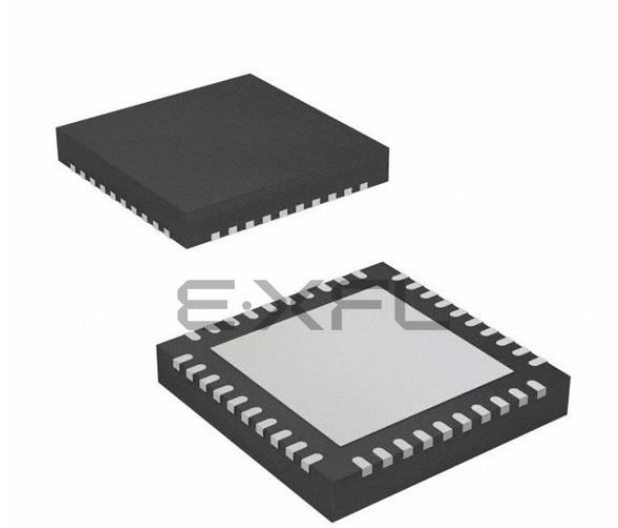


**Welcome to E-XFL.COM**

## 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 | 48MHz |
| Connectivity | I²C, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, HLVD, POR, PWM, WDT |
| Number of I/O | 35 |
| Program Memory Size | 16KB (8K x 16) |
| Program Memory Type | FLASH |
| EEPROM Size | 256 x 8 |
| RAM Size | 768 x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.3V ~ 5.5V |
| Data Converters | A/D 30x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 125°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 40-UFQFN Exposed Pad |
| Supplier Device Package | 40-UQFN (5x5) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic18f44k22-e-mv |

**FIGURE 5: 44-PIN TQFP DIAGRAM**

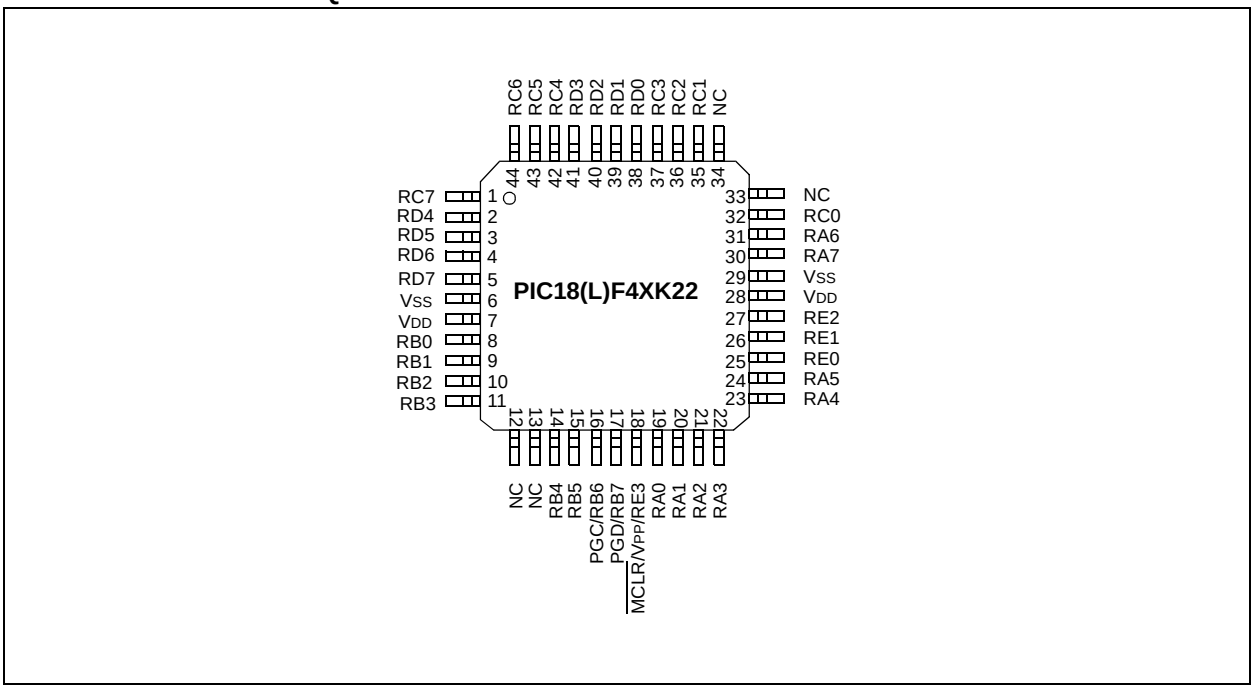
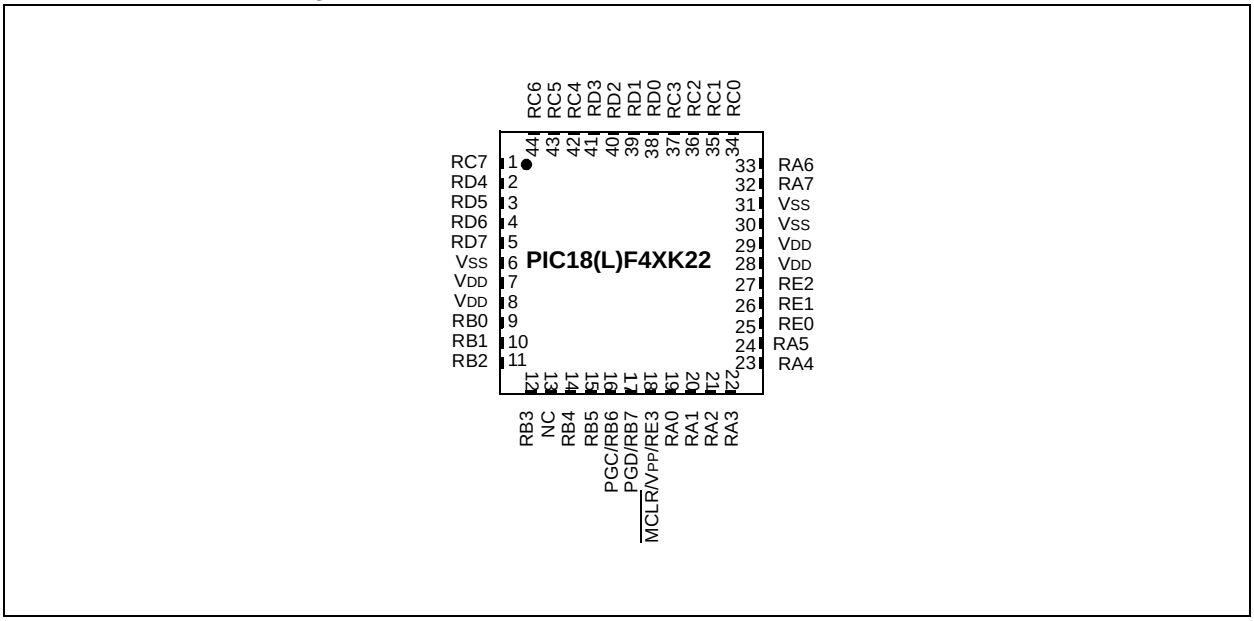



**FIGURE 6: 44-PIN QFN DIAGRAM**



**REGISTER 2-2: OSCCON2: OSCILLATOR CONTROL REGISTER 2**

| R-0/0 | R-0/q | U-0 | R/W-0/0 | R/W-0/u | R/W-1/1 | R-x/u | R-0/0 |
|---|---|---|---|---|---|---|---|
| PLLRDY | SOSCRUN | — | MFIOSEL | SOSCGO(1) | PRISD | MFIOFS | LFIOFS |
| bit 7 | | | | | | | bit 0 |

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

bit 7 **PLLRDY:** PLL Run Status bit
1 = System clock comes from 4xPLL
0 = System clock comes from an oscillator, other than 4xPLL

bit 6 **SOSCRUN:** SOSC Run Status bit
1 = System clock comes from secondary SOSC
0 = System clock comes from an oscillator, other than SOSC

bit 5 **Unimplemented:** Read as '0'.

bit 4 **MFIOSEL:** MFINTOSC Select bit
1 = MFINTOSC is used in place of HFINTOSC frequencies of 500 kHz, 250 kHz and 31.25 kHz
0 = MFINTOSC is not used

bit 3 **SOSCGO(1):** Secondary Oscillator Start Control bit
1 = Secondary oscillator is enabled.
0 = Secondary oscillator is shut off if no other sources are requesting it.

bit 2 **PRISD:** Primary Oscillator Drive Circuit Shutdown bit
1 = Oscillator drive circuit on
0 = Oscillator drive circuit off (zero power)

bit 1 **MFIOFS:** MFINTOSC Frequency Stable bit
1 = MFINTOSC is stable
0 = MFINTOSC is not stable

bit 0 **LFIOFS:** LFINTOSC Frequency Stable bit
1 = LFINTOSC is stable
0 = LFINTOSC is not stable

**Note 1:** The SOSCGO bit is only reset on a POR Reset.

**TABLE 2-3: OSC1 AND OSC2 PIN STATES IN SLEEP MODE**

| OSC Mode | OSC1 Pin | OSC2 Pin |
|---|---|---|
| RC, INTOSC with CLKOUT | Floating, external resistor should pull high | At logic low (clock/4 output) |
| RC with IO | Floating, external resistor should pull high | Configured as PORTA, bit 6 |
| INTOSC with IO | Configured as PORTA, bit 7 | Configured as PORTA, bit 6 |
| EC with IO | Floating, pulled by external clock | Configured as PORTA, bit 6 |
| EC with CLKOUT | Floating, pulled by external clock | At logic low (clock/4 output) |
| LP, XT, HS | Feedback inverter disabled at quiescent voltage level | Feedback inverter disabled at quiescent voltage level |

**Note:** See Table 4-2 in **Section 4.0 "Reset"** for time-outs due to Sleep and $\overline{\text{MCLR}}$ Reset.

## 2.11 Clock Switching

The system clock source can be switched between external and internal clock sources via software using the System Clock Select (SCS<1:0>) bits of the OSCCON register.

PIC18(L)F2X/4XK22 devices contain circuitry to prevent clock "glitches" when switching between clock sources. A short pause in the device clock occurs during the clock switch. The length of this pause is the sum of two cycles of the old clock source and three to four cycles of the new clock source. This formula assumes that the new clock source is stable.

Clock transitions are discussed in greater detail in **Section 3.1.2 "Entering Power-Managed Modes"**.

### 2.11.1 SYSTEM CLOCK SELECT (SCS<1:0>) BITS

The System Clock Select (SCS<1:0>) bits of the OSCCON register select the system clock source that is used for the CPU and peripherals.

- When SCS<1:0> = 00, the system clock source is determined by configuration of the FOSC<3:0> bits in the CONFIG1H Configuration register.
- When SCS<1:0> = 10, the system clock source is chosen by the internal oscillator frequency selected by the INTSRC bit of the OSCTUNE register, the MFIOSEL bit of the OSCCON2 register and the IRCF<2:0> bits of the OSCCON register.
- When SCS<1:0> = 01, the system clock source is the 32.768 kHz secondary oscillator shared with Timer1, Timer3 and Timer5.

After a Reset, the SCS<1:0> bits of the OSCCON register are always cleared.

**Note:** Any automatic clock switch, which may occur from Two-Speed Start-up or Fail-Safe Clock Monitor, does not update the SCS<1:0> bits of the OSCCON register. The user can monitor the SOSCRUN, MFIOFS and LFIOFS bits of the OSCCON2 register, and the HFIOFS and OSTS bits of the OSCCON register to determine the current system clock source.

### 2.11.2 OSCILLATOR START-UP TIME-OUT STATUS (OSTS) BIT

The Oscillator Start-up Time-out Status (OSTS) bit of the OSCCON register indicates whether the system clock is running from the external clock source, as defined by the FOSC<3:0> bits in the CONFIG1H Configuration register, or from the internal clock source. In particular, when the primary oscillator is the source of the primary clock, OSTS indicates that the Oscillator Start-up Timer (OST) has timed out for LP, XT or HS modes.

## 3.3 Sleep Mode

The Power-Managed Sleep mode in the PIC18(L)F2X/4XK22 devices is identical to the legacy Sleep mode offered in all other PIC microcontroller devices. It is entered by clearing the IDLEN bit of the OSCCON register and executing the SLEEP instruction. This shuts down the selected oscillator (Figure 3-4) and all clock source Status bits are cleared.

Entering the Sleep mode from either Run or Idle mode does not require a clock switch. This is because no clocks are needed once the controller has entered Sleep. If the WDT is selected, the LFINTOSC source will continue to operate. If the SOSC oscillator is enabled, it will also continue to run.

When a wake event occurs in Sleep mode (by interrupt, Reset or WDT time-out), the device will not be clocked until the clock source selected by the SCS<1:0> bits becomes ready (see Figure 3-5), or it will be clocked from the internal oscillator block if either the Two-Speed Start-up or the Fail-Safe Clock Monitor are enabled (see **Section 24.0 "Special Features of the CPU"**). In either case, the OSTS bit is set when the primary clock is providing the device clocks. The IDLEN and SCS bits are not affected by the wake-up.

## 3.4 Idle Modes

The Idle modes allow the controller's CPU to be selectively shut down while the peripherals continue to operate. Selecting a particular Idle mode allows users to further manage power consumption.

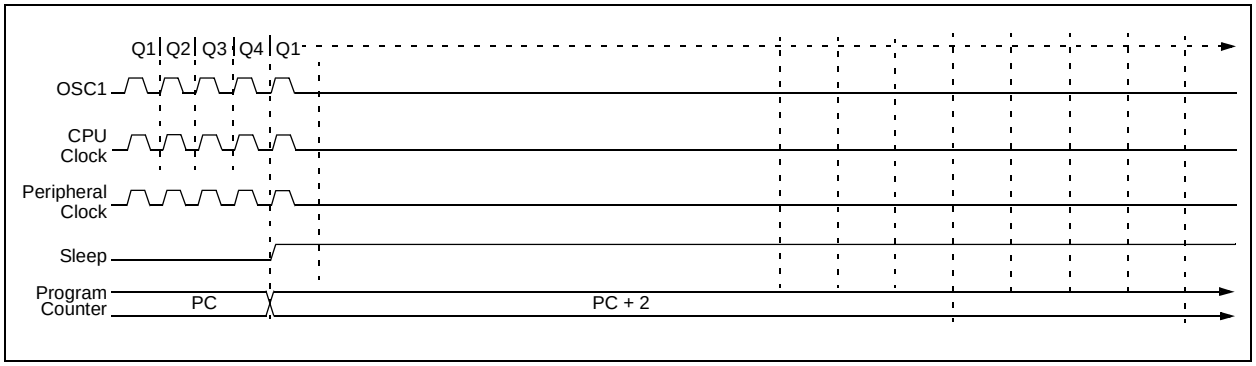
If the IDLEN bit is set to a '1' when a SLEEP instruction is executed, the peripherals will be clocked from the clock source selected by the SCS<1:0> bits; however, the CPU will not be clocked. The clock source status bits are not affected. Setting IDLEN and executing a SLEEP instruction provides a quick method of switching from a given Run mode to its corresponding Idle mode.

If the WDT is selected, the LFINTOSC source will continue to operate. If the SOSC oscillator is enabled, it will also continue to run.

Since the CPU is not executing instructions, the only exits from any of the Idle modes are by interrupt, WDT time-out, or a Reset. When a wake event occurs, CPU execution is delayed by an interval of $T_{CSD}$ while it becomes ready to execute code. When the CPU begins executing code, it resumes with the same clock source for the current Idle mode. For example, when waking from RC_IDLE mode, the internal oscillator block will clock the CPU and peripherals (in other words, RC_RUN mode). The IDLEN and SCS bits are not affected by the wake-up.

While in any Idle mode or the Sleep mode, a WDT time-out will result in a WDT wake-up to the Run mode currently specified by the SCS<1:0> bits.

**FIGURE 3-4: TRANSITION TIMING FOR ENTRY TO SLEEP MODE**



**REGISTER 10-10: LATx: PORTx OUTPUT LATCH REGISTER(1)**

| R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u | R/W-x/u |
|---|---|---|---|---|---|---|---|
| LATx7 | LATx6 | LATx5 | LATx4 | LATx3 | LATx2 | LATx1 | LATx0 |
| 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 7-0 **LATx<7:0>:** PORTx Output Latch bit value(2)

**Note 1:** Register Description for LATA, LATB, LATC and LATD.

**2:** Writes to PORTx are written to corresponding LATx register. Reads from PORTx register is return of I/O pin values.

**REGISTER 10-11: LATE: PORTE OUTPUT LATCH REGISTER(1)**

| 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' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

bit 7-3 **Unimplemented:** Read as '0'

bit 2-0 **LATE<2:0>:** PORTE Output Latch bit value(2)

**Note 1:** Available on PIC18(L)F4XK22 devices only.

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

**REGISTER 10-12: WPUB: WEAK PULL-UP PORTB REGISTER**

| R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| WPUB7 | WPUB6 | WPUB5 | WPUB4 | WPUB3 | WPUB2 | WPUB1 | WPUB0 |
| bit 7 | | | | | | | bit 0 |

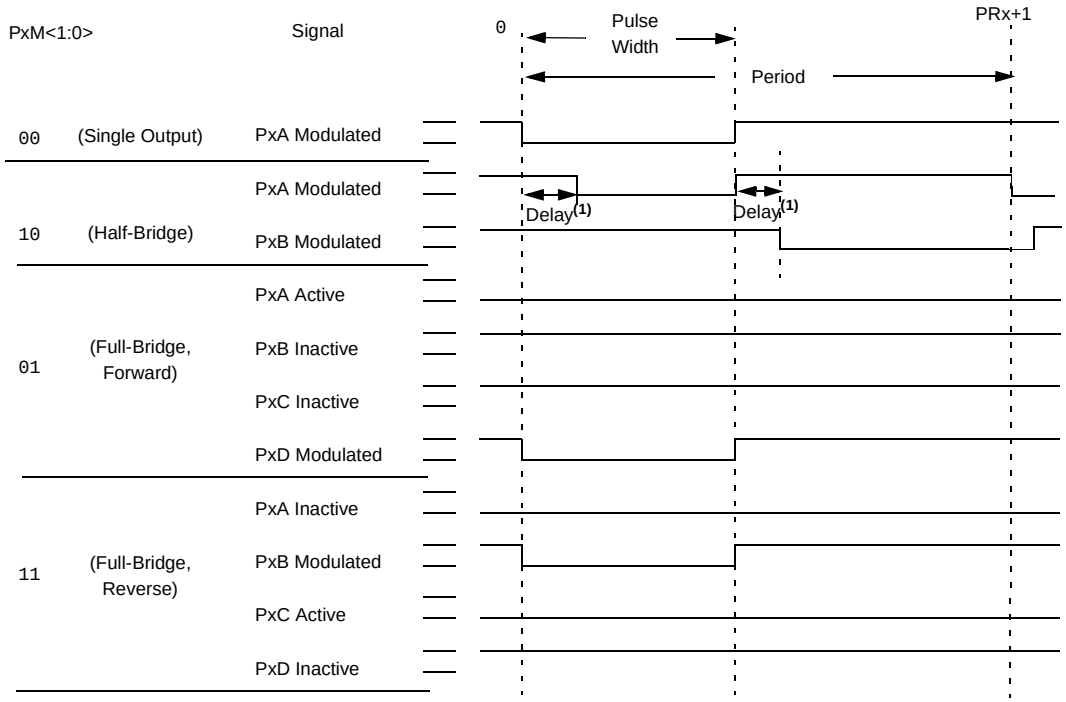
| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

bit 7-0 **WPUB<7:0>:** Weak Pull-up Register bits

1 = Pull-up enabled on PORT pin

0 = Pull-up disabled on PORT pin

**FIGURE 14-7: EXAMPLE ENHANCED PWM OUTPUT RELATIONSHIPS (ACTIVE-LOW STATE)**




Relationships:

- Period = 4 * TOSC * (PRx + 1) * (TMRx Prescale Value)
- Pulse Width = TOSC * (CCPRxL<7:0>:CCPxCON<5:4>) * (TMRx Prescale Value)
- Delay = 4 * TOSC * (PWMxCON<6:0>)

**Note 1:** Dead-band delay is programmed using the PWMxCON register (**Section 14.4.5 "Programmable Dead-Band Delay Mode"**).

## 15.4 I²C Mode Operation

All MSSPx I²C communication is byte oriented and shifted out MSb first. Six SFR registers and 2 interrupt flags interface the module with the PIC microcontroller and user software. Two pins, SDAx and SCLx, are exercised by the module to communicate with other external I²C devices.

### 15.4.1 BYTE FORMAT

All communication in I²C is done in 9-bit segments. A byte is sent from a master to a slave or vice-versa, followed by an Acknowledge bit sent back. After the 8th falling edge of the SCLx line, the device outputting data on the SDAx changes that pin to an input and reads in an acknowledge value on the next clock pulse.

The clock signal, SCLx, is provided by the master. Data is valid to change while the SCLx signal is low, and sampled on the rising edge of the clock. Changes on the SDAx line while the SCLx line is high define special conditions on the bus, explained below.

### 15.4.2 DEFINITION OF I²C TERMINOLOGY

There is language and terminology in the description of I²C communication that have definitions specific to I²C. That word usage is defined below and may be used in the rest of this document without explanation. This table was adapted from the Phillips I²C specification.

### 15.4.3 SDAx AND SCLx PINS

Selection of any I²C mode with the SSPxEN bit set, forces the SCLx and SDAx pins to be open-drain. These pins should be set by the user to inputs by setting the appropriate TRIS bits.

| | |
|---|---|
| **Note:** | Data is tied to output zero when an I²C mode is enabled. |

### 15.4.4 SDAx HOLD TIME

The hold time of the SDAx pin is selected by the SDAHT bit of the SSPxCON3 register. Hold time is the time SDAx is held valid after the falling edge of SCLx. Setting the SDAHT bit selects a longer 300 ns minimum hold time and may help on buses with large capacitance.

**TABLE 15-1: I²C BUS TERMS**

| TERM | Description |
|---|---|
| Transmitter | The device which shifts data out onto the bus. |
| Receiver | The device which shifts data in from the bus. |
| Master | The device that initiates a transfer, generates clock signals and terminates a transfer. |
| Slave | The device addressed by the master. |
| Multi-master | A bus with more than one device that can initiate data transfers. |
| Arbitration | Procedure to ensure that only one master at a time controls the bus. Winning arbitration ensures that the message is not corrupted. |
| Synchronization | Procedure to synchronize the clocks of two or more devices on the bus. |
| Idle | No master is controlling the bus, and both SDAx and SCLx lines are high. |
| Active | Any time one or more master devices are controlling the bus. |
| Addressed Slave | Slave device that has received a matching address and is actively being clocked by a master. |
| Matching Address | Address byte that is clocked into a slave that matches the value stored in SSPxADD. |
| Write Request | Slave receives a matching address with R/$\overline{W}$ bit clear, and is ready to clock in data. |
| Read Request | Master sends an address byte with the R/$\overline{W}$ bit set, indicating that it wishes to clock data out of the Slave. This data is the next and all following bytes until a Restart or Stop. |
| Clock Stretching | When a device on the bus holds SCLx low to stall communication. |
| Bus Collision | Any time the SDAx line is sampled low by the module while it is outputting and expected high state. |

**FIGURE 15-15: $I^2C$ SLAVE, 7-BIT ADDRESS, RECEPTION (SEN = 1, AHEN = 0, DHEN = 0)**

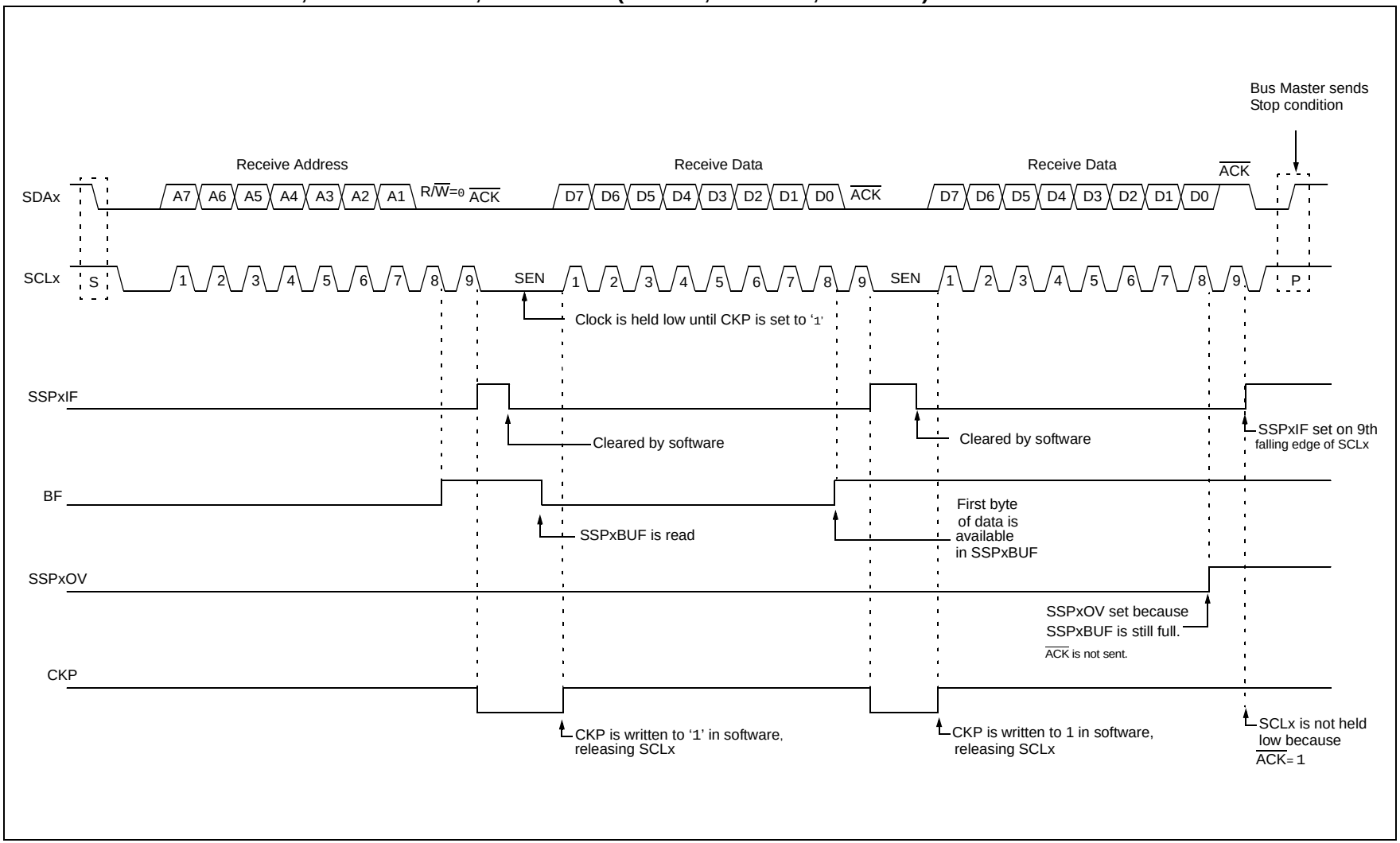


**TABLE 16-5: BAUD RATES FOR ASYNCHRONOUS MODES (CONTINUED)**

| BAUD RATE | SYNC = 0, BRGH = 0, BRG16 = 1 | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | FOSC = 8.000 MHz | | | FOSC = 4.000 MHz | | | FOSC = 3.6864 MHz | | | FOSC = 1.000 MHz | | |
| | Actual Rate | % Error | SPBRGHx: SPBRGx (decimal) | Actual Rate | % Error | SPBRGHx: SPBRGx (decimal) | Actual Rate | % Error | SPBRGHx :SPBRGx (decimal) | Actual Rate | % Error | SPBRGHx: SPBRGx (decimal) |
| 300 | 299.9 | -0.02 | 1666 | 300.1 | 0.04 | 832 | 300.0 | 0.00 | 767 | 300.5 | 0.16 | 207 |
| 1200 | 1199 | -0.08 | 416 | 1202 | 0.16 | 207 | 1200 | 0.00 | 191 | 1202 | 0.16 | 51 |
| 2400 | 2404 | 0.16 | 207 | 2404 | 0.16 | 103 | 2400 | 0.00 | 95 | 2404 | 0.16 | 25 |
| 9600 | 9615 | 0.16 | 51 | 9615 | 0.16 | 25 | 9600 | 0.00 | 23 | — | — | — |
| 10417 | 10417 | 0.00 | 47 | 10417 | 0.00 | 23 | 10473 | 0.53 | 21 | 10417 | 0.00 | 5 |
| 19.2k | 19.23k | 0.16 | 25 | 19.23k | 0.16 | 12 | 19.20k | 0.00 | 11 | — | — | — |
| 57.6k | 55556 | -3.55 | 8 | — | — | — | 57.60k | 0.00 | 3 | — | — | — |
| 115.2k | — | — | — | — | — | — | 115.2k | 0.00 | 1 | — | — | — |

| BAUD RATE | SYNC = 0, BRGH = 1, BRG16 = 1 or SYNC = 1, BRG16 = 1 | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | FOSC = 64.000 MHz | | | FOSC = 18.432 MHz | | | FOSC = 16.000 MHz | | | FOSC = 11.0592 MHz | | |
| | Actual Rate | % Error | SPBRGHx: SPBRGx (decimal) | Actual Rate | % Error | SPBRGHx: SPBRGx (decimal) | Actual Rate | % Error | SPBRGHx :SPBRGx (decimal) | Actual Rate | % Error | SPBRGHx: SPBRGx (decimal) |
| 300 | 300 | 0.00 | 53332 | 300.0 | 0.00 | 15359 | 300.0 | 0.00 | 13332 | 300.0 | 0.00 | 9215 |
| 1200 | 1200 | 0.00 | 13332 | 1200 | 0.00 | 3839 | 1200.1 | 0.01 | 3332 | 1200 | 0.00 | 2303 |
| 2400 | 2400 | 0.00 | 6666 | 2400 | 0.00 | 1919 | 2399.5 | -0.02 | 1666 | 2400 | 0.00 | 1151 |
| 9600 | 9598.1 | -0.02 | 1666 | 9600 | 0.00 | 479 | 9592 | -0.08 | 416 | 9600 | 0.00 | 287 |
| 10417 | 10417 | 0.00 | 1535 | 10425 | 0.08 | 441 | 10417 | 0.00 | 383 | 10433 | 0.16 | 264 |
| 19.2k | 19.21k | 0.04 | 832 | 19.20k | 0.00 | 239 | 19.23k | 0.16 | 207 | 19.20k | 0.00 | 143 |
| 57.6k | 57.55k | -0.08 | 277 | 57.60k | 0.00 | 79 | 57.97k | 0.64 | 68 | 57.60k | 0.00 | 47 |
| 115.2k | 115.11k | -0.08 | 138 | 115.2k | 0.00 | 39 | 114.29k | -0.79 | 34 | 115.2k | 0.00 | 23 |

| BAUD RATE | SYNC = 0, BRGH = 1, BRG16 = 1 or SYNC = 1, BRG16 = 1 | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | FOSC = 8.000 MHz | | | FOSC = 4.000 MHz | | | FOSC = 3.6864 MHz | | | FOSC = 1.000 MHz | | |
| | Actual Rate | % Error | SPBRGHx: SPBRGx (decimal) | Actual Rate | % Error | SPBRGHx: SPBRGx (decimal) | Actual Rate | % Error | SPBRGHx :SPBRGx (decimal) | Actual Rate | % Error | SPBRGHx: SPBRGx (decimal) |
| 300 | 300.0 | 0.00 | 6666 | 300.0 | 0.01 | 3332 | 300.0 | 0.00 | 3071 | 300.1 | 0.04 | 832 |
| 1200 | 1200 | -0.02 | 1666 | 1200 | 0.04 | 832 | 1200 | 0.00 | 767 | 1202 | 0.16 | 207 |
| 2400 | 2401 | 0.04 | 832 | 2398 | 0.08 | 416 | 2400 | 0.00 | 383 | 2404 | 0.16 | 103 |
| 9600 | 9615 | 0.16 | 207 | 9615 | 0.16 | 103 | 9600 | 0.00 | 95 | 9615 | 0.16 | 25 |
| 10417 | 10417 | 0.00 | 191 | 10417 | 0.00 | 95 | 10473 | 0.53 | 87 | 10417 | 0.00 | 23 |
| 19.2k | 19.23k | 0.16 | 103 | 19.23k | 0.16 | 51 | 19.20k | 0.00 | 47 | 19.23k | 0.16 | 12 |
| 57.6k | 57.14k | -0.79 | 34 | 58.82k | 2.12 | 16 | 57.60k | 0.00 | 15 | — | — | — |
| 115.2k | 117.6k | 2.12 | 16 | 111.1k | -3.55 | 8 | 115.2k | 0.00 | 7 | — | — | — |

## 21.0 FIXED VOLTAGE REFERENCE (FVR)

The Fixed Voltage Reference, or FVR, is a stable voltage reference, independent of VDD, with 1.024V, 2.048V or 4.096V selectable output levels. The output of the FVR can be configured to supply a reference voltage to the following:

- ADC input channel
- ADC positive reference
- Comparator positive input
- Digital-to-Analog Converter (DAC)

The FVR can be enabled by setting the FVREN bit of the VREFCON0 register.

### 21.1 Independent Gain Amplifiers

The output of the FVR supplied to the ADC, Comparators and DAC is routed through an independent programmable gain amplifier. The amplifier can be configured to amplify the 1.024V reference voltage by 1x, 2x or 4x, to produce the three possible voltage levels.

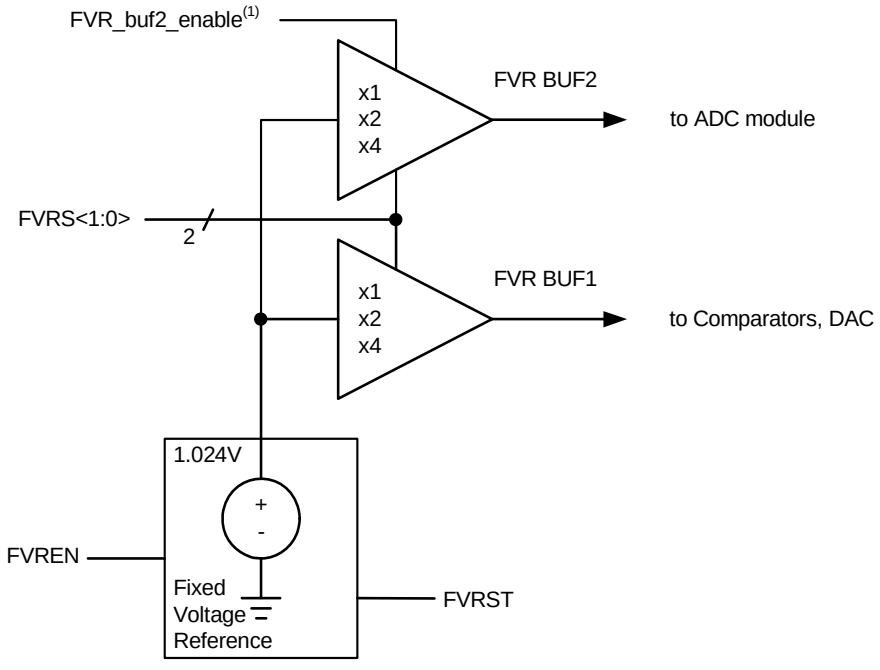
The FVRS<1:0> bits of the VREFCON0 register are used to enable and configure the gain amplifier settings for the reference supplied to the DAC and Comparator modules. When the ADC module is configured to use the FVR output, (FVR BUF2) the reference is buffered through an additional unity gain amplifier. This buffer is disabled if the ADC is not configured to use the FVR.

For specific use of the FVR, refer to the specific module sections: **Section 17.0 “Analog-to-Digital Converter (ADC) Module”**, **Section 22.0 “Digital-to-Analog Converter (DAC) Module”** and **Section 18.0 “Comparator Module”**.

### 21.2 FVR Stabilization Period

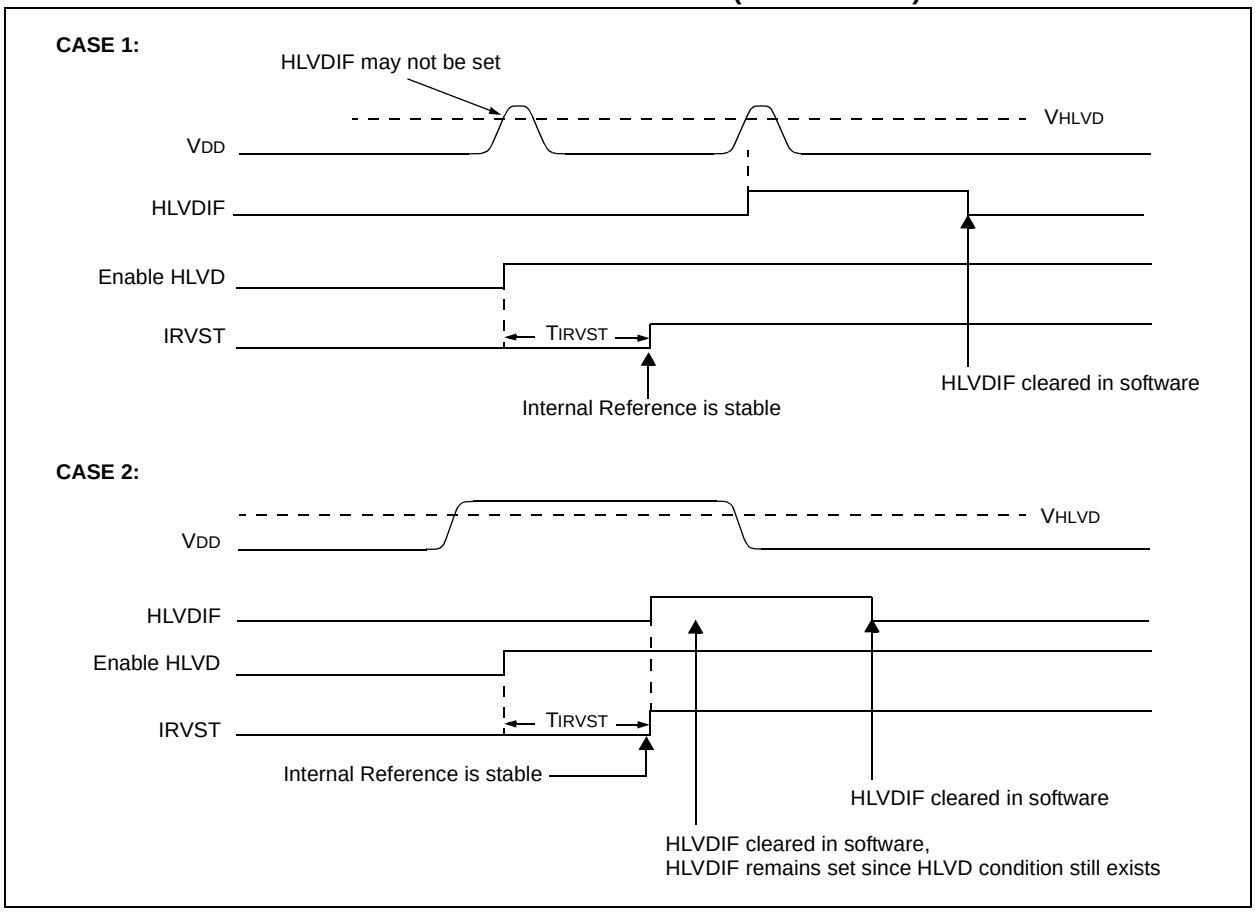
When the Fixed Voltage Reference module is enabled, it requires time for the reference and amplifier circuits to stabilize. Once the circuits stabilize and are ready for use, the FVRST bit of the VREFCON0 register will be set. See Table 27-3 for the minimum delay requirement.

**FIGURE 21-1: VOLTAGE REFERENCE BLOCK DIAGRAM**




**Note 1:** FVR_buf2_enable = ‘1’ when (ADON = ‘1’) AND [(PVCFG<1:0> = ‘10’) OR ( CHS<4:0> = ‘11111’)]

**FIGURE 23-3: HIGH-VOLTAGE DETECT OPERATION (VDIRMAG = 1)**




## 23.6 Applications

In many applications, it is desirable to detect a drop below, or rise above, a particular voltage threshold. For example, the HLVD module could be periodically enabled to detect Universal Serial Bus (USB) attach or detach. This assumes the device is powered by a lower voltage source than the USB when detached. An attach would indicate a high-voltage detect from, for example, 3.3V to 5V (the voltage on USB) and vice versa for a detach. This feature could save a design a few extra components and an attach signal (input pin).

For general battery applications, Figure 23-4 shows a possible voltage curve. Over time, the device voltage decreases. When the device voltage reaches voltage VA, the HLVD logic generates an interrupt at time, TA. The interrupt could cause the execution of an ISR, which would allow the application to perform "house-keeping tasks" and a controlled shutdown before the device voltage exits the valid operating range at TB. This would give the application a time window, represented by the difference between TA and TB, to safely exit.

**FIGURE 23-4: TYPICAL LOW-VOLTAGE DETECT APPLICATION**

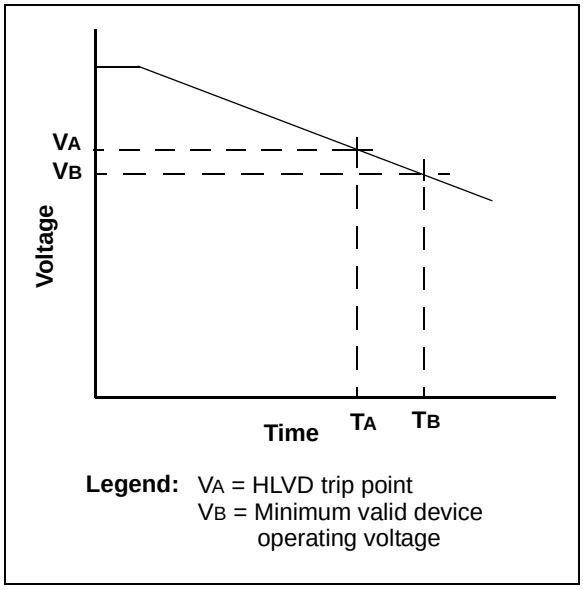


**TABLE 24-2: DEVICE ID TABLE FOR THE PIC18(L)F2X/4XK22 FAMILY**

| DEV<10:3> | DEV<2:0> | Part Number |
|---|---|---|
| 0101 0100 | 000 | PIC18F46K22 |
| | 001 | PIC18LF46K22 |
| | 010 | PIC18F26K22 |
| | 011 | PIC18LF26K22 |
| 0101 0101 | 000 | PIC18F45K22 |
| | 001 | PIC18LF45K22 |
| | 010 | PIC18F25K22 |
| | 011 | PIC18LF25K22 |
| 0101 0110 | 000 | PIC18F44K22 |
| | 001 | PIC18LF44K22 |
| | 010 | PIC18F24K22 |
| | 011 | PIC18LF24K22 |
| 0101 0111 | 000 | PIC18F43K22 |
| | 001 | PIC18LF43K22 |
| | 010 | PIC18F23K22 |
| | 011 | PIC18LF23K22 |

**NEGF** **Negate f**

Syntax: NEGF f {,a}

Operands: $0 \le f \le 255$
$a \in [0,1]$

Operation: $(\overline{f}) + 1 \rightarrow f$

Status Affected: N, OV, C, DC, Z

Encoding:

| | | | |
|---|---|---|---|
| 0110 | 110a | ffff | ffff |

Description: Location ‘f’ is negated using two’s complement. The result is placed in the data memory location ‘f’.
If ‘a’ is ‘0’, the Access Bank is selected.
If ‘a’ is ‘1’, the BSR is used to select the GPR bank.
If ‘a’ is ‘0’ and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \le 95$ (5Fh). See **Section 25.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode”** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | Read register ‘f’ | Process Data | Write register ‘f’ |

Example:
```
NEGF    REG, 1
```
Before Instruction
REG = 0011 1010 [3Ah]
After Instruction
REG = 1100 0110 [C6h]

**NOP** **No Operation**

Syntax: NOP

Operands: None

Operation: No operation

Status Affected: None

Encoding:

| | | | |
|---|---|---|---|
| 0000 1111 | 0000 xxxx | 0000 xxxx | 0000 xxxx |

Description: No operation.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | No operation | No operation | No operation |

Example:

None.

**POP** **Pop Top of Return Stack**

Syntax: POP

Operands: None

Operation: (TOS) → bit bucket

Status Affected: None

Encoding:

| | | | |
|---|---|---|---|
| 0000 | 0000 | 0000 | 0110 |

Description: The TOS value is pulled off the return stack and is discarded. The TOS value then becomes the previous value that was pushed onto the return stack. This instruction is provided to enable the user to properly manage the return stack to incorporate a software stack.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | No operation | POP TOS value | No operation |

Example:
```
POP
GOTO NEW
```
Before Instruction
TOS = 0031A2h
Stack (1 level down) = 014332h

After Instruction
TOS = 014332h
PC = NEW

**PUSH** **Push Top of Return Stack**

Syntax: PUSH

Operands: None

Operation: (PC + 2) → TOS

Status Affected: None

Encoding:

| | | | |
|---|---|---|---|
| 0000 | 0000 | 0000 | 0101 |

Description: The PC + 2 is pushed onto the top of the return stack. The previous TOS value is pushed down on the stack. This instruction allows implementing a software stack by modifying TOS and then pushing it onto the return stack.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | PUSH PC + 2 onto return stack | No operation | No operation |

Example:
```
PUSH
```
Before Instruction
TOS = 345Ah
PC = 0124h

After Instruction
PC = 0126h
TOS = 0126h
Stack (1 level down) = 345Ah

**TABLE 27-16: I²C BUS DATA REQUIREMENTS (SLAVE MODE)**

| Param. No. | Symbol | Characteristic | | Min | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 100 | THIGH | Clock High Time | 100 kHz mode | 4.0 | — | µs | Must operate at a minimum of 1.5 MHz |
| | | | 400 kHz mode | 0.6 | — | µs | Must operate at a minimum of 10 MHz |
| | | | SSP Module | 1.5 TCY | — | | |
| 101 | TLOW | Clock Low Time | 100 kHz mode | 4.7 | — | µs | Must operate at a minimum of 1.5 MHz |
| | | | 400 kHz mode | 1.3 | — | µs | Must operate at a minimum of 10 MHz |
| | | | SSP Module | 1.5 TCY | — | | |
| 102 | TR | SDA and SCL Rise Time | 100 kHz mode | — | 1000 | ns | |
| | | | 400 kHz mode | 20 + 0.1 CB | 300 | ns | CB is specified to be from 10 to 400 pF |
| 103 | TF | SDA and SCL Fall Time | 100 kHz mode | — | 300 | ns | |
| | | | 400 kHz mode | 20 + 0.1 CB | 300 | ns | CB is specified to be from 10 to 400 pF |
| 90 | TSU:STA | Start Condition Setup Time | 100 kHz mode | 4.7 | — | µs | Only relevant for Repeated Start condition |
| | | | 400 kHz mode | 0.6 | — | µs | |
| 91 | THD:STA | Start Condition Hold Time | 100 kHz mode | 4.0 | — | µs | After this period, the first clock pulse is generated |
| | | | 400 kHz mode | 0.6 | — | µs | |
| 106 | THD:DAT | Data Input Hold Time | 100 kHz mode | 0 | — | ns | |
| | | | 400 kHz mode | 0 | 0.9 | µs | |
| 107 | TSU:DAT | Data Input Setup Time | 100 kHz mode | 250 | — | ns | **(Note 2)** |
| | | | 400 kHz mode | 100 | — | ns | |
| 92 | TSU:STO | Stop Condition Setup Time | 100 kHz mode | 4.7 | — | µs | |
| | | | 400 kHz mode | 0.6 | — | µs | |
| 109 | TAA | Output Valid from Clock | 100 kHz mode | — | 3500 | ns | **(Note 1)** |
| | | | 400 kHz mode | — | — | ns | |
| 110 | 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 | |
| D102 | CB | Bus Capacitive Loading | | — | 400 | pF | |

**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 I²C bus device can be used in a standard mode 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 28-9: PIC18LF2X/4XK22 DELTA IPD SECONDARY OSCILLATOR**

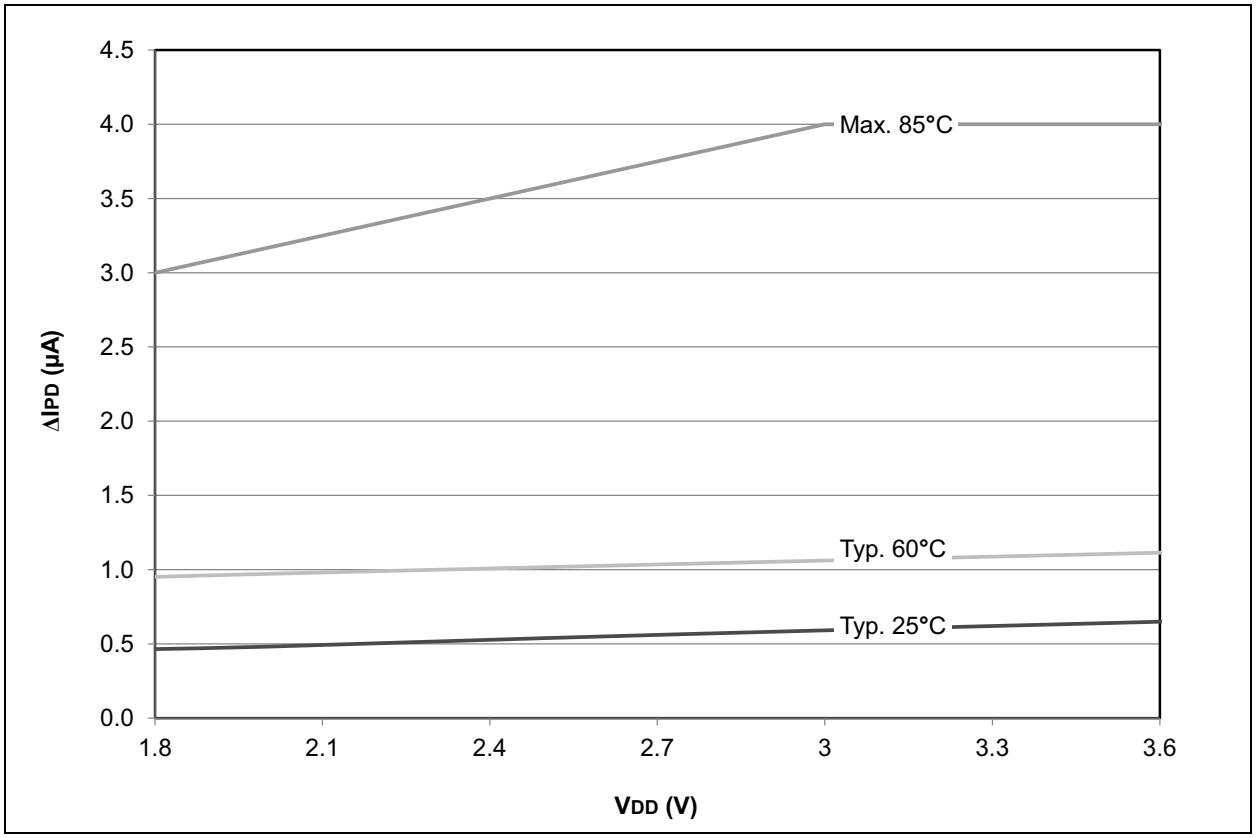



**FIGURE 28-10: PIC18F2X/4XK22 DELTA IPD SECONDARY OSCILLATOR**

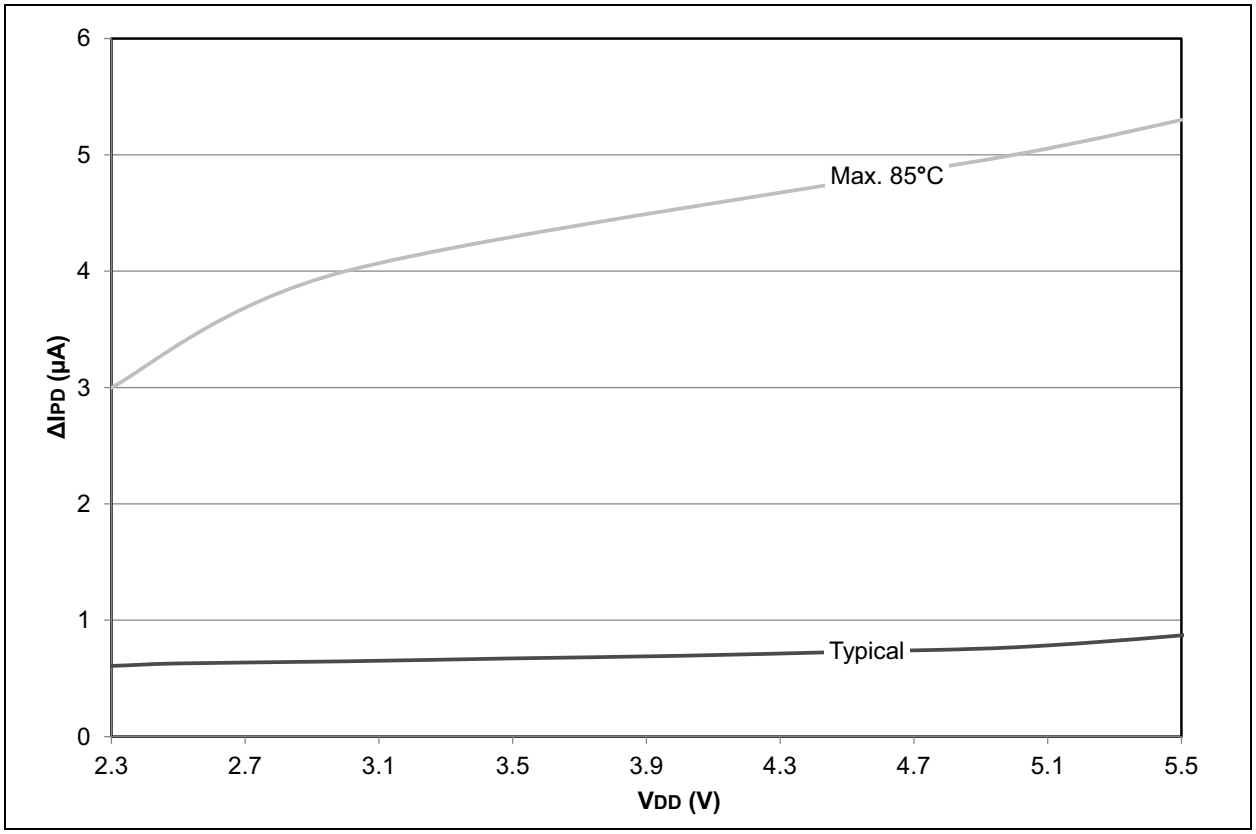


**FIGURE 28-24: PIC18LF2X/4XK22 IDD: RC_RUN MF-INTOSC 500 kHz**

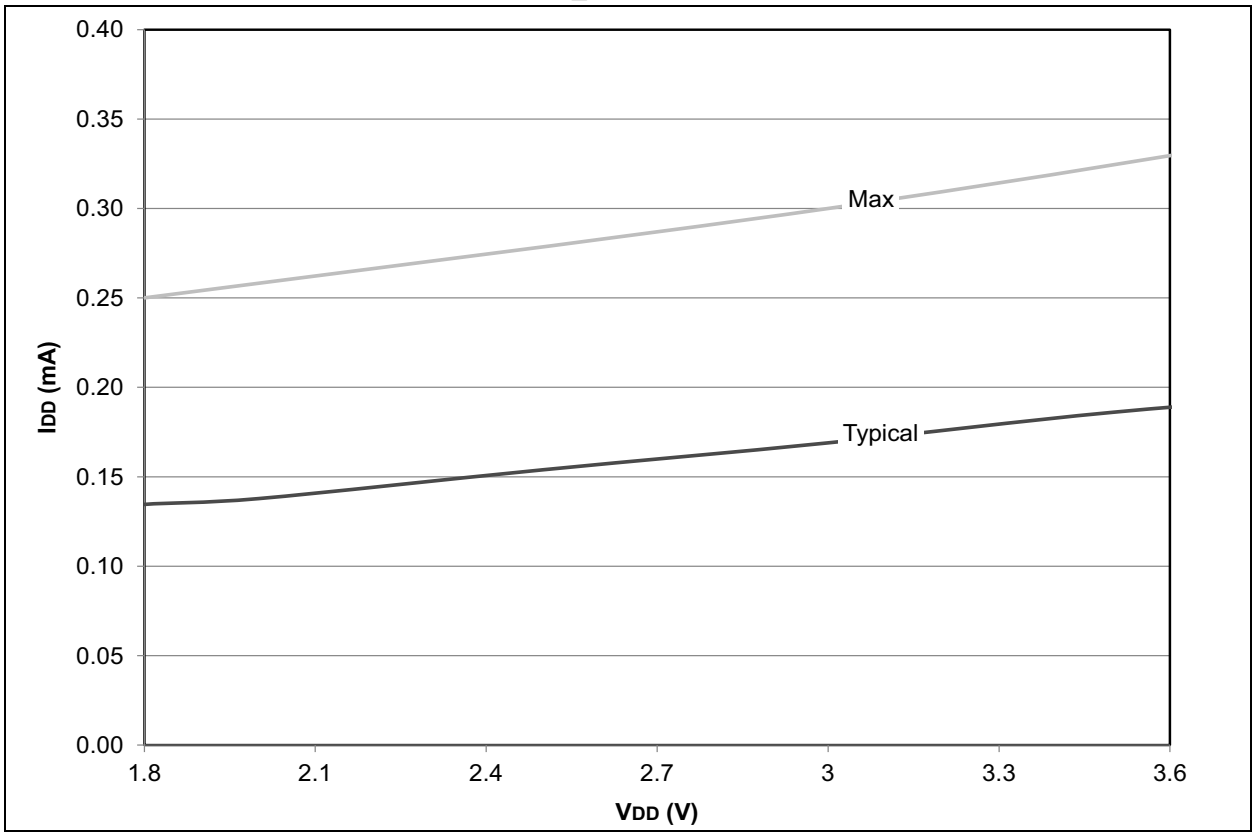



**FIGURE 28-25: PIC18F2X/4XK22 IDD: RC_RUN MF-INTOSC 500 kHz**

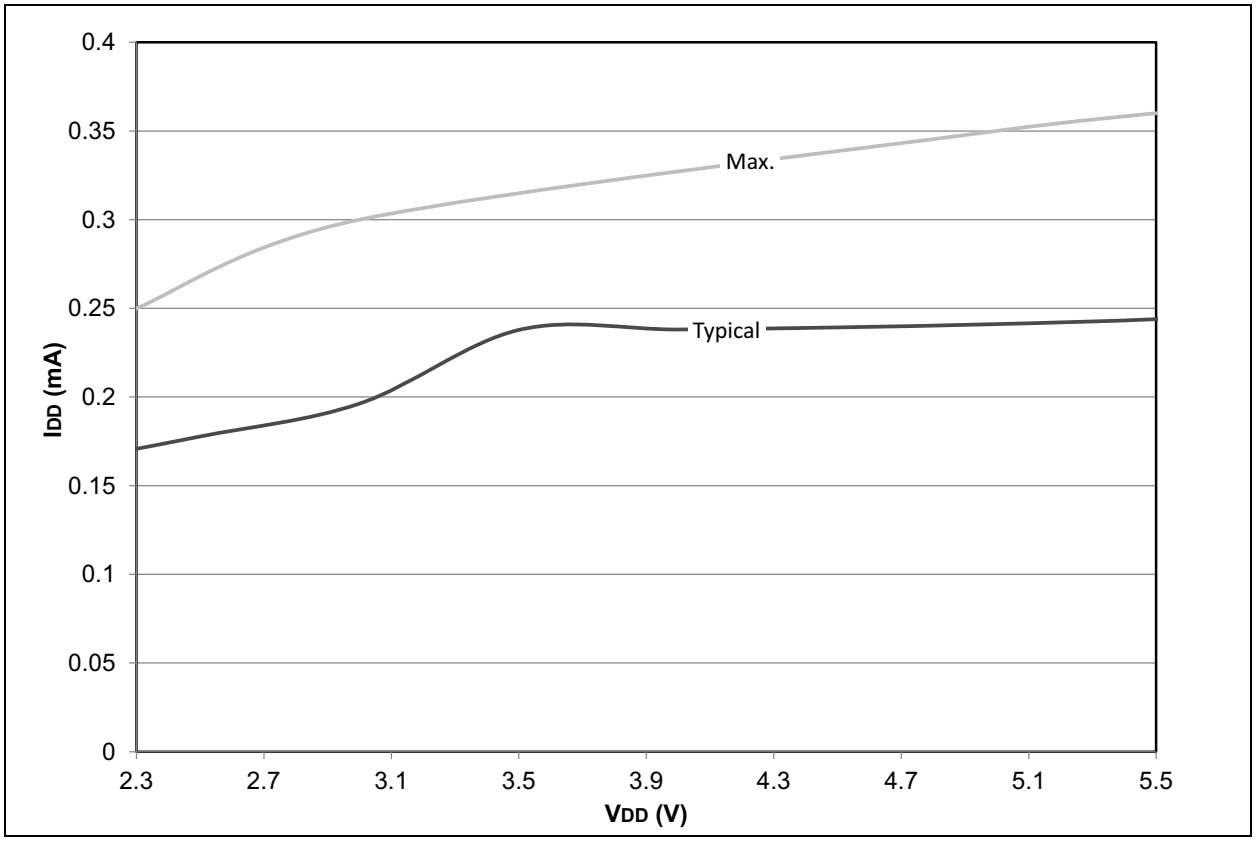


**FIGURE 28-32: PIC18F2X/4XK22 TYPICAL IDD: RC_RUN HF-INTOSC with PLL**

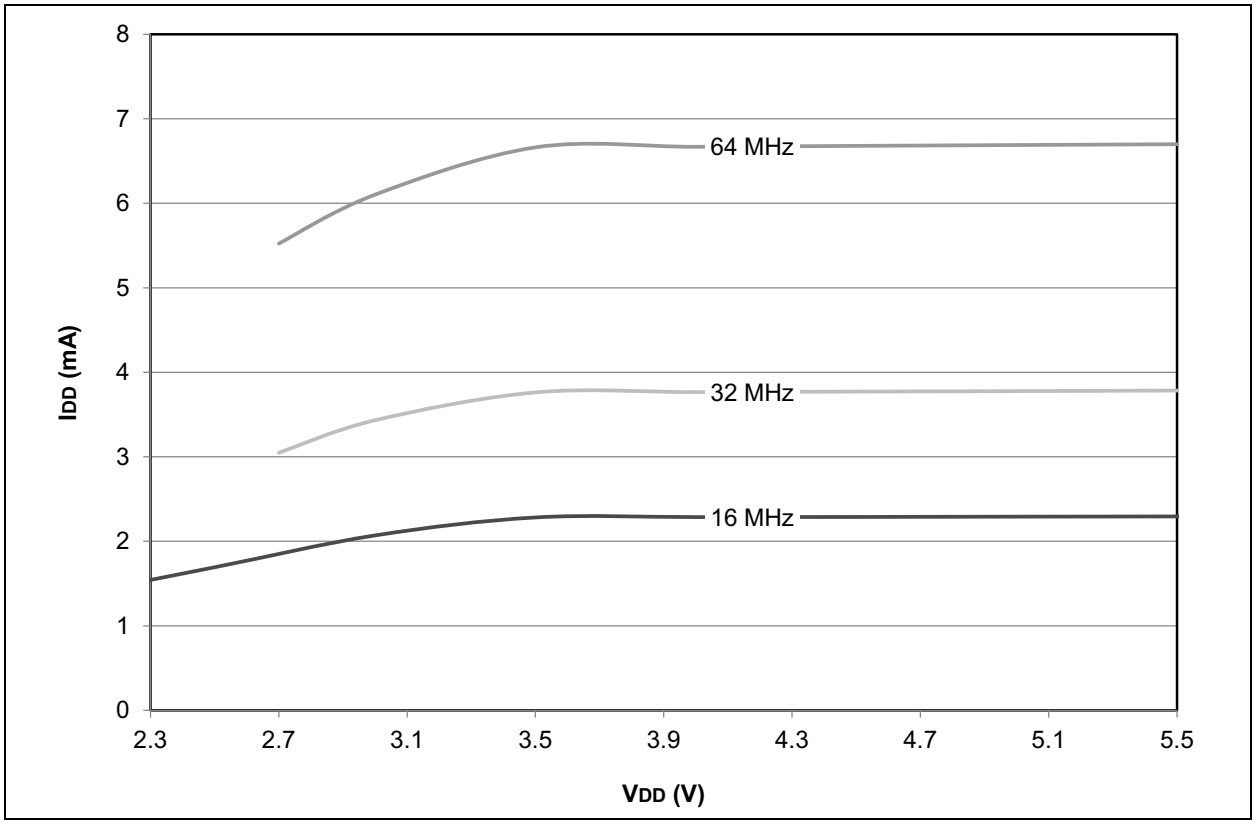



**FIGURE 28-33: PIC18F2X/4XK22 MAXIMUM IDD: RC_RUN HF-INTOSC with PLL**

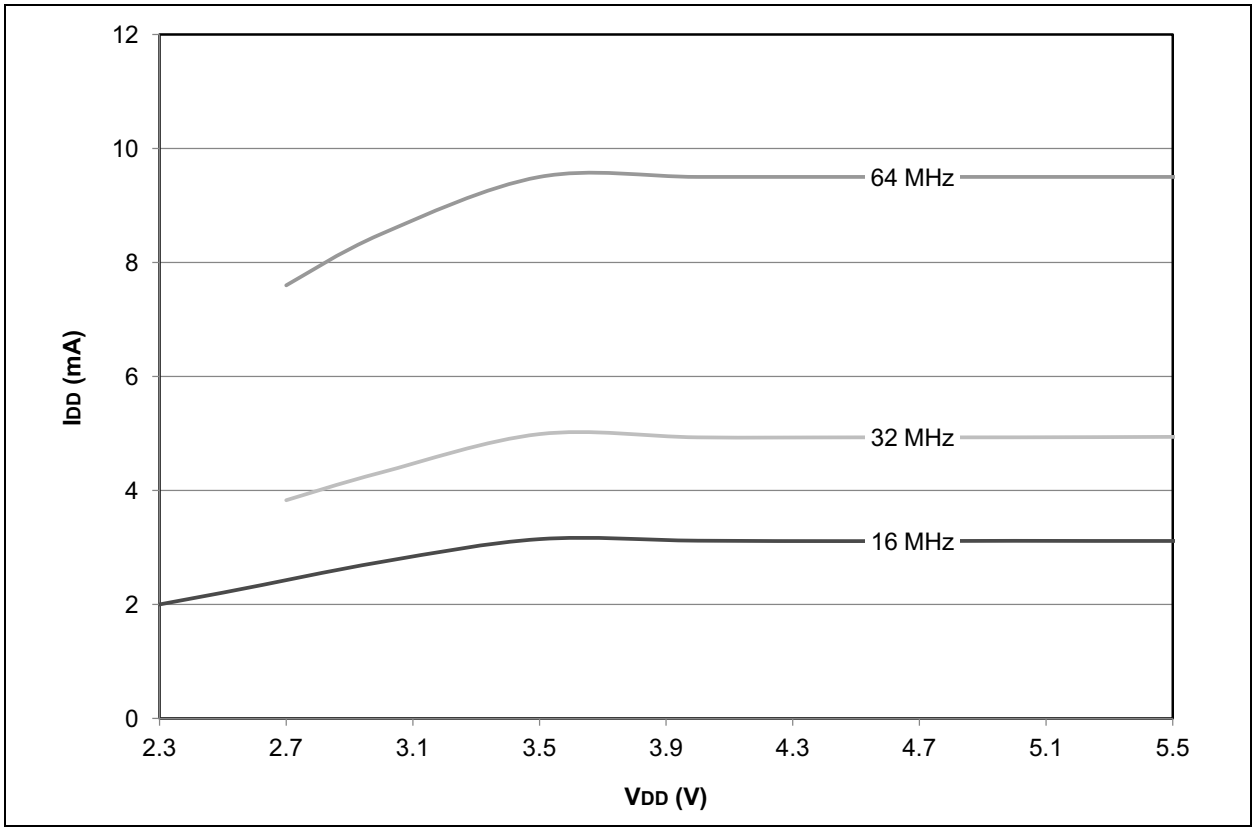


**FIGURE 28-74: PIC18F2X/4XK22 TYPICAL IDD: SEC_RUN 32.768 kHz**

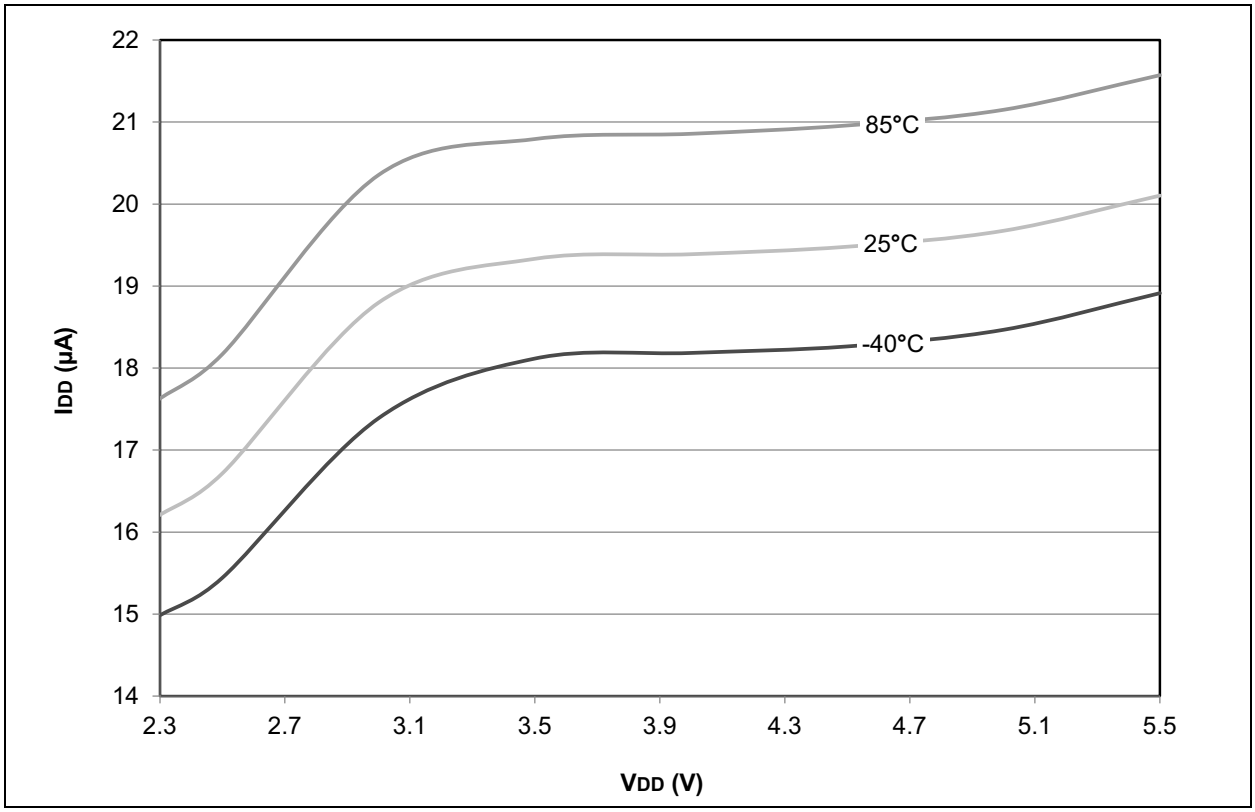
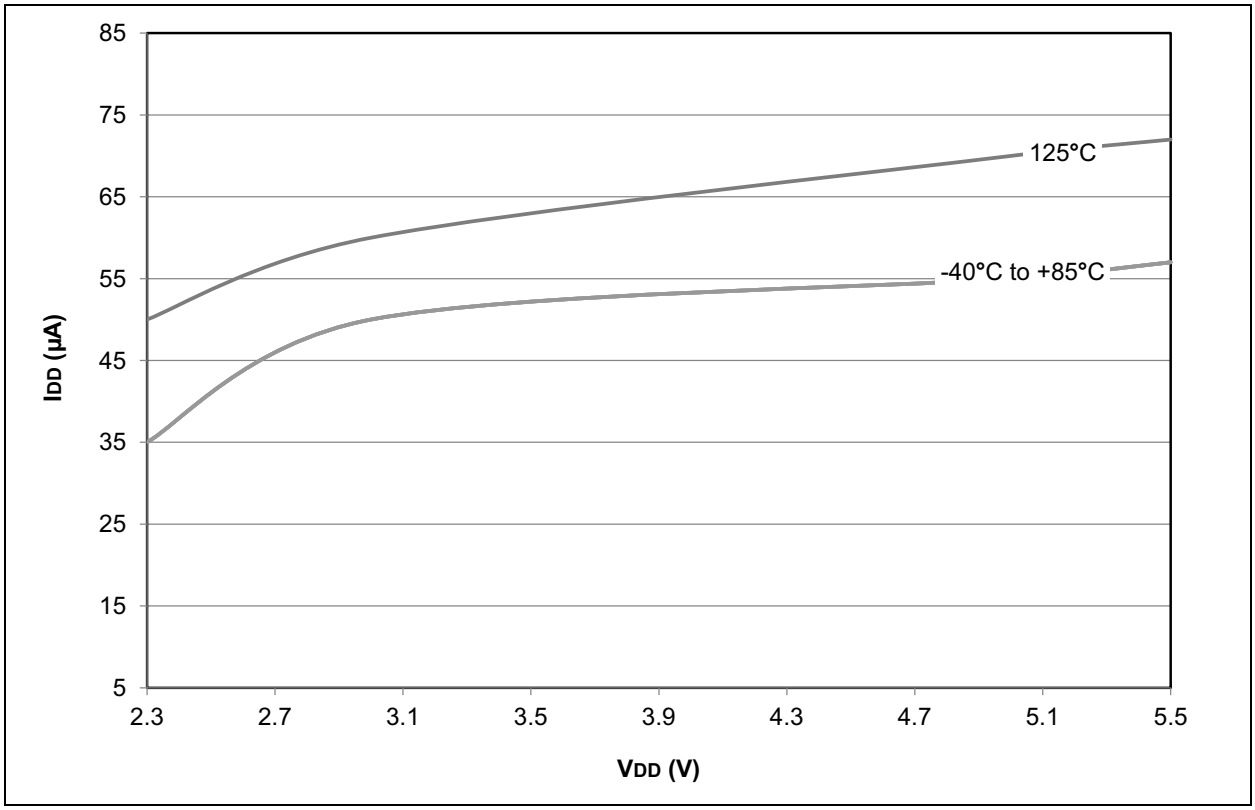
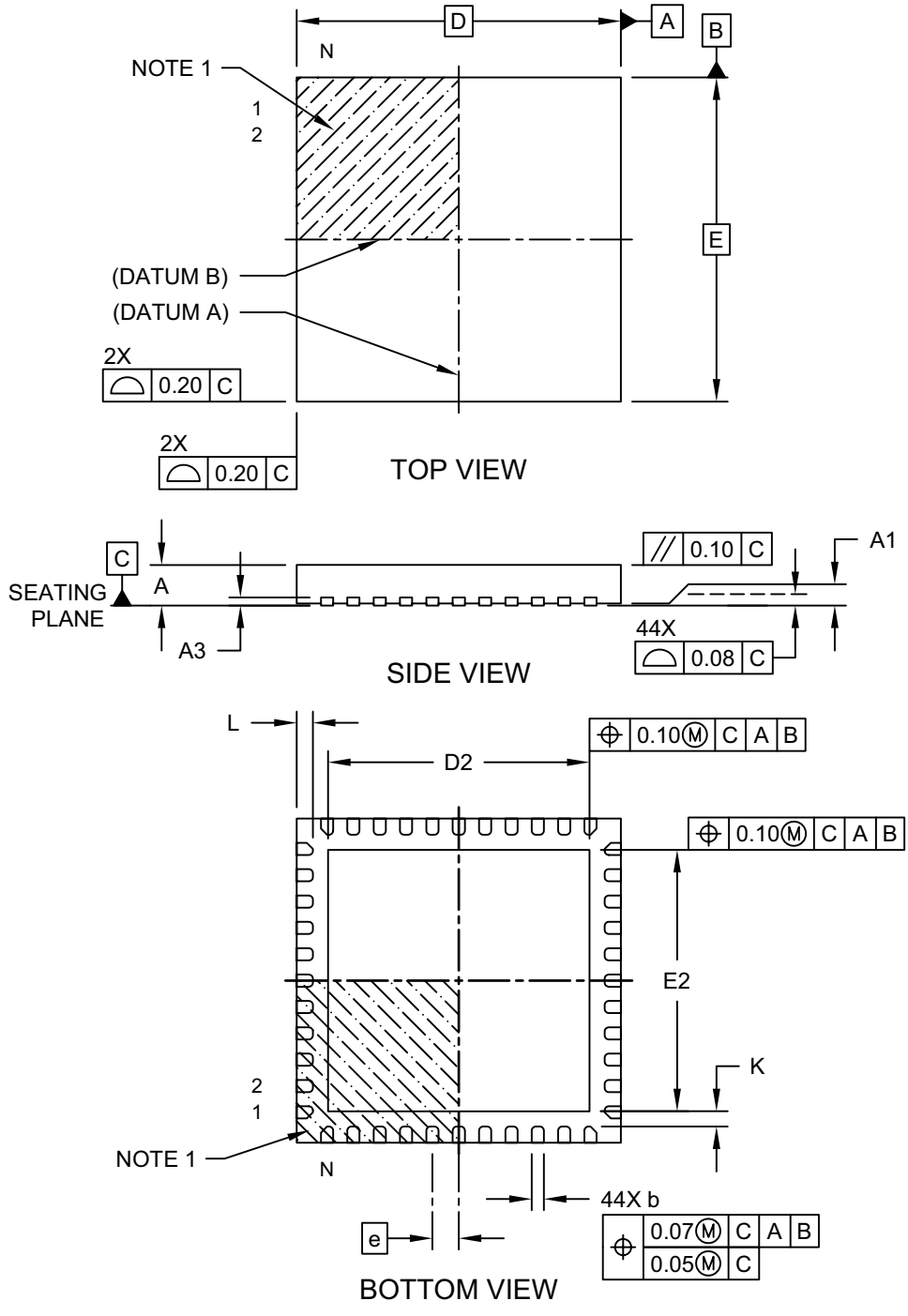



**FIGURE 28-75: PIC18F2X/4XK22 MAXIMUM IDD: SEC_RUN 32.768 kHz**



### 44-Lead Plastic Quad Flat, No Lead Package (ML) - 8x8 mm Body [QFN or VQFN]

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




Microchip Technology Drawing C04-103D Sheet 1 of 2