

Welcome to [E-XFL.COM](https://www.e-xfl.com)

### What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

### Applications of "[Embedded - Microcontrollers](#)"

#### Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	32MHz
Connectivity	I <sup>2</sup> C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	11
Program Memory Size	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	16-VQFN Exposed Pad
Supplier Device Package	16-QFN (4x4)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/pic16f1825-i-ml">https://www.e-xfl.com/product-detail/microchip-technology/pic16f1825-i-ml</a>

### 3.2.2 SPECIAL FUNCTION REGISTER

The Special Function Registers (FSR) are registers used by the application to control the desired operation of peripheral functions in the device. The Special Function Registers occupy the 20 bytes after the core registers of every data memory bank (addresses x0Ch/x8Ch through x1Fh/x9Fh). The registers associated with the operation of the peripherals are described in the appropriate peripheral chapter of this data sheet.

### 3.2.3 GENERAL PURPOSE RAM

There are up to 80 bytes of GPR in each data memory bank. The Special Function Registers occupy the 20 bytes after the core registers of every data memory bank (addresses x0Ch/x8Ch through x1Fh/x9Fh).

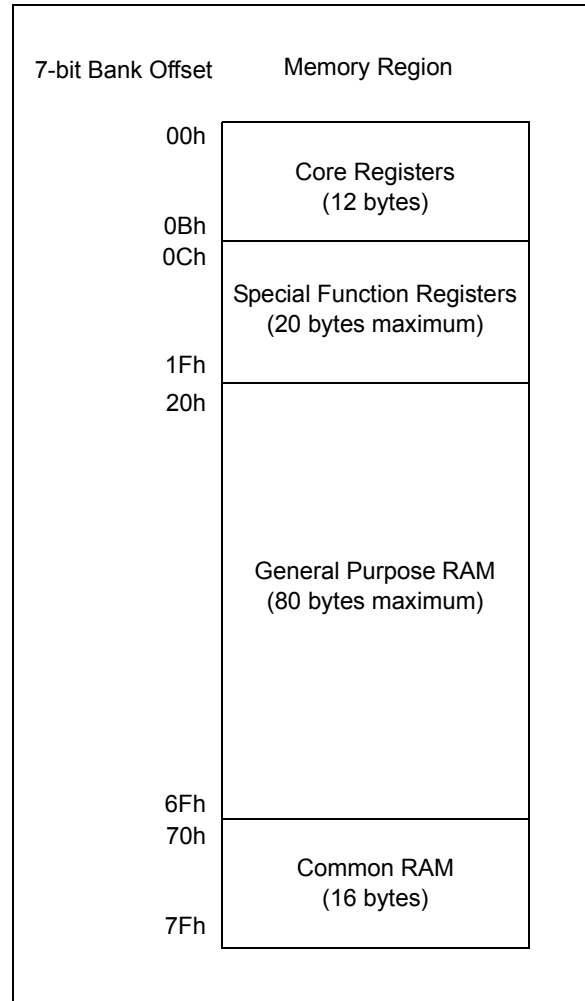
#### 3.2.3.1 Linear Access to GPR

The general purpose RAM can be accessed in a non-banked method via the FSRs. This can simplify access to large memory structures. See **Section 3.5.2 “Linear Data Memory”** for more information.

### 3.2.4 COMMON RAM

There are 16 bytes of common RAM accessible from all banks.

**FIGURE 3-2: BANKED MEMORY PARTITIONING**



### 3.2.5 DEVICE MEMORY MAPS

The memory maps for the device family are as shown in Table 3-2.

**TABLE 3-2: MEMORY MAP TABLES**

Device	Banks	Table No.
PIC16(L)F1825	0-7	Table 3-3
PIC16(L)F1829	8-15	Table 3-4
	16-23	Table 3-5
	24-31	Table 3-6
	31	Table 3-7

## 4.2 Code Protection

Code protection allows the device to be protected from unauthorized access. Program memory protection and data EEPROM protection are controlled independently. Internal access to the program memory and data EEPROM are unaffected by any code protection setting.

### 4.2.1 PROGRAM MEMORY PROTECTION

The entire program memory space is protected from external reads and writes by the  $\overline{CP}$  bit in Configuration Word 1. When  $\overline{CP} = 0$ , external reads and writes of program memory are inhibited and a read will return all '0's. The CPU can continue to read program memory, regardless of the protection bit settings. Writing the program memory is dependent upon the write protection setting. See **Section 4.3 "Write Protection"** for more information.

### 4.2.2 DATA EEPROM PROTECTION

The entire data EEPROM is protected from external reads and writes by the  $\overline{CPD}$  bit. When  $\overline{CPD} = 0$ , external reads and writes of data EEPROM are inhibited. The CPU can continue to read and write data EEPROM regardless of the protection bit settings.

## 4.3 Write Protection

Write protection allows the device to be protected from unintended self-writes. Applications, such as bootloader software, can be protected while allowing other regions of the program memory to be modified.

The  $WRT<1:0>$  bits in Configuration Word 2 define the size of the program memory block that is protected.

## 4.4 User ID

Four memory locations (8000h-8003h) are designated as ID locations where the user can store checksum or other code identification numbers. These locations are readable and writable during normal execution. See **Section 11.5 "User ID, Device ID and Configuration Word Access"** for more information on accessing these memory locations. For more information on checksum calculation, see the "*PIC16F/LF182X/PIC12F/LF1822 Memory Programming Specification*" (DS41390).

## 5.2.1.4 4xPLL

The oscillator module contains a 4xPLL that can be used with both external and internal clock sources to provide a system clock source. The input frequency for the 4xPLL must fall within specifications. See the PLL Clock Timing Specifications in **Section 30.0 “Electrical Specifications”**.

The 4xPLL may be enabled for use by one of two methods:

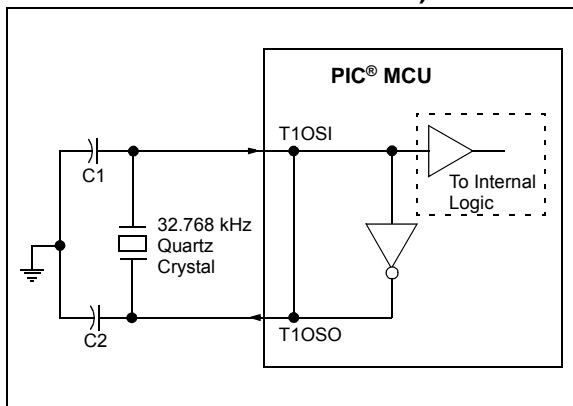
1. Program the PLEN bit in Configuration Word 2 to a ‘1’.
2. Write the SPLLEN bit in the OSCCON register to a ‘1’. If the PLEN bit in Configuration Word 2 is programmed to a ‘1’, then the value of SPLLEN is ignored.

## 5.2.1.5 TIMER1 Oscillator

The Timer1 Oscillator is a separate crystal oscillator that is associated with the Timer1 peripheral. It is optimized for timekeeping operations with a 32.768 kHz crystal connected between the T1OSO and T1OSI device pins.

The Timer1 Oscillator can be used as an alternate system clock source and can be selected during run-time using clock switching. Refer to **Section 5.3 “Clock Switching”** for more information.

**FIGURE 5-5: QUARTZ CRYSTAL OPERATION (TIMER1 OSCILLATOR)**



**Note 1:** Quartz crystal characteristics vary according to type, package and manufacturer. The user should consult the manufacturer data sheets for specifications and recommended application.

**2:** Always verify oscillator performance over the VDD and temperature range that is expected for the application.

**3:** For oscillator design assistance, reference the following Microchip Application Notes:

- AN826, “Crystal Oscillator Basics and Crystal Selection for rPIC® and PIC® Devices” (DS00826)
- AN849, “Basic PIC® Oscillator Design” (DS00849)
- AN943, “Practical PIC® Oscillator Analysis and Design” (DS00943)
- AN949, “Making Your Oscillator Work” (DS00949)
- TB097, “Interfacing a Micro Crystal MS1V-T1K 32.768 kHz Tuning Fork Crystal to a PIC16F690/SS” (DS91097)
- AN1288, “Design Practices for Low-Power External Oscillators” (DS01288)

## 5.2.1.6 External RC Mode

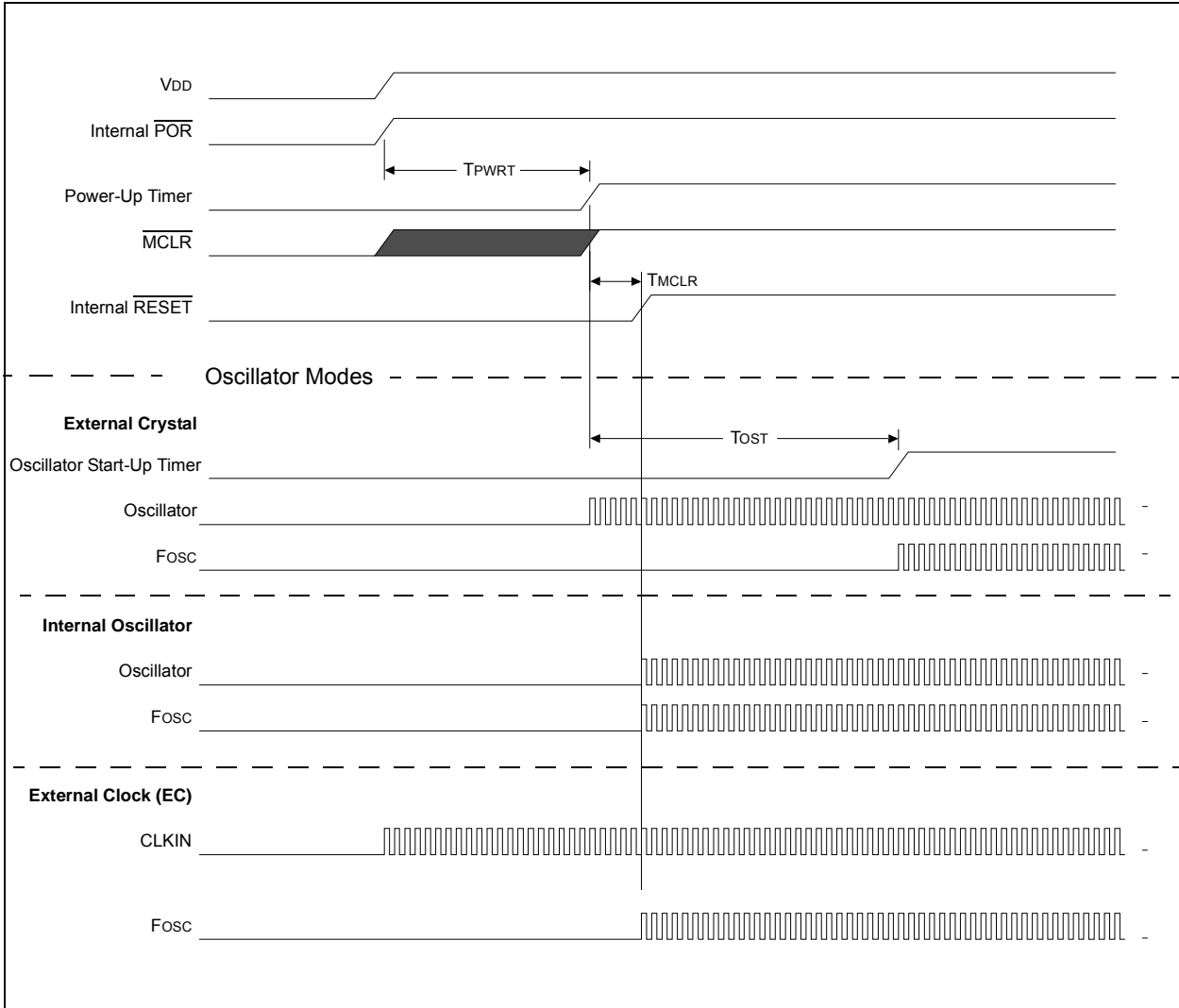
The external Resistor-Capacitor (RC) modes support the use of an external RC circuit. This allows the designer maximum flexibility in frequency choice while keeping costs to a minimum when clock accuracy is not required.

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 state of the CLKOUTEN bit in Configuration Word 1.

Figure 5-6 shows the external RC mode connections.

# PIC16(L)F1825/9

**FIGURE 7-3: RESET START-UP SEQUENCE**



## 8.6 Interrupt Control Registers

### 8.6.1 INTCON REGISTER

The INTCON register is a readable and writable register, that contains the various enable and flag bits for TMR0 register overflow, interrupt-on-change and external INT pin interrupts.

**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.

#### REGISTER 8-1: INTCON: INTERRUPT CONTROL REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R-0/0
GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF <sup>(1)</sup>
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      **GIE:** Global Interrupt Enable bit  
1 = Enables all active interrupts  
0 = Disables all interrupts
- bit 6      **PEIE:** Peripheral Interrupt Enable bit  
1 = Enables all active peripheral interrupts  
0 = Disables all peripheral interrupts
- bit 5      **TMR0IE:** Timer0 Overflow Interrupt Enable bit  
1 = Enables the Timer0 interrupt  
0 = Disables the Timer0 interrupt
- bit 4      **INTE:** INT External Interrupt Enable bit  
1 = Enables the INT external interrupt  
0 = Disables the INT external interrupt
- bit 3      **IOCIE:** Interrupt-on-Change Enable bit  
1 = Enables the interrupt-on-change  
0 = Disables the interrupt-on-change
- bit 2      **TMR0IF:** Timer0 Overflow Interrupt Flag bit  
1 = TMR0 register has overflowed  
0 = TMR0 register did not overflow
- bit 1      **INTF:** INT External Interrupt Flag bit  
1 = The INT external interrupt occurred  
0 = The INT external interrupt did not occur
- bit 0      **IOCIF:** Interrupt-on-Change Interrupt Flag bit<sup>(1)</sup>  
1 = When at least one of the interrupt-on-change pins changed state  
0 = None of the interrupt-on-change pins have changed state

**Note 1:** The IOCIF Flag bit is read-only and cleared when all the Interrupt-on-Change flags in the IOCxF register have been cleared by software.

**TABLE 10-3: SUMMARY OF REGISTERS ASSOCIATED WITH WATCHDOG TIMER**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
OSCCON	SPLLEN	IRCF<3:0>				—	SCS<1:0>		68
STATUS	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	22
WDTCON	—	—	WDTPS<4:0>					SWDTEN	100

**Legend:** x = unknown, u = unchanged, — = unimplemented locations read as '0'. Shaded cells are not used by Watchdog Timer.

**TABLE 10-4: SUMMARY OF CONFIGURATION WORD WITH WATCHDOG TIMER**

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	$\overline{CLKOUTEN}$	BOREN<1:0>	$\overline{CPD}$	48	
	7:0	$\overline{CP}$	MCLRE	$\overline{PWRTE}$	WDTE<1:0>	FOSC<2:0>				

**Legend:** — Unimplemented location, read as '0'. Shaded cells are not used by Watchdog Timer.

## 11.3.2 ERASING FLASH PROGRAM MEMORY

While executing code, program memory can only be erased by rows. To erase a row:

1. Load the EEADRH:EEADRL register pair with the address of new row to be erased.
2. Clear the CFGS bit of the EECON1 register.
3. Set the EEPGD, FREE, and WREN bits of the EECON1 register.
4. Write 55h, then AAh, to EECON2 (Flash programming unlock sequence).
5. Set control bit WR of the EECON1 register to begin the erase operation.
6. Poll the FREE bit in the EECON1 register to determine when the row erase has completed.

See Example 11-4.

After the “BSF EECON1, WR” instruction, the processor requires two cycles to set up the erase operation. The user must place two NOP instructions after the WR bit is set. The processor will halt internal operations for the typical 2 ms erase time. This is not Sleep mode as the clocks and peripherals will continue to run. After the erase cycle, the processor will resume operation with the third instruction after the EECON1 write instruction.

## 11.3.3 WRITING TO FLASH PROGRAM MEMORY

Program memory is programmed using the following steps:

1. Load the starting address of the word(s) to be programmed.
2. Load the write latches with data.
3. Initiate a programming operation.
4. Repeat steps 1 through 3 until all data is written.

Before writing to program memory, the word(s) to be written must be erased or previously unwritten. Program memory can only be erased one row at a time. No automatic erase occurs upon the initiation of the write.

Program memory can be written one or more words at a time. The maximum number of words written at one time is equal to the number of write latches. See Figure 11-2 for more details. The write latches are aligned to the address boundary defined by EEADRL as shown in Table 11-1. Write operations do not cross these boundaries. At the completion of a program memory write operation, the write latches are reset to contain 0x3FFF.

The following steps should be completed to load the write latches and program a block of program memory. These steps are divided into two parts. First, all write latches are loaded with data except for the last program memory location. Then, the last write latch is loaded and the programming sequence is initiated. A special unlock sequence is required to load a write latch with data or initiate a Flash programming operation. This unlock sequence should not be interrupted.

1. Set the EEPGD and WREN bits of the EECON1 register.
2. Clear the CFGS bit of the EECON1 register.
3. Set the LWLO bit of the EECON1 register. When the LWLO bit of the EECON1 register is ‘1’, the write sequence will only load the write latches and will not initiate the write to Flash program memory.
4. Load the EEADRH:EEADRL register pair with the address of the location to be written.
5. Load the EEDATH:EEDATL register pair with the program memory data to be written.
6. Write 55h, then AAh, to EECON2, then set the WR bit of the EECON1 register (Flash programming unlock sequence). The write latch is now loaded.
7. Increment the EEADRH:EEADRL register pair to point to the next location.
8. Repeat steps 5 through 7 until all but the last write latch has been loaded.
9. Clear the LWLO bit of the EECON1 register. When the LWLO bit of the EECON1 register is ‘0’, the write sequence will initiate the write to Flash program memory.
10. Load the EEDATH:EEDATL register pair with the program memory data to be written.
11. Write 55h, then AAh, to EECON2, then set the WR bit of the EECON1 register (Flash programming unlock sequence). The entire latch block is now written to Flash program memory.

It is not necessary to load the entire write latch block with user program data. However, the entire write latch block will be written to program memory.

An example of the complete write sequence for eight words is shown in Example 11-5. The initial address is loaded into the EEADRH:EEADRL register pair; the eight words of data are loaded using indirect addressing.



## 14.0 FIXED VOLTAGE REFERENCE (FVR)

The Fixed Voltage Reference, or FVR, is a stable voltage reference, independent of  $V_{DD}$ , with 1.024V, 2.048V or 4.096V selectable output levels. The output of the FVR can be configured to supply a reference voltage to the following:

- ADC input channel
- ADC positive reference
- Comparator positive input
- Digital-to-Analog Converter (DAC)
- Capacitive Sensing (CPS) module

The FVR can be enabled by setting the FVREN bit of the FVRCON register.

## 14.1 Independent Gain Amplifiers

The output of the FVR supplied to the ADC, Comparators, DAC and CPS are routed through two independent programmable gain amplifiers. Each amplifier can be configured to amplify the reference voltage by 1x, 2x or 4x, to produce the three possible voltage levels.

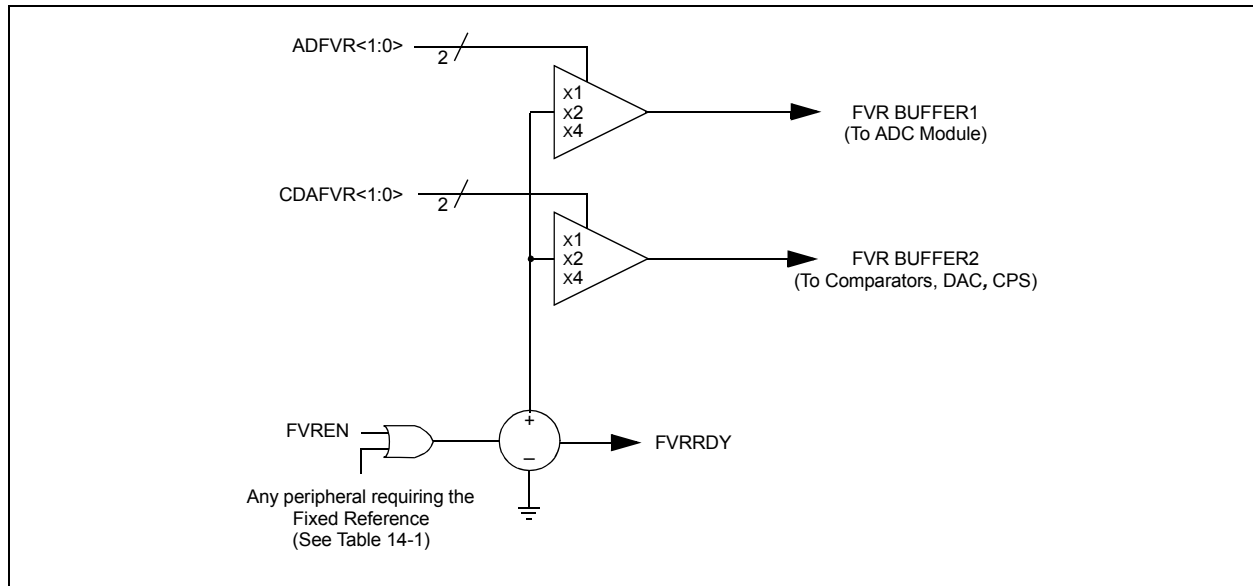
The ADFVR<1:0> bits of the FVRCON register are used to enable and configure the gain amplifier settings for the reference supplied to the ADC module. Reference **Section 16.0 “Analog-to-Digital Converter (ADC) Module”** for additional information.

The CDAFVR<1:0> bits of the FVRCON register are used to enable and configure the gain amplifier settings for the reference supplied to the DAC, CPS and comparator module. Reference **Section 17.0 “Digital-to-Analog Converter (DAC) Module”** and **Section 19.0 “Comparator Module”** for additional information.

## 14.2 FVR Stabilization Period

When the Fixed Voltage Reference module is enabled, it requires time for the reference and amplifier circuits to stabilize. Once the circuits stabilize and are ready for use, the FVRRDY bit of the FVRCON register will be set. See **Section 30.0 “Electrical Specifications”** for the minimum delay requirement.

**FIGURE 14-1: VOLTAGE REFERENCE BLOCK DIAGRAM**



# PIC16(L)F1825/9

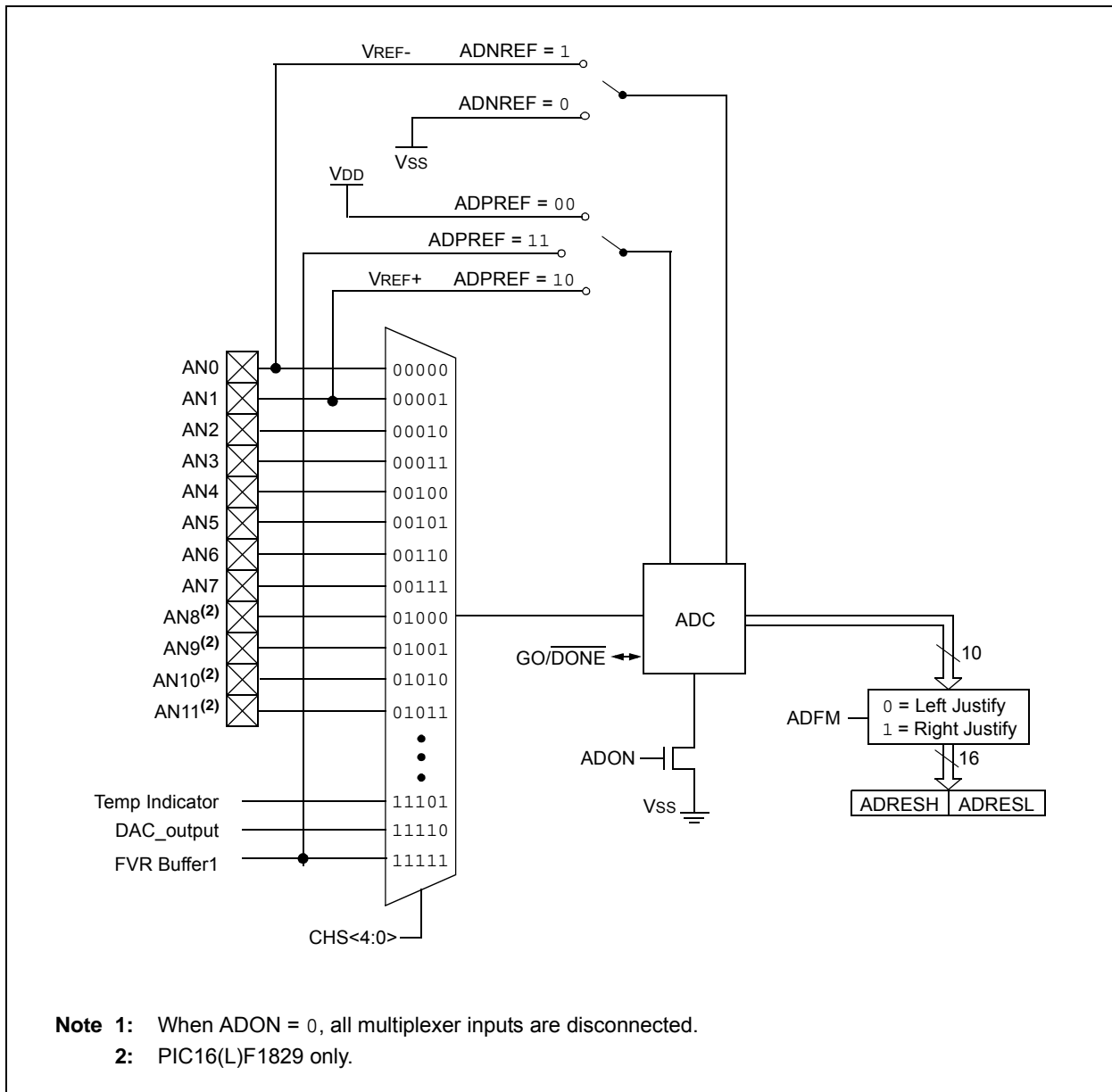
## 16.0 ANALOG-TO-DIGITAL CONVERTER (ADC) MODULE

The ADC can generate an interrupt upon completion of a conversion. This interrupt can be used to wake-up the device from Sleep.

The Analog-to-Digital Converter (ADC) allows conversion of an analog input signal to a 10-bit binary representation of that signal. This device uses analog inputs, which are multiplexed into a single sample and hold circuit. The output of the sample and hold is connected to the input of the converter. The converter generates a 10-bit binary result via successive approximation and stores the conversion result into the ADC result registers (ADRESH:ADRESL register pair). Figure 16-1 shows the block diagram of the ADC.

The ADC voltage reference is software selectable to be either internally generated or externally supplied.

**FIGURE 16-1: ADC BLOCK DIAGRAM**



- Note 1:** When ADON = 0, all multiplexer inputs are disconnected.
- 2:** PIC16(L)F1829 only.

## REGISTER 16-2: ADCON1: A/D CONTROL REGISTER 1

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
ADFM	ADCS<2:0>		—		ADNREF	ADPREF<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      **ADFM:** A/D Result Format Select bit  
 1 = Right justified. Six Most Significant bits of ADRESH are set to '0' when the conversion result is loaded.  
 0 = Left justified. Six Least Significant bits of ADRESL are set to '0' when the conversion result is loaded.
- bit 6-4    **ADCS<2:0>:** A/D Conversion Clock Select bits  
 000 = Fosc/2  
 001 = Fosc/8  
 010 = Fosc/32  
 011 = FRC (clock supplied from a dedicated RC oscillator)  
 100 = Fosc/4  
 101 = Fosc/16  
 110 = Fosc/64  
 111 = FRC (clock supplied from a dedicated RC oscillator)
- bit 3      **Unimplemented:** Read as '0'
- bit 2      **ADNREF:** A/D Negative Voltage Reference Configuration bit  
 0 = VREF- is connected to VSS  
 1 = VREF- is connected to external VREF- pin<sup>(1)</sup>
- bit 1-0    **ADPREF<1:0>:** A/D Positive Voltage Reference Configuration bits  
 00 = VREF+ is connected to VDD  
 01 = Reserved  
 10 = VREF+ is connected to external VREF+ pin<sup>(1)</sup>  
 11 = VREF+ is connected to internal Fixed Voltage Reference (FVR) module<sup>(1)</sup>

**Note 1:** When selecting the FVR or the VREF+ pin as the source of the positive reference, be aware that a minimum voltage specification exists. See **Section 30.0 "Electrical Specifications"** for details.

# PIC16(L)F1825/9

## 24.4.3 ENHANCED PWM AUTO-SHUTDOWN MODE

The PWM mode supports an Auto-Shutdown mode that will disable the PWM outputs when an external shutdown event occurs. Auto-Shutdown mode places the PWM output pins into a predetermined state. This mode is used to help prevent the PWM from damaging the application.

The auto-shutdown sources are selected using the CCPxAS<2:0> bits of the CCPxAS register. A shutdown event may be generated by:

- A logic '0' on the FLT0 pin
- A logic '1' on a Comparator (async\_CxOUT) output

A shutdown condition is indicated by the CCPxASE (Auto-Shutdown Event Status) bit of the CCPxAS register. If the bit is a '0', the PWM pins are operating normally. If the bit is a '1', the PWM outputs are in the shutdown state.

When a shutdown event occurs, two things happen:

The CCPxASE bit is set to '1'. The CCPxASE will remain set until cleared in firmware or an auto-restart occurs (see **Section 24.4.4 "Auto-restart Mode"**).

The enabled PWM pins are asynchronously placed in their shutdown states. The PWM output pins are grouped into pairs [PxA/PxC] and [PxB/PxD]. The state of each pin pair is determined by the PSSxAC and PSSxBD bits of the CCPxAS register. Each pin pair may be placed into one of three states:

- Drive logic '1'
- Drive logic '0'
- Tri-state (high-impedance)

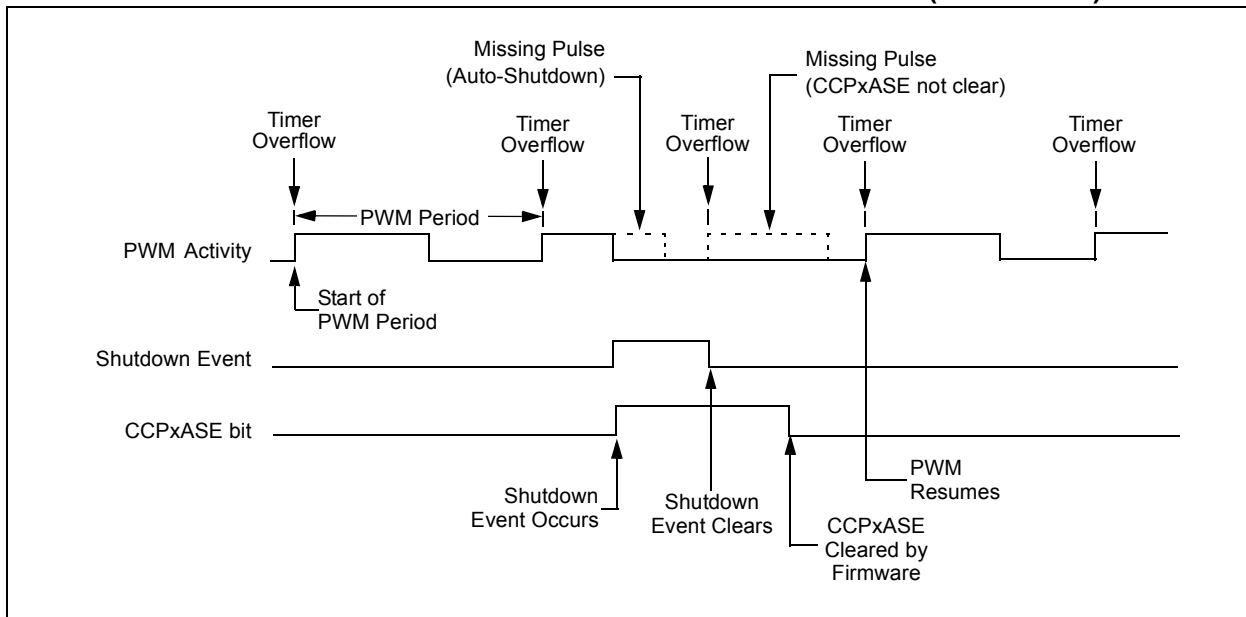
**Note 1:** The auto-shutdown condition is a level-based signal, not an edge-based signal. As long as the level is present, the auto-shutdown will persist.

**2:** Writing to the CCPxASE bit of the CCPxAS register is disabled while an auto-shutdown condition persists.

**3:** Once the auto-shutdown condition has been removed and the PWM restarted (either through firmware or auto-restart) the PWM signal will always restart at the beginning of the next PWM period.

**4:** Prior to an auto-shutdown event caused by a comparator output or FLT0 pin event, a software shutdown can be triggered in firmware by setting the CCPxASE bit of the CCPxAS register to '1'. The auto-restart feature tracks the active status of a shutdown caused by a comparator output or FLT0 pin event only. If it is enabled at this time, it will immediately clear this bit and restart the ECCP module at the beginning of the next PWM period.

**FIGURE 24-14: PWM AUTO-SHUTDOWN WITH FIRMWARE RESTART (PXRSEN = 0)**



# PIC16(L)F1825/9

The I<sup>2</sup>C interface supports the following modes and features:

- Master mode
- Slave mode
- Byte NACKing (Slave mode)
- Limited Multi-master support
- 7-bit and 10-bit addressing
- Start and Stop interrupts
- Interrupt masking
- Clock stretching
- Bus collision detection
- General call address matching
- Address masking
- Address Hold and Data Hold modes
- Selectable SDAx hold times

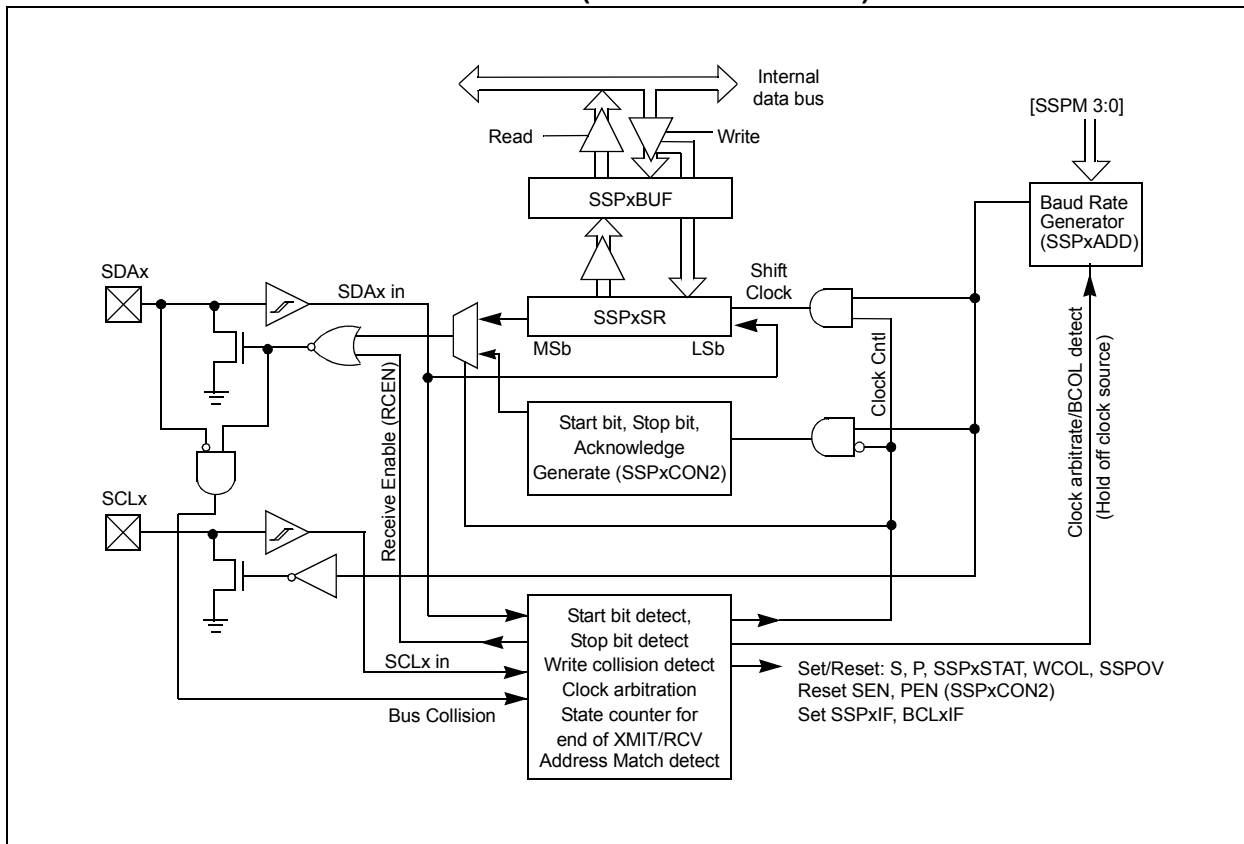
Figure 25-2 is a block diagram of the I<sup>2</sup>C interface module in Master mode. Figure 25-3 is a diagram of the I<sup>2</sup>C interface module in Slave mode.

The PIC16F1829 has two MSSP modules, MSSP1 and MSSP2, each module operating independently from the other.

**Note 1:** In devices with more than one MSSP module, it is very important to pay close attention to SSPxCONx register names. SSP1CON1 and SSP1CON2 registers control different operational aspects of the same module, while SSP1CON1 and SSP2CON1 control the same features for two different modules.

**2:** Throughout this section, generic references to an MSSP module in any of its operating modes may be interpreted as being equally applicable to MSSP1 or MSSP2. register names, module I/O signals, and bit names may use the generic designator 'x' to indicate the use of a numeral to distinguish a particular module when required.

**FIGURE 25-2: MSSPx BLOCK DIAGRAM (I<sup>2</sup>C™ MASTER MODE)**



## 25.2.6 SPI OPERATION IN SLEEP MODE

In SPI Master mode, module clocks may be operating at a different speed than when in full power mode; in the case of the Sleep mode, all clocks are halted.

Special care must be taken by the user when the MSSPx clock is much faster than the system clock.

In Slave mode, when MSSPx interrupts are enabled, after the master completes sending data, an MSSPx interrupt will wake the controller from Sleep.

If an exit from Sleep mode is not desired, MSSPx interrupts should be disabled.

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

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

**TABLE 25-1: SUMMARY OF REGISTERS ASSOCIATED WITH SPI OPERATION**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	122
ANSELB <sup>(1)</sup>	—	—	ANSB5	ANSB4	—	—	—	—	129
ANSELC	ANSC7 <sup>(1)</sup>	ANSC6 <sup>(1)</sup>	—	—	ANSC3	ANSC2	ANSC1	ANSC0	134
APFCON0	RXDTSEL	SDO1SEL <sup>(2)</sup>	SS1SEL <sup>(2)</sup>	—	T1GSEL	TXCKSEL	—	—	118
APFCON1	—	—	SDO2SEL <sup>(1)</sup>	SS2SEL <sup>(1)</sup>	P1DSEL	P1CSEL	P2BSEL	CCP2SEL	119
INLVLA	—	—	INLVLA5 <sup>(1)</sup>	INLVLA4 <sup>(1)</sup>	INLVLA3	INLVLA2	INLVLA1	INLVLA0	124
INVLVB <sup>(1)</sup>	INVLVB7	INVLVB6	INVLVB5	INVLVB4	—	—	—	—	129
INLVLC	INLVLC7 <sup>(1)</sup>	INLVLC6 <sup>(1)</sup>	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1 <sup>(1)</sup>	INLVLC0 <sup>(1)</sup>	135
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	87
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	88
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	92
SSP1BUF	Synchronous Serial Port Receive Buffer/Transmit Register								233*
SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				277
SSP1CON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	279
SSP1STAT	SMP	CKE	D/Ā	P	S	R/W	UA	BF	276
TRISA	—	—	TRISA5 <sup>(1)</sup>	TRISA4 <sup>(1)</sup>	TRISA3	TRISA2	TRISA1	TRISA0	122
TRISB <sup>(1)</sup>	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	128
TRISC	TRISC7 <sup>(1)</sup>	TRISC6 <sup>(1)</sup>	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1 <sup>(1)</sup>	TRISC0 <sup>(1)</sup>	133

**Legend:** — Unimplemented location, read as '0'. Shaded cells are not used by the MSSP1 in SPI mode.  
\* Page provides register information.

**Note 1:** PIC16(L)F1829 only.  
**2:** PIC16(L)F1825 only.

# PIC16(L)F1825/9

**TABLE 25-3: SUMMARY OF REGISTERS ASSOCIATED WITH I<sup>2</sup>C™ OPERATION**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page
INLVLA	—	—	INLVLA5 <sup>(1)</sup>	INLVLA4	INLVLA3 <sup>(2)</sup>	INLVLA2	INLVLA1	INLVLA0	124
INVLVB <sup>(1)</sup>	INVLVB7	INVLVB6	INVLVB5	INVLVB4	—	—	—	—	129
INLVLC	INLVLC7 <sup>(1)</sup>	INLVLC6 <sup>(1)</sup>	INLVLC5	INLVLC4	INLVLC3 <sup>(2)</sup>	INLVLC2 <sup>(2)</sup>	INLVLC1	INLVLC0	135
INTCON	GIE	PEIE	TMR0IE	INTE	IOCFIE	TMR0IF	INTF	IOCFIF	87
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	88
PIE2	OSFIE	C2IE	C1IE	EEIE	BCL1IE	—	—	CCP2IE	89
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	92
PIR2	OSFIF	C2IF	C1IF	EEIF	BCL1IF	—	—	CCP2IF	93
SSP1ADD	ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0	280
SSP1BUF	Synchronous Serial Port Receive Buffer/Transmit Register								233*
SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				277
SSP1CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	278
SSP1CON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	279
SSP1MSK	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0	280
SSP1STAT	SMP	CKE	D/Ā	P	S	R/W	UA	BF	276
TRISA	—	—	TRISA5 <sup>(1)</sup>	TRISA4	TRISA3 <sup>(2)</sup>	TRISA2	TRISA1	TRISA0	122
TRISB <sup>(1)</sup>	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	128
TRISC	TRISC7 <sup>(1)</sup>	TRISC6 <sup>(1)</sup>	TRISC5	TRISC4	TRISC3 <sup>(2)</sup>	TRISC2 <sup>(2)</sup>	TRISC1	TRISC0	133

**Legend:** — Unimplemented location, read as '0'. Shaded cells are not used by the MSSP module in I<sup>2</sup>C™ mode.

\* Page provides register information.

- Note 1:** PIC16(L)F1829 only.  
**Note 2:** PIC16(L)F1825 only.

# PIC16(L)F1825/9

## 26.3.1 AUTO-BAUD DETECT

The EUSART module supports automatic detection and calibration of the baud rate.

In the Auto-Baud Detect (ABD) mode, the clock to the BRG is reversed. Rather than the BRG clocking the incoming RX signal, the RX signal is timing the BRG. The Baud Rate Generator is used to time the period of a received 55h (ASCII “U”) which is the Sync character for the LIN bus. The unique feature of this character is that it has five rising edges including the Stop bit edge.

Setting the ABDEN bit of the BAUDCON register starts the auto-baud calibration sequence (Figure 26-6). While the ABD sequence takes place, the EUSART state machine is held in Idle. On the first rising edge of the receive line, after the Start bit, the SPBRG begins counting up using the BRG counter clock as shown in Figure 26-6. The fifth rising edge will occur on the RX pin at the end of the eighth bit period. At that time, an accumulated value totaling the proper BRG period is left in the SPBRGH, SPBRGL register pair, the ABDEN bit is automatically cleared and the RCIF interrupt flag is set. The value in the RCREG needs to be read to clear the RCIF interrupt. RCREG content should be discarded. When calibrating for modes that do not use the SPBRGH register the user can verify that the SPBRGL register did not overflow by checking for 00h in the SPBRGH register.

The BRG auto-baud clock is determined by the BRG16 and BRGH bits as shown in Table 26-6. During ABD, both the SPBRGH and SPBRGL registers are used as a 16-bit counter, independent of the BRG16 bit setting. While calibrating the baud rate period, the SPBRGH

and SPBRGL registers are clocked at 1/8th the BRG base clock rate. The resulting byte measurement is the average bit time when clocked at full speed.

**Note 1:** If the WUE bit is set with the ABDEN bit, auto-baud detection will occur on the byte following the Break character (see Section 26.3.3 “Auto-Wake-up on Break”).

**2:** It is up to the user to determine that the incoming character baud rate is within the range of the selected BRG clock source. Some combinations of oscillator frequency and EUSART baud rates are not possible.

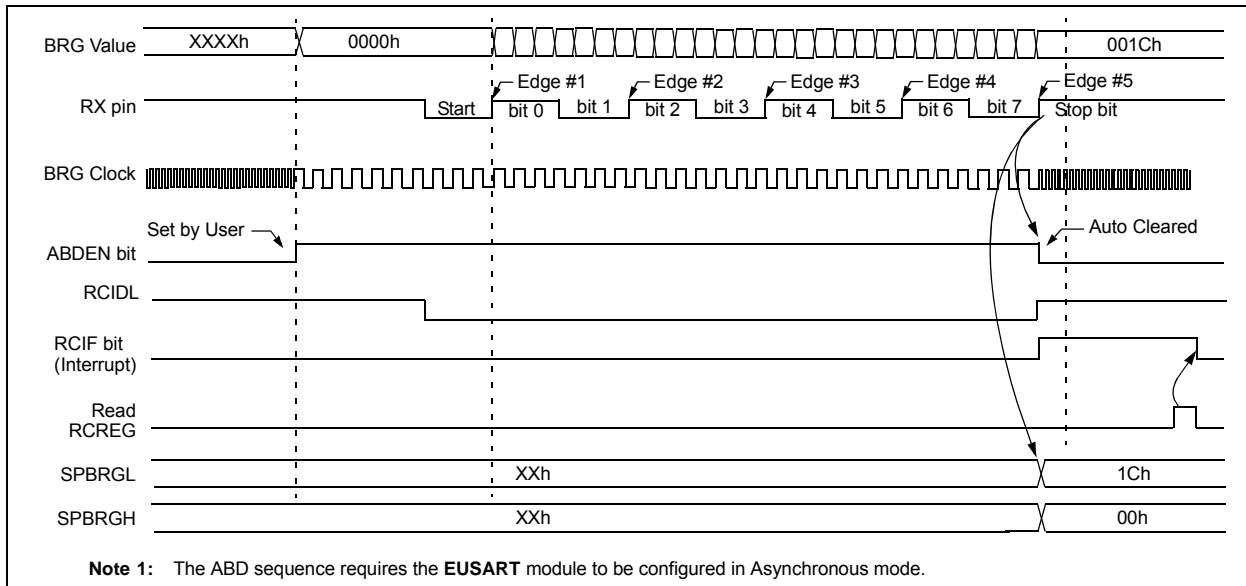
**3:** During the auto-baud process, the auto-baud counter starts counting at 1. Upon completion of the auto-baud sequence, to achieve maximum accuracy, subtract 1 from the SPBRGH:SPBRGL register pair.

**TABLE 26-6: BRG COUNTER CLOCK RATES**

BRG16	BRGH	BRG Base Clock	BRG ABD Clock
0	0	Fosc/64	Fosc/512
0	1	Fosc/16	Fosc/128
1	0	Fosc/16	Fosc/128
1	1	Fosc/4	Fosc/32

**Note:** During the ABD sequence, SPBRGL and SPBRGH registers are both used as a 16-bit counter, independent of BRG16 setting.

**FIGURE 26-6: AUTOMATIC BAUD RATE CALIBRATION**





# PIC16(L)F1825/9

## 27.4 Current Ranges

The Capacitive Sensing Oscillator can operate within several different current ranges, depending on the Voltage Reference mode and current range selections. Within each of the two Voltage Reference modes there are four current ranges.

Selection between the Voltage Reference modes is controlled by the CPSRM bit of the CPSCON0 register. Clearing this bit selects the Fixed Voltage References provided by the Capacitive Sensing Oscillator module. Setting this bit selects the variable voltage references supplied by the Fixed Voltage Reference (FVR) module and the Digital-to-Analog Converter (DAC) module. See **Section 27.3 “Voltage References”** for more information on configuring the voltage references. Selecting the current range within the voltage reference mode is controlled by configuring the CPSRNG<1:0> bits in the CPSCON0 register. See Table 27-1 for proper current mode

selection.

The Noise Detection mode is unique in that it disables the constant current source associated with the selected input pin, but leaves the rest of the oscillator circuitry and pin structure active. This eliminates the oscillation frequency on the analog pin and greatly reduces the current consumed by the Oscillator module. When noise is introduced onto the pin, the oscillator is driven at the frequency determined by the noise. This produces a detectable signal at the comparator stage, indicating the presence of activity on the pin. Figure 27-2 shows a more detailed drawing of the constant current sources and comparators associated with the oscillator and input pin.

**TABLE 27-1: CURRENT MODE SELECTION**

CPSRM	Voltage Reference Mode	CPSRNG<1:0>	Current Range <sup>(1)</sup>
0	Fixed	00	Off
		01	Low
		10	Medium
		11	High
1	Variable	00	Noise Detection
		01	Low
		10	Medium
		11	High

**Note 1:** See Power-Down Currents (IPD) in **Section 30.3 “DC Characteristics: PIC16(L)F1825/9-I/E (Power-Down)”** for more information.

# PIC16(L)F1825/9

**TABLE 30-4: CLKOUT AND I/O TIMING PARAMETERS**

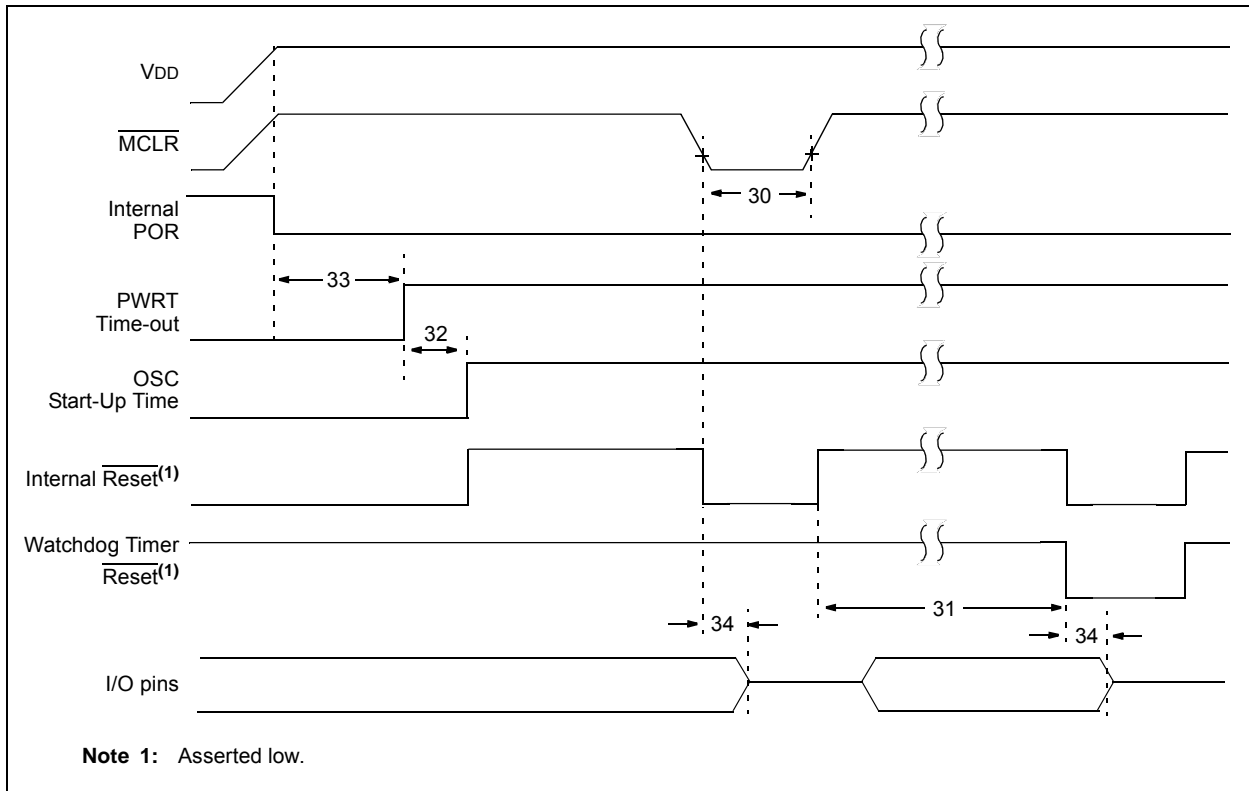
Standard Operating Conditions (unless otherwise stated) Operating Temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$							
Param No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
OS11	TosH2ckL	Fosc↑ to CLKOUT↓ <sup>(1)</sup>	—	—	70	ns	VDD = 3.3-5.0V
OS12	TosH2ckH	Fosc↑ to CLKOUT↑ <sup>(1)</sup>	—	—	72	ns	VDD = 3.3-5.0V
OS13	TckL2ioV	CLKOUT↓ to Port out valid <sup>(1)</sup>	—	—	20	ns	
OS14	TioV2ckH	Port input valid before CLKOUT↑ <sup>(1)</sup>	Tosc + 200 ns	—	—	ns	
OS15	TosH2ioV	Fosc↑ (Q1 cycle) to Port out valid	—	50	70*	ns	VDD = 3.3-5.0V
OS16	TosH2ioI	Fosc↑ (Q2 cycle) to Port input invalid (I/O in hold time)	50	—	—	ns	VDD = 3.3-5.0V
OS17	TioV2osH	Port input valid to Fosc↑ (Q2 cycle) (I/O in setup time)	20	—	—	ns	
OS18*	TioR	Port output rise time	—	40 15	72 32	ns	VDD = 1.8V VDD = 3.3-5.0V
OS19*	TioF	Port output fall time	—	28 15	55 30	ns	VDD = 1.8V VDD = 3.3-5.0V
OS20*	Tinp	INT pin input high or low time	25	—	—	ns	
OS21*	Tioc	Interrupt-on-change new input level time	25	—	—	ns	

\* These parameters are characterized but not tested.

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

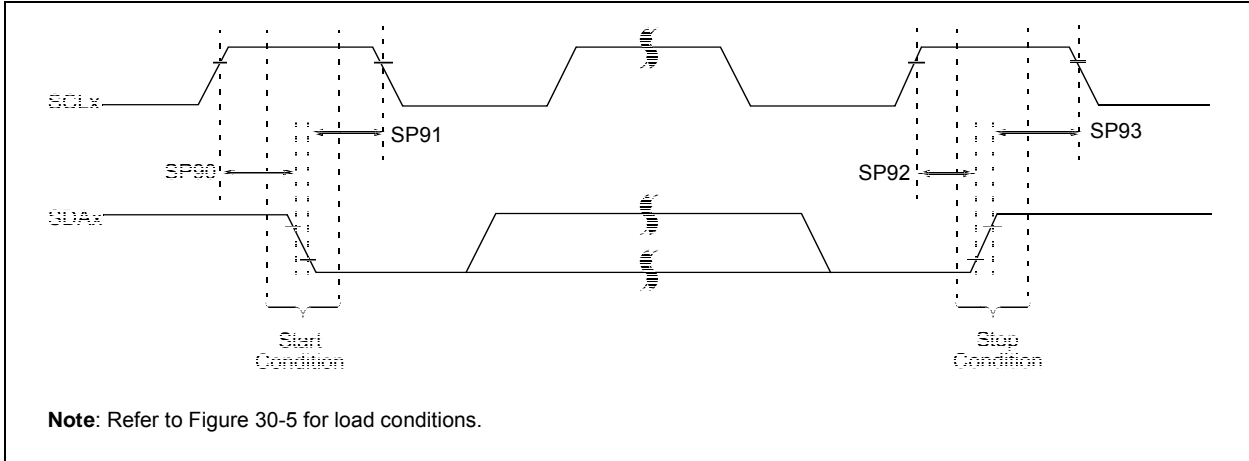
**Note 1:** Measurements are taken in RC mode where CLKOUT output is 4 x Tosc.

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



**Note 1:** Asserted low.

**FIGURE 30-20: I<sup>2</sup>C™ BUS START/STOP BITS TIMING**

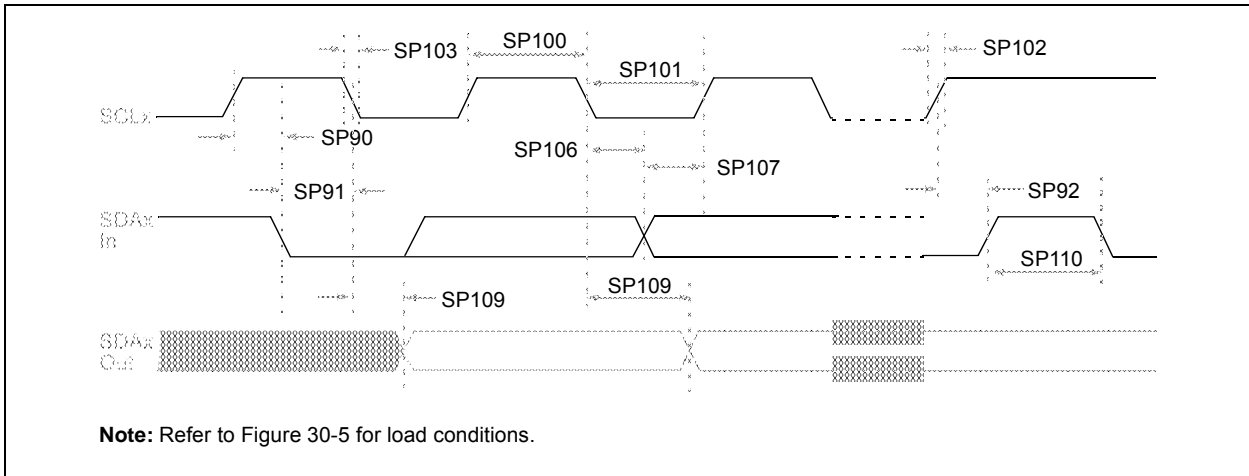


**TABLE 30-16: I<sup>2</sup>C™ BUS START/STOP BITS REQUIREMENTS**

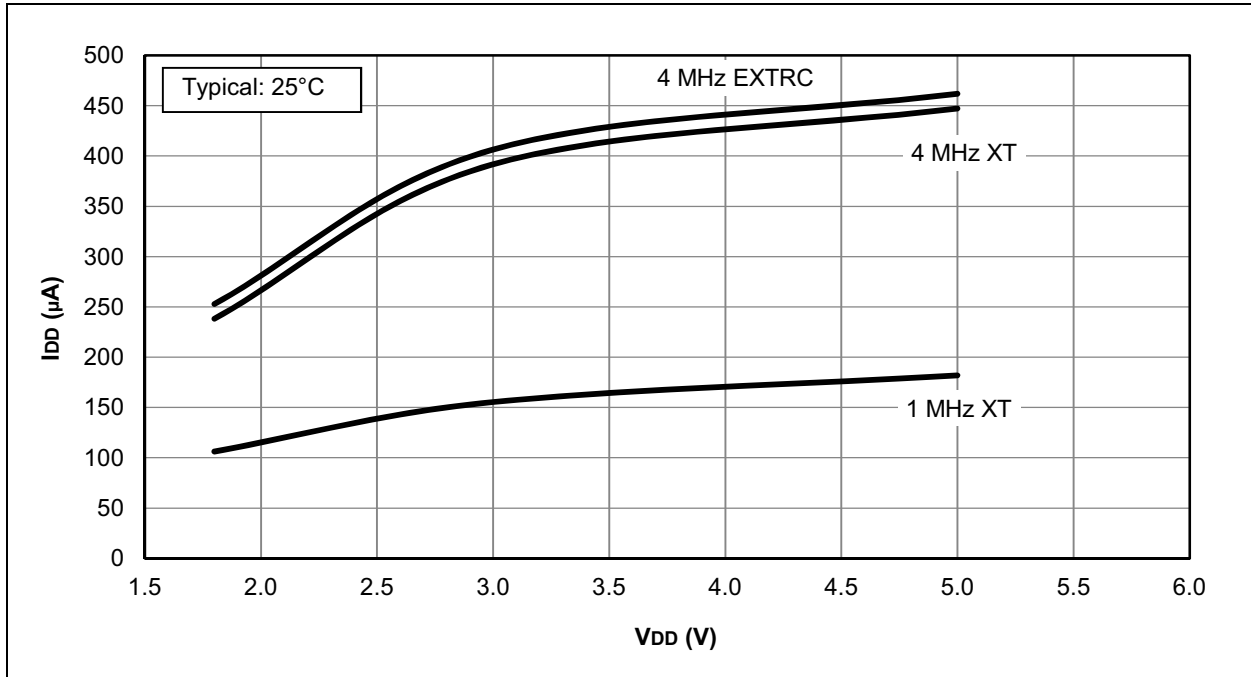
Param No.	Symbol	Characteristic	Min.	Typ	Max.	Units	Conditions	
SP90*	TSU:STA	Start condition	100 kHz mode	4700	—	—	ns	Only relevant for Repeated Start condition
		Setup time	400 kHz mode	600	—	—		
SP91*	THD:STA	Start condition	100 kHz mode	4000	—	—	ns	After this period, the first clock pulse is generated
		Hold time	400 kHz mode	600	—	—		
SP92*	TSU:STO	Stop condition	100 kHz mode	4700	—	—	ns	
		Setup time	400 kHz mode	600	—	—		
SP93	THD:STO	Stop condition	100 kHz mode	4000	—	—	ns	
		Hold time	400 kHz mode	600	—	—		

\* These parameters are characterized but not tested.

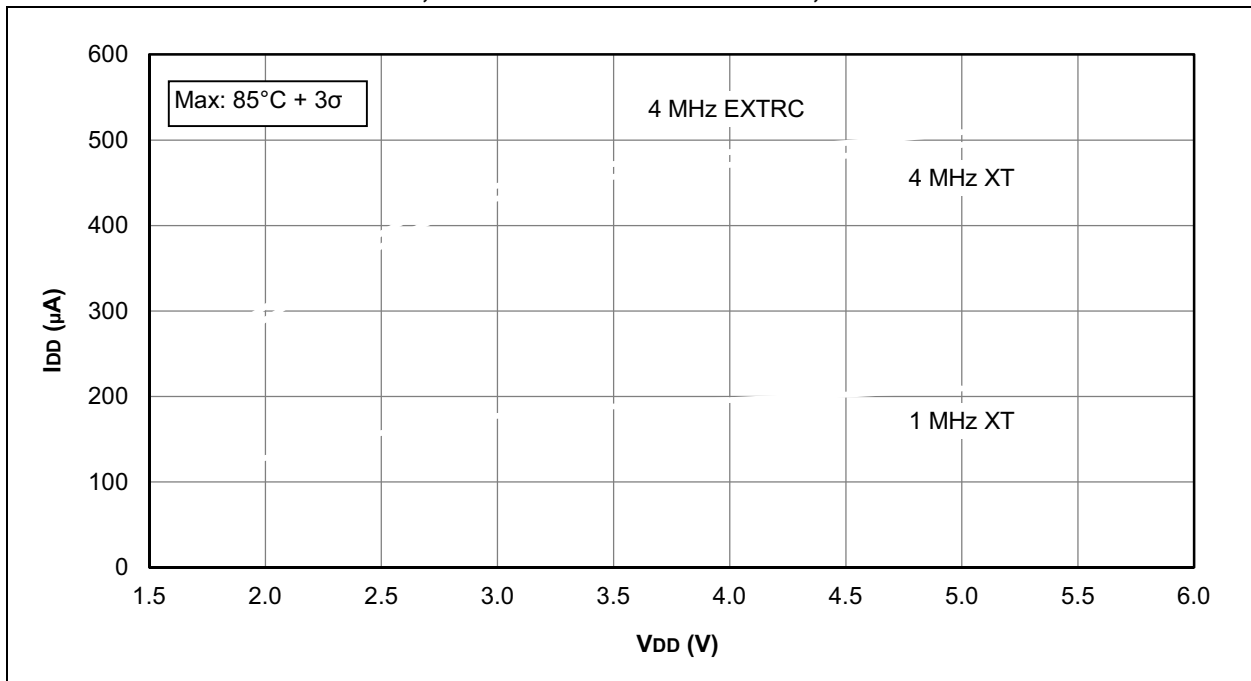
**FIGURE 30-21: I<sup>2</sup>C™ BUS DATA TIMING**



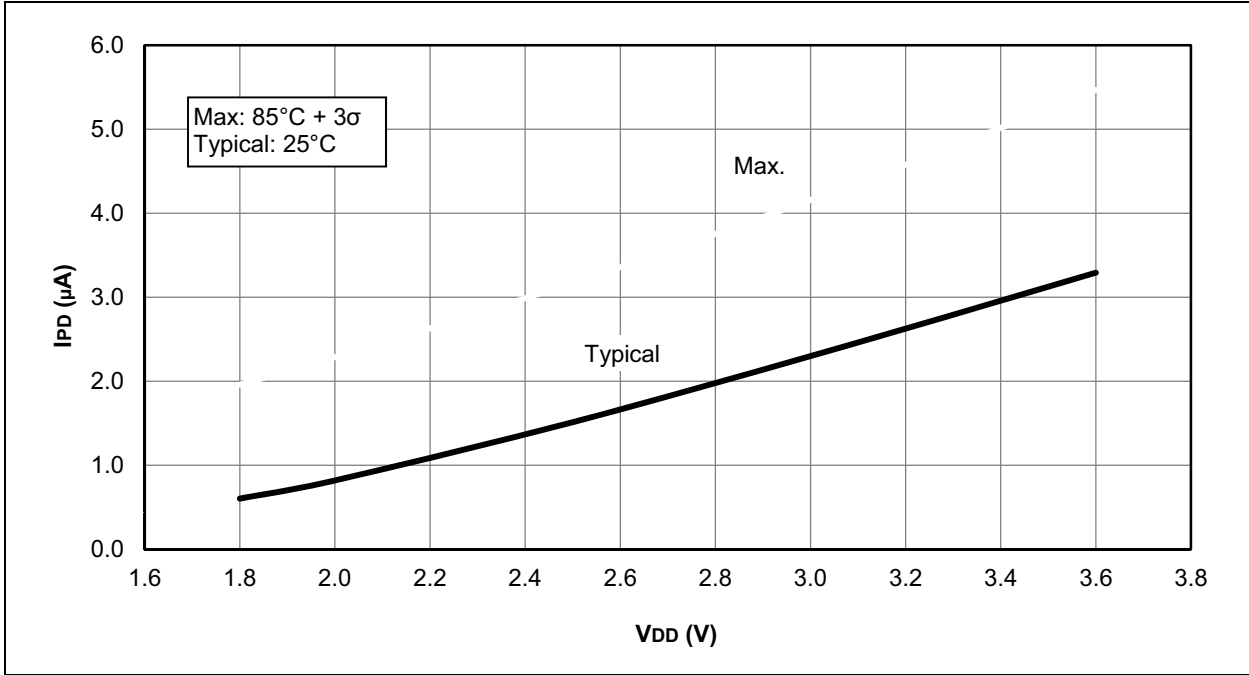
**FIGURE 31-5: I<sub>DD</sub> TYPICAL, XT AND EXTRC OSCILLATOR, PIC16F1825/9 ONLY**



**FIGURE 31-6: I<sub>DD</sub> MAXIMUM, XT AND EXTRC OSCILLATOR, PIC16F1825/9 ONLY**



**FIGURE 31-29: I<sub>PD</sub>, TIMER1 OSCILLATOR (F<sub>osc</sub> = 32 kHz), PIC16LF1825/9 ONLY**



**FIGURE 31-30: I<sub>PD</sub>, TIMER1 OSCILLATOR (F<sub>osc</sub> = 32 kHz), PIC16F1825/9 ONLY**

