



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	UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	16
Program Memory Size	3.5KB (2K x 14)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	224 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Through Hole
Package / Case	18-DIP (0.300", 7.62mm)
Supplier Device Package	18-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f628-20e-p

Email: info@E-XFL.COM

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

PIC16F62X

Pin Diagrams



Device Differences

Device	Voltage Range	Oscillator	Process Technology (Microns)
PIC16F627	3.0 - 5.5	(Note 1)	0.7
PIC16F628	3.0 - 5.5	(Note 1)	0.7
PIC16LF627	2.0 - 5.5	(Note 1)	0.7
PIC16LF628	2.0 - 5.5	(Note 1)	0.7
Note 1: If you change from the application.	nis device to another devic	e, please verify oscillator ch	aracteristics in your

3.2.2 SPECIAL FUNCTION REGISTERS

The SFRs are registers used by the CPU and Peripheral functions for controlling the desired operation of the device (Table 3-1). These registers are static RAM.

The special registers can be classified into two sets (core and peripheral). The SFRs 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.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR Reset ⁽¹⁾	Details on Page
Bank 0											
00h	INDF	Addressi	ng this locatio	n uses conte	nts of FSR to	address data	a memory (n	ot a physica	l register)	xxxx xxxx	25
01h	TMR0	Timer0 M	lodule's Regis	ster						xxxx xxxx	43
02h	PCL	Program	Counter's (PC	C) Least Sign	ficant Byte					0000 0000	13
03h	STATUS	IRP	RP1	RP0	TO	PD	Z	DC	С	0001 1xxx	19
04h	FSR	Indirect d	ata memory a	ddress point	er	•	•		•	xxxx xxxx	25
05h	PORTA	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	xxxx 0000	29
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	34
07h	_	Unimplen	nented								
08h	—	Unimplen	nented							_	_
09h	—	Unimplen	nented		-					_	_
0Ah	PCLATH		_	—	Write buffer	for upper 5 b	oits of progra	m counter		0 0000	25
0Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	21
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	23
0Dh	_	Unimplen	Jnimplemented —								_
0Eh	TMR1L	Holding r	egister for the	Least Signifi	cant Byte of	the 16-bit TN	R1			XXXX XXXX	46
0Fh	TMR1H	Holding r	egister for the	Most Signific	cant Byte of t	he 16-bit TM	R1			xxxx xxxx	46
10h	T1CON	_	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR10N	00 0000	46
11h	TMR2	TMR2 mo	odule's registe	er						0000 0000	50
12h	T2CON	_	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	50
13h	_	Unimplen	nented								
14h	_	Unimplen	nented							_	_
15h	CCPR1L	Capture/0	Compare/PWI	M register (LS	SB)					xxxx xxxx	61
16h	CCPR1H	Capture/0	Compare/PWI	M register (M	SB)		-			xxxx xxxx	61
17h	CCP1CON		_	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	00 0000	61
18h	RCSTA	SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D	0000 -00x	67
19h	TXREG	USART T	ransmit data	register						0000 0000	74
1Ah	RCREG	USART F	Receive data r	egister						0000 0000	77
1Bh	_	Unimplen	nented							_	_
1Ch	_	Unimplen	nented							—	_
1Dh	_	Unimplen	Unimplemented — — —							_	
1Eh	—	Unimplen	nented							—	_
1Fh	CMCON	C2OUT	C10UT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0000	53

TABLE 3-1: SPECIAL REGISTERS SUMMARY BANK 0

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

Note 1: For the Initialization Condition for Registers Tables, refer to Table 14-7 and Table 14-8 on page 98.



FIGURE 5-3: BLOCK DIAGRAM OF THE RA3/AN3 PIN





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





© 2003 Microchip Technology Inc.

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 Bit 5 of PORTB will cause all eight bits of PORTB to be read into the CPU. Then the BSF operation takes place on Bit 5 and PORTB is written to the output latches. If another bit of PORTB is used as a bi-directional I/O pin (e.g., Bit 0) 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 rewritten 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 Bit 0 is switched into Output mode later on, the content of the data latch may now be unknown.

Reading a 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., ${\tt BCF}\,,\ {\tt BSF},$ 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 latchPORT Pins
BCF STATUS, RPO	;
BCF PORTB, 7	;01pp pppp 11pp pppp
BSF STATUS, RPO	;
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 5-16). 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 5-16: SUCCESSIVE I/O OPERATION

FIGURE 9-4: ANALOG INPUT MODE



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
1Fh	CMCON	C2OUT	C10UT	C2INV	C1NV	CIS	CM2	CM1	CM0	0000 0000	0000 0000
0Bh/8Bh/ 10Bh/18Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	_	CCP1IF	TMR2IF	TMR1IF	0000 -000	0000 -000
8Ch	PIE1	EEIE	CMIE	RCIE	TXIE	_	CCP1IE	TMR2IE	TMR1IE	0000 -000	0000 -000
85h	TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	1111 1111	1111 1111

Legend: x = Unknown, u = Unchanged, - = Unimplemented, read as '0'

10.0 VOLTAGE REFERENCE MODULE

The Voltage Reference is a 16-tap resistor ladder network that provides a selectable voltage reference. The resistor ladder is segmented to provide two ranges of VREF values and has a power-down function to conserve power when the reference is not being used. The VRCON register controls the operation of the reference as shown in Figure 10-1. The block diagram is given in Figure 10-1.

10.1 Configuring the Voltage Reference

The Voltage Reference can output 16 distinct voltage levels for each range.

The equations used to calculate the output of the Voltage Reference are as follows:

if VRR = 1: VREF = (VR<3:0>/24) x VDD

voltage of 1.25V with VDD = 5.0V.

if VRR = 0: VREF = (VDD x 1/4) + (VR<3:0>/32) x VDD

The setting time of the Voltage Reference must be considered when changing the VREF output (Table 17-2). Example 10-1 shows an example of how to configure the Voltage Reference for an output

REGISTER 10-1:	VRCON R	EGISTER	(ADDRES	S: 9Fh)						
	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0		
	VREN	VROE	Vrr	_	VR3	VR2	VR1	VR0		
	bit 7							bi		
bit 7	Vren: Vre	F Enable								
	1 = VREF C 0 = VREF C	1 = VREF circuit powered on 0 = VREF circuit powered down, no IDD drain								
bit 6	VROE: VRE	F Output Er	able							
	1 = VREF is output on RA2 pin 0 = VREF is disconnected from RA2 pin									
bit 5	VRR: VREF	VRR: VREF Range selection								
	1 = Low Ra 0 = High R	1 = Low Range 0 = High Range								
bit 4	Unimplem	ented: Rea	d as '0'							
bit 3-0	Vr<3:0> : ∖ When Vrr When Vrr	/REF value s = 1: VREF = = 0: VREF =	election 0 ⊴ = (VR<3:0>/ = 1/4 * VDD ·	≤ VR [3:0] ≤ ′ 24) * VDD + (VR<3:0>/	15 32) * Vdd					

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

FIGURE 10-1: VOLTAGE REFERENCE BLOCK DIAGRAM







Steps to follow when setting up an Asynchronous Transmission:

- 1. Initialize the SPBRG register for the appropriate baud rate. If a high speed baud rate is desired, set bit BRGH. (Section 12.1)
- 2. Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
- 3. If interrupts are desired, then set enable bit TXIE.
- 4. If 9-bit transmission is desired, then set transmit bit TX9.
- 5. Enable the transmission by setting bit TXEN, which will also set bit TXIF.
- 6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
- 7. Load data to the TXREG register (starts transmission).







TABLE 12-6: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION

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
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF		CCP1IF	TMR2IF	TMR1IF	0000 -000	0000 -000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D	0000 -00x	0000 -00x
19h	TXREG	USART Trai	nsmit Re	gister						0000 0000	0000 0000
8Ch	PIE1	EEIE	CMIE	RCIE	TXIE	_	CCP1IE	TMR2IE	TMR1IE	0000 -000	0000 -000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	_	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register							0000 0000	0000 0000	

Legend: x = unknown, - = unimplemented locations read as '0'. Shaded cells are not used for Asynchronous Transmission.

12.4.2 USART SYNCHRONOUS MASTER RECEPTION

Once Synchronous mode is selected, reception is enabled by setting either enable bit SREN (RCSTA<5>) or enable bit CREN (RCSTA<4>). Data is sampled on the RB1/RX/DT pin on the falling edge of the clock. If enable bit SREN is set, then only a single word is received. If enable bit CREN is set, the reception is continuous until CREN is cleared. If both bits are set, then CREN takes precedence. After clocking the last bit, the received data in the Receive Shift Register (RSR) is transferred to the RCREG register (if it is empty). When the transfer is complete, interrupt flag bit RCIF (PIR1<5>) is set. The actual interrupt can be enabled/disabled by setting/clearing enable bit RCIE (PIE1<5>). Flag bit RCIF is a read only bit which is RESET by the hardware. In this case, it is RESET when the RCREG register has been read and is empty. The RCREG is a double buffered register (i.e., it is a two-deep FIFO). It is possible for two bytes of data to be received and transferred to the RCREG FIFO and a third byte to begin shifting into the RSR register. On the clocking of the last bit of the third byte, if the RCREG register is still full then overrun error bit OERR (RCSTA<1>) is set. The word in the RSR will be lost. The RCREG register can be read twice to retrieve the two bytes in the FIFO. Bit OERR has to be cleared in software (by clearing bit CREN). If bit OERR is set, transfers from the RSR to the RCREG are inhibited, so it is essential to clear bit OERR if it is set. The 9th

receive bit is buffered the same way as the receive data. Reading the RCREG register, will load bit RX9D with a new value, therefore it is essential for the user to read the RCSTA register before reading RCREG in order not to lose the old RX9D information.

Steps to follow when setting up a Synchronous Master Reception:

- 1. Initialize the SPBRG register for the appropriate baud rate. (Section 12.1)
- 2. Enable the synchronous master serial port by setting bits SYNC, SPEN, and CSRC.
- 3. Ensure bits CREN and SREN are clear.
- 4. If interrupts are desired, then set enable bit RCIE.
- 5. If 9-bit reception is desired, then set bit RX9.
- 6. If a single reception is required, set bit SREN. For continuous reception set bit CREN.
- Interrupt flag bit RCIF will be set when reception is complete and an interrupt will be generated if enable bit RCIE was set.
- 8. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
- 9. Read the 8-bit received data by reading the RCREG register.
- 10. If any error occurred, clear the error by clearing bit CREN.

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
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	_	CCP1IF	TMR2IF	TMR1IF	0000 -000	0000 -000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D	0000 -00x	0000 -00x
1Ah	RCREG	USART R	eceive F	Register						0000 0000	0000 0000
8Ch	PIE1	EEPIE	CMIE	RCIE	TXIE	_	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 -000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	_	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate	Baud Rate Generator Register							0000 0000	0000 0000

TABLE 12-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION

Legend: x = unknown, - = unimplemented read as '0'. Shaded cells are not used for Synchronous Master Reception.

14.5 Power-on Reset (POR), Power-up Timer (PWRT), Oscillator Start-up Timer (OST) and Brown-out Detect (BOD)

14.5.1 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".

14.5.2 POWER-UP TIMER (PWRT)

The PWRT provides a fixed 72 ms (nominal) timeout on power-up only, from POR or Brown-out Detect Reset. The PWRT 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, PWRTE can disable (if set) or enable (if cleared or programmed) the PWRT. The PWRT should always be enabled when Brown-out Detect 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.

14.5.3 OSCILLATOR START-UP TIMER (OST)

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

14.5.4 BROWN-OUT DETECT (BOD) RESET

The PIC16F62X members have on-chip BOD circuitry. A configuration bit, BODEN, can disable (if clear/ programmed) or enable (if set) the BOD Reset circuitry. If VDD falls below VBOD for longer than TBOD, the brown-out situation will RESET the chip. A RESET is not guaranteed to occur if VDD falls below VBOD for shorter than TBOD. VBOD and TBOD are defined in Table 17-1 and Table 17-6, respectively.

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

If VDD drops below VBOD while the Power-up Timer is running, the chip will go back into a Brown-out Detect Reset and the Power-up Timer will be re-initialized. Once VDD rises above VBOD, the Power-Up Timer will execute a 72 ms RESET. The Power-up Timer should always be enabled when Brown-out Detect is enabled. Figure 14-7 shows typical Brown-out situations.



FIGURE 14-7: BROWN-OUT SITUATIONS

14.6 Interrupts

The PIC16F62X has 10 sources of interrupt:

- External Interrupt RB0/INT
- TMR0 Overflow Interrupt
- PORTB Change Interrupts (pins RB7:RB4)
- Comparator Interrupt
- USART Interrupt TX
- USART Interrupt RX
- CCP Interrupt
- TMR1 Overflow Interrupt
- TMR2 Match Interrupt
- EEPROM

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. 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 14-15). 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. Individual interrupt flag bits are set regardless of the status of their corresponding mask bit or the GIE bit.

- 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.



FIGURE 14-14: INTERRUPT LOGIC

14.12 In-Circuit Serial Programming

The PIC16F62X microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data, and three other lines for power, ground, and the programming voltage. This allows customers to manufacture boards with unprogrammed devices, and then program the microcontroller just before shipping the product. This also allows the most recent firmware, or a custom firmware to be programmed.

The device is placed into a Program/Verify mode by holding the RB6 and RB7 pins low while raising the $\overline{\text{MCLR}}$ (VPP) pin from VIL to VIHH (see programming specification). RB6 becomes the programming clock and RB7 becomes the programming data. Both RB6 and RB7 are Schmitt Trigger inputs in this mode.

After RESET, to place the device into Programming/ Verify mode, the program counter (PC) is at location 00h. A 6-bit command is then supplied to the device. Depending on the command, 14 bits of program data are then supplied to or from the device, depending if the command was a load or a read. For complete details of serial programming, please refer to the Programming Specifications.

A typical in-circuit serial programming connection is shown in Figure 14-18.

FIGURE 14-18: TYPICAL IN-CIRCUIT SERIAL PROGRAMMING CONNECTION



14.13 Low Voltage Programming

The LVP bit of the configuration word, enables the low voltage programming. This mode allows the microcontroller to be programmed via ICSP using only a 5V source. This mode removes the requirement of VIHH to be placed on the MCLR pin. The LVP bit is normally erased to '1', which enables the low voltage programming. In this mode, the RB4/PGM pin is dedicated to the programming function and ceases to be a general purpose I/O pin. The device will enter Programming mode when a '1' is placed on the RB4/PGM pin. The HV Programming mode is still available by placing VIHH on the MCLR pin.

- Note 1: While in this mode, the RB4 pin can no longer be used as a general purpose I/O pin.
 - 2: VDD must be 5.0V <u>+</u>10% during erase/ program operations while in low voltage Programming mode.

If Low voltage Programming mode is not used, the LVP bit can be programmed to a '0', and RB4/PGM becomes a digital I/O pin. To program the device, VIHH must be placed onto MCLR during programming. The LVP bit may only be programmed when programming is entered with VIHH on MCLR. The LVP bit cannot be programmed when programming is entered with RB4/PGM.

It should be noted, that once the LVP bit is programmed to 0, High voltage Programming mode can be used to program the device.

PIC16F62X

CLRW	Clear V	V				COMF
Syntax:	[label]	CLRW				Syntax
Operands:	None					Opera
Operation:	$\begin{array}{c} 00h \rightarrow 0\\ 1 \rightarrow Z \end{array}$	(W)				Opera
Status Affected:	Z					Status
Encoding:	00	0001	0000	0011		Encod
Description:	W regis (Z) is se	ter is cle et.	ared. Zer	o bit	1	Descri
Words:	1					
Cycles:	1					
Example	CLRW					Words
	Before	Instructio	on F A			Cycles
	After In	vv = 0x struction W = 0x Z = 1	5A 00			Examp

COMF	Complement f
Syntax:	[label] COMF f,d
Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ d \in [0,1] \end{array}$
Operation:	$(\overline{f}) \rightarrow (dest)$
Status Affected:	Z
Encoding:	00 1001 dfff ffff
Description:	complemented. If 'd' is 0 the result is stored in W. If 'd' is 1 the result is stored back in register 'f'.
Words:	1
Cycles:	1
Example	COMF REG1, 0
	Before Instruction REG1 = $0x13$ After Instruction REG1 = $0x13$ W = $0xEC$

CLRWDT	Clear Watchdog Timer	DECF
Syntax:	[label] CLRWDT	Syntax:
Operands:	None	Operands:
Operation:	$\begin{array}{l} 00h \rightarrow WDT \\ 0 \rightarrow \underline{WDT} \text{ prescaler,} \\ 1 \rightarrow \underline{TO} \\ 1 \rightarrow \overline{PD} \end{array}$	Operation: Status Affecte
Status Affected:	TO, PD	Encoding:
Encoding:	00 0000 0110 0100	Description:
Description:	CLRWDT instruction resets the Watchdog Timer. It also resets the prescaler of the WDT. STATUS bits TO and PD are set.	Words:
Words:	1	Cycles:
Cycles:	1	Example
Example	CLRWDT	
	Before Instruction WDT counter = ? After Instruction WDT counter = $0x00$ WDT prescaler = 0 TO = 1 PD = 1	

DECF	Decrement f							
Syntax:	[<i>label</i>] DECF f,d							
Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ d \in [0,1] \end{array}$							
Operation:	(f) - 1 \rightarrow (dest)							
Status Affected:	Z							
Encoding:	00 0011 dfff ffff							
Description:	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'.							
Words:	1							
Cycles:	1							
Example	DECF CNT, 1							
	Before Instruction CNT = 0x01 Z = 0 After Instruction CNT = 0x00 Z = 1							

				 aupm1		- `	SJOIP		neprddel	suamu	 Program			SILV	IBV	BUB	soards	owa			*
	MPLAB Integrated Development Environment	MPLAB C17 C Compiler	MPLAB C18 C Compiler	MPASM Assembler/ MPLINK Object Linker	MPLAB C30 C Compiler	MPLAB ASM30 Assembler/Linker/Librarian	MPLAB ICE 2000 In-Circuit Emulator	MPLAB ICE 4000 In-Circuit Emulator	MPLAB ICD 2 In-Circuit Debugger	PICSTART Plus Entry Level Development Programmer	PRO MATE II Universal Device Programmer	PICDEM 1 Demonstration Board	PICDEM.net Demonstration Board	PICDEM 2 Plus Demonstration Board	PICDEM 3 Demonstration Board	PICDEM 14A Demonstration Board	PICDEM 17 Demonstration Board	PICDEM 18R Demonstration Board	PICDEM LIN Demonstration Board	PICDEM USB Demonstration Board	contact the Microchip web site at w
PIC12CXXX	>			>			>	>		>	>										ww.micre
PIC12FXXX	>			>			>		>	>	>										ochip.co
PIC14000	>			>			>			>	>					>					m for in
PIC16C5X	>			>			>	>		>	>	>									formatic
PIC16C6X	>			>			>	>	* >	>	>		L	^ +			L		L	L	on ho
PIC16CXXX	>			>			>	>		>	>	>									w to use
PIC16C43X	>			>			>			>	>								>		the MP.
PIC16F62X	>			>			**/			**	**>										LAB ICD
X7281219	>			>			>	>	*^	>	>	∕†		+							In-Circu
XX209101d	`			`			· 、	>		`	``````````````````````````````````````									-	uit Debu
4X/091014	`			`			`	>		`	<u> </u>	>									ager (D)
PIC16F8XX	``````````````````````````````````````			``````````````````````````````````````			``````````````````````````````````````		>	``````````````````````````````````````	<u>``</u>	`							>	>	/164001
PIC16C9XX	>			>			>	>		>	<u>></u>				>) with P
X4371919	>	>		>			>			>	>	>									C16C62
XXTOTIOI9	>	>		>			>	>		>	>						>				. 63. 64.
PIC18CXX2	>		>	>			>	>		>	>		>	>							65. 72.
PI18CX01								>	>									>			73.74.7
PIC18FXXX	>		>	>			>	^	^	~	>			^							6. 77.
dsPIC30F					>	>		>	>												

TABLE 16-1: DEVELOPMENT TOOLS FROM MICROCHIP

DS40300C-page 126





Param No.	Sym	Characteristic		Min	Тур†	Max	Units
10*	TosH2ckL	OSC1↑ to CLKOUT↓	16F62X	—	75	200	ns
10A*			16LF62X	—	_	400	ns
11*	TosH2ckH	OSC1 [↑] to CLKOUT [↑]	16F62X	—	75	200	ns
11A*			16LF62X	—	—	400	ns
12*	TckR	CLKOUT rise time	16F62X	—	35	100	ns
12A*			16LF62X	—	_	200	ns
13*	TckF	CLKOUT fall time	16F62X	—	35	100	ns
13A*			16LF62X	—		200	ns
14*	TckL2ioV	CLKOUT \downarrow to Port out valid	—	_	20	ns	
15*	TioV2ckH	Port in valid before	16F62X	Tosc+200 ns	_	—	ns
		CLKOUT ↑	16LF62X	Tosc=400 ns	_	—	ns
16*	TckH2iol	Port in hold after CLKOUT ↑		0	_	—	ns
17*	TosH2ioV	OSC1↑ (Q1 cycle) to	16F62X	—	50	150*	ns
		Port out valid	16LF62X	—	_	300	ns
18*	TosH2iol	OSC1 [↑] (Q2 cycle) to Port input invalic (I/O in hold time)	100 200	_	—	ns	

* These parameters are characterized but not tested.

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





FIGURE 18-4: TYPICAL IDD vs Fosc OVER VDD (XT MODE)





ON-LINE SUPPORT

Microchip provides on-line support on the Microchip World Wide Web site.

The web site is used by Microchip as a means to make files and information easily available to customers. To view the site, the user must have access to the Internet and a web browser, such as Netscape[®] or Microsoft[®] Internet Explorer. Files are also available for FTP download from our FTP site.

Connecting to the Microchip Internet Web Site

The Microchip web site is available at the following URL:

www.microchip.com

The file transfer site is available by using an FTP service to connect to:

ftp://ftp.microchip.com

The web site and file transfer site provide a variety of services. Users may download files for the latest Development Tools, Data Sheets, Application Notes, User's Guides, Articles and Sample Programs. A variety of Microchip specific business information is also available, including listings of Microchip sales offices, distributors and factory representatives. Other data available for consideration is:

- Latest Microchip Press Releases
- Technical Support Section with Frequently Asked
 Questions
- Design Tips
- Device Errata
- Job Postings
- Microchip Consultant Program Member Listing
- Links to other useful web sites related to Microchip Products
- Conferences for products, Development Systems, technical information and more
- Listing of seminars and events

SYSTEMS INFORMATION AND UPGRADE HOT LINE

The Systems Information and Upgrade Line provides system users a listing of the latest versions of all of Microchip's development systems software products. Plus, this line provides information on how customers can receive the most current upgrade kits.The Hot Line Numbers are:

1-800-755-2345 for U.S. and most of Canada, and

1-480-792-7302 for the rest of the world.

092002

READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 792-4150.

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

To:	Technical Publications Manager	Total Pages Sent							
RE:	Reader Response								
Fron	ו: Name								
	Company								
	Address								
	City / State / ZIP / Country								
	Telephone: ()	FAX: ()							
Appl	Application (optional):								
Wou	ld you like a reply?YN								
Devi	ce: PIC16F62X Lite	erature Number: DS40300C							
Que	stions:								
1 \	What are the best features of this do	cument?							
-									
2.	How does this document meet your h	nardware and software development needs?							
-									
-									
3. I	Do you find the organization of this d	ocument easy to follow? If not, why?							
-									
-									
4.	What additions to the document do y	ou think would enhance the structure and subject?							
-									
5 \	Nhat deletions from the document or	ould be made without affecting the overall usefulness?							
5.		Sud be made without anecting the overall usefulness?							
-									
6. I	s there any incorrect or misleading in	nformation (what and where)?							
	. 0								
-									
7. I	How would you improve this docume	nt?							
-									