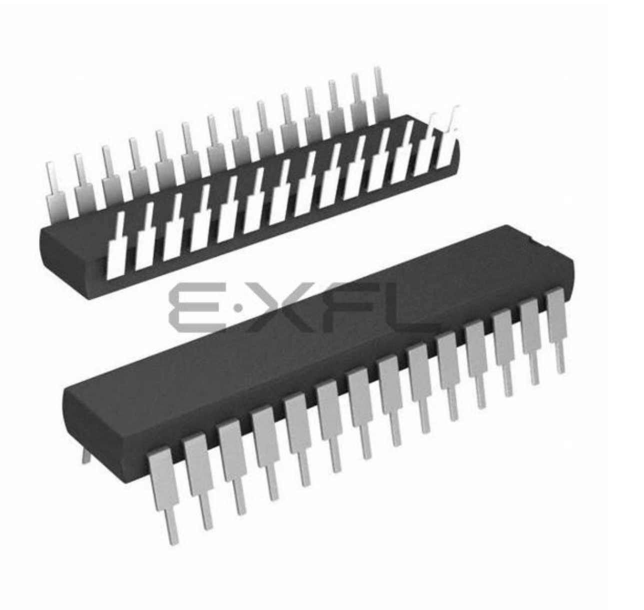


What is "Embedded - Microcontrollers"?

"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
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	22
Program Memory Size	3.5KB (2K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	4V ~ 5.5V
Data Converters	A/D 5x8b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	28-DIP (0.300", 7.62mm)
Supplier Device Package	28-SPDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f72-i-sp

2.0 MEMORY ORGANIZATION

There are two memory blocks in the PIC16F72 device. These are the program memory and the data memory. Each block has separate buses so that concurrent access can occur. Program memory and data memory are explained in this section. Program memory can be read internally by the user code (see Section 7.0).

The data memory can further be 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.

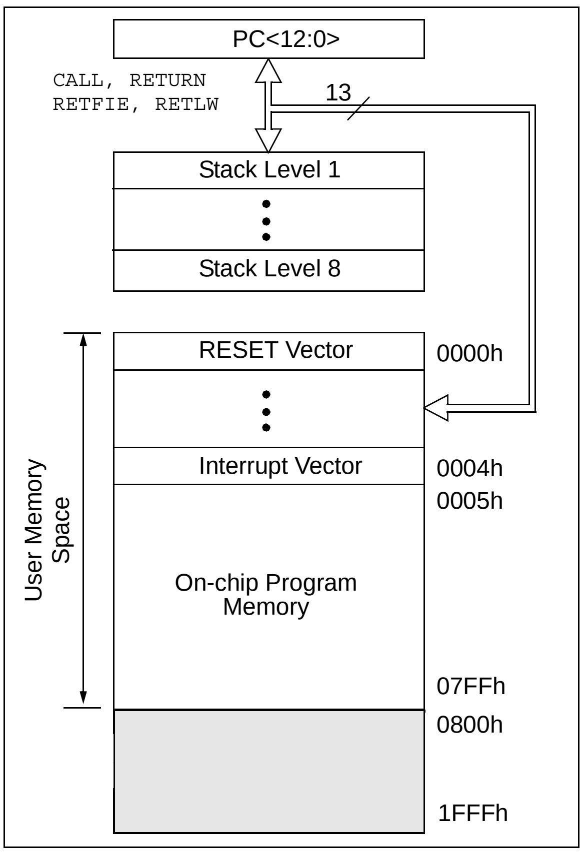
Additional information on device memory may be found in the PIC™ Mid-Range Reference Manual, (DS33023).

2.1 Program Memory Organization

PIC16F72 devices have a 13-bit program counter capable of addressing a 8K x 14 program memory space. The address range for this program memory is 0000h - 07FFh. Accessing a location above the physically implemented address will cause a wraparound.

The RESET Vector is at 0000h and the Interrupt Vector is at 0004h.

FIGURE 2-1: PROGRAM MEMORY MAP AND STACK



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 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).

2.2.1 GENERAL PURPOSE REGISTER FILE

The register file can be accessed either directly, or indirectly, through the File Select Register FSR (see Section 2.5).

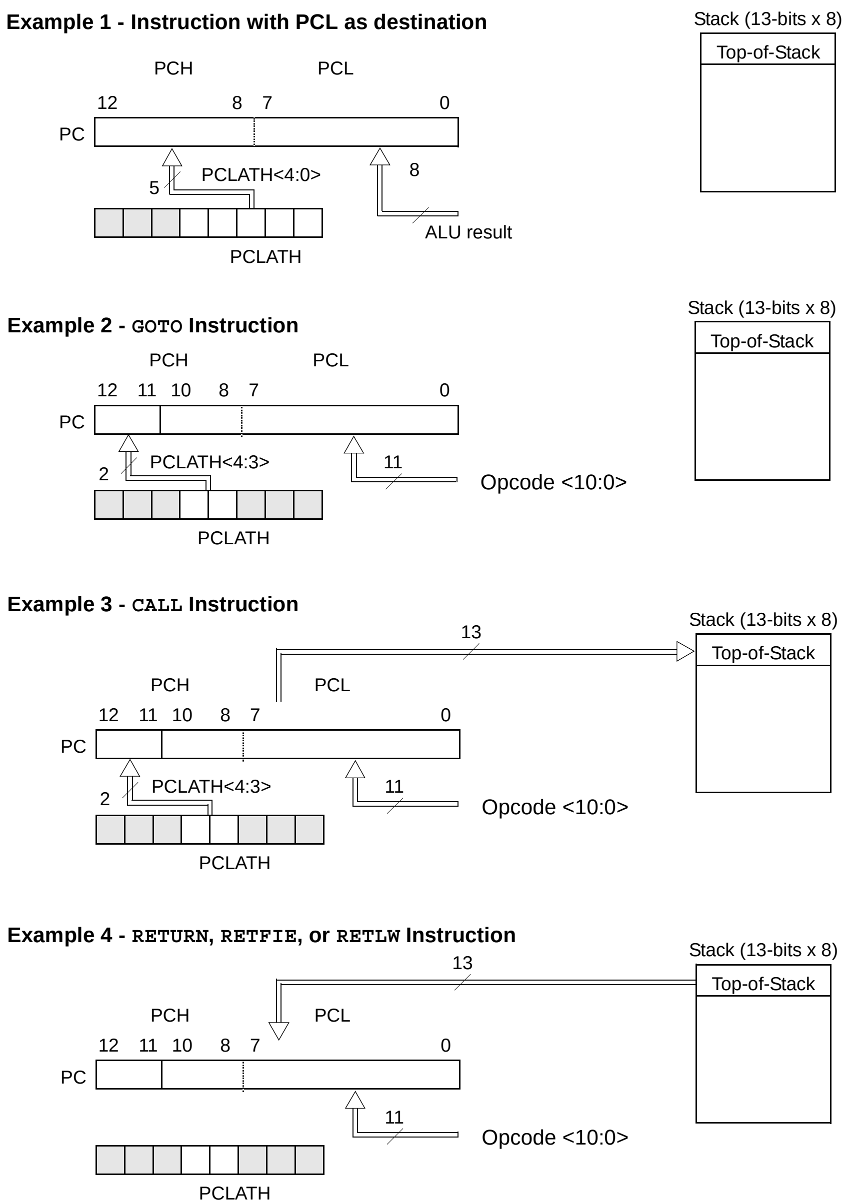
2.3 PCL and PCLATH

The program counter (PC) specifies the address of the instruction to fetch for execution. The PC is 13-bits wide. The low byte is called the PCL register. This register is readable and writable. The high byte is called the PCH register. This register contains the PC<12:8> bits and is not directly readable or writable. All updates to the PCH register go through the PCLATH register.

Figure 2-3 shows the four situations for the loading of the PC.

- Example 1 shows how the PC is loaded on a write to PCL (PCLATH<4:0> → PCH).
- Example 2 shows how the PC is loaded during a GOTO instruction (PCLATH<4:3> → PCH).
- Example 3 shows how the PC is loaded during a CALL instruction (PCLATH<4:3> → PCH), with the PC loaded (PUSH'd) onto the Top-of-Stack.
- Example 4 shows how the PC is loaded during one of the return instructions, where the PC is loaded (POP'd) from the Top-of-Stack.

FIGURE 2-3: LOADING OF PC IN DIFFERENT SITUATIONS



Note: PCLATH is not updated with the contents of PCH.

3.0 I/O PORTS

Some pins for these I/O ports are multiplexed with an alternate function for the peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin.

Additional information on I/O ports may be found in the PIC™ Mid-Range MCU Reference Manual, (DS33023).

3.1 PORTA and the TRISA Register

PORTA is a 6-bit wide, bi-directional port. The corresponding data direction register is TRISA. Setting a TRISA bit (= 1) will make the corresponding PORTA pin an input (i.e., put the corresponding output driver in a Hi-Impedance mode). Clearing a TRISA bit (= 0) will make the corresponding PORTA pin an output (i.e., put the contents of the output latch on the selected pin).

Reading the PORTA register, reads the status of the pins, whereas writing to it will write to the port latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, this value is modified and then written to the port data latch.

Pin RA4 is multiplexed with the Timer0 module clock input to become the RA4/T0CKI pin. The RA4/T0CKI pin is a Schmitt Trigger input and an open drain output. All other RA port pins have TTL input levels and full CMOS output drivers.

Other PORTA pins are multiplexed with analog inputs and analog VREF input. The operation of each pin is selected by clearing/setting the control bits in the ADCON1 register (A/D Control Register1).

Note: On a Power-on Reset, these pins are configured as analog inputs and read as '0'.

The TRISA register controls the direction of the RA pins, even when they are being used as analog inputs. The user must ensure the bits in the TRISA register are maintained set when using them as analog inputs.

EXAMPLE 3-1: INITIALIZING PORTA

```
BANKSEL   PORTA    ; select bank for PORTA
CLRF      PORTA    ; Initialize PORTA by
                   ; clearing output
                   ; data latches
BANKSEL   ADCON1   ; Select Bank for ADCON1
MOVLW     0x06     ; Configure all pins
MOVWF     ADCON1   ; as digital inputs
MOVLW     0xCF     ; Value used to
                   ; initialize data
                   ; direction
MOVWF     TRISA    ; Set RA<3:0> as inputs
                   ; RA<5:4> as outputs
                   ; TRISA<7:6> are always
                   ; read as '0'.
```

FIGURE 3-1: BLOCK DIAGRAM OF RA3:RA0 AND RA5 PINS

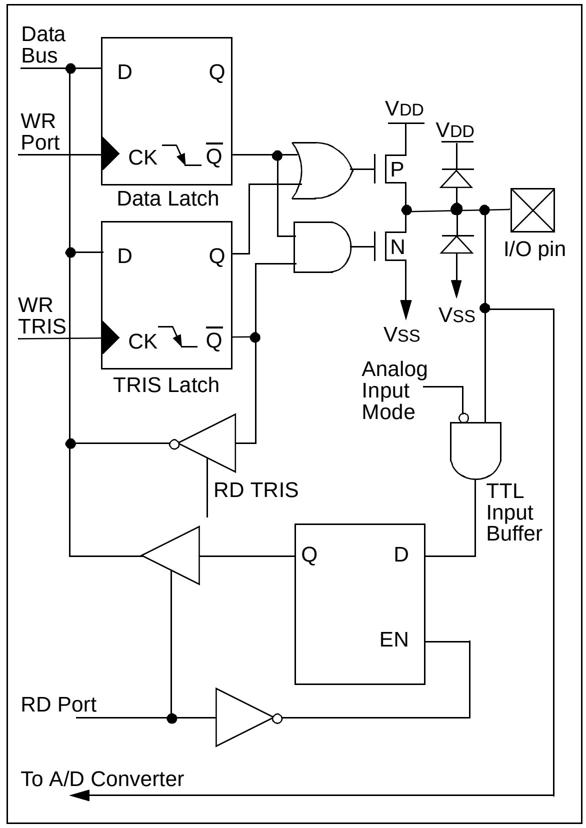
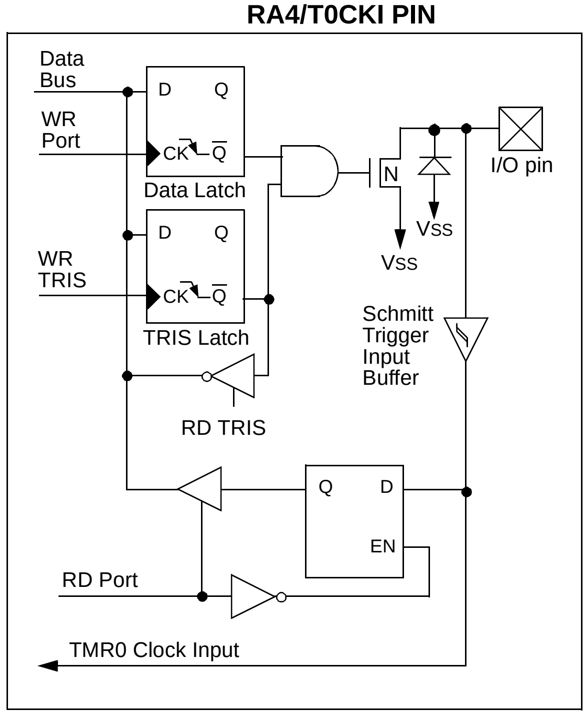


FIGURE 3-2: BLOCK DIAGRAM OF RA4/T0CKI PIN



7.0 READING PROGRAM MEMORY

The FLASH Program Memory is readable during normal operation over the entire V_{DD} range. It is indirectly addressed through Special Function Registers (SFR). Up to 14-bit wide numbers can be stored in memory for use as calibration parameters, serial numbers, packed 7-bit ASCII, etc. Executing a program memory location containing data that forms an invalid instruction results in a `NOP`.

There are five SFRs used to read the program and memory:

- PMCON1
- PMDATL
- PMDATH
- PMADRL
- PMADRH

The program memory allows word reads. Program memory access allows for checksum calculation and reading calibration tables.

When interfacing to the program memory block, the PMDATH:PMDATL registers form a two-byte word, which holds the 14-bit data for reads. The PMADRH:PMADRL registers form a two-byte word, which holds the 13-bit address of the FLASH location being accessed. This device has up to 2K words of program FLASH, with an address range from 0h to 07FFh. The unused upper bits PMDATH<7:6> and PMADRH<7:5> are not implemented and read as zeros.

7.1 PMADR

The address registers can address up to a maximum of 8K words of program FLASH.

When selecting a program address value, the MSByte of the address is written to the PMADRH register and the LSByte is written to the PMADRL register. The upper MSbits of PMADRH must always be clear.

7.2 PMCON1 Register

PMCON1 is the control register for memory accesses.

The control bit RD initiates read operations. This bit cannot be cleared, only set, in software. It is cleared in hardware at the completion of the read operation.

REGISTER 7-1: PMCON1: PROGRAM MEMORY CONTROL REGISTER 1 (ADDRESS 18Ch)

R-1	U-0	U-0	U-0	U-0	U-0	U-0	R/S-0
reserved	—	—	—	—	—	—	RD
bit 7							bit 0

bit 7 **Reserved:** Read as ‘1’

bit 6-1 **Unimplemented:** Read as ‘0’

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

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

8.3 PWM Mode

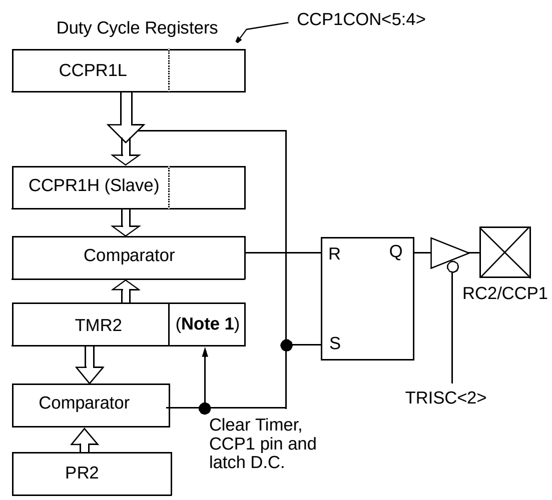
In Pulse Width Modulation (PWM) mode, the CCP1 pin produces up to a 10-bit resolution PWM output. Since the CCP1 pin is multiplexed with the PORTC data latch, the TRISC<2> bit must be cleared to make the CCP1 pin an output.

> **Note:** Clearing the CCP1CON register will force the CCP1 PWM output latch to the default low level. This is not the PORTC I/O data latch.

Figure 8-3 shows a simplified block diagram of the CCP module in PWM mode.

For a step by step procedure on how to set up the CCP module for PWM operation, see Section 8.3.3.

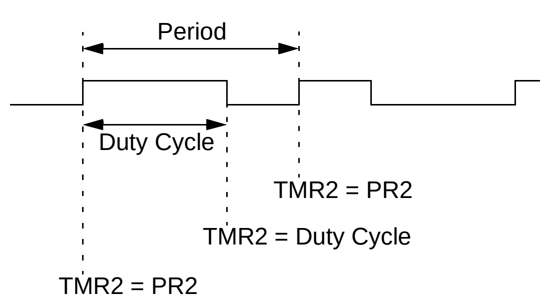
FIGURE 8-3: SIMPLIFIED PWM BLOCK DIAGRAM



Note 1: 8-bit timer is concatenated with 2-bit internal Q clock or 2 bits of the prescaler to create 10-bit time-base.

A PWM output (Figure 8-4) has a time-base (period) and a time that the output stays high (duty cycle). The frequency of the PWM is the inverse of the period (1/period).

FIGURE 8-4: PWM OUTPUT



8.3.1 PWM PERIOD

The PWM period is specified by writing to the PR2 register. The PWM period can be calculated using the formula in Equation 8-1.

EQUATION 8-1: PWM PERIOD

$$\text{PWM period} = [(PR2) + 1] \bullet 4 \bullet T_{OSC} \bullet (\text{TMR2 prescale value})$$

PWM frequency is defined as 1 / [PWM period].

When TMR2 is equal to PR2, the following three events occur on the next increment cycle:

- TMR2 is cleared
- The CCP1 pin is set (exception: if PWM duty cycle = 0%, the CCP1 pin will not be set)
- The PWM duty cycle is latched from CCPR1L into CCPR1H

> **Note:** The Timer2 postscaler (see Section 6.0) is not used in the determination of the PWM frequency. The postscaler could be used to have a servo update rate at a different frequency than the PWM output.

8.3.2 PWM DUTY CYCLE

The PWM duty cycle is specified by writing to the CCPR1L register and to the CCP1CON<5:4> bits. Up to 10-bit resolution is available: the CCPR1L contains the eight MSbs and the CCP1CON<5:4> contains the two LSbs. This 10-bit value is represented by CCPR1L:CCP1CON<5:4>. Equation 8-2 is used to calculate the PWM duty cycle in time.

EQUATION 8-2: PWM DUTY CYCLE

$$\text{PWM duty cycle} = (CCPR1L{:}CCP1CON<5:4>) \bullet T_{OSC} \bullet (\text{TMR2 prescale value})$$

CCPR1L and CCP1CON<5:4> can be written to at any time, but the duty cycle value is not latched into CCPR1H until after a match between PR2 and TMR2 occurs (i.e., the period is complete). In PWM mode, CCPR1H is a read only register.

The CCPR1H register and a 2-bit internal latch are used to double buffer the PWM duty cycle. This double buffering is essential for glitchless PWM operation.

When the CCPR1H and 2-bit latch match TMR2, concatenated with an internal 2-bit Q clock or 2 bits of the TMR2 prescaler, the CCP1 pin is cleared.

10.0 ANALOG-TO-DIGITAL CONVERTER (A/D) MODULE

The analog-to-digital (A/D) converter module has five inputs for the PIC16F72.

The A/D allows conversion of an analog input signal to a corresponding 8-bit digital number. The output of the sample and hold is the input into the converter, which generates the result via successive approximation. The analog reference voltage is software selectable to either the device's positive supply voltage (V_{DD}) or the voltage level on the RA3/AN3/V_{REF} pin.

The A/D converter has a unique feature of being able to operate while the device is in SLEEP mode. To operate in SLEEP, the A/D conversion clock must be derived from the A/D's internal RC oscillator.

The A/D module has three registers:

- A/D Result Register ADRES
- A/D Control Register 0 ADCON0
- A/D Control Register 1 ADCON1

A device RESET forces all registers to their RESET state. This forces the A/D module to be turned off and any conversion is aborted.

The ADCON0 register, shown in Register 10-1, controls the operation of the A/D module. The ADCON1 register, shown in Register 10-2, configures the functions of the port pins. The port pins can be configured as analog inputs (RA3 can also be a voltage reference) or a digital I/O.

For more information on use of the A/D Converter, see *AN546 - Use of A/D Converter*, or refer to the PIC™ Mid-Range MCU Family Reference Manual (DS33023).

REGISTER 10-1: ADCON0: A/D CONTROL REGISTER 0 (ADDRESS 1Fh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	—	ADON
bit 7							bit 0

bit 7-6 **ADCS<1:0>:** A/D Conversion Clock Select bits
00 = FOSC/2
01 = FOSC/8
10 = FOSC/32
11 = FRC (clock derived from the internal A/D module RC oscillator)

bit 5-3 **CHS<2:0>:** Analog Channel Select bits
000 = Channel 0, (RA0/AN0)
001 = Channel 1, (RA1/AN1)
010 = Channel 2, (RA2/AN2)
011 = Channel 3, (RA3/AN3)
100 = Channel 4, (RA5/AN4)

bit 2 **GO/$\overline{\text{DONE}}$:** A/D Conversion Status bit
<u>If ADON = 1:</u>
1 = A/D conversion in progress (setting this bit starts the A/D conversion)
0 = A/D conversion not in progress (this bit is automatically cleared by hardware when the A/D conversion is complete)

bit 1 **Unimplemented:** Read as '0'

bit 0 **ADON:** A/D On bit
1 = A/D converter module is operating
0 = A/D converter module is shut-off and consumes no operating current

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 10-1: A/D BLOCK DIAGRAM

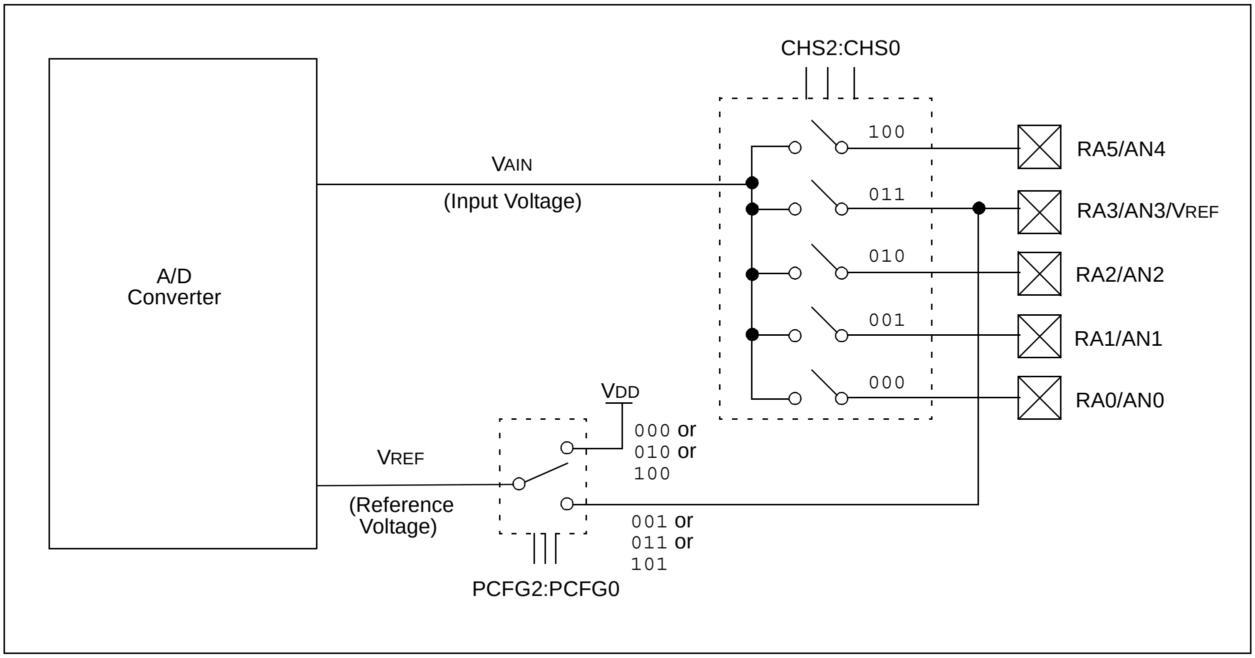
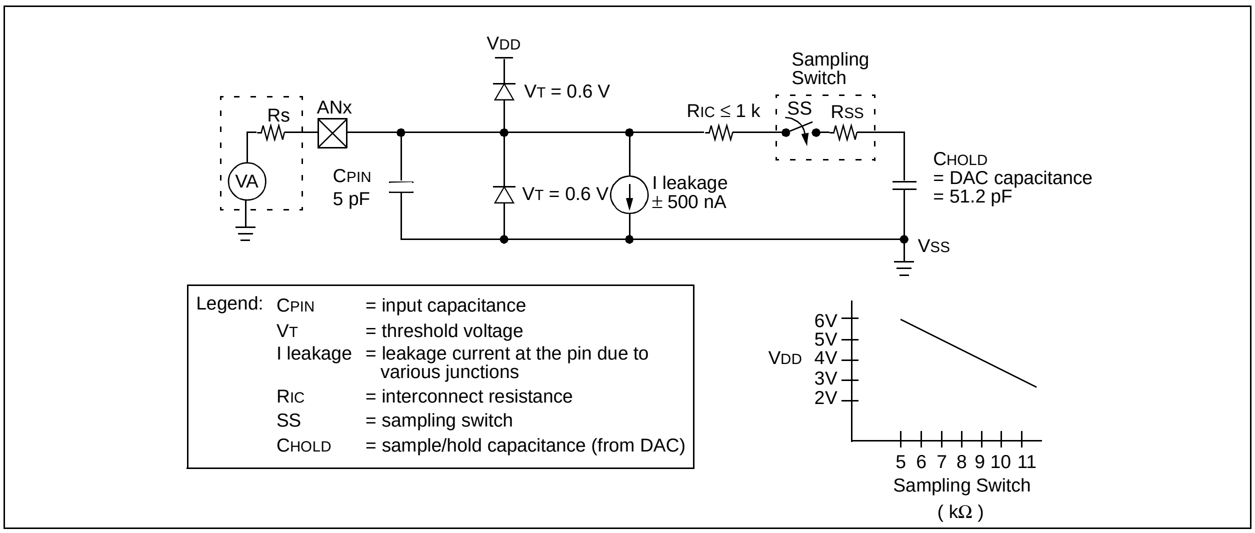


FIGURE 10-2: ANALOG INPUT MODEL



Legend:

CPIN	= input capacitance
VT	= threshold voltage
I leakage	= leakage current at the pin due to various junctions
RIC	= interconnect resistance
SS	= sampling switch
CHOLD	= sample/hold capacitance (from DAC)

10.1 A/D Acquisition Requirements

For the A/D converter to meet its specified accuracy, the charge holding capacitor (C_{HOLD}) must be allowed to fully charge to the input channel voltage level. The analog input model is shown in Figure 10-2. The source impedance (R_S) and the internal sampling switch (R_{SS}) impedance directly affect the time required to charge the capacitor C_{HOLD}. The sampling switch (R_{SS}) impedance varies over the device voltage (V_{DD}). The source impedance affects the offset voltage at the analog input (due to pin leakage current).

The maximum recommended impedance for analog sources is 10 kΩ. After the analog input channel is selected (changed), this acquisition must be done before the conversion can be started.

To calculate the minimum acquisition time, T_{ACQ}, see the PIC™ Mid-Range MCU Reference Manual, (DS33023). In general, however, given a max of 10 kΩ and at a temperature of 100°C, T_{ACQ} will be no more than 16 µs.

10.2 Selecting the A/D Conversion Clock

The A/D conversion time per bit is defined as T_{AD}. The A/D conversion requires 9.0 T_{AD} per 8-bit conversion. The source of the A/D conversion clock is software selectable. The four possible options for T_{AD} are:

- 2 T_{OSC}
- 8 T_{OSC}
- 32 T_{OSC}
- Internal RC oscillator (2 - 6 µs)

For correct A/D conversions, the A/D conversion clock (T_{AD}) must be selected to ensure a minimum T_{AD} time as small as possible, but no less than 1.6 µs and not greater than 6.4 µs.

Table 10-1 shows the resultant T_{AD} times derived from the device operating frequencies and the A/D clock source selected.

10.3 Configuring Analog Port Pins

The ADCON1, and TRISA registers control the operation of the A/D port pins. The port pins that are desired as analog inputs must have their corresponding TRIS bits set (input). If the TRIS bit is cleared (output), the digital output level (V_{OH} or V_{OL}) will be converted.

The A/D operation is independent of the state of the CHS<2:0> bits and the TRIS bits.

> **Note 1:** When reading the port register, all pins configured as analog input channels will read as cleared (a low level). Pins configured as digital inputs, will convert an analog input. Analog levels on a digitally configured input will not affect the conversion accuracy.
>
> **2:** Analog levels on any pin that is defined as a digital input (including the AN4:AN0 pins), may cause the input buffer to consume current out of the device specification.

10.4 A/D Conversions

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

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

TABLE 10-1: T_{AD} vs. MAXIMUM DEVICE OPERATING FREQUENCIES (STANDARD DEVICES (C))

AD Clock Source (T_{AD})		Maximum Device Frequency
Operation	**ADCS<1:0>**	**Max.**
2 T_{OSC}	`00`	1.25 MHz
8 T_{OSC}	`01`	5 MHz
32 T_{OSC}	`10`	20 MHz
RC(1, 2)	`11`	**(Note 1)**

Note 1: The RC source has a typical T_{AD} time of 4 µs, but can vary between 2-6 µs.
2: When the device frequencies are greater than 1 MHz, the RC A/D conversion clock source is only recommended for SLEEP operation.

10.5 A/D Operation During SLEEP

The A/D module can operate during SLEEP mode. This requires that the A/D clock source be set to RC (ADCS1:ADCS0 = `11`). When the RC clock source is selected, the A/D module waits one instruction cycle before starting the conversion. This allows the `SLEEP` instruction to be executed, which eliminates all digital switching noise from the conversion. When the conversion is completed, the GO/DONE bit will be cleared, and the result loaded into the ADRES register. If the A/D interrupt is enabled, the device will wake-up from SLEEP. If the A/D interrupt is not enabled, the A/D module will then be turned off, although the ADON bit will remain set.

When the A/D clock source is another clock option (not RC), a `SLEEP` instruction will cause the present conversion to be aborted and the A/D module to be turned off, though the ADON bit will remain set.

Turning off the A/D places the A/D module in its lowest current consumption state.

Note:	For the A/D module to operate in SLEEP, the A/D clock source must be set to RC (ADCS1:ADCS0 = `11`). To perform an A/D conversion in SLEEP, ensure the `SLEEP` instruction immediately follows the instruction that sets the GO/DONE bit.

10.6 Effects of a RESET

A device RESET forces all registers to their RESET state. The A/D module is disabled and any conversion in progress is aborted. All A/D input pins are configured as analog inputs.

The ADRES register will contain unknown data after a Power-on Reset.

10.7 Use of the CCP Trigger

An A/D conversion can be started by the “special event trigger” of the CCP1 module. This requires that the CCP1M3:CCP1M0 bits (CCP1CON<3:0>) be programmed as `1011` and that the A/D module is enabled (ADON bit is set). When the trigger occurs, the GO/DONE bit will be set, starting the A/D conversion, and the Timer1 counter will be reset to zero. Timer1 is reset to automatically repeat the A/D acquisition period with minimal software overhead (moving the ADRES to the desired location). The appropriate analog input channel must be selected and the minimum acquisition done before the “special event trigger” sets the GO/DONE bit (starts a conversion).

If the A/D module is not enabled (ADON is cleared), then the “special event trigger” will be ignored by the A/D module, but will still reset the Timer1 counter.

TABLE 10-2: REGISTERS/BITS ASSOCIATED WITH A/D

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`
1Eh	ADRES	A/D Result Register								`xxxx xxxx`	`uuuu uuuu`
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON	`0000 00-0`	`0000 00-0`
9Fh	ADCON1	—	—	—	—	—	PCFG2	PCFG1	PCFG0	`---- -000`	`---- -000`
05h	PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	`--0x 0000`	`--0u 0000`
85h	TRISA	—	—	PORTA Data Direction Register						`--11 1111`	`--11 1111`

Legend: `x` = unknown, `u` = unchanged, `-` = unimplemented, read as ‘0’. Shaded cells are not used for A/D conversion.

11.2 Oscillator Configurations

11.2.1 OSCILLATOR TYPES

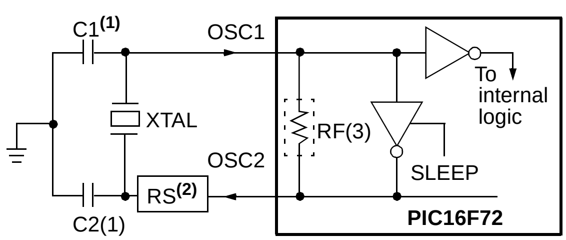
The PIC16F72 can be operated in four different Oscillator modes. The user can program two configuration bits (FOSC1 and FOSC0) to select one of these four modes:

- LP Low Power Crystal
- XT Crystal/Resonator
- HS High Speed Crystal/Resonator
- RC Resistor/Capacitor

11.2.2 CRYSTAL OSCILLATOR/CERAMIC RESONATORS

In XT, LP or HS modes, a crystal or ceramic resonator is connected to the OSC1/CLKI and OSC2/CLKO pins to establish oscillation (Figure 11-1). The PIC16F72 oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturers specifications. When in HS mode, the device can accept an external clock source to drive the OSC1/CLKI pin (Figure 11-2). See Figure 14-1 or Figure 14-2 (depending on the part number and VDD range) for valid external clock frequencies.

FIGURE 11-1: CRYSTAL/CERAMIC RESONATOR OPERATION (HS, XT OR LP OSC CONFIGURATION)



Note 1: See Table 11-1 and Table 11-2 for typical values of C1 and C2.

2: A series resistor (RS) may be required for AT strip cut crystals.

3: RF varies with the crystal chosen.

FIGURE 11-2: EXTERNAL CLOCK INPUT OPERATION (HS OSC CONFIGURATION)

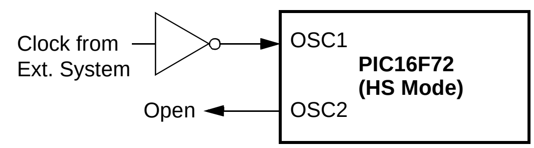


TABLE 11-1: CERAMIC RESONATORS (FOR DESIGN GUIDANCE ONLY)

Typical Capacitor Values Used:			
Mode	**Freq**	**OSC1**	**OSC2**
XT	455 kHz 2.0 MHz 4.0 MHz	56 pF 47 pF 33 pF	56 pF 47 pF 33 pF
HS	8.0 MHz 16.0 MHz	27 pF 22 pF	27 pF 22 pF

Capacitor values are for design guidance only.

These capacitors were tested with the resonators listed below for basic start-up and operation. These values were not optimized.

Different capacitor values may be required to produce acceptable oscillator operation. The user should test the performance of the oscillator over the expected VDD and temperature range for the application.

See the notes at the bottom of page 62 for additional information.

11.14 Power-down Mode (SLEEP)

Power-down mode is entered by executing a `SLEEP` instruction.

If enabled, the Watchdog Timer will be cleared but keeps running, the $\overline{\text{PD}}$ bit (STATUS<3>) is cleared, the $\overline{\text{TO}}$ (STATUS<4>) bit is set, and the oscillator driver is turned off. The I/O ports maintain the status they had before the `SLEEP` instruction was executed (driving high, low, or hi-impedance).

For lowest current consumption in this mode, place all I/O pins at either V_{DD} or V_{SS}, ensure no external circuitry is drawing current from the I/O pin, power-down the A/D and disable external clocks. Pull all I/O pins that are hi-impedance inputs, high or low externally, to avoid switching currents caused by floating inputs. The T0CKI input should also be at V_{DD} or V_{SS} for lowest current consumption. The contribution from on-chip pull-ups on PORTB should also be considered.

The $\overline{\text{MCLR}}$ pin must be at a logic high level (V_{IHMC}).

11.14.1 WAKE-UP FROM SLEEP

The device can wake-up from SLEEP through one of the following events:

1. External RESET input on $\overline{\text{MCLR}}$ pin.
2. Watchdog Timer wake-up (if WDT was enabled).
3. Interrupt from INT pin, RB port change or a peripheral interrupt.

External $\overline{\text{MCLR}}$ Reset will cause a device RESET. All other events are considered a continuation of program execution and cause a "wake-up". The $\overline{\text{TO}}$ and $\overline{\text{PD}}$ bits in the STATUS register can be used to determine the cause of the device RESET. The $\overline{\text{PD}}$ bit, which is set on power-up, is cleared when SLEEP is invoked. The $\overline{\text{TO}}$ bit is cleared if a WDT time-out occurred and caused wake-up.

The following peripheral interrupts can wake the device from SLEEP:

1. TMR1 interrupt. Timer1 must be operating as an asynchronous counter.
2. CCP Capture mode interrupt.
3. Special event trigger (Timer1 in Asynchronous mode using an external clock).
4. SSP (START/STOP) bit detect interrupt.
5. SSP transmit or receive in Slave mode (SPI/I^2C).
6. A/D conversion (when A/D clock source is RC).

Other peripherals cannot generate interrupts since during SLEEP, no on-chip clocks are present.

When the `SLEEP` instruction is being executed, the next instruction (PC + 1) is pre-fetched. For the device to wake-up through an interrupt event, the corresponding interrupt enable bit must be set (enabled). Wake-up occurs regardless of the state of the GIE bit. If the GIE bit is clear (disabled), the device continues execution at the instruction after the `SLEEP` instruction. If the GIE bit is set (enabled), the device executes the instruction after the `SLEEP` instruction and then branches to the interrupt address (0004h). In cases where the execution of the instruction following `SLEEP` is not desirable, the user should have a `NOP` after the `SLEEP` instruction.

11.14.2 WAKE-UP USING INTERRUPTS

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

- If the interrupt occurs **before** the execution of a `SLEEP` instruction, the `SLEEP` instruction will complete as a `NOP`. Therefore, the WDT and WDT postscaler will not be cleared, the $\overline{\text{TO}}$ bit will not be set and $\overline{\text{PD}}$ bits will not be cleared.
- If the interrupt occurs **during or after** the execution of a `SLEEP` instruction, the device will immediately wake-up from SLEEP. The `SLEEP` instruction will be completely executed before the wake-up. Therefore, the WDT and WDT postscaler will be cleared, the $\overline{\text{TO}}$ bit will be set and the $\overline{\text{PD}}$ bit will be cleared.

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

To ensure that the WDT is cleared, a `CLRWDT` instruction should be executed before a `SLEEP` 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'd with the eight-bit literal 'k'. The result is placed in the W register.

MOVLW	Move Literal to W
Syntax:	[*label*] MOVLW k
Operands:	$0 \le k \le 255$
Operation:	k → (W)
Status Affected:	None
Description:	The eight-bit literal 'k' is loaded into W register. The don't cares will assemble as '0's.

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' = '0', the result is placed in the W register. If 'd' = '1', the result is placed back in register 'f'.

MOVWF	Move W to f
Syntax:	[*label*] MOVWF f
Operands:	$0 \le f \le 127$
Operation:	(W) → (f)
Status Affected:	None
Description:	Move data from W register to register 'f'.

MOVF	Move f
Syntax:	[*label*] MOVF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f) → (destination)
Status Affected:	Z
Description:	The contents of register 'f' are moved to a destination dependant upon the status of 'd'. If 'd' = '0', the destination is W register. If 'd' = '1', the destination is file register 'f' itself. 'd' = '1' is useful to test a file register, since status flag Z is affected.

NOP	No Operation
Syntax:	[*label*] NOP
Operands:	None
Operation:	No operation
Status Affected:	None
Description:	No operation.

13.8 MPLAB ICD In-Circuit Debugger

Microchip's In-Circuit Debugger, MPLAB ICD, is a powerful, low cost, run-time development tool. This tool is based on the FLASH PIC MCUs and can be used to develop for this and other PIC microcontrollers. The MPLAB ICD utilizes the in-circuit debugging capability built into the FLASH devices. This feature, along with Microchip's In-Circuit Serial Programming™ protocol, offers cost-effective in-circuit FLASH debugging from the graphical user interface of the MPLAB Integrated Development Environment. This enables a designer to develop and debug source code by watching variables, single-stepping and setting break points. Running at full speed enables testing hardware in real-time.

13.9 PRO MATE II Universal Device Programmer

The PRO MATE II universal device programmer is a full-featured programmer, capable of operating in stand-alone mode, as well as PC-hosted mode. The PRO MATE II device programmer is CE compliant.

The PRO MATE II device programmer has programmable V_{DD} and V_{PP} supplies, which allow it to verify programmed memory at V_{DD} min and V_{DD} max for maximum reliability. It has an LCD display for instructions and error messages, keys to enter commands and a modular detachable socket assembly to support various package types. In stand-alone mode, the PRO MATE II device programmer can read, verify, or program PIC devices. It can also set code protection in this mode.

13.10 PICSTART Plus Entry Level Development Programmer

The PICSTART Plus development programmer is an easy-to-use, low cost, prototype programmer. It connects to the PC via a COM (RS-232) port. MPLAB Integrated Development Environment software makes using the programmer simple and efficient.

The PICSTART Plus development programmer supports all PIC devices with up to 40 pins. Larger pin count devices, such as the PIC16C92X and PIC17C76X, may be supported with an adapter socket. The PICSTART Plus development programmer is CE compliant.

13.11 PICDEM 1 Low Cost PIC Demonstration Board

The PICDEM 1 demonstration board is a simple board which demonstrates the capabilities of several of Microchip's microcontrollers. The microcontrollers supported are: PIC16C5X (PIC16C54 to PIC16C58A), PIC16C61, PIC16C62X, PIC16C71, PIC16C8X, PIC17C42, PIC17C43 and PIC17C44. All necessary hardware and software is included to run basic demo programs. The user can program the sample microcontrollers provided with the PICDEM 1 demonstration board on a PRO MATE II device programmer, or a PICSTART Plus development programmer, and easily test firmware. The user can also connect the PICDEM 1 demonstration board to the MPLAB ICE in-circuit emulator and download the firmware to the emulator for testing. A prototype area is available for the user to build some additional hardware and connect it to the microcontroller socket(s). Some of the features include an RS-232 interface, a potentiometer for simulated analog input, push button switches and eight LEDs connected to PORTB.

13.12 PICDEM 2 Low Cost PIC16CXX Demonstration Board

The PICDEM 2 demonstration board is a simple demonstration board that supports the PIC16C62, PIC16C64, PIC16C65, PIC16C73 and PIC16C74 microcontrollers. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM 2 demonstration board on a PRO MATE II device programmer, or a PICSTART Plus development programmer, and easily test firmware. The MPLAB ICE in-circuit emulator may also be used with the PICDEM 2 demonstration board to test firmware. A prototype area has been provided to the user for adding additional hardware and connecting it to the microcontroller socket(s). Some of the features include a RS-232 interface, push button switches, a potentiometer for simulated analog input, a serial EEPROM to demonstrate usage of the I^2C™ bus and separate headers for connection to an LCD module and a keypad.

14.1 DC Characteristics: PIC16F72 (Industrial, Extended) PIC16LF72 (Industrial) (Continued)

PIC16LF72 (Industrial)			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial				
PIC16F72 (Industrial, Extended)			**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended				
Param No.	**Sym**	**Characteristic**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
	IDD	**Supply Current (Notes 2, 5)**					
D010		PIC16LF72	—	0.4	2.0	mA	XT, RC osc configuration FOSC = 4 MHz, VDD = 3.0V **(Note 4)**
D010A			—	25	48	µA	LP osc configuration FOSC = 32 kHz, VDD = 3.0V, WDT disabled
D010		PIC16F72	-	0.9	4	mA	XT, RC osc configuration FOSC = 4 MHz, VDD = 5.5V **(Note 4)**
D013			—	5.2	15	mA	HS osc configuration FOSC = 20 MHz, VDD = 5.5V
D015*	ΔIBOR	**Brown-out Reset Current (Note 6)**	—	25	200	µA	BOR enabled, VDD = 5.0V
	IPD	**Power-down Current (Notes 3, 5)**					
D020		PIC16LF72	—	2.0	30	µA	VDD = 3.0V, WDT enabled, -40°C to +85°C
D021			—	0.1	5	µA	VDD = 3.0V, WDT disabled, -40°C to +85°C
D020		PIC16F72	—	5.0	42	µA	VDD = 4.0V, WDT enabled, -40°C to +85°C
D021				0.1	19	µA	VDD = 4.0V, WDT disabled, -40°C to +85°C
D023*	ΔIBOR	**Brown-out Reset Current (Note 6)**	—	25	200	µA	BOR enabled, VDD = 5.0V

* 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: This is the limit to which VDD can be lowered without losing RAM data.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors, such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature, also have an impact on the current consumption. The test conditions for all IDD measurements in active Operation mode are:

OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD
MCLR = VDD; WDT enabled/disabled as specified.

3: The power-down current in SLEEP mode does not depend on the oscillator type. Power-down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to VDD and VSS.

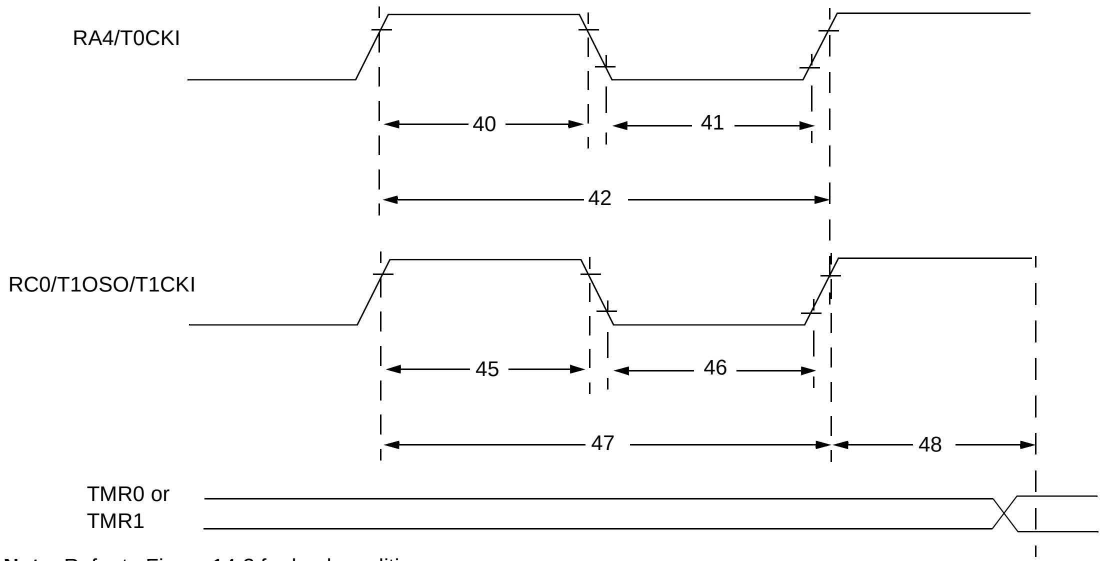
4: For RC osc configuration, current through REXT is not included. The current through the resistor can be estimated by the formula Ir = VDD/2REXT (mA) with REXT in kΩ.

5: Timer1 oscillator (when enabled) adds approximately 20 µA to the specification. This value is from characterization and is for design guidance only. This is not tested.

6: The Δ current is the additional current consumed when this peripheral is enabled. This current should be added to the base IDD or IPD measurement.

7: When BOR is enabled, the device will operate correctly until the VBOR voltage trip point is reached.

FIGURE 14-8: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS



Note: Refer to Figure 14-3 for load conditions.

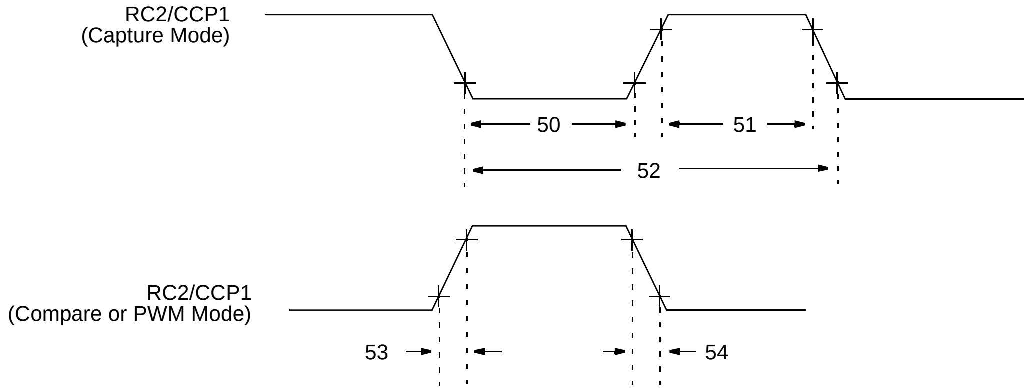
TABLE 14-4: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS

Param No.	Symbol	Characteristic			Min	Typ†	Max	Units	Conditions
40*	Tt0H	T0CKI High Pulse Width		No Prescaler	0.5 TCY + 20	—	—	ns	Must also meet parameter 42
				With Prescaler	10	—	—	ns	
41*	Tt0L	T0CKI Low Pulse Width		No Prescaler	0.5 TCY + 20	—	—	ns	Must also meet parameter 42
				With Prescaler	10	—	—	ns	
42*	Tt0P	T0CKI Period		No Prescaler	TCY + 40	—	—	ns	
				With Prescaler	Greater of: 20 or $\frac{T_{CY} + 40}{N}$	—	—	ns	N = prescale value (2, 4, ..., 256)
45*	Tt1H	T1CKI High Time	Synchronous, Prescaler = 1		0.5 TCY + 20	—	—	ns	Must also meet parameter 47
			Synchronous, Prescaler = 2,4,8	Standard(**F**)	15	—	—	ns	
				Extended(**LF**)	25	—	—	ns	
			Asynchronous	Standard(**F**)	30	—	—	ns	
				Extended(**LF**)	50	—	—	ns	
46*	Tt1L	T1CKI Low Time	Synchronous, Prescaler = 1		0.5 TCY + 20	—	—	ns	Must also meet parameter 47
			Synchronous, Prescaler = 2,4,8	Standard(**F**)	15	—	—	ns	
				Extended(**LF**)	25	—	—	ns	
			Asynchronous	Standard(**F**)	30	—	—	ns	
				Extended(**LF**)	50	—	—	ns	
47*	Tt1P	T1CKI Input Period	Synchronous	Standard(**F**)	Greater of: 30 or $\frac{T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1, 2, 4, 8)
				Extended(**LF**)	Greater of: 50 or $\frac{T_{CY} + 40}{N}$				N = prescale value (1, 2, 4, 8)
			Asynchronous	Standard(**F**)	60	—	—	ns	
				Extended(**LF**)	100	—	—	ns	
	Ft1	Timer1 Oscillator Input Frequency Range (oscillator enabled by setting bit T1OSCEN)			DC	—	200	kHz	
48	TCKEZtmr1	Delay from External Clock Edge to Timer Increment			2 TOSC	—	7 TOSC	—	

* 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.

FIGURE 14-9: CAPTURE/COMPARE/PWM TIMINGS (CCP1)



Note: Refer to Figure 14-3 for load conditions.

TABLE 14-5: CAPTURE/COMPARE/PWM REQUIREMENTS (CCP1)

Param No.	Symbol	Characteristic			Min	Typ†	Max	Units	Conditions
50*	TccL	CCP1 input low time	No Prescaler		0.5 TCY + 20	—	—	ns	
			With Prescaler	Standard(**F**)	10	—	—	ns	
				Extended(**LF**)	20	—	—	ns	
51*	TccH	CCP1 input high time	No Prescaler		0.5 TCY + 20	—	—	ns	
			With Prescaler	Standard(**F**)	10	—	—	ns	
				Extended(**LF**)	20	—	—	ns	
52*	TccP	CCP1 input period			$\frac{3\,T_{CY} + 40}{N}$	—	—	ns	N = prescale value (1,4 or 16)
53*	TccR	CCP1 output rise time		Standard(**F**)	—	10	25	ns	
				Extended(**LF**)	—	25	50	ns	
54*	TccF	CCP1 output fall time		Standard(**F**)	—	10	25	ns	
				Extended(**LF**)	—	25	45	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.

FIGURE 15-15: TYPICAL, MINIMUM AND MAXIMUM V_{OH} vs. I_{OH} (V_{DD} = 5V, -40°C TO +125°C)

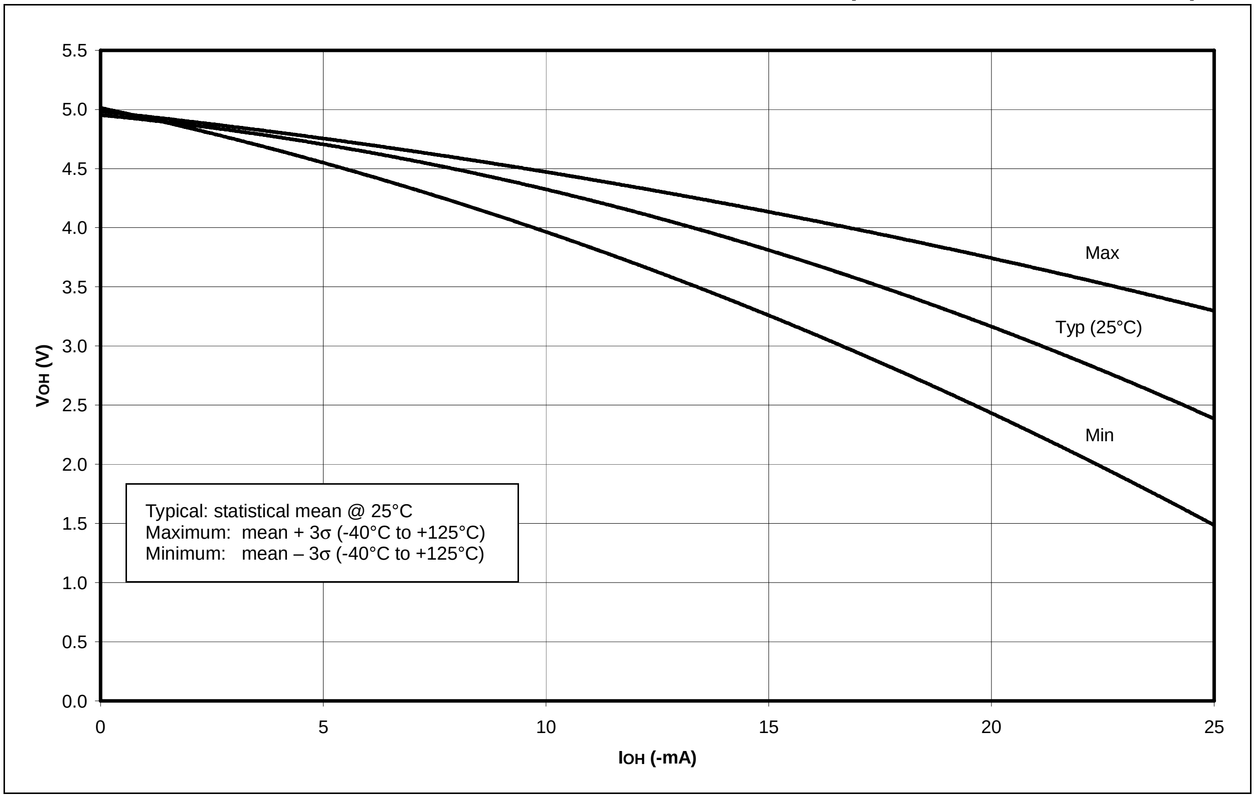
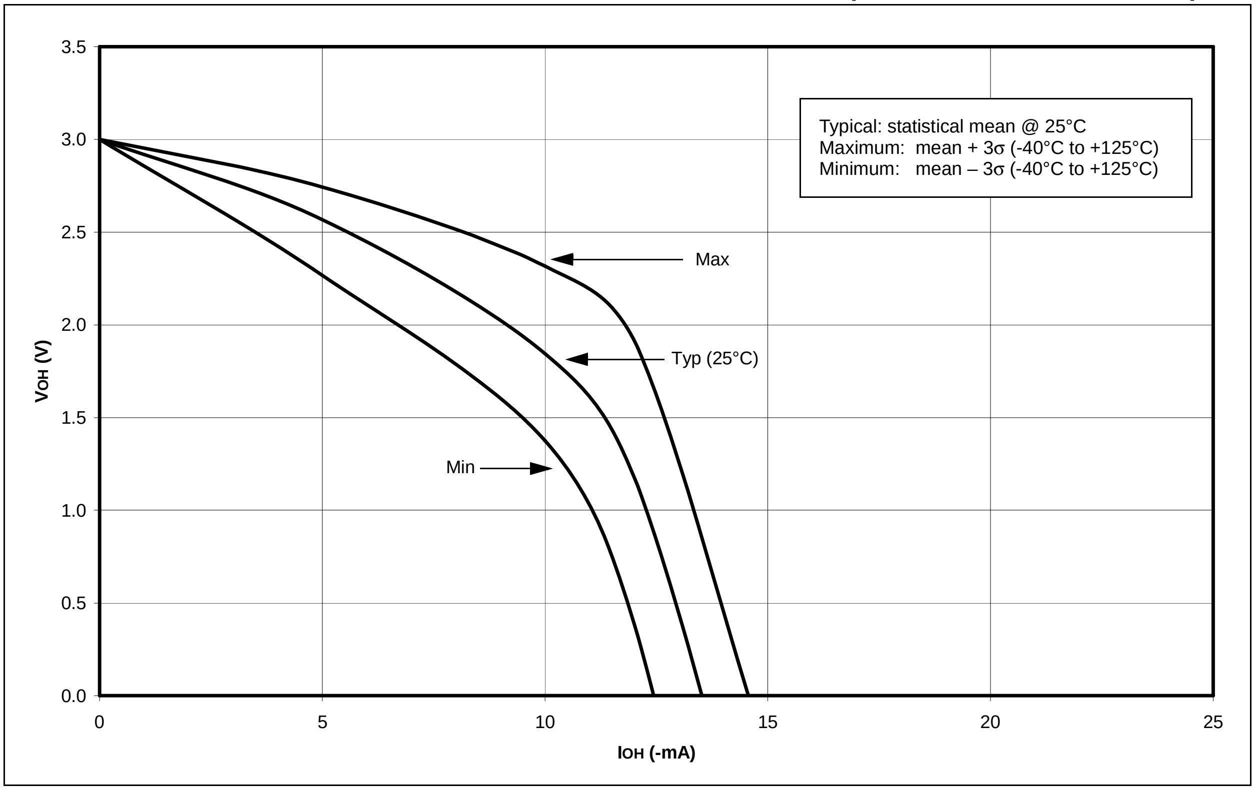
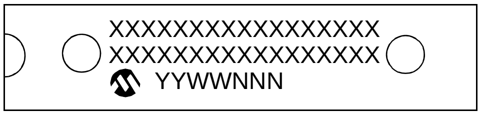


FIGURE 15-16: TYPICAL, MINIMUM AND MAXIMUM V_{OH} vs. I_{OH} (V_{DD} = 3V, -40°C TO +125°C)

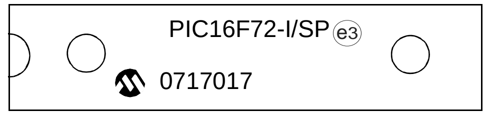


16.0 PACKAGE MARKING INFORMATION

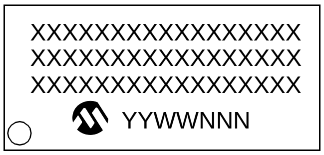
28-Lead PDIP (Skinny DIP)



Example



28-Lead SOIC

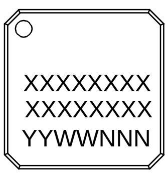


Example

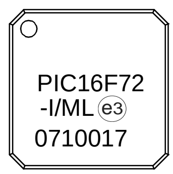
28-Lead SSOP

Example

28-Lead QFN



Example



Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week '01')
	NNN	Alphanumeric traceability code
	(e3)	Pb-free JEDEC designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator ((e3)) can be found on the outer packaging for this package.

Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

APPENDIX A: REVISION HISTORY

Revision A (April 2002)

This is a new data sheet. However, this device is similar to the PIC16C72 device found in the PIC16C7X Data Sheet (DS30390), the PIC16C72A Data Sheet (DS35008) or the PIC16F872 device (DS30221).

Revision B (May 2002)

Final data sheet. Includes device characterization data. Minor typographic revisions throughout.

Revision C (January 2007)

This revision includes updates to the packaging diagrams.

APPENDIX B: CONVERSION CONSIDERATIONS

Considerations for converting from previous versions of devices to the ones listed in this data sheet are listed in Table B-1.

TABLE B-1: CONVERSION CONSIDERATIONS

Characteristic	PIC16C72/72A	PIC16F872	PIC16F72
Pins	28	28	28
Timers	3	3	3
Interrupts	8	10	8
Communication	Basic SSP/SSP (SPI, I^2C Slave)	MSSP (SPI, I^2C Master/Slave)	SSP (SPI, I^2C Slave)
Frequency	20 MHz	20 MHz	20 MHz
A/D	8-bit, 5 Channels	10-bit, 5 Channels	8-bit, 5 Channels
CCP	1	1	1
Program Memory	2K EPROM	2K FLASH (1,000 E/W cycles)	2K FLASH (1000 E/W cycles)
RAM	128 bytes	128 bytes	128 bytes
EEPROM Data	None	64 bytes	None
Other	—	In-Circuit Debugger, Low Voltage Programming	—

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, 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
- Development Systems Information Line

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://support.microchip.com