

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

E·XFI

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	40MHz
Connectivity	I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	36
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	3.8K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	A/D 13x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf4620-i-pt

Email: info@E-XFL.COM

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

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families 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 Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is 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 products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION. QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

QUALITY MANAGEMENT SYSTEM CERTIFIED BY DNV ISO/TS 16949:2002

Trademarks

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, rfPIC and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Linear Active Thermistor, MXDEV, MXLAB, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, In-Circuit Serial Programming, ICSP, ICEPIC, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, PICkit, PICDEM, PICDEM.net, PICtail, PIC³² logo, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rfLAB, Select Mode, Total Endurance, UNI/O, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

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.

© 2008, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.



Microchip received ISO/TS-16949:2002 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

2.7 Clock Sources and Oscillator Switching

Like previous PIC18 devices, the PIC18F2525/2620/ 4525/4620 family includes a feature that allows the device clock source to be switched from the main oscillator to an alternate, low-frequency clock source. PIC18F2525/2620/4525/4620 devices offer two alternate clock sources. When an alternate clock source is enabled, the various power-managed operating modes are available.

Essentially, there are three clock sources for these devices:

- Primary oscillators
- · Secondary oscillators
- Internal oscillator block

The **primary oscillators** include the External Crystal and Resonator modes, the External RC modes, the External Clock modes and the internal oscillator block. The particular mode is defined by the FOSC3:FOSC0 Configuration bits. The details of these modes are covered earlier in this chapter. The **secondary oscillators** are those external sources not connected to the OSC1 or OSC2 pins. These sources may continue to operate even after the controller is placed in a power-managed mode.

PIC18F2525/2620/4525/4620 devices offer the Timer1 oscillator as a secondary oscillator. This oscillator, in all power-managed modes, is often the time base for functions such as a Real-Time Clock (RTC).

Most often, a 32.768 kHz watch crystal is connected between the RC0/T1OSO/T13CKI and RC1/T1OSI pins. Like the LP Oscillator mode circuit, loading capacitors are also connected from each pin to ground.

The Timer1 oscillator is discussed in greater detail in **Section 12.3 "Timer1 Oscillator"**.

In addition to being a primary clock source, the **internal oscillator block** is available as a power-managed mode clock source. The INTRC source is also used as the clock source for several special features, such as the WDT and Fail-Safe Clock Monitor.

The clock sources for the PIC18F2525/2620/4525/4620 devices are shown in Figure 2-8. See **Section 23.0 "Special Features of the CPU"** for Configuration register details.





NOTES:

13.2 Timer2 Interrupt

Timer2 can also generate an optional device interrupt. The Timer2 output signal (TMR2 to PR2 match) provides the input for the 4-bit output counter/ postscaler. This counter generates the TMR2 match interrupt flag which is latched in TMR2IF (PIR1<1>). The interrupt is enabled by setting the TMR2 Match Interrupt Enable bit, TMR2IE (PIE1<1>).

A range of 16 postscale options (from 1:1 through 1:16 inclusive) can be selected with the postscaler control bits, T2OUTPS3:T2OUTPS0 (T2CON<6:3>).

13.3 Timer2 Output

The unscaled output of TMR2 is available primarily to the CCP modules, where it is used as a time base for operations in PWM mode.

Timer2 can be optionally used as the shift clock source for the MSSP module operating in SPI mode. Additional information is provided in Section 17.0 "Master Synchronous Serial Port (MSSP) Module".



FIGURE 13-1: TIMER2 BLOCK DIAGRAM

TABLE 13-1: REGISTERS ASSOCIATED WITH TIMER2 AS A TIMER/COUNTER

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	49
PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	52
PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	52
IPR1	PSPIP ⁽¹⁾	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	52
TMR2	2 Timer2 Register							50	
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	50
PR2	Timer2 Period Register								50

Legend: — = unimplemented, read as '0'. Shaded cells are not used by the Timer2 module.

Note 1: These bits are unimplemented on 28-pin devices and read as '0'.

15.3 Compare Mode

In Compare mode, the 16-bit CCPRx register value is constantly compared against either the TMR1 or TMR3 register pair value. When a match occurs, the CCPx pin can be:

- driven high
- driven low
- toggled (high-to-low or low-to-high)
- remain unchanged (that is, reflects the state of the I/O latch)

The action on the pin is based on the value of the mode select bits (CCPxM3:CCPxM0). At the same time, the interrupt flag bit, CCPxIF, is set.

15.3.1 CCP PIN CONFIGURATION

The user must configure the CCPx pin as an output by clearing the appropriate TRIS bit.

Clearing the CCP2CON register will force				
the RB3 or RC1 compare output latch				
(depending on device configuration) to the				
default low level. This is not the PORTB or				
PORTC I/O data latch.				

15.3.2 TIMER1/TIMER3 MODE SELECTION

Timer1 and/or Timer3 must be running in Timer mode or Synchronized Counter mode if the CCP module is using the compare feature. In Asynchronous Counter mode, the compare operation may not work.

15.3.3 SOFTWARE INTERRUPT MODE

When the Generate Software Interrupt mode is chosen (CCPxM3:CCPxM0 = 1010), the corresponding CCPx pin is not affected. Only a CCP interrupt is generated, if enabled, and the CCPxIE bit is set.

15.3.4 SPECIAL EVENT TRIGGER

Both CCP modules are equipped with a Special Event Trigger. This is an internal hardware signal generated in Compare mode to trigger actions by other modules. The Special Event Trigger is enabled by selecting the Compare Special Event Trigger mode (CCPxM3:CCPxM0 = 1011).

For either CCP module, the Special Event Trigger resets the Timer register pair for whichever timer resource is currently assigned as the module's time base. This allows the CCPRx registers to serve as a programmable Period register for either timer.

The Special Event Trigger for CCP2 can also start an A/D conversion. In order to do this, the A/D converter must already be enabled.

FIGURE 15-2: COMPARE MODE OPERATION BLOCK DIAGRAM



17.3.2 OPERATION

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits (SSPCON1<5:0> and SSPSTAT<7:6>). These control bits allow the following to be specified:

- Master mode (SCK is the clock output)
- Slave mode (SCK is the clock input)
- Clock Polarity (Idle state of SCK)
- Data Input Sample Phase (middle or end of data output time)
- Clock Edge (output data on rising/falling edge of SCK)
- Clock Rate (Master mode only)
- Slave Select mode (Slave mode only)

The MSSP consists of a transmit/receive shift register (SSPSR) and a buffer register (SSPBUF). The SSPSR shifts the data in and out of the device, MSb first. The SSPBUF holds the data that was written to the SSPSR until the received data is ready. Once the 8 bits of data have been received, that byte is moved to the SSPBUF register. Then, the Buffer Full detect bit, BF (SSPSTAT<0>) and the interrupt flag bit, SSPIF, are set. This double-buffering of the received data (SSPBUF) allows the next byte to start reception before reading the data that was just received. Any write to the SSPBUF register during transmission/reception of data will be ignored and the write collision detect bit, WCOL

(SSPCON1<7>), will be set. User software must clear the WCOL bit so that it can be determined if the following write(s) to the SSPBUF register completed successfully.

When the application software is expecting to receive valid data, the SSPBUF should be read before the next byte of data to transfer is written to the SSPBUF. The Buffer Full bit, BF (SSPSTAT<0>), indicates when SSPBUF has been loaded with the received data (transmission is complete). When the SSPBUF is read, the BF bit is cleared. This data may be irrelevant if the SPI is only a transmitter. Generally, the MSSP interrupt is used to determine when the transmission/reception has completed. The SSPBUF must be read and/or written. If the interrupt method is not going to be used, then software polling can be done to ensure that a write collision does not occur. Example 17-1 shows the loading of the SSPBUF (SSPSR) for data transmission.

The SSPSR is not directly readable or writable and can only be accessed by addressing the SSPBUF register. Additionally, the MSSP status register (SSPSTAT) indicates the various status conditions.

Note:	The SSPBUF register cannot be used with					
	read-modify-write	instructions	such	as		
	BCF, BTFSC and COMF, etc.					

Note: To avoid lost data in Master mode, a read of the SSPBUF must be performed to clear the Buffer Full (BF) detect bit (SSPSTAT<0>) between each transmission.

EXAMPLE 17-1: LOADING THE SSPBUF (SSPSR) REGISTER

LOOP	BTFSS BRA MOVF	SSPSTAT, BF LOOP SSPBUF, W	<pre>;Has data been received (transmit complete)? ;No ;WREG reg = contents of SSPBUF</pre>
	MOVWF	RXDATA	;Save in user RAM, if data is meaningful
	MOVF MOVWF	TXDATA, W SSPBUF	;W reg = contents of TXDATA ;New data to xmit

17.4.6.1 I²C Master Mode Operation

The master device generates all of the serial clock pulses and the Start and Stop conditions. A transfer is ended with a Stop condition or with a Repeated Start condition. Since the Repeated Start condition is also the beginning of the next serial transfer, the I²C bus will not be released.

In Master Transmitter mode, serial data is output through SDA, while SCL outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the Read/Write (R/W) bit. In this case, the R/W bit will be logic '0'. Serial data is transmitted 8 bits at a time. After each byte is transmitted, an Acknowledge bit is received. Start and Stop conditions are output to indicate the beginning and the end of a serial transfer.

In Master Receive mode, the first byte transmitted contains the slave address of the transmitting device (7 bits) and the R/W bit. In this case, the R/W bit will be logic '1'. Thus, the first byte transmitted is a 7-bit slave address followed by a '1' to indicate the receive bit. Serial data is received via SDA, while SCL outputs the serial clock. Serial data is received 8 bits at a time. After each byte is received, an Acknowledge bit is transmitted. Start and Stop conditions indicate the beginning and end of transmission.

The Baud Rate Generator used for the SPI mode operation is used to set the SCL clock frequency for either 100 kHz, 400 kHz or 1 MHz I²C operation. See **Section 17.4.7 "Baud Rate"** for more detail.

A typical transmit sequence would go as follows:

- 1. The user generates a Start condition by setting the Start Enable bit, SEN (SSPCON2<0>).
- 2. SSPIF is set. The MSSP module will wait the required start time before any other operation takes place.
- 3. The user loads the SSPBUF with the slave address to transmit.
- 4. Address is shifted out the SDA pin until all 8 bits are transmitted.
- 5. The MSSP module shifts in the ACK bit from the slave device and writes its value into the SSPCON2 register.
- 6. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPIF bit.
- 7. The user loads the SSPBUF with eight bits of data.
- 8. Data is shifted out the SDA pin until all 8 bits are transmitted.
- The MSSP module shifts in the ACK bit from the slave device and writes its value into the SSPCON2 register.
- 10. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPIF bit.
- 11. The user generates a Stop condition by setting the Stop Enable bit, PEN (SSPCON2<2>).
- 12. Interrupt is generated once the Stop condition is complete.

17.4.14 SLEEP OPERATION

While in Sleep mode, the I²C module can receive addresses or data and when an address match or complete byte transfer occurs, wake the processor from Sleep (if the MSSP interrupt is enabled).

17.4.15 EFFECTS OF A RESET

A Reset disables the MSSP module and terminates the current transfer.

17.4.16 MULTI-MASTER MODE

In Multi-Master mode, the interrupt generation on the detection of the Start and Stop conditions allows the determination of when the bus is free. The Stop (P) and Start (S) bits are cleared from a Reset or when the MSSP module is disabled. Control of the I^2C bus may be taken when the P bit (SSPSTAT<4>) is set, or the bus is Idle, with both the S and P bits clear. When the bus is busy, enabling the MSSP interrupt will generate the interrupt when the Stop condition occurs.

In multi-master operation, the SDA line must be monitored for arbitration to see if the signal level is the expected output level. This check is performed in hardware with the result placed in the BCLIF bit.

The states where arbitration can be lost are:

- Address Transfer
- Data Transfer
- A Start Condition
- A Repeated Start Condition
- An Acknowledge Condition

17.4.17 MULTI -MASTER COMMUNICATION, BUS COLLISION AND BUS ARBITRATION

Multi-Master mode support is achieved by bus arbitration. When the master outputs address/data bits onto the SDA pin, arbitration takes place when the master outputs a '1' on SDA, by letting SDA float high and another master asserts a '0'. When the SCL pin floats high, data should be stable. If the expected data on SDA is a '1' and the data sampled on the SDA pin = 0, then a bus collision has taken place. The master will set the Bus Collision Interrupt Flag, BCLIF, and reset the I^2C port to its Idle state (Figure 17-25).

If a transmit was in progress when the bus collision occurred, the transmission is halted, the BF flag is cleared, the SDA and SCL lines are deasserted and the SSPBUF can be written to. When the user services the bus collision Interrupt Service Routine, and if the I^2C bus is free, the user can resume communication by asserting a Start condition.

If a Start, Repeated Start, Stop or Acknowledge condition was in progress when the bus collision occurred, the condition is aborted, the SDA and SCL lines are deasserted and the respective control bits in the SSPCON2 register are cleared. When the user services the bus collision Interrupt Service Routine, and if the I^2C bus is free, the user can resume communication by asserting a Start condition.

The master will continue to monitor the SDA and SCL pins. If a Stop condition occurs, the SSPIF bit will be set.

A write to the SSPBUF will start the transmission of data at the first data bit, regardless of where the transmitter left off when the bus collision occurred.

In Multi-Master mode, the interrupt generation on the detection of Start and Stop conditions allows the determination of when the bus is free. Control of the I^2C bus can be taken when the P bit is set in the SSPSTAT register, or the bus is Idle and the S and P bits are cleared.

FIGURE 17-25: BUS COLLISION TIMING FOR TRANSMIT AND ACKNOWLEDGE



18.2.4.1 Special Considerations Using Auto-Wake-up

Since auto-wake-up functions by sensing rising edge transitions on RX/DT, information with any state changes before the Stop bit may signal a false End-Of-Character (EOC) and cause data or framing errors. To work properly, therefore, the initial character in the transmission must be all '0's. This can be 00h (8 bytes) for standard RS-232 devices or 000h (12 bits) for LIN bus.

Oscillator start-up time must also be considered, especially in applications using oscillators with longer start-up intervals (i.e., XT or HS mode). The Sync Break (or wake-up signal) character must be of sufficient length and be followed by a sufficient interval to allow enough time for the selected oscillator to start and provide proper initialization of the EUSART.

18.2.4.2 Special Considerations Using the WUE Bit

The timing of WUE and RCIF events may cause some confusion when it comes to determining the validity of received data. As noted, setting the WUE bit places the EUSART in an Idle mode. The wake-up event causes a receive interrupt by setting the RCIF bit. The WUE bit is cleared after this when a rising edge is seen on RX/DT. The interrupt condition is then cleared by reading the RCREG register. Ordinarily, the data in RCREG will be dummy data and should be discarded.

The fact that the WUE bit has been cleared (or is still set) and the RCIF flag is set should not be used as an indicator of the integrity of the data in RCREG. Users should consider implementing a parallel method in firmware to verify received data integrity.

To assure that no actual data is lost, check the RCIDL bit to verify that a receive operation is not in process. If a receive operation is not occurring, the WUE bit may then be set just prior to entering the Sleep mode.





FIGURE 18-9: AUTO-WAKE-UP BIT (WUE) TIMINGS DURING SLEEP



					DAMA	D/M/ O	
	0-0				R/W-U		R/W-U
		ACQ12	ACQTT	ACQTU	ADC52	ADCS1	ADCSU
Dit 7							Dit U
Legena:	1.5						
R = Readable	e bit	VV = VVritable	Dit		nented bit, read		
-n = Value at	POR	'1' = Bit is set		'0' = Bit is clea	ared	x = Bit is unkr	nown
bit 7	ADFM: A/D F 1 = Right just 0 = Left justif	Result Format S tified ied	elect bit				
bit 6	Unimplemer	ted: Read as '	כי				
bit 5-3	ACQT2:ACQ	T0: A/D Acquis	ition Time Sel	ect bits			
	111 = 20 TAD $110 = 16 TAD$ $101 = 12 TAD$ $100 = 8 TAD$ $011 = 6 TAD$ $010 = 4 TAD$ $001 = 2 TAD$ $001 = 2 TAD$						
bit 2-0	ADCS2:ADCS0: A/D Conversion Clock Select bits 111 = FRC (clock derived from A/D RC oscillator) ⁽¹⁾ 110 = Fosc/64 101 = Fosc/16 100 = Fosc/4 011 = FRC (clock derived from A/D RC oscillator) ⁽¹⁾ 010 = Fosc/32 001 = Fosc/8 000 = Fosc/2						

REGISTER 19-3: ADCON2: A/D CONTROL REGISTER 2

Note 1: If the A/D FRC clock source is selected, a delay of one TCY (instruction cycle) is added before the A/D clock starts. This allows the SLEEP instruction to be executed before starting a conversion.

REGISTER 23-14: WDTCON: WATCHDOG TIMER CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	SWDTEN ⁽¹⁾
bit 7							bit 0

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

bit 7-1	Unimplemented: Read as '0'
---------	----------------------------

bit 0 SWDTEN: Software Controlled Watchdog Timer Enable bit⁽¹⁾

- 1 = Watchdog Timer is on
- 0 = Watchdog Timer is off

Note 1: This bit has no effect if the Configuration bit, WDTEN, is enabled.

TABLE 23-2: SUMMARY OF WATCHDOG TIMER REGISTERS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
RCON	IPEN	SBOREN ⁽¹⁾		RI	TO	PD	POR	BOR	50
WDTCON	_	—	—	_	_	_	_	SWDTEN	50

Legend: — = unimplemented, read as '0'. Shaded cells are not used by the Watchdog Timer.

Note 1: The SBOREN bit is only available when the BOREN1:BOREN0 Configuration bits = 01; otherwise, it is disabled and reads as '0'. See Section 4.4 "Brown-out Reset (BOR)".

ANDWF AND W with f						
Syntax		ANDWF	f {,d {,a}	}		
Operai	nds:	$0 \le f \le 255$ $d \in [0,1]$ $a \in [0,1]$	$0 \le f \le 255$ d $\in [0,1]$ a $\in [0,1]$			
Operat	tion:	(W) .AND.	(f) \rightarrow des	st		
Status	Affected:	N, Z				
Encodi	ing:	0001	01da	ffff	ffff	
Descri	ption:	The contents of W are ANDed with register 'f'. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default). If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank. If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \le 95$ (5Fh). See Section 24.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode" for details				
Words	:	1				
Cycles	:	1				
Q Cyc	cle Activity:					
_	Q1	Q2	Q3	}	Q4	
	Decode	Read register 'f'	Proce Dat	ess a c	Write to destination	
Example:		ANDWF	REG,	0, 0		
В	efore Instruc	tion				
A	W REG fter Instructio	= 17h = C2h on				
	W REG	= 02h = C2h				

вС		Branch if	Carry		
Synta	ax:	BC n			
Oper	ands:	-128 ≤ n ≤	127		
Oper	ation:	if Carry bit (PC) + 2 +	is '1', 2n → PC		
Statu	s Affected:	None			
Enco	ding:	1110	0010 nn	nn nnnn	
Desc	ription:	If the Carry bit is '1', then the program will branch. The 2's complement number '2n' is added to the PC. Since the PC will hav incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is then a two-cycle instruction.			
Word	ls:	1			
Cycle	es:	1(2)			
Q C If Ju	ycle Activity:				
	Q1	Q2	Q3	Q4	
	Decode	Read literal 'n'	Process Data	Write to PC	
	No operation	No operation	No operation	No operation	
lf No	o Jump:				
	Q1	Q2	Q3	Q4	
	Decode	Read literal 'n'	Process Data	No operation	
<u>Exan</u>	nple:	HERE	BC 5		
	Before Instruct PC After Instructio If Carry PC If Carry PC	tion = ac on = 1; = ac = 0; = ac	ldress (HERE ldress (HERE ldress (HERE) + 12) + 2)	

MOVFF	Move f to	o f				
Syntax:	MOVFF f	s,f _d				
Operands:	$\begin{array}{l} 0 \leq f_s \leq 40 \\ 0 \leq f_d \leq 40 \end{array}$	95 95				
Operation:	$(\mathrm{f}_{\mathrm{S}}) \to \mathrm{f}_{\mathrm{d}}$					
Status Affected:	None					
Encoding: 1st word (source) 2nd word (destin.)	1100 1111	1100 ffff ffff ffff _s 1111 ffff ffff ffff _d				
	The contents of source register f_s' are moved to destination register f_d' . Location of source f_s' can be anywhere in the 4096-byte data space (000h to FFFh) and location of destination f_d' can also be anywhere from 000h to FFFh. Either source or destination can be W (a useful special situation). MOVFF is particularly useful for transferring a data memory location to a peripheral register (such as the transmit buffer or an I/O port). The MOVFF instruction cannot use the PCL, TOSU, TOSH or TOSL as the destination register.					
Words:	2					
Cycles:	2 (3)					
Q Cycle Activity:						
Q1	Q2	Q3	3	Q4		

MOVLB	Move Literal to Low Nibble in BSR			
Syntax:	MOVLW k			
Operands:	$0 \leq k \leq 255$			
Operation:	$k \to BSR$			
Status Affected:	None			
Encoding:	0000	0001	kkkk	kkkk
Description:	The eight-bi Bank Select BSR<7:4> a of the value	it literal 'k t Register always rer of k ₇ :k ₄ .	' is loade '(BSR). ' mains '0	ed into the The value of ', regardless
Words:	1			
Cycles:	1			
Q Cycle Activity:				
Q1	Q2	G	13	Q4
Decode	Read literal 'k'	Pro Da	cess ata	Write literal 'k' to BSR
Example:	MOVLB	5		
Before Instruction BSR Register = 02h				

After Instruction	
BSR Register =	05h

Q1	Q2	Q3	Q4
Decode	Read register 'f' (src)	Process Data	No operation
Decode	No operation No dummy read	No operation	Write register 'f' (dest)

Example: MOVFF REG1, REG2

Before Instruction REG1 REG2	= =	33h 11h
After Instruction		
REG1 REG2	= =	33h 33h

24.2.3 BYTE-ORIENTED AND BIT-ORIENTED INSTRUCTIONS IN INDEXED LITERAL OFFSET MODE

Note:	Enabling	the	PIC18	instruction	set
	extension	may	cause le	gacy applicat	tions
	to behave	erratically or fail entirely.			

In addition to eight new commands in the extended set, enabling the extended instruction set also enables Indexed Literal Offset Addressing mode (**Section 5.5.1 "Indexed Addressing with Literal Offset**"). This has a significant impact on the way that many commands of the standard PIC18 instruction set are interpreted.

When the extended set is disabled, addresses embedded in opcodes are treated as literal memory locations: either as a location in the Access Bank ('a' = 0) or in a GPR bank designated by the BSR ('a' = 1). When the extended instruction set is enabled and 'a' = 0, however, a file register argument of 5Fh or less is interpreted as an offset from the pointer value in FSR2 and not as a literal address. For practical purposes, this means that all instructions that use the Access RAM bit as an argument – that is, all byte-oriented and bitoriented instructions, or almost half of the core PIC18 instructions – may behave differently when the extended instruction set is enabled.

When the content of FSR2 is 00h, the boundaries of the Access RAM are essentially remapped to their original values. This may be useful in creating backward compatible code. If this technique is used, it may be necessary to save the value of FSR2 and restore it when moving back and forth between C and assembly routines in order to preserve the Stack Pointer. Users must also keep in mind the syntax requirements of the extended instruction set (see Section 24.2.3.1 "Extended Instruction Syntax with Standard PIC18 Commands").

Although the Indexed Literal Offset Addressing mode can be very useful for dynamic stack and pointer manipulation, it can also be very annoying if a simple arithmetic operation is carried out on the wrong register. Users who are accustomed to the PIC18 programming must keep in mind that, when the extended instruction set is enabled, register addresses of 5Fh or less are used for Indexed Literal Offset Addressing mode.

Representative examples of typical byte-oriented and bit-oriented instructions in the Indexed Literal Offset Addressing mode are provided on the following page to show how execution is affected. The operand conditions shown in the examples are applicable to all instructions of these types.

24.2.3.1 Extended Instruction Syntax with Standard PIC18 Commands

When the extended instruction set is enabled, the file register argument, 'f', in the standard byte-oriented and bit-oriented commands is replaced with the literal offset value, 'k'. As already noted, this occurs only when 'f' is less than or equal to 5Fh. When an offset value is used, it must be indicated by square brackets ("[]"). As with the extended instructions, the use of brackets indicates to the compiler that the value is to be interpreted as an index or an offset. Omitting the brackets, or using a value greater than 5Fh within brackets, will generate an error in the MPASM Assembler.

If the index argument is properly bracketed for Indexed Literal Offset Addressing mode, the Access RAM argument is never specified; it will automatically be assumed to be '0'. This is in contrast to standard operation (extended instruction set disabled) when 'a' is set on the basis of the target address. Declaring the Access RAM bit in this mode will also generate an error in the MPASM Assembler.

The destination argument, 'd', functions as before.

In the latest versions of the MPASM assembler, language support for the extended instruction set must be explicitly invoked. This is done with either the command line option, $/_{Y}$, or the PE directive in the source listing.

24.2.4 CONSIDERATIONS WHEN ENABLING THE EXTENDED INSTRUCTION SET

It is important to note that the extensions to the instruction set may not be beneficial to all users. In particular, users who are not writing code that uses a software stack may not benefit from using the extensions to the instruction set.

Additionally, the Indexed Literal Offset Addressing mode may create issues with legacy applications written to the PIC18 assembler. This is because instructions in the legacy code may attempt to address registers in the Access Bank below 5Fh. Since these addresses are interpreted as literal offsets to FSR2 when the instruction set extension is enabled, the application may read or write to the wrong data addresses.

When porting an application to the PIC18F2525/2620/ 4525/4620, it is very important to consider the type of code. A large, re-entrant application that is written in 'C' and would benefit from efficient compilation will do well when using the instruction set extensions. Legacy applications that heavily use the Access Bank will most likely not benefit from using the extended instruction set.

ADD	OWF	ADD W to Indexed (Indexed Literal Offset mode)				
Synta	ax:	ADDWF	[k] {,d}			
Oper	ands:	$\begin{array}{l} 0 \leq k \leq 95 \\ d \in [0,1] \end{array}$				
Oper	ation:	(W) + ((FS	SR2) + k) -	\rightarrow dest		
Statu	is Affected:	N, OV, C,	DC, Z			
Enco	oding:	0010	01d0	kkkk	kkkk	
Desc	cription:	The contents of W are added to the contents of the register indicated by FSR2, offset by the value 'k'. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default).				
Word	ds:	1				
Cycles:		1				
QC	ycle Activity:					
	Q1	Q2	Q3		Q4	
	Decode	Read 'k'	Proce Dat	ess \ a de	Write to stination	
<u>Exar</u>	nple:	ADDWF	[OFST]	, 0		
	Before Instructi	on				
	W OFST FSR2 Contents of 0A2Ch	= = =	17h 2Ch 0A00h 20h	1		
	After Instruction W Contents	ו =	37h			
	of 0A2Ch	=	20h			

BSF Bit Set Indexed (Indexed Literal Offset mode)					ode)		
Synta	ax:	BSF [k],	b				
Oper	ands:	$\begin{array}{l} 0 \leq f \leq 95 \\ 0 \leq b \leq 7 \end{array}$	$\begin{array}{l} 0 \leq f \leq 95 \\ 0 \leq b \leq 7 \end{array}$				
Oper	ation:	$1 \rightarrow ((FSR2) + k) < b >$					
Statu	s Affected:	None					
Enco	ding:	1000		bbb0	kkk	k	kkkk
Desc	Description: Bit 'b' of the register indicated by FS offset by the value 'k', is set.				by FSR2,		
Word	ls:	1	1				
Cycles:		1	1				
QC	ycle Activity:						
	Q1	Q2		Q3			Q4
	Decode	Read register 'f'		Proce Data	SS A	V de	Vrite to stination
<u>Exan</u>	nple:	BSF	[]	FLAG_0	FST]	, 7	
Before Instructio FLAG_OFS FSR2 Contents of 0A0Ah		tion FST = =	=	0Ah 0A00h 55h	I		
	After Instruction Contents of 0A0Ah	on I =	=	D5h			

SET	F	Set Indexed (Indexed Literal Offset mode)				
Synt	ax:	SETF [k]				
Ope	rands:	$0 \leq k \leq 95$				
Ope	ration:	$FFh \to ((F$	SR2) + k)			
Statu	us Affected:	None				
Enco	oding:	0110 1000 kkkk kkkk				
Des	cription:	The contents of the register indicated by FSR2, offset by 'k', are set to FFh.				
Wor	ds:	1				
Cycl	es:	1				
QC	Cycle Activity:					
Q1		Q2	Q3	3	Q4	
	Decode	Read 'k'	Proce Dat	ess a r	Write egister	
<u>Exar</u>	<u>mple:</u>	SETF	[OFST]			
	Before Instruct OFST FSR2	tion = 20 = 04	Ch A00h			

of 0A2Ch	=	00h
After Instruction		
Contents of 0A2Ch	=	FFh



	TABLE 26-25:	A/D CONVERSION REQUIREMENTS
--	--------------	-----------------------------

Param No.	Symbol	Characteristic		Min	Мах	Units	Conditions
130	Tad	A/D Clock Period	PIC18FXXXX	0.7	25.0 ⁽¹⁾	μS	Tosc based, VREF $\ge 3.0V$
			PIC18LFXXXX	1.4	25.0 ⁽¹⁾	μS	VDD = 2.0V; Tosc based, VREF full range
			PIC18FXXXX		1	μS	A/D RC mode
			PIC18LFXXXX		3	μS	VDD = 2.0V; A/D RC mode
131	TCNV	Conversion Time (not including acquisition)	on time) (Note 2)	11	12	Tad	
132	TACQ	Acquisition Time (Note	3)	1.4	—	μS	-40°C to +85°C
135	Tswc	Switching Time from Convert \rightarrow Sample			(Note 4)		
TBD	TDIS	Discharge Time		0.2	_	μS	

Note 1: The time of the A/D clock period is dependent on the device frequency and the TAD clock divider.

2: ADRES register may be read on the following TCY cycle.

3: The time for the holding capacitor to acquire the "New" input voltage when the voltage changes full scale after the conversion (VDD to Vss or Vss to VDD). The source impedance (*Rs*) on the input channels is 50Ω.

4: On the following cycle of the device clock.











FIGURE 27-39: WDT PERIOD vs. VDD ACROSS TEMPERATURE (1:1 POSTSCALER, -40°C TO +125°C)

NOTES:

44-Lead Plastic Thin Quad Flatpack (PT) – 10x10x1 mm Body, 2.00 mm [TQFP]

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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.80 BSC		
Contact Pad Spacing	C1		11.40	
Contact Pad Spacing	C2		11.40	
Contact Pad Width (X44)	X1			0.55
Contact Pad Length (X44)	Y1			1.50
Distance Between Pads	G	0.25		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2076A