Welcome to [E-XFL.COM](#)

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

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

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

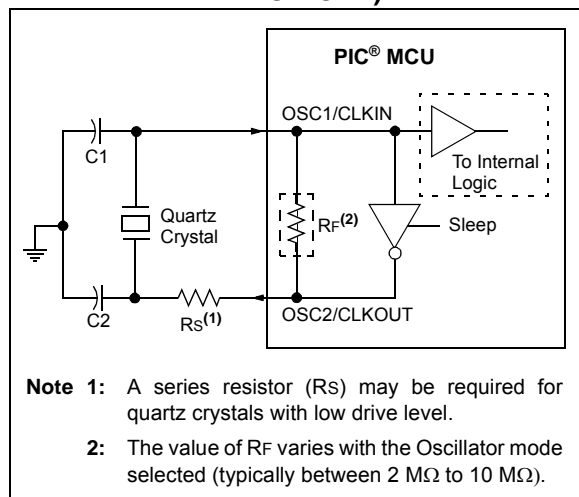| Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	32MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	11
Program Memory Size	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Through Hole
Package / Case	14-DIP (0.300", 7.62mm)
Supplier Device Package	14-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf1825-e-p

TABLE 3-6: PIC16(L)F1825/9 MEMORY MAP, BANKS 24-31

	BANK 24		BANK 25		BANK 26		BANK 27		BANK 28		BANK 29		BANK 30		BANK 31
C00h	INDF0	C80h	INDF0	D00h	INDF0	D80h	INDF0	E00h	INDF0	E80h	INDF0	F00h	INDF0	F80h	INDF0
C01h	INDF1	C81h	INDF1	D01h	INDF1	D81h	INDF1	E01h	INDF1	E81h	INDF1	F01h	INDF1	F81h	INDF1
C02h	PCL	C82h	PCL	D02h	PCL	D82h	PCL	E02h	PCL	E82h	PCL	F02h	PCL	F82h	PCL
C03h	STATUS	C83h	STATUS	D03h	STATUS	D83h	STATUS	E03h	STATUS	E83h	STATUS	F03h	STATUS	F83h	STATUS
C04h	FSR0L	C84h	FSR0L	D04h	FSR0L	D84h	FSR0L	E04h	FSR0L	E84h	FSR0L	F04h	FSR0L	F84h	FSR0L
C05h	FSR0H	C85h	FSR0H	D05h	FSR0H	D85h	FSR0H	E05h	FSR0H	E85h	FSR0H	F05h	FSR0H	F85h	FSR0H
C06h	FSR1L	C86h	FSR1L	D06h	FSR1L	D86h	FSR1L	E06h	FSR1L	E86h	FSR1L	F06h	FSR1L	F86h	FSR1L
C07h	FSR1H	C87h	FSR1H	D07h	FSR1H	D87h	FSR1H	E07h	FSR1H	E87h	FSR1H	F07h	FSR1H	F87h	FSR1H
C08h	BSR	C88h	BSR	D08h	BSR	D88h	BSR	E08h	BSR	E88h	BSR	F08h	BSR	F88h	BSR
C09h	WREG	C89h	WREG	D09h	WREG	D89h	WREG	E09h	WREG	E89h	WREG	F09h	WREG	F89h	WREG
C0Ah	PCLATH	C8Ah	PCLATH	D0Ah	PCLATH	D8Ah	PCLATH	E0Ah	PCLATH	E8Ah	PCLATH	F0Ah	PCLATH	F8Ah	PCLATH
C0Bh	INTCON	C8Bh	INTCON	D0Bh	INTCON	D8Bh	INTCON	E0Bh	INTCON	E8Bh	INTCON	F0Bh	INTCON	F8Bh	INTCON
C0Ch	—	C8Ch	—	D0Ch	—	D8Ch	—	E0Ch	—	E8Ch	—	F0Ch	—	F8Ch	
C0Dh	—	C8Dh	—	D0Dh	—	D8Dh	—	E0Dh	—	E8Dh	—	F0Dh	—	F8Dh	
C0Eh	—	C8Eh	—	D0Eh	—	D8Eh	—	E0Eh	—	E8Eh	—	F0Eh	—	F8Eh	
C0Fh	—	C8Fh	—	D0Fh	—	D8Fh	—	E0Fh	—	E8Fh	—	F0Fh	—	F8Fh	
C10h	—	C90h	—	D10h	—	D90h	—	E10h	—	E90h	—	F10h	—	F90h	
C11h	—	C91h	—	D11h	—	D91h	—	E11h	—	E91h	—	F11h	—	F91h	
C12h	—	C92h	—	D12h	—	D92h	—	E12h	—	E92h	—	F12h	—	F92h	
C13h	—	C93h	—	D13h	—	D93h	—	E13h	—	E93h	—	F13h	—	F93h	
C14h	—	C94h	—	D14h	—	D94h	—	E14h	—	E94h	—	F14h	—	F94h	
C15h	—	C95h	—	D15h	—	D95h	—	E15h	—	E95h	—	F15h	—	F95h	
C16h	—	C96h	—	D16h	—	D96h	—	E16h	—	E96h	—	F16h	—	F96h	
C17h	—	C97h	—	D17h	—	D97h	—	E17h	—	E97h	—	F17h	—	F97h	See Table 3-7 for register mapping details
C18h	—	C98h	—	D18h	—	D98h	—	E18h	—	E98h	—	F18h	—	F98h	
C19h	—	C99h	—	D19h	—	D99h	—	E19h	—	E99h	—	F19h	—	F99h	
C1Ah	—	C9Ah	—	D1Ah	—	D9Ah	—	E1Ah	—	E9Ah	—	F1Ah	—	F9Ah	
C1Bh	—	C9Bh	—	D1Bh	—	D9Bh	—	E1Bh	—	E9Bh	—	F1Bh	—	F9Bh	
C1Ch	—	C9Ch	—	D1Ch	—	D9Ch	—	E1Ch	—	E9Ch	—	F1Ch	—	F9Ch	
C1Dh	—	C9Dh	—	D1Dh	—	D9Dh	—	E1Dh	—	E9Dh	—	F1Dh	—	F9Dh	
C1Eh	—	C9Eh	—	D1Eh	—	D9Eh	—	E1Eh	—	E9Eh	—	F1Eh	—	F9Eh	
C1Fh	—	C9Fh	—	D1Fh	—	D9Fh	—	E1Fh	—	E9Fh	—	F1Fh	—	F9Fh	
C20h	Unimplemented Read as '0'	CA0h	Unimplemented Read as '0'	D20h	Unimplemented Read as '0'	DA0h	Unimplemented Read as '0'	E20h	Unimplemented Read as '0'	EA0h	Unimplemented Read as '0'	F20h	Unimplemented Read as '0'	FA0h	
C6Fh		CEFh		D6Fh		DEFh		E6Fh		EEFh		F6Fh		FEFh	
C70h	Accesses 70h – 7Fh	CF0h	Accesses 70h – 7Fh	D70h	Accesses 70h – 7Fh	DF0h	Accesses 70h – 7Fh	E70h	Accesses 70h – 7Fh	EF0h	Accesses 70h – 7Fh	F70h	Accesses 70h – 7Fh	FF0h	Accesses 70h – 7Fh
CFFh		CFFh		D7Fh		DFFh		E7Fh		EFFh		F7Fh		FFFh	

Legend: ▦ = Unimplemented data memory locations, read as '0'.

PIC16(L)F1825/9

FIGURE 5-3: **QUARTZ CRYSTAL OPERATION (LP, XT OR HS MODE)**

Note 1: A series resistor (Rs) may be required for quartz crystals with low drive level.

2: The value of Rᖴ varies with the Oscillator mode selected (typically between 2 MΩ to 10 MΩ).

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 Vᴅᴅ 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 rfPIC® and PIC® Devices*" (DS00826)

• AN849, "*Basic PIC® Oscillator Design*" (DS00849)

• AN943, "*Practical PIC® Oscillator Analysis and Design*" (DS00943)

• AN949, "*Making Your Oscillator Work*" (DS00949)

FIGURE 5-4: **CERAMIC RESONATOR OPERATION (XT OR HS MODE)**

Note 1: A series resistor (Rs) may be required for ceramic resonators with low drive level.

2: The value of Rᖴ varies with the Oscillator mode selected (typically between 2 MΩ to 10 MΩ).

3: An additional parallel feedback resistor (Rᴘ) may be required for proper ceramic resonator operation.
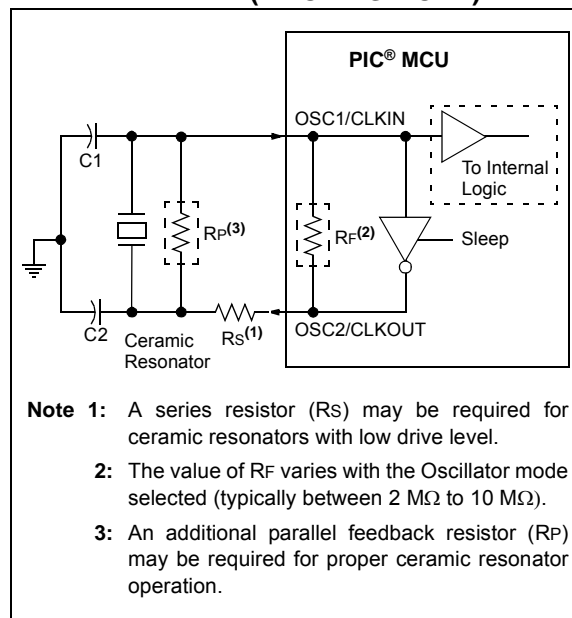
5.2.1.3 Oscillator Start-up Timer (OST)

If the oscillator module is configured for LP, XT or HS modes, the Oscillator Start-up Timer (OST) counts 1024 oscillations from OSC1. This occurs following a Power-on Reset (POR) and when the Power-up Timer (PWRT) has expired (if configured), or a wake-up from Sleep. During this time, the program counter does not increment and program execution is suspended unless either FSCM or Two-Speed Start-up are enabled. In this case, the code will continue to execute at the selected INTOSC frequency while the OST is counting. The OST ensures that the oscillator circuit, using a quartz crystal resonator or ceramic resonator, has started and is providing a stable system clock to the oscillator module.

In order to minimize latency between external oscillator start-up and code execution, the Two-Speed Clock Start-up mode can be selected (see **Section 5.4 "Two-Speed Clock Start-up Mode"**).

FIGURE 7-2: BROWN-OUT SITUATIONS

Note 1: TPWRT delay only if $\overline{\text{PWRTE}}$ bit is programmed to '0'.
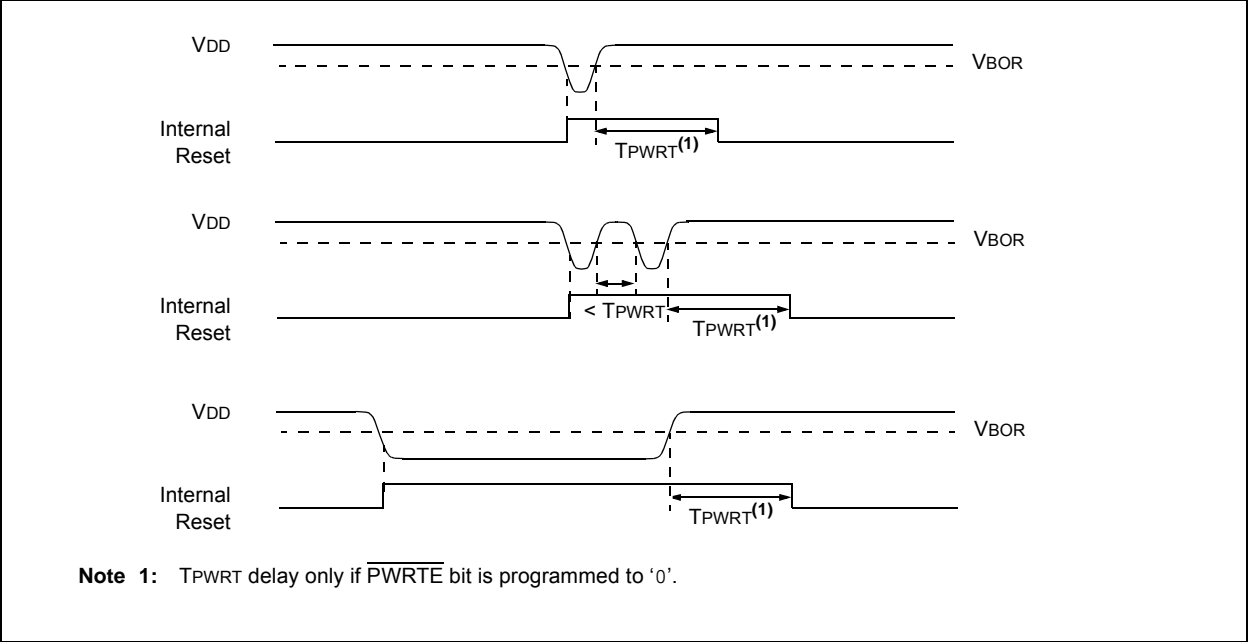
REGISTER 7-1: BORCON: BROWN-OUT RESET CONTROL REGISTER

R/W-1/u	U-0	U-0	U-0	U-0	U-0	U-0	R-q/u
SBOREN	—	—	—	—	—	—	BORRDY
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	q = Value depends on condition

bit 7 **SBOREN:** Software Brown-out Reset Enable bit
<u>If BOREN <1:0> in Configuration Word 1 $\neq$ 01:</u>
SBOREN is read/write, but has no effect on the BOR.
<u>If BOREN <1:0> in Configuration Word 1 = 01:</u>
1 = BOR Enabled
0 = BOR Disabled

bit 6-1 **Unimplemented:** Read as '0'

bit 0 **BORRDY:** Brown-out Reset Circuit Ready Status bit
1 = The Brown-out Reset circuit is active
0 = The Brown-out Reset circuit is inactive

PIC16(L)F1825/9

FIGURE 8-2: **INTERRUPT LATENCY**

© 2010-2015 Microchip Technology Inc.
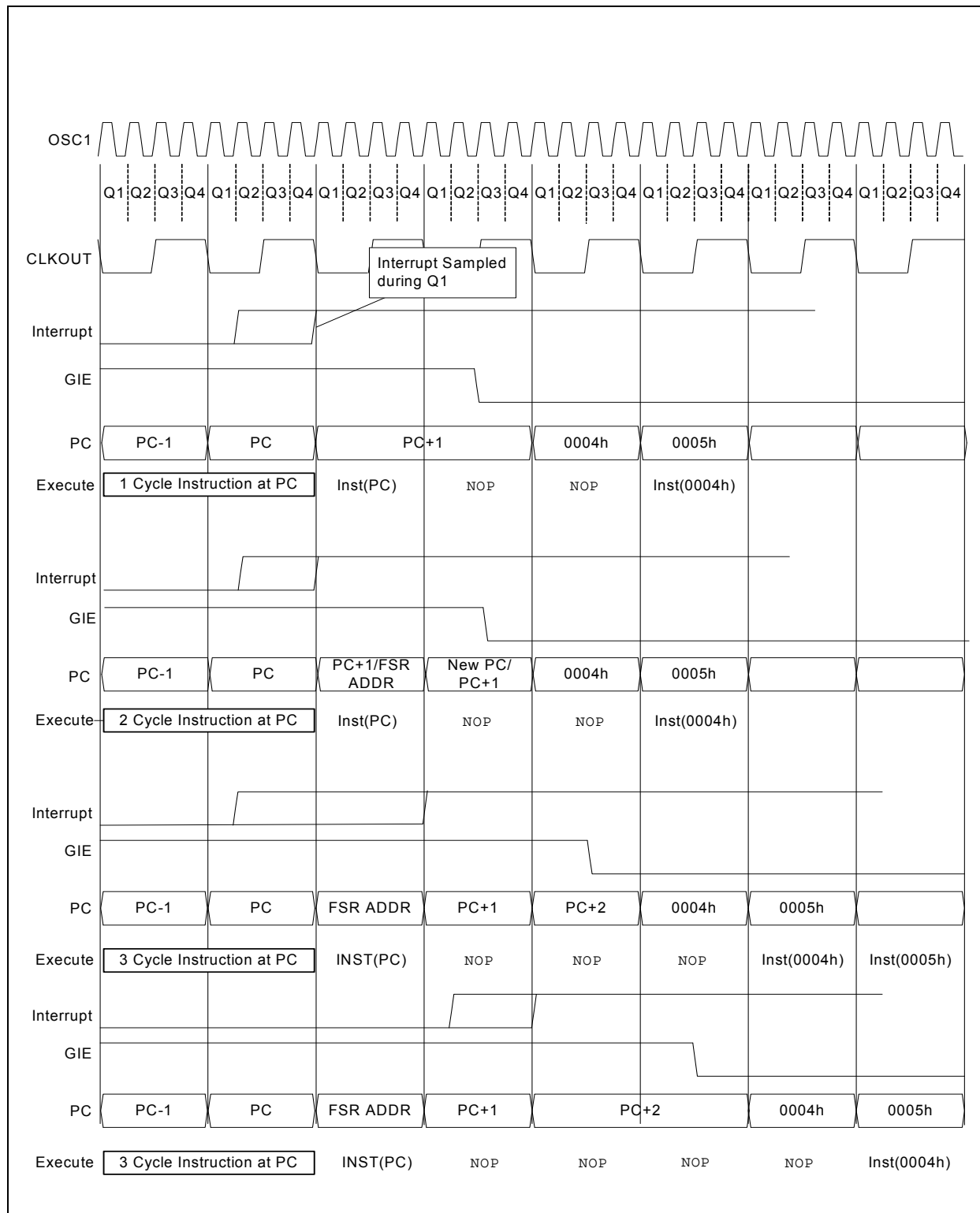
8.6.3 PIE2 REGISTER

The PIE2 register contains the interrupt enable bits, as shown in Register 8-3.

> **Note:** Bit PEIE of the INTCON register must be set to enable any peripheral interrupt.

REGISTER 8-3: PIE2: PERIPHERAL INTERRUPT ENABLE REGISTER 2

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0
OSFIE	C2IE	C1IE	EEIE	BCL1IE	—	—	CCP2IE
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   **OSFIE:** Oscillator Fail Interrupt Enable bit
          1 = Enables the Oscillator Fail interrupt
          0 = Disables the Oscillator Fail interrupt

bit 6   **C2IE:** Comparator C2 Interrupt Enable bit
          1 = Enables the Comparator C2 interrupt
          0 = Disables the Comparator C2 interrupt

bit 5   **C1IE:** Comparator C1 Interrupt Enable bit
          1 = Enables the Comparator C1 interrupt
          0 = Disables the Comparator C1 interrupt

bit 4   **EEIE:** EEPROM Write Completion Interrupt Enable bit
          1 = Enables the EEPROM write completion interrupt
          0 = Disables the EEPROM write completion interrupt

bit 3   **BCL1IE:** MSSP Bus Collision Interrupt Enable bit
          1 = Enables the MSSP bus collision interrupt
          0 = Disables the MSSP bus collision interrupt

bit 2-1   **Unimplemented:** Read as '0'

bit 0   **CCP2IE:** CCP2 Interrupt Enable bit
          1 = Enables the CCP2 interrupt
          0 = Disables the CCP2 interrupt

PIC16(L)F1825/9

11.0 DATA EEPROM AND FLASH PROGRAM MEMORY CONTROL

The data EEPROM and Flash program memory are readable and writable during normal operation (full V_{DD} range). These memories are not directly mapped in the register file space. Instead, they are indirectly addressed through the Special Function Registers (SFRs). There are six SFRs used to access these memories:

- EECON1
- EECON2
- EEDATL
- EEDATH
- EEADRL
- EEADRH

When interfacing the data memory block, EEDATL holds the 8-bit data for read/write, and EEADRL holds the address of the EEDATL location being accessed. These devices have 256 bytes of data EEPROM with an address range from 0h to 0FFh.

When accessing the program memory block, the EEDATH:EEDATL register pair forms a 2-byte word that holds the 14-bit data for read/write, and the EEADRL and EEADRH registers form a 2-byte word that holds the 15-bit address of the program memory location being read.

The EEPROM data memory allows byte read and write. An EEPROM byte write automatically erases the location and writes the new data (erase before write).

The write time is controlled by an on-chip timer. The write/erase voltages are generated by an on-chip charge pump rated to operate over the voltage range of the device for byte or word operations.

Depending on the setting of the Flash Program Memory Self Write Enable bits WRT<1:0> of the Configuration Word 2, the device may or may not be able to write certain blocks of the program memory. However, reads from the program memory are always allowed.

When the device is code-protected, the device programmer can no longer access data or program memory. When code-protected, the CPU may continue to read and write the data EEPROM memory and Flash program memory.

11.1 EEADRL and EEADRH Registers

The EEADRH:EEADRL register pair can address up to a maximum of 256 bytes of data EEPROM or up to a maximum of 32K words of program memory.

When selecting a program address value, the MSB of the address is written to the EEADRH register and the LSB is written to the EEADRL register. When selecting a EEPROM address value, only the LSB of the address is written to the EEADRL register.

11.1.1 EECON1 AND EECON2 REGISTERS

EECON1 is the control register for EE memory accesses.

Control bit EEPGD determines if the access will be a program or data memory access. When clear, any subsequent operations will operate on the EEPROM memory. When set, any subsequent operations will operate on the program memory. On Reset, EEPROM is selected by default.

Control bits RD and WR initiate read and write, respectively. These bits cannot be cleared, only set, in software. They are cleared in hardware at completion of the read or write operation. The inability to clear the WR bit in software prevents the accidental, premature termination of a write operation.

The WREN bit, when set, will allow a write operation to occur. On power-up, the WREN bit is clear. The WRERR bit is set when a write operation is interrupted by a Reset during normal operation. In these situations, following Reset, the user can check the WRERR bit and execute the appropriate error handling routine.

Interrupt flag bit EEIF of the PIR2 register is set when write is complete. It must be cleared in the software.

Reading EECON2 will read all '0's. The EECON2 register is used exclusively in the data EEPROM write sequence. To enable writes, a specific pattern must be written to EECON2.

PIC16(L)F1825/9

EXAMPLE 11-5: WRITING TO FLASH PROGRAM MEMORY

```
; This write routine assumes the following:
; 1. The 16 bytes of data are loaded, starting at the address in DATA_ADDR
; 2. Each word of data to be written is made up of two adjacent bytes in DATA_ADDR,
;    stored in little endian format
; 3. A valid starting address (the least significant bits = 000) is loaded in ADDRH:ADDRL
; 4. ADDRH and ADDRL are located in shared data memory 0x70 - 0x7F
;
        BCF         INTCON,GIE    ; Disable ints so required sequences will execute properly
        BANKSEL     EEADRH        ; Bank 3
        MOVF        ADDRH,W       ; Load initial address
        MOVWF       EEADRH        ;
        MOVF        ADDRL,W       ;
        MOVWF       EEADRL        ;
        MOVLW       LOW DATA_ADDR ; Load initial data address
        MOVWF       FSR0L         ;
        MOVLW       HIGH DATA_ADDR ; Load initial data address
        MOVWF       FSR0H         ;
        BSF         EECON1,EEPGD  ; Point to program memory
        BCF         EECON1,CFGS   ; Not configuration space
        BSF         EECON1,WREN   ; Enable writes
        BSF         EECON1,LWLO   ; Only Load Write Latches
LOOP
        MOVIW       FSR0++        ; Load first data byte into lower
        MOVWF       EEDATL        ;
        MOVIW       FSR0++        ; Load second data byte into upper
        MOVWF       EEDATH        ;

        MOVF        EEADRL,W      ; Check if lower bits of address are '000'
        XORLW       0x07          ; Check if we're on the last of 8 addresses
        ANDLW       0x07          ;
        BTFSC       STATUS,Z      ; Exit if last of eight words,
        GOTO        START_WRITE   ;

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

                                  ; After write processor continues with 3rd instruction.

        INCF        EEADRL,F      ; Still loading latches Increment address
        GOTO        LOOP          ; Write next latches

START_WRITE
        BCF         EECON1,LWLO   ; No more loading latches - Actually start Flash program
                                  ; memory write

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

                                  ; after write processor continues with 3rd instruction
        BCF         EECON1,WREN   ; Disable writes
        BSF         INTCON,GIE    ; Enable interrupts
```

The "Required Sequence" bracket spans the two MOVLW 55h / MOVWF EECON2 / MOVLW 0AAh / MOVWF EECON2 / BSF EECON1,WR / NOP / NOP blocks.

REGISTER 12-9: PORTB: PORTB REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	U-0	U-0	U-0	U-0
RB7	RB6	RB5	RB4	—	—	—	—
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 **RB<7:4>**: PORTB General Purpose I/O Pin bits
1 = Port pin is $\geq$ V$_{IH}$
0 = Port pin is $\leq$ V$_{IL}$

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

REGISTER 12-10: TRISB: PORTB TRI-STATE REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	U-0	U-0	U-0	U-0
TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—
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 **TRISB<7:4>:** PORTB Tri-State Control bits
1 = PORTB pin configured as an input (tri-stated)
0 = PORTB pin configured as an output

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

REGISTER 12-11: LATB: PORTB DATA LATCH REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	U-0	U-0	U-0	U-0
LATB7	LATB6	LATB5	LATB4	—	—	—	—
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 **LATB<7:4>**: PORTB Output Latch Value bits[1]

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

Note 1: Writes to PORTB are actually written to corresponding LATB register. Reads from PORTB register is return of actual I/O pin values.

13.0   INTERRUPT-ON-CHANGE

The PORTA pins can be configured to operate as Interrupt-on-Change (IOC) pins. On the PIC16(L)F1829 devices, the PORTB pins can also be configured to operate as IOC pins. An interrupt can be generated by detecting a signal that has either a rising edge or a falling edge. Any individual port pin, or combination of port pins, can be configured to generate an interrupt. The interrupt-on-change module has the following features:

- Interrupt-on-change enable (Master Switch)
- Individual pin configuration
- Rising and falling edge detection
- Individual pin interrupt flags

Figure 13-1 is a block diagram of the IOC module.

13.1   Enabling the Module

To allow individual port pins to generate an interrupt, the IOCIE bit of the INTCON register must be set. If the IOCIE bit is disabled, the edge detection on the pin will still occur, but an interrupt will not be generated.

13.2   Individual Pin Configuration

For each port pin, a rising edge detector and a falling edge detector are present. To enable a pin to detect a rising edge, the associated bit of the IOCxP register is set. To enable a pin to detect a falling edge, the associated bit of the IOCxN register is set.

A pin can be configured to detect rising and falling edges simultaneously by setting both associated bits of the IOCxP and IOCxN registers, respectively.

13.3   Interrupt Flags

The IOCAFx and IOCBFx bits located in the IOCAF and IOCBF registers, respectively, are status flags that correspond to the interrupt-on-change pins of the associated port. If an expected edge is detected on an appropriately enabled pin, then the status flag for that pin will be set, and an interrupt will be generated if the IOCIE bit is set. The IOCIF bit of the INTCON register reflects the status of all IOCAFx and IOCBFx bits.

13.4   Clearing Interrupt Flags

The individual status flags, (IOCAFx and IOCBFx bits), can be cleared by resetting them to zero. If another edge is detected during this clearing operation, the associated status flag will be set at the end of the sequence, regardless of the value actually being written.

In order to ensure that no detected edge is lost while clearing flags, only AND operations masking out known changed bits should be performed. The following sequence is an example of what should be performed.

EXAMPLE 13-1:   CLEARING INTERRUPT FLAGS (PORTA EXAMPLE)

```
    MOVLW   0xff
    XORWF   IOCAF, W
    ANDWF   IOCAF, F
```

13.5   Operation in Sleep

The interrupt-on-change interrupt sequence will wake the device from Sleep mode, if the IOCIE bit is set.

If an edge is detected while in Sleep mode, the IOCxF register will be updated prior to the first instruction executed out of Sleep.

REGISTER 16-5: **ADRESH: ADC RESULT REGISTER HIGH (ADRESH) ADFM = 1**

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	—	—	—	—	ADRES<9:8>	
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-2   **Reserved**: Do not use.

bit 1-0   **ADRES<9:8>**: ADC Result Register bits
                     Upper two bits of 10-bit conversion result

REGISTER 16-6: **ADRESL: ADC RESULT REGISTER LOW (ADRESL) ADFM = 1**

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
ADRES<7:0>							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

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

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)
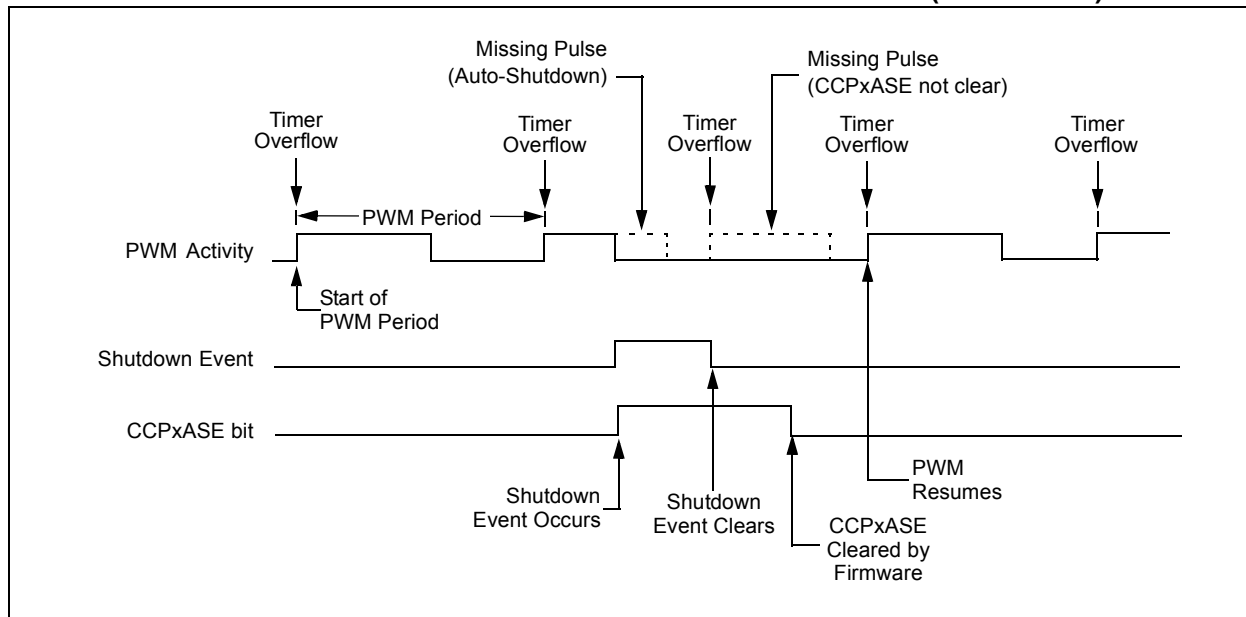
24.4.4 AUTO-RESTART MODE

The Enhanced PWM can be configured to automatically restart the PWM signal once the auto-shutdown condition has been removed. Auto-restart is enabled by setting the PxRSEN bit in the PWMxCON register.

If auto-restart is enabled, the CCPxASE bit will remain set as long as the auto-shutdown condition is active. When the auto-shutdown condition is removed, the CCPxASE bit will be cleared via hardware and normal operation will resume.

FIGURE 24-15: PWM AUTO-SHUTDOWN WITH AUTO-RESTART (PXRSEN = 1)
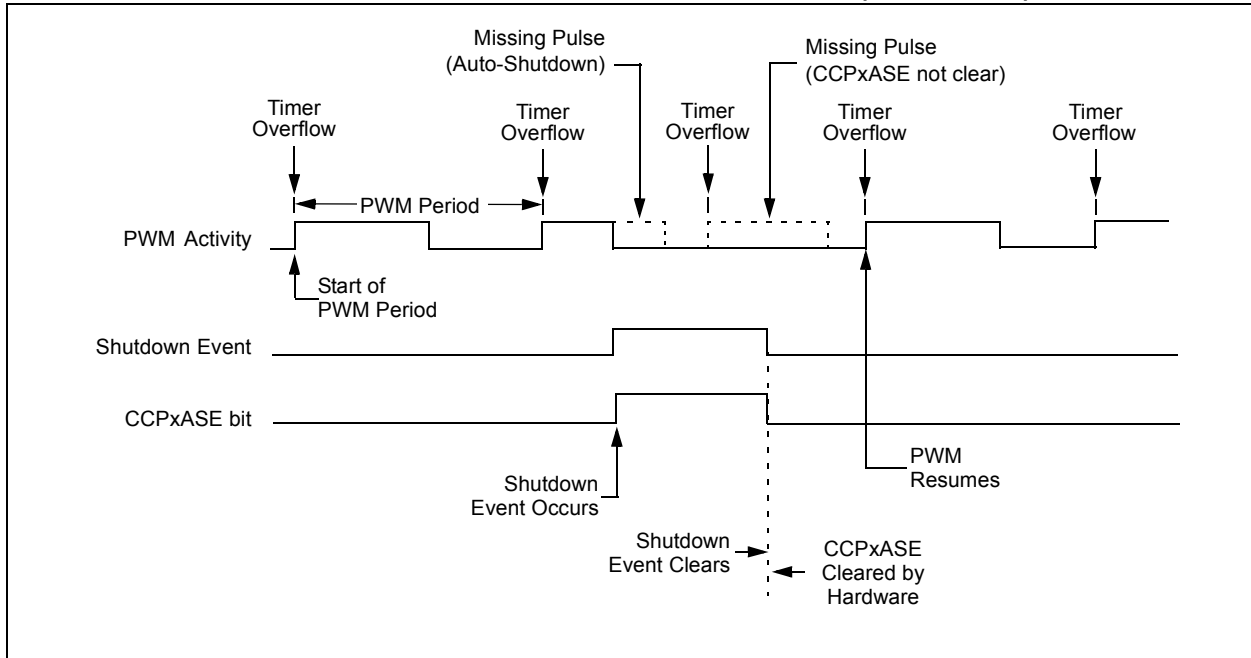
TABLE 24-10: SUMMARY OF REGISTERS ASSOCIATED WITH ENHANCED PWM

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
APFCON1	—	—	SDO2SEL[2]	SS2SEL[2]	P1DSEL	P1CSEL	P2BSEL	CCP2SEL	119
CCP1CON	P1M<1:0>[1]		DC1B<1:0>		CCP1M<3:0>				224
CCP2CON	P2M<1:0>[1]		DC2B<1:0>		CCP2M<3:0>				224
CCP3CON	—	—	DC3B<1:0>		CCP3M<3:0>				224
CCP4CON	—	—	DC4B<1:0>		CCP4M<3:0>				224
CCP1AS	CCP1ASE	CCP1AS<2:0>			PSS1AC<1:0>		PSS1BD<1:0>		226
CCP2AS	CCP2ASE	CCP2AS<2:0>			PSS2AC<1:0>		PSS2BD<1:0>		226
CCPTMRS	C4TSEL<1:0>		C3TSEL<1:0>		C2TSEL<1:0>		C1TSEL<1:0>		225
INLVLA	—	—	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0	124
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
PIE2	OSFIE	C2IE	C1IE	EEIE	BCL1IE	—	—	CCP2IE	89
PIE3	—	—	CCP4IE	CCP3IE	TMR6IE	—	TMR4IE	—	90
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	92
PIR2	OSFIF	C2IF	C1IF	EEIF	BCL1IF	—	—	CCP2IF	93
PIR3	—	—	CCP4IF	CCP3IF	TMR6IF	—	TMR4IF	—	94
PRx	Timer2/4/6 Period Register								188*
PSTR1CON	—	—	—	STR1SYNC	STR1D	STR1C	STR1B	STR1A	228
PSTR2CON	—	—	—	STR2SYNC	STR2D	STR2C	STR2B	STR2A	228
PWM1CON	P1RSEN	P1DC<6:0>							227
PWM2CON	P2RSEN	P2DC<6:0>							227
T2CON	—	T2OUTPS<3:0>				TMR2ON	T2CKPS<1:0>		190
T4CON	—	T4OUTPS<3:0>				TMR4ON	T4CKPS<1:0>		190
T6CON	—	T6OUTPS<3:0>				TMR6ON	T6CKPS<1:0>		190
TMRx	Timer2/4/6 Module Register								188*
TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	122
TRISC	TRISC7[2]	TRISC6[2]	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	133

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

* Page provides register information.

Note 1: Applies to ECCP modules only.
  2: PIC16(L)F1829 only.

25.6.2 CLOCK ARBITRATION

Clock arbitration occurs when the master, during any receive, transmit or Repeated Start/Stop condition, releases the SCLx pin (SCLx allowed to float high). When the SCLx pin is allowed to float high, the Baud Rate Generator (BRG) is suspended from counting until the SCLx pin is actually sampled high. When the SCLx pin is sampled high, the Baud Rate Generator is reloaded with the contents of SSPxADD<7:0> and begins counting. This ensures that the SCLx high time will always be at least one BRG rollover count in the event that the clock is held low by an external device (Figure 25-25).

FIGURE 25-25: BAUD RATE GENERATOR TIMING WITH CLOCK ARBITRATION
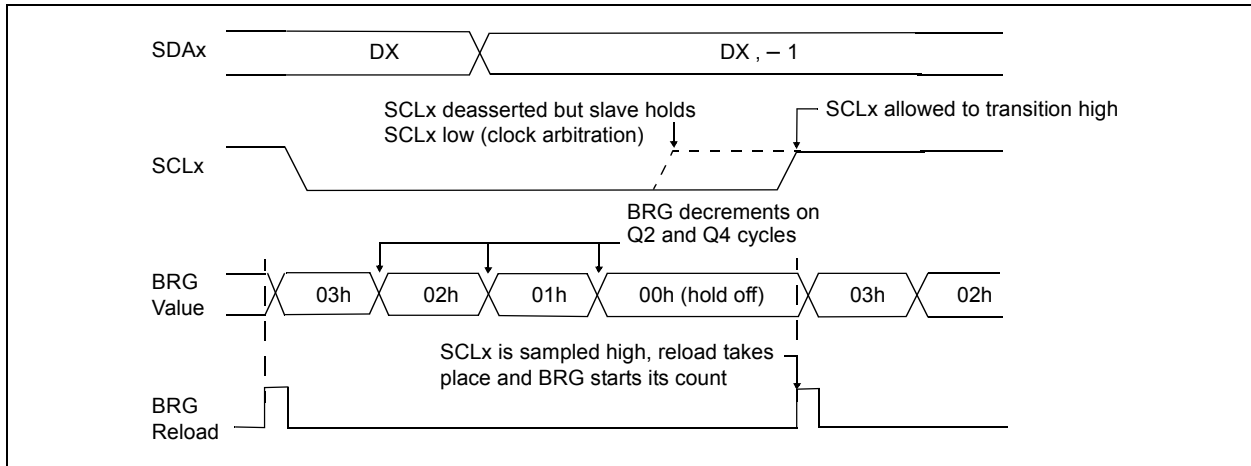
25.6.3 WCOL STATUS FLAG

If the user writes the SSPxBUF when a Start, Restart, Stop, Receive or Transmit sequence is in progress, the WCOL bit is set and the contents of the buffer are unchanged (the write does not occur). Any time the WCOL bit is set it indicates that an action on SSPxBUF was attempted while the module was not Idle.

Note:	Because queuing of events is not allowed, writing to the lower five bits of SSPxCON2 is disabled until the Start condition is complete.

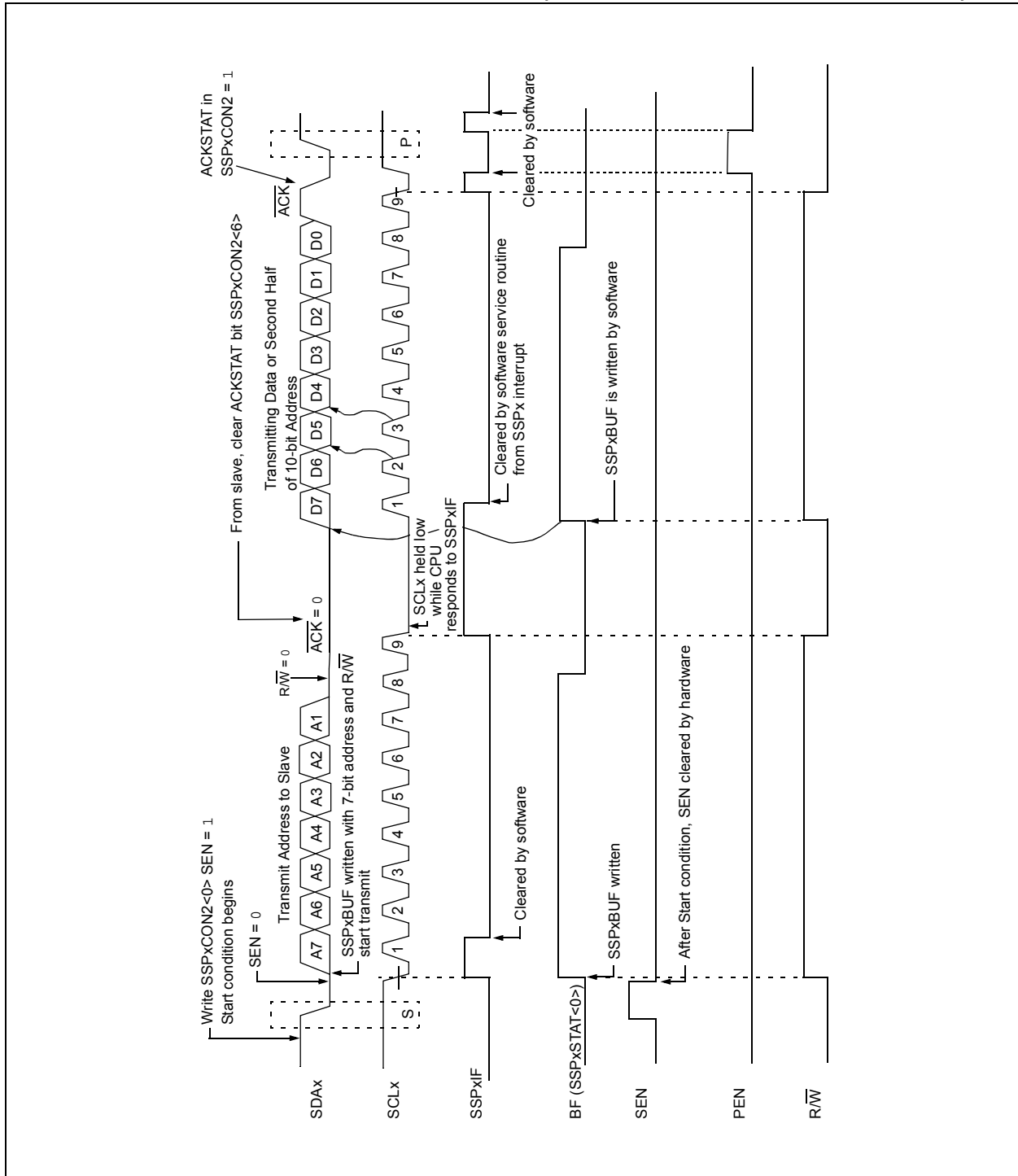**FIGURE 25-28:** **I²C MASTER MODE WAVEFORM (TRANSMISSION, 7 OR 10-BIT ADDRESS)**

FIGURE 26-4: ASYNCHRONOUS TRANSMISSION (BACK-TO-BACK)

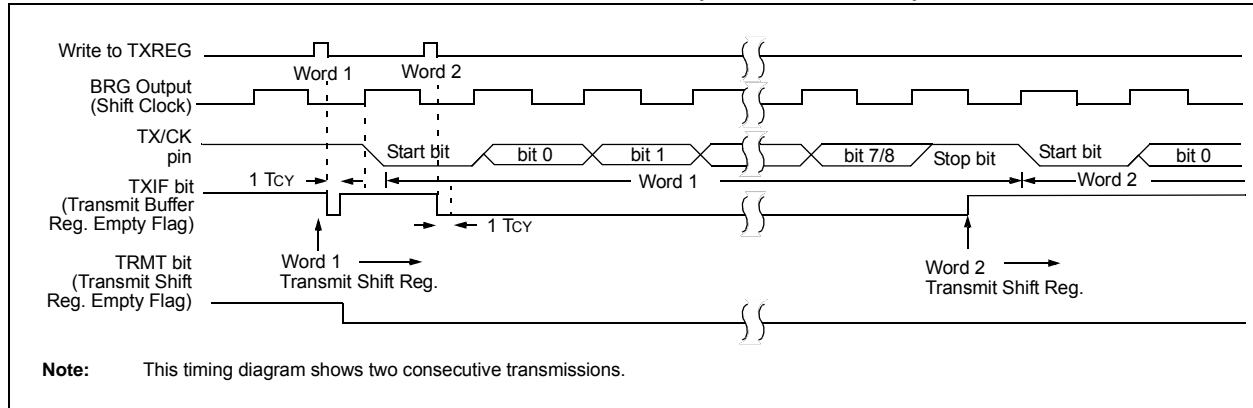**Note:** This timing diagram shows two consecutive transmissions.

TABLE 26-1: SUMMARY OF REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
APFCON0	RXDTSEL	SDO1SEL[2]	SS1SEL[2]	—	T1GSEL	TXCKSEL	—	—	118
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	292
INLVLA[3]	—	—	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0	124
INLVLB[1]	INLVLB7	INLVLB6	INLVLB5	INLVLB4	—	—	—	—	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
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	291
SPBRGL	SPBRG<7:0>								293*
SPBRGH	SPBRG<15:8>								293*
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
TXREG	EUSART Transmit Data Register								283
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	290

Legend: — Unimplemented location, read as '0'. Shaded cells are not used for asynchronous transmission.
       * Page provides register information.
Note 1: PIC16(L)F1829 only.
      2: PIC16(L)F1825 only.
      3: Unshaded cells apply to PIC16(L)F1825 only.

26.3 EUSART Baud Rate Generator (BRG)

The Baud Rate Generator (BRG) is an 8-bit or 16-bit timer that is dedicated to the support of both the asynchronous and synchronous EUSART operation. By default, the BRG operates in 8-bit mode. Setting the BRG16 bit of the BAUDCON register selects 16-bit mode.

The SPBRGH, SPBRGL register pair determines the period of the free running baud rate timer. In Asynchronous mode the multiplier of the baud rate period is determined by both the BRGH bit of the TXSTA register and the BRG16 bit of the BAUDCON register. In Synchronous mode, the BRGH bit is ignored.

Table 26-3 contains the formulas for determining the baud rate. Example 26-1 provides a sample calculation for determining the baud rate and baud rate error.

Typical baud rates and error values for various asynchronous modes have been computed for your convenience and are shown in Table 26-3. It may be advantageous to use the high baud rate (BRGH = 1), or the 16-bit BRG (BRG16 = 1) to reduce the baud rate error. The 16-bit BRG mode is used to achieve slow baud rates for fast oscillator frequencies.

Writing a new value to the SPBRGH, SPBRGL register pair causes the BRG timer to be reset (or cleared). This ensures that the BRG does not wait for a timer overflow before outputting the new baud rate.

If the system clock is changed during an active receive operation, a receive error or data loss may result. To avoid this problem, check the status of the RCIDL bit to make sure that the receive operation is Idle before changing the system clock.

EXAMPLE 26-1: CALCULATING BAUD RATE ERROR

For a device with F_{OSC} of 16 MHz, desired baud rate of 9600, Asynchronous mode, 8-bit BRG:

$$Desired\ Baud\ Rate = \frac{F_{OSC}}{64([SPBRGH:SPBRGL]+1)}$$

Solving for SPBRGH:SPBRGL:

$$X = \frac{\frac{F_{OSC}}{Desired\ Baud\ Rate}}{64} - 1$$

$$= \frac{\frac{16000000}{9600}}{64} - 1$$

$$= [25.042] = 25$$

$$Calculated\ Baud\ Rate = \frac{16000000}{64(25+1)}$$

$$= 9615$$

$$Error = \frac{Calc.\ Baud\ Rate - Desired\ Baud\ Rate}{Desired\ Baud\ Rate}$$

$$= \frac{(9615-9600)}{9600} = 0.16\%$$

MOVIW	Move INDFn to W
Syntax:	[*label*] MOVIW ++FSRn [*label*] MOVIW --FSRn [*label*] MOVIW FSRn++ [*label*] MOVIW FSRn-- [*label*] MOVIW k[FSRn]
Operands:	$n \in [0,1]$ $mm \in [00,01, 10, 11]$ $-32 \leq k \leq 31$
Operation:	INDFn $\rightarrow$ W Effective address is determined by • FSR + 1 (preincrement) • FSR - 1 (predecrement) • FSR + k (relative offset) After the Move, the FSR value will be either: • FSR + 1 (all increments) • FSR - 1 (all decrements) • Unchanged
Status Affected:	Z

Mode	Syntax	mm
Preincrement	++FSRn	00
Predecrement	--FSRn	01
Postincrement	FSRn++	10
Postdecrement	FSRn--	11

Description:	This instruction is used to move data between W and one of the indirect registers (INDFn). Before/after this move, the pointer (FSRn) is updated by pre/post incrementing/decrementing it. Note: The INDFn registers are not physical registers. Any instruction that accesses an INDFn register actually accesses the register at the address specified by the FSRn. FSRn is limited to the range 0000h - FFFFh. Incrementing/decrementing it beyond these bounds will cause it to wrap-around.

MOVLB	Move literal to BSR
Syntax:	[*label*] MOVLB k
Operands:	$0 \leq k \leq 15$
Operation:	k $\rightarrow$ BSR
Status Affected:	None
Description:	The 5-bit literal 'k' is loaded into the Bank Select Register (BSR).

MOVLP	Move literal to PCLATH
Syntax:	[*label*] MOVLP k
Operands:	$0 \leq k \leq 127$
Operation:	k $\rightarrow$ PCLATH
Status Affected:	None
Description:	The 7-bit literal 'k' is loaded into the PCLATH register.

MOVLW	Move literal to W
Syntax:	[*label*] MOVLW k
Operands:	$0 \leq k \leq 255$
Operation:	k $\rightarrow$ (W)
Status Affected:	None
Description:	The 8-bit literal 'k' is loaded into W register. The "don't cares" will assemble as '0's.
Words:	1
Cycles:	1
Example:	MOVLW 0x5A

After Instruction
W   =   0x5A

MOVWF	Move W to f
Syntax:	[*label*] MOVWF f
Operands:	$0 \leq f \leq 127$
Operation:	(W) $\rightarrow$ (f)
Status Affected:	None
Description:	Move data from W register to register 'f'.
Words:	1
Cycles:	1
Example:	MOVWF OPTION_REG

Before Instruction
OPTION_REG = 0xFF
W = 0x4F
After Instruction
OPTION_REG = 0x4F
W = 0x4F

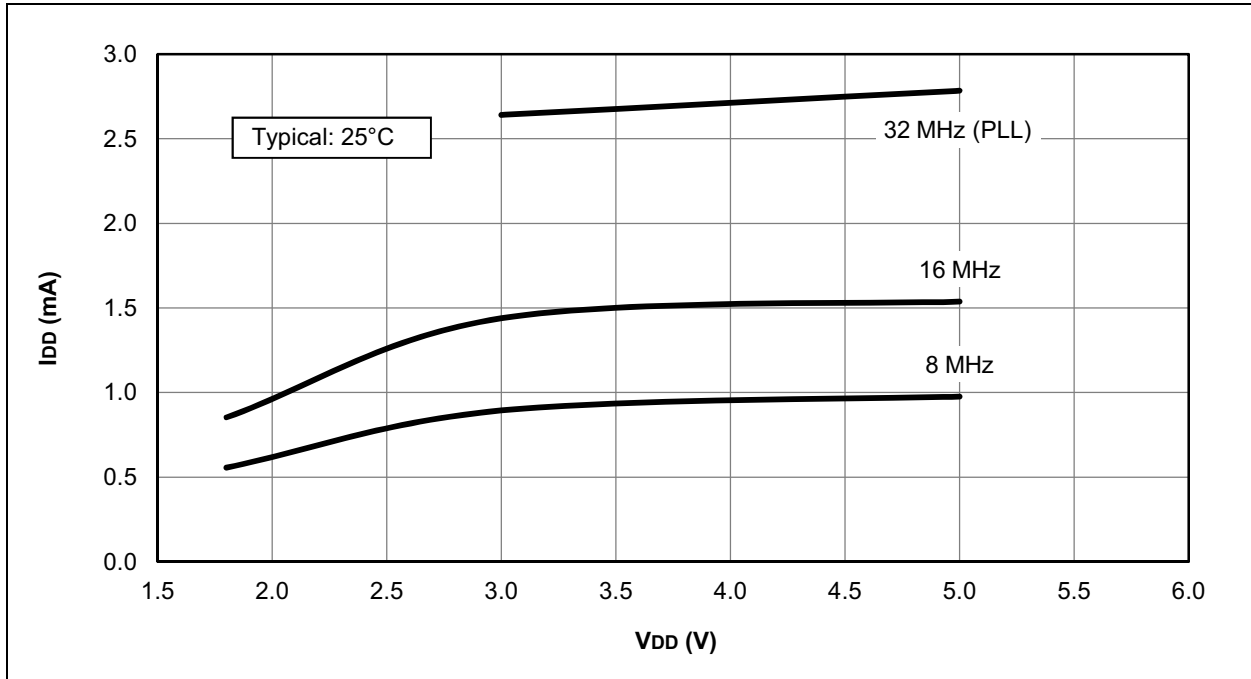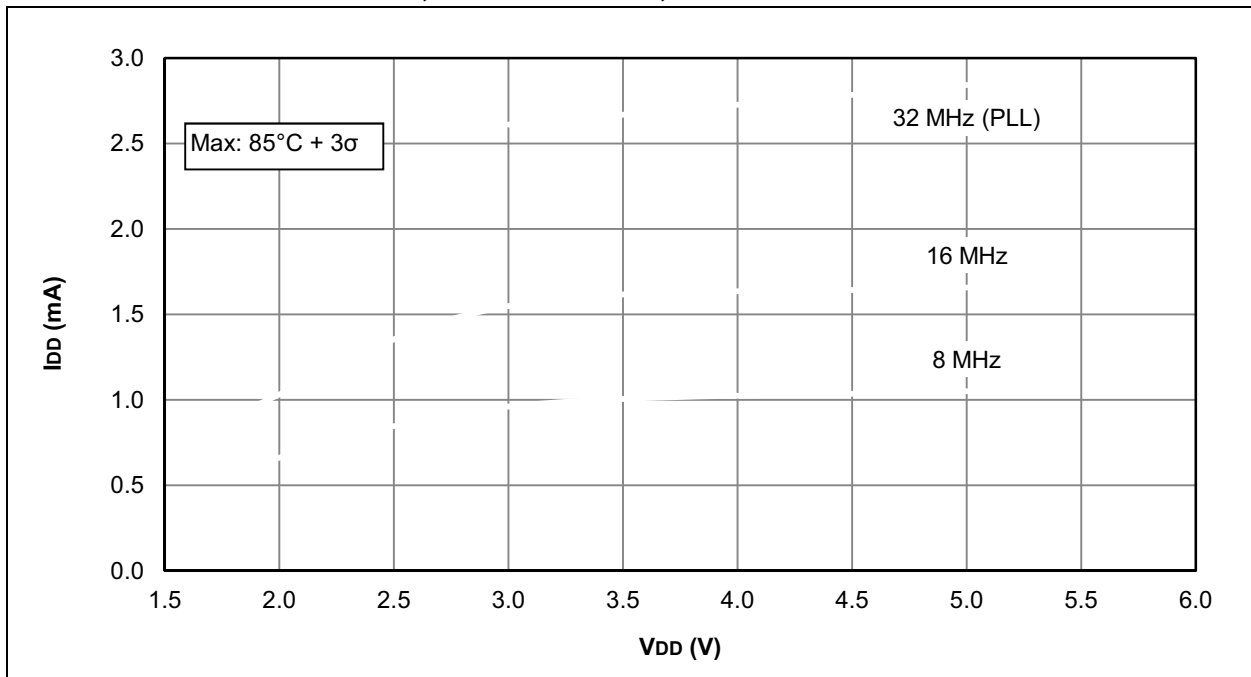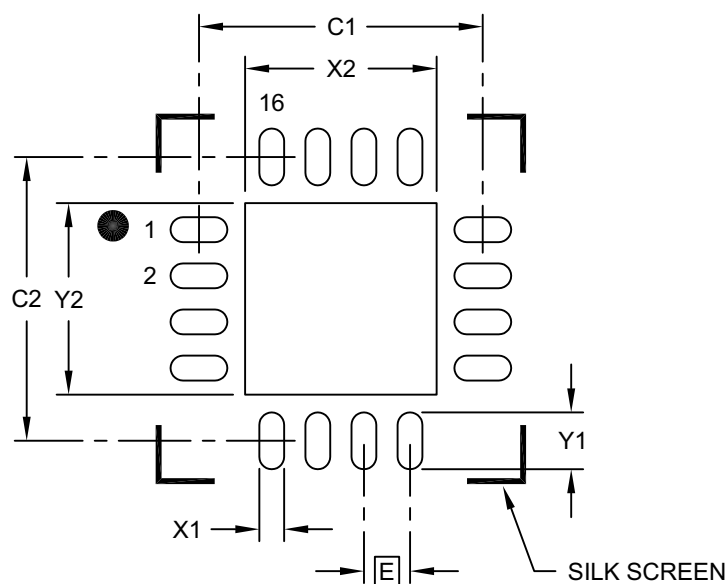**FIGURE 31-17:** I_{DD} TYPICAL, HFINTOSC MODE, PIC16F1825/9 ONLY

FIGURE 31-18: I_{DD} MAXIMUM, HFINTOSC MODE, PIC16F1825/9 ONLY

16-Lead Ultra Thin Plastic Quad Flat, No Lead Package (JQ) - 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

RECOMMENDED LAND PATTERN

	Units	MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Optional Center Pad Width	X2			2.70
Optional Center Pad Length	Y2			2.70
Contact Pad Spacing	C1		4.00	
Contact Pad Spacing	C2		4.00	
Contact Pad Width (X16)	X1			0.35
Contact Pad Length (X16)	Y1			0.80

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing C04-2257A