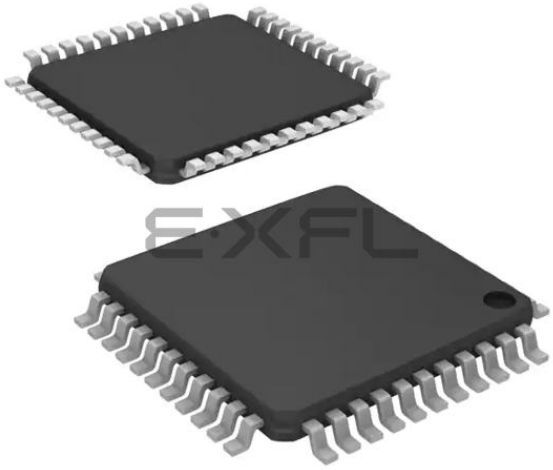


**Welcome to E-XFL.COM**

## 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"

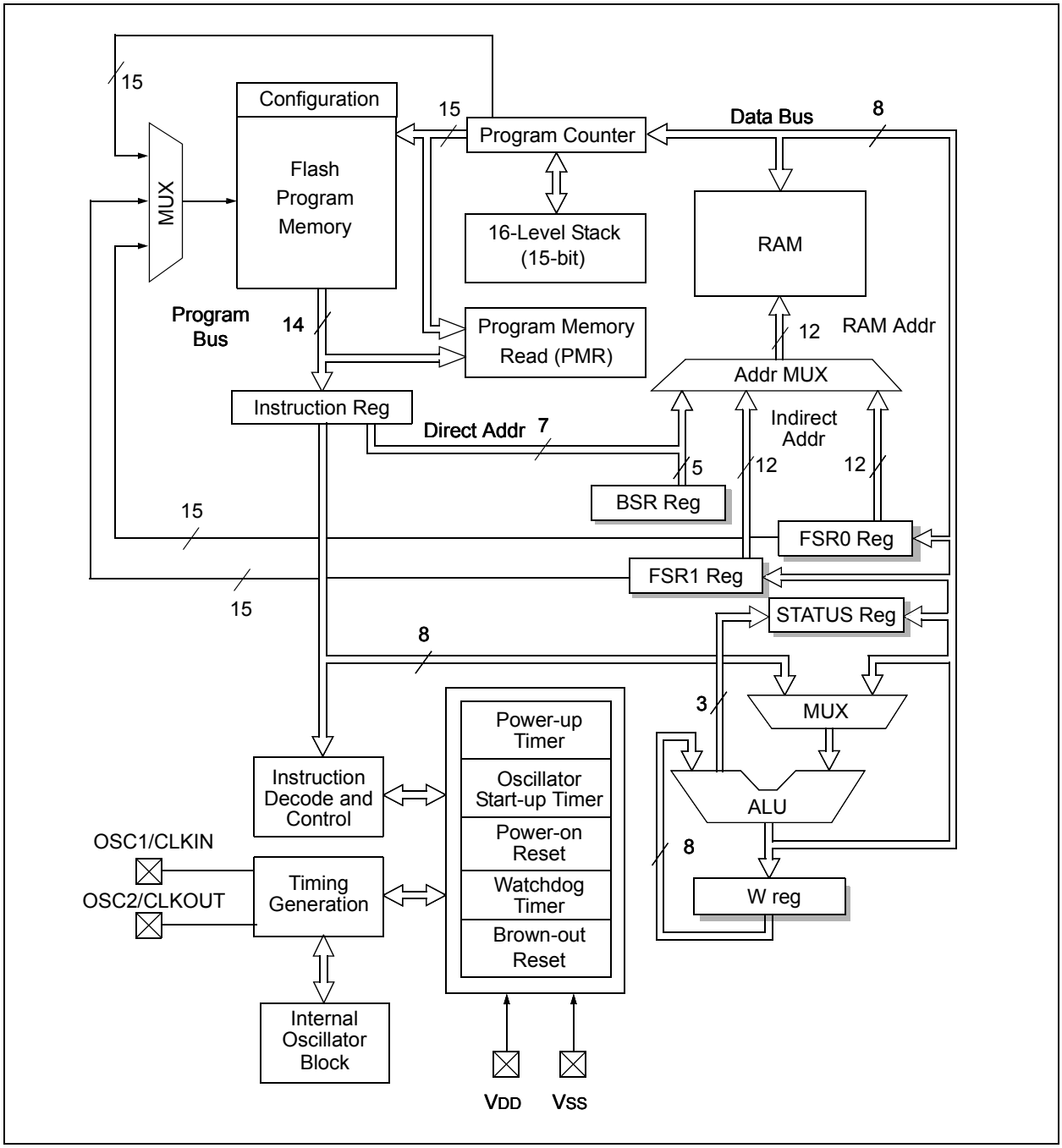
| Details | |
|---|---|
| Product Status | Active |
| Core Processor | PIC |
| Core Size | 8-Bit |
| Speed | 20MHz |
| Connectivity | I²C, LINbus, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, POR, PWM, WDT |
| Number of I/O | 36 |
| Program Memory Size | 14KB (8K x 14) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 512 x 8 |
| Voltage - Supply (Vcc/Vdd) | 1.8V ~ 3.6V |
| Data Converters | A/D 28x10b |
| 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/pic16lf1517-i-pt |

## 2.0 ENHANCED MID-RANGE CPU

This family of devices contain an enhanced mid-range 8-bit CPU core. The CPU has 49 instructions. Interrupt capability includes automatic context saving. The hardware stack is 16 levels deep and has Overflow and Underflow Reset capability. Direct, Indirect, and Relative addressing modes are available. Two File Select Registers (FSRs) provide the ability to read program and data memory.

- Automatic Interrupt Context Saving
- 16-level Stack with Overflow and Underflow
- File Select Registers
- Instruction Set

**FIGURE 2-1: CORE BLOCK DIAGRAM**



#### 3.3.1.1 STATUS Register

The STATUS register, shown in Register 3-1, contains:

- the arithmetic status of the ALU
- the Reset status

The STATUS register can be the destination for any instruction, like any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the $\overline{TO}$ and $\overline{PD}$ bits are not writable. Therefore, the result of an instruction with the STATUS register as destination may be different than intended.

For example, `CLRF STATUS` will clear the upper three bits and set the Z bit. This leaves the STATUS register as '`000u u1uu`' (where u = unchanged).

It is recommended, therefore, that only `BCF`, `BSF`, `SWAPF` and `MOVWF` instructions are used to alter the STATUS register, because these instructions do not affect any Status bits. For other instructions not affecting any Status bits (Refer to **Section 24.0 "Instruction Set Summary"**).

| | |
|---|---|
| **Note 1:** | The C and DC bits operate as $\overline{\text{Borrow}}$ and $\overline{\text{Digit Borrow}}$ out bits, respectively, in subtraction. |

## 3.4 Register Definitions: Status

**REGISTER 3-1: STATUS: STATUS REGISTER**

| U-0 | U-0 | U-0 | R-1/q | R-1/q | R/W-0/u | R/W-0/u | R/W-0/u |
|---|---|---|---|---|---|---|---|
| — | — | — | $\overline{TO}$ | $\overline{PD}$ | Z | DC(1) | C(1) |
| 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 7-5 **Unimplemented:** Read as '0'

bit 4 **$\overline{TO}$:** Time-out bit
- 1 = After power-up, `CLRWDT` instruction or `SLEEP` instruction
- 0 = A WDT time-out occurred

bit 3 **$\overline{PD}$:** Power-down bit
- 1 = After power-up or by the `CLRWDT` instruction
- 0 = By execution of the `SLEEP` instruction

bit 2 **Z:** Zero bit
- 1 = The result of an arithmetic or logic operation is zero
- 0 = The result of an arithmetic or logic operation is not zero

bit 1 **DC:** Digit Carry/$\overline{\text{Digit Borrow}}$ bit(1)
- 1 = A carry-out from the 4th low-order bit of the result occurred
- 0 = No carry-out from the 4th low-order bit of the result

bit 0 **C:** Carry/$\overline{\text{Borrow}}$ bit(1)
- 1 = A carry-out from the Most Significant bit of the result occurred
- 0 = No carry-out from the Most Significant bit of the result occurred

**Note 1:** For $\overline{\text{Borrow}}$, the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand.

**FIGURE 3-6: ACCESSING THE STACK EXAMPLE 2**

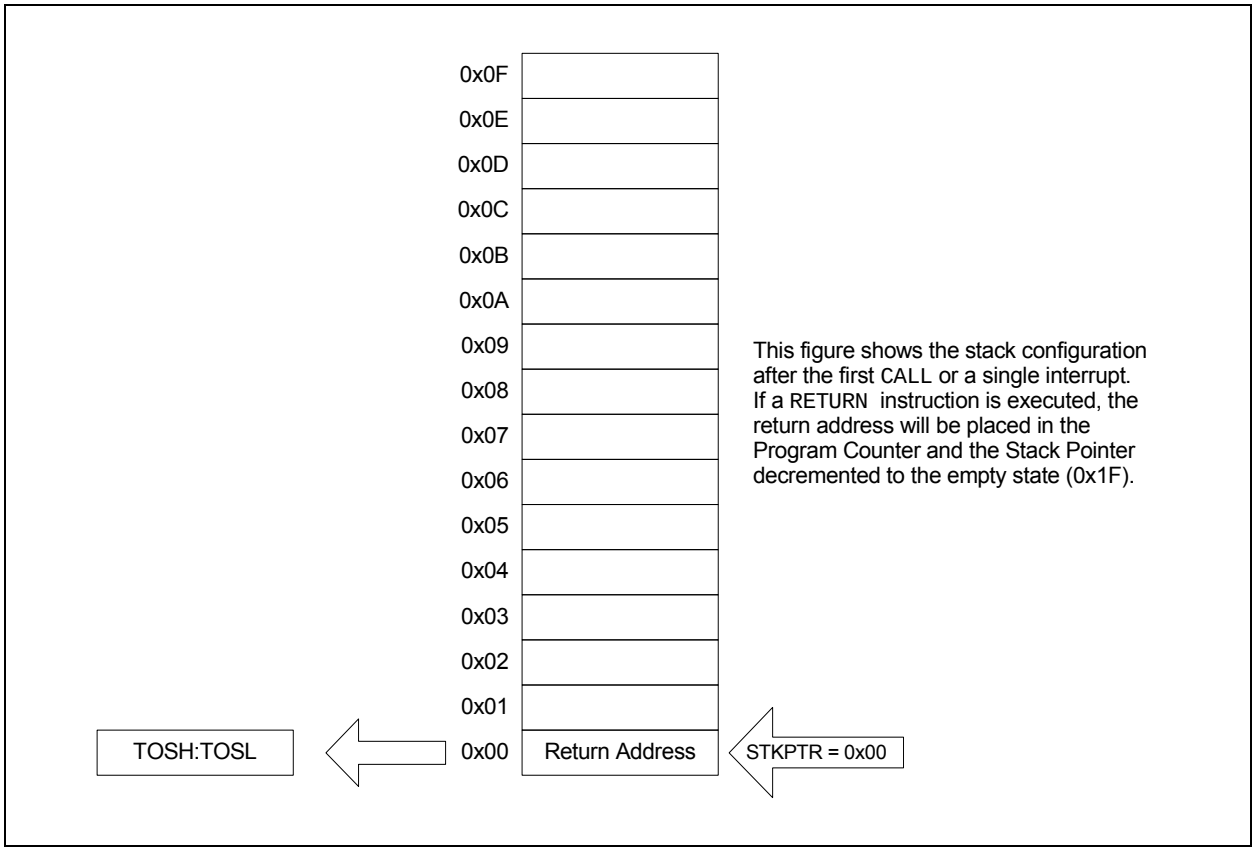



**FIGURE 3-7: ACCESSING THE STACK EXAMPLE 3**

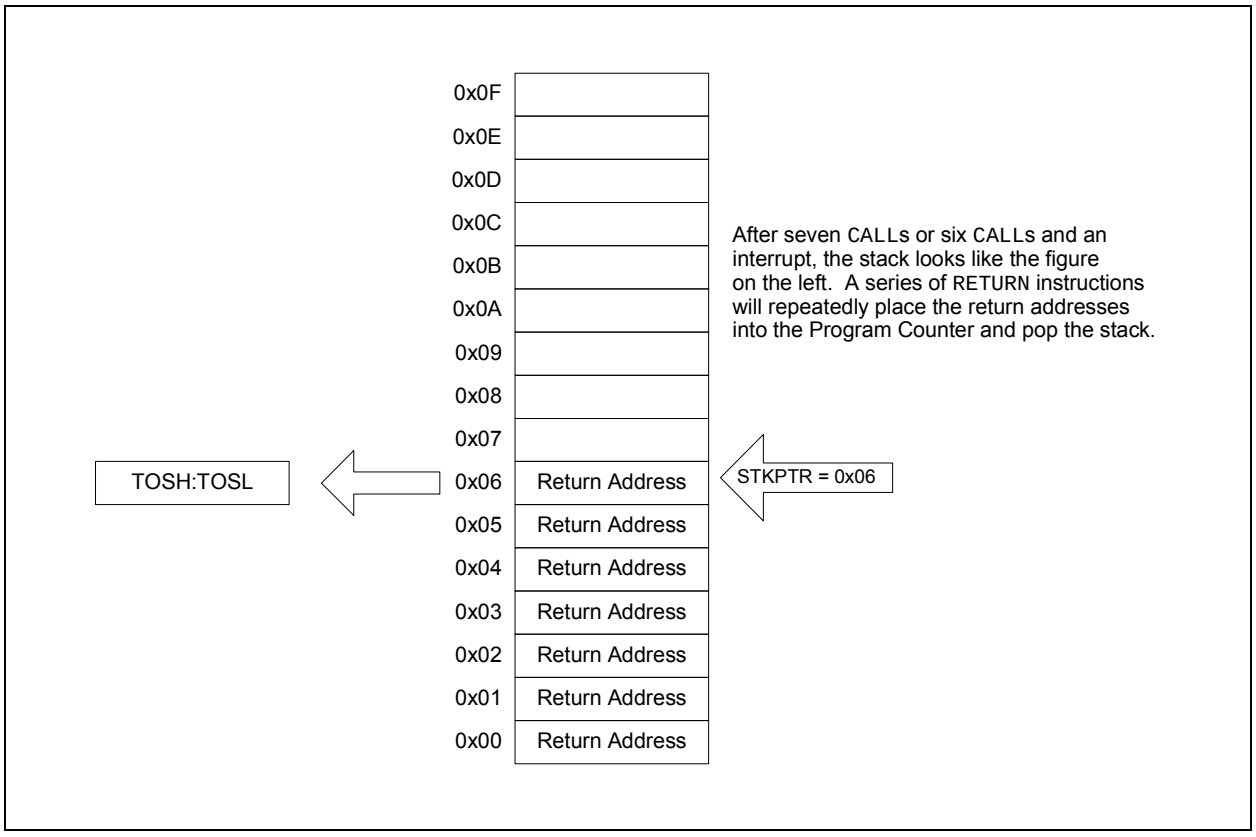
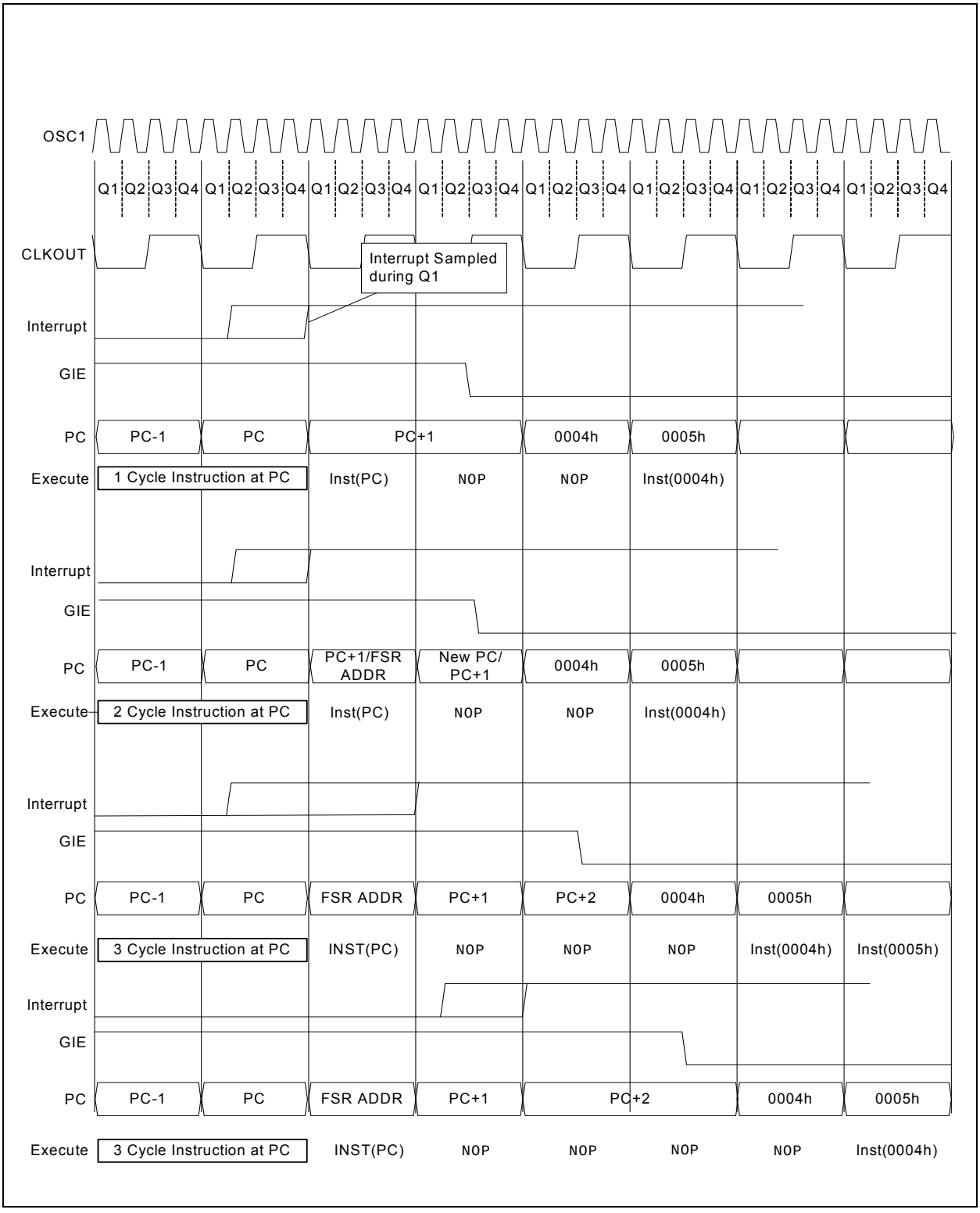


**FIGURE 7-2: INTERRUPT LATENCY**



## 11.0 FLASH PROGRAM MEMORY CONTROL

The Flash program memory is readable and writable during normal operation over the full VDD range. Program memory is indirectly addressed using Special Function Registers (SFRs). The SFRs used to access program memory are:

- PMCON1
- PMCON2
- PMDATL
- PMDATH
- PMADRL
- PMADRH

When accessing the program memory, the PMDATH:PMDATL register pair forms a 2-byte word that holds the 14-bit data for read/write, and the PMADRH:PMADRL register pair forms a 2-byte word that holds the 15-bit address of the program memory location being read.

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 operating voltage range of the device.

The Flash program memory can be protected in two ways; by code protection ($\overline{CP}$ bit in Configuration Words) and write protection (WRT<1:0> bits in Configuration Words).

Code protection ($\overline{CP}$ = 0)**(1)**, disables access, reading and writing, to the Flash program memory via external device programmers. Code protection does not affect the self-write and erase functionality. Code protection can only be reset by a device programmer performing a Bulk Erase to the device, clearing all Flash program memory, Configuration bits and User IDs.

Write protection prohibits self-write and erase to a portion or all of the Flash program memory as defined by the bits WRT<1:0>. Write protection does not affect a device programmers ability to read, write or erase the device.

| | |
|---|---|
| **Note 1:** | Code protection of the entire Flash program memory array is enabled by clearing the $\overline{CP}$ bit of Configuration Words. |

## 11.1 PMADRL and PMADRH Registers

The PMADRH:PMADRL register pair can address 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 PMADRH register and the LSB is written to the PMADRL register.

### 11.1.1 PMCON1 AND PMCON2 REGISTERS

PMCON1 is the control register for Flash program memory accesses.

Control bits RD and WR initiate read and write, respectively. These bits cannot be cleared, only set, in software. They are cleared by 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.

The PMCON2 register is a write-only register. Attempting to read the PMCON2 register will return all '0's.

To enable writes to the program memory, a specific pattern (the unlock sequence), must be written to the PMCON2 register. The required unlock sequence prevents inadvertent writes to the program memory write latches and Flash program memory.

## 11.2 Flash Program Memory Overview

It is important to understand the Flash program memory structure for erase and programming operations. Flash program memory is arranged in rows. A row consists of a fixed number of 14-bit program memory words. A row is the minimum size that can be erased by user software.

After a row has been erased, the user can reprogram all or a portion of this row. Data to be written into the program memory row is written to 14-bit wide data write latches. These write latches are not directly accessible to the user, but may be loaded via sequential writes to the PMDATH:PMDATL register pair.

| | |
|---|---|
| **Note:** | If the user wants to modify only a portion of a previously programmed row, then the contents of the entire row must be read and saved in RAM prior to the erase. Then, new data and retained data can be written into the write latches to reprogram the row of Flash program memory. However, any unprogrammed locations can be written without first erasing the row. In this case, it is not necessary to save and rewrite the other previously programmed locations. |

See Table 11-1 for Erase Row size and the number of write latches for Flash program memory.

**REGISTER 12-21: LATE: PORTE DATA LATCH REGISTER(2)**

| U-0 | U-0 | U-0 | U-0 | U-0 | R/W-x/u | R/W-x/u | R/W-x/u |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | LATE2 | LATE1 | LATE0 |
| 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 7-3 **Unimplemented**: Read as ‘0’

bit 2-0 **LATE<2:0>**: PORTE Output Latch Value bits(1)

**Note 1:** Writes to PORTE are actually written to corresponding LATE register. Reads from PORTE register is the return of actual I/O pin values.

**2:** PIC16(L)F1517/9 only.

**REGISTER 12-22: ANSELE: PORTE ANALOG SELECT REGISTER(2)**

| U-0 | U-0 | U-0 | U-0 | U-0 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | ANSE2 | ANSE1 | ANSE0 |
| 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 7-3 **Unimplemented:** Read as ‘0’

bit 2-0 **ANSE<2:0>**: Analog Select between Analog or Digital Function on pins RE<2:0>, respectively

0 = Digital I/O. Pin is assigned to port or digital special function.

1 = Analog input. Pin is assigned as analog input(1). Digital input buffer disabled.

**Note 1:** When setting a pin to an analog input, the corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.

**2:** PIC16(L)F1517/9 only.

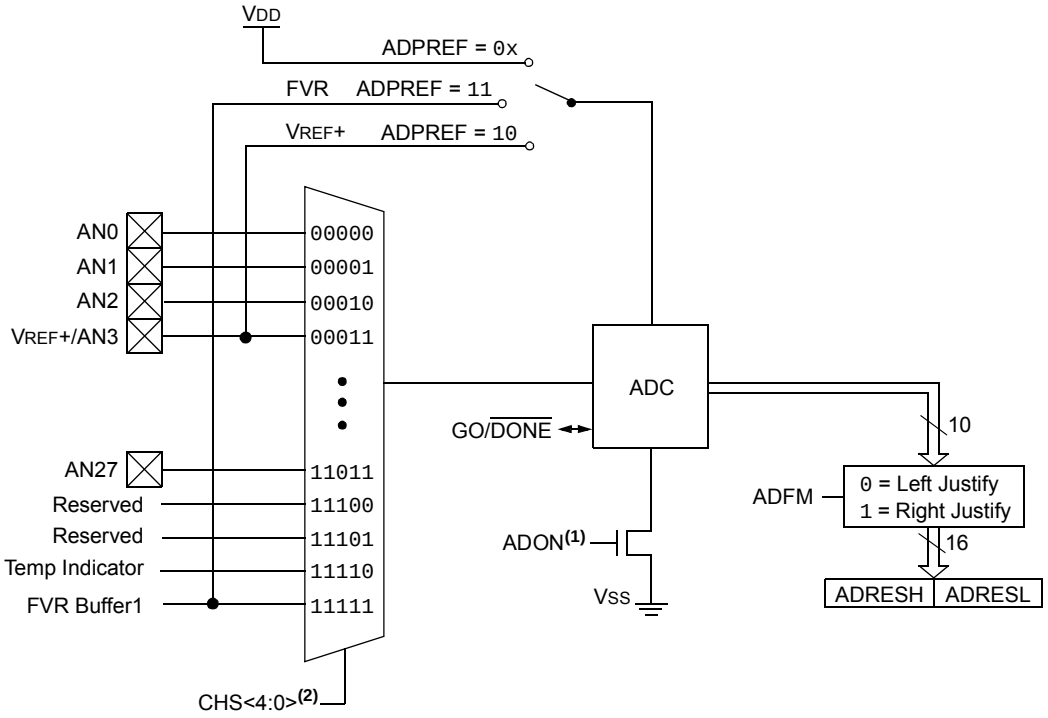
## 16.0 ANALOG-TO-DIGITAL CONVERTER (ADC) MODULE

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

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

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

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




**Note 1:** When ADON = 0, all multiplexer inputs are disconnected.

**2:** See ADCON0 register (Register 16-1) for detailed analog channel selection per device.

### 16.2.6 ADC CONVERSION PROCEDURE

This is an example procedure for using the ADC to perform an Analog-to-Digital conversion:

1. Configure Port:
   - Disable pin output driver (Refer to the TRIS register)
   - Configure pin as analog (Refer to the ANSEL register)
   - Disable weak pull-ups, either globally (Refer to the OPTION_REG register) or individually (Refer to the appropriate WPUx register)
2. Configure the ADC module:
   - Select ADC conversion clock
   - Configure voltage reference
   - Select ADC input channel
   - Turn on ADC module
3. Configure ADC interrupt (optional):
   - Clear ADC interrupt flag
   - Enable ADC interrupt
   - Enable peripheral interrupt
   - Enable global interrupt**(1)**
4. Wait the required acquisition time**(2)**.
5. Start conversion by setting the GO/$\overline{\text{DONE}}$ bit.
6. Wait for ADC conversion to complete by one of the following:
   - Polling the GO/$\overline{\text{DONE}}$ bit
   - Waiting for the ADC interrupt (interrupts enabled)
7. Read ADC Result.
8. Clear the ADC interrupt flag (required if interrupt is enabled).

**Note 1:** The global interrupt can be disabled if the user is attempting to wake up from Sleep and resume in-line code execution.

**2:** Refer to **Section 16.4 "ADC Acquisition Requirements"**.

**EXAMPLE 16-1: ADC CONVERSION**

```
;This code block configures the ADC
;for polling, Vdd and Vss references, Frc
;clock and AN0 input.
;
;Conversion start & polling for completion
; are included.
;
BANKSEL   ADCON1      ;
MOVLW     B'11110000' ;Right justify, Frc
                      ;clock
MOVWF     ADCON1      ;Vdd and Vss Vref
BANKSEL   TRISA       ;
BSF       TRISA,0     ;Set RA0 to input
BANKSEL   ANSEL       ;
BSF       ANSEL,0     ;Set RA0 to analog
BANKSEL   WPUA
BCF       WPUA, 0     ;Disable weak pull-
                      up on RA0
BANKSEL   ADCON0      ;
MOVLW     B'00000001' ;Select channel AN0
MOVWF     ADCON0      ;Turn ADC On
CALL      SampleTime  ;Acquisiton delay
BSF       ADCON0,ADGO ;Start conversion
BTFSC     ADCON0,ADGO ;Is conversion done?
GOTO      $-1         ;No, test again
BANKSEL   ADRESH      ;
MOVF      ADRESH,W    ;Read upper 2 bits
MOVWF     RESULTHI    ;store in GPR space
BANKSEL   ADRESL      ;
MOVF      ADRESL,W    ;Read lower 8 bits
MOVWF     RESULTLO    ;Store in GPR space
```

## 19.5 Register Definitions: Timer2 Control

**REGISTER 19-1: T2CON: TIMER2 CONTROL REGISTER**

| U-0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| — | T2OUTPS<3:0> | | | | TMR2ON | T2CKPS<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 **Unimplemented:** Read as ‘0’

bit 6-3 **T2OUTPS<3:0>:** Timer2 Output Postscaler Select bits
- 1111 = 1:16 Postscaler
- 1110 = 1:15 Postscaler
- 1101 = 1:14 Postscaler
- 1100 = 1:13 Postscaler
- 1011 = 1:12 Postscaler
- 1010 = 1:11 Postscaler
- 1001 = 1:10 Postscaler
- 1000 = 1:9 Postscaler
- 0111 = 1:8 Postscaler
- 0110 = 1:7 Postscaler
- 0101 = 1:6 Postscaler
- 0100 = 1:5 Postscaler
- 0011 = 1:4 Postscaler
- 0010 = 1:3 Postscaler
- 0001 = 1:2 Postscaler
- 0000 = 1:1 Postscaler

bit 2 **TMR2ON:** Timer2 On bit
- 1 = Timer2 is ON
- 0 = Timer2 is OFF

bit 1-0 **T2CKPS<1:0>:** Timer2 Clock Prescale Select bits
- 11 = Prescaler is 64
- 10 = Prescaler is 16
- 01 = Prescaler is 4
- 00 = Prescaler is 1

### 21.5.8 GENERAL CALL ADDRESS SUPPORT

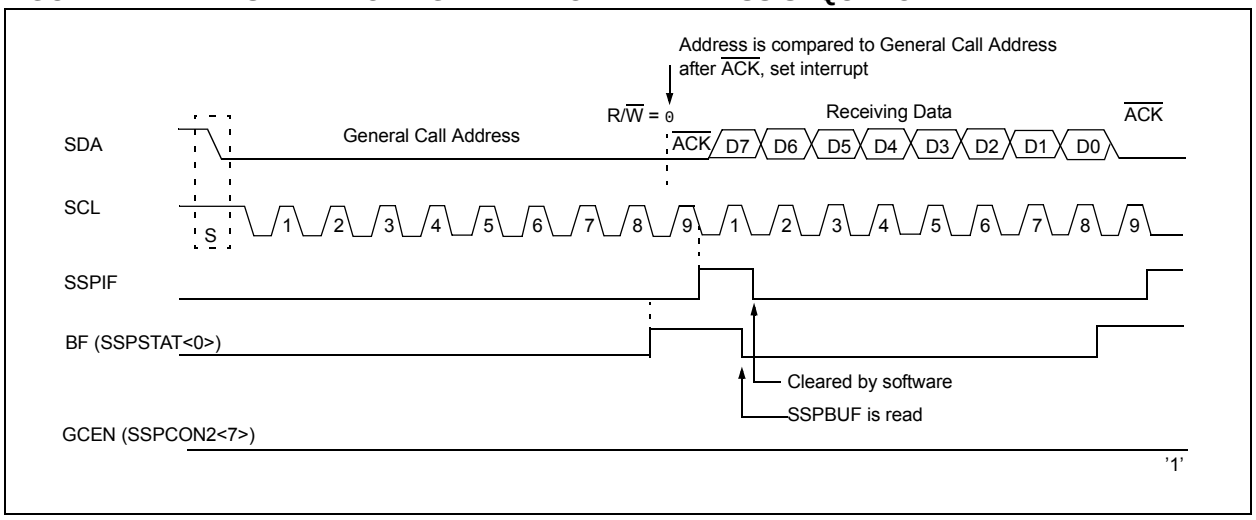
The addressing procedure for the $I^2C$ bus is such that the first byte after the Start condition usually determines which device will be the slave addressed by the master device. The exception is the general call address which can address all devices. When this address is used, all devices should, in theory, respond with an acknowledge.

The general call address is a reserved address in the $I^2C$ protocol, defined as address 0x00. When the GCEN bit of the SSPCON2 register is set, the slave module will automatically $\overline{ACK}$ the reception of this address regardless of the value stored in SSPADD. After the slave clocks in an address of all zeros with the R/$\overline{W}$ bit clear, an interrupt is generated and slave software can read SSPBUF and respond. Figure 21-24 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 SSPCON3 register is set, just as with any other address reception, the slave hardware will stretch the clock after the 8th falling edge of SCL. The slave must then set its ACKDT value and release the clock with communication progressing as it would normally.

**FIGURE 21-24: SLAVE MODE GENERAL CALL ADDRESS SEQUENCE**




### 21.5.9 SSP MASK REGISTER

An SSP Mask (SSPMSK) register (Register 21-7) is available in $I^2C$ Slave mode as a mask for the value held in the SSPSR 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 SSP operation until written with a mask value.

The SSP 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 SSP mask has no effect during the reception of the first (high) byte of the address.

## 21.7 BAUD RATE GENERATOR

The MSSP module has a Baud Rate Generator available for clock generation in both $I^2C$ and SPI Master modes. The Baud Rate Generator (BRG) reload value is placed in the SSPADD register (Register 21-8). When a write occurs to SSPBUF, the Baud Rate Generator will automatically begin counting down.

Once the given operation is complete, the internal clock will automatically stop counting and the clock pin will remain in its last state.

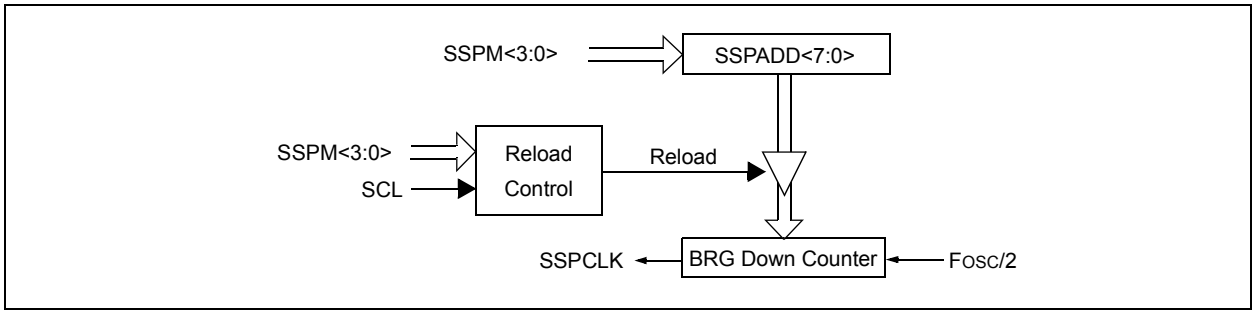
An internal signal “Reload” in Figure 21-40 triggers the value from SSPADD to be loaded into the BRG counter. This occurs twice for each oscillation of the module clock line. The logic dictating when the reload signal is asserted depends on the mode the MSSP is being operated in.

Table 21-2 demonstrates clock rates based on instruction cycles and the BRG value loaded into SSPADD.

**EQUATION 21-1: BRG CLOCK FREQUENCY**

$$F_{CLOCK} = \frac{F_{OSC}}{(SSPxADD + 1)(4)}$$

**FIGURE 21-40: BAUD RATE GENERATOR BLOCK DIAGRAM**



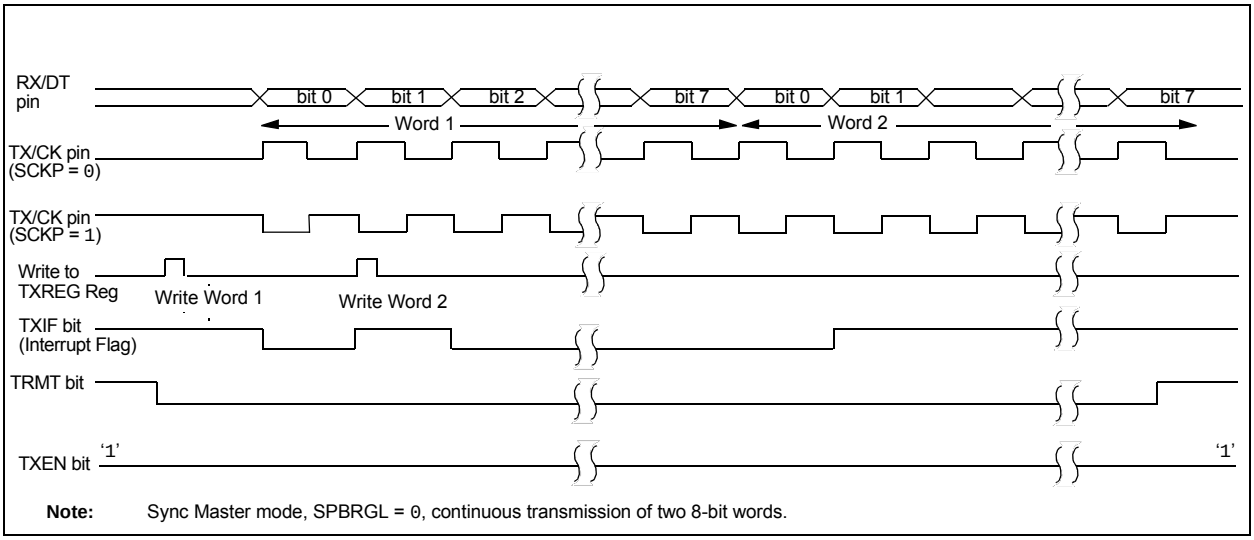

**Note:** Values of 0x00, 0x01 and 0x02 are not valid for SSPADD when used as a Baud Rate Generator for $I^2C$. This is an implementation limitation.

**TABLE 21-2: MSSP CLOCK RATE W/BRG**

| FOSC | FCY | BRG Value | FCLOCK (2 Rollovers of BRG) |
|---|---|---|---|
| 16 MHz | 4 MHz | 09h | 400 kHz(1) |
| 16 MHz | 4 MHz | 0Ch | 308 kHz |
| 16 MHz | 4 MHz | 27h | 100 kHz |
| 4 MHz | 1 MHz | 09h | 100 kHz |

**Note 1:** The $I^2C$ interface does not conform to the 400 kHz $I^2C$ specification (which applies to rates greater than 100 kHz) in all details, but may be used with care where higher rates are required by the application.

**FIGURE 22-10: SYNCHRONOUS TRANSMISSION**




**Note:** Sync Master mode, SPBRGL = 0, continuous transmission of two 8-bit words.

**FIGURE 22-11: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)**

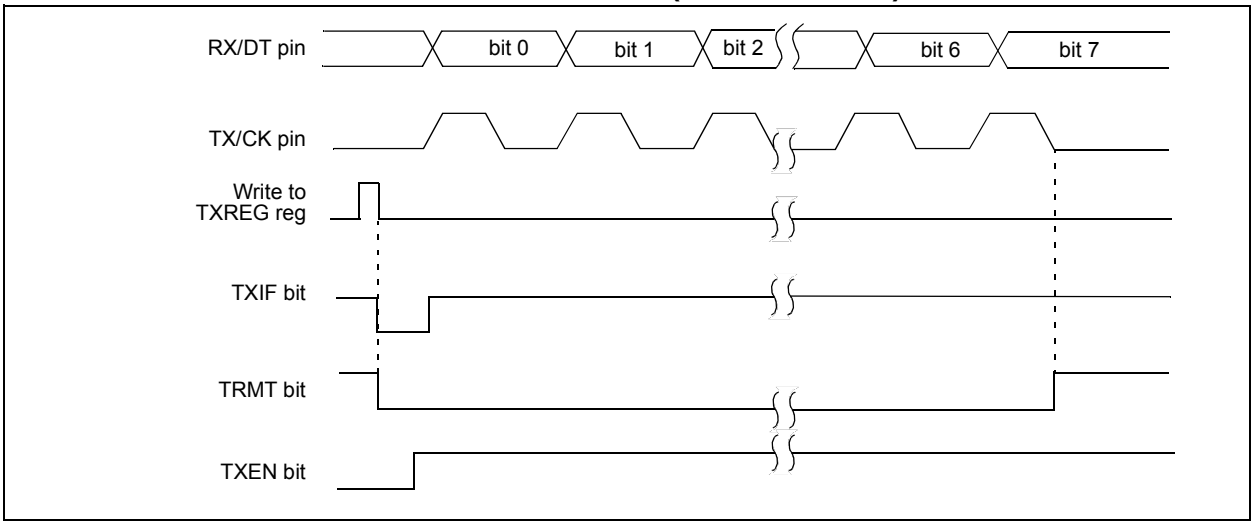



**TABLE 22-5: SUMMARY OF REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| BAUDCON | ABDOVF | RCIDL | — | SCKP | BRG16 | — | WUE | ABDEN | 232 |
| INTCON | GIE | PEIE | TMR0IE | INTE | IOCIE | TMR0IF | INTF | IOCIF | 74 |
| PIE1 | TMR1GIE | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 75 |
| PIR1 | TMR1GIF | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 77 |
| RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 231 |
| SPBRGL | BRG<7:0> | | | | | | | | 233* |
| SPBRGH | BRG<15:8> | | | | | | | | 233* |
| TRISC | TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 114 |
| TXREG | EUSART Transmit Data Register | | | | | | | | 222* |
| TXSTA | CSRC | TX9 | TXEN | SYNC | SENDB | BRGH | TRMT | TX9D | 230 |

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used for synchronous master transmission.

* Page provides register information.

#### 22.5.1.5 Synchronous Master Reception

Data is received at the RX/DT pin. The RX/DT pin output driver is automatically disabled when the EUSART is configured for synchronous master receive operation.

In Synchronous mode, reception is enabled by setting either the Single Receive Enable bit (SREN of the RCSTA register) or the Continuous Receive Enable bit (CREN of the RCSTA register).

When SREN is set and CREN is clear, only as many clock cycles are generated as there are data bits in a single character. The SREN bit is automatically cleared at the completion of one character. When CREN is set, clocks are continuously generated until CREN is cleared. If CREN is cleared in the middle of a character the CK clock stops immediately and the partial character is discarded. If SREN and CREN are both set, then SREN is cleared at the completion of the first character and CREN takes precedence.

To initiate reception, set either SREN or CREN. Data is sampled at the RX/DT pin on the trailing edge of the TX/CK clock pin and is shifted into the Receive Shift Register (RSR). When a complete character is received into the RSR, the RCIF bit is set and the character is automatically transferred to the two character receive FIFO. The Least Significant eight bits of the top character in the receive FIFO are available in RCREG. The RCIF bit remains set as long as there are unread characters in the receive FIFO.

| | |
|---|---|
| **Note:** | If the RX/DT function is on an analog pin, the corresponding ANSEL bit must be cleared for the receiver to function. |

#### 22.5.1.6 Slave Clock

Synchronous data transfers use a separate clock line, which is synchronous with the data. A device configured as a slave receives the clock on the TX/CK line. The TX/CK pin output driver is automatically disabled when the device is configured for synchronous slave 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 data bit is transferred for each clock cycle. Only as many clock cycles should be received as there are data bits.

| | |
|---|---|
| **Note:** | If the device is configured as a slave and the TX/CK function is on an analog pin, the corresponding ANSEL bit must be cleared. |

#### 22.5.1.7 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 RCREG is read to access the FIFO. When this happens the OERR bit of the RCSTA register is set. Previous data in the FIFO will not be overwritten. The two characters in the FIFO buffer can be read, however, no additional characters will be received until the error is cleared. The OERR bit can only be cleared by clearing the overrun condition. If the overrun error occurred when the SREN bit is set and CREN is clear then the error is cleared by reading RCREG. If the overrun occurred when the CREN bit is set then the error condition is cleared by either clearing the CREN bit of the RCSTA register or by clearing the SPEN bit which resets the EUSART.

#### 22.5.1.8 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 9-bits into the RSR for each character received. The RX9D bit of the RCSTA register is the 9th, 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.

#### 22.5.1.9 Synchronous Master Reception Setup:

1. Initialize the SPBRGH, SPBRGL register pair for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Clear the ANSEL bit for the RX pin (if applicable).
3. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
4. Ensure bits CREN and SREN are clear.
5. If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
6. If 9-bit reception is desired, set bit RX9.
7. Start reception by setting the SREN bit or for continuous reception, set the CREN bit.
8. Interrupt flag bit RCIF will be set when reception of a character is complete. An interrupt will be generated if the enable bit RCIE was set.
9. Read the RCSTA register to get the 9th bit (if enabled) and determine if any error occurred during reception.
10. Read the 8-bit received data by reading the RCREG register.
11. If an overrun error occurs, clear the error by either clearing the CREN bit of the RCSTA register or by clearing the SPEN bit which resets the EUSART.

## 25.2 Standard Operating Conditions

The standard operating conditions for any device are defined as:

Operating Voltage: $V_{DDMIN} \leq V_{DD} \leq V_{DDMAX}$
Operating Temperature: $T_{A\_MIN} \leq T_A \leq T_{A\_MAX}$

**$V_{DD}$ — Operating Supply Voltage(1)**

PIC16LF1516/7/8/9

$V_{DDMIN}$ ($F_{OSC} \leq$ 16 MHz) ..... +1.8V
$V_{DDMIN}$ (16 MHz $\leq F_{OSC} \leq$ 20 MHz) ..... +2.5V
$V_{DDMAX}$ ..... +3.6V

PIC16F1516/7/8/9

$V_{DDMIN}$ ($F_{OSC} \leq$ 16 MHz) ..... +2.3V
$V_{DDMIN}$ (16 MHz $\leq F_{OSC} \leq$ 20 MHz) ..... +2.5V
$V_{DDMAX}$ ..... +5.5V

**$T_A$ — Operating Ambient Temperature Range**

Industrial Temperature

$T_{A\_MIN}$ ..... -40°C
$T_{A\_MAX}$ ..... +85°C

Extended Temperature

$T_{A\_MIN}$ ..... -40°C
$T_{A\_MAX}$ ..... +125°C

**Note 1:** See Parameter D001, DC Characteristics: Supply Voltage.

## 25.3 DC Characteristics: Supply Voltage

| PIC16LF1516/7/8/9 | | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial<br>-40°C ≤ TA ≤ +125°C for extended | | | | |
|---|---|---|---|---|---|---|---|
| **PIC16F1516/7/8/9** | | | **Standard Operating Conditions (unless otherwise stated)**<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial<br>-40°C ≤ TA ≤ +125°C for extended | | | | |
| **Param. No.** | **Sym.** | **Characteristic** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| D001 | VDD | **Supply Voltage (VDDMIN, VDDMAX)** | | | | | |
| | | | 1.8<br>2.5 | —<br>— | 3.6<br>3.6 | V<br>V | FOSC ≤ 16 MHz:<br>FOSC ≤ 20 MHz |
| D001 | | | 2.3<br>2.5 | —<br>— | 5.5<br>5.5 | V<br>V | FOSC ≤ 16 MHz:<br>FOSC ≤ 20 MHz |
| D002* | VDR | **RAM Data Retention Voltage(1)** | | | | | |
| | | | 1.5 | — | — | V | Device in Sleep mode |
| D002* | | | 1.7 | — | — | V | Device in Sleep mode |
| D002A* | VPOR | **Power-on Reset Release Voltage** | — | 1.6 | — | V | |
| D002B* | VPORR | **Power-on Reset Rearm Voltage** | | | | | |
| | | | — | 0.8 | — | V | |
| D002B* | | | — | 1.42 | — | V | |
| D003 | VADFVR | **Fixed Voltage Reference Voltage for ADC** | -8 | — | 6 | % | 1.024V, VDD ≥ 2.5V<br>2.048V, VDD ≥ 2.5V<br>4.096V, VDD ≥ 4.75V |
| D004* | SVDD | **VDD Rise Rate to ensure internal Power-on Reset signal** | 0.05 | — | — | V/ms | See Section **6.1 "Power-On Reset (POR)"** for details. |

* 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:** This is the limit to which VDD can be lowered in Sleep mode without losing RAM data.

## 25.4 DC Characteristics: Supply Current ($I_{DD}$)

| PIC16LF1516/7/8/9 | | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial<br>-40°C ≤ TA ≤ +125°C for extended | | | | |
|---|---|---|---|---|---|---|---|
| **PIC16F1516/7/8/9** | | | **Standard Operating Conditions (unless otherwise stated)**<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial<br>-40°C ≤ TA ≤ +125°C for extended | | | | |
| **Param No.** | **Device Characteristics** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** | |
| | | | | | | **VDD** | **Note** |
| | **Supply Current ($I_{DD}$)(1, 2, 3)** | | | | | | |
| D010 | | — | 8.0 | 14 | µA | 1.8 | FOSC = 32 kHz<br>LP Oscillator<br>-40°C ≤ TA ≤ +85°C |
| | | — | 12.0 | 31 | µA | 3.0 | |
| D010 | | — | 11 | 28 | µA | 2.3 | FOSC = 32 kHz<br>LP Oscillator<br>-40°C ≤ TA ≤ +85°C |
| | | — | 13 | 38 | µA | 3.0 | |
| | | — | 14 | 45 | µA | 5.0 | |
| D011 | | — | 60 | 95 | µA | 1.8 | FOSC = 1 MHz<br>XT Oscillator |
| | | — | 110 | 180 | µA | 3.0 | |
| D011 | | — | 92 | 170 | µA | 2.3 | FOSC = 1 MHz<br>XT Oscillator |
| | | — | 140 | 230 | µA | 3.0 | |
| | | — | 170 | 350 | µA | 5.0 | |
| D012 | | — | 150 | 240 | µA | 1.8 | FOSC = 4 MHz<br>XT Oscillator |
| | | — | 260 | 430 | µA | 3.0 | |
| D012 | | — | 190 | 450 | µA | 2.3 | FOSC = 4 MHz<br>XT Oscillator |
| | | — | 310 | 500 | µA | 3.0 | |
| | | — | 370 | 650 | µA | 5.0 | |
| D013 | | — | 25 | 31 | µA | 1.8 | FOSC = 500 kHz<br>EC Oscillator<br>Low-Power mode |
| | | — | 35 | 50 | µA | 3.0 | |
| D013 | | — | 25 | 40 | µA | 2.3 | FOSC = 500 kHz<br>EC Oscillator<br>Low-Power mode |
| | | — | 35 | 55 | µA | 3.0 | |
| | | — | 40 | 60 | µA | 5.0 | |
| D014 | | — | 120 | 210 | µA | 1.8 | FOSC = 4 MHz<br>EC Oscillator<br>Medium-Power mode |
| | | — | 210 | 380 | µA | 3.0 | |
| D014 | | — | 160 | 250 | µA | 2.3 | FOSC = 4 MHz<br>EC Oscillator<br>Medium-Power mode |
| | | — | 260 | 380 | µA | 3.0 | |
| | | — | 330 | 480 | µA | 5.0 | |

* 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:** The test conditions for all IDD measurements in active operation mode are: OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD; $\overline{MCLR}$ = VDD; WDT disabled.

**2:** The supply current is mainly a function of the operating voltage and frequency. Other factors, such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature, also have an impact on the current consumption.

**3:** 0.1 µF capacitor on VCAP pin, PIC16F1516/7/8/9 only.

**4:** For RC oscillator configurations, current through REXT is not included. The current through the resistor can be extended by the formula IR = VDD/2REXT (mA) with REXT in kΩ.

## 25.6 DC Characteristics: I/O Ports (Continued)

| DC CHARACTERISTICS | | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial<br>-40°C ≤ TA ≤ +125°C for extended | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| | VOH | **Output High Voltage(4)** | | | | | |
| D090 | | I/O ports | VDD - 0.7 | — | — | V | IOH = 3.5 mA, VDD = 5V<br>IOH = 3 mA, VDD = 3.3V<br>IOH = 1 mA, VDD = 1.8V |
| | | **Capacitive Loading Specs on Output Pins** | | | | | |
| D101* | COSC2 | OSC2 pin | — | — | 15 | pF | In XT, HS and LP modes when external clock is used to drive OSC1 |
| D101A* | CIO | All I/O pins | — | — | 50 | pF | |
| | | **VCAP Capacitor Charging** | | | | | |
| D102* | | Charging current | — | — | 200 | μA | |
| D102A* | | Source/Sink capability when charging complete | — | — | 0.0 | mA | |

* 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.

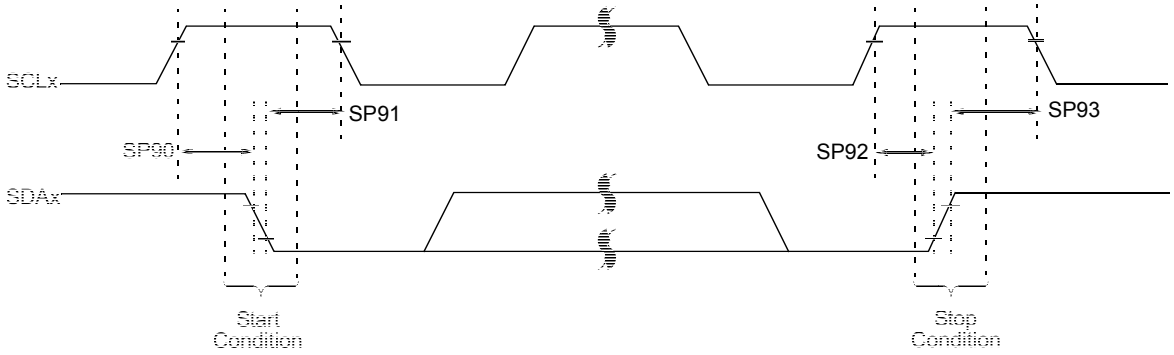
**Note 1:** In RC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended to use an external clock in RC mode.

**2:** Negative current is defined as current sourced by the pin.

**3:** The leakage current on the $\overline{\text{MCLR}}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

**4:** Including OSC2 in CLKOUT mode.

**FIGURE 25-20: I²C BUS START/STOP BITS TIMING**




**Note**: Refer to Figure 25-5 for load conditions.

**TABLE 25-13: I²C BUS START/STOP BITS REQUIREMENTS**

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

* These parameters are characterized but not tested.

**FIGURE 26-1: IDD, LP OSCILLATOR MODE, FOSC = 32 kHz, PIC16LF1516/7/8/9 ONLY**

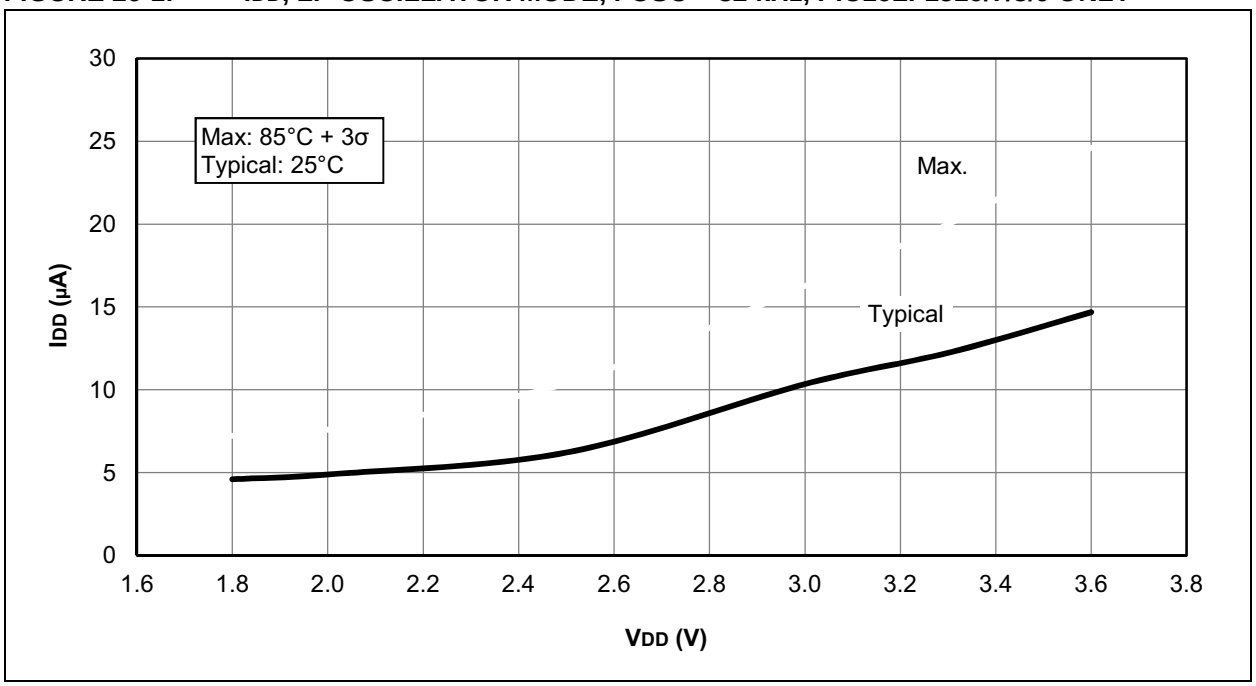



**FIGURE 26-2: IDD, LP OSCILLATOR MODE, FOSC = 32 kHz, PIC16F1516/7/8/9 ONLY**

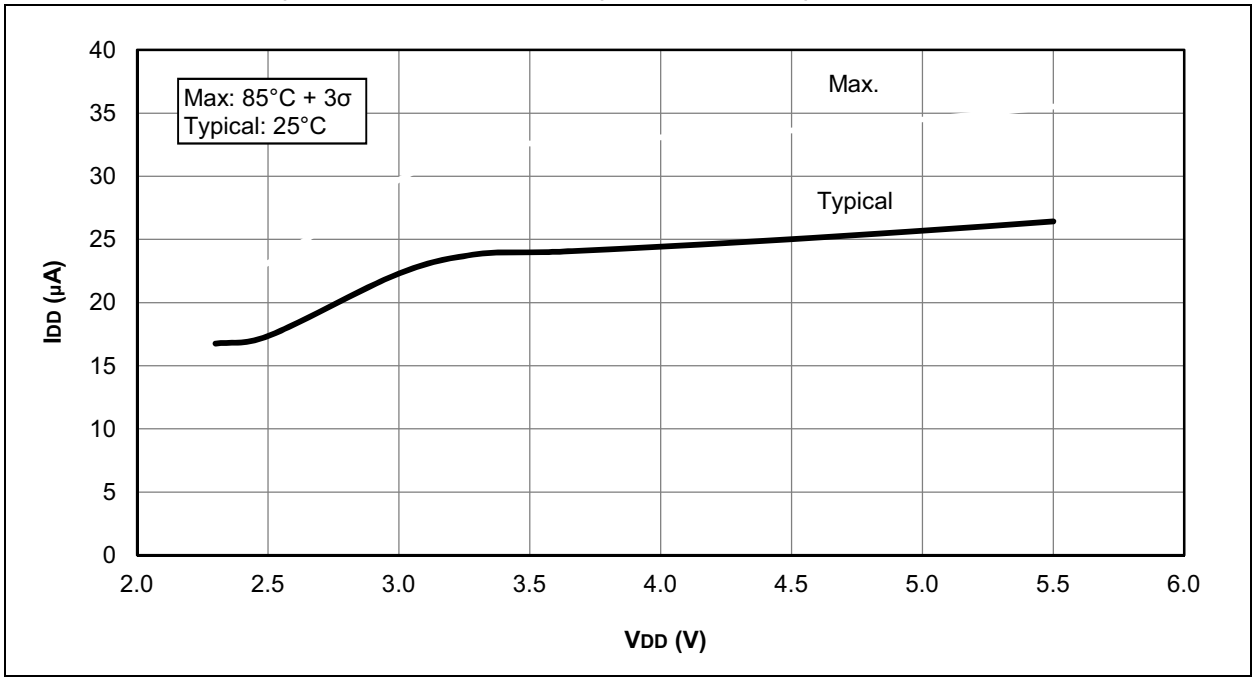


**FIGURE 26-3: IDD TYPICAL, XT AND EXTRC OSCILLATOR, PIC16LF1516/7/8/9 ONLY**

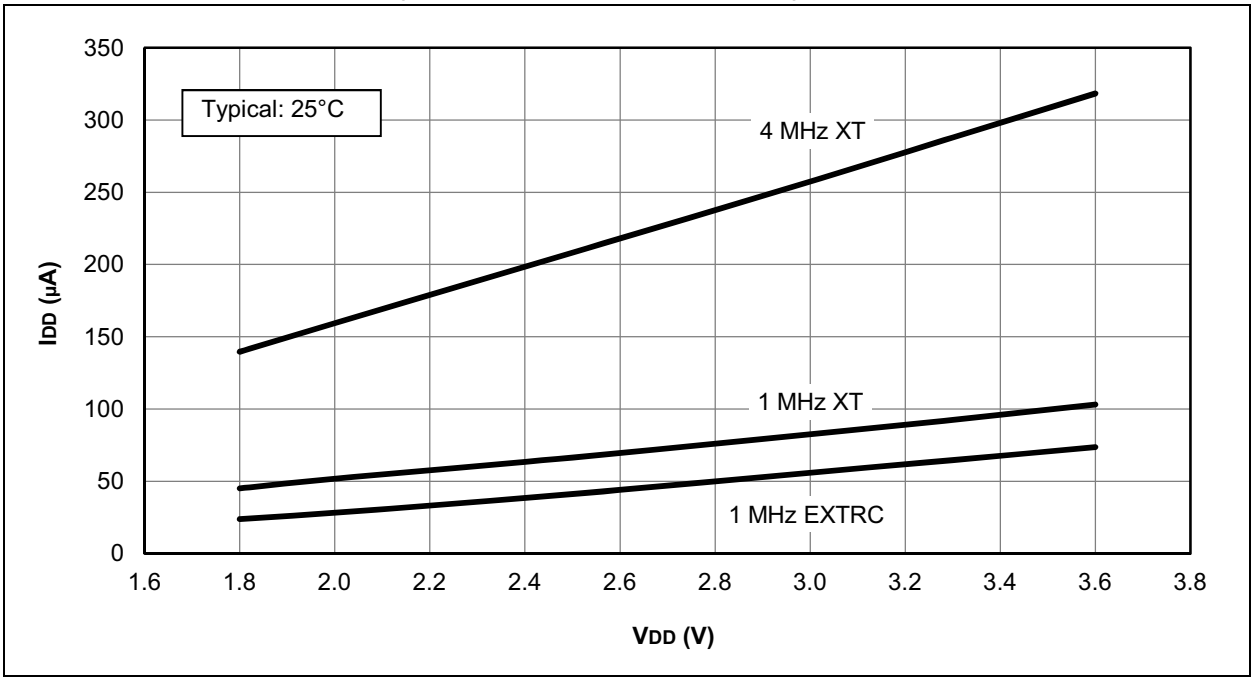



**FIGURE 26-4: IDD MAXIMUM, XT AND EXTRC OSCILLATOR, PIC16LF1516/7/8/9 ONLY**