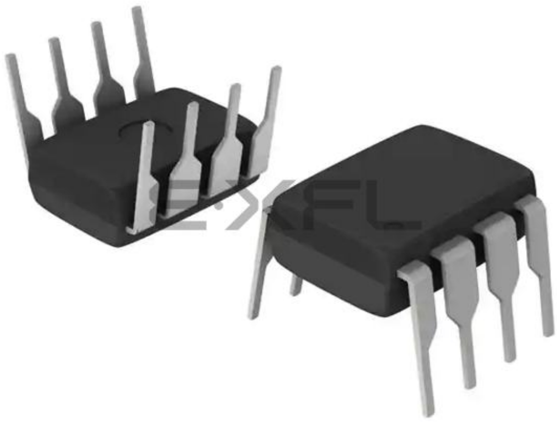


Welcome to E-XFL.COM

## 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 | 32MHz |
| Connectivity | I²C, LINbus, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, POR, PWM, WDT |
| Number of I/O | 6 |
| Program Memory Size | 3.5KB (2K x 14) |
| Program Memory Type | FLASH |
| EEPROM Size | 256 x 8 |
| RAM Size | 128 x 8 |
| Voltage - Supply (Vcc/Vdd) | 1.8V ~ 3.6V |
| Data Converters | A/D 4x10b; D/A 1x5b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Through Hole |
| Package / Case | 8-DIP (0.300", 7.62mm) |
| Supplier Device Package | 8-PDIP |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic12lf1822-i-p |

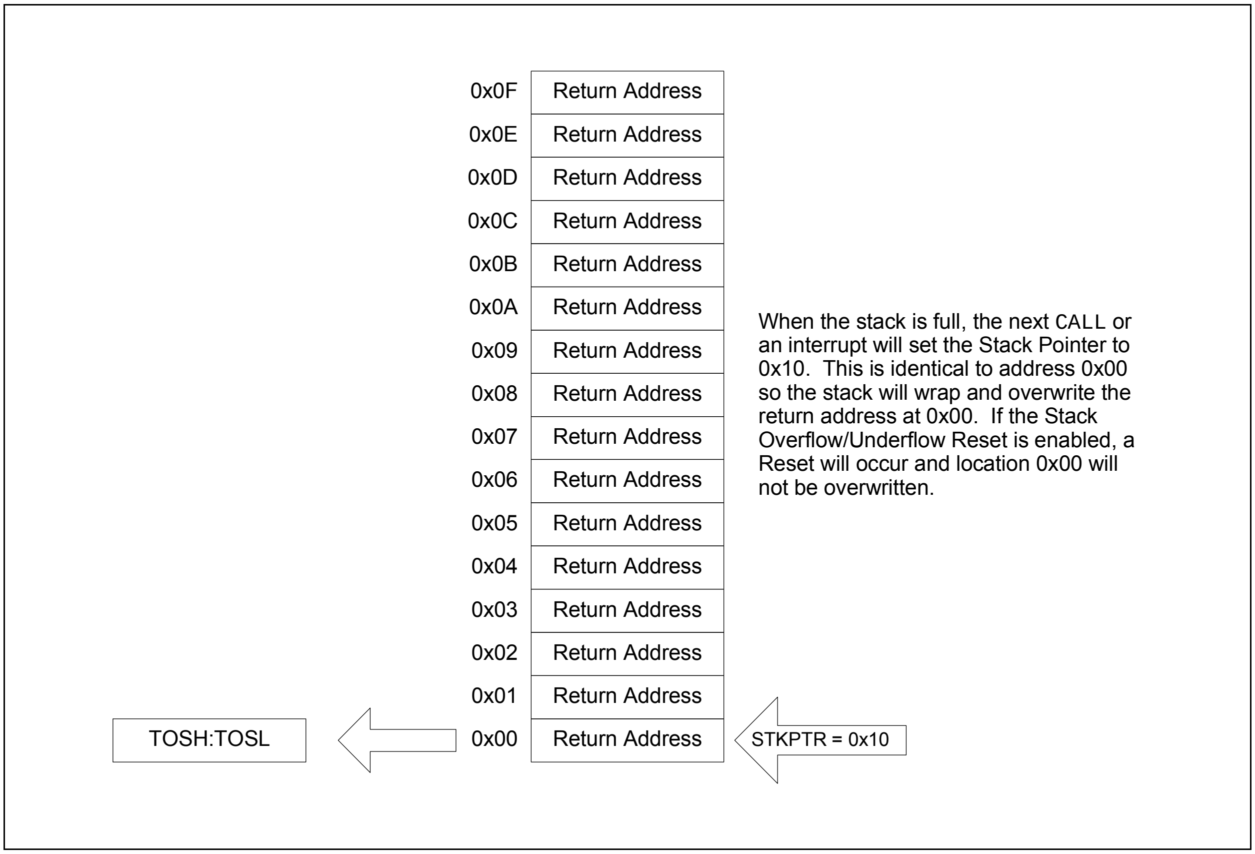
**TABLE 3-8: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 31** | | | | | | | | | | | |
| F80h(1) | INDF0 | Addressing this location uses contents of FSR0H/FSR0L to address data memory (not a physical register) | | | | | | | | xxxx xxxx | xxxx xxxx |
| F81h(1) | INDF1 | Addressing this location uses contents of FSR1H/FSR1L to address data memory (not a physical register) | | | | | | | | xxxx xxxx | xxxx xxxx |
| F82h(1) | PCL | Program Counter (PC) Least Significant Byte | | | | | | | | 0000 0000 | 0000 0000 |
| F83h(1) | STATUS | — | — | — | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | ---1 1000 | ---q quuu |
| F84h(1) | FSR0L | Indirect Data Memory Address 0 Low Pointer | | | | | | | | 0000 0000 | uuuu uuuu |
| F85h(1) | FSR0H | Indirect Data Memory Address 0 High Pointer | | | | | | | | 0000 0000 | 0000 0000 |
| F86h(1) | FSR1L | Indirect Data Memory Address 1 Low Pointer | | | | | | | | 0000 0000 | uuuu uuuu |
| F87h(1) | FSR1H | Indirect Data Memory Address 1 High Pointer | | | | | | | | 0000 0000 | 0000 0000 |
| F88h(1) | BSR | — | — | — | BSR<4:0> | | | | | ---0 0000 | ---0 0000 |
| F89h(1) | WREG | Working Register | | | | | | | | 0000 0000 | uuuu uuuu |
| F8Ah(1) | PCLATH | — | Write Buffer for the upper 7 bits of the Program Counter | | | | | | | -000 0000 | -000 0000 |
| F8Bh(1) | INTCON | GIE | PEIE | TMR0IE | INTE | IOCIE | TMR0IF | INTF | IOCIF | 0000 000x | 0000 000u |
| F8Ch — FE3h | — | Unimplemented | | | | | | | | — | — |
| FE4h | STATUS_ SHAD | — | — | — | — | — | Z_SHAD | DC_SHAD | C_SHAD | ---- -xxx | ---- -uuu |
| FE5h | WREG_ SHAD | Working Register Shadow | | | | | | | | 0000 0000 | uuuu uuuu |
| FE6h | BSR_ SHAD | — | — | — | Bank Select Register Shadow | | | | | ---x xxxx | ---u uuuu |
| FE7h | PCLATH_ SHAD | — | Program Counter Latch High Register Shadow | | | | | | | -xxx xxxx | uuuu uuuu |
| FE8h | FSR0L_ SHAD | Indirect Data Memory Address 0 Low Pointer Shadow | | | | | | | | xxxx xxxx | uuuu uuuu |
| FE9h | FSR0H_ SHAD | Indirect Data Memory Address 0 High Pointer Shadow | | | | | | | | xxxx xxxx | uuuu uuuu |
| FEAh | FSR1L_ SHAD | Indirect Data Memory Address 1 Low Pointer Shadow | | | | | | | | xxxx xxxx | uuuu uuuu |
| FEBh | FSR1H_ SHAD | Indirect Data Memory Address 1 High Pointer Shadow | | | | | | | | xxxx xxxx | uuuu uuuu |
| FECh | — | Unimplemented | | | | | | | | — | — |
| FEDh | STKPTR | — | — | — | Current Stack pointer | | | | | ---1 1111 | ---1 1111 |
| FEEh | TOSL | Top-of-Stack Low byte | | | | | | | | xxxx xxxx | uuuu uuuu |
| FEFh | TOSH | — | Top-of-Stack High byte | | | | | | | -xxx xxxx | -uuu uuuu |

**Legend:** x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, r = reserved.
Shaded locations are unimplemented, read as '0'.

**Note 1:** These registers can be addressed from any bank.
**2:** PIC16(L)F1823 only.
**3:** Unimplemented. Read as '1'.
**4:** PIC12(L)F1822 only.

**FIGURE 3-7: ACCESSING THE STACK EXAMPLE 4**




### 3.4.2 OVERFLOW/UNDERFLOW RESET

If the STVREN bit in Configuration Word 2 is programmed to ‘1’, the device will be reset if the stack is PUSHed beyond the sixteenth level or POPed beyond the first level, setting the appropriate bits (STKOVF or STKUNF, respectively) in the PCON register.

## 3.5 Indirect Addressing

The INDFn registers are not physical registers. Any instruction that accesses an INDFn register actually accesses the register at the address specified by the File Select Registers (FSR). If the FSRn address specifies one of the two INDFn registers, the read will return ‘0’ and the write will not occur (though Status bits may be affected). The FSRn register value is created by the pair FSRnH and FSRnL.

The FSR registers form a 16-bit address that allows an addressing space with 65536 locations. These locations are divided into three memory regions:

- Traditional Data Memory
- Linear Data Memory
- Program Flash Memory

### REGISTER 4-1: CONFIGURATION WORD 1

| R/P-1/1 | R/P-1/1 | R/P-1/1 | R/P-1/1 | R/P-1/1 | R/P-1/1 |
|---|---|---|---|---|---|
| FCMEN | IESO | $\overline{\text{CLKOUTEN}}$ | BOREN<1:0> | | $\overline{\text{CPD}}$ |
| bit 13 | | | | | bit 8 |

| R/P-1/1 | R/P-1/1 | R/P-1/1 | R/P-1/1 | R/P-1/1 | R/P-1/1 | R/P-1/1 | R/P-1/1 |
|---|---|---|---|---|---|---|---|
| $\overline{\text{CP}}$ | MCLRE | $\overline{\text{PWRTE}}$ | WDTE<1:0> | | FOSC<2:0> | | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | P = Programmable bit | U = Unimplemented bit, read as ‘1’ |
| ‘0’ = Bit is cleared | ‘1’ = Bit is set | -n = Value when blank or after Bulk Erase |

bit 13 **FCMEN:** Fail-Safe Clock Monitor Enable bit
1 = Fail-Safe Clock Monitor is enabled
0 = Fail-Safe Clock Monitor is disabled

bit 12 **IESO:** Internal External Switchover bit
1 = Internal/External Switchover mode is enabled
0 = Internal/External Switchover mode is disabled

bit 11 **$\overline{\text{CLKOUTEN}}$:** Clock Out Enable bit
If FOSC configuration bits are set to LP, XT, HS modes:
This bit is ignored, CLKOUT function is disabled. Oscillator function on the CLKOUT pin
All other FOSC modes:
1 = CLKOUT function is disabled. I/O function on the CLKOUT pin
0 = CLKOUT function is enabled on the CLKOUT pin

bit 10-9 **BOREN<1:0>:** Brown-out Reset Enable bits(1)
11 = BOR enabled
10 = BOR enabled during operation and disabled in Sleep
01 = BOR controlled by SBOREN bit of the BORCON register
00 = BOR disabled

bit 8 **$\overline{\text{CPD}}$:** Data Code Protection bit(2)
1 = Data memory code protection is disabled
0 = Data memory code protection is enabled

bit 7 **$\overline{\text{CP}}$:** Code Protection bit(3)
1 = Program memory code protection is disabled
0 = Program memory code protection is enabled

bit 6 **MCLRE:** $\overline{\text{MCLR}}$/VPP Pin Function Select bit
If LVP bit = 1:
This bit is ignored.
If LVP bit = 0:
1 = $\overline{\text{MCLR}}$/VPP pin function is $\overline{\text{MCLR}}$; Weak pull-up enabled.
0 = $\overline{\text{MCLR}}$/VPP pin function is digital input; $\overline{\text{MCLR}}$ internally disabled; Weak pull-up under control of the WPUA register.

bit 5 **$\overline{\text{PWRTE}}$:** Power-up Timer Enable bit(1)
1 = PWRT disabled
0 = PWRT enabled

bit 4-3 **WDTE<1:0>:** Watchdog Timer Enable bit
11 = WDT enabled
10 = WDT enabled while running and disabled in Sleep
01 = WDT controlled by the SWDTEN bit in the WDTCON register
00 = WDT disabled

**Note 1:** Enabling Brown-out Reset does not automatically enable Power-up Timer.
**2:** The entire data EEPROM will be erased when the code protection is turned off during an erase.
**3:** The entire program memory will be erased when the code protection is turned off.

### REGISTER 6-1: CLKRCON: REFERENCE CLOCK CONTROL REGISTER

| R/W-0/0 | R/W-0/0 | R/W-1/1 | R/W-1/1 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| CLKREN | CLKROE | CLKRSLR | CLKRDC<1:0> | | CLKRDIV<2:0> | | |
| 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 **CLKREN:** Reference Clock Module Enable bit
1 = Reference clock module is enabled
0 = Reference clock module is disabled

bit 6 **CLKROE:** Reference Clock Output Enable bit(3)
1 = Reference clock output is enabled on CLKR pin
0 = Reference clock output disabled on CLKR pin

bit 5 **CLKRSLR:** Reference Clock Slew Rate Control limiting enable bit
1 = Slew rate limiting is enabled
0 = Slew rate limiting is disabled

bit 4-3 **CLKRDC<1:0>:** Reference Clock Duty Cycle bits
11 = Clock outputs duty cycle of 75%
10 = Clock outputs duty cycle of 50%
01 = Clock outputs duty cycle of 25%
00 = Clock outputs duty cycle of 0%

bit 2-0 **CLKRDIV<2:0>** Reference Clock Divider bits
111 = Base clock value divided by 128
110 = Base clock value divided by 64
101 = Base clock value divided by 32
100 = Base clock value divided by 16
011 = Base clock value divided by 8
010 = Base clock value divided by 4
001 = Base clock value divided by 2(1)
000 = Base clock value(2)

**Note 1:** In this mode, the 25% and 75% duty cycle accuracy will be dependent on the source clock duty cycle.

**2:** In this mode, the duty cycle will always be equal to the source clock duty cycle, unless a duty cycle of 0% is selected.

**3:** To route CLKR to pin, $\overline{\text{CLKOUTEN}}$ of Configuration Word 1 = 1 is required. $\overline{\text{CLKOUTEN}}$ of Configuration Word 1 = 0 will result in FOSC/4. See **Section 6.3 “Conflicts with the CLKR pin”** for details.

## 7.10 Determining the Cause of a Reset

Upon any Reset, multiple bits in the STATUS and PCON register are updated to indicate the cause of the Reset. Table 7-3 and Table 7-4 show the Reset conditions of these registers.

**TABLE 7-3: RESET STATUS BITS AND THEIR SIGNIFICANCE**

| STKOVF | STKUNF | $\overline{\text{RMCLR}}$ | $\overline{\text{RI}}$ | $\overline{\text{POR}}$ | $\overline{\text{BOR}}$ | $\overline{\text{TO}}$ | $\overline{\text{PD}}$ | Condition |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 0 | x | 1 | 1 | Power-on Reset |
| 0 | 0 | 1 | 1 | 0 | x | 0 | x | Illegal, $\overline{\text{TO}}$ is set on $\overline{\text{POR}}$ |
| 0 | 0 | 1 | 1 | 0 | x | x | 0 | Illegal, $\overline{\text{PD}}$ is set on $\overline{\text{POR}}$ |
| 0 | 0 | 1 | 1 | u | 0 | 1 | 1 | Brown-out Reset |
| u | u | u | u | u | u | 0 | u | WDT Reset |
| u | u | u | u | u | u | 0 | 0 | WDT Wake-up from Sleep |
| u | u | u | u | u | u | 1 | 0 | Interrupt Wake-up from Sleep |
| u | u | 0 | u | u | u | u | u | $\overline{\text{MCLR}}$ Reset during normal operation |
| u | u | 0 | u | u | u | 1 | 0 | $\overline{\text{MCLR}}$ Reset during Sleep |
| u | u | u | 0 | u | u | u | u | RESET Instruction Executed |
| 1 | u | u | u | u | u | u | u | Stack Overflow Reset (STVREN = 1) |
| u | 1 | u | u | u | u | u | u | Stack Underflow Reset (STVREN = 1) |

**TABLE 7-4: RESET CONDITION FOR SPECIAL REGISTERS(2)**

| Condition | Program Counter | STATUS Register | PCON Register |
|---|---|---|---|
| Power-on Reset | 0000h | ---1 1000 | 00-- 110x |
| $\overline{\text{MCLR}}$ Reset during normal operation | 0000h | ---u uuuu | uu-- 0uuu |
| $\overline{\text{MCLR}}$ Reset during Sleep | 0000h | ---1 0uuu | uu-- 0uuu |
| WDT Reset | 0000h | ---0 uuuu | uu-- uuuu |
| WDT Wake-up from Sleep | PC + 1 | ---0 0uuu | uu-- uuuu |
| Brown-out Reset | 0000h | ---1 1uuu | 00-- 11u0 |
| Interrupt Wake-up from Sleep | PC + 1(1) | ---1 0uuu | uu-- uuuu |
| RESET Instruction Executed | 0000h | ---u uuuu | uu-- u0uu |
| Stack Overflow Reset (STVREN = 1) | 0000h | ---u uuuu | 1u-- uuuu |
| Stack Underflow Reset (STVREN = 1) | 0000h | ---u uuuu | u1-- uuuu |

**Legend:** u = unchanged, x = unknown, - = unimplemented bit, reads as '0'.

**Note 1:** When the wake-up is due to an interrupt and Global Enable bit (GIE) is set, the return address is pushed on the stack and PC is loaded with the interrupt vector (0004h) after execution of PC + 1.

**2:** If a Status bit is not implemented, that bit will be read as '0'.

## 8.0 INTERRUPTS

The interrupt feature allows certain events to preempt normal program flow. Firmware is used to determine the source of the interrupt and act accordingly. Some interrupts can be configured to wake the MCU from Sleep mode.

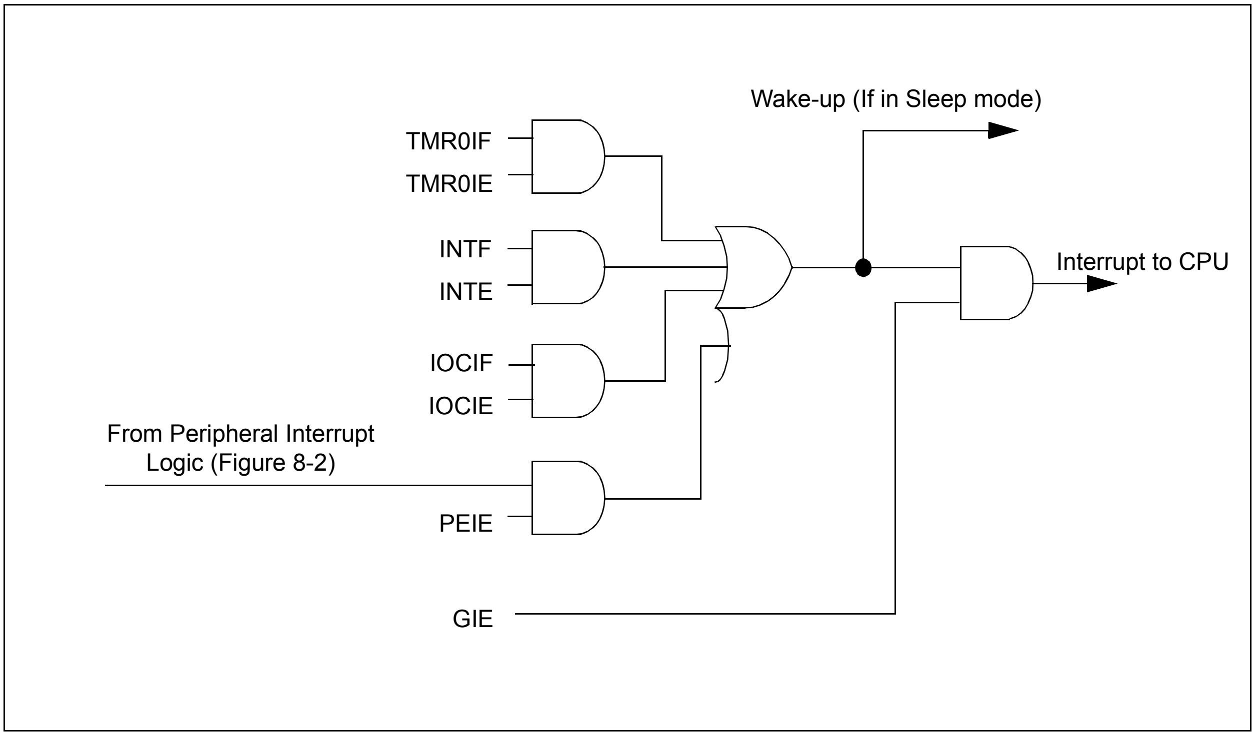
This chapter contains the following information for Interrupts:

- Operation
- Interrupt Latency
- Interrupts During Sleep
- INT Pin
- Automatic Context Saving

Many peripherals produce Interrupts. Refer to the corresponding chapters for details.

A block diagram of the interrupt logic is shown in Figure 8-1 and Figure 8-2.

**FIGURE 8-1: INTERRUPT LOGIC**



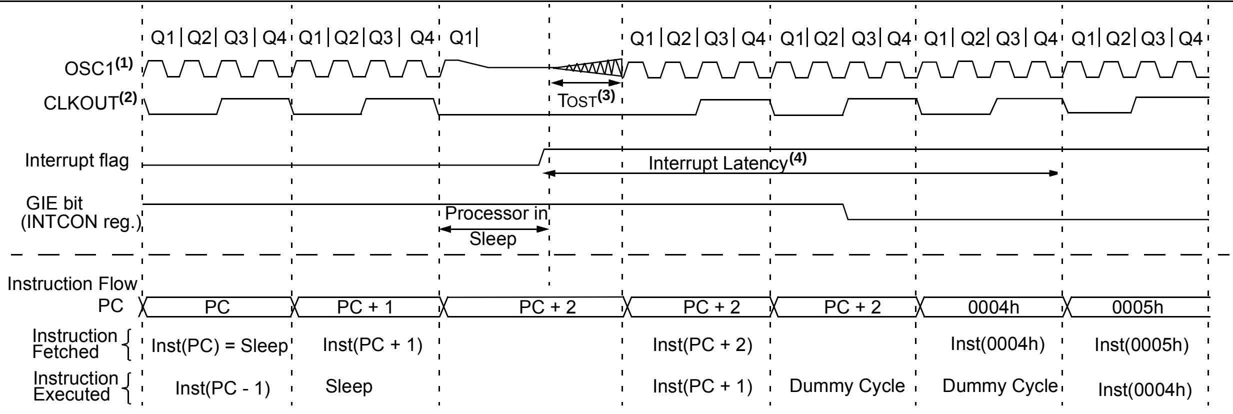
### 9.1.1 WAKE-UP USING INTERRUPTS

When global interrupts are disabled (GIE cleared) and any interrupt source has both its interrupt enable bit and interrupt flag bit set, one of the following will occur:

- If the interrupt occurs **before** the execution of a SLEEP instruction
  - SLEEP instruction will execute as a NOP.
  - WDT and WDT prescaler will not be cleared
  - $\overline{\text{TO}}$ bit of the STATUS register will not be set
  - $\overline{\text{PD}}$ bit of the STATUS register will not be cleared.
- If the interrupt occurs **during or after** the execution of a SLEEP instruction
  - SLEEP instruction will be completely executed
  - Device will immediately wake-up from Sleep
  - WDT and WDT prescaler will be cleared
  - $\overline{\text{TO}}$ bit of the STATUS register will be set
  - $\overline{\text{PD}}$ bit of the STATUS register will be cleared.

Even if the flag bits were checked before executing a SLEEP instruction, it may be possible for flag bits to become set before the SLEEP instruction completes. To determine whether a SLEEP instruction executed, test the $\overline{\text{PD}}$ bit. If the $\overline{\text{PD}}$ bit is set, the SLEEP instruction was executed as a NOP.

**FIGURE 9-1: WAKE-UP FROM SLEEP THROUGH INTERRUPT**




**Note 1:** XT, HS or LP Oscillator mode assumed.

**2:** CLKOUT is not available in XT, HS, or LP Oscillator modes, but shown here for timing reference.

**3:** TOST = 1024 TOSC (drawing not to scale). This delay applies only to XT, HS or LP Oscillator modes.

**4:** GIE = 1 assumed. In this case after wake-up, the processor calls the ISR at 0004h. If GIE = 0, execution will continue in-line.

## 12.3 PORTC Registers (PIC16(L)F1823 only)

PORTC is a 6-bit wide, bidirectional port. The corresponding data direction register is TRISC (Register 12-8). Setting a TRISC bit (= 1) will make the corresponding PORTC pin an input (i.e., put the corresponding output driver in a High-Impedance mode). 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). Example 12-2 shows how to initialize PORTC.

Reading the PORTC register (Register 12-7) reads the status of the pins, whereas writing to it will write to the PORT latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, this value is modified and then written to the PORT data latch (LATC).

The TRISC register (Register 12-8) controls the PORTC pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISC register are maintained set when using them as analog inputs. I/O pins configured as analog input always read ‘0’.

### 12.3.1 ANSELC REGISTER

The ANSELC register (Register 12-10) is used to configure the Input mode of an I/O pin to analog. Setting the appropriate ANSELC bit high will cause all digital reads on the pin to be read as ‘0’ and allow analog functions on the pin to operate correctly.

The state of the ANSELC bits has no affect on digital output functions. A pin with TRIS clear and ANSELC set will still operate as a digital output, but the Input mode will be analog. This can cause unexpected behavior when executing read-modify-write instructions on the affected port.

| Note: | The ANSELC register must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read ‘0’. |
|---|---|

**EXAMPLE 12-2: INITIALIZING PORTC**

```
BANKSEL PORTC      ;
CLRF    PORTC      ;Init PORTC
BANKSEL LATC       ;Data Latch
CLRF    LATC       ;
BANKSEL ANSELC
CLRF    ANSELC     ;Make RC<5:0> digital
BANKSEL TRISB      ;
MOVLW   B'00110000';Set RC<5:4> as inputs
                   ;and RC<3:0> as outputs
MOVWF   TRISC      ;
```

### 12.3.2 PORTC FUNCTIONS AND OUTPUT PRIORITIES

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

When multiple outputs are enabled, the actual pin control goes to the peripheral with the lowest number in the following lists.

Analog input and some digital input functions are not included in the list below. These input functions can remain active when the pin is configured as an output. Certain digital input functions override other port functions and are included in the priority list.

RC0

1. SCL (MSSP)
2. SCK (MSSP)

RC1

1. SDA (MSSP)

RC2

1. SDO (MSSP)
2. P1D

RC3

1. P1C

RC4

1. MDOUT
2. SRNQ
3. C2OUT
4. TX/CK
5. P1B

RC5

1. RX/DT
2. CCP1/P1A

## 21.1 Timer1 Operation

The Timer1 module is a 16-bit incrementing counter which is accessed through the TMR1H:TMR1L register pair. Writes to TMR1H or TMR1L directly update the counter.

When used with an internal clock source, the module is a timer and increments on every instruction cycle. When used with an external clock source, the module can be used as either a timer or counter and increments on every selected edge of the external source.

Timer1 is enabled by configuring the TMR1ON and TMR1GE bits in the T1CON and T1GCON registers, respectively. Table 21-1 displays the Timer1 enable selections.

**TABLE 21-1: TIMER1 ENABLE SELECTIONS**

| TMR1ON | TMR1GE | Timer1 Operation |
|---|---|---|
| 0 | 0 | Off |
| 0 | 1 | Off |
| 1 | 0 | Always On |
| 1 | 1 | Count Enabled |

## 21.2 Clock Source Selection

The TMR1CS<1:0> and T1OSCEN bits of the T1CON register are used to select the clock source for Timer1. Table 21-2 displays the clock source selections.

### 21.2.1 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.

When the FOSC internal clock source is selected, the Timer1 register value will increment by four counts every instruction clock cycle. Due to this condition, a 2 LSB error in resolution will occur when reading the Timer1 value. To utilize the full resolution of Timer1, an asynchronous input signal must be used to gate the Timer1 clock input.

The following asynchronous sources may be used:

- Asynchronous event on the T1G pin to Timer1 Gate
- C1 or C2 comparator input to Timer1 Gate

### 21.2.2 EXTERNAL CLOCK SOURCE

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

When enabled to count, Timer1 is incremented on the rising edge of the external clock input T1CKI or the capacitive sensing oscillator signal. Either of these external clock sources can be synchronized to the microcontroller system clock or they can run asynchronously.

When used as a timer with a clock oscillator, an external 32.768 kHz crystal can be used in conjunction with the dedicated internal oscillator circuit.

**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.

**TABLE 21-2: CLOCK SOURCE SELECTIONS**

| TMR1CS1 | TMR1CS0 | T1OSCEN | Clock Source |
|---|---|---|---|
| 0 | 1 | x | System Clock (FOSC) |
| 0 | 0 | x | Instruction Clock (FOSC/4) |
| 1 | 1 | x | Capacitive Sensing Oscillator |
| 1 | 0 | 0 | External Clocking on T1CKI Pin |
| 1 | 0 | 1 | Osc.Circuit On T1OSI/T1OSO Pins |

## 21.3 Timer1 Prescaler

Timer1 has four prescaler options allowing 1, 2, 4 or 8 divisions of the clock input. The T1CKPS bits of the T1CON register control the prescale counter. The prescale counter is not directly readable or writable; however, the prescaler counter is cleared upon a write to TMR1H or TMR1L.

## 21.4 Timer1 Oscillator

A dedicated low-power 32.768 kHz oscillator circuit is built-in between pins T1OSI (input) and T1OSO (amplifier output). This internal circuit is to be used in conjunction with an external 32.768 kHz crystal.

The oscillator circuit is enabled by setting the T1OSCEN bit of the T1CON register. The oscillator will continue to run during Sleep.

> **Note:** The oscillator requires a start-up and stabilization time before use. Thus, T1OSCEN should be set and a suitable delay observed prior to enabling Timer1.

## 21.5 Timer1 Operation in Asynchronous Counter Mode

If control bit $\overline{\text{T1SYNC}}$ of the T1CON register is set, the external clock input is not synchronized. The timer increments asynchronously to the internal phase clocks. If external clock source is selected then the timer will continue to run during Sleep and can generate an interrupt on overflow, which will wake-up the processor. However, special precautions in software are needed to read/write the timer (see **Section 21.5.1 "Reading and Writing Timer1 in Asynchronous Counter Mode"**).

> **Note:** When switching from synchronous to asynchronous operation, it is possible to skip an increment. When switching from asynchronous to synchronous operation, it is possible to produce an additional increment.

### 21.5.1 READING AND WRITING TIMER1 IN ASYNCHRONOUS COUNTER MODE

Reading TMR1H or TMR1L while the timer is running from an external asynchronous clock will ensure a valid read (taken care of in hardware). However, the user should keep in mind that reading the 16-bit timer in two 8-bit values itself, poses certain problems, since the timer may overflow between the reads.

For writes, it is recommended that the user simply stop the timer and write the desired values. A write contention may occur by writing to the timer registers, while the register is incrementing. This may produce an unpredictable value in the TMR1H:TMR1L register pair.

## 21.6 Timer1 Gate

Timer1 can be configured to count freely or the count can be enabled and disabled using Timer1 Gate circuitry. This is also referred to as Timer1 Gate Enable.

Timer1 Gate can also be driven by multiple selectable sources.

### 21.6.1 TIMER1 GATE ENABLE

The Timer1 Gate Enable mode is enabled by setting the TMR1GE bit of the T1GCON register. The polarity of the Timer1 Gate Enable mode is configured using the T1GPOL bit of the T1GCON register.

When Timer1 Gate Enable mode is enabled, Timer1 will increment on the rising edge of the Timer1 clock source. When Timer1 Gate Enable mode is disabled, no incrementing will occur and Timer1 will hold the current count. See Figure 21-3 for timing details.

**TABLE 21-3: TIMER1 GATE ENABLE SELECTIONS**

| T1CLK | T1GPOL | T1G | Timer1 Operation |
|---|---|---|---|
| ↑ | 0 | 0 | Counts |
| ↑ | 0 | 1 | Holds Count |
| ↑ | 1 | 0 | Holds Count |
| ↑ | 1 | 1 | Counts |

### 21.6.2 TIMER1 GATE SOURCE SELECTION

The Timer1 Gate source can be selected from one of four different sources. Source selection is controlled by the T1GSS bits of the T1GCON register. The polarity for each available source is also selectable. Polarity selection is controlled by the T1GPOL bit of the T1GCON register.

**TABLE 21-4: TIMER1 GATE SOURCES**

| T1GSS | Timer1 Gate Source |
|---|---|
| 00 | Timer1 Gate Pin |
| 01 | Overflow of Timer0<br>(TMR0 increments from FFh to 00h) |
| 10 | Comparator 1 Output sync_C1OUT<br>(optionally Timer1 synchronized output) |
| 11 | Comparator 2 Output sync_C2OUT<br>(optionally Timer1 synchronized output) |

### 24.3.6 OPERATION IN SLEEP MODE

In Sleep mode, the TMR2 register will not increment and the state of the module will not change. If the CCP1 pin is driving a value, it will continue to drive that value. When the device wakes up, TMR2 will continue from its previous state.

### 24.3.7 CHANGES IN SYSTEM CLOCK FREQUENCY

The PWM frequency is derived from the system clock frequency. Any changes in the system clock frequency will result in changes to the PWM frequency. See **Section 5.0 "Oscillator Module (With Fail-Safe Clock Monitor)"** for additional details.

### 24.3.8 EFFECTS OF RESET

Any Reset will force all ports to Input mode and the CCP registers to their Reset states.

### 24.3.9 ALTERNATE PIN LOCATIONS

This module incorporates I/O pins that can be moved to other locations with the use of the alternate pin function register, APFCON. To determine which pins can be moved and what their default locations are upon a Reset, see **Section 12.1 "Alternate Pin Function"** for more information.

**TABLE 24-8: SUMMARY OF REGISTERS ASSOCIATED WITH STANDARD PWM**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Register on Page |
|---|---|---|---|---|---|---|---|---|---|
| APFCON | RXDTSEL | SDOSEL | SSSEL | — | T1GSEL | TXCKSEL | P1BSEL(2) | CCP1SEL(2) | 114 |
| CCP1CON | P1M<1:0> | | DC1B<1:0> | | CCP1M<3:0> | | | | 213 |
| CCPR1L | Capture/Compare/PWM Register x Low Byte (LSB) | | | | | | | | 191 |
| INTCON | GIE | PEIE | TMR0IE | INTE | IOCIE | TMR0IF | INTF | IOCIF | 86 |
| PIE1 | TMR1GIE | ADIE | RCIE | TXIE | SSP1IE | CCP1IE | TMR2IE | TMR1IE | 87 |
| PIR1 | TMR1GIF | ADIF | RCIF | TXIF | SSP1IF | CCP1IF | TMR2IF | TMR1IF | 89 |
| PR2 | Timer2 Period Register | | | | | | | | 176* |
| T2CON | — | T2OUTPS<3:0> | | | | TMR2ON | T2CKPS<:0>1 | | 178 |
| TMR2 | Timer2 Module Register | | | | | | | | 176* |
| TRISA | — | — | TRISA5 | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | 117 |
| TRISC(1) | — | — | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 121 |

**Legend:** — = Unimplemented location, read as '0'. Shaded cells are not used by the PWM.
* Page provides register information.
**Note 1:** PIC16(L)F1823 only.
**2:** PIC12(L)F1822 only.

### 24.4.2 FULL-BRIDGE MODE (PIC16(L)F1823 ONLY)

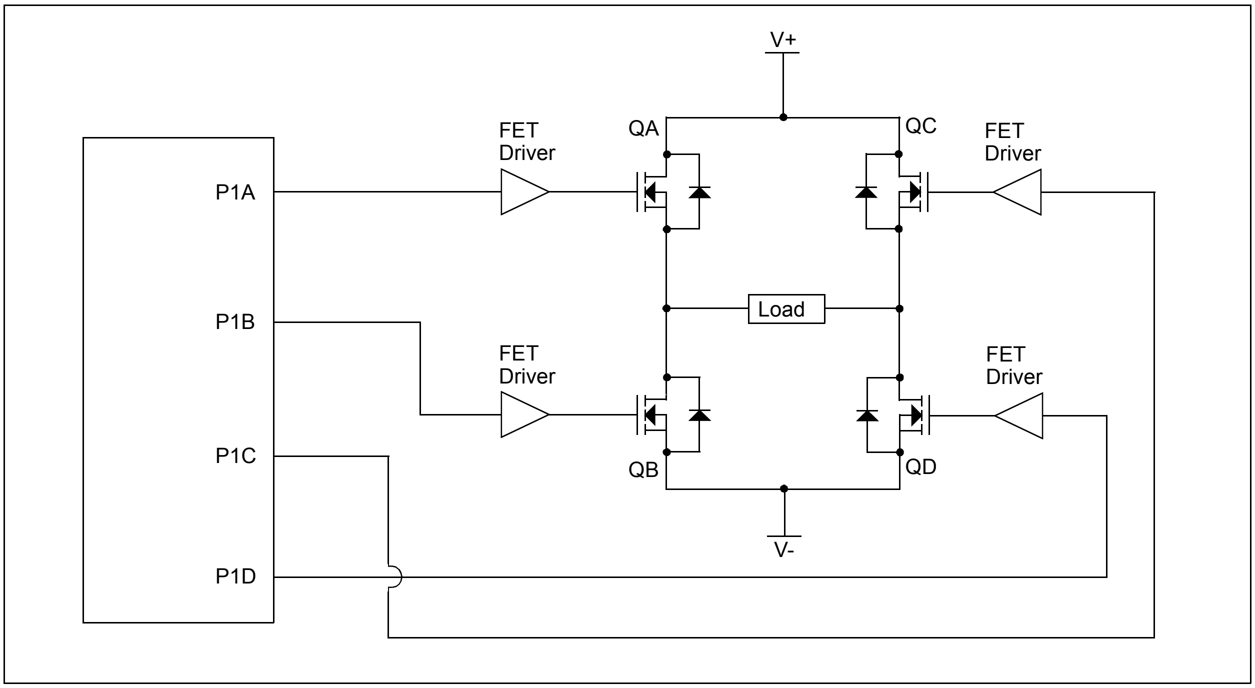
In Full-Bridge mode, all four pins are used as outputs. An example of Full-Bridge application is shown in Figure 24-10.

In the Forward mode, pin CCP1/P1A is driven to its active state, pin P1D is modulated, while P1B and P1C will be driven to their inactive state as shown in Figure 24-11.

In the Reverse mode, P1C is driven to its active state, pin P1B is modulated, while P1A and P1D will be driven to their inactive state as shown Figure 24-11.

P1A, P1B, P1C and P1D outputs are multiplexed with the PORT data latches. The associated TRIS bits must be cleared to configure the P1A, P1B, P1C and P1D pins as outputs.

**FIGURE 24-10: EXAMPLE OF FULL-BRIDGE APPLICATION**



#### 25.6.13.1 Bus Collision During a Start Condition

During a Start condition, a bus collision occurs if:

a) SDA or SCL are sampled low at the beginning of the Start condition (Figure 25-33).
b) SCL is sampled low before SDA is asserted low (Figure 25-34).

During a Start condition, both the SDA and the SCL pins are monitored.

If the SDA pin is already low, or the SCL pin is already low, then all of the following occur:

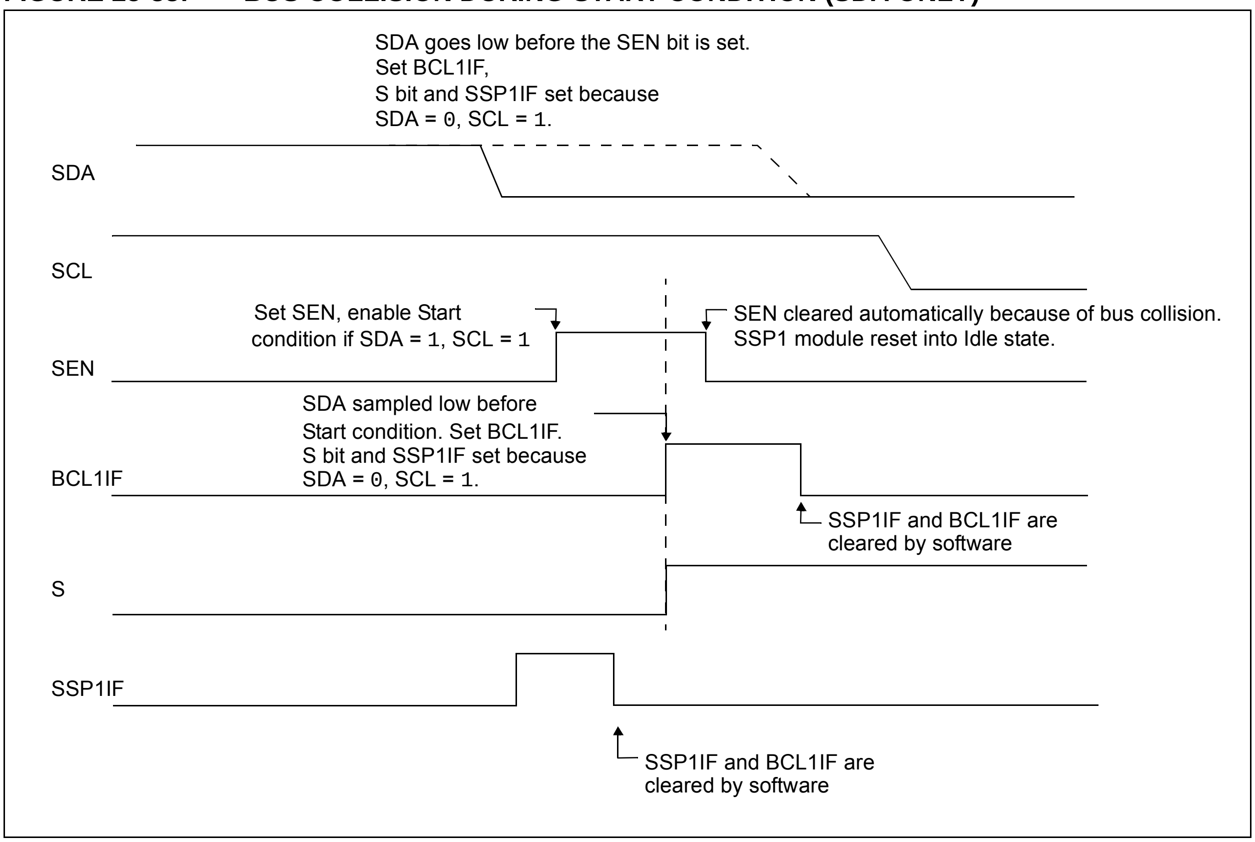
- the Start condition is aborted,
- the BCL1IF flag is set and
- the MSSP1 module is reset to its Idle state (Figure 25-33).

The Start condition begins with the SDA and SCL pins deasserted. When the SDA pin is sampled high, the Baud Rate Generator is loaded and counts down. If the SCL pin is sampled low while SDA is high, a bus collision occurs because it is assumed that another master is attempting to drive a data '1' during the Start condition.

If the SDA pin is sampled low during this count, the BRG is reset and the SDA line is asserted early (Figure 25-35). If, however, a '1' is sampled on the SDA pin, the SDA pin is asserted low at the end of the BRG count. The Baud Rate Generator is then reloaded and counts down to zero; if the SCL pin is sampled as '0' during this time, a bus collision does not occur. At the end of the BRG count, the SCL pin is asserted low.

| **Note:** | The reason that bus collision is not a factor during a Start condition is that no two bus masters can assert a Start condition at the exact same time. Therefore, one master will always assert SDA before the other. This condition does not cause a bus collision because the two masters must be allowed to arbitrate the first address following the Start condition. If the address is the same, arbitration must be allowed to continue into the data portion, Repeated Start or Stop conditions. |
|---|---|

**FIGURE 25-33: BUS COLLISION DURING START CONDITION (SDA ONLY)**



## 25.7 BAUD RATE GENERATOR

The MSSP1 module has a Baud Rate Generator available for clock generation in both I$^2$C and SPI Master modes. The Baud Rate Generator (BRG) reload value is placed in the SSP1ADD register (Register 25-6). When a write occurs to SSP1BUF, the Baud Rate Generator will automatically begin counting down.

Once the given operation is complete, the internal clock will automatically stop counting and the clock pin will remain in its last state.

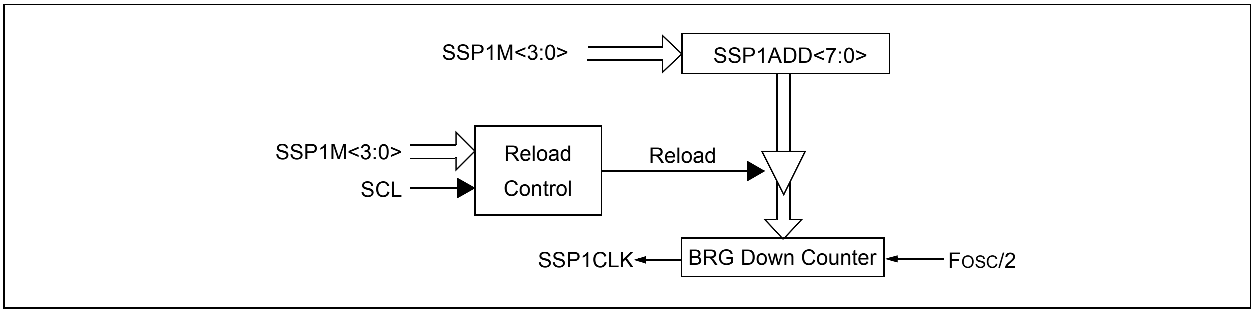
An internal signal “Reload” in Figure 25-40 triggers the value from SSP1ADD to be loaded into the BRG counter. This occurs twice for each oscillation of the module clock line. The logic dictating when the reload signal is asserted depends on the mode the MSSP1 is being operated in.

Table 25-4 demonstrates clock rates based on instruction cycles and the BRG value loaded into SSP1ADD.

**EQUATION 25-1:**

$$F_{CLOCK} = \frac{F_{OSC}}{(SSPxADD + 1)(4)}$$

**FIGURE 25-40: BAUD RATE GENERATOR BLOCK DIAGRAM**




| **Note:** | Values of 0x00, 0x01 and 0x02 are not valid for SSP1ADD when used as a Baud Rate Generator for I$^2$C. This is an implementation limitation. |
|---|---|

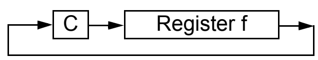
**TABLE 25-4: MSSP1 CLOCK RATE W/BRG**

| FOSC | FCY | BRG Value | FCLOCK (2 Rollovers of BRG) |
|---|---|---|---|
| 32 MHz | 8 MHz | 13h | 400 kHz |
| 32 MHz | 8 MHz | 19h | 308 kHz |
| 32 MHz | 8 MHz | 4Fh | 100 kHz |
| 16 MHz | 4 MHz | 09h | 400 kHz |
| 16 MHz | 4 MHz | 0Ch | 308 kHz |
| 16 MHz | 4 MHz | 27h | 100 kHz |
| 4 MHz | 1 MHz | 09h | 100 kHz |

**Note 1:** Refer to the I/O port electrical and timing specifications in Table 30-4 and Figure 30-7 to ensure the system is designed to support the I/O requirements.

### RRF — Rotate Right f through Carry

| | |
|---|---|
| Syntax: | [ *label* ] RRF f,d |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | See description below |
| Status Affected: | C |
| Description: | The contents of register 'f' are rotated one bit to the right through the Carry flag. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'. |




### SLEEP — Enter Sleep mode

| | |
|---|---|
| Syntax: | [ *label* ] SLEEP |
| Operands: | None |
| Operation: | 00h → WDT,<br>0 → WDT prescaler,<br>1 → $\overline{TO}$,<br>0 → $\overline{PD}$ |
| Status Affected: | $\overline{TO}$, $\overline{PD}$ |
| Description: | The power-down Status bit, $\overline{PD}$ is cleared. Time-out Status bit, $\overline{TO}$ is set. Watchdog Timer and its prescaler are cleared.<br>The processor is put into Sleep mode with the oscillator stopped. |

### SUBLW — Subtract W from literal

| | |
|---|---|
| Syntax: | [ *label* ] SUBLW k |
| Operands: | 0 ≤ k ≤ 255 |
| Operation: | k - (W) → (W) |
| Status Affected: | C, DC, Z |
| Description: | The W register is subtracted (2's complement method) from the 8-bit literal 'k'. The result is placed in the W register. |

| | |
|---|---|
| C = 0 | W > k |
| C = 1 | W ≤ k |
| DC = 0 | W<3:0> > k<3:0> |
| DC = 1 | W<3:0> ≤ k<3:0> |

### SUBWF — Subtract W from f

| | |
|---|---|
| Syntax: | [ *label* ] SUBWF f,d |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | (f) - (W) → (destination) |
| Status Affected: | C, DC, Z |
| Description: | Subtract (2's complement method) W register from register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f. |

| | |
|---|---|
| C = 0 | W > f |
| C = 1 | W ≤ f |
| DC = 0 | W<3:0> > f<3:0> |
| DC = 1 | W<3:0> ≤ f<3:0> |

### SUBWFB — Subtract W from f with Borrow

| | |
|---|---|
| Syntax: | SUBWFB f {,d} |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | (f) – (W) – ($\overline{B}$) → dest |
| Status Affected: | C, DC, Z |
| Description: | Subtract W and the BORROW flag (CARRY) 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'. |

### TABLE 30-2: OSCILLATOR PARAMETERS

**Standard Operating Conditions (unless otherwise stated)**
Operating Temperature -40°C ≤ TA ≤ +125°C

| Param No. | Sym. | Characteristic | Freq. Tolerance | Min. | Typ† | Max. | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| OS08 | HFOSC | Internal Calibrated HFINTOSC Frequency(2) | ±2% | — | 16.0 | — | MHz | 0°C ≤ TA ≤ +60°C, VDD ≥ 2.5V |
| | | | ±3% | — | 16.0 | — | MHz | 60°C ≤ TA ≤ +85°C, VDD ≥ 2.5V |
| | | | ±5% | — | 16.0 | — | MHz | -40°C ≤ TA ≤ +125°C |
| OS08A | MFOSC | Internal Calibrated MFINTOSC Frequency(2) | ±2% | — | 500 | — | kHz | 0°C ≤ TA ≤ +60°C, VDD ≥ 2.5V |
| | | | ±3% | — | 500 | — | kHz | 60°C ≤ TA ≤ +85°C, VDD ≥ 2.5V |
| | | | ±5% | — | 500 | — | kHz | -40°C ≤ TA ≤ +125°C |
| OS09 | LFOSC | Internal LFINTOSC Frequency | ±25% | — | 31 | — | kHz | -40°C ≤ TA ≤ +125°C |
| OS10* | TIOSC ST | HFINTOSC Wake-up from Sleep Start-up Time | — | — | 3.2 | 8 | μs | |
| | | MFINTOSC Wake-up from Sleep Start-up Time | — | — | 24 | 35 | μs | |

* These parameters are characterized but not tested.

† Data in “Typ” column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** Instruction cycle period (TCY) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at “min” values with an external clock applied to the OSC1 pin. When an external clock input is used, the “max” cycle time limit is “DC” (no clock) for all devices.

**2:** To ensure these oscillator frequency tolerances, VDD and VSS must be capacitively decoupled as close to the device as possible. 0.1 μF and 0.01 μF values in parallel are recommended.

**3:** By design.

### TABLE 30-3: PLL CLOCK TIMING SPECIFICATIONS (VDD = 2.7V TO 5.5V)

| Param No. | Sym. | Characteristic | Min. | Typ† | Max. | Units | Conditions |
|---|---|---|---|---|---|---|---|
| F10 | FOSC | Oscillator Frequency Range | 4 | — | 8 | MHz | |
| F11 | FSYS | On-Chip VCO System Frequency | 16 | — | 32 | MHz | |
| F12 | TRC | PLL Start-up Time (Lock Time) | — | — | 2 | ms | |
| F13* | ΔCLK | CLKOUT Stability (Jitter) | -0.25% | — | +0.25% | % | |

* These parameters are characterized but not tested.

† Data in “Typ” column is at 3V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**FIGURE 31-5: IDD TYPICAL, XT AND EXTRC OSCILLATOR, PIC12F1822 AND PIC16F1823 ONLY**

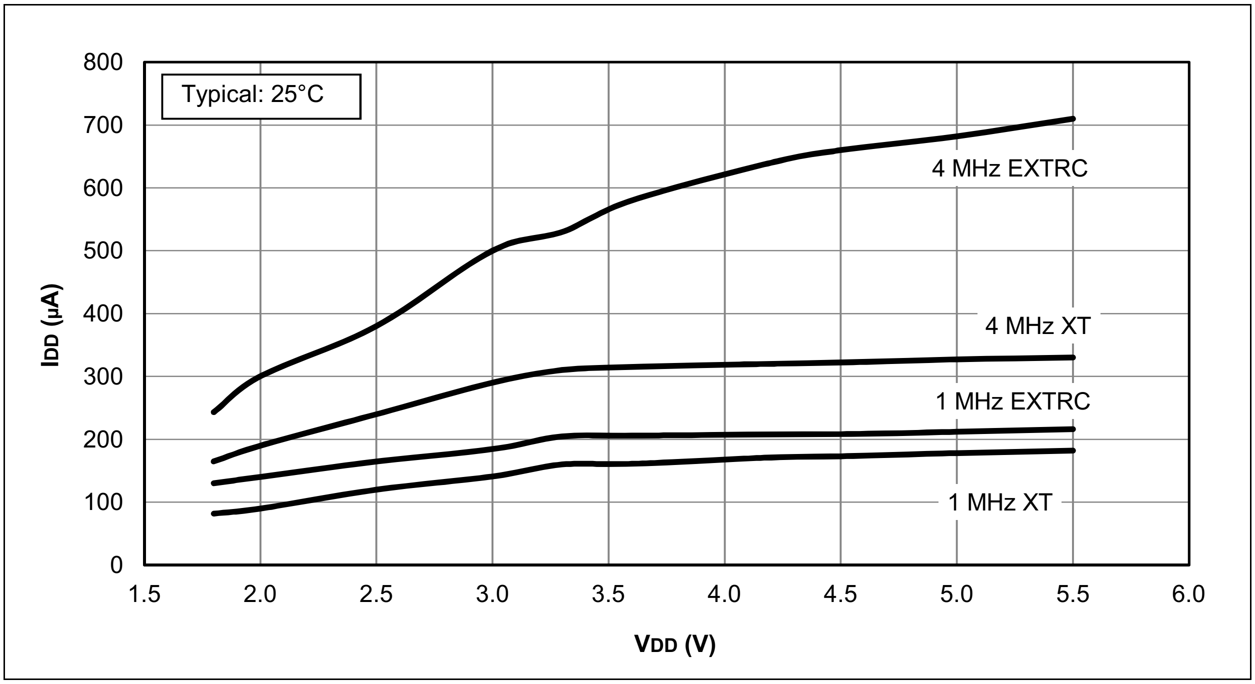
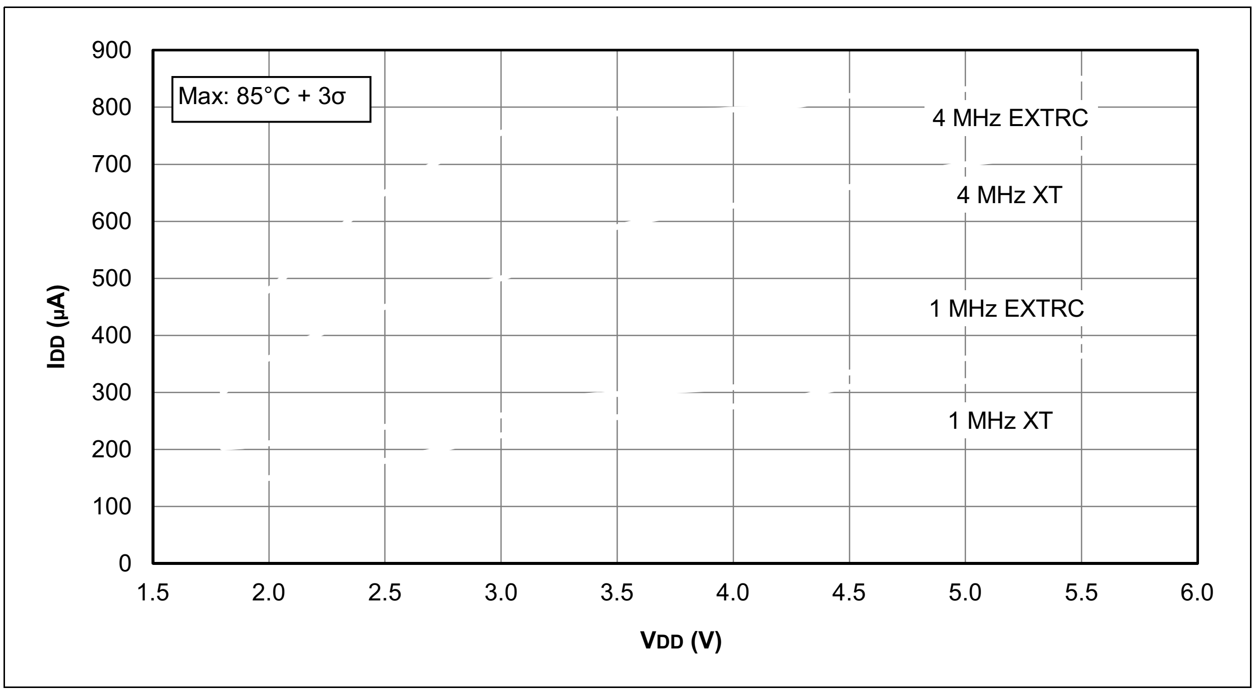
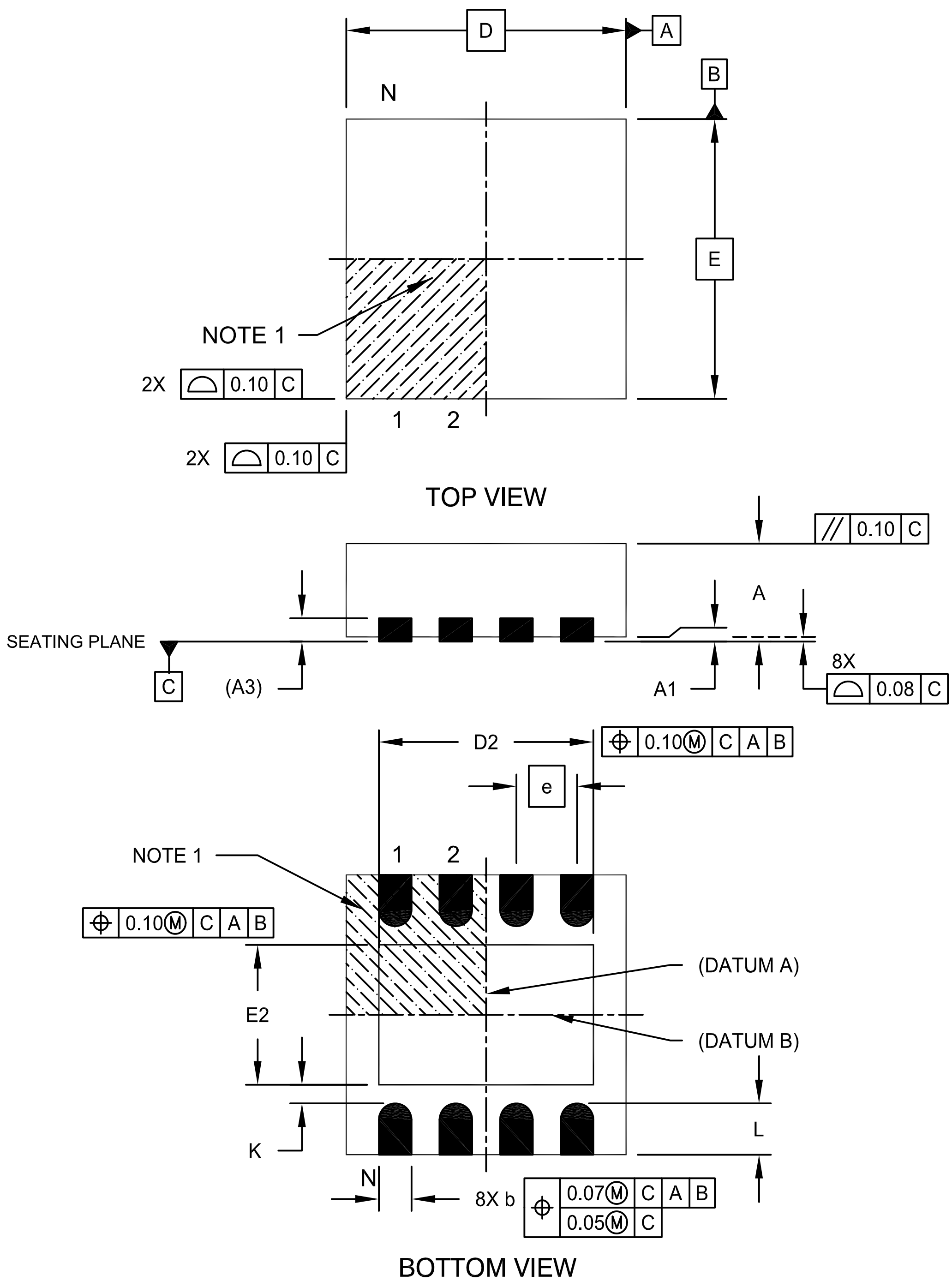



**FIGURE 31-6: IDD MAXIMUM, XT AND EXTRC OSCILLATOR, PIC12F1822 AND PIC16F1823 ONLY**



### 8-Lead Plastic Dual Flat, No Lead Package (MF) - 3x3x0.9mm Body [DFN]

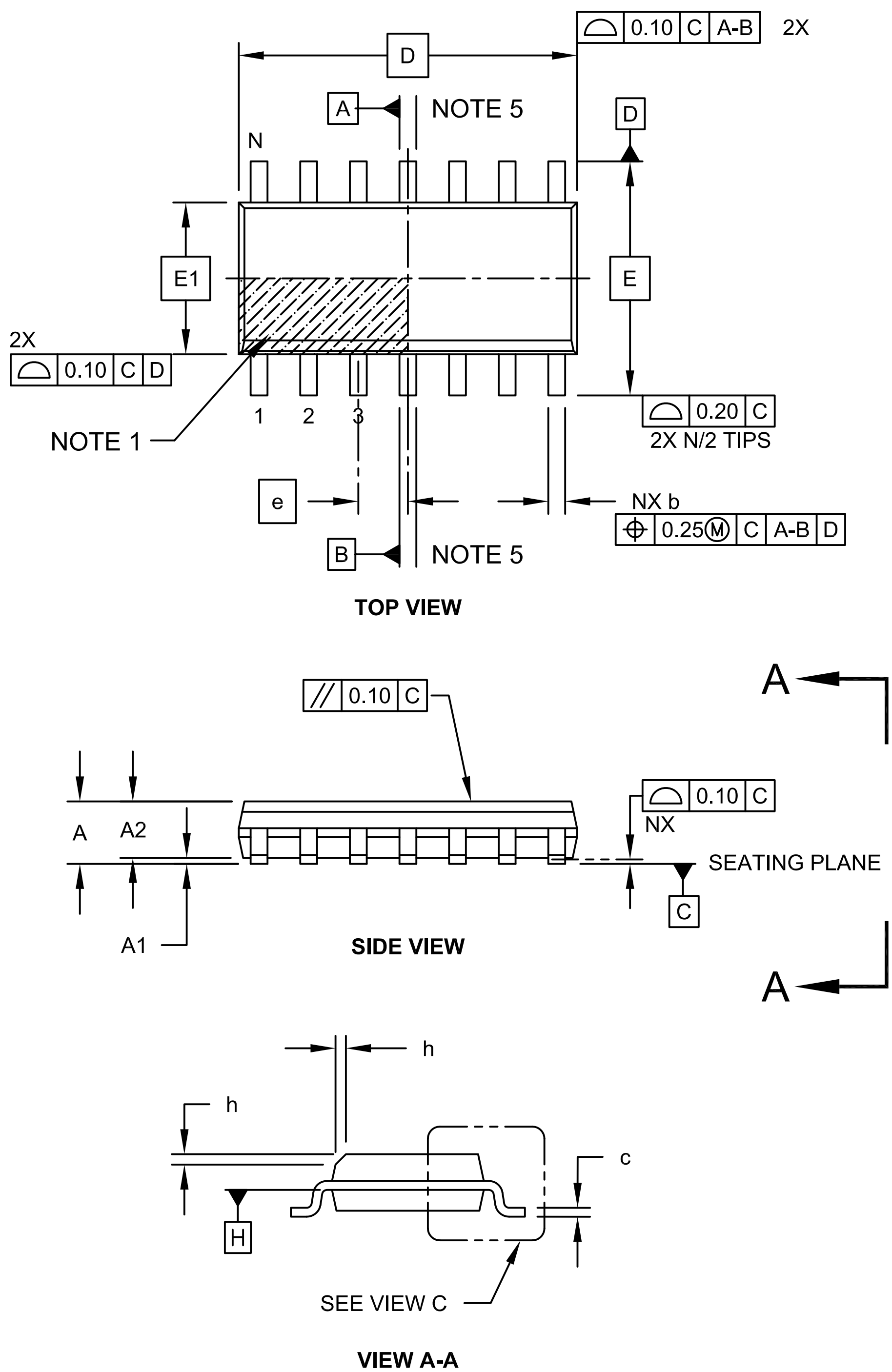
| | |
|---|---|
| **Note:** | For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging |




Microchip Technology Drawing No. C04-062C Sheet 1 of 2

### 14-Lead Plastic Small Outline (SL) - Narrow, 3.90 mm Body [SOIC]

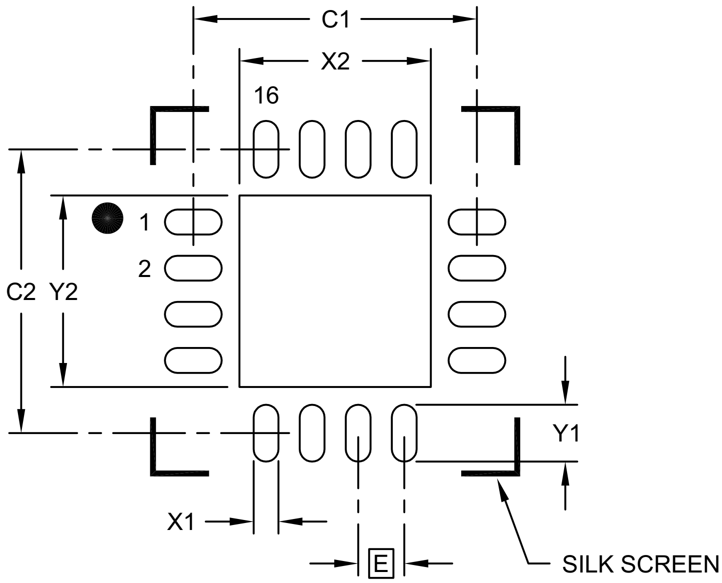
| | |
|---|---|
| **Note:** | For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging |




Microchip Technology Drawing No. C04-065C Sheet 1 of 2

### 16-Lead Ultra Thin Plastic Quad Flat, No Lead Package (JQ) - 4x4x0.5 mm Body [UQFN]

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




RECOMMENDED LAND PATTERN

| | Units | MILLIMETERS | | |
|---|---|---|---|---|
| | Dimension Limits | MIN | NOM | MAX |
| Contact Pitch | E | | 0.65 BSC | |
| Optional Center Pad Width | X2 | | | 2.70 |
| Optional Center Pad Length | Y2 | | | 2.70 |
| Contact Pad Spacing | C1 | | 4.00 | |
| Contact Pad Spacing | C2 | | 4.00 | |
| Contact Pad Width (X16) | X1 | | | 0.35 |
| Contact Pad Length (X16) | Y1 | | | 0.80 |

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing C04-2257A