



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](#)"

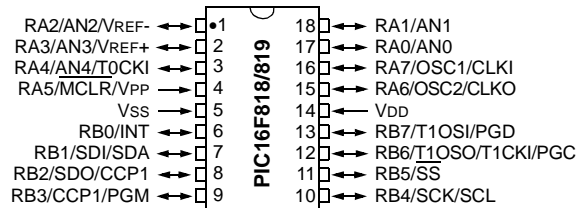
#### Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	10MHz
Connectivity	I <sup>2</sup> C, SPI
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	16
Program Memory Size	1.75KB (1K x 14)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	A/D 5x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	18-SOIC (0.295", 7.50mm Width)
Supplier Device Package	18-SOIC
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/pic16lf818t-i-so">https://www.e-xfl.com/product-detail/microchip-technology/pic16lf818t-i-so</a>

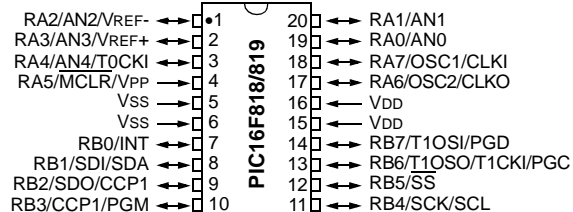
# PIC16F818/819

## Pin Diagrams

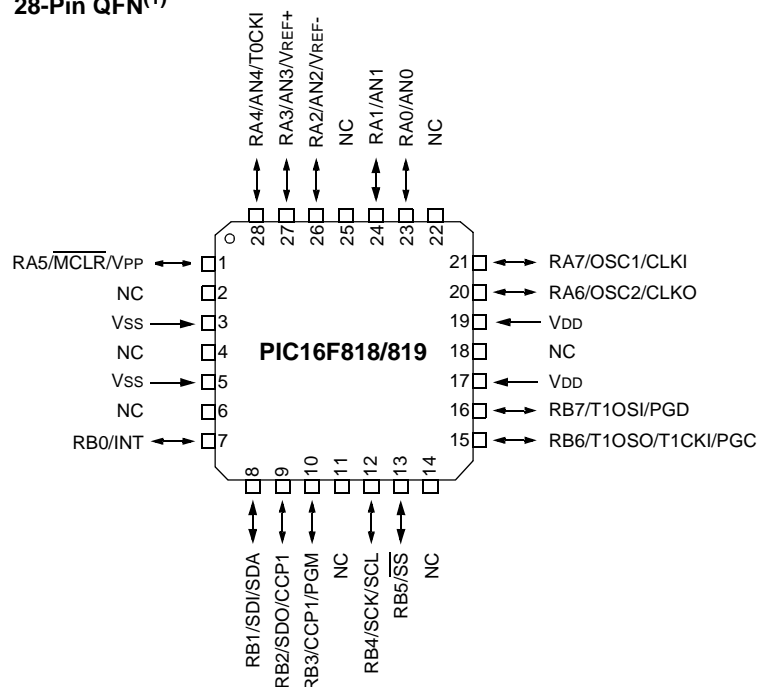
### 18-Pin PDIP, SOIC



### 20-Pin SSOP



### 28-Pin QFN<sup>(1)</sup>



**Note 1:** For the QFN package, it is recommended that the bottom pad be connected to VSS.

## Table of Contents

1.0	Device Overview .....	5
2.0	Memory Organization .....	9
3.0	Data EEPROM and Flash Program Memory .....	25
4.0	Oscillator Configurations .....	33
5.0	I/O Ports .....	39
6.0	Timer0 Module .....	53
7.0	Timer1 Module .....	57
8.0	Timer2 Module .....	63
9.0	Capture/Compare/PWM (CCP) Module .....	65
10.0	Synchronous Serial Port (SSP) Module .....	71
11.0	Analog-to-Digital Converter (A/D) Module .....	81
12.0	Special Features of the CPU .....	89
13.0	Instruction Set Summary .....	103
14.0	Development Support .....	111
15.0	Electrical Characteristics .....	115
16.0	DC and AC Characteristics Graphs and Tables .....	141
17.0	Packaging Information .....	155
	Appendix A: Revision History .....	165
	Appendix B: Device Differences .....	165
	INDEX .....	167
	The Microchip Web Site .....	173
	Customer Change Notification Service .....	173
	Customer Support .....	173
	Reader Response .....	174
	PIC16F818/819 Product Identification System .....	175

## TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at [docerrors@microchip.com](mailto:docerrors@microchip.com) or fax the **Reader Response Form** in the back of this data sheet to (480) 792-4150. We welcome your feedback.

### Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

<http://www.microchip.com>

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000A is version A of document DS30000).

### Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; <http://www.microchip.com>
- Your local Microchip sales office (see last page)

When contacting a sales office, please specify which device, revision of silicon and data sheet (include literature number) you are using.

### Customer Notification System

Register on our web site at [www.microchip.com](http://www.microchip.com) to receive the most current information on all of our products.

# PIC16F818/819

## 2.2 Data Memory Organization

The data memory is partitioned into multiple banks that contain the General Purpose Registers and the Special Function Registers. Bits RP1 (Status<6>) and RP0 (Status<5>) are the bank select bits.

RP1:RP0	Bank
00	0
01	1
10	2
11	3

Each bank extends up to 7Fh (128 bytes). The lower locations of each bank are reserved for the Special Function Registers. Above the Special Function Registers are the General Purpose Registers, implemented as static RAM. All implemented banks contain SFRs. Some “high use” SFRs from one bank may be mirrored in another bank for code reduction and quicker access (e.g., the Status register is in Banks 0-3).

**Note:** EEPROM data memory description can be found in **Section 3.0 “Data EEPROM and Flash Program Memory”** of this data sheet.

### 2.2.1 GENERAL PURPOSE REGISTER FILE

The register file can be accessed either directly or indirectly through the File Select Register, FSR.

# PIC16F818/819

## 2.2.2.3 INTCON Register

The INTCON register is a readable and writable register that contains various enable and flag bits for the TMR0 register overflow, RB port change and external RB0/INT pin interrupts.

**Note:** Interrupt flag bits get set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the Global Interrupt Enable bit, GIE (INTCON<7>). User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

### REGISTER 2-3: INTCON: INTERRUPT CONTROL REGISTER (ADDRESS 0Bh, 8Bh, 10Bh, 18Bh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF

bit 7

bit 0

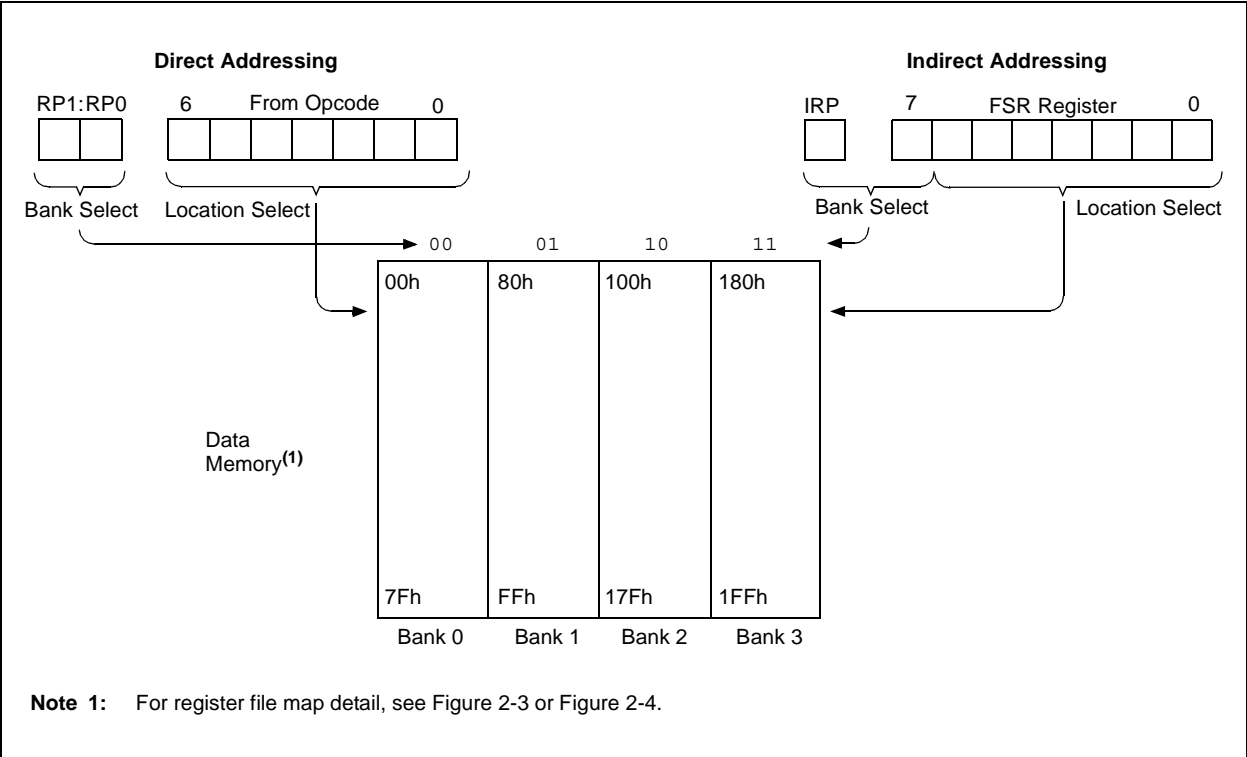
- bit 7 **GIE:** Global Interrupt Enable bit  
1 = Enables all unmasked interrupts  
0 = Disables all interrupts
- bit 6 **PEIE:** Peripheral Interrupt Enable bit  
1 = Enables all unmasked peripheral interrupts  
0 = Disables all peripheral interrupts
- bit 5 **TMR0IE:** TMR0 Overflow Interrupt Enable bit  
1 = Enables the TMR0 interrupt  
0 = Disables the TMR0 interrupt
- bit 4 **INTE:** RB0/INT External Interrupt Enable bit  
1 = Enables the RB0/INT external interrupt  
0 = Disables the RB0/INT external interrupt
- bit 3 **RBIE:** RB Port Change Interrupt Enable bit  
1 = Enables the RB port change interrupt  
0 = Disables the RB port change interrupt
- bit 2 **TMR0IF:** TMR0 Overflow Interrupt Flag bit  
1 = TMR0 register has overflowed (must be cleared in software)  
0 = TMR0 register did not overflow
- bit 1 **INTF:** RB0/INT External Interrupt Flag bit  
1 = The RB0/INT external interrupt occurred (must be cleared in software)  
0 = The RB0/INT external interrupt did not occur
- bit 0 **RBIF:** RB Port Change Interrupt Flag bit  
A mismatch condition will continue to set flag bit RBIF. Reading PORTB will end the mismatch condition and allow flag bit RBIF to be cleared.  
1 = At least one of the RB7:RB4 pins changed state (must be cleared in software)  
0 = None of the RB7:RB4 pins have changed state

#### 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

# PIC16F818/819

FIGURE 2-6: DIRECT/INDIRECT ADDRESSING



## 3.7 Writing to Flash Program Memory

Flash program memory may only be written to if the destination address is in a segment of memory that is not write-protected, as defined in bits WRT1:WRT0 of the device Configuration Word (Register 12-1). Flash program memory must be written in four-word blocks. A block consists of four words with sequential addresses, with a lower boundary defined by an address, where  $EEADR<1:0> = 00$ . At the same time, all block writes to program memory are done as write-only operations. The program memory must first be erased. The write operation is edge-aligned and cannot occur across boundaries.

To write to the program memory, the data must first be loaded into the buffer registers. There are four 14-bit buffer registers and they are addressed by the low 2 bits of  $EEADR$ .

The following sequence of events illustrate how to perform a write to program memory:

- Set the  $EEPGD$  and  $WREN$  bits in the  $EECON1$  register
- Clear the  $FREE$  bit in  $EECON1$
- Write address to  $EEADRH:EEADR$
- Write data to  $EEDATH:EEDATA$
- Write 55 to  $EECON2$
- Write AA to  $EECON2$
- Set  $WR$  bit in  $EECON1$

The user must follow the same specific sequence to initiate the write for each word in the program block by writing each program word in sequence (00, 01, 10, 11).

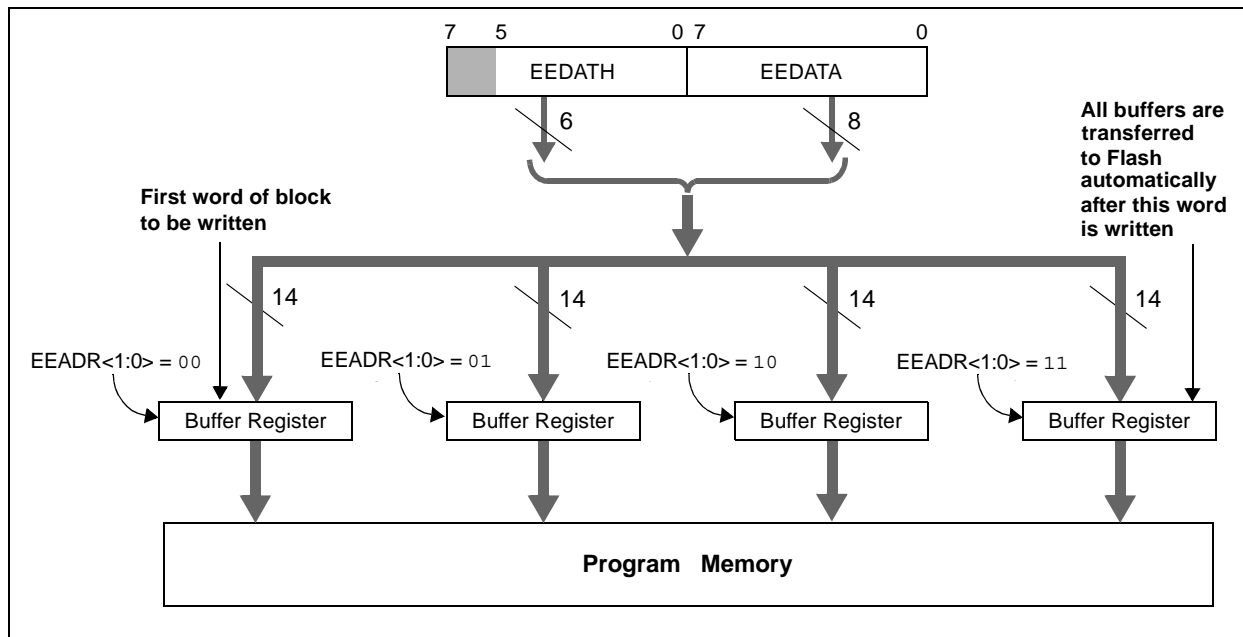
There are 4 buffer register words and all four locations **MUST** be written to with correct data.

After the “BSF  $EECON1$ ,  $WR$ ” instruction, if  $EEADR \neq \text{xxxxxx}11$ , then a short write will occur. This short write-only transfers the data to the buffer register. The  $WR$  bit will be cleared in hardware after one cycle.

After the “BSF  $EECON1$ ,  $WR$ ” instruction, if  $EEADR = \text{xxxxxx}11$ , then a long write will occur. This will simultaneously transfer the data from  $EEDATH:EEDATA$  to the buffer registers and begin the write of all four words. The processor will execute the next instruction and then ignore the subsequent instruction. The user should place NOP instructions into the second words. The processor will then halt internal operations for typically 2 msec in which the write takes place. This is not a Sleep mode, as the clocks and peripherals will continue to run. After the write cycle, the processor will resume operation with the 3rd instruction after the  $EECON1$  write instruction.

After each long write, the 4 buffer registers will be reset to 3FFF.

**FIGURE 3-1: BLOCK WRITES TO FLASH PROGRAM MEMORY**



# PIC16F818/819

## 7.2 Timer1 Operation in Timer Mode

Timer mode is selected by clearing the TMR1CS (T1CON<1>) bit. In this mode, the input clock to the timer is  $F_{osc}/4$ . The synchronize control bit,  $\overline{T1SYNC}$  (T1CON<2>), has no effect since the internal clock is always in sync.

## 7.3 Timer1 Counter Operation

Timer1 may operate in Asynchronous or Synchronous mode depending on the setting of the TMR1CS bit.

When Timer1 is being incremented via an external source, increments occur on a rising edge. After Timer1 is enabled in Counter mode, the module must first have a falling edge before the counter begins to increment.

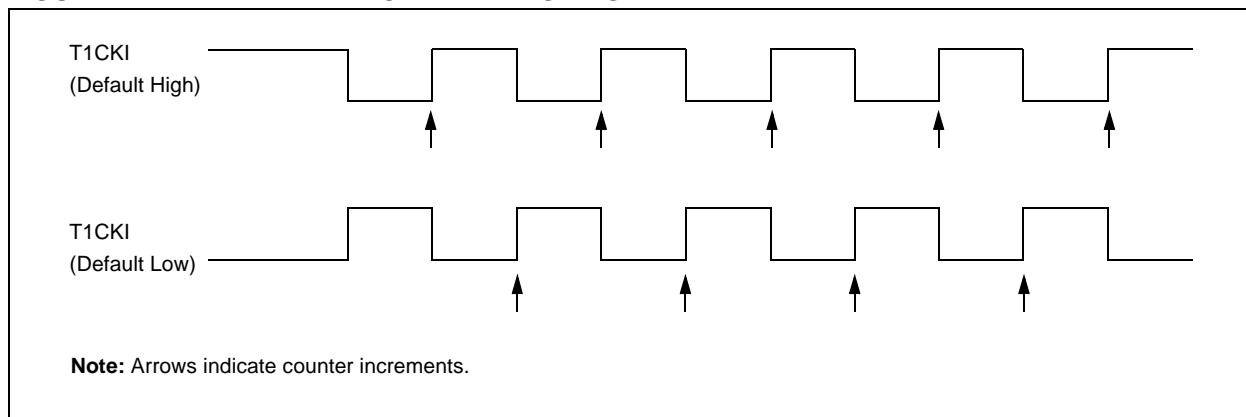
## 7.4 Timer1 Operation in Synchronized Counter Mode

Counter mode is selected by setting bit TMR1CS. In this mode, the timer increments on every rising edge of clock input on pin RB7/T1OSI/PGD when bit T1OSCEN is set, or on pin RB6/T1OSO/T1CKI/PGC when bit T1OSCEN is cleared.

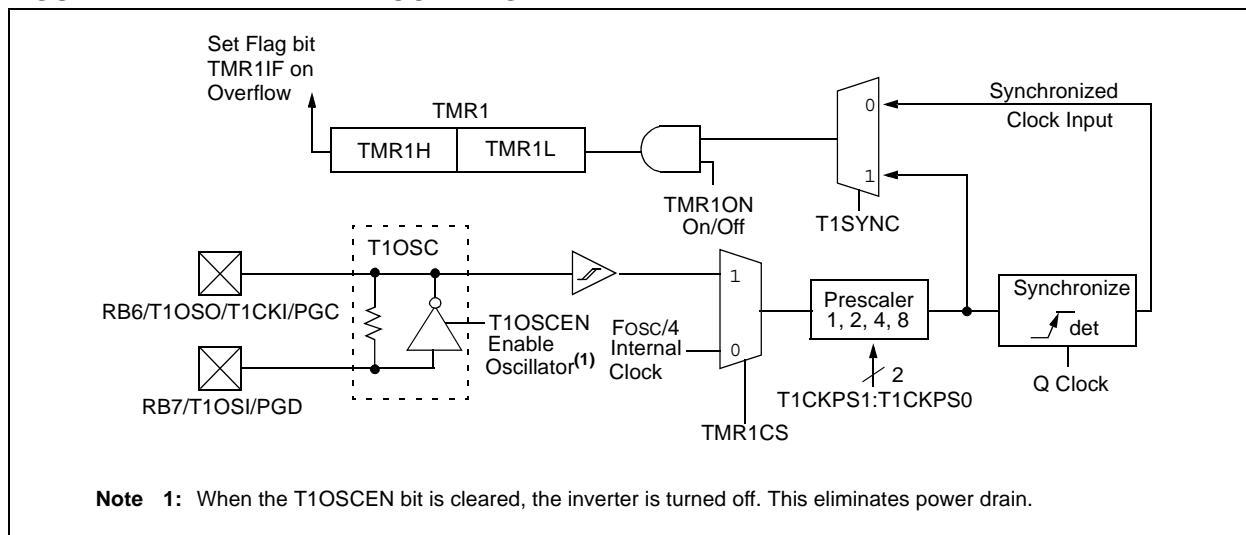
If  $\overline{T1SYNC}$  is cleared, then the external clock input is synchronized with internal phase clocks. The synchronization is done after the prescaler stage. The prescaler stage is an asynchronous ripple counter.

In this configuration, during Sleep mode, Timer1 will not increment even if the external clock is present, since the synchronization circuit is shut-off. The prescaler, however, will continue to increment.

**FIGURE 7-1: TIMER1 INCREMENTING EDGE**



**FIGURE 7-2: TIMER1 BLOCK DIAGRAM**





## 7.8 Resetting Timer1 Using a CCP Trigger Output

If the CCP1 module is configured in Compare mode to generate a “special event trigger” signal (CCP1M3:CCP1M0 = 1011), the signal will reset Timer1 and start an A/D conversion (if the A/D module is enabled).

Timer1 must be configured for either Timer or Synchronized Counter mode to take advantage of this feature. If Timer1 is running in Asynchronous Counter mode, this Reset operation may not work.

In the event that a write to Timer1 coincides with a special event trigger from CCP1, the write will take precedence.

In this mode of operation, the CCPR1H:CCPR1L register pair effectively becomes the period register for Timer1.

## 7.9 Resetting Timer1 Register Pair (TMR1H, TMR1L)

TMR1H and TMR1L registers are not reset to 00h on a POR or any other Reset, except by the CCP1 special event triggers.

T1CON register is reset to 00h on a Power-on Reset or a Brown-out Reset, which shuts off the timer and leaves a 1:1 prescale. In all other Resets, the register is unaffected.

## 7.10 Timer1 Prescaler

The prescaler counter is cleared on writes to the TMR1H or TMR1L registers.

## 7.11 Using Timer1 as a Real-Time Clock

Adding an external LP oscillator to Timer1 (such as the one described in **Section 7.6 “Timer1 Oscillator”**), gives users the option to include RTC functionality in their applications. This is accomplished with an inexpensive watch crystal to provide an accurate time base and several lines of application code to calculate the time. When operating in Sleep mode and using a battery or supercapacitor as a power source, it can completely eliminate the need for a separate RTC device and battery backup.

The application code routine, *RTClisr*, shown in Example 7-3, demonstrates a simple method to increment a counter at one-second intervals using an Interrupt Service Routine. Incrementing the TMR1 register pair to overflow, triggers the interrupt and calls the routine which increments the seconds counter by one; additional counters for minutes and hours are incremented as the previous counter overflows.

Since the register pair is 16 bits wide, counting up to overflow the register directly from a 32.768 kHz clock would take 2 seconds. To force the overflow at the required one-second intervals, it is necessary to preload it; the simplest method is to set the MSb of TMR1H with a BSF instruction. Note that the TMR1L register is never preloaded or altered; doing so may introduce cumulative error over many cycles.

For this method to be accurate, Timer1 must operate in Asynchronous mode and the Timer1 overflow interrupt must be enabled (PIE1<0> = 1) as shown in the routine, *RTCinit*. The Timer1 oscillator must also be enabled and running at all times.

# PIC16F818/819

---

NOTES:

## 10.0 SYNCHRONOUS SERIAL PORT (SSP) MODULE

### 10.1 SSP Module Overview

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

- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I<sup>2</sup>C)

An overview of I<sup>2</sup>C operations and additional information on the SSP module can be found in the “PIC® Mid-Range MCU Family Reference Manual” (DS33023).

Refer to Application Note AN578, “Use of the SSP Module in the I<sup>2</sup>C™ Multi-Master Environment” (DS00578).

### 10.2 SPI Mode

This section contains register definitions and operational characteristics of the SPI module.

SPI mode allows 8 bits of data to be synchronously transmitted and received simultaneously. To accomplish communication, typically three pins are used:

- Serial Data Out (SDO) RB2/SDO/CCP1
- Serial Data In (SDI) RB1/SDI/SDA
- Serial Clock (SCK) RB4/SCK/SCL

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

- Slave Select ( $\overline{SS}$ ) RB5/ $\overline{SS}$

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits in the SSPCON register (SSPCON<5:0>) and the SSPSTAT register (SSPSTAT<7:6>). These control bits allow the following to be specified:

- Master mode (SCK is the clock output)
- Slave mode (SCK is the clock input)
- Clock Polarity (Idle state of SCK)
- Clock Edge (output data on rising/falling edge of SCK)
- Clock Rate (Master mode only)
- Slave Select mode (Slave mode only)

**Note:** Before enabling the module in SPI Slave mode, the state of the clock line (SCK) must match the polarity selected for the Idle state. The clock line can be observed by reading the SCK pin. The polarity of the Idle state is determined by the CKP bit (SSPCON<4>).

## 10.3.1 SLAVE MODE

In Slave mode, the SCL and SDA pins must be configured as inputs (TRISB<4,1> set). The SSP module will override the input state with the output data when required (slave-transmitter).

When an address is matched, or the data transfer after an address match is received, the hardware automatically will generate the Acknowledge (ACK) pulse and then load the SSPBUF register with the received value currently in the SSPSR register.

Either or both of the following conditions will cause the SSP module not to give this ACK pulse:

- a) The Buffer Full bit, BF (SSPSTAT<0>), was set before the transfer was received.
- b) The overflow bit, SSPOV (SSPCON<6>), was set before the transfer was received.

In this case, the SSPSR register value is not loaded into the SSPBUF but bit, SSPIF (PIR1<3>), is set. Table 10-2 shows what happens when a data transfer byte is received, given the status of bits BF and SSPOV. The shaded cells show the condition where user software did not properly clear the overflow condition. Flag bit BF is cleared by reading the SSPBUF register while bit SSPOV is cleared through software.

The SCL clock input must have a minimum high and low for proper operation. The high and low times of the I<sup>2</sup>C specification, as well as the requirement of the SSP module, are shown in timing parameter #100 and parameter #101.

### 10.3.1.1 Addressing

Once the SSP module has been enabled, it waits for a Start condition to occur. Following the Start condition, the eight bits are shifted into the SSPSR register. All incoming bits are sampled with the rising edge of the clock (SCL) line. The value of register SSPSR<7:1> is compared to the value of the SSPADD register. The address is compared on the falling edge of the eighth clock (SCL) pulse. If the addresses match and the BF and SSPOV bits are clear, the following events occur:

- a) The SSPSR register value is loaded into the SSPBUF register.
- b) The Buffer Full bit, BF, is set.
- c) An  $\overline{\text{ACK}}$  pulse is generated.
- d) SSP Interrupt Flag bit, SSPIF (PIR1<3>), is set (interrupt is generated if enabled) – on the falling edge of the ninth SCL pulse.

In 10-bit Address mode, two address bytes need to be received by the slave device. The five Most Significant bits (MSbs) of the first address byte specify if this is a 10-bit address. Bit R/W (SSPSTAT<2>) must specify a write so the slave device will receive the second address byte. For a 10-bit address, the first byte would equal '1111 0 A9 A8 0', where A9 and A8 are the two MSbs of the address.

The sequence of events for 10-bit address is as follows, with steps 7-9 for slave-transmitter:

1. Receive first (high) byte of address (bits SSPIF, BF and bit UA (SSPSTAT<1>) are set).
2. Update the SSPADD register with second (low) byte of address (clears bit UA and releases the SCL line).
3. Read the SSPBUF register (clears bit BF) and clear flag bit, SSPIF.
4. Receive second (low) byte of address (bits SSPIF, BF and UA are set).
5. Update the SSPADD register with the first (high) byte of address; if match releases SCL line, this will clear bit UA.
6. Read the SSPBUF register (clears bit BF) and clear flag bit, SSPIF.
7. Receive Repeated Start condition.
8. Receive first (high) byte of address (bits SSPIF and BF are set).
9. Read the SSPBUF register (clears bit BF) and clear flag bit, SSPIF.

### 10.3.1.2 Reception

When the R/W bit of the address byte is clear and an address match occurs, the R/W bit of the SSPSTAT register is cleared. The received address is loaded into the SSPBUF register.

When the address byte overflow condition exists, then a no Acknowledge ( $\overline{\text{ACK}}$ ) pulse is given. An overflow condition is indicated if either bit, BF (SSPSTAT<0>), is set or bit, SSPOV (SSPCON<6>), is set.

An SSP interrupt is generated for each data transfer byte. Flag bit, SSPIF (PIR1<3>), must be cleared in software. The SSPSTAT register is used to determine the status of the byte.

### 10.3.1.3 Transmission

When the R/W bit of the incoming address byte is set and an address match occurs, the R/W bit of the SSPSTAT register is set. The received address is loaded into the SSPBUF register. The  $\overline{\text{ACK}}$  pulse will be sent on the ninth bit and pin RB4/SCK/SCL is held low. The transmit data must be loaded into the SSPBUF register which also loads the SSPSR register. Then pin RB4/SCK/SCL should be enabled by setting bit, CKP (SSPCON<4>). The master device must monitor the SCL pin prior to asserting another clock pulse. The slave devices may be holding off the master device by stretching the clock. The eight data bits are shifted out on the falling edge of the SCL input. This ensures that the SDA signal is valid during the SCL high time (Figure 10-7).

# PIC16F818/819

**TABLE 12-4: INITIALIZATION CONDITIONS FOR ALL REGISTERS**

Register	Power-on Reset, Brown-out Reset	MCLR Reset, WDT Reset	Wake-up via WDT or Interrupt
W	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF	N/A	N/A	N/A
TMR0	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	0000h	0000h	PC + 1 <sup>(2)</sup>
STATUS	0001 1xxx	000q quuu <sup>(3)</sup>	uuuq quuu <sup>(3)</sup>
FSR	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA	xxx0 0000	uuu0 0000	uuuu uuuu
PORTB	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCLATH	---0 0000	---0 0000	---u uuuu
INTCON	0000 000x	0000 000u	uuuu uuuu <sup>(1)</sup>
PIR1	-0-- 0000	-0-- 0000	-u-- uuuu <sup>(1)</sup>
PIR2	---0 ----	---0 ----	---u ---- <sup>(1)</sup>
TMR1L	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1H	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CON	--00 0000	--uu uuuu	--uu uuuu
TMR2	0000 0000	0000 0000	uuuu uuuu
T2CON	-000 0000	-000 0000	-uuu uuuu
SSPBUF	xxxx xxxx	uuuu uuuu	uuuu uuuu
SSPCON	0000 0000	0000 0000	uuuu uuuu
CCPR1L	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1H	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON	--00 0000	--00 0000	--uu uuuu
ADRESH	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0	0000 00-0	0000 00-0	uuuu uu-u
OPTION_REG	1111 1111	1111 1111	uuuu uuuu
TRISA	1111 1111	1111 1111	uuuu uuuu
TRISB	1111 1111	1111 1111	uuuu uuuu
PIE1	-0-- 0000	-0-- 0000	-u-- uuuu
PIE2	---0 ----	---0 ----	---u ----
PCON	---- -qqr	---- -uu	---- -uu
OSCCON	-000 -0--	-000 -0--	-uuu -u--
OSCTUNE	--00 0000	--00 0000	--uu uuuu
PR2	1111 1111	1111 1111	1111 1111
SSPADD	0000 0000	0000 0000	uuuu uuuu
SSPSTAT	0000 0000	0000 0000	uuuu uuuu
ADRESL	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON1	00-- 0000	00-- 0000	uu-- uuuu
EEDATA	xxxx xxxx	uuuu uuuu	uuuu uuuu
EEADR	xxxx xxxx	uuuu uuuu	uuuu uuuu
EEDATH	--xx xxxx	--uu uuuu	--uu uuuu
EEADRH	---- -xxx	---- -uuu	---- -uuu
EECON1	x--x x000	u--x u000	u--u uuuu
EECON2	---- ----	---- ----	---- ----

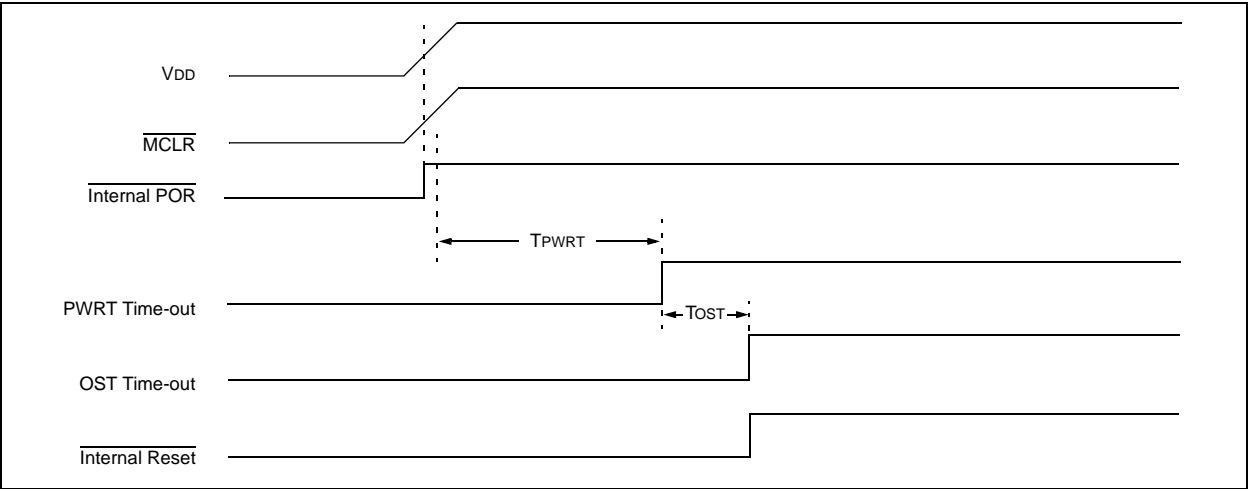
**Legend:** u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition, r = reserved, maintain clear

**Note 1:** One or more bits in INTCON, PIR1 and PR2 will be affected (to cause wake-up).

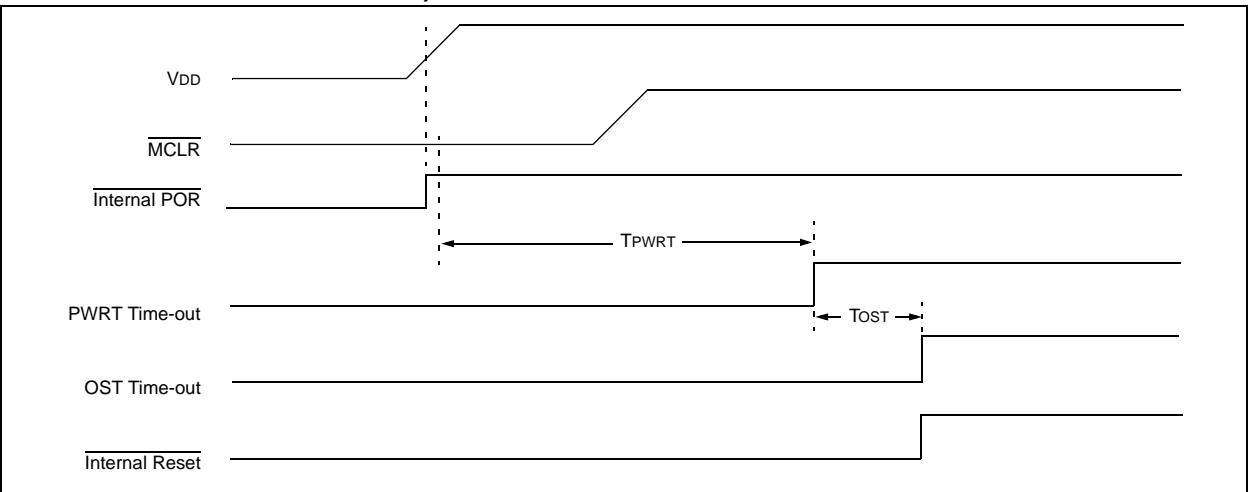
**2:** When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

**3:** See Table 12-3 for Reset value for specific conditions.

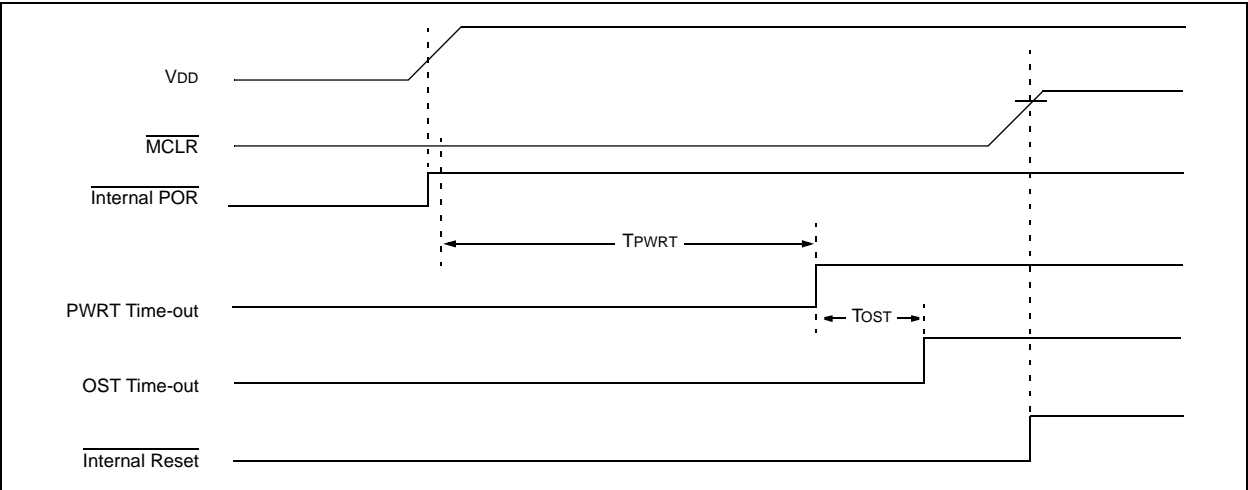
**FIGURE 12-3: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  TIED TO  $V_{\text{DD}}$  THROUGH PULL-UP RESISTOR)**



**FIGURE 12-4: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  TIED TO  $V_{\text{DD}}$  THROUGH RC NETWORK): CASE 1**



**FIGURE 12-5: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  TIED TO  $V_{\text{DD}}$  THROUGH RC NETWORK): CASE 2**



## 12.18 Low-Voltage ICSP Programming

The LVP bit of the Configuration Word register enables Low-Voltage ICSP Programming. This mode allows the microcontroller to be programmed via ICSP using a VDD source in the operating voltage range. This only means that VPP does not have to be brought to VIHh but can instead be left at the normal operating voltage. In this mode, the RB3/PGM pin is dedicated to the programming function and ceases to be a general purpose I/O pin.

If Low-Voltage Programming mode is not used, the LVP bit can be programmed to a '0' and RB3/PGM becomes a digital I/O pin. However, the LVP bit may only be programmed when Programming mode is entered with VIHh on MCLR. The LVP bit can only be changed when using high voltage on MCLR.

It should be noted that once the LVP bit is programmed to '0', only the High-Voltage Programming mode is available and only this mode can be used to program the device.

When using Low-Voltage ICSP, the part must be supplied at 4.5V to 5.5V if a bulk erase will be executed. This includes reprogramming of the code-protect bits from an ON state to an OFF state. For all other cases of Low-Voltage ICSP, the part may be programmed at the normal operating voltage. This means calibration values, unique user IDs or user code can be reprogrammed or added.

The following LVP steps assume the LVP bit is set in the Configuration Word register.

1. Apply VDD to the VDD pin.
2. Drive MCLR low.
3. Apply VDD to the RB3/PGM pin.
4. Apply VDD to the MCLR pin.
5. Follow with the associated programming steps.

**Note 1:** The High-Voltage Programming mode is always available, regardless of the state of the LVP bit, by applying VIHh to the MCLR pin.

- 2: While in Low-Voltage ICSP mode (LVP = 1), the RB3 pin can no longer be used as a general purpose I/O pin.
- 3: When using Low-Voltage ICSP Programming (LVP) and the pull-ups on PORTB are enabled, bit 3 in the TRISB register must be cleared to disable the pull-up on RB3 and ensure the proper operation of the device.
- 4: RB3 should not be allowed to float if LVP is enabled. An external pull-down device should be used to default the device to normal operating mode. If RB3 floats high, the PIC16F818/819 device will enter Programming mode.
- 5: LVP mode is enabled by default on all devices shipped from Microchip. It can be disabled by clearing the LVP bit in the Configuration Word register.
- 6: Disabling LVP will provide maximum compatibility to other PIC16CXXX devices.

# PIC16F818/819

---

## **SUBLW**      **Subtract W from Literal**

---

Syntax:      [ *label* ]    SUBLW   k

Operands:       $0 \leq k \leq 255$

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

Status Affected:    C, DC, Z

Description:      The W register is subtracted (2's complement method) from the eight-bit literal 'k'. The result is placed in the W register.

---

## **XORLW**      **Exclusive OR Literal with W**

---

Syntax:      [ *label* ]    XORLW   k

Operands:       $0 \leq k \leq 255$

Operation:       $(W) .XOR. k \rightarrow (W)$

Status Affected:    Z

Description:      The contents of the W register are XORed with the eight-bit literal 'k'. The result is placed in the W register.

---

## **SUBWF**      **Subtract W from f**

---

Syntax:      [ *label* ]    SUBWF   f,d

Operands:       $0 \leq f \leq 127$   
                     $d \in [0,1]$

Operation:       $(f) - (W) \rightarrow (\text{destination})$

Status Affected:    C, DC, Z

Description:      Subtract (2's complement method) W register from register 'f'. If 'd' = 0, the result is stored in the W register. If 'd' = 1, the result is stored back in register 'f'.

---

## **XORWF**      **Exclusive OR W with f**

---

Syntax:      [ *label* ]    XORWF   f,d

Operands:       $0 \leq f \leq 127$   
                     $d \in [0,1]$

Operation:       $(W) .XOR. (f) \rightarrow (\text{destination})$

Status Affected:    Z

Description:      Exclusive OR the contents of the W register with register 'f'. If 'd' = 0, the result is stored in the W register. If 'd' = 1, the result is stored back in register 'f'.

---

## **SWAPF**      **Swap Nibbles in f**

---

Syntax:      [ *label* ]    SWAPF   f,d

Operands:       $0 \leq f \leq 127$   
                     $d \in [0,1]$

Operation:       $(f<3:0>) \rightarrow (\text{destination}<7:4>),$   
                     $(f<7:4>) \rightarrow (\text{destination}<3:0>)$

Status Affected:    None

Description:      The upper and lower nibbles of register 'f' are exchanged. If 'd' = 0, the result is placed in W register. If 'd' = 1, the result is placed in register 'f'.



## 15.0 ELECTRICAL CHARACTERISTICS

### Absolute Maximum Ratings †

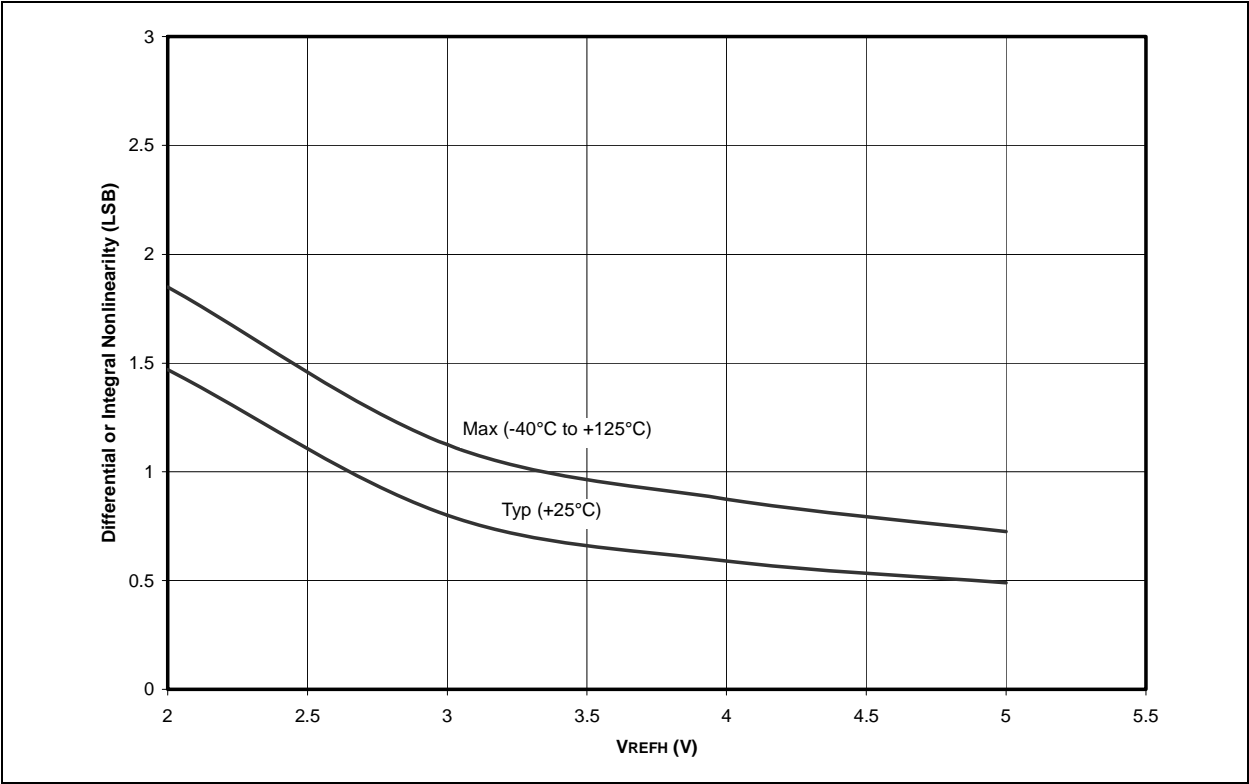
Ambient temperature under bias .....	-40°C to +125°C
Storage temperature .....	-65°C to +150°C
Voltage on any pin with respect to VSS (except VDD and $\overline{\text{MCLR}}$ ) .....	-0.3V to (VDD + 0.3V)
Voltage on VDD with respect to VSS .....	-0.3 to +7.5V
Voltage on $\overline{\text{MCLR}}$ with respect to VSS ( <b>Note 2</b> ) .....	-0.3 to +14V
Total power dissipation ( <b>Note 1</b> ) .....	1W
Maximum current out of VSS pin .....	200 mA
Maximum current into VDD pin .....	200 mA
Input clamp current, I <sub>IK</sub> (V <sub>I</sub> < 0 or V <sub>I</sub> > VDD) .....	±20 mA
Output clamp current, I <sub>OK</sub> (V <sub>O</sub> < 0 or V <sub>O</sub> > VDD) .....	±20 mA
Maximum output current sunk by any I/O pin.....	25 mA
Maximum output current sourced by any I/O pin .....	25 mA
Maximum current sunk by PORTA.....	100 mA
Maximum current sourced by PORTA.....	100 mA
Maximum current sunk by PORTB.....	100 mA
Maximum current sourced by PORTB .....	100 mA

**Note 1:** Power dissipation is calculated as follows:  $P_{dis} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (V_{OL} \times I_{OL})$

**2:** Voltage spikes at the  $\overline{\text{MCLR}}$  pin may cause latch-up. A series resistor of greater than 1 k $\Omega$  should be used to pull  $\overline{\text{MCLR}}$  to VDD, rather than tying the pin directly to VDD.

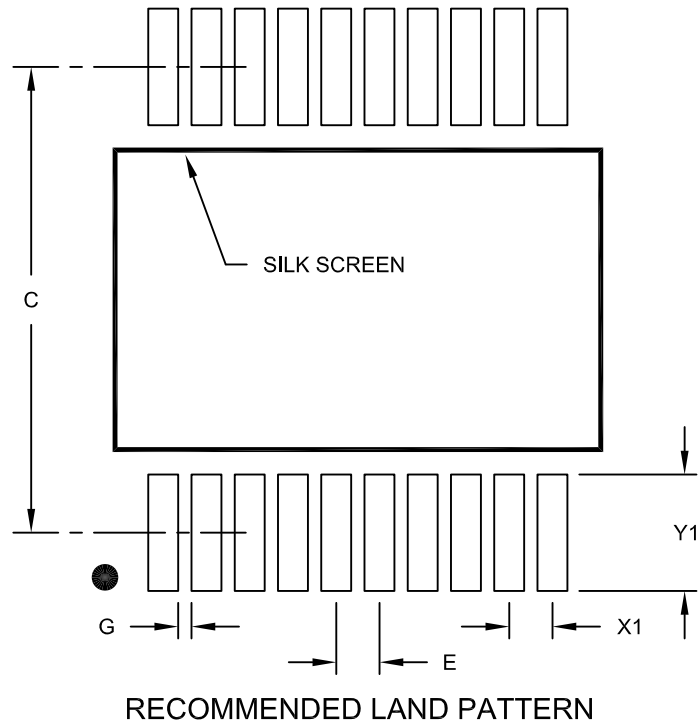
† NOTICE: Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

FIGURE 16-25: A/D NONLINEARITY vs. VREFH (VDD = 5V, -40°C TO +125°C)



## 20-Lead Plastic Shrink Small Outline (SS) - 5.30 mm Body [SSOP]

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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Contact Pad Spacing	C		7.20	
Contact Pad Width (X20)	X1			0.45
Contact Pad Length (X20)	Y1			1.75
Distance Between Pads	G	0.20		

### Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2072A

Timer1 .....	57	TRISA Register .....	14
Associated Registers .....	62	TRISB Register .....	14, 15
Capacitor Selection .....	60	<b>V</b>	
Counter Operation .....	58	Vdd Pin .....	8
Operation .....	57	Vss Pin .....	8
Operation in Asynchronous Counter Mode .....	59	<b>W</b>	
Operation in Synchronized Counter Mode .....	58	Wake-up from Sleep .....	89, 99
Operation in Timer Mode .....	58	Interrupts .....	93, 94
Oscillator .....	60	MCLR Reset .....	94
Oscillator Layout Considerations .....	60	WDT Reset .....	94
Prescaler .....	61	Wake-up Using Interrupts .....	99
Resetting Register Pair (TMR1H, TMR1L) .....	61	Watchdog Timer (WDT) .....	89, 98
Resetting Using a CCP Trigger Output .....	61	Associated Registers .....	98
TMR1H .....	59	Enable (WDTEN Bit) .....	98
TMR1L .....	59	INTRC Oscillator .....	98
Use as a Real-Time Clock .....	61	Postscaler. See Postscaler, WDT.	
Timer2 .....	63	Programming Considerations .....	98
Associated Registers .....	64	Time-out Period .....	98
Output .....	63	WDT Reset, Normal Operation .....	91, 93, 94
Postscaler .....	63	WDT Reset, Sleep .....	91, 94
Prescaler .....	63	WDT Wake-up .....	93
Prescaler and Postscaler .....	63	WCOL .....	73
Timing Diagrams		Write Collision Detect Bit, WCOL .....	73
A/D Conversion .....	140	WWW Address .....	173
Brown-out Reset .....	131	WWW, On-Line Support .....	3
Capture/Compare/PWM (CCP1) .....	133		
CLKO and I/O .....	130		
External Clock .....	129		
I <sup>2</sup> C Bus Data .....	137		
I <sup>2</sup> C Bus Start/Stop Bits .....	136		
I <sup>2</sup> C Reception (7-Bit Address) .....	78		
I <sup>2</sup> C Transmission (7-Bit Address) .....	78		
PWM Output .....	68		
Reset, Watchdog Timer, Oscillator Start-up			
Timer and Power-up Timer .....	131		
Slow Rise Time (MCLR Tied to Vdd			
Through RC Network) .....	96		
SPI Master Mode .....	75		
SPI Master Mode (CKE = 0, SMP = 0) .....	134		
SPI Master Mode (CKE = 1, SMP = 1) .....	134		
SPI Slave Mode (CKE = 0) .....	75, 135		
SPI Slave Mode (CKE = 1) .....	75, 135		
Time-out Sequence on Power-up (MCLR			
Tied to Vdd Through Pull-up Resistor) .....	95		
Time-out Sequence on Power-up (MCLR			
Tied to Vdd Through RC Network): Case 1 .....	95		
Time-out Sequence on Power-up (MCLR			
Tied to Vdd Through RC Network): Case 2 .....	95		
Timer0 and Timer1 External Clock .....	132		
Timer1 Incrementing Edge .....	58		
Wake-up from Sleep through Interrupt .....	100		
Timing Parameter Symbolology .....	128		
Timing Requirements			
External Clock .....	129		
TMR0 Register .....	15		
TMR1CS Bit .....	57		
TMR1H Register .....	13		
TMR1L Register .....	13		
TMR1ON Bit .....	57		
TMR2 Register .....	13		
TMR2ON Bit .....	64		
TOUTPS0 Bit .....	64		
TOUTPS1 Bit .....	64		
TOUTPS2 Bit .....	64		
TOUTPS3 Bit .....	64		

# PIC16F818/819

---

NOTES: