



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	32MHz
Connectivity	I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	17
Program Memory Size	7KB (4K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SSOP (0.209", 5.30mm Width)
Supplier Device Package	20-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf1828-i-ss

Email: info@E-XFL.COM

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

5.2.2.5 Internal Oscillator Frequency Selection

The system clock speed can be selected via software using the Internal Oscillator Frequency Select bits IRCF<3:0> of the OSCCON register.

The outputs of the 16 MHz HFINTOSC postscaler and the LFINTOSC connect to a multiplexer (see Figure 5-1). The Internal Oscillator Frequency Select bits IRCF<3:0> of the OSCCON register select the frequency output of the internal oscillators. One of the following frequencies can be selected via software:

- 32 MHz (requires 4xPLL)
- 16 MHz
- 8 MHz
- 4 MHz
- 2 MHz
- 1 MHz
- 500 kHz (Default after Reset)
- 250 kHz
- 125 kHz
- 62.5 kHz
- 31.25 kHz
- 31 kHz (LFINTOSC)
 - Note: Following any Reset, the IRCF<3:0> bits of the OSCCON register are set to '0111' and the frequency selection is set to 500 kHz. The user can modify the IRCF bits to select a different frequency.

The IRCF<3:0> bits of the OSCCON register allow duplicate selections for some frequencies. These duplicate choices can offer system design trade-offs. Lower power consumption can be obtained when changing oscillator sources for a given frequency. Faster transition times can be obtained between frequency changes that use the same oscillator source.

5.2.2.6 32 MHz Internal Oscillator Frequency Selection

The Internal Oscillator Block can be used with the 4xPLL associated with the External Oscillator Block to produce a 32 MHz internal system clock source. The following settings are required to use the 32 MHz internal clock source:

- The FOSC bits in Configuration Word 1 must be set to use the INTOSC source as the device system clock (FOSC<2:0> = 100).
- The SCS bits in the OSCCON register must be cleared to use the clock determined by FOSC<2:0> in Configuration Word 1 (SCS<1:0> = 00).
- The IRCF bits in the OSCCON register must be set to the 8 MHz HFINTOSC set to use (IRCF<3:0> = 1110).
- The SPLLEN bit in the OSCCON register must be set to enable the 4xPLL, or the PLLEN bit of the Configuration Word 2 must be programmed to a '1'.
- Note: When using the PLLEN bit of the Configuration Word 2, the 4xPLL cannot be disabled by software and the 8 MHz HFINTOSC option will no longer be available.

The 4xPLL is not available for use with the internal oscillator when the SCS bits of the OSCCON register are set to '1x'. The SCS bits must be set to '00' to use the 4xPLL with the internal oscillator.

11.2 Using the Data EEPROM

The data EEPROM is a high-endurance, byte addressable array that has been optimized for the storage of frequently changing information (e.g., program variables or other data that are updated often). When variables in one section change frequently, while variables in another section do not change, it is possible to exceed the total number of write cycles to the EEPROM without exceeding the total number of write cycles to a single byte. Refer to **Section 30.0 "Electrical Specifications"**. If this is the case, then a refresh of the array must be performed. For this reason, variables that change infrequently (such as constants, IDs, calibration, etc.) should be stored in Flash program memory.

11.2.1 READING THE DATA EEPROM MEMORY

To read a data memory location, the user must write the address to the EEADRL register, clear the EEPGD and CFGS control bits of the EECON1 register, and then set control bit RD. The data is available at the very next cycle, in the EEDATL register; therefore, it can be read in the next instruction. EEDATL will hold this value until another read or until it is written to by the user (during a write operation).

EXAMPLE 11-1: DATA EEPROM READ

BANKSEL	EEADRL	i
MOVLW	DATA_EE_ADD	R ;
MOVWF	EEADRL	;Data Memory
		;Address to read
BCF	EECON1, CFG	S ;Deselect Config space
BCF	EECON1, EEP	GD;Point to DATA memory
BSF	EECON1, RD	;EE Read
MOVF	EEDATL, W	;W = EEDATL

Note: Data EEPROM can be read regardless of the setting of the CPD bit.

11.2.2 WRITING TO THE DATA EEPROM MEMORY

To write an EEPROM data location, the user must first write the address to the EEADRL register and the data to the EEDATL register. Then the user must follow a specific sequence to initiate the write for each byte.

The write will not initiate if the above sequence is not followed exactly (write 55h to EECON2, write AAh to EECON2, then set WR bit) for each byte. Interrupts should be disabled during this code segment.

Additionally, the WREN bit in EECON1 must be set to enable write. This mechanism prevents accidental writes to data EEPROM due to errant (unexpected) code execution (i.e., lost programs). The user should keep the WREN bit clear at all times, except when updating EEPROM. The WREN bit is not cleared by hardware.

After a write sequence has been initiated, clearing the WREN bit will not affect this write cycle. The WR bit will be inhibited from being set unless the WREN bit is set.

At the completion of the write cycle, the WR bit is cleared in hardware and the EE Write Complete Interrupt Flag bit (EEIF) is set. The user can either enable this interrupt or poll this bit. EEIF must be cleared by software.

11.2.3 PROTECTION AGAINST SPURIOUS WRITE

There are conditions when the user may not want to write to the data EEPROM memory. To protect against spurious EEPROM writes, various mechanisms have been built-in. On power-up, WREN is cleared. Also, the Power-up Timer (64 ms duration) prevents EEPROM write.

The write initiate sequence and the WREN bit together help prevent an accidental write during:

- Brown-out
- Power Glitch
- Software Malfunction

11.2.4 DATA EEPROM OPERATION DURING CODE-PROTECT

Data memory can be code-protected by programming the CPD bit in the Configuration Word 1 (Register 5-1) to '0'.

When the data memory is code-protected, only the CPU is able to read and write data to the data EEPROM. It is recommended to code-protect the program memory when code-protecting data memory. This prevents anyone from replacing your program with a program that will access the contents of the data EEPROM.

EXAMPLE 11-2: DATA EEPROM WRITE

		BANKSEL	EEADRL	i
		MOVLW	DATA_EE_ADDR	;
		MOVWF	EEADRL	;Data Memory Address to write
		MOVLW	DATA_EE_DATA	i
		MOVWF	EEDATL	;Data Memory Value to write
		BCF	EECON1, CFGS	;Deselect Configuration space
		BCF	EECON1, EEPGD	;Point to DATA memory
		BSF	EECON1, WREN	;Enable writes
		BCF	INTCON, GIE	;Disable INTs.
ſ		MOVLW	55h	i
	ce ed	MOVWF	EECON2	;Write 55h
	huird	MOVLW	0AAh	i
	Sec	MOVWF	EECON2	;Write AAh
	ш 0	BSF	EECON1, WR	;Set WR bit to begin write
		BSF	INTCON, GIE	;Enable Interrupts
		BCF	EECON1, WREN	;Disable writes
		BTFSC	EECON1, WR	;Wait for write to complete
		GOTO	\$-2	;Done



	Q1 Q2 Q3 Q4	Q1 Q2 Q3 Q4	Q1 Q2 Q3 Q4	Q1 Q2 Q3 Q4	Q1 Q2 Q3 Q4	Q1 Q2 Q3 Q4
Flash ADDR	 ()	PC + 1	EEADRH,EEADRL	PC + 3	PC + 4	PC + 5
Flash Data		I STR (PC) X INSTI	R (PC + 1) EEDA		R (PC + 3) INST	R (PC + 4)
	INSTR(PC - 1) executed here	BSF EECON1,RD executed here	INSTR(PC + 1) executed here	Forced NOP executed here	INSTR(PC + 3) executed here	INSTR(PC + 4) executed here
RD bit	 	 			 	
EEDATH EEDATL Register	 	 	 	X		
EERHLT	ı 	 	، ا			

16.3 A/D Acquisition Requirements

For the ADC to meet its specified accuracy, the charge holding capacitor (CHOLD) must be allowed to fully charge to the input channel voltage level. The Analog Input model is shown in Figure 16-4. The source impedance (Rs) and the internal sampling switch (Rss) impedance directly affect the time required to charge the capacitor CHOLD. The sampling switch (Rss) impedance varies over the device voltage (VDD), refer to Figure 16-4. The maximum recommended impedance for analog sources is 10 k Ω . As the

source impedance is decreased, the acquisition time may be decreased. After the analog input channel is selected (or changed), an A/D acquisition must be done before the conversion can be started. To calculate the minimum acquisition time, Equation 16-1 may be used. This equation assumes that 1/2 LSb error is used (1,024 steps for the ADC). The 1/2 LSb error is the maximum error allowed for the ADC to meet its specified resolution.

EQUATION 16-1: ACQUISITION TIME EXAMPLE

Assumptions: Temperature =
$$50^{\circ}C$$
 and external impedance of $10k\Omega$ 5.0V VDD
 $TACQ = Amplifier Settling Time + Hold Capacitor Charging Time + Temperature Coefficient$
 $= TAMP + TC + TCOFF$
 $= 2\mu s + TC + [(Temperature - 25^{\circ}C)(0.05\mu s/^{\circ}C)]$

The value for TC can be approximated with the following equations:

$$V_{APPLIED}\left(1 - \frac{1}{(2^{n+1}) - 1}\right) = V_{CHOLD} \qquad ;[1] V_{CHOLD} charged to within 1/2 lsb$$

$$V_{APPLIED}\left(1 - e^{\frac{-Tc}{RC}}\right) = V_{CHOLD} \qquad ;[2] V_{CHOLD} charge response to V_{APPLIED} \\V_{APPLIED}\left(1 - e^{\frac{-Tc}{RC}}\right) = V_{APPLIED}\left(1 - \frac{1}{(2^{n+1}) - 1}\right) \qquad ;combining [1] and [2]$$

Note: Where n = number of bits of the ADC.

Solving for TC:

$$Tc = -C_{HOLD}(R_{IC} + R_{SS} + R_{S}) \ln(1/2047)$$

= -12.5pF(1k\Omega + 7k\Omega + 10k\Omega) \ln(0.0004885)
= 1.72\mus

Therefore:

$$TACQ = 2\mu s + 1.72\mu s + [(50^{\circ}C - 25^{\circ}C)(0.05\mu s/^{\circ}C)]$$

= 4.97\mu s

Note 1: The reference voltage (VREF) has no effect on the equation, since it cancels itself out.

- 2: The charge holding capacitor (CHOLD) is not discharged after each conversion.
- **3:** The maximum recommended impedance for analog sources is $10 \text{ k}\Omega$. This is required to meet the pin leakage specification.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CPSCON0	CPSON	CPSRM	_		CPSRNG<1:0>		CPSOUT	T0XCS	319
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		CDAFVR<1:0> ADFVR<1:0>		141
INLVLA	_	_	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0	123
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	89
OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA		PS<2:0>		176
TMR0	Timer0 Mo	ner0 Module Register					174*		
TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	121

Legend: — = Unimplemented location, read as '0'. Shaded cells are not used by the Timer0 module.

* Page provides register information.



FIGURE 25-4: SPI MASTER AND MULTIPLE SLAVE CONNECTION



25.2.1 SPI MODE REGISTERS

The MSSP1 module has five registers for SPI mode operation. These are:

- MSSP1 STATUS Register (SSP1STAT)
- MSSP1 Control Register 1 (SSP1CON1)
- MSSP1 Control Register 3 (SSP1CON3)
- MSSP1 Data Buffer Register (SSP1BUF)
- MSSP1 Address Register (SSP1ADD)
- MSSP1 Shift Register (SSP1SR) (Not directly accessible)

SSP1CON1 and SSP1STAT are the control and STATUS registers in SPI mode operation. The SSP1CON1 register is readable and writable. The lower six bits of the SSP1STAT are read-only. The upper two bits of the SSP1STAT are read/write.

In SPI master mode, SSP1ADD can be loaded with a value used in the Baud Rate Generator. More information on the Baud Rate Generator is available in **Section 25.7 "Baud Rate Generator"**.

SSP1SR is the shift register used for shifting data in and out. SSP1BUF provides indirect access to the SSP1SR register. SSP1BUF is the buffer register to which data bytes are written, and from which data bytes are read.

In receive operations, SSP1SR and SSP1BUF together create a buffered receiver. When SSP1SR receives a complete byte, it is transferred to SSP1BUF and the SSP1IF interrupt is set.

During transmission, the SSP1BUF is not buffered. A write to SSP1BUF will write to both SSP1BUF and SSP1SR.

FIGURE 26-2: EUSART RECEIVE BLOCK DIAGRAM



The operation of the EUSART module is controlled through three registers:

- Transmit Status and Control (TXSTA)
- Receive Status and Control (RCSTA)
- Baud Rate Control (BAUDCON)

These registers are detailed in Register 26-1, Register 26-2 and Register 26-3, respectively.

When the receiver or transmitter section is not enabled then the corresponding RX or TX pin may be used for general purpose input and output.

26.1.2.4 Receive Framing Error

Each character in the receive FIFO buffer has a corresponding framing error Status bit. A framing error indicates that a Stop bit was not seen at the expected time. The framing error status is accessed via the FERR bit of the RCSTA register. The FERR bit represents the status of the top unread character in the receive FIFO. Therefore, the FERR bit must be read before reading the RCREG.

The FERR bit is read-only and only applies to the top unread character in the receive FIFO. A framing error (FERR = 1) does not preclude reception of additional characters. It is not necessary to clear the FERR bit. Reading the next character from the FIFO buffer will advance the FIFO to the next character and the next corresponding framing error.

The FERR bit can be forced clear by clearing the SPEN bit of the RCSTA register which resets the EUSART. Clearing the CREN bit of the RCSTA register does not affect the FERR bit. A framing error by itself does not generate an interrupt.

Note:	If all receive characters in the receive
	FIFO have framing errors, repeated reads
	of the RCREG will not clear the FERR bit.

26.1.2.5 Receive Overrun Error

The receive FIFO buffer can hold two characters. An overrun error will be generated if a third character, in its entirety, is received before the FIFO is accessed. When this happens the OERR bit of the RCSTA register is set. The characters already in the FIFO buffer can be read but no additional characters will be received until the error is cleared. The error must be cleared by either clearing the CREN bit of the RCSTA register or by resetting the EUSART by clearing the SPEN bit of the RCSTA register.

26.1.2.6 Receiving 9-bit Characters

The EUSART supports 9-bit character reception. When the RX9 bit of the RCSTA register is set the EUSART will shift nine bits into the RSR for each character received. The RX9D bit of the RCSTA register is the ninth and Most Significant data bit of the top unread character in the receive FIFO. When reading 9-bit data from the receive FIFO buffer, the RX9D data bit must be read before reading the eight Least Significant bits from the RCREG.

26.1.2.7 Address Detection

A special Address Detection mode is available for use when multiple receivers share the same transmission line, such as in RS-485 systems. Address detection is enabled by setting the ADDEN bit of the RCSTA register.

Address detection requires 9-bit character reception. When address detection is enabled, only characters with the ninth data bit set will be transferred to the receive FIFO buffer, thereby setting the RCIF interrupt bit. All other characters will be ignored.

Upon receiving an address character, user software determines if the address matches its own. Upon address match, user software must disable address detection by clearing the ADDEN bit before the next Stop bit occurs. When user software detects the end of the message, determined by the message protocol used, software places the receiver back into the Address Detection mode by setting the ADDEN bit.

					SYNC = 0, BRGH = 1, BRG16 = 0							
BAUD	Fosc = 8.000 MHz			Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 1.000 MHz		
RATE	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	—	_			_	_	—	_	_	300	0.16	207
1200		_	_	1202	0.16	207	1200	0.00	191	1202	0.16	51
2400	2404	0.16	207	2404	0.16	103	2400	0.00	95	2404	0.16	25
9600	9615	0.16	51	9615	0.16	25	9600	0.00	23	—	—	_
10417	10417	0.00	47	10417	0.00	23	10473	0.53	21	10417	0.00	5
19.2k	19231	0.16	25	19.23k	0.16	12	19.2k	0.00	11	—	—	_
57.6k	55556	-3.55	8	—	—	—	57.60k	0.00	3	—	—	—
115.2k	_	_		_	_	_	115.2k	0.00	1	—	_	

TABLE 26-5: BAUD RATES FOR ASYNCHRONOUS MODES (CONTINUED)

					SYNC	= 0, BRG	l = 0, BRC	316 = 1				
BAUD	Fosc = 32.000 MHz			Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 11.0592 MHz		
RATE	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	300.0	0.00	6666	300.0	-0.01	4166	300.0	0.00	3839	300.0	0.00	2303
1200	1200	-0.02	3332	1200	-0.03	1041	1200	0.00	959	1200	0.00	575
2400	2401	-0.04	832	2399	-0.03	520	2400	0.00	479	2400	0.00	287
9600	9615	0.16	207	9615	0.16	129	9600	0.00	119	9600	0.00	71
10417	10417	0.00	191	10417	0.00	119	10378	-0.37	110	10473	0.53	65
19.2k	19.23k	0.16	103	19.23k	0.16	64	19.20k	0.00	59	19.20k	0.00	35
57.6k	57.14k	-0.79	34	56.818	-1.36	21	57.60k	0.00	19	57.60k	0.00	11
115.2k	117.6k	2.12	16	113.636	-1.36	10	115.2k	0.00	9	115.2k	0.00	5

					SYNC	C = 0, BRGH	l = 0, BRC	G16 = 1				
BAUD	Fosc = 8.000 MHz			Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 1.000 MHz		
RATE	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	299.9	-0.02	1666	300.1	0.04	832	300.0	0.00	767	300.5	0.16	207
1200	1199	-0.08	416	1202	0.16	207	1200	0.00	191	1202	0.16	51
2400	2404	0.16	207	2404	0.16	103	2400	0.00	95	2404	0.16	25
9600	9615	0.16	51	9615	0.16	25	9600	0.00	23	_	_	_
10417	10417	0.00	47	10417	0.00	23	10473	0.53	21	10417	0.00	5
19.2k	19.23k	0.16	25	19.23k	0.16	12	19.20k	0.00	11	_	_	_
57.6k	55556	-3.55	8	—	_	_	57.60k	0.00	3	—	_	_
115.2k		_	_	—	_	_	115.2k	0.00	1	_	_	_

R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0	R-0/0	R/W-0/0	
CPSON	CPSRM	<u> </u>		CPSRN	IG<1:0>	CPSOUT	TOXCS	
bit 7								
Legend:								
R = Readable I	bit	W = Writable	bit	U = Unimplen	nented bit, read	1 as '0'		
u = Bit is uncha	anged	x = Bit is unkr	nown	-n/n = Value a	t POR and BO	R/Value at all o	ther Resets	
'1' = Bit is set		'0' = Bit is clea	ared			_		
bit 7	CPSON: Cap 1 = CPS mod 0 = CPS mod	acitive Sensing dule is enabled dule is disabled	I Module Ena	ble bit				
bit 6	CPSRM: Cap 1 = Capacitiv 0 = Capacitiv	acitive Sensing /e Sensing moo /e Sensing moo	Reference N lule is in Varia lule is in Fixe	/lode bit able Voltage Re d Voltage Refer	ference mode. ence mode			
bit 5-4	Unimplemen	ted: Read as '	כ'					
bit 3-2	CPSRNG<1:0>: Capacitive Sensing Current Range bits <u>If CPSRM = 0 (Fixed Voltage Reference mode):</u> 00 = Oscillator is off 01 = Oscillator is in low range 10 = Oscillator is in medium range 11 = Oscillator is in high range							
	If CPSRM = 1 00 = Oscillato 01 = Oscillato 10 = Oscillato 11 = Oscillato	<u>(Variable Volta</u> or is on. Noise l or is in low rang or is in medium or is in high ran	age Referenc Detection mod le range ge	<u>e mode):</u> de. No Charge/	Discharge curr	ent is supplied.		
bit 1	CPSOUT: Capacitive Sensing Oscillator Status bit 1 = Oscillator is sourcing current (Current flowing out of the pin) 0 = Oscillator is sinking current (Current flowing into the pin)							
bit 0	 TOXCS: Timer0 External Clock Source Select bit <u>If TMR0CS = 1:</u> The T0XCS bit controls which clock external to the core/Timer0 module supplies Timer0: 1 = Timer0 clock source is the capacitive sensing oscillator 0 = Timer0 clock source is the T0CKI pin <u>If TMR0CS = 0:</u> Timer0 clock source is controlled by the core/Timer0 module and is Fosc/4 							

REGISTER 27-1: CPSCON0: CAPACITIVE SENSING CONTROL REGISTER 0

RETFIE	Return from Interrupt
Syntax:	[label] RETFIE
Operands:	None
Operation:	$\begin{array}{l} TOS \to PC, \\ 1 \to GIE \end{array}$
Status Affected:	None
Description:	Return from Interrupt. Stack is POPed and Top-of-Stack (TOS) is loaded in the PC. Interrupts are enabled by setting Global Interrupt Enable bit, GIE (INTCON<7>). This is a 2-cycle instruction.
Words:	1
Cycles:	2
Example:	RETFIE
	After Interrupt PC = TOS GIE = 1

RETURN	Return from Subroutine					
Syntax:	[label] RETURN					
Operands:	None					
Operation:	$TOS\toPC$					
Status Affected:	None					
Description:	Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a 2-cycle instruction.					

RETLW	Return with literal in W		Pototo Loft f through Corry
Syntax:	[<i>label</i>] RETLW k	RLF	
Operands:	$0 \le k \le 255$	Syntax:	[<i>label</i>] RLF f,d
Operation:	$k \rightarrow (W);$ TOS \rightarrow PC	Operands:	$0 \le f \le 127$ $d \in [0,1]$
Status Affected	None	Operation:	See description below
Description:	The W register is leaded with the 9 hit	Status Affected:	С
Description.	literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a 2-cycle instruction.	Description:	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
Words:	1		stored back in register T.
Cycles:	2		
Example:	CALL TABLE;W contains table	Words:	1
	;offset value . :W now has table value	Cycles:	1
TABLE	•	Example:	RLF REG1,0
	• ADDWF PC ;W = offset RETLW k1 ;Begin table RETLW k2 ; • • RETLW kn ; End of table		Before Instruction REG1 = 1110 0110 C = 0 After Instruction REG1 = 1110 0110 W = 1100 1100 C = 1
	Before Instruction W = 0x07 After Instruction W = value of k8		

TABLE 30-2: OSCILLATOR PARAMETERS

Operating Temperature $-40^{\circ}C \le TA \le +125^{\circ}C$									
Param No.	Sym.	Characteristic	Freq. Tolerance	Min.	Тур†	Max.	Units	Conditions	
OS08	HFosc	Internal Calibrated HFINTOSC	±2%		16.0	—	MHz	$0^{\circ}C \leq TA \leq \text{+}60^{\circ}C, \text{VDD} \geq 2.5 \text{V}$	
		Frequency ⁽¹⁾	±3%		16.0		MHz	$60^{\circ}C \leq TA \leq \text{+}85^{\circ}C, V\text{DD} \geq 2.5V$	
			±5%		16.0	_	MHz	$-40^{\circ}C \leq TA \leq +125^{\circ}C$	
OS08A MFosc		Internal Calibrated MFINTOSC	±2%		500		kHz	$0^{\circ}C \leq TA \leq +60^{\circ}C, \ V\text{dd} \geq 2.5V$	
		Frequency ⁽¹⁾	±3%		500		kHz	$60^{\circ}C \leq TA \leq \text{+}85^{\circ}C, \ V\text{DD} \geq 2.5V$	
			±5%	Ι	500	_	kHz	$-40^{\circ}C \leq TA \leq +125^{\circ}C$	
OS09	LFosc	Internal LFINTOSC Frequency	±25%	Ι	31	_	kHz	$-40^{\circ}C \leq TA \leq +125^{\circ}C$	
OS10*	TIOSC ST	HFINTOSC Wake-up from Sleep Start-up Time	Ι	_	5	8	μs		
		MFINTOSC Wake-up from Sleep Start-up Time	_	_	20	30	μS		

Standard Operating Conditions (unless otherwise stated)

These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: To ensure these oscillator frequency tolerances, VDD and VSS must be capacitively decoupled as close to the device as possible. $0.1 \ \mu\text{F}$ and $0.01 \ \mu\text{F}$ values in parallel are recommended.

TABLE 30-3: PLL CLOCK TIMING SPECIFICATIONS (VDD = 2.7V to 5.5V)

Param No.	Sym.	Characteristic	Min.	Тур†	Max.	Units	Conditions
F10	Fosc	Oscillator Frequency Range	4	—	8	MHz	
F11	Fsys	On-Chip VCO System Frequency	16	—	32	MHz	
F12	TRC	PLL Start-up Time (Lock Time)		—	2	ms	
F13*	ΔCLK	CLKOUT Stability (Jitter)	-0.25%	_	+0.25%	%	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

TABLE 30-5: **RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER** AND BROWN-OUT RESET PARAMETERS

Standard Operating Conditions (unless otherwise stated) Operating Temperature -40°C \leq TA \leq +125°C									
Param No.	Sym.	Characteristic	Min.	Тур†	Max.	Units	Conditions		
30	ТмсL	MCLR Pulse Width (low)	2			μS			
31	TWDTLP	Watchdog Timer Time-out Period (No Prescaler)	12	16	20	ms	Vdd = 3.3V-5V		
32	Tost	Oscillator Start-up Timer Period ⁽¹⁾		1024		Tosc			
33*	TPWRT	Power-up Timer Period, $\overline{PWRTE} = 0$	40	65	140	ms			
34*	Tioz	I/O high-impedance from MCLR Low or Watchdog Timer Reset	_	—	2.0	μS			
35	VBOR	Brown-out Reset Voltage ⁽²⁾	2.55 1.80	2.70 1.9	2.85 2.05	V V	BORV = 0 BORV = 1		
36*	VHYST	Brown-out Reset Hysteresis	20	35	75	mV	-40°C to +85°C		
37*	TBORDC	Brown-out Reset DC Response Time	0	1	35	μS	$V \text{DD} \leq V \text{BOR}$		
*	* These parameters are characterized but not tested								

These parameters are characterized but not tested.

Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance t only and are not tested.

Note 1: By design, the Oscillator Start-up (OST) counts the first 1.024 cycles, independent of frequency.

2: To ensure these voltage tolerances, VDD and Vss must be capacitively decoupled as close to the device as possible. 0.1 μ F and 0.01 μ F values in parallel are recommended.

FIGURE 30-10: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS









FIGURE 31-14: IDD, MFINTOSC MODE (Fosc = 500 kHz), PIC16F1824/8 ONLY









32.2 MPLAB XC Compilers

The MPLAB XC Compilers are complete ANSI C compilers for all of Microchip's 8, 16, and 32-bit MCU and DSC devices. These compilers provide powerful integration capabilities, superior code optimization and ease of use. MPLAB XC Compilers run on Windows, Linux or MAC OS X.

For easy source level debugging, the compilers provide debug information that is optimized to the MPLAB X IDE.

The free MPLAB XC Compiler editions support all devices and commands, with no time or memory restrictions, and offer sufficient code optimization for most applications.

MPLAB XC Compilers include an assembler, linker and utilities. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. MPLAB XC Compiler uses the assembler to produce its object file. Notable features of the assembler include:

- · Support for the entire device instruction set
- · Support for fixed-point and floating-point data
- Command-line interface
- · Rich directive set
- · Flexible macro language
- MPLAB X IDE compatibility

32.3 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel[®] standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code, and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB X IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multipurpose source files
- Directives that allow complete control over the assembly process

32.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

32.5 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC DSC devices. MPLAB XC Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- · Support for the entire device instruction set
- · Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

33.4 Package Details

The following sections give the technical details of the packages.

14-Lead Plastic Dual In-Line (P) – 300 mil Body [PDIP]

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



	INCHES					
Dimensior	n Limits	MIN	NOM	MAX		
Number of Pins	Ν	14				
Pitch	е		.100 BSC			
Top to Seating Plane	Α	-	-	.210		
Molded Package Thickness	A2	.115	.130	.195		
Base to Seating Plane	A1	.015	-	-		
Shoulder to Shoulder Width	E	.290	.310	.325		
Molded Package Width	E1	.240	.250	.280		
Overall Length	D	.735	.750	.775		
Tip to Seating Plane	L	.115	.130	.150		
Lead Thickness	С	.008	.010	.015		
Upper Lead Width	b1	.045	.060	.070		
Lower Lead Width	b	.014	.018	.022		
Overall Row Spacing §	eB	_	_	.430		

Notes:

- 1. Pin 1 visual index feature may vary, but must be located with the hatched area.
- 2. § Significant Characteristic.
- 3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
- 4. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-005B

20-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

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











Microchip Technology Drawing C04-094C Sheet 1 of 2

20-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

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



RECOMMENDED LAND PATTERN

	MILLIMETERS				
Dimension Limits		MIN	NOM	MAX	
Contact Pitch	E	1.27 BSC			
Contact Pad Spacing	С		9.40		
Contact Pad Width (X20)	Х			0.60	
Contact Pad Length (X20)	Y			1.95	
Distance Between Pads	Gx	0.67			
Distance Between Pads	G	7.45			

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2094A