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, PSMC, PWM, WDT
Number of I/O	24
Program Memory Size	3.5KB (2K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 11x12b; D/A 1x8b
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/pic16lf1782-i-ss

TABLE 3-8: 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 7											
38Ch	INLVLA	Input Type Control for PORTA								0000 0000	0000 0000
38Dh	INLVLB	Input Type Control for PORTB								0000 0000	0000 0000
38Eh	INLVLC	Input Type Control for PORTC								1111 1111	1111 1111
38Fh	—	Unimplemented								—	—
390h	INLVLE	—	—	—	—	INLVLE3	—	—	—	---- 1---	---- 1---
391h	IOCAP	IOCAP<7:0>								0000 0000	0000 0000
392h	IOCAN	IOCAN<7:0>								0000 0000	0000 0000
393h	IOCAF	IOCAF<7:0>								0000 0000	0000 0000
394h	IOCBP	IOCBP<7:0>								0000 0000	0000 0000
395h	IOCBN	IOCBN<7:0>								0000 0000	0000 0000
396h	IOCBF	IOCBF<7:0>								0000 0000	0000 0000
397h	IOCCP	IOCCP<7:0>								0000 0000	0000 0000
398h	IOCCN	IOCCN<7:0>								0000 0000	0000 0000
399h	IOCCF	IOCCF<7:0>								0000 0000	0000 0000
39Ah — 39Ch	—	Unimplemented								—	—
39Dh	IOCEP	—	—	—	—	IOCEP3	—	—	—	---- 0---	---- 0---
39Eh	IOCEN	—	—	—	—	IOCEN3	—	—	—	---- 0---	---- 0---
39Fh	IOCEF	—	—	—	—	IOCEF3	—	—	—	---- 0---	---- 0---
Bank 8-9											
40Ch or 41Fh and 48Ch or 49Fh	—	Unimplemented								—	—
Bank 10											
50Ch — 510h	—	Unimplemented								—	—
511h	OPA1CON	OPA1EN	OPA1SP	—	—	—	—	OPA1PCH<1:0>		00-- --00	00-- --00
512h	—	Unimplemented								—	—
513h	OPA2CON	OPA2EN	OPA2SP	—	—	—	—	OPA2PCH<1:0>		00-- --00	00-- --00
514h — 519h	—	Unimplemented								—	—
51Ah	CLKRCON	CLKREN	CLKROE	CLKRSLR	CLKRDC<1:0>		CLKRDIV<2:0>			0011 0000	0011 0000
51Bh — 51Fh	—	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: These registers can be addressed from any bank.
2: Unimplemented, read as '1'.
3: PIC16F1782/3 only.

REGISTER 8-7: PIR4: PERIPHERAL INTERRUPT REQUEST REGISTER 4

U-0	U-0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
—	—	PSMC2TIF	PSMC1TIF	—	—	PSMC2SIF	PSMC1SIF
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 **Unimplemented:** Read as '0'

bit 5 **PSMC2TIF:** PSMC2 Time Base Interrupt Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

bit 4 **PSMC1TIF:** PSMC1 Time Base Interrupt Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

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

bit 1 **PSMC2SIF:** PSMC2 Auto-shutdown Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

bit 0 **PSMC1SIF:** PSMC1 Auto-shutdown Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

Note: Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the Global Enable bit, GIE, of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

TABLE 13-3: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELA	ANSA7	—	ANSA5	ANSA4	ANSA3	ANSA2	ANSA1	ANSA0	115
INLVLA	INLVLA7	INLVLA6	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0	116
LATA	LATA7	LATA6	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0	114
ODCONA	ODA7	ODA6	ODA5	ODA4	ODA3	ODA2	ODA1	ODA0	116
OPTION_REG	$\overline{\text{WPUEN}}$	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			174
PORTA	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	114
SLRCONA	SLRA7	SLRA6	SLRA5	SLRA4	SLRA3	SLRA2	SLRA1	SLRA0	116
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	114
WPUA	WPUA7	WPUA6	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0	115

Legend: x = unknown, u = unchanged, – = unimplemented locations read as '0'. Shaded cells are not used by PORTA.

TABLE 13-4: SUMMARY OF CONFIGURATION WORD WITH PORTA

Name	Bits	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	Register on Page
CONFIG1	13:8	—	—	FCMEN	IESO	$\overline{\text{CLKOUTEN}}$	BOREN<1:0>		$\overline{\text{CPD}}$	40
	7:0	$\overline{\text{CP}}$	MCLRE	$\overline{\text{PWRTE}}$	WDTE<1:0>		FOSC<2:0>			

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

REGISTER 17-4: ADRESH: ADC RESULT REGISTER HIGH (ADRESH) 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
AD<11:4>							
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 **AD<11:4>**: ADC Result Register bits
Upper 8 bits of 12-bit conversion result

REGISTER 17-5: 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
AD<3:0>				—	—	—	ADSIGN
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 **AD<3:0>**: ADC Result Register bits
Lower 4 bits of 12-bit conversion result

bit 3-1 **Extended LSb bits**: These are cleared to zero by DC conversion.

bit 0 **ADSIGN**: ADC Result Sign bit

17.4 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 17-4. 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 17-4. **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 17-1 may be used. This equation assumes that 1/2 LSb error is used (4,096 steps for the ADC). The 1/2 LSb error is the maximum error allowed for the ADC to meet its specified resolution.

EQUATION 17-1: ACQUISITION TIME EXAMPLE

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

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

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

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

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

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

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

Solving for T_C*:*

$$\begin{aligned} T_C &= -C_{HOLD}(R_{IC} + R_{SS} + R_S)\ ln(1/8191) \\ &= -10pF(1k\Omega + 7k\Omega + 10k\Omega)\ ln(0.000122) \\ &= 1.62\mu s \end{aligned}$$

Therefore:

$$\begin{aligned} T_{ACQ} &= 2\mu s + 1.62\mu s + [(50°C - 25°C)(0.05\mu s/°C)] \\ &= 4.87\mu s \end{aligned}$$

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

2: Maximum source impedance feeding the input pin should be considered so that the pin leakage does not cause a voltage divider, thereby limiting the absolute accuracy.

19.0 DIGITAL-TO-ANALOG CONVERTER (DAC) MODULE

The Digital-to-Analog Converter supplies a variable voltage reference, ratiometric with the input source, with 256 selectable output levels.

The input of the DAC can be connected to:

- External VREF pins
- VDD supply voltage
- FVR (Fixed Voltage Reference)

The output of the DAC can be configured to supply a reference voltage to the following:

- Comparator positive input
- Op amp positive input
- ADC input channel
- DACOUT1 pin
- DACOUT2 pin

The Digital-to-Analog Converter (DAC) is enabled by setting the DACEN bit of the DACCON0 register.

19.1 Output Voltage Selection

The DAC has 256 voltage level ranges. The 256 levels are set with the DACR<7:0> bits of the DACCON1 register.

The DAC output voltage is determined by Equation 19-1:

EQUATION 19-1: DAC OUTPUT VOLTAGE

$$\underline{IF\ DACxEN = 1}$$

$$V_{OUT} = \left((V_{SOURCE+} - V_{SOURCE-}) \times \frac{DACxR[7:0]}{2^8} \right) + V_{SOURCE-}$$

$$V_{SOURCE+} = V_{DD},\ V_{REF},\ or\ FVR\ BUFFER\ 2$$

$$V_{SOURCE-} = V_{SS}$$

19.2 Ratiometric Output Level

The DAC output value is derived using a resistor ladder with each end of the ladder tied to a positive and negative voltage reference input source. If the voltage of either input source fluctuates, a similar fluctuation will result in the DAC output value.

The value of the individual resistors within the ladder can be found in **Section 30.0 "Electrical Specifications"**.

19.3 DAC Voltage Reference Output

The DAC voltage can be output to the DACOUT1 and DACOUT2 pins by setting the respective DACOE1 and DACOE2 pins of the DACCON0 register. Selecting the DAC reference voltage for output on either DACOUTx pin automatically overrides the digital output buffer and digital input threshold detector functions of that pin. Reading the DACOUTx pin when it has been configured for DAC reference voltage output will always return a '0'.

Due to the limited current drive capability, a buffer must be used on the DAC voltage reference output for external connections to either DACOUTx pin. Figure 19-2 shows an example buffering technique.

TABLE 22-5: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER1

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	121
CCP1CON	—	—	DC1B<1:0>		CCP1M<3:0>				255
CCP2CON	—	—	DC2B<1:0>		CCP2M<3:0>				255
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	79
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	80
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	83
TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								175*
TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								175*
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	120
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	125
T1CON	TMR1CS<1:0>		T1CKPS<1:0>		T1OSCEN	$\overline{\text{T1SYNC}}$	—	TMR1ON	183
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/$\overline{\text{DONE}}$	T1GVAL	T1GSS<1:0>		184

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by the Timer1 module.

* Page provides register information.

TABLE 23-1: 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>				255
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	79
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	80
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	83
PR2	Timer2 Module Period Register								186*
T2CON	—	T2OUTPS<3:0>				TMR2ON	T2CKPS<1:0>		188
TMR2	Holding Register for the 8-bit TMR2 Register								186*

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

* Page provides register information.

25.4 Register Definitions: CCP Control

REGISTER 25-1: CCPxCON: CCPx CONTROL 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
—	—	DCxB<1:0>		CCPxM<3:0>			
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 Reset
'1' = Bit is set	'0' = Bit is cleared	

bit 7-6 **Unimplemented:** Read as '0'

bit 5-4 **DCxB<1:0>:** PWM Duty Cycle Least Significant bits

Capture mode:
Unused

Compare mode:
Unused

PWM mode:
These bits are the two LSbs of the PWM duty cycle. The eight MSbs are found in CCPRxL.

bit 3-0 **CCPxM<3:0>:** CCPx Mode Select bits

11xx = PWM mode

1011 = Compare mode: Auto-conversion Trigger (sets CCPxIF bit (CCP2), starts ADC conversion if ADC module is enabled)**(1)**
1010 = Compare mode: generate software interrupt only
1001 = Compare mode: clear output on compare match (set CCPxIF)
1000 = Compare mode: set output on compare match (set CCPxIF)

0111 = Capture mode: every 16th rising edge
0110 = Capture mode: every 4th rising edge
0101 = Capture mode: every rising edge
0100 = Capture mode: every falling edge

0011 = Reserved
0010 = Compare mode: toggle output on match
0001 = Reserved
0000 = Capture/Compare/PWM off (resets CCPx module)

26.4.9 ACKNOWLEDGE SEQUENCE

The 9th SCL pulse for any transferred byte in I^2C is dedicated as an Acknowledge. It allows receiving devices to respond back to the transmitter by pulling the SDA line low. The transmitter must release control of the line during this time to shift in the response. The Acknowledge ($\overline{ACK}$) is an active-low signal, pulling the SDA line low indicated to the transmitter that the device has received the transmitted data and is ready to receive more.

The result of an $\overline{ACK}$ is placed in the ACKSTAT bit of the SSPCON2 register.

Slave software, when the AHEN and DHEN bits are set, allow the user to set the $\overline{ACK}$ value sent back to the transmitter. The ACKDT bit of the SSPCON2 register is set/cleared to determine the response.

Slave hardware will generate an $\overline{ACK}$ response if the AHEN and DHEN bits of the SSPCON3 register are clear.

There are certain conditions where an $\overline{ACK}$ will not be sent by the slave. If the BF bit of the SSPSTAT register or the SSPOV bit of the SSPCON1 register are set when a byte is received.

When the module is addressed, after the 8th falling edge of SCL on the bus, the ACKTIM bit of the SSPCON3 register is set. The ACKTIM bit indicates the acknowledge time of the active bus. The ACKTIM Status bit is only active when the AHEN bit or DHEN bit is enabled.

26.5 I^2C SLAVE MODE OPERATION

The MSSP Slave mode operates in one of four modes selected in the SSPM bits of SSPCON1 register. The modes can be divided into 7-bit and 10-bit Addressing mode. 10-bit Addressing modes operate the same as 7-bit with some additional overhead for handling the larger addresses.

Modes with Start and Stop bit interrupts operated the same as the other modes with SSP1IF additionally getting set upon detection of a Start, Restart, or Stop condition.

26.5.1 SLAVE MODE ADDRESSES

The SSPADD register (Register 26-6) contains the Slave mode address. The first byte received after a Start or Restart condition is compared against the value stored in this register. If the byte matches, the value is loaded into the SSPBUF register and an interrupt is generated. If the value does not match, the module goes idle and no indication is given to the software that anything happened.

The SSP Mask register (Register 26-5) affects the address matching process. See **Section 26.5.9 "SSP Mask Register"** for more information.

26.5.1.1 I^2C Slave 7-bit Addressing Mode

In 7-bit Addressing mode, the LSb of the received data byte is ignored when determining if there is an address match.

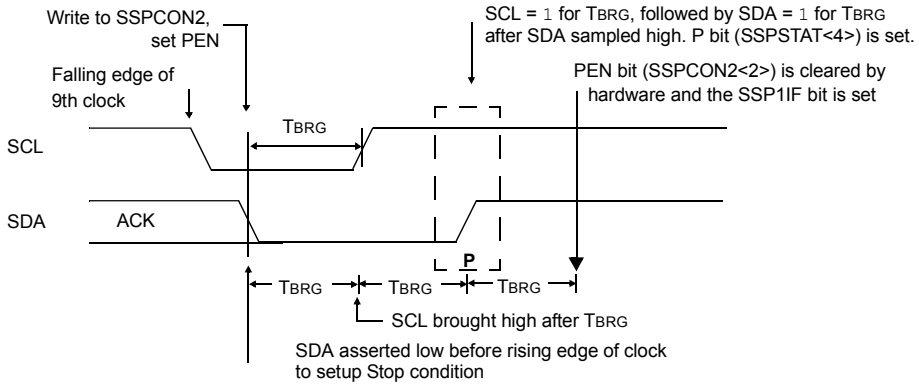
26.5.1.2 I^2C Slave 10-bit Addressing Mode

In 10-bit Addressing mode, the first received byte is compared to the binary value of '1 1 1 1 0 A9 A8 0'. A9 and A8 are the two MSb of the 10-bit address and stored in bits 2 and 1 of the SSPADD register.

After the acknowledge of the high byte the UA bit is set and SCL is held low until the user updates SSPADD with the low address. The low address byte is clocked in and all 8 bits are compared to the low address value in SSPADD. Even if there is not an address match; SSP1IF and UA are set, and SCL is held low until SSPADD is updated to receive a high byte again. When SSPADD is updated the UA bit is cleared. This ensures the module is ready to receive the high address byte on the next communication.

A high and low address match as a write request is required at the start of all 10-bit addressing communication. A transmission can be initiated by issuing a Restart once the slave is addressed, and clocking in the high address with the R/$\overline{W}$ bit set. The slave hardware will then acknowledge the read request and prepare to clock out data. This is only valid for a slave after it has received a complete high and low address byte match.

FIGURE 26-31: STOP CONDITION RECEIVE OR TRANSMIT MODE



Note: TBRG = one Baud Rate Generator period.

26.6.10 SLEEP OPERATION

While in Sleep mode, the I^2C slave module can receive addresses or data and when an address match or complete byte transfer occurs, wake the processor from Sleep (if the MSSP interrupt is enabled).

26.6.11 EFFECTS OF A RESET

A Reset disables the MSSP module and terminates the current transfer.

26.6.12 MULTI-MASTER MODE

In Multi-Master mode, the interrupt generation on the detection of the Start and Stop conditions allows the determination of when the bus is free. The Stop (P) and Start (S) bits are cleared from a Reset or when the MSSP module is disabled. Control of the I^2C bus may be taken when the P bit of the SSPSTAT register is set, or the bus is Idle, with both the S and P bits clear. When the bus is busy, enabling the SSP interrupt will generate the interrupt when the Stop condition occurs.

In multi-master operation, the SDA line must be monitored for arbitration to see if the signal level is the expected output level. This check is performed by hardware with the result placed in the BCL1IF bit.

The states where arbitration can be lost are:

- Address Transfer
- Data Transfer
- A Start Condition
- A Repeated Start Condition
- An Acknowledge Condition

26.6.13 MULTI -MASTER COMMUNICATION, BUS COLLISION AND BUS ARBITRATION

Multi-Master mode support is achieved by bus arbitration. When the master outputs address/data bits onto the SDA pin, arbitration takes place when the master outputs a '1' on SDA, by letting SDA float high and another master asserts a '0'. When the SCL pin floats high, data should be stable. If the expected data on SDA is a '1' and the data sampled on the SDA pin is '0', then a bus collision has taken place. The master will set the Bus Collision Interrupt Flag, BCL1IF and reset the I^2C port to its Idle state (Figure 26-31).

If a transmit was in progress when the bus collision occurred, the transmission is halted, the BF flag is cleared, the SDA and SCL lines are deasserted and the SSPBUF can be written to. When the user services the bus collision Interrupt Service Routine and if the I^2C bus is free, the user can resume communication by asserting a Start condition.

If a Start, Repeated Start, Stop or Acknowledge condition was in progress when the bus collision occurred, the condition is aborted, the SDA and SCL lines are deasserted and the respective control bits in the SSPCON2 register are cleared. When the user services the bus collision Interrupt Service Routine and if the I^2C bus is free, the user can resume communication by asserting a Start condition.

The master will continue to monitor the SDA and SCL pins. If a Stop condition occurs, the SSP1IF bit will be set.

A write to the SSPBUF will start the transmission of data at the first data bit, regardless of where the transmitter left off when the bus collision occurred.

In Multi-Master mode, the interrupt generation on the detection of Start and Stop conditions allows the determination of when the bus is free. Control of the I^2C bus can be taken when the P bit is set in the SSPSTAT register, or the bus is Idle and the S and P bits are cleared.

26.6.13.2 Bus Collision During a Repeated Start Condition

During a Repeated Start condition, a bus collision occurs if:

a) A low level is sampled on SDA when SCL goes from low level to high level.

b) SCL goes low before SDA is asserted low, indicating that another master is attempting to transmit a data '1'.

When the user releases SDA and the pin is allowed to float high, the BRG is loaded with SSPADD and counts down to zero. The SCL pin is then deasserted and when sampled high, the SDA pin is sampled.

If SDA is low, a bus collision has occurred (i.e., another master is attempting to transmit a data '0', Figure 26-35). If SDA is sampled high, the BRG is reloaded and begins counting. If SDA goes from high-to-low before the BRG times out, no bus collision occurs because no two masters can assert SDA at exactly the same time.

If SCL goes from high-to-low before the BRG times out and SDA has not already been asserted, a bus collision occurs. In this case, another master is attempting to transmit a data '1' during the Repeated Start condition, see Figure 26-36.

If, at the end of the BRG time-out, both SCL and SDA are still high, the SDA pin is driven low and the BRG is reloaded and begins counting. At the end of the count, regardless of the status of the SCL pin, the SCL pin is driven low and the Repeated Start condition is complete.

FIGURE 26-36: BUS COLLISION DURING A REPEATED START CONDITION (CASE 1)

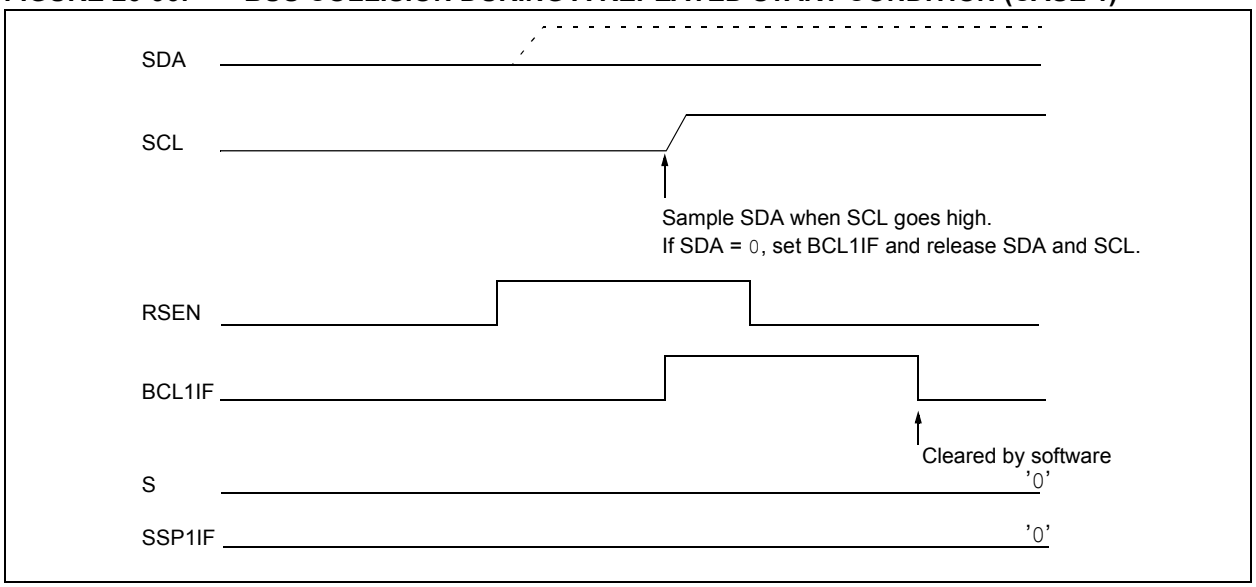


FIGURE 26-37: BUS COLLISION DURING REPEATED START CONDITION (CASE 2)

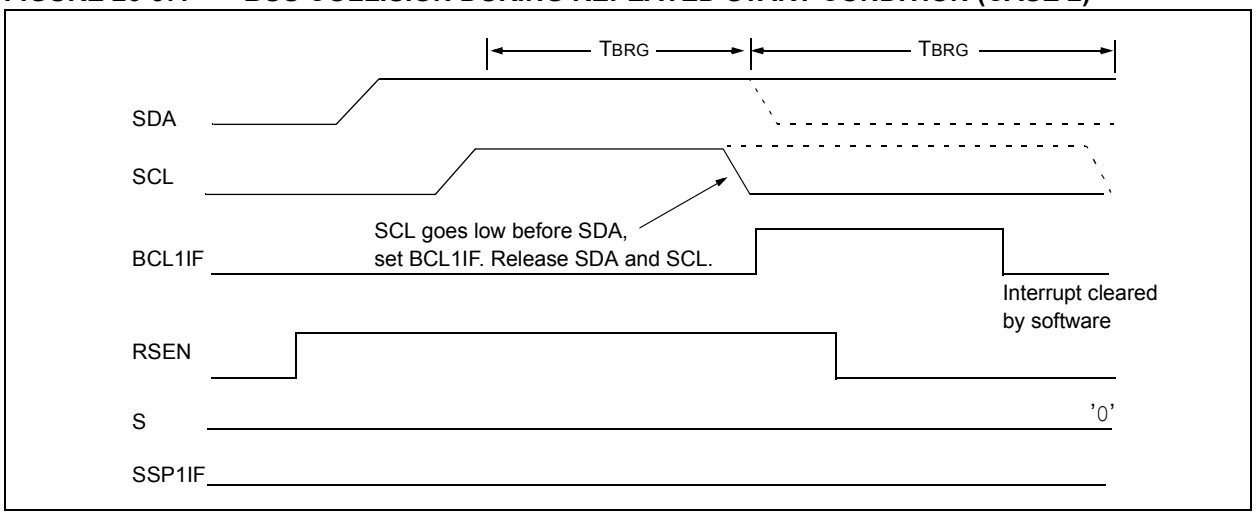
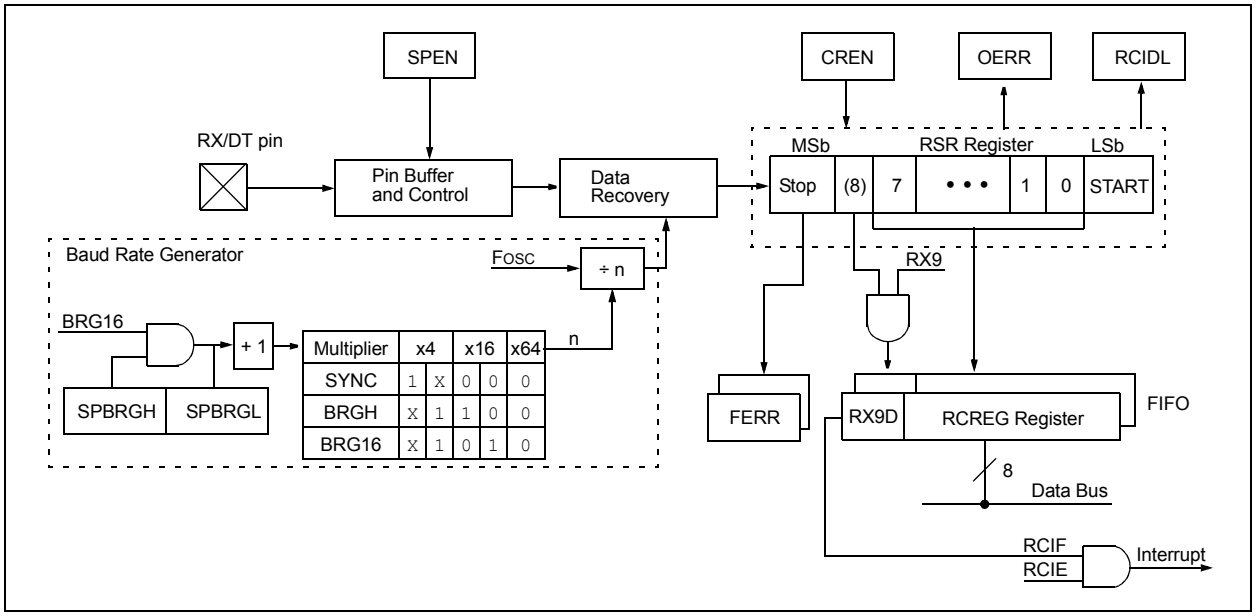


FIGURE 27-2: EUSART RECEIVE BLOCK DIAGRAM



The operation of the EUSART module is controlled through three registers:

- Transmit Status and Control (TXSTA)
- Receive Status and Control (RCSTA)
- Baud Rate Control (BAUDCON)

These registers are detailed in Register 27-1, Register 27-2 and Register 27-3, respectively.

When the receiver or transmitter section is not enabled then the corresponding RX or TX pin may be used for general purpose input and output.

RRF **Rotate Right f through Carry**

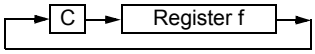
Syntax: [*label*] RRF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: See description below

Status Affected: C

Description: The contents of register 'f' are rotated one bit to the right through the Carry flag. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.



SLEEP **Enter Sleep mode**

Syntax: [*label*] SLEEP

Operands: None

Operation: 00h → WDT,
0 → WDT prescaler,
1 → $\overline{TO}$,
0 → $\overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$

Description: The power-down Status bit, $\overline{PD}$ is cleared. Time-out Status bit, $\overline{TO}$ is set. Watchdog Timer and its prescaler are cleared.
The processor is put into Sleep mode with the oscillator stopped.

SUBLW **Subtract W from literal**

Syntax: [*label*] SUBLW k

Operands: $0 \le k \le 255$

Operation: k - (W) → (W)

Status Affected: C, DC, Z

Description: The W register is subtracted (2's complement method) from the 8-bit literal 'k'. The result is placed in the W register.

C = 0	W > k
C = 1	W ≤ k
DC = 0	W<3:0> > k<3:0>
DC = 1	W<3:0> ≤ k<3:0>

SUBWF **Subtract W from f**

Syntax: [*label*] SUBWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f) - (W) → (destination)

Status Affected: C, DC, Z

Description: Subtract (2's complement method) W register from 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.

C = 0	W > f
C = 1	W ≤ f
DC = 0	W<3:0> > f<3:0>
DC = 1	W<3:0> ≤ f<3:0>

SUBWFB **Subtract W from f with Borrow**

Syntax: SUBWFB f {,d}

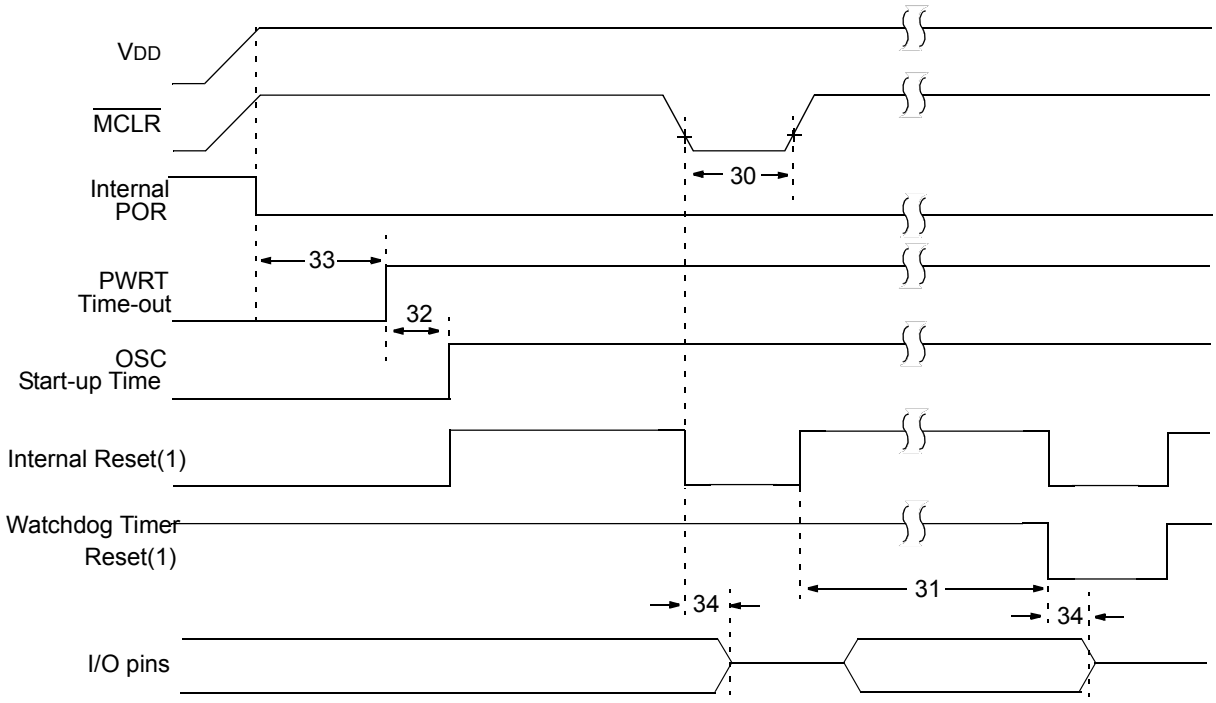
Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f) – (W) – ($\overline{B}$) → dest

Status Affected: C, DC, Z

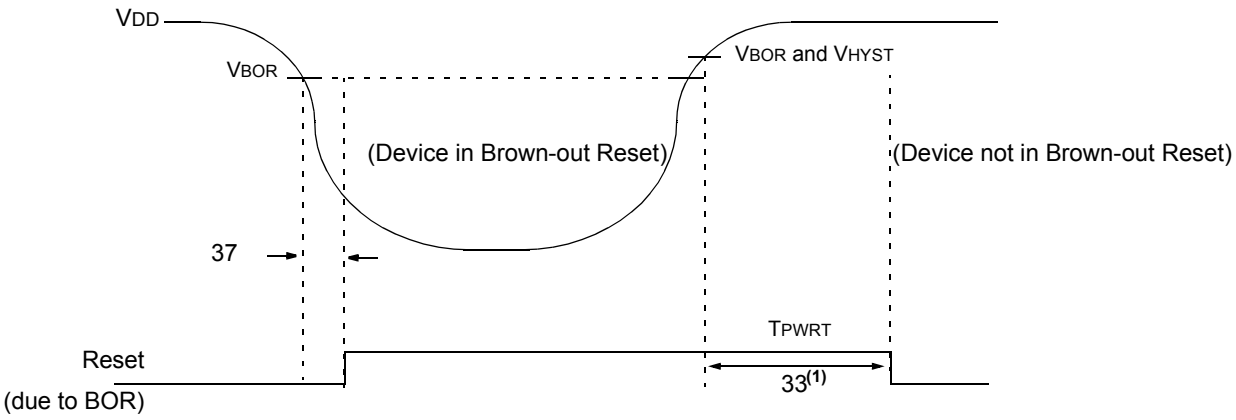
Description: Subtract W and the BORROW flag (CARRY) from register 'f' (2's complement method). If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f'.

FIGURE 30-8: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING



Note 1: Asserted low.

FIGURE 30-9: BROWN-OUT RESET TIMING AND CHARACTERISTICS



Note 1: The delay, (TPWRT) releasing Reset, only occurs when the Power-up Timer is enabled, ($\overline{PWRTE}$ = 0).

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

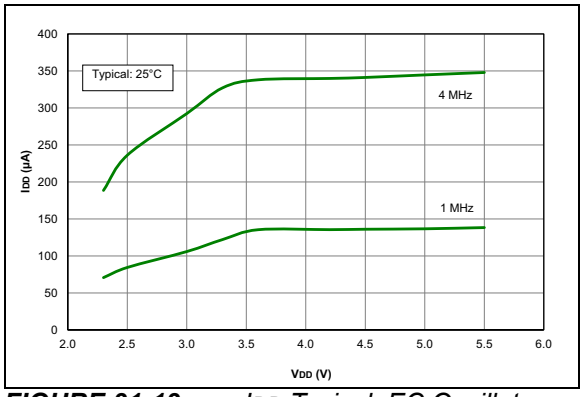


FIGURE 31-13: *IDD Typical, EC Oscillator MP Mode, PIC16F1782/3 Only.*

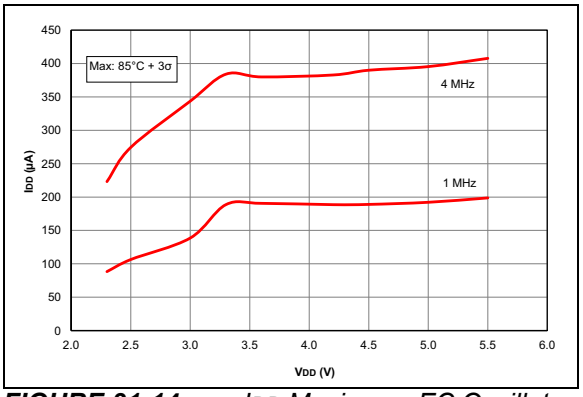


FIGURE 31-14: *IDD Maximum, EC Oscillator MP Mode, PIC16F1782/3 Only.*

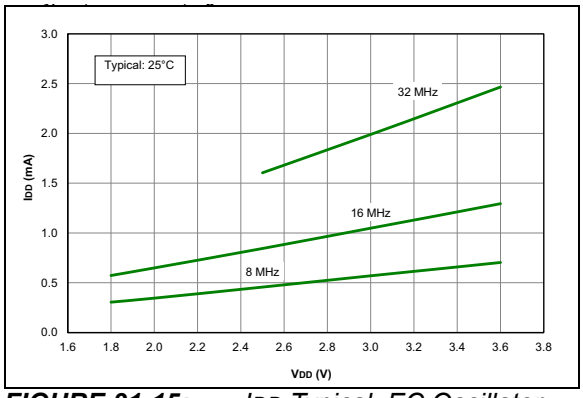


FIGURE 31-15: *IDD Typical, EC Oscillator HP Mode, PIC16LF1782/3 Only.*

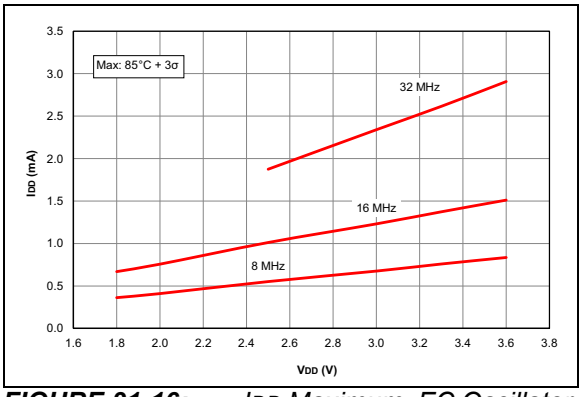


FIGURE 31-16: *IDD Maximum, EC Oscillator HP Mode, PIC16LF1782/3 Only.*

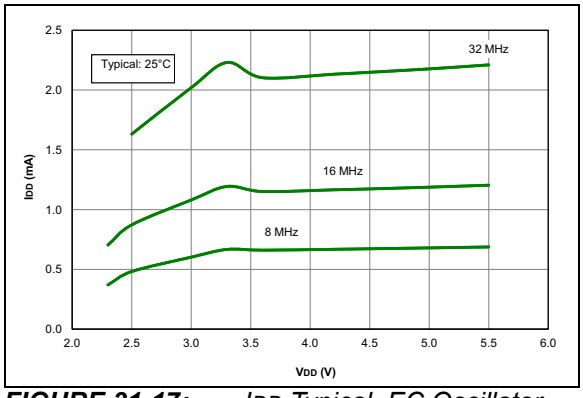


FIGURE 31-17: *IDD Typical, EC Oscillator HP Mode, PIC16F1782/3 Only.*

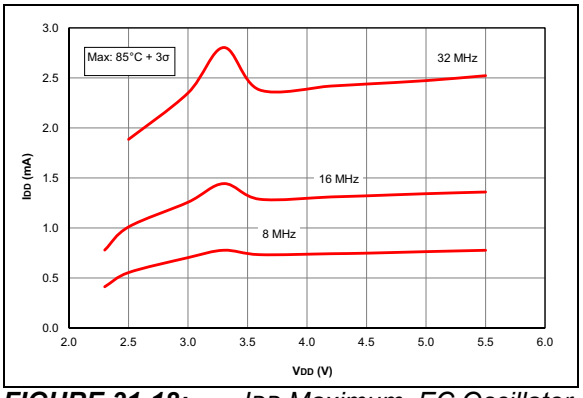


FIGURE 31-18: *IDD Maximum, EC Oscillator HP Mode, PIC16F1782/3 Only.*

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

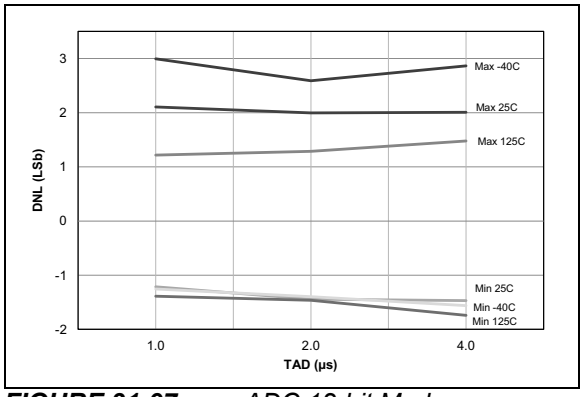


FIGURE 31-97: *ADC 12-bit Mode, Single-Ended DNL, VDD = 5.5V, VREF = 5.5V.*

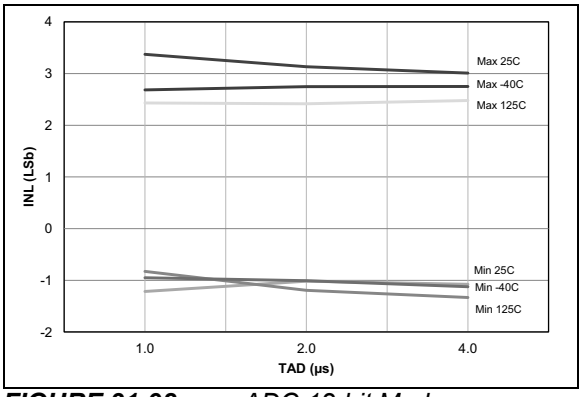


FIGURE 31-98: *ADC 12-bit Mode, Single-Ended INL, VDD = 5.5V, VREF = 5.5V.*

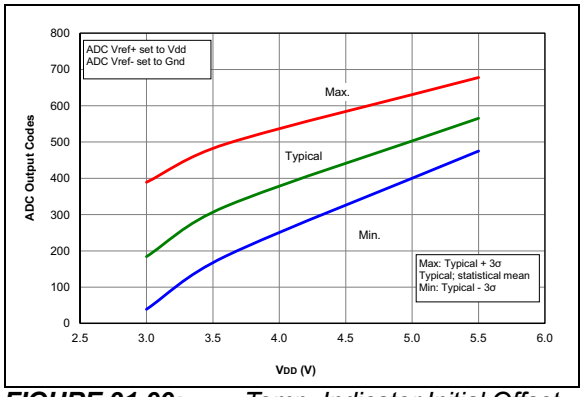


FIGURE 31-99: *Temp. Indicator Initial Offset, High Range, Temp. = 20°C, PIC16F1782/3 Only.*

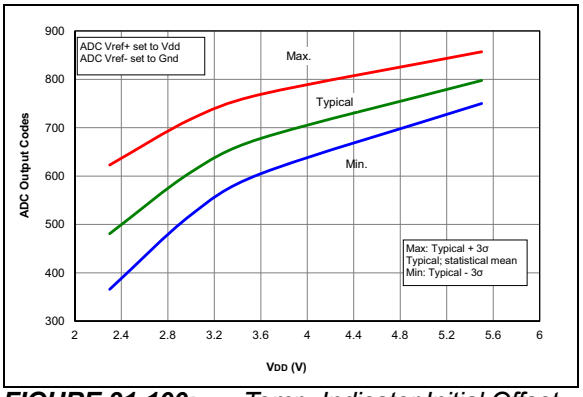


FIGURE 31-100: *Temp. Indicator Initial Offset, Low Range, Temp. = 20°C, PIC16F1782/3 Only.*

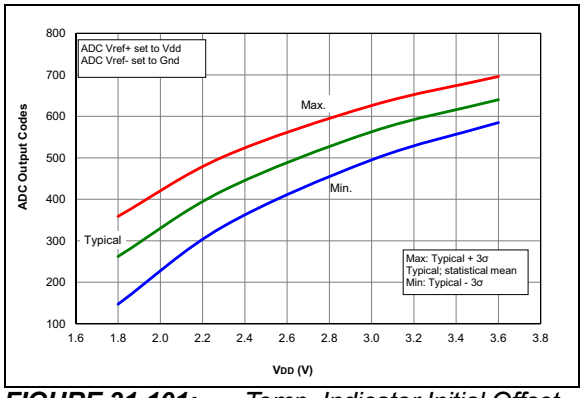


FIGURE 31-101: *Temp. Indicator Initial Offset, Low Range, Temp. = 20°C, PIC16LF1782/3 Only.*

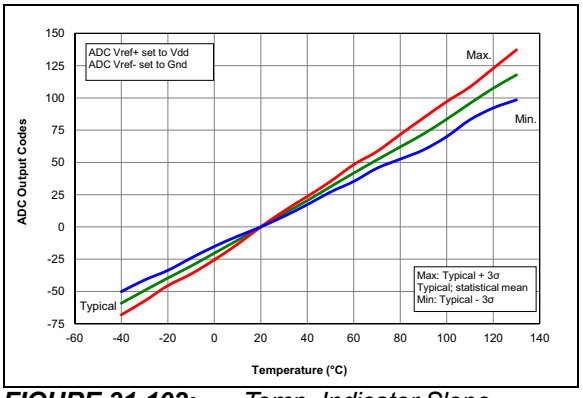


FIGURE 31-102: *Temp. Indicator Slope Normalized to 20°C, High Range, VDD = 5.5V, PIC16F1782/3 Only.*

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

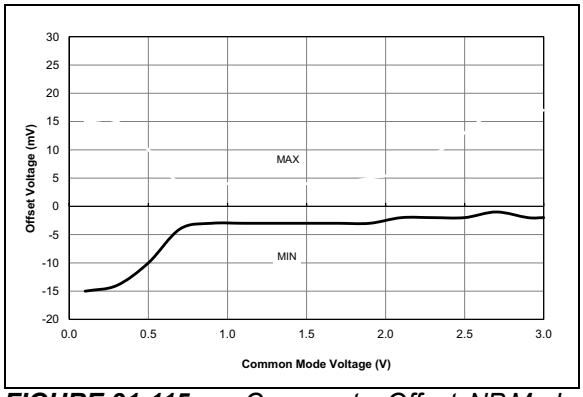


FIGURE 31-115: *Comparator Offset, NP Mode (CxSP = 1), VDD = 3.0V, Typical Measured Values at 25°C.*

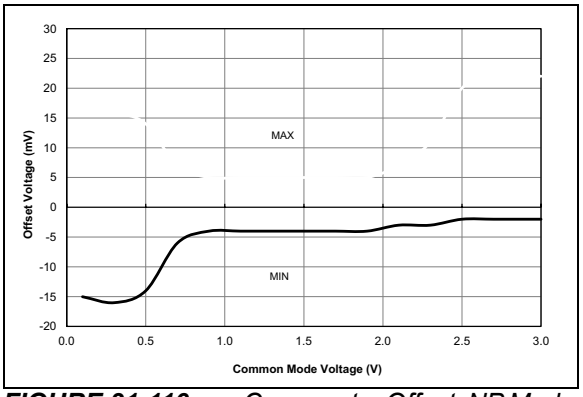


FIGURE 31-116: *Comparator Offset, NP Mode (CxSP = 1), VDD = 3.0V, Typical Measured Values From -40°C to 125°C.*

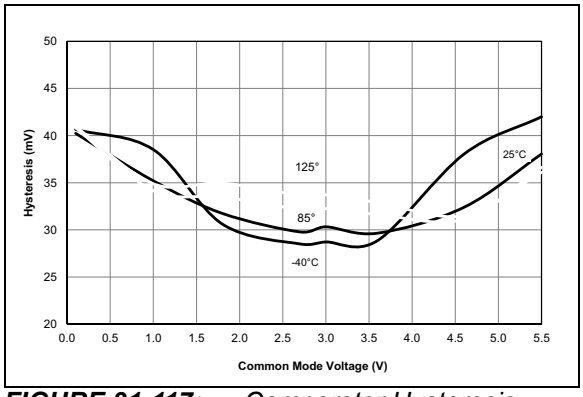


FIGURE 31-117: *Comparator Hysteresis, NP Mode (CxSP = 1), VDD = 5.5V, Typical Measured Values, PIC16F1782/3 Only.*

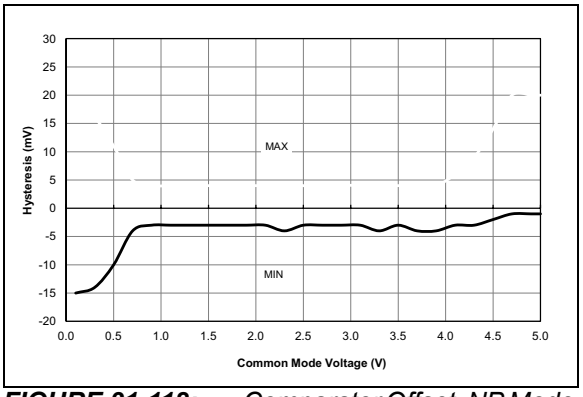


FIGURE 31-118: *Comparator Offset, NP Mode (CxSP = 1), VDD = 5.0V, Typical Measured Values at 25°C, PIC16F1782/3 Only.*

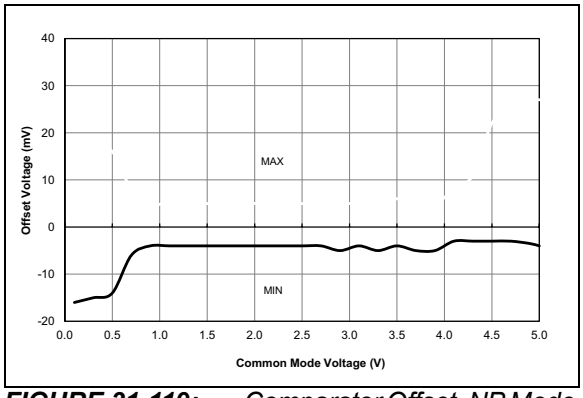


FIGURE 31-119: *Comparator Offset, NP Mode (CxSP = 1), VDD = 5.0V, Typical Measured Values From -40°C to 125°C, PIC16F1782/3 Only.*

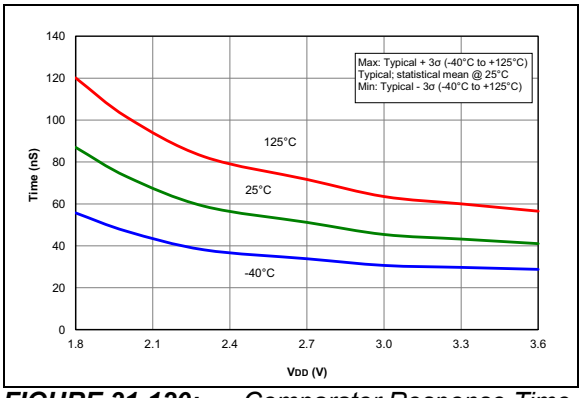


FIGURE 31-120: *Comparator Response Time Over Voltage, NP Mode (CxSP = 1), Typical Measured Values, PIC16LF1782/3 Only.*

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

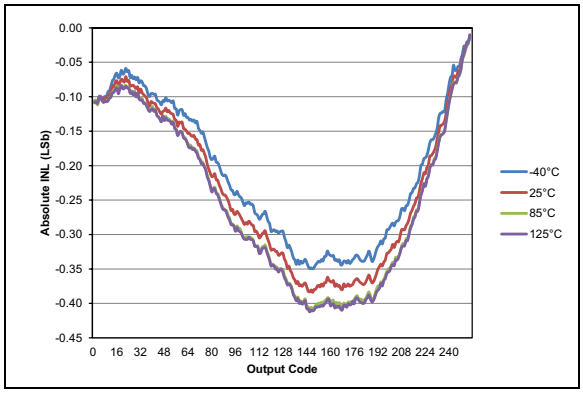


FIGURE 31-127: *Typical DAC INL Error, VDD = 5.0V, VREF = External 5V, PIC16F1782/3 Only.*

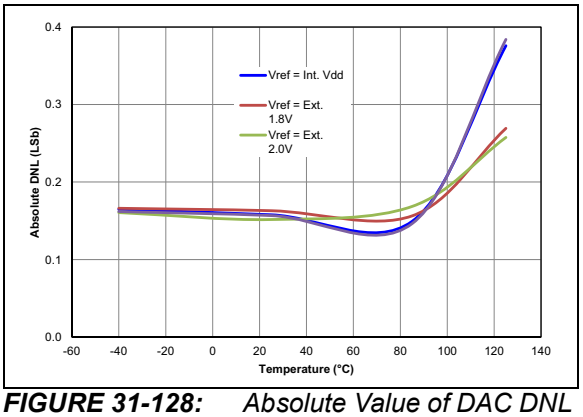


FIGURE 31-128: *Absolute Value of DAC DNL Error, VDD = 3.0V, VREF = VDD.*

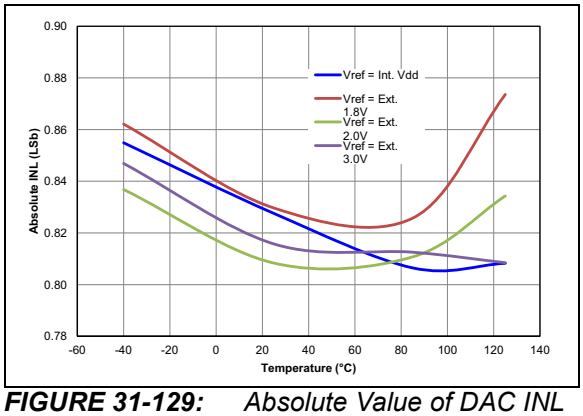


FIGURE 31-129: *Absolute Value of DAC INL Error, VDD = 3.0V.*

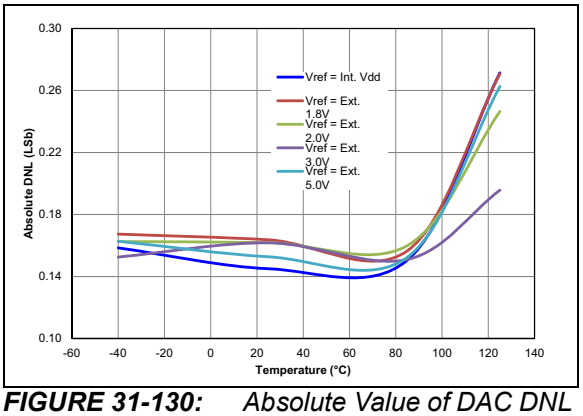


FIGURE 31-130: *Absolute Value of DAC DNL Error, VDD = 5.0V, PIC16F1782/3 Only.*

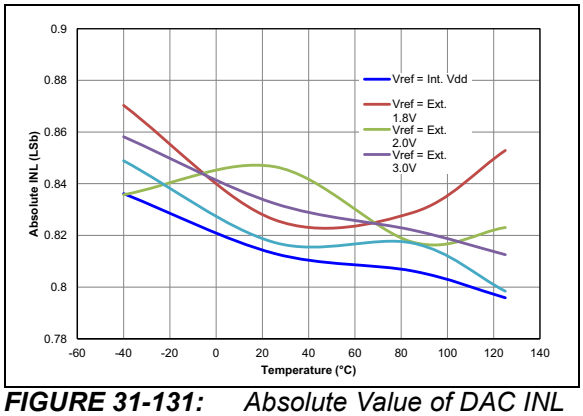


FIGURE 31-131: *Absolute Value of DAC INL Error, VDD = 5.0V, PIC16F1782/3 Only.*

32.2 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

32.3 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

32.4 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

32.5 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