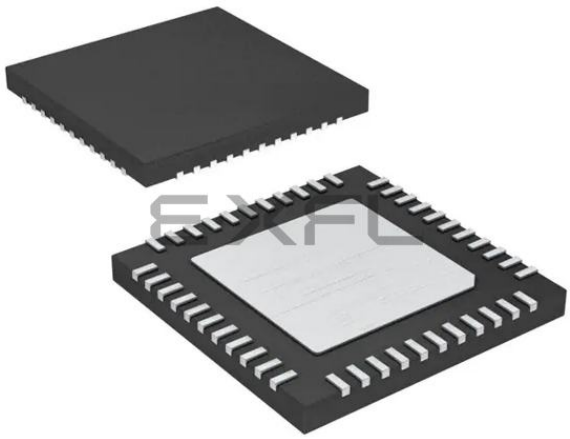


**Welcome to E-XFL.COM**

## 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, LCD, POR, PWM, WDT |
| Number of I/O | 36 |
| Program Memory Size | 28KB (16K x 14) |
| Program Memory Type | FLASH |
| EEPROM Size | 256 x 8 |
| RAM Size | 1K x 8 |
| Voltage - Supply (Vcc/Vdd) | 1.8V ~ 5.5V |
| Data Converters | A/D 14x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 125°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 44-VQFN Exposed Pad |
| Supplier Device Package | 44-QFN (8x8) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic16f1939-e-ml |

**TABLE 1-2: PIC16(L)F1938/9 PINOUT DESCRIPTION (CONTINUED)**

| Name | Function | Input Type | Output Type | Description |
|---|---|---|---|---|
| RA6/OSC2/CLKOUT/VCAP(2)/ SEG1 | RA6 | TTL | CMOS | General purpose I/O. |
| | OSC2 | — | XTAL | Crystal/Resonator (LP, XT, HS modes). |
| | CLKOUT | — | CMOS | FOSC/4 output. |
| | VCAP | Power | Power | Filter capacitor for Voltage Regulator (PIC16F1938/9 only). |
| | SEG1 | — | AN | LCD Analog output. |
| RA7/OSC1/CLKIN/SEG2 | RA7 | TTL | CMOS | General purpose I/O. |
| | OSC1 | XTAL | — | Crystal/Resonator (LP, XT, HS modes). |
| | CLKIN | CMOS | — | External clock input (EC mode). |
| | SEG2 | — | AN | LCD Analog output. |
| RB0/AN12/CPS0/CCP4/SRI/INT/ SEG0 | RB0 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN12 | AN | — | A/D Channel 12 input. |
| | CPS0 | AN | — | Capacitive sensing input 0. |
| | CCP4 | ST | CMOS | Capture/Compare/PWM4. |
| | SRI | — | ST | SR Latch input. |
| | INT | ST | — | External interrupt. |
| | SEG0 | — | AN | LCD analog output. |
| RB1/AN10/C12IN3-/CPS1/P1C/ VLCD1 | RB1 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN10 | AN | — | A/D Channel 10 input. |
| | C12IN3- | AN | — | Comparator C1 or C2 negative input. |
| | CPS1 | AN | — | Capacitive sensing input 1. |
| | P1C | — | CMOS | PWM output. |
| | VLCD1 | AN | — | LCD analog input. |
| RB2/AN8/CPS2/P1B/VLCD2 | RB2 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN8 | AN | — | A/D Channel 8 input. |
| | CPS2 | AN | — | Capacitive sensing input 2. |
| | P1B | — | CMOS | PWM output. |
| | VLCD2 | AN | — | LCD analog input. |
| RB3/AN9/C12IN2-/CPS3/ CCP2(1)/P2A(1)/VLCD3 | RB3 | TTL | CMOS | General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up. |
| | AN9 | AN | — | A/D Channel 9 input. |
| | C12IN2- | AN | — | Comparator C1 or C2 negative input. |
| | CPS3 | AN | — | Capacitive sensing input 3. |
| | CCP2 | ST | CMOS | Capture/Compare/PWM2. |
| | P2A | — | CMOS | PWM output. |
| | VLCD3 | AN | — | LCD analog input. |

**Legend:** AN = Analog input or output CMOS = CMOS compatible input or output OD = Open Drain
TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels XTAL = Crystal
HV = High Voltage $I^2C$™ = Schmitt Trigger input with $I^2C$ levels

**Note 1:** Pin function is selectable via the APFCON register.
**2:** PIC16F1938/9 devices only.
**3:** PIC16(L)F1938 devices only.
**4:** PORTD is available on PIC16(L)F1939 devices only.
**5:** RE<2:0> are available on PIC16(L)F1939 devices only.

**TABLE 1-2: PIC16(L)F1938/9 PINOUT DESCRIPTION (CONTINUED)**

| Name | Function | Input Type | Output Type | Description |
|---|---|---|---|---|
| RC4/SDI/SDA/T1G(1)/SEG11 | RC4 | ST | CMOS | General purpose I/O. |
| | SDI | ST | — | SPI data input. |
| | SDA | $I^2C$ | OD | $I^2C$™ data input/output. |
| | T1G | ST | — | Timer1 Gate input. |
| | SEG11 | — | AN | LCD Analog output. |
| RC5/SDO/SEG10 | RC5 | ST | CMOS | General purpose I/O. |
| | SDO | — | CMOS | SPI data output. |
| | SEG10 | — | AN | LCD Analog output. |
| RC6/TX/CK/CCP3/P3A/SEG9 | RC6 | ST | CMOS | General purpose I/O. |
| | TX | — | CMOS | USART asynchronous transmit. |
| | CK | ST | CMOS | USART synchronous clock. |
| | CCP3 | ST | CMOS | Capture/Compare/PWM3. |
| | P3A | — | CMOS | PWM output. |
| | SEG9 | — | AN | LCD Analog output. |
| RC7/RX/DT/P3B/SEG8 | RC7 | ST | CMOS | General purpose I/O. |
| | RX | ST | — | USART asynchronous input. |
| | DT | ST | CMOS | USART synchronous data. |
| | P3B | — | CMOS | PWM output. |
| | SEG8 | — | AN | LCD Analog output. |
| RD0(4)/CPS8/COM3 | RD0 | ST | CMOS | General purpose I/O. |
| | CPS8 | AN | — | Capacitive sensing input 8. |
| | COM3 | — | AN | LCD analog output. |
| RD1(4)/CPS9/CCP4 | RD1 | ST | CMOS | General purpose I/O. |
| | CPS9 | AN | — | Capacitive sensing input 9. |
| | CCP4 | ST | CMOS | Capture/Compare/PWM4. |
| RD2(4)/CPS10/P2B | RD2 | ST | CMOS | General purpose I/O. |
| | CPS10 | AN | — | Capacitive sensing input 10. |
| | P2B | — | CMOS | PWM output. |
| RD3(4)/CPS11/P2C/SEG16 | RD3 | ST | CMOS | General purpose I/O. |
| | CPS11 | AN | — | Capacitive sensing input 11. |
| | P2C | — | CMOS | PWM output. |
| | SEG16 | — | AN | LCD analog output. |
| RD4(4)/CPS12/P2D/SEG17 | RD4 | ST | CMOS | General purpose I/O. |
| | CPS12 | AN | — | Capacitive sensing input 12. |
| | P2D | — | CMOS | PWM output. |
| | SEG17 | — | AN | LCD analog output. |
| RD5(4)/CPS13/P1B/SEG18 | RD5 | ST | CMOS | General purpose I/O. |
| | CPS13 | AN | — | Capacitive sensing input 13. |
| | P1D | — | CMOS | PWM output. |
| | SEG18 | — | AN | LCD analog output. |

**Legend:** AN = Analog input or output CMOS = CMOS compatible input or output OD = Open Drain
TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels XTAL = Crystal
HV = High Voltage $I^2C$™ = Schmitt Trigger input with $I^2C$ levels

**Note 1:** Pin function is selectable via the APFCON register.
**2:** PIC16F1938/9 devices only.
**3:** PIC16(L)F1938 devices only.
**4:** PORTD is available on PIC16(L)F1939 devices only.
**5:** RE<2:0> are available on PIC16(L)F1939 devices only.

**TABLE 3-10: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 15 (Continued)** | | | | | | | | | | | |
| 7A6h | LCDDATA6 | SEG7 COM2 | SEG6 COM2 | SEG5 COM2 | SEG4 COM2 | SEG3 COM2 | SEG2 COM2 | SEG1 COM2 | SEG0 COM2 | xxxx xxxx | uuuu uuuu |
| 7A7h | LCDDATA7 | SEG15 COM2 | SEG14 COM2 | SEG13 COM2 | SEG12 COM2 | SEG11 COM2 | SEG10 COM2 | SEG9 COM2 | SEG8 COM2 | xxxx xxxx | uuuu uuuu |
| 7A8h | LCDDATA8(3) | SEG23 COM2 | SEG22 COM2 | SEG21 COM2 | SEG20 COM2 | SEG19 COM2 | SEG18 COM2 | SEG17 COM2 | SEG16 COM2 | xxxx xxxx | uuuu uuuu |
| 7A9h | LCDDATA9 | SEG7 COM3 | SEG6 COM3 | SEG5 COM3 | SEG4 COM3 | SEG3 COM3 | SEG2 COM3 | SEG1 COM3 | SEG0 COM3 | xxxx xxxx | uuuu uuuu |
| 7AAh | LCDDATA10 | SEG15 COM3 | SEG14 COM3 | SEG13 COM3 | SEG12 COM3 | SEG11 COM3 | SEG10 COM3 | SEG9 COM3 | SEG8 COM3 | xxxx xxxx | uuuu uuuu |
| 7ABh | LCDDATA11(3) | SEG23 COM3 | SEG22 COM3 | SEG21 COM3 | SEG20 COM3 | SEG19 COM3 | SEG18 COM3 | SEG17 COM3 | SEG16 COM3 | xxxx xxxx | uuuu uuuu |
| 7ACh — 7EFh | — | Unimplemented | | | | | | | | — | — |

**Legend:** x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations are unimplemented, read as '0'.

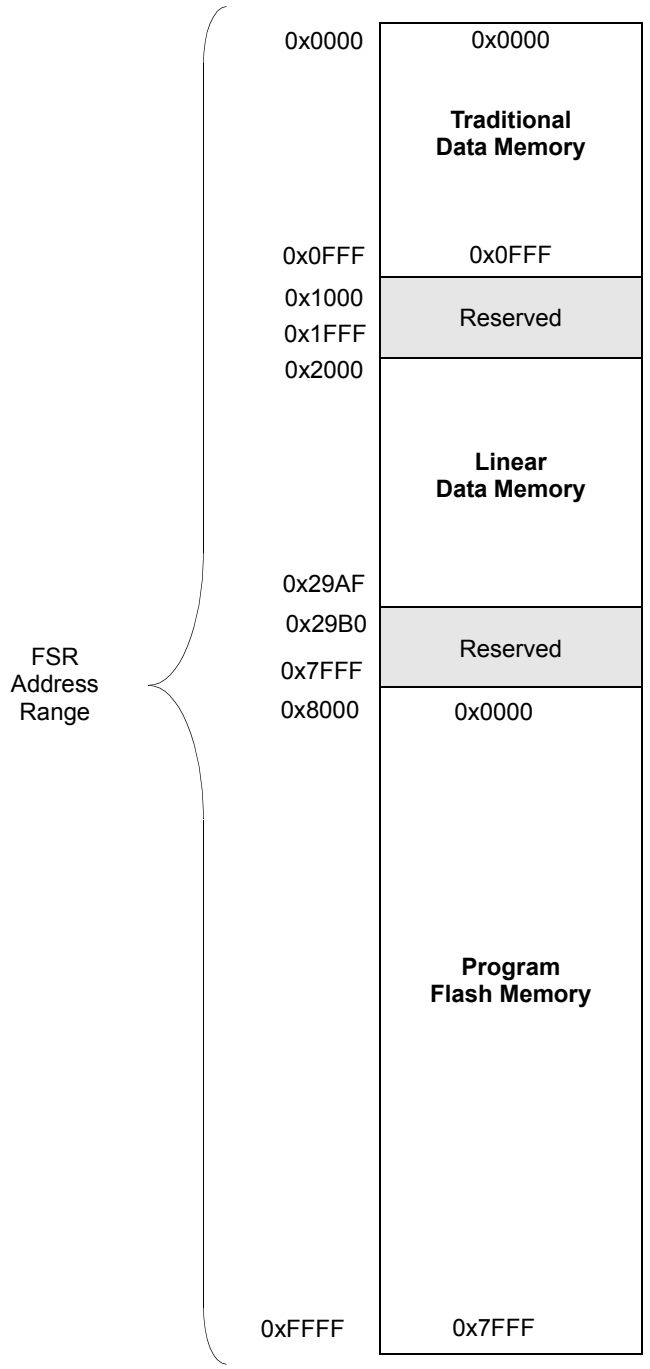
**Note 1:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<14:8>, whose contents are transferred to the upper byte of the program counter.

**2:** These registers can be addressed from any bank.

**3:** These registers/bits are not implemented on PIC16(L)F1938 devices, read as '0'.

**4:** Unimplemented, read as '1'.

**FIGURE 3-8: INDIRECT ADDRESSING**




**Note:** Not all memory regions are completely implemented. Consult device memory tables for memory limits.

## 6.1 Power-on Reset (POR)

The POR circuit holds the device in Reset until VDD has reached an acceptable level for minimum operation. Slow rising VDD, fast operating speeds or analog performance may require greater than minimum VDD. The PWRT, BOR or MCLR features can be used to extend the start-up period until all device operation conditions have been met.

### 6.1.1 POWER-UP TIMER (PWRT)

The Power-up Timer provides a nominal 64 ms time-out on POR or Brown-out Reset.

The device is held in Reset as long as PWRT is active. The PWRT delay allows additional time for the VDD to rise to an acceptable level. The Power-up Timer is enabled by clearing the PWRTE bit in Configuration Words.

The Power-up Timer starts after the release of the POR and BOR.

For additional information, refer to Application Note AN607, *"Power-up Trouble Shooting"* (DS00607).

## 6.2 Brown-Out Reset (BOR)

The BOR circuit holds the device in Reset when VDD reaches a selectable minimum level. Between the POR and BOR, complete voltage range coverage for execution protection can be implemented.

The Brown-out Reset module has four operating modes controlled by the BOREN<1:0> bits in Configuration Words. The four operating modes are:

- BOR is always on
- BOR is off when in Sleep
- BOR is controlled by software
- BOR is always off

Refer to Table 6-3 for more information.

The Brown-out Reset voltage level is selectable by configuring the BORV bit in Configuration Words.

A VDD noise rejection filter prevents the BOR from triggering on small events. If VDD falls below VBOR for a duration greater than parameter TBORDC, the device will reset. See Figure 6-2 for more information.

**TABLE 6-1: BOR OPERATING MODES**

| BOREN<1:0> | SBOREN | Device Mode | BOR Mode | Device Operation upon release of POR | Device Operation upon wake- up from Sleep |
|---|---|---|---|---|---|
| 11 | X | X | Active | Waits for BOR ready(1) | |
| 10 | X | Awake | Active | Waits for BOR ready | |
| | | Sleep | Disabled | | |
| 01 | 1 | X | Active | Begins immediately | |
| | 0 | | Disabled | Begins immediately | |
| 00 | X | X | Disabled | Begins immediately | |

**Note 1:** In these specific cases, "Release of POR" and "Wake-up from Sleep", there is no delay in start-up. The BOR ready flag, (BORRDY = 1), will be set before the CPU is ready to execute instructions because the BOR circuit is forced on by the BOREN<1:0> bits.

### 6.2.1 BOR IS ALWAYS ON

When the BOREN bits of Configuration Words are programmed to '11', the BOR is always on. The device start-up will be delayed until the BOR is ready and VDD is higher than the BOR threshold.

BOR protection is active during Sleep. The BOR does not delay wake-up from Sleep.

### 6.2.2 BOR IS OFF IN SLEEP

When the BOREN bits of Configuration Words are programmed to '10', the BOR is on, except in Sleep. The device start-up will be delayed until the BOR is ready and VDD is higher than the BOR threshold.

BOR protection is not active during Sleep. The device wake-up will be delayed until the BOR is ready.

### 6.2.3 BOR CONTROLLED BY SOFTWARE

When the BOREN bits of Configuration Words are programmed to '01', the BOR is controlled by the SBOREN bit of the BORCON register. The device start-up is not delayed by the BOR ready condition or the VDD level.

BOR protection begins as soon as the BOR circuit is ready. The status of the BOR circuit is reflected in the BORRDY bit of the BORCON register.

BOR protection is unchanged by Sleep.

**REGISTER 12-16: LATD: PORTD DATA LATCH REGISTER**

| 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 |
|---|---|---|---|---|---|---|---|
| LATD7 | LATD6 | LATD5 | LATD4 | LATD3 | LATD2 | LATD1 | LATD0 |
| 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 **LATD<7:0>**: PORTD Output Latch Value bits(1,2)

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

**2:** PORTD implemented on PIC16(L)F1939 devices only.

**REGISTER 12-17: ANSELD: PORTD ANALOG SELECT REGISTER(2)**

| R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 |
|---|---|---|---|---|---|---|---|
| ANSD7 | ANSD6 | ANSD5 | ANSD4 | ANSD3 | ANSD2 | ANSD1 | ANSD0 |
| 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 **ANSD<7:0>**: Analog Select between Analog or Digital Function on Pins RD<7:0>, 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:** ANSELD register is not implemented on the PIC16(L)F1938. Read as '0'.

**3:** PORTD implemented on PIC16(L)F1939 devices only.

**TABLE 12-11: SUMMARY OF REGISTERS ASSOCIATED WITH PORTD(1)**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| ANSELD | ANSD7 | ANSD6 | ANSD5 | ANSD4 | ANSD3 | ANSD2 | ANSD1 | ANSD0 | 138 |
| CCPxCON | PxM<1:0> | | DCxB<1:0> | | CCPxM<3:0> | | | | 228 |
| CPSCON0 | CPSON | CPSRM | — | — | CPSRNG<1:0> | | CPSOUT | T0XCS | 321 |
| CPSCON1 | — | — | — | — | CPSCH<3:0> | | | | 322 |
| LATD | LATD7 | LATD6 | LATD5 | LATD4 | LATD3 | LATD2 | LATD1 | LATD0 | 138 |
| LCDCON | LCDEN | SLPEN | WERR | — | CS<1:0> | | LMUX<1:0> | | 327 |
| LCDSE2 | SE23 | SE22 | SE21 | SE20 | SE19 | SE18 | SE17 | SE16 | 331 |
| PORTD | RD7 | RD6 | RD5 | RD4 | RD3 | RD2 | RD1 | RD0 | 137 |
| TRISD | TRISD7 | TRISD6 | TRISD5 | TRISD4 | TRISD3 | TRISD2 | TRISD1 | TRISD0 | 137 |

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

**Note 1:** These registers are not implemented on the PIC16(L)F1938 devices, read as '0'.

**REGISTER 12-20: LATE: PORTE DATA LATCH REGISTER**

| U-0 | U-0 | U-0 | U-0 | U-0 | R/W-x/u | R/W-x/u | R/W-x/u |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | LATE2(2) | LATE1(2) | LATE0(2) |
| 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 **Unimplemented**: Read as '0'

bit 3-0 **LATE<3:0>**: PORTE Output Latch Value bits(1)

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

**2:** LATE register is not implemented on the PIC16(L)F1938. Read as '0'

**REGISTER 12-21: ANSELE: PORTE ANALOG SELECT REGISTER**

| U-0 | U-0 | U-0 | U-0 | U-0 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | ANSE2(2) | ANSE1(2) | ANSE0(2) |
| 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:** ANSELE register is not implemented on the PIC16(L)F1938. Read as '0'

### 15.2.6 A/D 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)
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 15.4 "A/D Acquisition Requirements"**.

**EXAMPLE 15-1: A/D 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   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
```

**TABLE 19-1: SRCLK FREQUENCY TABLE**

| SRCLK | Divider | Fosc = 32 MHz | Fosc = 20 MHz | Fosc = 16 MHz | Fosc = 4 MHz | Fosc = 1 MHz |
|---|---|---|---|---|---|---|
| 111 | 512 | 62.5 kHz | 39.0 kHz | 31.3 kHz | 7.81 kHz | 1.95 kHz |
| 110 | 256 | 125 kHz | 78.1 kHz | 62.5 kHz | 15.6 kHz | 3.90 kHz |
| 101 | 128 | 250 kHz | 156 kHz | 125 kHz | 31.25 kHz | 7.81 kHz |
| 100 | 64 | 500 kHz | 313 kHz | 250 kHz | 62.5 kHz | 15.6 kHz |
| 011 | 32 | 1 MHz | 625 kHz | 500 kHz | 125 kHz | 31.3 kHz |
| 010 | 16 | 2 MHz | 1.25 MHz | 1 MHz | 250 kHz | 62.5 kHz |
| 001 | 8 | 4 MHz | 2.5 MHz | 2 MHz | 500 kHz | 125 kHz |
| 000 | 4 | 8 MHz | 5 MHz | 4 MHz | 1 MHz | 250 kHz |

The I²C interface supports the following modes and features:

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

Figure 24-2 is a block diagram of the I²C Interface module in Master mode. Figure 24-3 is a diagram of the I²C Interface module in Slave mode.

**FIGURE 24-2: MSSP BLOCK DIAGRAM (I²C™ MASTER MODE)**

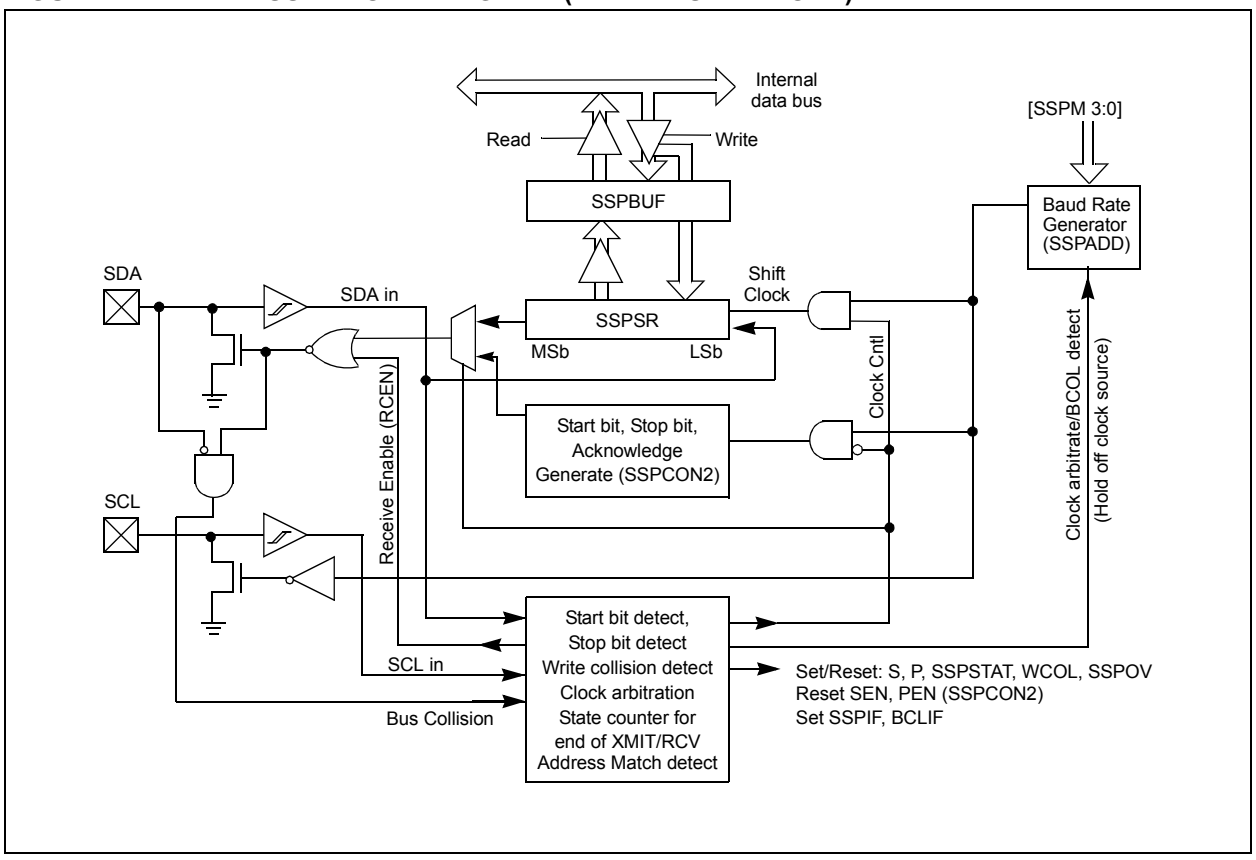
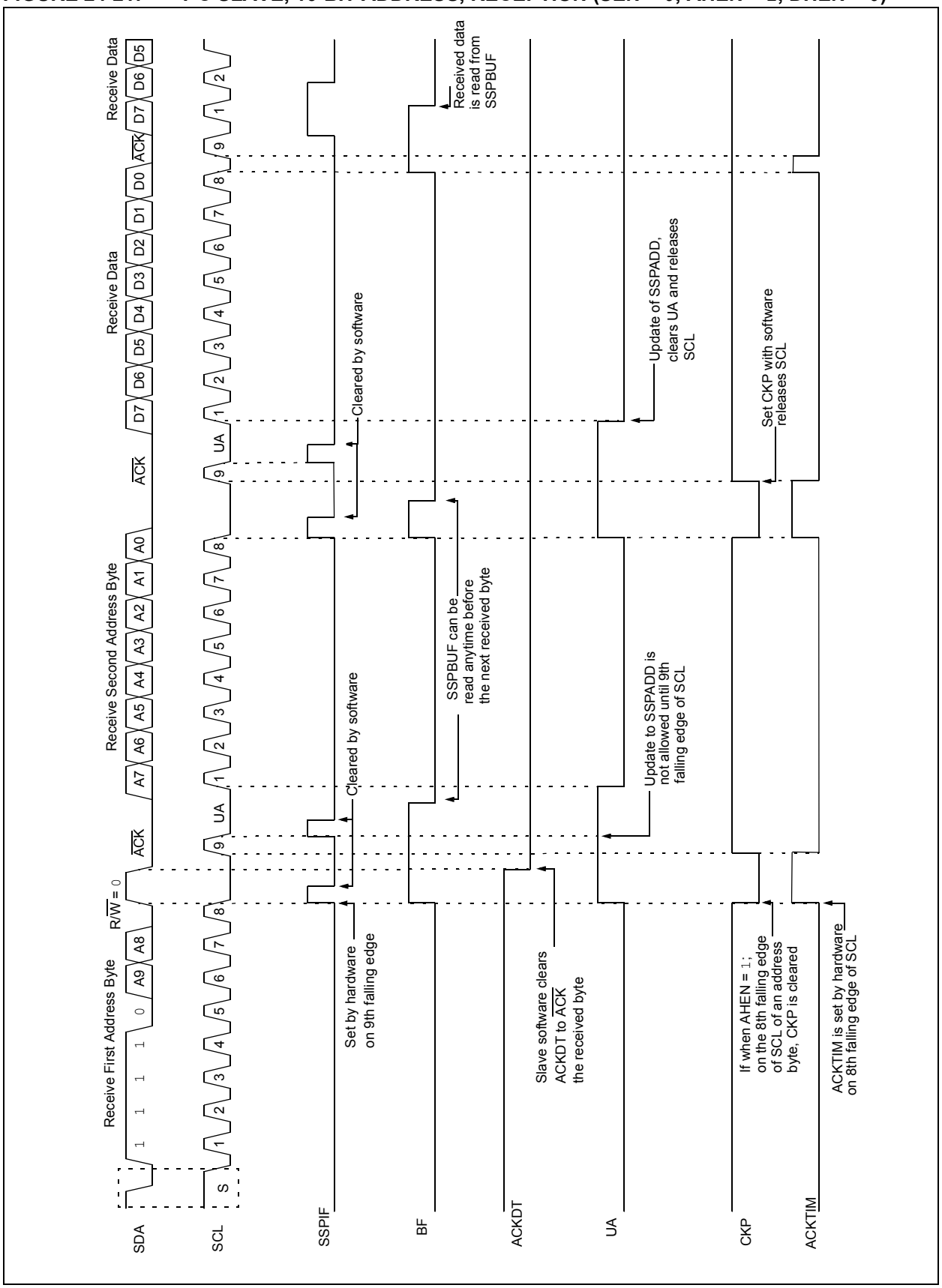


**FIGURE 24-21: I²C SLAVE, 10-BIT ADDRESS, RECEPTION (SEN = 0, AHEN = 1, DHEN = 0)**



#### 24.6.13.1 Bus Collision During a Start Condition

During a Start condition, a bus collision occurs if:

a) SDA or SCL are sampled low at the beginning of the Start condition (Figure 24-33).

b) SCL is sampled low before SDA is asserted low (Figure 24-34).

During a Start condition, both the SDA and the SCL pins are monitored.

If the SDA pin is already low, or the SCL pin is already low, then all of the following occur:

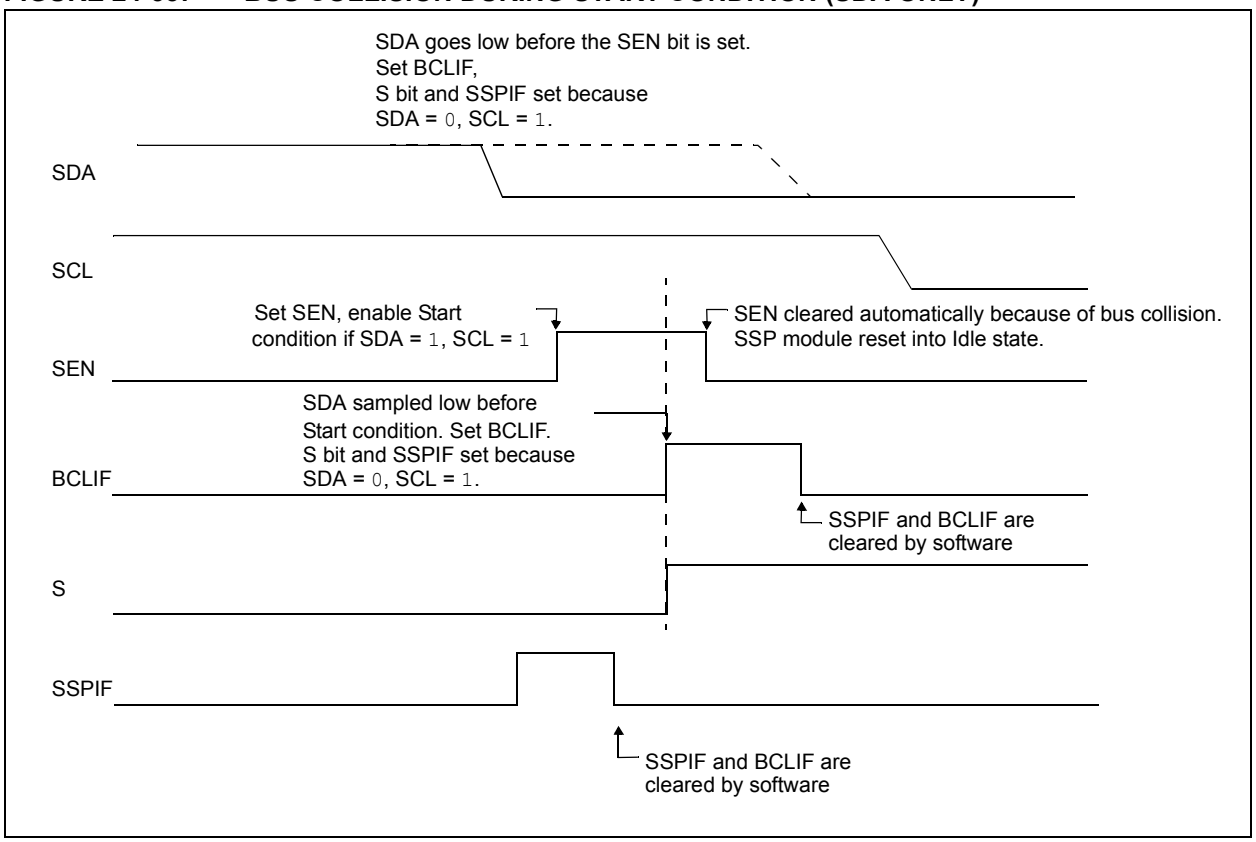
- the Start condition is aborted,
- the BCLIF flag is set and
- the MSSP module is reset to its Idle state (Figure 24-33).

The Start condition begins with the SDA and SCL pins deasserted. When the SDA pin is sampled high, the Baud Rate Generator is loaded and counts down. If the SCL pin is sampled low while SDA is high, a bus collision occurs because it is assumed that another master is attempting to drive a data '1' during the Start condition.

If the SDA pin is sampled low during this count, the BRG is reset and the SDA line is asserted early (Figure 24-35). If, however, a '1' is sampled on the SDA pin, the SDA pin is asserted low at the end of the BRG count. The Baud Rate Generator is then reloaded and counts down to zero; if the SCL pin is sampled as '0' during this time, a bus collision does not occur. At the end of the BRG count, the SCL pin is asserted low.

**Note:** The reason that bus collision is not a factor during a Start condition is that no two bus masters can assert a Start condition at the exact same time. Therefore, one master will always assert SDA before the other. This condition does not cause a bus collision because the two masters must be allowed to arbitrate the first address following the Start condition. If the address is the same, arbitration must be allowed to continue into the data portion, Repeated Start or Stop conditions.

**FIGURE 24-33: BUS COLLISION DURING START CONDITION (SDA ONLY)**



## 27.5 LCD Bias Internal Reference Ladder

The internal reference ladder can be used to divide the LCD bias voltage two or three equally spaced voltages that will be supplied to the LCD segment pins. To create this, the reference ladder consists of three matched resistors. Refer to Figure 27-3.

### 27.5.1 BIAS MODE INTERACTION

When in 1/2 Bias mode (BIASMD = 1), then the middle resistor of the ladder is shorted out so that only two voltages are generated. The current consumption of the ladder is higher in this mode, with the one resistor removed.

**TABLE 27-3: LCD INTERNAL LADDER POWER MODES (1/3 BIAS)**

| Power Mode | Nominal Resistance of Entire Ladder | Nominal IDD |
|---|---|---|
| Low | 3 Mohm | 1 µA |
| Medium | 300 kohm | 10 µA |
| High | 30 kohm | 100 µA |

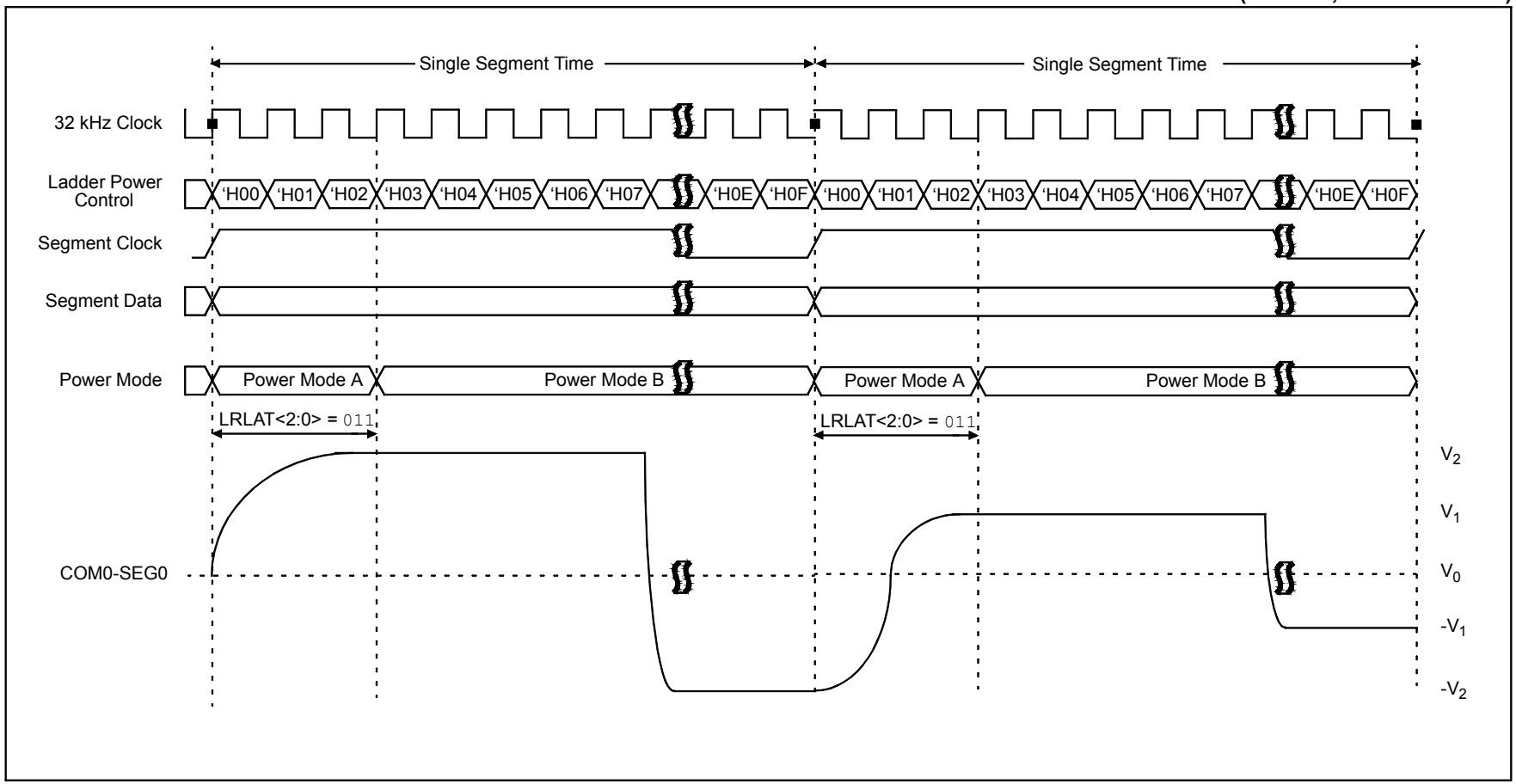
### 27.5.2 POWER MODES

The internal reference ladder may be operated in one of three power modes. This allows the user to trade off LCD contrast for power in the specific application. The larger the LCD glass, the more capacitance is present on a physical LCD segment, requiring more current to maintain the same contrast level.

Three different power modes are available, LP, MP and HP. The internal reference ladder can also be turned off for applications that wish to provide an external ladder or to minimize power consumption. Disabling the internal reference ladder results in all of the ladders being disconnected, allowing external voltages to be supplied.

Whenever the LCD module is inactive (LCDA = 0), the internal reference ladder will be turned off.

**FIGURE 27-5: LCD INTERNAL REFERENCE LADDER POWER MODE SWITCHING DIAGRAM – TYPE A WAVEFORM (1/2 MUX, 1/2 BIAS DRIVE)**



## 29.2 Instruction Descriptions

**ADDFSR — Add Literal to FSRn**

| | |
|---|---|
| Syntax: | [ *label* ] ADDFSR FSRn, k |
| Operands: | -32 ≤ k ≤ 31<br>n ∈ [ 0, 1] |
| Operation: | FSR(n) + k → FSR(n) |
| Status Affected: | None |
| Description: | The signed 6-bit literal 'k' is added to the contents of the FSRnH:FSRnL register pair.<br><br>FSRn is limited to the range 0000h - FFFFh. Moving beyond these bounds will cause the FSR to wrap-around. |

**ADDLW — Add literal and W**

| | |
|---|---|
| Syntax: | [ *label* ] ADDLW k |
| Operands: | 0 ≤ k ≤ 255 |
| Operation: | (W) + k → (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 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | (W) + (f) → (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'. |

**ADDWFC — ADD W and CARRY bit to f**

| | |
|---|---|
| Syntax: | [ *label* ] ADDWFC f {,d} |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | (W) + (f) + (C) → dest |
| Status Affected: | C, DC, Z |
| Description: | Add W, the Carry flag and data memory location 'f'. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed in data memory location 'f'. |

**ANDLW — AND literal with W**

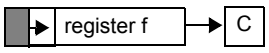
| | |
|---|---|
| Syntax: | [ *label* ] ANDLW k |
| Operands: | 0 ≤ k ≤ 255 |
| Operation: | (W) .AND. (k) → (W) |
| Status Affected: | Z |
| Description: | The contents of W register are AND'ed with the 8-bit literal 'k'. The result is placed in the W register. |

**ANDWF — AND W with f**

| | |
|---|---|
| Syntax: | [ *label* ] ANDWF f,d |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | (W) .AND. (f) → (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'. |

**ASRF — Arithmetic Right Shift**

| | |
|---|---|
| Syntax: | [ *label* ] ASRF f {,d} |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | (f<7>)→ dest<7><br>(f<7:1>) → dest<6:0>,<br>(f<0>) → C, |
| Status Affected: | C, Z |
| Description: | The contents of register 'f' are shifted one bit to the right through the Carry flag. The MSb remains unchanged. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f'. |



**DECFSZ** **Decrement f, Skip if 0**

| | |
|---|---|
| Syntax: | [ *label* ] DECFSZ f,d |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | (f) - 1 → (destination);<br>skip if result = 0 |
| Status Affected: | None |
| Description: | The contents of register 'f' are decremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.<br>If the result is '1', the next instruction is executed. If the result is '0', then a NOP is executed instead, making it a 2-cycle instruction. |

**GOTO** **Unconditional Branch**

| | |
|---|---|
| Syntax: | [ *label* ] GOTO k |
| Operands: | 0 ≤ k ≤ 2047 |
| Operation: | k → PC<10:0><br>PCLATH<6:3> → PC<14:11> |
| Status Affected: | None |
| Description: | GOTO is an unconditional branch. The eleven-bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. GOTO is a 2-cycle instruction. |

**INCF** **Increment f**

| | |
|---|---|
| Syntax: | [ *label* ] INCF f,d |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | (f) + 1 → (destination) |
| Status Affected: | Z |
| Description: | The contents of register 'f' are incremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'. |

**INCFSZ** **Increment f, Skip if 0**

| | |
|---|---|
| Syntax: | [ *label* ] INCFSZ f,d |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | (f) + 1 → (destination),<br>skip if result = 0 |
| Status Affected: | None |
| Description: | The contents of register 'f' are incremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.<br>If the result is '1', the next instruction is executed. If the result is '0', a NOP is executed instead, making it a 2-cycle instruction. |

**IORLW** **Inclusive OR literal with W**

| | |
|---|---|
| Syntax: | [ *label* ] IORLW k |
| Operands: | 0 ≤ k ≤ 255 |
| Operation: | (W) .OR. k → (W) |
| Status Affected: | Z |
| Description: | The contents of the W register are OR'ed with the 8-bit literal 'k'. The result is placed in the W register. |

**IORWF** **Inclusive OR W with f**

| | |
|---|---|
| Syntax: | [ *label* ] IORWF f,d |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | (W) .OR. (f) → (destination) |
| Status Affected: | Z |
| Description: | Inclusive OR the W register with register 'f'. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'. |

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

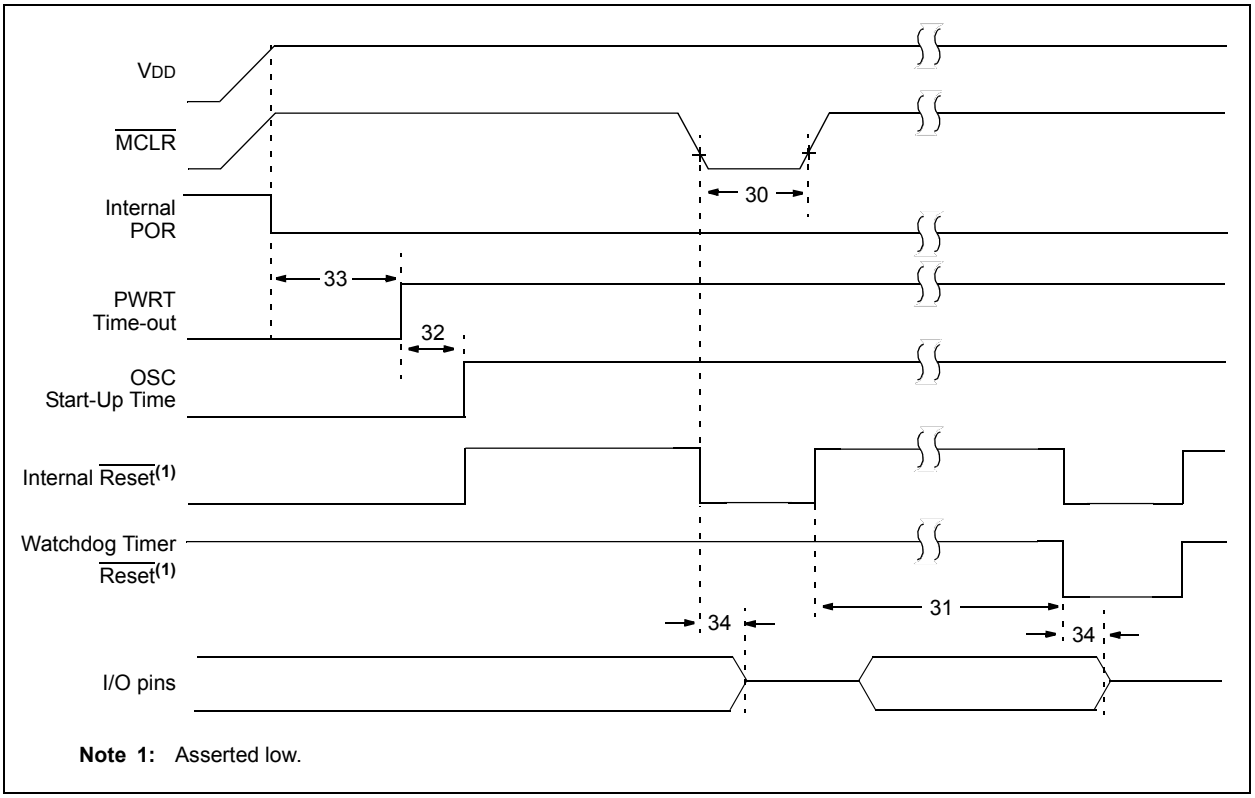
| **Standard Operating Conditions (unless otherwise stated)**<br>Operating Temperature -40°C ≤ TA ≤ +125°C | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| OS11 | TosH2ckL | FOSC↑ to CLKOUT↓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

* These parameters are characterized but not tested.

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

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

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




**Note 1:** Asserted low.

**TABLE 30-16: I²C™ BUS DATA REQUIREMENTS**

| Param. No. | Symbol | Characteristic | | Min. | Max. | Units | Conditions |
|---|---|---|---|---|---|---|---|
| SP100* | THIGH | Clock high time | 100 kHz mode | 4.0 | — | μs | Device must operate at a minimum of 1.5 MHz |
| | | | 400 kHz mode | 0.6 | — | μs | Device must operate at a minimum of 10 MHz |
| | | | SSP module | 1.5TCY | — | | |
| SP101* | TLOW | Clock low time | 100 kHz mode | 4.7 | — | μs | Device must operate at a minimum of 1.5 MHz |
| | | | 400 kHz mode | 1.3 | — | μs | Device must operate at a minimum of 10 MHz |
| | | | SSP module | 1.5TCY | — | | |
| SP102* | TR | SDA and SCL rise time | 100 kHz mode | — | 1000 | ns | |
| | | | 400 kHz mode | 20 + 0.1CB | 300 | ns | CB is specified to be from 10-400 pF |
| SP103* | TF | SDA and SCL fall time | 100 kHz mode | — | 250 | ns | |
| | | | 400 kHz mode | 20 + 0.1CB | 250 | ns | CB is specified to be from 10-400 pF |
| SP106* | THD:DAT | Data input hold time | 100 kHz mode | 0 | — | ns | |
| | | | 400 kHz mode | 0 | 0.9 | μs | |
| SP107* | TSU:DAT | Data input setup time | 100 kHz mode | 250 | — | ns | **(Note 2)** |
| | | | 400 kHz mode | 100 | — | ns | |
| SP109* | TAA | Output valid from clock | 100 kHz mode | — | 3500 | ns | **(Note 1)** |
| | | | 400 kHz mode | — | — | ns | |
| SP110* | TBUF | Bus free time | 100 kHz mode | 4.7 | — | μs | Time the bus must be free before a new transmission can start |
| | | | 400 kHz mode | 1.3 | — | μs | |
| SP111 | CB | Bus capacitive loading | | — | 400 | pF | |

* These parameters are characterized but not tested.

**Note 1:** As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of Start or Stop conditions.

**2:** A Fast mode (400 kHz) I²C™ bus device can be used in a Standard mode (100 kHz) I²C bus system, but the requirement TSU:DAT ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the low period of the SCL signal. If such a device does stretch the low period of the SCL signal, it must output the next data bit to the SDA line TR max. + TSU:DAT = 1000 + 250 = 1250 ns (according to the Standard mode I²C bus specification), before the SCL line is released.

**FIGURE 31-17: IDD TYPICAL, EXTERNAL CLOCK (ECH), HIGH-POWER MODE, PIC16F1938/9 ONLY**

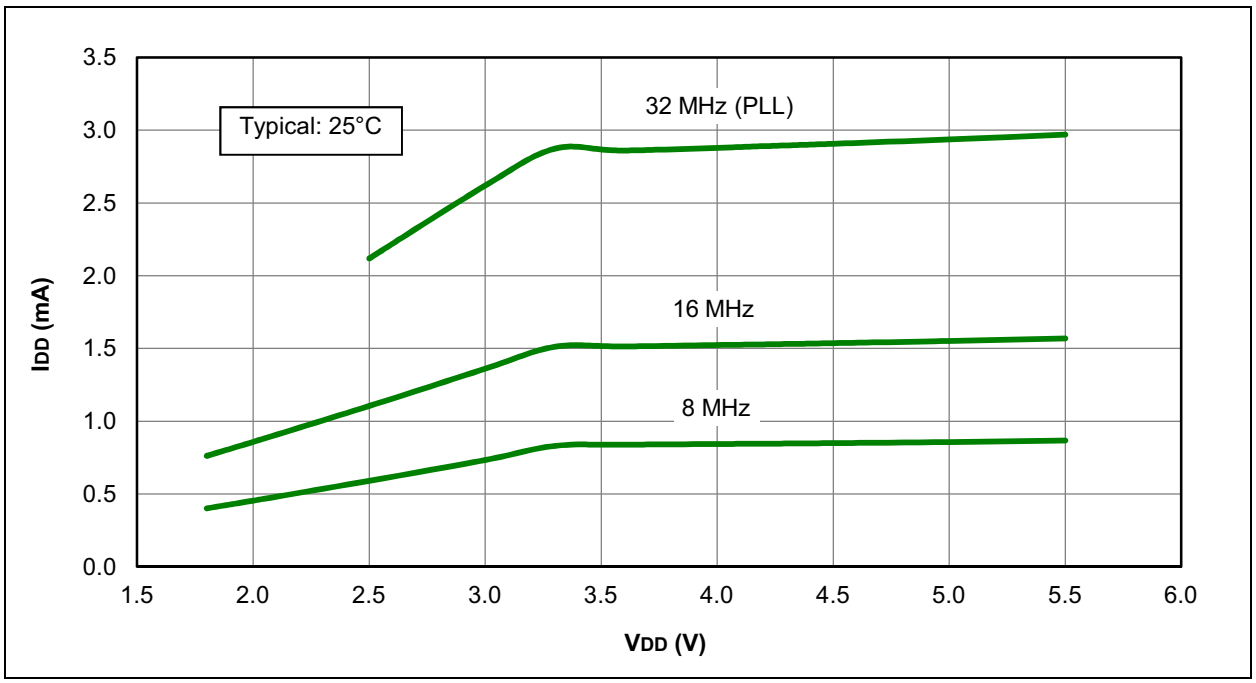



**FIGURE 31-18: IDD MAXIMUM, EXTERNAL CLOCK (ECH), HIGH-POWER MODE, PIC16F1938/9 ONLY**

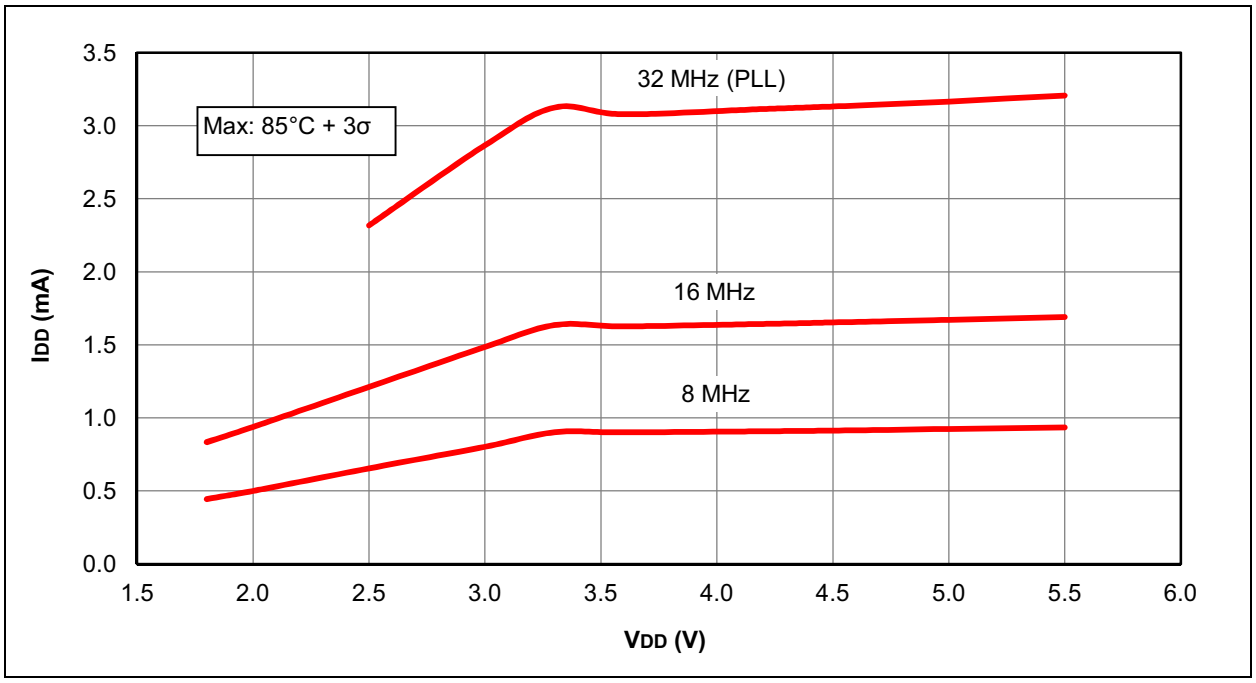


**FIGURE 31-49: IPD, COMPARATOR, NORMAL-POWER MODE, (CxSP = 1), PIC16LF1938/9 ONLY**

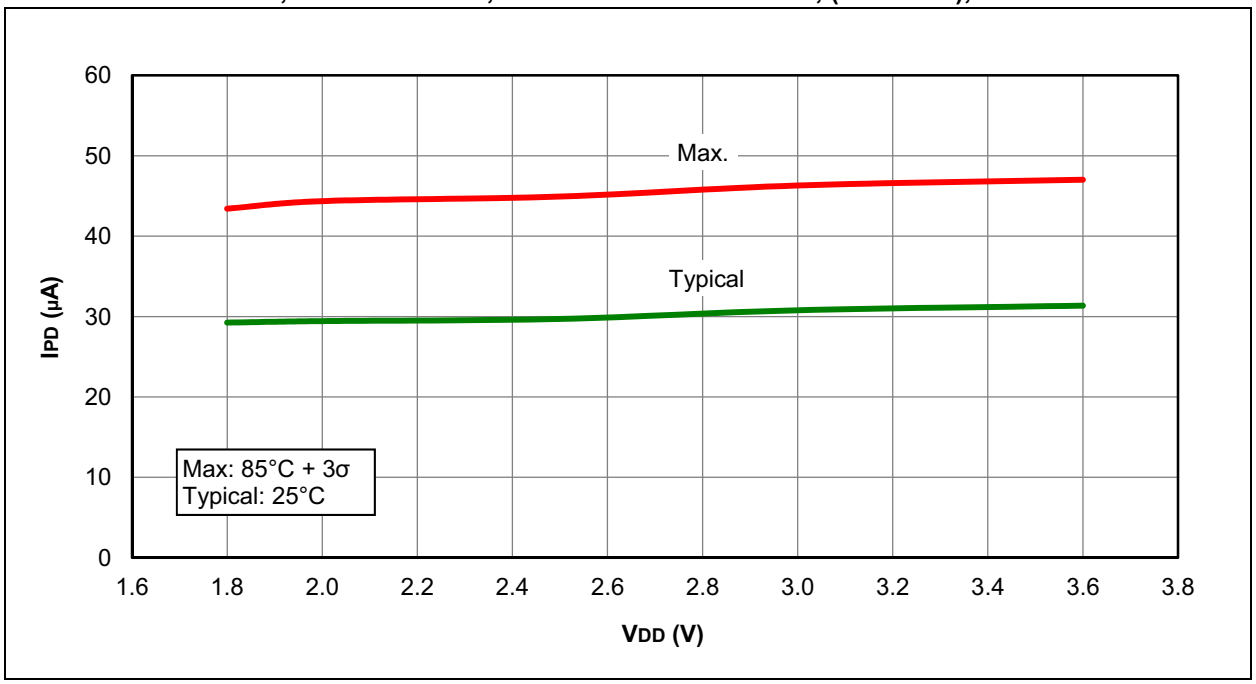



**FIGURE 31-50: IPD, COMPARATOR, NORMAL-POWER MODE, (CxSP = 1), PIC16F1938/9 ONLY**