



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	I ² C, SPI
Peripherals	LCD, POR, PWM, WDT
Number of I/O	25
Program Memory Size	7KB (4K x 14)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	176 x 8
Voltage - Supply (Vcc/Vdd)	4V ~ 6V
Data Converters	A/D 5x8b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16c924-04i-pt

Email: info@E-XFL.COM

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



FIGURE 4-2: REGISTER FILE MAP

	Address		Address		Address	A	Address
Indirect addr.(1)	00h	Indirect addr.(1)	80h	Indirect addr.(1)	100h	Indirect addr.(1)	180h
TMR0	01h	OPTION	81h	TMR0	101h	OPTION	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h	PORTF	107h	TRISF	187h
PORTD	08h	TRISD	88h	PORTG	108h	TRISG	188h
PORTE	09h	TRISE	89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch		10Ch		18Ch
	0Dh		8Dh	LCDSE	10Dh		18Dh
TMR1L	0Eh	PCON	8Eh	LCDPS	10Eh		18Eh
TMR1H	0Fh		8Fh	LCDCON	10Fh		18Fh
T1CON	10h		90h	LCDD00	110h		190h
TMR2	11h		91h	LCDD01	111h		191h
T2CON	12h	PR2	92h	LCDD02	112h		192h
SSPBUF	13h	SSPADD	93h	LCDD03	113h		193h
SSPCON	14h	SSPSTAT	94h	LCDD04	114h		194h
CCPR1L	15h		95h	LCDD05	115h		195h
CCPR1H	16h		96h	LCDD06	116h		196h
CCP1CON	17h		97h	LCDD07	117h		197h
	18h		98h	LCDD08	118h		198h
	19h		99h	LCDD09	119h		199h
	1Ah		9Ah	LCDD10	11Ah		19Ah
	1Bh		9Bh	LCDD11	11Bh		19Bh
	1Ch		9Ch	LCDD12	11Ch		19Ch
	1Dh		9Dh	LCDD13	11Dh		19Dh
ADRES ⁽²⁾	1Eh		9Eh	LCDD14	11Eh		19Eh
ADCON0 ⁽²⁾	1Fh	ADCON1 ⁽²⁾	9Fh	LCDD15	11Fh		19Fh
	20h		A0h		120h		1A0h
General Purpose		General Purpose Register					
Register			EFh		16F		1EFh
	75	Mapped in Bank 0 70h-7Fh	F0h	Mapped in Bank 0 70h-7Fh	170	Mapped in Bank 0 70h-7Fh	1F0h
Bank 0	」/⊢h	Bank 1	FFN	Bank 2	176	Bank 3	TFFh
Bank 0		Bank 1		Bank 2		Bank 3	

4.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers are registers used by the CPU and Peripheral Modules for controlling the desired operation of the device. These registers are implemented as static RAM.

The special function registers can be classified into two sets (core and peripheral). Those registers associated with the "core" functions are described in this section, and 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 Power-on Reset	Value on all other resets
Bank 0	Bank 0										
00h	INDF	Addressing	this location	egister)	0000 0000	0000 0000					
01h	TMR0	Timer0 mod	lule's register	r						xxxx xxxx	uuuu uuuu
02h	PCL	Program Co	ounter's (PC)	Least Signifi	cant Byte					0000 0000	0000 0000
03h	STATUS	IRP	RP1	RP0	TO	PD	Z	DC	С	0001 1xxx	000q quuu
04h	FSR	Indirect data	a memory ad	dress pointe	r					XXXX XXXX	uuuu uuuu
05h	PORTA	—	_	PORTA Dat	a Latch wher	n written: POI	RTA pins whe	en read		(4)	(4)
06h	PORTB	PORTB Dat	a Latch wher	n written: PO	RTB pins wh	en read				XXXX XXXX	uuuu uuuu
07h	PORTC	—	_	PORTC Dat	a Latch whe	n written: PO	RTC pins wh	en read		xx xxxx	uu uuuu
08h	PORTD	PORTD Dat	a Latch whe	n written: PO	RTD pins wh	ien read				0000 0000	0000 0000
09h	PORTE	PORTE pins	s when read							0000 0000	0000 0000
0Ah	PCLATH	—	_	—	Write Buffer	for the uppe	r 5 bits of the	e Program Co	ounter	0 0000	0 0000
0Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	LCDIF	ADIF ⁽²⁾	—	_	SSPIF	CCP1IF	TMR2IF	TMR1IF	00 0000	00 0000
0Dh	_	Unimpleme	nted							_	_
0Eh	TMR1L	Holding regi	ister for the L		XXXX XXXX	uuuu uuuu					
0Fh	TMR1H	Holding regi	ister for the N	lost Significa	ant Byte of th	e 16-bit TMR	1 register	_		xxxx xxxx	uuuu uuuu
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR10N	00 0000	uu uuuu
11h	TMR2	Timer2 mod	lule's register	r						0000 0000	0000 0000
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
13h	SSPBUF	Synchronou	is Serial Port	Receive Buf	fer/Transmit	Register			-	XXXX XXXX	uuuu uuuu
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
15h	CCPR1L	Capture/Co	mpare/PWM	Register (LS	iB)					XXXX XXXX	uuuu uuuu
16h	CCPR1H	Capture/Co	mpare/PWM	Register (MS	SB)		-			XXXX XXXX	uuuu uuuu
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	00 0000	00 0000
18h	_	Unimpleme	nted							—	—
19h	_	Unimpleme	nted							—	—
1Ah	_	Unimpleme	nted							—	—
1Bh	_	Unimpleme	nted							—	—
1Ch	-	Unimpleme	nted							—	—
1Dh	-	Unimpleme	nted							-	—
1Eh ⁽¹⁾	ADRES	A/D Result I	Register				-			XXXX XXXX	uuuu uuuu
1Fh ⁽¹⁾	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	(5)	ADON	0000 0000	0000 0000

TABLE 4-1: SPECIAL FUNCTION REGISTER SUMMARY

 ${\rm x}$ = unknown, ${\rm u}$ = unchanged, ${\rm q}$ = value depends on condition, - = unimplemented read as '0', Legend: shaded locations are unimplemented, read as '0'.

1: Registers ADRES, ADCON0, and ADCON1 are not implemented in the PIC16C923, read as '0'. Note

2: These bits are reserved on the PIC16C923, always maintain these bits clear.

3: These pixels do not display, but can be used as general purpose RAM.

4: PIC16C923 reset values for PORTA: --xx xxxx for a POR, and --uu uuuu for all other resets, PIC16C924 reset values for PORTA: --0x 0000 when read.

5: Bit1 of ADCON0 is reserved on the PIC16C924, always maintain this bit clear.

4.2.2.2 OPTION REGISTER

The OPTION register is a readable and writable register which contains various control bits to configure the TMR0/WDT prescaler, the external RB0/INT pin interrupt, TMR0, and the weak pull-ups on PORTB. **Note:** To achieve a 1:1 prescaler assignment for the TMR0 register, assign the prescaler to the Watchdog Timer.

FIGURE 4-4: OPTION REGISTER (ADDRESS 81h, 181h)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1					
RBPU	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0	R = Readable bit				
bit7							bit0	W = Writable bit U = Unimplemented bit, read as '0' - n = Value at POR reset				
bit 7:	RBPU : PORTB Pull-up Enable bit 1 = PORTB pull-ups are disabled 0 = PORTB pull-ups are enabled by individual port latch values											
bit 6:	INTEDG: Interrupt Edge Select bit 1 = Interrupt on rising edge of RB0/INT pin 0 = Interrupt on falling edge of RB0/INT pin											
bit 5:	TOCS : TMR0 Clock Source Select bit 1 = Transition on RA4/T0CKI pin 0 = Internal instruction cycle clock (CLKOUT)											
bit 4:	TOSE : TMF 1 = Increm 0 = Increm	R0 Source ent on hig ent on lov	e Edge Se gh-to-low w-to-high	elect bit transition transition	on RA4/T00 on RA4/T00	CKI pin CKI pin						
bit 3:	PSA : Pres 1 = Presca 0 = Presca	caler Ass Iler is ass Iler is ass	ignment b igned to t igned to t	bit he WDT he Timer0	module							
bit 2-0:	PS2:PS0 :	Prescaler	Rate Sel	ect bits								
	Bit Value	TMR0 R	ate WD	Γ Rate								
	000 001 010 100 101 110 111	1:2 1:4 1:8 1:16 1:32 1:64 1:12 1:25	8 1: 8 1: 6 1:	1 2 4 8 16 32 64 128								

5.2 PORTB and TRISB Register

PORTB is an 8-bit wide bi-directional port. The corresponding data direction register is TRISB. Setting a bit in the TRISB register puts the corresponding output driver in a hi-impedance input mode. Clearing a bit in the TRISB register puts the contents of the output latch on the selected pin(s).

EXAMPLE 5-2: INITIALIZING PORTB

BCF	STATUS,	RP0	;	Select Bank0
BCF	STATUS,	RP1		
CLRF	PORTB		;	Initialize PORTB
BSF	STATUS,	RP0	;	
MOVLW	0xCF		;	Value used to
			;	initialize data
			;	direction
MOVWF	TRISB		;	Set RB<3:0> as inputs
			;	RB<5:4> as outputs
			;	RB<7:6> as inputs

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

FIGURE 5-3: BLOCK DIAGRAM OF RB3:RB0 PINS



Four of PORTB's pins, RB7:RB4, have an interrupt on change feature. Only pins configured as inputs can cause this interrupt to occur (i.e. any RB7:RB4 pin configured as an output is excluded from the interrupt on change comparison). The input pins (of RB7:RB4) are compared with the old value latched on the last read of PORTB. The "mismatch" outputs of RB7:RB4 are OR'ed together to generate the RB Port Change Interrupt with flag bit RBIF (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 keypad and make it possible for wake-up on key-depression. Refer to the *Embedded Control Handbook, "Implementing Wake-Up on Key Stroke"* (AN552).

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.

FIGURE 5-4: BLOCK DIAGRAM OF RB7:RB4 PINS



Name	Bit#	Buffer	Function
RB0/INT	bit0	TTL/ST	Input/output pin or external interrupt input. Internal software programmable weak pull-up. This buffer is a Schmitt Trigger input when configured as the external interrupt.
RB1	bit1	TTL	Input/output pin. Internal software programmable weak pull-up.
RB2	bit2	TTL	Input/output pin. Internal software programmable weak pull-up.
RB3	bit3	TTL	Input/output pin. Internal software programmable weak pull-up.
RB4	bit4	TTL	Input/output pin (with interrupt on change). Internal software programmable weak pull-up.
RB5	bit5	TTL	Input/output pin (with interrupt on change). Internal software programmable weak pull-up.
RB6	bit6	TTL/ST	Input/output pin (with interrupt on change). Internal software programmable weak pull-up. Serial programming clock. This buffer is a Schmitt Trigger input when used in serial programming mode.
RB7	bit7	TTL/ST	Input/output pin (with interrupt on change). Internal software programmable weak pull-up. Serial programming data. This buffer is a Schmitt Trigger input when used in serial programming mode.

TABLE 5-3: PORTB FUNCTIONS

Legend: TTL = TTL input, ST = Schmitt Trigger input

TABLE 5-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other resets
06h, 106h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	uuuu uuuu
86h, 186h	TRISB	PORTB	Data Direct	1111 1111	1111 1111						
81h, 181h	OPTION	RBPU	INTEDG	TOCS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

Legend: x = unknown, u = unchanged. Shaded cells are not used by PORTB.

5.8 <u>I/O Programming Considerations</u>

5.8.1 BI-DIRECTIONAL I/O PORTS

Any instruction which writes, operates internally as a read followed by a write operation. The BCF and BSF instructions, for example, read the register into the CPU, execute the bit operation and write the result back to the register. Caution must be used when these instructions are applied to a port with both inputs and outputs defined. For example, a BSF operation on bit5 of PORTB will cause all eight bits of PORTB to be read into the CPU. Then the BSF operation takes place on bit5 and PORTB is written to the output latches. If another bit of PORTB is used as a bi-directional I/O pin (e.g., bit0) and it is defined as an input at this time, the input signal present on the pin itself would be read into the CPU and 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 bit0 is switched into output mode later on, the contents of the data latch may now be unknown.

Reading the port register, reads the values of the port pins. Writing to the port register writes the value to the port latch. When using read-modify-write instructions (ex. BCF, BSF) 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 5-8 shows the effect of two sequential read-modify-write instructions on an I/O port.

EXAMPLE 5-8: 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-ups and are
;not connected to other circuitry
.

/			
;		PORT latch	PORT pins
;			
BCF PORTB, 7	;	01pp pppp	11pp pppp
BCF PORTB, 6	;	10pp pppp	11pp pppp
BCF STATUS, RP1	;		
BSF STATUS, RPO	;		
BCF TRISB, 7	;	10pp pppp	11pp pppp
BCF TRISB, 6	;	10pp pppp	10pp pppp
;			
;Note that the use	er	may have exp	ected the
;pin values to be	0	0pp ppp. The	2nd BCF
	-		

;caused RB7 to be latched as the pin value ;(high).

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.

5.8.2 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 5-11). 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.

PC	X PC	X PC + 1	X	PC + 2	PC + 3	This example shows a write to PORT
fetched	MOVWF PORTB write to PORTB	MOVF PORTB,W		NOP	NOP	Note that:
RB7:RB0		<u>.</u>	X			data setup time = (0.25TCY - TPD)
1		1 1 1 1		Port pin sampled here	1 1 1	where Tcy = instruction cycle TPD = propagation delay
Instruction executed		MOVWF PORTB write to PORTB	MO	VF PORTB,W	NOP	Therefore, at higher clock frequencie a write followed by a read may be pro lematic.
1		1	1	1	1 1	

FIGURE 5-11: SUCCESSIVE I/O OPERATION

TABLE 9-1: REGISTERS ASSOCIATED WITH TIMER2 AS A TIMER/COUNTER

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other resets
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	LCDIF	ADIF ⁽¹⁾	_		SSPIF	CCP1IF	TMR2IF	TMR1IF	00 0000	00 0000
8Ch	PIE1	LCDIE	ADIE ⁽¹⁾	_	-	SSPIE	CCP1IE	TMR2IE	TMR1IE	00 0000	00 0000
11h	TMR2	Timer2 module's register									0000 0000
12h	T2CON	_	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
92h	PR2	Timer2 Period Register									1111 1111

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used by the Timer2 module. Note 1: Bits ADIE and ADIF are reserved on the PIC16C923, always maintain these bits clear.

13.2 LCD Interrupts

The LCD timing generation provides an interrupt that defines the LCD frame timing. This interrupt can be used to coordinate the writing of the pixel data with the start of a new frame. Writing pixel data at the frame boundary allows a visually crisp transition of the image. This interrupt can also be used to synchronize external events to the LCD. For example, the interface to an external segment driver, such as a Microchip AY0438, can be synchronized for segment data update to the LCD frame. A new frame is defined to begin at the leading edge of the COM0 common signal. The interrupt will be set immediately after the LCD controller completes accessing all pixel data required for a frame. This will occur at a certain fixed time before the frame boundary as shown in Figure 13-9. The LCD controller will begin to access data for the next frame within TFWR after the interrupt.





TABLE 14-6: INITIALIZATION CONDITIONS FOR ALL REGISTERS	(Cont.'d)
---	-----------

Register	Applicab	le Devices	Power-on Reset	MCLR Resets WDT Reset	Wake-up via WDT or Interrupt
LCDSE	923	924	1111 1111	1111 1111	սսսս սսսս
LCDPS	923	924	0000	0000	uuuu
LCDCON	923	924	00-0 0000	00-0 0000	uu-u uuuu
LCDD00 to LCDD15	923	924	XXXX XXXX	นนนน นนนน	սսսս սսսս
TRISF	923	924	1111 1111	1111 1111	uuuu uuuu
TRISG	923	924	1111 1111	1111 1111	uuuu uuuu

Legend: u = unchanged, x = unknown, -= unimplemented bit, read as '0', q = value depends on condition Note 1: One or more bits in INTCON and/or PIR1 will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

3: See Table 14-5 for reset value for specific condition.

4: Bits PIE1<6> and PIR1<6> are reserved on the PIC16C923, always maintain these bits clear.

5: PORTA values when read.

TABLE 15-2: PIC16CXXX INSTRUCTION SET

Mnemonic,		Description	Cycles		14-Bit C	Opcode	e	Status	Notes
Operands				MSb			LSb	Affected	
BYTE-ORIE	NTED	FILE REGISTER OPERATIONS							
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C,DC,Z	1,2
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	1,2
CLRF	f	Clear f	1	00	0001	lfff	ffff	Z	2
CLRW	-	Clear W	1	00	0001	0xxx	xxxx	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	1,2
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1,2,3
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	1,2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1,2,3
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	1,2
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	1,2
MOVWF	f	Move W to f	1	00	0000	lfff	ffff		
NOP	-	No Operation	1	00	0000	0xx0	0000		
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	С	1,2
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	С	1,2
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z	1,2
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff		1,2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	1,2
BIT-ORIENT	ED FIL	E REGISTER OPERATIONS							
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		1,2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		3
LITERAL AI	ND CO	NTROL OPERATIONS							
ADDLW	k	Add literal and W	1	11	111x	kkkk	kkkk	C,DC,Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
CALL	k	Call subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDT	-	Clear Watchdog Timer	1	00	0000	0110	0100	TO,PD	
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk		
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLW	k	Move literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	-	Return from interrupt	2	00	0000	0000	1001		
RETLW	k	Return with literal in W	2	11	01xx	kkkk	kkkk		
RETURN	-	Return from Subroutine	2	00	0000	0000	1000		
SLEEP	-	Go into standby mode	1	00	0000	0110	0011	TO,PD	
SUBLW	k	Subtract W from literal	1	11	110x	kkkk	kkkk	C,DC,Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

Note 1: When an I/O register is modified as a function of itself (e.g., MOVF PORTB, 1), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

2: If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 Module.

3: If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

PIC16C9XX

BTFSS	Bit Test f	, Skip if S	Set		CALL	Call Sub	oroutine		
Syntax:	[<i>label</i>] BT	FSS f,b			Syntax:	[label]	CALL 4	(
Operands:	$0 \le f \le 12$	7			Operands:	$0 \leq k \leq 2047$			
Operation:	0 ≤ b < 7 skip if (f<	b>) = 1			Operation:	(PC)+1- $k \rightarrow PC <$	→ TOS, <10:0>,	> DC -12	.11.
Status Affected:	None		1		•	(PCLAIF	1<4.3>) -	→ PG<12	.11>
Encoding:	01	11bb	bfff	ffff	Status Affected:	None			
Description:	If bit 'b' in register 'f' is '0' then the next instruction is executed. If bit 'b' is '1', then the next instruction is discarded and a NOP is executed instead, making this a 2Tcy instruction.			Encoding: Description:	10 0kkk kkkk kkkk Call Subroutine. First, return address (PC+1) is pushed onto the stack. The eleven bit immediate address is loader into PC bits <10:0>. The upper bits of			ddress ck. The s loaded bits of	
Words:	1					the PC are	e loaded fr /cle instrue	om PCLA1	TH. CALL
Cycles:	1(2)				Words:	1			
Q Cycle Activity:	Q1	Q2	Q3	Q4	Cycles:	2			
	Decode	Read register 'f'	Process data	No- Operation	Q Cycle Activity:	Q1	Q2	Q3	Q4
If Skip:	(2nd Cycle)				1st Cycle	Decode	Read literal 'k',	Process data	Write to PC
	Q1	Q2	Q3	Q4			Push PC to Stack		
	No- Operation	No- Operation	No- Operation	No- Operation	2nd Cycle	No- Operation	No- Operation	No- Operation	No- Operation
Example	HERE FALSE TRUE	BTFSC GOTO •	FLAG,1 PROCESS_	_CODE	Example	HERE Before Ir	CALL	THERE	
		•				After Ins	PC = A truction	ddress HE	RE
	• Before Instruction PC = address HERE After Instruction if FLAG<1> = 0, PC = address FALSE if FLAG<1> = 1, FLAG<1> = 1,						PC = A TOS = A	ddress TH ddress HE	IERE RE+1

IORLW	Inclusive OR Literal with W								
Syntax:	[label]	IORLW	k						
Operands:	$0 \le k \le 2$	$0 \le k \le 255$							
Operation:	(W) .OR.	k ightarrow (W)							
Status Affected:	Z								
Encoding:	11	1000	kkkk	kkkk					
Description:	The contents of the W register is OR'ed with the eight bit literal 'k'. The result is placed in the W register.								
Words:	1								
Cycles:	1								
Q Cycle Activity:	Q1	Q2	Q3	Q4					
	Decode	Read literal 'k'	Process data	Write to W					
Example	IORLW	0x35							
	Before Instruction W = 0x9A After Instruction W = 0xBF Z = 1								

IORWF	Inclusive	e OR W v	with f				
Syntax:	[label]	IORWF	f,d				
Operands:	$\begin{array}{l} 0 \leq f \leq 12 \\ d \in \left[0,1\right] \end{array}$	27					
Operation:	(W) .OR.	$(f) \rightarrow (de)$	estination)			
Status Affected:	Z						
Encoding:	00	0100	dfff	ffff			
Description:	Inclusive OR the W register with regis- ter 'f'. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.						
Words:	1						
Cycles:	1						
Q Cycle Activity:	Q1	Q2	Q3	Q4			
	Decode	Read register 'f'	Process data	Write to destination			
Example	IORWF		RESULT,	0			
	Before In	struction					
		RESULT W	= 0x13 = 0x91	3			
	After Inst	ruction					
		RESULT W	= 0x13 = 0x93	3			

vv	=	07
Z	=	1

PIC16C9XX

XORLW	Exclusive OR Literal with W	XORWF	Exclusive OR W with f				
Syntax:	[<i>label</i>] XORLW k	Syntax:	[<i>label</i>] XORWF f,d				
Operands:	$0 \le k \le 255$	Operands:	0 ≤ f ≤ 127				
Operation:	(W) .XOR. $k \rightarrow (W)$		d ∈ [0,1]				
Status Affected:	Z	Operation:	(W) .XOR. (f) \rightarrow (destination)				
Encoding:	11 1010 kkkk kkkk	Status Affected:	Z				
Description:	The contents of the W register are	Encoding:	00 0110 dfff ffff				
	XOR'ed with the eight bit literal 'k'. The result is placed in the W regis- ter.	Description:	Exclusive OR 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	Words:	1				
Cycles:	1	vvolus.	1				
Q Cycle Activity:	Q1 Q2 Q3 Q4	Cycles:	1				
	Decode Read literal 'k' Process Write to W	Q Cycle Activity:	Q1 Q2 Q3 Q4 Decode Read register 'r' Process data Write to destination				
Example:	XORLW 0xAF						
	Before Instruction	Example	XORWF REG 1				
	W = 0xB5		Before Instruction				
	After Instruction W = 0x1A		REG = 0xAF W = 0xB5				
			After Instruction				
			$\begin{array}{rcl} REG &=& 0x1A\\ W &=& 0xB5 \end{array}$				

MPASM has the following features to assist in developing software for specific use applications.

- Provides translation of Assembler source code to object code for all Microchip microcontrollers.
- Macro assembly capability.
- Produces all the files (Object, Listing, Symbol, and special) required for symbolic debug with Microchip's emulator systems.
- Supports Hex (default), Decimal and Octal source and listing formats.

MPASM provides a rich directive language to support programming of the PICmicro. Directives are helpful in making the development of your assemble source code shorter and more maintainable.

16.11 Software Simulator (MPLAB-SIM)

The MPLAB-SIM Software Simulator allows code development in a PC host environment. It allows the user to simulate the PICmicro series microcontrollers on an instruction level. On any given instruction, the user may examine or modify any of the data areas or provide external stimulus to any of the pins. The input/ output radix can be set by the user and the execution can be performed in; single step, execute until break, or in a trace mode.

MPLAB-SIM fully supports symbolic debugging using MPLAB-C and MPASM. The Software Simulator offers the low cost flexibility to develop and debug code outside of the laboratory environment making it an excellent multi-project software development tool.

16.12 C Compiler (MPLAB-C)

The MPLAB-C Code Development System is a complete 'C' compiler and integrated development environment for Microchip's PICmicro family of micro-controllers. The compiler provides powerful integration capabilities and ease of use not found with other compilers.

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

16.13 <u>Fuzzy Logic Development System</u> (*fuzzy*TECH-MP)

*fuzzy*TECH-MP fuzzy logic development tool is available in two versions - a low cost introductory version, MP Explorer, for designers to gain a comprehensive working knowledge of fuzzy logic system design; and a full-featured version, *fuzzy*TECH-MP, edition for implementing more complex systems.

Both versions include Microchip's *fuzzy*LAB[™] demonstration board for hands-on experience with fuzzy logic systems implementation.

16.14 <u>MP-DriveWay™ – Application Code</u> <u>Generator</u>

MP-DriveWay is an easy-to-use Windows-based Application Code Generator. With MP-DriveWay you can visually configure all the peripherals in a PICmicro device and, with a click of the mouse, generate all the initialization and many functional code modules in C language. The output is fully compatible with Microchip's MPLAB-C C compiler. The code produced is highly modular and allows easy integration of your own code. MP-DriveWay is intelligent enough to maintain your code through subsequent code generation.

16.15 <u>SEEVAL[®] Evaluation and</u> <u>Programming System</u>

The SEEVAL SEEPROM Designer's Kit supports all Microchip 2-wire and 3-wire Serial EEPROMs. The kit includes everything necessary to read, write, erase or program special features of any Microchip SEEPROM product including Smart Serials[™] and secure serials. The Total Endurance[™] Disk is included to aid in tradeoff analysis and reliability calculations. The total kit can significantly reduce time-to-market and result in an optimized system.

16.16 <u>KEELOQ[®] Evaluation and</u> <u>Programming Tools</u>

KEELOQ evaluation and programming tools support Microchips HCS Secure Data Products. The HCS evaluation kit includes an LCD display to show changing codes, a decoder to decode transmissions, and a programming interface to program test transmitters.

17.3 DC Characteristics:

PIC16C923/924-04 (Commercial, Industrial) PIC16C923/924-08 (Commercial, Industrial) PIC16LC923/924-04 (Commercial, Industrial)

		Standa	rd Opera	ting	Conditio	ns (un	less otherwise stated)	
	APACTERISTICS	Operating temperature $-40^{\circ}C \leq TA \leq +85^{\circ}C$ for industrial and						
		$0^{\circ}C \leq TA \leq +70^{\circ}C$ for commercial						
		Operati	ng voltage	e Vd	D range a	s desc	ribed in DC spec	
Param	Characteristic	Sym	Min	Тур	Max	Units	Conditions	
No.				1				
	Input Low Voltage							
	I/O ports	VIL						
D030	with TTL buffer		Vss	-	0.15Vdd	V	For entire VDD range	
			Vss	-	0.8V	V	$4.5V \le VDD \le 5.5V$	
D031	with Schmitt Trigger buffer		Vss	-	0.2Vdd	V		
D032	MCLR, OSC1 (in RC mode)		Vss	-	0.2Vdd	V		
D033	OSC1 (in XT, HS and LP)		Vss	-	0.3Vdd	V	Note1	
	Input High Voltage							
	I/O ports	Vih		-				
D040	with TTL buffer		2.0	-	Vdd	V	$4.5V \le VDD \le 5.5V$	
D040A			0.25Vdd	-	Vdd	V	For entire VDD range	
			+ 0.8V					
D041	with Schmitt Trigger buffer		0.8Vdd	-	Vdd	V		
D042	MCLR		0.8Vdd	-	Vdd	V		
D042A	OSC1 (XT, HS and LP)		0.7Vdd	-	Vdd	V	Note1	
D043	OSC1 (in RC mode)		0.9Vdd	-	Vdd	V		
D070	PORTB weak pull-up current	IPURB	50	250	400	μΑ	VDD = 5V, VPIN = VSS	
	Input Leakage Current (Notes 2, 3)							
D060	I/O ports	lı.	-	-	±1.0	uΑ	Vss ≤ VPIN ≤ Vpp. Pin at hi-Z	
D061	MCLR. RA4/T0CKI		-	-	±5	μA	$Vss \leq VPIN \leq VDD$	
D063	OSC1		-	-	±5	μA	Vss \leq VPIN \leq VDD, XT, HS and LP osc	
					_	1.	configuration	
	Output Low Voltage						-	
D080	I/O ports	Vol	-	-	0.6	V	IOL = 4.0 mA, VDD = 4.5V	
D083	OSC2/CLKOUT (RC osc mode)		-	-	0.6	V	IOL = 1.6 mA, VDD = 4.5V	
	Output High Voltage							
D090	I/O ports (Note 3)	Vон	Vdd - 0.7	-	-	V	IOH = -3.0 mA, VDD = 4.5V	
D092	OSC2/CLKOUT (RC osc mode)		Vdd - 0.7	-	-	V	IOH = -1.3 mA, VDD = 4.5V	
	Capacitive Loading Specs on							
	Output Pins							
D100*	OSC2 pin	Cosc2	-	-	15	pF	In XT, HS and LP modes when exter- nal clock is used to drive OSC1.	
D101*	All I/O pins and OSC2 (in RC)	Сю	-	-	50	pF		
D102*	SCL, SDA in I ² C mode	Св	-	-	400	pF		
D150*	Open -Drain High Voltage	Vdd	-	-	14	V	RA4 pin	

These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: In RC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC16C9XX be driven with external clock in RC mode.

2: The leakage current on the MCLR pin is strongly dependent on the 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 current sourced by the pin.

FIGURE 17-6: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS



Param No.	Sym	Characteristic			Min	Тур†	Max	Units	Conditions	
40*	Tt0H	T0CKI High Pulse V	Vidth	No Prescaler	0.5Tcy + 20	-	—	ns	Must also meet	
			V		10	_	_	ns	parameter 42	
41*	Tt0L	T0CKI Low Pulse W	/idth	No Prescaler	0.5TCY + 20	—	—	ns	Must also meet	
				With Prescaler	10	—	—	ns	parameter 42	
42*	Tt0P	T0CKI Period		No Prescaler	TCY + 40	—	—	ns		
				With Prescaler	Greater of:	-	—	ns	N = prescale value	
					20 or <u>Tcy + 40</u>				(2, 4,, 256)	
45*	T+411	T4 OKU Llink Time	Curacharan avec							
45"	ITTH	I TCKI High Time	Synchronous, P		0.51CY + 20			ns	Must also meet	
			Synchronous,	PIC16C923/924	15			ns		
			2,4,8	PIC16 LC 923/924	25	_	_	ns		
			Asynchronous	PIC16 C 923/924	30	—	—	ns		
				PIC16LC923/924	50	—	—	ns		
46*	Tt1L	T1CKI Low Time	Synchronous, F	rescaler = 1	0.5Tcy + 20	—	—	ns	Must also meet	
			Synchronous,	PIC16 C 923/924	15	—	—	ns	parameter 47	
			Prescaler = 2,4,8	PIC16 LC 923/924	25	-	_	ns		
			Asynchronous	PIC16 C 923/924	30	-	—	ns		
				PIC16LC923/924	50	—	—	ns		
47*	Tt1P	T1CKI input period	Synchronous	PIC16 C 923/924	Greater of:	-	—	ns	N = prescale value	
					30 OR <u>TCY + 40</u>				(1, 2, 4, 8)	
					N					
				PIC16LC923/924	Greater of:				N = prescale value	
					50 OR <u>TCY + 40</u>				(1, 2, 4, 8)	
			Asynchronous	PIC16C023/02/	60			ne		
			Asylicilionous	PIC16I C023/024	100			ne		
	Et1	Timer1 oscillator inr	L	1.1010 E0 323/924		+=-	200	kH7		
		(oscillator enabled b	ov setting bit T1C	SCEN)			200			
48	TCKEZtmr1	Delay from external	clock edge to tir	ner increment	2Tosc	-	7Tosc			
		,,				I				

These parameters are characterized but not tested.

+ Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

19.4 Package Marking Information



Legend:	MMM	Microchip part number information				
	XXX	Customer specific information*				
	AA	Year code (last 2 digits of calender year)				
	BB	Week code (week of January 1 is week '01')				
	С	Facility code of the plant at which wafer is manufactured. C = Chandler, Arizona, U.S.A. S = Tempe, Arizona, U.S.A.				
	D ₁	Mask revision number for microcontroller				
	E	Assembly code of the plant or country of origin in which part was assembled.				
Note:	In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.					

Standard OTP marking consists of Microchip part number, year code, week code, facility code, mask revision number, and assembly code. For OTP marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

List of Equations And Examples

Example 3-1:	Instruction Pipeline Flow15
Example 4-1:	Call of a Subroutine in Page 1 from Page 030
Example 4-2:	Indirect Addressing
Example 5-1:	Initializing PORTA
Example 5-2:	Initializing PORTB
Example 5-3:	Initializing PORTC
Example 5-4:	Initializing PORTD
Example 5-5:	Initializing PORTE
Example 5-6:	Initializing PORTF
Example 5-7:	Initializing PORTG 40
Example 5-8:	Read-Modify-Write Instructions on an
	I/O Port
Example 7-1:	Changing Prescaler (Timer0→WDT)49
Example 7-2:	Changing Prescaler (WDT→Timer0)49
Example 8-1:	Reading a 16-bit Free-Running Timer 53
Example 10-1:	Changing Between Capture Prescalers 58
Example 10-2:	PWM Period and Duty Cycle Calculation 60
Example 11-1:	Loading the SSPBUF (SSPSR) Register 65
Equation 12-1:	A/D Minimum Charging Time82
Example 12-1:	Calculating the Minimum Required
	Sample Time82
Example 12-2:	Doing an A/D Conversion
Example 12-3:	4-bit vs. 8-bit Conversion Times
Example 13-1:	Static MUX with 32 Segments 100
Example 13-2:	1/3 MUX with 13 Segments 100
Example 14-1:	Saving STATUS, W, and PCLATH
	Registers in RAM115

List of Figures

Figure 3-1:	PIC16C923 Block Diagram	. 10
Figure 3-2:	PIC16C924 Block Diagram	. 11
Figure 3-3:	Clock/Instruction Cycle	. 15
Figure 4-1:	Program Memory Map and Stack	. 17
Figure 4-2:	Register File Map	. 18
Figure 4-3:	Status Register (Address 03h, 83h, 103h,	
	183h)	. 23
Figure 4-4:	OPTION Register (Address 81h, 181h)	. 24
Figure 4-5:	INTCON Register (Address 0Bh, 8Bh,	
	10Bh, 18Bh)	. 25
Figure 4-6:	PIE1 Register (Address 8Ch)	. 26
Figure 4-7:	PIR1 Register (Address 0Ch)	. 27
Figure 4-8:	PCON Register (Address 8Eh)	. 28
Figure 4-9:	Loading of PC In Different Situations	. 29
Figure 4-10:	Direct/Indirect Addressing	. 30
Figure 5-1:	Block Diagram of pins RA3:RA0 and RA5.	. 31
Figure 5-2:	Block Diagram of RA4/T0CKI Pin	. 31
Figure 5-3:	Block Diagram of RB3:RB0 Pins	. 33
Figure 5-4:	Block Diagram of RB7:RB4 Pins	. 33
Figure 5-5:	PORTC Block Diagram (Peripheral Output	
	Override)	. 35
Figure 5-6:	PORTD<4:0> Block Diagram	. 36
Figure 5-7:	PORTD<7:5> Block Diagram	. 37
Figure 5-8:	PORTE Block Diagram	. 38
Figure 5-9:	PORTF Block Diagram	. 39
Figure 5-10:	PORTG Block Diagram	. 40
Figure 5-11:	Successive I/O Operation	. 41
Figure 7-1:	Timer0 Block Diagram	. 45
Figure 7-2:	Timer0 Timing: Internal Clock/No Prescale	45
Figure 7-3:	Timer0 Timing: Internal Clock/Prescale 1:2	46
Figure 7-4:	Timer0 Interrupt Timing	. 46
Figure 7-5:	Timer0 Timing with External Clock	. 47
Figure 7-6:	Block Diagram of the Timer0/WDT	
	Prescaler	. 48

Figure 8-1:	T1CON: Timer1 Control Register (Address	
	10h)5 ⁻	1
Figure 8-2:	Timer1 Block Diagram52	2
Figure 9-1:	Timer2 Block Diagram5	5
Figure 9-2:	T2CON: Timer2 Control Register (Address	5
Figure 10-1	CCP1CON Register (Address 17b) 5	э 7
Figure 10-1.	Capture Mode Operation Block Diagram	/ 0
Figure 10-2.	Compare Mode Operation Block Diagram	0
Figure 10-3.	Compare Mode Operation Block Diagram	0
Figure 10-4:		9
Figure 10-5:		9
Figure 11-1:	(Address 94b)	R
Figure 11-2	SSPCON: Sync Serial Port Control Register	0
riguio i i 2.	(Address 14b)	л
Figure 11-3	SSP Block Diagram (SPI Mode)	4 5
Figure 11-5.	SDI Mostor/Slove Connection	6
Figure 11-4.	SPI Maslel/Slave Connection	-
Figure 11-5:	SPI Mode Timing, Master Mode	1
Figure 11-6:	SPI Mode Timing	_
	(Slave Mode With $CKE = 0$)6	1
Figure 11-7:	SPI Mode Timing	_
	(Slave Mode With CKE = 1)68	8
Figure 11-8:	Start and Stop Conditions69	9
Figure 11-9:	7-bit Address Format70	0
Figure 11-10:	I ² C 10-bit Address Format70	0
Figure 11-11:	Slave-receiver Acknowledge70	0
Figure 11-12:	Data Transfer Wait State70	0
Figure 11-13:	Master-transmitter Sequence	1
Figure 11-14:	Master-receiver Sequence	1
Figure 11-15:	Combined Format	1
Figure 11-16:	Multi-master Arbitration	
g	(Two Masters) 7	2
Figure 11-17	Clock Synchronization 7	2
Figure 11-18:	SSP Block Diagram	-
riguie rielo.	$(l^2 C Mode)$ 7	2
Figure 11 10:	(1°C Woveforms for Recention	5
Figure 11-19.	(Z bit Address)	5
Figure 11 20	(7-bit Address)	5
Figure 11-20:	(Z bit Address)	~
E '	(7-bit Address)	b
Figure 11-21:	Operation of the I ² C Module in IDLE_MODE,	_
	RCV_MODE or XMI1_MODE	8
Figure 12-1:	ADCON0 Register (Address 1Fh)79	9
Figure 12-2:	ADCON1 Register (Address 9Fh)80	0
Figure 12-3:	A/D Block Diagram8	1
Figure 12-4:	Analog Input Model82	2
Figure 12-5:	A/D Transfer Function8	7
Figure 12-6:	Flowchart of A/D Operation88	8
Figure 13-1:	LCDCON Register (Address 10Fh)89	9
Figure 13-2:	LCD Module Block Diagram90	0
Figure 13-3:	LCDPS Register (Address 10Eh)90	0
Figure 13-4:	Waveforms in Static Drive9	1
Figure 13-5:	Waveforms in 1/2 MUX, 1/3 Bias Drive92	2
Figure 13-6:	Waveforms in 1/3 MUX, 1/3 Bias93	3
Figure 13-7:	Waveforms in 1/4 MUX, 1/3 Bias	4
Figure 13-8:	LCD Clock Generation	5
Figure 13-9:	Example Waveforms in 1/4 MUX Drive 9	7
Figure 13-10	Generic I CDD Register Lavout	8
Figure 13-11	Sleep Entry/exit When SI PEN = 1 or	0
. iguio 10-11.	$CS1 \cdot CS0 = 00$	o
Figure 12 12:	LCDSE Register (Address 10Db) 10	0 0
Figure 13-12:	Charge Pump and Posister Lodder 40	1
Figure 13-13:	Configuration Word	1 2
	Configuration voord	3
rigure 14-2:	Crystal/Ceramic Resonator Operation	
	(HS, XT or LP OSC Configuration) 104	4
Figure 14-3:	(HS, XT or LP OSC Configuration)	4

Note the following details of the code protection feature on PICmicro[®] MCUs.

- The PICmicro family meets the specifications contained in the Microchip Data Sheet.
- Microchip believes that its family of PICmicro microcontrollers is one of the most secure products of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the PICmicro microcontroller in a manner outside the operating specifications contained in the data sheet. The person doing so may be engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not
 mean that we are guaranteeing the product as "unbreakable".
- Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our product.

If you have any further questions about this matter, please contact the local sales office nearest to you.

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

Trademarks

The Microchip name and logo, the Microchip logo, FilterLab, KEELOQ, microID, MPLAB, PIC, PICmicro, PICMASTER, PICSTART, PRO MATE, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

dsPIC, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, microPort, Migratable Memory, MPASM, MPLIB, MPLINK, MPSIM, MXDEV, PICC, PICDEM, PICDEM.net, rfPIC, Select Mode and Total Endurance are trademarks of Microchip Technology Incorporated in the U.S.A.

Serialized Quick Turn Programming (SQTP) is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2002, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.





Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999. The Company's quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, KEELoq® code hopping devices, Serial EEPROMs and microperipheral products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001 certified.