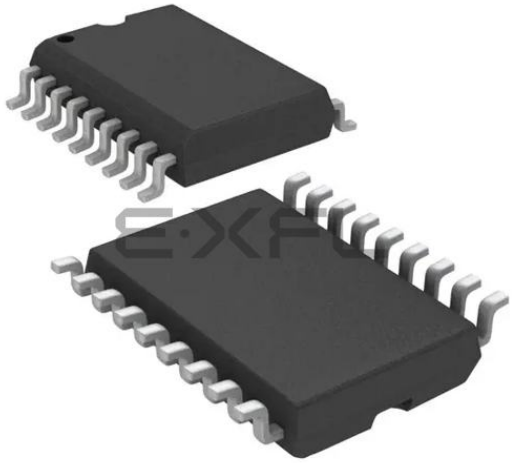


Welcome to E-XFL.COM

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	Obsolete
Core Processor	PIC
Core Size	8-Bit
Speed	20MHz
Connectivity	I²C, SPI
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	16
Program Memory Size	3.5KB (2K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	4V ~ 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	https://www.e-xfl.com/product-detail/microchip-technology/pic16f819t-i-sotsl

TABLE 1-2: PIC16F818/819 PINOUT DESCRIPTIONS (CONTINUED)

Pin Name	PDIP/ SOIC Pin#	SSOP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
						PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.
RB0/INT	6	7	7			
RB0				I/O	TTL	Bidirectional I/O pin.
INT				I	ST(1)	External interrupt pin.
RB1/SDI/SDA	7	8	8			
RB1				I/O	TTL	Bidirectional I/O pin.
SDI				I	ST	SPI data in.
SDA				I/O	ST	I^2C™ data.
RB2/SDO/CCP1	8	9	9			
RB2				I/O	TTL	Bidirectional I/O pin.
SDO				O	ST	SPI data out.
CCP1				I/O	ST	Capture input, Compare output, PWM output.
RB3/CCP1/PGM	9	10	10			
RB3				I/O	TTL	Bidirectional I/O pin.
CCP1				I/O	ST	Capture input, Compare output, PWM output.
PGM				I	ST	Low-Voltage ICSP™ Programming enable pin.
RB4/SCK/SCL	10	11	12			
RB4				I/O	TTL	Bidirectional I/O pin. Interrupt-on-change pin.
SCK				I/O	ST	Synchronous serial clock input/output for SPI.
SCL				I	ST	Synchronous serial clock input for I^2C.
RB5/$\overline{SS}$	11	12	13			
RB5				I/O	TTL	Bidirectional I/O pin. Interrupt-on-change pin.
$\overline{SS}$				I	TTL	Slave select for SPI in Slave mode.
RB6/T1OSO/T1CKI/PGC	12	13	15			
RB6				I/O	TTL	Interrupt-on-change pin.
T1OSO				O	ST	Timer1 Oscillator output.
T1CKI				I	ST	Timer1 clock input.
PGC				I	ST(2)	In-circuit debugger and ICSP programming clock pin.
RB7/T1OSI/PGD	13	14	16			
RB7				I/O	TTL	Interrupt-on-change pin.
T1OSI				I	ST	Timer1 oscillator input.
PGD				I	ST(2)	In-circuit debugger and ICSP programming data pin.
VSS	5	5, 6	3, 5	P	–	Ground reference for logic and I/O pins.
VDD	14	15, 16	17, 19	P	–	Positive supply for logic and I/O pins.

Legend: I = Input O = Output I/O = Input/Output P = Power
– = Not used TTL = TTL Input ST = Schmitt Trigger Input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.

2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.

3: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

2.0 MEMORY ORGANIZATION

There are two memory blocks in the PIC16F818/819. These are the program memory and the data memory. Each block has its own bus, so access to each block can occur during the same oscillator cycle.

The data memory can be further broken down into the general purpose RAM and the Special Function Registers (SFRs). The operation of the SFRs that control the “core” are described here. The SFRs used to control the peripheral modules are described in the section discussing each individual peripheral module.

The data memory area also contains the data EEPROM memory. This memory is not directly mapped into the data memory but is indirectly mapped. That is, an indirect address pointer specifies the address of the data EEPROM memory to read/write. The PIC16F818 device’s 128 bytes of data EEPROM memory have the address range of 00h-7Fh and the PIC16F819 device’s 256 bytes of data EEPROM memory have the address range of 00h-FFh. More details on the EEPROM memory can be found in **Section 3.0 “Data EEPROM and Flash Program Memory”**.

Additional information on device memory may be found in the *“PIC® Mid-Range Reference Manual”* (DS33023).

2.1 Program Memory Organization

The PIC16F818/819 devices have a 13-bit program counter capable of addressing an 8K x 14 program memory space. For the PIC16F818, the first 1K x 14 (0000h-03FFh) is physically implemented (see Figure 2-1). For the PIC16F819, the first 2K x 14 is located at 0000h-07FFh (see Figure 2-2). Accessing a location above the physically implemented address will cause a wraparound. For example, the same instruction will be accessed at locations 020h, 420h, 820h, C20h, 1020h, 1420h, 1820h and 1C20h.

The Reset vector is at 0000h and the interrupt vector is at 0004h.

FIGURE 2-1: PROGRAM MEMORY MAP AND STACK FOR PIC16F818

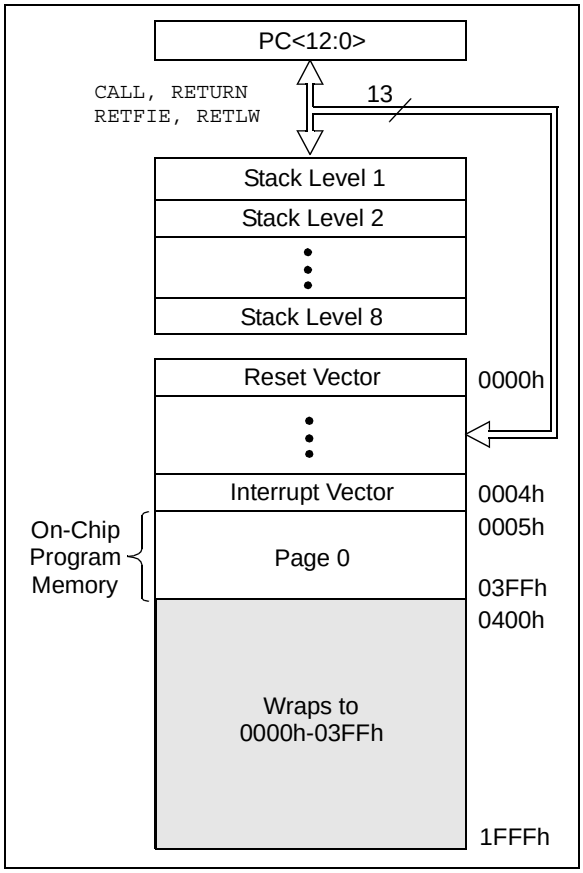
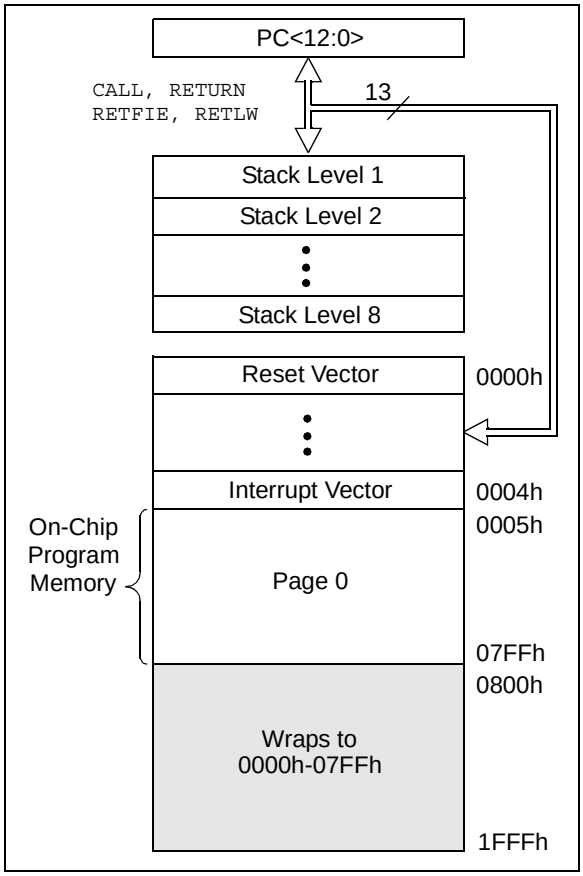


FIGURE 2-2: PROGRAM MEMORY MAP AND STACK FOR PIC16F819



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

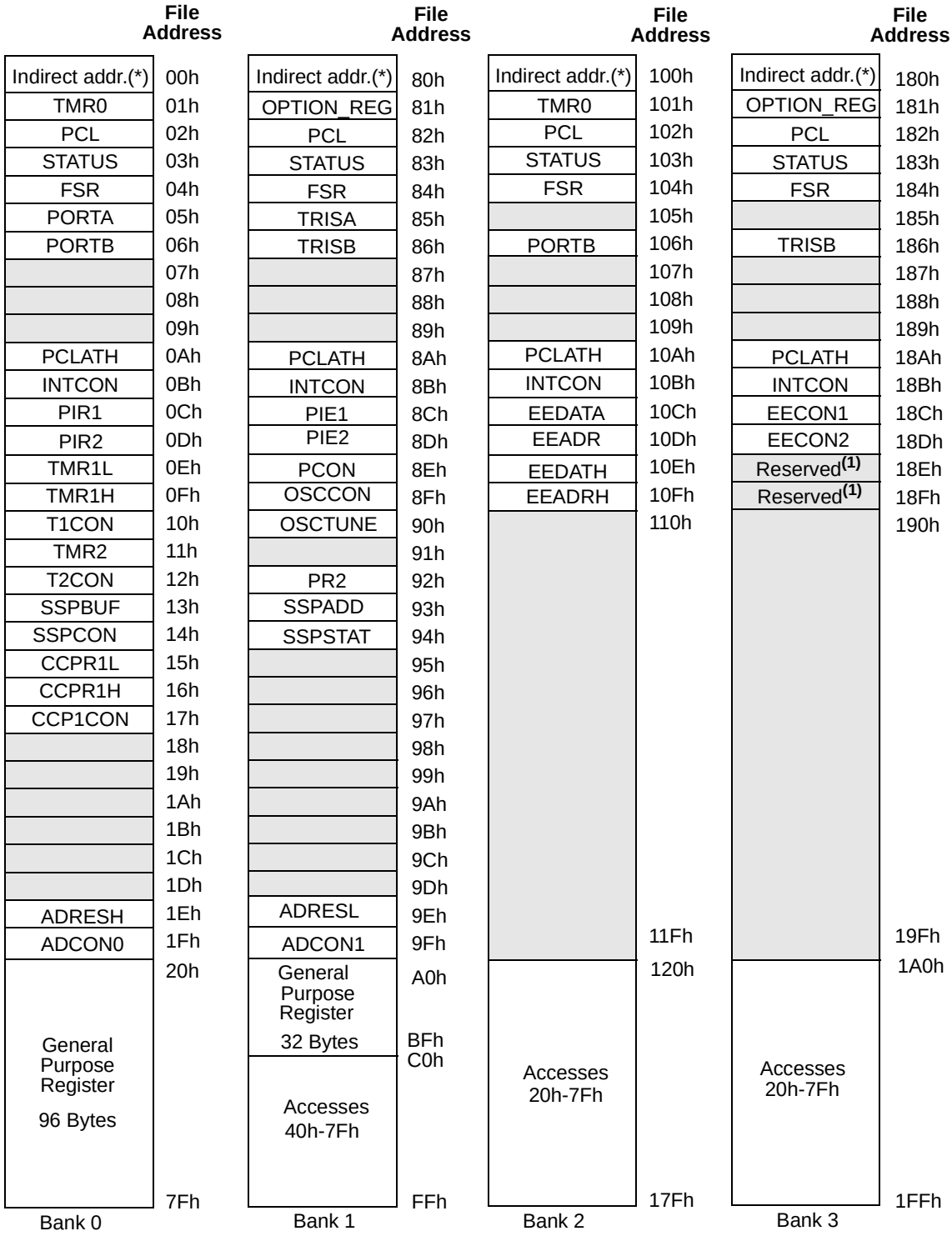
Each bank extends up to 7Fh (128 bytes). The lower locations of each bank are reserved for the Special Function Registers. Above the Special Function Registers are the General Purpose Registers, implemented as static RAM. All implemented banks contain SFRs. Some “high use” SFRs from one bank may be mirrored 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.

FIGURE 2-3: PIC16F818 REGISTER FILE MAP



Unimplemented data memory locations, read as '0'.

* Not a physical register.

Note 1: These registers are reserved; maintain these registers clear.

2.2.2.5 PIR1 Register

This register contains the individual flag bits for the peripheral interrupts.

Note:	Interrupt flag bits are 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 2-5: PIR1: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 1 (ADDRESS 0Ch)

U-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	ADIF	—	—	SSPIF	CCP1IF	TMR2IF	TMR1IF
bit 7							bit 0

bit 7 **Unimplemented:** Read as '0'

bit 6 **ADIF:** A/D Converter Interrupt Flag bit
1 = An A/D conversion completed
0 = The A/D conversion is not complete

bit 5-4 **Unimplemented:** Read as '0'

bit 3 **SSPIF:** Synchronous Serial Port (SSP) Interrupt Flag bit
1 = The SSP interrupt condition has occurred and must be cleared in software before returning from the Interrupt Service Routine. The conditions that will set this bit are a transmission/ reception has taken place.
0 = No SSP interrupt condition has occurred

bit 2 **CCP1IF:** CCP1 Interrupt Flag bit
Capture mode:
1 = A TMR1 register capture occurred (must be cleared in software)
0 = No TMR1 register capture occurred
Compare mode:
1 = A TMR1 register compare match occurred (must be cleared in software)
0 = No TMR1 register compare match occurred
PWM mode:
Unused in this mode.

bit 1 **TMR2IF:** TMR2 to PR2 Match Interrupt Flag bit
1 = TMR2 to PR2 match occurred (must be cleared in software)
0 = No TMR2 to PR2 match occurred

bit 0 **TMR1IF:** TMR1 Overflow Interrupt Flag bit
1 = TMR1 register overflowed (must be cleared in software)
0 = TMR1 register did not overflow

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

REGISTER 3-1: EECON1: EEPROM ACCESS CONTROL REGISTER 1 (ADDRESS 18Ch)

R/W-x	U-0	U-0	R/W-x	R/W-x	R/W-0	R/S-0	R/S-0
EEPGD	—	—	FREE	WRERR	WREN	WR	RD
bit 7							bit 0

bit 7 **EEPGD**: Program/Data EEPROM Select bit
1 = Accesses program memory
0 = Accesses data memory
Reads '0' after a POR; this bit cannot be changed while a write operation is in progress.

bit 6-5 **Unimplemented:** Read as '0'

bit 4 **FREE:** EEPROM Forced Row Erase bit
1 = Erase the program memory row addressed by EEADRH:EEADR on the next WR command
0 = Perform write-only

bit 3 **WRERR:** EEPROM Error Flag bit
1 = A write operation is prematurely terminated (any $\overline{\text{MCLR}}$ or any WDT Reset during normal operation)
0 = The write operation completed

bit 2 **WREN:** EEPROM Write Enable bit
1 = Allows write cycles
0 = Inhibits write to the EEPROM

bit 1 **WR:** Write Control bit
1 = Initiates a write cycle. The bit is cleared by hardware once write is complete. The WR bit can only be set (not cleared) in software.
0 = Write cycle to the EEPROM is complete

bit 0 **RD:** Read Control bit
1 = Initiates an EEPROM read, RD is cleared in hardware. The RD bit can only be set (not cleared) in software.
0 = Does not initiate an EEPROM read

Legend:			
R = Readable bit	W = Writable bit	S = Set only	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

An example of the complete four-word write sequence is shown in Example 3-5. The initial address is loaded into the EEADRH:EEADR register pair; the four words of data are loaded using indirect addressing, assuming that a row erase sequence has already been performed.

EXAMPLE 3-5: WRITING TO FLASH PROGRAM MEMORY

```
; This write routine assumes the following:

; 1. The 32 words in the erase block have already been erased.
; 2. A valid starting address (the least significant bits = '00') is loaded into EEADRH:EEADR
; 3. This example is starting at 0x100, this is an application dependent setting.
; 4. The 8 bytes (4 words) of data are loaded, starting at an address in RAM called ARRAY.
; 5. This is an example only, location of data to program is application dependent.
; 6. word_block is located in data memory.

        BANKSEL  EECON1            ;prepare for WRITE procedure
        BSF      EECON1, EEPGD     ;point to program memory
        BSF      EECON1, WREN      ;allow write cycles
        BCF      EECON1, FREE      ;perform write only

        BANKSEL  word_block
        MOVLW    .4
        MOVWF    word_block        ;prepare for 4 words to be written

        BANKSEL  EEADRH            ;Start writing at 0x100
        MOVLW    0x01
        MOVWF    EEADRH            ;load HIGH address
        MOVLW    0x00
        MOVWF    EEADR             ;load LOW address
        BANKSEL  ARRAY
        MOVLW    ARRAY             ;initialize FSR to start of data
        MOVWF    FSR
LOOP
        BANKSEL  EEDATA
        MOVF     INDF, W           ;indirectly load EEDATA
        MOVWF    EEDATA
        INCF     FSR, F            ;increment data pointer
        MOVF     INDF, W           ;indirectly load EEDATH
        MOVWF    EEDATH
        INCF     FSR, F            ;increment data pointer

        BANKSEL  EECON1
        MOVLW    0x55              ;required sequence
        MOVWF    EECON2
        MOVLW    0xAA
        MOVWF    EECON2
        BSF      EECON1, WR        ;set WR bit to begin write
        NOP                        ;instructions here are ignored as processor
        NOP

        BANKSEL  EEADR
        INCF     EEADR, f          ;load next word address
        BANKSEL  word_block
        DECFSZ   word_block, f     ;have 4 words been written?
        GOTO     loop              ;NO, continue with writing

        BANKSEL  EECON1
        BCF      EECON1, WREN      ;YES, 4 words complete, disable writes
        BSF      INTCON, GIE       ;enable interrupts
```

Required Sequence

FIGURE 5-1: BLOCK DIAGRAM OF RA0/AN0:RA1/AN1 PINS

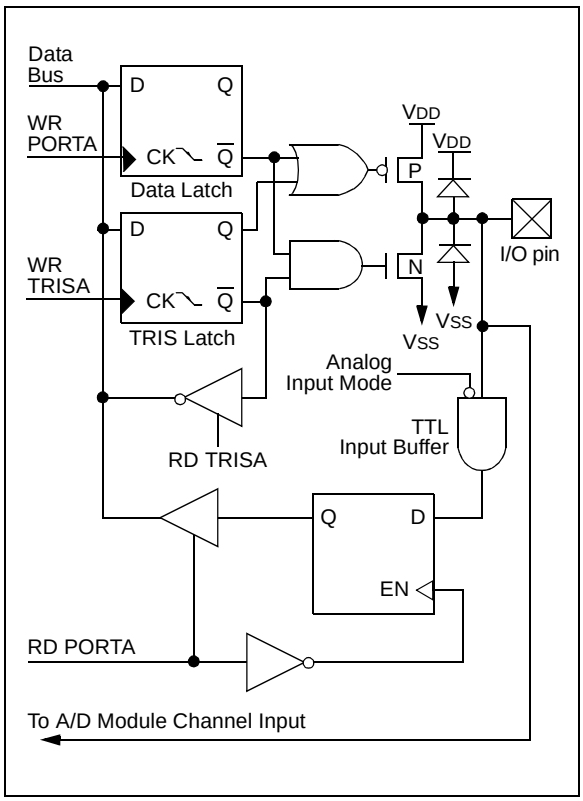


FIGURE 5-2: BLOCK DIAGRAM OF RA3/AN3/VREF+ PIN

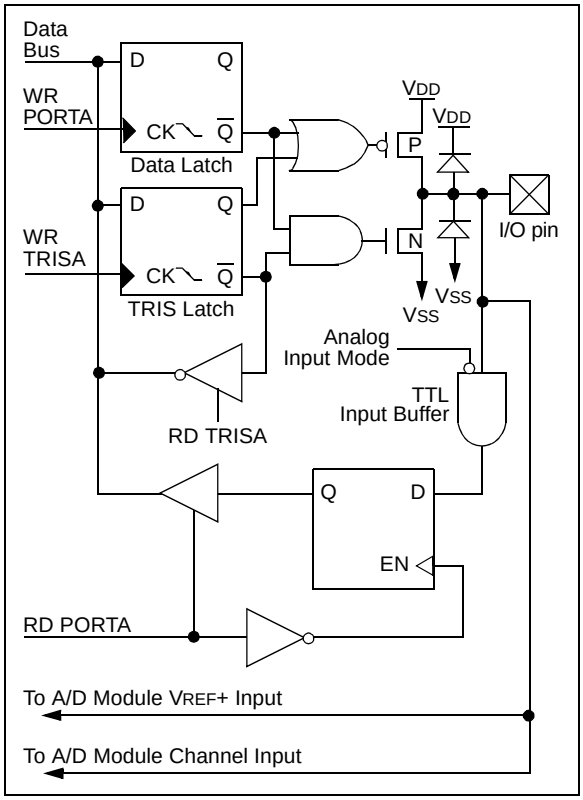


FIGURE 5-3: BLOCK DIAGRAM OF RA2/AN2/VREF- PIN

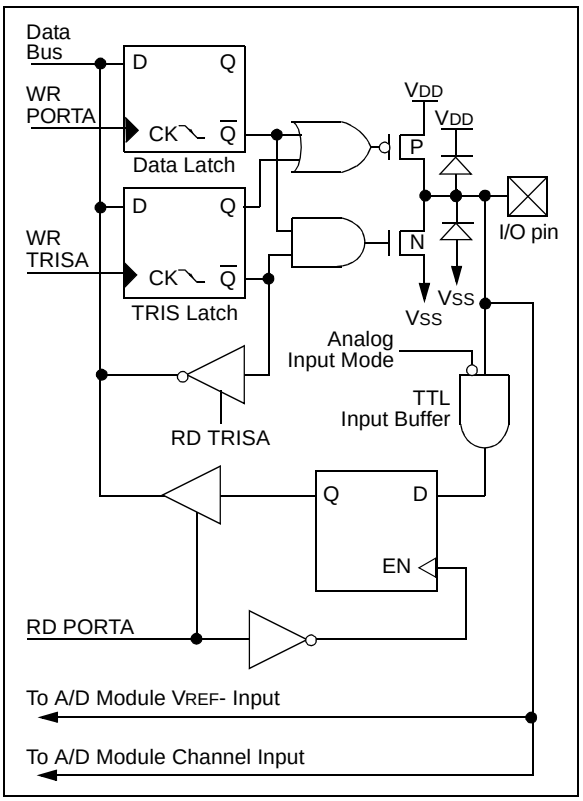


FIGURE 5-4: BLOCK DIAGRAM OF RA4/AN4/T0CKI PIN

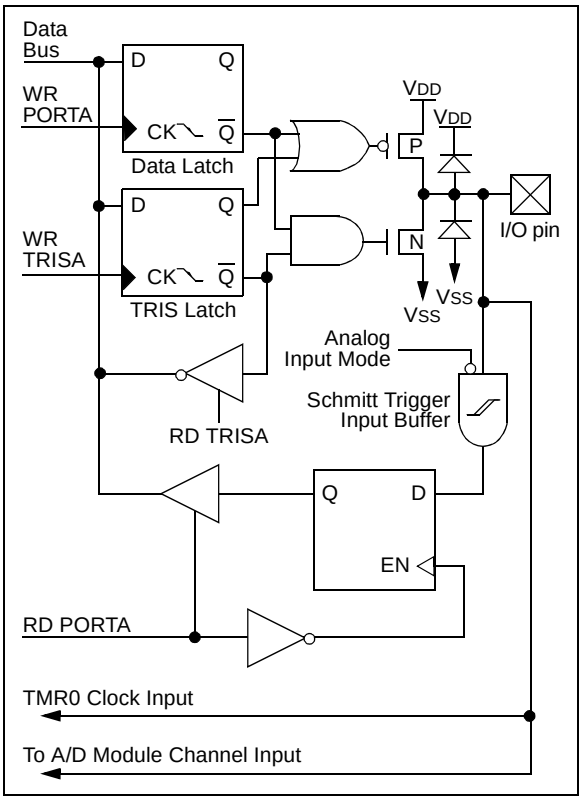


TABLE 5-3: PORTB FUNCTIONS

Name	Bit#	Buffer	Function
RB0/INT	bit 0	TTL/ST(1)	Input/output pin or external interrupt input. Internal software programmable weak pull-up.
RB1/SDI/SDA	bit 1	TTL/ST(5)	Input/output pin, SPI data input pin or I^2C™ data I/O pin. Internal software programmable weak pull-up.
RB2/SDO/CCP1	bit 2	TTL/ST(4)	Input/output pin, SPI data output pin or Capture input/Compare output/PWM output pin. Internal software programmable weak pull-up.
RB3/CCP1/PGM(3)	bit 3	TTL/ST(2)	Input/output pin, Capture input/Compare output/PWM output pin or programming in LVP mode. Internal software programmable weak pull-up.
RB4/SCK/SCL	bit 4	TTL/ST(5)	Input/output pin or SPI and I^2C clock pin (with interrupt-on-change). Internal software programmable weak pull-up.
RB5/$\overline{SS}$	bit 5	TTL	Input/output pin or SPI slave select pin (with interrupt-on-change). Internal software programmable weak pull-up.
RB6/T1OSO/T1CKI/ PGC	bit 6	TTL/ST(2)	Input/output pin, Timer1 oscillator output pin, Timer1 clock input pin or serial programming clock (with interrupt-on-change). Internal software programmable weak pull-up.
RB7/T1OSI/PGD	bit 7	TTL/ST(2)	Input/output pin, Timer1 oscillator input pin or serial programming data (with interrupt-on-change). Internal software programmable weak pull-up.

Legend: TTL = TTL input, ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.

2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.

3: Low-Voltage ICSP™ Programming (LVP) is enabled by default which disables the RB3 I/O function. LVP must be disabled to enable RB3 as an I/O pin and allow maximum compatibility to the other 18-pin mid-range devices.

4: This buffer is a Schmitt Trigger input when configured for CCP or SSP mode.

5: This buffer is a Schmitt Trigger input when configured for SPI or I^2C mode.

TABLE 5-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB

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
06h, 106h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	uuuu uuuu
86h, 186h	TRISB	PORTB Data Direction Register								1111 1111	1111 1111
81h, 181h	OPTION_REG	$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

Legend: x = unknown, u = unchanged. Shaded cells are not used by PORTB.

EXAMPLE 6-1: CHANGING THE PRESCALER ASSIGNMENT FROM TIMER0 TO WDT

```
    BANKSEL  OPTION_REG     ; Select Bank of OPTION_REG
    MOVLW    b'xx0x0xxx'    ; Select clock source and prescale value of
    MOVWF    OPTION_REG     ; other than 1:1
    BANKSEL  TMR0           ; Select Bank of TMR0
    CLRF     TMR0           ; Clear TMR0 and prescaler
    BANKSEL  OPTION_REG     ; Select Bank of OPTION_REG
    MOVLW    b'xxxx1xxx'    ; Select WDT, do not change prescale value
    MOVWF    OPTION_REG
    CLRWDT                  ; Clears WDT and prescaler
    MOVLW    b'xxxx1xxx'    ; Select new prescale value and WDT
    MOVWF    OPTION_REG
```

EXAMPLE 6-2: CHANGING THE PRESCALER ASSIGNMENT FROM WDT TO TIMER0

```
CLRWDT                  ; Clear WDT and prescaler
BANKSEL  OPTION_REG     ; Select Bank of OPTION_REG
MOVLW    b'xxxx0xxx'    ; Select TMR0, new prescale
MOVWF    OPTION_REG     ; value and clock source
```

TABLE 6-1: REGISTERS ASSOCIATED WITH TIMER0

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
01h,101h	TMR0	Timer0 Module Register								xxxx xxxx	uuuu uuuu
0Bh,8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
81h,181h	OPTION_REG	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'. Shaded cells are not used by Timer0.

7.6 Timer1 Oscillator

A crystal oscillator circuit is built-in between pins T1OSI (input) and T1OSO (amplifier output). It is enabled by setting control bit, T1OSCEN (T1CON<3>). The oscillator is a low-power oscillator, rated up to 32.768 kHz. It will continue to run during Sleep. It is primarily intended for a 32 kHz crystal. The circuit for a typical LP oscillator is shown in Figure 7-3. Table 7-1 shows the capacitor selection for the Timer1 oscillator.

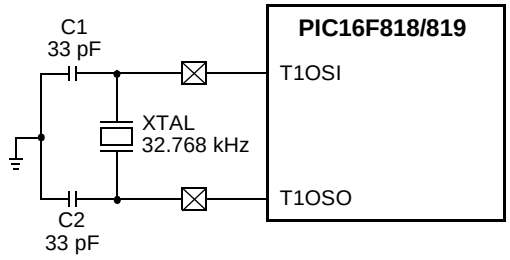
The user must provide a software time delay to ensure proper oscillator start-up.

Note: The Timer1 oscillator shares the T1OSI and T1OSO pins with the PGD and PGC pins used for programming and debugging.

When using the Timer1 oscillator, In-Circuit Serial Programming™ (ICSP™) may not function correctly (high-voltage or low-voltage) or the In-Circuit Debugger (ICD) may not communicate with the controller. As a result of using either ICSP or ICD, the Timer1 crystal may be damaged.

If ICSP or ICD operations are required, the crystal should be disconnected from the circuit (disconnect either lead) or installed after programming. The oscillator loading capacitors may remain in-circuit during ICSP or ICD operation.

FIGURE 7-3: EXTERNAL COMPONENTS FOR THE TIMER1 LP OSCILLATOR



Note: See the Notes with Table 7-1 for additional information about capacitor selection.

TABLE 7-1: CAPACITOR SELECTION FOR THE TIMER1 OSCILLATOR

Osc Type	Freq	C1	C2
LP	32 kHz	33 pF	33 pF

Note 1: Microchip suggests this value as a starting point in validating the oscillator circuit.

2: Higher capacitance increases the stability of the oscillator but also increases the start-up time.

3: Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components.

4: Capacitor values are for design guidance only.

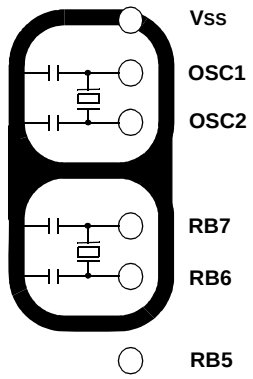
7.7 Timer1 Oscillator Layout Considerations

The Timer1 oscillator circuit draws very little power during operation. Due to the low-power nature of the oscillator, it may also be sensitive to rapidly changing signals in close proximity.

The oscillator circuit, shown in Figure 7-3, should be located as close as possible to the microcontroller. There should be no circuits passing within the oscillator circuit boundaries other than VSS or VDD.

If a high-speed circuit must be located near the oscillator, a grounded guard ring around the oscillator circuit, as shown in Figure 7-4, may be helpful when used on a single-sided PCB or in addition to a ground plane.

FIGURE 7-4: OSCILLATOR CIRCUIT WITH GROUNDED GUARD RING



EXAMPLE 7-3: IMPLEMENTING A REAL-TIME CLOCK USING A TIMER1 INTERRUPT SERVICE

```
RTCinit     BANKSEL   TMR1H
            MOVLW     0x80              ; Preload TMR1 register pair
            MOVWF     TMR1H             ; for 1 second overflow
            CLRF      TMR1L
            MOVLW     b'00001111'       ; Configure for external clock,
            MOVWF     T1CON             ; Asynchronous operation, external oscillator
            CLRF      secs              ; Initialize timekeeping registers
            CLRF      mins
            MOVLW     .12
            MOVWF     hours
            BANKSEL   PIE1
            BSF       PIE1, TMR1IE      ; Enable Timer1 interrupt
            RETURN
RTCisr      BANKSEL   TMR1H
            BSF       TMR1H, 7          ; Preload for 1 sec overflow
            BCF       PIR1, TMR1IF      ; Clear interrupt flag
            INCF      secs, F           ; Increment seconds
            MOVF      secs, w
            SUBLW     .60
            BTFSS     STATUS, Z         ; 60 seconds elapsed?
            RETURN                      ; No, done
            CLRF      seconds           ; Clear seconds
            INCF      mins, f           ; Increment minutes
            MOVF      mins, w
            SUBLW     .60
            BTFSS     STATUS, Z         ; 60 seconds elapsed?
            RETURN                      ; No, done
            CLRF      mins              ; Clear minutes
            INCF      hours, f          ; Increment hours
            MOVF      hours, w
            SUBLW     .24
            BTFSS     STATUS, Z         ; 24 hours elapsed?
            RETURN                      ; No, done
            CLRF      hours             ; Clear hours
            RETURN                      ; Done
```

TABLE 7-2: REGISTERS ASSOCIATED WITH TIMER1 AS A TIMER/COUNTER

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
0Bh,8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	—	ADIF	—	—	SSPIF	CCP1IF	TMR2IF	TMR1IF	-0-- 0000	-0-- 0000
8Ch	PIE1	—	ADIE	—	—	SSPIE	CCP1IE	TMR2IE	TMR1IE	-0-- 0000	-0-- 0000
0Eh	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
0Fh	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	--00 0000	--uu uuuu

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by the Timer1 module.

REGISTER 10-1: SSPSTAT: SYNCHRONOUS SERIAL PORT STATUS REGISTER (ADDRESS 94h)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF
bit 7							bit 0

bit 7 **SMP:** SPI Data Input Sample Phase bit

SPI Master mode:

1 = Input data sampled at end of data output time

0 = Input data sampled at middle of data output time (Microwire)

SPI Slave mode:

This bit must be cleared when SPI is used in Slave mode.

I^2C mode:

This bit must be maintained clear.

bit 6 **CKE:** SPI Clock Edge Select bit

1 = Transmit occurs on transition from active to Idle clock state

0 = Transmit occurs on transition from Idle to active clock state

Note: Polarity of clock state is set by the CKP bit (SSPCON<4>).

I^2C mode:

This bit must be maintained clear.

bit 5 **D/$\overline{A}$:** Data/$\overline{\text{Address}}$ bit (I^2C mode only)

In I^2C Slave mode:

1 = Indicates that the last byte received was data

0 = Indicates that the last byte received was address

bit 4 **P:** Stop bit**(1)** (I^2C mode only)

1 = Indicates that a Stop bit has been detected last

0 = Stop bit was not detected last

bit 3 **S:** Start bit**(1)** (I^2C mode only)

1 = Indicates that a Start bit has been detected last (this bit is '0' on Reset)

0 = Start bit was not detected last

bit 2 **R/$\overline{W}$:** Read/$\overline{\text{Write}}$ Information bit (I^2C mode only)

Holds the R/$\overline{W}$ bit information following the last address match and is only valid from address match to the next Start bit, Stop bit or $\overline{\text{ACK}}$ bit.

1 = Read

0 = Write

bit 1 **UA:** Update Address bit (10-bit I^2C mode only)

1 = Indicates that the user needs to update the address in the SSPADD register

0 = Address does not need to be updated

bit 0 **BF:** Buffer Full Status bit

Receive (SPI and I^2C modes):

1 = Receive complete, SSPBUF is full

0 = Receive not complete, SSPBUF is empty

Transmit (In I^2C mode only):

1 = Transmit in progress, SSPBUF is full (8 bits)

0 = Transmit complete, SSPBUF is empty

Note 1: This bit is cleared when the SSP module is disabled (i.e., the SSPEN bit is cleared).

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

FIGURE 10-2: SPI MODE TIMING, MASTER MODE

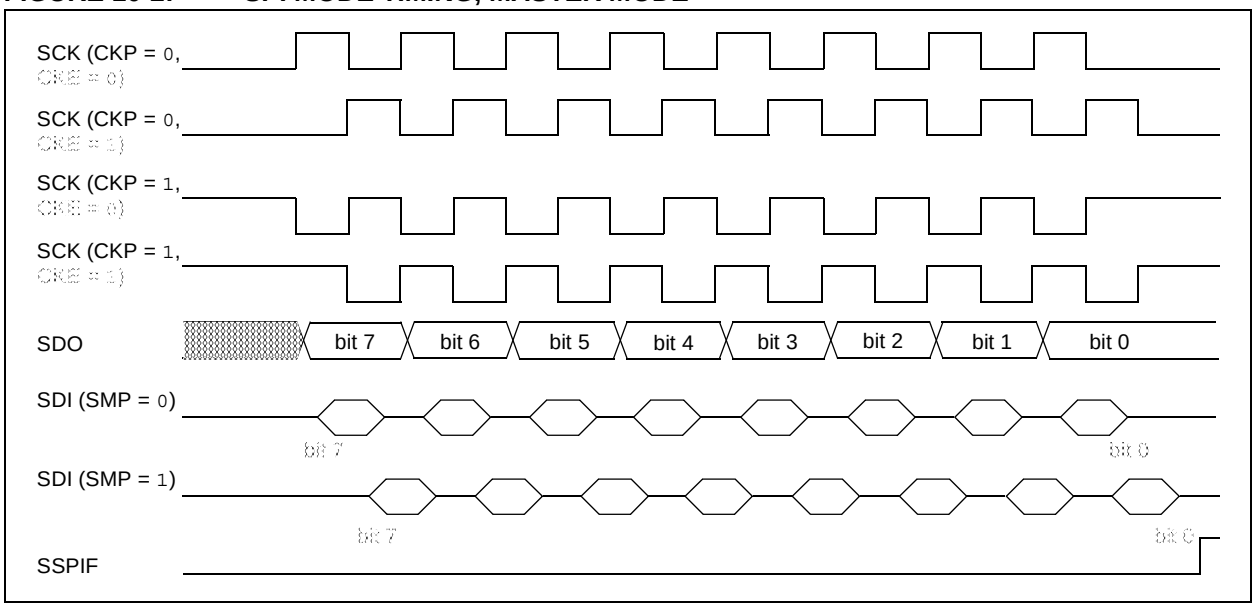


FIGURE 10-3: SPI MODE TIMING (SLAVE MODE WITH CKE = 0)

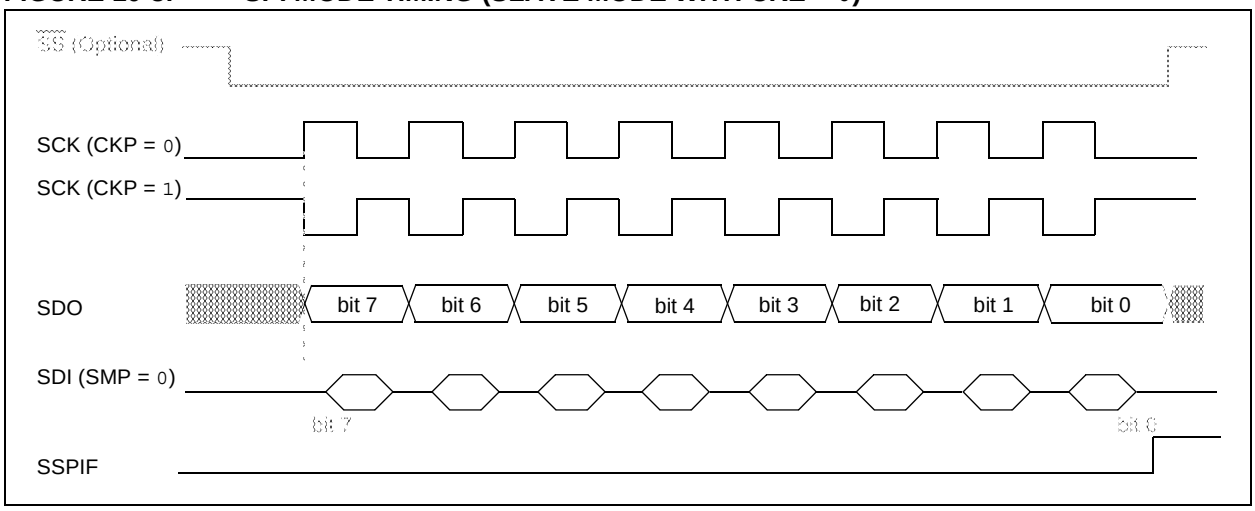
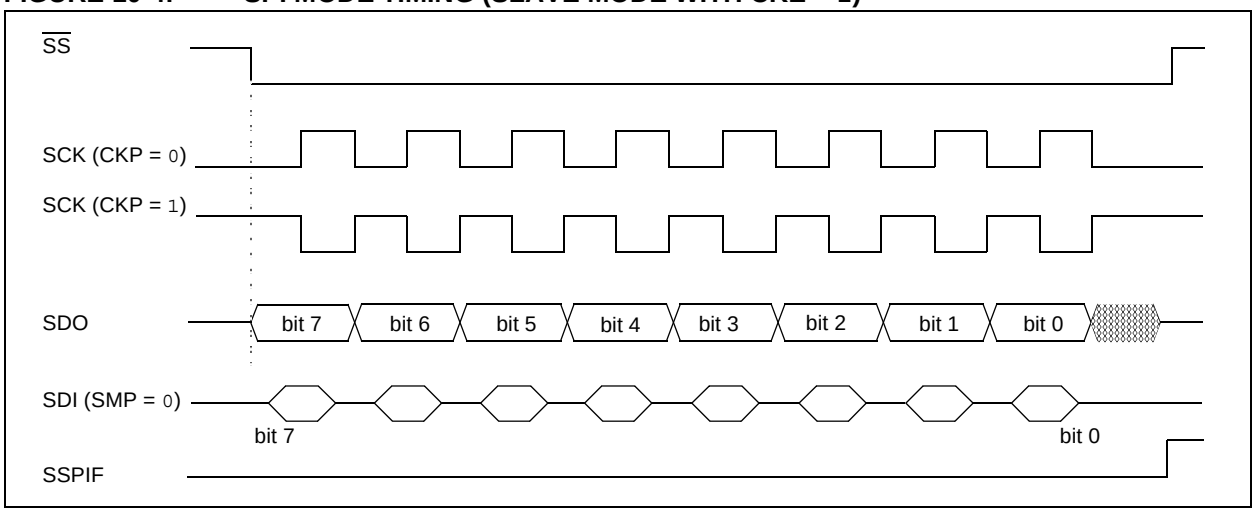


FIGURE 10-4: SPI MODE TIMING (SLAVE MODE WITH CKE = 1)



REGISTER 11-2: ADCON1: A/D CONTROL REGISTER 1 (ADDRESS 9Fh)

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	ADCS2	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

bit 7 **ADFM:** A/D Result Format Select bit
1 = Right justified, 6 Most Significant bits of ADRESH are read as '0'
0 = Left justified, 6 Least Significant bits of ADRESL are read as '0'

bit 6 **ADCS2:** A/D Clock Divide by 2 Select bit
1 = A/D clock source is divided by 2 when system clock is used
0 = Disabled

bit 5-4 **Unimplemented:** Read as '0'

bit 3-0 **PCFG<3:0>:** A/D Port Configuration Control bits

PCFG	AN4	AN3	AN2	AN1	AN0	VREF+	VREF-	C/R
0000	A	A	A	A	A	AVDD	AVSS	5/0
0001	A	VREF+	A	A	A	AN3	AVSS	4/1
0010	A	A	A	A	A	AVDD	AVSS	5/0
0011	A	VREF+	A	A	A	AN3	AVSS	4/1
0100	D	A	D	A	A	AVDD	AVSS	3/0
0101	D	VREF+	D	A	A	AN3	AVSS	2/1
011x	D	D	D	D	D	AVDD	AVSS	0/0
1000	A	VREF+	VREF-	A	A	AN3	AN2	3/2
1001	A	A	A	A	A	AVDD	AVSS	5/0
1010	A	VREF+	A	A	A	AN3	AVSS	4/1
1011	A	VREF+	VREF-	A	A	AN3	AN2	3/2
1100	A	VREF+	VREF-	A	A	AN3	AN2	3/2
1101	D	VREF+	VREF-	A	A	AN3	AN2	2/2
1110	D	D	D	D	A	AVDD	AVSS	1/0
1111	D	VREF+	VREF-	D	A	AN3	AN2	1/2

A = Analog input D = Digital I/O
C/R = Number of analog input channels/Number of A/D voltage references

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

11.4 A/D Conversions

Clearing the GO/$\overline{\text{DONE}}$ bit during a conversion will abort the current conversion. The A/D Result register pair will NOT be updated with the partially completed A/D conversion sample. That is, the ADRESH:ADRESL registers will continue to contain the value of the last completed conversion (or the last value written to the ADRESH:ADRESL registers). After the A/D conversion is aborted, a 2-TAD wait is required before the next acquisition is started. After this 2-TAD wait, acquisition on the selected channel is automatically started. The GO/$\overline{\text{DONE}}$ bit can then be set to start the conversion.

In Figure 11-3, after the GO bit is set, the first time segment has a minimum of TCY and a maximum of TAD.

Note:	The GO/$\overline{\text{DONE}}$ bit should **NOT** be set in the same instruction that turns on the A/D.

11.4.1 A/D RESULT REGISTERS

The ADRESH:ADRESL register pair is the location where the 10-bit A/D result is loaded at the completion of the A/D conversion. This register pair is 16 bits wide. The A/D module gives the flexibility to left or right justify the 10-bit result in the 16-bit result register. The A/D Format Select bit (ADFM) controls this justification. Figure 11-4 shows the operation of the A/D result justification. The extra bits are loaded with '0's. When an A/D result will not overwrite these locations (A/D disable), these registers may be used as two general purpose 8-bit registers.

FIGURE 11-3: A/D CONVERSION TAD CYCLES

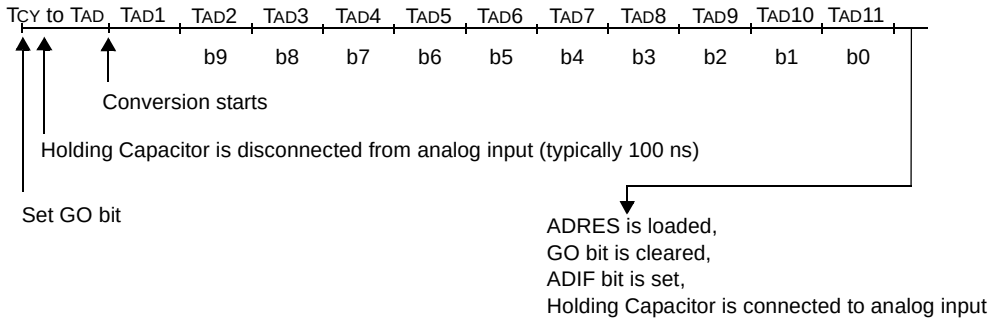
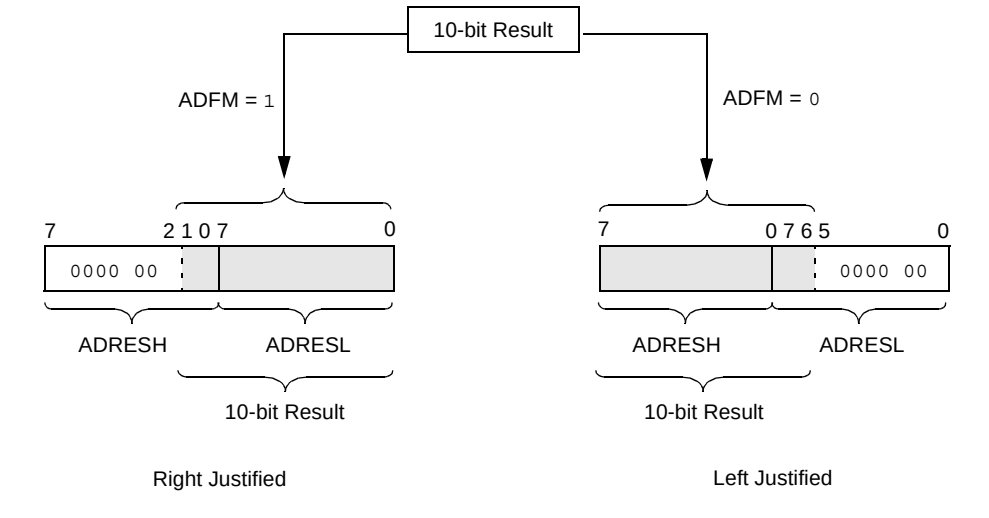


FIGURE 11-4: A/D RESULT JUSTIFICATION



BTFSS	Bit Test f, Skip if Set
Syntax:	[*label*] BTFSS f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b < 7
Operation:	skip if (f<b>) = 1
Status Affected:	None
Description:	If bit 'b' in register 'f' = 0, the next instruction is executed. If bit 'b' = 1, then the next instruction is discarded and a NOP is executed instead, making this a 2 TCY instruction.

BTFSC	Bit Test, Skip if Clear
Syntax:	[*label*] BTFSC f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b ≤ 7
Operation:	skip if (f<b>) = 0
Status Affected:	None
Description:	If bit 'b' in register 'f' = 1, the next instruction is executed. If bit 'b' in register 'f' = 0, the next instruction is discarded and a NOP is executed instead, making this a 2 TCY instruction.

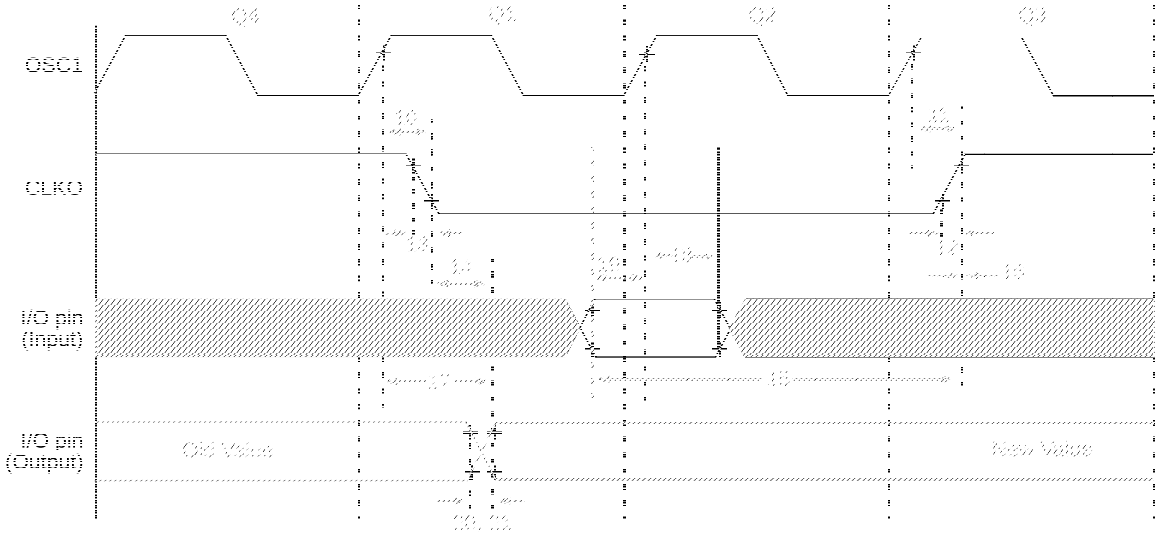
CALL	Call Subroutine
Syntax:	[*label*] CALL k
Operands:	0 ≤ k ≤ 2047
Operation:	(PC) + 1 → TOS, k → PC<10:0>, (PCLATH<4:3>) → PC<12:11>
Status Affected:	None
Description:	Call subroutine. First, return address (PC + 1) is pushed onto the stack. The eleven-bit immediate address is loaded into PC bits<10:0>. The upper bits of the PC are loaded from PCLATH. CALL is a two-cycle instruction.

CLRF	Clear f
Syntax:	[*label*] CLRF f
Operands:	0 ≤ f ≤ 127
Operation:	00h → (f) 1 → Z
Status Affected:	Z
Description:	The contents of register 'f' are cleared and the Z bit is set.

CLRW	Clear W
Syntax:	[*label*] CLRW
Operands:	None
Operation:	00h → (W) 1 → Z
Status Affected:	Z
Description:	W register is cleared. Zero bit (Z) is set.

CLRWDT	Clear Watchdog Timer
Syntax:	[*label*] CLRWDT
Operands:	None
Operation:	00h → WDT 0 → WDT prescaler, 1 → $\overline{TO}$ 1 → $\overline{PD}$
Status Affected:	$\overline{TO}$, $\overline{PD}$
Description:	CLRWDT instruction resets the Watchdog Timer. It also resets the prescaler of the WDT. Status bits $\overline{TO}$ and $\overline{PD}$ are set.

FIGURE 15-5: CLKO AND I/O TIMING



Note: Refer to Figure 15-3 for load conditions.

TABLE 15-2: CLKO AND I/O TIMING REQUIREMENTS

Param No.	Symbol	Characteristic		Min	Typ†	Max	Units	Conditions
10*	TosH2ckL	OSC1 ↑ to CLKO ↓		—	75	200	ns	**(Note 1)**
11*	TosH2ckH	OSC1 ↑ to CLKO ↑		—	75	200	ns	**(Note 1)**
12*	TckR	CLKO Rise Time		—	35	100	ns	**(Note 1)**
13*	TckF	CLKO Fall Time		—	35	100	ns	**(Note 1)**
14*	TckL2ioV	CLKO ↓ to Port Out Valid		—	—	0.5 Tcy + 20	ns	**(Note 1)**
15*	TioV2ckH	Port In Valid before CLKO ↑		Tosc + 200	—	—	ns	**(Note 1)**
16*	TckH2ioI	Port In Hold after CLKO ↑		0	—	—	ns	**(Note 1)**
17*	TosH2ioV	OSC1 ↑ (Q1 cycle) to Port Out Valid		—	100	255	ns	
18*	TosH2ioI	OSC1 ↑ (Q2 cycle) to Port Input Invalid (I/O in hold time)	PIC16**F**818/819	100	—	—	ns	
			PIC16**LF**818/819	200	—	—	ns	
19*	TioV2osH	Port Input Valid to OSC1 ↑ (I/O in setup time)		0	—	—	ns	
20*	TioR	Port Output Rise Time	PIC16**F**818/819	—	10	40	ns	
			PIC16**LF**818/819	—	—	145	ns	
21*	TioF	Port Output Fall Time	PIC16**F**818/819	—	10	40	ns	
			PIC16**LF**818/819	—	—	145	ns	
22††*	Tinp	INT pin High or Low Time		Tcy	—	—	ns	
23††*	Trbp	RB7:RB4 Change INT High or Low Time		Tcy	—	—	ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

†† These parameters are asynchronous events, not related to any internal clock edges.

Note 1: Measurements are taken in RC mode, where CLKO output is 4 x Tosc.

FIGURE 15-10: SPI MASTER MODE TIMING (CKE = 0, SMP = 0)

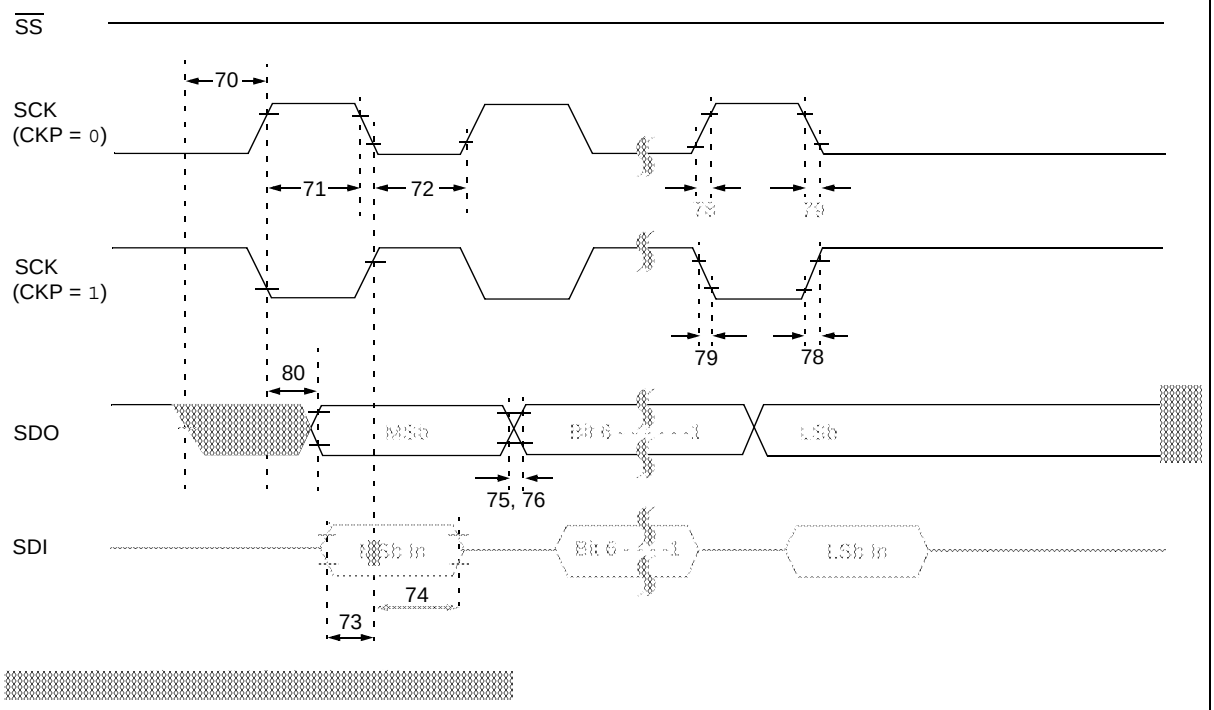
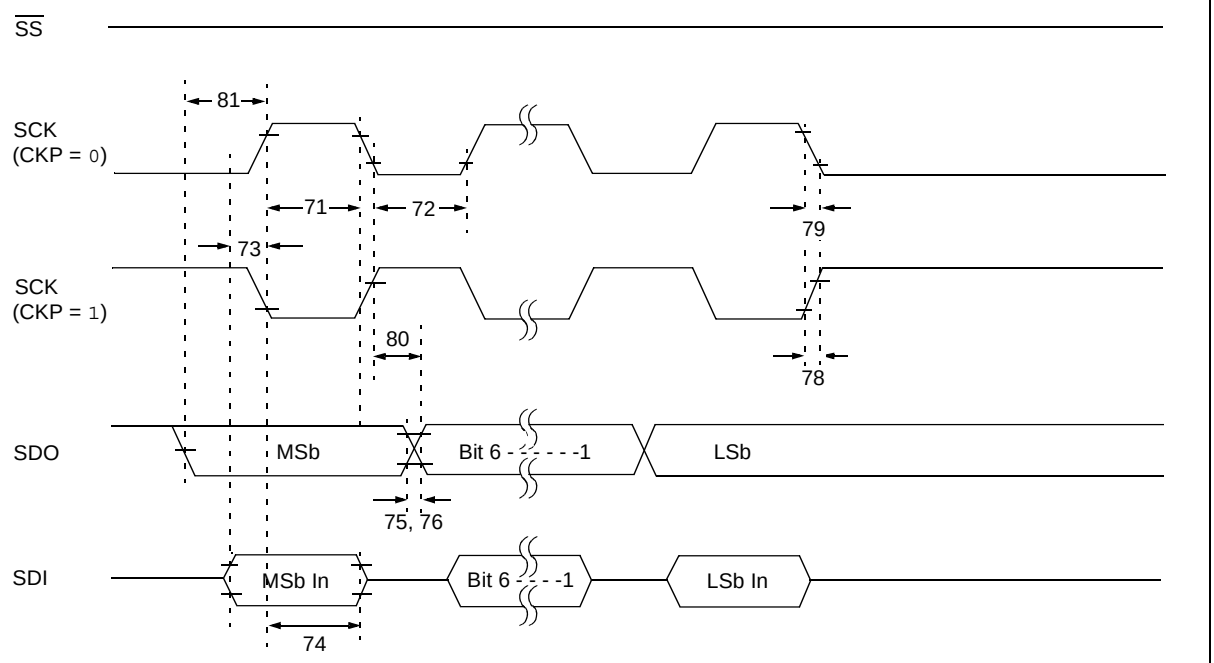


FIGURE 15-11: SPI MASTER MODE TIMING (CKE = 1, SMP = 1)



Note: Refer to Figure 15-3 for load conditions.

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Trademarks

Printed on recycled paper.

ISBN: 9781620769393

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
═ ISO/TS 16949 ═

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.