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, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	17
Program Memory Size	8KB (4K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SOIC (0.295", 7.50mm Width)
Supplier Device Package	20-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf13k22t-i-so

1.0 DEVICE OVERVIEW

This family offers the advantages of all PIC18 microcontrollers – namely, high computational performance with the addition of high-endurance, Flash program memory. On top of these features, the PIC18(L)F1XK22 family introduces design enhancements that make these microcontrollers a logical choice for many high-performance, power sensitive applications.

1.1 New Core Features

1.1.1 XLP TECHNOLOGY

All of the devices in the PIC18(L)F1XK22 family incorporate a range of features that can significantly reduce power consumption during operation. Key items include:

- **Multiple Idle Modes:** The controller can also run with its CPU core disabled but the peripherals still active. In these states, power consumption can be reduced even further, to as little as 4% of normal operation requirements.
- **On-the-fly Mode Switching:** The power-managed modes are invoked by user code during operation, allowing the user to incorporate power-saving ideas into their application's software design.
- **Low Consumption in Key Modules:** The power requirements for both Timer1 and the Watchdog Timer are minimized. See **Section 26.0 "Electrical Specifications"** for values.

1.1.2 MULTIPLE OSCILLATOR OPTIONS AND FEATURES

All of the devices in the PIC18(L)F1XK22 family offer ten different oscillator options, allowing users a wide range of choices in developing application hardware. These include:

- Four Crystal modes, using crystals or ceramic resonators
- External Clock modes, offering the option of using two pins (oscillator input and a divide-by-4 clock output) or one pin (oscillator input, with the second pin reassigned as general I/O)
- External RC Oscillator modes with the same pin options as the External Clock modes
- An internal oscillator block which contains a 16 MHz HFINTOSC oscillator and a 31 kHz LFINTOSC oscillator which together provide eight user selectable clock frequencies, from 31 kHz to 16 MHz. This option frees the two oscillator pins for use as additional general purpose I/O.
- A Phase Lock Loop (PLL) frequency multiplier, available to both the high-speed crystal and internal oscillator modes, which allows clock speeds of up to 64 MHz. Used with the internal oscillator, the PLL gives users a complete selection of clock speeds, from 31 kHz to 64 MHz – all without using an external crystal or clock circuit.

Besides its availability as a clock source, the internal oscillator block provides a stable reference source that gives the family additional features for robust operation:

- **Fail-Safe Clock Monitor:** This option constantly monitors the main clock source against a reference signal provided by the LFINTOSC. If a clock failure occurs, the controller is switched to the internal oscillator block, allowing for continued operation or a safe application shutdown.
- **Two-Speed Start-up:** This option allows the internal oscillator to serve as the clock source from Power-on Reset, or wake-up from Sleep mode, until the primary clock source is available.

4.0 FLASH PROGRAM MEMORY

The Flash program memory is readable, writable and erasable during normal operation over the entire VDD range.

A read from program memory is executed one byte at a time. A write to program memory is executed on blocks of 16 or 8 bytes at a time depending on the specific device (See Table 4-1). Program memory is erased in blocks of 64 bytes at a time. The difference between the write and erase block sizes requires from 4 to 8 block writes to restore the contents of a single block erase. A Bulk Erase operation can not be issued from user code.

TABLE 4-1: WRITE/ERASE BLOCK SIZES

Device	Write Block Size (bytes)	Erase Block Size (bytes)
PIC18(L)F13K22	8	64
PIC18(L)F14K22	16	64

Writing or erasing program memory will cease instruction fetches until the operation is complete. The program memory cannot be accessed during the write or erase, therefore, code cannot execute. An internal programming timer terminates program memory writes and erases.

A value written to program memory does not need to be a valid instruction. Executing a program memory location that forms an invalid instruction results in a NOP.

4.1 Table Reads and Table Writes

In order to read and write program memory, there are two operations that allow the processor to move bytes between the program memory space and the data RAM:

- Table Read (TBLRD)
- Table Write (TBLWT)

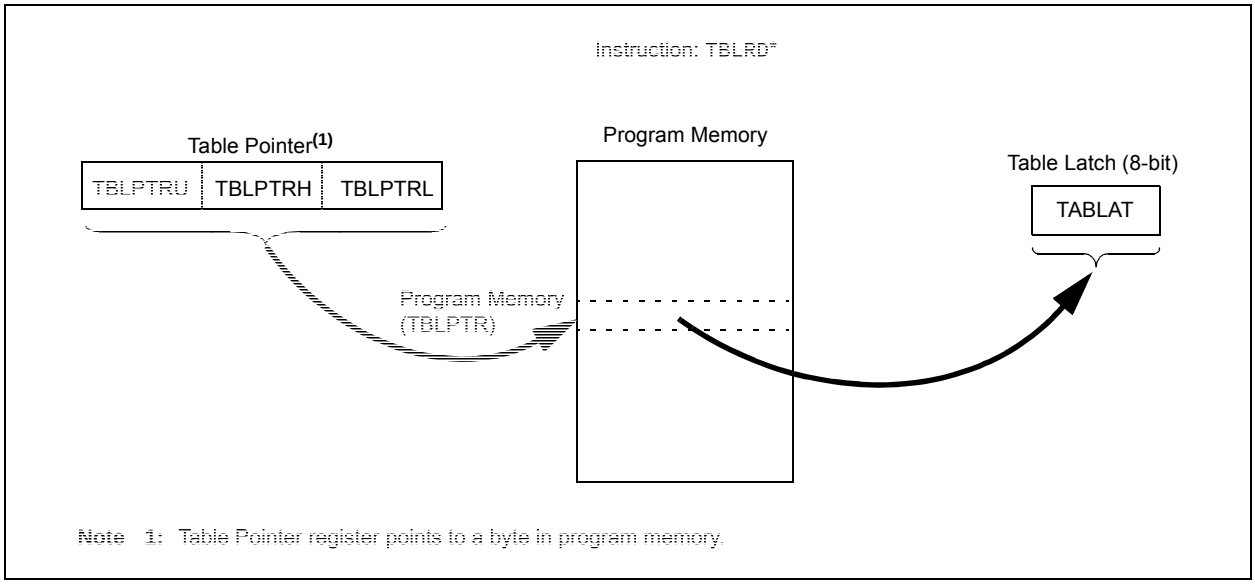
The program memory space is 16-bit wide, while the data RAM space is 8-bit wide. Table reads and table writes move data between these two memory spaces through an 8-bit register (TABLAT).

The table read operation retrieves one byte of data directly from program memory and places it into the TABLAT register. Figure 4-1 shows the operation of a table read.

The table write operation stores one byte of data from the TABLAT register into a write block holding register. The procedure to write the contents of the holding registers into program memory is detailed in **Section 4.5 "Writing to Flash Program Memory"**. Figure 4-2 shows the operation of a table write with program memory and data RAM.

Table operations work with byte entities. Tables containing data, rather than program instructions, are not required to be word-aligned. Therefore, a table can start and end at any byte address. If a table write is being used to write executable code into program memory, program instructions will need to be word-aligned.

FIGURE 4-1: TABLE READ OPERATION



REGISTER 7-3: INTCON3: INTERRUPT CONTROL 3 REGISTER

R/W-1	R/W-1	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
INT2IP	INT1IP	—	INT2IE	INT1IE	—	INT2IF	INT1IF
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 **INT2IP:** INT2 External Interrupt Priority bit
1 = High priority
0 = Low priority

bit 6 **INT1IP:** INT1 External Interrupt Priority bit
1 = High priority
0 = Low priority

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

bit 4 **INT2IE:** INT2 External Interrupt Enable bit
1 = Enables the INT2 external interrupt
0 = Disables the INT2 external interrupt

bit 3 **INT1IE:** INT1 External Interrupt Enable bit
1 = Enables the INT1 external interrupt
0 = Disables the INT1 external interrupt

bit 2 **Unimplemented:** Read as '0'

bit 1 **INT2IF:** INT2 External Interrupt Flag bit
1 = The INT2 external interrupt occurred (must be cleared by software)
0 = The INT2 external interrupt did not occur

bit 0 **INT1IF:** INT1 External Interrupt Flag bit
1 = The INT1 external interrupt occurred (must be cleared by software)
0 = The INT1 external interrupt did not occur

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 might ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. This feature allows for software polling.

REGISTER 7-9: IPR2: PERIPHERAL INTERRUPT PRIORITY REGISTER 2

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	U-0	R/W-1	U-0
OSCFIP	C1IP	C2IP	EEIP	BCLIP	—	TMR3IP	—
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 **OSCFIP:** Oscillator Fail Interrupt Priority bit
1 = High priority
0 = Low priority

bit 6 **C1IP:** Comparator C1 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 5 **C2IP:** Comparator C2 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 4 **EEIP:** Data EEPROM/Flash Write Operation Interrupt Priority bit
1 = High priority
0 = Low priority

bit 3 **BCLIP:** Bus Collision Interrupt Priority bit
1 = High priority
0 = Low priority

bit 2 **Unimplemented:** Read as '0'

bit 1 **TMR3IP:** TMR3 Overflow Interrupt Priority bit
1 = High priority
0 = Low priority

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

8.3 PORTC, TRISC and LATC 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 PORTC 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.

All the pins on PORTC are implemented with Schmitt Trigger input buffer. Each pin is individually configurable as an input or output.

Note:	On a Power-on Reset, RC<7:6> and RC<3:0> are configured as analog inputs and read as '0'.

EXAMPLE 8-3: INITIALIZING PORTC

```
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
```

REGISTER 8-11: PORTC: PORTC REGISTER

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0
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 **RC<7:0>**: PORTC I/O Pin bits
1 = Port pin is > V_{IH}
0 = Port pin is < V_{IL}

REGISTER 8-12: TRISC: PORTC TRI-STATE 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
TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0
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 **TRISC<7:0>:** PORTC Tri-State Control bits
1 = PORTC pin configured as an input (tri-stated)
0 = PORTC pin configured as an output

8.4 Port Analog Control

Some port pins are multiplexed with analog functions such as the Analog-to-Digital Converter and comparators. When these I/O pins are to be used as analog inputs it is necessary to disable the digital input buffer to avoid excessive current caused by improper biasing of the digital input. Individual control of the digital input buffers on pins which share analog functions is provided by the ANSEL and ANSELH registers. Setting an ANSx bit high will disable the associated digital input buffer and cause all reads of that pin to return '0' while allowing analog functions of that pin to operate correctly.

The state of the ANSx bits has no affect on digital output functions. A pin with the associated TRISx bit clear and ANSx bit set will still operate as a digital output but the Input mode will be analog.

REGISTER 8-14: ANSEL: 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
ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0
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 **ANS7:** RC3 Analog Select Control bit
1 = Digital input buffer of RC3 is disabled
0 = Digital input buffer of RC3 is enabled

bit 6 **ANS6:** RC2 Analog Select Control bit
1 = Digital input buffer of RC2 is disabled
0 = Digital input buffer of RC2 is enabled

bit 5 **ANS5:** RC1 Analog Select Control bit
1 = Digital input buffer of RC1 is disabled
0 = Digital input buffer of RC1 is enabled

bit 4 **ANS4:** RC0 Analog Select Control bit
1 = Digital input buffer of RC0 is disabled
0 = Digital input buffer of RC0 is enabled

bit 3 **ANS3:** RA4 Analog Select Control bit
1 = Digital input buffer of RA4 is disabled
0 = Digital input buffer of RA4 is enabled

bit 2 **ANS2:** RA2 Analog Select Control bit
1 = Digital input buffer of RA2 is disabled
0 = Digital input buffer of RA2 is enabled

bit 1 **ANS1:** RA1 Analog Select Control bit
1 = Digital input buffer of RA1 is disabled
0 = Digital input buffer of RA1 is enabled

bit 0 **ANS0:** RA0 Analog Select Control bit
1 = Digital input buffer of RA0 is disabled
0 = Digital input buffer of RA0 is enabled

10.1 Timer1 Operation

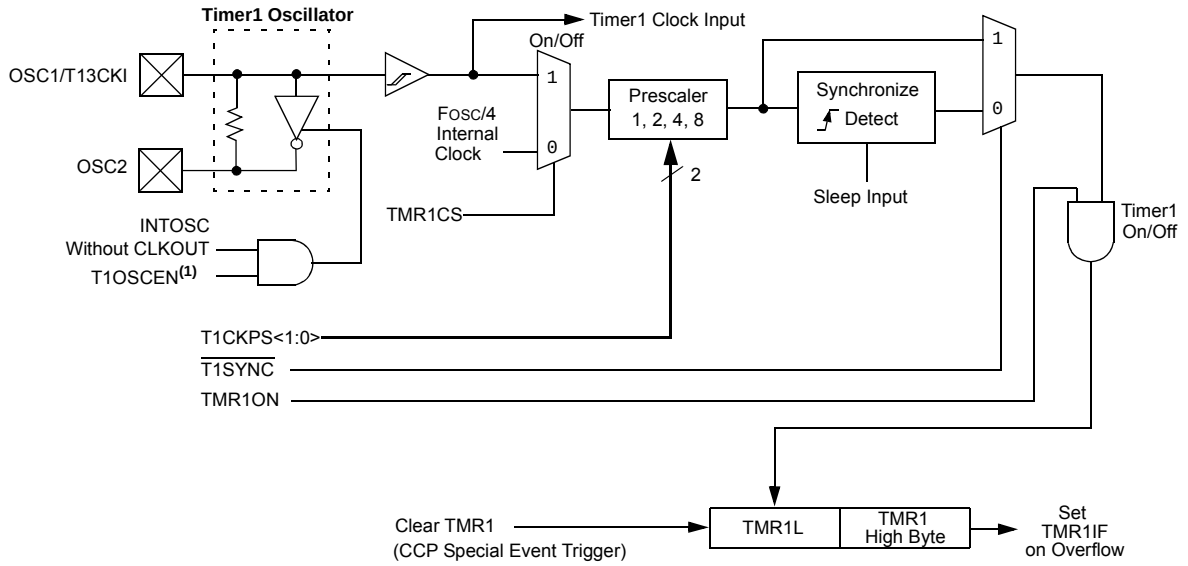
Timer1 can operate in one of the following modes:

- Timer
- Synchronous Counter
- Asynchronous Counter

The operating mode is determined by the clock select bit, TMR1CS of the T1CON register. When TMR1CS is cleared (= 0), Timer1 increments on every internal instruction cycle (FOSC/4). When the bit is set, Timer1 increments on every rising edge of either the Timer1 external clock input or the Timer1 oscillator, if enabled.

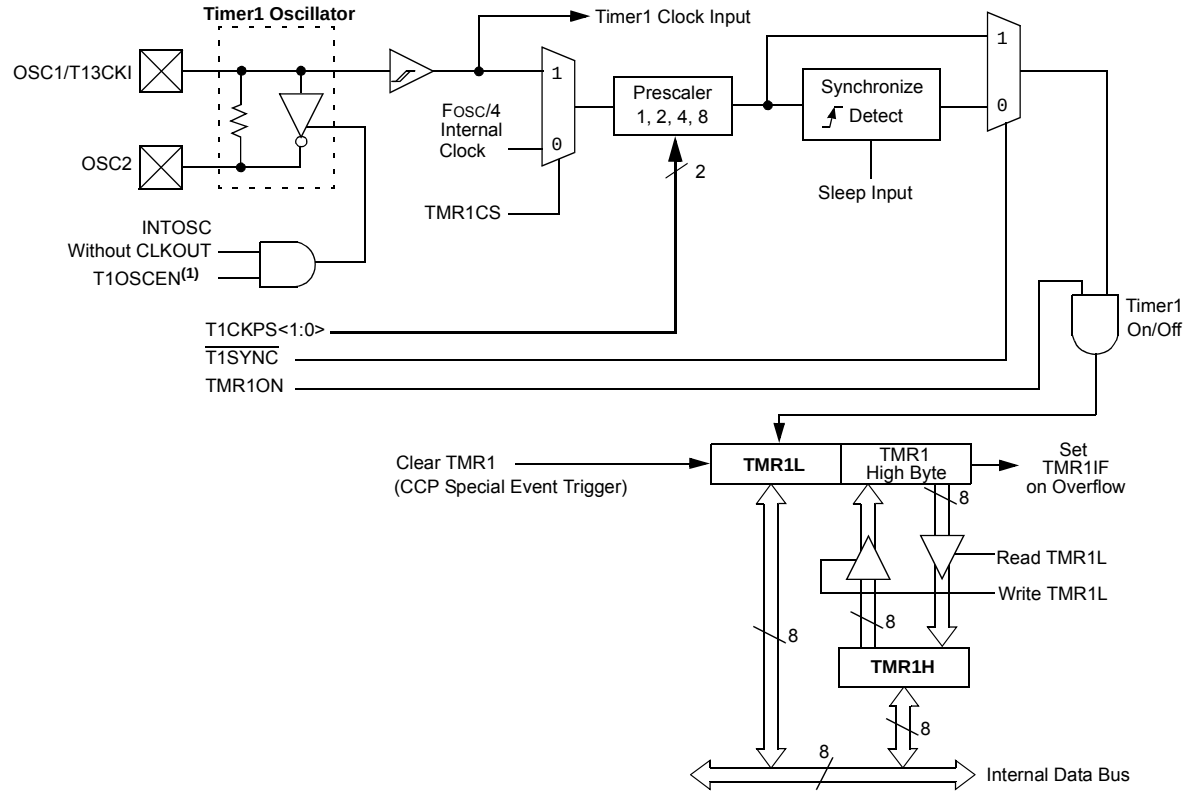
When the Timer1 oscillator is enabled, the digital circuitry associated with the OSC1 and OSC2 pins is disabled. This means the values of TRISA<5:4> are ignored and the pins are read as '0'.

FIGURE 10-1: TIMER1 BLOCK DIAGRAM



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

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.

14.2.6 SLAVE MODE

In Slave mode, the data is transmitted and received as external clock pulses appear on SCK. When the last bit is latched, the SSPIF interrupt flag bit is set.

Before enabling the module in SPI Slave mode, the clock line must match the proper Idle state. The clock line can be observed by reading the SCK pin. The Idle state is determined by the CKP bit of the SSPCON1 register.

While in Slave mode, the external clock is supplied by the external clock source on the SCK pin. This external clock must meet the minimum high and low times as specified in the electrical specifications.

While in Sleep mode, the slave can transmit/receive data. When a byte is received, the device will wake-up from Sleep.

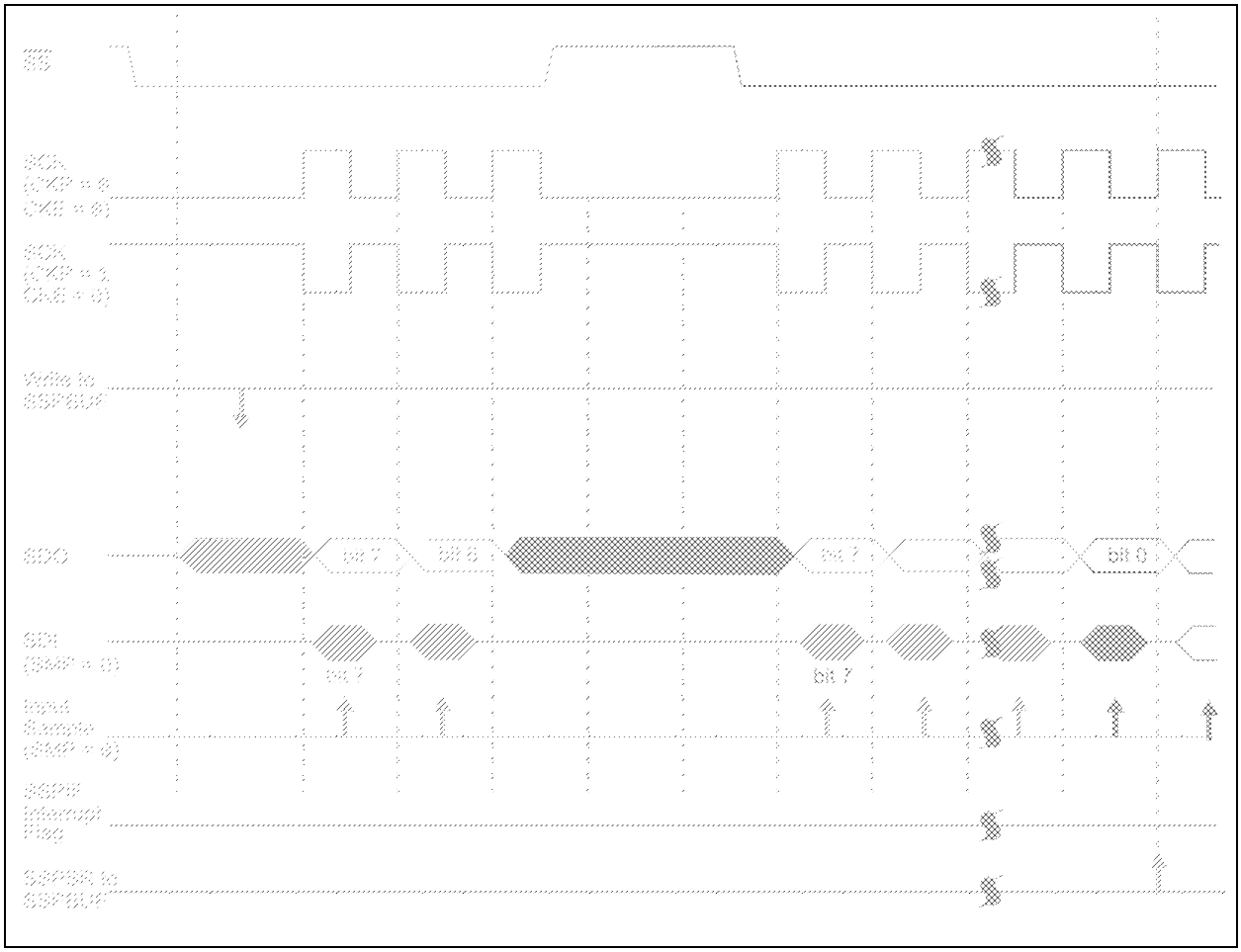
14.2.7 SLAVE SELECT SYNCHRONIZATION

The $\overline{SS}$ pin allows a Synchronous Slave mode. The SPI must be in Slave mode with $\overline{SS}$ pin control enabled (SSPCON1<3:0> = 0100). When the $\overline{SS}$ pin is low, transmission and reception are enabled and the SDO pin is driven. When the $\overline{SS}$ pin goes high, the SDO pin is no longer driven, even if in the middle of a transmitted byte and becomes a floating output. External pull-up/pull-down resistors may be desirable depending on the application.

Note 1:	When the SPI is in Slave mode with $\overline{SS}$ pin control enabled (SSPCON<3:0> = 0100), the SPI module will reset if the $\overline{SS}$ pin is set to VDD.
2:	When the SPI is used in Slave mode with CKE set the $\overline{SS}$ pin control must also be enabled.

When the SPI module resets, the bit counter is forced to '0'. This can be done by either forcing the $\overline{SS}$ pin to a high level or clearing the SSPEN bit.

FIGURE 14-4: SLAVE SYNCHRONIZATION WAVEFORM



REGISTER 14-7: SSPADD: MSSP ADDRESS AND BAUD RATE REGISTER (I²C MODE)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0
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

Master mode:

bit 7-0 **ADD<7:0>:** Baud Rate Clock Divider bits
SCL pin clock period = ((ADD<7:0> + 1) *4)/FOSC

10-Bit Slave mode — Most Significant Address Byte:

bit 7-3 **Not used:** Unused for Most Significant Address Byte. Bit state of this register is a "don't care." Bit pattern sent by master is fixed by I²C specification and must be equal to '11110'. However, those bits are compared by hardware and are not affected by the value in this register.

bit 2-1 **ADD<9:8>:** Two Most Significant bits of 10-bit address

bit 0 **Not used:** Unused in this mode. Bit state is a "don't care."

10-Bit Slave mode — Least Significant Address Byte:

bit 7-0 **ADD<7:0>:** Eight Least Significant bits of 10-bit address

7-Bit Slave mode:

bit 7-1 **ADD<6:0>:** 7-bit address

bit 0 **Not used:** Unused in this mode. Bit state is a "don't care."

16.1.6 INTERRUPTS

The ADC module allows for the ability to generate an interrupt upon completion of an Analog-to-Digital Conversion. The ADC interrupt flag is the ADIF bit in the PIR1 register. The ADC interrupt enable is the ADIE bit in the PIE1 register. The ADIF bit must be cleared by software.

Note:	The ADIF bit is set at the completion of every conversion, regardless of whether or not the ADC interrupt is enabled.

This interrupt can be generated while the device is operating or while in Sleep. If the device is in Sleep, the interrupt will wake-up the device. Upon waking from Sleep, the next instruction following the SLEEP instruction is always executed. If the user is attempting to wake-up from Sleep and resume in-line code execution, the global interrupt must be disabled. If the global interrupt is enabled, execution will switch to the Interrupt Service Routine. Please see **Section 16.1.6 "Interrupts"** for more information.

TABLE 16-1: ADC CLOCK PERIOD (T_{AD}) vs. DEVICE OPERATING FREQUENCIES

ADC Clock Period (T_{AD})		Device Frequency (F_{OSC})			
ADC Clock Source	**ADCS<2:0>**	**48 MHz**	**16 MHz**	**4 MHz**	**1 MHz**
F_{OSC}/2	000	41.67 ns(2)	125 ns(2)	500 ns(2)	2.0 μs
F_{OSC}/4	100	83.33 ns(2)	250 ns(2)	1.0 μs	4.0 μs
F_{OSC}/8	001	167 ns(2)	500 ns(2)	2.0 μs	8.0 μs(3)
F_{OSC}/16	101	333 ns(2)	1.0 μs	4.0 μs	16.0 μs(3)
F_{OSC}/32	010	667 ns(2)	2.0 μs	8.0 μs(3)	32.0 μs(3)
F_{OSC}/64	110	1.33 μs	4.0 μs	16.0 μs(3)	64.0 μs(3)
F_{RC}	x11	1-4 μs(1,4)	1-4 μs(1,4)	1-4 μs(1,4)	1-4 μs(1,4)

Legend: Shaded cells are outside of recommended range.

Note 1: The F_{RC} source has a typical T_{AD} time of 1.7 μs.

2: These values violate the minimum required T_{AD} time.

3: For faster conversion times, the selection of another clock source is recommended.

4: When the device frequency is greater than 1 MHz, the F_{RC} clock source is only recommended if the conversion will be performed during Sleep.

16.1.7 RESULT FORMATTING

The 10-bit A/D conversion result can be supplied in two formats, left justified or right justified. The ADFM bit of the ADCON2 register controls the output format.

Figure 16-2 shows the two output formats.

FIGURE 16-2: 10-BIT A/D CONVERSION RESULT FORMAT

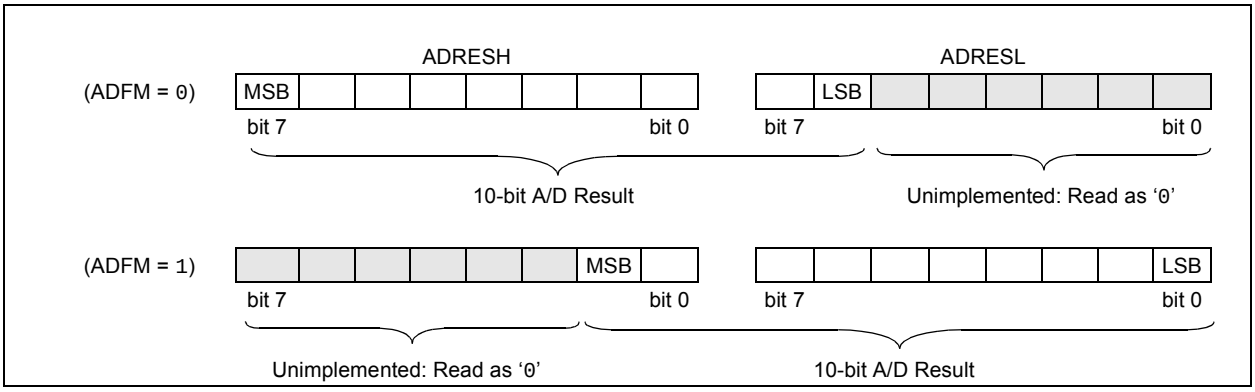


TABLE 19-1: SRCLK FREQUENCY TABLE

SRCLK	Divider	Fosc = 20 MHz	Fosc = 16 MHz	Fosc = 8 MHz	Fosc = 4 MHz	Fosc = 1 MHz
111	512	25.6 μs	32 μs	64 μs	128 μs	512 μs
110	256	12.8 μs	16 μs	32 μs	64 μs	256 μs
101	128	6.4 μs	8 μs	16 μs	32 μs	128 μs
100	64	3.2 μs	4 μs	8 μs	16 μs	64 μs
011	32	1.6 μs	2 μs	4 μs	8 μs	32 μs
010	16	0.8 μs	1 μs	2 μs	4 μs	16 μs
001	8	0.4 μs	0.5 μs	1 μs	2 μs	8 μs
000	4	0.2 μs	0.25 μs	0.5 μs	1 μs	4 μs

REGISTER 19-1: SRCON0: SR LATCH 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-0
SRLEN	SRCLK2	SRCLK1	SRCLK0	SRQEN	SRNQEN	SRPS	SRPR
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 **SRLEN:** SR Latch Enable bit(1)
- 1 = SR latch is enabled
- 0 = SR latch is disabled

bit 6-4 **SRCLK<2:0>(1):** SR Latch Clock divider bits
- 000 = 1/4 Peripheral cycle clock
- 001 = 1/8 Peripheral cycle clock
- 010 = 1/16 Peripheral cycle clock
- 011 = 1/32 Peripheral cycle clock
- 100 = 1/64 Peripheral cycle clock
- 101 = 1/128 Peripheral cycle clock
- 110 = 1/256 Peripheral cycle clock
- 111 = 1/512 Peripheral cycle clock

bit 3 **SRQEN:** SR Latch Q Output Enable bit
- 1 = Q is present on the SRQ pin
- 0 = Q is internal only

bit 2 **SRNQEN:** SR Latch $\overline{Q}$ Output Enable bit
- 1 = $\overline{Q}$ is present on the SRNQ pin
- 0 = $\overline{Q}$ is internal only

bit 1 **SRPS:** Pulse Set Input of the SR Latch bit
- 1 = Pulse input
- 0 = Always reads back '0'

bit 0 **SRPR:** Pulse Reset Input of the SR Latch bit
- 1 = Pulse input
- 0 = Always reads back '0'

Note 1: Changing the SRCLK bits while the SR latch is enabled may cause false triggers to the set and Reset inputs of the latch.

22.5 Device Reset Timers

PIC18(L)F1XK22 devices incorporate three separate on-chip timers that help regulate the Power-on Reset process. Their main function is to ensure that the device clock is stable before code is executed. These timers are:

- Power-up Timer (PWRT)
- Oscillator Start-up Timer (OST)
- PLL Lock Time-out

22.5.1 POWER-UP TIMER (PWRT)

The Power-up Timer (PWRT) of PIC18(L)F1XK22 devices is an 11-bit counter which uses the LFINTOSC source as the clock input. This yields an approximate time interval of 2048 x 32 μs = 65.6 ms. While the PWRT is counting, the device is held in Reset.

The power-up time delay depends on the LFINTOSC clock and will vary from chip-to-chip due to temperature and process variation. See **Section 26.0 "Electrical Specifications"** for details.

The PWRT is enabled by clearing the $\overline{\text{PWRTEN}}$ Configuration bit.

22.5.2 OSCILLATOR START-UP TIMER (OST)

The Oscillator Start-up Timer (OST) provides a 1024 oscillator cycle (from OSC1 input) delay after the PWRT delay is over. This ensures that the crystal oscillator or resonator has started and stabilized.

The OST time-out is invoked only for XT, LP, HS and HSPLL modes and only on Power-on Reset, or on exit from all power-managed modes that stop the external oscillator.

22.5.3 PLL LOCK TIME-OUT

With the PLL enabled in its PLL mode, the time-out sequence following a Power-on Reset is slightly different from other oscillator modes. A separate timer is used to provide a fixed time-out that is sufficient for the PLL to lock to the main oscillator frequency. This PLL lock time-out (T_{PLL}) is typically 2 ms and follows the oscillator start-up time-out.

22.5.4 TIME-OUT SEQUENCE

On power-up, the time-out sequence is as follows:

1. After the POR pulse has cleared, PWRT time-out is invoked (if enabled).
2. Then, the OST is activated.

The total time-out will vary based on oscillator configuration and the status of the PWRT. Figure 22-3, Figure 22-4, Figure 22-5, Figure 22-6 and Figure 22-7 all depict time-out sequences on power-up, with the Power-up Timer enabled and the device operating in HS Oscillator mode. Figures 22-3 through 22-6 also apply to devices operating in XT or LP modes. For devices in RC mode and with the PWRT disabled, on the other hand, there will be no time-out at all.

Since the time-outs occur from the POR pulse, if $\overline{\text{MCLR}}$ is kept low long enough, all time-outs will expire, after which, bringing $\overline{\text{MCLR}}$ high will allow program execution to begin immediately (Figure 22-5). This is useful for testing purposes or to synchronize more than one PIC18(L)F1XK22 device operating in parallel.

TABLE 22-2: TIME-OUT IN VARIOUS SITUATIONS

Oscillator Configuration	Power-up(2) and Brown-out		Exit from Power-Managed Mode
	$\overline{\text{PWRTEN}}$ = 0	$\overline{\text{PWRTEN}}$ = 1	
HSPLL	66 ms(1) + 1024 TOSC + 2 ms(2)	1024 TOSC + 2 ms(2)	1024 TOSC + 2 ms(2)
HS, XT, LP	66 ms(1) + 1024 TOSC	1024 TOSC	1024 TOSC
EC, ECIO	66 ms(1)	—	—
RC, RCIO	66 ms(1)	—	—
INTIO1, INTIO2	66 ms(1)	—	—

Note 1: 66 ms (65.5 ms) is the nominal Power-up Timer (PWRT) delay.
2: 2 ms is the nominal time required for the PLL to lock.

TABLE 22-4: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Address	Power-on Reset, Brown-out Reset	MCLR Resets, WDT Reset, RESET Instruction, Stack Resets	Wake-up via WDT or Interrupt
ADRESH	FC4h	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADRESL	FC3h	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0	FC2h	--00 0000	--00 0000	--uu uuuu
ADCON1	FC1h	---- 0000	---- 0000	---- uuuu
ADCON2	FC0h	0-00 0000	0-00 0000	u-uu uuuu
CCPR1H	FBFh	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1L	FBEh	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON	FBDh	0000 0000	0000 0000	uuuu uuuu
VREFCON2	FBCh	---0 0000	---0 0000	---u uuuu
VREFCON1	FBBh	000- 00-0	000- 00-0	uuu- uu-u
VREFCON0	FBAh	0001 ----	0001 ----	uuuu ----
PSTRCON	FB9h	---0 0001	---0 0001	---u uuuu
BAUDCON	FB8h	0100 0-00	0100 0-00	uuuu u-uu
PWM1CON	FB7h	0000 0000	0000 0000	uuuu uuuu
ECCP1AS	FB6h	0000 0000	0000 0000	uuuu uuuu
TMR3H	FB3h	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR3L	FB2h	xxxx xxxx	uuuu uuuu	uuuu uuuu
T3CON	FB1h	0-00 0000	u-uu uuuu	u-uu uuuu
SPBRGH	FB0h	0000 0000	0000 0000	uuuu uuuu
SPBRG	FAFh	0000 0000	0000 0000	uuuu uuuu
RCREG	FAEh	0000 0000	0000 0000	uuuu uuuu
TXREG	FADh	0000 0000	0000 0000	uuuu uuuu
TXSTA	FACh	0000 0010	0000 0010	uuuu uuuu
RCSTA	FABh	0000 000x	0000 000x	uuuu uuuu
EEADR	FAAh	0000 0000	0000 0000	uuuu uuuu
EEDATA	FA8h	0000 0000	0000 0000	uuuu uuuu
EECON2	FA7h	0000 0000	0000 0000	0000 0000
EECON1	FA6h	xx-0 x000	uu-0 u000	uu-0 u000

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition. Shaded cells indicate conditions do not apply for the designated device.

Note 1: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

3: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

4: See Table 22-3 for Reset value for specific condition.

MOVLW **Move literal to W**

Syntax: MOVLW k

Operands: $0 \le k \le 255$

Operation: $k \rightarrow W$

Status Affected: None

Encoding:

0000	1110	kkkk	kkkk

Description: The 8-bit literal ‘k’ is loaded into W.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal ‘k’	Process Data	Write to W

Example: MOVLW 5Ah

After Instruction

W = 5Ah

MOVWF **Move W to f**

Syntax: MOVWF f {,a}

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

Operation: $(W) \rightarrow f$

Status Affected: None

Encoding:

0110	111a	ffff	ffff

Description: Move data from W to register ‘f’. Location ‘f’ can be anywhere in the 256-byte bank.
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 register ‘f’

Example: MOVWF REG, 0

Before Instruction

W = 4Fh
REG = FFh

After Instruction

W = 4Fh
REG = 4Fh

25.6 MPLAB X SIM Software Simulator

The MPLAB X SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB X SIM Software Simulator fully supports symbolic debugging using the MPLAB XC Compilers, and the MPASM and MPLAB Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

25.7 MPLAB REAL ICE In-Circuit Emulator System

The MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs all 8, 16 and 32-bit MCU, and DSC devices with the easy-to-use, powerful graphical user interface of the MPLAB X IDE.

The emulator is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ-11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradable through future firmware downloads in MPLAB X IDE. MPLAB REAL ICE offers significant advantages over competitive emulators including full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, logic probes, a ruggedized probe interface and long (up to three meters) interconnection cables.

25.8 MPLAB ICD 3 In-Circuit Debugger System

The MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost-effective, high-speed hardware debugger/programmer for Microchip Flash DSC and MCU devices. It debugs and programs PIC Flash microcontrollers and dsPIC DSCs with the powerful, yet easy-to-use graphical user interface of the MPLAB IDE.

The MPLAB ICD 3 In-Circuit Debugger probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with a connector compatible with the MPLAB ICD 2 or MPLAB REAL ICE systems (RJ-11). MPLAB ICD 3 supports all MPLAB ICD 2 headers.

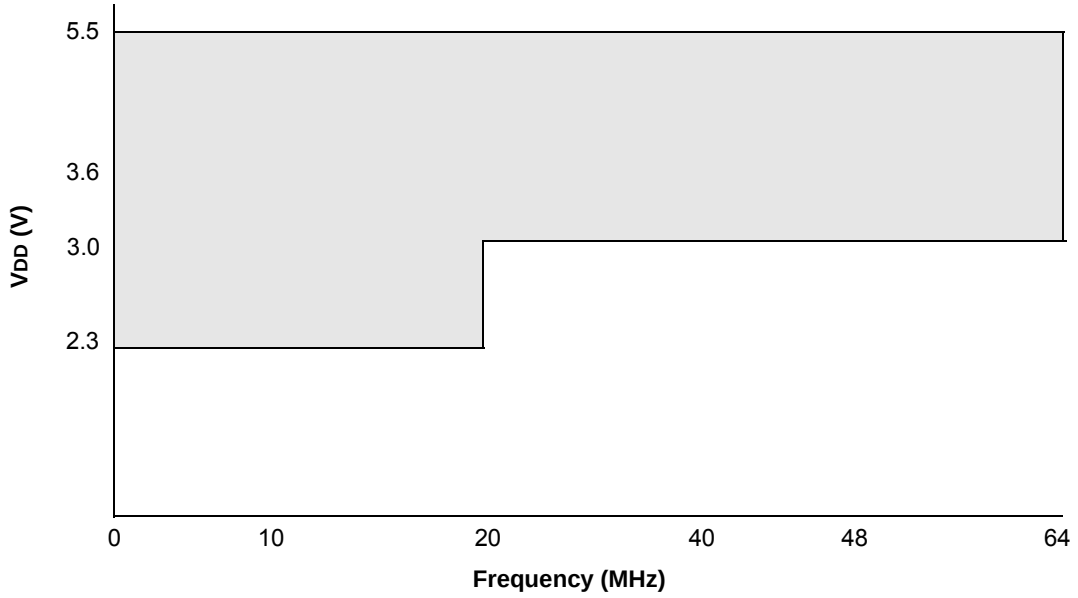
25.9 PICkit 3 In-Circuit Debugger/ Programmer

The MPLAB PICkit 3 allows debugging and programming of PIC and dsPIC Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB IDE. The MPLAB PICkit 3 is connected to the design engineer's PC using a full-speed USB interface and can be connected to the target via a Microchip debug (RJ-11) connector (compatible with MPLAB ICD 3 and MPLAB REAL ICE). The connector uses two device I/O pins and the Reset line to implement in-circuit debugging and In-Circuit Serial Programming™ (ICSP™).

25.10 MPLAB PM3 Device Programmer

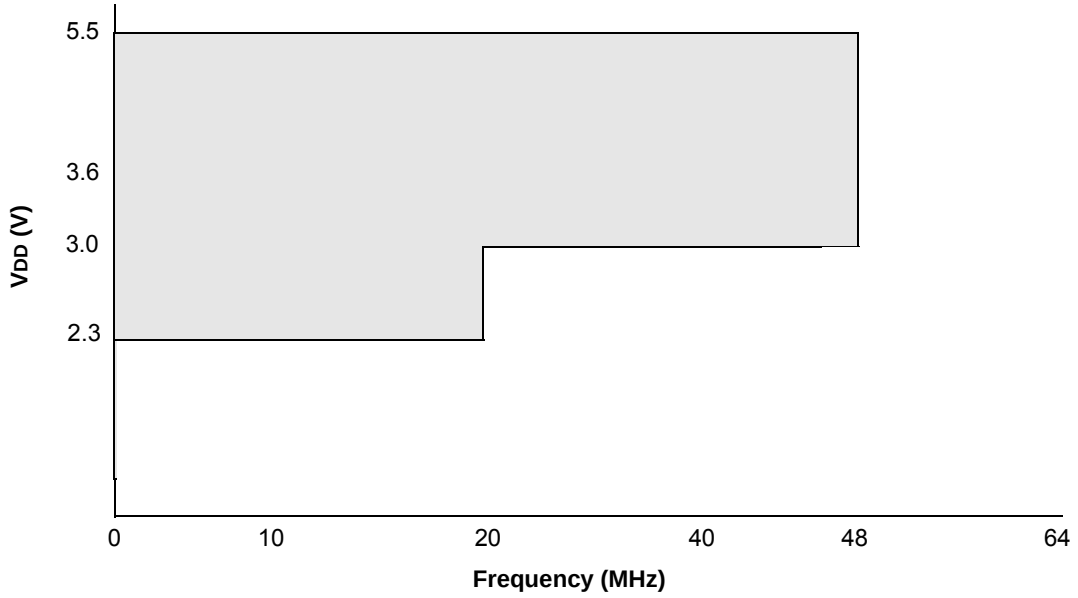
The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at V_{DDMIN} and V_{DDMAX} for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages, and a modular, detachable socket assembly to support various package types. The ICSP cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices, and incorporates an MMC card for file storage and data applications.

FIGURE 26-1: PIC18F1XK22 VOLTAGE FREQUENCY GRAPH, -40°C ≤ TA ≤ +85°C



Note 1: The shaded region indicates the permissible combinations of voltage and frequency.
2: Refer to Table 26-13 for each Oscillator mode's supported frequencies.

FIGURE 26-2: PIC18F1XK22 VOLTAGE FREQUENCY GRAPH, -40°C ≤ TA ≤ +125°C



Note 1: The shaded region indicates the permissible combinations of voltage and frequency.
2: Refer to Table 26-13 for each Oscillator mode's supported frequencies.

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

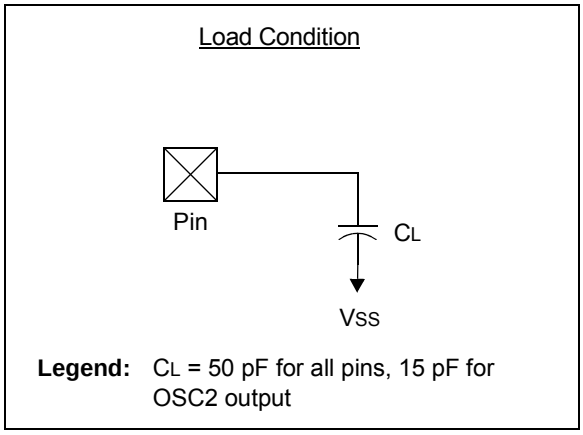
Lowercase letters (pp) and their meanings:

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



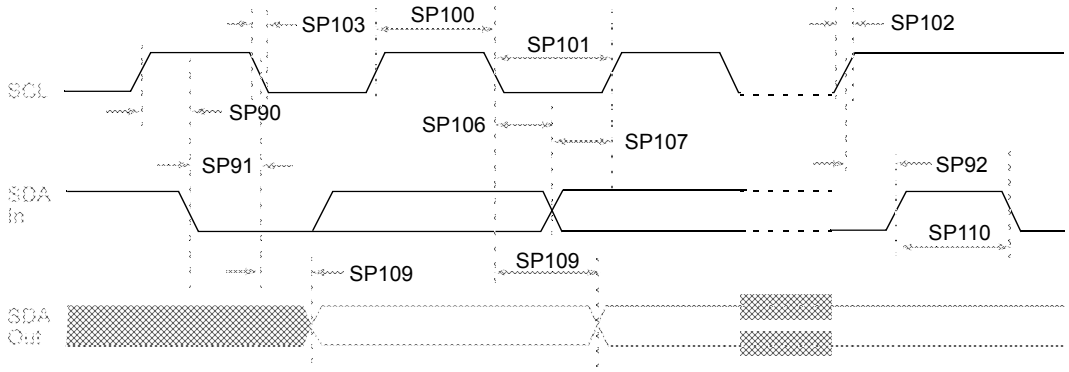
Legend: CL = 50 pF for all pins, 15 pF for OSC2 output

TABLE 26-27: I²C BUS START/STOP BITS REQUIREMENTS

Param. No.	Symbol	Characteristic		Min.	Typ.	Max.	Units	Conditions
SP90*	TSU:STA	Start condition Setup time	100 kHz mode	4700	—	—	ns	Only relevant for Repeated Start condition
			400 kHz mode	600	—	—		
SP91*	THD:STA	Start condition Hold time	100 kHz mode	4000	—	—	ns	After this period, the first clock pulse is generated
			400 kHz mode	600	—	—		
SP92*	TSU:STO	Stop condition Setup time	100 kHz mode	4700	—	—	ns	
			400 kHz mode	600	—	—		
SP93	THD:STO	Stop condition Hold time	100 kHz mode	4000	—	—	ns	
			400 kHz mode	600	—	—		

* These parameters are characterized but not tested.

FIGURE 26-22: I²C BUS DATA TIMING



Note: Refer to Figure 26-7 for load conditions.

FIGURE 27-27: PIC18LF1XK22 TYPICAL LF-INTOSC FREQUENCY (MAX./MIN. = 31.25 kHz ± 15%)

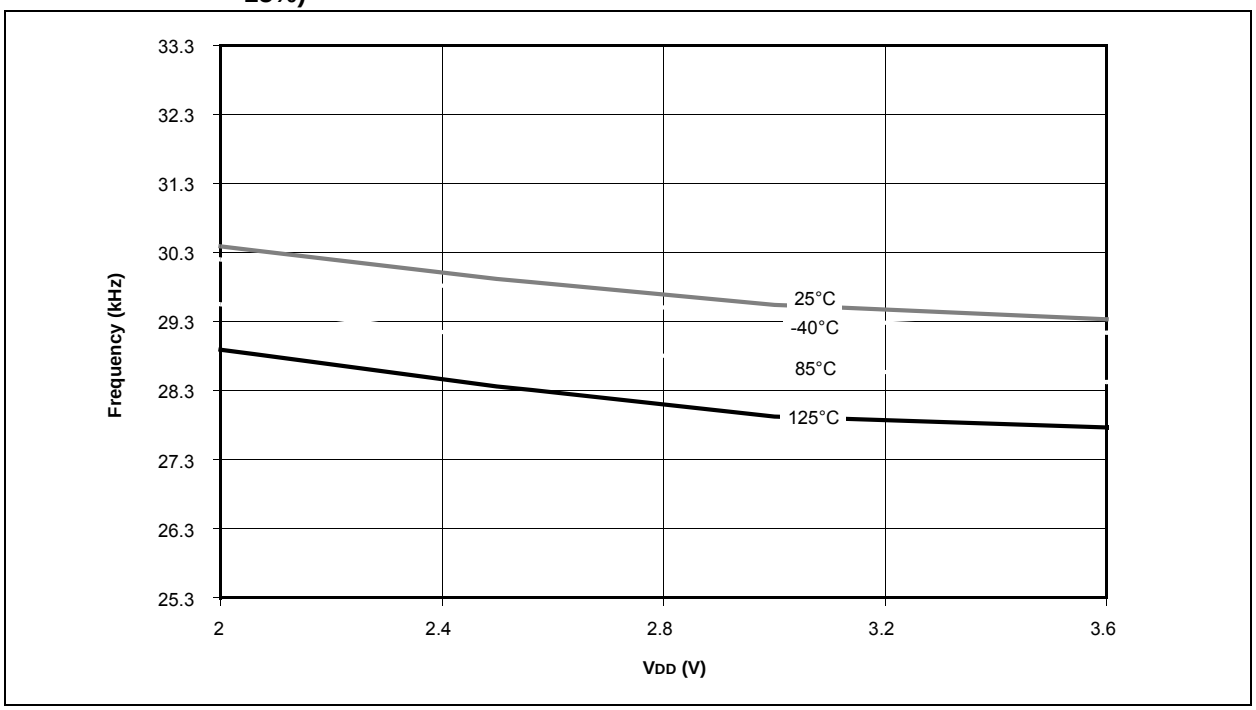


FIGURE 27-28: PIC18LF1XK22 TYPICAL LF-INTOSC FREQUENCY (MAX./MIN. = 31.25 kHz ± 15%)