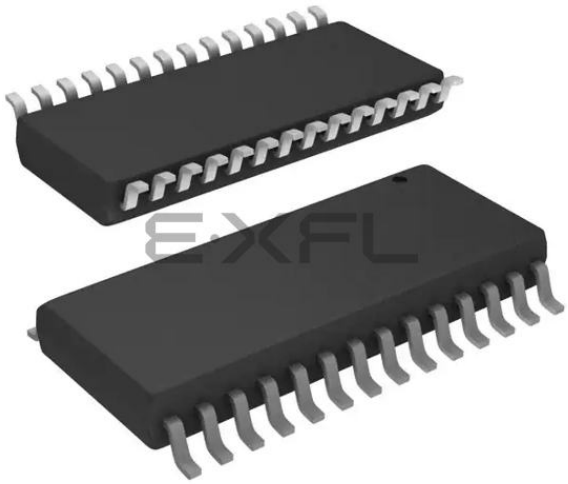


**Welcome to E-XFL.COM**

## 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 | 24 |
| 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) | 1.8V ~ 3.6V |
| Data Converters | A/D 19x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 125°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 28-SOIC (0.295", 7.50mm Width) |
| Supplier Device Package | 28-SOIC |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic18lf24k22-e-so |

**TABLE 2: PIC18(L)F2XK22 PIN SUMMARY**

| 28-SSOP, SOIC 28-SPDIP | 28-QFN, UQFN | I/O | Analog | Comparator | CTMU | SR Latch | Reference | (E)CCP | EUSART | MSSP | Timers | Interrupts | Pull-up | Basic |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 27 | RA0 | AN0 | C12IN0- | | | | | | | | | | |
| 3 | 28 | RA1 | AN1 | C12IN1- | | | | | | | | | | |
| 4 | 1 | RA2 | AN2 | C2IN+ | | | VREF-<br>DACOUT | | | | | | | |
| 5 | 2 | RA3 | AN3 | C1IN+ | | | VREF+ | | | | | | | |
| 6 | 3 | RA4 | | C1OUT | | SRQ | | CCP5 | | | T0CKI | | | |
| 7 | 4 | RA5 | AN4 | C2OUT | | SRNQ | HLVDIN | | | $\overline{SS}$1 | | | | |
| 10 | 7 | RA6 | | | | | | | | | | | | OSC2<br>CLKO |
| 9 | 6 | RA7 | | | | | | | | | | | | OSC1<br>CLKI |
| 21 | 18 | RB0 | AN12 | | | SRI | | CCP4<br>FLT0 | | $\overline{SS}$2 | | INT0 | Y | |
| 22 | 19 | RB1 | AN10 | C12IN3- | | | | P1C | | SCK2<br>SCL2 | | INT1 | Y | |
| 23 | 20 | RB2 | AN8 | | CTED1 | | | P1B | | SDI2<br>SDA2 | | INT2 | Y | |
| 24 | 21 | RB3 | AN9 | C12IN2- | CTED2 | | | CCP2<br>P2A(1) | | SDO2 | | | Y | |
| 25 | 22 | RB4 | AN11 | | | | | P1D | | | T5G | IOC | Y | |
| 26 | 23 | RB5 | AN13 | | | | | CCP3<br>P3A(4)<br>P2B(3) | | | T1G<br>T3CKI(2) | IOC | Y | |
| 27 | 24 | RB6 | | | | | | | TX2/CK2 | | | IOC | Y | PGC |
| 28 | 25 | RB7 | | | | | | | RX2/DT2 | | | IOC | Y | PGD |
| 11 | 8 | RC0 | | | | | | P2B(3) | | | SOSCO<br>T1CKI<br>T3CKI(2)<br>T3G | | | |
| 12 | 9 | RC1 | | | | | | CCP2<br>P2A(1) | | | SOSCI | | | |
| 13 | 10 | RC2 | AN14 | | CTPLS | | | CCP1<br>P1A | | | T5CKI | | | |
| 14 | 11 | RC3 | AN15 | | | | | | | SCK1<br>SCL1 | | | | |
| 15 | 12 | RC4 | AN16 | | | | | | | SDI1<br>SDA1 | | | | |
| 16 | 13 | RC5 | AN17 | | | | | | | SDO1 | | | | |
| 17 | 14 | RC6 | AN18 | | | | | CCP3<br>P3A(4) | TX1/CK1 | | | | | |
| 18 | 15 | RC7 | AN19 | | | | | P3B | RX1/DT1 | | | | | |
| 1 | 26 | RE3 | | | | | | | | | | | | $\overline{MCLR}$<br>VPP |
| 8, 19<br>19 | 5, 16<br>16 | VSS | | | | | | | | | | | | VSS |
| 20 | 17 | VDD | | | | | | | | | | | | VDD |

**Note 1:** CCP2/P2A multiplexed in fuses.
**2:** T3CKI multiplexed in fuses.
**3:** P2B multiplexed in fuses.
**4:** CCP3/P3A multiplexed in fuses.

**TABLE 1-2: PIC18(L)F2XK22 PINOUT I/O DESCRIPTIONS**

| Pin Number | | Pin Name | Pin Type | Buffer Type | Description |
|---|---|---|---|---|---|
| PDIP, SOIC | QFN, UQFN | | | | |
| 2 | 27 | RA0/C12IN0-/AN0 | | | |
| | | RA0 | I/O | TTL | Digital I/O. |
| | | C12IN0- | I | Analog | Comparators C1 and C2 inverting input. |
| | | AN0 | I | Analog | Analog input 0. |
| 3 | 28 | RA1/C12IN1-/AN1 | | | |
| | | RA1 | I/O | TTL | Digital I/O. |
| | | C12IN1- | I | Analog | Comparators C1 and C2 inverting input. |
| | | AN1 | I | Analog | Analog input 1. |
| 4 | 1 | RA2/C2IN+/AN2/DACOUT/$V_{REF-}$ | | | |
| | | RA2 | I/O | TTL | Digital I/O. |
| | | C2IN+ | I | Analog | Comparator C2 non-inverting input. |
| | | AN2 | I | Analog | Analog input 2. |
| | | DACOUT | O | Analog | DAC Reference output. |
| | | $V_{REF-}$ | I | Analog | A/D reference voltage (low) input. |
| 5 | 2 | RA3/C1IN+/AN3/$V_{REF+}$ | | | |
| | | RA3 | I/O | TTL | Digital I/O. |
| | | C1IN+ | I | Analog | Comparator C1 non-inverting input. |
| | | AN3 | I | Analog | Analog input 3. |
| | | $V_{REF+}$ | I | Analog | A/D reference voltage (high) input. |
| 6 | 3 | RA4/CCP5/C1OUT/SRQ/T0CKI | | | |
| | | RA4 | I/O | ST | Digital I/O. |
| | | CCP5 | I/O | ST | Capture 5 input/Compare 5 output/PWM 5 output. |
| | | C1OUT | O | CMOS | Comparator C1 output. |
| | | SRQ | O | TTL | SR latch Q output. |
| | | T0CKI | I | ST | Timer0 external clock input. |
| 7 | 4 | RA5/C2OUT/SRNQ/$\overline{SS1}$/HLVDIN/AN4 | | | |
| | | RA5 | I/O | TTL | Digital I/O. |
| | | C2OUT | O | CMOS | Comparator C2 output. |
| | | SRNQ | O | TTL | SR latch $\overline{Q}$ output. |
| | | $\overline{SS1}$ | I | TTL | SPI slave select input (MSSP). |
| | | HLVDIN | I | Analog | High/Low-Voltage Detect input. |
| | | AN4 | I | Analog | Analog input 4. |
| 10 | 7 | RA6/CLKO/OSC2 | | | |
| | | RA6 | I/O | TTL | Digital I/O. |
| | | CLKO | O | | In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate. |
| | | OSC2 | O | | Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. |

**Legend:** TTL = TTL compatible input CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; I = Input; O = Output; P = Power.

**Note 1:** Default pin assignment for P2B, T3CKI, CCP3 and CCP2 when Configuration bits PB2MX, T3CMX, CCP3MX and CCP2MX are set.

**2:** Alternate pin assignment for P2B, T3CKI, CCP3 and CCP2 when Configuration bits PB2MX, T3CMX, CCP3MX and CCP2MX are clear.

**FIGURE 5-5: DATA MEMORY MAP FOR PIC18(L)F23K22 AND PIC18(L)F43K22 DEVICES**

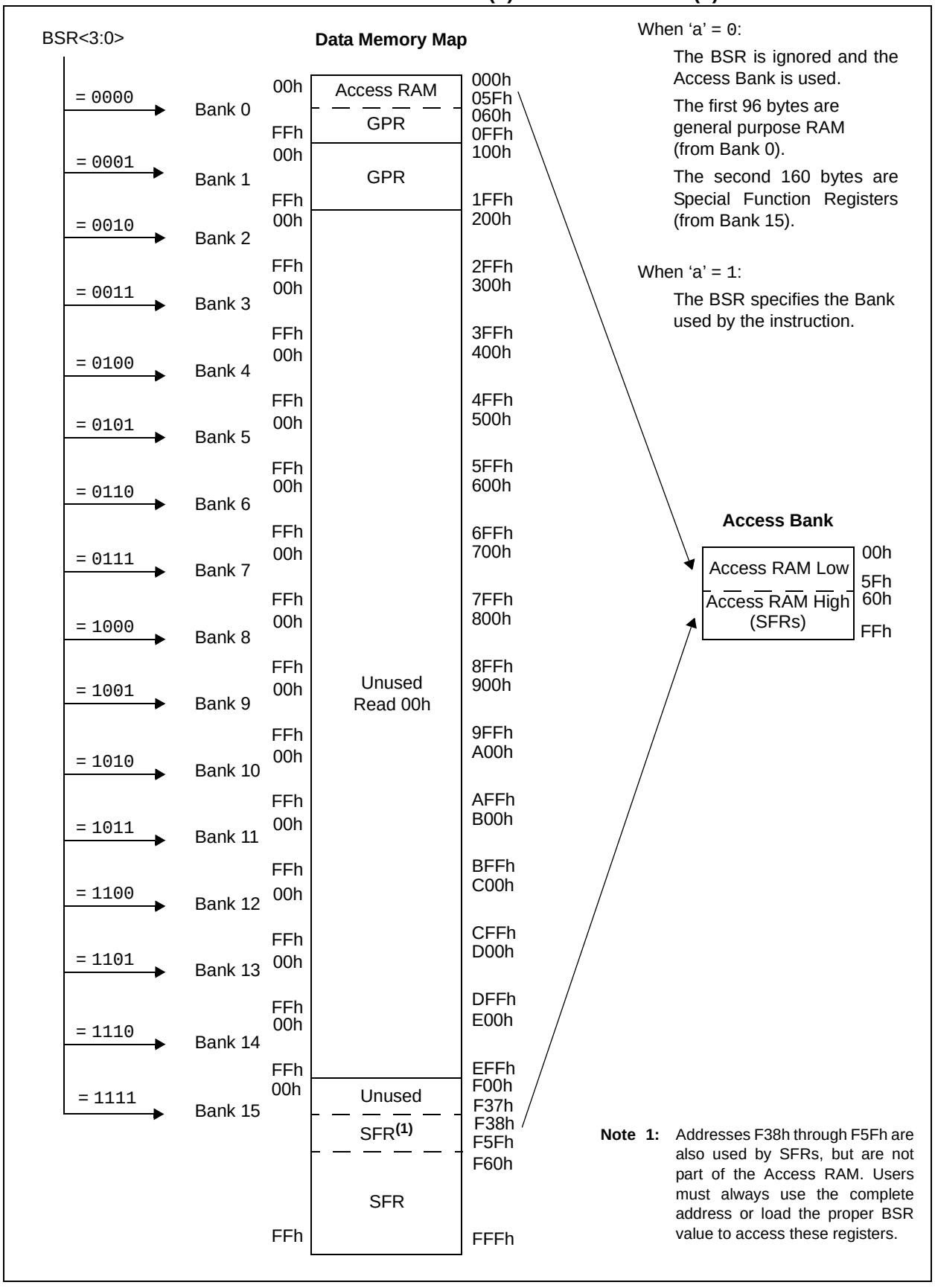


**TABLE 5-2: REGISTER FILE SUMMARY FOR PIC18(L)F2X/4XK22 DEVICES**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR |
|---|---|---|---|---|---|---|---|---|---|---|
| FFFh | TOSU | — | — | — | Top-of-Stack, Upper Byte (TOS<20:16>) | | | | | ---0 0000 |
| FFEh | TOSH | Top-of-Stack, High Byte (TOS<15:8>) | | | | | | | | 0000 0000 |
| FFDh | TOSL | Top-of-Stack, Low Byte (TOS<7:0>) | | | | | | | | 0000 0000 |
| FFCh | STKPTR | STKFUL | STKUNF | — | STKPTR<4:0> | | | | | 00-0 0000 |
| FFBh | PCLATU | — | — | — | Holding Register for PC<20:16> | | | | | ---0 0000 |
| FFAh | PCLATH | Holding Register for PC<15:8> | | | | | | | | 0000 0000 |
| FF9h | PCL | Holding Register for PC<7:0> | | | | | | | | 0000 0000 |
| FF8h | TBLPTRU | — | — | Program Memory Table Pointer Upper Byte(TBLPTR<21:16>) | | | | | | --00 0000 |
| FF7h | TBLPTRH | Program Memory Table Pointer High Byte(TBLPTR<15:8>) | | | | | | | | 0000 0000 |
| FF6h | TBLPTRL | Program Memory Table Pointer Low Byte(TBLPTR<7:0>) | | | | | | | | 0000 0000 |
| FF5h | TABLAT | Program Memory Table Latch | | | | | | | | 0000 0000 |
| FF4h | PRODH | Product Register, High Byte | | | | | | | | xxxx xxxx |
| FF3h | PRODL | Product Register, Low Byte | | | | | | | | xxxx xxxx |
| FF2h | INTCON | GIE/GIEH | PEIE/GIEL | TMR0IE | INT0IE | RBIE | TMR0IF | INT0IF | RBIF | 0000 000x |
| FF1h | INTCON2 | $\overline{\text{RBPU}}$ | INTEDG0 | INTEDG1 | INTEDG2 | — | TMR0IP | — | RBIP | 1111 -1-1 |
| FF0h | INTCON3 | INT2IP | INT1IP | — | INT2IE | INT1IE | — | INT2IF | INT1IF | 11-0 0-00 |
| FEFh | INDF0 | Uses contents of FSR0 to address data memory – value of FSR0 not changed (not a physical register) | | | | | | | | ---- ---- |
| FEEh | POSTINC0 | Uses contents of FSR0 to address data memory – value of FSR0 post-incremented (not a physical register) | | | | | | | | ---- ---- |
| FEDh | POSTDEC0 | Uses contents of FSR0 to address data memory – value of FSR0 post-decremented (not a physical register) | | | | | | | | ---- ---- |
| FECh | PREINC0 | Uses contents of FSR0 to address data memory – value of FSR0 pre-incremented (not a physical register) | | | | | | | | ---- ---- |
| FEBh | PLUSW0 | Uses contents of FSR0 to address data memory – value of FSR0 pre-incremented (not a physical register) – value of FSR0 offset by W | | | | | | | | ---- ---- |
| FEAh | FSR0H | — | — | — | — | Indirect Data Memory Address Pointer 0, High Byte | | | | ---- 0000 |
| FE9h | FSR0L | Indirect Data Memory Address Pointer 0, Low Byte | | | | | | | | xxxx xxxx |
| FE8h | WREG | Working Register | | | | | | | | xxxx xxxx |
| FE7h | INDF1 | Uses contents of FSR1 to address data memory – value of FSR1 not changed (not a physical register) | | | | | | | | ---- ---- |
| FE6h | POSTINC1 | Uses contents of FSR1 to address data memory – value of FSR1 post-incremented (not a physical register) | | | | | | | | ---- ---- |
| FE5h | POSTDEC1 | Uses contents of FSR1 to address data memory – value of FSR1 post-decremented (not a physical register) | | | | | | | | ---- ---- |
| FE4h | PREINC1 | Uses contents of FSR1 to address data memory – value of FSR1 pre-incremented (not a physical register) | | | | | | | | ---- ---- |
| FE3h | PLUSW1 | Uses contents of FSR1 to address data memory – value of FSR1 pre-incremented (not a physical register) – value of FSR1 offset by W | | | | | | | | ---- ---- |
| FE2h | FSR1H | — | — | — | — | Indirect Data Memory Address Pointer 1, High Byte | | | | ---- 0000 |
| FE1h | FSR1L | Indirect Data Memory Address Pointer 1, Low Byte | | | | | | | | xxxx xxxx |
| FE0h | BSR | — | — | — | — | Bank Select Register | | | | ---- 0000 |
| FDFh | INDF2 | Uses contents of FSR2 to address data memory – value of FSR2 not changed (not a physical register) | | | | | | | | ---- ---- |
| FDEh | POSTINC2 | Uses contents of FSR2 to address data memory – value of FSR2 post-incremented (not a physical register) | | | | | | | | ---- ---- |
| FDDh | POSTDEC2 | Uses contents of FSR2 to address data memory – value of FSR2 post-decremented (not a physical register) | | | | | | | | ---- ---- |
| FDCh | PREINC2 | Uses contents of FSR2 to address data memory – value of FSR2 pre-incremented (not a physical register) | | | | | | | | ---- ---- |
| FDBh | PLUSW2 | Uses contents of FSR2 to address data memory – value of FSR2 pre-incremented (not a physical register) – value of FSR2 offset by W | | | | | | | | ---- ---- |
| FDAh | FSR2H | — | — | — | — | Indirect Data Memory Address Pointer 2, High Byte | | | | ---- 0000 |
| FD9h | FSR2L | Indirect Data Memory Address Pointer 2, Low Byte | | | | | | | | xxxx xxxx |
| FD8h | STATUS | — | — | — | N | OV | Z | DC | C | ---x xxxx |
| FD7h | TMR0H | Timer0 Register, High Byte | | | | | | | | 0000 0000 |
| FD6h | TMR0L | Timer0 Register, Low Byte | | | | | | | | xxxx xxxx |
| FD5h | T0CON | TMR0ON | T08BIT | T0CS | T0SE | PSA | T0PS<2:0> | | | 1111 1111 |
| FD3h | OSCCON | IDLEN | IRCF<2:0> | | | OSTS | HFIOFS | SCS<1:0> | | 0011 q000 |
| FD2h | OSCCON2 | PLLRDY | SOSCRUN | — | MFIOSEL | SOSCGO | PRISD | MFIOFS | LFIOFS | 00-0 01x0 |

**Legend:** x = unknown, u = unchanged, — = unimplemented, q = value depends on condition

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

**2:** PIC18(L)F2XK22 devices only.

**3:** PIC18(L)F23/24K22 and PIC18(L)F43/44K22 devices only.

**4:** PIC18(L)F26K22 and PIC18(L)F46K22 devices only.

## 6.6 Writing to Flash Program Memory

The programming block size is 64 bytes. Word or byte programming is not supported.

Table writes are used internally to load the holding registers needed to program the Flash memory. There are only as many holding registers as there are bytes in a write block (64 bytes).

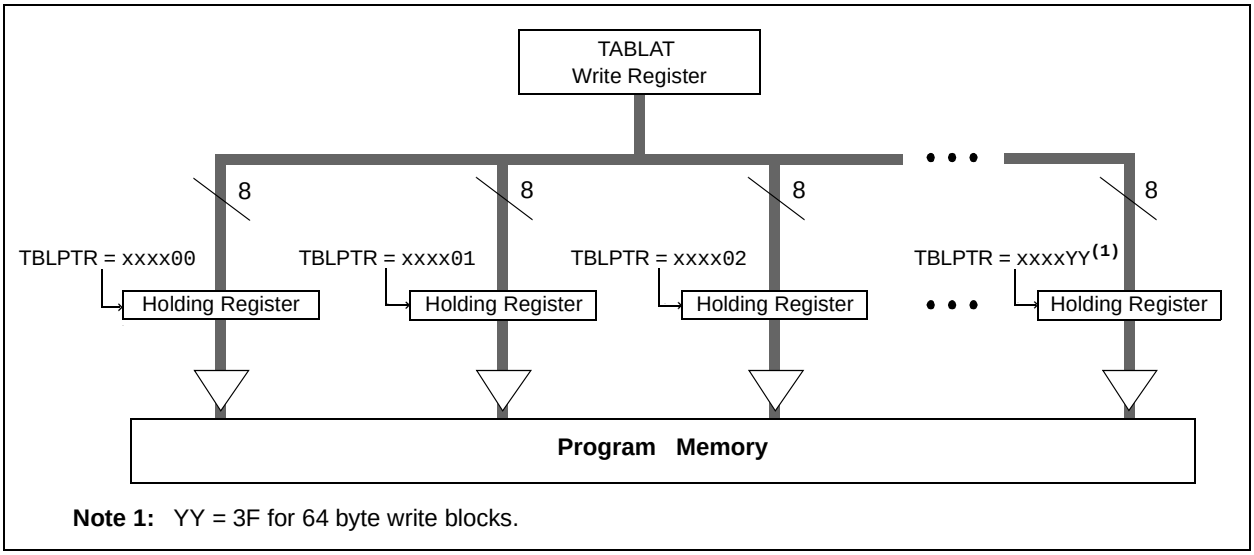
Since the Table Latch (TABLAT) is only a single byte, the TBLWT instruction needs to be executed 64 times for each programming operation. All of the table write operations will essentially be short writes because only the holding registers are written. After all the holding registers have been written, the programming operation of that block of memory is started by configuring the EECON1 register for a program memory write and performing the long write sequence.

The long write is necessary for programming the internal Flash. Instruction execution is halted during a long write cycle. The long write will be terminated by the internal programming timer.

The EEPROM on-chip timer controls the write time. The write/erase voltages are generated by an on-chip charge pump, rated to operate over the voltage range of the device.

| | |
|---|---|
| **Note:** | The default value of the holding registers on device Resets and after write operations is FFh. A write of FFh to a holding register does not modify that byte. This means that individual bytes of program memory may be modified, provided that the change does not attempt to change any bit from a '0' to a '1'. When modifying individual bytes, it is not necessary to load all holding registers before executing a long write operation. |

**FIGURE 6-5: TABLE WRITES TO FLASH PROGRAM MEMORY**




**Note 1:** YY = 3F for 64 byte write blocks.

### 6.6.1 FLASH PROGRAM MEMORY WRITE SEQUENCE

The sequence of events for programming an internal program memory location should be:

1. Read 64 bytes into RAM.
2. Update data values in RAM as necessary.
3. Load Table Pointer register with address being erased.
4. Execute the block erase procedure.
5. Load Table Pointer register with address of first byte being written.
6. Write the 64-byte block into the holding registers with auto-increment.
7. Set the EECON1 register for the write operation:
   - set EEPGD bit to point to program memory;
   - clear the CFGS bit to access program memory;
   - set WREN to enable byte writes.
8. Disable interrupts.
9. Write 55h to EECON2.
10. Write 0AAh to EECON2.
11. Set the WR bit. This will begin the write cycle.
12. The CPU will stall for duration of the write (about 2 ms using internal timer).
13. Re-enable interrupts.
14. Verify the memory (table read).

This procedure will require about 6 ms to update each write block of memory. An example of the required code is given in Example 6-3.

| | |
|---|---|
| **Note:** | Before setting the WR bit, the Table Pointer address needs to be within the intended address range of the bytes in the holding registers. |

## 14.0 CAPTURE/COMPARE/PWM MODULES

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

This family of devices contains three Enhanced Capture/Compare/PWM modules (ECCP1, ECCP2, and ECCP3) and two standard Capture/Compare/PWM modules (CCP4 and CCP5).

The Capture and Compare functions are identical for all CCP/ECCP modules. The difference between CCP and ECCP modules are in the Pulse-Width Modulation (PWM) function. In CCP modules, the standard PWM function is identical. In ECCP modules, the Enhanced PWM function has either full-bridge or half-bridge PWM output. Full-bridge ECCP modules have four available I/O pins while half-bridge ECCP modules only have two available I/O pins. ECCP PWM modules are backward compatible with CCP PWM modules and can be configured as standard PWM modules. See Table 14-1 to determine the CCP/ECCP functionality available on each device in this family.

**Note 1:** In devices with more than one CCP module, it is very important to pay close attention to the register names used. A number placed after the module acronym is used to distinguish between separate modules. For example, the CCP1CON and CCP2CON control the same operational aspects of two completely different CCP modules.

**2:** Throughout this section, generic references to a CCP module in any of its operating modes may be interpreted as being equally applicable to ECCP1, ECCP2, ECCP3, CCP4 and CCP5. Register names, module signals, I/O pins, and bit names may use the generic designator ‘x’ to indicate the use of a numeral to distinguish a particular module, when required.

**TABLE 14-1: PWM RESOURCES**

| Device Name | ECCP1 | ECCP2 | ECCP3 | CCP4 | CCP5 |
|---|---|---|---|---|---|
| PIC18(L)F23K22<br>PIC18(L)F24K22<br>PIC18(L)F25K22<br>PIC18(L)F26K22 | Enhanced PWM<br>Full-Bridge | Enhanced PWM<br>Half-Bridge | Enhanced PWM<br>Half-Bridge | Standard PWM | Standard PWM<br>(Special Event Trigger) |
| PIC18(L)F43K22<br>PIC18(L)F44K22<br>PIC18(L)F45K22<br>PIC18(L)F46K22 | Enhanced PWM<br>Full-Bridge | Enhanced PWM<br>Full-Bridge | Enhanced PWM<br>Half-Bridge | Standard PWM | Standard PWM<br>(Special Event Trigger) |

## 14.2 Compare Mode

The Compare mode function described in this section is identical for all CCP and ECCP modules available on this device family.

Compare mode makes use of the 16-bit TimerX resources, Timer1, Timer3 and Timer5. The 16-bit value of the CCPRxH:CCPRxL register pair is constantly compared against the 16-bit value of the TMRxH:TMRxL register pair. When a match occurs, one of the following events can occur:

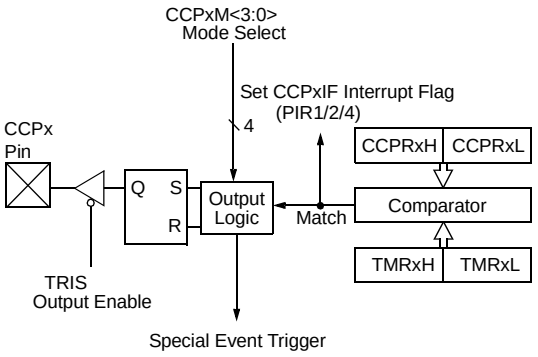
- Toggle the CCPx output
- Set the CCPx output
- Clear the CCPx output
- Generate a Special Event Trigger
- Generate a Software Interrupt

The action on the pin is based on the value of the CCPxM<3:0> control bits of the CCPxCON register. At the same time, the interrupt flag CCPxIF bit is set.

All Compare modes can generate an interrupt.

Figure 14-2 shows a simplified diagram of the Compare operation.

**FIGURE 14-2: COMPARE MODE OPERATION BLOCK DIAGRAM**




Special Event Trigger function on

- ECCP1, ECCP2, ECCP3, CCP4 and CCP5 will:
  - Reset TimerX – TMRxH:TMRxL = 0x0000
  - TimerX Interrupt Flag, (TMRxIF) is not set

Additional Function on

- CCP5 will
  - Set ADCON0<1>, GO/$\overline{\text{DONE}}$ bit to start an ADC Conversion if ADCON<0>, ADON = 1.

### 14.2.1 CCP PIN CONFIGURATION

The user must configure the CCPx pin as an output by clearing the associated TRIS bit.

Some CCPx outputs are multiplexed on a couple of pins. Table 14-2 shows the CCP output pin Multiplexing. Selection of the output pin is determined by the CCPxMX bits in Configuration register 3H (CONFIG3H). Refer to Register 24-4 for more details.

| **Note:** | Clearing the CCPxCON register will force the CCPx compare output latch to the default low level. This is not the PORT I/O data latch. |
|---|---|

### 14.2.2 TimerX MODE RESOURCE

In Compare mode, 16-bit TimerX resource must be running in either Timer mode or Synchronized Counter mode. The compare operation may not work in Asynchronous Counter mode.

See **Section 12.0 "Timer1/3/5 Module with Gate Control"** for more information on configuring the 16-bit TimerX resources.

| **Note:** | Clocking TimerX from the system clock (FOSC) should not be used in Compare mode. In order for Compare mode to recognize the trigger event on the CCPx pin, TimerX must be clocked from the instruction clock (FOSC/4) or from an external clock source. |
|---|---|

### 14.2.3 SOFTWARE INTERRUPT MODE

When Generate Software Interrupt mode is chosen (CCPxM<3:0> = 1010), the CCPx module does not assert control of the CCPx pin (see the CCPxCON register).

## 18.9 Register Definitions: Comparator Control

**REGISTER 18-1: CMxCON0: COMPARATOR x CONTROL REGISTER**

| R/W-0 | R-0 | R/W-0 | R/W-0 | R/W-1 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| CxON | CxOUT | CxOE | CxPOL | CxSP | CxR | CxCH<1:0> | |
| 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 **CxON:** Comparator Cx Enable bit
1 = Comparator Cx is enabled
0 = Comparator Cx is disabled

bit 6 **CxOUT:** Comparator Cx Output bit
If CxPOL = 1 (inverted polarity):
CxOUT = 0 when CxVIN+ > CxVIN-
CxOUT = 1 when CxVIN+ < CxVIN-
If CxPOL = 0 (non-inverted polarity):
CxOUT = 1 when CxVIN+ > CxVIN-
CxOUT = 0 when CxVIN+ < CxVIN-

bit 5 **CxOE:** Comparator Cx Output Enable bit
1 = CxOUT is present on the CxOUT pin(1)
0 = CxOUT is internal only

bit 4 **CxPOL:** Comparator Cx Output Polarity Select bit
1 = CxOUT logic is inverted
0 = CxOUT logic is not inverted

bit 3 **CxSP:** Comparator Cx Speed/Power Select bit
1 = Cx operates in Normal-Power, Higher Speed mode
0 = Cx operates in Low-Power, Low-Speed mode

bit 2 **CxR:** Comparator Cx Reference Select bit (non-inverting input)
1 = CxVIN+ connects to CxVREF output
0 = CxVIN+ connects to C12IN+ pin

bit 1-0 **CxCH<1:0>:** Comparator Cx Channel Select bit
00 = C12IN0- pin of Cx connects to CxVIN-
01 = C12IN1- pin of Cx connects to CxVIN-
10 = C12IN2- pin of Cx connects to CxVIN-
11 = C12IN3- pin of Cx connects to CxVIN-

**Note 1:** Comparator output requires the following three conditions: CxOE = 1, CxON = 1 and corresponding port TRIS bit = 0.

The module is enabled by setting the HLVDEN bit (HLVDCON<4>). Each time the HLVD module is enabled, the circuitry requires some time to stabilize. The IRVST bit (HLVDCON<5>) is a read-only bit used to indicate when the circuit is stable. The module can only generate an interrupt after the circuit is stable and IRVST is set.

The VDIRMAG bit (HLVDCON<7>) determines the overall operation of the module. When VDIRMAG is cleared, the module monitors for drops in VDD below a predetermined set point. When the bit is set, the module monitors for rises in VDD above the set point.

## 23.2 Operation

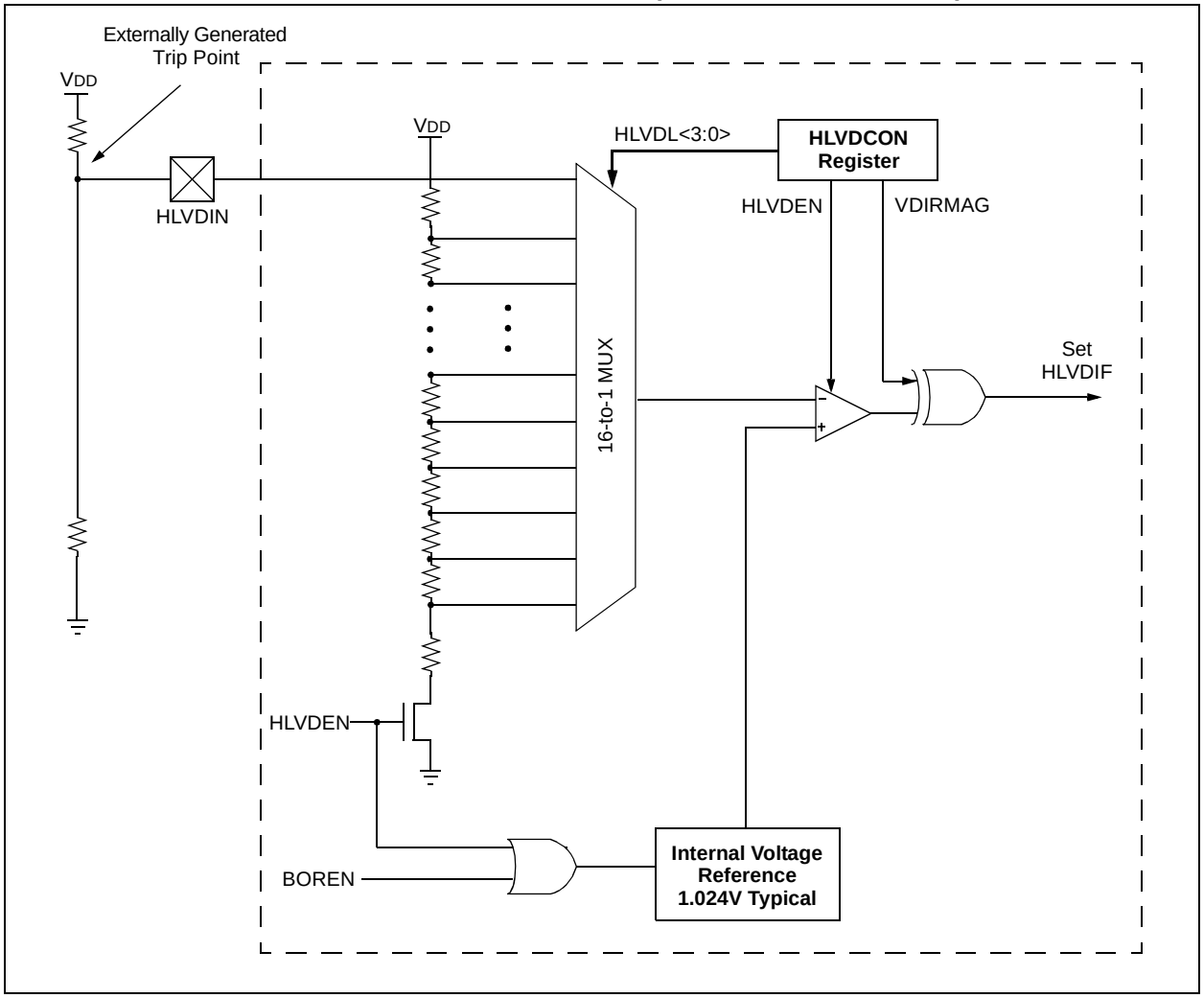
When the HLVD module is enabled, a comparator uses an internally generated reference voltage as the set point. The set point is compared with the trip point, where each node in the resistor divider represents a trip point voltage. The “trip point” voltage is the voltage level at which the device detects a high or low-voltage event, depending on the configuration of the module.

When the supply voltage is equal to the trip point, the voltage tapped off of the resistor array is equal to the internal reference voltage generated by the voltage reference module. The comparator then generates an interrupt signal by setting the HLVDIF bit.

The trip point voltage is software programmable to any of 16 values. The trip point is selected by programming the HLVDL<3:0> bits (HLVDCON<3:0>).

The HLVD module has an additional feature that allows the user to supply the trip voltage to the module from an external source. This mode is enabled when bits, HLVDL<3:0>, are set to ‘1111’. In this state, the comparator input is multiplexed from the external input pin, HLVDIN. This gives users the flexibility of configuring the High/Low-Voltage Detect interrupt to occur at any voltage in the valid operating range.

**FIGURE 23-1: HLVD MODULE BLOCK DIAGRAM (WITH EXTERNAL INPUT)**



### 25.1.1 STANDARD INSTRUCTION SET

**ADDLW** **ADD literal to W**

Syntax: ADDLW k

Operands: $0 \le k \le 255$

Operation: $(W) + k \rightarrow W$

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

Encoding:

| | | | |
|---|---|---|---|
| 0000 | 1111 | kkkk | kkkk |

Description: The contents of W are added to the 8-bit literal 'k' and the result is placed in W.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | Read literal 'k' | Process Data | Write to W |

Example: ADDLW 15h

Before Instruction
W = 10h
After Instruction
W = 25h

**ADDWF** **ADD W to f**

Syntax: ADDWF f {,d {,a}}

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

Operation: $(W) + (f) \rightarrow$ dest

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

Encoding:

| | | | |
|---|---|---|---|
| 0010 | 01da | ffff | ffff |

Description: Add W to register 'f'. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default).
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 to destination |

Example: ADDWF REG, 0, 0

Before Instruction
W = 17h
REG = 0C2h
After Instruction
W = 0D9h
REG = 0C2h

**Note:** All PIC18 instructions may take an optional label argument preceding the instruction mnemonic for use in symbolic addressing. If a label is used, the instruction format then becomes: {label} instruction argument(s).

**TBLRD Table Read**

Syntax: TBLRD ( *; *+; *-; +*)

Operands: None

Operation: if TBLRD *,
(Prog Mem (TBLPTR)) → TABLAT;
TBLPTR – No Change;
if TBLRD *+,
(Prog Mem (TBLPTR)) → TABLAT;
(TBLPTR) + 1 → TBLPTR;
if TBLRD *-,
(Prog Mem (TBLPTR)) → TABLAT;
(TBLPTR) – 1 → TBLPTR;
if TBLRD +*,
(TBLPTR) + 1 → TBLPTR;
(Prog Mem (TBLPTR)) → TABLAT;

Status Affected: None

Encoding:

| | | | |
|---|---|---|---|
| 0000 | 0000 | 0000 | 10nn<br>nn=0 *<br>=1 *+<br>=2 *-<br>=3 +* |

Description: This instruction is used to read the contents of Program Memory (P.M.). To address the program memory, a pointer called Table Pointer (TBLPTR) is used.
The TBLPTR (a 21-bit pointer) points to each byte in the program memory. TBLPTR has a 2-Mbyte address range.

TBLPTR[0] = 0: Least Significant Byte of Program Memory Word
TBLPTR[0] = 1: Most Significant Byte of Program Memory Word

The TBLRD instruction can modify the value of TBLPTR as follows:

- no change
- post-increment
- post-decrement
- pre-increment

Words: 1

Cycles: 2

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | No operation | No operation | No operation |
| No operation | No operation (Read Program Memory) | No operation | No operation (Write TABLAT) |

**TBLRD Table Read (Continued)**

Example1:
```
TBLRD   *+ ;
```
Before Instruction
TABLAT = 55h
TBLPTR = 00A356h
MEMORY (00A356h) = 34h
After Instruction
TABLAT = 34h
TBLPTR = 00A357h

Example2:
```
TBLRD   +* ;
```
Before Instruction
TABLAT = AAh
TBLPTR = 01A357h
MEMORY (01A357h) = 12h
MEMORY (01A358h) = 34h
After Instruction
TABLAT = 34h
TBLPTR = 01A358h

## 26.2 MPLAB XC Compilers

The MPLAB XC Compilers are complete ANSI C compilers for all of Microchip's 8, 16, and 32-bit MCU and DSC devices. These compilers provide powerful integration capabilities, superior code optimization and ease of use. MPLAB XC Compilers run on Windows, Linux or MAC OS X.

For easy source level debugging, the compilers provide debug information that is optimized to the MPLAB X IDE.

The free MPLAB XC Compiler editions support all devices and commands, with no time or memory restrictions, and offer sufficient code optimization for most applications.

MPLAB XC Compilers include an assembler, linker and utilities. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. MPLAB XC Compiler uses the assembler to produce its object file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

## 26.3 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code, and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB X IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multipurpose source files
- Directives that allow complete control over the assembly process

## 26.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

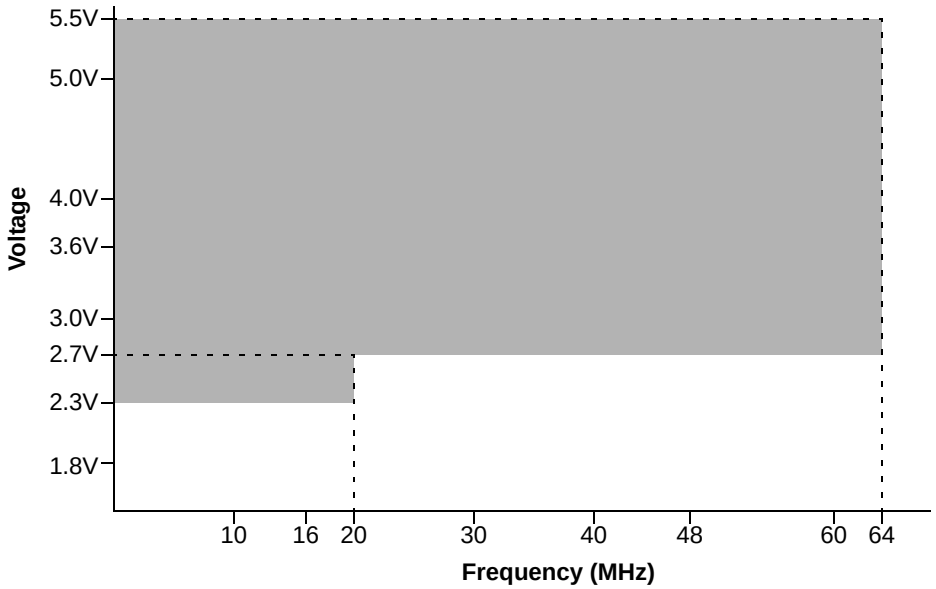
- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

## 26.5 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC DSC devices. MPLAB XC Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

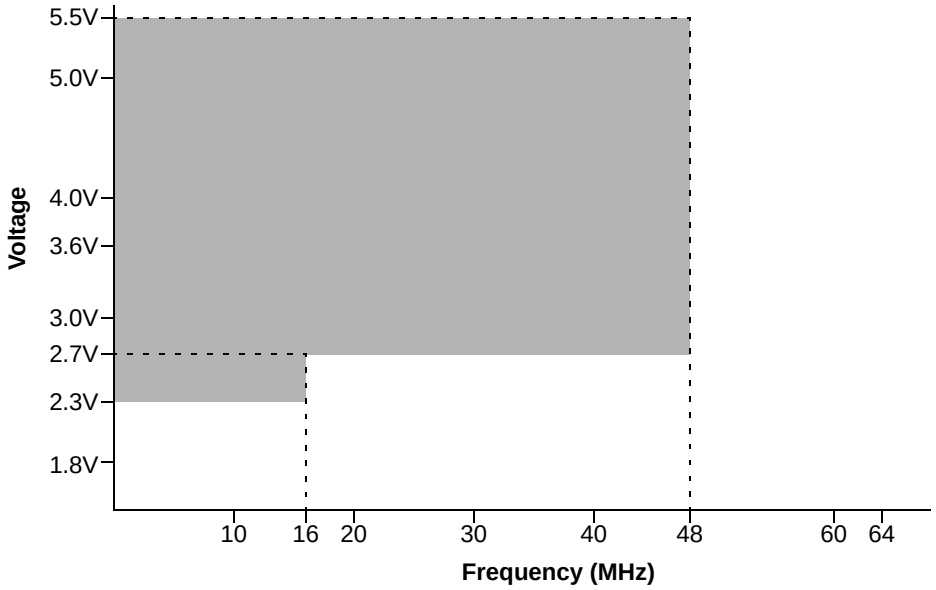
**FIGURE 27-3: PIC18F2X/4XK22 FAMILY VOLTAGE-FREQUENCY GRAPH (INDUSTRIAL TEMPERATURE)**




**Note 1:** Maximum Frequency 20 MHz, 2.3V to 2.7V, -40°C to +85°C

**2:** Maximum Frequency 64 MHz, 2.7V to 5.5V, -40°C to +85°C

**FIGURE 27-4: PIC18F2X/4XK22 FAMILY VOLTAGE-FREQUENCY GRAPH (EXTENDED TEMPERATURE)**




**Note 1:** Maximum Frequency 16 MHz, 2.3V to 2.7V, +85°C to +125°C

**2:** Maximum Frequency 48 MHz, 2.7V to 5.5V, +85°C to +125°C

## 27.4 DC Characteristics: RC Idle Supply Current, PIC18(L)F2X/4XK22

| PIC18LF2X/4XK22 | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +125°C | | | | | |
|---|---|---|---|---|---|---|---|
| **PIC18F2X/4XK22** | | **Standard Operating Conditions (unless otherwise stated)**<br>Operating temperature -40°C ≤ TA ≤ +125°C | | | | | |
| **Param No.** | **Device Characteristics** | **Typ** | **Max** | **Units** | **Conditions** | | |
| D045 | Supply Current (IDD)(1),(2) | 0.5 | 18 | µA | -40°C | VDD = 1.8V | FOSC = 31 kHz (**RC_IDLE** mode, LFINTOSC source) |
| | | 0.6 | 18 | µA | +25°C | | |
| | | 0.7 | — | µA | +60°C | | |
| | | 0.75 | 20 | µA | +85°C | | |
| | | 2.3 | 22 | µA | +125°C | | |
| D046 | | 1.1 | 20 | µA | -40°C | VDD = 3.0V | |
| | | 1.2 | 20 | µA | +25°C | | |
| | | 1.3 | — | µA | +60°C | | |
| | | 1.4 | 22 | µA | +85°C | | |
| | | 3.2 | 25 | µA | +125°C | | |
| D047 | | 17 | 30 | µA | -40°C | VDD = 2.3V | FOSC = 31 kHz (**RC_IDLE** mode, LFINTOSC source) |
| | | 13 | 30 | µA | +25°C | | |
| | | 14 | 30 | µA | +85°C | | |
| | | 15 | 45 | µA | +125°C | | |
| D048 | | 19 | 35 | µA | -40°C | VDD = 3.0V | |
| | | 15 | 35 | µA | +25°C | | |
| | | 16 | 35 | µA | +85°C | | |
| | | 17 | 50 | µA | +125°C | | |
| D049 | | 21 | 40 | µA | -40°C | VDD = 5.0V | |
| | | 15 | 40 | µA | +25°C | | |
| | | 16 | 40 | µA | +85°C | | |
| | | 18 | 60 | µA | +125°C | | |
| D050 | | 0.11 | 0.20 | mA | -40°C to +125°C | VDD = 1.8V | FOSC = 500 kHz (**RC_IDLE** mode, MFINTOSC source) |
| D051 | | 0.12 | 0.25 | mA | -40°C to +125°C | VDD = 3.0V | |
| D052 | | 0.14 | 0.21 | mA | -40°C to +125°C | VDD = 2.3V | FOSC = 500 kHz (**RC_IDLE** mode, MFINTOSC source) |
| D053 | | 0.15 | 0.25 | mA | -40°C to +125°C | VDD = 3.0V | |
| D054 | | 0.20 | 0.31 | mA | -40°C to +125°C | VDD = 5.0V | |

**Note 1:** The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption.

Test condition: All Peripheral Module Control bits in PMD0, PMD1 and PMD2 set to '1'.

**2:** The test conditions for all IDD measurements in active operation mode are:

All I/O pins set as outputs driven to Vss;

$\overline{MCLR}$ = VDD;

OSC1 = external square wave, from rail-to-rail (PRI_RUN and PRI_IDLE only).

27.11.3 TIMING DIAGRAMS AND SPECIFICATIONS

**FIGURE 27-7: EXTERNAL CLOCK TIMING (ALL MODES EXCEPT PLL)**

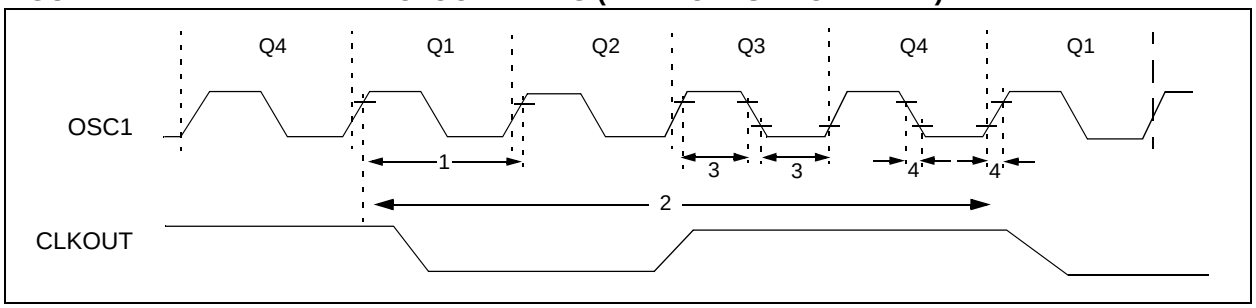



**TABLE 27-7: EXTERNAL CLOCK TIMING REQUIREMENTS**

| Param. No. | Symbol | Characteristic | Min | Max | Units | Conditions |
|---|---|---|---|---|---|---|
| 1A | FOSC | External CLKIN Frequency(1) | DC | 0.5 | MHz | EC, ECIO Oscillator mode (low power) |
| | | | DC | 16 | MHz | EC, ECIO Oscillator mode (medium power) |
| | | | DC | 64 | MHz | EC, ECIO Oscillator mode (high power) |
| | | Oscillator Frequency(1) | DC | 4 | MHz | RC Oscillator mode |
| | | | 5 | 200 | kHz | LP Oscillator mode |
| | | | 0.1 | 4 | MHz | XT Oscillator mode |
| | | | 4 | 4 | MHz | HS Oscillator mode, VDD < 2.7V |
| | | | 4 | 16 | MHz | HS Oscillator mode, VDD ≥ 2.7V, Medium-Power mode (HSMP) |
| | | | 4 | 20 | MHz | HS Oscillator mode, VDD ≥ 2.7V, High-Power mode (HSHP) |
| 1 | TOSC | External CLKIN Period(1) | 2.0 | — | µs | EC, ECIO Oscillator mode (low power) |
| | | | 62.5 | — | ns | EC, ECIO Oscillator mode (medium power) |
| | | | 15.6 | — | ns | EC, ECIO Oscillator mode (high power) |
| | | Oscillator Period(1) | 250 | — | ns | RC Oscillator mode |
| | | | 5 | 200 | µs | LP Oscillator mode |
| | | | 0.25 | 10 | µs | XT Oscillator mode |
| | | | 250 | 250 | ns | HS Oscillator mode, VDD < 2.7V |
| | | | 62.5 | 250 | ns | HS Oscillator mode, VDD ≥ 2.7V, Medium-Power mode (HSMP) |
| | | | 50 | 250 | ns | HS Oscillator mode, VDD ≥ 2.7V, High-Power mode (HSHP) |
| 2 | TCY | Instruction Cycle Time(1) | 62.5 | — | ns | TCY = 4/FOSC |
| 3 | TOSL, TOSH | External Clock in (OSC1) High or Low Time | 2.5 | — | µs | LP Oscillator mode |
| | | | 30 | — | ns | XT Oscillator mode |
| | | | 10 | — | ns | HS Oscillator mode |
| 4 | TOSR, TOSF | External Clock in (OSC1) Rise or Fall Time | — | 50 | ns | LP Oscillator mode |
| | | | — | 20 | ns | XT Oscillator mode |
| | | | — | 7.5 | ns | HS Oscillator mode |

**Note 1:** Instruction cycle period (TCY) equals four times the input oscillator time base period for all configurations except PLL. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at “min.” values with an external clock applied to the OSC1/CLKIN pin. When an external clock input is used, the “max.” cycle time limit is “DC” (no clock) for all devices.

**TABLE 27-8: PLL CLOCK TIMING SPECIFICATIONS**

| Param. No. | Sym | Characteristic | Min | Max | Units | Conditions |
|---|---|---|---|---|---|---|
| F10 | FOSC | Oscillator Frequency Range | 4 | 5 | MHz | VDD < 2.7V, -40°C to +85°C |
| | | | 4 | 4 | MHz | VDD < 2.7V, +85°C to +125°C |
| | | | 4 | 16 | MHz | 2.7V ≤ VDD, -40°C to +85°C |
| | | | 4 | 12 | MHz | 2.7V ≤ VDD, +85°C to +125°C |
| F11 | FSYS | On-Chip VCO System Frequency | 16 | 20 | MHz | VDD < 2.7V, -40°C to +85°C |
| | | | 16 | 16 | MHz | VDD < 2.7V, +85°C to +125°C |
| | | | 16 | 64 | MHz | 2.7V ≤ VDD, -40°C to +85°C |
| | | | 16 | 48 | MHz | 2.7V ≤ VDD, +85°C to +125°C |
| F12 | $t_{rc}$ | PLL Start-up Time (Lock Time) | — | 2 | ms | |

**TABLE 27-9: AC CHARACTERISTICS:INTERNAL OSCILLATORS ACCURACY PIC18(L)F46K22**

| **Standard Operating Conditions (unless otherwise stated)**<br>Operating temperature -40°C ≤ TA ≤ +125°C | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param. No.** | **Characteristics** | **Freq. Tolerance** | **Min** | **Typ†** | **Max** | **Units** | **Conditions** |
| OA1 | Internal Calibrated HFINTOSC Frequency(1) | ± 2% | — | 16.0 | — | MHz | 0°C ≤ TA ≤ +60°C, VDD ≥ 2.5V |
| | | ± 3% | — | 16.0 | — | MHz | +60°C ≤ TA ≤ +85°C, VDD ≥ 2.5V |
| | | ± 5% | — | 16.0 | — | MHz | -40°C ≤ TA ≤ +125°C |
| OA2 | Internal Calibrated MFINTOSC Frequency(1) | ± 2% | — | 500 | — | kHz | 0°C ≤ TA ≤ +60°C, VDD ≥ 2.5V |
| | | ± 3% | — | 500 | — | kHz | +60°C ≤ TA ≤ +85°C, VDD ≥ 2.5V |
| | | ± 5% | — | 500 | — | kHz | -40°C ≤ TA ≤ +125°C |
| OA3 | Internal Calibrated LFINTOSC Frequency(1) | ± 20% | — | 31 | — | kHz | -40°C ≤ TA ≤ +125°C |

† 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:** To ensure these oscillator frequency tolerances, VDD and VSS must be capacitively decoupled as close to the device as possible. 0.1 µF and 0.01 µF values in parallel are recommended.

**FIGURE 28-66: PIC18F2X/4XK22 TYPICAL IDD: PRI_IDLE EC HIGH POWER**

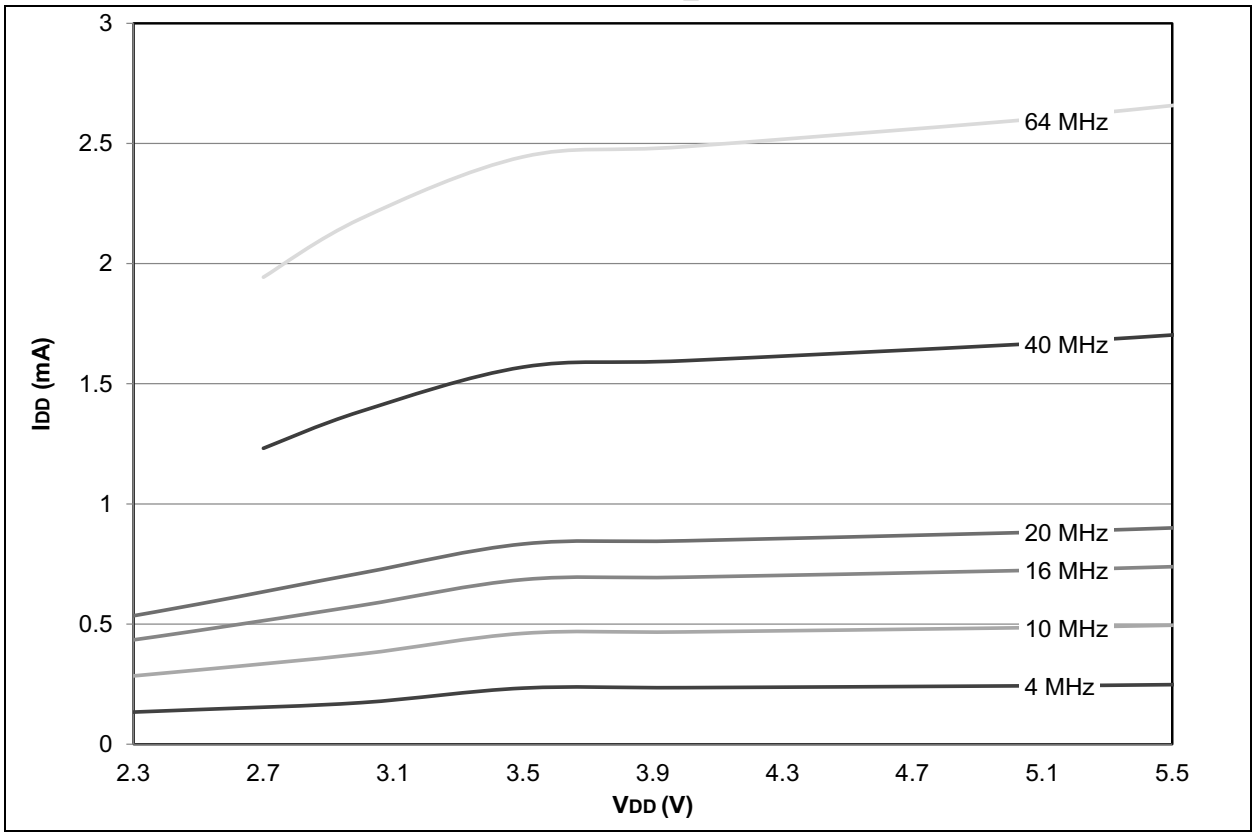



**FIGURE 28-67: PIC18F2X/4XK22 MAXIMUM IDD: PRI_IDLE EC HIGH POWER**

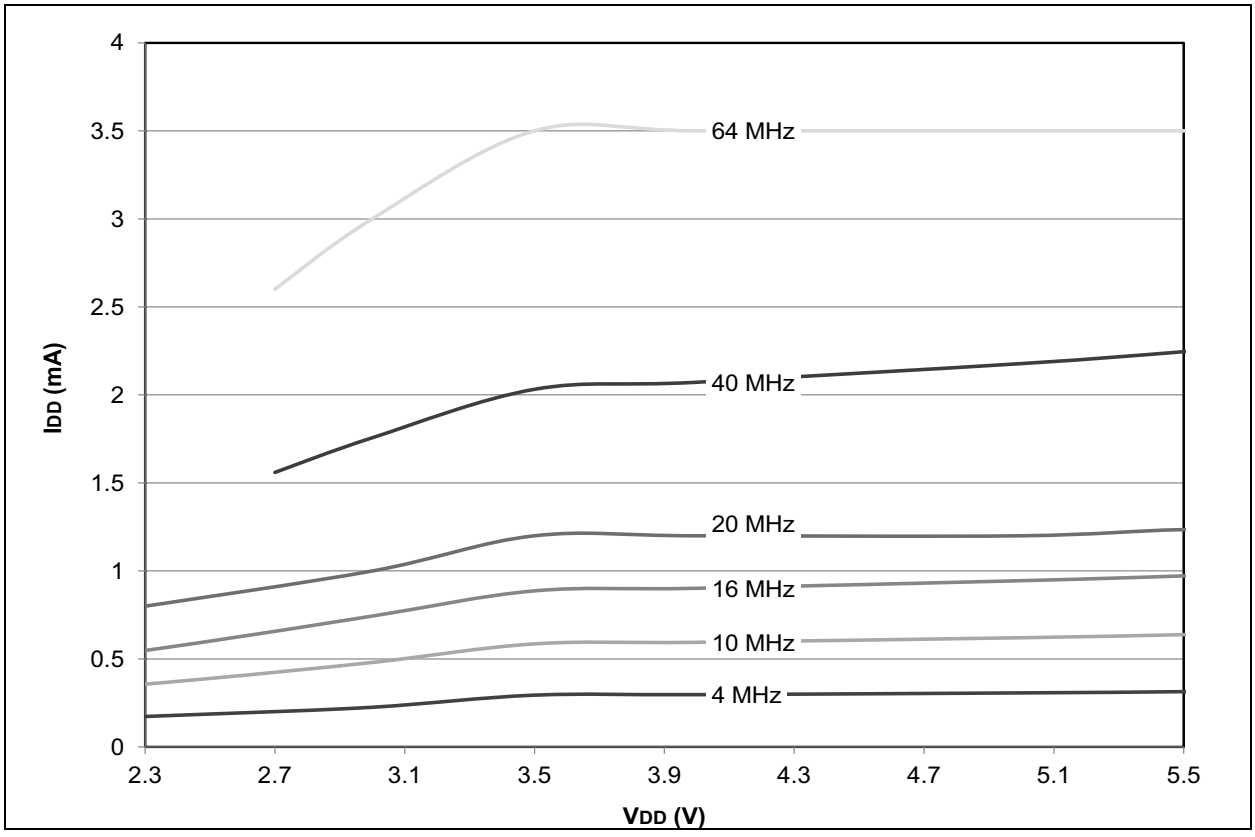


**FIGURE 28-82: PIC18(L)F2X/4XK22 TTL BUFFER INPUT HIGH VOLTAGE**

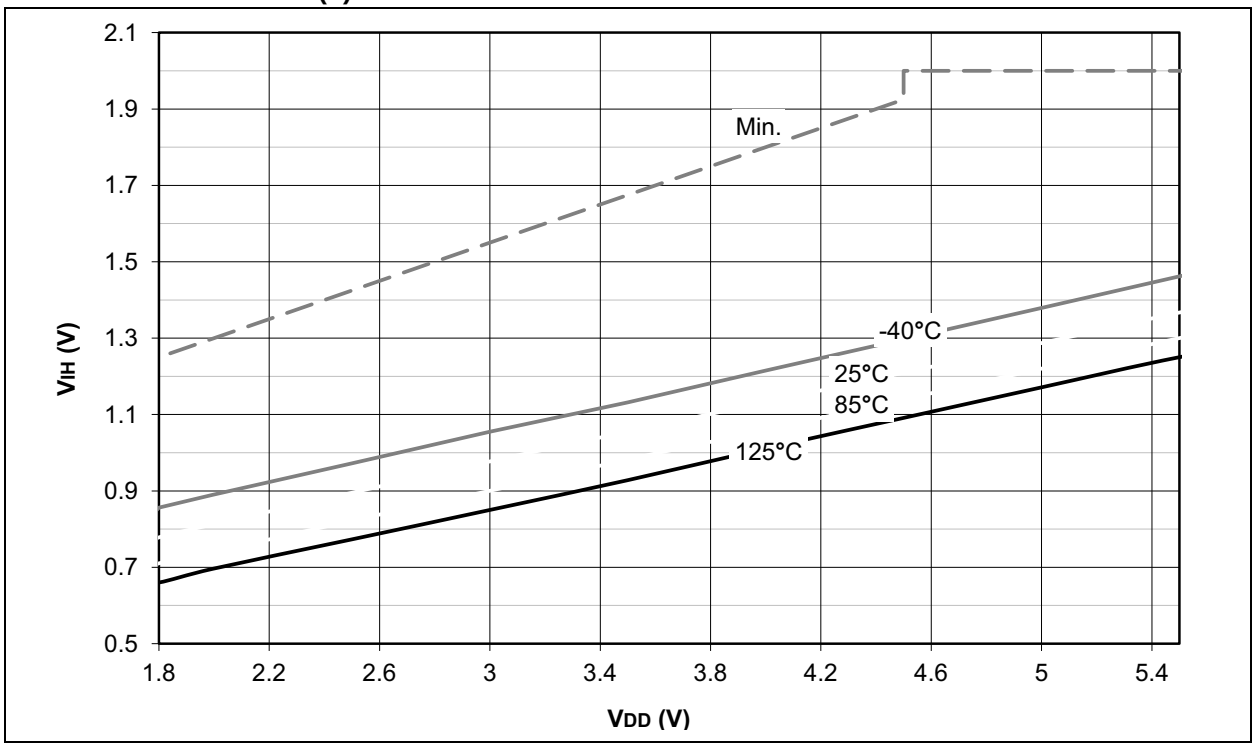



**FIGURE 28-83: PIC18(L)F2X/4XK22 SCHMITT TRIGGER BUFFER INPUT HIGH VOLTAGE**

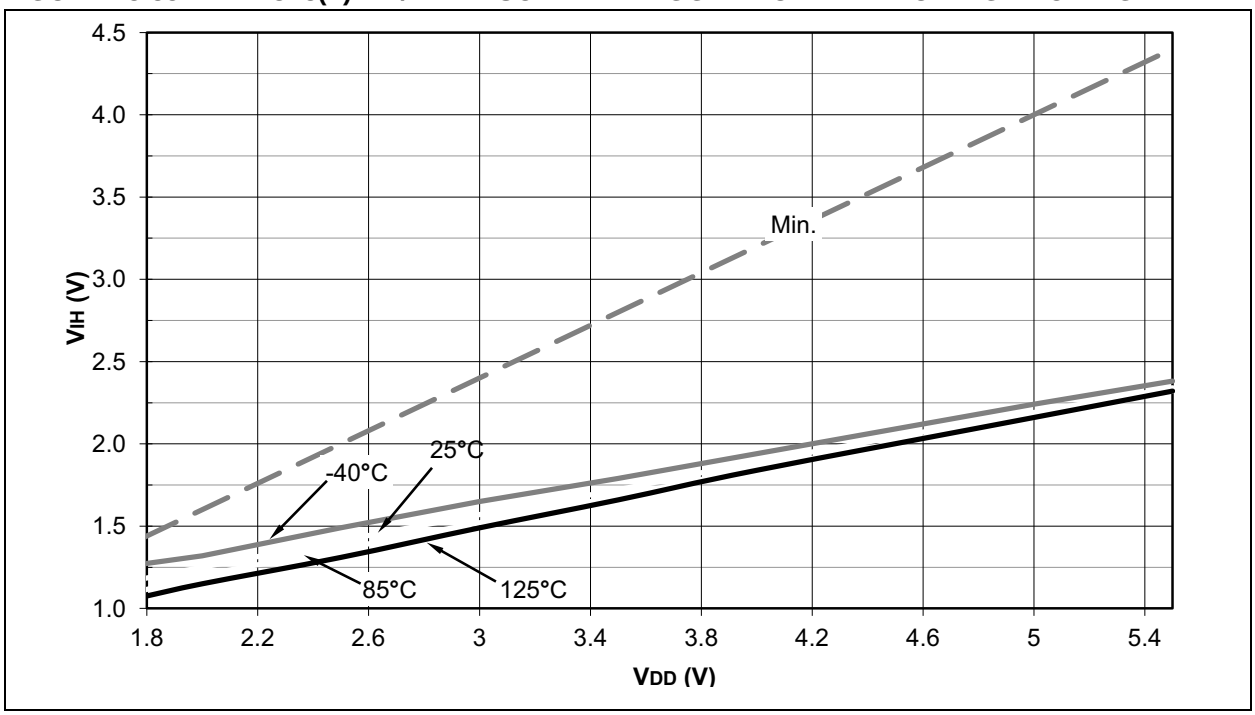
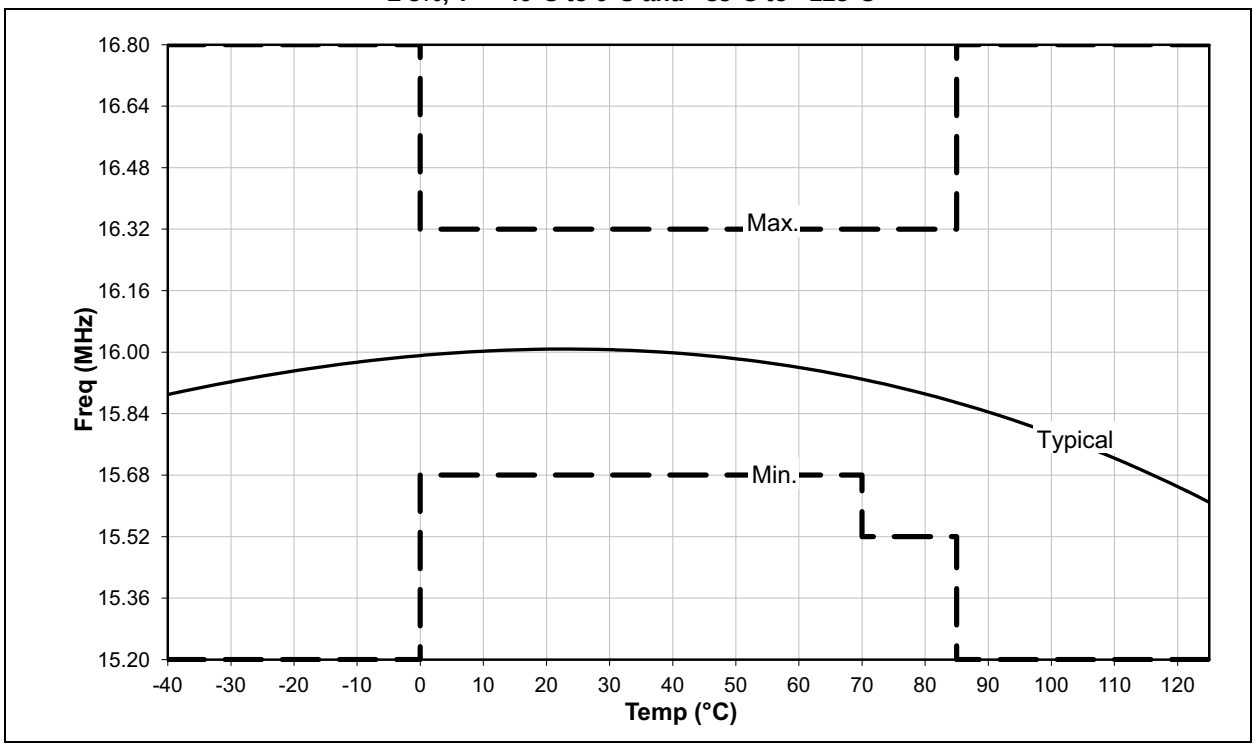


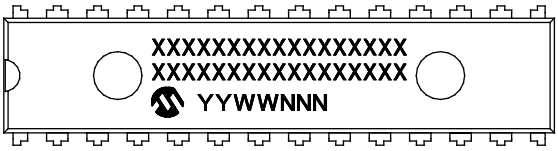
**FIGURE 28-100: PIC18(L)F2X/4XK22 HF-INTOSC FREQUENCY vs. TEMPERATURE at 16 MHz**
**MIN / MAX: ± 2%, T = 0°C to +70°C**
**+2% / -3%, T = +70°C to +85°C**
**± 5%, T = -40°C to 0°C and +85°C to +125°C**



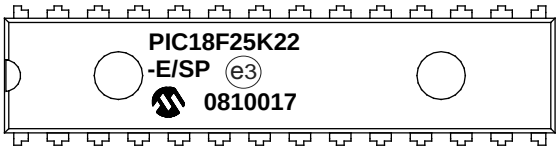
## 29.0 PACKAGING INFORMATION

### 29.1 Package Marking Information

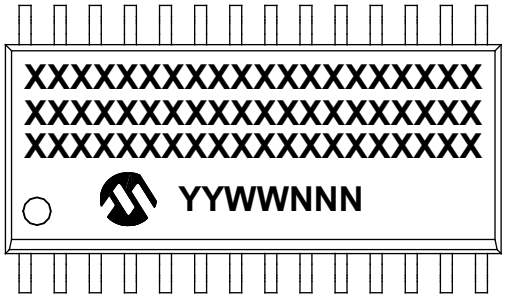
28-Lead SPDIP (.300”)




Example

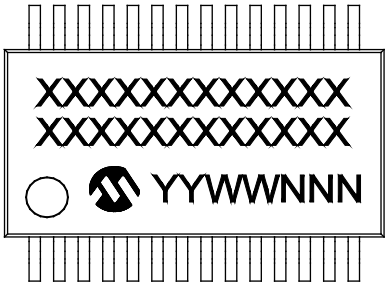



28-Lead SOIC (7.50 mm)

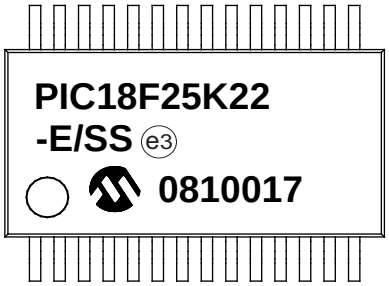



Example


28-Lead SSOP (5.30 mm)




Example




| | | |
|---|---|---|
| **Legend:** | XX...X | Customer-specific information or Microchip part number |
| | Y | Year code (last digit of calendar year) |
| | YY | Year code (last 2 digits of calendar year) |
| | WW | Week code (week of January 1 is week ‘01’) |
| | NNN | Alphanumeric traceability code |
| | (e3) | Pb-free JEDEC® designator for Matte Tin (Sn) |
| | * | This package is Pb-free. The Pb-free JEDEC designator ((e3)) can be found on the outer packaging for this package. |

**Note**: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.