



Welcome to [E-XFL.COM](https://www.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	Obsolete
Core Processor	PIC
Core Size	8-Bit
Speed	4MHz
Connectivity	-
Peripherals	Brown-out Detect/Reset, POR, WDT
Number of I/O	13
Program Memory Size	3.5KB (2K x 14)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	2.5V ~ 5.5V
Data Converters	-
Oscillator Type	External
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SSOP (0.209", 5.30mm Width)
Supplier Device Package	20-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lc622at-04e-ss

4.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers are registers used by the CPU and Peripheral functions for controlling the desired operation of the device (Table 4-1). These registers are static RAM.

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

TABLE 4-1: SPECIAL REGISTERS FOR THE PIC16C62X

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR Reset	Value on all other RESETS ⁽¹⁾
Bank 0											
00h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx
01h	TMR0	Timer0 Module's Register								xxxx xxxx	uuuu uuuu
02h	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
03h	STATUS	IRP ⁽²⁾	RP1 ⁽²⁾	RP0	\overline{TO}	\overline{PD}	Z	DC	C	0001 1xxx	000q quuu
04h	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
05h	PORTA	—	—	—	$\overline{RA4}$	$\overline{RA3}$	$\overline{RA2}$	$\overline{RA1}$	RA0	---x 0000	---u 0000
06h	PORTB	RB7	RB6	RB5	$\overline{RB4}$	$\overline{RB3}$	$\overline{RB2}$	$\overline{RB1}$	RB0	xxxx xxxx	uuuu uuuu
07h-09h	Unimplemented									—	—
0Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of program counter					---0 0000	---0 0000
0Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	—	CMIF	—	—	—	—	—	—	-0-- ----	-0-- ----
0Dh-1Eh	Unimplemented									—	—
1Fh	CMCON	C2OUT	C1OUT	—	—	CIS	CM2	CM1	CM0	00-- 0000	00-- 0000
Bank 1											
80h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx
81h	OPTION	\overline{RBPU}	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
82h	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
83h	STATUS	IRP ⁽²⁾	RP1 ⁽²⁾	RP0	\overline{TO}	\overline{PD}	Z	DC	C	0001 1xxx	000q quuu
84h	FSR	Indirect data memory address pointer								xxxx xxxx	uuuu uuuu
85h	TRISA	—	—	—	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	---1 1111	---1 1111
86h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	1111 1111
87h-89h	Unimplemented									—	—
8Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of program counter					---0 0000	---0 0000
8Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
8Ch	PIE1	—	CMIE	—	—	—	—	—	—	-0-- ----	-0-- ----
8Dh	Unimplemented									—	—
8Eh	PCON	—	—	—	—	—	—	POR	\overline{BOR}	---- --0x	---- --uq
8Fh-9Eh	Unimplemented									—	—
9Fh	VRCON	VREN	VROE	VRR	—	VR3	VR2	VR1	VR0	000- 0000	000- 0000

Legend: — = Unimplemented locations read as '0', u = unchanged, x = unknown,
q = value depends on condition, shaded = unimplemented

Note 1: Other (non Power-up) Resets include MCLR Reset, Brown-out Reset and Watchdog Timer Reset during normal operation.

2: IRP & RP1 bits are reserved; always maintain these bits clear.

4.2.2.4 PIE1 Register

This register contains the individual enable bit for the comparator interrupt.

REGISTER 4-4: PIE1 REGISTER (ADDRESS 8CH)

U-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
—	CMIE	—	—	—	—	—	—
bit 7							bit 0

- bit 7 **Unimplemented:** Read as '0'
- bit 6 **CMIE:** Comparator Interrupt Enable bit
1 = Enables the Comparator interrupt
0 = Disables the Comparator interrupt
- bit 5-0 **Unimplemented:** Read as '0'

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

4.2.2.5 PIR1 Register

This register contains the individual flag bit for the comparator interrupt.

Note: Interrupt flag bits get set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>). User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

REGISTER 4-5: PIR1 REGISTER (ADDRESS 0CH)

U-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
—	CMIF	—	—	—	—	—	—
bit 7							bit 0

- bit 7 **Unimplemented:** Read as '0'
- bit 6 **CMIF:** Comparator Interrupt Flag bit
1 = Comparator input has changed
0 = Comparator input has not changed
- bit 5-0 **Unimplemented:** Read as '0'

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

5.0 I/O PORTS

The PIC16C62X have two ports, PORTA and PORTB. 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.

5.1 PORTA and TRISA Registers

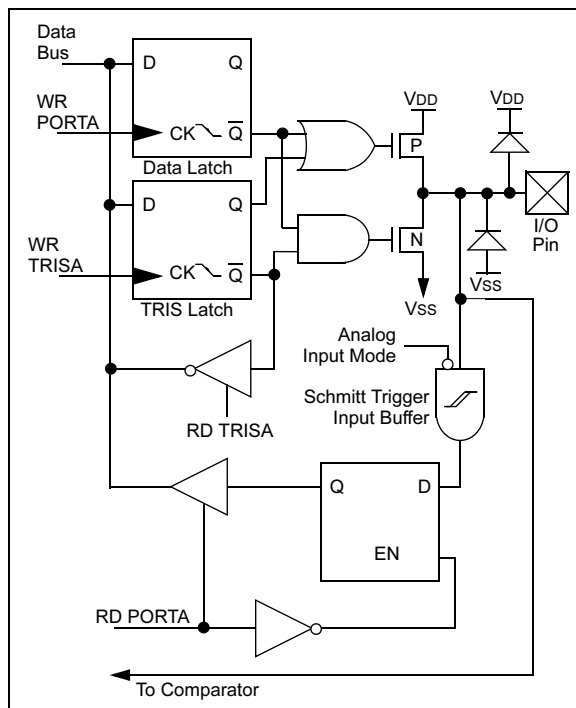
PORTA is a 5-bit wide latch. RA4 is a Schmitt Trigger input and an open drain output. Port RA4 is multiplexed with the T0CKI clock input. All other RA port pins have Schmitt Trigger input levels and full CMOS output drivers. All pins have data direction bits (TRIS registers), which can configure these pins as input or output.

A '1' in the TRISA register puts the corresponding output driver in a Hi-impedance mode. A '0' in the TRISA register puts the contents of the output latch on the selected pin(s).

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. So a write to a port implies that the port pins are first read, then this value is modified and written to the port data latch.

The PORTA pins are multiplexed with comparator and voltage reference functions. The operation of these pins are selected by control bits in the CMCON (comparator control register) register and the VRCON (voltage reference control register) register. When selected as a comparator input, these pins will read as '0's.

FIGURE 5-1: BLOCK DIAGRAM OF RA1:RA0 PINS



Note: On RESET, the TRISA register is set to all inputs. The digital inputs are disabled and the comparator inputs are forced to ground to reduce excess current consumption.

TRISA controls the direction of the RA pins, even when they are being used as comparator inputs. The user must make sure to keep the pins configured as inputs when using them as comparator inputs.

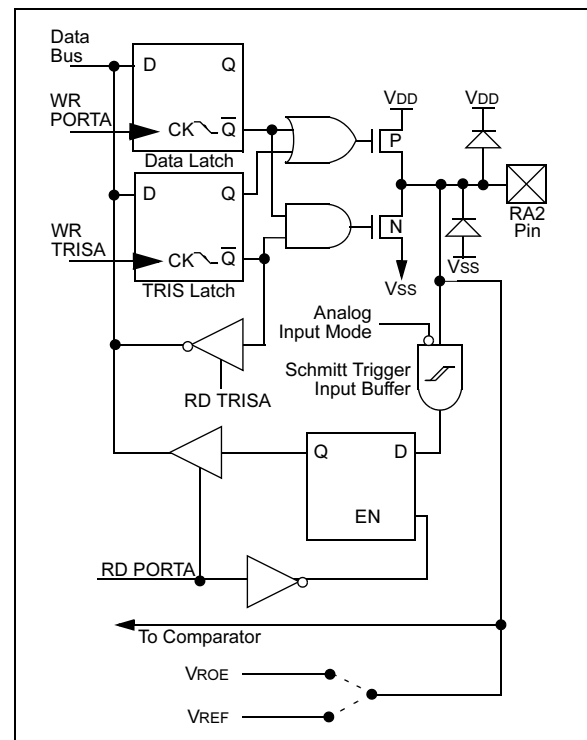
The RA2 pin will also function as the output for the voltage reference. When in this mode, the VREF pin is a very high impedance output and must be buffered prior to any external load. The user must configure TRISA<2> bit as an input and use high impedance loads.

In one of the Comparator modes defined by the CMCON register, pins RA3 and RA4 become outputs of the comparators. The TRISA<4:3> bits must be cleared to enable outputs to use this function.

EXAMPLE 5-1: INITIALIZING PORTA

```
CLRF    PORTA      ;Initialize PORTA by setting
                   ;output data latches
MOVLW   0X07       ;Turn comparators off and
MOVWF   CMCON      ;enable pins for I/O
                   ;functions
BSF     STATUS, RP0 ;Select Bank1
MOVLW   0x1F       ;Value used to initialize
                   ;data direction
MOVWF   TRISA      ;Set RA<4:0> as inputs
                   ;TRISA<7:5> are always
                   ;read as '0'.
```

FIGURE 5-2: BLOCK DIAGRAM OF RA2 PIN



PIC16C62X

5.3 I/O Programming Considerations

5.3.1 BI-DIRECTIONAL I/O PORTS

Any instruction which writes, operates internally as a read followed by a write operation. The `BCF` and `BSF` instructions, for example, read the register into the CPU, execute the bit operation and write the result back to the register. Caution must be used when these instructions are applied to a port with both inputs and outputs defined. For example, a `BSF` operation on bit5 of `PORTB` will cause all eight bits of `PORTB` to be read into the CPU. Then the `BSF` operation takes place on bit5 and `PORTB` is written to the output latches. If another bit of `PORTB` is used as a bi-directional I/O pin (e.g., bit0) and it is defined as an input at this time, the input signal present on the pin itself would be read into the CPU and re-written to the data latch of this particular pin, overwriting the previous content. As long as the pin stays in the Input mode, no problem occurs. However, if bit0 is switched into Output mode later on, the content of the data latch may now be unknown.

Reading the port register reads the values of the port pins. Writing to the port register writes the value to the port latch. When using read-modify-write instructions (ex. `BCF`, `BSF`, etc.) on a port, the value of the port pins is read, the desired operation is done to this value, and this value is then written to the port latch.

Example 5-2 shows the effect of two sequential read-modify-write instructions (ex., `BCF`, `BSF`, etc.) on an I/O port.

A pin actively outputting a Low or High should not be driven from external devices at the same time in order to change the level on this pin ("wired-or", "wired-and"). The resulting high output currents may damage the chip.

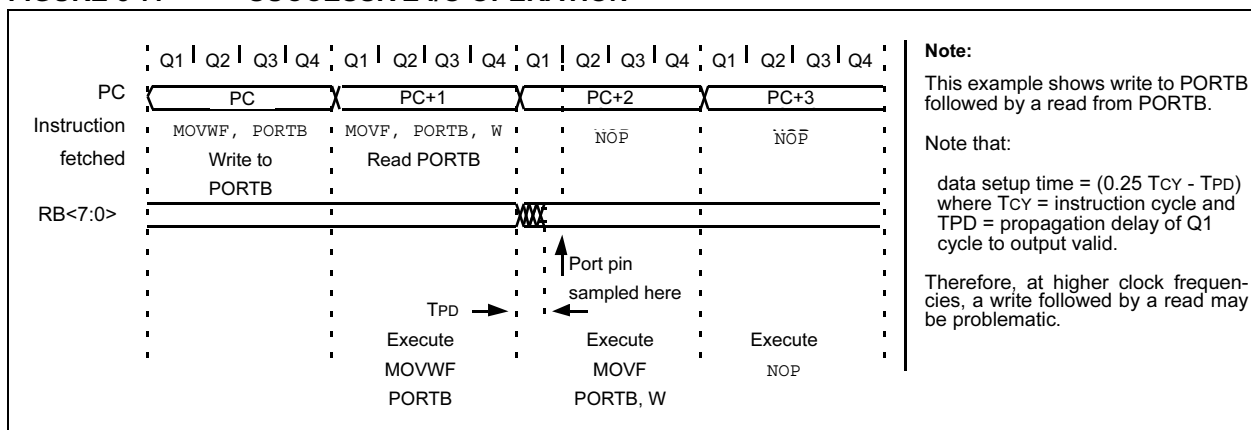
EXAMPLE 5-2: READ-MODIFY-WRITE INSTRUCTIONS ON AN I/O PORT

```
; Initial PORT settings:   PORTB<7:4> Inputs
;
;                           PORTB<3:0> Outputs
; PORTB<7:6> have external pull-up and are not
; connected to other circuitry
;
;                           PORT latch   PORT pins
;                           -----
;
;
; BCF PORTB, 7           ; 01pp pppp   11pp pppp
; BCF PORTB, 6           ; 10pp pppp   11pp pppp
; BSF STATUS,RP0         ;
; BCF TRISB, 7           ; 10pp pppp   11pp pppp
; BCF TRISB, 6           ; 10pp pppp   10pp pppp
;
;
; Note that the user may have expected the pin
; values to be 00pp pppp. The 2nd BCF caused
; RB7 to be latched as the pin value (High).
```

5.3.2 SUCCESSIVE OPERATIONS ON I/O PORTS

The actual write to an I/O port happens at the end of an instruction cycle, whereas for reading, the data must be valid at the beginning of the instruction cycle (Figure 5-7). Therefore, care must be exercised if a write followed by a read operation is carried out on the same I/O port. The sequence of instructions should be such to allow the pin voltage to stabilize (load dependent) before the next instruction which causes that file to be read into the CPU is executed. Otherwise, the previous state of that pin may be read into the CPU rather than the new state. When in doubt, it is better to separate these instructions with a `NOP` or another instruction not accessing this I/O port.

FIGURE 5-7: SUCCESSIVE I/O OPERATION



PIC16C62X

FIGURE 6-3: TIMER0 TIMING: INTERNAL CLOCK/PRESCALE 1:2

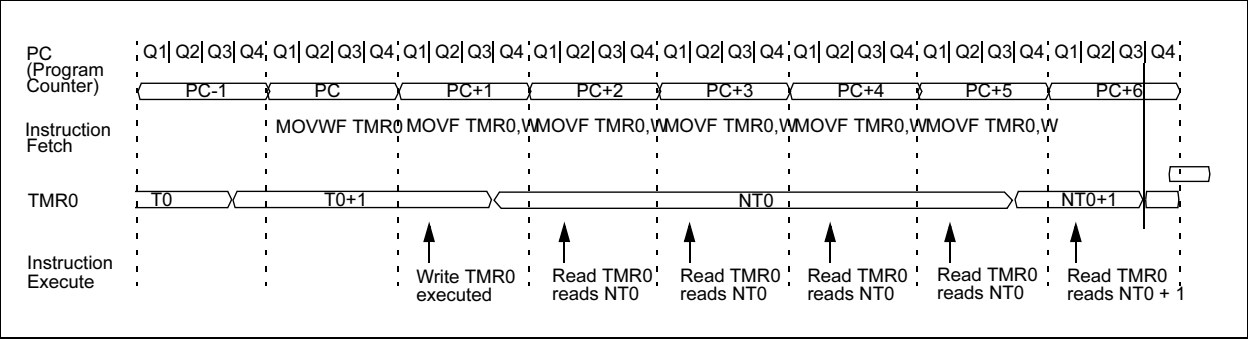
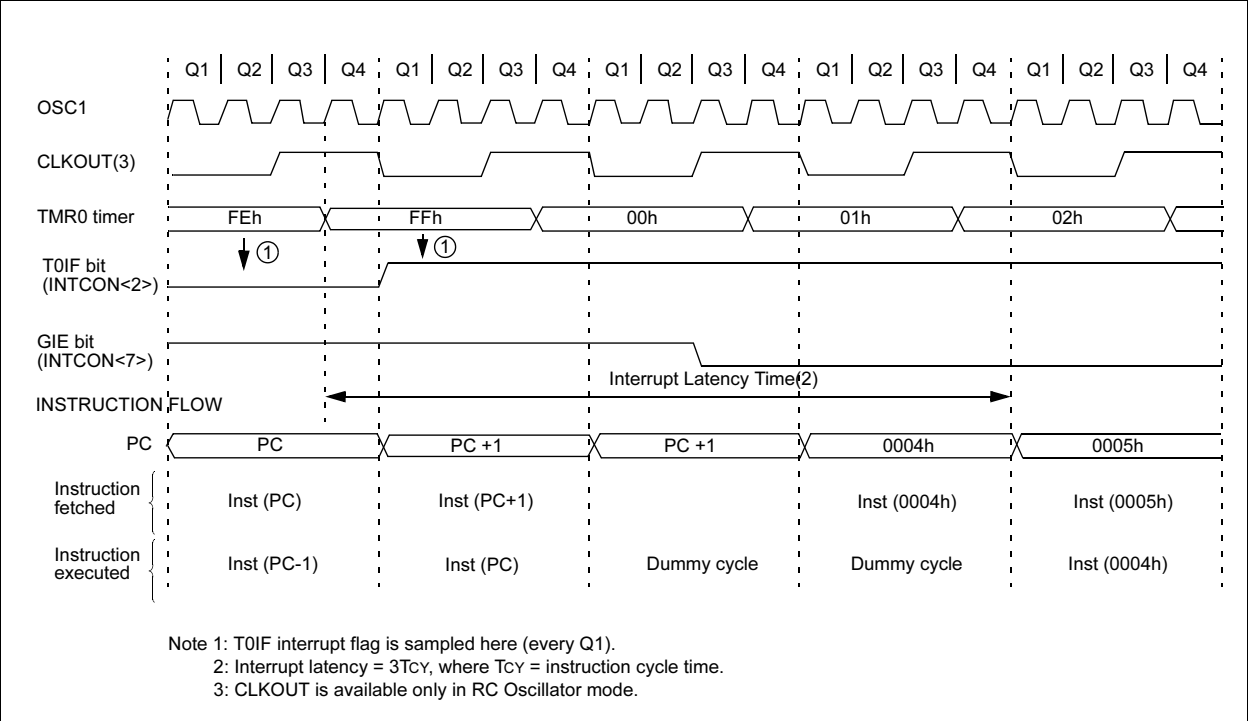


FIGURE 6-4: TIMER0 INTERRUPT TIMING



6.2 Using Timer0 with External Clock

When an external clock input is used for Timer0, it must meet certain requirements. The external clock requirement is due to internal phase clock (Tosc) synchronization. Also, there is a delay in the actual incrementing of Timer0 after synchronization.

6.2.1 EXTERNAL CLOCK SYNCHRONIZATION

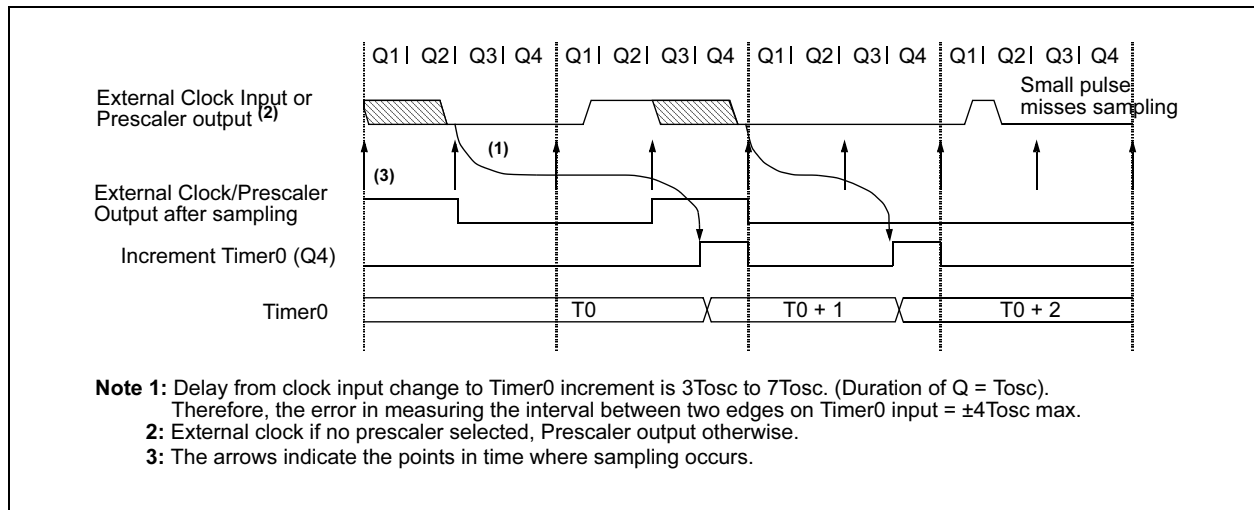
When no prescaler is used, the external clock input is the same as the prescaler output. The synchronization of T0CKI with the internal phase clocks is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks (Figure 6-5). Therefore, it is necessary for T0CKI to be high for at least 2Tosc (and a small RC delay of 20 ns) and low for at least 2Tosc (and a small RC delay of 20 ns). Refer to the electrical specification of the desired device.

When a prescaler is used, the external clock input is divided by the asynchronous ripple-counter type prescaler, so that the prescaler output is symmetrical. For the external clock to meet the sampling requirement, the ripple-counter must be taken into account. Therefore, it is necessary for T0CKI to have a period of at least 4Tosc (and a small RC delay of 40 ns) divided by the prescaler value. The only requirement on T0CKI high and low time is that they do not violate the minimum pulse width requirement of 10 ns. Refer to parameters 40, 41 and 42 in the electrical specification of the desired device.

6.2.2 TIMER0 INCREMENT DELAY

Since the prescaler output is synchronized with the internal clocks, there is a small delay from the time the external clock edge occurs to the time the TMR0 is actually incremented. Figure 6-5 shows the delay from the external clock edge to the timer incrementing.

FIGURE 6-5: TIMER0 TIMING WITH EXTERNAL CLOCK



PIC16C62X

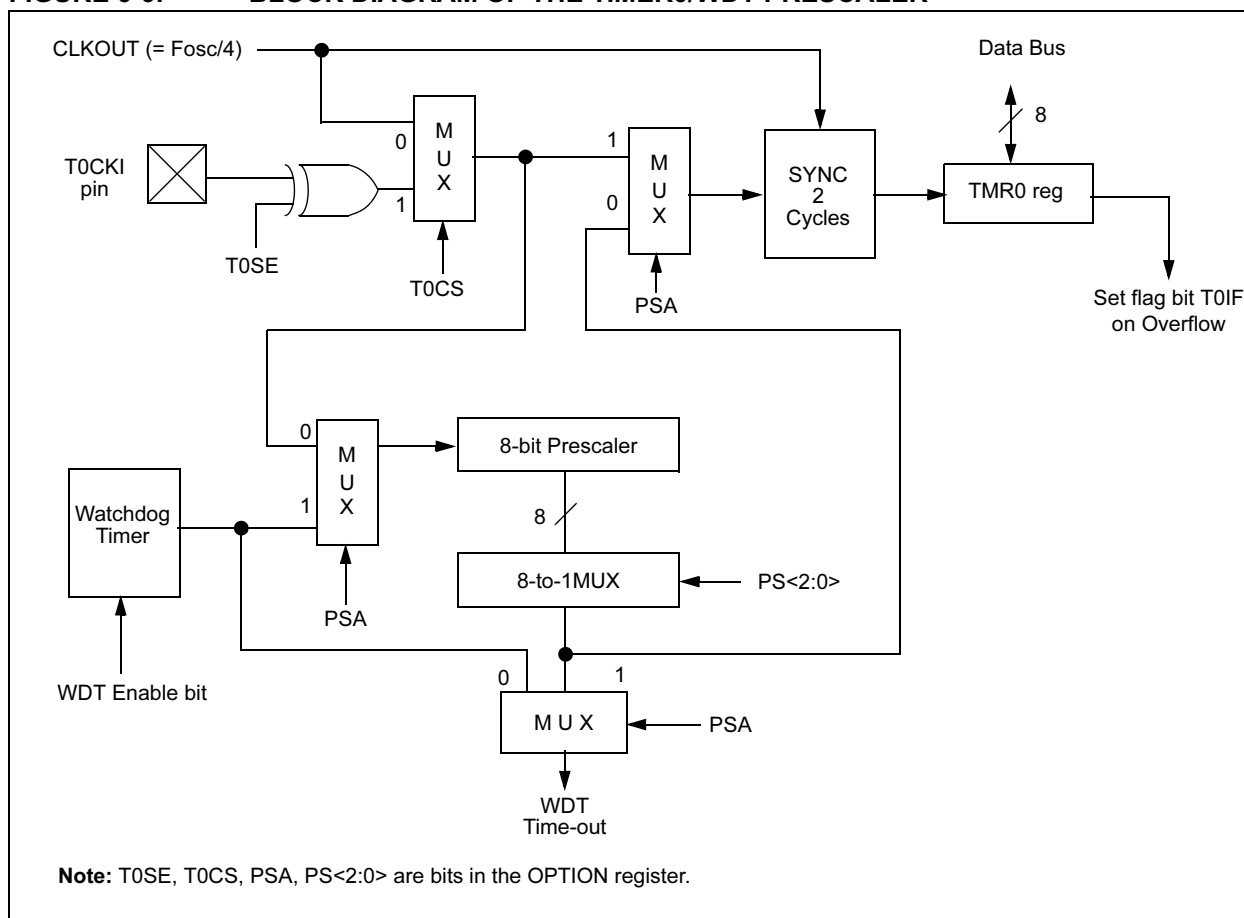
6.3 Prescaler

An 8-bit counter is available as a prescaler for the Timer0 module, or as a postscaler for the Watchdog Timer, respectively (Figure 6-6). For simplicity, this counter is being referred to as “prescaler” throughout this data sheet. Note that there is only one prescaler available which is mutually exclusive between the Timer0 module and the Watchdog Timer. Thus, a prescaler assignment for the Timer0 module means that there is no prescaler for the Watchdog Timer and vice-versa.

The PSA and PS<2:0> bits (OPTION<3:0>) determine the prescaler assignment and prescale ratio.

When assigned to the Timer0 module, all instructions writing to the TMR0 register (e.g., `CLRF 1`, `MOVWF 1`, `BSF 1,x....etc.`) will clear the prescaler. When assigned to WDT, a `CLRWDT` instruction will clear the prescaler along with the Watchdog Timer. The prescaler is not readable or writable.

FIGURE 6-6: BLOCK DIAGRAM OF THE TIMER0/WDT PRESCALER



PIC16C62X

NOTES:

7.5 Comparator Outputs

The comparator outputs are read through the CMCON register. These bits are read only. The comparator outputs may also be directly output to the RA3 and RA4 I/O pins. When the CM<2:0> = 110, multiplexors in the output path of the RA3 and RA4 pins will switch and the output of each pin will be the unsynchronized output of the comparator. The uncertainty of each of the comparators is related to the input offset voltage and the response time given in the specifications. Figure 7-3 shows the comparator output block diagram.

The TRISA bits will still function as an output enable/disable for the RA3 and RA4 pins while in this mode.

2: Analog levels on any pin that is defined as a digital input may cause the input buffer to consume more current than is specified.

The diagram illustrates the logic for the RA3 or RA4 pin multiplexer. It features two D-type flip-flops and several logic gates. The top flip-flop's D input is connected to the output of an inverter that takes 'Bus Data' as input. Its Q output is connected to the 'RD CMCON' input of the bottom flip-flop. The bottom flip-flop's D input is connected to the output of a 3-input AND gate. The inputs to this AND gate are 'Set CMIF Bit', 'FROM OTHER COMPARATOR', and the Q output of the top flip-flop. The Q output of the bottom flip-flop is connected to the 'RD CMCON' input of the top flip-flop. The 'EN' (enable) input of the top flip-flop is connected to the output of an inverter that takes the Q output of the bottom flip-flop as input. The 'EN' input of the bottom flip-flop is connected to the output of an inverter that takes the 'RD CMCON' input as input. The 'CL' (clear) input of the bottom flip-flop is connected to the 'NRESET' input. The output of the multiplexer is connected to the 'PORT PINS' and is also fed back to the 'Bus Data' input of the top flip-flop's D input.

PIC16C62X

EXAMPLE 8-1: VOLTAGE REFERENCE CONFIGURATION

MOVLW	0x02	; 4 Inputs Muxed
MOVWF	CMCON	; to 2 comps.
BSF	STATUS,RP0	; go to Bank 1
MOVLW	0x0F	; RA3-RA0 are
MOVWF	TRISA	; inputs
MOVLW	0xA6	; enable VREF
MOVWF	VRCON	; low range
		; set VR<3:0>=6
BCF	STATUS,RP0	; go to Bank 0
CALL	DELAY10	; 10μs delay

8.2 Voltage Reference Accuracy/Error

The full range of VSS to VDD cannot be realized due to the construction of the module. The transistors on the top and bottom of the resistor ladder network (Figure 8-1) keep VREF from approaching VSS or VDD. The voltage reference is VDD derived and therefore, the VREF output changes with fluctuations in VDD. The tested absolute accuracy of the voltage reference can be found in Table 12-2.

8.3 Operation During SLEEP

When the device wakes up from SLEEP through an interrupt or a Watchdog Timer time-out, the contents of the VRCON register are not affected. To minimize current consumption in SLEEP mode, the voltage reference should be disabled.

8.4 Effects of a RESET

A device RESET disables the voltage reference by clearing bit VREN (VRCON<7>). This reset also disconnects the reference from the RA2 pin by clearing bit VROE (VRCON<6>) and selects the high voltage range by clearing bit VRR (VRCON<5>). The VREF value select bits, VRCON<3:0>, are also cleared.

8.5 Connection Considerations

The voltage reference module operates independently of the comparator module. The output of the reference generator may be connected to the RA2 pin if the TRISA<2> bit is set and the VROE bit, VRCON<6>, is set. Enabling the voltage reference output onto the RA2 pin with an input signal present will increase current consumption. Connecting RA2 as a digital output with VREF enabled will also increase current consumption.

The RA2 pin can be used as a simple D/A output with limited drive capability. Due to the limited drive capability, a buffer must be used in conjunction with the voltage reference output for external connections to VREF. Figure 8-2 shows an example buffering technique.

FIGURE 8-2: VOLTAGE REFERENCE OUTPUT BUFFER EXAMPLE

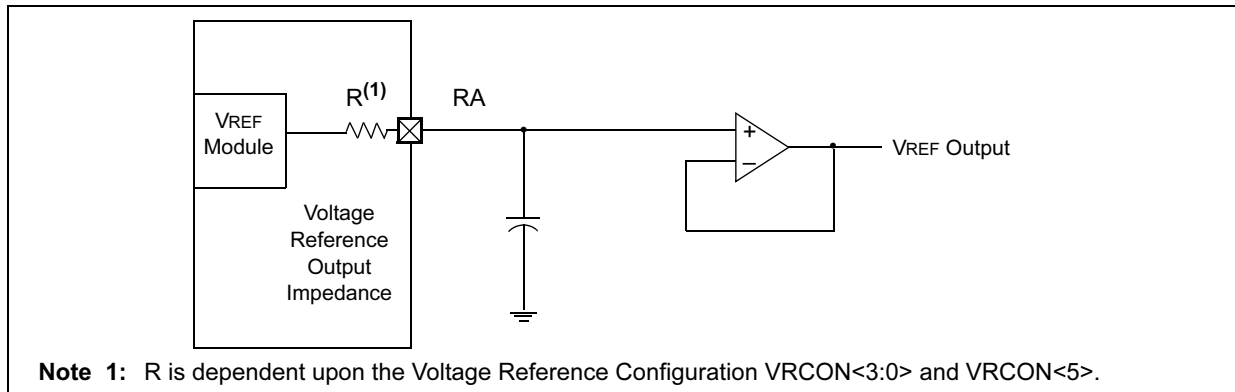


TABLE 8-1: REGISTERS ASSOCIATED WITH VOLTAGE REFERENCE

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
9Fh	VRCON	VREN	VROE	VRR	—	VR3	VR2	VR1	VR0	000- 0000	000- 0000
1Fh	CMCON	C2OUT	C1OUT	—	—	CIS	CM2	CM1	CM0	00-- 0000	00-- 0000
85h	TRISA	—	—	—	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	---1 1111	---1 1111

Note: - = Unimplemented, read as "0"

PIC16C62X

9.4 Power-on Reset (POR), Power-up Timer (PWRT), Oscillator Start-up Timer (OST) and Brown-out Reset (BOR)

9.4.1 POWER-ON RESET (POR)

The on-chip POR circuit holds the chip in RESET until VDD has reached a high enough level for proper operation. To take advantage of the POR, just tie the MCLR pin through a resistor to VDD. This will eliminate external RC components usually needed to create Power-on Reset. A maximum rise time for VDD is required. See Electrical Specifications for details.

The POR circuit does not produce an internal RESET when VDD declines.

When the device starts normal operation (exits the RESET condition), device operating parameters (voltage, frequency, temperature, etc.) must be met to ensure operation. If these conditions are not met, the device must be held in RESET until the operating conditions are met.

For additional information, refer to Application Note AN607, "Power-up Trouble Shooting".

9.4.2 POWER-UP TIMER (PWRT)

The Power-up Timer provides a fixed 72 ms (nominal) time-out on power-up only, from POR or Brown-out Reset. The Power-up Timer operates on an internal RC oscillator. The chip is kept in RESET as long as PWRT is active. The PWRT delay allows the VDD to rise to an acceptable level. A configuration bit, **PWRT**, can disable (if set) or enable (if cleared or programmed) the Power-up Timer. The Power-up Timer should always be enabled when Brown-out Reset is enabled.

The Power-up Time delay will vary from chip-to-chip and due to VDD, temperature and process variation. See DC parameters for details.

9.4.3 OSCILLATOR START-UP TIMER (OST)

The Oscillator Start-Up Timer (OST) provides a 1024 oscillator cycle (from OSC1 input) delay after the PWRT delay is over. This ensures that the crystal oscillator or resonator has started and stabilized.

The OST time-out is invoked only for XT, LP and HS modes and only on Power-on Reset or wake-up from SLEEP.

9.4.4 BROWN-OUT RESET (BOR)

The PIC16C62X members have on-chip Brown-out Reset circuitry. A configuration bit, **BODEN**, can disable (if clear/programmed) or enable (if set) the Brown-out Reset circuitry. If VDD falls below 4.0V refer to VBOR parameter D005 (VBOR) for greater than parameter (TBOR) in Table 12-5. The brown-out situation will RESET the chip. A RESET won't occur if VDD falls below 4.0V for less than parameter (TBOR).

On any RESET (Power-on, Brown-out, Watchdog, etc.) the chip will remain in RESET until VDD rises above BVDD. The Power-up Timer will now be invoked and will keep the chip in RESET an additional 72 ms.

If VDD drops below BVDD while the Power-up Timer is running, the chip will go back into a Brown-out Reset and the Power-up Timer will be re-initialized. Once VDD rises above BVDD, the Power-Up Timer will execute a 72 ms RESET. The Power-up Timer should always be enabled when Brown-out Reset is enabled. Figure 9-7 shows typical Brown-out situations.

FIGURE 9-7: BROWN-OUT SITUATIONS

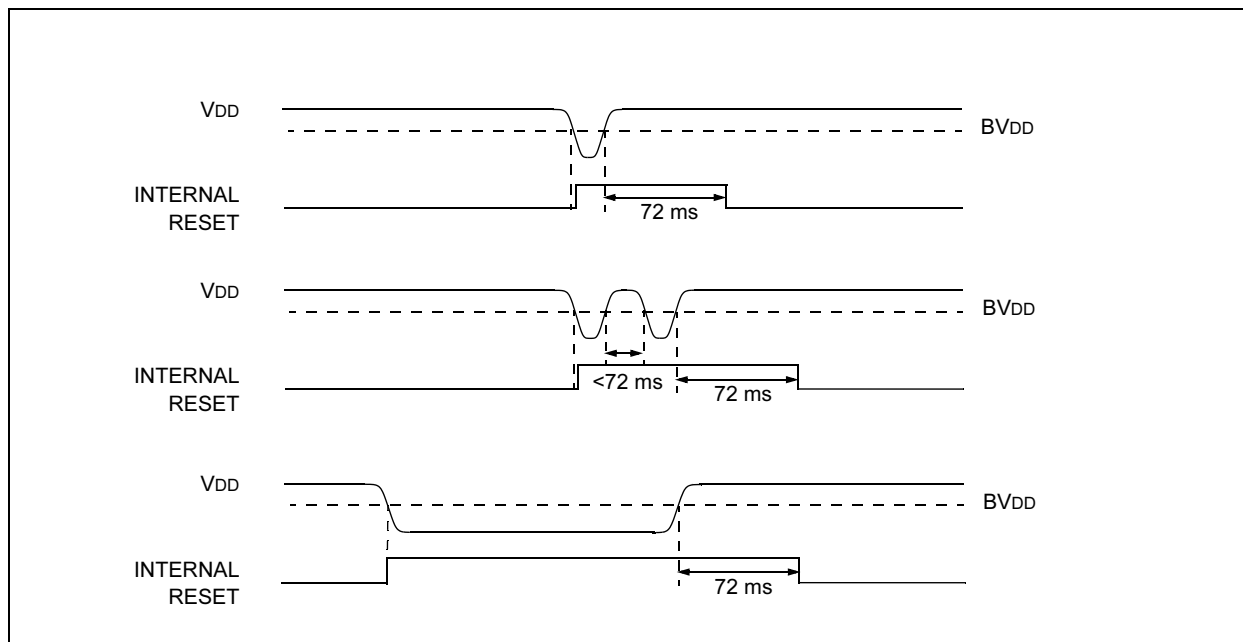


TABLE 9-6: SUMMARY OF INTERRUPT REGISTERS

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR Reset	Value on all other RESETS ⁽¹⁾
0Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	—	CMIF	—	—	—	—	—	—	-0-- ----	-0-- ----
8Ch	PIE1	—	CMIE	—	—	—	—	—	—	-0-- ----	-0-- ----

Note 1: Other (non Power-up) Resets include MCLR Reset, Brown-out Reset and Watchdog Timer Reset during normal operation.

9.6 Context Saving During Interrupts

During an interrupt, only the return PC value is saved on the stack. Typically, users may wish to save key registers during an interrupt (e.g., W register and STATUS register). This will have to be implemented in software.

Example 9-3 stores and restores the STATUS and W registers. The user register, W_TEMP, must be defined in both banks and must be defined at the same offset from the bank base address (i.e., W_TEMP is defined at 0x20 in Bank 0 and it must also be defined at 0xA0 in Bank 1). The user register, STATUS_TEMP, must be defined in Bank 0. The Example 9-3:

- Stores the W register
- Stores the STATUS register in Bank 0
- Executes the ISR code
- Restores the STATUS (and bank select bit register)
- Restores the W register

EXAMPLE 9-3: SAVING THE STATUS AND W REGISTERS IN RAM

```

MOVWF  W_TEMP      ;copy W to temp register,
                    ;could be in either bank
SWAPF  STATUS,W     ;swap status to be saved
                    ;into W
BCF    STATUS,RP0    ;change to bank 0 regardless
                    ;of current bank
MOVWF  STATUS_TEMP  ;save status to bank 0
                    ;register
:
:    (ISR)
:
SWAPF  STATUS_TEMP, ;swap STATUS_TEMP register
W      ;into W, sets bank to origi-
      ;nal
      ;state
MOVWF  STATUS       ;move W into STATUS register
SWAPF  W_TEMP,F     ;swap W_TEMP
SWAPF  W_TEMP,W     ;swap W_TEMP into W

```

PIC16C62X

BCF		Bit Clear f						
Syntax:	[<i>label</i>] BCF f,b							
Operands:	$0 \leq f \leq 127$ $0 \leq b \leq 7$							
Operation:	$0 \rightarrow (f)$							
Status Affected:	None							
Encoding:	<table border="1"><tr><td>01</td><td>00bb</td><td>bfff</td><td>ffff</td></tr></table>				01	00bb	bfff	ffff
01	00bb	bfff	ffff					
Description:	Bit 'b' in register 'f' is cleared.							
Words:	1							
Cycles:	1							
Example	<div>BCF FLAG_REG, 7</div> <div>Before Instruction</div> <div>FLAG_REG = 0xC7</div> <div>After Instruction</div> <div>FLAG_REG = 0x47</div>							

BSF		Bit Set f						
Syntax:	[<i>label</i>] BSF f,b							
Operands:	$0 \leq f \leq 127$ $0 \leq b \leq 7$							
Operation:	$1 \rightarrow (f)$							
Status Affected:	None							
Encoding:	<table border="1"><tr><td>01</td><td>01bb</td><td>bfff</td><td>ffff</td></tr></table>				01	01bb	bfff	ffff
01	01bb	bfff	ffff					
Description:	Bit 'b' in register 'f' is set.							
Words:	1							
Cycles:	1							
Example	<div>BSFFLAG_REG,7</div> <div>Before Instruction</div> <div>FLAG_REG = 0x0A</div> <div>After Instruction</div> <div>FLAG_REG = 0x8A</div>							

BTFSC		Bit Test, Skip if Clear							
Syntax:	[<i>label</i>] BTFSC f,b								
Operands:	$0 \leq f \leq 127$ $0 \leq b \leq 7$								
Operation:	skip if (f) = 0								
Status Affected:	None								
Encoding:	<table><tr><td>01</td><td>10bb</td><td>bfff</td><td>ffff</td></tr></table>					01	10bb	bfff	ffff
01	10bb	bfff	ffff						
Description:	<p>If bit 'b' in register 'f' is '0', then the next instruction is skipped.</p> <p>If bit 'b' is '0', then the next instruction fetched during the current instruction execution is discarded, and a NOP is executed instead, making this a two-cycle instruction.</p>								
Words:	1								
Cycles:	1(2)								
Example	HERE	BTFSC	FLAG,1						
	FALSE	GOTO	PROCESS_CO						
	TRUE	.	DE						
		.							
		.							

Before Instruction
PC = address HERE

After Instruction
if FLAG<1> = 0,
PC = address TRUE
if FLAG<1>=1,
PC = address FALSE

PIC16C62X

INCFSZ Increment f, Skip if 0

Syntax: `[label] INCFSZ f,d`

Operands: $0 \leq f \leq 127$
 $d \in [0,1]$

Operation: $(f) + 1 \rightarrow (\text{dest})$, skip if result = 0

Status Affected: None

Encoding:

00	1111	dfff	ffff
----	------	------	------

Description: The contents of register 'f' are incremented. If 'd' is 0 the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.
 If the result is 0, the next instruction, which is already fetched, is discarded. A NOP is executed instead making it a two-cycle instruction.

Words: 1

Cycles: 1(2)

Example

```

HERE      INCFSZ    CNT, 1
          GOTO      LOOP
CONTINUE  .
          .
          .
  
```

Before Instruction
 PC = address HERE

After Instruction
 CNT = CNT + 1
 if CNT= 0,
 PC = address CONTINUE
 if CNT≠ 0,
 PC = address HERE +1

IORLW Inclusive OR Literal with W

Syntax: `[label] IORLW k`

Operands: $0 \leq k \leq 255$

Operation: $(W) .OR. k \rightarrow (W)$

Status Affected: Z

Encoding:

11	1000	kkkk	kkkk
----	------	------	------

Description: The contents of the W register is OR'ed with the eight bit literal 'k'. The result is placed in the W register.

Words: 1

Cycles: 1

Example

```

IORLW    0x35
  
```

Before Instruction
 W = 0x9A

After Instruction
 W = 0xBF
 Z = 1

IORWF Inclusive OR W with f

Syntax: `[label] IORWF f,d`

Operands: $0 \leq f \leq 127$
 $d \in [0,1]$

Operation: $(W) .OR. (f) \rightarrow (\text{dest})$

Status Affected: \bar{Z}

Encoding:

00	0100	dfff	ffff
----	------	------	------

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'.

Words: 1

Cycles: 1

Example

```

IORWF          RESULT, 0
  
```

Before Instruction
 RESULT = 0x13
 W = 0x91

After Instruction
 RESULT = 0x13
 W = 0x93
 Z = 1

MOVLW Move Literal to W

Syntax: `[label] MOVLW k`

Operands: $0 \leq k \leq 255$

Operation: $k \rightarrow (W)$

Status Affected: None

Encoding:

11	00xx	kkkk	kkkk
----	------	------	------

Description: The eight bit literal 'k' is loaded into W register. The don't cares will assemble as 0's.

Words: 1

Cycles: 1

Example

```

MOVLW    0x5A
  
```

After Instruction
 W = 0x5A

11.3 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.

11.4 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/librarian 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

11.5 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 capabilities, 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.

11.6 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 its 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

11.7 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.

11.8 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.

PIC16C62X

FIGURE 12-5: PIC16LC620A/LC621A/LC622A VOLTAGE-FREQUENCY GRAPH, $-40^{\circ}\text{C} \leq T_A \leq 0^{\circ}\text{C}$

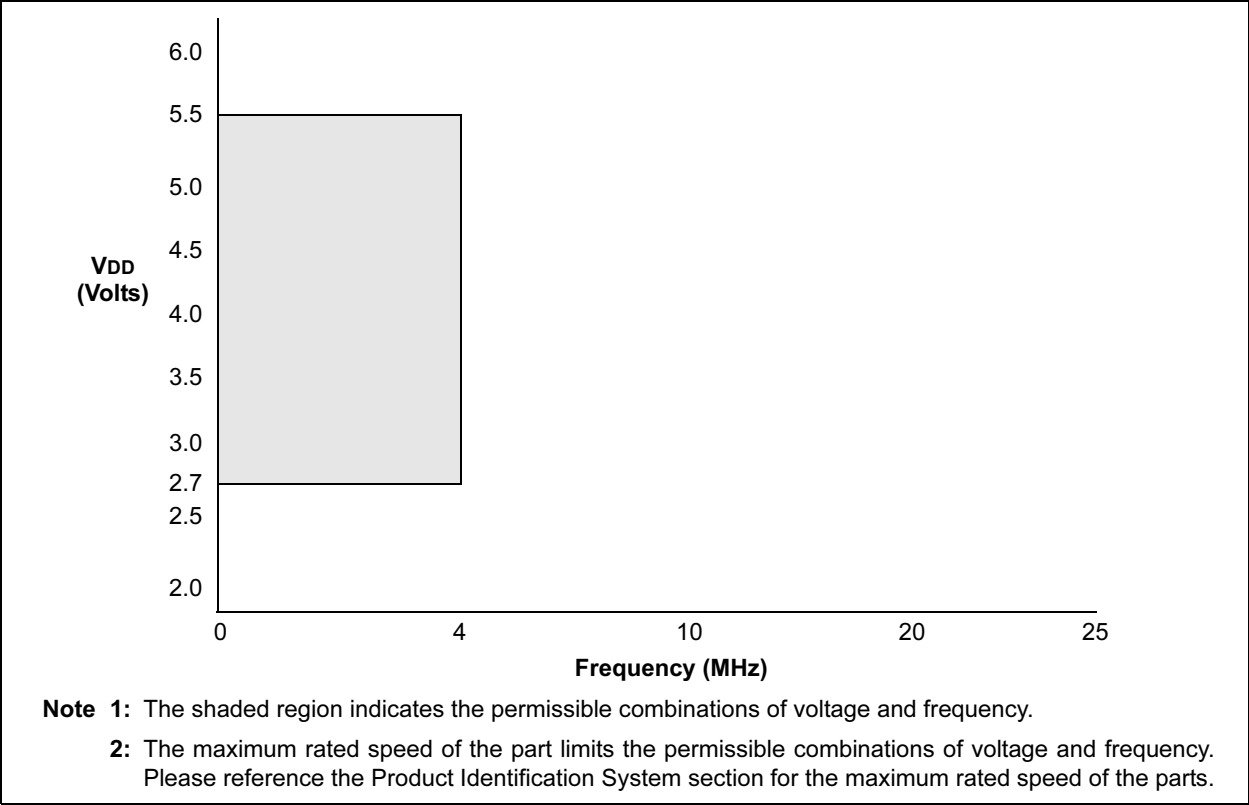


FIGURE 12-6: PIC16LC620A/LC621A/LC622A VOLTAGE-FREQUENCY GRAPH, $0^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$

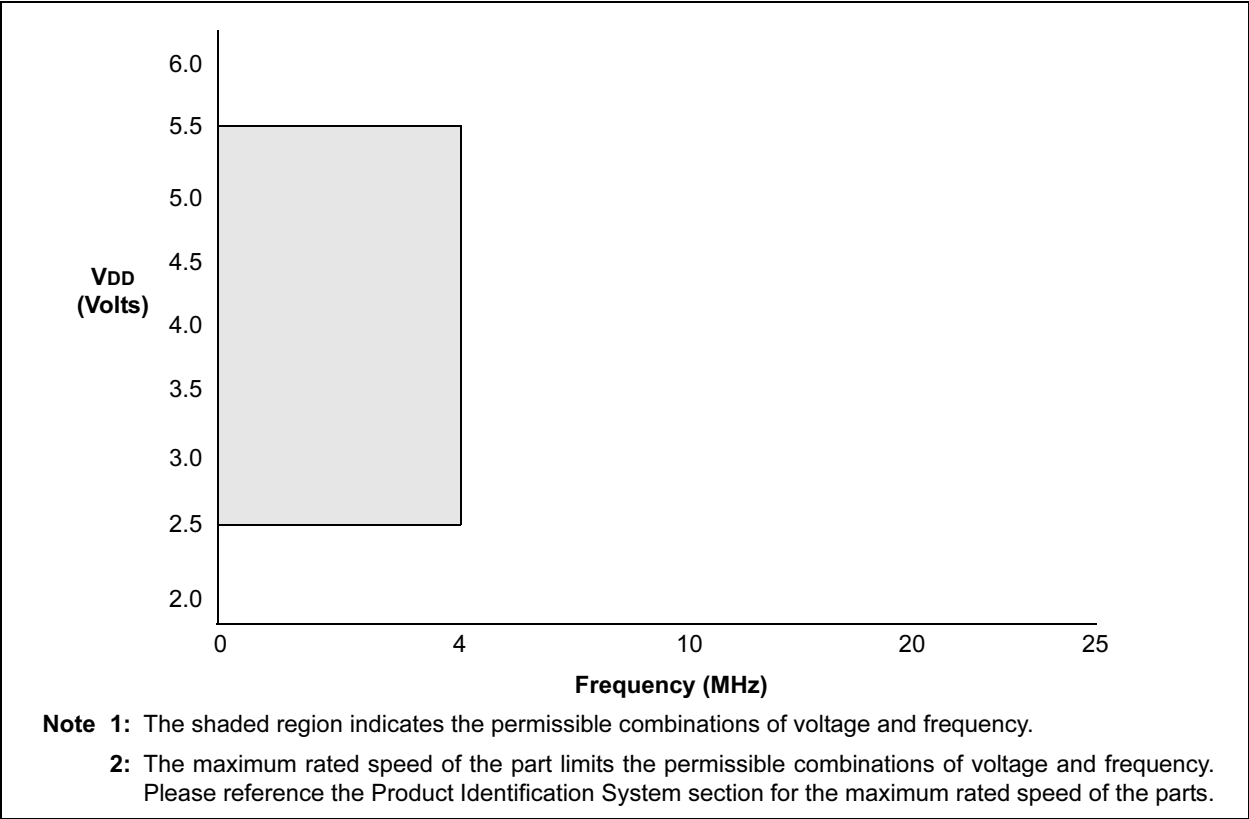
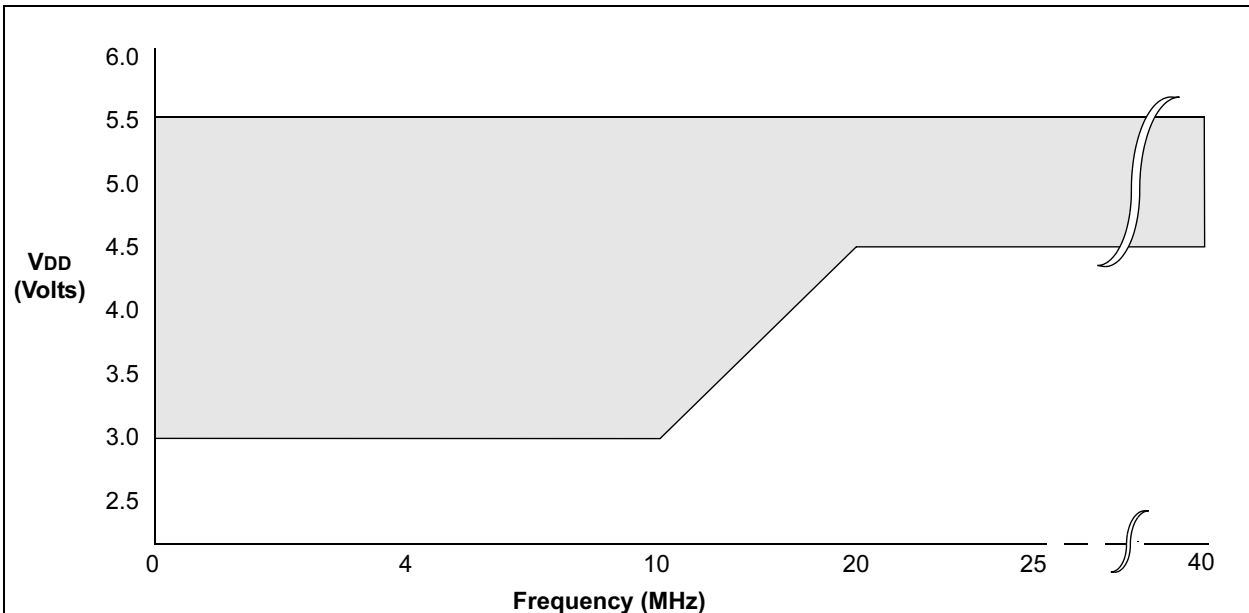


FIGURE 12-10: PIC16C620A/C621A/C622A/CR620A - 40 VOLTAGE-FREQUENCY GRAPH, $0^{\circ}\text{C} \leq T_A \leq +70^{\circ}\text{C}$



- Note 1:** The shaded region indicates the permissible combinations of voltage and frequency.
- Note 2:** The maximum rated speed of the part limits the permissible combinations of voltage and frequency. Please reference the Product Identification System section for the maximum rated speed of the parts.
- Note 3:** Operation between 20 to 40 MHz requires the following:
- VDD between 4.5V. and 5.5V
 - OSC1 externally driven
 - OSC2 not connected
 - HS mode
 - Commercial temperatures
- Devices qualified for 40 MHz operation have -40 designation (ex: PIC16C620A-40/P).

PIC16C62X

12.4 DC Characteristics: PIC16C62X/C62XA/CR62XA (Commercial, Industrial, Extended) PIC16LC62X/LC62XA/LCR62XA (Commercial, Industrial, Extended) (CONT.)

PIC16C62X/C62XA/CR62XA			Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial and $0^{\circ}\text{C} \leq T_A \leq +70^{\circ}\text{C}$ for commercial and $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for extended				
PIC16LC62X/LC62XA/LCR62XA			Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial and $0^{\circ}\text{C} \leq T_A \leq +70^{\circ}\text{C}$ for commercial and $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for extended				
Param. No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
D080	VOL	Output Low Voltage I/O ports	—	—	0.6	V	IOL = 8.5 mA, VDD = 4.5V, -40° to $+85^{\circ}\text{C}$
			—	—	0.6	V	IOL = 7.0 mA, VDD = 4.5V, $+125^{\circ}\text{C}$
D083		OSC2/CLKOUT (RC only)	—	—	0.6	V	IOL = 1.6 mA, VDD = 4.5V, -40° to $+85^{\circ}\text{C}$
			—	—	0.6	V	IOL = 1.2 mA, VDD = 4.5V, $+125^{\circ}\text{C}$
D090	VOH	Output High Voltage ⁽³⁾ I/O ports (Except RA4)	VDD-0.7	—	—	V	IOH = -3.0 mA, VDD = 4.5V, -40° to $+85^{\circ}\text{C}$
			VDD-0.7	—	—	V	IOH = -2.5 mA, VDD = 4.5V, $+125^{\circ}\text{C}$
D092		OSC2/CLKOUT (RC only)	VDD-0.7	—	—	V	IOH = -1.3 mA, VDD = 4.5V, -40° to $+85^{\circ}\text{C}$
			VDD-0.7	—	—	V	IOH = -1.0 mA, VDD = 4.5V, $+125^{\circ}\text{C}$
D090	VOH	Output High Voltage ⁽³⁾ I/O ports (Except RA4)	VDD-0.7	—	—	V	IOH = -3.0 mA, VDD = 4.5V, -40° to $+85^{\circ}\text{C}$
			VDD-0.7	—	—	V	IOH = -2.5 mA, VDD = 4.5V, $+125^{\circ}\text{C}$
D092		OSC2/CLKOUT (RC only)	VDD-0.7	—	—	V	IOH = -1.3 mA, VDD = 4.5V, -40° to $+85^{\circ}\text{C}$
			VDD-0.7	—	—	V	IOH = -1.0 mA, VDD = 4.5V, $+125^{\circ}\text{C}$
D150	VOD	Open-Drain High Voltage			10 8.5*	V	RA4 pin PIC16C62X, PIC16LC62X RA4 pin PIC16C62XA, PIC16LC62XA, PIC16CR62XA, PIC16LCR62XA
D150	VOD	Open-Drain High Voltage			10 8.5*	V	RA4 pin PIC16C62X, PIC16LC62X RA4 pin PIC16C62XA, PIC16LC62XA, PIC16CR62XA, PIC16LCR62XA
D100	COSC 2	Capacitive Loading Specs on Output Pins 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 RC mode)			50	pF	
D100	COSC 2	Capacitive Loading Specs on Output Pins 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 RC mode)			50	pF	

* 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:** In RC oscillator configuration, the OSC1 pin is a Schmitt Trigger input. It is not recommended that the PIC16C62X(A) be driven with external clock in RC 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.

PIC16C62X

12.5 DC CHARACTERISTICS: PIC16C620A/C621A/C622A-40⁽⁷⁾ (Commercial) PIC16CR620A-40⁽⁷⁾ (Commercial)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated)				
			Operating temperature 0°C ≤ TA ≤ +70°C for commercial				
Param No.	Sym	Characteristic	Min	Typ†	Max	Unit	Conditions
D030	V _{IL}	Input Low Voltage I/O ports with TTL buffer	V _{SS}	—	0.8V 0.15V _{DD}	V	V _{DD} = 4.5V to 5.5V, otherwise
D031		with Schmitt Trigger input	V _{SS}	—	0.2V _{DD}	V	(Note 1)
D032		$\overline{\text{MCLR}}$, RA4/T0CKI, OSC1 (in RC mode)	V _{SS}	—	0.2V _{DD}	V	
D033		OSC1 (in XT and HS) OSC1 (in LP)	V _{SS} V _{SS}	— —	0.3V _{DD} 0.6V _{DD} - 1.0	V V	
D040	V _{IH}	Input High Voltage I/O ports with TTL buffer	2.0V 0.25 V _{DD} + 0.8	—	V _{DD} V _{DD}	V	V _{DD} = 4.5V to 5.5V, otherwise
D041		with Schmitt Trigger input	0.8 V _{DD}	—	V _{DD}	V	(Note 1)
D042		$\overline{\text{MCLR}}$ RA4/T0CKI	0.8 V _{DD}	—	V _{DD}	V	
D043		OSC1 (XT, HS and LP)	0.7 V _{DD}	—	V _{DD}	V	
D043A		OSC1 (in RC mode)	0.9 V _{DD}	—			(Note 1)
D070	IP _{URB}	PORTB Weak Pull-up Current	50	200	400	μA	V _{DD} = 5.0V, V _{PIN} = V _{SS}
D060	I _{IL}	Input Leakage Current ^(2, 3) I/O ports (except PORTA)	—	—	±1.0	μA	V _{SS} ≤ V _{PIN} ≤ V _{DD} , pin at hi-impedance
D061		PORTA	—	—	±0.5	μA	V _{SS} ≤ V _{PIN} ≤ V _{DD} , pin at hi-impedance
D063		RA4/T0CKI	—	—	±1.0	μA	V _{SS} ≤ V _{PIN} ≤ V _{DD}
		OSC1, $\overline{\text{MCLR}}$	—	—	±5.0	μA	V _{SS} ≤ V _{PIN} ≤ V _{DD} , XT, HS and LP osc configuration
D080	V _{OL}	Output Low Voltage I/O ports	—	—	0.6	V	I _{OL} = 8.5 mA, V _{DD} = 4.5V, -40° to +85°C
			—	—	0.6	V	I _{OL} = 7.0 mA, V _{DD} = 4.5V, +125°C
D083		OSC2/CLKOUT (RC only)	—	—	0.6	V	I _{OL} = 1.6 mA, V _{DD} = 4.5V, -40° to +85°C
			—	—	0.6	V	I _{OL} = 1.2 mA, V _{DD} = 4.5V, +125°C
D090	V _{OH}	Output High Voltage ⁽³⁾ I/O ports (except RA4)	V _{DD} -0.7 V _{DD} -0.7	— —	— —	V V	I _{OH} = -3.0 mA, V _{DD} = 4.5V, -40° to +85°C I _{OH} = -2.5 mA, V _{DD} = 4.5V, +125°C
D092		OSC2/CLKOUT (RC only)	V _{DD} -0.7 V _{DD} -0.7	— —	— —	V V	I _{OH} = -1.3 mA, V _{DD} = 4.5V, -40° to +85°C I _{OH} = -1.0 mA, V _{DD} = 4.5V, +125°C
*D150	V _{OD}	Open Drain High Voltage			8.5	V	RA4 pin
D100	C _{osc2}	Capacitive Loading Specs on Output Pins OSC2 pin			15	pF	In XT, HS and LP modes when external clock used to drive OSC1.
D101	C _{io}	All I/O pins/OSC2 (in RC 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: 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 V_{DD}, $\overline{\text{MCLR}}$ = V_{DD}; 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 V_{DD} or V_{SS}.

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

5: 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.

6: Commercial temperature range only.

7: See Section 12.1 and Section 12.3 for 16C62X and 16CR62X devices for operation between 20 MHz and 40 MHz for valid modified characteristics.

READER RESPONSE

Please list the following information, and use this outline to provide us with your comments about this document.

Application (optional):

Would you like a reply? Y N

Device: PIC16C62X Literature Number: DS30235J

Questions:

1. What are the best features of this document?

2. How does this document meet your hardware and software development needs?

3. Do you find the organization of this document easy to follow? If not, why?

4. What additions to the document do you think would enhance the structure and subject?

5. What deletions from the document could be made without affecting the overall usefulness?

6. Is there any incorrect or misleading information (what and where)?

7. How would you improve this document?
