



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	20MHz
Connectivity	-
Peripherals	Brown-out Detect/Reset, POR, WDT
Number of I/O	13
Program Memory Size	896B (512 x 14)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	96 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	18-SOIC (0.295", 7.50mm Width)
Supplier Device Package	18-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16c620a-20e-so

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

1.0 GENERAL DESCRIPTION

The PIC16C62X devices are 18 and 20-Pin ROM/ EPROM-based members of the versatile PICmicro[®] family of low cost, high performance, CMOS, fullystatic, 8-bit microcontrollers.

All PICmicro microcontrollers employ an advanced RISC architecture. The PIC16C62X devices have enhanced core features, eight-level deep stack, and multiple internal and external interrupt sources. The separate instruction and data buses of the Harvard architecture allow a 14-bit wide instruction word with the separate 8-bit wide data. The two-stage instruction pipeline allows all instructions to execute in a single cycle, except for program branches (which require two cycles). A total of 35 instructions (reduced instruction set) are available. Additionally, a large register set gives some of the architectural innovations used to achieve a very high performance.

PIC16C62X microcontrollers typically achieve a 2:1 code compression and a 4:1 speed improvement over other 8-bit microcontrollers in their class.

The PIC16C620A, PIC16C621A and PIC16CR620A have 96 bytes of RAM. The PIC16C622(A) has 128 bytes of RAM. Each device has 13 I/O pins and an 8-bit timer/counter with an 8-bit programmable prescaler. In addition, the PIC16C62X adds two analog comparators with a programmable on-chip voltage reference module. The comparator module is ideally suited for applications requiring a low cost analog interface (e.g., battery chargers, threshold detectors, white goods controllers, etc).

PIC16C62X devices have special features to reduce external components, thus reducing system cost, enhancing system reliability and reducing power consumption. There are four oscillator options, of which the single pin RC oscillator provides a low cost solution, the LP oscillator minimizes power consumption, XT is a standard crystal, and the HS is for High Speed crystals. The SLEEP (Power-down) mode offers power savings. The user can wake-up the chip from SLEEP through several external and internal interrupts and RESET.

A highly reliable Watchdog Timer with its own on-chip RC oscillator provides protection against software lock- up.

A UV-erasable CERDIP-packaged version is ideal for code development while the cost effective One-Time-Programmable (OTP) version is suitable for production in any volume.

Table 1-1 shows the features of the PIC16C62X midrange microcontroller families.

A simplified block diagram of the PIC16C62X is shown in Figure 3-1.

The PIC16C62X series fits perfectly in applications ranging from battery chargers to low power remote sensors. The EPROM technology makes

customization of application programs (detection levels, pulse generation, timers, etc.) extremely fast and convenient. The small footprint packages make this microcontroller series perfect for all applications with space limitations. Low cost, low power, high performance, ease of use and I/O flexibility make the PIC16C62X very versatile.

1.1 Family and Upward Compatibility

Those users familiar with the PIC16C5X family of microcontrollers will realize that this is an enhanced version of the PIC16C5X architecture. Please refer to Appendix A for a detailed list of enhancements. Code written for the PIC16C5X can be easily ported to PIC16C62X family of devices (Appendix B). The PIC16C62X family fills the niche for users wanting to migrate up from the PIC16C5X family and not needing various peripheral features of other members of the PIC16XX mid-range microcontroller family.

1.2 Development Support

The PIC16C62X family is supported by a full-featured macro assembler, a software simulator, an in-circuit emulator, a low cost development programmer and a full-featured programmer. Third Party "C" compilers are also available.

FIGURE 3-1: BLOCK DIAGRAM



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	Unimpleme	nted: Read	d as '0'					
bit 6	CMIE: Com	parator Inte	errupt Enable	e bit				
	 1 = Enables the Comparator interrupt 0 = Disables the Comparator interrupt 							
bit 5-0	Unimpleme	nted: Read	d as '0'					
	Legend:							
	R = Readab	le bit	W = W	ritable bit	U = Unim	plemented	bit, read as '	0'
	- n = Value a	at POR	'1' = Bi	it is set	'0' = Bit i	s cleared	x = Bit is u	nknown

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)

_I\ 4 -J.	FINTNEO										
	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0			
	—	CMIF		_	—	_	—	_			
	bit 7							bit 0			
bit 7	Unimplem	ented: Rea	d as '0'								
bit 6	CMIF: Com	CMIF: Comparator Interrupt Flag bit									
	1 = Compa	rator input h	nas changed								
	0 = Comparator input has not changed										
bit 5-0	Unimplemented: Read as '0'										
	Legend:										
	R = Reada	ble bit	W = W	ritable bit	U = Unim	plemented	bit, read as '	0'			
	- n = Value	at POR	'1' = B	it is set	'0' = Bit is	s cleared	x = Bit is u	nknown			

4.3 PCL and PCLATH

The program counter (PC) is 13-bits wide. The low byte comes from the PCL register, which is a readable and writable register. The high byte (PC<12:8>) is not directly readable or writable and comes from PCLATH. On any RESET, the PC is cleared. Figure 4-8 shows the two situations for the loading of the PC. The upper example in the figure shows how the PC is loaded on a write to PCL (PCLATH<4:0> \rightarrow PCH). The lower example in the figure shows how the PC is loaded during a CALL or GOTO instruction (PCLATH<4:3> \rightarrow PCH).

FIGURE 4-8: LOADING OF PC IN DIFFERENT SITUATIONS



4.3.1 COMPUTED GOTO

A computed GOTO is accomplished by adding an offset to the program counter (ADDWF PCL). When doing a table read using a computed GOTO method, care should be exercised if the table location crosses a PCL memory boundary (each 256 byte block). Refer to the application note, *"Implementing a Table Read"* (AN556).

4.3.2 STACK

The PIC16C62X family has an 8-level deep x 13-bit wide hardware stack (Figure 4-2 and Figure 4-3). The stack space is not part of either program or data space and the stack pointer is not readable or writable. The PC is PUSHed onto the stack when a CALL instruction is executed or an interrupt causes a branch. The stack is POPed in the event of a RETURN, RETLW or a RETFIE instruction execution. PCLATH is not affected by a PUSH or POP operation.

The stack operates as a circular buffer. This means that after the stack has been PUSHed eight times, the ninth push overwrites the value that was stored from the first push. The tenth push overwrites the second push (and so on).

- Note 1: There are no STATUS bits to indicate stack overflow or stack underflow conditions.
 - 2: There are no instructions/mnemonics called PUSH or POP. These are actions that occur from the execution of the CALL, RETURN, RETLW and RETFIE instructions, or the vectoring to an interrupt address.

5.2 PORTB and TRISB Registers

PORTB is an 8-bit wide, bi-directional port. The corresponding data direction register is TRISB. A '1' in the TRISB register puts the corresponding output driver in a High Impedance mode. A '0' in the TRISB register puts the contents of the output latch on the selected pin(s).

Reading PORTB 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.

Each of the PORTB pins has a weak internal pull-up ($\approx 200 \ \mu A \ typical$). A single control bit can turn on all the pull-ups. This is done by clearing the RBPU (OPTION<7>) bit. The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on Power-on Reset.

Four of PORTB's pins, RB<7:4>, have an interrupt on change feature. Only pins configured as inputs can cause this interrupt to occur (e.g., any RB<7:4> pin configured as an output is excluded from the interrupt on change comparison). The input pins (of RB<7:4>) are compared with the old value latched on the last read of PORTB. The "mismatch" outputs of RB<7:4> are OR'ed together to generate the RBIF interrupt (flag latched in INTCON<0>).

FIGURE 5-5: BLOCK DIAGRAM OF RB<7:4> PINS



This interrupt can wake the device from SLEEP. The user, in the interrupt service routine, can clear the interrupt in the following manner:

- a) Any read or write of PORTB. This will end the mismatch condition.
- b) Clear flag bit RBIF.

A mismatch condition will continue to set flag bit RBIF. Reading PORTB will end the mismatch condition and allow flag bit RBIF to be cleared.

This interrupt on mismatch feature, together with software configurable pull-ups on these four pins allow easy interface to a key pad and make it possible for wake-up on key-depression. (See AN552, "Implementing Wake-Up on Key Strokes.)

Note:	If a change on the I/O pin should occur					
	when the read operation is being executed					
	(start of the Q2 cycle), then the RBIF inter-					
	rupt flag may not get set.					

The interrupt-on-change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt on change feature. Polling of PORTB is not recommended while using the interrupt-on-change feature.





6.3.1 SWITCHING PRESCALER ASSIGNMENT

The prescaler assignment is fully under software control (i.e., it can be changed "on-the-fly" during program execution). To avoid an unintended device RESET, the following instruction sequence (Example 6-1) must be executed when changing the prescaler assignment from Timer0 to WDT.)

EXAMPLE 6-1: CHANGING PRESCALER (TIMER0→WDT)

	-	
1.BCF	STATUS, RPO	;Skip if already in ;Bank 0
2.CLRWDT		;Clear WDT
3.CLRF	TMR0	;Clear TMR0 & Prescaler
4.BSF	STATUS, RPO	;Bank 1
5.MOVLW	'00101111'b;	;These 3 lines (5, 6, 7)
6.MOVWF	OPTION	;are required only if ;desired PS<2:0> are
7.CLRWDT		;000 or 001
8.MOVLW	'00101xxx'b	;Set Postscaler to
9.MOVWF	OPTION	;desired WDT rate
10.BCF	STATUS, RPO	;Return to Bank 0

To change prescaler from the WDT to the TMR0 module, use the sequence shown in Example 6-2. This precaution must be taken even if the WDT is disabled.

EXAMPLE 6-2:

CHANGING PRESCALER (WDT→TIMER0)

	•	
CLRWDT		;Clear WDT and
		;prescaler
BSF	STATUS, RPO	
MOVLW	b'xxxx0xxx'	;Select TMR0, new
		prescale value and
		, plock gourgo
		,CIOCK SOULCE
MOVWF	OPTION REG	
BCF	STATUS, RPO	

TABLE 6-1: REGISTERS ASSOCIATED WITH TIMER0

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
01h	TMR0	Timer0	module regi	ister						XXXX XXXX	uuuu uuuu
0Bh/8Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
81h	OPTION	RBPU	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
85h	TRISA	_	_		TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	1 1111	1 1111

Legend: — = Unimplemented locations, read as '0', u = unchanged, x = unknown

Note: Shaded bits are not used by TMR0 module.

7.6 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 read or write 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.

7.7 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

Vdd ∆Vt = 0.6V RIC Rs < 10K Δικ **I**LEAKAGE CPIN VT = 0.6V ±500 nA 5 pF Vss Input Capacitance Legend CPIN = Threshold Voltage Vт = Leakage Current at the pin due to various junctions ILEAKAGE = = Interconnect Resistance RIC Rs = Source Impedance Analog Voltage VA =

FIGURE 7-4: ANALOG INPUT MODEL

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.

7.8 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, CM<2:0> = 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.

7.9 Analog Input Connection Considerations

A simplified circuit for an analog input is shown in Figure 7-4. 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 $10 \ k\Omega$ 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.

9.2 Oscillator Configurations

9.2.1 OSCILLATOR TYPES

The PIC16C62X devices can be operated in four different oscillator options. The user can program two configuration bits (FOSC1 and FOSC0) to select one of these four modes:

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

9.2.2 CRYSTAL OSCILLATOR / CERAMIC RESONATORS

In XT, LP or HS modes, a crystal or ceramic resonator is connected to the OSC1 and OSC2 pins to establish oscillation (Figure 9-1). The PIC16C62X oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturers specifications. When in XT, LP or HS modes, the device can have an external clock source to drive the OSC1 pin (Figure 9-2).

FIGURE 9-1: CRYSTAL OPERATION (OR CERAMIC RESONATOR) (HS, XT OR LP OSC CONFIGURATION)



See Table 9-1 and Table 9-2 for recommended values of C1 and C2.

Note: A series resistor may be required for AT strip cut crystals.

FIGURE 9-2: EXTERNAL CLOCK INPUT OPERATION (HS, XT OR LP OSC



TABLE 9-1:CAPACITOR SELECTION FOR
CERAMIC RESONATORS

R	anges Chara	~[]			
Mode	Freq	OSC1(C1)	OSC2(C2)		
ХТ	455 kHz 2.0 MHz 4.0 MHz	22 - 100 pF 15 - 68 pF 15 - 68 pF	82 - 100 pF 15 - 68 pF 15 - 68 pF		
HS	8.0 MHz 16.0 MHz 🔨	10-68 bF 10-22 pF	10 - 68 pF 10 - 22 pF		
Higher capacitance increases the stability of the oscil- lator bur also increases the start-up time. These welloes are for design guidance only. Since each vesonator has its own characteristics, the user should consult the resonator manufacturer for appropriate values of external components					

TABLE 9-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR

Mode	Freq	OSC1(C1)	OSC2(C2)		
LP	32 kHz	68 - 100 pF	68 - 100 pF		
	200 kHz	15 - 30 pF	15 - 30 pF		
хт	100 kHz	68 - 150 pF	150 - 300 pF		
	2 MHz	15 - 30 pF	15 - 30 pF		
	4 MHz	15 - 30 pF	15 - 30 pF		
HS	8 MHz	15-30 pF	^V 15 - 30 pF		
	10 MHz	15-30 pF	15 - 30 pF		
	20 MHz 🔨	15-30 pF	15 - 30 pF		
Higher capacitance increases the stability of the oscillator but also increases the start-up time. These values are for design guidance only. Rs may be required in HS mode as well as XT mode to avoid overdriving crystals with low drive level specification. Since each crystal has its own characteristics, the user should consult the crystal manufacturer for appropriate values of external components.					

9.2.3 EXTERNAL CRYSTAL OSCILLATOR CIRCUIT

Either a prepackaged oscillator can be used or a simple oscillator circuit with TTL gates can be built. Prepackaged oscillators provide a wide operating range and better stability. A well-designed crystal oscillator will provide good performance with TTL gates. Two types of crystal oscillator circuits can be used; one with series resonance or one with parallel resonance.

Figure 9-3 shows implementation of a parallel resonant oscillator circuit. The circuit is designed to use the fundamental frequency of the crystal. The 74AS04 inverter performs the 180° phase shift that a parallel oscillator requires. The 4.7 k Ω resistor provides the negative feedback for stability. The 10 k Ω potentiometers bias the 74AS04 in the linear region. This could be used for external oscillator designs.

FIGURE 9-3: EXTERNAL PARALLEL RESONANT CRYSTAL OSCILLATOR CIRCUIT



Figure 9-4 shows a series resonant oscillator circuit. This circuit is also designed to use the fundamental frequency of the crystal. The inverter performs a 180° phase shift in a series resonant oscillator circuit. The 330 k Ω resistors provide the negative feedback to bias the inverters in their linear region.

FIGURE 9-4: EXTERNAL SERIES RESONANT CRYSTAL OSCILLATOR CIRCUIT



9.2.4 RC OSCILLATOR

For timing insensitive applications the "RC" device option offers additional cost savings. The RC oscillator frequency is a function of the supply voltage, the resistor (REXT) and capacitor (CEXT) values, and the operating temperature. In addition to this, the oscillator frequency will vary from unit to unit due to normal process parameter variation. Furthermore, the difference in lead frame capacitance between package types will also affect the oscillation frequency, especially for low CEXT values. The user also needs to take into account variation due to tolerance of external R and C components used. Figure 9-5 shows how the R/C combination is connected to the PIC16C62X. For REXT values below 2.2 k Ω , the oscillator operation may become unstable or stop completely. For very high REXT values (e.g., 1 M Ω), the oscillator becomes sensitive to noise, humidity and leakage. Thus, we recommend to keep REXT between 3 k Ω and 100 k Ω .

Although the oscillator will operate with no external capacitor (CEXT = 0 pF), we recommend using values above 20 pF for noise and stability reasons. With no or small external capacitance, the oscillation frequency can vary dramatically due to changes in external capacitances, such as PCB trace capacitance or package lead frame capacitance.

See Section 13.0 for RC frequency variation from part to part due to normal process variation. The variation is larger for larger R (since leakage current variation will affect RC frequency more for large R) and for smaller C (since variation of input capacitance will affect RC frequency more).

See Section 13.0 for variation of oscillator frequency due to VDD for given REXT/CEXT values, as well as frequency variation due to operating temperature for given R, C and VDD values.

The oscillator frequency, divided by 4, is available on the OSC2/CLKOUT pin, and can be used for test purposes or to synchronize other logic (Figure 3-2 for waveform).

FIGURE 9-5: RC OSCILLATOR MODE



9.4.5 TIME-OUT SEQUENCE

On power-up the time-out sequence is as follows: First PWRT time-out is invoked after POR has expired. Then OST is activated. The total time-out will vary based on oscillator configuration and <u>PWRTE</u> bit status. For example, in RC mode with <u>PWRTE</u> bit erased (<u>PWRT</u> disabled), there will be no time-out at all. Figure 9-8, Figure 9-9 and Figure 9-10 depict time-out sequences.

Since the time-outs occur from the POR pulse, if $\overline{\text{MCLR}}$ is kept low long enough, the time-outs will expire. Then bringing $\overline{\text{MCLR}}$ high will begin execution immediately (see Figure 9-9). This is useful for testing purposes or to synchronize more than one PIC16C62X device operating in parallel.

Table 9-4 shows the RESET conditions for some special registers, while Table 9-5 shows the RESET conditions for all the registers.

9.4.6 POWER CONTROL (PCON)/ STATUS REGISTER

The power control/STATUS register, PCON (address 8Eh), has two bits.

Bit0 is $\overline{\text{BOR}}$ (Brown-out). $\overline{\text{BOR}}$ is unknown on Poweron Reset. It must then be set by the user and checked on subsequent RESETS to see if $\overline{\text{BOR}} = 0$, indicating that a brown-out has occurred. The $\overline{\text{BOR}}$ STATUS bit is a don't care and is not necessarily predictable if the brown-out circuit is disabled (by setting BODEN bit = 0 in the Configuration word).

Bit1 is POR (Power-on Reset). It is a '0' on Power-on Reset and unaffected otherwise. The user must write a '1' to this bit following a Power-on Reset. On a subsequent RESET, if POR is '0', it will indicate that a Power-on Reset must have occurred (VDD may have gone too low).

Oscillator Configuration	Powe	er-up	Brown-out Reset	Wake-up from SLEEP	
	PWRTE = 0	PWRTE = 1	Brown out Rooot		
XT, HS, LP	72 ms + 1024 Tosc	1024 Tosc	72 ms + 1024 Tosc	1024 Tosc	
RC	72 ms	_	72 ms	_	

TABLE 9-1: TIME-OUT IN VARIOUS SITUATIONS

TABLE 9-2:	STATUS/PCON BITS AND THEIR SIGNIFICANCE

POR	BOR	то	PD	
0	Х	1	1	Power-on Reset
0	Х	0	Х	Illegal, TO is set on POR
0	Х	Х	0	Illegal, PD is set on POR
1	0	Х	Х	Brown-out Reset
1	1	0	u	WDT Reset
1	1	0	0	WDT Wake-up
1	1	u	u	MCLR Reset during normal operation
1	1	1	0	MCLR Reset during SLEEP

Legend: u = unchanged, x = unknown

TABLE 9-3: SUMMARY OF REGISTERS ASSOCIATED WITH BROWN-OUT

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⁾
83h	STATUS				TO	PD				0001 1xxx	000q quuu
8Eh	PCON		_				_	POR	BOR	0x	uq

Legend: u = unchanged, x = unknown, - = unimplemented bit, reads as '0', q = value depends on condition.

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

9.5 Interrupts

The PIC16C62X has 4 sources of interrupt:

- External interrupt RB0/INT
- TMR0 overflow interrupt
- PORTB change interrupts (pins RB<7:4>)
- · Comparator interrupt

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

A global interrupt enable bit, GIE (INTCON<7>) enables (if set) all un-masked interrupts or disables (if cleared) all interrupts. Individual interrupts can be disabled through their corresponding enable bits in INTCON register. GIE is cleared on RESET.

The "return from interrupt" instruction, RETFIE, exits interrupt routine, as well as sets the GIE bit, which reenable RB0/INT interrupts.

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

The peripheral interrupt flag is contained in the special register PIR1. The corresponding interrupt enable bit is contained in special registers PIE1.

When an interrupt is responded to, the GIE is cleared to disable any further interrupt, the return address is pushed into the stack and the PC is loaded with 0004h.

FIGURE 9-15: INTERRUPT LOGIC

Once in the interrupt service routine, the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid RB0/ INT 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. The exact latency depends when the interrupt event occurs (Figure 9-16). The latency is the same for one or two cycle instructions. Once in the interrupt service routine, the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid multiple interrupt requests.

- Note 1: Individual interrupt flag bits are set regardless of the status of their corresponding mask bit or the GIE bit.
 - 2: When an instruction that clears the GIE bit is executed, any interrupts that were pending for execution in the next cycle are ignored. The CPU will execute a NOP in the cycle immediately following the instruction which clears the GIE bit. The interrupts which were ignored are still pending to be serviced when the GIE bit is set again.



PIC16C62X

CLRW	Clear W	COMF	Complement f
Syntax:	[label] CLRW	Syntax:	[<i>label</i>] COMF f,d
Operands:	None	Operands:	$0 \leq f \leq 127$
Operation:	$00h \rightarrow (W)$		d ∈ [0,1]
	$1 \rightarrow Z$	Operation:	$(f) \rightarrow (dest)$
Status Affected:	Z	Status Affected:	Z
Encoding:	00 0001 0000 0011	Encoding:	00 1001 dfff ffff
Description:	W register is cleared. Zero bit (Z) is set.	Description:	The contents of register 'f' are complemented. If 'd' is 0, the
Words:	1		result is stored in W. If 'd' is 1, the
Cycles:	1	Words:	1
Example	CLRW	Cycles:	1
	Before Instruction	Everale	COME DECI 0
	W = 0x5A	Example	COMF REGI,U
	W = 0x00		REG1 = $0x13$
	Z = 1		After Instruction
			$\begin{array}{rcl} REG1 &= & 0x13 \\ W &= & 0xEC \end{array}$
CLRWDT	Clear Watchdog Timer		
Syntax:	[label] CLRWDT	DEOE	
,		DECF	Decrement f
Operands:	None	DECF Syntax:	Decrement f
Operands: Operation:	None $00h \rightarrow WDT$	DECF Syntax: Operands:	Decrement f [<i>label</i>] DECF f,d 0 < f < 127
Operands: Operation:	None $00h \rightarrow WDT$ $0 \rightarrow WDT$ prescaler, $1 \rightarrow \overline{TO}$	Syntax: Operands:	Decrement f [<i>label</i>] DECF f,d $0 \le f \le 127$ $d \in [0,1]$
Operands: Operation:	None $00h \rightarrow WDT$ $0 \rightarrow WDT$ prescaler, $1 \rightarrow \overline{TO}$ $1 \rightarrow PD$	Syntax: Operands: Operation:	Decrement f [label] DECF f,d $0 \le f \le 127$ $d \in [0,1]$ (f) - 1 \rightarrow (dest)
Operands: Operation: Status Affected:	None $00h \rightarrow WDT$ $0 \rightarrow WDT$ prescaler, $1 \rightarrow \overline{TO}$ $1 \rightarrow PD$ \overline{TO}, PD	DECF Syntax: Operands: Operation: Status Affected:	Decrement f [label] DECF f,d $0 \le f \le 127$ $d \in [0,1]$ (f) - 1 \rightarrow (dest) Z
Operands: Operation: Status Affected: Encoding:	None $00h \rightarrow WDT$ $0 \rightarrow WDT \text{ prescaler,}$ $1 \rightarrow \overline{TO}$ $1 \rightarrow PD$ $\overline{TO, PD}$ $00 0000 0110 0100$	Syntax: Operands: Operation: Status Affected: Encoding:	Decrement f $[label]$ DECF f,d $0 \le f \le 127$ $d \in [0,1]$ $(f) - 1 \rightarrow (dest)$ Z 00 0011dfff
Operands: Operation: Status Affected: Encoding: Description:	None $00h \rightarrow WDT$ $0 \rightarrow WDT$ prescaler, $1 \rightarrow TO$ $1 \rightarrow PD$ TO, PD 00 0000 0110 0100 CLRWDT instruction resets the	DECF Syntax: Operands: Operation: Status Affected: Encoding: Description:	Decrement f $[label]$ DECF f,d $0 \le f \le 127$ $d \in [0,1]$ $(f) - 1 \rightarrow (dest)$ Z 00 0011 dfffdfffDecrement register 'f'. If 'd' is 0,
Operands: Operation: Status Affected: Encoding: Description:	None $00h \rightarrow WDT$ $0 \rightarrow WDT$ prescaler, $1 \rightarrow \overline{TO}$ $1 \rightarrow PD$ \overline{TO}, PD 00 0000 0110 0100 CLRWDT instruction resets the Watchdog Timer. It also resets the	Syntax: Operands: Operation: Status Affected: Encoding: Description:	Decrement f $[label]$ DECF f,d $0 \le f \le 127$ $d \in [0,1]$ $(f) - 1 \rightarrow (dest)$ Z 00 0011 dfffffffDecrement register 'f'. If 'd' is 0, the result is stored in the W
Operands: Operation: Status Affected: Encoding: Description:	None $00h \rightarrow WDT$ $0 \rightarrow WDT$ prescaler, $1 \rightarrow \overline{TO}$ $1 \rightarrow PD$ \overline{TO}, PD O CLRWDT instruction resets the Watchdog Timer. It also resets the prescaler of the WDT. STATUS http://doi.org/	DECF Syntax: Operands: Operation: Status Affected: Encoding: Description:	Decrement f $[label]$ DECF f,d $0 \le f \le 127$ $d \in [0,1]$ $(f) - 1 \rightarrow (dest)$ Z 00 0011 dfffffffDecrement register 'f'. If 'd' is 0,the result is stored in the Wregister. If 'd' is 1, the result isstored back in register 'f'
Operands: Operation: Status Affected: Encoding: Description:	None $00h \rightarrow WDT$ $0 \rightarrow WDT$ prescaler, $1 \rightarrow \overline{TO}$ $1 \rightarrow PD$ $\overline{TO}, \overline{PD}$ 00 0000 0110 0100 CLRWDT instruction resets the Watchdog Timer. It also resets the pres <u>cal</u> er of the WDT. STATUS bits TO and PD are set. 1	DECF Syntax: Operands: Operation: Status Affected: Encoding: Description:	Decrement f $[label]$ DECF f,d $0 \le f \le 127$ $d \in [0,1]$ $(f) - 1 \rightarrow (dest)$ Z000011dfffffffDecrement register 'f'. If 'd' is 0,the result is stored in the Wregister. If 'd' is 1, the result isstored back in register 'f'.1
Operands: Operation: Status Affected: Encoding: Description: Words:	None $00h \rightarrow WDT$ $0 \rightarrow WDT$ prescaler, $1 \rightarrow TO$ $1 \rightarrow PD$ TO, PD O 000 0110 0100 CLRWDT instruction resets the Watchdog Timer. It also resets the prescaler of the WDT. STATUS bits TO and PD are set. 1	DECF Syntax: Operands: Operation: Status Affected: Encoding: Description: Words:	Decrement f $[label]$ DECF f,d $0 \le f \le 127$ $d \in [0,1]$ $(f) - 1 \rightarrow (dest)$ Z 00 0011 dfffffffDecrement register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.1
Operands: Operation: Status Affected: Encoding: Description: Words: Cycles:	None $00h \rightarrow WDT$ $0 \rightarrow WDT prescaler,$ $1 \rightarrow TO$ $1 \rightarrow PD$ TO, PD 00 0000 0110 0100 CLRWDT instruction resets the Watchdog Timer. It also resets the prescaler of the WDT. STATUS bits TO and PD are set. 1 1	DECF Syntax: Operands: Operation: Status Affected: Encoding: Description: Words: Cycles:	Decrement f $[label]$ DECF f,d $0 \le f \le 127$ $d \in [0,1]$ $(f) - 1 \rightarrow (dest)$ Z 00 0011 dfffffffDecrement register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.111
Operands: Operation: Status Affected: Encoding: Description: Words: Cycles: Example	None $00h \rightarrow WDT$ $0 \rightarrow WDT prescaler,$ $1 \rightarrow TO$ $1 \rightarrow PD$ TO, PD 00 0000 0110 0100 CLRWDT instruction resets the Watchdog Timer. It also resets the pres <u>caler of the</u> WDT. STATUS bits TO and PD are set. 1 1 CLRWDT Defense landmattice	DECF Syntax: Operands: Operation: Status Affected: Encoding: Description: Words: Cycles: Example	Decrement f $[label]$ DECF f,d $0 \le f \le 127$ $d \in [0,1]$ $(f) - 1 \rightarrow (dest)$ Z 00 0011 dfffffffDecrement register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.11DECFCNT, 1Decrement register
Operands: Operation: Status Affected: Encoding: Description: Words: Cycles: Example	None $00h \rightarrow WDT$ $0 \rightarrow WDT$ prescaler, $1 \rightarrow TO$ $1 \rightarrow PD$ TO, PD 00 0000 0110 0100 CLRWDT instruction resets the Watchdog Timer. It also resets the prescaler of the WDT. STATUS bits TO and PD are set. 1 1 CLRWDT Before Instruction WDT counter = 2	DECF Syntax: Operands: Operation: Status Affected: Encoding: Description: Words: Cycles: Example	Decrement f $[label]$ DECF f,d $0 \le f \le 127$ $d \in [0,1]$ $(f) - 1 \rightarrow (dest)$ Z 00 0011 dfffDecrement register 'f'. If 'd' is 0,the result is stored in the Wregister. If 'd' is 1, the result isstored back in register 'f'.11DECFCNT, 1Before InstructionCNT CNT $= 0x01$
Operands: Operation: Status Affected: Encoding: Description: Words: Cycles: Example	None $00h \rightarrow WDT$ $0 \rightarrow WDT prescaler,$ $1 \rightarrow TO$ $1 \rightarrow PD$ TO, PD 00 0000 0110 0100 CLRWDT instruction resets the Watchdog Timer. It also resets the prescaler of the WDT. STATUS bits TO and PD are set. 1 1 CLRWDT Before Instruction WDT counter = ? After Instruction	DECF Syntax: Operands: Operation: Status Affected: Encoding: Description: Words: Cycles: Example	Decrement f[/abe/]DECF f,d $0 \le f \le 127$ $d \in [0,1]$ $(f) - 1 \rightarrow (dest)$ Z000011dfffffffDecrement register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.11DECFCNT, 1Before Instruction $CNT = 0x01$ $Z = 0$
Operands: Operation: Status Affected: Encoding: Description: Words: Cycles: Example	None $00h \rightarrow WDT$ $0 \rightarrow WDT prescaler,$ $1 \rightarrow TO$ $1 \rightarrow PD$ TO, PD 00 0000 0110 0100 CLRWDT instruction resets the Watchdog Timer. It also resets the prescaler of the WDT. STATUS bits TO and PD are set. 1 1 CLRWDT Before Instruction WDT counter = ? After Instruction WDT counter = 0x00 WDT respective 0	DECF Syntax: Operands: Operation: Status Affected: Encoding: Description: Words: Cycles: Example	Decrement f $\begin{bmatrix} label \end{bmatrix} DECF f,d$ $0 \le f \le 127$ $d \in [0,1]$ $(f) - 1 \rightarrow (dest)$ Z $\boxed{00 0011 dfff ffff}$ Decrement register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'. 1 1 $DECF CNT, 1$ Before Instruction $CNT = 0x01$ $Z = 0$ After Instruction
Operands: Operation: Status Affected: Encoding: Description: Words: Cycles: Example	None $\begin{array}{c} 00h \rightarrow WDT \\ 0 \rightarrow WDT \ prescaler, \\ 1 \rightarrow \overline{TO} \\ 1 \rightarrow PD \\ \hline TO, PD \\ \hline 00 & 0000 & 0110 & 0100 \\ \hline \\ CLRWDT \ instruction \ resets the \\ Watchdog Timer. It also resets the \\ Watchdog Timer. It also resets the \\ prescaler of the WDT. STATUS \\ bits TO and PD are set. \\ 1 \\ 1 \\ CLRWDT \\ \hline \\ Before \ Instruction \\ WDT \ counter \ = \ ? \\ After \ Instruction \\ WDT \ counter \ = \ 0 \\ \hline TO \ = \ 1 \end{array}$	DECF Syntax: Operands: Operation: Status Affected: Encoding: Description: Words: Cycles: Example	Decrement f $[label]$ DECF f,d $0 \le f \le 127$ $d \in [0,1]$ $(f) - 1 \rightarrow (dest)$ Z 00 0011 dffffffDecrement register 'f'. If 'd' is 0,the result is stored in the Wregister. If 'd' is 1, the result isstored back in register 'f'.11DECFCNT, 1Before InstructionCNTZ0After InstructionCNTCNT0 x00Z=1

SUBLW	Subtract W from Literal	SUBWF	Subtract W from f
Syntax:	[<i>label</i>] SUBLW k	Syntax:	[<i>label</i>] SUBWF f,d
Operands:	$0 \le k \le 255$	Operands:	$0 \le f \le 127$
Operation:	$k - (W) \rightarrow (W)$		d ∈ [0,1]
Status Affected:	C, DC, Z	Operation: Status	(f) - (W) \rightarrow (dest) C, DC, Z
Encoding:	11 110x kkkk kkkk	Affected:	
Description:	The W register is subtracted (2's complement method) from the eight bit literal 'k'. The result is placed in the W register.	Encoding: Description:	000010dfffffffSubtract (2's complement method)W register from register 'f'. If 'd' is 0,the result is stored in the W register.If 'd' is 1, the result is stored head in
Words:	1		register 'f'.
Cycles:	1	Words:	1
Example 1:	SUBLW 0x02	Cycles:	1
	Before Instruction	Example 1:	SUBWF REG1,1
	W = 1 $C = ?$		Before Instruction
	After Instruction		REG1= 3
	W = 1		W = 2 C = ?
Example 2:	Before Instruction		After Instruction
Example 2.	W = 2 $C = ?$		REG1= 1 W = 2 C = 1; result is positive
	After Instruction	Example 2:	Before Instruction
	W = 0 C = 1; result is zero		REG1= 2 W = 2
Example 3:	Before Instruction		C = ?
	W = 3 C = ?		After Instruction REG1= 0
	After Instruction		W = 2
	W = 0xFF	Example 3	C = 1; result is zero Before Instruction
	C – 0, result is negative		REG1= 1 W = 2 C = ?
			After Instruction
			REG1= 0xFF W = 2 C = 0; result is negative

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/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

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 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

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.

11.9 MPLAB ICE 2000 High Performance Universal In-Circuit Emulator

The MPLAB ICE 2000 universal in-circuit emulator is intended to provide the product development engineer with a complete microcontroller design tool set for PICmicro microcontrollers. Software control of the MPLAB ICE 2000 in-circuit emulator is advanced by the MPLAB Integrated Development Environment, which allows editing, building, downloading and source debugging from a single environment.

The MPLAB ICE 2000 is a full-featured emulator system with enhanced trace, trigger and data monitoring features. Interchangeable processor modules allow the system to be easily reconfigured for emulation of different processors. The universal architecture of the MPLAB ICE in-circuit emulator allows expansion to support new PICmicro microcontrollers.

The MPLAB ICE 2000 in-circuit emulator system has been designed as a real-time emulation system with advanced features that are typically found on more expensive development tools. The PC platform and Microsoft[®] Windows 32-bit operating system were chosen to best make these features available in a simple, unified application.

11.10 MPLAB ICE 4000 High Performance Universal In-Circuit Emulator

The MPLAB ICE 4000 universal in-circuit emulator is intended to provide the product development engineer with a complete microcontroller design tool set for highend PICmicro microcontrollers. Software control of the MPLAB ICE in-circuit emulator is provided by the MPLAB Integrated Development Environment, which allows editing, building, downloading and source debugging from a single environment.

The MPLAB ICD 4000 is a premium emulator system, providing the features of MPLAB ICE 2000, but with increased emulation memory and high speed performance for dsPIC30F and PIC18XXXX devices. Its advanced emulator features include complex triggering and timing, up to 2 Mb of emulation memory, and the ability to view variables in real-time.

The MPLAB ICE 4000 in-circuit emulator system has been designed as a real-time emulation system with advanced features that are typically found on more expensive development tools. The PC platform and Microsoft Windows 32-bit operating system were chosen to best make these features available in a simple, unified application.

11.11 MPLAB ICD 2 In-Circuit Debugger

Microchip's In-Circuit Debugger, MPLAB ICD 2, is a powerful, low cost, run-time development tool, connecting to the host PC via an RS-232 or high speed USB interface. This tool is based on the FLASH PICmicro MCUs and can be used to develop for these and other PICmicro microcontrollers. The MPLAB ICD 2 utilizes the in-circuit debugging capability built into the FLASH devices. This feature, along with Microchip's In-Circuit Serial Programming[™] (ICSP[™]) protocol, offers cost effective in-circuit FLASH debugging from the graphical user interface of the MPLAB Integrated Development Environment. This enables a designer to develop and debug source code by setting breakpoints, single-stepping and watching variables, CPU status and peripheral registers. Running at full speed enables testing hardware and applications in real-time. MPLAB ICD 2 also serves as a development programmer for selected PICmicro devices.

11.12 PRO MATE II Universal Device Programmer

The PRO MATE II is a universal, CE compliant device programmer with programmable voltage verification at VDDMIN and VDDMAX for maximum reliability. It features an LCD display for instructions and error messages and a modular detachable socket assembly to support various package types. In Stand-Alone mode, the PRO MATE II device programmer can read, verify, and program PICmicro devices without a PC connection. It can also set code protection in this mode.

11.13 PICSTART Plus Development Programmer

The PICSTART Plus development programmer is an easy-to-use, low cost, prototype programmer. It connects to the PC via a COM (RS-232) port. MPLAB Integrated Development Environment software makes using the programmer simple and efficient. The PICSTART Plus development programmer supports most PICmicro devices up to 40 pins. Larger pin count devices, such as the PIC16C92X and PIC17C76X, may be supported with an adapter socket. The PICSTART Plus development programmer is CE compliant.



FIGURE 12-8: PIC16CR62XA VOLTAGE-FREQUENCY GRAPH, -40°C \leq TA \leq 0°C, +70°C \leq TA \leq +125°C



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

PIC16C62X/C62XA/CR62XA			$\begin{array}{llllllllllllllllllllllllllllllllllll$					
PIC16LC62X/LC62XA/LCR62XA		$\begin{array}{llllllllllllllllllllllllllllllllllll$						
Param. No.	Sym	Characteristic	Min	Тур†	Мах	Units	Conditions	
D040	Vih	Input High Voltage I/O ports with TTL buffer	2.0V	_	1/22	V	VDD = 4.5V to 5.5V	
D041		with Schmitt Trigger input	0.25 VDD + 0.8V		VDD VDD		otherwise	
D041			0.8 VDD	_	VDD	V		
D043 D043A		OSC1 (XT, HS and LP) OSC1 (in RC mode)	0.7 VDD 0.9 VDD	—	VDD	V	(Note 1)	
D070	IPURB	PORTB weak pull-up current	50	200	400	μA	VDD = 5.0V, VPIN = VSS	
D070	IPURB	PORTB weak pull-up current	50	200	400	μA	VDD = 5.0V, VPIN = VSS	
	lı∟	Input Leakage Current ^(2, 3) I/O ports (Except PORTA)			±1.0	μA	Vss \leq VPIN \leq VDD, pin at hi-impedance	
D060		PORTA	_	_	±0.5	μA	Vss \leq VPIN \leq VDD, pin at hi-impedance	
D061		RA4/T0CKI	_	_	±1.0	μA	$Vss \leq V \text{PIN} \leq V \text{DD}$	
D063		OSC1, MCLR			±5.0	μΑ	Vss \leq VPIN \leq VDD, XT, HS and LP osc configuration	
	lı∟	Input Leakage Current ^(2, 3)						
					±1.0	μΑ	$Vss \leq V PIN \leq V DD, \ pin \ at \ hi\text{-impedance}$	
D060		PORTA	—	—	±0.5	μA	$Vss \le VPIN \le VDD$, pin at hi-impedance	
D061		RA4/T0CKI	—	—	±1.0	μA	$Vss \leq V \text{PIN} \leq V \text{DD}$	
D063		OSC1, MCLR	-		±5.0	μA	Vss \leq VPIN \leq VDD, XT, HS and LP osc configuration	
	Vol	Output Low Voltage						
D080		I/O ports	—	—	0.6	V	$IOL = 8.5 \text{ mA}, \text{ VDD} = 4.5 \text{V}, -40^{\circ} \text{ to } +85^{\circ}\text{C}$	
			—	—	0.6	V	IOL = 7.0 mA, VDD = 4.5V, +125°C	
D083		OSC2/CLKOUT (RC only)	—	—	0.6	V	$IOL = 1.6 \text{ mA}, \text{ VDD} = 4.5 \text{V}, -40^{\circ} \text{ to } +85^{\circ}\text{C}$	
			_	—	0.6	V	Iol = 1.2 mA, VDD = 4.5V, +125°C	

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 t 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.

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

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated)Operating temperature $0^{\circ}C \leq TA \leq +70^{\circ}C$ for commercial					
Param No.	Sym	Characteristic	Min	Тур†	Мах	Unit	Conditions	
	VIL	Input Low Voltage						
		I/O ports						
D030		with TTL buffer	Vss	—	0.8V 0.15Vdd	V	VDD = 4.5V to 5.5V, otherwise	
D031		with Schmitt Trigger input	Vss		0.2VDD	V		
D032		MCLR, RA4/T0CKI, OSC1 (in RC mode)	Vss	—	0.2Vdd	V	(Note 1)	
D033		OSC1 (in XT and HS)	Vss	—	0.3VDD	V		
		OSC1 (in LP)	Vss	—	0.6Vdd - 1.0	V		
	Viн	Input High Voltage						
		I/O ports						
D040		with TTL buffer	2.0V	—	VDD	V	VDD = 4.5V to 5.5V, otherwise	
D044		with Ochavitt Triansations t	0.25 VDD + 0.8		VDD			
D041					VDD			
D042		MCLR RA4/TUCKI		_	VDD	V		
D043		OSC1 (A1, HS and LP) OSC1 (in RC mode)		_	VDD	v	(Note 1)	
D070	IPURB	PORTB Weak Pull-up Current	50	200	400	μА	$V_{DD} = 5.0V$. VPIN = Vss	
	lil	Input Leakage Current ^(2, 3)						
		I/O ports (except PORTA)			±1.0	μA	VSS \leq VPIN \leq VDD, pin at hi-impedance	
D060		PORTA	_	_	±0.5	μA	Vss \leq VPIN \leq VDD, pin at hi-impedance	
D061		RA4/T0CKI	—	—	±1.0	μA	$Vss \le VPIN \le VDD$	
D063		OSC1, MCLR	_	—	±5.0	μA	$Vss \leq VPIN \leq VDD,$ XT, HS and LP osc configuration	
	Vol	Output Low Voltage						
D080		I/O ports	_	—	0.6	V	IOL = 8.5 mA, VDD = 4.5V, -40° to +85°C	
			—	—	0.6	V	IOL = 7.0 mA, VDD = 4.5V, +125°C	
D083		OSC2/CLKOUT (RC only)	—	—	0.6	V	IOL = 1.6 mA, VDD = 4.5V, -40° to +85°C	
		(2)	_		0.6	V	IOL = 1.2 mA, VDD = 4.5V, +125°C	
	Vон	Output High Voltage ⁽³⁾						
D090		I/O ports (except RA4)	VDD-0.7	—	—	V	IOH = -3.0 mA, VDD = 4.5V, -40° to +85°C	
			VDD-0.7	—	—	V	IOH = -2.5 mA, VDD = 4.5V, +125°C	
D092		OSC2/CLKOUT (RC only)	VDD-0.7	—	—	V	IOH = -1.3 mA, VDD = 4.5V, -40° to +85°C	
*0450	1/25	On an Duain Ulink Matterna	VDD-0.7	_		V	IOH = -1.0 mA, VDD = 4.5V, +125°C	
"D150	VOD	Open Drain High Voltage			8.5	V	RA4 pin	
		Output Pins						
D100	Cosc2	OSC2 pin			15	pF	In XT, HS and LP modes when external clock used to drive OSC1	
D101	Сю	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 VDD 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 IDD measurements in Active Operation mode are:

OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD, MCLR = VDD; WDT enabled/disabled as specified.
 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 bi-impedance state and tied to VDD or VSS.

 mode, with all I/O pins in hi-impedance state and tied to VDD or VSs.
 For RC osc configuration, current through REXT is not included. The current through the resistor can be estimated by the formula Ir = VDD/ 2REXT (mA) with REXT in kΩ.

5: 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.

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.

PIC16C62X









PIC16C62X

N
NOP Instruction
0
One-Time-Programmable (OTP) Devices7
OPTION Instruction
OPTION Register
Oscillator Configurations
Oscillator Start-up Timer (OST)50
Р
Package Marking Information
Packaging Information
PCL and PCLATH
PCON Register
PICkit 1 FLASH Starter Kit79
PICSTART Plus Development Programmer77
PIE1 Register
PIR1 Register
Port RB Interrupt
PORTA
PORTB
Power Control/Status Register (PCON)
Power-Down Mode (SLEEP)
Power-On Reset (POR)
Power-up Timer (PWRT)
Prescaler
PRO MATE II Universal Device Programmer
Program Memory Organization
Q
Quick-Turnaround-Production (QTP) Devices
R
RC Oscillator
Reset49
RETFIE Instruction70
RETLW Instruction70
RETURN Instruction70
RLF Instruction71
RRF Instruction71
S

0	
Serialized Quick-Turnaround-Production (SQTP) De	vices 7
SLEEP Instruction	71
Software Simulator (MPLAB SIM)	76
Software Simulator (MPLAB SIM30)	76
Special Features of the CPU	45
Special Function Registers	17
Stack	23
Status Register	18
SUBLW Instruction	72
SUBWF Instruction	72
SWAPF Instruction	73

Т

Timer0	
TIMER0	
TIMER0 (TMR0) Interrupt	
TIMER0 (TMR0) Module	
TMR0 with External Clock	
Timer1	
Switching Prescaler Assignment	35
Timing Diagrams and Specifications	
TMR0 Interrupt	
TRIS Instruction	73
TRISA	
TRISB	

v

Voltage Reference Module VRCON Register	43 43
W	
Watchdog Timer (WDT)	. 58
WWW, On-Line Support	3
X	
XORLW Instruction	. 73
XORWF Instruction	.73