



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	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	128 × 8
RAM Size	128 × 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	-
Oscillator Type	External
Operating Temperature	0°C ~ 70°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/pic16ce625-04-p

Email: info@E-XFL.COM

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

Table of Contents

1.0	General Description	3				
2.0	PIC16CE62X Device Varieties	5				
3.0	Architectural Overview	7				
4.0	Memory Organization	11				
5.0	I/O Ports	23				
6.0	EEPROM Peripheral Operation	29				
7.0	Timer0 Module	35				
8.0	Comparator Module	41				
9.0	Voltage Reference Module	47				
10.0	Special Features of the CPU	49				
11.0	Instruction Set Summary	65				
12.0	Development Support	77				
13.0	Electrical Specifications	83				
14.0	Packaging Information	97				
Appe	ndix A: Code for Accessing EEPROM Data Memory	103				
Index		105				
On Li	ne Support	107				
Read	Reader Response					
PIC1	IC16CE62X Product Identification System109					

To Our Valued Customers

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number. e.g., DS30000A is version A of document DS30000.

New Customer Notification System

Register on our web site (www.microchip.com/cn) to receive the most current information on our products.

Errata

An errata sheet may exist for current devices, describing minor operational differences (from the data sheet) and recommended workarounds. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; http://www.microchip.com
- Your local Microchip sales office (see last page)
- The Microchip Corporate Literature Center; U.S. FAX: (480) 786-7277

When contacting a sales office or the literature center, please specify which device, revision of silicon and data sheet (include literature number) you are using.

Corrections to this Data Sheet

We constantly strive to improve the quality of all our products and documentation. We have spent a great deal of time to ensure that this document is correct. However, we realize that we may have missed a few things. If you find any information that is missing or appears in error, please:

- Fill out and mail in the reader response form in the back of this data sheet.
- E-mail us at webmaster@microchip.com.

We appreciate your assistance in making this a better document.

4.0 MEMORY ORGANIZATION

4.1 <u>Program Memory Organization</u>

The PIC16CE62X has a 13-bit program counter capable of addressing an 8K x 14 program memory space. Only the first 512 x 14 (0000h - 01FFh) for the PIC16CE623, 1K x 14 (0000h - 03FFh) for the PIC16CE624 and 2K x 14 (0000h - 07FFh) for the PIC16CE625 are physically implemented. Accessing a location above these boundaries will cause a wrap-around within the first 512 x 14 space (PIC16CE623) or 1K x 14 space (PIC16CE624) or 2K x 14 space (PIC16CE625). The reset vector is at 0000h and the interrupt vector is at 0004h (Figure 4-1, Figure 4-2, Figure 4-3).

FIGURE 4-1: PROGRAM MEMORY MAP AND STACK FOR THE PIC16CE623



FIGURE 4-2: PROGRAM MEMORY MAP AND STACK FOR THE PIC16CE624



FIGURE 4-3: PROGRAM MEMORY MAP AND STACK FOR THE PIC16CE625



4.2.2.1 STATUS REGISTER

The STATUS register, shown in Register 4-1, contains the arithmetic status of the ALU, the RESET status and the bank select bits for data memory.

The STATUS register can be the destination for any instruction, like any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the TO and PD bits are not writable. Therefore, the result of an instruction with the STATUS register as destination may be different than intended.

For example, CLRF STATUS will clear the upper-three bits and set the Z bit. This leaves the status register as 000uu1uu (where u = unchanged).

It is recommended, therefore, that only BCF, BSF, SWAPF and MOVWF instructions are used to alter the STATUS register, because these instructions do not affect any status bit. For other instructions, not affecting any status bits, see the "Instruction Set Summary".

Note 1:	The IRP and RP1 bits (STATUS<7:6>) are not used by the PIC16CE62X and should be programmed as '0'. Use of these bits as general purpose R/W bits is NOT recommended, since this may affect upward compatibility with future products.
Note 2:	The C and DC bits operate as a Borrow and Digit Borrow out bit, respectively, in subtraction. See the SUBLW and SUBWF instructions for examples.

REGISTER 4-1: STATUS REGISTER (ADDRESS 03H OR 83H)

Reserved	Reserved	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x	
IRP bit7	RP1	RP0	TO	PD	Z	DC	C bit0	R = Readable bit W = Writable bit
								U = Unimplemented bit, read as '0'
								-n = Value at POR reset -x = Unknown at POR reset
bit 7:	IRP: The IF	RP bit is r	eserved or	the PIC1	6CE62X, a	lways mair	ntain this bit	t clear.
bit 6:5	SRP<1:O>: Register Bank Select bits (used for direct addressing) 11 = Bank 3 (180h - 1FFh) 10 = Bank 2 (100h - 17Fh) 01 = Bank 1 (80h - FFh) 00 = Bank 0 (00h - 7Fh) Fach bank is 128 bytes. The BP1 bit is reserved, always maintain this bit clear.							
bit 4:	TO: Time-out bit 1 = After power-up, CLRWDT instruction, or SLEEP instruction 0 = A WDT time-out occurred							
bit 3:	PD : Power-down bit 1 = After power-up or by the CLRWDT instruction 0 = By execution of the SLEEP instruction							
bit 2:	Z : Zero bit 1 = The res 0 = The res	sult of an sult of an	arithmetic arithmetic	or logic op or logic op	peration is a	zero not zero		
bit 1:	DC : Digit c 1 = A carry 0 = No carr	arry/borro v-out from ry-out fro	bw bit (ADD the 4th low m the 4th l	WF, ADDLW w order bit ow order b	of the result of the result	SUBWF instr ult occurred sult	uctions) (for I	or borrow the polarity is reversed)
bit 0:	C: Carry/bc 1 = A carry 0 = No carr Note: For b second ope the source	orrow bit -out from ry-out from porrow the erand. Fo register.	(ADDWF, AD the most s m the mos e polarity is r rotate (RH	DLW, SUB: significant t significan s reversed RF, RLF) in	LW, SUBWF bit of the ro t bit of the . A subtrac structions,	instructior esult occurr result occu tion is exec this bit is lo	ns) red urred suted by add baded with e	ding the two's complement of the either the high or low order bit of

4.2.2.4 PIE1 REGISTER

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

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



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)



4.4 Indirect Addressing, INDF and FSR Registers

The INDF register is not a physical register. Addressing the INDF register will cause indirect addressing.

Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses data pointed to by the File Select Register (FSR). Reading INDF itself indirectly will produce 00h. Writing to the INDF register indirectly results in a no-operation (although status bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit (STATUS<7>), as shown in Figure 4-7. However, IRP is not used in the PIC16CE62X. A simple program to clear RAM location 20h-2Fh using indirect addressing is shown in Example 4-1.

EXAMPLE 4-1:		INDIRECT ADDRESSING				
	movlw	0x20	;initialize pointer			
	movwf	FSR	;to RAM			
NEXT	clrf	INDF	;clear INDF register			
	incf	FSR	;inc pointer			
	btfss	FSR,4	;all done?			
	goto	NEXT	;no clear next			
			;yes continue			
CONTINUE:						

FIGURE 4-7: DIRECT/INDIRECT ADDRESSING PIC16CE62X



NOTES:

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 \overline{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 (i.e., 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>).





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.





6.0 EEPROM PERIPHERAL OPERATION

The PIC16CE623/624/625 each have 128 bytes of EEPROM data memory. The EEPROM data memory supports a bi-directional, 2-wire bus and data transmission protocol. These two-wires are serial data (SDA) and serial clock (SCL), and are mapped to bit1 and bit2, respectively, of the EEINTF register (SFR 90h). In addition, the power to the EEPROM can be controlled using bit0 (EEVDD) of the EEINTF register. For most applications, all that is required is calls to the following functions:

;	Byte_Write: Byte write routine
;	Inputs: EEPROM Address EEADDR
;	EEPROM Data EEDATA
;	Outputs: Return 01 in W if OK, else
;	return 00 in W
;	
;	Read_Current: Read EEPROM at address
cι	irrently held by EE device.
;	Inputs: NONE
;	Outputs: EEPROM Data EEDATA
;	Return 01 in W if OK, else
;	return 00 in W
;	
;	Read_Random: Read EEPROM byte at supplied
;	address
;	Inputs: EEPROM Address EEADDR
;	Outputs: EEPROM Data EEDATA
;	Return 01 in W if OK,
;	else return 00 in W

The code for these functions is available on our web site (www.microchip.com). The code will be accessed by either including the source code FL62XINC.ASM or by linking FLASH62X.ASM. FLASH62.IMC provides external definition to the calling program.

6.0.1 SERIAL DATA

SDA is a bi-directional pin used to transfer addresses and data into and data out of the memory.

For normal data transfer, SDA is allowed to change only during SCL low. Changes during SCL high are reserved for indicating the START and STOP conditions.

6.0.2 SERIAL CLOCK

This SCL input is used to synchronize the data transfer to and from the memory.

6.0.3 EEINTF REGISTER

The EEINTF register (SFR 90h) controls the access to the EEPROM. Register 6-1 details the function of each bit. User code must generate the clock and data signals.

REGISTER 6-1: EEINTF REGISTER (ADDRESS 90h)

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	
	—	_	_	_	EESCL	EESDA	EEVDD	R = Readable bit
bit7							bit0	 W = Writable bit U = Unimplemented bit, read as '0' n = Value at POR reset
bit 7-3:	Unimplen	nented: F	lead as '0'					
bit 2:	EESCL : C 1 = Clock 0 = Clock	EESCL: Clock line to the EEPROM 1 = Clock high 0 = Clock low						
bit 1:	EESDA : Data line to EEPROM 1 = Data line is high (pin is tri-stated, line is pulled high by a pull-up resistor) 0 = Data line is low							
bit 0:	EEVDD : VDD control bit for EEPROM 1 = VDD is turned on to EEPROM 0 = VDD is turned off to EEPROM (all pins are tri-stated and the EEPROM is powered down)							
Note:	EESDA, EESCL and EEVDD will read '0' if EEVDD is turned off.							

6.1 Bus Characteristics

In this section, the term "processor" refers to the portion of the PIC16CE62X that interfaces to the EEPROM through software manipulating the EEINTF register. The following **bus protocol** is to be used with the EEPROM data memory.

- Data transfer may be initiated only when the bus is not busy.
- During data transfer, the data line must remain stable whenever the clock line is HIGH. Changes in the data line while the clock line is HIGH will be interpreted by the EEPROM as a START or STOP condition.

Accordingly, the following bus conditions have been defined (Figure 6-1).

6.1.1 BUS NOT BUSY (A)

Both data and clock lines remain HIGH.

6.1.2 START DATA TRANSFER (B)

A HIGH to LOW transition of the SDA line while the clock (SCL) is HIGH determines a START condition. All commands must be preceded by a START condition.

6.1.3 STOP DATA TRANSFER (C)

A LOW to HIGH transition of the SDA line while the clock (SCL) is HIGH determines a STOP condition. All operations must be ended with a STOP condition.

6.1.4 DATA VALID (D)

The state of the data line represents valid data when, after a START condition, the data line is stable for the duration of the HIGH period of the clock signal.

The data on the line must be changed during the LOW period of the clock signal. There is one bit of data per clock pulse.

Each data transfer is initiated with a START condition and terminated with a STOP condition. The number of the data bytes transferred between the START and STOP conditions is determined by the processor and is theoretically unlimited, although only the last sixteen will be stored when doing a write operation. When an overwrite does occur, it will replace data in a first-in, first-out fashion.

6.1.5 ACKNOWLEDGE

The EEPROM will generate an acknowledge after the reception of each byte. The processor must generate an extra clock pulse which is associated with this acknowledge bit.

Note:	Acknowledge bits are not generated if an
	internal programming cycle is in progress.

When the EEPROM acknowledges, it pulls down the SDA line during the acknowledge clock pulse in such a way that the SDA line is stable LOW during the HIGH period of the acknowledge related clock pulse. Of course, setup and hold times must be taken into account. The processor must signal an end of data to the EEPROM by not generating an acknowledge bit on the last byte that has been clocked out of the EEPROM. In this case, the EEPROM must leave the data line HIGH to enable the processor to generate the STOP condition (Figure 6-2).

7.3.1 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 7-1) must be executed when changing the prescaler assignment from Timer0 to WDT.

EXAMPLE 7-1: CHANGING PRESCALER (TIMER0 \rightarrow 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 7-2. This precaution must be taken even if the WDT is disabled.

EXAMPLE 7-2: CHANGING PRESCALER (WDT \rightarrow TIMER0)

CLRWDT		;Clear WDT and
		;prescaler
BSF	STATUS, RPO	
MOVLW	b'xxxx0xxx'	;Select TMR0, new
		;prescale value and
		;clock source
MOVWF	OPTION_REG	
BCF	STATUS, RPO	

TABLE 7-1: 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	Timer0 module register							xxxx xxxx	uuuu uuuu
0Bh/8Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
81h	OPTION	RBPU	INTEDG	TOCS	T0SE	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', x = unknown, u = unchanged.

Note: Shaded bits are not used by TMR0 module.

9.0 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 Register 9-1. The block diagram is given in Figure 9-1.

9.1 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

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 13-1). Example 9-1 shows an example of how to configure the Voltage Reference for an output voltage of 1.25V with VDD = 5.0V.

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	R = Readable bit	
bit7		-			-		bit0	 W = Writable bit U = Unimplemented bit, read as '0' n = Value at POR reset 	
bit 7:	VREN: VREF 1 = VR 0 = VR	Enable EF circuit p EF circuit p	owerec	l on I down, no	IDD drain				
bit 6:	bit 6: VREF Output Enable 1 = VREF is output on RA2 pin 0 = VREF is disconnected from RA2 pin								
bit 5:	VRR: VREF 1 = LO 0 = Hig	Range sel w Range gh Range	ection						
bit 4:	Unimplem	ented: Rea	ad as '0	li.					
bit 3-0:	VR<3:0>: V when V when V	/REF value /RR = 1: VI /RR = 0: VI	selectio REF = (\ REF = 1/	on 0 ≤ Vr [3 /r<3:0>/ 24 /4 * Vdd +	3:0] ≤ 15 4) * VDD (VR<3:0>/ 3	32) * Vdd			

REGISTER 9-1: VRCON REGISTER (ADDRESS 9Fh)

FIGURE 9-1: VOLTAGE REFERENCE BLOCK DIAGRAM



10.4 <u>Power-on Reset (POR), Power-up</u> <u>Timer (PWRT), Oscillator Start-up</u> <u>Timer (OST) and Brown-out Reset</u> (BOD)

10.4.1 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 $\overline{\text{MCLR}}$ 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".

10.4.2 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, PWRTE, 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.

10.4.3 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.

10.4.4 BROWN-OUT RESET (BOD)

The PIC16CE62X members have on-chip Brown-out Reset circuitry. A configuration bit, BOREN, can disable (if clear/programmed) or enable (if set) the Brown-out Reset circuitry. If VDD falls below 4.0V (refer to BVDD parameter D005) for greater than parameter (TBOR) in Table 13-5, 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, Watch-dog, etc.) the chip will remain in reset until VDD rises above BVDD. The Power-up Timer will then 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 10-7 shows typical Brown-out situations.



FIGURE 10-7: BROWN-OUT SITUATIONS

11.1 Instruction Descriptions

ADDLW	Add Literal and W								
Syntax:	[<i>label</i>] ADDLW k								
Operands:	$0 \le k \le 2$	$0 \le k \le 255$							
Operation:	$(W) + k \rightarrow (W)$								
Status Affected:	C, DC, Z								
Encoding:	11	111x	kkkk	kkkk					
Description:	The conte added to the result is pl	The contents of the W register are added to the eight bit literal 'k' and the result is placed in the W register.							
Words:	1								
Cycles:	1								
Example	ADDLW	0x15							
	Before In After Inst	struction W = ruction	0x10						
		vv =	UX25						

ANDLW	AND Literal with W								
Syntax:	[<i>label</i>] ANDLW k								
Operands:	$0 \le k \le 255$								
Operation:	(W) .AND. (k) \rightarrow (W)								
Status Affected:	Z								
Encoding:	11	1001	kkkk	kkkk					
Description:	The contents of W register are AND'ed with the eight bit literal 'k'. The result is placed in the W register.								
Words:	1								
Cycles:	1								
Example	ANDLW	0x5F							
	Before In After Inst	struction W = ruction W =	0xA3 0x03						

ADDWF	Add W and f						
Syntax:	[label] ADDWF f,d						
Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ d \in [0,1] \end{array}$						
Operation:	$(W) + (f) \rightarrow (dest)$						
Status Affected:	C, DC, Z						
Encoding:	00 0111 dfff ffff						
Description:	Add the contents of the W register with 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	ADDWF FSR, 0						
	Before Instruction W = 0x17 FSR = 0xC2 After Instruction W = 0xD9 FSR = 0xC2						

ANDWF	AND W with f						
Syntax:	[label] ANDWF f,d						
Operands:	$0 \le f \le 127$ $d \in [0,1]$						
Operation:	(W) .AND. (f) \rightarrow (dest)						
Status Affected:	Z						
Encoding:	00 0101 dfff ffff						
Description:	AND the W register with 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	ANDWF FSR, 1						
	Before Instruction W = 0x17 FSR = 0xC2 After Instruction W = 0x17 FSR = 0x02						

RETURN	Return from Subroutine	RRF	Rotate Right f through Carry			
Syntax:	[label] RETURN	Syntax:	[<i>label</i>] RRF f,d			
Operands:	None	Operands:	$0 \leq f \leq 127$			
Operation:	$TOS \rightarrow PC$		$d \in [0,1]$			
Status Affected:	None	Operation:	See description below			
Encoding:	00 0000 0000 1000	Status Affected:	С			
Description:	Return from subroutine. The stack is	Encoding:	00 1100 dfff ffff			
Words:	POPed and the top of the stack (TOS) is loaded into the program counter. This is a two cycle instruction. 1	Description:	The contents of register 'f' are rotated one bit to the right through the Carry Flag. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is			
Cycles:	2					
Example	RETURN					
	After Interrupt	Words:	1			
	PC = TOS	Cycles:	1			
		Example	RRF REG1,0			
			Before Instruction			
			REG1 = 1110 0110 C = 0			
			After Instruction			
			$\begin{array}{rcl} \mathbf{REG1} &=& 1110 & 0110 \\ \mathbf{W} &=& 0111 & 0011 \end{array}$			
			C = 0			

RLF	Rotate	Left f thi	rough	o Carr	у				
Syntax:	[label]	RLF	f,d						
Operands:	0 ≤ f ≤ 1 d ∈ [0,1	27]							
Operation:	See des	See description below							
Status Affected:	С	С							
Encoding:	0 0	1101	df	ff	ffff				
	one bit to Flag. If 'c the W res stored ba	The contents of register 'f' are rotated one bit to the left through the Carry Flag. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is stored back in register 'f'.							
Words:	1								
Cycles:	1								
Example	RLF	RE	G1,0						
	Before I	nstructio	n						
		REG1	=	1110	0110				
	A ft a v lua e	C	=	0					
	Alterins			1110	0110				
		W	=	1100	1100				
		C	=	1	1100				

SLEEP

Syntax:	[label]	SLEEP)			
Operands:	None					
Operation:	$\begin{array}{l} 00h \rightarrow WDT, \\ 0 \rightarrow WDT \ prescaler, \\ 1 \rightarrow \overline{TO}, \\ 0 \rightarrow \overline{PD} \end{array}$					
Status Affected:	TO, PD					
Encoding:	0 0	0000	0110	0011		
Description:	The power-down status bit, \overline{PD} is cleared. Time-out status bit, \overline{TO} is set. Watchdog Timer and its prescaler are cleared. The processor is put into SLEEP mode with the oscillator stopped.					
Words:	1					
Cycles:	1					
Example:	SLEEP					

SWAPF	Swap Nib	bles in	f		XORLW	Exclusiv	ve OR L	iteral wit	th W	
Syntax:	[label] S	SWAPF	f,d		Syntax:	[<i>label</i>] XORLW k				
Operands:	$0 \le f \le 127$ $d \in [0,1]$ $(f<3:0>) \rightarrow (dest<7:4>),$ $(f<7:4>) \rightarrow (dest<3:0>)$			Operands:	$0 \le k \le 255$ (W) .XOR. $k \rightarrow$ (W) Z					
Operation:				Status Affected:						
Status Affected:	None				Encoding:	11	1010	kkkk	kkkk	
Encoding:	0 0	1110	dfff	ffff	Description:	The contents of the W register are				
Description:	The upper and lower nibbles of register 'f' are exchanged. If 'd' is 0, the result is placed in W register. If 'd'			Words:	The result is placed in the W register.					
Marda	is 1, the res	sult is pla	ced in reg	ister 't'.	Cycles:	1				
Cycles:	1				Example:	XORLW	0xAF			
Example	SWAPF R	EG,	0			Before I	nstructio	n		
	Before Ins	truction					W =	0xB5		
		REG1	= 0x/	45		After Instruction				
	After Instru	uction					W =	0x1A		
		REG1 W	= 0x/ = 0x5	45 5A						

TRIS	Load TRIS Register
Syntax:	[label] TRIS f
Operands:	$5 \le f \le 7$
Operation:	(W) \rightarrow TRIS register f;
Status Affected:	None
Encoding:	00 0000 0110 0fff
Description: Words: Cycles: Example	The instruction is supported for code compatibility with the PIC16C5X products. Since TRIS registers are readable and writable, the user can directly address them. 1
	To maintain upward compatibility with future PIC [®] MCU products, do not use this instruction.

XORWF	Exclusive OR W with f							
Syntax:	[label]	XORWF	f,d					
Operands:	$\begin{array}{l} 0 \leq f \leq 12 \\ d \in [0,1] \end{array}$	7						
Operation:	(W) .XOR. (f) \rightarrow (dest)							
Status Affected:	Z							
Encoding:	0 0	0110	dff	Ē	ffff			
Description:	Exclusive (W register the result is 'd' is 1, the ister 'f'.	OR the co with regis s stored ir result is s	ntents ster 'f'. In the V stored	s of t If 'd V reg bac	he ' is 0, gister. If k in reg-			
Words:	1							
Cycles:	1							
Example	XORWF	REG 3	1					
	Before In:	struction						
		REG W	= =	0xA 0xE	AF 35			
	After Inst	ruction						
		REG W	= =	Ox1 OxE	I A 35			

12.0 DEVELOPMENT SUPPORT

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

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

12.1 <u>MPLAB Integrated Development</u> <u>Environment Software</u>

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

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

MPLAB allows you to:

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

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

12.2 MPASM Assembler

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

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

MPASM features include:

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

12.3 <u>MPLAB-C17 and MPLAB-C18</u> <u>C Compilers</u>

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

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

12.4 MPLINK/MPLIB Linker/Librarian

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

13.2 DC CHARACTERISTICS: F

PIC16LCE62X-04 (Commercial, Industrial)

			Standard Operating Conditions (unless otherwise stated)					
DC CH		STICS	Operating temperature $-40^{\circ}C \le TA \le +85^{\circ}C$ for industrial and $0^{\circ}C \le TA \le +70^{\circ}C$ for commercial and					
						-4	$0^{\circ}C \leq TA \leq +125^{\circ}C$ for extended	
Param	Sym	Characteristic	Min	Typ†	Max	Units	Conditions	
No.								
D001	Vdd	Supply Voltage	2.5	-	5.5	V	See Figure 13-1 through Figure 13-3	
D002	Vdr	RAM Data Retention Voltage (Note 1)	-	1.5*	-	V	Device in SLEEP mode	
D003	VPOR	VDD start voltage to ensure Power-on Reset	-	Vss	-	V	See section on power-on reset for details	
D004	SVDD	VDD rise rate to ensure Power-on Reset	.05*	-	-	V/ms	See section on power-on reset for details	
D005	VBOR	Brown-out Detect Voltage	3.7	4.0	4.35	V	BOREN configuration bit is cleared	
D010	IDD	Supply Current (Note 2)	-	1.2	2.0	mA	Fosc = 4 MHz, VDD = 5.5V, WDT disabled,	
							XT osc mode, (Note 4)*	
			_	_	1.1	mA	FOSC = 4 MIHZ, $VDD = 2.5$ V, WDT disabled, XT osc mode (Note 4)	
			_	35	70	μA	Fosc = 32 kHz , VDD = 2.5V, WDT disabled,	
						•	LP osc mode	
D020	IPD	Power Down Current (Note 3)	-	-	2.0	μA	VDD = 2.5V	
			-	-	2.2	μA	VDD = 3.0V*	
			-	-	9.0	μA A	VDD = 5.5V	
D000	ALWET	WDT Ourrent (Note 5)	_	-	10	μΑ		
D022	ΔIWDT	WDT Current (Note 5)	-	6.0	10	μΑ	VDD=4.0V (125°C)	
D022A	ΔIBOR	Brown-out Reset Current	_	75	125	μA uA	$\frac{(123)}{BOD}$ enabled. VDD = 5.0V	
		(Note 5)				P		
D023		Comparator Current for each	-	30	60	μA	VDD = 4.0V	
00224		Comparator (Note 5)		80	125		$V_{DD} = 4.0 V$	
DUZSA		Operating Current	_	80	2	μA mA	$V_{00} = 4.0V$	
		Operating Current	_		1	mA	VCC = 5.5V, SCL = 400 KHz	
		Standby Current	_		30	μA	VCC = 3.0V, EE VDD = VCC	
	ΔIEE	Standby Current	-		100	μΑ	VCC = 3.0V, EE VDD = VCC	
1A	Fosc	LP Oscillator Operating Frequency	0		200	kHz	All temperatures	
		RC Oscillator Operating Frequency	0	—	4	MHz	All temperatures	
		XT Oscillator Operating Frequency	0	—	4	MHz	All temperatures	
		HS Oscillator Operating Frequency	0	—	20	MHz	All temperatures	

* 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.

3: The power down current in SLEEP mode does not depend on the oscillator type. Power down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to VDD or Vss.

4: 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 k Ω .

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.

18-Lead Plastic Small Outline (SO) - Wide, 300 mil (SOIC)

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



	Units		INCHES*		N	IILLIMETERS	6
Dimension	Limits	MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		18			18	
Pitch	р		.050			1.27	
Overall Height	Α	.093	.099	.104	2.36	2.50	2.64
Molded Package Thickness	A2	.088	.091	.094	2.24	2.31	2.39
Standoff	A1	.004	.008	.012	0.10	0.20	0.30
Overall Width	E	.394	.407	.420	10.01	10.34	10.67
Molded Package Width	E1	.291	.295	.299	7.39	7.49	7.59
Overall Length	D	.446	.454	.462	11.33	11.53	11.73
Chamfer Distance	h	.010	.020	.029	0.25	0.50	0.74
Foot Length	L	.016	.033	.050	0.41	0.84	1.27
Foot Angle	φ	0	4	8	0	4	8
Lead Thickness	с	.009	.011	.012	0.23	0.27	0.30
Lead Width	В	.014	.017	.020	0.36	0.42	0.51
Mold Draft Angle Top	α	0	12	15	0	12	15
Mold Draft Angle Bottom	β	0	12	15	0	12	15

*Controlling Parameter

Notes:

Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side. JEDEC Equivalent: MS-013

Drawing No. C04-051

APPENDIX A: CODE FOR ACCESSING EEPROM DATA MEMORY

Please check our web site at www.microchip.com for code availability.

APPENDIX B:REVISION HISTORY

Revision D (January 2013)

Added a note to each package outline drawing.

INDEX

Α	
ADDLW Instruction	
ADDWF Instruction	
ANDLW Instruction	
ANDWF Instruction	
Architectural Overview	7
Assembler	
MPASM Assembler	77
в	

В

BCF Instruction	
Block Diagram	
TIMER0	
TMR0/WDT PRESCALER	
Brown-Out Detect (BOD)	
BSF Instruction	
BTFSC Instruction	
BTFSS Instruction	
С	

CALL Instruction	
Clocking Scheme/Instruction Cycle	10
CLRF Instruction	69
CLRW Instruction	69
CLRWDT Instruction	70
CMCON Register	41
Code Protection	64
COMF Instruction	70
Comparator Configuration	42
Comparator Interrupts	45
Comparator Module	41
Comparator Operation	
Comparator Reference	
Configuration Bits	50
Configuring the Voltage Reference	47
Crystal Operation	51
_	

D

Data Memory Organization	12
DECF Instruction	70
DECFSZ Instruction	70
Development Support	77
E	

Е

EEPROM Peripheral Operation	29
Errata	2
External Crystal Oscillator Circuit	52

G

General purpose Register File	12
GOTO Instruction	71

L

I/O Ports	23
I/O Programming Considerations	
ID Locations	64
INCF Instruction	71
INCFSZ Instruction	71
In-Circuit Serial Programming	64
Indirect Addressing, INDF and FSR Registers	21
Instruction Flow/Pipelining	10
Instruction Set	
ADDLW	67
ADDWF	67
ANDLW	67
ANDWF	67
BCF	68
BSF	
-	

BTFSC	68
BTFSS	69
CALL	69
	69
	09 70
COMF	70
DECF	70
DECFSZ	70
GOTO	71
	71
	/1 71
IORWE	72
MOVF	72
MOVLW	72
MOVWF	72
NOP	73
OPTION	73
	73
BETUBN	73
RLF	74
RRF	74
SLEEP	74
SUBLW	75
SUBWF	75
TRIS	76
	76
XORUW	76
XORWF	76 65
XORWF	76 65 60
XORWF	76 65 60 17
XORWF	76 65 60 17 59 71
XORWF	76 65 60 17 59 71 72
XORWF Instruction Set Summary INT Interrupt INTCON Register Interrupts IORLW Instruction IORWF Instruction K	76 65 17 59 71 72
XORWF	76 65 60 17 59 71 72 80
XORWF	76 65 60 17 59 71 72 80
XORWF	76 65 60 17 59 71 72 80
XORWF	 76 65 60 17 59 71 72 80 72 72 72
XORWF	76 65 60 17 59 71 72 80 72 72 72 72
XORWF	76 65 60 17 59 71 72 80 72 72 72 72 72
XORWF	76 65 60 17 59 71 72 80 72 72 72 77
XORWF	76 65 60 17 59 71 72 80 72 72 72 72 77 73
XORWF	76 65 60 17 59 71 72 80 72 72 72 77 73
XORWF	76 65 60 17 59 71 72 80 72 72 77 73 . 5
XORWF	76 65 60 17 59 71 72 80 72 72 72 77 73 . 5 73
XORWF	76 65 60 17 59 71 72 80 72 72 72 77 73 . 5 73 16
XORWF	76 65 60 17 59 71 72 80 72 72 72 73 .5 73 16 51
XORWF	76 60 17 59 71 72 80 72 72 77 73 . 5 73 16 51 54
XORWF	76 60 17 59 71 72 80 72 72 72 77 73 . 5 73 16 51 54
XORWF	76 65 60 17 59 71 72 80 72 72 77 73 . 5 73 16 51 54 01 77
XORWF	76 60 17 59 71 72 80 72 72 77 73 .5 73 16 51 54 01 97 20
XORWF	76 60 17 57 72 80 72 72 77 73 .5 73 61 51 40 97 20 19
XORWF	76 65 60 17 59 71 72 80 72 72 77 73 .5 316 51 97 20 197 20 197 9
XORWF	76 65 60 17 59 71 72 80 72 72 77 73 .5 73 61 51 4 97 20 197 79

PICSTART® Plus Entry Level Development System 79 PIE1 Register 18 PIR1 Register 18