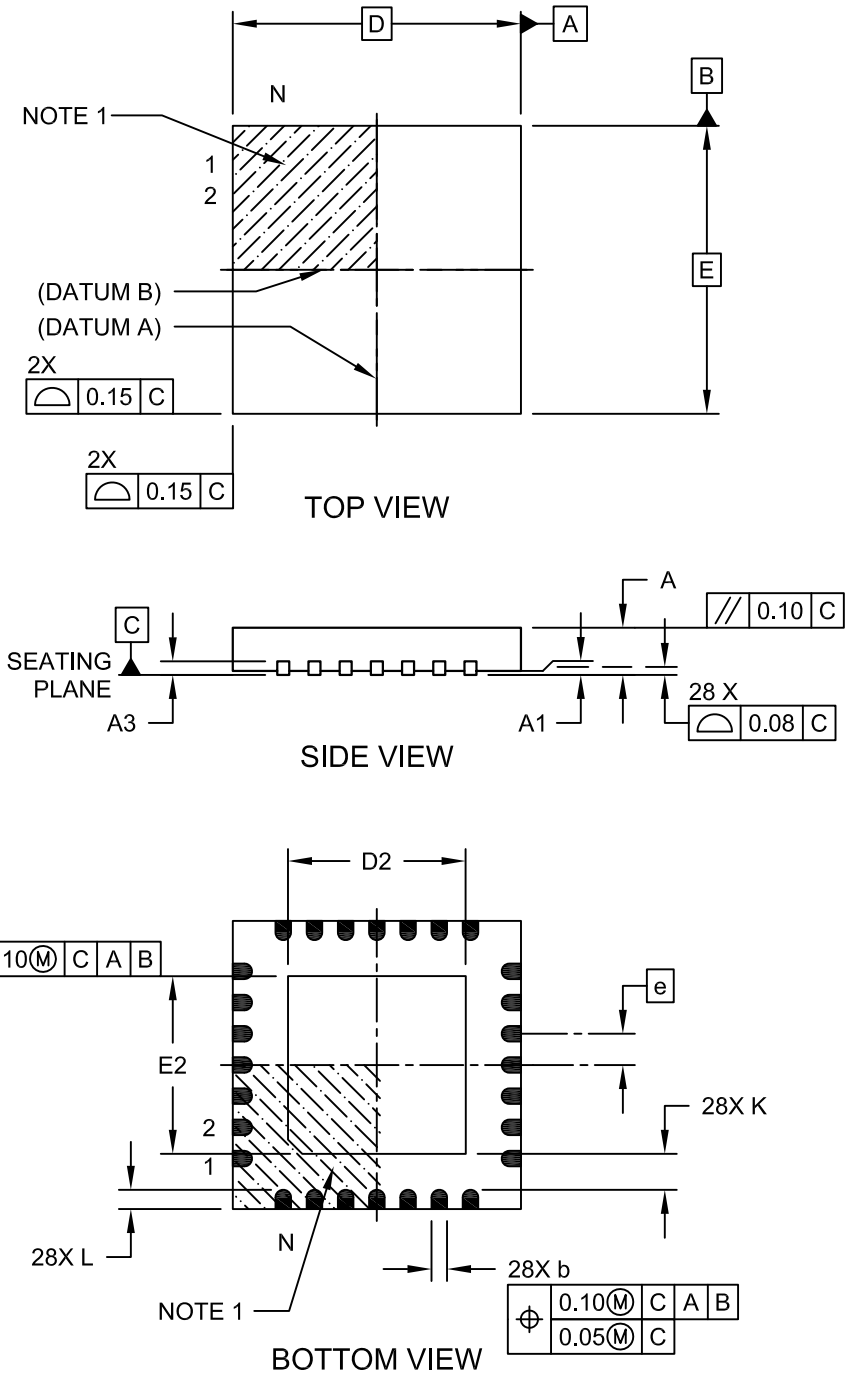


FIGURE 35-102: *Comparator Offset, NP Mode (CxSP = 1), V_{DD} = 3.0V, Typical Measured Values From -40°C to 125°C.*

28-Lead Plastic Quad Flat, No Lead Package (MM) - 6x6x0.9mm Body [QFN-S]
With 0.40 mm Terminal Length

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



Microchip Technology Drawing C04-124C Sheet 1 of 2