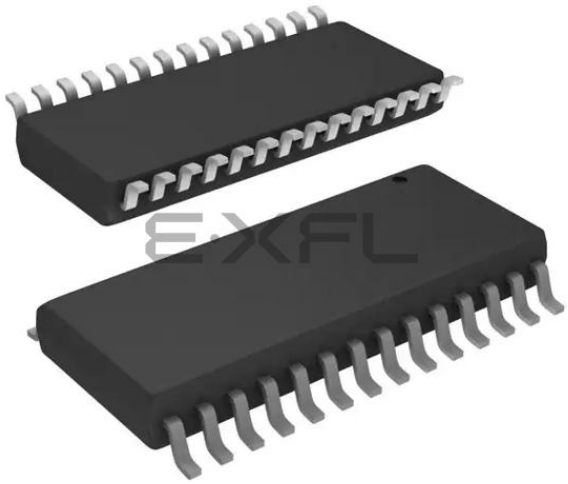
Microchip Technology - PIC16F876AT-I/SO Datasheet



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	20MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	22
Program Memory Size	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	368 x 8
Voltage - Supply (Vcc/Vdd)	4V ~ 5.5V
Data Converters	A/D 5x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f876at-i-so

TABLE 1-2: PIC16F873A/876A PINOUT DESCRIPTION (CONTINUED)

Pin Name	PDIP, SOIC, 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-ups on all inputs.
RB0/INT	21	18		TTL/ST(1)	
RB0			I/O		Digital I/O.
INT			I		External interrupt.
RB1	22	19	I/O	TTL	Digital I/O.
RB2	23	20	I/O	TTL	Digital I/O.
RB3/PGM	24	21		TTL	
RB3			I/O		Digital I/O.
PGM			I		Low-voltage (single-supply) ICSP programming enable pin.
RB4	25	22	I/O	TTL	Digital I/O.
RB5	26	23	I/O	TTL	Digital I/O.
RB6/PGC	27	24		TTL/ST(2)	
RB6			I/O		Digital I/O.
PGC			I		In-circuit debugger and ICSP programming clock.
RB7/PGD	28	25		TTL/ST(2)	
RB7			I/O		Digital I/O.
PGD			I/O		In-circuit debugger and ICSP programming data.
					PORTC is a bidirectional I/O port.
RC0/T1OSO/T1CKI	11	8		ST	
RC0			I/O		Digital I/O.
T1OSO			O		Timer1 oscillator output.
T1CKI			I		Timer1 external clock input.
RC1/T1OSI/CCP2	12	9		ST	
RC1			I/O		Digital I/O.
T1OSI			I		Timer1 oscillator input.
CCP2			I/O		Capture2 input, Compare2 output, PWM2 output.
RC2/CCP1	13	10		ST	
RC2			I/O		Digital I/O.
CCP1			I/O		Capture1 input, Compare1 output, PWM1 output.
RC3/SCK/SCL	14	11		ST	
RC3			I/O		Digital I/O.
SCK			I/O		Synchronous serial clock input/output for SPI mode.
SCL			I/O		Synchronous serial clock input/output for I^2C mode.
RC4/SDI/SDA	15	12		ST	
RC4			I/O		Digital I/O.
SDI			I		SPI data in.
SDA			I/O		I^2C data I/O.
RC5/SDO	16	13		ST	
RC5			I/O		Digital I/O.
SDO			O		SPI data out.
RC6/TX/CK	17	14		ST	
RC6			I/O		Digital I/O.
TX			O		USART asynchronous transmit.
CK			I/O		USART1 synchronous clock.
RC7/RX/DT	18	15		ST	
RC7			I/O		Digital I/O.
RX			I		USART asynchronous receive.
DT			I/O		USART synchronous data.
VSS	8, 19	5, 6	P	—	Ground reference for logic and I/O pins.
VDD	20	17	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.

TABLE 1-3: PIC16F874A/877A PINOUT DESCRIPTION (CONTINUED)

Pin Name	PDIP Pin#	PLCC Pin#	TQFP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
							PORTD is a bidirectional I/O port or Parallel Slave Port when interfacing to a microprocessor bus.
RD0/PSP0	19	21	38	38		ST/TTL(3)	
RD0					I/O		Digital I/O.
PSP0					I/O		Parallel Slave Port data.
RD1/PSP1	20	22	39	39		ST/TTL(3)	
RD1					I/O		Digital I/O.
PSP1					I/O		Parallel Slave Port data.
RD2/PSP2	21	23	40	40		ST/TTL(3)	
RD2					I/O		Digital I/O.
PSP2					I/O		Parallel Slave Port data.
RD3/PSP3	22	24	41	41		ST/TTL(3)	
RD3					I/O		Digital I/O.
PSP3					I/O		Parallel Slave Port data.
RD4/PSP4	27	30	2	2		ST/TTL(3)	
RD4					I/O		Digital I/O.
PSP4					I/O		Parallel Slave Port data.
RD5/PSP5	28	31	3	3		ST/TTL(3)	
RD5					I/O		Digital I/O.
PSP5					I/O		Parallel Slave Port data.
RD6/PSP6	29	32	4	4		ST/TTL(3)	
RD6					I/O		Digital I/O.
PSP6					I/O		Parallel Slave Port data.
RD7/PSP7	30	33	5	5		ST/TTL(3)	
RD7					I/O		Digital I/O.
PSP7					I/O		Parallel Slave Port data.
							PORTE is a bidirectional I/O port.
RE0/$\overline{RD}$/AN5	8	9	25	25		ST/TTL(3)	
RE0					I/O		Digital I/O.
$\overline{RD}$					I		Read control for Parallel Slave Port.
AN5					I		Analog input 5.
RE1/$\overline{WR}$/AN6	9	10	26	26		ST/TTL(3)	
RE1					I/O		Digital I/O.
$\overline{WR}$					I		Write control for Parallel Slave Port.
AN6					I		Analog input 6.
RE2/$\overline{CS}$/AN7	10	11	27	27		ST/TTL(3)	
RE2					I/O		Digital I/O.
$\overline{CS}$					I		Chip select control for Parallel Slave Port.
AN7					I		Analog input 7.
VSS	12, 31	13, 34	6, 29	6, 30, 31	P	—	Ground reference for logic and I/O pins.
VDD	11, 32	12, 35	7, 28	7, 8, 28, 29	P	—	Positive supply for logic and I/O pins.
NC	—	1, 17, 28, 40	12,13, 33, 34	13	—	—	These pins are not internally connected. These pins should be left unconnected.

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.

3.6 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 14-1). 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 erase and write operations. The write operation is edge-aligned and cannot occur across boundaries.

To write program data, it must first be loaded into the buffer registers (see Figure 3-1). This is accomplished by first writing the destination address to EEADR and EEADRH and then writing the data to EEDATA and EEDATH. After the address and data have been set up, then the following sequence of events must be executed:

1. Set the EEPGD control bit (EECON1<7>).
2. Write 55h, then AAh, to EECON2 (Flash programming sequence).
3. Set the WR control bit (EECON1<1>).

All four buffer register locations **MUST** be written to with correct data. If only one, two or three words are being written to in the block of four words, then a read from the program memory location(s) not being written to must be performed. This takes the data from the program location(s) not being written and loads it into the EEDATA and EEDATH registers. Then the sequence of events to transfer data to the buffer registers must be executed.

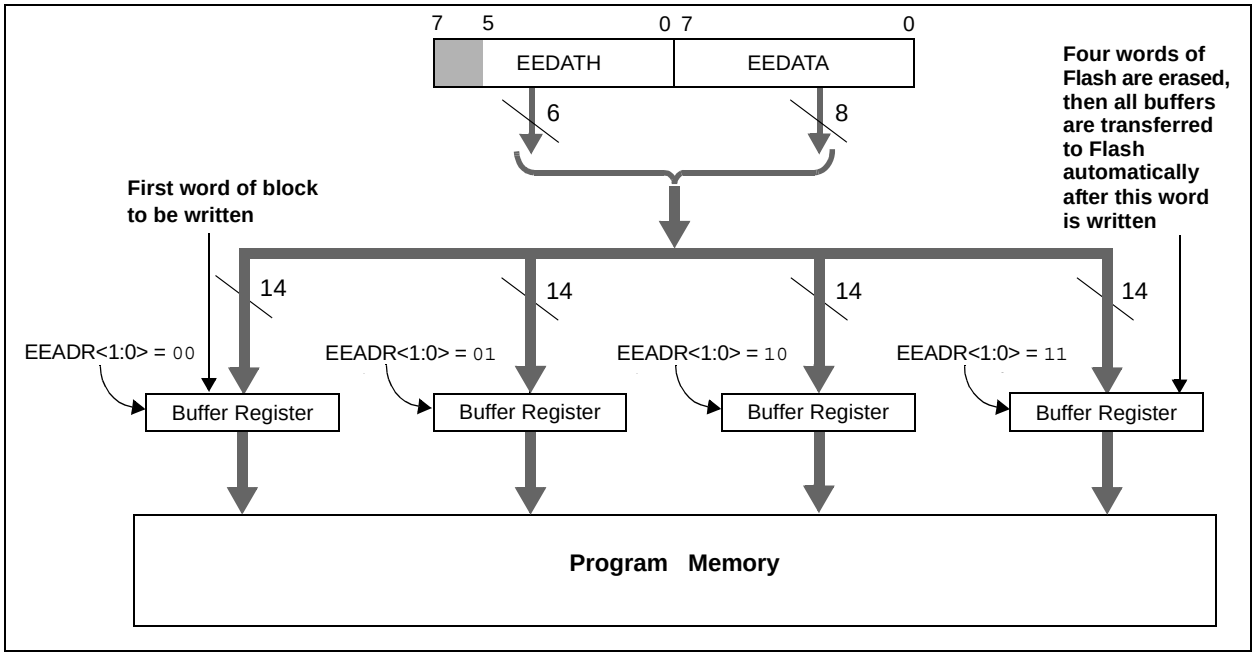
To transfer data from the buffer registers to the program memory, the EEADR and EEADRH must point to the last location in the four-word block (EEADR<1:0> = 11). Then the following sequence of events must be executed:

1. Set the EEPGD control bit (EECON1<7>).
2. Write 55h, then AAh, to EECON2 (Flash programming sequence).
3. Set control bit WR (EECON1<1>) to begin the write operation.

The user must follow the same specific sequence to initiate the write for each word in the program block, writing each program word in sequence (00,01,10,11). When the write is performed on the last word (EEADR<1:0> = 11), the block of four words are automatically erased and the contents of the buffer registers are written into the program memory.

After the "BSF EECON1,WR" instruction, the processor requires two cycles to set up the erase/write operation. The user must place two NOP instructions after the WR bit is set. Since data is being written to buffer registers, the writing of the first three words of the block appears to occur immediately. The processor will halt internal operations for the typical 4 ms, only during the cycle in which the erase takes place (i.e., the last word of the four-word block). This is not Sleep mode as the clocks and peripherals will continue to run. After the write cycle, the processor will resume operation with the third instruction after the EECON1 write instruction. If the sequence is performed to any other location, the action is ignored.

FIGURE 3-1: BLOCK WRITES TO FLASH PROGRAM MEMORY



An example of the complete four-word write sequence is shown in Example 3-4. The initial address is loaded into the EEADRH:EEADR register pair; the four words of data are loaded using indirect addressing.

EXAMPLE 3-4: WRITING TO FLASH PROGRAM MEMORY

```
; This write routine assumes the following:
;
; 1. A valid starting address (the least significant bits = '00')is loaded in ADDRH:ADDRL
; 2. The 8 bytes of data are loaded, starting at the address in DATADDR
; 3. ADDRH, ADDRL and DATADDR are all located in shared data memory 0x70 - 0x7f
;
        BSF     STATUS,RP1          ;
        BCF     STATUS,RP0          ; Bank 2
        MOVF    ADDRH,W             ; Load initial address
        MOVWF   EEADRH              ;
        MOVF    ADDRL,W             ;
        MOVWF   EEADR               ;
        MOVF    DATAADDR,W          ; Load initial data address
        MOVWF   FSR                 ;
LOOP    MOVF    INDF,W              ; Load first data byte into lower
        MOVWF   EEDATA              ;
        INCF    FSR,F               ; Next byte
        MOVF    INDF,W              ; Load second data byte into upper
        MOVWF   EEDATH              ;
        INCF    FSR,F               ;
        BSF     STATUS,RP0          ; Bank 3
        BSF     EECON1,EEPGD        ; Point to program memory
        BSF     EECON1,WREN         ; Enable writes
        BCF     INTCON,GIE          ; Disable interrupts (if using)
        MOVLW   55h                 ; Start of required write sequence:
        MOVWF   EECON2              ; Write 55h
        MOVLW   AAh                 ;
        MOVWF   EECON2              ; Write AAh
        BSF     EECON1,WR           ; Set WR bit to begin write
        NOP                         ; Any instructions here are ignored as processor
                                    ; halts to begin write sequence
        NOP                         ; processor will stop here and wait for write complete
                                    ; after write processor continues with 3rd instruction
        BCF     EECON1,WREN         ; Disable writes
        BSF     INTCON,GIE          ; Enable interrupts (if using)
        BCF     STATUS,RP0          ; Bank 2
        INCF    EEADR,F             ; Increment address
        MOVF    EEADR,W             ; Check if lower two bits of address are '00'
        ANDLW   0x03                ; Indicates when four words have been programmed
        XORLW   0x03                ;
        BTFSC   STATUS,Z            ; Exit if more than four words,
        GOTO    LOOP                ; Continue if less than four words
```

Required Sequence

8.0 CAPTURE/COMPARE/PWM MODULES

Each Capture/Compare/PWM (CCP) module contains a 16-bit register which can operate as a:

- 16-bit Capture register
- 16-bit Compare register
- PWM Master/Slave Duty Cycle register

Both the CCP1 and CCP2 modules are identical in operation, with the exception being the operation of the special event trigger. Table 8-1 and Table 8-2 show the resources and interactions of the CCP module(s). In the following sections, the operation of a CCP module is described with respect to CCP1. CCP2 operates the same as CCP1 except where noted.

CCP1 Module:

Capture/Compare/PWM Register 1 (CCPR1) is comprised of two 8-bit registers: CCPR1L (low byte) and CCPR1H (high byte). The CCP1CON register controls the operation of CCP1. The special event trigger is generated by a compare match and will reset Timer1.

CCP2 Module:

Capture/Compare/PWM Register 2 (CCPR2) is comprised of two 8-bit registers: CCPR2L (low byte) and CCPR2H (high byte). The CCP2CON register controls the operation of CCP2. The special event trigger is generated by a compare match and will reset Timer1 and start an A/D conversion (if the A/D module is enabled).

Additional information on CCP modules is available in the PIC® Mid-Range MCU Family Reference Manual (DS33023) and in application note *AN594, "Using the CCP Module(s)"* (DS00594).

TABLE 8-1: CCP MODE – TIMER RESOURCES REQUIRED

CCP Mode	Timer Resource
Capture	Timer1
Compare	Timer1
PWM	Timer2

TABLE 8-2: INTERACTION OF TWO CCP MODULES

CCPx Mode	CCPy Mode	Interaction
Capture	Capture	Same TMR1 time base
Capture	Compare	The compare should be configured for the special event trigger which clears TMR1
Compare	Compare	The compare(s) should be configured for the special event trigger which clears TMR1
PWM	PWM	The PWMs will have the same frequency and update rate (TMR2 interrupt)
PWM	Capture	None
PWM	Compare	None

9.0 MASTER SYNCHRONOUS SERIAL PORT (MSSP) MODULE

9.1 Master SSP (MSSP) Module Overview

The Master Synchronous Serial Port (MSSP) 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 MSSP module can operate in one of two modes:

- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I^2C)
  - Full Master mode
  - Slave mode (with general address call)

The I^2C interface supports the following modes in hardware:

- Master mode
- Multi-Master mode
- Slave mode

9.2 Control Registers

The MSSP module has three associated registers. These include a status register (SSPSTAT) and two control registers (SSPCON and SSPCON2). The use of these registers and their individual configuration bits differ significantly, depending on whether the MSSP module is operated in SPI or I^2C mode.

Additional details are provided under the individual sections.

9.3 SPI Mode

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

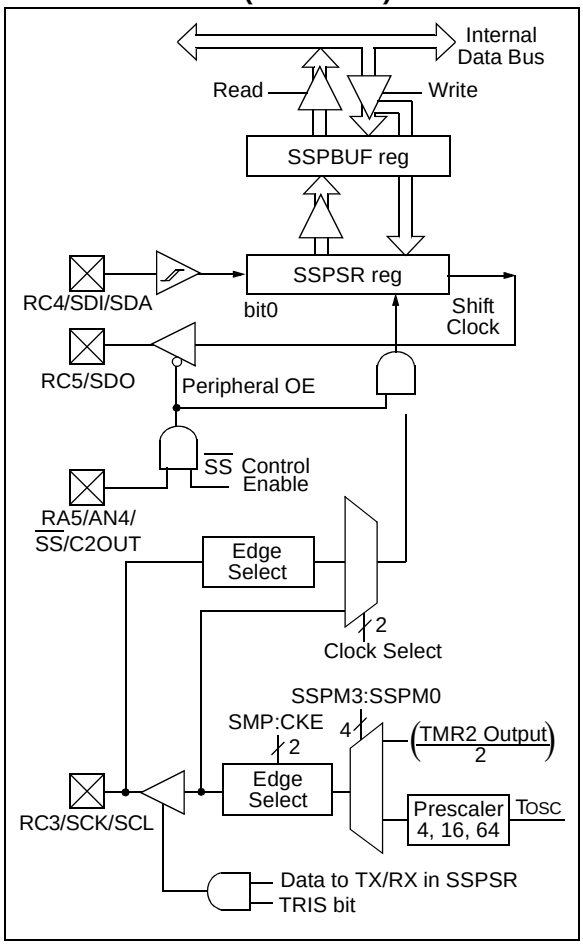
- Serial Data Out (SDO) – RC5/SDO
- Serial Data In (SDI) – RC4/SDI/SDA
- Serial Clock (SCK) – RC3/SCK/SCL

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

- Slave Select ($\overline{SS}$) – RA5/AN4/$\overline{SS}$/C2OUT

Figure 9-1 shows the block diagram of the MSSP module when operating in SPI mode.

FIGURE 9-1: MSSP BLOCK DIAGRAM (SPI MODE)



Note:	When the SPI is in Slave mode with $\overline{SS}$ pin control enabled (SSPCON<3:0> = 0100), the state of the $\overline{SS}$ pin can affect the state read back from the TRISC<5> bit. The Peripheral OE signal from the SSP module in PORTC controls the state that is read back from the TRISC<5> bit (see **Section 4.3 "PORTC and the TRISC Register"** for information on PORTC). If Read-Modify-Write instructions, such as BSF, are performed on the TRISC register while the $\overline{SS}$ pin is high, this will cause the TRISC<5> bit to be set, thus disabling the SDO output.

9.3.2 OPERATION

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits (SSPCON<5:0> and 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)
- Data Input Sample Phase (middle or end of data output time)
- Clock Edge (output data on rising/falling edge of SCK)
- Clock Rate (Master mode only)
- Slave Select mode (Slave mode only)

The MSSP consists of a transmit/receive shift register (SSPSR) and a buffer register (SSPBUF). The SSPSR shifts the data in and out of the device, MSb first. The SSPBUF holds the data that was written to the SSPSR until the received data is ready. Once the eight bits of data have been received, that byte is moved to the SSPBUF register. Then, the Buffer Full detect bit, BF (SSPSTAT<0>), and the interrupt flag bit, SSPIF, are set. This double-buffering of the received data (SSPBUF) allows the next byte to start reception before reading the data that was just received. Any write to the SSPBUF register during transmission/reception of data will be ignored and the write collision detect bit, WCOL (SSPCON<7>), will be set. User software must clear the WCOL bit so that it can be determined if the following write(s) to the SSPBUF register completed successfully.

When the application software is expecting to receive valid data, the SSPBUF should be read before the next byte of data to transfer is written to the SSPBUF. Buffer Full bit, BF (SSPSTAT<0>), indicates when SSPBUF has been loaded with the received data (transmission is complete). When the SSPBUF is read, the BF bit is cleared. This data may be irrelevant if the SPI is only a transmitter. Generally, the MSSP interrupt is used to determine when the transmission/reception has completed. The SSPBUF must be read and/or written. If the interrupt method is not going to be used, then software polling can be done to ensure that a write collision does not occur. Example 9-1 shows the loading of the SSPBUF (SSPSR) for data transmission.

The SSPSR is not directly readable or writable and can only be accessed by addressing the SSPBUF register. Additionally, the MSSP Status register (SSPSTAT) indicates the various status conditions.

EXAMPLE 9-1: LOADING THE SSPBUF (SSPSR) REGISTER

```
LOOP  BTFSS  SSPSTAT, BF    ;Has data been received(transmit complete)?
      BRA    LOOP           ;No
      MOVF   SSPBUF, W      ;WREG reg = contents of SSPBUF
      MOVWF  RXDATA         ;Save in user RAM, if data is meaningful
      MOVF   TXDATA, W      ;W reg = contents of TXDATA
      MOVWF  SSPBUF         ;New data to xmit
```

FIGURE 9-5: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 0)

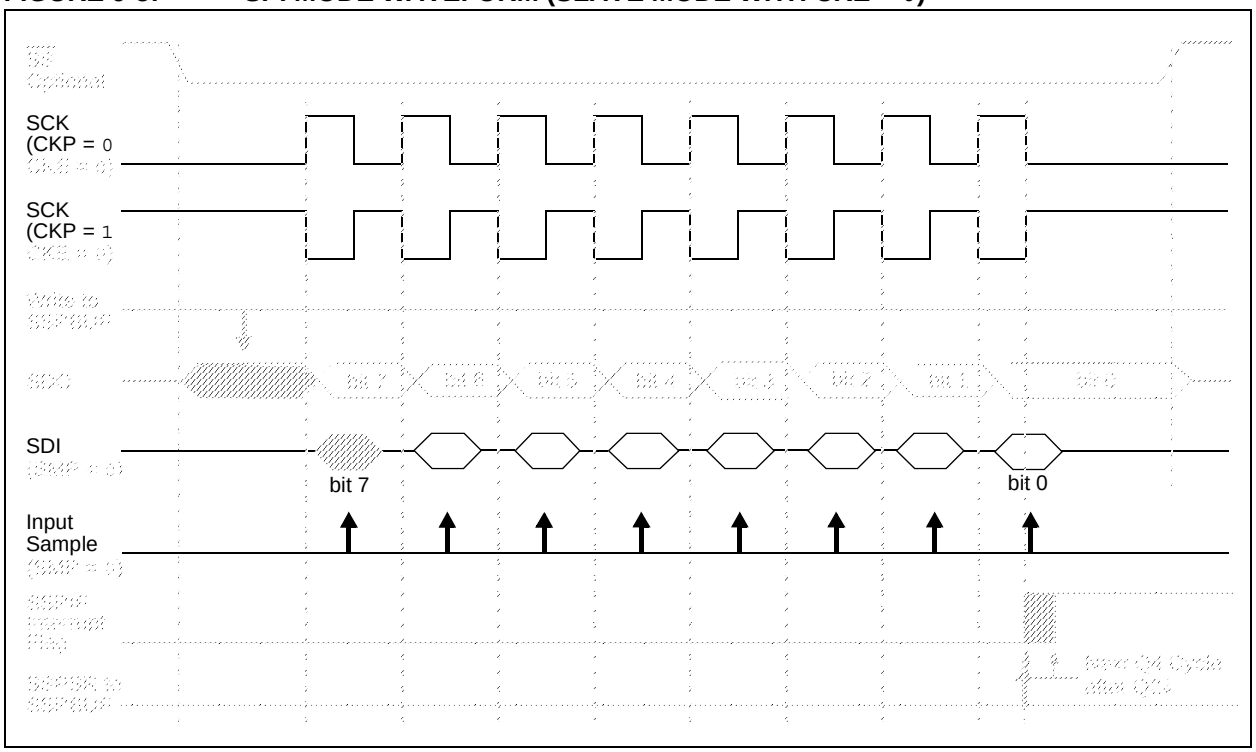
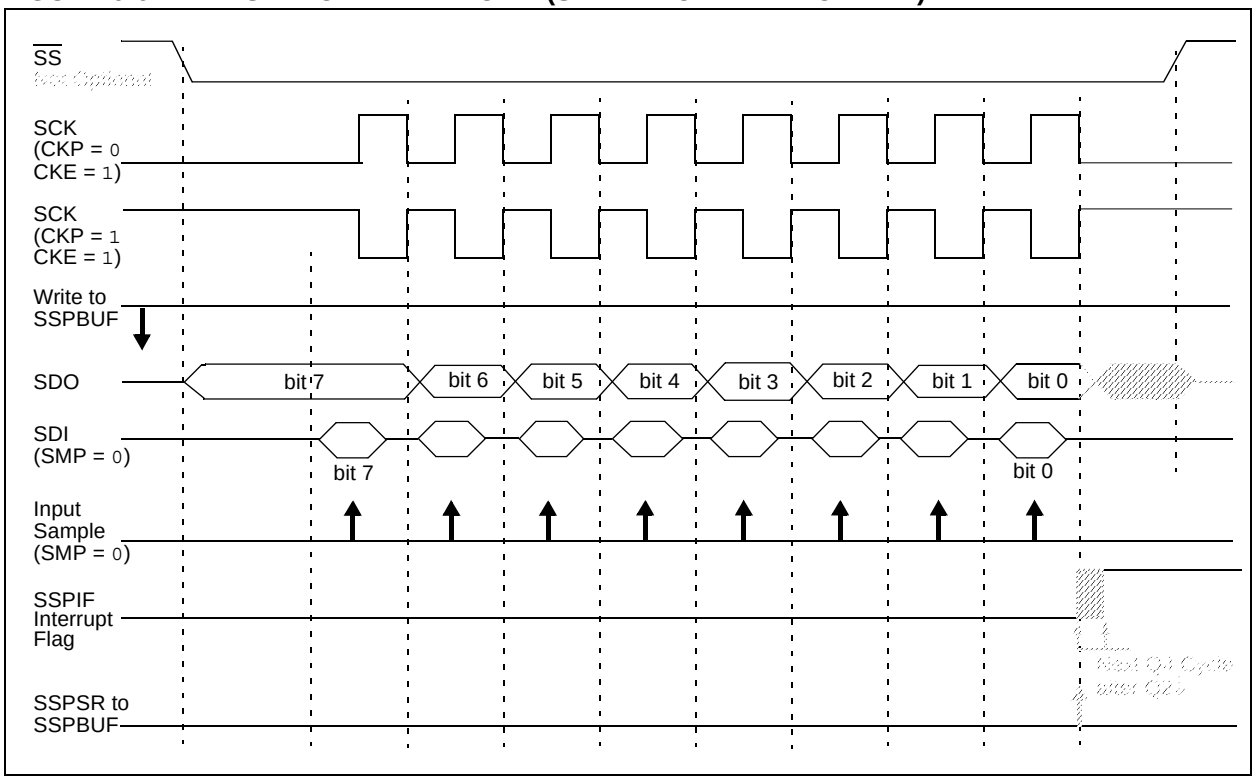


FIGURE 9-6: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 1)



9.4 I²C Mode

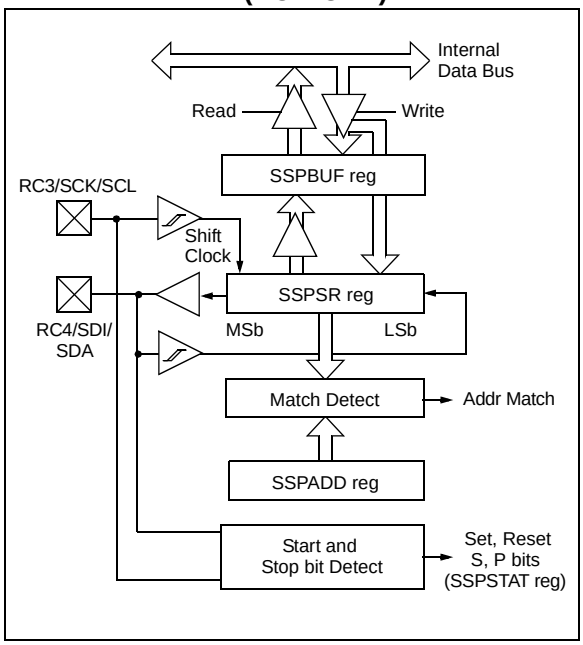
The MSSP module in I²C mode fully implements all master and slave functions (including general call support) and provides interrupts on Start and Stop bits in hardware to determine a free bus (multi-master function). The MSSP module implements the standard mode specifications, as well as 7-bit and 10-bit addressing.

Two pins are used for data transfer:

- Serial clock (SCL) – RC3/SCK/SCL
- Serial data (SDA) – RC4/SDI/SDA

The user must configure these pins as inputs or outputs through the TRISC<4:3> bits.

FIGURE 9-7: MSSP BLOCK DIAGRAM (I²C MODE)



9.4.1 REGISTERS

The MSSP module has six registers for I²C operation. These are:

- MSSP Control Register (SSPCON)
- MSSP Control Register 2 (SSPCON2)
- MSSP Status Register (SSPSTAT)
- Serial Receive/Transmit Buffer Register (SSPBUF)
- MSSP Shift Register (SSPSR) – Not directly accessible
- MSSP Address Register (SSPADD)

SSPCON, SSPCON2 and SSPSTAT are the control and status registers in I²C mode operation. The SSPCON and SSPCON2 registers are readable and writable. The lower six bits of the SSPSTAT are read-only. The upper two bits of the SSPSTAT are read/write.

SSPSR is the shift register used for shifting data in or out. SSPBUF is the buffer register to which data bytes are written to or read from.

SSPADD register holds the slave device address when the SSP is configured in I²C Slave mode. When the SSP is configured in Master mode, the lower seven bits of SSPADD act as the baud rate generator reload value.

In receive operations, SSPSR and SSPBUF together create a double-buffered receiver. When SSPSR receives a complete byte, it is transferred to SSPBUF and the SSPIF interrupt is set.

During transmission, the SSPBUF is not double-buffered. A write to SSPBUF will write to both SSPBUF and SSPSR.

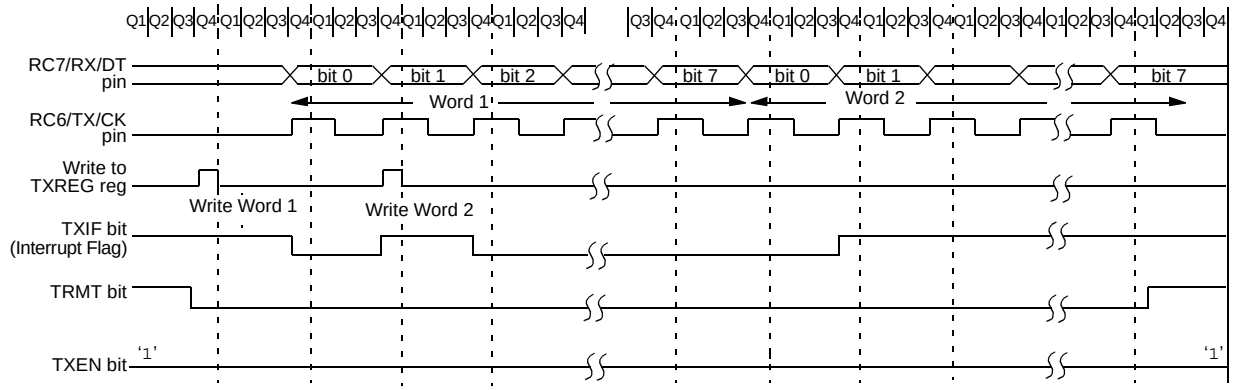
TABLE 10-8: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION

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	R0IF	0000 000x	0000 000u
0Ch	PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
19h	TXREG	USART Transmit Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented, read as '0'. Shaded cells are not used for synchronous master transmission.

Note 1: Bits PSPIE and PSPIF are reserved on 28-pin devices; always maintain these bits clear.

FIGURE 10-9: SYNCHRONOUS TRANSMISSION



Note: Sync Master mode; SPBRG = 0. Continuous transmission of two 8-bit words.

FIGURE 10-10: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)

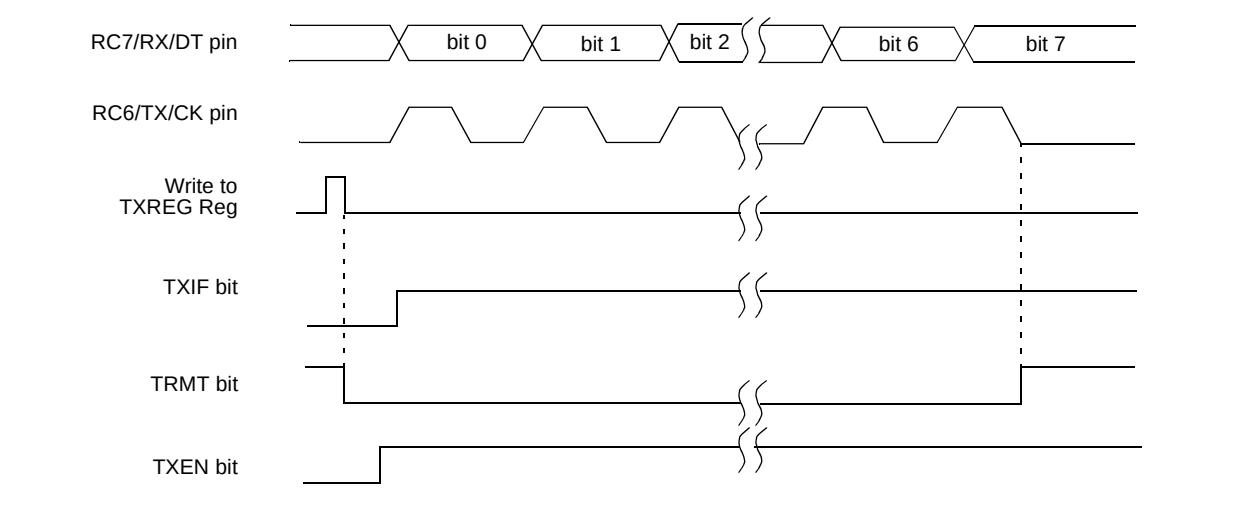


TABLE 14-6: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Devices				Power-on Reset, Brown-out Reset	$\overline{MCLR}$ Resets, WDT Reset	Wake-up via WDT or Interrupt
TRISD	73A	74A	76A	77A	1111 1111	1111 1111	uuuu uuuu
TRISE	73A	74A	76A	77A	0000 -111	0000 -111	uuuu -uuu
PIE1	73A	74A	76A	77A	r000 0000	r000 0000	ruuu uuuu
	73A	74A	76A	77A	0000 0000	0000 0000	uuuu uuuu
PIE2	73A	74A	76A	77A	-0-0 0--0	-0-0 0--0	-u-u u--u
PCON	73A	74A	76A	77A	---- --qq	---- --uu	---- --uu
SSPCON2	73A	74A	76A	77A	0000 0000	0000 0000	uuuu uuuu
PR2	73A	74A	76A	77A	1111 1111	1111 1111	1111 1111
SSPADD	73A	74A	76A	77A	0000 0000	0000 0000	uuuu uuuu
SSPSTAT	73A	74A	76A	77A	--00 0000	--00 0000	--uu uuuu
TXSTA	73A	74A	76A	77A	0000 -010	0000 -010	uuuu -uuu
SPBRG	73A	74A	76A	77A	0000 0000	0000 0000	uuuu uuuu
CMCON	73A	974	76A	77A	0000 0111	0000 0111	uuuu uuuu
CVRCON	73A	74A	76A	77A	000- 0000	000- 0000	uuu- uuuu
ADRESL	73A	74A	76A	77A	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON1	73A	74A	76A	77A	00-- 0000	00-- 0000	uu-- uuuu
EEDATA	73A	74A	76A	77A	xxxx xxxx	uuuu uuuu	uuuu uuuu
EEADR	73A	74A	76A	77A	xxxx xxxx	uuuu uuuu	uuuu uuuu
EEDATH	73A	74A	76A	77A	xxxx xxxx	uuuu uuuu	uuuu uuuu
EEADRH	73A	74A	76A	77A	xxxx xxxx	uuuu uuuu	uuuu uuuu
EECON1	73A	74A	76A	77A	x--- x000	u--- u000	u--- uuuu
EECON2	73A	74A	76A	77A	---- ----	---- ----	---- ----

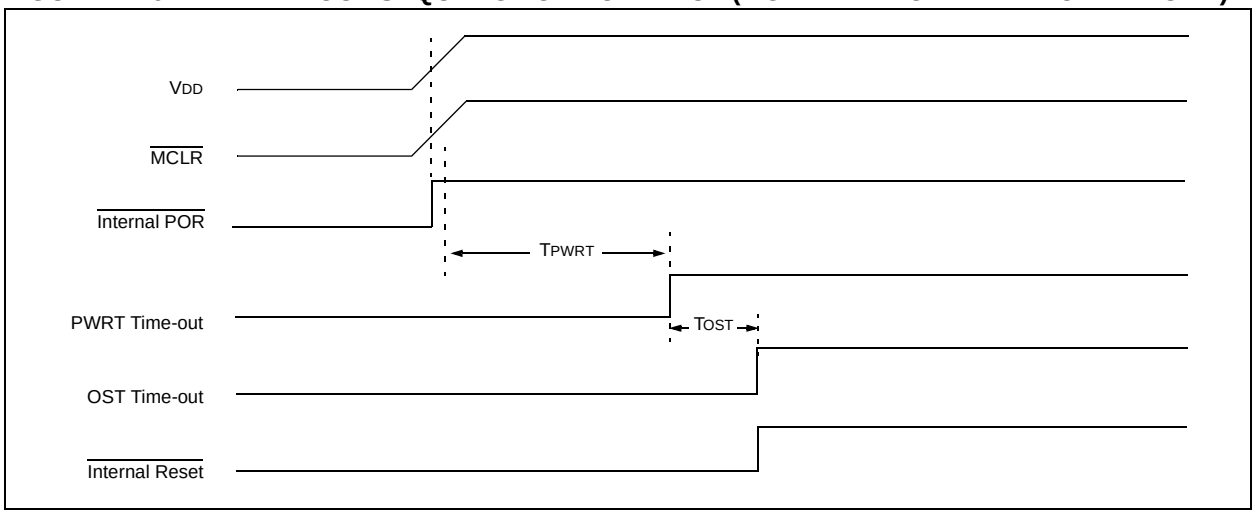
Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition, r = reserved, maintain clear. Shaded cells indicate conditions do not apply for the designated device.

Note 1: One or more bits in INTCON, PIR1 and/or PIR2 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 14-5 for Reset value for specific condition.

FIGURE 14-6: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ TIED TO V_{DD} VIA RC NETWORK)



DECFSZ **Decrement f, Skip if 0**

Syntax:	[*label*] DECFSZ f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f) - 1 → (destination); skip if result = 0
Status Affected:	None
Description:	The contents of register 'f' are decremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'. If the result is '1', the next instruction is executed. If the result is '0', then a NOP is executed instead, making it a 2 TCY instruction.

GOTO **Unconditional Branch**

Syntax:	[*label*] GOTO k
Operands:	$0 \le k \le 2047$
Operation:	k → PC<10:0> PCLATH<4:3> → PC<12:11>
Status Affected:	None
Description:	GOTO is an unconditional branch. The eleven-bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. GOTO is a two-cycle instruction.

INCF **Increment f**

Syntax:	[*label*] INCF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f) + 1 → (destination)
Status Affected:	Z
Description:	The contents of register 'f' are incremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.

INCFSZ **Increment f, Skip if 0**

Syntax:	[*label*] INCFSZ f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f) + 1 → (destination), skip if result = 0
Status Affected:	None
Description:	The contents of register 'f' are incremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'. If the result is '1', the next instruction is executed. If the result is '0', a NOP is executed instead, making it a 2 TCY instruction.

IORLW **Inclusive OR Literal with W**

Syntax:	[*label*] IORLW k
Operands:	$0 \le k \le 255$
Operation:	(W) .OR. k → (W)
Status Affected:	Z
Description:	The contents of the W register are OR'ed with the eight-bit literal 'k'. The result is placed in the W register.

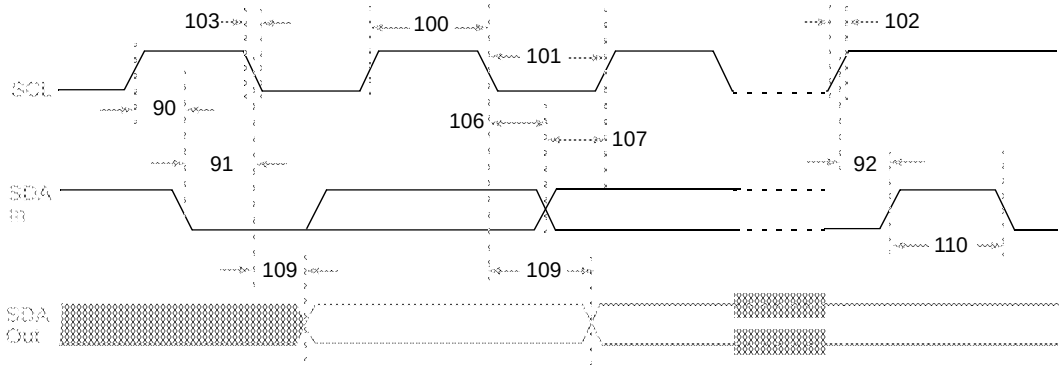
IORWF **Inclusive OR W with f**

Syntax:	[*label*] IORWF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(W) .OR. (f) → (destination)
Status Affected:	Z
Description:	Inclusive OR the W register with register 'f'. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.

TABLE 17-10: I²C BUS START/STOP BITS REQUIREMENTS

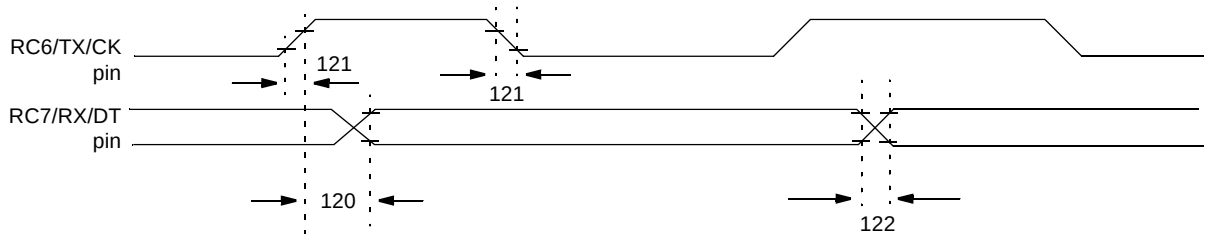
Param No.	Symbol	Characteristic		Min	Typ	Max	Units	Conditions
90	TSU:STA	Start condition	100 kHz mode	4700	—	—	ns	Only relevant for Repeated Start condition
		Setup time	400 kHz mode	600	—	—		
91	THD:STA	Start condition	100 kHz mode	4000	—	—	ns	After this period, the first clock pulse is generated
	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

FIGURE 17-16: I²C BUS DATA TIMING



Note: Refer to Figure 17-3 for load conditions.

FIGURE 17-17: USART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING



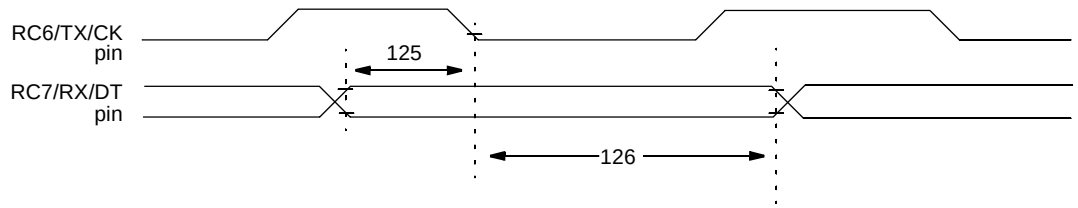
Note: Refer to Figure 17-3 for load conditions.

TABLE 17-12: USART SYNCHRONOUS TRANSMISSION REQUIREMENTS

Param No.	Symbol	Characteristic		Min	Typ†	Max	Units	Conditions
120	TCKH2DTV	SYNC XMIT (MASTER & SLAVE) Clock High to Data Out Valid	Standard(**F**)	—	—	80	ns	
			Extended(**LF**)	—	—	100	ns	
121	TCKRF	Clock Out Rise Time and Fall Time (Master mode)	Standard(**F**)	—	—	45	ns	
			Extended(**LF**)	—	—	50	ns	
122	TDTRF	Data Out Rise Time and Fall Time	Standard(**F**)	—	—	45	ns	
			Extended(**LF**)	—	—	50	ns	

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

FIGURE 17-18: USART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING



Note: Refer to Figure 17-3 for load conditions.

TABLE 17-13: USART SYNCHRONOUS RECEIVE REQUIREMENTS

Param No.	Symbol	Characteristic	Min	Typ†	Max	Units	Conditions
125	TDTV2CKL	SYNC RCV (MASTER & SLAVE) Data Setup before CK ↓ (DT setup time)	15	—	—	ns	
126	TCKL2DTL	Data Hold after CK ↓ (DT hold time)	15	—	—	ns	

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

TABLE 17-14: A/D CONVERTER CHARACTERISTICS:PIC16F873A/874A/876A/877A (INDUSTRIAL)
PIC16LF873A/874A/876A/877A (INDUSTRIAL)

Param No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
A01	NR	Resolution		—	—	10-bits	bit	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF
A03	EIL	Integral Linearity Error		—	—	< ± 1	LSb	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF
A04	EDL	Differential Linearity Error		—	—	< ± 1	LSb	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF
A06	EOFF	Offset Error		—	—	< ± 2	LSb	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF
A07	EGN	Gain Error		—	—	< ± 1	LSb	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF
A10	—	Monotonicity		—	guaranteed(3)	—	—	VSS ≤ VAIN ≤ VREF
A20	VREF	Reference Voltage (VREF+ – VREF-)		2.0	—	VDD + 0.3	V	
A21	VREF+	Reference Voltage High		AVDD – 2.5V		AVDD + 0.3V	V	
A22	VREF-	Reference Voltage Low		AVSS – 0.3V		VREF+ – 2.0V	V	
A25	VAIN	Analog Input Voltage		VSS – 0.3V	—	VREF + 0.3V	V	
A30	ZAIN	Recommended Impedance of Analog Voltage Source		—	—	2.5	kΩ	**(Note 4)**
A40	IAD	A/D Conversion Current (VDD)	PIC16F87XA	—	220	—	μA	Average current consumption when A/D is on **(Note 1)**
			PIC16LF87XA	—	90	—	μA	
A50	IREF	VREF Input Current **(Note 2)**		—	—	5	μA	During VAIN acquisition. Based on differential of VHOLD to VAIN to charge CHOLD, see **Section 11.1 "A/D Acquisition Requirements"**.
				—	—	150	μA	During A/D conversion cycle

* 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.

Note 1: When A/D is off, it will not consume any current other than minor leakage current. The power-down current spec includes any such leakage from the A/D module.

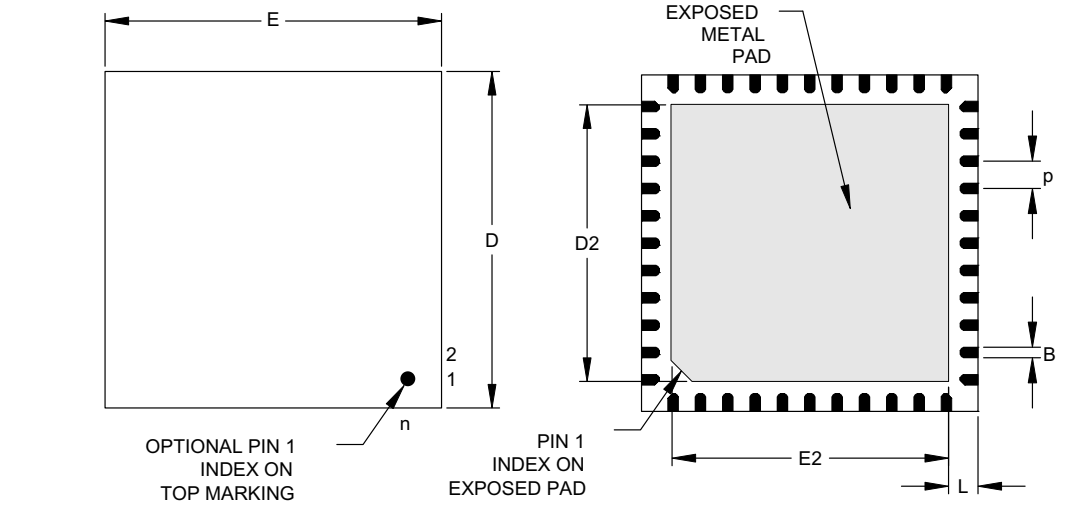
2: VREF current is from RA3 pin or VDD pin, whichever is selected as reference input.

3: The A/D conversion result never decreases with an increase in the input voltage and has no missing codes.

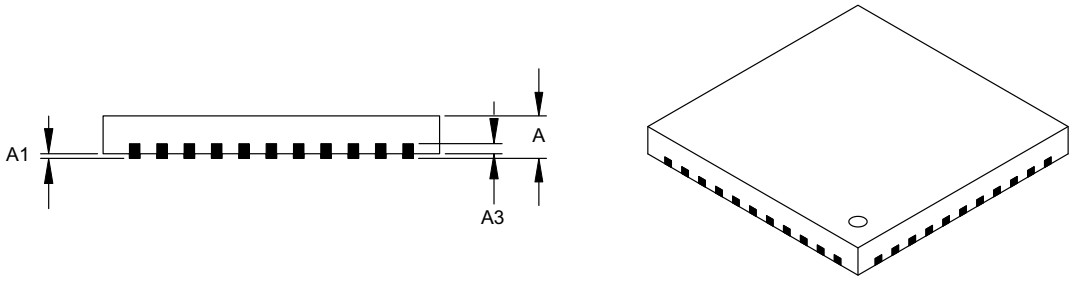
4: Maximum allowed impedance for analog voltage source is 10 kΩ. This requires higher acquisition time.

44-Lead Plastic Quad Flat No Lead Package (ML) 8x8 mm Body (QFN)

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



TOP VIEW BOTTOM VIEW



Units		INCHES			MILLIMETERS*		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		44			44	
Pitch	p		.026 BSC			0.65 BSC	
Overall Height	A	.031	.035	.039	0.80	0.90	1.00
Standoff	A1	.000	.001	.002	0	0.02	0.05
Base Thickness	A3		.010 REF			0.25 REF	
Overall Width	E		.315 BSC			8.00 BSC	
Exposed Pad Width	E2	.262	.268	.274	6.65	6.80	6.95
Overall Length	D		.315 BSC			8.00 BSC	
Exposed Pad Length	D2	.262	.268	.274	6.65	6.80	6.95
Lead Width	B	.012	.013	.013	0.30	0.33	0.35
Lead Length	L	.014	.016	.018	0.35	0.40	0.45

*Controlling Parameter

Notes:

Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.

JEDEC equivalent: M0-220

Drawing No. C04-103

INDEX

R

S

T

THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://microchip.com/support