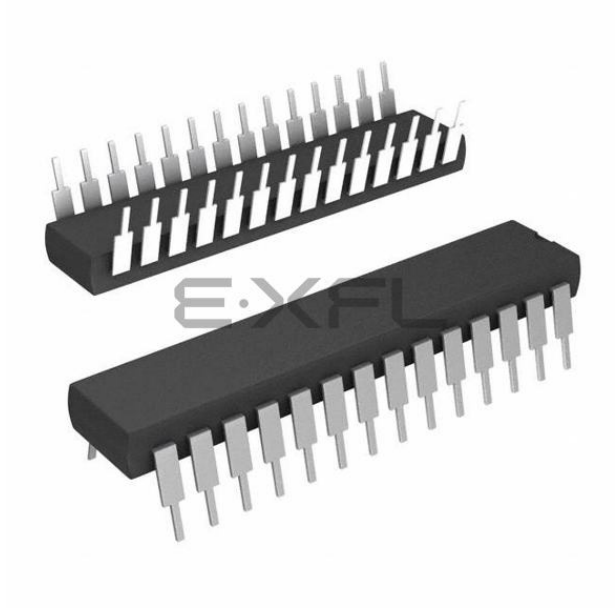


Welcome to E-XFL.COM

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	64MHz
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)	2.3V ~ 5.5V
Data Converters	A/D 19x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	28-DIP (0.300", 7.62mm)
Supplier Device Package	28-SPDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f24k22-i-sp

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

Pin Number		Pin Name	Pin Type	Buffer Type	Description
PDIP, SOIC	QFN, UQFN				
25	22	RB4/IOC0/P1D/T5G/AN11			
		RB4	I/O	TTL	Digital I/O.
		IOC0	I	TTL	Interrupt-on-change pin.
		P1D	O	CMOS	Enhanced CCP1 PWM output.
		T5G	I	ST	Timer5 external clock gate input.
		AN11	I	Analog	Analog input 11.
26	23	RB5/IOC1/P2B/P3A/CCP3/T3CKI/T1G/AN13			
		RB5	I/O	TTL	Digital I/O.
		IOC1	I	TTL	Interrupt-on-change pin.
		P2B(1)	O	CMOS	Enhanced CCP2 PWM output.
		P3A(1)	O	CMOS	Enhanced CCP3 PWM output.
		CCP3(1)	I/O	ST	Capture 3 input/Compare 3 output/PWM 3 output.
		T3CKI(2)	I	ST	Timer3 clock input.
		T1G	I	ST	Timer1 external clock gate input.
		AN13	I	Analog	Analog input 13.
27	24	RB6/IOC2/TX2/CK2/PGC			
		RB6	I/O	TTL	Digital I/O.
		IOC2	I	TTL	Interrupt-on-change pin.
		TX2	O	—	EUSART asynchronous transmit.
		CK2	I/O	ST	EUSART synchronous clock (see related RXx/DTx).
		PGC	I/O	ST	In-Circuit Debugger and ICSP™ programming clock pin.
28	25	RB7/IOC3/RX2/DT2/PGD			
		RB7	I/O	TTL	Digital I/O.
		IOC3	I	TTL	Interrupt-on-change pin.
		RX2	I	ST	EUSART asynchronous receive.
		DT2	I/O	ST	EUSART synchronous data (see related TXx/CKx).
		PGD	I/O	ST	In-Circuit Debugger and ICSP™ programming data pin.
11	8	RC0/P2B/T3CKI/T3G/T1CKI/SOSCO			
		RC0	I/O	ST	Digital I/O.
		P2B(2)	O	CMOS	Enhanced CCP1 PWM output.
		T3CKI(1)	I	ST	Timer3 clock input.
		T3G	I	ST	Timer3 external clock gate input.
		T1CKI	I	ST	Timer1 clock input.
		SOSCO	O	—	Secondary oscillator output.
12	9	RC1/P2A/CCP2/SOSCI			
		RC1	I/O	ST	Digital I/O.
		P2A	O	CMOS	Enhanced CCP2 PWM output.
		CCP2(1)	I/O	ST	Capture 2 input/Compare 2 output/PWM 2 output.
		SOSCI	I	Analog	Secondary oscillator input.

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.

TABLE 1-3: PIC18(L)F4XK22 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Number				Pin Name	Pin Type	Buffer Type	Description
PDIP	TQFP	QFN	UQFN				
37	14	14	12	RB4/IOC0/T5G/AN11			
				RB4	I/O	TTL	Digital I/O.
				IOC0	I	TTL	Interrupt-on-change pin.
				T5G	I	ST	Timer5 external clock gate input.
				AN11	I	Analog	Analog input 11.
38	15	15	13	RB5/IOC1/P3A/CCP3/T3CKI/T1G/AN13			
				RB5	I/O	TTL	Digital I/O.
				IOC1	I	TTL	Interrupt-on-change pin.
				P3A(1)	O	CMOS	Enhanced CCP3 PWM output.
				CCP3(1)	I/O	ST	Capture 3 input/Compare 3 output/PWM 3 output.
				T3CKI(2)	I	ST	Timer3 clock input.
				T1G	I	ST	Timer1 external clock gate input.
				AN13	I	Analog	Analog input 13.
39	16	16	14	RB6/IOC2/PGC			
				RB6	I/O	TTL	Digital I/O.
				IOC2	I	TTL	Interrupt-on-change pin.
				PGC	I/O	ST	In-Circuit Debugger and ICSP™ programming clock pin.
40	17	17	15	RB7/IOC3/PGD			
				RB7	I/O	TTL	Digital I/O.
				IOC3	I	TTL	Interrupt-on-change pin.
				PGD	I/O	ST	In-Circuit Debugger and ICSP™ programming data pin.
15	32	34	30	RC0/P2B/T3CKI/T3G/T1CKI/SOSCO			
				RC0	I/O	ST	Digital I/O.
				P2B(2)	O	CMOS	Enhanced CCP1 PWM output.
				T3CKI(1)	I	ST	Timer3 clock input.
				T3G	I	ST	Timer3 external clock gate input.
				T1CKI	I	ST	Timer1 clock input.
				SOSCO	O	—	Secondary oscillator output.
16	35	35	31	RC1/P2A/CCP2/SOSCI			
				RC1	I/O	ST	Digital I/O.
				P2A(1)	O	CMOS	Enhanced CCP2 PWM output.
				CCP2(1)	I/O	ST	Capture 2 input/Compare 2 output/PWM 2 output.
				SOSCI	I	Analog	Secondary oscillator input.
17	36	36	32	RC2/CTPLS/P1A/CCP1/T5CKI/AN14			
				RC2	I/O	ST	Digital I/O.
				CTPLS	O	—	CTMU pulse generator output.
				P1A	O	CMOS	Enhanced CCP1 PWM output.
				CCP1	I/O	ST	Capture 1 input/Compare 1 output/PWM 1 output.
				T5CKI	I	ST	Timer5 clock input.
				AN14	I	Analog	Analog input 14.

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/P3A and CCP2/P2A when Configuration bits PB2MX, T3CMX, CCP3MX and CCP2MX are set.

2: Alternate pin assignment for P2B, T3CKI, CCP3/P3A and CCP2/P2A when Configuration bits PB2MX, T3CMX, CCP3MX and CCP2MX are clear.

4.3 Master Clear ($\overline{MCLR}$)

The $\overline{MCLR}$ pin provides a method for triggering an external Reset of the device. A Reset is generated by holding the pin low. These devices have a noise filter in the $\overline{MCLR}$ Reset path which detects and ignores small pulses. An internal weak pull-up is enabled when the pin is configured as the $\overline{MCLR}$ input.

The $\overline{MCLR}$ pin is not driven low by any internal Resets, including the WDT.

In PIC18(L)F2X/4XK22 devices, the $\overline{MCLR}$ input can be disabled with the MCLRE Configuration bit. When $\overline{MCLR}$ is disabled, the pin becomes a digital input. See **Section 10.6 "PORTE Registers"** for more information.

4.4 Power-on Reset (POR)

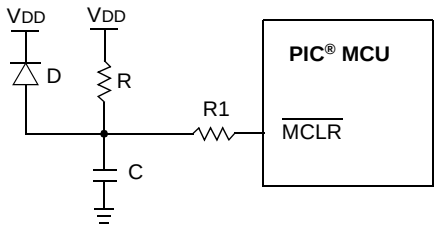
A Power-on Reset pulse is generated on-chip whenever VDD rises above a certain threshold. This allows the device to start in the initialized state when VDD is adequate for operation.

To take advantage of the POR circuitry either leave the pin floating, or tie the $\overline{MCLR}$ pin through a resistor to VDD. This will eliminate external RC components usually needed to create a Power-on Reset delay. A minimum rise rate for VDD is specified. For a slow rise time, see Figure 4-2.

When the device starts normal operation (i.e., exits the Reset condition), device operating parameters (voltage, frequency, temperature, etc.) must be met to ensure proper operation. If these conditions are not met, the device must be held in Reset until the operating conditions are met.

POR events are captured by the $\overline{POR}$ bit of the RCON register. The state of the bit is set to '0' whenever a POR occurs; it does not change for any other Reset event. $\overline{POR}$ is not reset to '1' by any hardware event. To capture multiple events, the user must manually set the bit to '1' by software following any POR.

FIGURE 4-2: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW VDD POWER-UP)



Note 1: External Power-on Reset circuit is required only if the VDD power-up slope is too slow. The diode D helps discharge the capacitor quickly when VDD powers down.

2: 15 kΩ < R < 40 kΩ is recommended to make sure that the voltage drop across R does not violate the device's electrical specification.

3: R1 ≥ 1 kΩ will limit any current flowing into $\overline{MCLR}$ from external capacitor C, in the event of $\overline{MCLR}$/VPP pin breakdown, due to Electrostatic Discharge (ESD) or Electrical Overstress (EOS).

Operations on the FSRs with POSTDEC, POSTINC and PREINC affect the entire register pair; that is, rollovers of the FSRnL register from FFh to 00h carry over to the FSRnH register. On the other hand, results of these operations do not change the value of any flags in the STATUS register (e.g., Z, N, OV, etc.).

The PLUSW register can be used to implement a form of indexed addressing in the data memory space. By manipulating the value in the W register, users can reach addresses that are fixed offsets from pointer addresses. In some applications, this can be used to implement some powerful program control structure, such as software stacks, inside of data memory.

5.6.3.3 Operations by FSRs on FSRs

Indirect addressing operations that target other FSRs or virtual registers represent special cases. For example, using an FSR to point to one of the virtual registers will not result in successful operations. As a specific case, assume that FSR0H:FSR0L contains FE7h, the address of INDF1. Attempts to read the value of the INDF1 using INDF0 as an operand will return 00h. Attempts to write to INDF1 using INDF0 as the operand will result in a NOP.

On the other hand, using the virtual registers to write to an FSR pair may not occur as planned. In these cases, the value will be written to the FSR pair but without any incrementing or decrementing. Thus, writing to either the INDF2 or POSTDEC2 register will write the same value to the FSR2H:FSR2L.

Since the FSRs are physical registers mapped in the SFR space, they can be manipulated through all direct operations. Users should proceed cautiously when working on these registers, particularly if their code uses indirect addressing.

Similarly, operations by indirect addressing are generally permitted on all other SFRs. Users should exercise the appropriate caution that they do not inadvertently change settings that might affect the operation of the device.

5.7 Data Memory and the Extended Instruction Set

Enabling the PIC18 extended instruction set (XINST Configuration bit = 1) significantly changes certain aspects of data memory and its addressing. Specifically, the use of the Access Bank for many of the core PIC18 instructions is different; this is due to the introduction of a new addressing mode for the data memory space.

What does not change is just as important. The size of the data memory space is unchanged, as well as its linear addressing. The SFR map remains the same. Core PIC18 instructions can still operate in both Direct and Indirect Addressing mode; inherent and literal instructions do not change at all. Indirect addressing with FSR0 and FSR1 also remain unchanged.

5.7.1 INDEXED ADDRESSING WITH LITERAL OFFSET

Enabling the PIC18 extended instruction set changes the behavior of indirect addressing using the FSR2 register pair within Access RAM. Under the proper conditions, instructions that use the Access Bank – that is, most bit-oriented and byte-oriented instructions – can invoke a form of indexed addressing using an offset specified in the instruction. This special addressing mode is known as Indexed Addressing with Literal Offset, or Indexed Literal Offset mode.

When using the extended instruction set, this addressing mode requires the following:

- The use of the Access Bank is forced (‘a’ = 0) and
- The file address argument is less than or equal to 5Fh.

Under these conditions, the file address of the instruction is not interpreted as the lower byte of an address (used with the BSR in direct addressing), or as an 8-bit address in the Access Bank. Instead, the value is interpreted as an offset value to an Address Pointer, specified by FSR2. The offset and the contents of FSR2 are added to obtain the target address of the operation.

5.7.2 INSTRUCTIONS AFFECTED BY INDEXED LITERAL OFFSET MODE

Any of the core PIC18 instructions that can use direct addressing are potentially affected by the Indexed Literal Offset Addressing mode. This includes all byte-oriented and bit-oriented instructions, or almost one-half of the standard PIC18 instruction set. Instructions that only use Inherent or Literal Addressing modes are unaffected.

Additionally, byte-oriented and bit-oriented instructions are not affected if they do not use the Access Bank (Access RAM bit is ‘1’), or include a file address of 60h or above. Instructions meeting these criteria will continue to execute as before. A comparison of the different possible addressing modes when the extended instruction set is enabled is shown in Figure 5-11.

Those who desire to use byte-oriented or bit-oriented instructions in the Indexed Literal Offset mode should note the changes to assembler syntax for this mode. This is described in more detail in **Section 25.2.1 “Extended Instruction Syntax”**.

9.8 Register Definitions: Interrupt Control

REGISTER 9-1: INTCON: INTERRUPT CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
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 **GIE/GIEH:** Global Interrupt Enable bit
When IPEN = 0:
1 = Enables all unmasked interrupts
0 = Disables all interrupts including peripherals
When IPEN = 1:
1 = Enables all high priority interrupts
0 = Disables all interrupts including low priority

bit 6 **PEIE/GIEL:** Peripheral Interrupt Enable bit
When IPEN = 0:
1 = Enables all unmasked peripheral interrupts
0 = Disables all peripheral interrupts
When IPEN = 1:
1 = Enables all low priority interrupts
0 = Disables all low priority interrupts

bit 5 **TMR0IE:** TMR0 Overflow Interrupt Enable bit
1 = Enables the TMR0 overflow interrupt
0 = Disables the TMR0 overflow interrupt

bit 4 **INT0IE:** INT0 External Interrupt Enable bit
1 = Enables the INT0 external interrupt
0 = Disables the INT0 external interrupt

bit 3 **RBIE:** Port B Interrupt-On-Change (IOCx) Interrupt Enable bit(2)
1 = Enables the IOCx port change interrupt
0 = Disables the IOCx port change interrupt

bit 2 **TMR0IF:** TMR0 Overflow Interrupt Flag bit
1 = TMR0 register has overflowed (must be cleared by software)
0 = TMR0 register did not overflow

bit 1 **INT0IF:** INT0 External Interrupt Flag bit
1 = The INT0 external interrupt occurred (must be cleared by software)
0 = The INT0 external interrupt did not occur

bit 0 **RBIF:** Port B Interrupt-On-Change (IOCx) Interrupt Flag bit(1)
1 = At least one of the IOC<3:0> (RB<7:4>) pins changed state (must be cleared by software)
0 = None of the IOC<3:0> (RB<7:4>) pins have changed state

Note 1: A mismatch condition will continue to set the RBIF bit. Reading PORTB will end the mismatch condition and allow the bit to be cleared.
2: RB port change interrupts also require the individual pin IOCB enables.

Note: Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. This feature allows for software polling.

10.4 PORTC Registers

PORTC is an 8-bit wide, bidirectional port. The corresponding data direction register is TRISC. Setting a TRISC bit (= 1) will make the corresponding PORTC pin an input (i.e., disable the output driver). Clearing a TRISC bit (= 0) will make the corresponding PORTC pin an output (i.e., enable the output driver and put the contents of the output latch on the selected pin).

The Data Latch register (LATC) is also memory mapped. Read-modify-write operations on the LATC register read and write the latched output value for PORTC.

PORTC is multiplexed with several peripheral functions (Table 10-8). The pins have Schmitt Trigger input buffers.

Some of these pin functions can be relocated to alternate pins using the Control fuse bits in CONFIG3H. RC0 is the default pin for T3CKI. Clearing the T3CMX bit moves the pin function to RB5. RC1 is the default pin for the CCP2 peripheral pin. Clearing the CCP2MX bit moves the pin function to the RB3 pin.

Two other pin functions, P2B and CCP3, can be relocated from their default pins to PORTC pins by clearing the control fuses in CONFIG3H. Clearing P2BMX and CCP3MX moves the pin functions to RC0 and RC6**(1)**/RE0**(2)**, respectively.

When enabling peripheral functions, care should be taken in defining TRIS bits for each PORTC pin. The EUSART and MSSP peripherals override the TRIS bit to make a pin an output or an input, depending on the peripheral configuration. Refer to the corresponding peripheral section for additional information.

Note:	On a Power-on Reset, these pins are configured as analog inputs.

The contents of the TRISC register are affected by peripheral overrides. Reading TRISC always returns the current contents, even though a peripheral device may be overriding one or more of the pins.

EXAMPLE 10-3: INITIALIZING PORTC

```
MOVLB   0xF     ; Set BSR for banked SFRs
CLRF    PORTC   ; Initialize PORTC by
                ; clearing output
                ; data latches
CLRF    LATC    ; Alternate method
                ; to clear output
                ; data latches
MOVLW   0CFh    ; Value used to
                ; initialize data
                ; direction
MOVWF   TRISC   ; Set RC<3:0> as inputs
                ; RC<5:4> as outputs
                ; RC<7:6> as inputs
MOVLW   30h     ; Value used to
                ; enable digital inputs
MOVWF   ANSELC  ; RC<3:2> dig input enable
                ; No ANSEL bits for RC<1:0>
                ; RC<7:6> dig input enable
```

10.4.1 PORTC OUTPUT PRIORITY

Each PORTC pin is multiplexed with other functions. The pins, their combined functions and their output priorities are briefly described here. For additional information, refer to the appropriate section in this data sheet.

When multiple outputs are enabled, the actual pin control goes to the peripheral with the higher priority. Table 10-4 lists the PORTC pin functions from the highest to the lowest priority.

Analog input functions, such as ADC, comparator and SR latch inputs, are not shown in the priority lists.

These inputs are active when the I/O pin is set for Analog mode using the ANSELx registers. Digital output functions may control the pin when it is in Analog mode with the priority shown below.

10.5 PORTD Registers

Note: PORTD is only available on 40-pin and 44-pin devices.

PORTD is an 8-bit wide, bidirectional port. The corresponding data direction register is TRISD. Setting a TRISD bit (= 1) will make the corresponding PORTD pin an input (i.e., disable the output driver). Clearing a TRISD bit (= 0) will make the corresponding PORTD pin an output (i.e., enable the output driver and put the contents of the output latch on the selected pin).

The Data Latch register (LATD) is also memory mapped. Read-modify-write operations on the LATD register read and write the latched output value for PORTD.

All pins on PORTD are implemented with Schmitt Trigger input buffers. Each pin is individually configurable as an input or output.

All of the PORTD pins are multiplexed with analog and digital peripheral modules. See Table 10-11.

Note: On a Power-on Reset, these pins are configured as analog inputs.

EXAMPLE 10-4: INITIALIZING PORTD

```
MOVLB   0xF     ; Set BSR for banked SFRs
CLRF    PORTD   ; Initialize PORTD by
                ; clearing output
                ; data latches
CLRF    LATD    ; Alternate method
                ; to clear output
                ; data latches
MOVLW   0CFh    ; Value used to
                ; initialize data
                ; direction
MOVWF   TRISD   ; Set RD<3:0> as inputs
                ; RD<5:4> as outputs
                ; RD<7:6> as inputs
MOVLW   30h     ; Value used to
                ; enable digital inputs
MOVWF   ANSELD  ; RD<3:0> dig input enable
                ; RC<7:6> dig input enable
```

10.5.1 PORTD OUTPUT PRIORITY

Each PORTD pin is multiplexed with other functions. The pins, their combined functions and their output priorities are briefly described here. For additional information, refer to the appropriate section in this data sheet.

When multiple outputs are enabled, the actual pin control goes to the peripheral with the higher priority. Table 10-4 lists the PORTD pin functions from the highest to the lowest priority.

Analog input functions, such as ADC, comparator and SR latch inputs, are not shown in the priority lists.

These inputs are active when the I/O pin is set for Analog mode using the ANSELx registers. Digital output functions may control the pin when it is in Analog mode with the priority shown below.

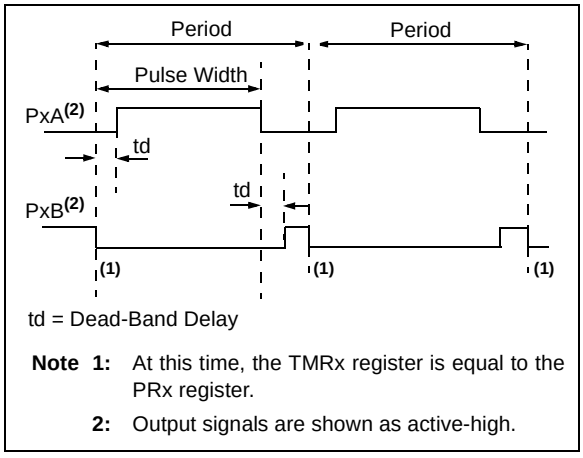
14.4.1 HALF-BRIDGE MODE

In Half-Bridge mode, two pins are used as outputs to drive push-pull loads. The PWM output signal is output on the CCPx/PxA pin, while the complementary PWM output signal is output on the PxB pin (see Figure 14-9). This mode can be used for half-bridge applications, as shown in Figure 14-9, or for full-bridge applications, where four power switches are being modulated with two PWM signals.

In Half-Bridge mode, the programmable dead-band delay can be used to prevent shoot-through current in half-bridge power devices. The value of the PDC<6:0> bits of the PWMxCON register sets the number of instruction cycles before the output is driven active. If the value is greater than the duty cycle, the corresponding output remains inactive during the entire cycle. See **Section 14.4.5 "Programmable Dead-Band Delay Mode"** for more details of the dead-band delay operations.

Since the PxA and PxB outputs are multiplexed with the PORT data latches, the associated TRIS bits must be cleared to configure PxA and PxB as outputs.

FIGURE 14-8: EXAMPLE OF HALF-BRIDGE PWM OUTPUT

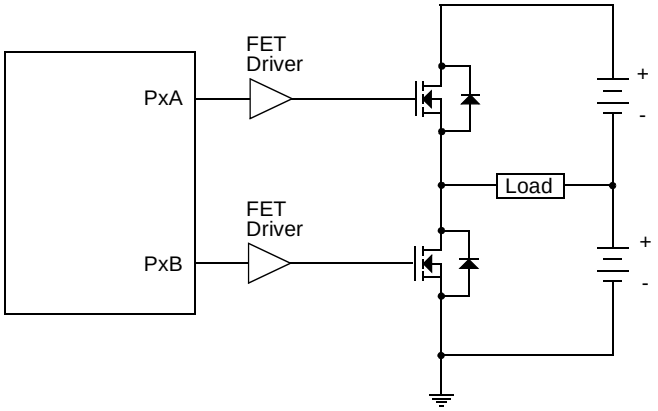


Note 1: At this time, the TMRx register is equal to the PRx register.

2: Output signals are shown as active-high.

FIGURE 14-9: EXAMPLE OF HALF-BRIDGE APPLICATIONS

Standard Half-Bridge Circuit ("Push-Pull")



Half-Bridge Output Driving a Full-Bridge Circuit

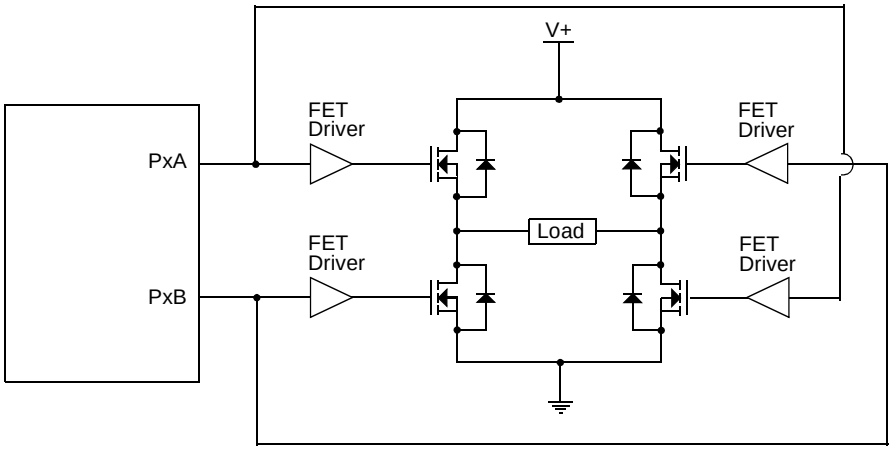
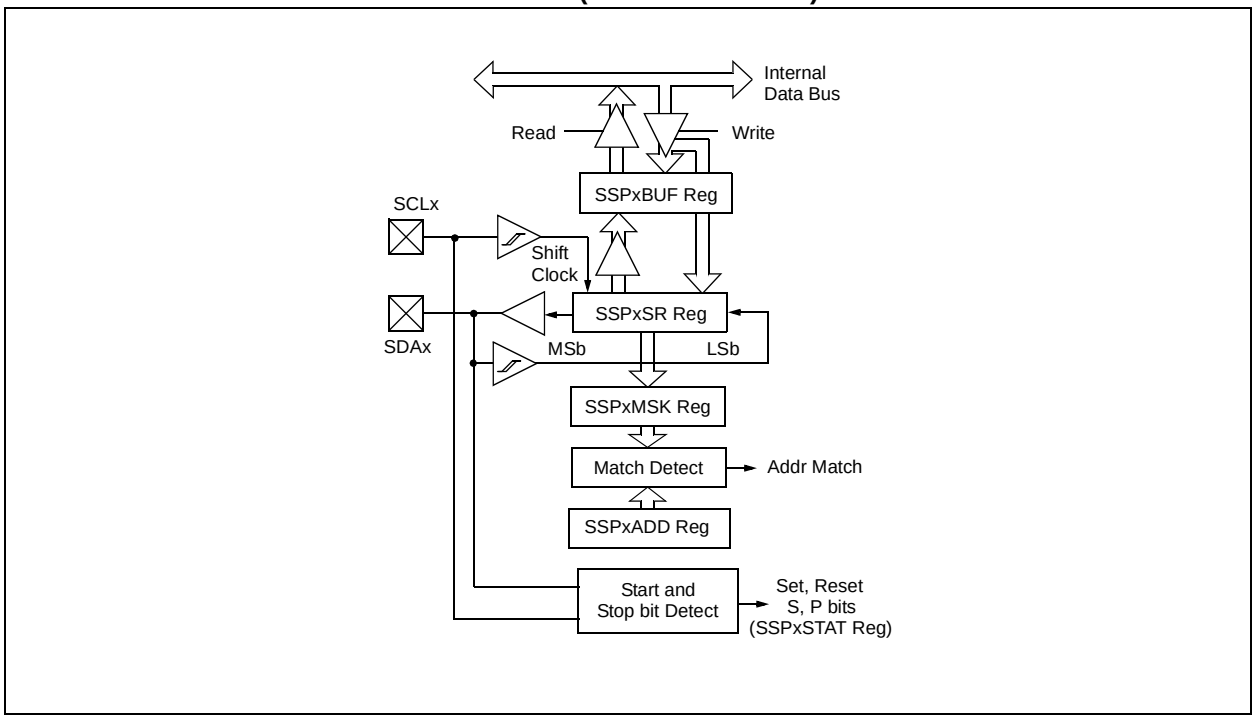


FIGURE 15-3: MSSPx BLOCK DIAGRAM (I^2C SLAVE MODE)



15.6.8 ACKNOWLEDGE SEQUENCE TIMING

An Acknowledge sequence is enabled by setting the Acknowledge Sequence Enable bit, ACKEN, of the SSPxCON2 register. When this bit is set, the SCLx pin is pulled low and the contents of the Acknowledge data bit are presented on the SDAx pin. If the user wishes to generate an Acknowledge, then the ACKDT bit should be cleared. If not, the user should set the ACKDT bit before starting an Acknowledge sequence. The Baud Rate Generator then counts for one rollover period (TBRG) and the SCLx pin is deasserted (pulled high). When the SCLx pin is sampled high (clock arbitration), the Baud Rate Generator counts for TBRG. The SCLx pin is then pulled low. Following this, the ACKEN bit is automatically cleared, the Baud Rate Generator is turned off and the MSSPx module then goes into Idle mode (Figure 15-30).

15.6.8.1 WCOL Status Flag

If the user writes the SSPxBUF when an Acknowledge sequence is in progress, then WCOL is set and the contents of the buffer are unchanged (the write does not occur).

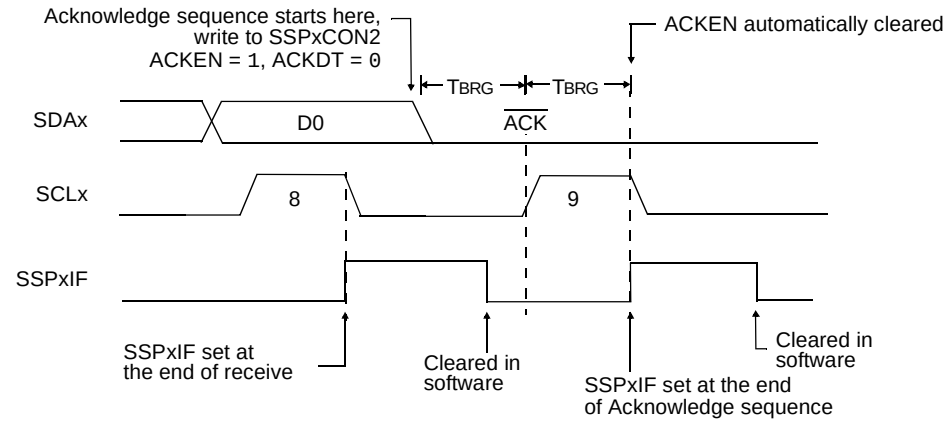
15.6.9 STOP CONDITION TIMING

A Stop bit is asserted on the SDAx pin at the end of a receive/transmit by setting the Stop Sequence Enable bit, PEN, of the SSPxCON2 register. At the end of a receive/transmit, the SCLx line is held low after the falling edge of the ninth clock. When the PEN bit is set, the master will assert the SDAx line low. When the SDAx line is sampled low, the Baud Rate Generator is reloaded and counts down to '0'. When the Baud Rate Generator times out, the SCLx pin will be brought high and one TBRG (Baud Rate Generator rollover count) later, the SDAx pin will be deasserted. When the SDAx pin is sampled high while SCLx is high, the P bit of the SSPxSTAT register is set. A TBRG later, the PEN bit is cleared and the SSPxIF bit is set (Figure 15-31).

15.6.9.1 WCOL Status Flag

If the user writes the SSPxBUF when a Stop sequence is in progress, then the WCOL bit is set and the contents of the buffer are unchanged (the write does not occur).

FIGURE 15-30: ACKNOWLEDGE SEQUENCE WAVEFORM



Note: TBRG = one Baud Rate Generator period.

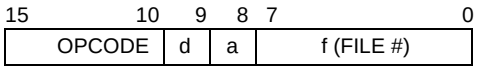
EXAMPLE 19-1: SETUP FOR CTMU CALIBRATION ROUTINES

```
#include "p18cxxx.h"
/*****************************************************************************/
/*Set up CTMU ****************************************************************/
/*****************************************************************************/
void setup(void)

{ //CTMUCONH/1 - CTMU Control registers

    CTMUCONH = 0x00;    //make sure CTMU is disabled
    CTMUCONL = 0x90;
    //CTMU continues to run when emulator is stopped,CTMU continues
    //to run in idle mode,Time Generation mode disabled, Edges are blocked
    //No edge sequence order, Analog current source not grounded, trigger
    //output disabled, Edge2 polarity = positive level, Edge2 source =
    //source 0, Edge1 polarity = positive level, Edge1 source = source 0,


    //CTMUICON - CTMU Current Control Register
    CTMUICON = 0x01;    //0.55uA, Nominal - No Adjustment

/*****************************************************************************/
//Set up AD converter;
/*****************************************************************************/

    TRISA=0x04;                 //set channel 2 as an input

    // Configure AN2 as an analog channel
    ANSELAbits.ANSA2=1;
    TRISAbits.TRISA2=1;

   // ADCON2
   ADCON2bits.ADFM=1;           // Results format 1= Right justified
   ADCON2bits.ACQT=1;           // Acquition time 7 = 20TAD 2 = 4TAD 1=2TAD
   ADCON2bits.ADCS=2;           // Clock conversion bits 6= FOSC/64 2=FOSC/32



   // ADCON1
   ADCON1bits.PVCFG0 =0;         // Vref+ = AVdd
   ADCON1bits.NVCFG1 =0;         // Vref- = AVss
  //  ADCON0
   ADCON0bits.CHS=2;            // Select ADC channel

   ADCON0bits.ADON=1;           // Turn on ADC

}
```

FIGURE 25-1: GENERAL FORMAT FOR INSTRUCTIONS

Byte-oriented file register operations

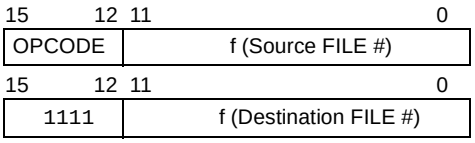
Example Instruction



ADDWF MYREG, W, B

d = 0 for result destination to be WREG register
d = 1 for result destination to be file register (f)
a = 0 to force Access Bank
a = 1 for BSR to select bank
f = 8-bit file register address

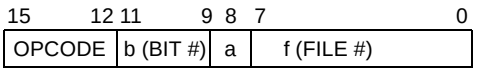
Byte to Byte move operations (2-word)



MOVFF MYREG1, MYREG2

f = 12-bit file register address

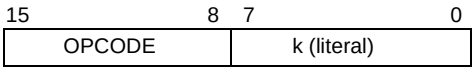
Bit-oriented file register operations



BSF MYREG, bit, B

b = 3-bit position of bit in file register (f)
a = 0 to force Access Bank
a = 1 for BSR to select bank
f = 8-bit file register address

Literal operations

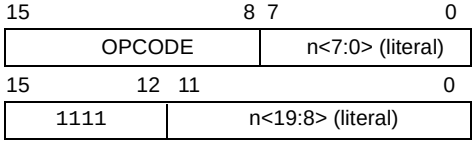


MOVLW 7Fh

k = 8-bit immediate value

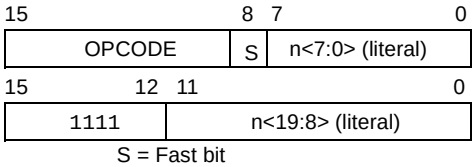
Control operations

CALL, GOTO and Branch operations



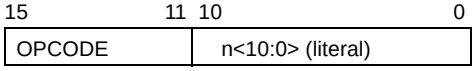
GOTO Label

n = 20-bit immediate value

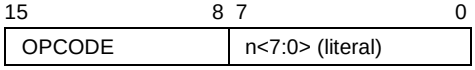


CALL MYFUNC

S = Fast bit



BRA MYFUNC



BC MYFUNC

MOVSS **Move Indexed to Indexed**

Syntax: MOVSS [z_s], [z_d]

Operands: $0 \le z_s \le 127$
$0 \le z_d \le 127$

Operation: ((FSR2) + z_s) → ((FSR2) + z_d)

Status Affected: None

Encoding:

1st word (source)	1110	1011	1zzz	$zzzz_s$
2nd word (dest.)	1111	xxxx	xzzz	$zzzz_d$

Description: The contents of the source register are moved to the destination register. The addresses of the source and destination registers are determined by adding the 7-bit literal offsets 'z_s' or 'z_d', respectively, to the value of FSR2. Both registers can be located anywhere in the 4096-byte data memory space (000h to FFFh).
The MOVSS instruction cannot use the PCL, TOSU, TOSH or TOSL as the destination register.
If the resultant source address points to an indirect addressing register, the value returned will be 00h. If the resultant destination address points to an indirect addressing register, the instruction will execute as a NOP.

Words: 2

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Determine source addr	Determine source addr	Read source reg
Decode	Determine dest addr	Determine dest addr	Write to dest reg

Example: MOVSS [05h], [06h]

Before Instruction
FSR2 = 80h
Contents of 85h = 33h
Contents of 86h = 11h
After Instruction
FSR2 = 80h
Contents of 85h = 33h
Contents of 86h = 33h

PUSHL **Store Literal at FSR2, Decrement FSR2**

Syntax: PUSHL k

Operands: $0 \le k \le 255$

Operation: k → (FSR2),
FSR2 – 1 → FSR2

Status Affected: None

Encoding:

1111	1010	kkkk	kkkk

Description: The 8-bit literal 'k' is written to the data memory address specified by FSR2. FSR2 is decremented by 1 after the operation. This instruction allows users to push values onto a software stack.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read 'k'	Process data	Write to destination

Example: PUSHL 08h

Before Instruction
FSR2H:FSR2L = 01ECh
Memory (01ECh) = 00h

After Instruction
FSR2H:FSR2L = 01EBh
Memory (01ECh) = 08h

26.11 Demonstration/Development Boards, Evaluation Kits, and Starter Kits

A wide variety of demonstration, development and evaluation boards for various PIC MCUs and dsPIC DSCs allows quick application development on fully functional systems. Most boards include prototyping areas for adding custom circuitry and provide application firmware and source code for examination and modification.

The boards support a variety of features, including LEDs, temperature sensors, switches, speakers, RS-232 interfaces, LCD displays, potentiometers and additional EEPROM memory.

The demonstration and development boards can be used in teaching environments, for prototyping custom circuits and for learning about various microcontroller applications.

In addition to the PICDEM™ and dsPICDEM™ demonstration/development board series of circuits, Microchip has a line of evaluation kits and demonstration software for analog filter design, KEELOQ® security ICs, CAN, IrDA®, PowerSmart battery management, SEEVAL® evaluation system, Sigma-Delta ADC, flow rate sensing, plus many more.

Also available are starter kits that contain everything needed to experience the specified device. This usually includes a single application and debug capability, all on one board.

Check the Microchip web page (www.microchip.com) for the complete list of demonstration, development and evaluation kits.

26.12 Third-Party Development Tools

Microchip also offers a great collection of tools from third-party vendors. These tools are carefully selected to offer good value and unique functionality.

- Device Programmers and Gang Programmers from companies, such as SoftLog and CCS
- Software Tools from companies, such as Gimpel and Trace Systems
- Protocol Analyzers from companies, such as Saleae and Total Phase
- Demonstration Boards from companies, such as MikroElektronika, Digilent® and Olimex
- Embedded Ethernet Solutions from companies, such as EZ Web Lynx, WIZnet and IPLogika®

27.6 DC Characteristics: Primary Idle Supply Current, PIC18(L)F2X/4XK22

PIC18LF2X/4XK22		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +125°C					
PIC18F2X/4XK22		**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +125°C					
Param No.	**Device Characteristics**	**Typ**	**Max**	**Units**	**Conditions**		
D100	Supply Current (IDD)[1],[2]	0.030	0.050	mA	-40°C to +125°C	VDD = 1.8V	Fosc = 1 MHz (**PRI_IDLE** mode, ECM source)
D101		0.045	0.065	mA	-40°C to +125°C	VDD = 3.0V	
D102		0.06	0.12	mA	-40°C to +125°C	VDD = 2.3V	Fosc = 1 MHz (**PRI_IDLE** mode, ECM source)
D103		0.08	0.15	mA	-40°C to +125°C	VDD = 3.0V	
D104		0.13	0.20	mA	-40°C to +125°C	VDD = 5.0V	
D105		0.45	0.8	mA	-40°C to +125°C	VDD = 1.8V	Fosc = 20 MHz (**PRI_IDLE** mode, ECH source)
D106		0.70	1.0	mA	-40°C to +125°C	VDD = 3.0V	
D107		0.55	0.8	mA	-40°C to +125°C	VDD = 2.3V	Fosc = 20 MHz (**PRI_IDLE** mode, ECH source)
D108		0.75	1.0	mA	-40°C to +125°C	VDD = 3.0V	
D109		0.90	1.2	mA	-40°C to +125°C	VDD = 5.0V	
D110		2.25	3.0	mA	-40°C to +125°C	VDD = 3.0V	Fosc = 64 MHz (**PRI_IDLE** mode, ECH source)
D111		2.25	3.0	mA	-40°C to +125°C	VDD = 3.0V	Fosc = 64 MHz (**PRI_IDLE** mode, ECH source)
D112		2.60	3.5	mA	-40°C to +125°C	VDD = 5.0V	
D113		0.35	0.6	mA	-40°C to +125°C	VDD = 1.8V	Fosc = 4 MHz 16 MHz Internal (**PRI_IDLE** mode, ECM + PLL source)
D114		0.55	0.8	mA	-40°C to +125°C	VDD = 3.0V	
D115		0.45	0.6	mA	-40°C to +125°C	VDD = 2.3V	Fosc = 4 MHz 16 MHz Internal (**PRI_IDLE** mode, ECM + PLL source)
D116		0.60	0.9	mA	-40°C to +125°C	VDD = 3.0V	
D117		0.70	1.0	mA	-40°C to +125°C	VDD = 5.0V	
D118		2.2	3.0	mA	-40°C to +125°C	VDD = 3.0V	Fosc = 16 MHz 64 MHz Internal (**PRI_IDLE** mode, ECH + PLL source)
D119		2.2	3.0	mA	-40°C to +125°C	VDD = 3.0V	Fosc = 16 MHz 64 MHz Internal (**PRI_IDLE** mode, ECH + PLL source)
D120		2.5	3.5	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;
MCLR = VDD;
OSC1 = external square wave, from rail-to-rail (PRI_RUN and PRI_IDLE only).

FIGURE 28-60: PIC18LF2X/4XK22 TYPICAL IDD: PRI_IDLE EC MEDIUM POWER

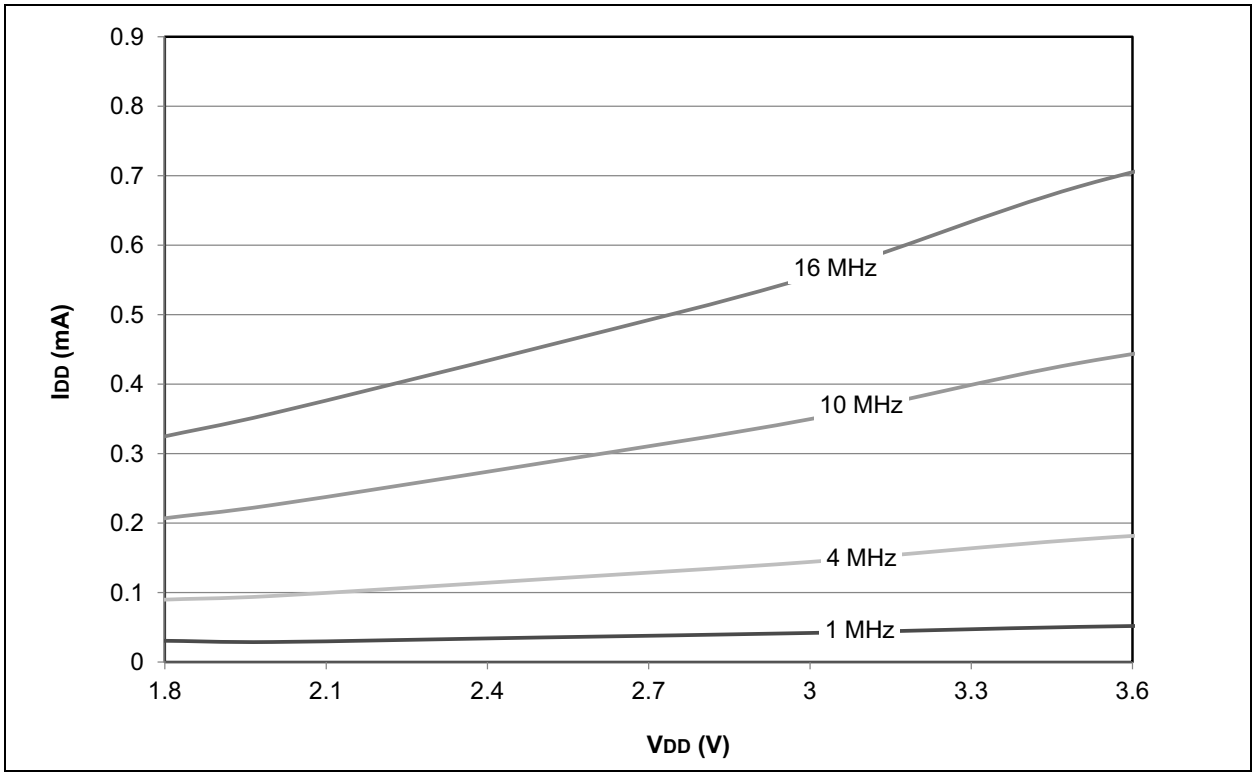


FIGURE 28-61: PIC18LF2X/4XK22 MAXIMUM IDD: PRI_IDLE EC MEDIUM POWER

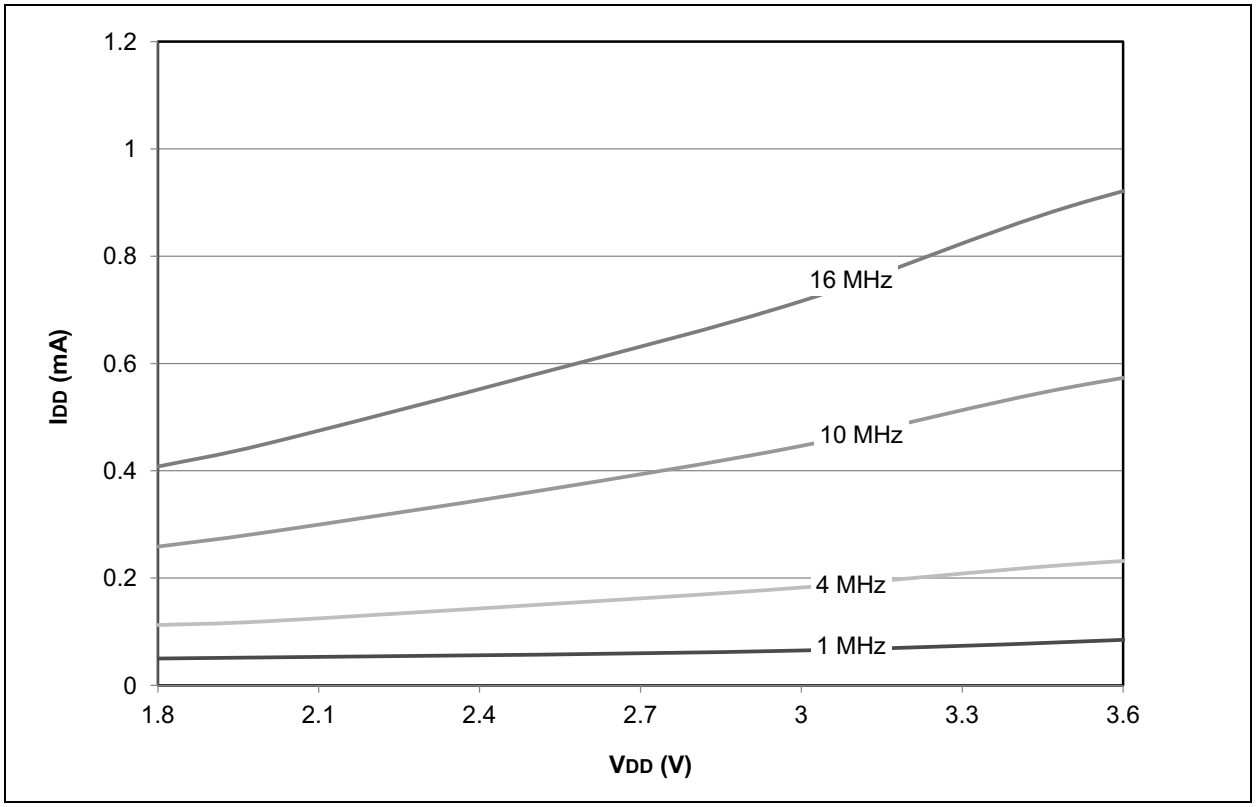


FIGURE 28-72: PIC18LF2X/4XK22 TYPICAL IDD: SEC_RUN 32.768 kHz

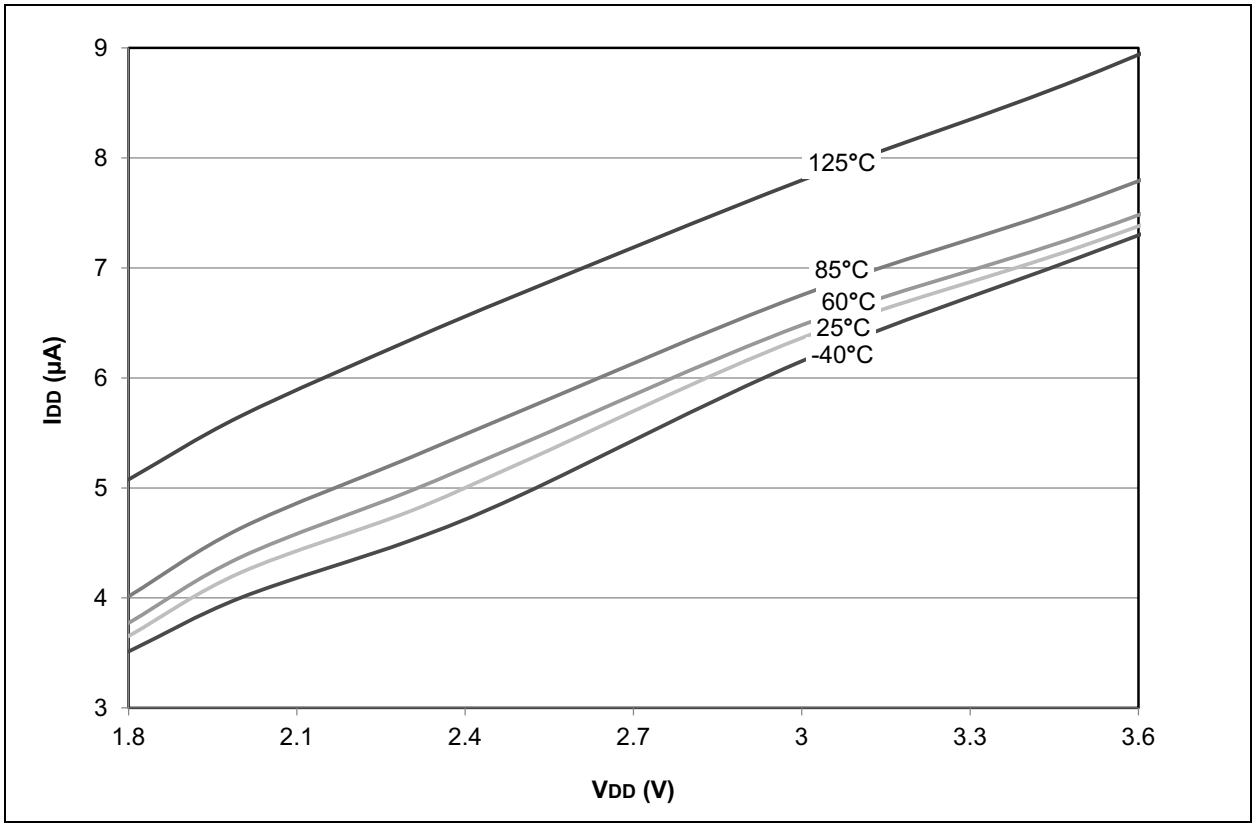


FIGURE 28-73: PIC18LF2X/4XK22 MAXIMUM IDD: SEC_RUN 32.768 kHz

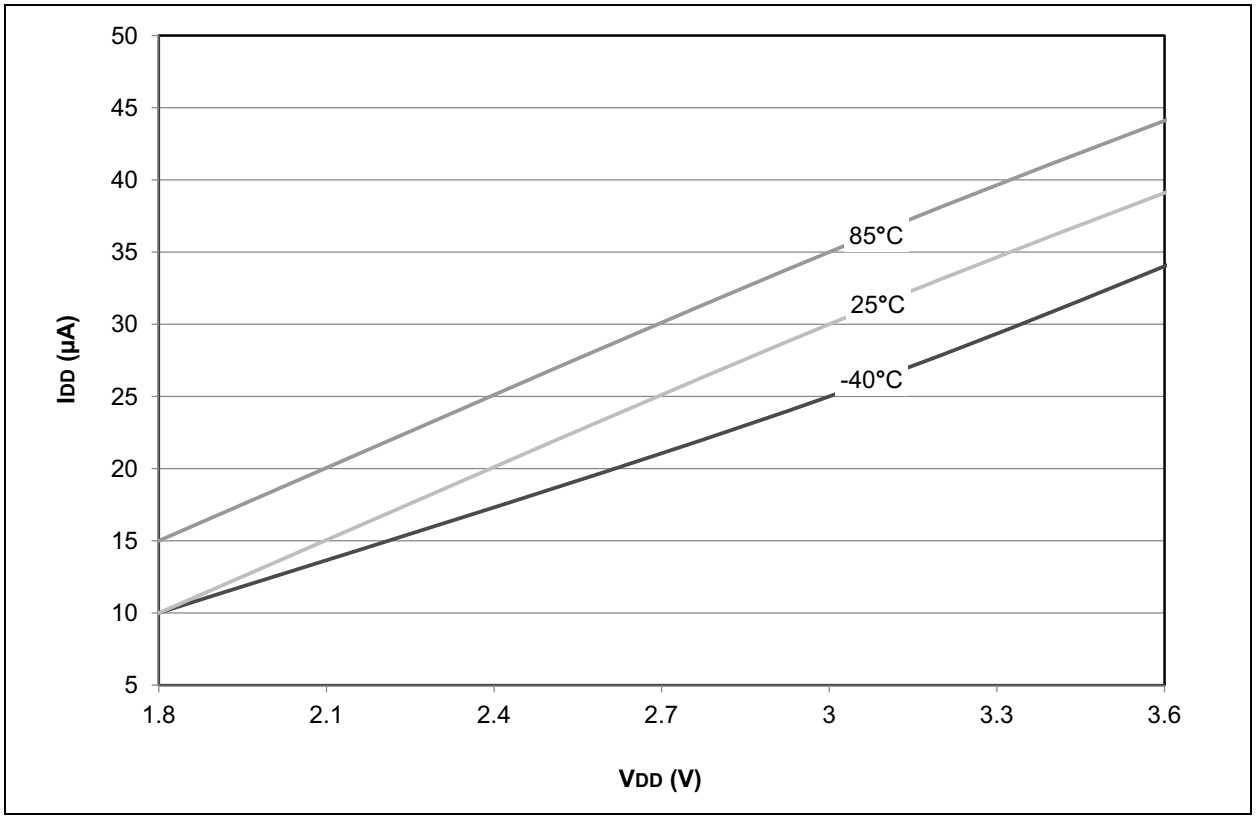
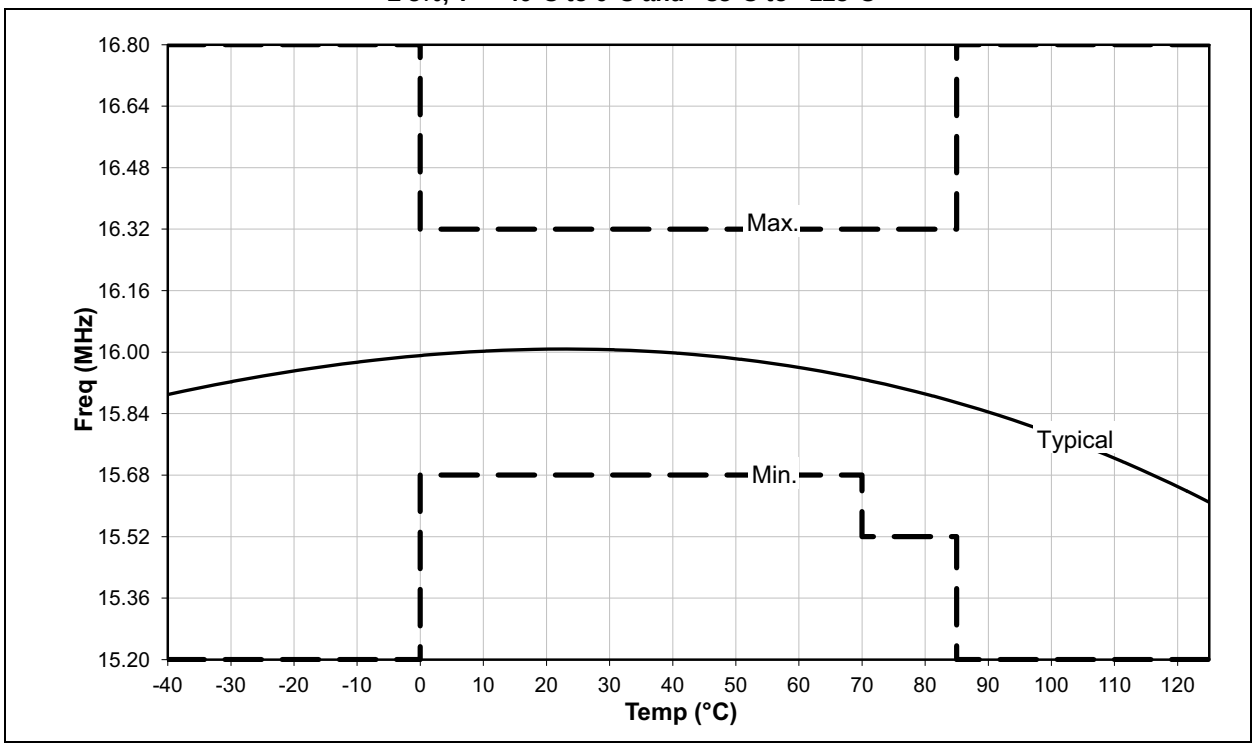
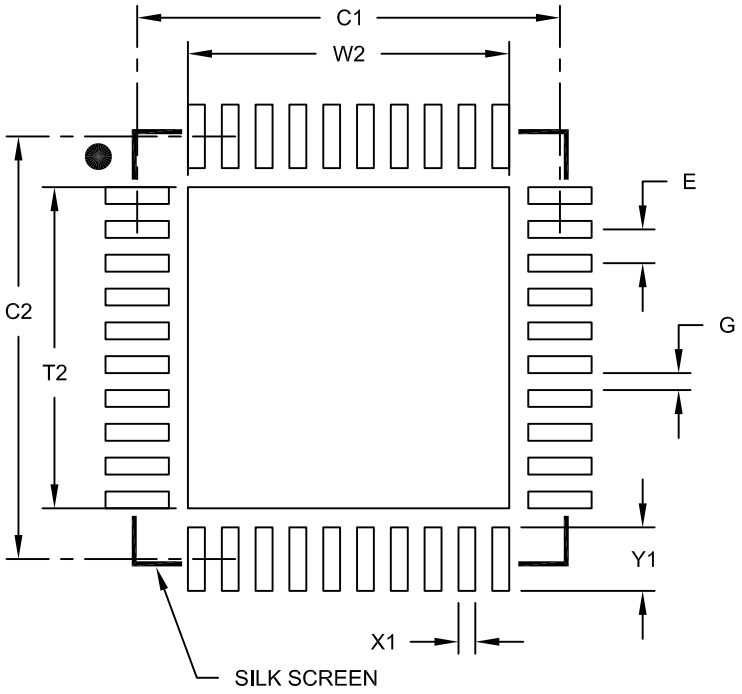


FIGURE 28-100: PIC18(L)F2X/4XK22 HF-INTOSC FREQUENCY vs. TEMPERATURE at 16 MHz
MIN / MAX: ± 2%, T = 0°C to +70°C
+2% / -3%, T = +70°C to +85°C
± 5%, T = -40°C to 0°C and +85°C to +125°C



40-Lead Plastic Ultra Thin Quad Flat, No Lead Package (MV) - 5x5 mm Body [UQFN]

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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.40 BSC	
Optional Center Pad Width	W2			3.80
Optional Center Pad Length	T2			3.80
Contact Pad Spacing	C1		5.00	
Contact Pad Spacing	C2		5.00	
Contact Pad Width (X40)	X1			0.20
Contact Pad Length (X40)	Y1			0.75
Distance Between Pads	G	0.20		

Notes:

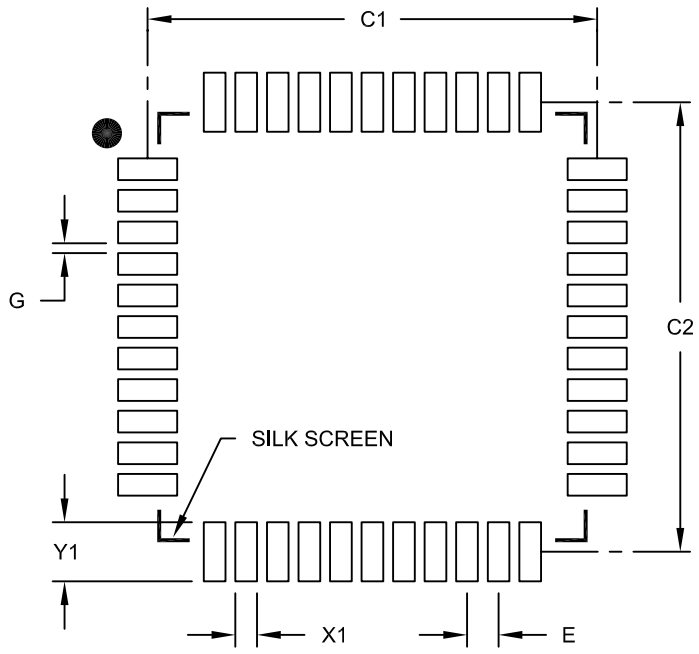
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2156B

44-Lead Plastic Thin Quad Flatpack (PT) 10X10X1 mm Body, 2.00 mm Footprint [TQFP]

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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.80 BSC	
Contact Pad Spacing	C1		11.40	
Contact Pad Spacing	C2		11.40	
Contact Pad Width (X44)	X1			0.55
Contact Pad Length (X44)	Y1			1.50
Distance Between Pads	G	0.25		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2076B