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	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	3.8K 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-SSOP (0.209", 5.30mm Width)
Supplier Device Package	28-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf26k22-e-ss

FIGURE 3: 40-PIN PDIP DIAGRAM

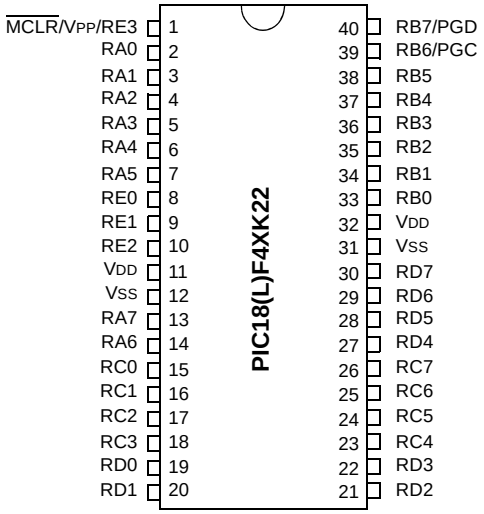


FIGURE 4: 40-PIN UQFN DIAGRAM

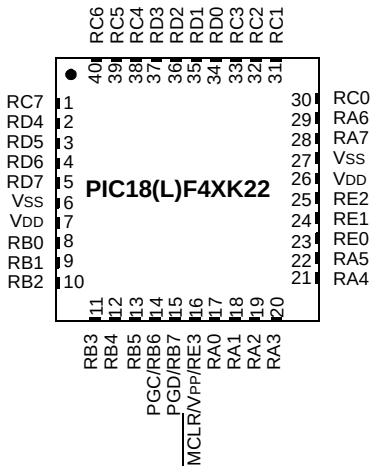


TABLE 1-2: PIC18(L)F2XK22 PINOUT I/O DESCRIPTIONS (CONTINUED)

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.

5.1.2.3 PUSH and POP Instructions

Since the Top-of-Stack is readable and writable, the ability to push values onto the stack and pull values off the stack without disturbing normal program execution is a desirable feature. The PIC18 instruction set includes two instructions, PUSH and POP, that permit the TOS to be manipulated under software control. TOSU, TOSH and TOSL can be modified to place data or a return address on the stack.

The PUSH instruction places the current PC value onto the stack. This increments the Stack Pointer and loads the current PC value onto the stack.

The POP instruction discards the current TOS by decrementing the Stack Pointer. The previous value pushed onto the stack then becomes the TOS value.

5.2 Register Definitions: Stack Pointer

REGISTER 5-1: STKPTR: STACK POINTER REGISTER

R/C-0	R/C-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
STKFUL(1)	STKUNF(1)	—	STKPTR<4:0>				
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented	C = Clearable only bit
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7 **STKFUL:** Stack Full Flag bit(1)
1 = Stack became full or overflowed
0 = Stack has not become full or overflowed

bit 6 **STKUNF:** Stack Underflow Flag bit(1)
1 = Stack Underflow occurred
0 = Stack Underflow did not occur

bit 5 **Unimplemented:** Read as ‘0’

bit 4-0 **STKPTR<4:0>:** Stack Pointer Location bits

Note 1: Bit 7 and bit 6 are cleared by user software or by a POR.

5.2.0.1 Stack Full and Underflow Resets

Device Resets on Stack Overflow and Stack Underflow conditions are enabled by setting the STVREN bit in Configuration Register 4L. When STVREN is set, a full or underflow will set the appropriate STKFUL or STKUNF bit and then cause a device Reset. When STVREN is cleared, a full or underflow condition will set the appropriate STKFUL or STKUNF bit but not cause a device Reset. The STKFUL or STKUNF bits are cleared by the user software or a Power-on Reset.

5.2.1 FAST REGISTER STACK

A fast register stack is provided for the Status, WREG and BSR registers, to provide a “fast return” option for interrupts. The stack for each register is only one level deep and is neither readable nor writable. It is loaded with the current value of the corresponding register when the processor vectors for an interrupt. All interrupt sources will push values into the stack registers. The values in the registers are then loaded back into their associated registers if the RETFIE,FAST instruction is used to return from the interrupt.

If both low and high priority interrupts are enabled, the stack registers cannot be used reliably to return from low priority interrupts. If a high priority interrupt occurs while servicing a low priority interrupt, the stack register values stored by the low priority interrupt will be overwritten. In these cases, users must save the key registers by software during a low priority interrupt.

If interrupt priority is not used, all interrupts may use the fast register stack for returns from interrupt. If no interrupts are used, the fast register stack can be used to restore the Status, WREG and BSR registers at the end of a subroutine call. To use the fast register stack for a subroutine call, a CALL label, FAST instruction must be executed to save the Status, WREG and BSR registers to the fast register stack. A RETURN, FAST instruction is then executed to restore these registers from the fast register stack.

Example 5-1 shows a source code example that uses the fast register stack during a subroutine call and return.

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

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR
F3Ah	ANSELC	ANSC7	ANSC6	ANSC5	ANSC4	ANSC3	ANSC2	—	—	1111 11--
F39h	ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	--11 1111
F38h	ANSELA	—	—	ANSA5	—	ANSA3	ANSA2	ANSA1	ANSA0	--1- 1111

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

Note 1: PIC18(L)F4XK22 devices only.

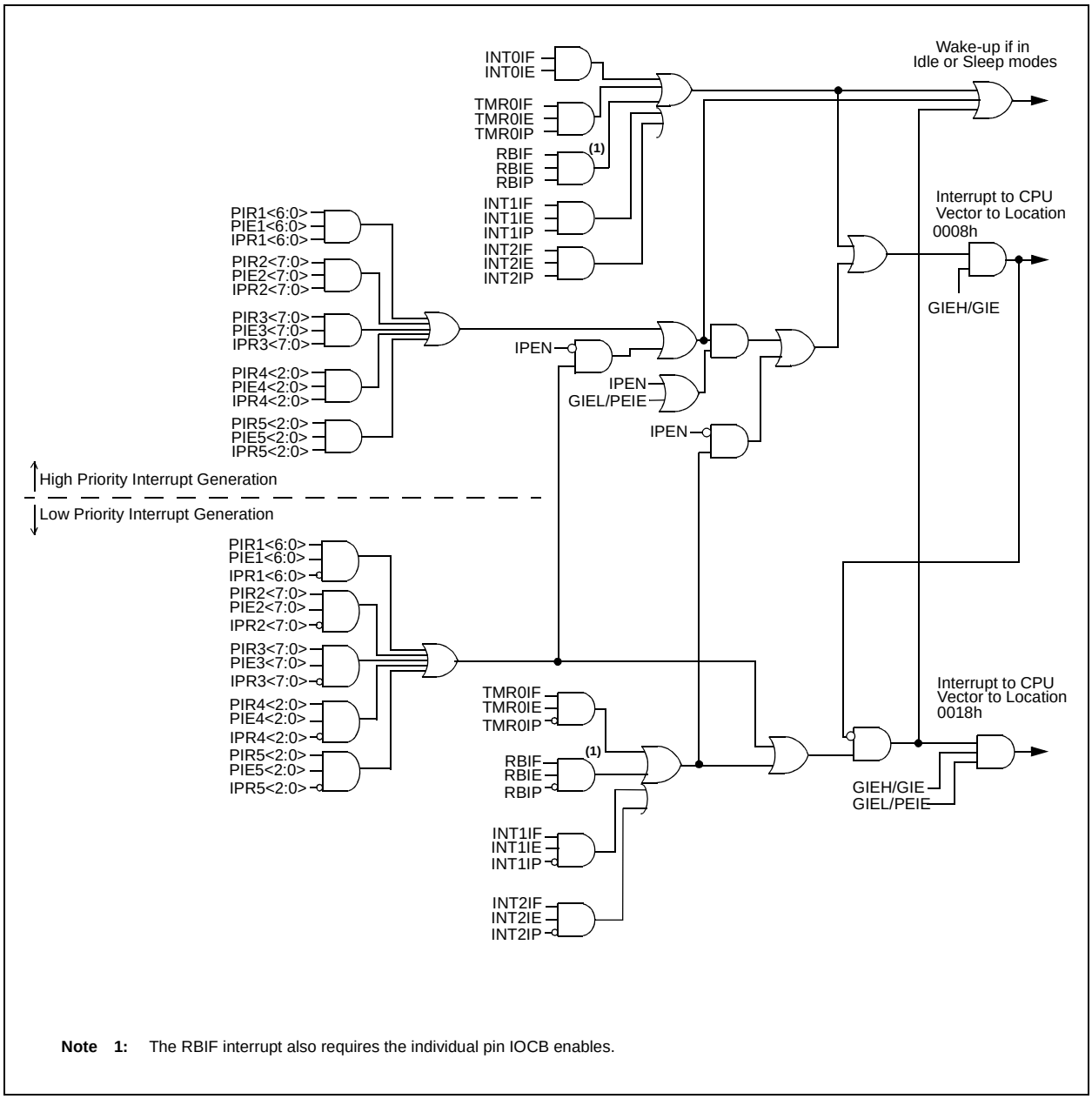
2: PIC18(L)F2XK22 devices only.

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

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

Note:	Do not use the MOVFF instruction to modify any of the interrupt control registers while **any** interrupt is enabled. Doing so may cause erratic microcontroller behavior.

FIGURE 9-1: PIC18 INTERRUPT LOGIC



Note 1: The RBIF interrupt also requires the individual pin IOCB enables.

REGISTER 9-8: PIR5: PERIPHERAL INTERRUPT (FLAG) REGISTER 5

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	TMR6IF	TMR5IF	TMR4IF
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 **TMR6IF:** TMR6 to PR6 Match Interrupt Flag bit
1 = TMR6 to PR6 match occurred (must be cleared in software)
0 = No TMR6 to PR6 match occurred

bit 1 **TMR5IF:** TMR5 Overflow Interrupt Flag bit
1 = TMR5 register overflowed (must be cleared in software)
0 = TMR5 register did not overflow

bit 0 **TMR4IF:** TMR4 to PR4 Match Interrupt Flag bit
1 = TMR4 to PR4 match occurred (must be cleared in software)
0 = No TMR4 to PR4 match occurred

10.1.1 PORTA OUTPUT PRIORITY

Each PORTA 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 PORTA pin functions from the highest to the lowest priority.

Analog input functions, such as ADC and comparator, 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.

TABLE 10-4: PORT PIN FUNCTION PRIORITY

Port bit	Port Function Priority by Port Pin				
	PORTA	**PORTB**	**PORTC**	**PORTD(2)**	**PORTE(2)**
0	RA0	CCP4(1)	SOSCO	SCL2	CCP3(8)
		RB0	P2B(6)	SCK2	P3A(8)
			RC0	RD0	RE0
1	RA1	SCL2(1)	SOSCI	SDA2	P3B
		SCK2(1)	CCP2(3)	CCP4	RE1
		P1C(1)	P2A(3)	RD1	
		RB1	RC1		
2	RA2	SDA2(1)	CCP1	P2B	CCP5
		P1B(1)	P1A	RD2(4)	RE2
		RB2	CTPLS		
			RC2		
3	RA3	SDO2(1)	SCL1	P2C	$\overline{\text{MCLR}}$
		CCP2(6)	SCK1	RD3	V_{PP}
		P2A(6)	RC3		RE3
		RB3			
4	SRQ	P1D(1)	SDA1	SDO2	
	C1OUT	RB4	RC4	P2D	
	CCP5(1)			RD4	
	RA4				

Note 1: PIC18(L)F2XK22 devices.
2: PIC18(L)F4XK22 devices.
3: Function default pin.
4: Function default pin (28-pin devices).
5: Function default pin (40/44-pin devices).
6: Function alternate pin.
7: Function alternate pin (28-pin devices).
8: Function alternate pin (40/44-pin devices)

REGISTER 10-4: ANSELB – PORTB ANALOG SELECT REGISTER

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0
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 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **ANSB<5:0>:** RB<5:0> Analog Select bit
1 = Digital input buffer disabled
0 = Digital input buffer enabled

REGISTER 10-5: ANSELC – PORTC ANALOG SELECT REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	U-0	U-0
ANSC7	ANSC6	ANSC5	ANSC4	ANSC3	ANSC2	—	—
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 7-2 **ANSC<7:2>:** RC<7:2> Analog Select bit
1 = Digital input buffer disabled
0 = Digital input buffer enabled

bit 1-0 **Unimplemented:** Read as ‘0’

REGISTER 10-6: ANSELD – PORTD ANALOG SELECT 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
ANSD7	ANSD6	ANSD5	ANSD4	ANSD3	ANSD2	ANSD1	ANSD0
bit 7							bit 0

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

bit 7-0 **ANSD<7:0>:** RD<7:0> Analog Select bit
1 = Digital input buffer disabled
0 = Digital input buffer enabled

FIGURE 12-5: TIMER1/3/5 GATE TOGGLE MODE

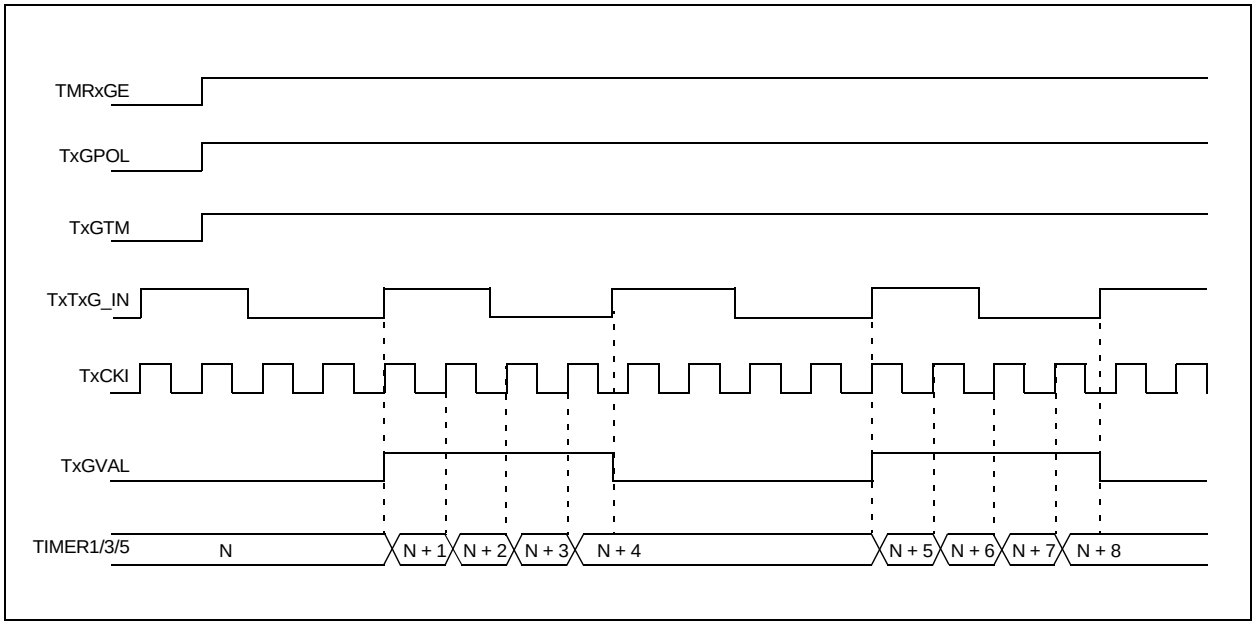
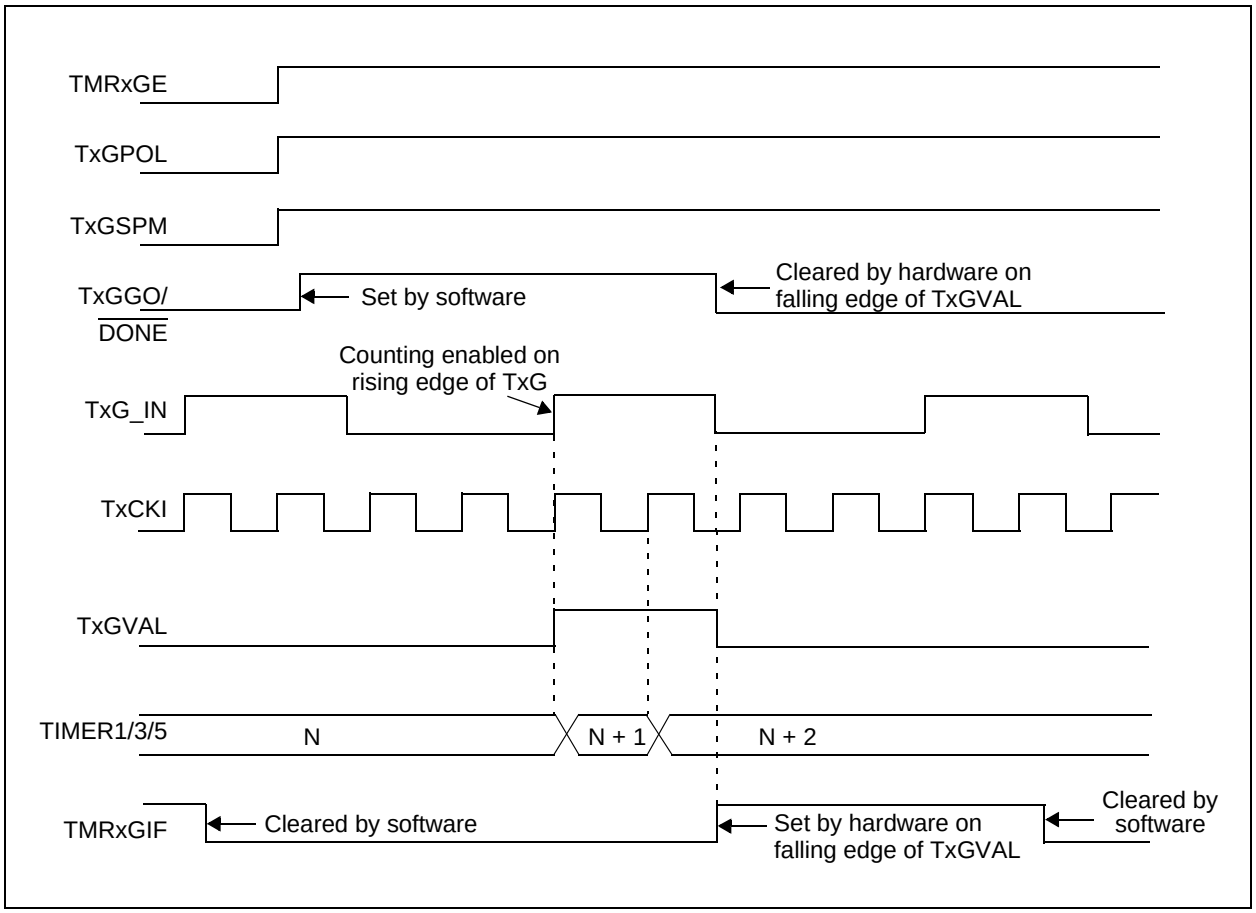


FIGURE 12-6: TIMER1/3/5 GATE SINGLE-PULSE MODE



15.4.9 ACKNOWLEDGE SEQUENCE

The 9th SCLx pulse for any transferred byte in I^2C is dedicated as an Acknowledge. It allows receiving devices to respond back to the transmitter by pulling the SDAx line low. The transmitter must release control of the line during this time to shift in the response. The Acknowledge ($\overline{ACK}$) is an active-low signal, pulling the SDAx line low indicated to the transmitter that the device has received the transmitted data and is ready to receive more.

The result of an $\overline{ACK}$ is placed in the ACKSTAT bit of the SSPxCON2 register.

Slave software, when the AHEN and DHEN bits are set, allow the user to set the $\overline{ACK}$ value sent back to the transmitter. The ACKDT bit of the SSPxCON2 register is set/cleared to determine the response.

Slave hardware will generate an $\overline{ACK}$ response if the AHEN and DHEN bits of the SSPxCON3 register are clear.

There are certain conditions where an $\overline{ACK}$ will not be sent by the slave. If the BF bit of the SSPxSTAT register or the SSPxOV bit of the SSPxCON1 register are set when a byte is received.

When the module is addressed, after the 8th falling edge of SCLx on the bus, the ACKTIM bit of the SSPxCON3 register is set. The ACKTIM bit indicates the acknowledge time of the active bus.

The ACKTIM Status bit is only active when the AHEN bit or DHEN bit is enabled.

15.5 I^2C Slave Mode Operation

The MSSPx Slave mode operates in one of four modes selected in the SSPxM bits of SSPxCON1 register. The modes can be divided into 7-bit and 10-bit Addressing mode. 10-bit Addressing modes operate the same as 7-bit with some additional overhead for handling the larger addresses.

Modes with Start and Stop bit interrupts operated the same as the other modes with SSPxIF additionally getting set upon detection of a Start, Restart, or Stop condition.

15.5.1 SLAVE MODE ADDRESSES

The SSPxADD register (Register 15-7) contains the Slave mode address. The first byte received after a Start or Restart condition is compared against the value stored in this register. If the byte matches, the value is loaded into the SSPxBUF register and an interrupt is generated. If the value does not match, the module goes Idle and no indication is given to the software that anything happened.

The SSPx Mask register (Register 15-6) affects the address matching process. See **Section 15.5.9 "SSPx Mask Register"** for more information.

15.5.1.1 I^2C Slave 7-bit Addressing Mode

In 7-bit Addressing mode, the LSb of the received data byte is ignored when determining if there is an address match.

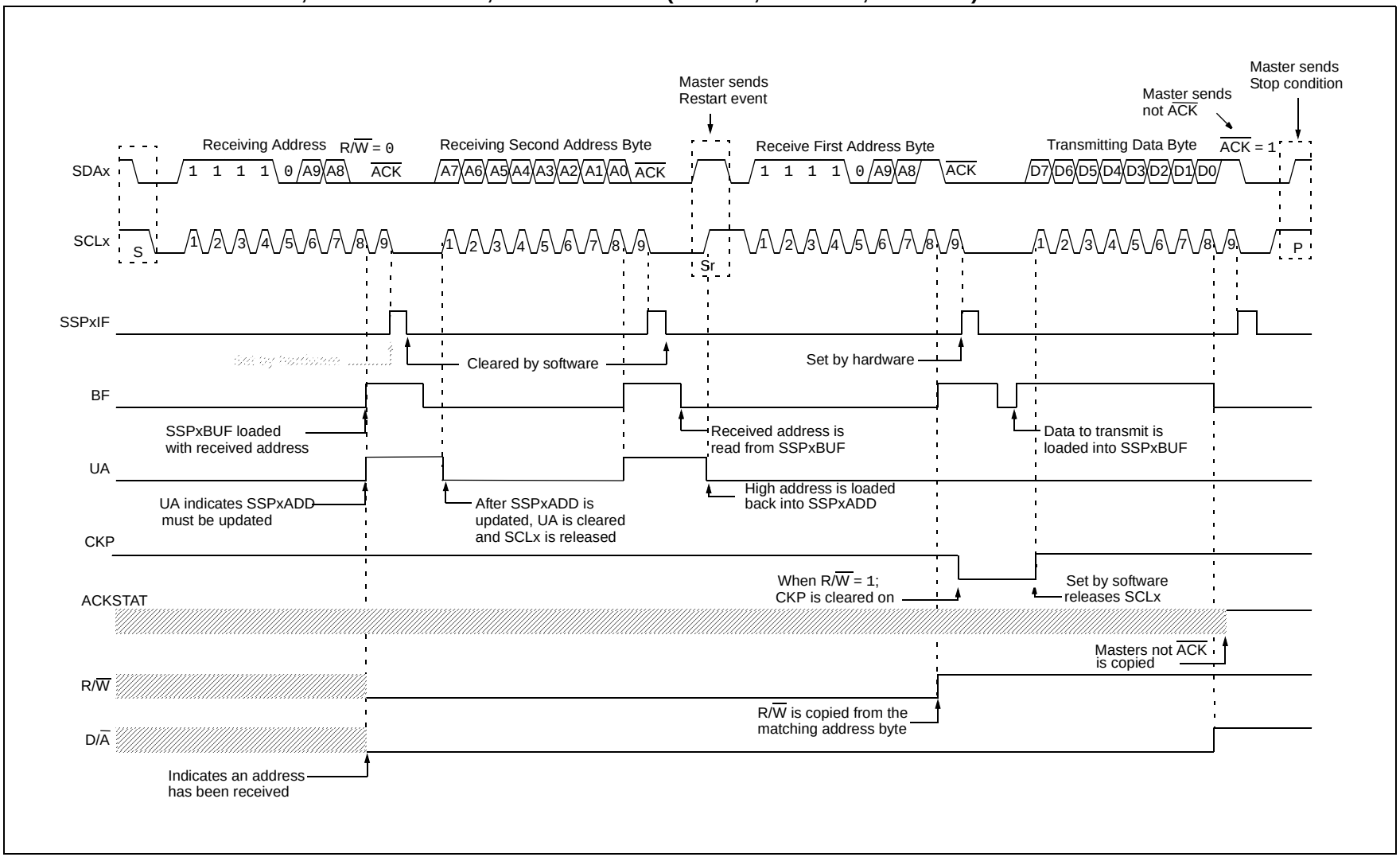
15.5.1.2 I^2C Slave 10-bit Addressing Mode

In 10-bit Addressing mode, the first received byte is compared to the binary value of '1 1 1 1 0 A9 A8 0'. A9 and A8 are the two MSb of the 10-bit address and stored in bits 2 and 1 of the SSPxADD register.

After the acknowledge of the high byte the UA bit is set and SCLx is held low until the user updates SSPxADD with the low address. The low address byte is clocked in and all eight bits are compared to the low address value in SSPxADD. Even if there is not an address match; SSPxIF and UA are set, and SCLx is held low until SSPxADD is updated to receive a high byte again. When SSPxADD is updated the UA bit is cleared. This ensures the module is ready to receive the high address byte on the next communication.

A high and low address match as a write request is required at the start of all 10-bit addressing communication. A transmission can be initiated by issuing a Restart once the slave is addressed, and clocking in the high address with the R/$\overline{W}$ bit set. The slave hardware will then acknowledge the read request and prepare to clock out data. This is only valid for a slave after it has received a complete high and low address byte match.

FIGURE 15-22: I²C SLAVE, 10-BIT ADDRESS, TRANSMISSION (SEN = 0, AHEN = 0, DHEN = 0)



16.3 Register Definitions: EUSART Control

REGISTER 16-1: TxSTAx: TRANSMIT STATUS AND CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN(1)	SYNC	SENDB	BRGH	TRMT	TX9D
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 **CSRC:** Clock Source Select bit
Asynchronous mode:
Don't care
Synchronous mode:
1 = Master mode (clock generated internally from BRG)
0 = Slave mode (clock from external source)

bit 6 **TX9:** 9-bit Transmit Enable bit
1 = Selects 9-bit transmission
0 = Selects 8-bit transmission

bit 5 **TXEN:** Transmit Enable bit(1)
1 = Transmit enabled
0 = Transmit disabled

bit 4 **SYNC:** EUSART Mode Select bit
1 = Synchronous mode
0 = Asynchronous mode

bit 3 **SENDB:** Send Break Character bit
Asynchronous mode:
1 = Send Sync Break on next transmission (cleared by hardware upon completion)
0 = Sync Break transmission completed
Synchronous mode:
Don't care

bit 2 **BRGH:** High Baud Rate Select bit
Asynchronous mode:
1 = High speed
0 = Low speed
Synchronous mode:
Unused in this mode

bit 1 **TRMT:** Transmit Shift Register Status bit
1 = TSR empty
0 = TSR full

bit 0 **TX9D:** Ninth bit of Transmit Data
Can be address/data bit or a parity bit.

Note 1: SREN/CREN overrides TXEN in Sync mode.

EXAMPLE 19-3: CAPACITANCE CALIBRATION ROUTINE

```
#include "p18cxxx.h"

#define COUNT 25                            //@ 8MHz INTFRC = 62.5 us.
#define ETIME COUNT*2.5                     //time in uS
#define DELAY for(i=0;i<COUNT;i++)
#define ADSCALE 1023                        //for unsigned conversion 10 sig
bits
#define ADREF 3.3                           //Vdd connected to A/D Vr+
#define RCAL .027                           //R value is 4200000 (4.2M)
                                            //scaled so that result is in
                                            //1/100th of uA

int main(void)
{
    int i;
    int j = 0;                              //index for loop
    unsigned int Vread = 0;
    float CTMUISrc, CTMUCap, Vavg, VTot, Vcal;

//assume CTMU and A/D have been set up correctly
//see Example 25-1 for CTMU & A/D setup
setup();

CTMUCONHbits.CTMUEN = 1;                    //Enable the CTMU
CTMUCONLbits.EDG1STAT = 0;                  // Set Edge status bits to zero
CTMUCONLbits.EDG2STAT = 0;
    for(j=0;j<10;j++)
    {
        CTMUCONHbits.IDISSEN = 1;           //drain charge on the circuit
        DELAY;                              //wait 125us
        CTMUCONHbits.IDISSEN = 0;           //end drain of circuit

        CTMUCONLbits.EDG1STAT = 1;          //Begin charging the circuit
                                            //using CTMU current source
        DELAY;                              //wait for 125us
        CTMUCONLbits.EDG1STAT = 0;          //Stop charging circuit

        PIR1bits.ADIF = 0;                  //make sure A/D Int not set
        ADCON0bits.GO=1;                    //and begin A/D conv.
        while(!PIR1bits.ADIF);              //Wait for A/D convert complete


        Vread = ADRES;                      //Get the value from the A/D
        PIR1bits.ADIF = 0;                  //Clear A/D Interrupt Flag
        VTot += Vread;                      //Add the reading to the total
    }

    Vavg = (float)(VTot/10.000);            //Average of 10 readings
    Vcal = (float)(Vavg/ADSCALE*ADREF);
    CTMUISrc = Vcal/RCAL;                   //CTMUISrc is in 1/100ths of uA
    CTMUCap = (CTMUISrc*ETIME/Vcal)/100;
}
```

22.7 Operation During Sleep

When the device wakes up from Sleep through an interrupt or a Watchdog Timer time-out, the contents of the VREFCON1 register are not affected. To minimize current consumption in Sleep mode, the voltage reference should be disabled.

22.8 Effects of a Reset

A device Reset affects the following:

- DAC is disabled
- DAC output voltage is removed from the DACOUT pin
- The DACR<4:0> range select bits are cleared

22.9 Register Definitions: DAC Control

REGISTER 22-1: VREFCON1: VOLTAGE REFERENCE CONTROL REGISTER 0

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	U-0	R/W-0
DACEN	DACLPS	DACOE	—	DACPSS<1:0>		—	DACNSS
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7 **DACEN:** DAC Enable bit
1 = DAC is enabled
0 = DAC is disabled

bit 6 **DACLPS:** DAC Low-Power Voltage Source Select bit
1 = DAC Positive reference source selected
0 = DAC Negative reference source selected

bit 5 **DACOE:** DAC Voltage Output Enable bit
1 = DAC voltage level is also an output on the DACOUT pin
0 = DAC voltage level is disconnected from the DACOUT pin

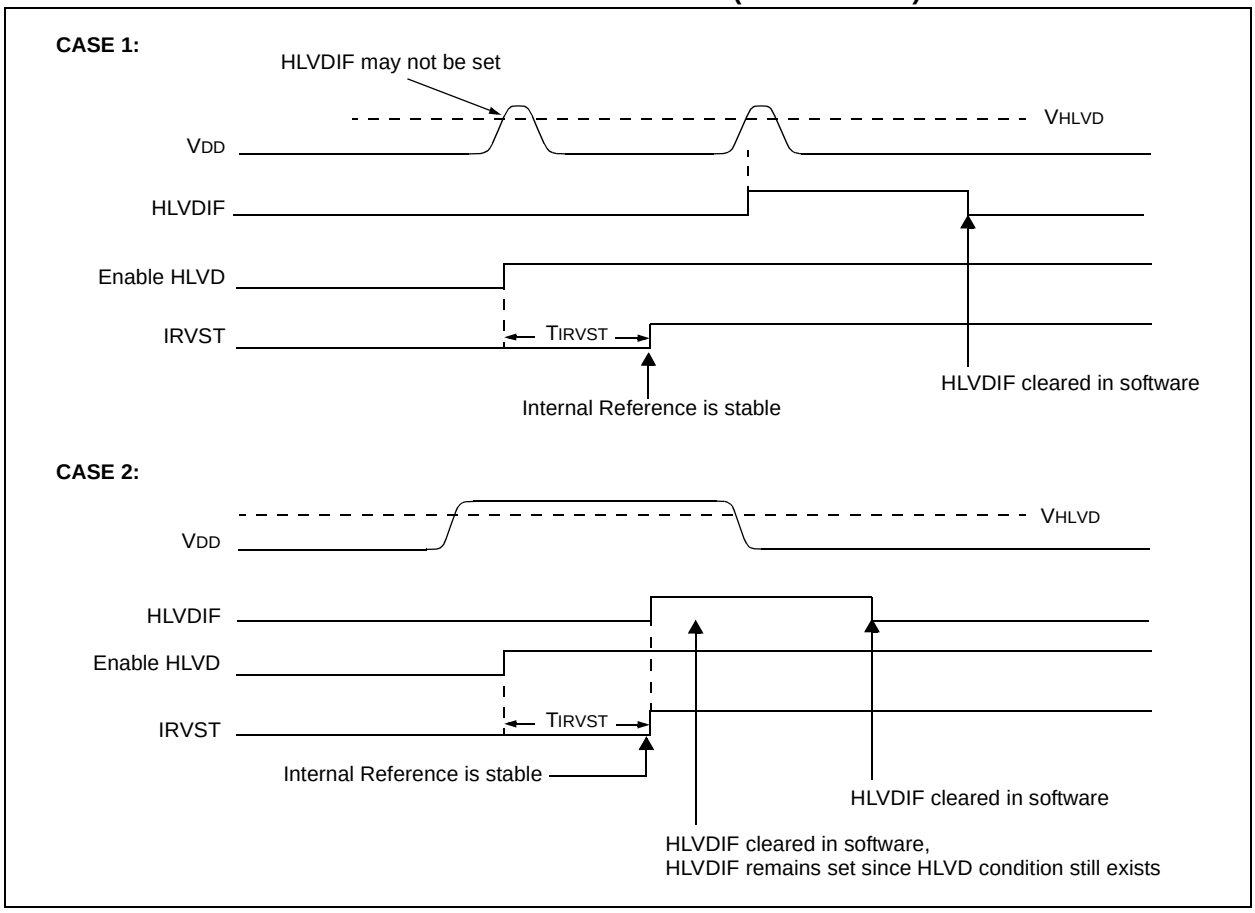
bit 4 **Unimplemented:** Read as ‘0’

bit 3-2 **DACPSS<1:0>:** DAC Positive Source Select bits
00 = V_{DD}
01 = $V_{REF}+$
10 = FVR BUF1 output
11 = Reserved, do not use

bit 1 **Unimplemented:** Read as ‘0’

bit 0 **DACNSS:** DAC Negative Source Select bits
1 = $V_{REF}-$
0 = V_{SS}

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

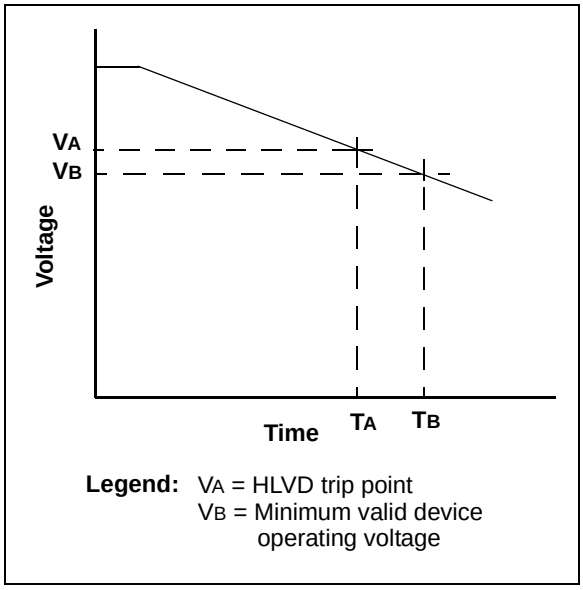


23.6 Applications

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



MOVFF **Move f to f**

Syntax: MOVFF f_s,f_d

Operands: $0 \le f_s \le 4095$
$0 \le f_d \le 4095$

Operation: $(f_s) \rightarrow f_d$

Status Affected: None

Encoding:

1st word (source)	1100	ffff	ffff$_s$
2nd word (destin.)	1111	ffff	ffff$_d$

Description: The contents of source register 'f_s' are moved to destination register 'f_d'. Location of source 'f_s' can be anywhere in the 4096-byte data space (000h to FFFh) and location of destination 'f_d' can also be anywhere from 000h to FFFh.
Either source or destination can be W (a useful special situation).
MOVFF is particularly useful for transferring a data memory location to a peripheral register (such as the transmit buffer or an I/O port).
The MOVFF instruction cannot use the PCL, TOSU, TOSH or TOSL as the destination register.

Words: 2

Cycles: 2 (3)

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f' (src)	Process Data	No operation
Decode	No operation No dummy read	No operation	Write register 'f' (dest)

Example: MOVFF REG1, REG2

Before Instruction
REG1 = 33h
REG2 = 11h
After Instruction
REG1 = 33h
REG2 = 33h

MOVLB **Move literal to low nibble in BSR**

Syntax: MOVLW k

Operands: $0 \le k \le 255$

Operation: $k \rightarrow$ BSR

Status Affected: None

Encoding:

0000	0001	kkkk	kkkk

Description: The 8-bit literal 'k' is loaded into the Bank Select Register (BSR). The value of BSR<7:4> always remains '0', regardless of the value of k_7:k_4.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write literal 'k' to BSR

Example: MOVLB 5

Before Instruction
BSR Register = 02h
After Instruction
BSR Register = 05h

27.2 DC Characteristics: Power-Down 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 +25°C**	**Typ +60°C**	**Max +85°C**	**Max +125°C**	**Units**	**Conditions**	
							VDD	**Notes**
Power-down Base Current (IPD)(1)								
D006	Sleep mode	0.01	0.04	2	10	µA	1.8V	WDT, BOR, FVR and SOSC disabled, all Peripherals inactive
		0.01	0.06	2	10	µA	3.0V	
		12	13	25	35	µA	2.3V	
		13	14	30	40	µA	3.0V	
		13	14	35	50	µA	5.0V	
Power-down Module Differential Current (delta IPD)								
D007	Watchdog Timer	0.3	0.3	2.5	2.5	µA	1.8V	
		0.5	0.5	2.5	2.5	µA	3.0V	
		0.35	0.35	5.0	5.0	µA	2.3V	
		0.5	0.5	5.0	5.0	µA	3.0V	
		0.5	0.5	5.0	5.0	µA	5.0V	
D008	Brown-out Reset(2)	8	8.5	15	16	µA	2.0V	
		9	9.5	15	16	µA	3.0V	
		3.4	3.4	15	16	µA	2.3V	
		3.8	3.8	15	16	µA	3.0V	
		5.2	5.2	15	16	µA	5.0V	
D010	High/Low Voltage Detect(2)	6.5	6.7	15	15	µA	2.0V	
		7	7.5	15	15	µA	3.0V	
		2.1	2.1	15	15	µA	2.3V	
		2.4	2.4	15	15	µA	3.0V	
		3.2	3.2	15	15	µA	5.0V	
D011	Secondary Oscillator	0.5	1	3	10	µA	1.8V	32 kHz on SOSC
		0.6	1.1	4	10	µA	3.0V	
		0.5	1	3	10	µA	2.3V	
		0.6	1.1	4	10	µA	3.0V	
		0.6	1.1	5	10	µA	5.0V	

Note 1: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD or VSS and all features that add delta current disabled (such as WDT, Timer1 Oscillator, BOR, etc.).

2: On PIC18LF2X/4XK22 the BOR, HLVD and FVR enable internal band gap reference. With more than one of these modules enabled, the current consumption will be less than the sum of the specifications. On PIC18F2X/4XK22, the internal band gap reference is always enabled and its current consumption is included in the Power-down Base Current (IPD).

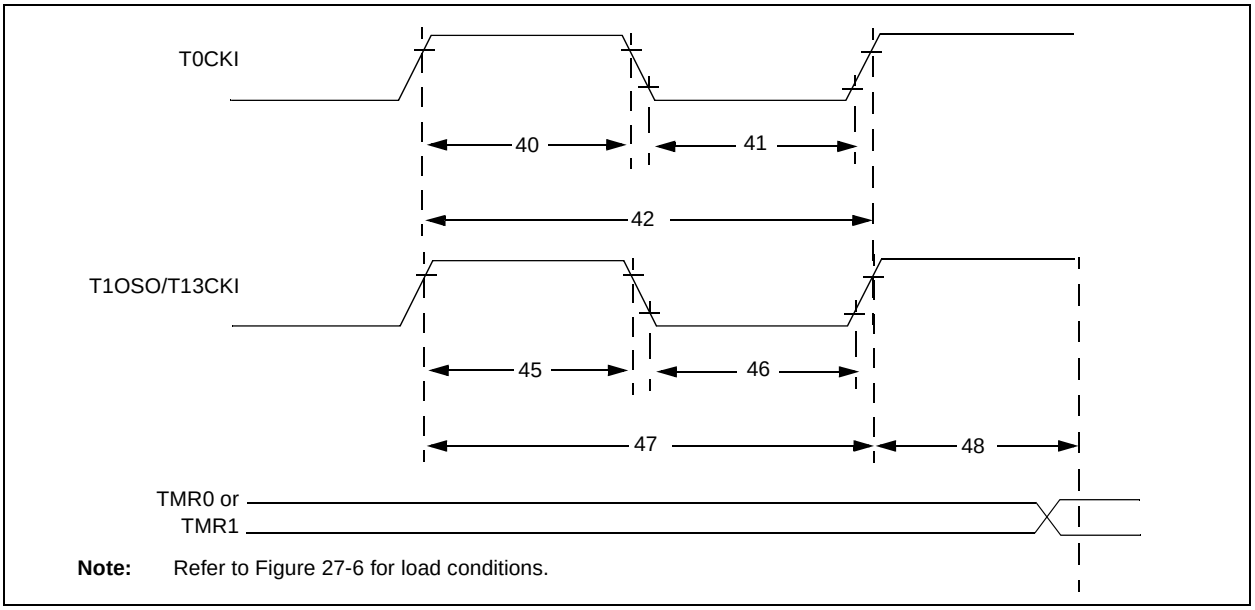
3: A/D converter differential currents apply only in Run mode. In Sleep or Idle mode both the ADC and the FRC turn off as soon as conversion (if any) is complete.

TABLE 27-11: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER AND BROWN-OUT RESET REQUIREMENTS

Param. No.	Symbol	Characteristic	Min	Typ	Max	Units	Conditions
30	TmcL	$\overline{MCLR}$ Pulse Width (low)	2	—	—	µs	
31	TWDT	Watchdog Timer Time-out Period (no postscaler)	3.5	4.1	4.7	ms	1:1 prescaler
32	TOST	Oscillation Start-up Timer Period	1024 TOSC	—	1024 TOSC	—	TOSC = OSC1 period
33	TPWRT	Power-up Timer Period	54.8	64.4	74.1	ms	
34	TIOZ	I/O High-Impedance from $\overline{MCLR}$ Low or Watchdog Timer Reset	—	2	—	µs	
35	TBOR	Brown-out Reset Pulse Width	200[1]	—	—	µs	VDD ≤ BVDD (see D005)
36	TIVRST	Internal Reference Voltage Stable	—	25	35	µs	
37	THLVD	High/Low-Voltage Detect Pulse Width	200[1]	—	—	µs	VDD ≤ VHLVD
38	TCSD	CPU Start-up Time	5	—	10	µs	
39	TIOBST	Time for HF-INTOSC to Stabilize	—	0.25	1	ms	
40	TIOSC_ST	Time for HF-INTOSC to Start	—	TBD	TBD	µs	

Note 1: Minimum pulse width that will consistently trigger a reset or interrupt. Shorter pulses may intermittently trigger a response.

FIGURE 27-11: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS



Note: Refer to Figure 27-6 for load conditions.

FIGURE 28-46: PIC18F2X/4XK22 TYPICAL IDD: RC_IDLE HF-INTOSC with PLL

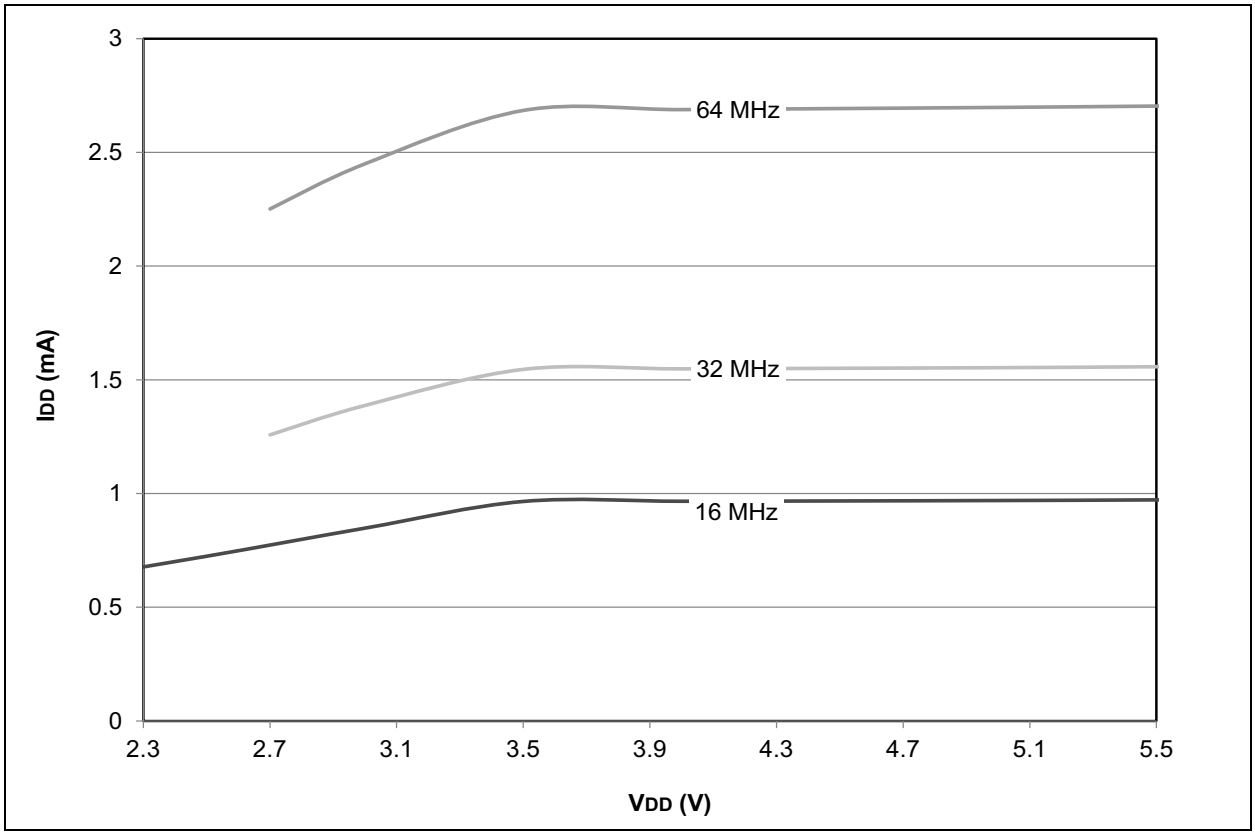


FIGURE 28-47: PIC18F2X/4XK22 MAXIMUM IDD: RC_IDLE HF-INTOSC with PLL

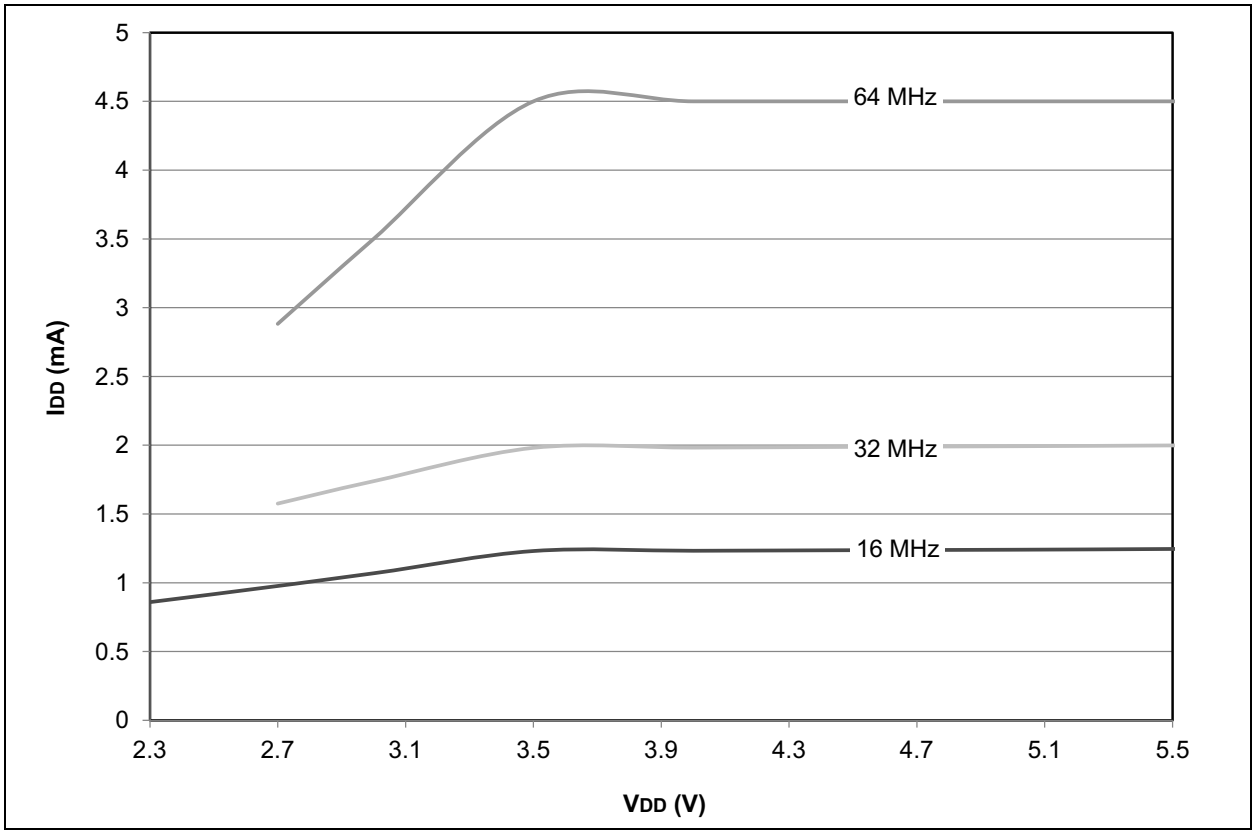


FIGURE 28-62: PIC18F2X/4XK22 TYPICAL IDD: PRI_IDLE EC MEDIUM POWER

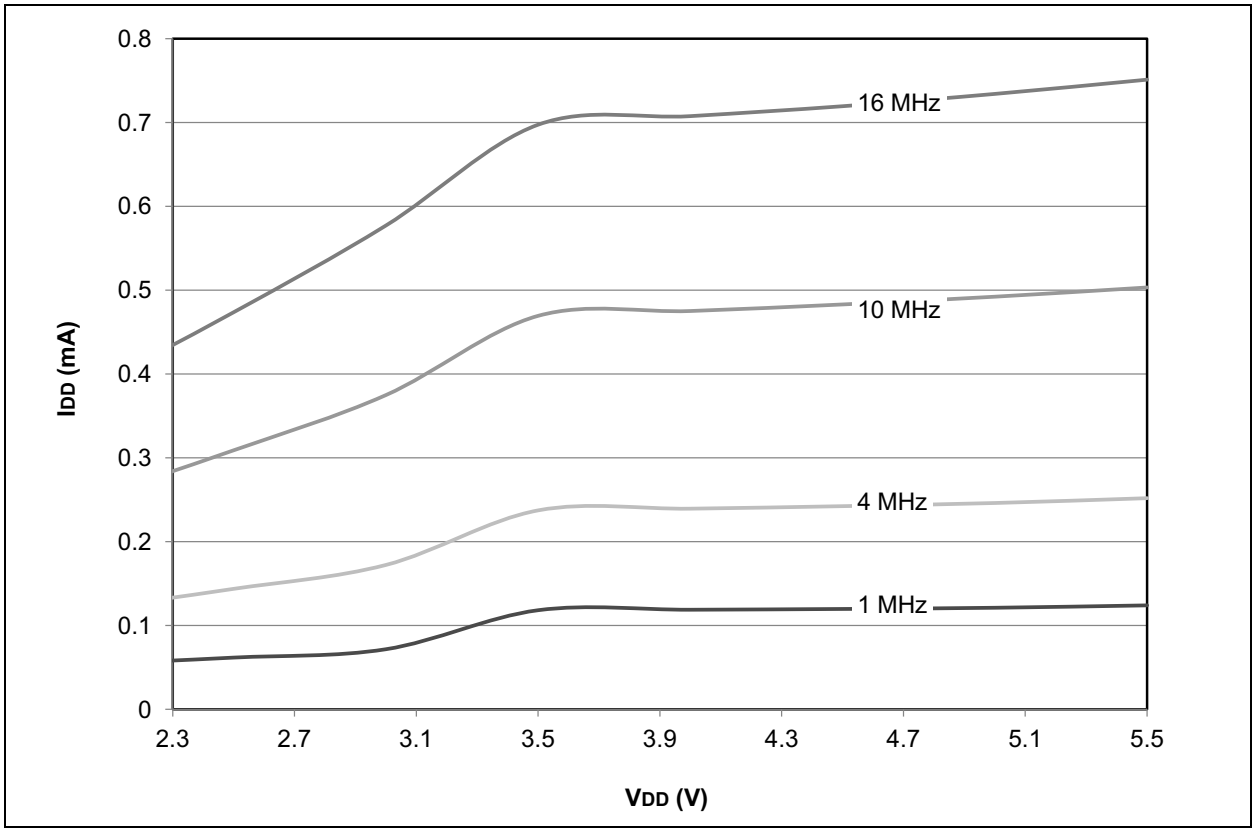


FIGURE 28-63: PIC18F2X/4XK22 MAXIMUM IDD: PRI_IDLE EC MEDIUM POWER