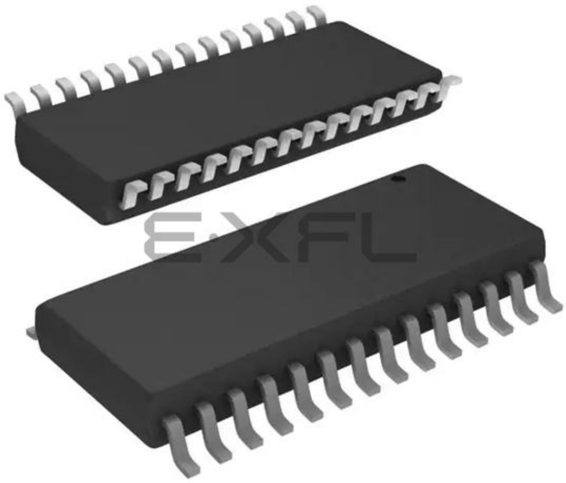


Welcome to E-XFL.COM

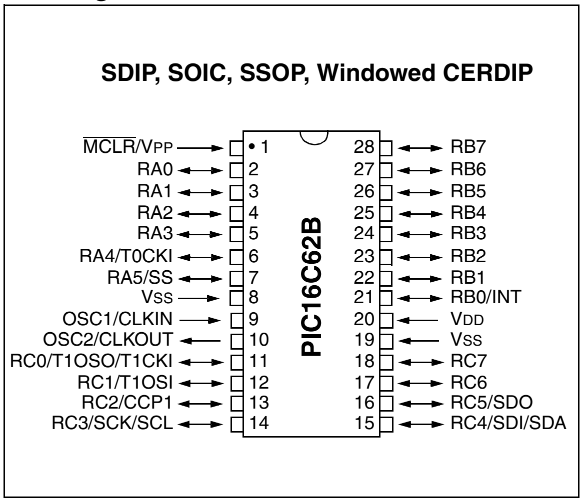
What is "Embedded - Microcontrollers"?

"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
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	22
Program Memory Size	3.5KB (2K x 14)
Program Memory Type	OTP
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 ~ 125°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/pic16c72a-04e-so

Pin Diagrams



Key Features PIC® Mid-Range Reference Manual (DS33023)	PIC16C62B	PIC16C72A
Operating Frequency	DC - 20 MHz	DC - 20 MHz
Resets (and Delays)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
Program Memory (14-bit words)	2K	2K
Data Memory (bytes)	128	128
Interrupts	7	8
I/O Ports	Ports A,B,C	Ports A,B,C
Timers	3	3
Capture/Compare/PWM modules	1	1
Serial Communications	SSP	SSP
8-bit Analog-to-Digital Module	—	5 input channels

 Preliminary

2.2 Data Memory Organization

The data memory is partitioned into multiple banks which contain the General Purpose Registers and the Special Function Registers. Bits RP1 and RP0 are the bank select bits.

RP1(1)	RP0	(STATUS<6:5>)

= 00 → Bank0
= 01 → Bank1
= 10 → Bank2 (not implemented)
= 11 → Bank3 (not implemented)

Note 1: Maintain this bit clear to ensure upward compatibility with future products.

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 Special Function Registers. Some "high use" Special Function Registers from one bank may be mirrored in another bank for code reduction and quicker access.

2.2.1 GENERAL PURPOSE REGISTER FILE

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

FIGURE 2-2: REGISTER FILE MAP

File Address	Bank 0	Bank 1	File Address
00h	INDF(1)	INDF(1)	80h
01h	TMR0	OPTION_REG	81h
02h	PCL	PCL	82h
03h	STATUS	STATUS	83h
04h	FSR	FSR	84h
05h	PORTA	TRISA	85h
06h	PORTB	TRISB	86h
07h	PORTC	TRISC	87h
08h	—	—	88h
09h	—	—	89h
0Ah	PCLATH	PCLATH	8Ah
0Bh	INTCON	INTCON	8Bh
0Ch	PIR1	PIE1	8Ch
0Dh	—	—	8Dh
0Eh	TMR1L	PCON	8Eh
0Fh	TMR1H	—	8Fh
10h	T1CON	—	90h
11h	TMR2	—	91h
12h	T2CON	PR2	92h
13h	SSPBUF	SSPADD	93h
14h	SSPCON	SSPSTAT	94h
15h	CCPR1L	—	95h
16h	CCPR1H	—	96h
17h	CCP1CON	—	97h
18h	—	—	98h
19h	—	—	99h
1Ah	—	—	9Ah
1Bh	—	—	9Bh
1Ch	—	—	9Ch
1Dh	—	—	9Dh
1Eh	ADRES(2)	—	9Eh
1Fh	ADCON0(2)	ADCON1(2)	9Fh
20h–7Fh	General Purpose Registers	General Purpose Registers	A0h–BFh
		—	C0h–FFh

Unimplemented data memory locations, read as '0'.

Note 1: Not a physical register.
2: These registers are not implemented on the PIC16C62B, read as '0'.

2.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers are registers used by the CPU and Peripheral Modules for controlling the desired operation of the device. These registers are implemented as static RAM. A list of these registers is given in Table 2-1.

The Special Function Registers can be classified into two sets; core (CPU) and peripheral. Those registers associated with the core functions are described in detail in this section. Those related to the operation of the peripheral features are described in detail in the peripheral feature section.

TABLE 2-1 SPECIAL FUNCTION REGISTER SUMMARY

Addr	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 (4)
Bank 0											
00h	INDF(1)	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	0000 0000
01h	TMR0	Timer0 module's register								xxxx xxxx	uuuu uuuu
02h	PCL(1)	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
03h	STATUS(1)	IRP(5)	RP1(5)	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	0001 1xxx	000q quuu
04h	FSR(1)	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
05h	PORTA(6,7)	—	—	PORTA Data Latch when written: PORTA pins when read						--0x 0000	--0u 0000
06h	PORTB(6,7)	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	uuuu uuuu
07h	PORTC(6,7)	PORTC Data Latch when written: PORTC pins when read								xxxx xxxx	uuuu uuuu
08h-09h	—	Unimplemented								—	—
0Ah	PCLATH(1,2)	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
0Bh	INTCON(1)	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	—	ADIF(3)	—	—	SSPIF	CCP1IF	TMR2IF	TMR1IF	-0-- 0000	-0-- 0000
0Dh	—	Unimplemented								—	—
0Eh	TMR1L	Holding register for the Least Significant Byte of the 16-bit TMR1 register								xxxx xxxx	uuuu uuuu
0Fh	TMR1H	Holding register for the Most Significant Byte of the 16-bit TMR1 register								xxxx xxxx	uuuu uuuu
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	--00 0000	--uu uuuu
11h	TMR2	Timer2 module's register								0000 0000	0000 0000
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
15h	CCPR1L	Capture/Compare/PWM Register1 (LSB)								xxxx xxxx	uuuu uuuu
16h	CCPR1H	Capture/Compare/PWM Register1 (MSB)								xxxx xxxx	uuuu uuuu
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
18h-1Dh	—	Unimplemented								—	—
1Eh	ADRES(3)	A/D Result Register								xxxx xxxx	uuuu uuuu
1Fh	ADCON0(3)	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	—	ADON	0000 00-0	0000 00-0

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', Shaded locations are unimplemented, read as '0'.

Note 1: These registers can be addressed from either bank.

2: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for PC<12:8> whose contents are transferred to the upper byte of the program counter.

3: A/D not implemented on the PIC16C62B, maintain as '0'.

4: Other (non power-up) resets include: external reset through $\overline{\text{MCLR}}$ and the Watchdog Timer Reset.

5: The IRP and RP1 bits are reserved. Always maintain these bits clear.

6: On any device reset, these pins are configured as inputs.

7: This is the value that will be in the port output latch.

TABLE 3-5 PORTC FUNCTIONS

Name	Bit#	Buffer Type	Function	TRISC Override
RC0/T1OSO/T1CKI	bit0	ST	Input/output port pin or Timer1 oscillator output/Timer1 clock input	Yes
RC1/T1OSI	bit1	ST	Input/output port pin or Timer1 oscillator input	Yes
RC2/CCP1	bit2	ST	Input/output port pin or Capture1 input/Compare1 output/PWM1 output	No
RC3/SCK/SCL	bit3	ST	RC3 can also be the synchronous serial clock for both SPI and I^2C modes.	No
RC4/SDI/SDA	bit4	ST	RC4 can also be the SPI Data In (SPI mode) or data I/O (I^2C mode).	No
RC5/SDO	bit5	ST	Input/output port pin or Synchronous Serial Port data output	No
RC6	bit6	ST	Input/output port pin	No
RC7	bit7	ST	Input/output port pin	No

Legend: ST = Schmitt Trigger input

TABLE 3-6 SUMMARY OF REGISTERS ASSOCIATED WITH PORTC

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
07h	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	uuuu uuuu
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111

Legend: x = unknown, u = unchanged.

6.0 TIMER2 MODULE

The Timer2 module timer has the following features:

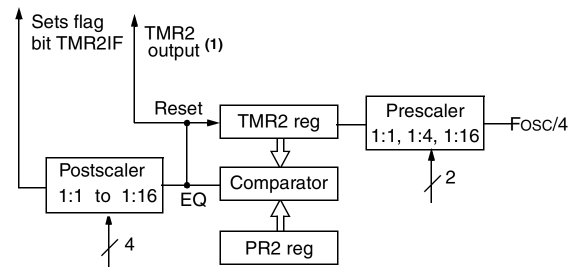
- 8-bit timer (TMR2 register)
  - Readable and writable
- 8-bit period register (PR2)
  - Readable and writable
- Software programmable prescaler (1:1, 1:4, 1:16)
- Software programmable postscaler (1:1 to 1:16)
- Interrupt on match (TMR2 = PR2)
- Timer2 can be used by SSP and CCP

Timer2 has a control register, shown in Register 6-1. Timer2 can be shut off by clearing control bit TMR2ON (T2CON<2>) to minimize power consumption.

Figure 6-1 is a simplified block diagram of the Timer2 module.

Additional information on timer modules is available in the PIC® MCU Mid-Range Reference Manual, (DS33023).

FIGURE 6-1: TIMER2 BLOCK DIAGRAM



Note 1: TMR2 register output can be software selected by the SSP Module as a baud clock.

REGISTER 6-1:T2CON: TIMER2 CONTROL REGISTER (ADDRESS 12h)

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit7							bit0

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

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

bit 6-3: **TOUTPS3:TOUTPS0**: Timer2 Output Postscale Select bits
`0000` = 1:1 Postscale
`0001` = 1:2 Postscale
`0010` = 1:3 Postscale
•
•
•
`1111` = 1:16 Postscale

bit 2: **TMR2ON**: Timer2 On bit
1 = Timer2 is on
0 = Timer2 is off

bit 1-0: **T2CKPS1:T2CKPS0**: Timer2 Clock Prescale Select bits
`00` = Prescaler is 1
`01` = Prescaler is 4
`1x` = Prescaler is 16

6.1 Timer2 Operation

The Timer2 output is also used by the CCP module to generate the PWM "On-Time", and the PWM period with a match with PR2.

The TMR2 register is readable and writable, and is cleared on any device reset.

The input clock (FOSC/4) has a prescale option of 1:1, 1:4 or 1:16, selected by control bits T2CKPS1:T2CKPS0 (T2CON<1:0>).

The match output of TMR2 goes through a 4-bit postscaler (which gives a 1:1 to 1:16 scaling) to generate a TMR2 interrupt (latched in flag bit TMR2IF, (PIR1<1>)).

The prescaler and postscaler counters are cleared when any of the following occurs:

- a write to the TMR2 register
- a write to the T2CON register
- any device reset (Power-on Reset, $\overline{MCLR}$ reset, Watchdog Timer reset or Brown-out Reset)

TMR2 is not cleared when T2CON is written.

6.2 Timer2 Interrupt

The Timer2 module has an 8-bit period register PR2. Timer2 increments from 00h until it matches PR2 and then resets to 00h on the next increment cycle. PR2 is a readable and writable register. The PR2 register is initialized to FFh upon reset.

6.3 Output of TMR2

The output of TMR2 (before the postscaler) is fed to the Synchronous Serial Port module, which optionally uses it to generate shift clock.

TABLE 6-1 REGISTERS ASSOCIATED WITH TIMER2 AS A TIMER/COUNTER

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other resets
0Bh,8Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	—	ADIF	—	—	SSPIF	CCP1IF	TMR2IF	TMR1IF	-00- 0000	0000 0000
8Ch	PIE1	—	ADIE	—	—	SSPIE	CCP1IE	TMR2IE	TMR1IE	-0-- 0000	0000 0000
11h	TMR2	Timer2 module's register								0000 0000	0000 0000
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
92h	PR2	Timer2 Period Register								1111 1111	1111 1111

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used by the Timer2 module.

7.0 CAPTURE/COMPARE/PWM (CCP) MODULE

The 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 PWM master/slave duty cycle register. Table 7-1 shows the timer resources of the CCP module modes.

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. All are readable and writable.

Additional information on the CCP module is available in the PIC® MCU Mid-Range Reference Manual, (DS33023).

TABLE 7-1 CCP MODE - TIMER RESOURCE

CCP Mode	Timer Resource
Capture	Timer1
Compare	Timer1
PWM	Timer2

TABLE 7-2 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.

REGISTER 7-1: CCP1CON REGISTER (ADDRESS 17h)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0
bit7							bit0

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

bit 7-6: **Unimplemented:** Read as '0'

bit 5-4: **CCP1X:CCP1Y**: PWM Least Significant bits
Capture Mode: Unused
Compare Mode: Unused
PWM Mode: These bits are the two LSbs of the PWM duty cycle. The eight MSbs are found in CCPR1L.

bit 3-0: **CCP1M3:CCP1M0**: CCP1 Mode Select bits
`0000` = Capture/Compare/PWM off (resets CCP1 module)
`0100` = Capture mode, every falling edge
`0101` = Capture mode, every rising edge
`0110` = Capture mode, every 4th rising edge
`0111` = Capture mode, every 16th rising edge
`1000` = Compare mode, set output on match (CCP1IF bit is set)
`1001` = Compare mode, clear output on match (CCP1IF bit is set)
`1010` = Compare mode, generate software interrupt on match (CCP1IF bit is set, CCP1 pin is unaffected)
`1011` = Compare mode, trigger special event (CCP1IF bit is set; CCP1 resets TMR1 and starts an A/D conversion (if A/D module is enabled))
`11xx` = PWM mode

8.0 SYNCHRONOUS SERIAL PORT (SSP) MODULE

8.1 SSP Module Overview

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

- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I^2C)

For more information on SSP operation (including an I^2C Overview), refer to the PIC® MCU Mid-Range Reference Manual, (DS33023). Also, refer to Application Note AN578, *"Use of the SSP Module in the I^2C Multi-Master Environment."*

8.2 SPI Mode

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

Additional information on SPI operation may be found in the PIC® MCU Mid-Range Reference Manual, (DS33023).

8.2.1 OPERATION OF SSP MODULE IN SPI MODE

A block diagram of the SSP Module in SPI Mode is shown in Figure 8-1.

The SPI mode allows 8-bits of data to be synchronously transmitted and received simultaneously. To accomplish communication, 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/$\overline{SS}$/AN4

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

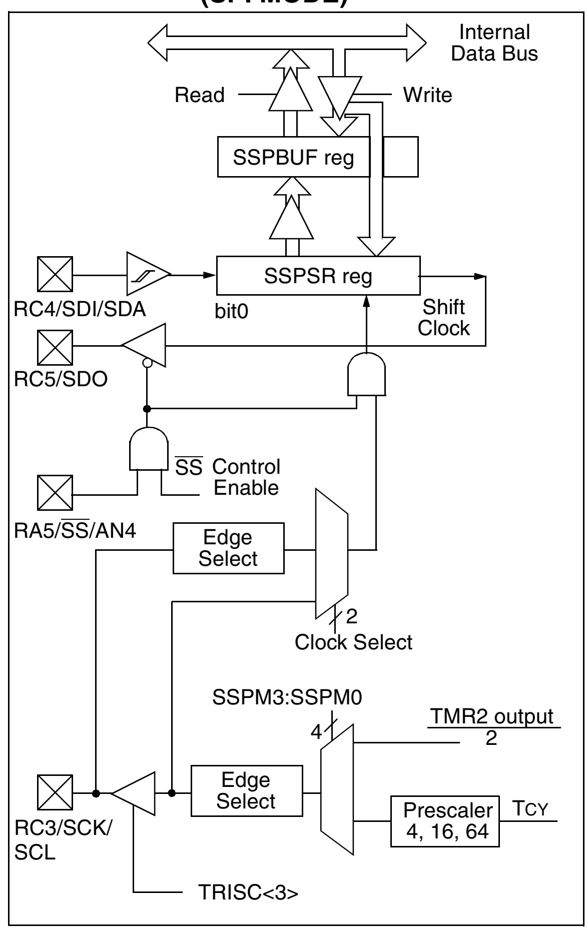
- Master Operation (SCK is the clock output)
- Slave Mode (SCK is the clock input)
- Clock Polarity (Idle state of SCK)
- Clock Edge (Output data on rising/falling edge of SCK)
- Clock Rate (master operation only)
- Slave Select Mode (Slave mode only)

To enable the serial port, SSP Enable bit, SSPEN (SSPCON<5>) must be set. To reset or reconfigure SPI mode, clear bit SSPEN, re-initialize the SSPCON register, and then set bit SSPEN. This configures the SDI, SDO, SCK and $\overline{SS}$ pins as serial port pins. For the pins to behave as the serial port function, they must have their data direction bits (in the TRISC register) appropriately programmed. That is:

- SDI must have TRISC<4> set
- SDO must have TRISC<5> cleared
- SCK (master operation) must have TRISC<3> cleared
- SCK (Slave mode) must have TRISC<3> set
- $\overline{SS}$ must have TRISA<5> set (if used)

Note:	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:	If the SPI is used in Slave Mode with CKE = '1', then the $\overline{SS}$ pin control must be enabled.

FIGURE 8-1: SSP BLOCK DIAGRAM (SPI MODE)



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

Note:	This section applies to the PIC16C72A only.

The analog-to-digital (A/D) converter module has five input channels.

The A/D allows conversion of an analog input signal to a corresponding 8-bit digital number (refer to Application Note AN546 for use of A/D Converter). 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 the 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.

Additional information on the A/D module is available in the PIC® MCU Mid-Range Reference Manual, (DS33023).

The A/D module has three registers. These registers are:

- 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 Figure 9-1, controls the operation of the A/D module. The ADCON1 register, shown in Figure 9-2, configures the functions of the port pins. The port pins can be configured as analog inputs (RA3 can also be a voltage reference) or as digital I/O.

REGISTER 9-1: ADCON0 REGISTER (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
bit7							bit0

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

bit 7-6: **ADCS1:ADCS0:** A/D Conversion Clock Select bits
`00` = FOSC/2
`01` = FOSC/8
`10` = FOSC/32
`11` = FRC (clock derived from an internal RC oscillator)

bit 5-3: **CHS2:CHS0**: 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
If ADON = 1
`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 shutoff and consumes no operating current

10.8 Time-out Sequence

When a POR reset occurs, the PWRT delay starts (if enabled). When PWRT ends, the OST counts 1024 oscillator cycles (LP, XT, HS modes only). When OST completes, the device comes out of reset. The total time-out will vary based on oscillator configuration and the status of the PWRT. For example, in RC mode with the PWRT disabled, there will be no time-out at all.

If $\overline{\text{MCLR}}$ is kept low long enough, the time-outs will expire. Bringing $\overline{\text{MCLR}}$ high will begin execution immediately. This is useful for testing purposes or to synchronize more than one PIC16CXXX device operating in parallel.

Table 10-5 shows the reset conditions for the STATUS, PCON and PC registers, while Table 10-6 shows the reset conditions for all the registers.

10.9 Power Control/Status Register (PCON)

The $\overline{\text{BOR}}$ bit is unknown on Power-on Reset. If the Brown-out Reset circuit is used, the $\overline{\text{BOR}}$ bit must be set by the user and checked on subsequent resets to see if it was cleared, indicating a Brown-out has occurred.

$\overline{\text{POR}}$ (Power-on Reset Status bit) is cleared on a Power-on Reset and unaffected otherwise. The user

Status Register

IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C

PCON Register

						$\overline{\text{POR}}$	$\overline{\text{BOR}}$

TABLE 10-3 TIME-OUT IN VARIOUS SITUATIONS

Oscillator Configuration	Power-up		Brown-out	Wake-up from SLEEP
	$\overline{\text{PWRTE}}$ = 0	$\overline{\text{PWRTE}}$ = 1		
XT, HS, LP	72 ms + 1024TOSC	1024TOSC	72 ms + 1024TOSC	1024TOSC
RC	72 ms	—	72 ms	—

TABLE 10-4 STATUS BITS AND THEIR SIGNIFICANCE

$\overline{\text{POR}}$	$\overline{\text{BOR}}$	$\overline{\text{TO}}$	$\overline{\text{PD}}$	
0	x	1	1	Power-on Reset
0	x	0	x	Illegal, $\overline{\text{TO}}$ is set on $\overline{\text{POR}}$
0	x	x	0	Illegal, $\overline{\text{PD}}$ is set on $\overline{\text{POR}}$
1	0	1	1	Brown-out Reset
1	1	0	1	WDT Reset
1	1	0	0	WDT Wake-up
1	1	u	u	$\overline{\text{MCLR}}$ Reset during normal operation
1	1	1	0	$\overline{\text{MCLR}}$ Reset during SLEEP or interrupt wake-up from SLEEP

TABLE 10-5 RESET CONDITION FOR SPECIAL REGISTERS

Condition	Program Counter	STATUS Register	PCON Register
Power-on Reset	000h	0001 1xxx	---- --0x
$\overline{\text{MCLR}}$ Reset during normal operation	000h	000u uuuu	---- --uu
$\overline{\text{MCLR}}$ Reset during SLEEP	000h	0001 0uuu	---- --uu
WDT Reset	000h	0000 1uuu	---- --uu
WDT Wake-up	PC + 1	uuu0 0uuu	---- --uu
Brown-out Reset	000h	0001 1uuu	---- --u0
Interrupt wake-up from SLEEP	PC + 1$^{(1)}$	uuu1 0uuu	---- --uu

Legend: u = unchanged, x = unknown, - = unimplemented bit read as '0'.

Note 1: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

10.10 Interrupts

The interrupt control register (INTCON) records individual interrupt requests in flag bits. It also has individual and global interrupt enable bits.

Note:	Individual interrupt flag bits are set regardless of the status of their corresponding mask bit or the GIE bit.

A global interrupt enable bit, GIE (INTCON<7>) enables or disables all interrupts. When bit GIE is enabled, and an interrupt's flag bit and mask bit are set, the interrupt will vector immediately. Individual interrupts can be disabled through their corresponding enable bits in various registers. Individual interrupt flag bits are set regardless of the status of the GIE bit. The GIE bit is cleared on reset.

The "return from interrupt" instruction, `RETFIE`, exits the interrupt routine and sets the GIE bit, which re-enables interrupts.

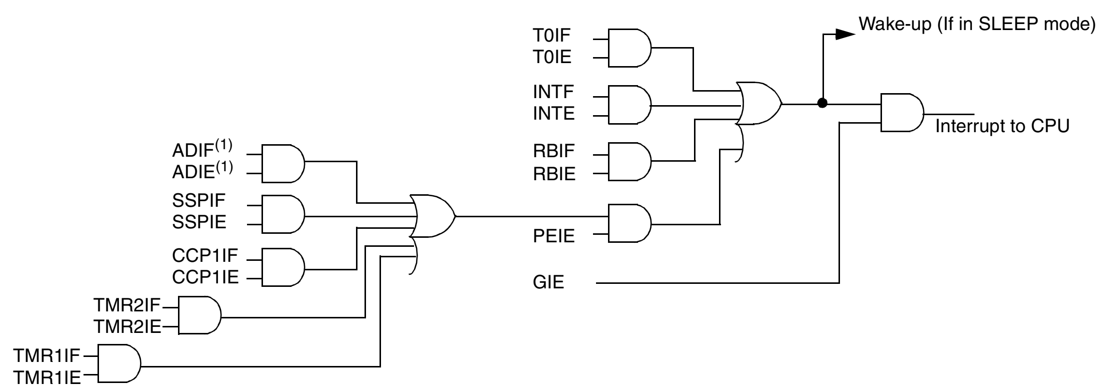
The RB0/INT pin interrupt, the RB port change interrupt and the TMR0 overflow interrupt flags are contained in the INTCON register.

The peripheral interrupt flags are contained in the special function registers PIR1 and PIR2. The corresponding interrupt enable bits are contained in special function registers PIE1 and PIE2, and the peripheral interrupt enable bit is contained in special function register INTCON.

When an interrupt is responded to, the GIE bit is cleared to disable any further interrupts, the return address is pushed onto the stack and the PC is loaded with 0004h. Once in the interrupt service routine, the source of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bit must be cleared in software before re-enabling interrupts to avoid recursive interrupts.

For external interrupt events, such as the INT pin or PORTB change interrupt, the interrupt latency will be three or four instruction cycles, depending on when the interrupt event occurs. The latency is the same for one or two cycle instructions. Individual interrupt flag bits are set regardless of the status of their corresponding mask bit or the GIE bit

FIGURE 10-7: INTERRUPT LOGIC



Note 1: The A/D module is not implemented on the PIC16C62B.

10.13 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 be considered.

The $\overline{\text{MCLR}}$ pin must be at a logic high level (V_{IHMC}, parameter D042).

10.13.1 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 some Peripheral Interrupts.

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 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. CCP1 is in compare mode).
4. SSP (Start/Stop) bit detect interrupt.
5. SSP transmit or receive in slave mode (SPI/I^2C).
6. USART RX or TX (synchronous slave mode).

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 is regardless of the state of the GIE bit. If the GIE bit is clear (disabled), the device resumes 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, a `NOP` should follow the `SLEEP` instruction.

10.13.2 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.

TABLE 11-2 PIC16CXXX INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode MSb			LSb	Status Affected	Notes
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	**f, d**	Add W and f	1	00	0111	dfff	ffff	C,DC,Z	1,2
ANDWF	**f, d**	AND W with f	1	00	0101	dfff	ffff	Z	1,2
CLRF	**f**	Clear f	1	00	0001	lfff	ffff	Z	2
CLRW	**-**	Clear W	1	00	0001	0000	0011	Z	
COMF	**f, d**	Complement f	1	00	1001	dfff	ffff	Z	1,2
DECF	**f, d**	Decrement f	1	00	0011	dfff	ffff	Z	1,2
DECFSZ	**f, d**	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1,2,3
INCF	**f, d**	Increment f	1	00	1010	dfff	ffff	Z	1,2
INCFSZ	**f, d**	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1,2,3
IORWF	**f, d**	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	1,2
MOVF	**f, d**	Move f	1	00	1000	dfff	ffff	Z	1,2
MOVWF	**f**	Move W to f	1	00	0000	lfff	ffff		
NOP	**-**	No Operation	1	00	0000	0xx0	0000		
RLF	**f, d**	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	1,2
RRF	**f, d**	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	1,2
SUBWF	**f, d**	Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z	1,2
SWAPF	**f, d**	Swap nibbles in f	1	00	1110	dfff	ffff		1,2
XORWF	**f, d**	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	**f, b**	Bit Clear f	1	01	00bb	bfff	ffff		1,2
BSF	**f, b**	Bit Set f	1	01	01bb	bfff	ffff		1,2
BTFSC	**f, b**	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff		3
BTFSS	**f, b**	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		3
LITERAL AND CONTROL OPERATIONS									
ADDLW	**k**	Add literal and W	1	11	111x	kkkk	kkkk	C,DC,Z	
ANDLW	**k**	AND literal with W	1	11	1001	kkkk	kkkk	Z	
CALL	**k**	Call subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDT	**-**	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}$,$\overline{PD}$	
GOTO	**k**	Go to address	2	10	1kkk	kkkk	kkkk		
IORLW	**k**	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLW	**k**	Move literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	**-**	Return from interrupt	2	00	0000	0000	1001		
RETLW	**k**	Return with literal in W	2	11	01xx	kkkk	kkkk		
RETURN	**-**	Return from Subroutine	2	00	0000	0000	1000		
SLEEP	**-**	Go into standby mode	1	00	0000	0110	0011	$\overline{TO}$,$\overline{PD}$	
SUBLW	**k**	Subtract W from literal	1	11	110x	kkkk	kkkk	C,DC,Z	
XORLW	**k**	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

Note 1: When an I/O register is modified as a function of itself (e.g., `MOVF PORTB, 1`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

2: If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 Module.

3: If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

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

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

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

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

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

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

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 is 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'.

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 is moved to a destination dependant upon the status of d. If d = 0, 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.

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.

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'.

NOP — No Operation

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

12.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers are supported with a full range of hardware and software development tools:

- Integrated Development Environment
  - MPLAB™ IDE Software
- Assemblers/Compilers/Linkers
  - MPASM Assembler
  - MPLAB-C17 and MPLAB-C18 C Compilers
  - MPLINK/MPLIB Linker/Librarian
- Simulators
  - MPLAB-SIM Software Simulator
- Emulators
  - MPLAB-ICE Real-Time In-Circuit Emulator
  - PICMASTER®/PICMASTER-CE In-Circuit Emulator
  - ICEPIC™
- In-Circuit Debugger
  - MPLAB-ICD for PIC16F877
- Device Programmers
  - PRO MATE® II Universal Programmer
  - PICSTART® Plus Entry-Level Prototype Programmer
- Low-Cost Demonstration Boards
  - SIMICE
  - PICDEM-1
  - PICDEM-2
  - PICDEM-3
  - PICDEM-17
  - SEEVAL®
  - KEELOQ®

12.1 MPLAB Integrated Development Environment Software

- The MPLAB IDE software brings an ease of software development previously unseen in the 8-bit microcontroller market. MPLAB is a Windows®-based application which contains:

- Multiple functionality
  - editor
  - simulator
  - programmer (sold separately)
  - emulator (sold separately)
- A full featured editor
- A project manager
- Customizable tool bar and key mapping
- A status bar
- On-line help

MPLAB allows you to:

- Edit your source files (either assembly or ‘C’)
- One touch assemble (or compile) and download to PIC MCU tools (automatically updates all project information)
- Debug using:
  - source files
  - absolute listing file
  - object code

The ability to use MPLAB with Microchip’s simulator, MPLAB-SIM, allows a consistent platform and the ability to easily switch from the cost-effective simulator to the full featured emulator with minimal retraining.

12.2 MPASM Assembler

MPASM is a full featured universal macro assembler for all PIC MCUs. It can produce absolute code directly in the form of HEX files for device programmers, or it can generate relocatable objects for MPLINK.

MPASM has a command line interface and a Windows shell and can be used as a standalone application on a Windows 3.x or greater system. MPASM generates relocatable object files, Intel standard HEX files, MAP files to detail memory usage and symbol reference, an absolute LST file which contains source lines and generated machine code, and a COD file for MPLAB debugging.

MPASM features include:

- MPASM and MPLINK are integrated into MPLAB projects.
- MPASM allows user defined macros to be created for streamlined assembly.
- MPASM allows conditional assembly for multi purpose source files.
- MPASM directives allow complete control over the assembly process.

12.3 MPLAB-C17 and MPLAB-C18 C Compilers

The MPLAB-C17 and MPLAB-C18 Code Development Systems are complete ANSI ‘C’ compilers and integrated development environments for Microchip’s PIC17CXXX and PIC18CXXX family of microcontrollers, respectively. These compilers provide powerful integration capabilities and ease of use not found with other compilers.

For easier source level debugging, the compilers provide symbol information that is compatible with the MPLAB IDE memory display.

12.4 MPLINK/MPLIB Linker/Librarian

MPLINK is a relocatable linker for MPASM and MPLAB-C17 and MPLAB-C18. It can link relocatable objects from assembly or C source files along with precompiled libraries using directives from a linker script.

NOTES:

13.2 DC Characteristics: PIC16LC62B/72A-04 (Commercial, Industrial)

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated) Operating temperature 0°C ≤ TA ≤ +70°C for commercial -40°C ≤ TA ≤ +85°C for industrial					
Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
D001	VDD	Supply Voltage	2.5 VBOR*	- -	5.5 5.5	V V	LP, XT, RC osc modes (DC - 4 MHz) BOR enabled (Note 7)
D002*	VDR	**RAM Data Retention Voltage** (Note 1)	-	1.5	-	V	
D003	VPOR	**VDD Start Voltage** to ensure internal Power-on Reset signal	-	VSS	-	V	See section on Power-on Reset for details
D004* D004A*	SVDD	**VDD Rise Rate** to ensure internal Power-on Reset signal	0.05 TBD	- -	- -	V/ms	PWRT enabled ($\overline{PWRTE}$ bit clear) PWRT disabled ($\overline{PWRTE}$ bit set) See section on Power-on Reset for details
D005	VBOR	Brown-out Reset voltage trip point	3.65	-	4.35	V	BODEN bit set
D010	IDD	**Supply Current** (Note 2, 5)	-	2.0	3.8	mA	XT, RC osc modes FOSC = 4 MHz, VDD = 3.0V (Note 4)
D010A			-	22.5	48	μA	LP OSC MODE FOSC = 32 kHz, VDD = 3.0V, WDT disabled
D020 D021 D021A	IPD	**Power-down Current** (Note 3, 5)	- - -	7.5 0.9 0.9	30 5 5	μA μA μA	VDD = 3.0V, WDT enabled, -40°C to +85°C VDD = 3.0V, WDT disabled, 0°C to +70°C VDD = 3.0V, WDT disabled, -40°C to +85°C
		Module Differential Current (Note 6)					
D022* D022A*	ΔIWDT ΔIBOR	Watchdog Timer Brown-out Reset	- -	6.0 TBD	20 200	μA μA	WDTE BIT SET, VDD = 4.0V BODEN bit set, 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 tristated, pulled to VDD,
$\overline{MCLR}$ = 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.

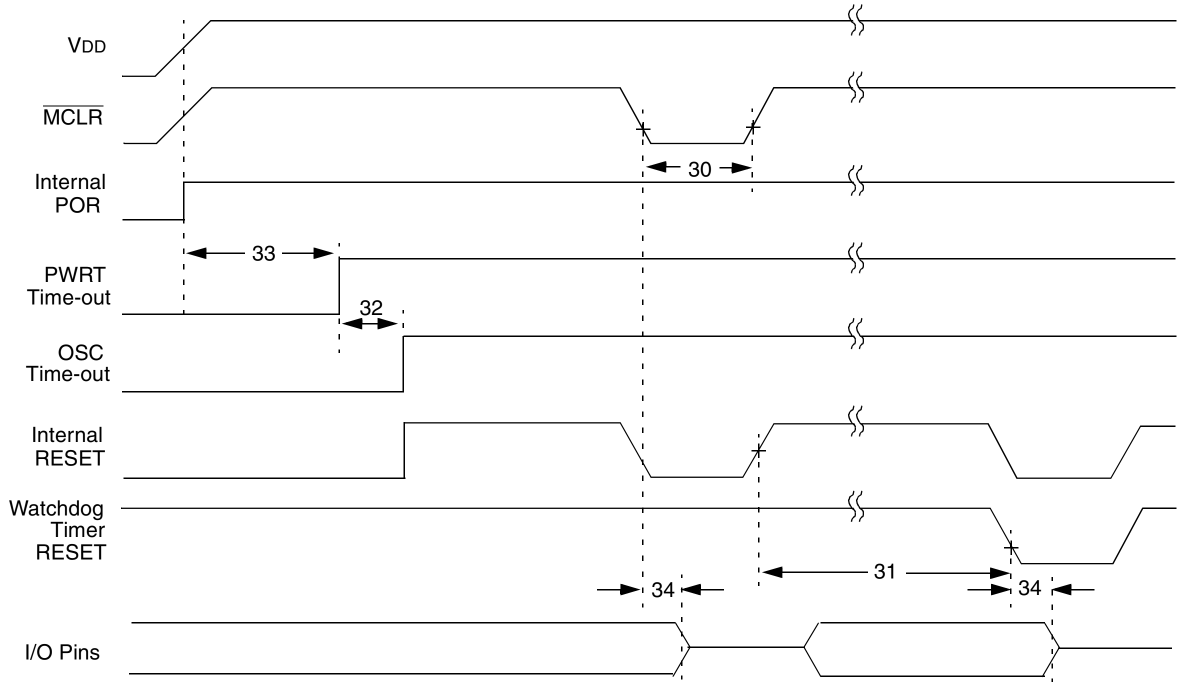
4: For RC osc mode, current through Rext is not included. The current through the resistor can be estimated by the formula Ir = VDD/2Rext (mA) with Rext in kOhm.

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: This is the voltage where the device enters the Brown-out Reset. When BOR is enabled, the device will perform a brown-out reset when VDD falls below VBOR.

FIGURE 13-7: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING



Note: Refer to Figure 13-4 for load conditions.

FIGURE 13-8: BROWN-OUT RESET TIMING

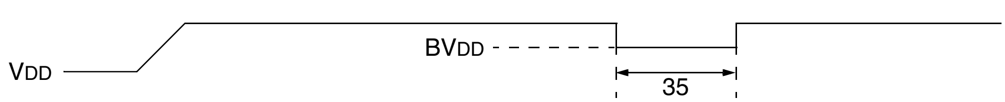


TABLE 13-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER AND BROWN-OUT RESET REQUIREMENTS

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
30	TmcL	$\overline{MCLR}$ Pulse Width (low)	2	—	—	μs	VDD = 5V, -40°C to +125°C
31*	Twdt	Watchdog Timer Time-out Period (No Prescaler)	7	18	33	ms	VDD = 5V, -40°C to +125°C
32	Tost	Oscillator Start-up Timer Period	—	1024 TOSC	—	—	TOSC = OSC1 period
33*	Tpwrt	Power-up Timer Period	28	72	132	ms	VDD = 5V, -40°C to +125°C
34	TIOZ	I/O Hi-impedance from $\overline{MCLR}$ Low or WDT reset	—	—	2.1	μs	
35	TBOR	Brown-out Reset Pulse Width	100	—	—	μs	VDD ≤ BVDD (D005)

* 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.

Q

R

S

 Preliminary