



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 ² 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 ~ 3.6V
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	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf1825t-i-ml

Table of Contents

1.0	Device Overview	9
2.0	Enhanced Mid-range CPU	17
3.0	Memory Organization	19
4.0	Device Configuration	47
5.0	Oscillator Module (With Fail-Safe Clock Monitor).....	53
6.0	Reference Clock Module	71
7.0	Resets	74
8.0	Interrupts	82
9.0	Power-Down Mode (Sleep)	96
10.0	Watchdog Timer	98
11.0	Data EEPROM and Flash Program Memory Control.....	102
12.0	I/O Ports	116
13.0	Interrupt-on-Change	136
14.0	Fixed Voltage Reference (FVR)	141
15.0	Temperature Indicator Module	143
16.0	Analog-to-Digital Converter (ADC) Module	144
17.0	Digital-to-Analog Converter (DAC) Module	157
18.0	SR Latch.....	162
19.0	Comparator Module.....	167
20.0	Timer0 Module	174
21.0	Timer1 Module with Gate Control.....	177
22.0	Timer2/4/6 Modules.....	188
23.0	Data Signal Modulator.....	192
24.0	Capture/Compare/PWM Modules	201
25.0	Master Synchronous Serial Port (MSSP1 and MSSP2) Module	229
26.0	Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART).....	281
27.0	Capacitive Sensing (CPS) Module	309
28.0	In-Circuit Serial Programming™ (ICSP™)	317
29.0	Instruction Set Summary	320
30.0	Electrical Specifications.....	334
31.0	DC and AC Characteristics Graphs and Charts	370
32.0	Development Support.....	401
33.0	Packaging Information.....	405
	Appendix A: Data Sheet Revision History.....	432
	Appendix B: Migrating From Other PIC® Devices	432
	The Microchip Web Site	433
	Customer Change Notification Service	433
	Customer Support.....	433
	Product Identification System	434

7.10 Determining the Cause of a Reset

Upon any Reset, multiple bits in the STATUS and PCON register are updated to indicate the cause of the Reset. Table 7-3 and Table 7-4 show the Reset conditions of these registers.

TABLE 7-3: RESET STATUS BITS AND THEIR SIGNIFICANCE

STKOVF	STKUNF	RMCLR	RI	POR	BOR	TO	PD	Condition
0	0	1	1	0	x	1	1	Power-on Reset
0	0	1	1	0	x	0	x	Illegal, $\overline{TO}$ is set on $\overline{POR}$
0	0	1	1	0	x	x	0	Illegal, $\overline{PD}$ is set on $\overline{POR}$
0	0	1	1	u	0	1	1	Brown-out Reset
u	u	u	u	u	u	0	u	WDT Reset
u	u	u	u	u	u	0	0	WDT Wake-up from Sleep
u	u	u	u	u	u	1	0	Interrupt Wake-up from Sleep
u	u	0	u	u	u	u	u	$\overline{MCLR}$ Reset during normal operation
u	u	0	u	u	u	1	0	$\overline{MCLR}$ Reset during Sleep
u	u	u	0	u	u	u	u	RESET Instruction Executed
1	u	u	u	u	u	u	u	Stack Overflow Reset (STVREN = 1)
u	1	u	u	u	u	u	u	Stack Underflow Reset (STVREN = 1)

TABLE 7-4: RESET CONDITION FOR SPECIAL REGISTERS⁽²⁾

Condition	Program Counter	STATUS Register	PCON Register
Power-on Reset	0000h	---1 1000	00-- 110x
$\overline{MCLR}$ Reset during normal operation	0000h	---u uuuu	uu-- 0uuu
$\overline{MCLR}$ Reset during Sleep	0000h	---1 0uuu	uu-- 0uuu
WDT Reset	0000h	---0 uuuu	uu-- uuuu
WDT Wake-up from Sleep	PC + 1	---0 0uuu	uu-- uuuu
Brown-out Reset	0000h	---1 1uuu	00-- 11u0
Interrupt Wake-up from Sleep	PC + 1 ⁽¹⁾	---1 0uuu	uu-- uuuu
RESET Instruction Executed	0000h	---u uuuu	uu-- u0uu
Stack Overflow Reset (STVREN = 1)	0000h	---u uuuu	1u-- uuuu
Stack Underflow Reset (STVREN = 1)	0000h	---u uuuu	u1-- uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, reads as '0'.

Note 1: When the wake-up is due to an interrupt and Global Enable bit (GIE) is set, the return address is pushed on the stack and PC is loaded with the interrupt vector (0004h) after execution of PC + 1.

2: If a Status bit is not implemented, that bit will be read as '0'.

EXAMPLE 11-4: ERASING ONE ROW OF PROGRAM MEMORY

```
; This row erase routine assumes the following:
; 1. A valid address within the erase block is loaded in ADDRH:ADDRL
; 2. ADDRH and ADDRL are located in shared data memory 0x70 - 0x7F

        BCF      INTCON,GIE      ; Disable ints so required sequences will execute properly
        BANKSEL  EEADRL
        MOVF     ADDRL,W         ; Load lower 8 bits of erase address boundary
        MOVWF    EEADRL
        MOVF     ADDRH,W        ; Load upper 6 bits of erase address boundary
        MOVWF    EEADRH
        BSF      EECON1,EEPGD    ; Point to program memory
        BCF      EECON1,CFGSR    ; Not configuration space
        BSF      EECON1,FREE     ; Specify an erase operation
        BSF      EECON1,WREN     ; Enable writes

        MOV LW   55h            ; Start of required sequence to initiate erase
        MOVWF    EECON2         ; Write 55h
        MOV LW   0AAh          ;
        MOVWF    EECON2         ; Write AAh
        BSF      EECON1,WR      ; Set WR bit to begin erase
        NOP      ; Any instructions here are ignored as processor
        NOP      ; halts to begin erase sequence
        NOP      ; Processor will stop here and wait for erase complete.

        ; after erase processor continues with 3rd instruction

        BCF      EECON1,WREN     ; Disable writes
        BSF      INTCON,GIE     ; Enable interrupts
```

PIC16(L)F1825/9

REGISTER 12-7: WPUA: WEAK PULL-UP PORTA REGISTER

U-0	U-0	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
—	—	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0
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 5-0 **WPUA<5:0>:** Weak Pull-up Register bits^(1,2)

1 = Pull-up enabled

0 = Pull-up disabled

Note 1: Global **WPUEN** bit of the **OPTION_REG** register must be cleared for individual pull-ups to be enabled.

Note 2: The weak pull-up device is automatically disabled if the pin is in configured as an output.

REGISTER 12-8: INLVLA: PORTA INPUT LEVEL CONTROL REGISTER

U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-1/1	R/W-0/0	R/W-0/0
—	—	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0
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 5-0 **INLVLA<5:0>:** PORTA Input Level Select bits

For RA<5:0> pins, respectively

1 = ST input used for port reads and interrupt-on-change

0 = TTL input used for port reads and interrupt-on-change

TABLE 12-3: 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	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	123
APFCON0	RXDTSEL	SDO1SEL ⁽²⁾	SS1SEL ⁽²⁾	—	T1GSEL	TXCKSEL	—	—	118
APFCON1	—	—	SDO2SEL ⁽¹⁾	SS2SEL ⁽¹⁾	P1DSEL	P1CSEL	P2BSEL	CCP2SEL	119
INLVLA	—	—	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0	124
LATA	—	—	LATA5	LATA4	—	LATA2	LATA1	LATA0	123
OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			176
PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	122
TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	122
WPUA	—	—	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0	124

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

Note 1: PIC16(L)F1829 only.

Note 2: PIC16(L)F1825 only.

TABLE 12-4: 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	CLKOUTEN	BOREN<1:0>		CPD	48
	7:0	CP	MCLRE	PWRTÉ	WDTE<1:0>		FOSC<2:0>			

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

15.0 TEMPERATURE INDICATOR MODULE

This family of devices is equipped with a temperature circuit designed to measure the operating temperature of the silicon die. The circuit's range of operating temperature falls between -40°C and +85°C. The output is a voltage that is proportional to the device temperature. The output of the temperature indicator is internally connected to the device ADC.

The circuit may be used as a temperature threshold detector or a more accurate temperature indicator, depending on the level of calibration performed. A one-point calibration allows the circuit to indicate a temperature closely surrounding that point. A two-point calibration allows the circuit to sense the entire range of temperature more accurately. Reference Application Note AN1333, "Use and Calibration of the Internal Temperature Indicator" (DS01333) for more details regarding the calibration process.

15.1 Circuit Operation

Figure 15-1 shows a simplified block diagram of the temperature circuit. The proportional voltage output is achieved by measuring the forward voltage drop across multiple silicon junctions.

Equation 15-1 describes the output characteristics of the temperature indicator.

EQUATION 15-1: V_{OUT} RANGES

High Range: $V_{OUT} = V_{DD} - 4V_T$

Low Range: $V_{OUT} = V_{DD} - 2V_T$

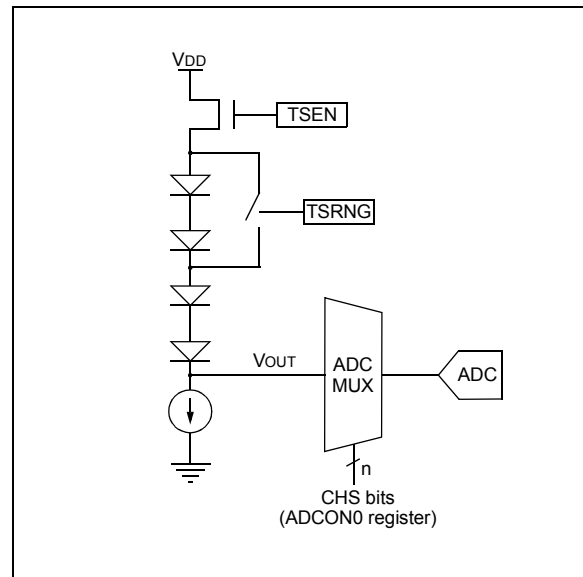
The temperature sense circuit is integrated with the Fixed Voltage Reference (FVR) module. See **Section 14.0 "Fixed Voltage Reference (FVR)"** for more information.

The circuit is enabled by setting the TSEN bit of the FVRCON register. When disabled, the circuit draws no current.

The circuit operates in either high or low range. The high range, selected by setting the TSRNG bit of the FVRCON register, provides a wider output voltage. This provides more resolution over the temperature range, but may be less consistent from part to part. This range requires a higher bias voltage to operate and thus, a higher V_{DD} is needed.

The low range is selected by clearing the TSRNG bit of the FVRCON register. The low range generates a lower voltage drop and thus, a lower bias voltage is needed to operate the circuit. The low range is provided for low voltage operation.

FIGURE 15-1: TEMPERATURE CIRCUIT DIAGRAM



15.2 Minimum Operating V_{DD} vs. Minimum Sensing Temperature

When the temperature circuit is operated in low range, the device may be operated at any operating voltage that is within specifications.

When the temperature circuit is operated in high range, the device operating voltage, V_{DD} , must be high enough to ensure that the temperature circuit is correctly biased.

Table 15-1 shows the recommended minimum V_{DD} vs. range setting.

TABLE 15-1: RECOMMENDED V_{DD} VS. RANGE

Min. V_{DD} , TSRNG = 1	Min. V_{DD} , TSRNG = 0
3.6V	1.8V

15.3 Temperature Output

The output of the circuit is measured using the internal Analog-to-Digital converter. A channel is reserved for the temperature circuit output. Refer to **Section 16.0 "Analog-to-Digital Converter (ADC) Module"** for detailed information.

15.4 ADC Acquisition Time

To ensure accurate temperature measurements, the user must wait at least 200 μ s after the ADC input multiplexer is connected to the temperature indicator output before the conversion is performed. In addition, the user must wait 200 μ s between sequential conversions of the temperature indicator output.

PIC16(L)F1825/9

TABLE 16-3: SUMMARY OF REGISTERS ASSOCIATED WITH ADC

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ADCON0	—	CHS<4:0>					GO/DONE	ADON	150
ADCON1	ADFM	ADCS<2:0>			—	ADNREF	ADPREF<1:0>		151
ADRESH	A/D Result Register High								152, 153
ADRESL	A/D Result Register Low								152, 153
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	123
ANSELB ⁽¹⁾	—	—	ANSB5	ANSB4	—	—	—	—	129
ANSELC	ANSC7 ⁽¹⁾	ANSC6 ⁽¹⁾	—	—	ANSC3	ANSC2	ANSC1	ANSC0	134
CCP4CON	P4M<1:0>		DC4B<1:0>		CCP4M<3:0>				224
INLVLA	—	—	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0	124
INVLVB ⁽¹⁾	INVLVB7	INVLVB6	INVLVB5	INVLVB4	—	—	—	—	129
INLVLC	INLVLC7 ⁽¹⁾	INLVLC6 ⁽¹⁾	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0	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
TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	122
TRISB ⁽¹⁾	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	128
TRISC	TRISC7 ⁽¹⁾	TRISC6 ⁽¹⁾	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	133
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		142
DACCON0	DACEN	DACLPS	DACOE	—	DACPSS<1:0>		—	DACNSS	160
DACCON1	—	—	—	DACR<4:0>					160

Legend: x = unknown, u = unchanged, — = unimplemented read as '0', q = value depends on condition. Shaded cells are not used for ADC module.

Note 1: PIC16(L)F1829 only.

REGISTER 24-2: CCPTMRS: PWM TIMER SELECTION CONTROL REGISTER 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	R/W-0/0
C4TSEL<1:0>		C3TSEL<1:0>		C2TSEL<1:0>		C1TSEL<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 **C4TSEL<1:0>**: CCP4 Timer Selection bits
00 = CCP4 is based off Timer2 in PWM mode
01 = CCP4 is based off Timer4 in PWM mode
10 = CCP4 is based off Timer6 in PWM mode
11 = Reserved
- bit 5-4 **C3TSEL<1:0>**: CCP3 Timer Selection bits
00 = CCP3 is based off Timer2 in PWM mode
01 = CCP3 is based off Timer4 in PWM mode
10 = CCP3 is based off Timer6 in PWM mode
11 = Reserved
- bit 3-2 **C2TSEL<1:0>**: CCP2 Timer Selection bits
00 = CCP2 is based off Timer2 in PWM mode
01 = CCP2 is based off Timer4 in PWM mode
10 = CCP2 is based off Timer6 in PWM mode
11 = Reserved
- bit 1-0 **C1TSEL<1:0>**: CCP1 Timer Selection bits
00 = CCP1 is based off Timer2 in PWM mode
01 = CCP1 is based off Timer4 in PWM mode
10 = CCP1 is based off Timer6 in PWM mode
11 = Reserved

REGISTER 24-4: PWMxCON: ENHANCED PWM 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/W-0/0
PxRSEN	PxDC<6:0>						
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 **PxRSEN:** PWM Restart Enable bit
1 = Upon auto-shutdown, the CCPxASE bit clears automatically once the shutdown event goes away; the PWM restarts automatically
0 = Upon auto-shutdown, CCPxASE must be cleared in software to restart the PWM
- bit 6-0 **PxDC<6:0>:** PWM Delay Count bits
PxDCx = Number of Fosc/4 (4 * Tosc) cycles between the scheduled time when a PWM signal **should** transition active and the **actual** time it transitions active

Note 1: Bit resets to '0' with Two-Speed Start-up and LP, XT or HS selected as the Oscillator mode or Fail-Safe mode is enabled.

25.5.2 SLAVE TRANSMISSION

When the $\overline{R/W}$ bit of the incoming address byte is set and an address match occurs, the $\overline{R/W}$ bit of the SSPxSTAT register is set. The received address is loaded into the SSPxBUF register, and an $\overline{ACK}$ pulse is sent by the slave on the ninth bit.

Following the $\overline{ACK}$, slave hardware clears the CKP bit and the SCLx pin is held low (see **Section 25.5.5 “Clock Stretching”** for more detail). By stretching the clock, the master will be unable to assert another clock pulse until the slave is done preparing the transmit data.

The transmit data must be loaded into the SSPxBUF register which also loads the SSPxSR register. Then the SCLx pin should be released by setting the CKP bit of the SSPxCON1 register. The eight data bits are shifted out on the falling edge of the SCLx input. This ensures that the SDAx signal is valid during the SCLx high time.

The $\overline{ACK}$ pulse from the master-receiver is latched on the rising edge of the ninth SCLx input pulse. This $\overline{ACK}$ value is copied to the ACKSTAT bit of the SSPxCON2 register. If ACKSTAT is set (not $\overline{ACK}$), then the data transfer is complete. In this case, when the not $\overline{ACK}$ is latched by the slave, the slave goes Idle and waits for another occurrence of the Start bit. If the SDAx line was low ($\overline{ACK}$), the next transmit data must be loaded into the SSPxBUF register. Again, the SCLx pin must be released by setting bit CKP.

An MSSPx interrupt is generated for each data transfer byte. The SSPxIF bit must be cleared by software and the SSPxSTAT register is used to determine the status of the byte. The SSPxIF bit is set on the falling edge of the ninth clock pulse.

25.5.2.1 Slave Mode Bus Collision

A slave receives a Read request and begins shifting data out on the SDAx line. If a bus collision is detected and the SBCDE bit of the SSPxCON3 register is set, the BCLxIF bit of the PIRx register is set. Once a bus collision is detected, the slave goes Idle and waits to be addressed again. User software can use the BCLxIF bit to handle a slave bus collision.

25.5.2.2 7-bit Transmission

A master device can transmit a read request to a slave, and then clock data out of the slave. The list below outlines what software for a slave will need to do to accomplish a standard transmission. Figure 25-18 can be used as a reference to this list.

1. Master sends a Start condition on SDAx and SCLx.
2. S bit of SSPxSTAT is set; SSPxIF is set if interrupt on Start detect is enabled.
3. Matching address with $\overline{R/W}$ bit set is received by the Slave setting SSPxIF bit.
4. Slave hardware generates an $\overline{ACK}$ and sets SSPxIF.
5. SSPxIF bit is cleared by user.
6. Software reads the received address from SSPxBUF, clearing BF.
7. $\overline{R/W}$ is set so CKP was automatically cleared after the $\overline{ACK}$.
8. The slave software loads the transmit data into SSPxBUF.
9. CKP bit is set releasing SCLx, allowing the master to clock the data out of the slave.
10. SSPxIF is set after the $\overline{ACK}$ response from the master is loaded into the ACKSTAT register.
11. SSPxIF bit is cleared.
12. The slave software checks the ACKSTAT bit to see if the master wants to clock out more data.

Note 1: If the master $\overline{ACK}$ s the clock will be stretched.

2: ACKSTAT is the only bit updated on the rising edge of SCLx (9th) rather than the falling.

13. Steps 9-13 are repeated for each transmitted byte.
14. If the master sends a not $\overline{ACK}$; the clock is not held, but SSPxIF is still set.
15. The master sends a Restart condition or a Stop.
16. The slave is no longer addressed.

PIC16(L)F1825/9

25.5.5 CLOCK STRETCHING

Clock stretching occurs when a device on the bus holds the SCLx line low effectively pausing communication. The slave may stretch the clock to allow more time to handle data or prepare a response for the master device. A master device is not concerned with stretching as anytime it is active on the bus and not transferring data it is stretching. Any stretching done by a slave is invisible to the master software and handled by the hardware that generates SCLx.

The CKP bit of the SSPxCON1 register is used to control stretching in software. Any time the CKP bit is cleared, the module will wait for the SCLx line to go low and then hold it. Setting CKP will release SCLx and allow more communication.

25.5.5.1 Normal Clock Stretching

Following an $\overline{\text{ACK}}$ if the R/W bit of SSPxSTAT is set, a read request, the slave hardware will clear CKP. This allows the slave time to update SSPxBUF with data to transfer to the master. If the SEN bit of SSPxCON2 is set, the slave hardware will always stretch the clock after the ACK sequence. Once the slave is ready, CKP is set by software and communication resumes.

Note 1: The BF bit has no effect on if the clock will be stretched or not. This is different than previous versions of the module that would not stretch the clock, clear CKP, if SSPxBUF was read before the ninth falling edge of SCLx.

2: Previous versions of the module did not stretch the clock for a transmission if SSPxBUF was loaded before the ninth falling edge of SCLx. It is now always cleared for read requests.

25.5.5.2 10-bit Addressing Mode

In 10-bit Addressing mode, when the UA bit is set, the clock is always stretched. This is the only time the SCLx is stretched without CKP being cleared. SCLx is released immediately after a write to SSPxADD.

Note: Previous versions of the module did not stretch the clock if the second address byte did not match.

25.5.5.3 Byte NACKing

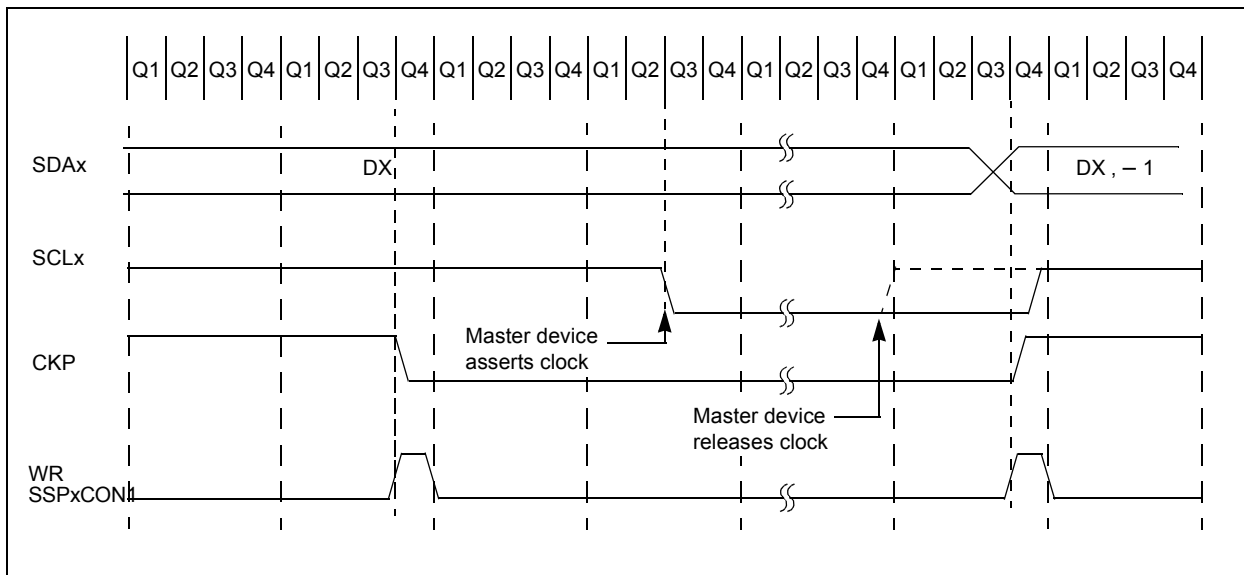
When AHEN bit of SSPxCON3 is set; CKP is cleared by hardware after the eighth falling edge of SCLx for a received matching address byte. When DHEN bit of SSPxCON3 is set; CKP is cleared after the eighth falling edge of SCLx for received data.

Stretching after the eighth falling edge of SCLx allows the slave to look at the received address or data and decide if it wants to ACK the received data.

25.5.6 CLOCK SYNCHRONIZATION AND THE CKP BIT

Any time the CKP bit is cleared, the module will wait for the SCLx line to go low and then hold it. However, clearing the CKP bit will not assert the SCLx output low until the SCLx output is already sampled low. Therefore, the CKP bit will not assert the SCLx line until an external I²C master device has already asserted the SCLx line. The SCLx output will remain low until the CKP bit is set and all other devices on the I²C bus have released SCLx. This ensures that a write to the CKP bit will not violate the minimum high time requirement for SCLx (see Figure 25-23).

FIGURE 25-23: CLOCK SYNCHRONIZATION TIMING



25.5.7 GENERAL CALL ADDRESS SUPPORT

The addressing procedure for the I²C bus is such that the first byte after the Start condition usually determines which device will be the slave addressed by the master device. The exception is the general call address which can address all devices. When this address is used, all devices should, in theory, respond with an acknowledge.

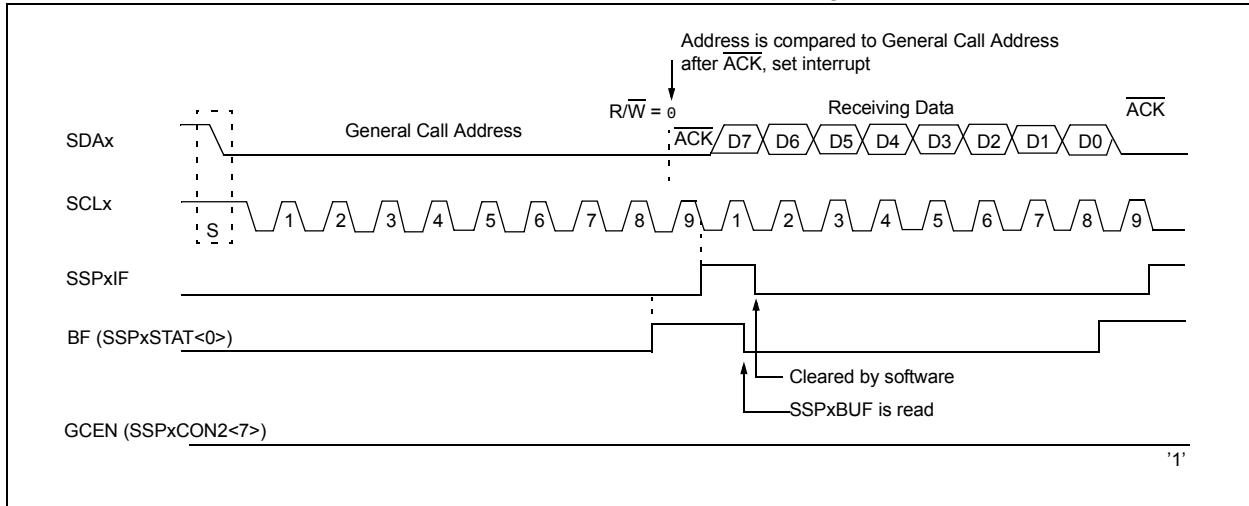
The general call address is a reserved address in the I²C protocol, defined as address 0x00. When the GCEN bit of the SSPxCON2 register is set, the slave module will automatically $\overline{\text{ACK}}$ the reception of this address regardless of the value stored in SSPxADD. After the slave clocks in an address of all zeros with the R/W bit clear, an interrupt is generated and slave

software can read SSPxBUF and respond. Figure 25-24 shows a general call reception sequence.

In 10-bit Address mode, the UA bit will not be set on the reception of the general call address. The slave will prepare to receive the second byte as data, just as it would in 7-bit mode.

If the AHEN bit of the SSPxCON3 register is set, just as with any other address reception, the slave hardware will stretch the clock after the eighth falling edge of SCLx. The slave must then set its ACKDT value and release the clock with communication progressing as it would normally.

FIGURE 25-24: SLAVE MODE GENERAL CALL ADDRESS SEQUENCE



25.5.8 SSPX MASK REGISTER

An SSPx Mask (SSPMSK) register (Register 25-5) is available in I²C Slave mode as a mask for the value held in the SSPxSR register during an address comparison operation. A zero ('0') bit in the SSPMSK register has the effect of making the corresponding bit of the received address a "don't care".

This register is reset to all '1's upon any Reset condition and, therefore, has no effect on standard SSPx operation until written with a mask value.

The SSPx Mask register is active during:

- 7-bit Address mode: address compare of A<7:1>.
- 10-bit Address mode: address compare of A<7:0> only. The SSPx mask has no effect during the reception of the first (high) byte of the address.

26.1 EUSART Asynchronous Mode

The EUSART transmits and receives data using the standard non-return-to-zero (NRZ) format. NRZ is implemented with two levels: a V_{OH} mark state which represents a '1' data bit, and a V_{OL} space state which represents a '0' data bit. NRZ refers to the fact that consecutively transmitted data bits of the same value stay at the output level of that bit without returning to a neutral level between each bit transmission. An NRZ transmission port idles in the mark state. Each character transmission consists of one Start bit followed by eight or nine data bits and is always terminated by one or more Stop bits. The Start bit is always a space and the Stop bits are always marks. The most common data format is eight bits. Each transmitted bit persists for a period of 1/(Baud Rate). An on-chip dedicated 8-bit/16-bit Baud Rate Generator is used to derive standard baud rate frequencies from the system oscillator. See Table 26-5 for examples of baud rate configurations.

The EUSART transmits and receives the LSb first. The EUSART's transmitter and receiver are functionally independent, but share the same data format and baud rate. Parity is not supported by the hardware, but can be implemented in software and stored as the ninth data bit.

26.1.1 EUSART ASYNCHRONOUS TRANSMITTER

The EUSART transmitter block diagram is shown in Figure 26-1. The heart of the transmitter is the serial Transmit Shift Register (TSR), which is not directly accessible by software. The TSR obtains its data from the transmit buffer, which is the TXREG register.

26.1.1.1 Enabling the Transmitter

The EUSART transmitter is enabled for asynchronous operations by configuring the following three control bits:

- TXEN = 1
- SYNC = 0
- SPEN = 1

All other EUSART control bits are assumed to be in their default state.

Setting the TXEN bit of the TXSTA register enables the transmitter circuitry of the EUSART. Clearing the SYNC bit of the TXSTA register configures the EUSART for asynchronous operation. Setting the SPEN bit of the RCSTA register enables the EUSART and automatically configures the TX/CK I/O pin as an output. If the TX/CK pin is shared with an analog peripheral, the analog I/O function must be disabled by clearing the corresponding ANSEL bit.

Note 1: The TXIF Transmitter Interrupt flag is set when the TXEN enable bit is set.

26.1.1.2 Transmitting Data

A transmission is initiated by writing a character to the TXREG register. If this is the first character, or the previous character has been completely flushed from the TSR, the data in the TXREG is immediately transferred to the TSR register. If the TSR still contains all or part of a previous character, the new character data is held in the TXREG until the Stop bit of the previous character has been transmitted. The pending character in the TXREG is then transferred to the TSR in one T_{cy} immediately following the Stop bit transmission. The transmission of the Start bit, data bits and Stop bit sequence commences immediately following the transfer of the data to the TSR from the TXREG.

26.1.1.3 Transmit Data Polarity

The polarity of the transmit data can be controlled with the SCKP bit of the BAUDCON register. The default state of this bit is '0' which selects high true transmit idle and data bits. Setting the SCKP bit to '1' will invert the transmit data resulting in low true idle and data bits. The SCKP bit controls transmit data polarity in Asynchronous mode only. In Synchronous mode, the SCKP bit has a different function. See **Section 26.4.1.2 "Clock Polarity"**.

26.1.1.4 Transmit Interrupt Flag

The TXIF interrupt flag bit of the PIR1 register is set whenever the EUSART transmitter is enabled and no character is being held for transmission in the TXREG. In other words, the TXIF bit is only clear when the TSR is busy with a character and a new character has been queued for transmission in the TXREG. The TXIF flag bit is not cleared immediately upon writing TXREG. TXIF becomes valid in the second instruction cycle following the write execution. Polling TXIF immediately following the TXREG write will return invalid results. The TXIF bit is read-only, it cannot be set or cleared by software.

The TXIF interrupt can be enabled by setting the TXIE interrupt enable bit of the PIE1 register. However, the TXIF flag bit will be set whenever the TXREG is empty, regardless of the state of TXIE enable bit.

To use interrupts when transmitting data, set the TXIE bit only when there is more data to send. Clear the TXIE interrupt enable bit upon writing the last character of the transmission to the TXREG.

26.1.2.4 Receive Framing Error

Each character in the receive FIFO buffer has a corresponding framing error Status bit. A framing error indicates that a Stop bit was not seen at the expected time. The framing error status is accessed via the FERR bit of the RCSTA register. The FERR bit represents the status of the top unread character in the receive FIFO. Therefore, the FERR bit must be read before reading the RCREG.

The FERR bit is read-only and only applies to the top unread character in the receive FIFO. A framing error (FERR = 1) does not preclude reception of additional characters. It is not necessary to clear the FERR bit. Reading the next character from the FIFO buffer will advance the FIFO to the next character and the next corresponding framing error.

The FERR bit can be forced clear by clearing the SPEN bit of the RCSTA register which resets the EUSART. Clearing the CREN bit of the RCSTA register does not affect the FERR bit. A framing error by itself does not generate an interrupt.

Note: If all receive characters in the receive FIFO have framing errors, repeated reads of the RCREG will not clear the FERR bit.

26.1.2.5 Receive Overrun Error

The receive FIFO buffer can hold two characters. An overrun error will be generated if a third character, in its entirety, is received before the FIFO is accessed. When this happens the OERR bit of the RCSTA register is set. The characters already in the FIFO buffer can be read but no additional characters will be received until the error is cleared. The error must be cleared by either clearing the CREN bit of the RCSTA register or by resetting the EUSART by clearing the SPEN bit of the RCSTA register.

26.1.2.6 Receiving 9-bit Characters

The EUSART supports 9-bit character reception. When the RX9 bit of the RCSTA register is set the EUSART will shift nine bits into the RSR for each character received. The RX9D bit of the RCSTA register is the ninth and Most Significant data bit of the top unread character in the receive FIFO. When reading 9-bit data from the receive FIFO buffer, the RX9D data bit must be read before reading the eight Least Significant bits from the RCREG.

26.1.2.7 Address Detection

A special Address Detection mode is available for use when multiple receivers share the same transmission line, such as in RS-485 systems. Address detection is enabled by setting the ADDEN bit of the RCSTA register.

Address detection requires 9-bit character reception. When address detection is enabled, only characters with the ninth data bit set will be transferred to the receive FIFO buffer, thereby setting the RCIF interrupt bit. All other characters will be ignored.

Upon receiving an address character, user software determines if the address matches its own. Upon address match, user software must disable address detection by clearing the ADDEN bit before the next Stop bit occurs. When user software detects the end of the message, determined by the message protocol used, software places the receiver back into the Address Detection mode by setting the ADDEN bit.

26.3.2 AUTO-BAUD OVERFLOW

During the course of automatic baud detection, the ABDOVF bit of the BAUDCON register will be set if the baud rate counter overflows before the fifth rising edge is detected on the RX pin. The ABDOVF bit indicates that the counter has exceeded the maximum count that can fit in the 16 bits of the SPBRGH:SPBRGL register pair. After the ABDOVF has been set, the counter continues to count until the fifth rising edge is detected on the RX pin. Upon detecting the fifth RX edge, the hardware will set the RCIF interrupt flag and clear the ABDEN bit of the BAUDCON register. The RCIF flag can be subsequently cleared by reading the RCREG register. The ABDOVF flag of the BAUDCON register can be cleared by software directly.

To terminate the auto-baud process before the RCIF flag is set, clear the ABDEN bit then clear the ABDOVF bit of the BAUDCON register. The ABDOVF bit will remain set if the ABDEN bit is not cleared first.

26.3.3 AUTO-WAKE-UP ON BREAK

During Sleep mode, all clocks to the EUSART are suspended. Because of this, the Baud Rate Generator is inactive and a proper character reception cannot be performed. The Auto-Wake-up feature allows the controller to wake-up due to activity on the RX/DT line. This feature is available only in Asynchronous mode.

The Auto-Wake-up feature is enabled by setting the WUE bit of the BAUDCON register. Once set, the normal receive sequence on RX/DT is disabled, and the EUSART remains in an Idle state, monitoring for a wake-up event independent of the CPU mode. A wake-up event consists of a high-to-low transition on the RX/DT line. (This coincides with the start of a Sync Break or a wake-up signal character for the LIN protocol.)

The EUSART module generates an RCIF interrupt coincident with the wake-up event. The interrupt is generated synchronously to the Q clocks in normal CPU operating modes (Figure 26-7), and asynchronously if the device is in Sleep mode (Figure 26-8). The interrupt condition is cleared by reading the RCREG register.

The WUE bit is automatically cleared by the low-to-high transition on the RX line at the end of the Break. This signals to the user that the Break event is over. At this point, the EUSART module is in Idle mode waiting to receive the next character.

26.3.3.1 Special Considerations

Break Character

To avoid character errors or character fragments during a wake-up event, the wake-up character must be all zeros.

When the wake-up is enabled the function works independent of the low time on the data stream. If the WUE bit is set and a valid non-zero character is received, the low time from the Start bit to the first rising edge will be interpreted as the wake-up event. The remaining bits in the character will be received as a fragmented character and subsequent characters can result in framing or overrun errors.

Therefore, the initial character in the transmission must be all '0's. This must be 10 or more bit times, 13-bit times recommended for LIN bus, or any number of bit times for standard RS-232 devices.

Oscillator Start-up Time

Oscillator start-up time must be considered, especially in applications using oscillators with longer start-up intervals (i.e., LP, XT or HS/PLL mode). The Sync Break (or wake-up signal) character must be of sufficient length, and be followed by a sufficient interval, to allow enough time for the selected oscillator to start and provide proper initialization of the EUSART.

WUE Bit

The wake-up event causes a receive interrupt by setting the RCIF bit. The WUE bit is cleared in hardware by a rising edge on RX/DT. The interrupt condition is then cleared in software by reading the RCREG register and discarding its contents.

To ensure that no actual data is lost, check the RCIDL bit to verify that a receive operation is not in process before setting the WUE bit. If a receive operation is not occurring, the WUE bit may then be set just prior to entering the Sleep mode.

TABLE 30-2: OSCILLATOR 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	Freq. Tolerance	Min.	Typ†	Max.	Units	Conditions
OS08	HFOSC	Internal Calibrated HFINTOSC Frequency ⁽¹⁾	$\pm 2\%$	—	16.0	—	MHz	$0^{\circ}\text{C} \leq T_A \leq +60^{\circ}\text{C}$, $V_{DD} \geq 2.5\text{V}$
			$\pm 3\%$	—	16.0	—	MHz	$60^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$, $V_{DD} \geq 2.5\text{V}$
			$\pm 5\%$	—	16.0	—	MHz	$-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$
OS08A	MFOSC	Internal Calibrated MFINTOSC Frequency ⁽¹⁾	$\pm 2\%$	—	500	—	kHz	$0^{\circ}\text{C} \leq T_A \leq +60^{\circ}\text{C}$, $V_{DD} \geq 2.5\text{V}$
			$\pm 3\%$	—	500	—	kHz	$60^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$, $V_{DD} \geq 2.5\text{V}$
			$\pm 5\%$	—	500	—	kHz	$-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$
OS09	LFOSC	Internal LFINTOSC Frequency	$\pm 25\%$	—	31	—	kHz	$-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$
OS10*	Tiosc ST	HFINTOSC Wake-up from Sleep Start-up Time	—	—	5	8	μs	
		MFINTOSC Wake-up from Sleep Start-up Time	—	—	20	30	μs	

* These parameters are characterized but not tested.

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

Note 1: To ensure these oscillator frequency tolerances, V_{DD} and V_{SS} must be capacitively decoupled as close to the device as possible. 0.1 μF and 0.01 μF values in parallel are recommended.

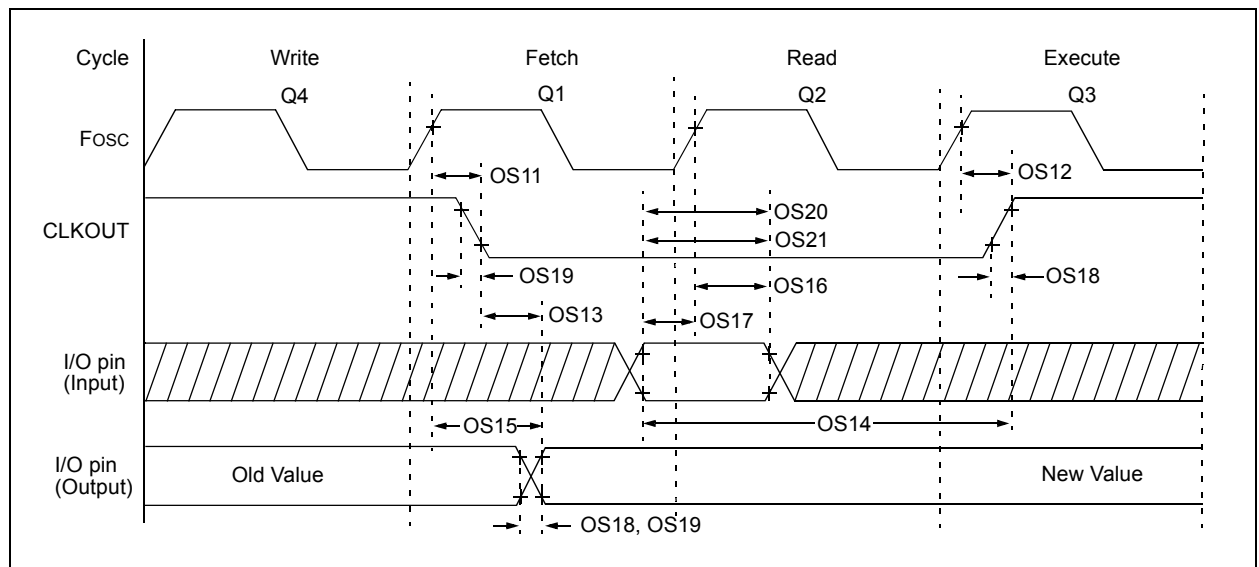
TABLE 30-3: PLL CLOCK TIMING SPECIFICATIONS ($V_{DD} = 2.7\text{V TO } 5.5\text{V}$)

Param No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
F10	FOSC	Oscillator Frequency Range	4	—	8	MHz	
F11	FSYS	On-Chip VCO System Frequency	16	—	32	MHz	
F12	TRC	PLL Start-up Time (Lock Time)	—	—	2	ms	
F13*	ΔCLK	CLKOUT Stability (Jitter)	-0.25%	—	+0.25%	%	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 30-7: CLKOUT AND I/O TIMING



PIC16(L)F1825/9

FIGURE 31-7: I_{DD} TYPICAL, EC OSCILLATOR, MEDIUM-POWER MODE, PIC16LF1825/9 ONLY

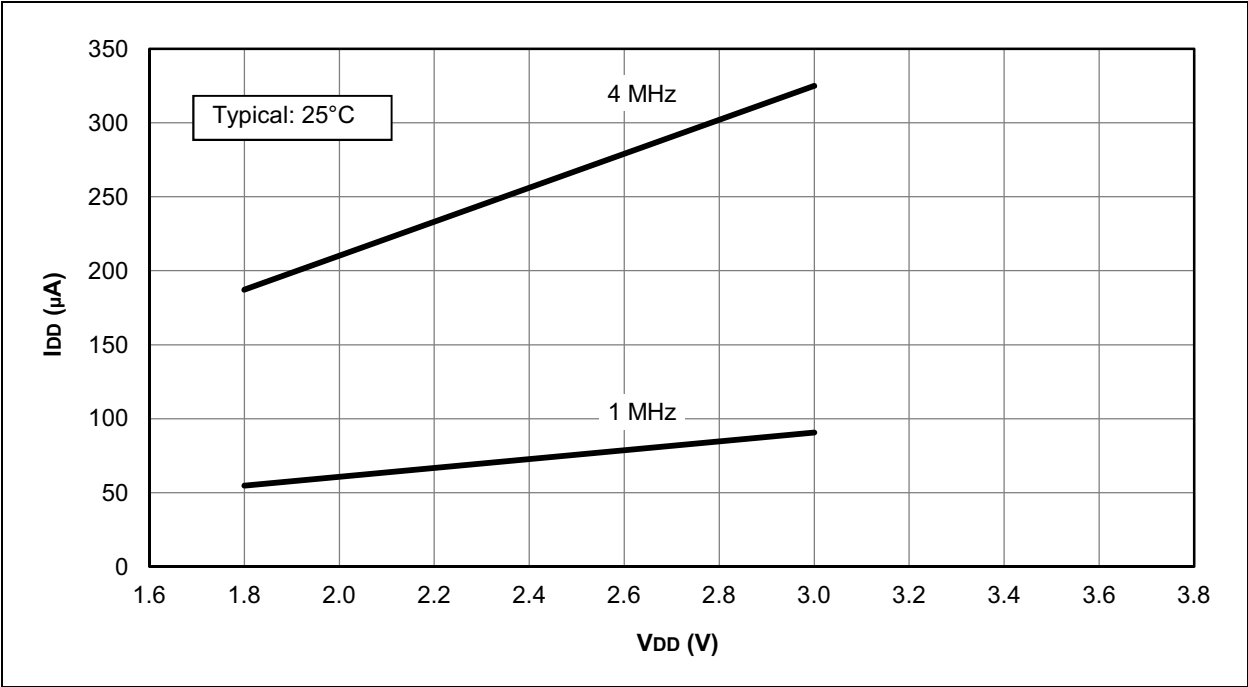
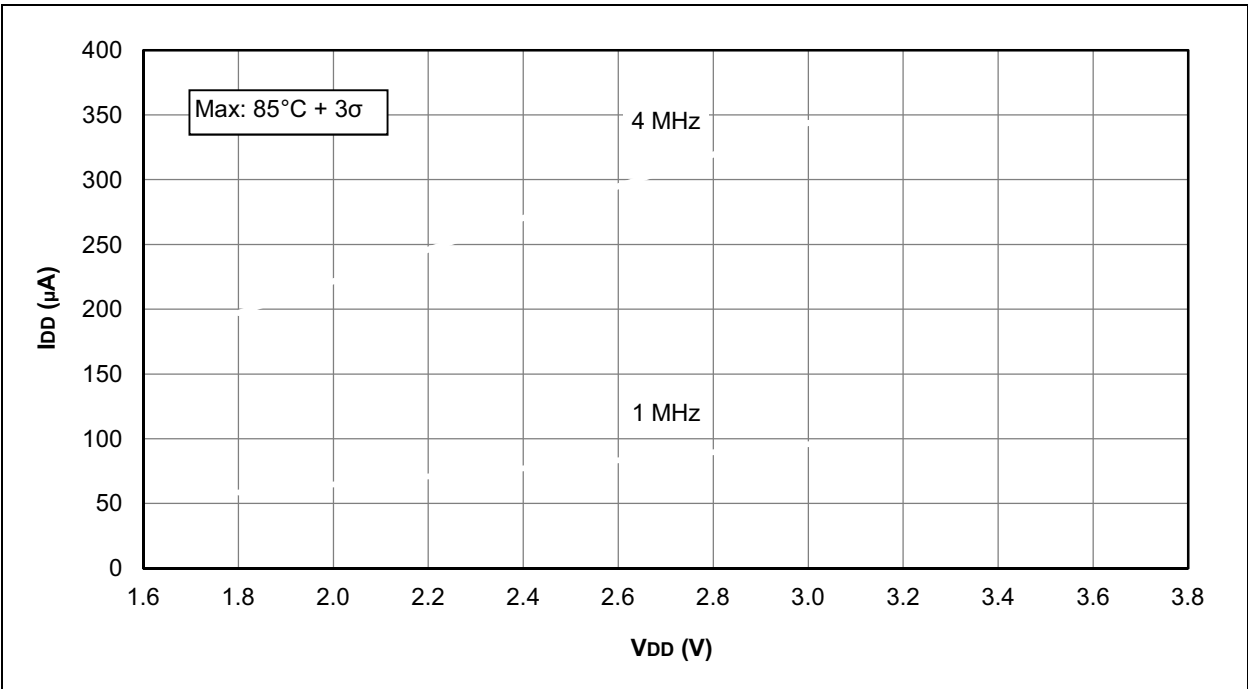


FIGURE 31-8: I_{DD} MAXIMUM, EC OSCILLATOR, MEDIUM-POWER MODE, PIC16LF1825/9 ONLY



PIC16(L)F1825/9

FIGURE 31-54: COMPARATOR HYSTERESIS, NORMAL-POWER MODE (CxSP = 1, CxHYS = 1)

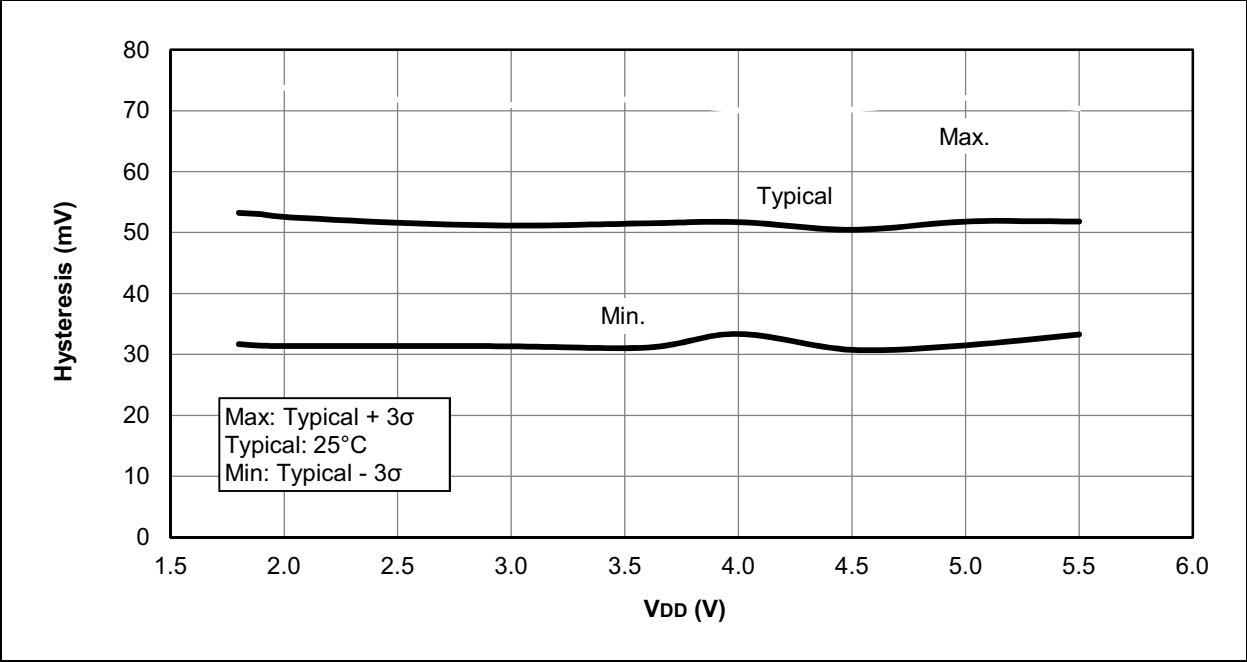
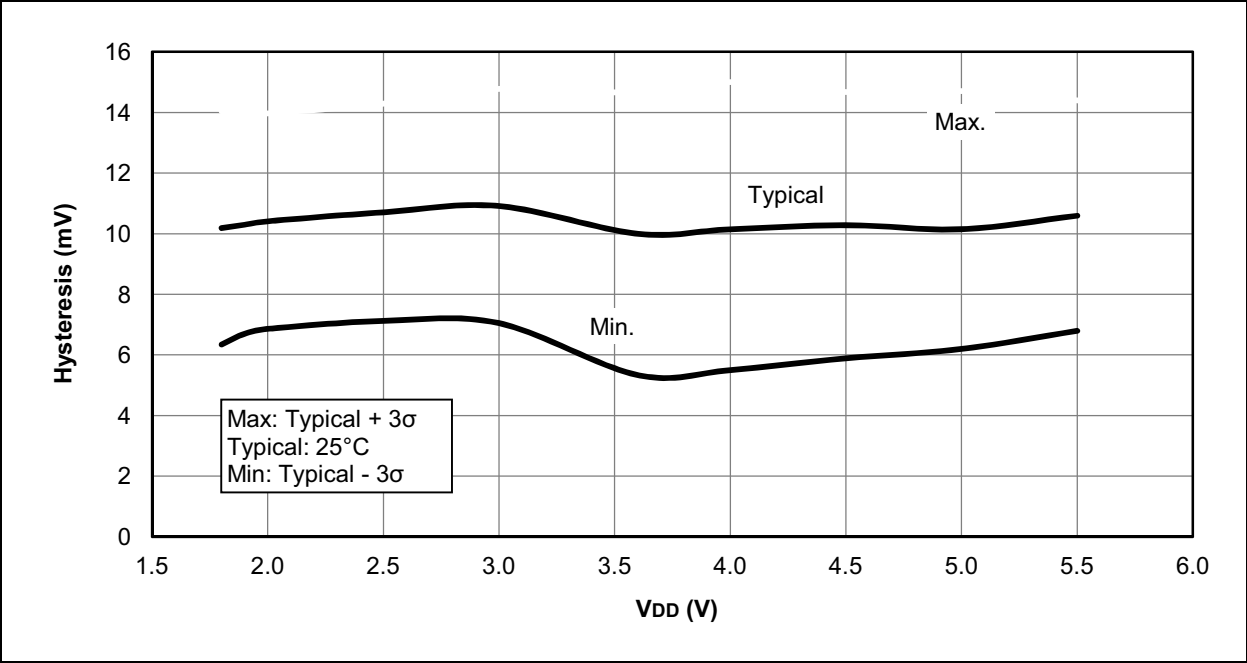


FIGURE 31-55: COMPARATOR HYSTERESIS, LOW-POWER MODE (CxSP = 0, CxHYS = 1)



32.6 MPLAB X SIM Software Simulator

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

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

32.7 MPLAB REAL ICE In-Circuit Emulator System

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

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

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

32.8 MPLAB ICD 3 In-Circuit Debugger System

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

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

32.9 PICkit 3 In-Circuit Debugger/Programmer

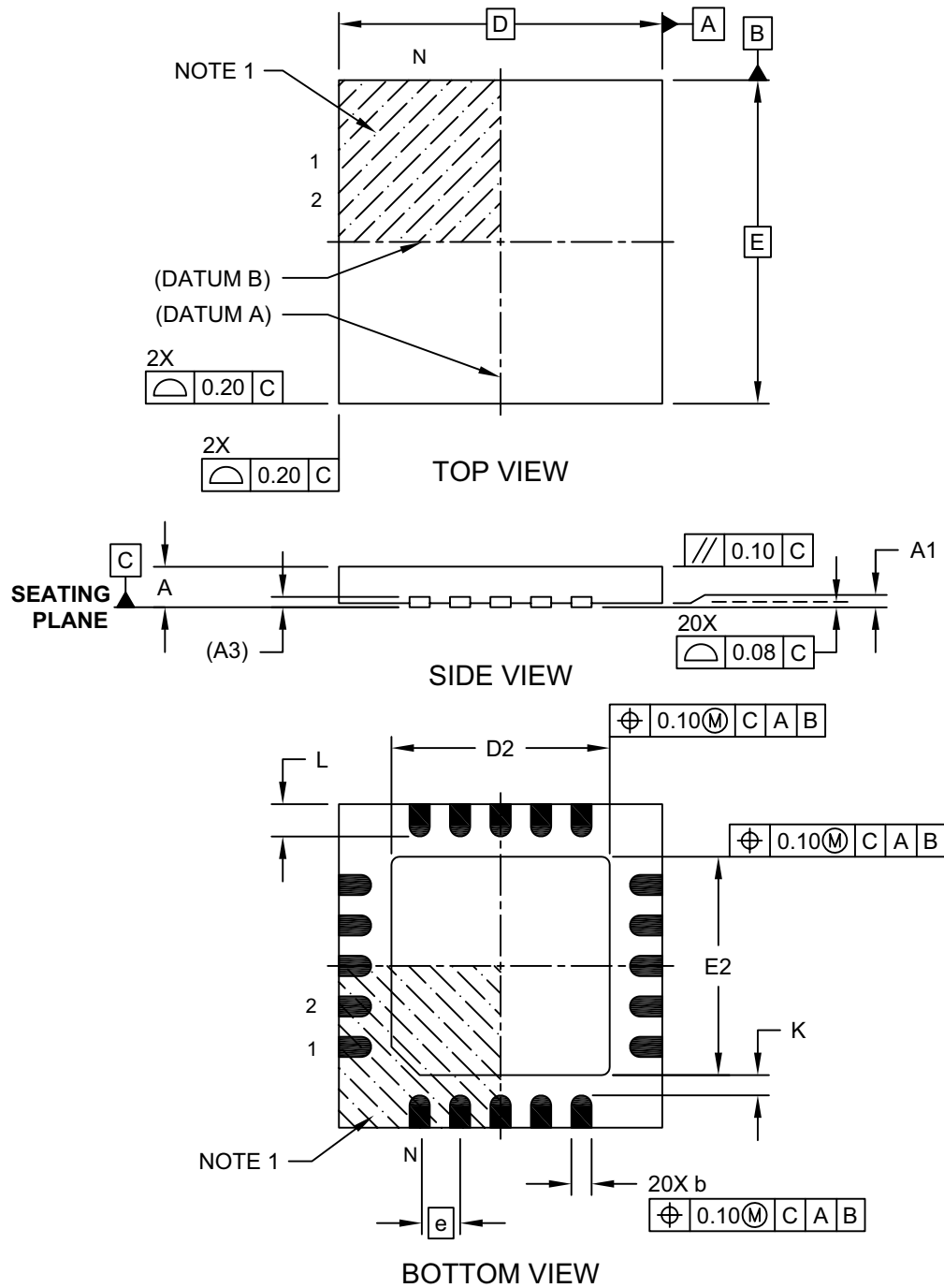
The MPLAB PICkit 3 allows debugging and programming of PIC and dsPIC Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB IDE. The MPLAB PICkit 3 is connected to the design engineer's PC using a full-speed USB interface and can be connected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

32.10 MPLAB PM3 Device Programmer

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

20-Lead Ultra Thin Plastic Quad Flat, No Lead Package (GZ) - 4x4x0.5 mm Body [UQFN]

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



Microchip Technology Drawing C04-255A Sheet 1 of 2