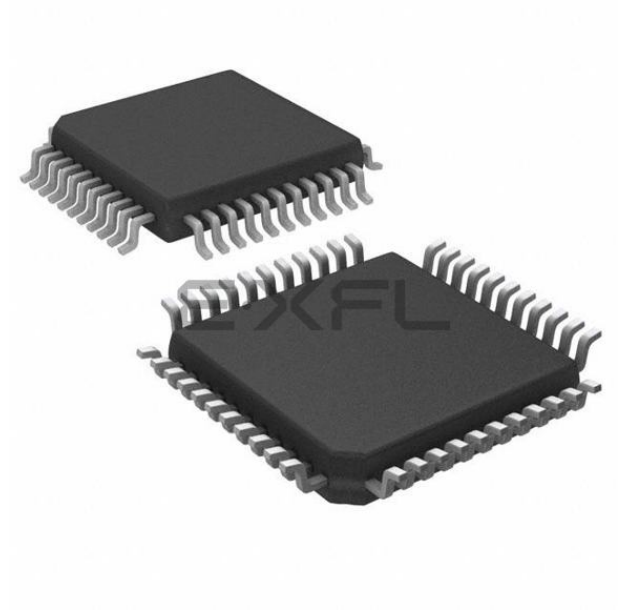


Welcome to E-XFL.COM

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	4MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	33
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 8x10b
Oscillator Type	External
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	44-QFP
Supplier Device Package	44-MQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f877-04-pq

Table of Contents

1.5.3.4 UV Erasable Devices

The UV erasable version of EPROM program memory devices is optimal for prototype development and pilot programs.

These devices can be erased and reprogrammed to any of the configuration modes. Third party programmers are also available; refer to Microchip's *Third Party Guide* (DS00104) for a list of sources.

The amount of time required to completely erase a UV erasable device depends on: the wavelength of the light, its intensity, distance from UV source, the process technology of the device (how small are the memory cells).

Note:	Fluorescent lights and sunlight both emit ultraviolet light at the erasure wavelength. Leaving a UV erasable device's window uncovered could cause, over time, the devices memory cells to become erased. The erasure time for a fluorescent light is about three years, while sunlight requires only about one week. To prevent the memory cells from losing data, an opaque label should be placed over the erasure window.

1.5.3.5 One-Time-Programmable (OTP) Devices

The availability of OTP devices is especially useful for customers expecting code changes and updates.

OTP devices, packaged in plastic packages, permit the user to program them once. In addition to the program and data EPROM memories, the configuration bits must be programmed.

1.5.3.6 Flash Devices

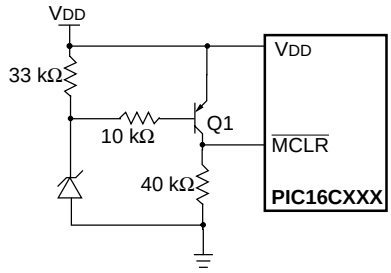
A Flash device allows its memory to be changed by an electric charge. This means that the system can be designed so that programming may be performed in-circuit. Since no window is required, the lower cost plastic packages can used for these devices.

1.5.3.7 EEPROM Devices

An EEPROM device allows its memory to be erased by an electric charge. This means that the system can be designed so that erasure and reprogramming may be performed in-circuit. Since no window is required, the lower cost plastic packages can used for these devices.

Some devices do not have the on-chip brown-out circuit, and in other cases there are some applications where the Brown-out Reset trip point of the device may not be at the desired level. Figure 3-10 and Figure 3-11 are two examples of external circuitry that may be implemented. Each needs to be evaluated to determine if they match the requirements of the application.

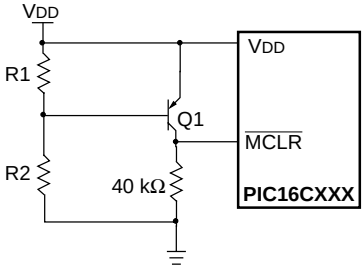
Figure 3-10: External Brown-out Protection Circuit 1



This circuit will activate reset when VDD goes below (Vz + 0.7V)
where Vz = Zener voltage.

Note 1: Internal Brown-out Reset circuitry should be disabled when using this circuit.
2: Resistors should be adjusted for the characteristics of the transistor.

Figure 3-11: External Brown-out Protection Circuit 2



Note 1: This brown-out circuit is less expensive, albeit less accurate. Transistor Q1 turns off when VDD is below a certain level such that:

$$V_{DD} \cdot \frac{R1}{R1 + R2} = 0.7V$$

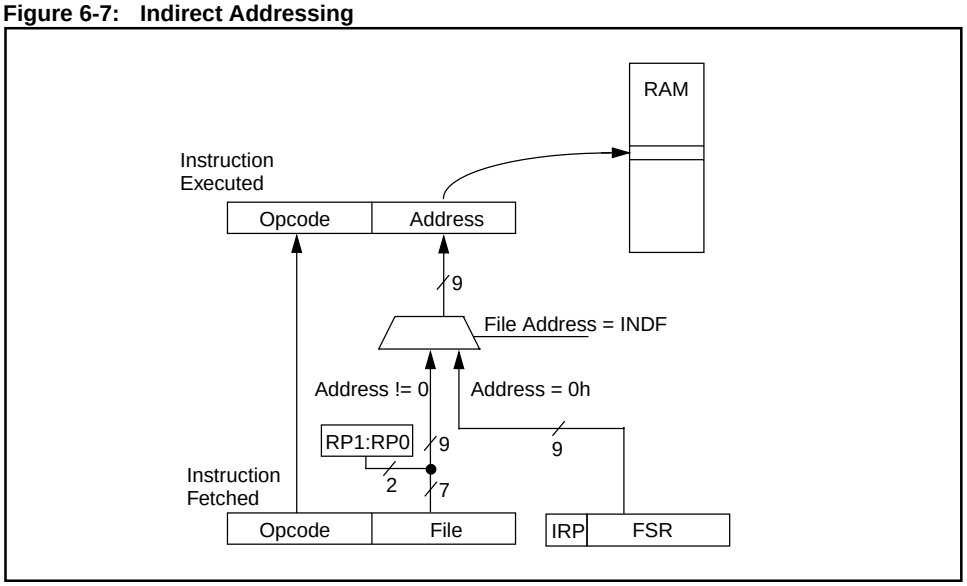
2: Internal Brown-out Reset circuitry should be disabled when using this circuit.
3: Resistors should be adjusted for the characteristics of the transistor.

6.3.4 Indirect Addressing, INDF, and FSR Registers

Indirect addressing is a mode of addressing data memory where the data memory address in the instruction is not fixed. An SFR register is used as a pointer to the data memory location that is to be read or written. Since this pointer is in RAM, the contents can be modified by the program. This can be useful for data tables in the data memory. Figure 6-7 shows the operation of indirect addressing. This shows the moving of the value to the data memory address specified by the value of the FSR register.

Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses the register pointed to by the File Select Register, FSR. Reading the INDF register itself indirectly (FSR = '0') will read 00h. Writing to the INDF register indirectly results in a no-operation (although status bits may be affected). An effective 9-bit address is generated by the concatenation of the IRP bit (STATUS<7>) with the 8-bit FSR register, as shown in Figure 6-8.

Figure 6-7: Indirect Addressing



8.2.2 PIE Register(s)

Depending on the number of peripheral interrupt sources, there may be multiple Peripheral Interrupt Enable registers (PIE1, PIE2). These registers contain the individual enable bits for the Peripheral interrupts. These registers will be generically referred to as PIE. If the device has a PIE register, The PEIE bit must be set to enable any of these peripheral interrupts.

Note:	Bit PEIE (INTCON<6>) must be set to enable any of the peripheral interrupts.

Although, the PIE register bits have a general bit location with each register, future devices may not have consistent placement. Bit location inconsistencies will not be a problem if you use the supplied Microchip Include files for the symbolic use of these bits. This will allow the Assembler/Compiler to automatically take care of the placement of these bits by specifying the correct register and bit name.

Register 8-2: PIE Register

R/W-0
(Note 1)
bit 7 ... bit 0

bit **TMR1IE**: TMR1 Overflow Interrupt Enable bit
1 = Enables the TMR1 overflow interrupt
0 = Disables the TMR1 overflow interrupt

bit **TMR2IE**: TMR2 to PR2 Match Interrupt Enable bit
1 = Enables the TMR2 to PR2 match interrupt
0 = Disables the TMR2 to PR2 match interrupt

bit **CCP1IE**: CCP1 Interrupt Enable bit
1 = Enables the CCP1 interrupt
0 = Disables the CCP1 interrupt

bit **CCP2IE**: CCP2 Interrupt Enable bit
1 = Enables the CCP2 interrupt
0 = Disables the CCP2 interrupt

bit **SSPIE**: Synchronous Serial Port Interrupt Enable bit
1 = Enables the SSP interrupt
0 = Disables the SSP interrupt

bit **RCIE**: USART Receive Interrupt Enable bit
1 = Enables the USART receive interrupt
0 = Disables the USART receive interrupt

bit **TXIE**: USART Transmit Interrupt Enable bit
1 = Enables the USART transmit interrupt
0 = Disables the USART transmit interrupt

bit **ADIE**: A/D Converter Interrupt Enable bit
1 = Enables the A/D interrupt
0 = Disables the A/D interrupt

bit **ADCIE**: Slope A/D Converter comparator Trip Interrupt Enable bit
1 = Enables the Slope A/D interrupt
0 = Disables the Slope A/D interrupt

bit **OVFIE**: Slope A/D TMR Overflow Interrupt Enable bit
1 = Enables the Slope A/D TMR overflow interrupt
0 = Disables the Slope A/D TMR overflow interrupt

bit **PSPIE:** Parallel Slave Port Read/Write Interrupt Enable bit
1 = Enables the PSP read/write interrupt
0 = Disables the PSP read/write interrupt

bit **EEIE**: EE Write Complete Interrupt Enable bit
1 = Enables the EE write complete interrupt
0 = Disables the EE write complete interrupt

bit **LCDIE**: LCD Interrupt Enable bit
1 = Enables the LCD interrupt
0 = Disables the LCD interrupt

bit **CMIE**: Comparator Interrupt Enable bit
1 = Enables the Comparator interrupt
0 = Disables the Comparator interrupt

Legend		
R = Readable bit	W = Writable bit	
U = Unimplemented bit, read as '0'		- n = Value at POR reset

Note 1: The bit position of the enable bits is device dependent. Please refer to the device data sheet for bit placement.

9.14 Revision History

Revision A

This is the initial released revision of the I/O Ports description.

14.1 Introduction

Each CCP (Capture/Compare/PWM) module contains a 16-bit register which can operate as a 16-bit capture register, as a 16-bit compare register or as a 10-bit PWM master/slave Duty Cycle register. The CCP modules are identical in operation, with the exception of the operation of the special event trigger.

Each CCP module has 3 registers. Multiple CCP modules may exist on a single device. Throughout this section we use generic names for the CCP registers. These generic names are shown in Table 14-1.

Table 14-1: Specific to Generic CCP Nomenclature

Generic Name	CCP1	CCP2	Comment
CCPxCON	CCP1CON	CCP2CON	CCP control register
CCPRxH	CCPR1H	CCPR2H	CCP High byte
CCPRxL	CCPR1L	CCPR2L	CCP Low byte
CCPx	CCP1	CCP2	CCP pin

Table 14-2 shows the resources of the CCP modules, in each of its modes. While Table 14-3 shows the interactions between the CCP modules, where CCPx is one CCP module and CCPy is another CCP module.

Table 14-2: CCP Mode - Timer Resource

CCP Mode	Timer Resource
Capture	Timer1
Compare	Timer1
PWM	Timer2

Table 14-3: 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

14.6 Initialization

The CCP module has three modes of operation. Example 14-3 shows the initialization of capture mode, Example 14-4 shows the initialization of compare mode, and Example 14-5 shows the initialization of PWM mode.

Example 14-3: Capture Initialization

```
    CLRF   CCP1CON       ; CCP Module is off
    CLRF   TMR1H         ; Clear Timer1 High byte
    CLRF   TMR1L         ; Clear Timer1 Low byte
    CLRF   INTCON        ; Disable interrupts and clear T0IF
    BSF    STATUS, RP0   ; Bank1
    BSF    TRISC, CCP1   ; Make CCP pin input
    CLRF   PIE1          ; Disable peripheral interrupts
    BCF    STATUS, RP0   ; Bank0
    CLRF   PIR1          ; Clear peripheral interrupts Flags
    MOVLW  0x06          ; Capture mode, every 4th rising edge
    MOVWF  CCP1CON       ;
    BSF    T1CON, TMR1ON ; Timer1 starts to increment
;
; The CCP1 interrupt is disabled,
; do polling on the CCP Interrupt flag bit
;
Capture_Event
    BTFSS  PIR1, CCP1IF
    GOTO   Capture_Event
;
; Capture has occurred
;
    BCF    PIR1, CCP1IF  ; This needs to be done before next compare
```

The SSP module has five registers for I^2C operation. They are:

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

The SSPCON register allows control of the I^2C operation. Four mode selection bits (SSPCON<3:0>) allow one of the following I^2C modes to be selected:

- I^2C Slave mode (7-bit address)
- I^2C Slave mode (10-bit address)
- I^2C Firmware controlled Multi-Master mode, 7-bit address (start and stop bit interrupts enabled)
- I^2C Firmware controlled Multi-Master mode, 10-bit address (start and stop bit interrupts enabled)
- I^2C Firmware controlled Master mode, slave is idle

Before selecting any I^2C mode, the SCL and SDA pins must be programmed to inputs by setting the appropriate TRIS bits. Selecting an I^2C mode, by setting the SSPEN bit, enables the SCL and SDA pins to be used as the clock and data lines in I^2C mode.

The SSPSTAT register gives the status of the data transfer. This information includes detection of a START or STOP bit, specifies if the received byte was data or address, if the next byte is the completion of 10-bit address, and if this will be a read or write data transfer. The SSPSTAT register is read only.

The SSPBUF is the register to which transfer data is written to or read from. The SSPSR register shifts the data in or out of the device. In receive operations, the SSPBUF and SSPSR create a doubled buffered receiver. This allows reception of the next byte to begin before reading the last byte of received data. When the complete byte is received, it is transferred to the SSPBUF register and the SSPIF flag bit is set. If another complete byte is received before the SSPBUF register is read, a receiver overflow has occurred and bit SSPOV (SSPCON<6>) is set.

The SSPADD register holds the slave address. In 10-bit mode, the user needs to write the high byte of the address (`1111 0 A9 A8 0`). Following the high byte address match, the low byte of the address needs to be loaded (A7:A0).

17.3.5 Slave Mode

In slave mode, the data is transmitted and received as the external clock pulses appear on SCK. When the last bit is latched, the SSPIF interrupt flag bit is set.

While in slave mode the external clock is supplied by the external clock source on the SCK pin. This external clock must meet the minimum high and low times as specified in the electrical specifications.

While in sleep mode, the slave can transmit/receive data. When a byte is receive the device will wake-up from sleep.

17.3.6 Slave Select Synchronization

The $\overline{SS}$ pin allows a synchronous slave mode. The SPI must be in slave mode with $\overline{SS}$ pin control enabled (SSPCON1<3:0> = 04h). The pin must not be driven low for the $\overline{SS}$ pin to function as an input. The Data Latch must be high. When the $\overline{SS}$ pin is low, transmission and reception are enabled and the SDO pin is driven. When the $\overline{SS}$ pin goes high, the SDO pin is no longer driven, even if in the middle of a transmitted byte, and becomes a floating output. External pull-up/ pull-down resistors may be desirable, depending on the application.

Note 1: When the SPI is in Slave Mode with $\overline{SS}$ pin control enabled, (SSPCON<3:0> = 0100) the SPI module will reset if the $\overline{SS}$ pin is set to VDD.

Note 2: If the SPI is used in Slave Mode with CKE is set, then the $\overline{SS}$ pin control must be enabled.

When the SPI module resets, the bit counter is forced to 0. This can be done by either by forcing the $\overline{SS}$ pin to a high level or clearing the SSPEN bit.

To emulate two-wire communication, the SDO pin can be connected to the SDI pin. When the SPI needs to operate as a receiver the SDO pin can be configured as an input. This disables transmissions from the SDO. The SDI can always be left as an input (SDI function) since it cannot create a bus conflict.

17.4.11 I^2C Master Mode Transmission

Transmission of a data byte, a 7-bit address, or the either half of a 10-bit address is accomplished by simply writing a value to SSPBUF register. This action will set the buffer full flag bit, BF, and allow the baud rate generator to begin counting and start the next transmission. Each bit of address/data will be shifted out onto the SDA pin after the falling edge of SCL is asserted (see data hold time specification parameters 106). SCL is held low for one baud rate generator roll over count (T_{BRG}). Data should be valid before SCL is released high (see Data setup time specification parameters 107). When the SCL pin is released high, it is held that way for T_{BRG}, the data on the SDA pin must remain stable for that duration and some hold time after the next falling edge of SCL. After the eighth bit is shifted out (the falling edge of the eighth clock), the BF flag is cleared and the master releases SDA allowing the slave device being addressed to respond with an $\overline{ACK}$ bit during the ninth bit time, if an address match occurs or if data was received properly. The status of $\overline{ACK}$ is written into the ACKDT bit on the falling edge of the ninth clock. If the master receives an acknowledge, the acknowledge status bit, ACKSTAT, is cleared. If not, the bit is set. After the ninth clock the SSPIF bit is set, and the master clock (baud rate generator) is suspended until the next data byte is loaded into the SSPBUF leaving SCL low and SDA unchanged (Figure 17-26).

After the write to the SSPBUF, each bit of address will be shifted out on the falling edge of SCL until all seven address bits and the R/$\overline{W}$ bit are completed. On the falling edge of the eighth clock the master will de-assert the SDA pin allowing the slave to respond with an acknowledge. On the falling edge of the ninth clock the master will sample the SDA pin to see if the address was recognized by a slave. The status of the ACK bit is loaded into the ACKSTAT status bit (SSPCON2<6>). Following the falling edge of the ninth clock transmission of the address, the SSPIF is set, the BF flag is cleared, and the baud rate generator is turned off until another write to the SSPBUF takes place, holding SCL low and allowing SDA to float.

17.4.11.1 BF Status Flag

In transmit mode, the BF bit (SSPSTAT<0>) is set when the CPU writes to SSPBUF and is cleared when all 8 bits are shifted out.

17.4.11.2 WCOL Status Flag

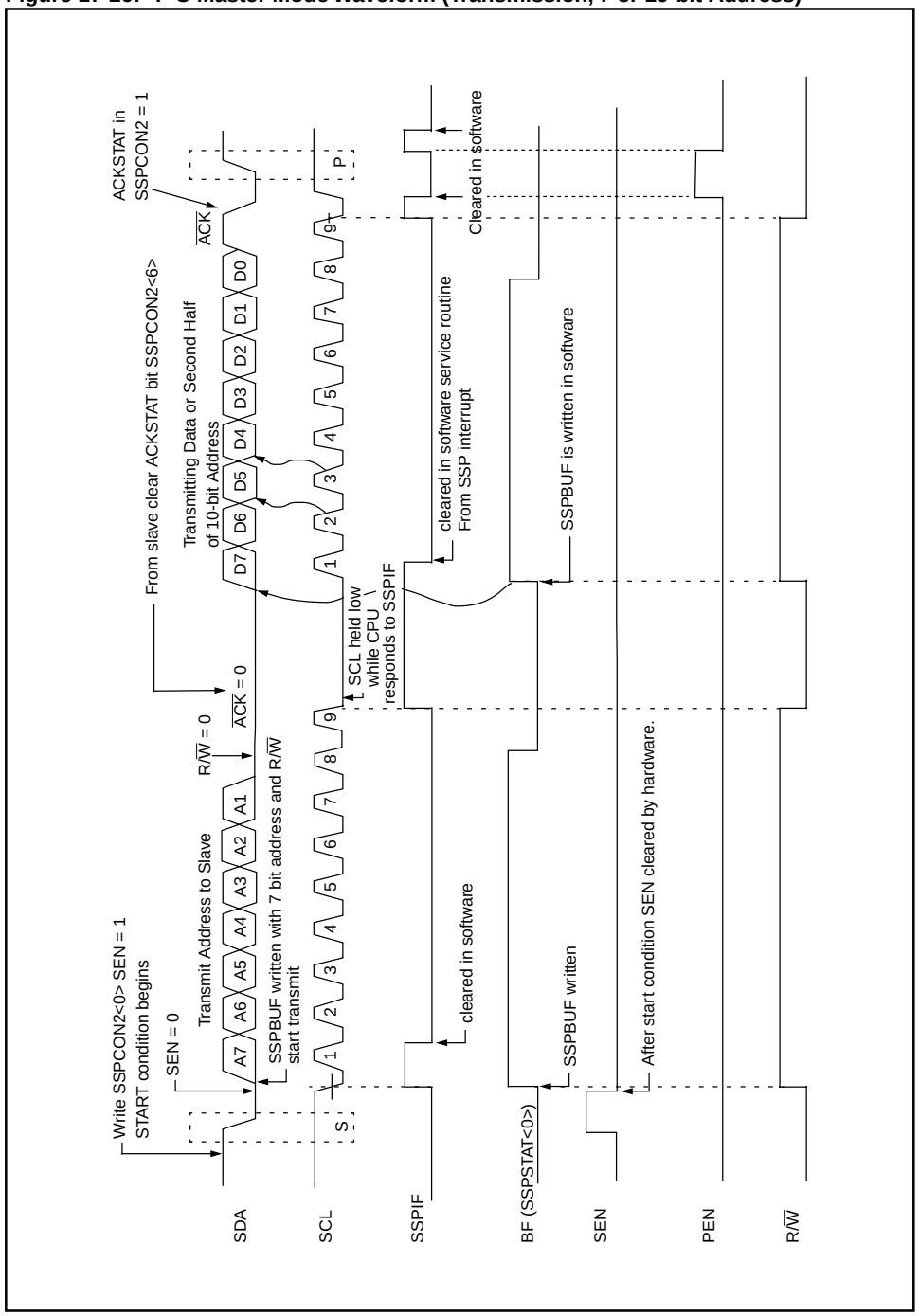
If the user writes the SSPBUF when a transmit is already in progress (i.e. SSPSR is still shifting out a data byte), then WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

WCOL must be cleared in software.

17.4.11.3 ACKSTAT Status Flag

In transmit mode, the ACKSTAT bit (SSPCON2<6>) is cleared when the slave has sent an acknowledge ($\overline{ACK}$ = 0), and is set when the slave does not acknowledge ($\overline{ACK}$ = 1). A slave sends an acknowledge when it has recognized its address (including a general call), or when the slave has properly received its data.

Figure 17-26: I^2C Master Mode Waveform (Transmission, 7 or 10-bit Address)



20.8 Comparator Interrupts

The comparator interrupt flag is set whenever the comparators value changes relative to the last value loaded into CMxOUT bits. Software will need to maintain information about the status of the output bits, as read from CMCON<7:6>, to determine the actual change that has occurred. The CMIF bit, is the comparator interrupt flag. The CMIF bit must be cleared. Since it is also possible to set this bit, a simulated interrupt may be initiated.

The CMIE bit and the PEIE bit (INTCON<6>) must be set to enable the interrupt. In addition, the GIE bit must also be set. If any of these bits are clear, the interrupt is not enabled, though the CMIF bit will still be set if an interrupt condition occurs.

The user, in the interrupt service routine, can clear the interrupt in the following manner:

a) Any read or write of the CMCON register. This will load the CMCON register with the new value with the CMxOUT bits.

b) Clear the CMIF flag bit.

An interrupt condition will continue to set the CMIF flag bit. Reading CMCON will end the interrupt condition, and allow the CMIF flag bit to be cleared.

20.9 Comparator Operation During SLEEP

When a comparator is active and the device is placed in SLEEP mode, the comparator remains active and the interrupt is functional if enabled. This interrupt will wake up the device from SLEEP mode when enabled. While the comparator is powered-up, each comparator that is operational will consume additional current as shown in the comparator specifications. To minimize power consumption while in SLEEP mode, turn off the comparators, CM2:CM0 = 111, before entering sleep. If the device wakes-up from sleep, the contents of the CMCON register are not affected.

20.10 Effects of a RESET

A device reset forces the CMCON register to its reset state. This forces the comparator module to be in the comparator reset mode, CM2:CM0 = 000. This ensures that all potential inputs are analog inputs. Device current is minimized when analog inputs are present at reset time. The comparators will be powered-down during the reset interval.

Register 25-2: LCDPS Register

U-0	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	—	LP3	LP2	LP1	LP0
bit 7							bit 0

bit 7:4 **Unimplemented,** read as '0'

bit 3:0 **LP3:LP0**: Frame Clock Prescale Selection bits

LMUX1:LMUX0	Multiplex	Frame Frequency =
00	Static	Clock source / (128 * (LP3:LP0 + 1))
01	1/2	Clock source / (128 * (LP3:LP0 + 1))
10	1/3	Clock source / (96 * (LP3:LP0 + 1))
11	1/4	Clock source / (128 * (LP3:LP0 + 1))

Legend		
R = Readable bit	W = Writable bit	
U = Unimplemented bit, read as '0'		- n = Value at POR reset

Register 25-3: Generic LCDD (Pixel Data) Register Layout

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
SEGs COMc	SEGs COMc	SEGs COMc	SEGs COMc	SEGs COMc	SEGs COMc	SEGs COMc	SEGs COMc
bit 7							bit 0

bit 7:0 **SEGsCOMc**: Pixel Data bit for segment s and common c

1 = Pixel on (dark)
0 = Pixel off (clear)

Legend		
R = Readable bit	W = Writable bit	
U = Unimplemented bit, read as '0'		- n = Value at POR reset

25.5.2 Segment Enables

The LCDSE register is used to select the pin function for groups of pins. The selection allows each group of pins to operate as either LCD drivers or digital only pins. To configure the pins as a digital port, the corresponding bits in the LCDSE register must be cleared.

If the pin is a digital input the corresponding TRIS bit controls the data direction. Any bit set in the LCDSE register overrides any bit settings in the corresponding TRIS register.

Note 1: On a Power-on Reset, the LCD pins are configured as LCD drivers.

Note 2: The LMUX1:LMUX0 bits take precedence over the LCDSE bit settings for pins RD7, RD6 and RD5.

Example 25-1: Static MUX with 32 Segments

```
    BCF   STATUS,RP0   ; Select Bank2
    BSF   STATUS,RP1   ;
    BCF   LCDCON,LMUX1 ; Select Static MUX
    BCF   LCDCON,LMUX0 ;
    MOVLW 0xFF         ; Make PortD,E,F,G LCD pins
    MOVWF LCDSE        ; configure rest of LCD
```

Example 25-2: 1/3 MUX with 13 Segments

```
    BCF   STATUS,RP0   ; Select Bank2
    BSF   STATUS,RP1   ;
    BSF   LCDCON,LMUX1 ; Select 1/3 MUX
    BCF   LCDCON,LMUX0 ;
    MOVLW 0x87         ; Make PORTD<7:0> & PORTE<6:0> LCD pins
    MOVWF LCDSE        ; configure rest of LCD
```

31.3.2 IDD vs. Frequency

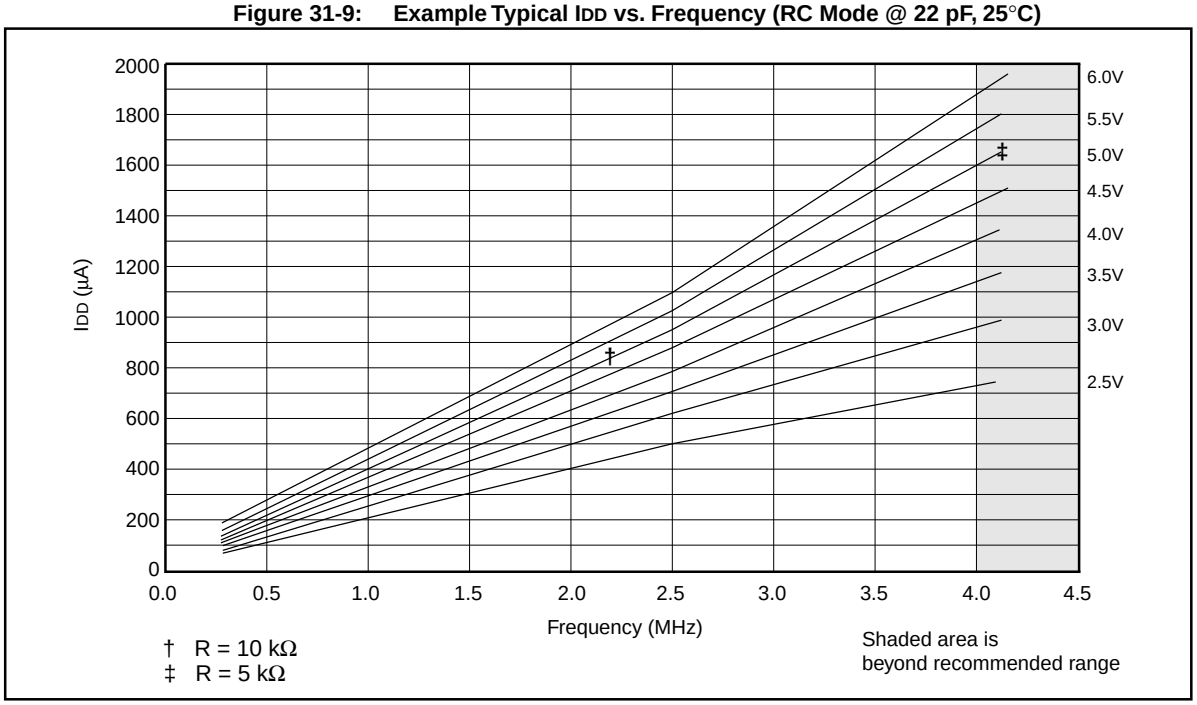
IDD is the current (I) that the device consumes when the device is in operating mode. This test is taken with all I/O as inputs, either pulled high or low. That is, there are no floating inputs, nor are any pins driving an output (with a load).

The IDD vs. Frequency charts measure the results on a Microchip automated bench setup, called the DCS (**D**ata **C**ollection **S**ystem). The DCS accurately reflects the device and specified component values, that is, it does not add stray capacitance or current.

31.3.2.1 RC Measurements

For the RC measurement, the DCS selects a resistor and capacitor value, and then varies the voltage over the specified range. As the voltage is changed, the frequency of operation changes. For a fixed RC, as VDD increases, the frequency increases. After the measurement, at this RC, has been taken, the RC value is changed and the measurements are taken again. Each point on the graph corresponds to a device voltage, resistor value (R), and capacitor value (C).

Figure 31-9: Example Typical IDD vs. Frequency (RC Mode @ 22 pF, 25°C)



A.1 Initiating and Terminating Data Transfer

During times of no data transfer (idle time), both the clock line (SCL) and the data line (SDA) are pulled high through the external pull-up resistors. The START and STOP conditions determine the start and stop of data transmission. The START condition is defined as a high to low transition of the SDA when the SCL is high. The STOP condition is defined as a low to high transition of the SDA when the SCL is high. Figure A-1 shows the START and STOP conditions. The master generates these conditions for starting and terminating data transfer. Due to the definition of the START and STOP conditions, when data is being transmitted, the SDA line can only change state when the SCL line is low.

Figure A-1: Start and Stop Conditions

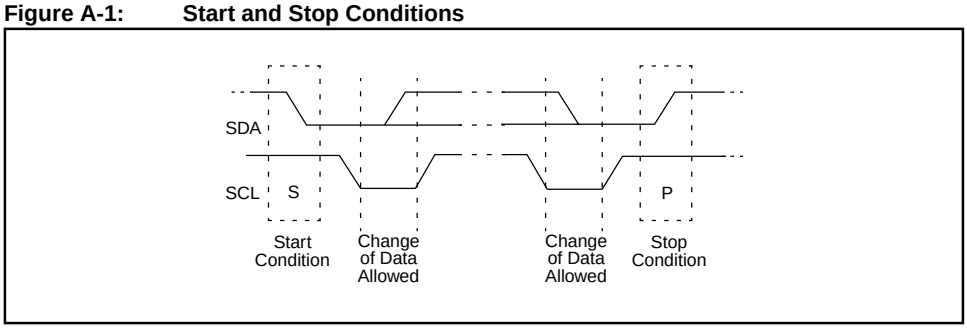


Table A-1: I^2C Bus Terminology

Term	Description
Transmitter	The device that sends the data to the bus.
Receiver	The device that receives the data from the bus.
Master	The device which initiates the transfer, generates the clock and terminates the transfer.
Slave	The device addressed by a master.
Multi-master	More than one master device in a system. These masters can attempt to control the bus at the same time without corrupting the message.
Arbitration	Procedure that ensures that only one of the master devices will control the bus. This ensure that the transfer data does not get corrupted.
Synchronization	Procedure where the clock signals of two or more devices are synchronized.

Figure A-12: I²C Bus Data Timing Specification

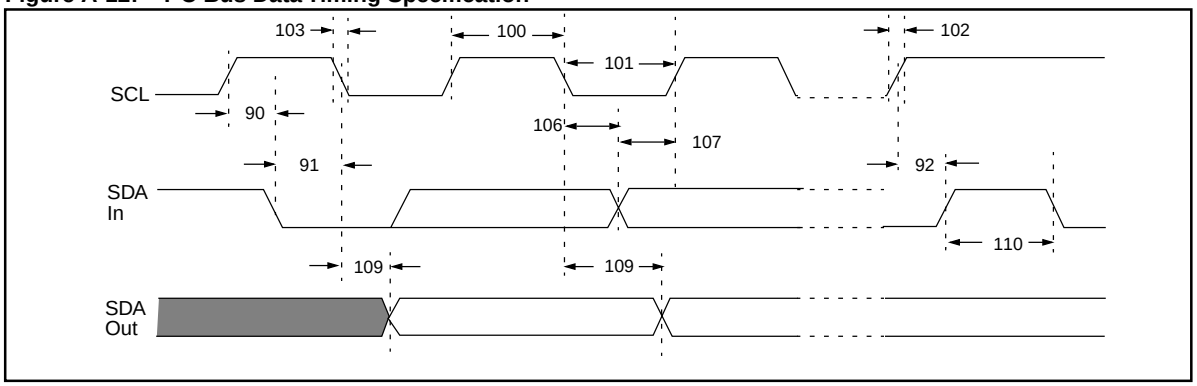


Table A-3: I²C Bus Data Timing Specification

Microchip Parameter No.	Sym	Characteristic		Min	Max	Units	Conditions
100	THIGH	Clock high time	100 kHz mode	4.0	—	µs	
			400 kHz mode	0.6	—	µs	
101	TLOW	Clock low time	100 kHz mode	4.7	—	µs	
			400 kHz mode	1.3	—	µs	
102	TR	SDA and SCL rise time	100 kHz mode	—	1000	ns	
			400 kHz mode	20 + 0.1Cb	300	ns	Cb is specified to be from 10 to 400 pF
103	TF	SDA and SCL fall time	100 kHz mode	—	300	ns	
			400 kHz mode	20 + 0.1Cb	300	ns	Cb is specified to be from 10 to 400 pF
90	TSU:STA	START condition setup time	100 kHz mode	4.7	—	µs	Only relevant for repeated START condition
			400 kHz mode	0.6	—	µs	
91	THD:STA	START condition hold time	100 kHz mode	4.0	—	µs	After this period the first clock pulse is generated
			400 kHz mode	0.6	—	µs	
106	THD:DAT	Data input hold time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	µs	
107	TSU:DAT	Data input setup time	100 kHz mode	250	—	ns	Note 2
			400 kHz mode	100	—	ns	
92	TSU:STO	STOP condition setup time	100 kHz mode	4.7	—	µs	
			400 kHz mode	0.6	—	µs	
109	TAA	Output valid from clock	100 kHz mode	—	3500	ns	Note 1
			400 kHz mode	—	1000	ns	
110	TBUF	Bus free time	100 kHz mode	4.7	—	µs	Time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	µs	
D102	Cb	Bus capacitive loading		—	400	pF	

Note 1: As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of START or STOP conditions.

2: A fast-mode I²C-bus device can be used in a standard-mode I²C-bus system, but the requirement tsu;DAT ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line
TR max.+tsu;DAT = 1000 + 250 = 1250 ns (according to the standard-mode I²C bus specification) before the SCL line is released.

C

Capture

A function of the CCP module in which the value of a timer/counter is “captured”, into a holding register, when a predetermined event occurs.

CCP

Capture, Compare, Pulse Width Modulation (PWM). This module can be configured to operate as an input capture, or a timer compare, or a PWM output.

Common RAM

This is a region of the data memory RAM that is the same RAM location across all banks. This common RAM maybe implemented between addresses 70h -7Fh (inclusive). This common area is useful for the saving of required variables during context switching (such as during an interrupt).

Compare

A function of the CCP module in which the device will perform an action when a timer’s register value matches the value in the compare register.

Compare Register

A 16-bit register that contains a value that is compared to the 16-bit TMR1 register. The compare function triggers when the counter matches the contents of the compare register.

Capture Register

A 16-bit register that gets loaded with the value of the 16-bit TMR1 register when a capture event occurs.

Configuration Word

This is a location that specifies the characteristics that the device will have for operation (such as oscillator mode, WDT enable, start-up timer enables). These characteristics can be specified at time of device programming. For EPROM memory devices, as long as the bit is a '1', it may at a later time be programmed as a '0'. The device must be erased for a '0' to be returned to a '1'.

Conversion Time (Tconv)

This is related to Analog to Digital (A/D) converters. This is the time that the A/D converter requires to convert the analog voltage level on the holding capacitor to a digital value.

CPU

Central Processing Unit. Decodes the instructions, and determines the operands that are needed and the operations that need to be done. Arithmetic, logical, or shift operations will be passed to the ALU.