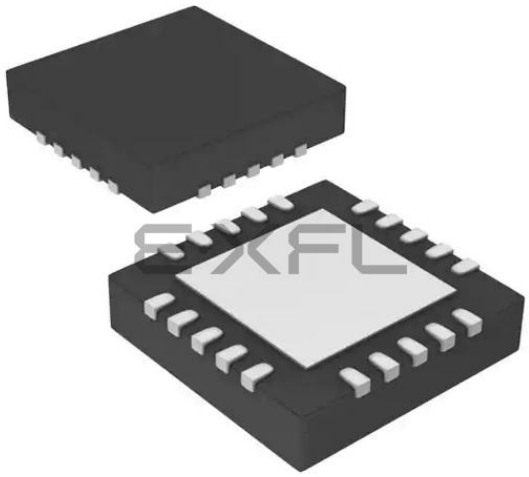


Welcome to E-XFL.COM

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, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	17
Program Memory Size	16KB (8K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	20-VFQFN Exposed Pad
Supplier Device Package	20-QFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf14k22-e-ml

TABLE 1-1: DEVICE FEATURES FOR THE PIC18(L)F1XK22 (20-PIN DEVICES)

Features	PIC18F13K22	PIC18LF13K22	PIC18F14K22	PIC18LF14K22
Voltage Range (1.8 - 5.5V)	2.3-5.5V	1.8V-3.6V	2.3-5.5V	1.8V-3.6V
Program Memory (Bytes)	8K		16K	
Program Memory (Instructions)	4096		8192	
Data Memory (Bytes)	256		512	
Operating Frequency	DC – 64 MHz			
Interrupt Sources	30			
I/O Ports	Ports A, B, C			
Timers	4			
Enhanced Capture/ Compare/PWM Modules	1			
Serial Communications	MSSP, Enhanced USART			
10-Bit Analog-to-Digital Module	12 Input Channels			
Resets (and Delays)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow, $\overline{\text{MCLR}}$, WDT (PWRT, OST)			
Instruction Set	75 Instructions, 83 with Extended Instruction Set Enabled			
Packages	20-Pin PDIP, SSOP, SOIC QFN (4x4x0.9mm)			

TABLE 1-2: PIC18(L)F1XK22 PIN SUMMARY (CONTINUED)

Pin Name	Pin Number		Pin Type	Buffer Type	Description
	PDIP/SSOP/SOIC	QFN			
RB5/AN11/RX/DT	12	9			
RB5			I/O	TLL	Digital I/O
AN11			I	Analog	ADC channel 11
RX			I	ST	EUSART asynchronous receive
DT			I/O	ST	EUSART synchronous data (see related RX/TX)
RB6/SCK/SCL	11	8			
RB6			I/O	TLL	Digital I/O
SCK			I/O	ST	Synchronous serial clock input/output for SPI mode
SCL			I/O	ST	Synchronous serial clock input/output for I^2C mode
RB7/TX/CK	10	7			
RB7			I/O	TLL	Digital I/O
TX			O	CMOS	EUSART asynchronous transmit
CK			I/O	ST	EUSART synchronous clock (see related RX/DT)
RC0/AN4/C2IN+	16	13			
RC0			I/O	ST	Digital I/O
AN4			I	Analog	ADC channel 4
C2IN+			I	Analog	Comparator C2 noninverting input
RC1/AN5/C12IN-	15	12			
RC1			I/O	ST	Digital I/O
AN5			I	Analog	ADC channel 5
C12IN-			I	Analog	Comparator C1 and C2 inverting input
RC2/AN6/C12IN2-/P1D	14	11			
RC2			I/O	ST	Digital I/O
AN6			I	Analog	ADC channel 6
C12IN2-			I	Analog	Comparator C1 and C2 inverting input
P1D			O	CMOS	Enhanced CCP1 PWM output
RC3/AN7/C12IN3-/P1C/PGM	7	4			
RC3			I/O	ST	Digital I/O
AN7			I	Analog	ADC channel 7
C12IN3-			I	Analog	Comparator C1 and C2 inverting input
P1C			O	CMOS	Enhanced CCP1 PWM output
PGM			I/O	ST	Low-Voltage ICSP Programming enable pin
RC4/C2OUT/P1B/SRNQ	6	3			
RC4			I/O	ST	Digital I/O
C2OUT			O	CMOS	Comparator C2 output
P1B			O	CMOS	Enhanced CCP1 PWM output
SRNQ			O	CMOS	SR latch inverted output
RC5/CCP1/P1A	5	2			
RC5			I/O	ST	Digital I/O
CCP1			I/O	ST	Capture 1 input/Compare 1 output/PWM 1 output
P1A			O	CMOS	Enhanced CCP1 PWM output
RC6/AN8/$\overline{SS}$	8	5			
RC6			I/O	ST	Digital I/O
AN8			I	Analog	ADC channel 8
$\overline{SS}$			I	TTL	SPI slave select input
RC7/AN9/SDO	9	6			
RC7			I/O	ST	Digital I/O
AN9			I	Analog	ADC channel 9
SDO			O	CMOS	SPI data out
VSS	20	17	P	—	Ground reference for logic and I/O pins
VDD	1	18	P	—	Positive supply for logic and I/O pins

Legend: TTL = TTL compatible input
ST = Schmitt Trigger input
O = Output
XTAL = Crystal Oscillator
CMOS = CMOS compatible input or output
I = Input
P = Power

FIGURE 2-7: FSCM TIMING DIAGRAM

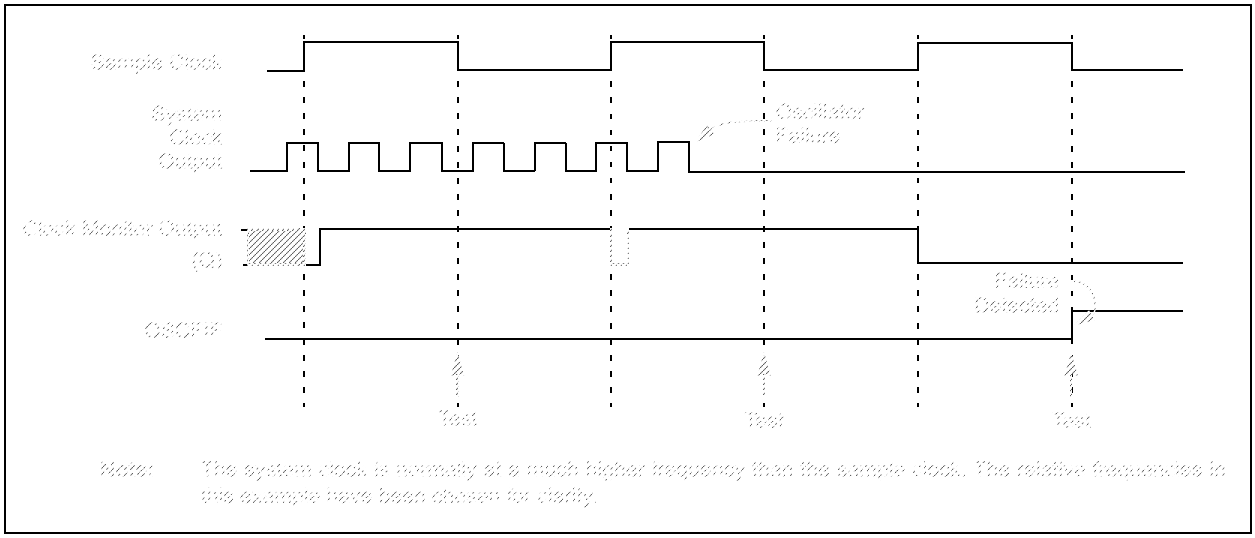


TABLE 2-5: SUMMARY OF REGISTERS ASSOCIATED WITH CLOCK SOURCES

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
CONFIG1H	IESO	FCMEN	PCLKEN	PLL_EN	FOSC3	FOSC2	FOSC1	FOSC0	251
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RABIE	TMR0IF	INT0IF	RABIF	245
OSCCON	IDLEN	IRCF2	IRCF1	IRCF0	OSTS	HFIOFS	SCS1	SCS0	246
OSCCON2	—	—	—	—	—	PRI_SD	HFIOFL	LFIOFS	246
OSCTUNE	INTSRC	PLLEN	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0	248
IPR2	OSCFIP	C1IP	C2IP	EEIP	BCLIP	—	TMR3IP	—	248
PIE2	OSCFIE	C1IE	C2IE	EEIE	BCLIE	—	TMR3IE	—	248
PIR2	OSCFIF	C1IF	C2IF	EEIF	BCLIF	—	TMR3IF	—	248
T1CON	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	246

Legend: x = unknown, u = unchanged, – = unimplemented locations read as '0'. Shaded cells are not used by oscillators.
Note 1: Other (non Power-up) Resets include $\overline{\text{MCLR}}$ Reset and Watchdog Timer Reset during normal operation.

EXAMPLE 4-3: WRITING TO FLASH PROGRAM MEMORY (CONTINUED)

```
                DECFSZ  COUNTER                 ; loop until holding registers are full
                BRA     WRITE_WORD_TO_HREGS
PROGRAM_MEMORY
                BSF     EECON1, EEPGD           ; point to Flash program memory
                BCF     EECON1, CFGS            ; access Flash program memory
                BSF     EECON1, WREN            ; enable write to memory
                BCF     INTCON, GIE             ; disable interrupts
                MOVLW   55h
    Required    MOVWF   EECON2                  ; write 55h
    Sequence    MOVLW   0AAh
                MOVWF   EECON2                  ; write 0AAh
                BSF     EECON1, WR              ; start program (CPU stall)
                DCFSZ   COUNTER2                ; repeat for remaining write blocks
                BRA     WRITE_BYTE_TO_HREGS     ;
                BSF     INTCON, GIE             ; re-enable interrupts
                BCF     EECON1, WREN            ; disable write to memory
```

4.5.2 WRITE VERIFY

Depending on the application, good programming practice may dictate that the value written to the memory should be verified against the original value. This should be used in applications where excessive writes can stress bits near the specification limit.

4.5.3 UNEXPECTED TERMINATION OF WRITE OPERATION

If a write is terminated by an unplanned event, such as loss of power or an unexpected Reset, the memory location just programmed should be verified and reprogrammed if needed. If the write operation is interrupted by a $\overline{\text{MCLR}}$ Reset or a WDT Time-out Reset during normal operation, the WRERR bit will be set which the user can check to decide whether a rewrite of the location(s) is needed.

4.5.4 PROTECTION AGAINST SPURIOUS WRITES

To protect against spurious writes to Flash program memory, the write initiate sequence must also be followed. See **Section 23.0 "Special Features of the CPU"** for more detail.

4.6 Flash Program Operation During Code Protection

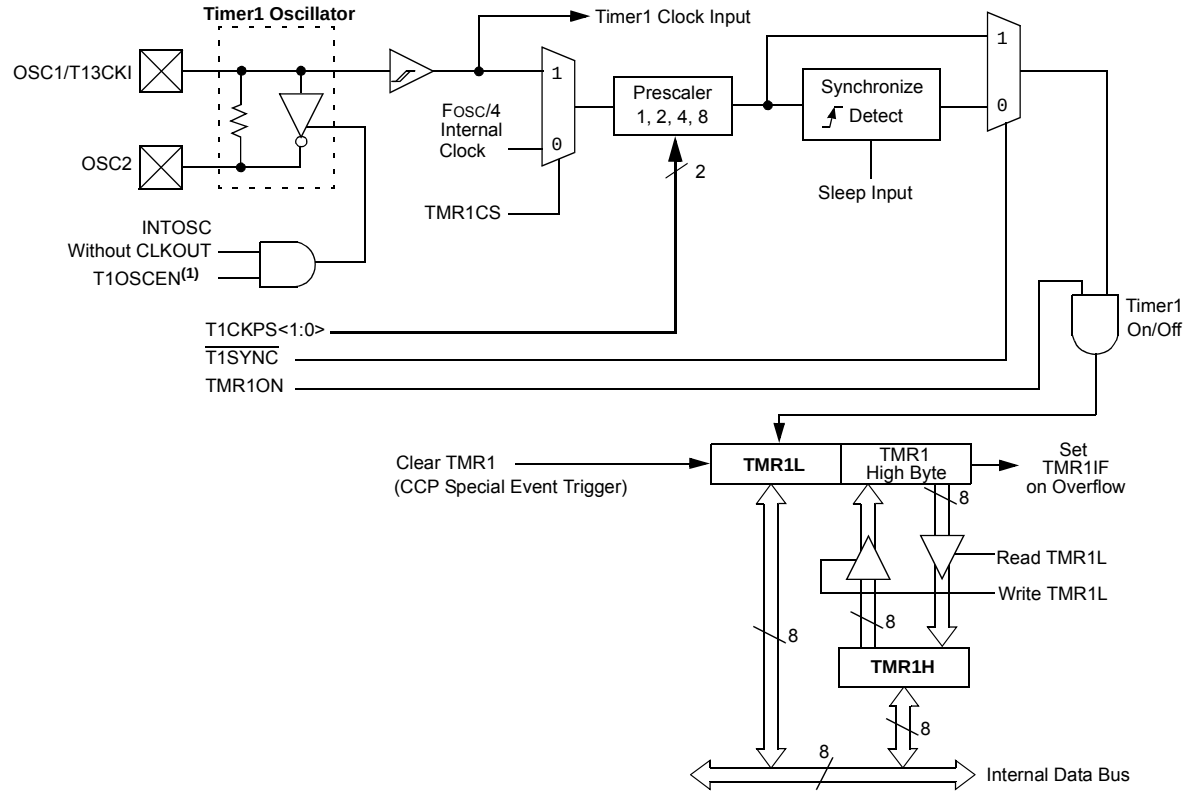
See **Section 23.3 "Program Verification and Code Protection"** for details on code protection of Flash program memory.

TABLE 4-3: REGISTERS ASSOCIATED WITH PROGRAM FLASH MEMORY

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
EECON1	EEPGD	CFGS	—	FREE	WRERR	WREN	WR	RD	247
EECON2	EEPROM Control Register 2 (not a physical register)								247
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RABIE	TMR0IF	INT0IF	RABIF	245
IPR2	OSCFIP	C1IP	C2IP	EEIP	BCLIP	—	TMR3IP	—	248
PIE2	OSCFIE	C1IE	C2IE	EEIE	BCLIE	—	TMR3IE	—	248
PIR2	OSCFIF	C1IF	C2IF	EEIF	BCLIF	—	TMR3IF	—	248
TABLAT	Program Memory Table Latch								245
TBLPTRL	Program Memory Table Pointer Low Byte (TBLPTR<7:0>)								245
TBLPTRU	—	—	bit 21	Program Memory Table Pointer Upper Byte (TBLPTR<20:16>)					245
TBPLTRH	Program Memory Table Pointer High Byte (TBLPTR<15:8>)								245

Legend: — = unimplemented, read as '0'. Shaded cells are not used during Flash/EEPROM access.

FIGURE 10-2: TIMER1 BLOCK DIAGRAM (16-BIT READ/WRITE MODE)



Note 1: When enable bit, T1OSCEN, is cleared, the inverter and feedback resistor are turned off to eliminate power drain.

11.0 TIMER2 MODULE

The Timer2 module timer incorporates the following features:

- 8-bit timer and period registers (TMR2 and PR2, respectively)
- Readable and writable (both registers)
- Software programmable prescaler (1:1, 1:4 and 1:16)
- Software programmable postscaler (1:1 through 1:16)
- Interrupt on TMR2-to-PR2 match
- Optional use as the shift clock for the MSSP module

The module is controlled through the T2CON register (Register 11-1), which enables or disables the timer and configures the prescaler and postscaler. Timer2 can be shut off by clearing control bit, TMR2ON of the T2CON register, to minimize power consumption.

A simplified block diagram of the module is shown in Figure 11-1.

11.1 Timer2 Operation

In normal operation, TMR2 is incremented from 00h on each clock (FOSC/4). A 4-bit counter/prescaler on the clock input gives direct input, divide-by-4 and divide-by-16 prescale options; these are selected by the prescaler control bits, T2CKPS<1:0> of the T2CON register. The value of TMR2 is compared to that of the period register, PR2, on each clock cycle. When the two values match, the comparator generates a match signal as the timer output. This signal also resets the value of TMR2 to 00h on the next cycle and drives the output counter/postscaler (see **Section 11.2 "Timer2 Interrupt"**).

The TMR2 and PR2 registers are both directly readable and writable. The TMR2 register is cleared on any device Reset, whereas the PR2 register initializes to FFh. Both the prescaler and postscaler counters are cleared on the following events:

- a write to the TMR2 register
- a write to the T2CON register
- any device Reset (Power-on Reset, $\overline{\text{MCLR}}$ Reset, Watchdog Timer Reset or Brown-out Reset)

TMR2 is not cleared when T2CON is written.

REGISTER 11-1: T2CON: TIMER2 CONTROL REGISTER

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0
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 **Unimplemented:** Read as '0'

bit 6-3 **T2OUTPS<3:0>**: Timer2 Output Postscale Select bits
0000 = 1:1 Postscale
0001 = 1:2 Postscale
•
•
•
1111 = 1:16 Postscale

bit 2 **TMR2ON**: Timer2 On bit
1 = Timer2 is on
0 = Timer2 is off

bit 1-0 **T2CKPS<1:0>**: Timer2 Clock Prescale Select bits
00 = Prescaler is 1
01 = Prescaler is 4
1x = Prescaler is 16

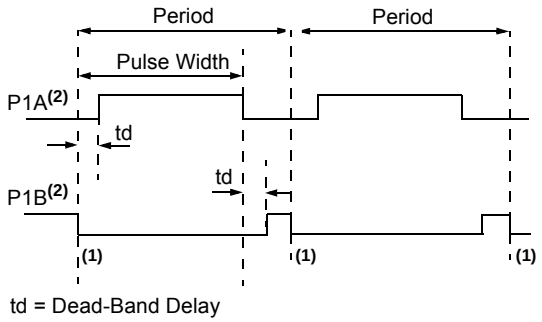
13.4.1 HALF-BRIDGE MODE

In Half-Bridge mode, two pins are used as outputs to drive push-pull loads. The PWM output signal is output on the CCP1/P1A pin, while the complementary PWM output signal is output on the P1B pin (see Figure 13-6). This mode can be used for half-bridge applications, as shown in Figure 13-7, 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 PWM1CON 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 13.4.6 "Programmable Dead-Band Delay Mode"** for more details of the dead-band delay operations.

Since the P1A and P1B outputs are multiplexed with the PORT data latches, the associated TRIS bits must be cleared to configure P1A and P1B as outputs.

FIGURE 13-6: EXAMPLE OF HALF-BRIDGE PWM OUTPUT

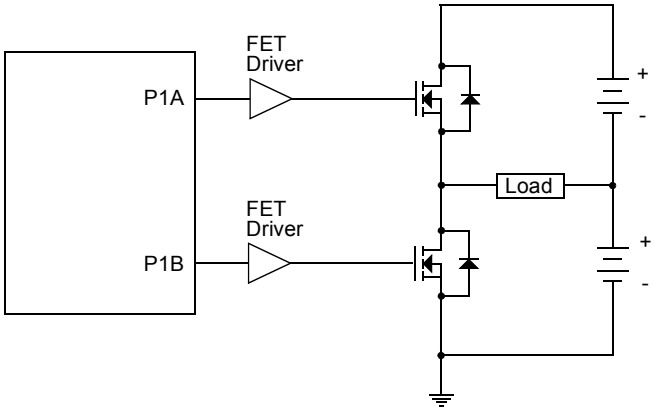


Note 1: At this time, the TMR2 register is equal to the PR2 register.

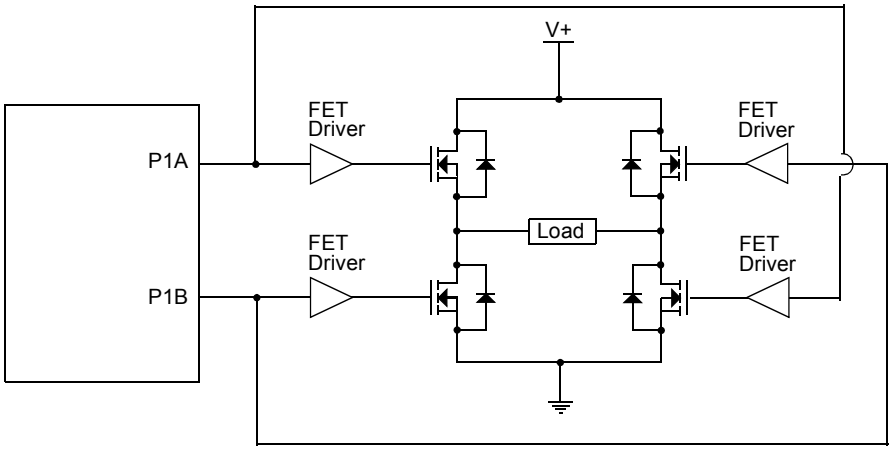
2: Output signals are shown as active-high.

FIGURE 13-7: EXAMPLE OF HALF-BRIDGE APPLICATIONS

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



Half-Bridge Output Driving a Full-Bridge Circuit



14.0 MASTER SYNCHRONOUS SERIAL PORT (MSSP) MODULE

14.1 Master SSP (MSSP) Module Overview

The Master Synchronous Serial Port (MSSP) module is a serial interface, useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be serial EEPROMs, shift registers, display drivers, A/D converters, etc. The MSSP module can operate in one of two modes:

- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I^2C)
  - Full Master mode
  - Slave mode (with general address call)

The I^2C interface supports the following modes in hardware:

- Master mode
- Multi-Master mode
- Slave mode

14.2 SPI Mode

The SPI mode allows eight bits of data to be synchronously transmitted and received simultaneously. All four modes of SPI are supported. To accomplish communication, typically three pins are used:

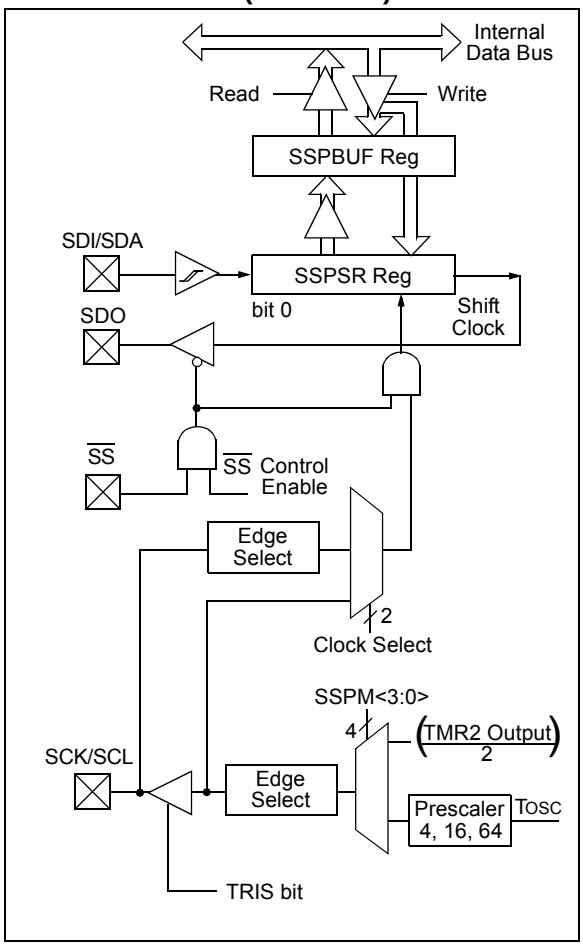
- Serial Data Out – SDO
- Serial Data In – SDI
- Serial Clock – SCK

Additionally, a fourth pin may be used when in a Slave mode of operation:

- Slave Select – $\overline{SS}$

Figure 14-1 shows the block diagram of the MSSP module when operating in SPI mode.

FIGURE 14-1: MSSP BLOCK DIAGRAM (SPI MODE)



REGISTER 14-3: SSPSTAT: MSSP STATUS REGISTER (I²C MODE)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/$\overline{A}$	P(1)	S(1)	R/$\overline{W}$(2, 3)	UA	BF
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 **SMP:** Slew Rate Control bit
In Master or Slave mode:
1 = Slew rate control disabled for standard Speed mode (100 kHz and 1 MHz)
0 = Slew rate control enabled for High-Speed mode (400 kHz)

bit 6 **CKE:** SMBus Select bit
In Master or Slave mode:
1 = Enable SMBus specific inputs
0 = Disable SMBus specific inputs

bit 5 **D/$\overline{A}$:** Data/$\overline{Address}$ bit
In Master mode:
Reserved.
In Slave mode:
1 = Indicates that the last byte received or transmitted was data
0 = Indicates that the last byte received was an address

bit 4 **P:** Stop bit(1)
1 = Indicates that a Stop bit has been detected last
0 = Stop bit was not detected last

bit 3 **S:** Start bit(1)
1 = Indicates that a Start bit has been detected last
0 = Start bit was not detected last

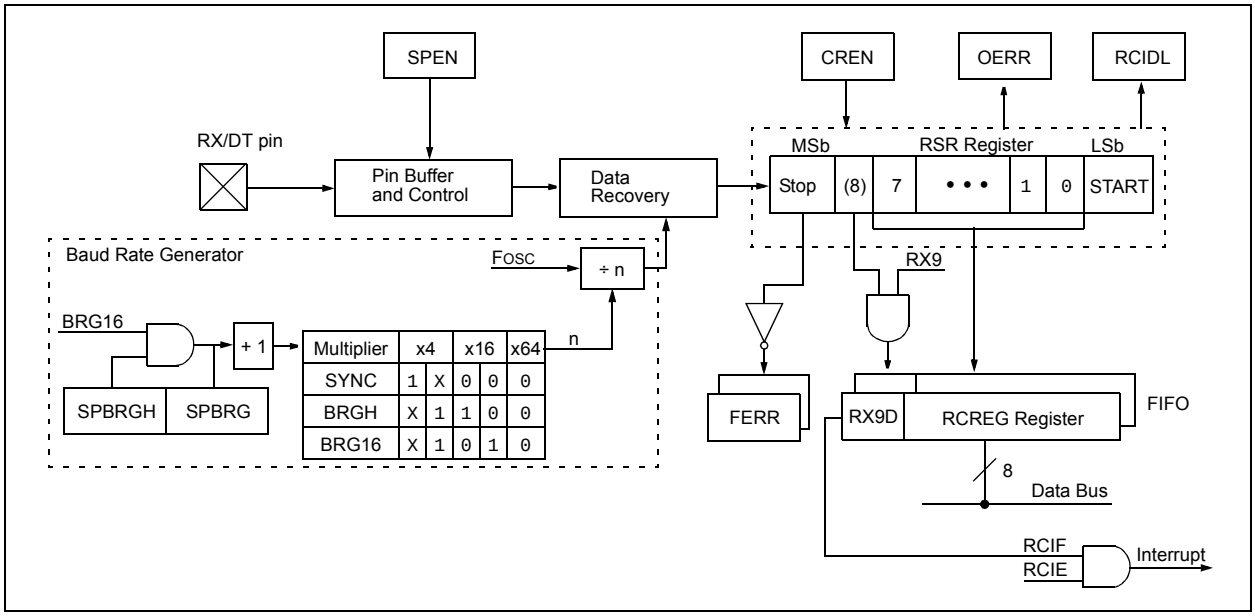
bit 2 **R/$\overline{W}$:** Read/$\overline{Write}$ Information bit (I²C mode only)(2, 3)
In Slave mode:
1 = Read
0 = Write
In Master mode:
1 = Transmit is in progress
0 = Transmit is not in progress

bit 1 **UA:** Update Address bit (10-bit Slave mode only)
1 = Indicates that the user needs to update the address in the SSPADD register
0 = Address does not need to be updated

bit 0 **BF:** Buffer Full Status bit
In Transmit mode:
1 = SSPBUF is full
0 = SSPBUF is empty
In Receive mode:
1 = SSPBUF is full (does not include the $\overline{ACK}$ and Stop bits)
0 = SSPBUF is empty (does not include the $\overline{ACK}$ and Stop bits)

Note 1: This bit is cleared on Reset and when SSPEN is cleared.
2: This bit holds the R/$\overline{W}$ bit information following the last address match. This bit is only valid from the address match to the next Start bit, Stop bit or not $\overline{ACK}$ bit.
3: ORing this bit with SEN, RSEN, PEN, RCEN or ACKEN will indicate if the Master mode is active.

FIGURE 15-2: EUSART RECEIVE BLOCK DIAGRAM



The operation of the EUSART module is controlled through three registers:

- Transmit Status and Control (TXSTA)
- Receive Status and Control (RCSTA)
- Baud Rate Control (BAUDCTL)

These registers are detailed in Register 15-1, Register 15-2 and Register 15-3, respectively.

For all modes of EUSART operation, the TRIS control bits corresponding to the RX/DT and TX/CK pins should be set to '1'. The EUSART control will automatically reconfigure the pin from input to output, as needed.

20.3 Register Definitions: FVR Control

REGISTER 20-1: VREFCON0: FIXED VOLTAGE REFERENCE CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-1	U-0	U-0	U-0	U-0
FVR1EN	FVR1ST	FVR1S<1:0>		—	—	—	—
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 **FVR1EN:** Fixed Voltage Reference Enable bit
0 = Fixed Voltage Reference is disabled
1 = Fixed Voltage Reference is enabled

bit 6 **FVR1ST:** Fixed Voltage Reference Ready Flag bit
0 = Fixed Voltage Reference output is not ready or not enabled
1 = Fixed Voltage Reference output is ready for use

bit 5-4 **FVR1S<1:0>:** Fixed Voltage Reference Selection bits
00 = Fixed Voltage Reference Peripheral output is off
01 = Fixed Voltage Reference Peripheral output is 1x (1.024V)
10 = Fixed Voltage Reference Peripheral output is 2x (2.048V)**(1)**
11 = Fixed Voltage Reference Peripheral output is 4x (4.096V)**(1)**

bit 3-2 **Reserved:** Read as '0'. Maintain these bits clear.

bit 1-0 **Unimplemented:** Read as '0'.

Note 1: Fixed Voltage Reference output cannot exceed V_{DD}.

TABLE 20-1: SUMMARY OF REGISTERS ASSOCIATED WITH FIXED VOLTAGE REFERENCE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page
VREFCON0	FVR1EN	FVR1ST	FVR1S<1:0>		—	—	—	—	232

Legend: — = unimplemented locations, read as '0'. Shaded bits are not used by the FVR module.

REGISTER 21-2: VREFCON2: VOLTAGE REFERENCE CONTROL REGISTER 1

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	DAC1R<4:0>				
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 7-5 **Unimplemented:** Read as '0'

bit 4-0 **DAC1R<4:0>:** DAC Voltage Output Select bits

$V_{OUT} = ((V_{SRC}+) - (V_{SRC}-))*(DAC1R<4:0>/(2^5)) + V_{SRC}-$

TABLE 21-1: REGISTERS ASSOCIATED WITH DAC MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page
VREFCON0	FVR1EN	FVR1ST	FVR1S<1:0>		—	—	—	—	232
VREFCON1	D1EN	D1LPS	DAC1OE	—	D1PSS<1:0>		—	D1NSS	235
VREFCON2	—	—	—	DAC1R<4:0>					236

Legend: — = Unimplemented locations, read as '0'. Shaded bits are not used by the DAC module.

24.0 INSTRUCTION SET SUMMARY

PIC18(L)F1XK22 devices incorporate the standard set of 75 PIC18 core instructions, as well as an extended set of eight new instructions, for the optimization of code that is recursive or that utilizes a software stack. The extended set is discussed later in this section.

24.1 Standard Instruction Set

The standard PIC18 instruction set adds many enhancements to the previous PIC® MCU instruction sets, while maintaining an easy migration from these PIC MCU instruction sets. Most instructions are a single program memory word (16 bits), but there are four instructions that require two program memory locations.

Each single-word instruction is a 16-bit word divided into an opcode, which specifies the instruction type and one or more operands, which further specify the operation of the instruction.

The instruction set is highly orthogonal and is grouped into four basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal** operations
- **Control** operations

The PIC18 instruction set summary in Table 24-2 lists **byte-oriented**, **bit-oriented**, **literal** and **control** operations. Table 24-1 shows the opcode field descriptions.

Most **byte-oriented** instructions have three operands:

1. The file register (specified by ‘f’)
2. The destination of the result (specified by ‘d’)
3. The accessed memory (specified by ‘a’)

The file register designator ‘f’ specifies which file register is to be used by the instruction. The destination designator ‘d’ specifies where the result of the operation is to be placed. If ‘d’ is zero, the result is placed in the WREG register. If ‘d’ is one, the result is placed in the file register specified in the instruction.

All **bit-oriented** instructions have three operands:

1. The file register (specified by ‘f’)
2. The bit in the file register (specified by ‘b’)
3. The accessed memory (specified by ‘a’)

The bit field designator ‘b’ selects the number of the bit affected by the operation, while the file register designator ‘f’ represents the number of the file in which the bit is located.

The **literal** instructions may use some of the following operands:

- A literal value to be loaded into a file register (specified by ‘k’)
- The desired FSR register to load the literal value into (specified by ‘f’)
- No operand required (specified by ‘—’)

The **control** instructions may use some of the following operands:

- A program memory address (specified by ‘n’)
- The mode of the CALL or RETURN instructions (specified by ‘s’)
- The mode of the table read and table write instructions (specified by ‘m’)
- No operand required (specified by ‘—’)

All instructions are a single word, except for four double-word instructions. These instructions were made double-word to contain the required information in 32 bits. In the second word, the 4 MSbs are ‘1’s. If this second word is executed as an instruction (by itself), it will execute as a NOP.

All single-word instructions are executed in a single instruction cycle, unless a conditional test is true or the program counter is changed as a result of the instruction. In these cases, the execution takes two instruction cycles, with the additional instruction cycle(s) executed as a NOP.

The double-word instructions execute in two instruction cycles.

One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1μs. If a conditional test is true, or the program counter is changed as a result of an instruction, the instruction execution time is 2μs. Two-word branch instructions (if true) would take 3μs.

Figure 24-1 shows the general formats that the instructions can have. All examples use the convention ‘nnh’ to represent a hexadecimal number.

The Instruction Set Summary, shown in Table 24-2, lists the standard instructions recognized by the Microchip Assembler (MPASM™).

Section 24.1.1 “Standard Instruction Set” provides a description of each instruction.

ADDWFC **ADD W and CARRY bit to f**

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

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

Operation: (W) + (f) + (C) → dest

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

Encoding:

0010	00da	ffff	ffff

Description: Add W, the CARRY flag and data memory location 'f'. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed in data memory location 'f'.
If 'a' is '0', the Access Bank is selected.
If 'a' is '1', the BSR is used to select the GPR bank (default).
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 24.2.3 "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: `ADDWFC REG, 0, 1`

Before Instruction
CARRY bit = 1
REG = 02h
W = 4Dh
After Instruction
CARRY bit = 0
REG = 02h
W = 50h

ANDLW **AND literal with W**

Syntax: ANDLW k

Operands: $0 \le k \le 255$

Operation: (W) .AND. k → W

Status Affected: N, Z

Encoding:

0000	1011	kkkk	kkkk

Description: The contents of W are AND'ed with the 8-bit literal 'k'. 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: `ANDLW 05Fh`

Before Instruction
W = A3h
After Instruction
W = 03h

SUBLW **Subtract W from literal**

Syntax: SUBLW k

Operands: $0 \le k \le 255$

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

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

Encoding:

0000	1000	kkkk	kkkk

Description: W is subtracted from the 8-bit literal 'k'. 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 1:
```
SUBLW   02h
Before Instruction
    W   =   01h
    C   =   ?
After Instruction
    W   =   01h
    C   =   1   ; result is positive
    Z   =   0
    N   =   0
```

Example 2:
```
SUBLW   02h
Before Instruction
    W   =   02h
    C   =   ?
After Instruction
    W   =   00h
    C   =   1   ; result is zero
    Z   =   1
    N   =   0
```

Example 3:
```
SUBLW   02h
Before Instruction
    W   =   03h
    C   =   ?
After Instruction
    W   =   FFh ; (2's complement)
    C   =   0   ; result is negative
    Z   =   0
    N   =   1
```

SUBWF **Subtract W from f**

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

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

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

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

Encoding:

0101	11da	ffff	ffff

Description: Subtract W from register 'f' (2's complement method). 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 (default).
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 24.2.3 "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 1:
```
SUBWF   REG, 1, 0
Before Instruction
    REG   =   3
    W     =   2
    C     =   ?
After Instruction
    REG   =   1
    W     =   2
    C     =   1   ; result is positive
    Z     =   0
    N     =   0
```

Example 2:
```
SUBWF   REG, 0, 0
Before Instruction
    REG   =   2
    W     =   2
    C     =   ?
After Instruction
    REG   =   2
    W     =   0
    C     =   1   ; result is zero
    Z     =   1
    N     =   0
```

Example 3:
```
SUBWF   REG, 1, 0
Before Instruction
    REG   =   1
    W     =   2
    C     =   ?
After Instruction
    REG   =   FFh ;(2's complement)
    W     =   2
    C     =   0   ; result is negative
    Z     =   0
    N     =   1
```

ADDWF **ADD W to Indexed (Indexed Literal Offset mode)**

Syntax: ADDWF [k] {,d}

Operands: $0 \le k \le 95$
$d \in [0,1]$

Operation: (W) + ((FSR2) + k) → dest

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

Encoding:

0010	01d0	kkkk	kkkk

Description: The contents of W are added to the contents of the register indicated by FSR2, offset by the value 'k'.
If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default).

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: ADDWF [OFST], 0

Before Instruction
W = 17h
OFST = 2Ch
FSR2 = 0A00h
Contents of 0A2Ch = 20h
After Instruction
W = 37h
Contents of 0A2Ch = 20h

BSF **Bit Set Indexed (Indexed Literal Offset mode)**

Syntax: BSF [k], b

Operands: $0 \le f \le 95$
$0 \le b \le 7$

Operation: 1 → ((FSR2) + k)<b>

Status Affected: None

Encoding:

1000	bbb0	kkkk	kkkk

Description: Bit 'b' of the register indicated by FSR2, offset by the value 'k', is set.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

Example: BSF [FLAG_OFST], 7

Before Instruction
FLAG_OFST = 0Ah
FSR2 = 0A00h
Contents of 0A0Ah = 55h
After Instruction
Contents of 0A0Ah = D5h

SETF **Set Indexed (Indexed Literal Offset mode)**

Syntax: SETF [k]

Operands: $0 \le k \le 95$

Operation: FFh → ((FSR2) + k)

Status Affected: None

Encoding:

0110	1000	kkkk	kkkk

Description: The contents of the register indicated by FSR2, offset by 'k', are set to FFh.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read 'k'	Process Data	Write register

Example: SETF [OFST]

Before Instruction
OFST = 2Ch
FSR2 = 0A00h
Contents of 0A2Ch = 00h
After Instruction
Contents of 0A2Ch = FFh

TABLE 26-4: PRIMARY RUN SUPPLY CURRENT

PIC18LF1XK22		Standard Operating Conditions (unless otherwise stated)					
PIC18F1XK22		**Standard Operating Conditions (unless otherwise stated)**					
Param. No.	**Device Characteristics**	**Typ.**	**Max.**	**Units**	**Conditions**		
D014	**Supply Current (I_{DD})(1, 2, 4, 5)**	0.20	0.32	mA	-40°C to +125°C	V_{DD} = 1.8V	F_{OSC} = 1 MHz (**PRI_RUN**, EC Med Osc)
D014A		0.27	0.39	mA	-40°C to +125°C	V_{DD} = 3.0V	
D014		.20	.32	mA	-40°C to +125°C	V_{DD} = 2.3V	F_{OSC} = 1 MHz (**PRI_RUN**, EC Med Osc)
D014A		.27	.39	mA	-40°C to +125°C	V_{DD} = 3.0V	
D014B		.30	.42	mA	-40°C to +125°C	V_{DD} = 5.0V	
D015		1.7	2.6	mA	-40°C to +125°C	V_{DD} = 1.8V	F_{OSC} = 16 MHz (**PRI_RUN**, EC High Osc)
D015A		3.0	4.2	mA	-40°C to +125°C	V_{DD} = 3.0V	
D015		2.4	3.2	mA	-40°C to +125°C	V_{DD} = 2.3V	F_{OSC} = 16 MHz (**PRI_RUN**, EC High Osc)
D015A		3.0	4.2	mA	-40°C to +125°C	V_{DD} = 3.0V	
D015B		3.3	4.4	mA	-40°C to +125°C	V_{DD} = 5.0V	
D016		11.5	14.0	mA	-40°C to +125°C	V_{DD} = 3.0V	F_{OSC} = 64 MHz (**PRI_RUN**, EC High Osc)
D016		11.9	14.4	mA	-40°C to +125°C	V_{DD} = 2.3V	F_{OSC} = 64 MHz (**PRI_RUN**, EC High Osc)
D016A		12.1	14.6	mA	-40°C to +125°C	V_{DD} = 5.0V	
D017		2.1	2.9	mA	-40°C to +125°C	V_{DD} = 1.8V	F_{OSC} = 4 MHz 16 MHz Internal (**PRI_RUN HS+PLL**)
D017A		3.1	4.0	mA	-40°C to +125°C	V_{DD} = 3.0V	
D017		2.1	2.9	mA	-40°C to +125°C	V_{DD} = 2.3V	F_{OSC} = 4 MHz 16 MHz Internal (**PRI_RUN HS+PLL**)
D017A		3.1	4.0	mA	-40°C to +125°C	V_{DD} = 3.0V	
D017B		3.3	4.5	mA	-40°C to +125°C	V_{DD} = 5.0V	
D018		10	15	mA	-40°C to +125°C	V_{DD} = 3.0V	F_{OSC} = 16 MHz 64 MHz Internal (**PRI_RUN HS+PLL**)
D018		12.4	15.4	mA	-40°C to +125°C	V_{DD} = 3.0V	F_{OSC} = 16 MHz 64 MHz Internal (**PRI_RUN HS+PLL**)
D018A		12.6	15.6	mA	-40°C to +125°C	V_{DD} = 5.0V	

* These parameters are characterized but not tested.

Note 1: The test conditions for all I_{DD} measurements in active operation mode are: OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to V_{DD}; $\overline{MCLR}$ = V_{DD}; WDT disabled.

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

3: For RC oscillator configurations, current through R_{EXT} is not included. The current through the resistor can be extended by the formula $I_R = V_{DD}/2R_{EXT}$ (mA) with R_{EXT} in kΩ.

4: FVR and BOR are disabled.

5: When a single temperature range is provided for a parameter, the specification applies to both industrial and extended temperature devices.

TABLE 26-7: SECONDARY IDLE SUPPLY CURRENT

PIC18LF1XK22		Standard Operating Conditions (unless otherwise stated)					
PIC18F1XK22		**Standard Operating Conditions (unless otherwise stated)**					
Param. No.	**Device Characteristics**	**Typ.**	**Max.**	**Units**	**Conditions**		
D023	**Supply Current (I_{DD})(1, 2, 4)**	2	5	µA	-40°C	V_{DD} = 1.8V	F_{OSC} = 32 kHz(3) (**SEC_IDLE** mode, Timer1 as clock)
		2	5	µA	+25°C		
		3	9	µA	+85°C		
		8	11	µA	+125°C		
D023A		4	8	µA	-40°C	V_{DD} = 3.0V	
		5	10	µA	+25°C		
		9	20	µA	+85°C		
		20	23	µA	+125°C		
D023		20	40	µA	-40°C	V_{DD} = 2.3V	F_{OSC} = 32 kHz(3) (**SEC_IDLE** mode, Timer1 as clock)
		21	41	µA	+25°C		
		23	44	µA	+85°C		
		24	47	µA	+125°C		
D023A		23	45	µA	-40°C	V_{DD} = 3.0V	
		25	47	µA	+25°C		
		28	49	µA	+85°C		
		30	52	µA	+125°C		
D023B		28	50	µA	-40°C	V_{DD} = 5.0V	
		30	54	µA	+25°C		
		32	59	µA	+85°C		
		33	62	µA	+125°C		

* These parameters are characterized but not tested.

Note 1: The test conditions for all I_{DD} measurements in active operation mode are: OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to V_{DD}; $\overline{MCLR}$ = V_{DD}; WDT disabled.

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

3: For RC oscillator configurations, current through R_{EXT} is not included. The current through the resistor can be extended by the formula $I_R = V_{DD}/2R_{EXT}$ (mA) with R_{EXT} in kΩ.

4: FVR and BOR are disabled.

26.4 Timing Parameter Symbology

The timing parameter symbols have been created with one of the following formats:

1. TppS2ppS
2. TppS

T			
F	Frequency	T	Time

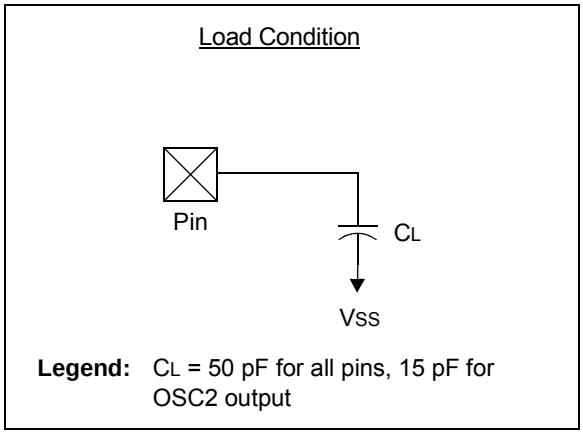
Lowercase letters (pp) and their meanings:

pp			
cc	CCP1	osc	OSC1
ck	CLKOUT	rd	$\overline{RD}$
cs	$\overline{CS}$	rw	$\overline{RD}$ or $\overline{WR}$
di	SDI	sc	SCK
do	SDO	ss	$\overline{SS}$
dt	Data in	t0	T0CKI
io	I/O PORT	t1	T1CKI
mc	$\overline{MCLR}$	wr	$\overline{WR}$

Uppercase letters and their meanings:

S			
F	Fall	P	Period
H	High	R	Rise
I	Invalid (High-impedance)	V	Valid
L	Low	Z	High-impedance

FIGURE 26-7: LOAD CONDITIONS



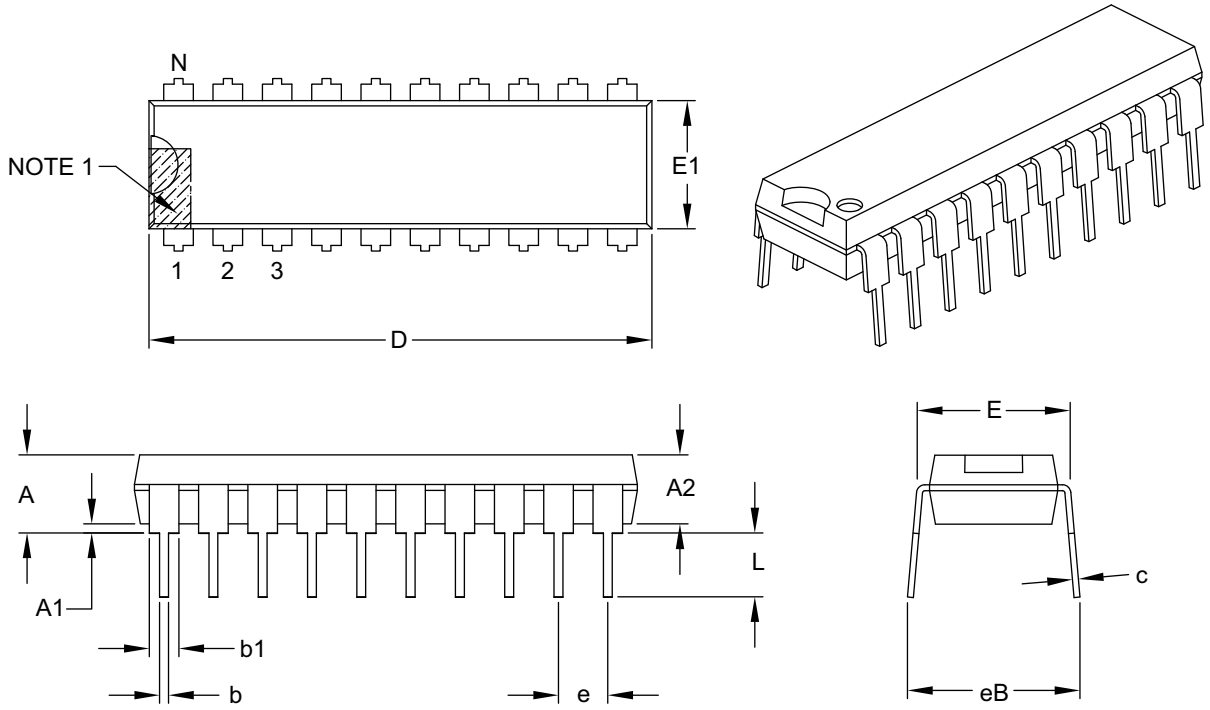
Legend: CL = 50 pF for all pins, 15 pF for OSC2 output

28.2 Package Details

The following sections give the technical details of the packages.

20-Lead Plastic Dual In-Line (P) – 300 mil Body [PDIP]

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



Units		INCHES		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		20	
Pitch	e		.100 BSC	
Top to Seating Plane	A	–	–	.210
Molded Package Thickness	A2	.115	.130	.195
Base to Seating Plane	A1	.015	–	–
Shoulder to Shoulder Width	E	.300	.310	.325
Molded Package Width	E1	.240	.250	.280
Overall Length	D	.980	1.030	1.060
Tip to Seating Plane	L	.115	.130	.150
Lead Thickness	c	.008	.010	.015
Upper Lead Width	b1	.045	.060	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	–	–	.430

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

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

Microchip Technology Drawing C04-019B