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	20MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	35
Program Memory Size	7KB (4K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	A/D 14x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f884-i-pt

TABLE 2: **28-PIN QFN ALLOCATION TABLE (PIC16F882/883/886)**

I/O	28-Pin QFN	Analog	Comparators	Timers	ECCP	EUSART	MSSP	Interrupt	Pull-up	Basic
RA0	27	AN0/ULPWU	C12IN0-	—	—	—	—	—	—	—
RA1	28	AN1	C12IN1-	—	—	—	—	—	—	—
RA2	1	AN2	C2IN+	—	—	—	—	—	—	VREF-/CVREF
RA3	2	AN3	C1IN+	—	—	—	—	—	—	VREF+
RA4	3	—	C1OUT	T0CKI	—	—	—	—	—	—
RA5	4	AN4	C2OUT	—	—	—	$\overline{SS}$	—	—	—
RA6	7	—	—	—	—	—	—	—	—	OSC2/CLKOUT
RA7	6	—	—	—	—	—	—	—	—	OSC1/CLKIN
RB0	18	AN12	—	—	—	—	—	IOC/INT	Y	—
RB1	19	AN10	C12IN3-	—	P1C	—	—	IOC	Y	—
RB2	20	AN8	—	—	P1B	—	—	IOC	Y	—
RB3	21	AN9	C12IN2-	—	—	—	—	IOC	Y	PGM
RB4	22	AN11	—	—	P1D	—	—	IOC	Y	—
RB5	23	AN13	—	$\overline{T1G}$	—	—	—	IOC	Y	—
RB6	24	—	—	—	—	—	—	IOC	Y	ICSPCLK
RB7	25	—	—	—	—	—	—	IOC	Y	ICSPDAT
RC0	8	—	—	T1OSO/T1CKI	—	—	—	—	—	—
RC1	9	—	—	T1OSI	CCP2	—	—	—	—	—
RC2	10	—	—	—	CCP1/P1A	—	—	—	—	—
RC3	11	—	—	—	—	—	SCK/SCL	—	—	—
RC4	12	—	—	—	—	—	SDI/SDA	—	—	—
RC5	13	—	—	—	—	—	SDO	—	—	—
RC6	14	—	—	—	—	TX/CK	—	—	—	—
RC7	15	—	—	—	—	RX/DT	—	—	—	—
RE3	26	—	—	—	—	—	—	—	Y[1]	$\overline{MCLR}$/VPP
—	17	—	—	—	—	—	—	—	—	VDD
—	5	—	—	—	—	—	—	—	—	VSS
—	16	—	—	—	—	—	—	—	—	VSS

Note 1: Pull-up activated only with external $\overline{MCLR}$ configuration.

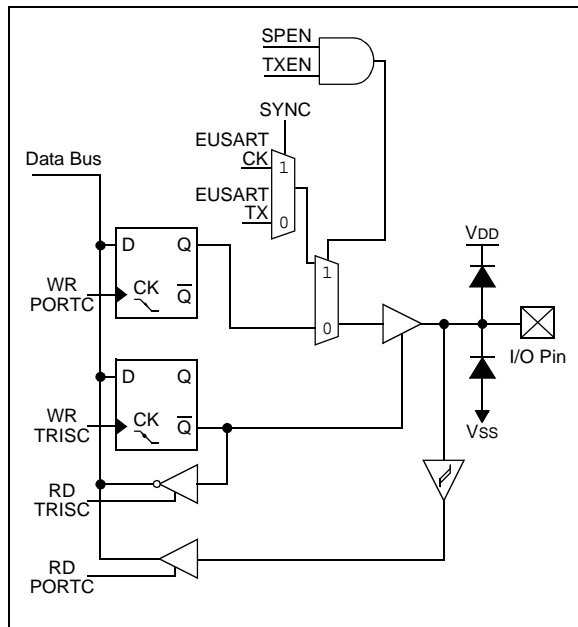### 3.5.7   RC6/TX/CK

Figure 3-17 shows the diagram for this pin. This pin is configurable to function as one of the following:

- a general purpose I/O
- an asynchronous serial output
- a synchronous clock I/O

3.5.8   RC7/RX/DT

Figure 3-18 shows the diagram for this pin. This pin is configurable to function as one of the following:

- a general purpose I/O
- an asynchronous serial input
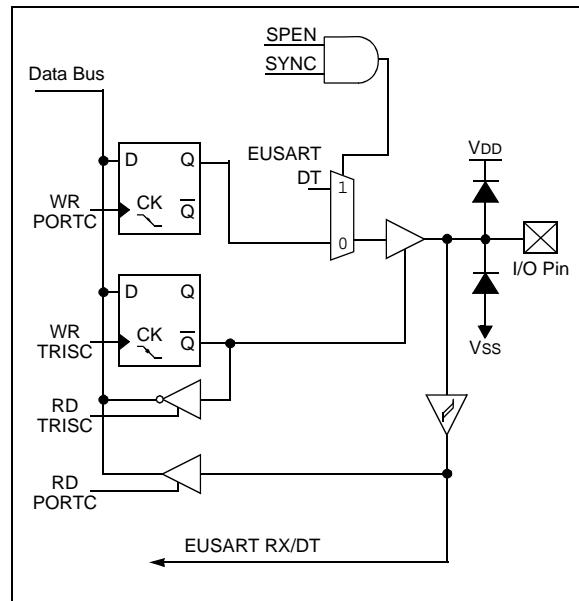- a synchronous serial data I/O

FIGURE 3-17:   BLOCK DIAGRAM OF RC6

FIGURE 3-18:   BLOCK DIAGRAM OF RC7

TABLE 3-3:   SUMMARY OF REGISTERS ASSOCIATED WITH PORTC

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CCP1CON	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	122
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	123
PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	54
PSTRCON	—	—	—	STRSYNC	STRD	STRC	STRB	STRA	144
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	158
SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	177
T1CON	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	81
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	54

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

6.12   Timer1 Control Register

The Timer1 Control register (T1CON), shown in Register 6-1, is used to control Timer1 and select the various features of the Timer1 module.

REGISTER DEFINITIONS: TIMER1 CONTROL

REGISTER 6-1:   T1CON: TIMER1 CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
T1GINV[(1)]	TMR1GE[(2)]	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared x = Bit is unknown

bit 7   **T1GINV:** Timer1 Gate Invert bit[(1)]

1 = Timer1 gate is active-high (Timer1 counts when gate is high)
0 = Timer1 gate is active-low (Timer1 counts when gate is low)

bit 6   **TMR1GE:** Timer1 Gate Enable bit[(2)]

If TMR1ON = 0:
This bit is ignored
If TMR1ON = 1:
1 = Timer1 counting is controlled by the Timer1 Gate function
0 = Timer1 is always counting

bit 5-4   **T1CKPS<1:0>:** Timer1 Input Clock Prescale Select bits

11 = 1:8 Prescale Value
10 = 1:4 Prescale Value
01 = 1:2 Prescale Value
00 = 1:1 Prescale Value

bit 3   **T1OSCEN:** LP Oscillator Enable Control bit

1 = LP oscillator is enabled for Timer1 clock
0 = LP oscillator is off

bit 2   **$\overline{\text{T1SYNC}}$:** Timer1 External Clock Input Synchronization Control bit

TMR1CS = 1:
1 = Do not synchronize external clock input
0 = Synchronize external clock input
TMR1CS = 0:
This bit is ignored. Timer1 uses the internal clock

bit 1   **TMR1CS:** Timer1 Clock Source Select bit

1 = External clock from T1CKI pin (on the rising edge)
0 = Internal clock (F$_{OSC}$/4)

bit 0   **TMR1ON:** Timer1 On bit

1 = Enables Timer1
0 = Stops Timer1

Note 1:   T1GINV bit inverts the Timer1 gate logic, regardless of source.
        2:   TMR1GE bit must be set to use either $\overline{\text{T1G}}$ pin or C2OUT, as selected by the T1GSS bit of the CM2CON1 register, as a Timer1 gate source.

REGISTER DEFINITIONS: VOLTAGE REFERENCE CONTROL

REGISTER 8-5: VRCON: VOLTAGE REFERENCE CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
VREN	VROE	VRR	VRSS	VR3	VR2	VR1	VR0
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7   **VREN:** Comparator C1 Voltage Reference Enable bit
            1 = CVREF circuit powered on
            0 = CVREF circuit powered down

bit 6   **VROE:** Comparator C2 Voltage Reference Enable bit
            1 = CVREF voltage level is also output on the RA2/AN2/VREF-/CVREF/C2IN+ pin
            0 = CVREF voltage is disconnected from the RA2/AN2/VREF-/CVREF/C2IN+ pin

bit 5   **VRR:** CVREF Range Selection bit
            1 = Low range
            0 = High range

bit 4   **VRSS:** Comparator VREF Range Selection bit
            1 = Comparator Reference Source, CVRSRC = (VREF+) - (VREF-)
            0 = Comparator Reference Source, CVRSRC = VDD - VSS

bit 3-0   **VR<3:0>:** CVREF Value Selection $0 \leq$ VR<3:0> ≤ 15
            <u>When VRR = 1</u>: CVREF = (VR<3:0>/24) * VDD
            <u>When VRR = 0</u>: CVREF = VDD/4 + (VR<3:0>/32) * VDD

TABLE 8-3: SUMMARY OF REGISTERS ASSOCIATED WITH THE COMPARATOR AND VOLTAGE REFERENCE MODULES

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	41
ANSELH	—	—	ANS13	ANS12	ANS11	ANS10	ANS9	ANS8	49
CM1CON0	C1ON	C1OUT	C1OE	C1POL	—	C1R	C1CH1	C1CH0	89
CM2CON0	C2ON	C2OUT	C2OE	C2POL	—	C2R	C2CH1	C2CH0	90
CM2CON1	MC1OUT	MC2OUT	C1RSEL	C2RSEL	—	—	T1GSS	C2SYNC	92
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	32
PIE2	OSFIE	C2IE	C1IE	EEIE	BCLIE	ULPWUIE	—	CCP2IE	34
PIR2	OSFIF	C2IF	C1IF	EEIF	BCLIF	ULPWUIF	—	CCP2IF	36
PORTA	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	40
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	49
SRCON	SR1	SR0	C1SEN	C2SEN	PULSS	PULSR	—	FVREN	94
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	40
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	49
VRCON	VREN	VROE	VRR	VRSS	VR3	VR2	VR1	VR0	98

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

11.2 Capture/Compare/PWM (CCP2)

The Capture/Compare/PWM module is a peripheral which allows the user to time and control different events. In Capture mode, the peripheral allows the timing of the duration of an event. The Compare mode allows the user to trigger an external event when a predetermined amount of time has expired. The PWM mode can generate a Pulse-Width Modulated signal of varying frequency and duty cycle.

The timer resources used by the module are shown in Table 11-2.

Additional information on CCP modules is available in the Application Note AN594, *"Using the CCP Modules"* (DS00594).

TABLE 11-2: CCP MODE – TIMER RESOURCES REQUIRED

CCP Mode	Timer Resource
Capture	Timer1
Compare	Timer1
PWM	Timer2

REGISTER 11-2: CCP2CON: CCP2 CONTROL REGISTER

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared x = Bit is unknown

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

bit 5-4   **DC2B<1:0>:** PWM Duty Cycle Least Significant bits
Capture mode:
Unused.
Compare mode:
Unused.
PWM mode:
These bits are the two LSbs of the PWM duty cycle. The eight MSbs are found in CCPR2L.

bit 3-0   **CCP2M<3:0>:** CCP2 Mode Select bits
0000 = Capture/Compare/PWM off (resets CCP2 module)
0001 = Unused (reserved)
0010 = Unused (reserved)
0011 = Unused (reserved)
0100 = Capture mode, every falling edge
0101 = Capture mode, every rising edge
0110 = Capture mode, every 4th rising edge
0111 = Capture mode, every 16th rising edge
1000 = Compare mode, set output on match (CCP2IF bit is set)
1001 = Compare mode, clear output on match (CCP2IF bit is set)
1010 = Compare mode, generate software interrupt on match (CCP2IF bit is set, CCP2 pin is unaffected)
1011 = Compare mode, trigger special event (CCP2IF bit is set, TMR1 is reset and A/D conversion is started if the ADC module is enabled. CCP2 pin is unaffected.)
11xx = PWM mode.

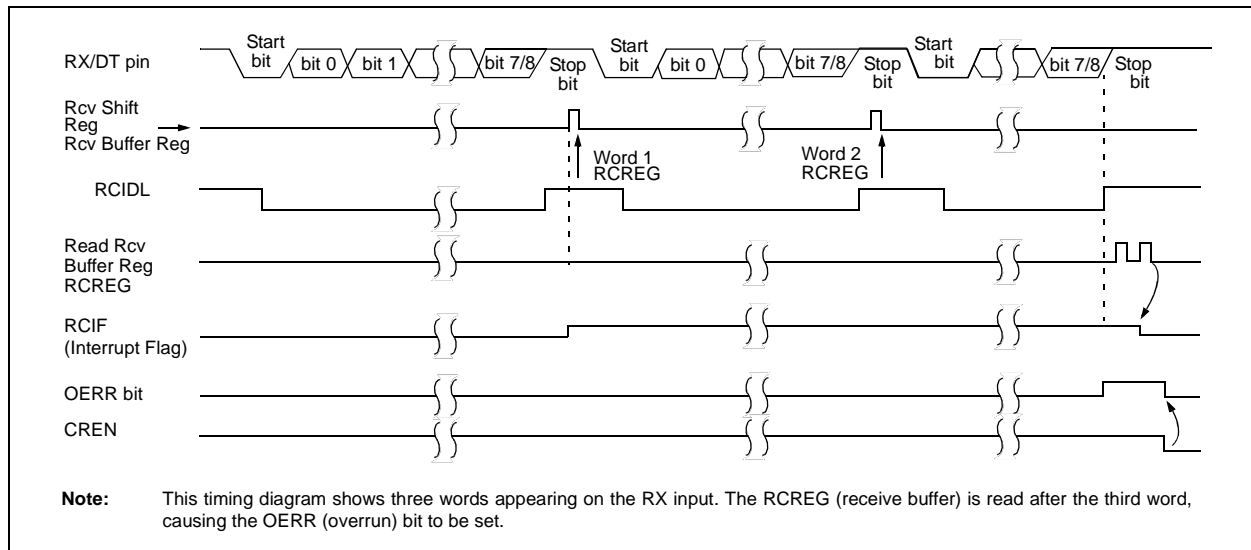### 12.1.2.8   Asynchronous Reception Setup:

1. Initialize the SPBRGH, SPBRG register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 12.3 "EUSART Baud Rate Generator (BRG)"**).

2. Enable the serial port by setting the SPEN bit. The SYNC bit must be clear for asynchronous operation.

3. If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.

4. If 9-bit reception is desired, set the RX9 bit.

5. Enable reception by setting the CREN bit.

6. The RCIF interrupt flag bit will be set when a character is transferred from the RSR to the receive buffer. An interrupt will be generated if the RCIE interrupt enable bit was also set.

7. Read the RCSTA register to get the error flags and, if 9-bit data reception is enabled, the ninth data bit.

8. Get the received eight Least Significant data bits from the receive buffer by reading the RCREG register.

9. If an overrun occurred, clear the OERR flag by clearing the CREN receiver enable bit.

12.1.2.9   9-bit Address Detection Mode Setup

This mode would typically be used in RS-485 systems. To set up an Asynchronous Reception with Address Detect Enable:

1. Initialize the SPBRGH, SPBRG register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 12.3 "EUSART Baud Rate Generator (BRG)"**).

2. Enable the serial port by setting the SPEN bit. The SYNC bit must be clear for asynchronous operation.

3. If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.

4. Enable 9-bit reception by setting the RX9 bit.

5. Enable address detection by setting the ADDEN bit.

6. Enable reception by setting the CREN bit.

7. The RCIF interrupt flag bit will be set when a character with the ninth bit set is transferred from the RSR to the receive buffer. An interrupt will be generated if the RCIE interrupt enable bit was also set.

8. Read the RCSTA register to get the error flags. The ninth data bit will always be set.

9. Get the received eight Least Significant data bits from the receive buffer by reading the RCREG register. Software determines if this is the device's address.

10. If an overrun occurred, clear the OERR flag by clearing the CREN receiver enable bit.

11. If the device has been addressed, clear the ADDEN bit to allow all received data into the receive buffer and generate interrupts.

FIGURE 12-5:   ASYNCHRONOUS RECEPTION

Note: This timing diagram shows three words appearing on the RX input. The RCREG (receive buffer) is read after the third word, causing the OERR (overrun) bit to be set.

12.3 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 BAUDCTL register selects 16-bit mode.

The SPBRGH, SPBRG 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 BAUDCTL register. In Synchronous mode, the BRGH bit is ignored.

Table 12-3 contains the formulas for determining the baud rate. Example 12-1 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 12-3. 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, SPBRG 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 12-1: CALCULATING BAUD RATE ERROR

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

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

Solving for SPBRGH:SPBRG:

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

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

$$= [25.042] = 25$$

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

$$= 9615$$

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

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

TABLE 12-3: BAUD RATE FORMULAS

Configuration Bits			BRG/EUSART Mode	Baud Rate Formula
SYNC	BRG16	BRGH		
0	0	0	8-bit/Asynchronous	$F_{OSC}/[64\ (n+1)]$
0	0	1	8-bit/Asynchronous	$F_{OSC}/[16\ (n+1)]$
0	1	0	16-bit/Asynchronous	
0	1	1	16-bit/Asynchronous	$F_{OSC}/[4\ (n+1)]$
1	0	x	8-bit/Synchronous	
1	1	x	16-bit/Synchronous	

Legend: x = Don't care, n = value of SPBRGH, SPBRG register pair

TABLE 12-4: REGISTERS ASSOCIATED WITH THE BAUD RATE GENERATOR

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BAUDCTL	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	159
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	158
SPBRG	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0	160
SPBRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	160
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	157

Legend: x = unknown, – = unimplemented read as '0'. Shaded cells are not used for the Baud Rate Generator.

12.4 EUSART Synchronous Mode

Synchronous serial communications are typically used in systems with a single master and one or more slaves. The master device contains the necessary circuitry for baud rate generation and supplies the clock for all devices in the system. Slave devices can take advantage of the master clock by eliminating the internal clock generation circuitry.

There are two signal lines in Synchronous mode: a bidirectional data line and a clock line. Slaves use the external clock supplied by the master to shift the serial data into and out of their respective receive and transmit shift registers. Since the data line is bidirectional, synchronous operation is half-duplex only. Half-duplex refers to the fact that master and slave devices can receive and transmit data but not both simultaneously. The EUSART can operate as either a master or slave device.

Start and Stop bits are not used in synchronous transmissions.

12.4.1 SYNCHRONOUS MASTER MODE

The following bits are used to configure the EUSART for Synchronous Master operation:

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

Setting the SYNC bit of the TXSTA register configures the device for synchronous operation. Setting the CSRC bit of the TXSTA register configures the device as a master. Clearing the SREN and CREN bits of the RCSTA register ensures that the device is in the Transmit mode, otherwise the device will be configured to receive. Setting the SPEN bit of the RCSTA register enables the EUSART. If the RX/DT or TX/CK pins are shared with an analog peripheral the analog I/O functions must be disabled by clearing the corresponding ANSEL bits.

12.4.1.1 Master Clock

Synchronous data transfers use a separate clock line, which is synchronous with the data. A device configured as a master transmits the clock on the TX/CK line. The TX/CK pin output driver is automatically enabled when the EUSART is configured for synchronous transmit or receive operation. Serial data bits change on the leading edge to ensure they are valid at the trailing edge of each clock. One clock cycle is generated for each data bit. Only as many clock cycles are generated as there are data bits.

12.4.1.2 Clock Polarity

A clock polarity option is provided for Microwire compatibility. Clock polarity is selected with the SCKP bit of the BAUDCTL register. Setting the SCKP bit sets the clock Idle state as high. When the SCKP bit is set, the data changes on the falling edge of each clock. Clearing the SCKP bit sets the Idle state as low. When the SCKP bit is cleared, the data changes on the rising edge of each clock.

12.4.1.3 Synchronous Master Transmission

Data is transferred out of the device on the RX/DT pin. The RX/DT and TX/CK pin output drivers are automatically enabled when the EUSART is configured for synchronous master transmit operation.

A transmission is initiated by writing a character to the TXREG register. If the TSR still contains all or part of a previous character the new character data is held in the TXREG until the last bit of the previous character has been transmitted. 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. The transmission of the character commences immediately following the transfer of the data to the TSR from the TXREG.

Each data bit changes on the leading edge of the master clock and remains valid until the subsequent leading clock edge.

> **Note:** The TSR register is not mapped in data memory, so it is not available to the user.

12.4.1.4 Synchronous Master Transmission Setup:

1. Initialize the SPBRGH, SPBRG register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 12.3 "EUSART Baud Rate Generator (BRG)"**).
2. Enable the synchronous master serial port by setting bits SYNC, SPEN, and CSRC.
3. Disable Receive mode by clearing bits SREN and CREN.
4. Enable Transmit mode by setting the TXEN bit.
5. If 9-bit transmission is desired, set the TX9 bit.
6. If interrupts are desired, set the TXIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
7. If 9-bit transmission is selected, the ninth bit should be loaded in the TX9D bit.
8. Start transmission by loading data to the TXREG register.

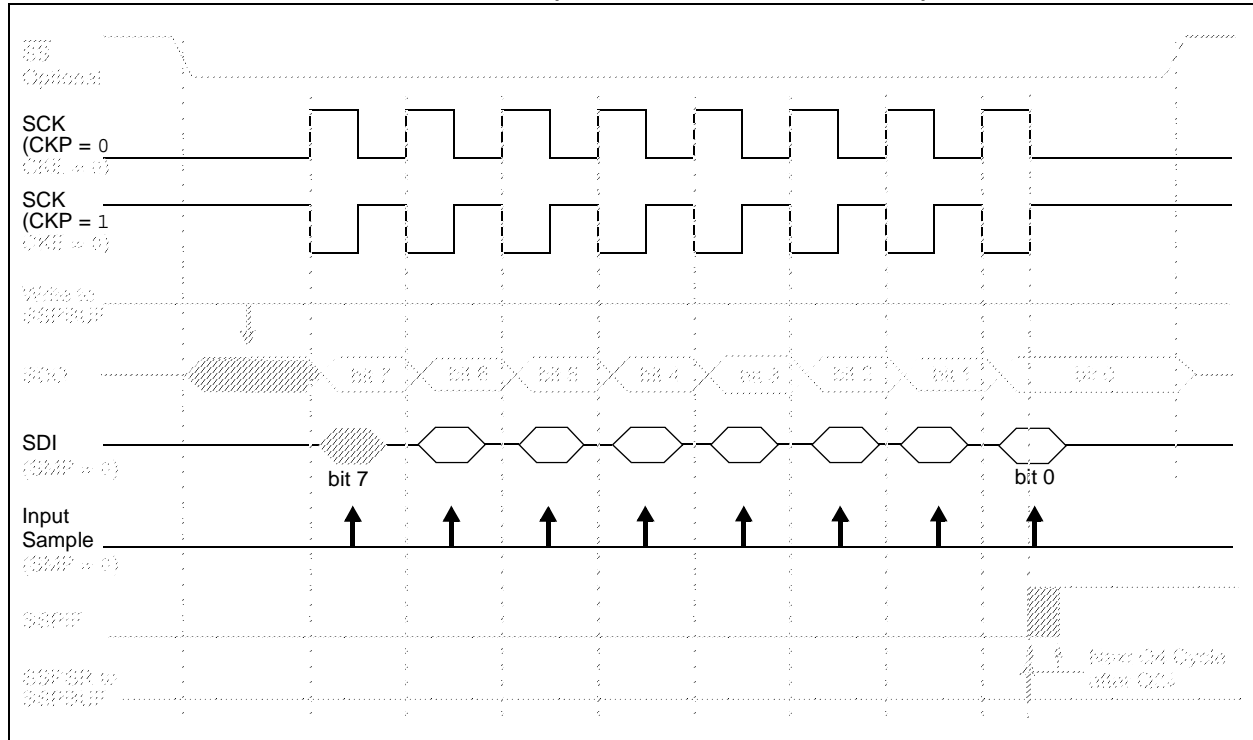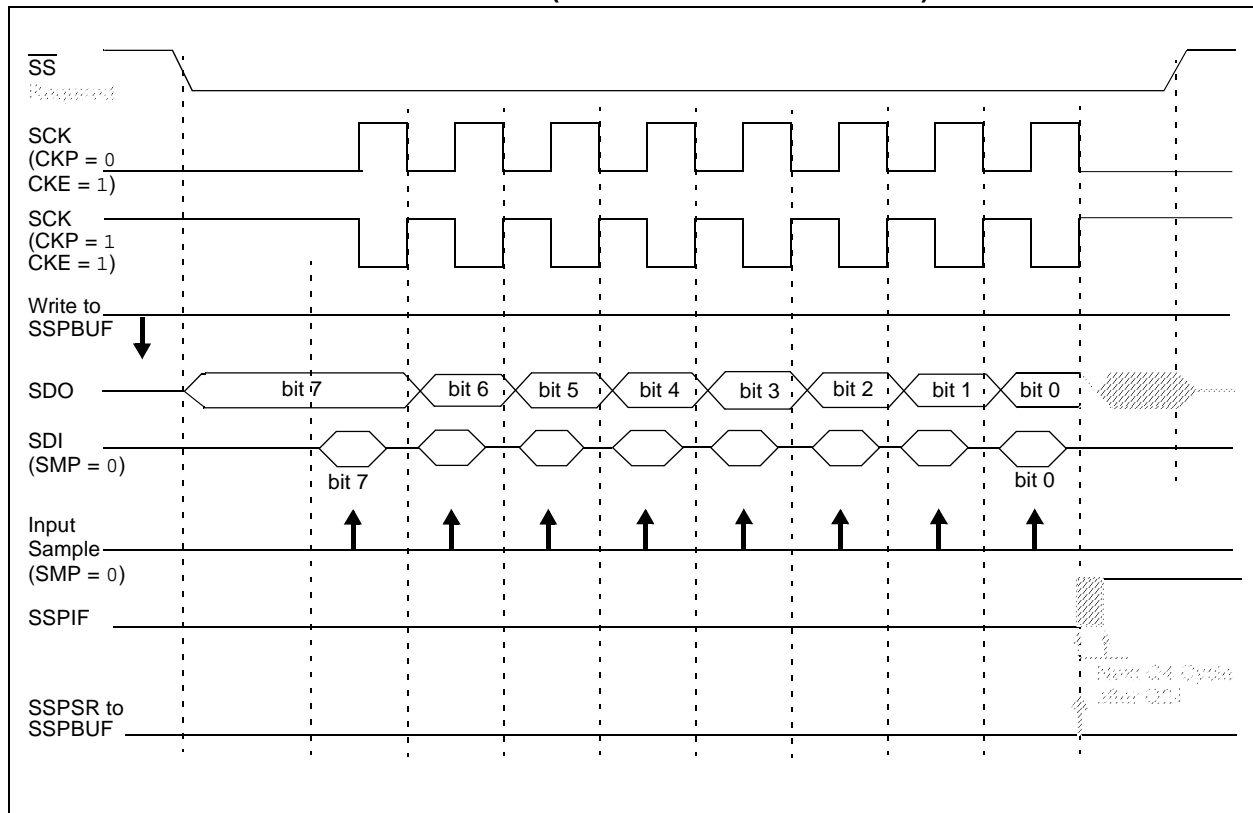**FIGURE 13-4:** SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 0)

FIGURE 13-5: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 1)

13.4.1.1 Addressing

Once the MSSP module has been enabled, it waits for a Start condition to occur. Following the Start condition, the eight bits are shifted into the SSPSR register. All incoming bits are sampled with the rising edge of the clock (SCL) line. The value of register SSPSR<7:1> is compared to the value of the SSPADD register. The address is compared on the falling edge of the eighth clock (SCL) pulse. If the addresses match, and the BF and SSPOV bits are clear, the following events occur:

a) The SSPSR register value is loaded into the SSPBUF register.

b) The buffer full bit BF is set.

c) An $\overline{ACK}$ pulse is generated.

d) MSSP interrupt flag bit, SSPIF of the PIR1 register, is set on the falling edge of the ninth SCL pulse (interrupt is generated, if enabled).

In 10-bit address mode, two address bytes need to be received by the slave. The five Most Significant bits (MSb) of the first address byte specify if this is a 10-bit address. The R/$\overline{W}$ bit (SSPSTAT register) must specify a write so the slave device will receive the second address byte. For a 10-bit address, the first byte would equal '1111 0 A9 A8 0', where A9 and A8 are the two MSb's of the address.

The sequence of events for 10-bit addressing is as follows, with steps 7-9 for slave-transmitter:

1. Receive first (high) byte of address (bit SSPIF of the PIR1 register and bits BF and UA of the SSPSTAT register are set).

2. Update the SSPADD register with second (low) byte of address (clears bit UA and releases the SCL line).

3. Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.

4. Receive second (low) byte of address (bits SSPIF, BF, and UA are set).

5. Update the SSPADD register with the first (high) byte of address. If match releases SCL line, this will clear bit UA.

6. Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.

7. Receive Repeated Start condition.

8. Receive first (high) byte of address (bits SSPIF and BF are set).

9. Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.

13.4.1.2 Reception

When the R/$\overline{W}$ bit of the address byte is clear and an address match occurs, the R/$\overline{W}$ bit of the SSPSTAT register is cleared. The received address is loaded into the SSPBUF register.

When the address byte overflow condition exists, then no Acknowledge ($\overline{ACK}$) pulse is given. An overflow condition is defined as either bit BF (SSPSTAT register) is set, or bit SSPOV (SSPCON register) is set.

An MSSP interrupt is generated for each data transfer byte. Flag bit SSPIF of the PIR1 register must be cleared in software. The SSPSTAT register is used to determine the status of the byte.

13.4.1.3 Transmission

When the R/$\overline{W}$ bit of the incoming address byte is set and an address match occurs, the R/$\overline{W}$ bit of the SSPSTAT register is set. The received address is loaded into the SSPBUF register. The $\overline{ACK}$ pulse will be sent on the ninth bit and pin RC3/SCK/SCL is held low. The transmit data must be loaded into the SSPBUF register, which also loads the SSPSR register. Then pin RC3/SCK/SCL should be enabled by setting bit CKP (SSPCON register). The master must monitor the SCL pin prior to asserting another clock pulse. The slave devices may be holding off the master by stretching the clock. The eight data bits are shifted out on the falling edge of the SCL input. This ensures that the SDA signal is valid during the SCL high time (Figure 13-8).

An MSSP interrupt is generated for each data transfer byte. The SSPIF bit must be cleared in software and the SSPSTAT register is used to determine the status of the byte. The SSPIF bit is set on the falling edge of the ninth clock pulse.

As a slave-transmitter, the $\overline{ACK}$ pulse from the master-receiver is latched on the rising edge of the ninth SCL input pulse. If the SDA line is high (not $\overline{ACK}$), then the data transfer is complete. When the $\overline{ACK}$ is latched by the slave, the slave logic is reset and the slave monitors for another occurrence of the Start bit. If the SDA line was low ($\overline{ACK}$), the transmit data must be loaded into the SSPBUF register, which also loads the SSPSR register. Pin RC3/SCK/SCL should be enabled by setting bit CKP.

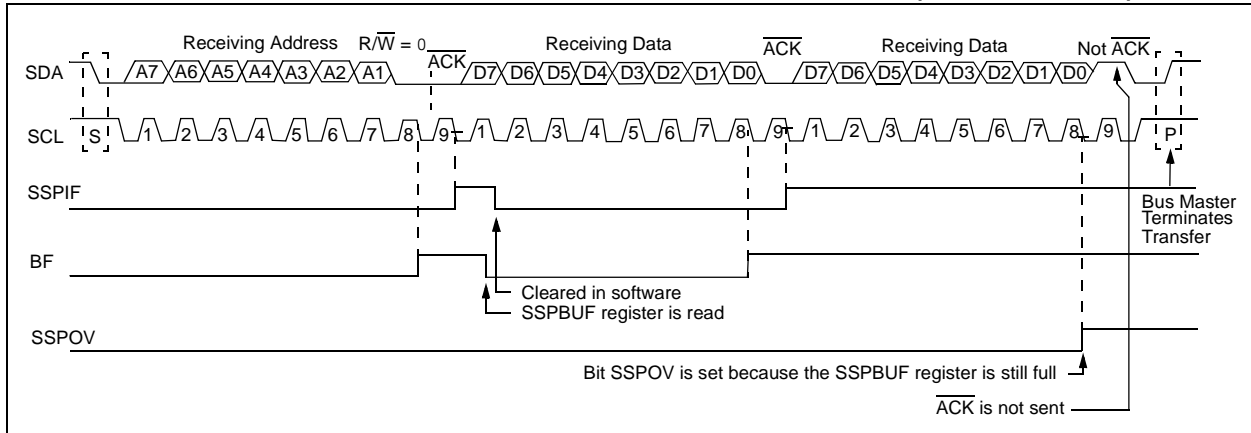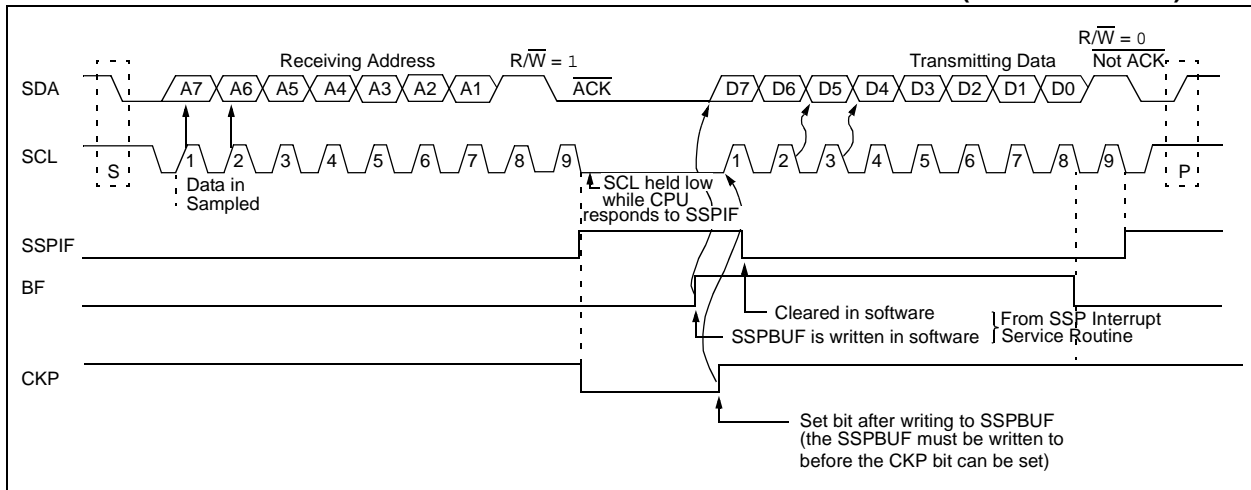**FIGURE 13-7:** **I²C™ SLAVE MODE WAVEFORMS FOR RECEPTION (7-BIT ADDRESS)**

FIGURE 13-8: **I²C™ SLAVE MODE WAVEFORMS FOR TRANSMISSION (7-BIT ADDRESS)**

PIC16F882/883/884/886/887

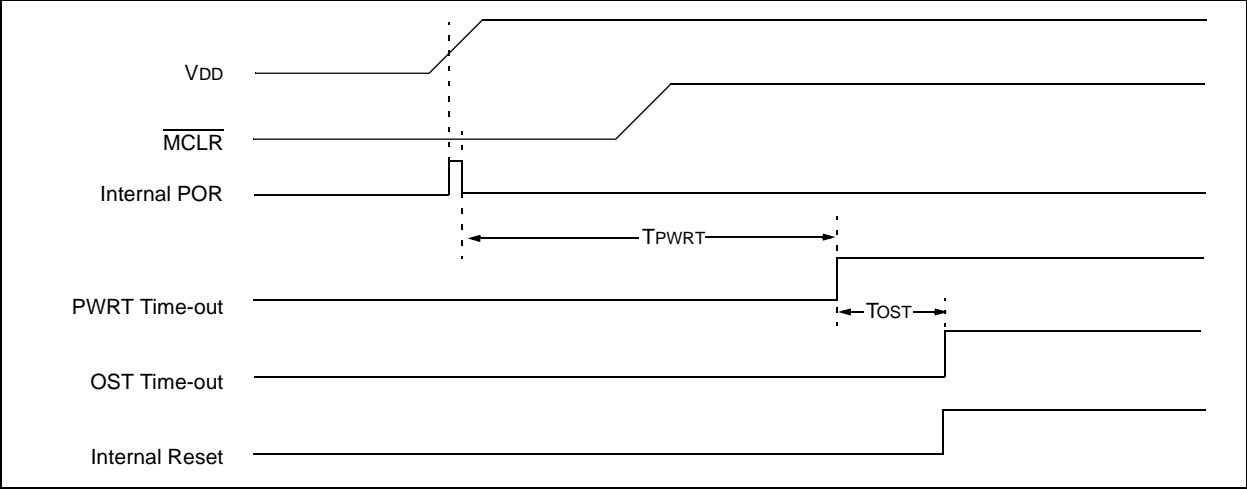**FIGURE 14-4:** TIME-OUT SEQUENCE ON POWER-UP (DELAYED $\overline{\text{MCLR}}$): CASE 1

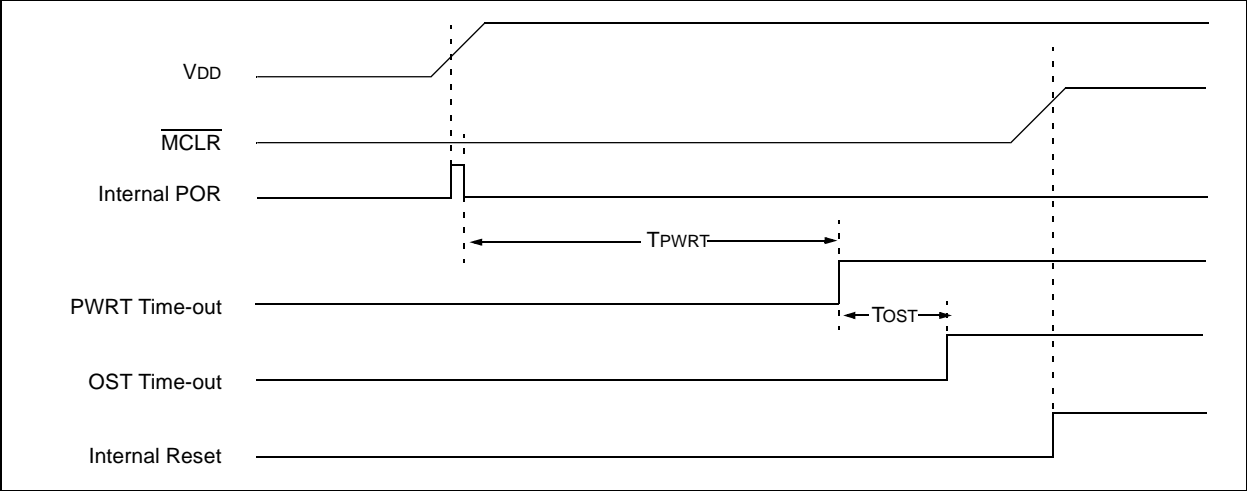**FIGURE 14-5:** TIME-OUT SEQUENCE ON POWER-UP (DELAYED $\overline{\text{MCLR}}$): CASE 2
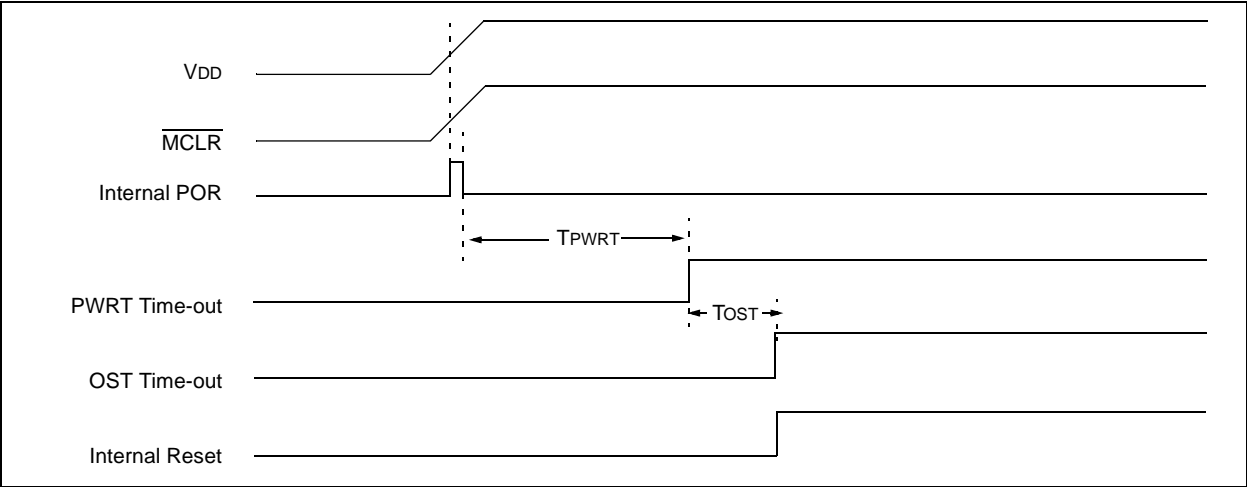

FIGURE 14-6: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ WITH V$_{DD}$)

FIGURE 14-8: **INT PIN INTERRUPT TIMING**

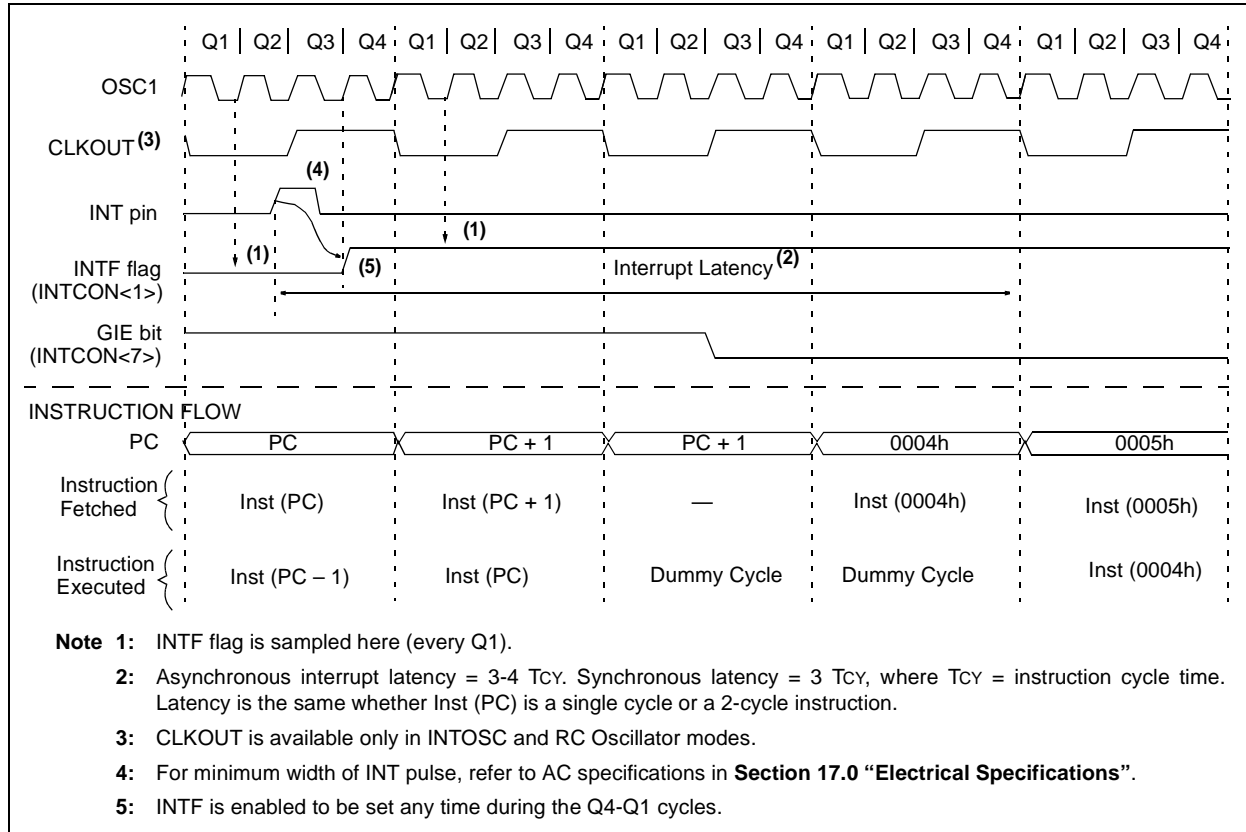**Note 1:** INTF flag is sampled here (every Q1).

2: Asynchronous interrupt latency = 3-4 TCY. Synchronous latency = 3 TCY, where TCY = instruction cycle time. Latency is the same whether Inst (PC) is a single cycle or a 2-cycle instruction.

3: CLKOUT is available only in INTOSC and RC Oscillator modes.

4: For minimum width of INT pulse, refer to AC specifications in **Section 17.0 "Electrical Specifications"**.

5: INTF is enabled to be set any time during the Q4-Q1 cycles.

TABLE 14-6: **SUMMARY OF INTERRUPT REGISTERS**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	32
PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	33
PIE2	OSFIE	C2IE	C1IE	EEIE	BCLIE	ULPWUIE	—	CCP2IE	34
PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	35
PIR2	OSFIF	C2IF	C1IF	EEIF	BCLIF	ULPWUIF	—	CCP2IF	36

Legend: x = unknown, u = unchanged, — = unimplemented read as '0', q = value depends upon condition.
Shaded cells are not used by the interrupt module.

14.5   Watchdog Timer (WDT)

The WDT has the following features:

- Operates from the LFINTOSC (31 kHz)
- Contains a 16-bit prescaler
- Shares an 8-bit prescaler with Timer0
- Time-out period is from 1 ms to 268 seconds
- Configuration bit and software controlled

WDT is cleared under certain conditions described in Table 14-7.

14.5.1   WDT OSCILLATOR

The WDT derives its time base from the 31 kHz LFINTOSC. The LTS bit of the OSCCON register does not reflect that the LFINTOSC is enabled.

The value of WDTCON is '`---0 1000`' on all Resets. This gives a nominal time base of 17 ms.

> **Note:** When the Oscillator Start-up Timer (OST) is invoked, the WDT is held in Reset, because the WDT Ripple Counter is used by the OST to perform the oscillator delay count. When the OST count has expired, the WDT will begin counting (if enabled).

14.5.2   WDT CONTROL

The WDTE bit is located in the Configuration Word Register 1. When set, the WDT runs continuously.

When the WDTE bit in the Configuration Word Register 1 is set, the SWDTEN bit of the WDTCON register has no effect. If WDTE is clear, then the SWDTEN bit can be used to enable and disable the WDT. Setting the bit will enable it and clearing the bit will disable it.

The PSA and PS<2:0> bits of the OPTION register have the same function as in previous versions of the PIC16F882/883/884/886/887 family of microcontrollers. See **Section 5.0 "Timer0 Module"** for more information.
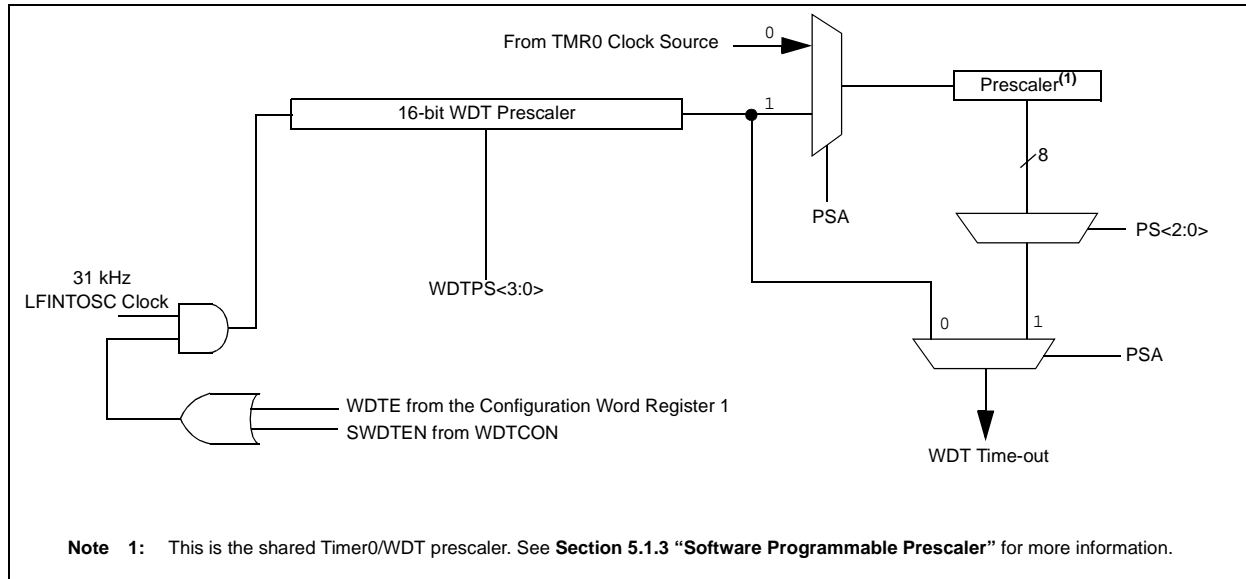
FIGURE 14-9:   WATCHDOG TIMER BLOCK DIAGRAM

    Note   1:   This is the shared Timer0/WDT prescaler. See **Section 5.1.3 "Software Programmable Prescaler"** for more information.

TABLE 14-7:   WDT STATUS

Conditions	WDT
WDTE = 0	Cleared
`CLRWDT` Command	
Oscillator Fail Detected	
Exit Sleep + System Clock = T1OSC, EXTRC, INTOSC, EXTCLK	
Exit Sleep + System Clock = XT, HS, LP	Cleared until the end of OST

15.2   Instruction Descriptions

ADDLW	**Add literal and W**
Syntax:	[*label*] ADDLW k
Operands:	$0 \leq k \leq 255$
Operation:	(W) + k $\rightarrow$ (W)
Status Affected:	C, DC, Z
Description:	The contents of the W register are added to the 8-bit literal 'k' and the result is placed in the W register.

ADDWF	**Add W and f**
Syntax:	[*label*] ADDWF f,d
Operands:	$0 \leq f \leq 127$ d $\in$ [0,1]
Operation:	(W) + (f) $\rightarrow$ (destination)
Status Affected:	C, DC, Z
Description:	Add the contents of the W register with register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

ANDLW	**AND literal with W**
Syntax:	[*label*] ANDLW k
Operands:	$0 \leq k \leq 255$
Operation:	(W) .AND. (k) $\rightarrow$ (W)
Status Affected:	Z
Description:	The contents of W register are AND'ed with the 8-bit literal 'k'. The result is placed in the W register.

ANDWF	**AND W with f**
Syntax:	[*label*] ANDWF f,d
Operands:	$0 \leq f \leq 127$ d $\in$ [0,1]
Operation:	(W) .AND. (f) $\rightarrow$ (destination)
Status Affected:	Z
Description:	AND the W register with register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

BCF	**Bit Clear f**
Syntax:	[*label*] BCF f,b
Operands:	$0 \leq f \leq 127$ $0 \leq b \leq 7$
Operation:	0 $\rightarrow$ (f<b>)
Status Affected:	None
Description:	Bit 'b' in register 'f' is cleared.

BSF	**Bit Set f**
Syntax:	[*label*] BSF f,b
Operands:	$0 \leq f \leq 127$ $0 \leq b \leq 7$
Operation:	1 $\rightarrow$ (f<b>)
Status Affected:	None
Description:	Bit 'b' in register 'f' is set.

BTFSC	**Bit Test f, Skip if Clear**
Syntax:	[*label*] BTFSC f,b
Operands:	$0 \leq f \leq 127$ $0 \leq b \leq 7$
Operation:	skip if (f<b>) = 0
Status Affected:	None
Description:	If bit 'b' in register 'f' is '1', the next instruction is executed. If bit 'b' in register 'f' is '0', the next instruction is discarded, and a NOP is executed instead, making this a 2-cycle instruction.

BTFSS	**Bit Test f, Skip if Set**
Syntax:	[*label*] BTFSS f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b < 7
Operation:	skip if (f<b>) = 1
Status Affected:	None
Description:	If bit 'b' in register 'f' is '0', the next instruction is executed. If bit 'b' is '1', then the next instruction is discarded and a NOP is executed instead, making this a 2-cycle instruction.

CALL	**Call Subroutine**
Syntax:	[*label*] CALL k
Operands:	0 ≤ k ≤ 2047
Operation:	(PC)+ 1→ TOS, k → PC<10:0>, (PCLATH<4:3>) → PC<12:11>
Status Affected:	None
Description:	Call Subroutine. First, return address (PC + 1) is pushed onto the stack. The 11-bit immediate address is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH. CALL is a 2-cycle instruction.

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

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

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

COMF	**Complement f**
Syntax:	[*label*] COMF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	$(\bar{f})$ → (destination)
Status Affected:	Z
Description:	The contents of register 'f' are complemented. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f'.

DECF	**Decrement f**
Syntax:	[*label*] DECF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) - 1 → (destination)
Status Affected:	Z
Description:	Decrement register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

FIGURE 17-20: HFINTOSC FREQUENCY ACCURACY OVER DEVICE V_{DD} AND TEMPERATURE

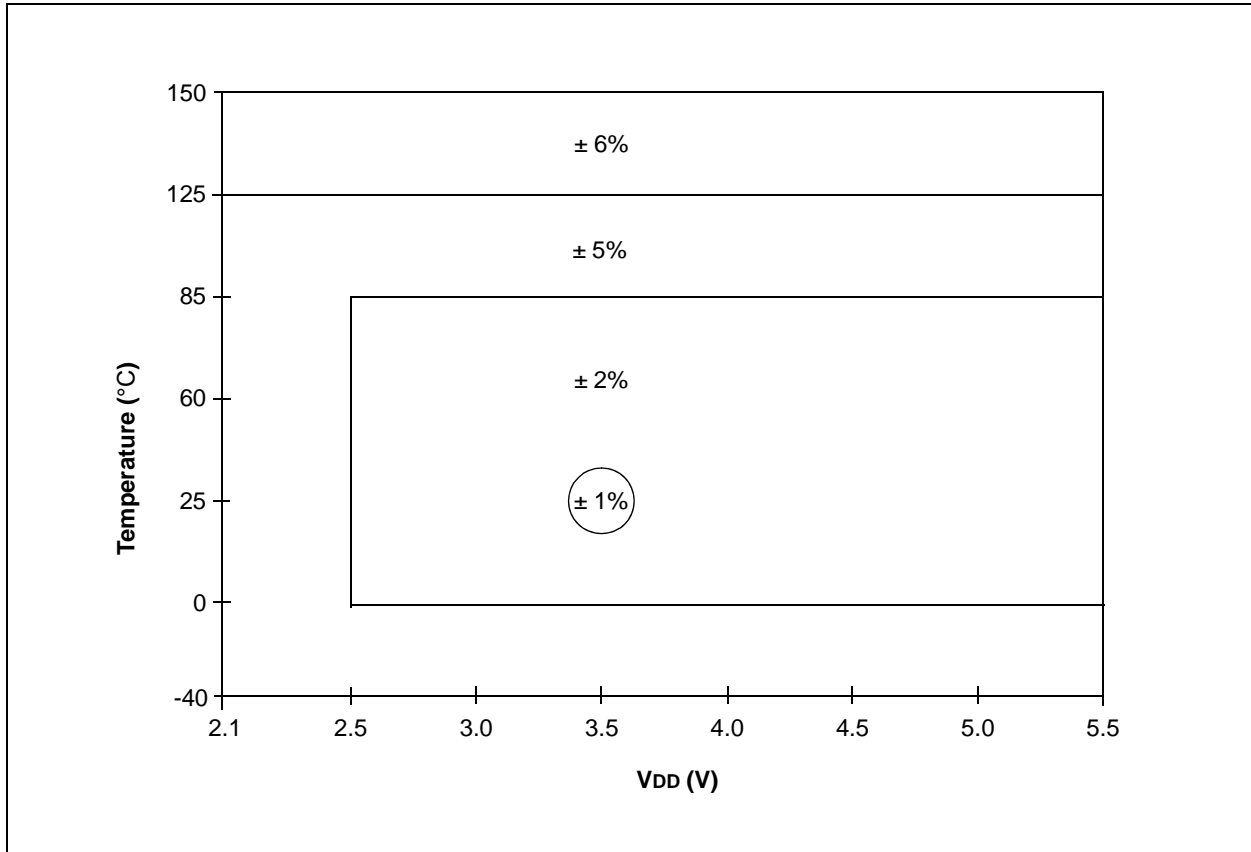**TABLE 17-18:** ADC CLOCK PERIOD (T_{AD}) Vs. DEVICE OPERATING FREQUENCIES (V_{DD} ≥ 3.0V, V_{REF} ≥ 2.5V)

ADC Clock Period (T_{AD})		Device Frequency (F_{OSC})			
ADC Clock Source	**ADCS<2:0>**	**20 MHz**	**8 MHz**	**4 MHz**	**1 MHz**
Fosc/2	`000`	100 ns	250 ns	500 ns	2.0 µs
Fosc/8	`001`	400 ns	1.0 µs	2.0 µs	8.0 µs
Fosc/32	`010`	1.6 µs	4.0 µs	8.0 µs	32.0 µs
Frc	`x11`	2-6 µs	2-6 µs	2-6 µs	2-6 µs

Legend: Shaded cells should not be used for conversions at temperatures above +125°C.

Note 1: T_{AD} must be between 1.6 µs and 6.0 µs.

PIC16F882/883/884/886/887

FIGURE 18-7: TYPICAL I_{DD} vs. V_{DD} OVER F_{OSC} (EXTRC MODE)
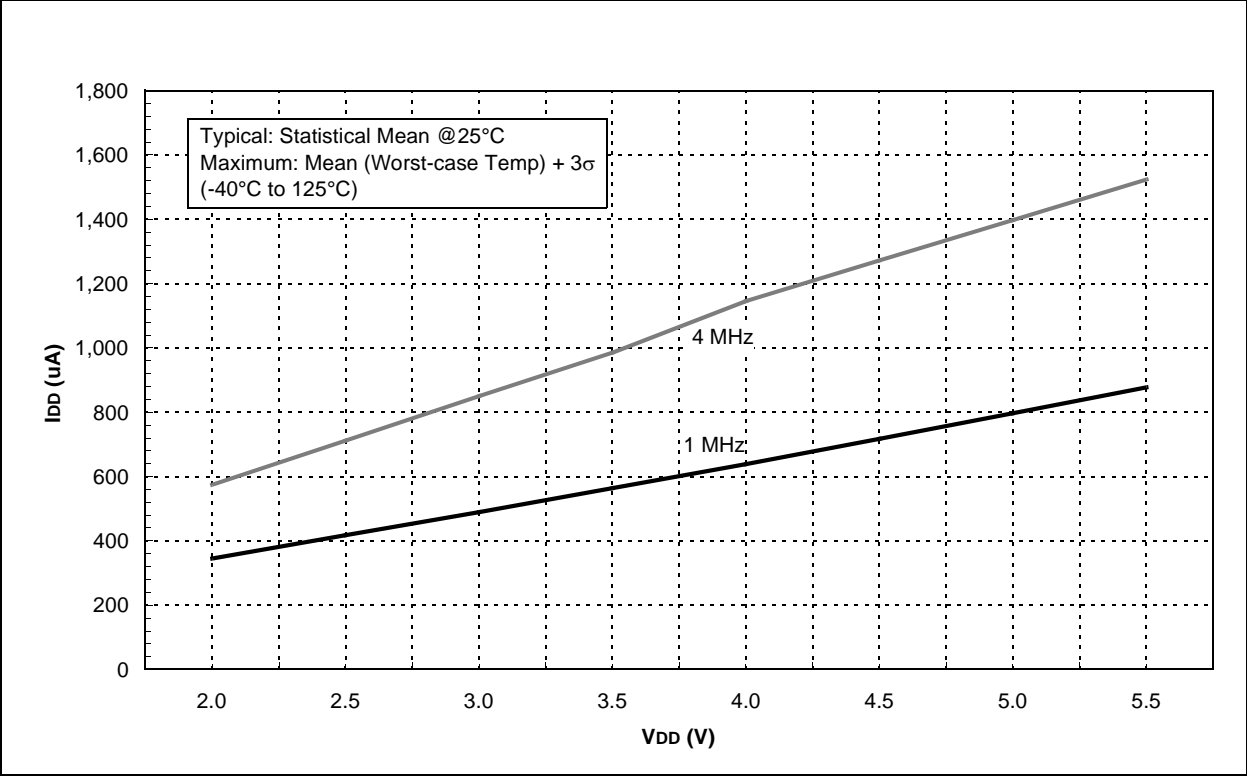

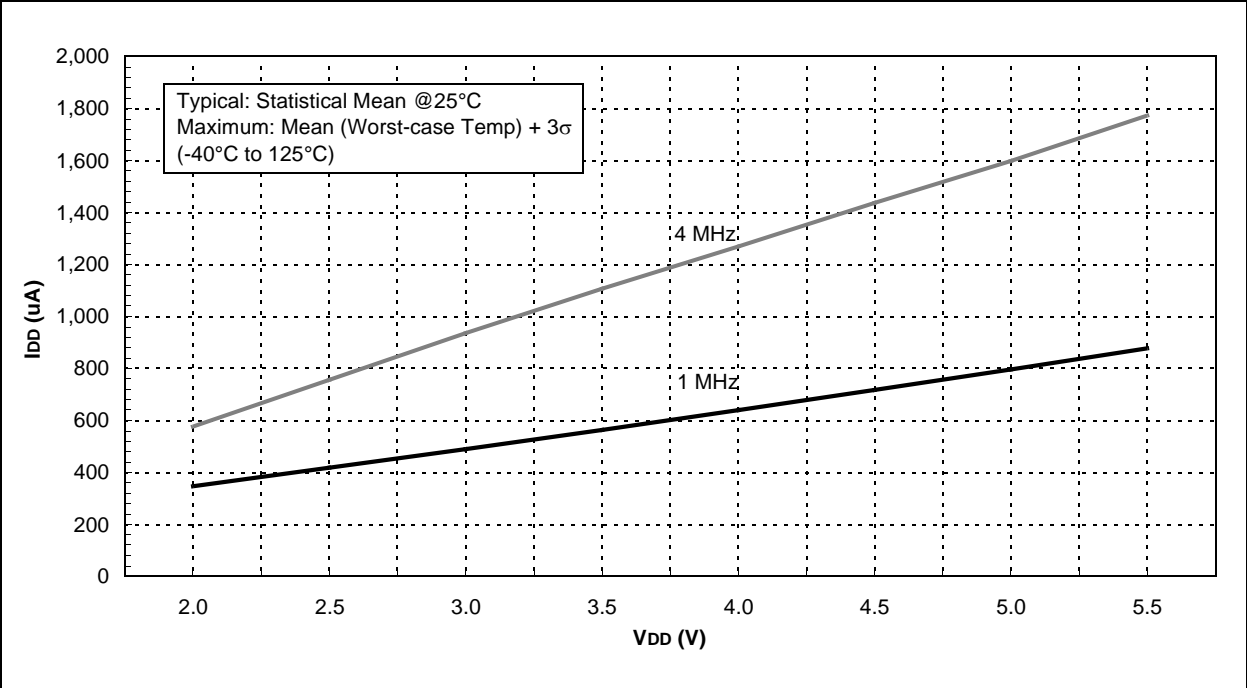**FIGURE 18-8:** MAXIMUM I_{DD} vs. V_{DD} (EXTRC MODE)

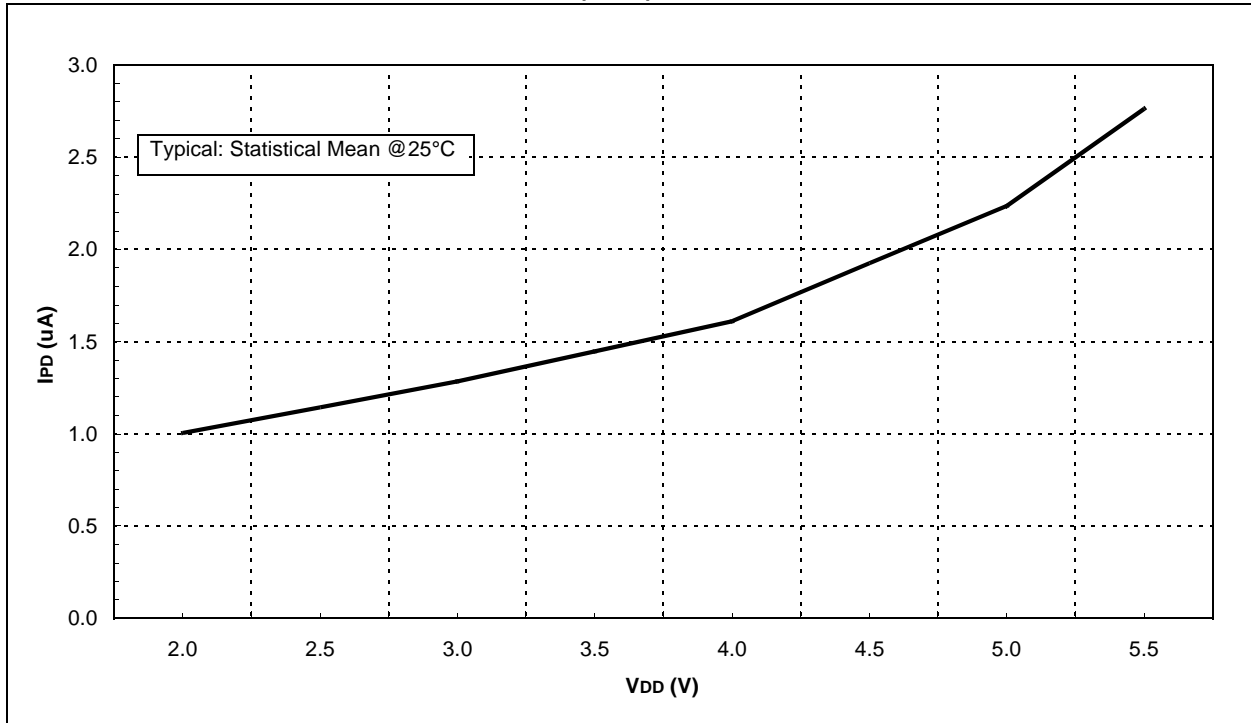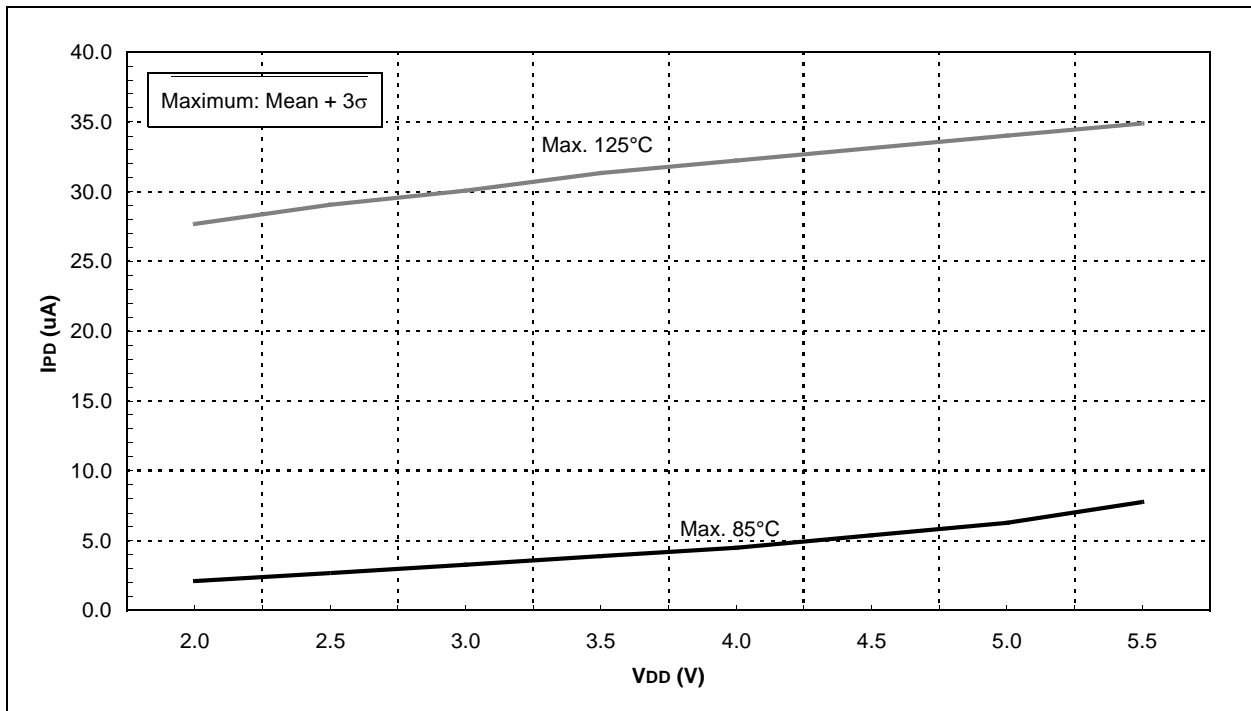**FIGURE 18-17:** TYPICAL WDT I_PD vs. V_DD (25°C)

FIGURE 18-18: MAXIMUM WDT I_PD vs. V_DD OVER TEMPERATURE

PIC16F882/883/884/886/887

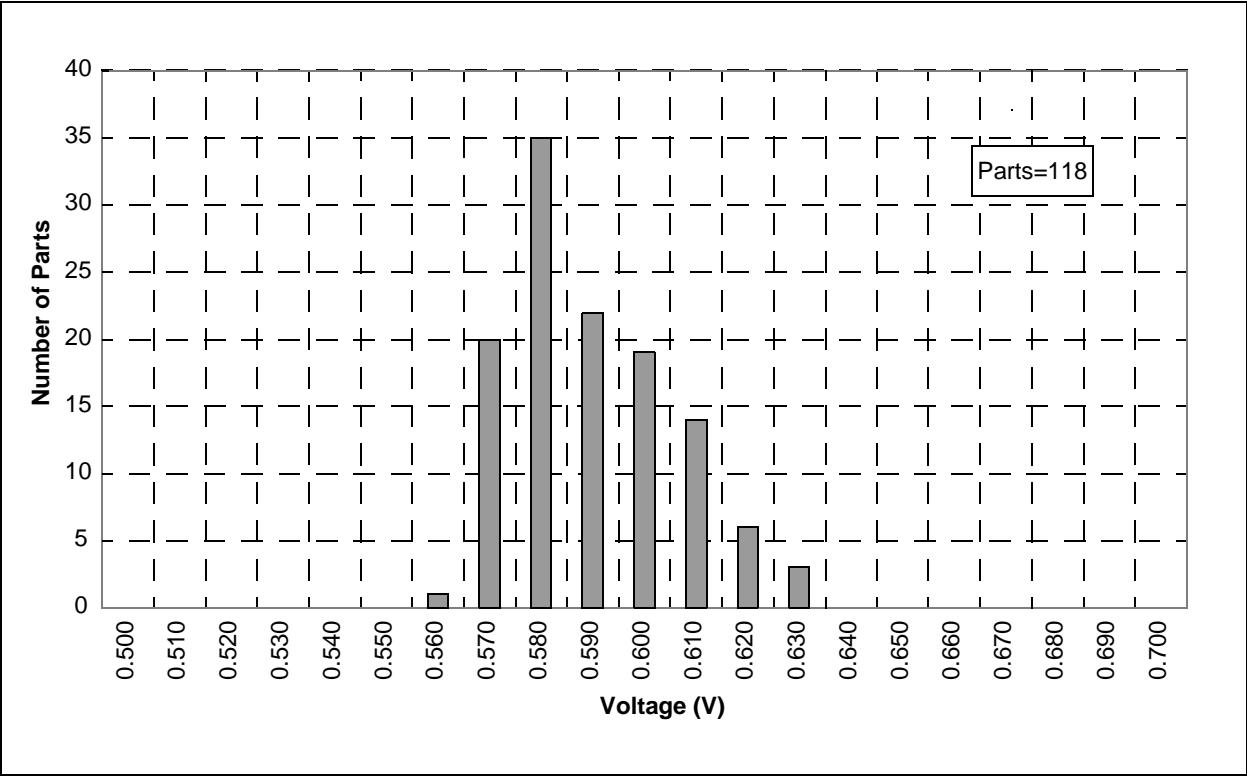**FIGURE 18-47:** **TYPICAL VP6 REFERENCE VOLTAGE DISTRIBUTION (3V, 85°C)**

FIGURE 18-48: **TYPICAL VP6 REFERENCE VOLTAGE DISTRIBUTION (3V, 125°C)**