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	UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	16
Program Memory Size	1.75KB (1K x 14)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	224 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	18-SOIC (0.295", 7.50mm Width)
Supplier Device Package	18-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f627t-04i-so

NOTES:

NOTES:

2.0   ARCHITECTURAL OVERVIEW

The high performance of the PIC16F62X family can be attributed to a number of architectural features commonly found in RISC microprocessors. To begin with, the PIC16F62X uses a Harvard architecture, in which, program and data are accessed from separate memories using separate buses. This improves bandwidth over traditional Von Neumann architecture where program and data are fetched from the same memory. Separating program and data memory further allows instructions to be sized differently than 8-bit wide data word. Instruction opcodes are 14-bits wide making it possible to have all single-word instructions. A 14-bit wide program memory access bus fetches a 14-bit instruction in a single cycle. A two-stage pipeline overlaps fetch and execution of instructions. Consequently, all instructions (35) execute in a single cycle (200 ns @ 20 MHz) except for program branches.

The Table below lists program memory (FLASH, Data and EEPROM).

TABLE 2-1:   DEVICE DESCRIPTION

Device	Memory		
	FLASH Program	RAM Data	EEPROM Data
PIC16F627	1024 x 14	224 x 8	128 x 8
PIC16F628	2048 x 14	224 x 8	128 x 8
PIC16LF627	1024 x 14	224 x 8	128 x 8
PIC16LF628	2048 x 14	224 x 8	128 x 8

The PIC16F62X can directly or indirectly address its register files or data memory. All Special Function registers, including the program counter, are mapped in the data memory. The PIC16F62X have an orthogonal (symmetrical) instruction set that makes it possible to carry out any operation, on any register, using any Addressing mode. This symmetrical nature, and lack of 'special optimal situations' make programming with the PIC16F62X simple yet efficient. In addition, the learning curve is reduced significantly.

The PIC16F62X devices contain an 8-bit ALU and working register. The ALU is a general purpose arithmetic unit. It performs arithmetic and Boolean functions between data in the working register and any register file.

The ALU is 8-bit wide and capable of addition, subtraction, shift and logical operations. Unless otherwise mentioned, arithmetic operations are two's complement in nature. In two-operand instructions, typically one operand is the working register (W register). The other operand is a file register or an immediate constant. In single operand instructions, the operand is either the W register or a file register.

The W register is an 8-bit working register used for ALU operations. It is not an addressable register.
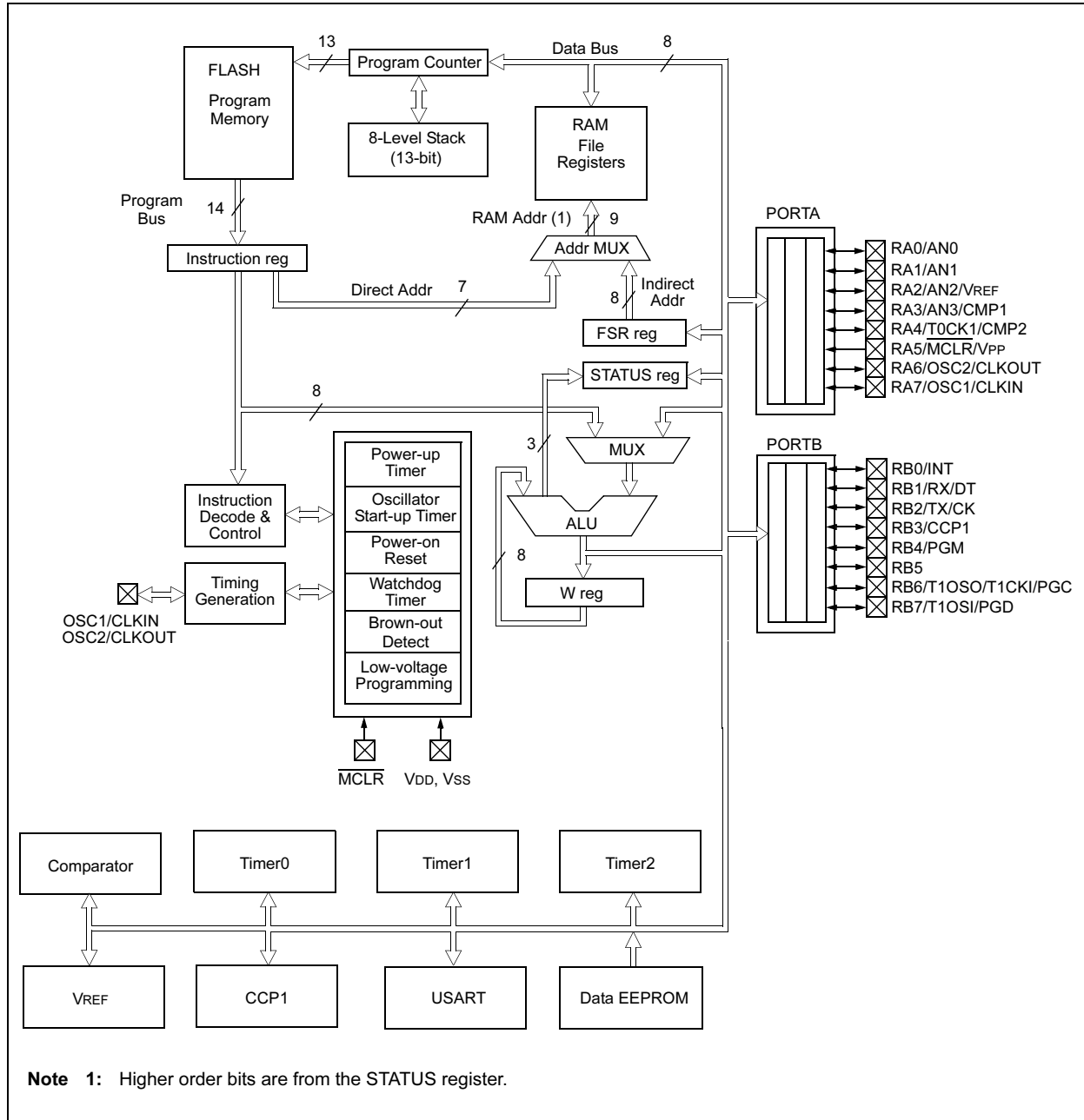
Depending on the instruction executed, the ALU may affect the values of the Carry (C), Digit Carry (DC), and Zero (Z) bits in the STATUS register. The C and DC bits operate as a Borrow and Digit Borrow out bit, respectively, bit in subtraction. See the SUBLW and SUBWF instructions for examples.

A simplified block diagram is shown in Figure 2-1, and a description of the device pins in Table 2-1.

Two types of data memory are provided on the PIC16F62X devices. Non-volatile EEPROM data memory is provided for long term storage of data such as calibration values, lookup table data, and any other data which may require periodic updating in the field. This data is not lost when power is removed. The other data memory provided is regular RAM data memory. Regular RAM data memory is provided for temporary storage of data during normal operation. It is lost when power is removed.

PIC16F62X

Note 1: Higher order bits are from the STATUS register.

Preliminary © 2003 Microchip Technology Inc.

PIC16F62X

3.2.2.4 PIE1 Register

This register contains interrupt enable bits.

REGISTER 3-4: PIE1 REGISTER (ADDRESS: 8Ch)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
EEIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE

bit 7   bit 0

bit 7   **EEIE:** EE Write Complete Interrupt Enable Bit
1 = Enables the EE write complete interrupt
0 = Disables the EE write complete interrupt

bit 6   **CMIE**: Comparator Interrupt Enable bit
1 = Enables the comparator interrupt
0 = Disables the comparator interrupt

bit 5   **RCIE**: USART Receive Interrupt Enable bit
1 = Enables the USART receive interrupt
0 = Disables the USART receive interrupt

bit 4   **TXIE**: USART Transmit Interrupt Enable bit
1 = Enables the USART transmit interrupt
0 = Disables the USART transmit interrupt

bit 3   **Unimplemented**: Read as '0'

bit 2   **CCP1IE**: CCP1 Interrupt Enable bit
1 = Enables the CCP1 interrupt
0 = Disables the CCP1 interrupt

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

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

Preliminary

PIC16F62X

3.2.2.6 PCON Register

The PCON register contains flag bits to differentiate between a Power-on Reset, an external MCLR Reset, WDT Reset or a Brown-out Detect.

> **Note:** BOD is unknown on Power-on Reset. It must then be set by the user and checked on subsequent RESETS to see if BOD is cleared, indicating a brown-out has occurred. The BOD STATUS bit is a "don't care" and is not necessarily predictable if the brown-out circuit is disabled (by clearing the BODEN bit in the Configuration word).

REGISTER 3-6: **PCON REGISTER (ADDRESS: 0Ch)**

U-0	U-0	U-0	U-0	R/W-1	U-0	R/W-q	R/W-q
—	—	—	—	OSCF	—	POR	BOD

bit 7   bit 0

bit 7-4   **Unimplemented:** Read as '0'

bit 3   **OSCF**: INTRC/ER oscillator frequency
1 = 4 MHz typical[1]
0 = 37 KHz typical

bit 2   **Unimplemented:** Read as '0'

bit 1   **POR**: Power-on Reset STATUS bit
1 = No Power-on Reset occurred
0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

bit 0   **BOD**: Brown-out Detect STATUS bit
1 = No Brown-out Reset occurred
0 = A Brown-out Reset occurred (must be set in software after a Brown-out Reset occurs)

Note 1: When in ER Oscillator mode, setting OSCF = 1 will cause the oscillator frequency to change to the frequency specified by the external resistor.

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

PIC16F62X

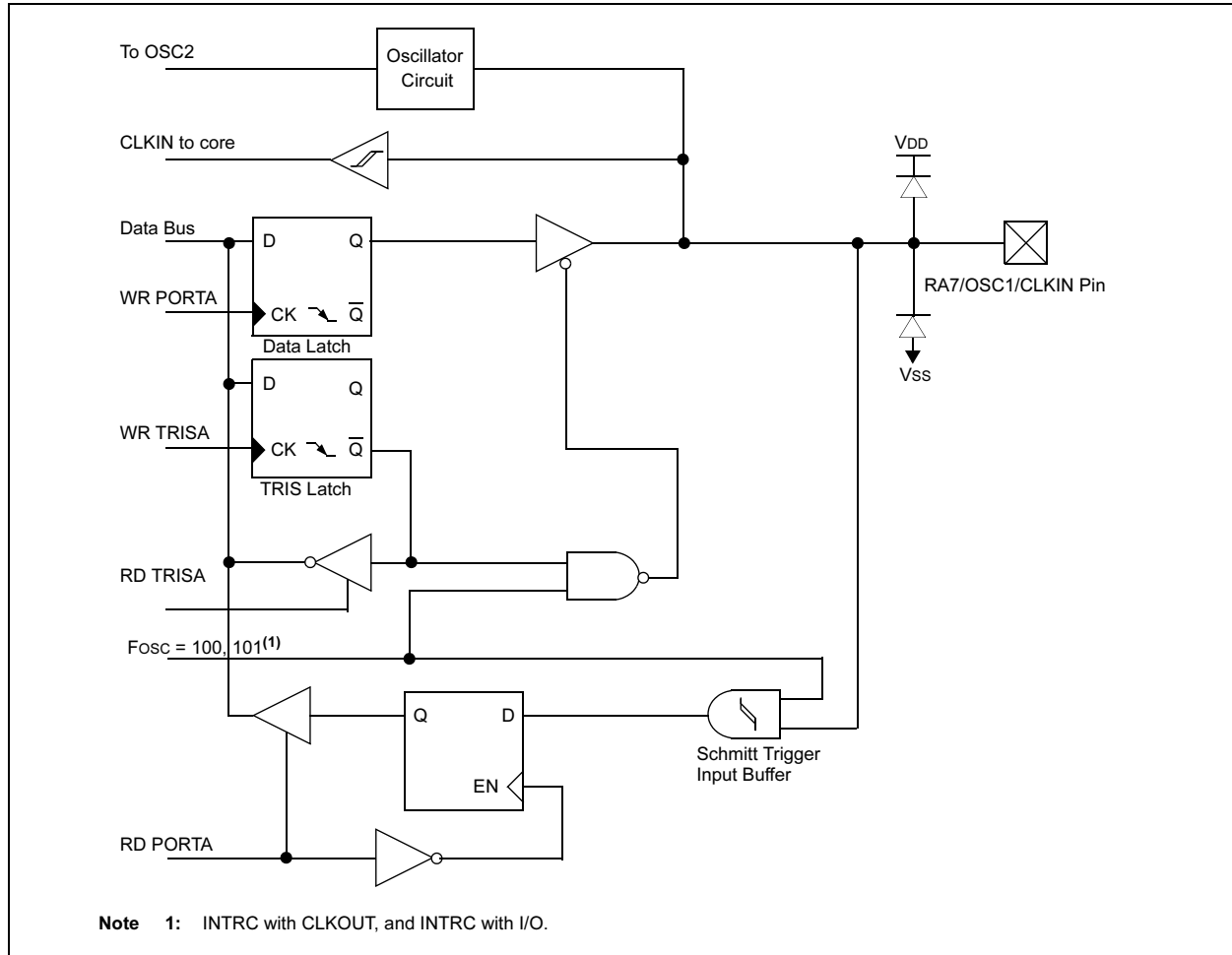**FIGURE 5-7:** **BLOCK DIAGRAM OF RA7/OSC1/CLKIN PIN**

Note 1: INTRC with CLKOUT, and INTRC with I/O.

TABLE 5-3: **PORTB FUNCTIONS**

Name	Function	Input Type	Output Type	Description
RB0/INT	RB0	TTL	CMOS	Bi-directional I/O port. Can be software programmed for internal weak pull-up.
	INT	ST	—	External interrupt.
RB1/RX/DT	RB1	TTL	CMOS	Bi-directional I/O port. Can be software programmed for internal weak pull-up.
	RX	ST	—	USART Receive Pin
	DT	ST	CMOS	Synchronous data I/O
RB2/TX/CK	RB2	TTL	CMOS	Bi-directional I/O port
	TX	—	CMOS	USART Transmit Pin
	CK	ST	CMOS	Synchronous Clock I/O. Can be software programmed for internal weak pull-up.
RB3/CCP1	RB3	TTL	CMOS	Bi-directional I/O port. Can be software programmed for internal weak pull-up.
	CCP1	ST	CMOS	Capture/Compare/PWM/I/O
RB4/PGM	RB4	TTL	CMOS	Bi-directional I/O port. Can be software programmed for internal weak pull-up.
	PGM	ST	—	Low voltage programming input pin. Interrupt-on-pin change. When low voltage programming is enabled, the interrupt-on-pin change and weak pull-up resistor are disabled.
RB5	RB5	TTL	CMOS	Bi-directional I/O port. Interrupt-on-pin change. Can be software programmed for internal weak pull-up.
RB6/T1OSO/T1CKI/ PGC	RB6	TTL	CMOS	Bi-directional I/O port. Interrupt-on-pin change. Can be software programmed for internal weak pull-up.
	T1OSO	—	XTAL	Timer1 Oscillator Output
	T1CKI	ST	—	Timer1 Clock Input
	PGC	ST	—	ICSP Programming Clock
RB7/T1OSI/PGD	RB7	TTL	CMOS	Bi-directional I/O port. Interrupt-on-pin change. Can be software programmed for internal weak pull-up.
	T1OSI	XTAL	—	Timer1 Oscillator Input
	PGD	ST	CMOS	ICSP Data I/O

Legend: O = Output   CMOS = CMOS Output   P = Power
      — = Not used   I = Input   ST = Schmitt Trigger Input
      TTL = TTL Input   OD = Open Drain Output   AN = Analog

TABLE 5-4: **SUMMARY OF REGISTERS ASSOCIATED WITH PORTB[1]**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR	Value on All Other RESETS
06h, 106h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	uuuu uuuu
86h, 186h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	1111 1111
81h, 181h	OPTION	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

Legend: u = unchanged, x = unknown
Note 1: Shaded bits are not used by PORTB.

PIC16F62X

9.1 Comparator Configuration

There are eight modes of operation for the comparators. The CMCON register is used to select the mode. Figure 9-1 shows the eight possible modes. The TRISA register controls the data direction of the comparator pins for each mode. If the Comparator mode is changed, the comparator output level may not be valid for the specified mode change delay shown in Table 17-1.

> **Note:** Comparator interrupts should be disabled during a Comparator mode change otherwise a false interrupt may occur.

FIGURE 9-1:   COMPARATOR I/O OPERATING MODES

A = Analog Input, port reads zeros always.
D = Digital Input.
CIS (CMCON<3>) is the Comparator Input Switch.

9.6   Comparator Interrupts

The Comparator Interrupt flag is set whenever there is a change in the output value of either comparator. 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, PIR1<6>, is the Comparator Interrupt Flag. The CMIF bit must be RESET by clearing '0'. Since it is also possible to write a '1' to this register, a simulated interrupt may be initiated.

The CMIE bit (PIE1<6>) 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.

Note:	If a change in the CMCON register (C1OUT or C2OUT) should occur when a read operation is being executed (start of the Q2 cycle), then the CMIF (PIR1<6>) interrupt flag may not get set.

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

a)   Any write or read of CMCON. This will end the mismatch condition.

b)   Clear flag bit CMIF.

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

9.7   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, higher SLEEP currents than shown in the power-down current specification will occur. 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, CM<2:0> = 111, before entering SLEEP. If the device wakes-up from SLEEP, the contents of the CMCON register are not affected.

9.8   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.

9.9   Analog Input Connection Considerations

A simplified circuit for an analog input is shown in Figure 9-4. Since the analog pins are connected to a digital output, they have reverse biased diodes to VDD and VSS. The analog input therefore, must be between VSS and VDD. If the input voltage deviates from this range by more than 0.6V in either direction, one of the diodes is forward biased and a latchup may occur. A maximum source impedance of 10 kΩ is recommended for the analog sources. Any external component connected to an analog input pin, such as a capacitor or a Zener diode, should have very little leakage current.
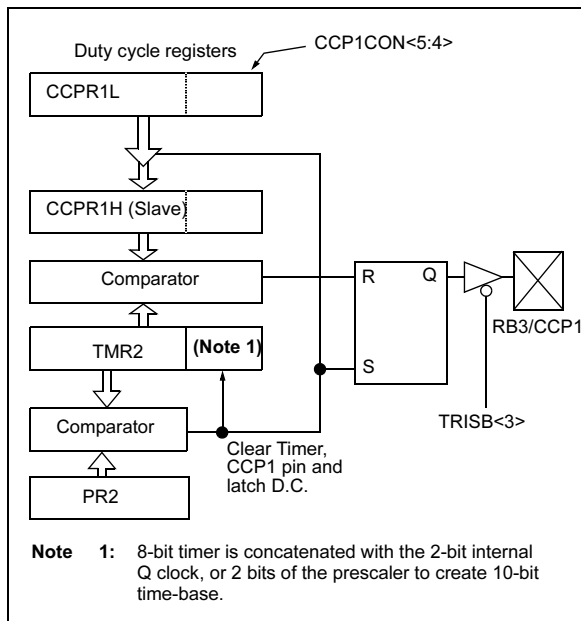
PIC16F62X

11.3   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 PORTB data latch, the TRISB<3> 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 PORTB I/O data latch.

Figure 11-2 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 11.3.3.

FIGURE 11-2:   SIMPLIFIED PWM BLOCK DIAGRAM

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

A PWM output (Figure 11-3) 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 11-3:   PWM OUTPUT
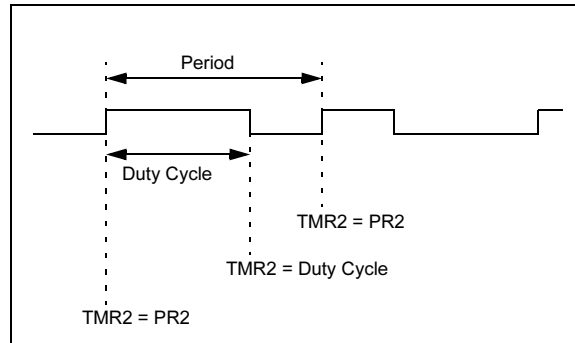
11.3.1   PWM PERIOD

The PWM period is specified by writing to the PR2register. The PWM period can be calculated using the following formula:

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

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 8.0) is not used in the determination of the PWM frequency. The postscaler could be used to have an interrupt occur at a different frequency than the PWM output.

FIGURE 12-5: **USART TRANSMIT BLOCK DIAGRAM**
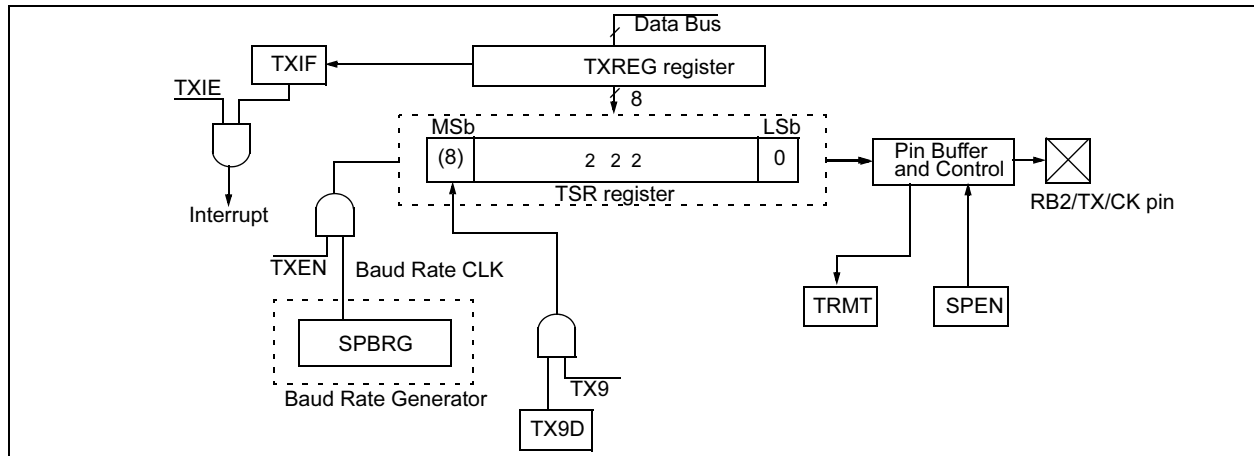
Steps to follow when setting up an Asynchronous Transmission:

1. Initialize the SPBRG register for the appropriate baud rate. If a high speed baud rate is desired, set bit BRGH. (Section 12.1)

2. Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.

3. If interrupts are desired, then set enable bit TXIE.

4. If 9-bit transmission is desired, then set transmit bit TX9.

5. Enable the transmission by setting bit TXEN, which will also set bit TXIF.

6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.

7. Load data to the TXREG register (starts transmission).
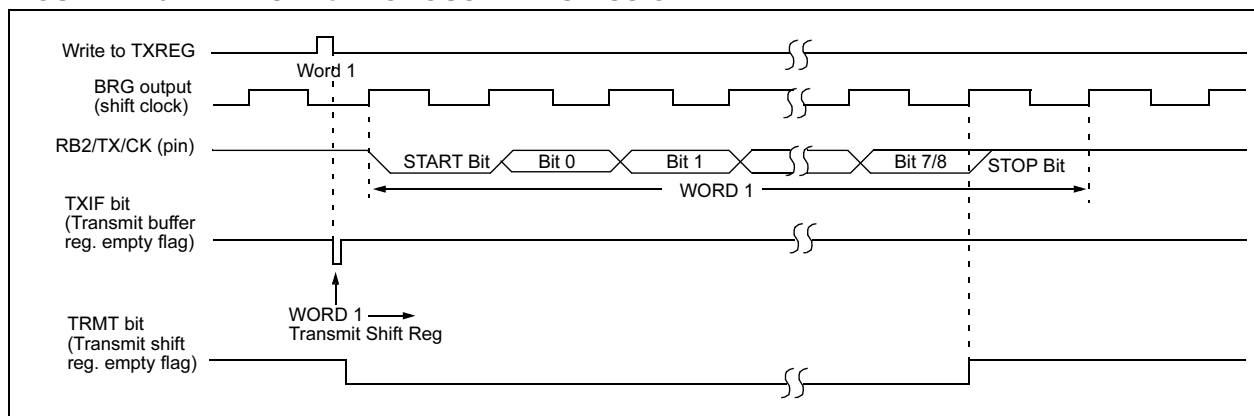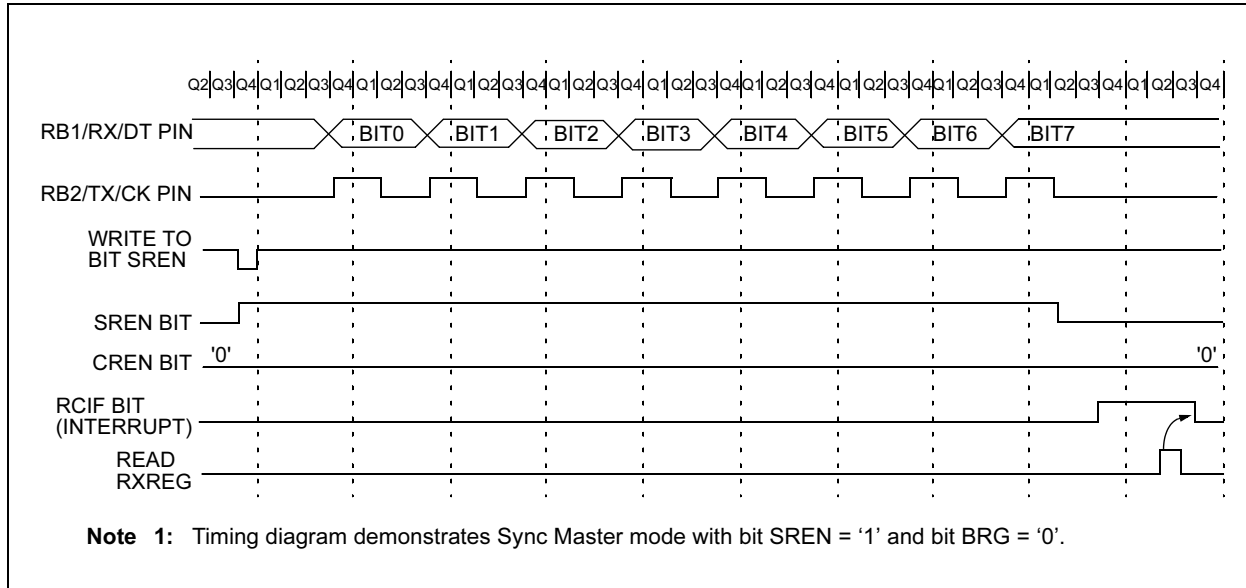
FIGURE 12-6: **ASYNCHRONOUS TRANSMISSION**

FIGURE 12-14:   SYNCHRONOUS RECEPTION (MASTER MODE, SREN)

Note   1: Timing diagram demonstrates Sync Master mode with bit SREN = '1' and bit BRG = '0'.

12.5   USART Synchronous Slave Mode

Synchronous Slave mode differs from the Master mode in the fact that the shift clock is supplied externally at the RB2/TX/CK pin (instead of being supplied internally in Master mode). This allows the device to transfer or receive data while in SLEEP mode. Slave mode is entered by clearing bit CSRC (TXSTA<7>).

12.5.1   USART SYNCHRONOUS SLAVE TRANSMIT

The operation of the Synchronous Master and Slave modes are identical except in the case of the SLEEP mode.

If two words are written to the TXREG and then the SLEEP instruction is executed, the following will occur:

a) The first word will immediately transfer to the TSR register and transmit.

b) The second word will remain in TXREG register.

c) Flag bit TXIF will not be set.

d) When the first word has been shifted out of TSR, the TXREG register will transfer the second word to the TSR and flag bit TXIF will now be set.

e) If enable bit TXIE is set, the interrupt will wake the chip from SLEEP and if the global interrupt is enabled, the program will branch to the interrupt vector (0004h).

Steps to follow when setting up a Synchronous Slave Transmission:

1. Enable the synchronous slave serial port by setting bits SYNC and SPEN and clearing bit CSRC.

2. Clear bits CREN and SREN.

3. If interrupts are desired, then set enable bit TXIE.

4. If 9-bit transmission is desired, then set bit TX9.

5. Enable the transmission by setting enable bit TXEN.

6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.

7. Start transmission by loading data to the TXREG register.

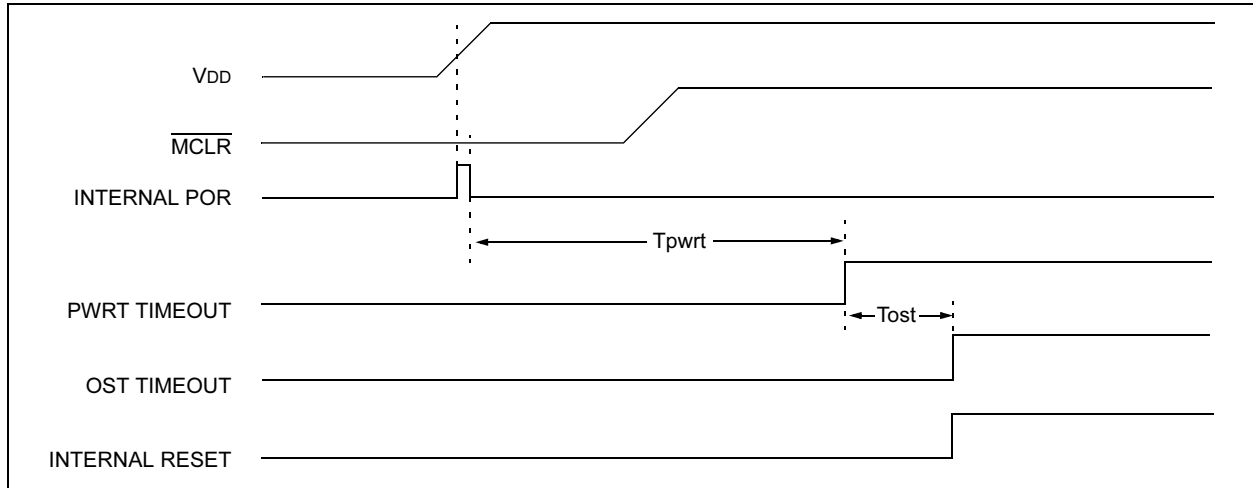**FIGURE 14-8:** TIMEOUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ NOT TIED TO V$_{DD}$): CASE

FIGURE 14-9: TIMEOUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ NOT TIED TO V$_{DD}$): CASE 2
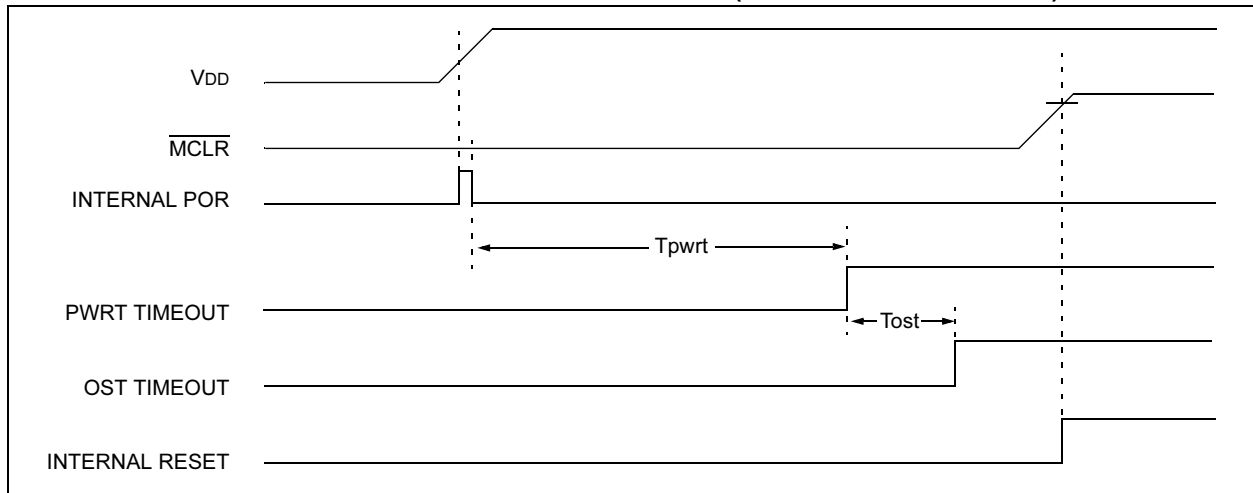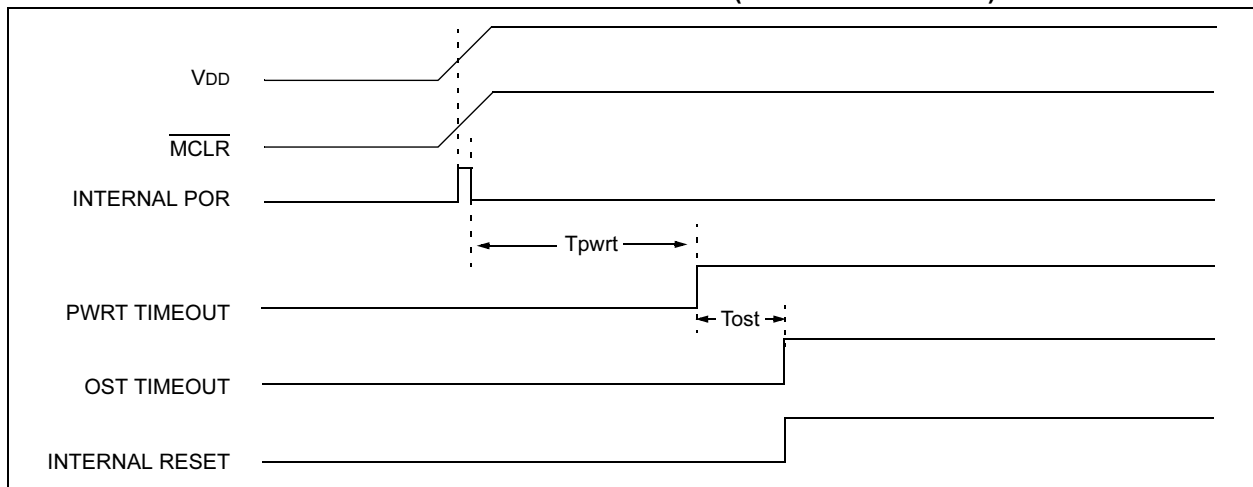

FIGURE 14-10: TIMEOUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ TIED TO V$_{DD}$)

15.0   INSTRUCTION SET SUMMARY

Each PIC16F62X instruction is a 14-bit word divided into an OPCODE which specifies the instruction type and one or more operands which further specify the operation of the instruction. The PIC16F62X instruction set summary in Table 15-2 lists byte-oriented, bit-oriented, and literal and control operations. Table 15-1 shows the opcode field descriptions.

For byte-oriented instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the W register. If 'd' is one, the result is placed in the file register specified in the instruction.

For bit-oriented instructions, 'b' represents a bit field designator which selects the number of the bit affected by the operation, while 'f' represents the number of the file in which the bit is located.

For literal and control operations, 'k' represents an eight or eleven bit constant or literal value.

TABLE 15-1:   OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1) The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1
label	Label name
TOS	Top of Stack
PC	Program Counter
PCLATH	Program Counter High Latch
GIE	Global Interrupt Enable bit
WDT	Watchdog Timer/Counter
TO	Timeout bit
PD	Power-down bit
dest	Destination either the W register or the specified register file location
[ ]	Options
( )	Contents
→	Assigned to
< >	Register bit field
∈	In the set of
italics	User defined term (font is courier)

The instruction set is highly orthogonal and is grouped into three basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal and control** operations

All instructions are executed within one single instruction cycle, unless a conditional test is true or the program counter is changed as a result of an instruction. In this case, the execution takes two instruction cycles with the second cycle executed as a NOP. One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 μs. If a conditional test is true or the program counter is changed as a result of an instruction, the instruction execution time is 2 μs.

Table 15-2 lists the instructions recognized by the MPASM™ assembler.

Figure 15-1 shows the three general formats that the instructions can have.

> **Note 1:** Any unused opcode is reserved. Use of any reserved opcode may cause unexpected operation.
>
> **2:** To maintain upward compatibility with future PICmicro® products, do not use the OPTION and TRIS instructions.

All examples use the following format to represent a hexadecimal number:

   0xhh

where h signifies a hexadecimal digit.

FIGURE 15-1:   GENERAL FORMAT FOR INSTRUCTIONS

Byte-oriented file register operations

13		8	7	6		0
	OPCODE		d		f (FILE #)	

d = 0 for destination W
d = 1 for destination f
f = 7-bit file register address

Bit-oriented file register operations

13		10	9		7	6		0
	OPCODE		b (BIT #)			f (FILE #)		

b = 3-bit bit address
f = 7-bit file register address

Literal and control operations
General

13		8	7		0
	OPCODE			k (literal)	

k = 8-bit immediate value

CALL and GOTO instructions only

13		11	10		0
	OPCODE			k (literal)	

k = 11-bit immediate value

RETLW	Return with Literal in W
Syntax:	[*label*] RETLW k
Operands:	$0 \leq k \leq 255$
Operation:	$k \rightarrow (W)$; $TOS \rightarrow PC$
Status Affected:	None
Encoding:	`11` `01xx` `kkkk` `kkkk`
Description:	The W register is loaded with the eight bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a two-cycle instruction.
Words:	1
Cycles:	2

Example

```
CALL TABLE;W contains table
          ;offset value
•         ;W now has table
value
•
TABLE •
•
ADDWF PC ;W = offset
RETLW k1 ;Begin table
RETLW k2  ;
•
•
•
RETLW kn  ; End of table
```

Before Instruction
      W = 0x07
After Instruction
      W = value of k8

RETURN	Return from Subroutine
Syntax:	[*label*] RETURN
Operands:	None
Operation:	$TOS \rightarrow PC$
Status Affected:	None
Encoding:	`00` `0000` `0000` `1000`
Description:	Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a two-cycle instruction.
Words:	1
Cycles:	2

Example   `RETURN`

After Interrupt
      PC = TOS

RLF	Rotate Left f through Carry
Syntax:	[*label*] RLF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	See description below
Status Affected:	C
Encoding:	`00` `1101` `dfff` `ffff`
Description:	The contents of register 'f' are rotated one bit to the left through the Carry Flag. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words:	1
Cycles:	1

Example   `RLF    REG1, 0`

Before Instruction
      REG1 = 1110 0110
      C   = 0
After Instruction
      REG1 = 1110 0110
      W   = 1100 1100
      C   = 1

16.3   MPLAB C17 and MPLAB C18 C Compilers

The MPLAB C17 and MPLAB C18 Code Development Systems are complete ANSI C compilers for Microchip's PIC17CXXX and PIC18CXXX family of microcontrollers. These compilers provide powerful integration capabilities, superior code optimization and ease of use not found with other compilers.

For easy source level debugging, the compilers provide symbol information that is optimized to the MPLAB IDE debugger.

16.4   MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK object linker combines relocatable objects created by the MPASM assembler and the MPLAB C17 and MPLAB C18 C compilers. It can link relocatable objects from pre-compiled libraries, using directives from a linker script.

The MPLIB object librarian manages the creation and modification of library files of pre-compiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

• Efficient linking of single libraries instead of many smaller files
• Enhanced code maintainability by grouping related modules together
• Flexible creation of libraries with easy module listing, replacement, deletion and extraction

16.5   MPLAB C30 C Compiler

The MPLAB C30 C compiler is a full-featured, ANSI compliant, optimizing compiler that translates standard ANSI C programs into dsPIC30F assembly language source. The compiler also supports many command-line options and language extensions to take full advantage of the dsPIC30F device hardware capabili-ties, and afford fine control of the compiler code generator.

MPLAB C30 is distributed with a complete ANSI C standard library. All library functions have been validated and conform to the ANSI C library standard. The library includes functions for string manipulation, dynamic memory allocation, data conversion, time-keeping, and math functions (trigonometric, exponential and hyperbolic). The compiler provides symbolic information for high level source debugging with the MPLAB IDE.

16.6   MPLAB ASM30 Assembler, Linker, and Librarian

MPLAB ASM30 assembler produces relocatable machine code from symbolic assembly language for dsPIC30F devices. MPLAB C30 compiler uses the assembler to produce it's object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

• Support for the entire dsPIC30F instruction set
• Support for fixed-point and floating-point data
• Command line interface
• Rich directive set
• Flexible macro language
• MPLAB IDE compatibility

16.7   MPLAB SIM Software Simulator

The MPLAB SIM software simulator allows code development in a PC hosted environment by simulating the PICmicro series microcontrollers on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a file, or user defined key press, to any pin. The execution can be performed in Single-Step, Execute Until Break, or Trace mode.

The MPLAB SIM simulator fully supports symbolic debugging using the MPLAB C17 and MPLAB C18 C Compilers, as well as the MPASM assembler. The software simulator offers the flexibility to develop and debug code outside of the laboratory environment, making it an excellent, economical software development tool.

16.8   MPLAB SIM30 Software Simulator

The MPLAB SIM30 software simulator allows code development in a PC hosted environment by simulating the dsPIC30F series microcontrollers on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a file, or user defined key press, to any of the pins.

The MPLAB SIM30 simulator fully supports symbolic debugging using the MPLAB C30 C Compiler and MPLAB ASM30 assembler. The simulator runs in either a Command Line mode for automated tasks, or from MPLAB IDE. This high speed simulator is designed to debug, analyze and optimize time intensive DSP routines.

PIC16F62X

17.1 DC Characteristics: PIC16F62X-04 (Commercial, Industrial, Extended)
PIC16F62X-20 (Commercial, Industrial, Extended)
PIC16LF62X-04 (Commercial, Industrial)

PIC16LF62X-04 (Commercial, Industrial)	**Standard Operating Conditions (unless otherwise stated)** Operating temperature $-40°C \leq Ta \leq +85°C$ for industrial and $0°C \leq Ta \leq +70°C$ for commercial
PIC16F62X-04 PIC16F62X-20 (Commercial, Industrial, Extended)	**Standard Operating Conditions (unless otherwise stated)** Operating temperature $-40°C \leq Ta \leq +85°C$ for industrial and $0°C \leq Ta \leq +70°C$ for commercial and $-40°C \leq Ta \leq +125°C$ for extended

Param No.	Sym	Characteristic/Device	Min	Typ†	Max	Units	Conditions
	V_{DD}	**Supply Voltage**					
D001		PIC16LF62X	2.0	—	5.5	V	
D001		PIC16F62X	3.0	—	5.5	V	
D002	V_{DR}	**RAM Data Retention Voltage**[1]	—	1.5	—	V	Device in SLEEP mode*
D003	V_{POR}	**V_{DD} Start Voltage** to ensure Power-on Reset	—	Vss	—	V	See section on Power-on Reset for details
D004	S_{VDD}	**V_{DD} Rise Rate** to ensure Power-on Reset	0.05	—	—	V/ms	See section on Power-on Reset for details*
D005	V_{BOD}	**Brown-out Detect Voltage**	3.65 3.65	4.0 —	4.35 4.4	V V	BODEN configuration bit is set BODEN configuration bit is set, Extended
	I_{DD}	**Supply Current**[2, 5]					
D010 D013		PIC16LF62X	— — — — — —	0.30 1.10 4.0 3.80 — 20	0.6 2.0 7.0 6.0 2.0 30	mA mA mA mA mA μA	Fosc = 4.0 MHz, VDD = 2.0[5] FOSC = 4.0 MHz, VDD = 5.5* Fosc = 20.0 MHz, VDD = 5.5 Fosc = 20.0 MHz, VDD = 4.5* FOSC = 10.0 MHz, VDD = 3.0[6] FOSC = 32 kHz, VDD = 2.0
D010 D013 D014		PIC16F62X	— — — — — 	0.60 1.10 4.0 3.80 — 20	0.7 2.0 7.0 6.0 2.0 30	mA mA mA mA mA μA	Fosc = 4.0 MHz, VDD = 3.0 Fosc = 4.0 MHz, VDD = 5.5* FOSC = 20.0 MHz, VDD = 5.5 FOSC = 20.0 MHz, VDD = 4.5* FOSC = 10.0 MHz, VDD = 3.0*[6] FOSC = 32 kHz, VDD = 3.0*

Legend: Rows with standard voltage device data only are shaded for improved readability.

    *   These parameters are characterized but not tested.

    †   Data in "Typ" column is at 5.0V, 25°C, unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which V_{DD} can be lowered in SLEEP mode 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 I_{DD} measurements in active Operation mode are:
   OSC1 = external square wave, from rail to rail; all I/O pins tri-stated, 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 or Vss.

  4: The Δ current is the additional current consumed when this peripheral is enabled. This current should be added to the base I_{DD} or I_{PD} measurement.

  5: For RC osc configuration, current through R_{EXT} is not included. The current through the resistor can be estimated by the formula $Ir = V_{DD}/2R_{EXT}$ (mA) with R_{EXT} in kΩ.

  6: Commercial temperature only.

17.2 DC Characteristics: PIC16F62X (Commercial, Industrial, Extended)
PIC16LF62X (Commercial, Industrial)

DC CHARACTERISTICS			**Standard Operating Conditions (unless otherwise stated)**
			 Operating temperature -40°C ≤ TA ≤ +85°C for industrial and
			                 0°C ≤ TA ≤ +70°C for commercial and
			                 -40°C ≤ TA ≤ +125°C for extended
			 Operating voltage VDD range as described in DC spec Table 17-1 and Table 17-2				

Param. No.	Sym	Characteristic/Device	Min	Typ†	Max	Unit	Conditions
	VIL	**Input Low Voltage**					
D030		I/O ports with TTL buffer	Vss	—	0.8 0.15 VDD	V V	VDD = 4.5V to 5.5V otherwise
D031		with Schmitt Trigger input	Vss	—	0.2 VDD	V	
D032		MCLR, RA4/T0CKI,OSC1 (in ER mode)	Vss	—	0.2 VDD	V	**(Note1)**
D033		OSC1 (in XT and HS) OSC1 (in LP)	Vss Vss	— —	0.3 VDD 0.6 VDD - 1.0	V V	
	VIH	**Input High Voltage**					
D040		I/O ports with TTL buffer	2.0V .25 VDD + 0.8V	—	VDD VDD	V V	VDD = 4.5V to 5.5V otherwise
D041		with Schmitt Trigger input	0.8 VDD	—	VDD	V	
D042		MCLR RA4/T0CKI	0.8 VDD	—	VDD	V	
D043		OSC1 (XT, HS and LP)	0.7 VDD	—	VDD	V	
D043A		OSC1 (in ER mode)	0.9 VDD			V	**(Note1)**
D070	IPURB	**PORTB weak pull-up current**	50	200	400	µA	VDD = 5.0V, VPIN = Vss
	IIL	**Input Leakage Current[2], [3]**					
		I/O ports (Except PORTA)			±1.0	µA	Vss ≤ VPIN ≤ VDD, pin at hi-impedance
D060		PORTA	—	—	±0.5	µA	Vss ≤ VPIN ≤ VDD, pin at hi-impedance
D061		RA4/T0CKI	—	—	±1.0	µA	Vss ≤ VPIN ≤ VDD
D063		OSC1, MCLR	—	—	±5.0	µA	Vss ≤ VPIN ≤ VDD, XT, HS and LP osc configuration
	VOL	**Output Low Voltage**					
D080		I/O ports	— —	— —	0.6 0.6	V V	IOL=8.5 mA, VDD=4.5V, -40° to +85°C IOL=7.0 mA, VDD=4.5V, +125°C
D083		OSC2/CLKOUT (ER only)	— —	— —	0.6 0.6	V V	IOL=1.6 mA, VDD=4.5V, -40° to +85°C IOL=1.2 mA, VDD=4.5V, +125°C
	VOH	**Output High Voltage[3]**					
D090		I/O ports (Except RA4)	VDD - 0.7 VDD - 0.7	— —	— —	V V	IOH=-3.0 mA, VDD=4.5V, -40° to +85°C IOH=-2.5 mA, VDD=4.5V, +125°C
D092		OSC2/CLKOUT (ER only)	VDD - 0.7 VDD - 0.7	— —	— —	V V	IOH=-1.3 mA, VDD=4.5V, -40° to +85°C IOH=-1.0 mA, VDD=4.5V, +125°C
D150	VOD	**Open-Drain High Voltage**		—	8.5	V	RA4 pin PIC16F62X, PIC16LF62X*
		Capacitive Loading Specs on Output Pins					
D100*	COSC2	OSC2 pin		—	15	pF	In XT, HS and LP modes when external clock used to drive OSC1.
D101*	Cio	All I/O pins/OSC2 (in ER mode)		—	50	pF	

*   These parameters are characterized but not tested.

†   Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: In ER oscillator configuration, the OSC1 pin is a Schmitt Trigger input. It is not recommended that the PIC16F62X be driven with external clock in ER mode.

2: The leakage current on the MCLR pin is strongly dependent on applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

3: Negative current is defined as coming out of the pin.
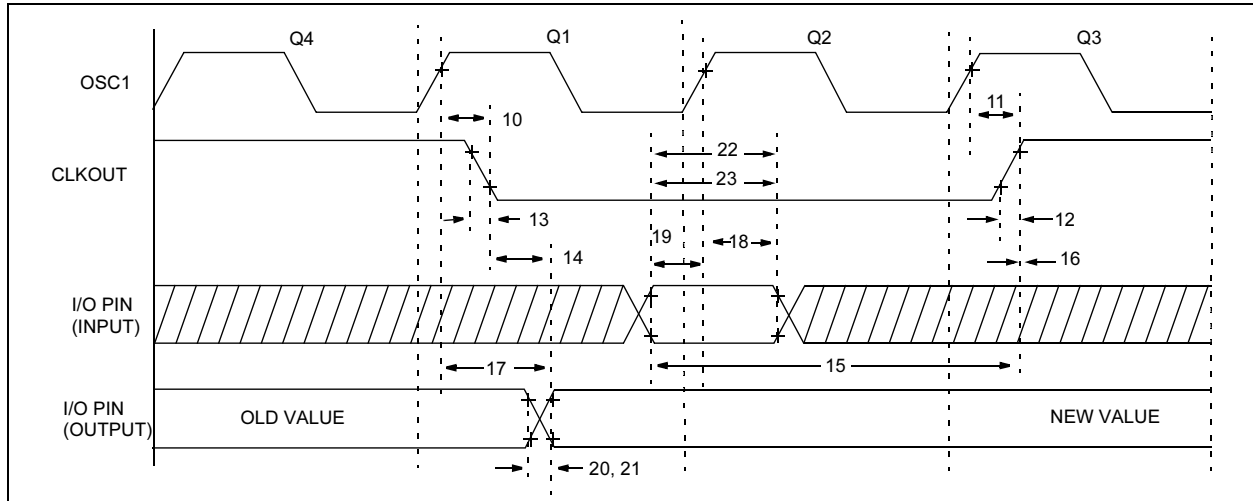
FIGURE 17-7: CLKOUT AND I/O TIMING

TABLE 17-5: CLKOUT AND I/O TIMING REQUIREMENTS

Param No.	Sym	Characteristic		Min	Typ†	Max	Units
10*	TosH2ckL	OSC1↑ to CLKOUT↓	16F62X	—	75	200	ns
10A*			16LF62X	—	—	400	ns
11*	TosH2ckH	OSC1↑ to CLKOUT↑	16F62X	—	75	200	ns
11A*			16LF62X	—	—	400	ns
12*	TckR	CLKOUT rise time	16F62X	—	35	100	ns
12A*			16LF62X	—	—	200	ns
13*	TckF	CLKOUT fall time	16F62X	—	35	100	ns
13A*			16LF62X	—	—	200	ns
14*	TckL2ioV	CLKOUT ↓ to Port out valid		—	—	20	ns
15*	TioV2ckH	Port in valid before	16F62X	Tosc+200 ns	—	—	ns
		CLKOUT ↑	16LF62X	Tosc=400 ns	—	—	ns
16*	TckH2ioI	Port in hold after CLKOUT ↑		0	—	—	ns
17*	TosH2ioV	OSC1↑ (Q1 cycle) to	16F62X	—	50	150*	ns
		Port out valid	16LF62X	—	—	300	ns
18*	TosH2ioI	OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time)		100 200	—	—	ns

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.