



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	Obsolete
Core Processor	PIC
Core Size	8-Bit
Speed	33MHz
Connectivity	UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	33
Program Memory Size	4KB (2K x 16)
Program Memory Type	ОТР
EEPROM Size	-
RAM Size	232 x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 6V
Data Converters	-
Oscillator Type	External
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LCC (J-Lead)
Supplier Device Package	44-PLCC (16.59x16.59)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic17c42at-33-l

Email: info@E-XFL.COM

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

6.4.1 INDIRECT ADDRESSING REGISTERS

The PIC17C4X has four registers for indirect addressing. These registers are:

- INDF0 and FSR0
- INDF1 and FSR1

Registers INDF0 and INDF1 are not physically implemented. Reading or writing to these registers activates indirect addressing, with the value in the corresponding FSR register being the address of the data. The FSR is an 8-bit register and allows addressing anywhere in the 256-byte data memory address range. For banked memory, the bank of memory accessed is specified by the value in the BSR.

If file INDF0 (or INDF1) itself is read indirectly via an FSR, all '0's are read (Zero bit is set). Similarly, if INDF0 (or INDF1) is written to indirectly, the operation will be equivalent to a NOP, and the status bits are not affected.

6.4.2 INDIRECT ADDRESSING OPERATION

The indirect addressing capability has been enhanced over that of the PIC16CXX family. There are two control bits associated with each FSR register. These two bits configure the FSR register to:

- Auto-decrement the value (address) in the FSR after an indirect access
- Auto-increment the value (address) in the FSR after an indirect access
- No change to the value (address) in the FSR after an indirect access

These control bits are located in the ALUSTA register. The FSR1 register is controlled by the FS3:FS2 bits and FSR0 is controlled by the FS1:FS0 bits.

When using the auto-increment or auto-decrement features, the effect on the FSR is not reflected in the ALUSTA register. For example, if the indirect address causes the FSR to equal '0', the Z bit will not be set.

If the FSR register contains a value of 0h, an indirect read will read 0h (Zero bit is set) while an indirect write will be equivalent to a NOP (status bits are not affected).

Indirect addressing allows single cycle data transfers within the entire data space. This is possible with the use of the MOVPF and MOVFP instructions, where either 'p' or 'f' is specified as INDF0 (or INDF1).

If the source or destination of the indirect address is in banked memory, the location accessed will be determined by the value in the BSR. A simple program to clear RAM from 20h - FFh is shown in Example 6-1.

EXAMPLE 6-1: INDIRECT ADDRESSING

	MOVLW	0x20	;	
	MOVWF	FSR0	;	FSR0 = 20h
	BCF	ALUSTA, FS1	;	Increment FSR
	BSF	ALUSTA, FSO	;	after access
	BCF	ALUSTA, C	;	C = 0
	MOVLW	END_RAM + 1	;	
LP	CLRF	INDF0	;	Addr(FSR) = 0
	CPFSEQ	FSR0	;	FSR0 = END_RAM+1?
	GOTO	LP	;	NO, clear next
	:		;	YES, All RAM is
	:		;	cleared

6.5 <u>Table Pointer (TBLPTRL and</u> <u>TBLPTRH)</u>

File registers TBLPTRL and TBLPTRH form a 16-bit pointer to address the 64K program memory space. The table pointer is used by instructions TABLWT and TABLRD.

The TABLRD and the TABLWT instructions allow transfer of data between program and data space. The table pointer serves as the 16-bit address of the data word within the program memory. For a more complete description of these registers and the operation of Table Reads and Table Writes, see Section 7.0.

6.6 <u>Table Latch (TBLATH, TBLATL)</u>

The table latch (TBLAT) is a 16-bit register, with TBLATH and TBLATL referring to the high and low bytes of the register. It is not mapped into data or program memory. The table latch is used as a temporary holding latch during data transfer between program and data memory (see descriptions of instructions TABLRD, TABLWT, TLRD and TLWT). For a more complete description of these registers and the operation of Table Reads and Table Writes, see Section 7.0.

FIGURE 9-5: BLOCK DIAGRAM OF RB3 AND RB2 PORT PINS



FIGURE 12-2: TCON2 REGISTER (ADDRESS: 17h, BANK 3)

<u> </u>	<u> </u>	
CA2OV	F CA1OVF PWM2ON PWM1ON CA1/PR3 TMR3ON TMR2ON TMR1ON	R = Readable bit
bit7	bitO	-n = Value at POR reset
bit 7:	CA2OVF : Capture2 Overflow Status bit This bit indicates that the capture value had not been read from the captur before the next capture event occurred. The capture register retains the older capture before overflow). Subsequent capture events will not update the capt value until the capture register has been read (both bytes). 1 = Overflow occurred on Capture2 register 0 = No overflow occurred on Capture2 register	e register pair (CA2H:CA2L) st unread capture value (last oture register with the Timer3
bit 6:	CA10VF : Capture1 Overflow Status bit This bit indicates that the capture value had not been read from (PR3H/CA2H:PR3L/CA2L) before the next capture event occurred. The cap est unread capture value (last capture before overflow). Subsequent captur capture register with the TMR3 value until the capture register has been read 1 = Overflow occurred on Capture1 register 0 = No overflow occurred on Capture1 register	the capture register pair oture register retains the old- re events will not update the ad (both bytes).
bit 5:	PWM2ON : PWM2 On bit 1 = PWM2 is enabled (The RB3/PWM2 pin ignores the state of the DDRB< 0 = PWM2 is disabled (The RB3/PWM2 pin uses the state of the DDRB<3>	3> bit) bit for data direction)
bit 4:	PWM10N : PWM1 On bit 1 = PWM1 is enabled (The RB2/PWM1 pin ignores the state of the DDRB< 0 = PWM1 is disabled (The RB2/PWM1 pin uses the state of the DDRB<2>	2> bit) bit for data direction)
bit 3:	CA1/PR3 : CA1/PR3 Register Mode Select bit 1 = Enables Capture1 (PR3H/CA1H:PR3L/CA1L is the Capture1 register. To a period register) 0 = Enables the Period register (PR3H/CA1H:PR3L/CA1L is the Period regi	imer3 runs without ster for Timer3)
bit 2:	TMR3ON : Timer3 On bit 1 = Starts Timer3 0 = Stops Timer3	
bit 1:	TMR2ON : Timer2 On bit This bit controls the incrementing of the Timer2 register. When Timer2:Time is set), TMR2ON must be set. This allows the MSB of the timer to increment 1 = Starts Timer2 (Must be enabled if the T16 bit (TCON1<3>) is set) 0 = Stops Timer2	er1 form the 16-bit timer (T16 ht.
bit 0:	TMR1ON : Timer1 On bit <u>When T16 is set (in 16-bit Timer Mode)</u> 1 = Starts 16-bit Timer2:Timer1 0 = Stops 16-bit Timer2:Timer1	
	<u>When T16 is clear (in 8-bit Timer Mode)</u> 1 = Starts 8-bit Timer1 0 = Stops 8-bit Timer1	

12.1 <u>Timer1 and Timer2</u>

12.1.1 TIMER1, TIMER2 IN 8-BIT MODE

Both Timer1 and Timer2 will operate in 8-bit mode when the T16 bit is clear. These two timers can be independently configured to increment from the internal instruction cycle clock or from an external clock source on the RB4/TCLK12 pin. The timer clock source is configured by the TMRxCS bit (x = 1 for Timer1 or = 2 for Timer2). When TMRxCS is clear, the clock source is internal and increments once every instruction cycle (Fosc/4). When TMRxCS is set, the clock source is the RB4/TCLK12 pin, and the timer will increment on every falling edge of the RB4/TCLK12 pin.

The timer increments from 00h until it equals the Period register (PRx). It then resets to 00h at the next increment cycle. The timer interrupt flag is set when the timer is reset. TMR1 and TMR2 have individual interrupt flag bits. The TMR1 interrupt flag bit is latched into TMR1IF, and the TMR2 interrupt flag bit is latched into TMR2IF.

Each timer also has a corresponding interrupt enable bit (TMRxIE). The timer interrupt can be enabled by setting this bit and disabled by clearing this bit. For peripheral interrupts to be enabled, the Peripheral Interrupt Enable bit must be enabled (PEIE is set) and global interrupts must be enabled (GLINTD is cleared).

The timers can be turned on and off under software control. When the Timerx On control bit (TMRxON) is set, the timer increments from the clock source. When TMRxON is cleared, the timer is turned off and cannot cause the timer interrupt flag to be set.

12.1.1.1 EXTERNAL CLOCK INPUT FOR TIMER1 OR TIMER2

When TMRxCS is set, the clock source is the RB4/TCLK12 pin, and the timer will increment on every falling edge on the RB4/TCLK12 pin. The TCLK12 input is synchronized with internal phase clocks. This causes a delay from the time a falling edge appears on TCLK12 to the time TMR1 or TMR2 is actually incremented. For the external clock input timing requirements, see the Electrical Specification section.



FIGURE 12-3: TIMER1 AND TIMER2 IN TWO 8-BIT TIMER/COUNTER MODE

12.1.2 TIMER1 & TIMER2 IN 16-BIT MODE

To select 16-bit mode, the T16 bit must be set. In this mode TMR1 and TMR2 are concatenated to form a 16-bit timer (TMR2:TMR1). The 16-bit timer increments until it matches the 16-bit period register (PR2:PR1). On the following timer clock, the timer value is reset to 0h, and the TMR1IF bit is set.

When selecting the clock source for the16-bit timer, the TMR1CS bit controls the entire 16-bit timer and TMR2CS is a "don't care." When TMR1CS is clear, the timer increments once every instruction cycle (Fosc/4). When TMR1CS is set, the timer increments on every falling edge of the RB4/TCLK12 pin. For the 16-bit timer to increment, both TMR1ON and TMR2ON bits must be set (Table 12-1).

12.1.2.1 EXTERNAL CLOCK INPUT FOR TMR1:TMR2

When TMR1CS is set, the 16-bit TMR2:TMR1 increments on the falling edge of clock input TCLK12. The input on the RB4/TCLK12 pin is sampled and synchronized by the internal phase clocks twice every instruction cycle. This causes a delay from the time a falling edge appears on RB4/TCLK12 to the time TMR2:TMR1 is actually incremented. For the external clock input timing requirements, see the Electrical Specification section.

TABLE 12-1. TORINING ON TO-DIT TIME

TMR2ON	TMR10N	Result
1	1	16-bit timer (TMR2:TMR1) ON
0	1	Only TMR1 increments
x	0	16-bit timer OFF

FIGURE 12-4: TMR1 AND TMR2 IN 16-BIT TIMER/COUNTER MODE



TABLE 12-2: SUMMARY OF TIMER1 AND TIMER2 REGISTERS

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 (Note1)
16h, Bank 3	TCON1	CA2ED1	CA2ED0	CA1ED1	CA1ED0	T16	TMR3CS	TMR2CS	TMR1CS	0000 0000	0000 0000
17h, Bank 3	TCON2	CA2OVF	CA10VF	PWM2ON	PWM10N	CA1/PR3	TMR3ON	TMR2ON	TMR10N	0000 0000	0000 0000
10h, Bank 2	TMR1	Timer1 re	gister							xxxx xxxx	uuuu uuuu
11h, Bank 2	TMR2	Timer2 re	gister							xxxx xxxx	uuuu uuuu
16h, Bank 1	PIR	RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TXIF	RCIF	0000 0010	0000 0010
17h, Bank 1	PIE	RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TXIE	RCIE	0000 0000	0000 0000
07h, Unbanked	INTSTA	PEIF	TOCKIF	T0IF	INTF	PEIE	T0CKIE	TOIE	INTE	0000 0000	0000 0000
06h, Unbanked	CPUSTA	—	—	STKAV	GLINTD	TO	PD		_	11 11	11 qq
14h, Bank 2	PR1	Timer1 pe	riod registe	r						xxxx xxxx	uuuu uuuu
15h, Bank 2	PR2	Timer2 pe	riod registe	r						xxxx xxxx	uuuu uuuu
10h, Bank 3	PW1DCL	DC1	DC0	—	_	—	_	_	—	xx	uu
11h, Bank 3	PW2DCL	DC1	DC0	TM2PW2	_	_	_	_	_	xx0	uu0
12h, Bank 3	PW1DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2	xxxx xxxx	uuuu uuuu
13h, Bank 3	PW2DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2	xxxx xxxx	uuuu uuuu

Legend: x = unknown, u = unchanged, - = unimplemented read as a '0', q - value depends on condition,

shaded cells are not used by Timer1 or Timer2.

Note 1: Other (non power-up) resets include: external reset through MCLR and WDT Timer Reset.

NOTES:

13.1 USART Baud Rate Generator (BRG)

The BRG supports both the Asynchronous and Synchronous modes of the USART. It is a dedicated 8-bit baud rate generator. The SPBRG register controls the period of a free running 8-bit timer. Table 13-1 shows the formula for computation of the baud rate for different USART modes. These only apply when the USART is in synchronous master mode (internal clock) and asynchronous mode.

Given the desired baud rate and Fosc, the nearest integer value between 0 and 255 can be calculated using the formula below. The error in baud rate can then be determined.

TABLE 13-1: BAUD RATE FORMULA

SYNC	Mode	Baud Rate			
0	Asynchronous	Fosc/(64(X+1))			
1	Synchronous	Fosc/(4(X+1))			

X = value in SPBRG (0 to 255)

Example 13-1 shows the calculation of the baud rate error for the following conditions:

Fosc = 16 MHz Desired Baud Rate = 9600 SYNC = 0

EXAMPLE 13-1: CALCULATING BAUD RATE ERROR

Desired Baud rate=Fosc / (64 (X + 1))

 $9600 = \frac{16000000}{(64 (X + 1))}$

X = 25.042 = 25

Calculated Baud Rate=16000000 / (64 (25 + 1))

= 9615

- Error = <u>(Calculated Baud Rate Desired Baud Rate)</u> Desired Baud Rate
 - = (9615 9600) / 9600
 - = 0.16%

Writing a new value to the SPBRG, causes the BRG timer to be reset (or cleared), this ensures that the BRG does not wait for a timer overflow before outputting the new baud rate.

TABLE 13-2: REGISTERS ASSOCIATED WITH BAUD RATE GENERATOR

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 (Note1)
13h, Bank 0	RCSTA	SPEN	RX9	SREN	CREN	_	FERR	OERR	RX9D	0000 -00x	0000 -00u
15h, Bank 0	TXSTA	CSRC	TX9	TXEN	SYNC	—	_	TRMT	TX9D	00001x	0000lu
17h, Bank 0 SPBRG Baud rate generator register								XXXX XXXX	uuuu uuuu		

Legend: x = unknown, u = unchanged, - = unimplemented read as a '0', shaded cells are not used by the Baud Rate Generator. Note 1: Other (non power-up) resets include: external reset through \overline{MCLR} and Watchdog Timer Reset.

13.2.2 USART ASYNCHRONOUS RECEIVER

The receiver block diagram is shown in Figure 13-4. The data comes in the RA4/RX/DT pin and drives the data recovery block. The data recovery block is actually a high speed shifter operating at 16 times the baud rate, whereas the main receive serial shifter operates at the bit rate or at Fosc.

Once asynchronous mode is selected, reception is enabled by setting bit CREN (RCSTA<4>).

The heart of the receiver is the receive (serial) shift register (RSR). After sampling the stop bit, the received data in the RSR is transferred to the RCREG (if it is empty). If the transfer is complete, the interrupt bit RCIF (PIR<0>) is set. The actual interrupt can be enabled/disabled by setting/clearing the RCIE (PIE<0>) bit. RCIF is a read only bit which is cleared by the hardware. It is cleared when RCREG has been read and is empty. 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 begin shifting to the RSR. On detection of the stop bit of the third byte, if the RCREG is still full, then the overrun error bit, OERR (RCSTA<1>) will be set. The word in the RSR will be lost. RCREG can be read twice to retrieve the two bytes in the FIFO. The OERR bit has to be cleared in software which is done by resetting the receive logic (CREN is set). If the OERR bit is set, transfers from the RSR to RCREG are inhibited, so it is essential to clear the OERR bit if it is set. The framing error bit FERR (RCSTA<2>) is set if a stop bit is not detected.

FIGURE 13-7: RX PIN SAMPLING SCHEME

Note: The FERR and the 9th receive bit are buffered the same way as the receive data. Reading the RCREG register will allow the RX9D and FERR bits to be loaded with values for the next received Received data; therefore, it is essential for the user to read the RCSTA register before reading RCREG in order not to lose the old FERR and RX9D information.

13.2.3 SAMPLING

The data on the RA4/RX/DT pin is sampled three times by a majority detect circuit to determine if a high or a low level is present at the RA4/RX/DT pin. The sampling is done on the seventh, eighth and ninth falling edges of a x16 clock (Figure 11-3).

The x16 clock is a free running clock, and the three sample points occur at a frequency of every 16 falling edges.

RX		Start bit	Bit0
(RA4/RX/DT pin)	-	Baud CLK for all but start bit	
Jaud CLK	1		
x16 CLK		2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 1	
		Samples	

Steps to follow when setting up an Asynchronous Reception:

- 1. Initialize the SPBRG register for the appropriate baud rate.
- 2. Enable the asynchronous serial port by clearing the SYNC bit and setting the SPEN bit.
- 3. If interrupts are desired, then set the RCIE bit.
- 4. If 9-bit reception is desired, then set the RX9 bit.
- 5. Enable the reception by setting the CREN bit.
- 6. The RCIF bit will be set when reception completes and an interrupt will be generated if the RCIE bit was set.

- Read RCSTA to get the ninth bit (if enabled) and FERR bit to determine if any error occurred during reception.
- 8. Read RCREG for the 8-bit received data.
- 9. If an overrun error occurred, clear the error by clearing the OERR bit.
- Note: To terminate a reception, either clear the SREN and CREN bits, or the SPEN bit. This will reset the receive logic, so that it will be in the proper state when receive is re-enabled.



FIGURE 13-8: ASYNCHRONOUS RECEPTION

TABLE 13-6:	REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION

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 (Note1)
16h, Bank 1	PIR	RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TXIF	RCIF	0000 0010	0000 0010
13h, Bank 0	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00u
14h, Bank 0	RCREG	RX7	RX6	RX5	RX4	RX3	RX2	RX1	RX0	xxxx xxxx	uuuu uuuu
17h, Bank 1	PIE	RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TXIE	RCIE	0000 0000	0000 0000
15h, Bank 0	TXSTA	CSRC	TX9	TXEN	SYNC	—	_	TRMT	TX9D	00001x	00001u
17h, Bank 0	7h, Bank 0 SPBRG Baud rate generator register								xxxx xxxx	uuuu uuuu	

Legend: x = unknown, u = unchanged, - = unimplemented read as a '0', shaded cells are not used for asynchronous reception. Note 1: Other (non power-up) resets include: external reset through MCLR and Watchdog Timer Reset.

14.4.2 MINIMIZING CURRENT CONSUMPTION

To minimize current consumption, all I/O pins should be either at VDD, or VSS, with no external circuitry drawing current from the I/O pin. I/O pins that are hi-impedance inputs should be pulled high or low externally to avoid switching currents caused by floating inputs. The TOCKI input should be at VDD or VSS. The contributions from on-chip pull-ups on PORTB should also be considered, and disabled when possible.

14.5 <u>Code Protection</u>

The code in the program memory can be protected by selecting the microcontroller in code protected mode (PM2:PM0 = '000').

Note:	PM2 de	oes not	exist on th	e PIC17C42. To					
	select	code	protected	microcontroller					
	mode. $PM1:PM0 = '0.0'$.								

In this mode, instructions that are in the on-chip program memory space, can continue to read or write the program memory. An instruction that is executed outside of the internal program memory range will be inhibited from writing to or reading from program memory.

Note: Microchip does not recommend code protecting windowed devices.

If the code protection bit(s) have not been programmed, the on-chip program memory can be read out for verification purposes.

CALL		Subroutin	ne Call		CLI	RF	Clear f				
Syntax:		[label] C	CALL k		Syr	itax:	[<i>label</i>] CL	.RF f,s			
Operan	ds:	$0 \le k \le 4095$		Ope	erands:	$0 \le f \le 25$	$0 \le f \le 255$				
Operation:		PC+ 1→ T k<12:8> –	PC+ 1 \rightarrow TOS, k \rightarrow PC<12:0>, k<12:8> \rightarrow PCLATH<4:0>;				$00h \rightarrow f,$ $00h \rightarrow de$	$\begin{array}{l} 00h \rightarrow f, \ s \in \ [0,1] \\ 00h \rightarrow dest \end{array}$			
		$PC<15:13> \rightarrow PCLATH<7:5>$			Sta	tus Affected:	None				
Status Affected:		None			Enc	oding:	0010	100s	ffff	ffff	
Encodir	Encoding: 111k kkkk kkkk kkkk			Des	scription:	Clears the	contents	of the sp	ecified rea-		
Description:		Subroutine return addre the stack. TI PC bits<12: bits of the F	call within 8K ess (PC+1) is he 13-bit value :0>. Then the u PC are copied	page. First, pushed onto is loaded into upper-eight into PCLATH.		·	ister(s). s = 0: Data WREG are s = 1: Data cleared.	a memory e cleared. a memory	location	'f' and 'f' is	
		Call is a tw	Call is a two-cycle instruction.			rds:	1	1			
		See LCALL for calls outside 8K memory space.			Сус	eles:	1				
Words:		1			QC	Cycle Activity:					
Cycles:		2				Q1	Q2	Q	3	Q4	
Q Cvcle	e Activitv:					Decode	Read	Exec	ute	Write	
,	Q1	Q2	Q3	Q4			register i		i a	and other	
[Decode	Read literal 'k'<7:0>	Execute	NOP					:	specified register	
Fo	rced NOP	NOP	Execute	NOP	Exa	imple:	CLRF	FLAC	G_REG		
Example: HERE CALL THERE			Before Instru FLAG_R	uction EG = 0	x5A						
PC = Address(HERE)		RE)			After Instruc	tion					
Afte	er Instruct	tion Address (TH	·			FLAG_R	EG = 02	x00			

TOS = Address(HERE + 1)

CPFSEQ	PFSEQ Compare f with WREG, skip if f = WREG		CPF	SGT	Compare skip if f >	f with WRE WREG	G,			
Syntax:	Syntax: [label] CPFSEQ f			Syn	tax:	[label] ([label] CPFSGT f			
Operands:	Operands: $0 \le f \le 255$		Ope	rands:	$0 \le f \le 255$	5				
Operation:	(f) – (WRE skip if (f) = (unsigned ((f) – (WREG), skip if (f) = (WREG) (unsigned comparison)			ration:	(f) – (WREG), skip if (f) > (WREG) (unsigned comparison)				
Status Affecte	ed: None	None			us Affected:	None	None			
Encoding:	0011	0001 fff	f ffff	Enc	oding:	0011	0010 ff	ff ffff		
Description: Compares the location 'f' to th performing an If 'f' = WREG t tion is discard cuted instead instruction.		the contents of the contents an unsigned s G then the fetc arded and an N ad making this	data memory of WREG by ubtraction. thed instruc- IOP is exe- a two-cycle	Des	cription:	Compares location 'f' t by performi If the conte WREG then discarded a instead ma	the contents o to the contents ing an unsigne ints of 'f' > the n the fetched i and an NOP is king this a two	f data memory of the WREG ed subtraction. contents of nstruction is executed o-cycle instruc-		
Words:	1			Wor	de.	1				
Cycles:	1 (2)				us. Iac	1 (2)				
Q Cycle Activ	rity:				velo Activity:	1 (2)				
Q1	Q2	Q3	Q4	QU		02	03	04		
Decode	e Read register 'f'	Execute	NOP		Decode	Read	Execute	NOP		
If skip:				lf sk	in:	register i				
Q1	Q2	Q3	Q4		Q1	Q2	Q3	Q4		
Forced N	OP NOP	Execute	NOP		Forced NOP	NOP	Execute	NOP		
Example: HERE CPFSEQ REG NEQUAL : EQUAL :			<u>Exa</u>	mple:	HERE NGREATER GREATER	CPFSGT RI : :	EG			
Before Instruction PC Address = HERE WREG = ? REG = ?			Before Instru PC WREG	= Ac = ?	dress (HERE))				
After Inst If RE If RE	$\begin{array}{rcl} G & = & W \\ PC & = & Ac \\ G & \neq & W \\ PC & = & Ac \end{array}$	REG; Idress (EQUAL REG; Idress (NEQUA) L)		After Instruct If REG PC If REG PC	tion	REG; Idress (great REG; Idress (ngrea	TER) ATER)		

DECF	Decreme	nt f		DEC	CFSZ	Decreme	nt f, ski	p if 0	
Syntax:	[<i>label</i>] [DECF f,d		Syn	tax:	[<i>label</i>] [DECFSZ	Z f,d	
Operands:	0 ≤ f ≤ 255 d ∈ [0,1]	5		Оре	erands:	$\begin{array}{l} 0 \leq f \leq 255 \\ d \in \ [0,1] \end{array}$	5		
Operation:	(f) – 1 \rightarrow (dest)			Ope	eration:	(f) – 1 \rightarrow (dest);		
Status Affected:	OV, C, DC	OV, C, DC, Z				skip if resu	ult = 0		
Encoding:	0000 011d ffff ffff		Stat	us Affected:	None				
Description:	Decrement	register 'f'. If 'o	d' is 0 the	Enc	oding:	0001	011d	ffff	ffff
·	result is stored in WREG. If 'd' is 1 the result is stored back in register 'f'.		Des	cription:	The content mented. If 'd	The contents of register 'f' are decre- mented. If 'd' is 0 the result is placed in			
Words:	1					WREG. If 'd	l' is 1 the stor 'f'	e result is	s placed
Cycles:	1					If the result	is 0. the	next ins	truction.
Q Cycle Activity:						which is alr	eady feto	ched, is o	discarded,
Q1	Q2	Q3	Q4			and an NOI	is exection	cuted ins	tead mak-
Decode	Read register 'f'	Execute	Write to destination	Wor	ds:	1			
Example:	DECF	CNT, 1		Сус	les:	1(2)			
Before Instru	iction			QC	ycle Activity:				
CNT	= 0x01				Q1	Q2	Q	3	Q4
Z	= 0				Decode	Read register 'f'	Exec	ute c	Write to lestination
CNT Z	= 0x00 = 1			<u>Exa</u>	mple:	HERE	DECFS GOTO	SZ CN	ЛТ, 1)ОР
						CONTINUE			
					Before Instru	uction			

PC	=	Address (HERE)
After Instruct	ion	
CNT	=	CNT - 1
If CNT	=	0;
PC	=	Address (CONTINUE)
If CNT	≠	0;
PC	=	Address (HERE+1)

TABLRD	Table Read				
<u>Example1</u> :	TABLRD	1, 1,	REG ;		
Before Instruct	tion				
REG		=	0x53		
TBLATH		=	0xAA		
TBLATL		=	0x55		
TBLPTR		=	0xA356		
MEMORY(TBLPTR)	=	0x1234		
After Instruction	n (table v	vrite cor	mpletion)		
REG		=	0xAA		
TBLATH		=	0x12		
TBLATL		=	0x34		
TBLPTR		=	0xA357		
MEMORY(TBLPTR)	=	0x5678		
Example2:	TABLRD	0, 0,	REG ;		
Before Instruct	tion				
REG		=	0x53		
TBLATH		=	0xAA		
TBLATL		=	0x55		
TBLPTR		=	0xA356		
MEMORY(TBLPTR)	=	0x1234		
After Instructio	n (table v	vrite cor	mpletion)		
REG		=	0x55		
TBLATH		=	0x12		
TBLATL		=	0x34		
TBLPTR		=	0xA356		
MEMORY(TBLPTR)	=	0x1234		

$ [label] T 0 \le f \le 255 i \in [0,1] t \in [0,1] If t = 0, f \rightarrow TBIf t = 1,TBLATIf i = 1,TBLPTNone10101. Load vlatch (If t = 01. Load vlatch (If t = 12. The ccto the pointerIf TBLprogratethe insIf TBLPRO$	TABLWT t,i,f 5 11ti 5 10ad into low byte; 10ad into high byte 5 10at by TBLPTR 10at by TB
$0 \le f \le 255$ $i \in [0,1]$ $I \in [0,1]$ If t = 0, $f \rightarrow TB$ If t = 1, $f \rightarrow TB$ TBLAT If i = 1, TBLPT None 1010 1. Load v latch (If t = 0 If t = 1 2. The cc to the pointed If TBL progra the ins If TBL EPRO	SLATL; SLATH; \rightarrow Prog Mem (TBLPTR) $TR + 1 \rightarrow TBLPTR$ 11ti ffff ffff value in 'f' into 16-bit table TBLAT) 2: load into low byte; 2: load into low byte; 2: load into high byte pontents of TBLAT is written program memory location d to by TBLPTR LPTR points to external am memory location, then struction takes two-cycle PTR points to an internal
If $t = 0$, $f \rightarrow TB$ If $t = 1$, TBLAT If $i = 1$, TBLPT None 1010 1. Load v latch (If $t = 1$ 2. The cc to the pointer If TBL progra the ins If TBL PRO	SLATL; SLATH; $T \rightarrow \text{Prog Mem (TBLPTR)}$ $TR + 1 \rightarrow \text{TBLPTR}$ 11ti ffff ffff value in 'f' into 16-bit table TBLAT) TBLAT TBLAT is written program memory location d to by TBLPTR LPTR points to external am memory location, then struction takes two-cycle PTR points to an internal
None 1010 1. Load v latch (If t = 0 If t = 1 2. The cc to the pointer If TBL progra the ins If TBL EPRO	11tiffffffffvalue in 'f' into 16-bit tableTBLAT)b: load into low byte;: load into high bytepontents of TBLAT is writtenprogram memory locationd to by TBLPTRLPTR points to externalam memory location, thenstruction takes two-cyclePTR points to an internal
1010 1. Load v latch (If t = 0 If t = 1 2. The cc to the pointer If TBL progra the ins If TBL EPRO	11tiffffffffvalue in 'f' into 16-bit table TBLAT)b: load into low byte; : load into high byteb: load into high byteontents of TBLAT is written program memory location d to by TBLPTR LPTR points to external am memory location, then struction takes two-cycle PTR points to an internal
 Load v latch (If t = 0 If t = 1 The cc to the pointer If TBL progra the ins If TBL EPRO 	value in 'f' into 16-bit table TBLAT) b: load into low byte; : load into high byte protents of TBLAT is written program memory location d to by TBLPTR _PTR points to external am memory location, then struction takes two-cycle PTR points to an internal
instruct an inter R/VPP pin m r successfu PP = VDD Imming sec xecuted, but the internal	M location, then the ction is terminated when errupt is received. nust be at the programming ul programming of internal quence of internal memory ut will not be successful I memory location may be
3. The T cally ir	BLPTR can be automati-
If $i = 0$; TBLPTR is not
lf i = 1	; TBLPTR is incremented
1	
2 (many if EPROM p	write is to on-chip program memory)
Q2	Q3 Q4
Read egister 'f'	Execute Write register TBLATH or
	amming sec xecuted, b the interna 3. The T cally in If i = 0 If i = 1 2 (many if EPROM p Q2 Read register 'f'

Applicable Devices 42 R42 42A 43 R43 44



FIGURE 17-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING

TABLE 17-4:RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP
TIMER REQUIREMENTS

Parameter	Sym	Characteristic	Min	Тур†	Max	Units	Conditions
No.							
30	TmcL	MCLR Pulse Width (low)	100 *	—	_	ns	
31	Twdt	Watchdog Timer Time-out Period (Prescale = 1)	5*	12	25 *	ms	
32	Tost	Oscillation Start-up Timer Period		1024 Tosc §		ms	Tosc = OSC1 period
33	Tpwrt	Power-up Timer Period	40 *	96	200 *	ms	
35	TmcL2adI	MCLR to System Interface bus (AD15:AD0) invalid	_	_	100 *	ns	

* 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.

t These parameters are for design guidance only and are not tested, nor characterized.

§ This specification ensured by design.

Applicable Devices 42 R42 42A 43 R43 44

FIGURE 18-2: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD



FIGURE 18-3: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD



Applicable Devices 42 R42 42A 43 R43 44



TABLE 19-11: MEMORY INTERFACE WRITE REQUIREMENTS (NOT SUPPORTED IN PIC17LC4X DEVICES)

Parameter No.	Sym	Characteristic	Min	Тур†	Max	Units	Conditions
150	TadV2alL	AD<15:0> (address) valid to ALE↓ (address setup time)	0.25Tcy - 10	_	_	ns	
151	TalL2adl	ALE↓ to address out invalid (address hold time)	0	_	_	ns	
152	TadV2wrL	Data out valid to $\overline{WR} \downarrow$ (data setup time)	0.25Tcy - 40	—	_	ns	
153	TwrH2adl	WR [↑] to data out invalid (data hold time)	_	0.25Tcy §	_	ns	
154	TwrL	WR pulse width	—	0.25TCY §		ns	

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.

§ This specification ensured by design.

21.2 <u>40-Lead Plastic Dual In-line (600 mil)</u>



Package Group: Plastic Dual In-Line (PLA)								
		Millimeters		Inches				
Symbol	Min	Max	Notes	Min	Max	Notes		
α	0°	10°		0°	10°			
A	_	5.080		_	0.200			
A1	0.381	_		0.015	_			
A2	3.175	4.064		0.125	0.160			
В	0.355	0.559		0.014	0.022			
B1	1.270	1.778	Typical	0.050	0.070	Typical		
С	0.203	0.381	Typical	0.008	0.015	Typical		
D	51.181	52.197		2.015	2.055			
D1	48.260	48.260	Reference	1.900	1.900	Reference		
E	15.240	15.875		0.600	0.625			
E1	13.462	13.970		0.530	0.550			
e1	2.489	2.591	Typical	0.098	0.102	Typical		
eA	15.240	15.240	Reference	0.600	0.600	Reference		
eB	15.240	17.272		0.600	0.680			
L	2.921	3.683		0.115	0.145			
N	40	40		40	40			
S	1.270	-		0.050	-			
S1	0.508	_		0.020	_			

MP-C C Compiler	145
MPSIM Software Simulator	143, 145
MULLW	129
Multiply Examples	
16 x 16 Routine	50
16 x 16 Signed Routine	51
8 x 8 Routine	49
8 x 8 Signed Routine	49
MULWF	129

Ν

NEGW 1	30
NOP1	30

0

OERR	84
Opcode Field Descriptions	107
OSC Selection	
Oscillator	
Configuration	100
Crystal	
External Clock	
External Crystal Circuit	102
External Parallel Resonant Crystal Circuit	102
External Series Resonant Crystal Circuit	102
RC	
RC Frequencies	165, 195
Oscillator Start-up Time (Figure)	
Oscillator Start-up Timer (OST)	15, 99
OST	15, 99
OV	
Overflow (OV)	9

Ρ

Package Marking Information	
Packaging Information	
Parameter Measurement Information	
PC (Program Counter)	
PCH	
PCL	34, 41, 108
PCLATH	
PD	
PEIE	
PEIF	
Peripheral Bank	
Peripheral Interrupt Enable	23
Peripheral Interrupt Request (PIR)	
PICDEM-1 Low-Cost PIC16/17 Demo Board	143, 144
PICDEM-2 Low-Cost PIC16CXX Demo Board	143, 144
PICDEM-3 Low-Cost PIC16C9XXX Demo Boar	d144
PICMASTER [®] RT In-Circuit Emulator	
PICSTART [®] Low-Cost Development System	
PIE	34, 92, 96, 98
Pin Compatible Devices	
PIR	34, 92, 96, 98
PM0	
PM1	
POP	
POR	
PORTA	19, 34, 53
PORTB	19, 34, 55
PORTC	19, 34, 58

PORTD	19,	34,	60
PORTE	19,	34,	62
Power-down Mode		1	05
Power-on Reset (POR)		15.	99
Power-up Timer (PWRT)		15	99
PR1		20	35
PR2		20,	35
		20,	20
			20
	•••••		20
			30
PR3L/CATL	•••••		35
	•••••		69
PRO MATE® Universal Programmer	•••••	1	43
PRODH	•••••		20
PRODL			20
Program Counter (PC)			41
Program Memory			
External Access Waveforms			31
External Connection Diagram			31
Мар			29
Modes			
Extended Microcontroller			29
Microcontroller			29
Microprocessor			29
Protected Microcontroller			29
			20
Organization			20
Transfore from Date Momeny			42
Prate start Misses as testler	•••••		43
Protected Microcontroller	•••••		29
PS0	•••••	38,	67
PS1	•••••	38,	67
PS2		38,	67
PS3		38,	67
PUSH		27,	39
PW1DCH		20,	35
PW1DCL		20,	35
PW2DCH		20,	35
PW2DCL		20,	35
PWM		71,	75
Duty Cycle			76
External Clock Source			76
Frequency vs. Resolution			76
			76
Max Resolution/Frequency for External			
Clock Input			77
			75
Poriode			76
			70
	•••••		12
	•••••	12,	15
	•••••		12
PWM2ON		72,	75
PWR I		15,	99

R

RA1/T0CKI pin	
RBIE	
RBIF	
RBPU	
RC Oscillator	
RC Oscillator Frequencies	
RCIE	
RCIF	
RCREG	19, 34, 91, 92, 96, 97
RCSTA	
Reading 16-bit Value	
5	

Figure 19-2:	External Clock Timing 184	
Figure 19-3:	CLKOUT and I/O Timing 185	
Figure 19-4:	Reset, Watchdog Timer.	
J	Oscillator Start-Up Timer, and	
	Power-I In Timer Timing 186	
Figure 19-5	Timer() Clock Timings 187	
Figure 19-5.	Timer1 Timer2 and Timer2 Clock	
Figure 19-0.	Timinan 197	
E: 40 T	107	
Figure 19-7:	Capture Timings	
Figure 19-8:	PWM Timings 188	
Figure 19-9:	USART Module: Synchronous	
	Transmission (Master/Slave) Timing 189	
Figure 19-10:	USART Module: Synchronous	
	Receive (Master/Slave) Timing 189	
Figure 19-11:	Memory Interface Write Timing	
0	(Not Supported in PIC17LC4X Devices) 190	
Figure 19-12:	Memory Interface Read Timing	
rigulo lo l2.	(Not Supported in PIC17I C4X Devices) 191	
Figure 20-1.	Typical RC Oscillator Frequency vs	
rigule 20-1.	Tomporoturo 102	
E		
Figure 20-2:	Typical RC Oscillator Frequency	
	vs. VDD	
Figure 20-3:	Typical RC Oscillator Frequency	
	vs. VDD194	
Figure 20-4:	Typical RC Oscillator Frequency	
	vs. VDD 195	
Figure 20-5:	Transconductance (gm) of LF Oscillator	
-	vs. VDD	
Figure 20-6:	Transconductance (gm) of XT Oscillator	
J	vs. VDD 196	
Figure 20-7	Typical Ipp vs. Frequency (External	
ga.o _o	Clock 25°C) 197	
Figure 20-8.	Maximum IDD vs. Frequency (External	
1 igule 20-0.	Clock 125° C to 40° C)	
	Clock 125 C to -40 C) 197	
Figure 20-9:	Disabled 25°C	
F : 00.40	Disabled 25°C 198	
Figure 20-10:	Maximum IPD vs. VDD watchdog	
	Disabled 198	
Figure 20-11:	Typical IPD vs. VDD Watchdog	
	Enabled 25°C 199	
Figure 20-12:	Maximum IPD vs. VDD Watchdog	
	Enabled 199	
Figure 20-13:	WDT Timer Time-Out Period vs. VDD 200	
Figure 20-14:	ЮН vs. VOH. VDD = 3V	
Figure 20-15:	IOH VS. VOH. VDD = 5V 201	
Figure 20-16	$ O \vee S \vee O = 3 \vee 201$	
Figure 20-17:	$101 \text{ vs. } \sqrt{01} \sqrt{00} = 5 \sqrt{01} 0$	
Figure 20-17.	VTH (Input Throshold Voltago) of	
Figure 20-16.		
E	1/O FILIS (TTL) VS. VDD	
Figure 20-19:	VTH, VIL of I/O PINS (Schmitt Trigger)	
F : 00.05	vs. vDD	
Figure 20-20:	VTH (Input Threshold Voltage) of OSC1	
	Input (In XT and LF Modes) vs. VDD 203	
LIST OF TAE		
Table 1-1:	PIC17CXX Family of Devices 6	
Table 3-1:	Pinout Descriptions12	
Table 4-1:	Time-Out in Various Situations	
Table 4-2:	STATUS Bits and Their Significance	

Mode Memory Access 30

Initialization Conditions For Special

Table 6-2:	EPROM Memory Access Time
	Ordering Suffix
Table 6-3:	Special Function Registers34
Table 7-1:	Interrupt - Table Write Interaction45
Table 8-1:	Performance Comparison49
Table 9-1:	PORTA Functions
Table 9-2:	Registers/Bits Associated with PORTA54
Table 9-3:	PORTB Functions
Table 9-4:	Registers/Bits Associated with PORTB57
Table 9-5.	PORTC FUNCTIONS
Table 9-0.	POPTD Functions
Table 9-7.	Registers/Bits Associated with PORTD 61
Table 9-0.	PORTE Functions 63
Table 9-10 [.]	Registers/Bits Associated with PORTE 63
Table 11-1	Registers/Bits Associated with Timer0 70
Table 12-1	Turning On 16-bit Timer 74
Table 12-2:	Summary of Timer1 and Timer2
	Registers
Table 12-3:	PWM Frequency vs. Resolution at
	25 MHz
Table 12-4:	Registers/Bits Associated with PWM77
Table 12-5:	Registers Associated with Capture79
Table 12-6:	Summary of TMR1, TMR2, and TMR3
	Registers81
Table 13-1:	Baud Rate Formula86
Table 13-2:	Registers Associated with Baud Rate
	Generator86
Table 13-3:	Baud Rates for Synchronous Mode87
Table 13-4:	Baud Rates for Asynchronous Mode88
Table 13-5:	Registers Associated with Asynchronous
	Transmission90
Table 13-6:	Registers Associated with Asynchronous
	Reception92
Table 13-7:	Registers Associated with Synchronous
	Master Transmission94
Table 13-8:	Registers Associated with Synchronous
T 40.0	Master Reception
Table 13-9:	Registers Associated with Synchronous
Table 40.40	Slave Transmission
Table 13-10:	Registers Associated with Synchronous
Table 14 1:	Slave Reception
Table 14-1.	Consister Selection for Coromia
	Posopators 101
Table 14-3	Capacitor Selection for Crystal
Table 14-5.	OscillatoR 101
Table 14-4	Registers/Bits Associated with the
	Watchdog Timer 104
Table 15-1:	Opcode Field Descriptions
Table 15-2:	PIC17CXX Instruction Set
Table 16-1:	development tools from microchip146
Table 17-1:	Cross Reference of Device Specs for
	Oscillator Configurations and Frequencies
	of Operation (Commercial Devices)148
Table 17-2:	External Clock Timing Requirements155
Table 17-3:	CLKOUT and I/O Timing Requirements156
Table 17-4:	Reset, Watchdog Timer,
	Oscillator Start-Up Timer and
	Power-Up Timer Requirements157
Table 17-5:	Timer0 Clock Requirements158
Table 17-6:	Timer1, Timer2, and Timer3 Clock
	Requirements158
Table 17-7:	Capture Requirements159
Table 17-8:	PWM Requirements159

Table 4-3:

Table 4-4:

Table 5-1:

Table 6-1: