

Welcome to [E-XFL.COM](https://www.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	I ² C, 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	Through Hole
Package / Case	20-DIP (0.300", 7.62mm)
Supplier Device Package	20-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf14k22-e-p

1.2 Other Special Features

- **Memory Endurance:** The Flash cells for both program memory and data EEPROM are rated to last for many thousands of erase/write cycles – up to 10K for program memory and 100K for EEPROM. Data retention without refresh is conservatively estimated to be greater than 40 years.
- **Self-programmability:** These devices can write to their own program memory spaces under internal software control. Using a bootloader routine located in the code protected Boot Block, it is possible to create an application that can update itself in the field.
- **Extended Instruction Set:** The PIC18(L)F1XK22 family introduces an optional extension to the PIC18 instruction set, which adds eight new instructions and an Indexed Addressing mode. This extension has been specifically designed to optimize re-entrant application code originally developed in high-level languages, such as C.
- **Enhanced CCP module:** In PWM mode, this module provides one, two or four modulated outputs for controlling half-bridge and full-bridge drivers. Other features include:
 - Auto-Shutdown, for disabling PWM outputs on interrupt or other select conditions
 - Auto-Restart, to reactivate outputs once the condition has cleared
 - Output steering to selectively enable one or more of four outputs to provide the PWM signal.
- **Enhanced Addressable USART:** This serial communication module is capable of standard RS-232 operation and provides support for the LIN bus protocol. Other enhancements include automatic baud rate detection and a 16-bit Baud Rate Generator for improved resolution.
- **10-bit A/D Converter:** This module incorporates programmable acquisition time, allowing for a channel to be selected and a conversion to be initiated without waiting for a sampling period and thus, reduce code overhead.
- **Extended Watchdog Timer (WDT):** This enhanced version incorporates a 16-bit postscaler, allowing an extended time-out range that is stable across operating voltage and temperature. See **Section 26.0 “Electrical Specifications”** for time-out periods.

1.3 Details on Individual Family Members

Devices in the PIC18(L)F1XK22 family are available in 20-pin packages. Block diagrams for the two groups are shown in Figure 1-1.

The devices are differentiated from each other in the following ways:

1. Flash program memory:
 - 8 Kbytes for PIC18(L)F13K22
 - 16 Kbytes for PIC18(L)F14K22

All other features for devices in this family are identical. These are summarized in Table 1-1.

The pinouts for all devices are listed in Table 1 and I/O description are in Table 1-2.

PIC18(L)F1XK22

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

Pin Name	Pin Number		Pin Type	Buffer Type	Description
	PDIP/SSOP/SOIC	QFN			
RA0/AN0/CVREF/VREF-/C1IN+/INT0/PGD RA0 AN0 CVREF/DAC1OUT VREF- C1IN+ INT0 PGD	19	16	I/O I O I I I I/O	TTL Analog Analog Analog Analog ST ST	Digital I/O ADC channel 0 DAC reference voltage output ADC and DAC reference voltage (low) input Comparator C1 noninverting input External interrupt 0 ICSP™ programming data pin
RA1/AN1/C12IN0-/VREF+/INT1/PGC RA1 AN1 C12IN0- VREF+ INT1 PGC	18	15	I/O I I I I I/O	TTL Analog Analog Analog ST ST	Digital I/O ADC channel 1 Comparator C1 and C2 inverting input ADC and DAC reference voltage (high) input External interrupt 1 ICSP programming clock pin
RA2/AN2/C1OUT/T0CKI/INT2/SRQ RA2 AN2 C1OUT T0CKI INT2 SRQ	17	14	I/O I — I I O	ST Analog CMOS ST ST CMOS	Digital I/O ADC channel 2 Comparator C1 output Timer0 external clock input External interrupt 2 SR latch output
RA3/MCLR/VPP RA3 MCLR VPP	4	1	I I P	ST ST —	Digital input Active-low Master Clear with internal pull-up High voltage programming input
RA4/AN3/OSC2/CLKOUT RA4 AN3 OSC2 CLKOUT	3	20	I/O I O O	TTL Analog XTAL CMOS	Digital I/O ADC channel 3 Oscillator crystal output. Connect to crystal or resonator in Crystal Oscillator mode In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate
RA5/OSC1/CLKIN/T13CKI RA5 OSC1 CLKIN T13CKI	2	19	I/O I I I	TTL XTAL CMOS ST	Digital I/O Oscillator crystal input or external clock input ST buffer when configured in RC mode; analog other wise External clock source input. Always associated with the pin function OSC1 (See related OSC1/CLKIN, OSC2, CLKOUT pins Timer0 and Timer3 external clock input
RB4/AN10/SDI/SDA RB4 AN10 SDI SDA	13	10	I/O I I I/O	TTL Analog ST ST	Digital I/O ADC channel 10 SPI data in I ² C data I/O

Legend: TTL = TTL compatible input
ST = Schmitt Trigger input
O = Output
XTAL = Crystal Oscillator

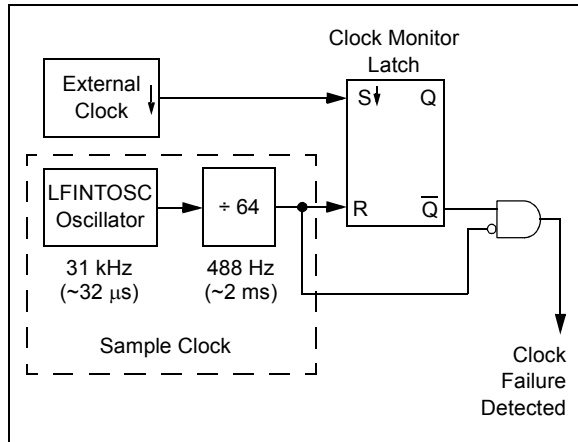
CMOS = CMOS compatible input or output
I = Input
P = Power

PIC18(L)F1XK22

2.12 Fail-Safe Clock Monitor

The Fail-Safe Clock Monitor (FSCM) allows the device to continue operating should the external oscillator fail. The FSCM can detect oscillator failure any time after the Oscillator Start-up Timer (OST) has expired. The FSCM is enabled by setting the FCMEN bit in the CONFIG1H Configuration register. The FSCM is applicable to all external oscillator modes (LP, XT, HS, EC and RC).

FIGURE 2-6: FSCM BLOCK DIAGRAM



2.12.1 FAIL-SAFE DETECTION

The FSCM module detects a failed oscillator by comparing the external oscillator to the FSCM sample clock. The sample clock is generated by dividing the LFINTOSC by 64. See Figure 2-6. Inside the fail detector block is a latch. The external clock sets the latch on each falling edge of the external clock. The sample clock clears the latch on each rising edge of the sample clock. A failure is detected when an entire half-cycle of the sample clock elapses before the primary clock goes low.

2.12.2 FAIL-SAFE OPERATION

When the external clock fails, the FSCM switches the device clock to an internal clock source and sets the bit flag OSCFIF of the PIR2 register. The OSCFIF flag will generate an interrupt if the OSCFIE bit of the PIE2 register is also set. The device firmware can then take steps to mitigate the problems that may arise from a failed clock. The system clock will continue to be sourced from the internal clock source until the device firmware successfully restarts the external oscillator and switches back to external operation. An automatic transition back to the failed clock source will not occur.

The internal clock source chosen by the FSCM is determined by the IRCF<2:0> bits of the OSCCON register. This allows the internal oscillator to be configured before a failure occurs.

2.12.3 FAIL-SAFE CONDITION CLEARING

The Fail-Safe condition is cleared by either one of the following:

- Any Reset
- By toggling the SCS1 bit of the OSCCON register

Both of these conditions restart the OST. While the OST is running, the device continues to operate from the INTOSC selected in OSCCON. When the OST times out, the Fail-Safe condition is cleared and the device automatically switches over to the external clock source. The Fail-Safe condition need not be cleared before the OSCFIF flag is cleared.

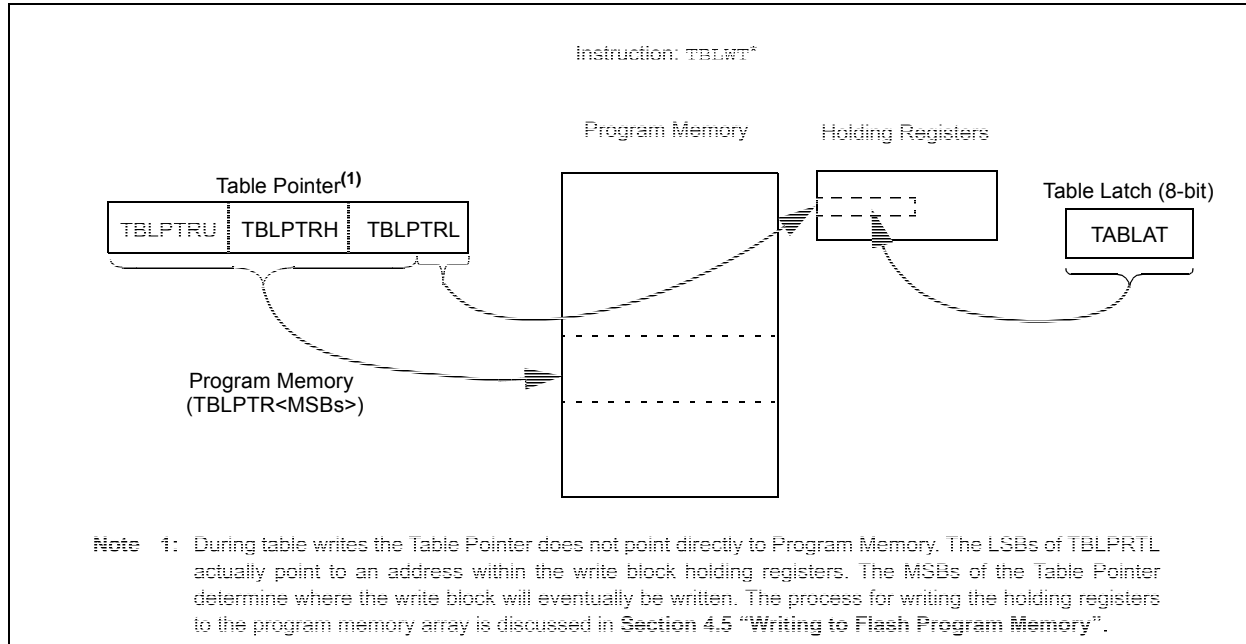
2.12.4 RESET OR WAKE-UP FROM SLEEP

The FSCM is designed to detect an oscillator failure after the Oscillator Start-up Timer (OST) has expired. The OST is used after waking up from Sleep and after any type of Reset. The OST is not used with the EC or RC Clock modes so that the FSCM will be active as soon as the Reset or wake-up has completed. When the FSCM is enabled, the Two-Speed Start-up is also enabled. Therefore, the device will always be executing code while the OST is operating.

Note: Due to the wide range of oscillator start-up times, the Fail-Safe circuit is not active during oscillator start-up (i.e., after exiting Reset or Sleep). After an appropriate amount of time, the user should check the OSTS bit of the OSCCON register to verify the oscillator start-up and that the system clock switchover has successfully completed.

PIC18(L)F1XK22

FIGURE 4-2: TABLE WRITE OPERATION



4.2 Control Registers

Several control registers are used in conjunction with the TBLRD and TBLWT instructions. These include the:

- EECON1 register
- EECON2 register
- TABLAT register
- TBLPTR registers

4.2.1 EECON1 AND EECON2 REGISTERS

The EECON1 register (Register 4-1) is the control register for memory accesses. The EECON2 register is not a physical register; it is used exclusively in the memory write and erase sequences. Reading EECON2 will read all '0's.

The EEPGD control bit determines if the access will be a program or data EEPROM memory access. When EEPGD is clear, any subsequent operations will operate on the data EEPROM memory. When EEPGD is set, any subsequent operations will operate on the program memory.

The CFGS control bit determines if the access will be to the Configuration/Calibration registers or to program memory/data EEPROM memory. When CFGS is set, subsequent operations will operate on Configuration registers regardless of EEPGD (see Section 23.0 "Special Features of the CPU"). When CFGS is clear, memory selection access is determined by EEPGD.

The FREE bit allows the program memory erase operation. When FREE is set, an erase operation is initiated on the next WR command. When FREE is clear, only writes are enabled.

The WREN bit, when set, will allow a write operation. The WREN bit is clear on power-up.

The WRERR bit is set by hardware when the WR bit is set and cleared when the internal programming timer expires and the write operation is complete.

Note: During normal operation, the WRERR is read as '1'. This can indicate that a write operation was prematurely terminated by a Reset, or a write operation was attempted improperly.

The WR control bit initiates write operations. The WR bit cannot be cleared, only set, by firmware. Then WR bit is cleared by hardware at the completion of the write operation.

Note: The EEIF interrupt flag bit of the PIR2 register is set when the write is complete. The EEIF flag stays set until cleared by firmware.

8.2 PORTB, TRISB and LATB Registers

PORTB is an 4-bit wide, bidirectional port. The corresponding data direction register is TRISB. Setting a TRISB bit (= 1) will make the corresponding PORTB pin an input (i.e., disable the output driver). Clearing a TRISB bit (= 0) will make the corresponding PORTB pin an output (i.e., enable the output driver and put the contents of the output latch on the selected pin).

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

EXAMPLE 8-2: INITIALIZING PORTB

```
CLRF    PORTB    ; Initialize PORTB by
                  ; clearing output
                  ; data latches
CLRF    LATB     ; Alternate method
                  ; to clear output
                  ; data latches
MOVLW   0F0h     ; Value used to
                  ; initialize data
                  ; direction
MOVWF   TRISB    ; Set RB<7:4> as outputs
```

All PORTB pins are individually configurable as interrupt-on-change pins. Control bits in the IOCB register enable (when set) or disable (when clear) the interrupt function for each pin.

When set, the RABIE bit of the INTCON register enables interrupts on all pins which also have their corresponding IOCB bit set. When clear, the RABIE bit disables all interrupt-on-changes.

Only pins configured as inputs can cause this interrupt to occur (i.e., any pin configured as an output is excluded from the interrupt-on-change comparison).

For enabled interrupt-on-change pins, the values are compared with the old value latched on the last read of PORTB. The 'mismatch' outputs of the last read are OR'd together to set the PORTB Change Interrupt flag bit (RABIF) in the INTCON register.

This interrupt can wake the device from the Sleep mode, or any of the Idle modes. The user, in the Interrupt Service Routine, can clear the interrupt in the following manner:

- Any read or write of PORTB to clear the mismatch condition (except when PORTB is the source or destination of a MOVFF instruction).
- Clear the flag bit, RABIF.

A mismatch condition will continue to set the RABIF flag bit. Reading or writing PORTB will end the mismatch condition and allow the RABIF bit to be cleared. The latch holding the last read value is not affected by a MCLR nor Brown-out Reset. After either one of these Resets, the RABIF flag will continue to be set if a mismatch is present.

Note: If a change on the I/O pin should occur when the read operation is being executed (start of the Q2 cycle), then the RABIF interrupt flag may not get set. Furthermore, since a read or write on a port affects all bits of that port, care must be taken when using multiple pins in Interrupt-on-Change mode. Changes on one pin may not be seen while servicing changes on another pin.

The interrupt-on-change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt-on-change feature. Polling of PORTB is not recommended while using the interrupt-on-change feature.

All PORTB pins have individually controlled weak internal pull-up. When set, each bit of the WPUB register enables the corresponding pin pull-up. When cleared, the RABPU bit of the INTCON2 register enables pull-ups on all pins which also have their corresponding WPUB bit set. When set, the RABPU bit disables all weak pull-ups. The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on a Power-on Reset.

Note: On a Power-on Reset, RB<5:4> are configured as analog inputs by default and read as '0'.

10.2 Timer1 16-Bit Read/Write Mode

Timer1 can be configured for 16-bit reads and writes (see Figure 10-2). When the RD16 control bit of the T1CON register is set, the address for TMR1H is mapped to a buffer register for the high byte of Timer1. A read from TMR1L will load the contents of the high byte of Timer1 into the Timer1 high byte buffer. This provides the user with the ability to accurately read all 16 bits of Timer1 without the need to determine whether a read of the high byte, followed by a read of the low byte, has become invalid due to a rollover or carry between reads.

Writing to TMR1H does not directly affect Timer1. Instead, the high byte of Timer1 is updated with the contents of TMR1H when a write occurs to TMR1L. This allows all 16 bits of Timer1 to be updated at once.

The high byte of Timer1 is not directly readable or writable in this mode. All reads and writes must take place through the Timer1 High Byte Buffer register. Writes to TMR1H do not clear the Timer1 prescaler. The prescaler is only cleared on writes to TMR1L.

10.3 Clock Source Selection

The TMR1CS bit of the T1CON register is used to select the clock source. When TMR1CS = 0, the clock source is Fosc/4. When TMR1CS = 1, the clock source is supplied externally.

Clock Source	T1OSCEN	FOSC Mode	TMR1CS
Fosc/4	x	xxx	0
T1CKI pin	0	xxx	1
T1LPOSC	1	LP or INTOSCIO	1

10.3.1 INTERNAL CLOCK SOURCE

When the internal clock source is selected the TMR1H:TMR1L register pair will increment on multiples of Fosc as determined by the Timer1 prescaler.

10.3.2 EXTERNAL CLOCK SOURCE

When the external clock source is selected, the Timer1 module may work as a timer or a counter.

When counting, Timer1 is incremented on the rising edge of the external clock input T1CKI. In addition, the Counter mode clock can be synchronized to the microcontroller system clock or run asynchronously.

If an external clock oscillator is needed (and the microcontroller is using the INTOSC without CLKOUT), Timer1 can use the LP oscillator as a clock source.

Note: In Counter mode, a falling edge must be registered by the counter prior to the first incrementing rising edge after any one or more of the following conditions:

- Timer1 enabled after POR
- Write to TMR1H or TMR1L
- Timer1 is disabled
- Timer1 is disabled (TMR1ON = 0) when T1CKI is high then Timer1 is enabled (TMR1ON = 1) when T1CKI is low.

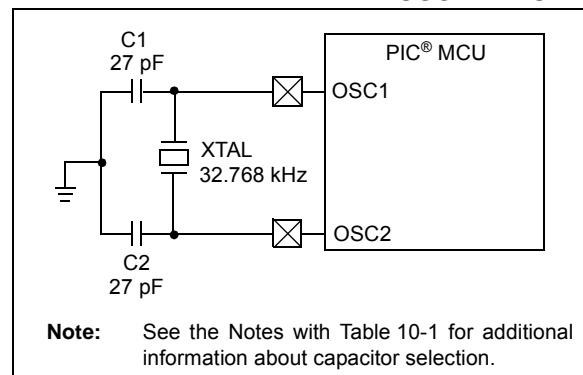
Note: See Figure 9-2.

10.4 Timer1 Oscillator

An on-chip crystal oscillator circuit is incorporated between pins OSC1 (input) and OSC2 (amplifier output). It is enabled by setting the Timer1 Oscillator Enable bit, T1OSCEN of the T1CON register. The oscillator is a low-power circuit rated for 32 kHz crystals. It will continue to run during all power-managed modes. The circuit for a typical LP oscillator is shown in Figure 10-3. Table 10-1 shows the capacitor selection for the Timer1 oscillator.

The Timer1 oscillator is shared with the system LP oscillator. Thus, Timer1 can use this mode only when the primary system clock is derived from the internal oscillator or when the oscillator is in the LP mode. The user must provide a software time delay to ensure proper oscillator start-up.

FIGURE 10-3: EXTERNAL COMPONENTS FOR THE TIMER1 LP OSCILLATOR



PIC18(L)F1XK22

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 ($F_{osc}/4$). 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{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

In addition to the expanded range of modes available through the CCP1CON register and ECCP1AS register, the ECCP module has two additional registers associated with Enhanced PWM operation and auto-shutdown features. They are:

- PWM1CON (Dead-band delay)
- PSTRCON (Output steering)

13.1 ECCP Outputs and Configuration

The enhanced CCP module may have up to four PWM outputs, depending on the selected operating mode. These outputs, designated P1A through P1D, are multiplexed with I/O pins on PORTC. The outputs that are active depend on the CCP operating mode selected. The pin assignments are summarized in Table 13-2.

To configure the I/O pins as PWM outputs, the proper PWM mode must be selected by setting the P1M<1:0> and CCP1M<3:0> bits. The appropriate TRISC direction bits for the port pins must also be set as outputs.

13.1.1 CCP MODULE AND TIMER RESOURCES

The CCP modules utilize Timers 1, 2 or 3, depending on the mode selected. Timer1 and Timer3 are available to modules in Capture or Compare modes, while Timer2 is available for modules in PWM mode.

TABLE 13-1: CCP MODE – TIMER RESOURCE

CCP/ECCP Mode	Timer Resource
Capture	Timer1 or Timer3
Compare	Timer1 or Timer3
PWM	Timer2

The assignment of a particular timer to a module is determined by the Timer-to-CCP enable bits in the T3CON register (Register 12-1). The interactions between the two modules are summarized in Figure 13-1. In Asynchronous Counter mode, the capture operation will not work reliably.

13.2 Capture Mode

In Capture mode, the CCPR1H:CCPR1L register pair captures the 16-bit value of the TMR1 or TMR3 registers when an event occurs on the corresponding CCP1 pin. An event is defined as one of the following:

- Every falling edge
- Every rising edge
- Every 4th rising edge
- Every 16th rising edge

The event is selected by the mode select bits, CCP1M<3:0> of the CCP1CON register. When a capture is made, the interrupt request flag bit, CCP1IF, is set; it must be cleared by software. If another capture occurs before the value in register CCPR1 is read, the old captured value is overwritten by the new captured value.

13.2.1 CCP PIN CONFIGURATION

In Capture mode, the appropriate CCP1 pin should be configured as an input by setting the corresponding TRIS direction bit.

Note: If the CCP1 pin is configured as an output, a write to the port can cause a capture condition.

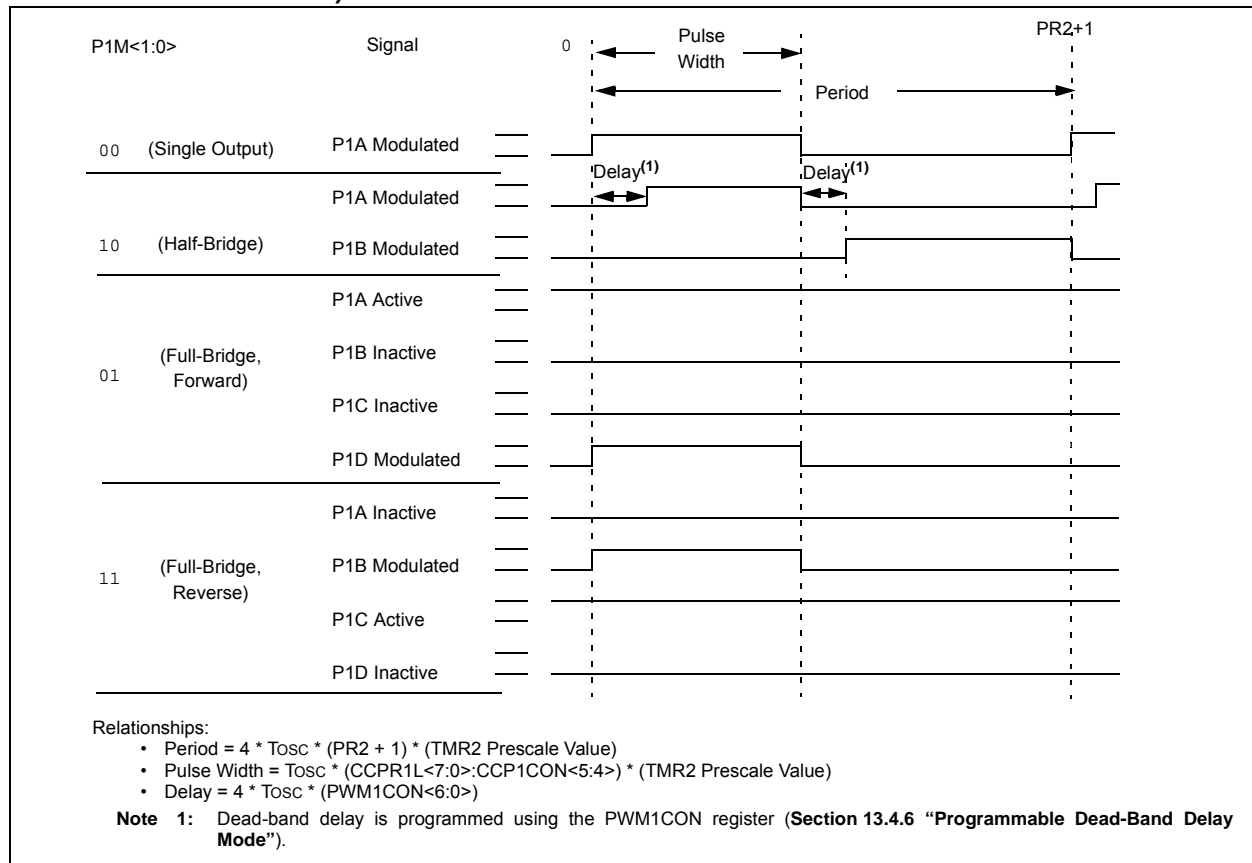
13.2.2 TIMER1/TIMER3 MODE SELECTION

The timers that are to be used with the capture feature (Timer1 and/or Timer3) must be running in Timer mode or Synchronized Counter mode. In Asynchronous Counter mode, the capture operation may not work. The timer to be used with each CCP module is selected in the T3CON register (see **Section 13.1.1 “CCP Module and Timer Resources”**).

13.2.3 SOFTWARE INTERRUPT

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep the CCP1IE interrupt enable bit clear to avoid false interrupts. The interrupt flag bit, CCP1IF, should also be cleared following any such change in operating mode.

FIGURE 13-4: EXAMPLE PWM (ENHANCED MODE) OUTPUT RELATIONSHIPS (ACTIVE-HIGH STATE)



13.4.7.1 Steering Synchronization

The STRSYNC bit of the PSTRCON register gives the user two selections of when the steering event will happen. When the STRSYNC bit is '0', the steering event will happen at the end of the instruction that writes to the PSTRCON register. In this case, the output signal at the P1<D:A> pins may be an incomplete PWM waveform. This operation is useful when the user firmware needs to immediately remove a PWM signal from the pin.

When the STRSYNC bit is '1', the effective steering update will happen at the beginning of the next PWM period. In this case, steering on/off the PWM output will always produce a complete PWM waveform.

Figures 13-17 and 13-18 illustrate the timing diagrams of the PWM steering depending on the STRSYNC setting.

FIGURE 13-17: EXAMPLE OF STEERING EVENT AT END OF INSTRUCTION (STRSYNC = 0)

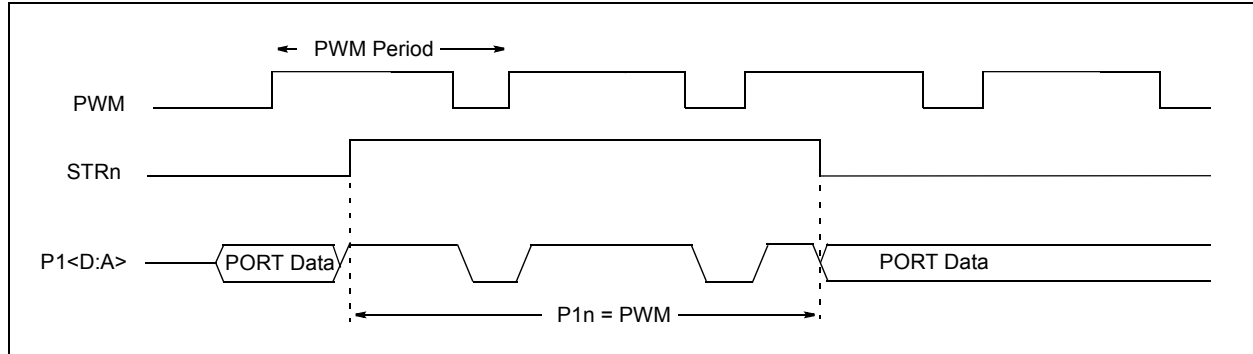
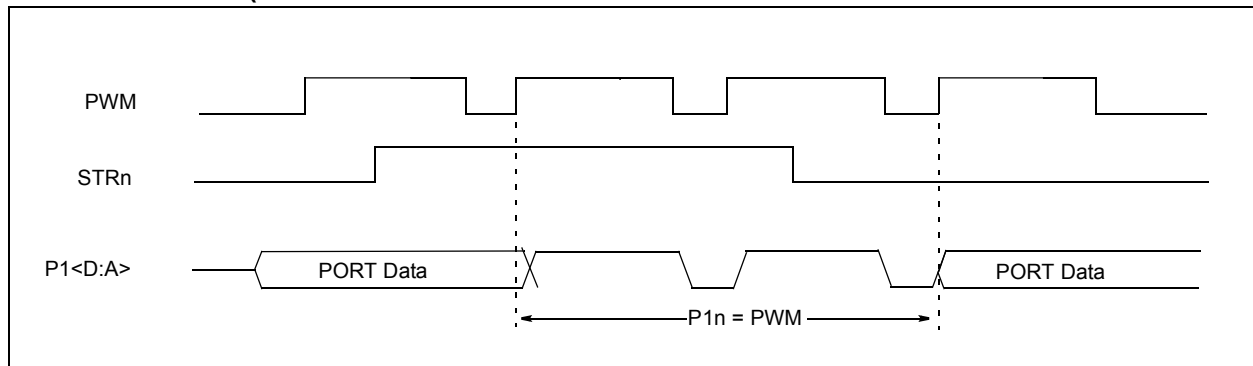


FIGURE 13-18: EXAMPLE OF STEERING EVENT AT BEGINNING OF INSTRUCTION (STRSYNC = 1)



14.2.8 OPERATION IN POWER-MANAGED MODES

In SPI Master mode, module clocks may be operating at a different speed than when in Full Power mode; in the case of the Sleep mode, all clocks are halted.

In all Idle modes, a clock is provided to the peripherals. That clock could be from the primary clock source, the secondary clock (Timer1 oscillator at 32.768 kHz) or the INTOSC source. See **Section 18.0 “Power-Managed Modes”** for additional information.

In most cases, the speed that the master clocks SPI data is not important; however, this should be evaluated for each system.

When MSSP interrupts are enabled, after the master completes sending data, an MSSP interrupt will wake the controller:

- From Sleep, in Slave mode
- From Idle, in Slave or Master mode

If an exit from Sleep or Idle mode is not desired, MSSP interrupts should be disabled.

In SPI Master mode, when the Sleep mode is selected, all module clocks are halted and the transmission/reception will remain in that state until the device wakes. After the device returns to Run mode, the module will resume transmitting and receiving data.

In SPI Slave mode, the SPI Transmit/Receive Shift register operates asynchronously to the device. This allows the device to be placed in any Power-Managed mode and data to be shifted into the SPI

Transmit/Receive Shift register. When all eight bits have been received, the MSSP interrupt flag bit will be set and if enabled, will wake the device.

14.2.9 EFFECTS OF A RESET

A Reset disables the MSSP module and terminates the current transfer.

14.2.10 BUS MODE COMPATIBILITY

Table 14-1 shows the compatibility between the standard SPI modes and the states of the CKP and CKE control bits.

TABLE 14-1: SPI BUS MODES

Standard SPI Mode Terminology	Control Bits State	
	CKP	CKE
0, 0	0	1
0, 1	0	0
1, 0	1	1
1, 1	1	0

There is also an SMP bit which controls when the data is sampled.

TABLE 14-2: REGISTERS ASSOCIATED WITH SPI OPERATION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RABIE	TMR0IF	INT0IF	RABIF	245
IPR1	—	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	248
PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	248
PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	248
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	248
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	248
SSPBUF	SSP Receive Buffer/Transmit Register								246
SSPCON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	246
SSPSTAT	SMP	CKE	D/Ā	P	S	R/Ŵ	UA	BF	246

Legend: Shaded cells are not used by the MSSP in SPI mode.

14.3.6 MASTER MODE

Master mode is enabled by setting and clearing the appropriate SSPM bits in SSPCON1 and by setting the SSPEN bit. In Master mode, the SCL and SDA lines are manipulated by the MSSP hardware.

Master mode of operation is supported by interrupt generation on the detection of the Start and Stop conditions. The Stop (P) and Start (S) bits are cleared from a Reset or when the MSSP module is disabled. Control of the I²C bus may be taken when the P bit is set, or the bus is Idle, with both the S and P bits clear.

In Firmware Controlled Master mode, user code conducts all I²C bus operations based on Start and Stop bit conditions.

Once Master mode is enabled, the user has six options.

1. Assert a Start condition on SDA and SCL.
2. Assert a Repeated Start condition on SDA and SCL.
3. Write to the SSPBUF register initiating transmission of data/address.
4. Configure the I²C port to receive data.
5. Generate an Acknowledge condition at the end of a received byte of data.
6. Generate a Stop condition on SDA and SCL.

Note: The MSSP module, when configured in I²C Master mode, does not allow queueing of events. For instance, the user is not allowed to initiate a Start condition and immediately write the SSPBUF register to initiate transmission before the Start condition is complete. In this case, the SSPBUF will not be written to and the WCOL bit will be set, indicating that a write to the SSPBUF did not occur.

The following events will cause the SSP Interrupt Flag bit, SSPIF, to be set (SSP interrupt, if enabled):

- Start condition
- Stop condition
- Data transfer byte transmitted/received
- Acknowledge transmit
- Repeated Start

FIGURE 14-16: MSSP BLOCK DIAGRAM (I²C MASTER MODE)

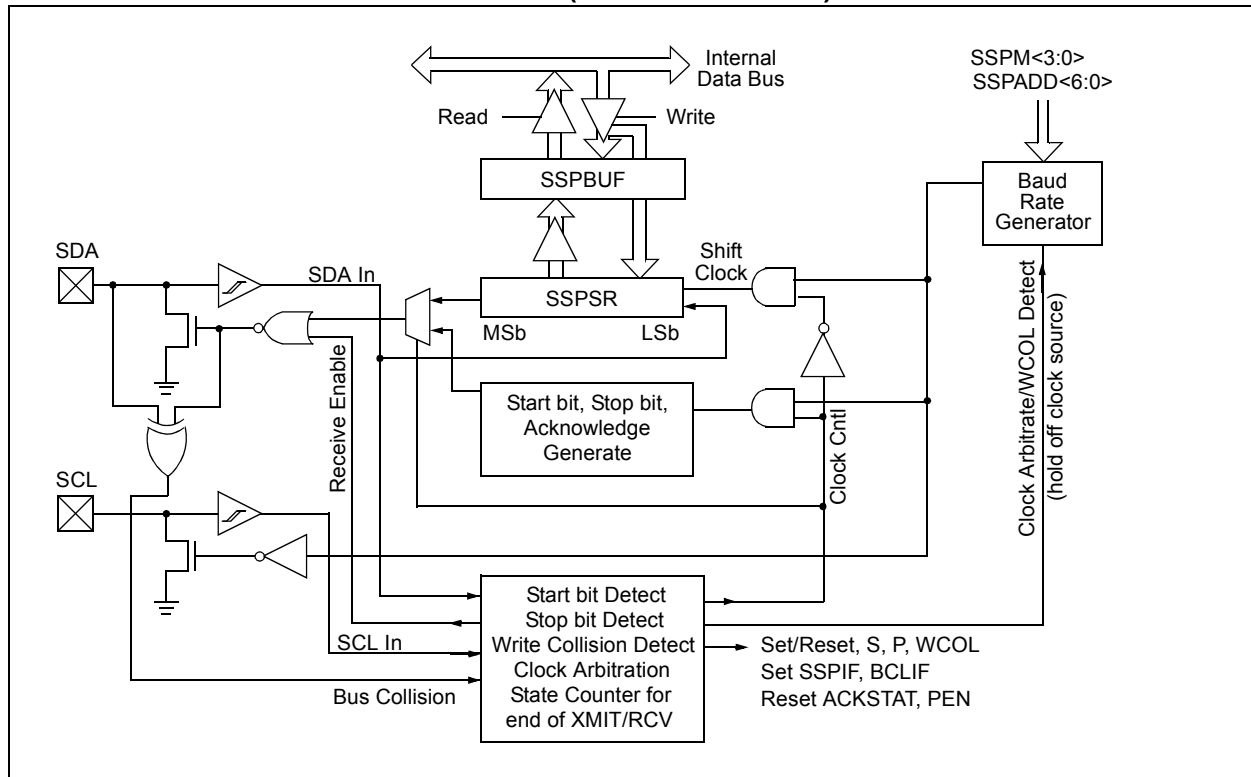


FIGURE 14-25: BUS COLLISION TIMING FOR TRANSMIT AND ACKNOWLEDGE

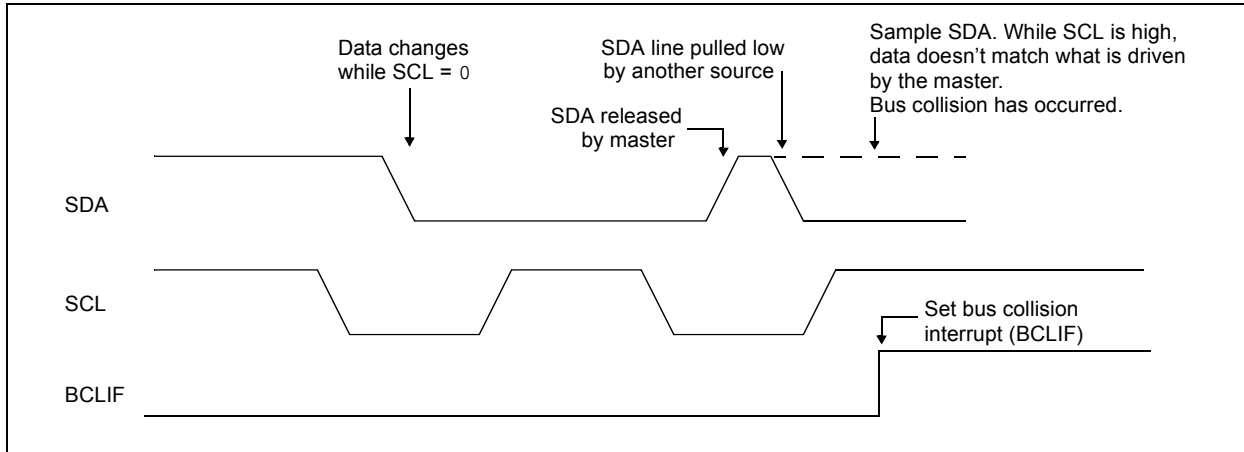


TABLE 15-5: BAUD RATES FOR ASYNCHRONOUS MODES (CONTINUED)

BAUD RATE	SYNC = 0, BRGH = 1, BRG16 = 1 or SYNC = 1, BRG16 = 1											
	Fosc = 48.000 MHz			Fosc = 18.432 MHz			Fosc = 12.000 MHz			Fosc = 11.0592 MHz		
	Actual Rate	% Error	SPBRGH :SPBRG (decimal)	Actual Rate	% Error	SPBRGH :SPBRG (decimal)	Actual Rate	% Error	SPBRGH :SPBRG (decimal)	Actual Rate	% Error	SPBRGH :SPBRG (decimal)
300	300	0.00	39999	300.0	0.00	15359	300	0.00	9999	300.0	0.00	9215
1200	1200	0.00	9999	1200	0.00	3839	1200	0.00	2499	1200	0.00	2303
2400	2400	0.00	4999	2400	0.00	1919	2400	0.00	1249	2400	0.00	1151
9600	9600	0.00	1249	9600	0.00	479	9615	0.16	311	9600	0.00	287
10417	10417	0.00	1151	10425	0.08	441	10417	0.00	287	10433	0.16	264
19.2k	19.20k	0.00	624	19.20k	0.00	239	19.23k	0.16	155	19.20k	0.00	143
57.6k	57.69k	0.16	207	57.60k	0.00	79	57.69k	0.16	51	57.60k	0.00	47
115.2k	115.38k	0.16	103	115.2k	0.00	39	115.38k	0.16	25	115.2k	0.00	23

BAUD RATE	SYNC = 0, BRGH = 1, BRG16 = 1 or SYNC = 1, BRG16 = 1											
	Fosc = 8.000 MHz			Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 1.000 MHz		
	Actual Rate	% Error	SPBRGH :SPBRG (decimal)	Actual Rate	% Error	SPBRGH :SPBRG (decimal)	Actual Rate	% Error	SPBRGH :SPBRG (decimal)	Actual Rate	% Error	SPBRGH :SPBRG (decimal)
300	300.0	0.00	6666	300.0	0.01	3332	300.0	0.00	3071	300.1	0.04	832
1200	1200	-0.02	1666	1200	0.04	832	1200	0.00	767	1202	0.16	207
2400	2401	0.04	832	2398	0.08	416	2400	0.00	383	2404	0.16	103
9600	9615	0.16	207	9615	0.16	103	9600	0.00	95	9615	0.16	25
10417	10417	0.00	191	10417	0.00	95	10473	0.53	87	10417	0.00	23
19.2k	19.23k	0.16	103	19.23k	0.16	51	19.20k	0.00	47	19.23k	0.16	12
57.6k	57.14k	-0.79	34	58.82k	2.12	16	57.60k	0.00	15	—	—	—
115.2k	117.6k	2.12	16	111.1k	-3.55	8	115.2k	0.00	7	—	—	—

18.1.3 MULTIPLE FUNCTIONS OF THE SLEEP COMMAND

The power-managed mode that is invoked with the `SLEEP` instruction is determined by the setting of the `IDLEN` bit of the `OSCCON` register at the time the instruction is executed. All clocks stop and minimum power is consumed when `SLEEP` is executed with the `IDLEN` bit cleared. The system clock continues to supply a clock to the peripherals but is disconnected from the CPU when `SLEEP` is executed with the `IDLEN` bit set.

18.2 Run Modes

In the Run modes, clocks to both the core and peripherals are active. The difference between these modes is the clock source.

18.2.1 PRI_RUN MODE

The `PRI_RUN` mode is the normal, full power execution mode of the microcontroller. This is also the default mode upon a device Reset, unless Two-Speed Start-up is enabled (see **Section 2.11 “Two-Speed Start-up Mode”** for details). In this mode, the device operated off the oscillator defined by the `FOSC` bits of the `CONFIG` Configuration register.

18.2.2 SEC_RUN MODE

In `SEC_RUN` mode, the CPU and peripherals are clocked from the secondary external oscillator. This gives users the option of lower power consumption while still using a high accuracy clock source.

`SEC_RUN` mode is entered by setting the `SCS<1:0>` bits of the `OSCCON` register to '01'. When `SEC_RUN` mode is active all of the following are true:

- The main clock source is switched to the secondary external oscillator
- Primary external oscillator is shut down
- `T1RUN` bit of the `T1CON` register is set
- `OSTS` bit is cleared.

Note: The secondary external oscillator should already be running prior to entering `SEC_RUN` mode. If the `T1OSCEN` bit is not set when the `SCS<1:0>` bits are set to '01', entry to `SEC_RUN` mode will not occur until `T1OSCEN` bit is set and secondary external oscillator is ready.

18.2.3 RC_RUN MODE

In `RC_RUN` mode, the CPU and peripherals are clocked from the internal oscillator. In this mode, the primary external oscillator is shut down. `RC_RUN` mode provides the best power conservation of all the Run modes when the `LFINTOSC` is the system clock.

`RC_RUN` mode is entered by setting the `SCS1` bit. When the clock source is switched from the primary oscillator to the internal oscillator, the primary oscillator is shut down and the `OSTS` bit is cleared. The `IRCF` bits may be modified at any time to immediately change the clock speed.

PIC18(L)F1XK22

22.4 Brown-out Reset (BOR)

PIC18(L)F1XK22 devices implement a BOR circuit that provides the user with a number of configuration and power-saving options. The BOR is controlled by the BORV<1:0> and BOREN<1:0> bits of the CONFIG2L Configuration register. There are a total of four BOR configurations which are summarized in Table 22-1.

The BOR threshold is set by the BORV<1:0> bits. If BOR is enabled (any values of BOREN<1:0>, except '00'), any drop of VDD below VBOR for greater than TBOR will reset the device. A Reset may or may not occur if VDD falls below VBOR for less than TBOR. The chip will remain in Brown-out Reset until VDD rises above VBOR.

If the Power-up Timer is enabled, it will be invoked after VDD rises above VBOR; it then will keep the chip in Reset for an additional time delay, TPWRT. If VDD drops below VBOR while the Power-up Timer is running, the chip will go back into a Brown-out Reset and the Power-up Timer will be initialized. Once VDD rises above VBOR, the Power-up Timer will execute the additional time delay.

BOR and the Power-on Timer (PWRT) are independently configured. Enabling BOR Reset does not automatically enable the PWRT.

22.4.1 SOFTWARE ENABLED BOR

When BOREN<1:0> = 01, the BOR can be enabled or disabled by the user in software. This is done with the SBOREN control bit of the RCON register. Setting SBOREN enables the BOR to function as previously described. Clearing SBOREN disables the BOR entirely. The SBOREN bit operates only in this mode; otherwise it is read as '0'.

Placing the BOR under software control gives the user the additional flexibility of tailoring the application to its environment without having to reprogram the device to change BOR configuration. It also allows the user to tailor device power consumption in software by eliminating the incremental current that the BOR consumes. While the BOR current is typically very small, it may have some impact in low-power applications.

Note: Even when BOR is under software control, the BOR Reset voltage level is still set by the BORV<1:0> Configuration bits. It cannot be changed by software.

22.4.2 DETECTING BOR

When BOR is enabled, the $\overline{\text{BOR}}$ bit always resets to '0' on any BOR or POR event. This makes it difficult to determine if a BOR event has occurred just by reading the state of $\overline{\text{BOR}}$ alone. A more reliable method is to simultaneously check the state of both $\overline{\text{POR}}$ and $\overline{\text{BOR}}$. This assumes that the $\overline{\text{POR}}$ and $\overline{\text{BOR}}$ bits are reset to '1' by software immediately after any POR event. If BOR is '0' while POR is '1', it can be reliably assumed that a BOR event has occurred.

22.4.3 DISABLING BOR IN SLEEP MODE

When BOREN<1:0> = 10, the BOR remains under hardware control and operates as previously described. Whenever the device enters Sleep mode, however, the BOR is automatically disabled. When the device returns to any other operating mode, BOR is automatically re-enabled.

This mode allows for applications to recover from brown-out situations, while actively executing code, when the device requires BOR protection the most. At the same time, it saves additional power in Sleep mode by eliminating the small incremental BOR current.

TABLE 22-1: BOR CONFIGURATIONS

BOR Configuration		Status of SBOREN (RCON<6>)	BOR Operation
BOREN1	BOREN0		
0	0	Unavailable	BOR disabled; must be enabled by reprogramming the Configuration bits.
0	1	Available	BOR enabled by software; operation controlled by SBOREN.
1	0	Unavailable	BOR enabled by hardware in Run and Idle modes, disabled during Sleep mode.
1	1	Unavailable	BOR enabled by hardware; must be disabled by reprogramming the Configuration bits.

CLRF		Clear f						
Syntax:	CLRF f{,a}							
Operands:	$0 \leq f \leq 255$ $a \in [0,1]$							
Operation:	$000h \rightarrow f$ $1 \rightarrow Z$							
Status Affected:	Z							
Encoding:	<table border="1"><tr><td>0110</td><td>101a</td><td>ffff</td><td>ffff</td></tr></table>				0110	101a	ffff	ffff
0110	101a	ffff	ffff					
Description:	<p>Clears the contents of the specified register.</p> <p>If 'a' is '0', the Access Bank is selected.</p> <p>If 'a' is '1', the BSR is used to select the GPR bank (default).</p> <p>If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \leq 95$ (5Fh). See Section 24.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode” for details.</p>							
Words:	1							
Cycles:	1							
Q Cycle Activity:								
	Q1	Q2	Q3	Q4				
	Decode	Read register 'f'	Process Data	Write register 'f'				

Example: CLRF FLAG_REG, 1

Before Instruction
FLAG_REG = 5Ah
After Instruction
FLAG_REG = 00h

CLRWDT		Clear Watchdog Timer						
Syntax:	CLRWDT							
Operands:	None							
Operation:	000h → WDT, 000h → WDT postscaler, 1 → \overline{TO} , 1 → \overline{PD}							
Status Affected:	\overline{TO} , \overline{PD}							
Encoding:	<table border="1"><tr><td>0000</td><td>0000</td><td>0000</td><td>0100</td></tr></table>				0000	0000	0000	0100
0000	0000	0000	0100					
Description:	CLRWDT instruction resets the Watchdog Timer. It also resets the postscaler of the WDT. Status bits, \overline{TO} and \overline{PD} , are set.							
Words:	1							
Cycles:	1							
Q Cycle Activity:								
	Q1	Q2	Q3	Q4				
	Decode	No operation	Process Data	No operation				

Example: CLRWDT

Before Instruction
WDT Counter = ?
After Instruction
WDT Counter = 00h
WDT Postscaler = 0
 \overline{TO} = 1
 \overline{PD} = 1

PIC18(L)F1XK22

MOVSS Move Indexed to Indexed

Syntax: MOVSS [z_s], [z_d]

Operands: 0 ≤ z_s ≤ 127
0 ≤ z_d ≤ 127

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

Status Affected: None

Encoding:

1110	1011	1zzz	zzzz _s
1111	xxxx	xzzz	zzzz _d

1st word (source)

2nd word (dest.)

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

Words: 2

Cycles: 2

Q Cycle Activity:

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

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

Before Instruction

FSR2 = 80h

Contents of 85h = 33h

Contents of 86h = 11h

After Instruction

FSR2 = 80h

Contents of 85h = 33h

Contents of 86h = 33h

PUSHL Store Literal at FSR2, Decrement FSR2

Syntax: PUSHL k

Operands: 0 ≤ k ≤ 255

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

Status Affected: None

Encoding:

1110	1010	kkkk	kkkk
------	------	------	------

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

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: PUSHL 08h

Before Instruction

FSR2H:FSR2L = 01ECh

Memory (01ECh) = 00h

After Instruction

FSR2H:FSR2L = 01EBh

Memory (01ECh) = 08h

PIC18(L)F1XK22

26.2 Standard Operating Conditions

The standard operating conditions for any device are defined as:

Operating Voltage: $V_{DDMIN} \leq V_{DD} \leq V_{DDMAX}$

Operating Temperature: $T_{A_MIN} \leq T_A \leq T_{A_MAX}$

V_{DD} — Operating Supply Voltage⁽¹⁾

PIC18LF1XK22

V _{DDMIN} (Fosc ≤ 16 MHz).....	+1.8V
V _{DDMIN} (Fosc ≤ 20 MHz).....	+2.0V
V _{DDMIN} (Fosc ≤ 64 MHz).....	+3.0V
V _{DDMAX}	+3.6V

PIC18F1XK22

V _{DDMIN} (Fosc ≤ 20 MHz).....	+2.3V
V _{DDMIN} (Fosc ≤ 64 MHz).....	+3.0V
V _{DDMAX}	+5.5V

T_A — Operating Ambient Temperature Range

Industrial Temperature

T _{A_MIN}	-40°C
T _{A_MAX}	+85°C

Extended Temperature

T _{A_MIN}	-40°C
T _{A_MAX}	+125°C

Note 1: See Parameter D001, DC Characteristics: Supply Voltage.