

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	64MHz
Connectivity	I ² C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	36
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 35x10b; D/A 1x5b
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/pic18lf45k40t-i-pt

Email: info@E-XFL.COM

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

U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
_	TMR6MD	TMR5MD	TMR4MD	TMR3MD	TMR2MD	TMR1MD	TMR0MD
bit 7							bit 0
Legend:							
R = Readable	e bit	W = Writable	bit	U = Unimplem	nented bit, read	as '0'	
u = Bit is unc	hanged	x = Bit is unkn	iown	-n/n = Value a	t POR and BO	R/Value at all o	ther Resets
'1' = Bit is set	t	'0' = Bit is clea	ared	q = Value dep	ends on condit	ion	
bit 7	Unimplemen	ted: Read as 'd)'				
bit 6	TMR6MD: Di	sable Timer TM	IR6 bit				
	1 = TMR6 m	odule disabled					
	0 = TMR6 m	odule enabled					
bit 5	TMR5MD: Di	sable Timer TM	IR5 bit				
	1 = TMR5 m	odule disabled					
	0 = TMR5 m	odule enabled					
bit 4	TMR4MD: Di	sable Timer TM	IR4 bit				
	1 = TMR4 m	odule disabled					
	0 = TMR4 m	odule enabled					
bit 3	TMR3MD: Di	sable Timer TM	IR3 bit				
	1 = TMR3 m	odule disabled					
	0 = TMR3 m	odule enabled					
bit 2	TMR2MD: Di	sable Timer TM	IR2 bit				
	1 = TMR2 m	odule disabled					
b : t . d							
DIC	1 mR1 mD: Dis	sable limer liv	IR'I DI				
	$\perp = TMRTm$	odule disabled					
hit 0		sable Timer TM	IR0 hit				
Situ	1 = TMR0 m	odule disabled					
	0 = TMR0 m	odule enabled					

REGISTER 7-2: PMD1: PMD CONTROL REGISTER 1

8.14 Power Control (PCON0) Register

The Power Control (PCON0) register contains flag bits to differentiate between a:

- Brown-out Reset (BOR)
- Power-on Reset (POR)
- Reset Instruction Reset (RI)
- MCLR Reset (RMCLR)
- Watchdog Timer Reset (RWDT)
- Watchdog Window Violation (WDTWV)
- Stack Underflow Reset (STKUNF)
- Stack Overflow Reset (STKOVF)

The PCON0 register bits are shown in Register 8-2.

Hardware will change the corresponding register bit during the Reset process; if the Reset was not caused by the condition, the bit remains unchanged (Table 8-3).

Software should reset the bit to the inactive state after restart (hardware will not reset the bit).

Software may also set any PCON0 bit to the active state, so that user code may be tested, but no Reset action will be generated.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BORCON	SBOREN							BORRDY	75
PCON0	STKOVF	STKUNF	WDTWV	RWDT	RMCLR	RI	POR	BOR	76
STATUS	_	TO	PD	Ν	OV	Z	DC	С	118
WDTCON0	_	_		WDTPS<4:0> SEN					85
WDTCON1	_	V	/DTCS<2:0>	>		W	/INDOW<2:()>	86

TABLE 8-4: SUMMARY OF REGISTERS ASSOCIATED WITH RESETS

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by Resets.



FIGURE 11-7: PFM ROW ERASE FLOWCHART



11.1.6 WRITING TO PROGRAM FLASH MEMORY

The programming write block size is described in Table 11-3. Word or byte programming is not supported. Table writes are used internally to load the holding registers needed to program the Flash memory. There are only as many holding registers as there are bytes in a write block. Refer to Table 11-3 for write latch size.

Since the table latch (TABLAT) is only a single byte, the TBLWT instruction needs to be executed multiple times for each programming operation. The write protection state is ignored for this operation. All of the table write operations will essentially be short writes because only the holding registers are written. NVMIF is not affected while writing to the holding registers.

After all the holding registers have been written, the programming operation of that block of memory is started by configuring the NVMCON1 register for a program memory write and performing the long write sequence.

If the PFM address in the TBLPTR is write-protected or if TBLPTR points to an invalid location, the WR bit is cleared without any effect and the WREER is signaled.

The long write is necessary for programming the internal Flash. CPU operation is suspended during a long write cycle and resumes when the operation is complete. The long write operation completes in one instruction cycle. When complete, WR is cleared in hardware and NVMIF is set and an interrupt will occur if NVMIE is also set. The latched data is reset to all '1s'. WREN is not changed.

The internal programming timer controls the write time. The write/erase voltages are generated by an on-chip charge pump, rated to operate over the voltage range of the device.

Note:	The default value of the holding registers on device Resets and after write operations is FFh. A write of FFh to a holding register does not modify that byte. This means that individual bytes of program memory may be modified, provided that the change does not attempt to change any bit from a '0' to a '1'. When modifying individual bytes, it is not necessary to load all holding registers
	before executing a long write operation.

The user needs to load the TBLPTR and TABLAT register with the address and data byte respectively before executing the write command. An unlock sequence needs to be followed for writing to the USER IDs/ DEVICE IDs/CONFIG words (Section 11.1.4, NVM Unlock Sequence). If WRTC = 0 or if TBLPTR points an invalid address location (see Table 11-3), WR bit is cleared without any effect and WRERR is set.

A single CONFIG word byte is written at once and the operation includes an implicit erase cycle for that byte (it is not necessary to set FREE). CPU execution is stalled and at the completion of the write cycle, the WR bit is cleared in hardware and the NVM Interrupt Flag bit (NVMIF) is set. The new CONFIG value takes effect when the CPU resumes operation.

TABLE 11-4: USER ID, DEV/REV ID AND CONFIGURATION WORD ACCESS (NVMREG<1:0> = x1)

Address	Function	Read Access	Write Access
20 0000h-20 000Fh	User IDs	Yes	Yes
3F FFFCh-3F FFFFh	Revision ID/Device ID	Yes	No
30 0000h-30 000Bh	Configuration Words 1-6	Yes	Yes

REGISTER 18-3: TMR0L: TIMER0 COUNT REGISTER

				-			
R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
			TMR	0<7:0>			
bit 7							bit 0
Legend:							
R = Readable	bit	W = Writable b	bit	U = Unimpler	mented bit, read	1 as '0'	
u = Bit is unch	anged	x = Bit is unkne	own	-n/n = Value a	at POR and BO	R/Value at all c	other Resets
'1' = Bit is set		'0' = Bit is clea	red				

bit 7-0 TMR0<7:0>:TMR0 Counter bits <7:0>

REGISTER 18-4: TMR0H: TIMER0 PERIOD REGISTER

| R/W-1/1 |
|---------|---------|---------|---------|---------|---------|---------|---------|
| | | | TMR0 | <15:8> | | | |
| bit 7 | | | | | | | bit 0 |

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-0 When T016BIT = 0 PR0<7:0>:TMR0 Period Register Bits <7:0> When T016BIT = 1 TMR0<15:8>: TMR0 Counter bits <15:8>

TABLE 18-1: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER0

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
TMR0L	TMR0<7:0>							226	
TMR0H				TMR0	<15:8>				226
T0CON0	T0EN	—	TOOUT	T016BIT		TOOUTPS	<3:0>		224
T0CON1		T0CS<2:0>		TOASYNC		T0CKPS<	:3:0>		225
T0CKIPPS	-	—	—		TOCK	IPPS<4:0>			216
TMR0PPS	_	—	—		TMRC)PPS<4:0>			216
INTCON	GIE/GIEH	PEIE/GIEL	IPEN	—	—	INT2EDG	INT1EDG	INT0EDG	170
PIR0	-	—	TMR0IF	IOCIF	—	INT2IF	INT1IF	INT0IF	171
PIE0	_	—	TMR0IE	IOCIE	IOCIE — INT2IE INT1IE INT0IE			179	
IPR0	_	_	TMR0IP	IOCIP	_	INT2IP	INT1IP	INT0IP	187
PMD1		TMR6MD	TMR5MD	TMR4MD	TMR3MD	TMR2MD	TMR1MD	TMR0MD	69

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

© 2015-2017 Microchip Technology Inc.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE/GIEH	PEIE/GIEL	IPEN	-		INT2EDG	INT1EDG	INT0EDG	170
PIE4	_	_	TMR6IE	TMR5IE	TMR4IE	TMR3IE	TMR2IE	TMR1IE	183
PIR4	_	—	TMR6IF	TMR5IF	TMR4IF	TMR3IF	TMR2IF	TMR1IF	175
IPR4	_	—	TMR6IP	TMR5IP	TMR4IP	TMR3IP	TMR2IP	TMR1IP	191
PMD1		TMR6MD	TMR5MD	TMR4MD	TMR3MD	TMR2MD	TMR1MD	TMR0MD	69
PR2			Tir	ner2 Module I	Period Regist	er			244*
TMR2			Holding F	Register for th	e 8-bit TMR2	Register			244*
T2CON	ON		CKPS<2:0>			OUTP	S<3:0>		262
T2CLKCON	_	—	_	_	_		CS<2:0>		264
T2RST	_	—	_	_		RSEL	.<3:0>		265
T2HLT	PSYNC	CPOL	CSYNC			MODE<4:0>			263
PR4			Tir	ner4 Module I	Period Regist	er			244*
TMR4			Holding F	Register for th	e 8-bit TMR4	Register			244*
T4CON	ON		CKPS<2:0>			OUTP	S<3:0>		262
T4CLKCON	_	_	_	_	_		CS<2:0>		264
T4RST	_	—	_	_		RSEL	.<3:0>		265
T4HLT	PSYNC	CPOL	CSYNC			MODE<4:0>			263
PR6			Tir	ner6 Module I	Period Regist	er			244*
TMR6			Holding F	Register for th	e 8-bit TMR6	Register			244*
T6CON	ON		CKPS<2:0>			OUTP	S<3:0>		262
T6CLKCON	_	—	_	_	_		CS<2:0>		264
T6RST	_	_	_	_		RSEL	.<3:0>		265
T6HLT	PSYNC	CPOL	CSYNC	MODE<4:0>					
T2INPPS	—	—	—		Т	2INPPS<4:0	>		216
T4INPPS	—	—	—		Т	4INPPS<4:0	>		216
T6INPPS	_	_	_		Т	6INPPS<4:0	>		216

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

Page provides register information.



FIGURE 21-4: SIMPLIFIED PWM BLOCK DIAGRAM



PIC18LF26/45/46K40



The SPI bus operates with a single master device and one or more slave devices. When multiple slave devices are used, an independent Slave Select connection is required from the master device to each slave device.

Figure 26-2 shows a typical connection between a master device and multiple slave devices.

The master selects only one slave at a time. Most slave devices have tri-state outputs so their output signal appears disconnected from the bus when they are not selected.

controlled through addressing. Figure 26-9 is a block diagram of the I²C interface module in Master mode.

Figure 26-10 is a diagram of the I^2C interface module

in Slave mode.

26.6 I²C Mode Overview

The Inter-Integrated Circuit (I²C) bus is a multi-master serial data communication bus. Devices communicate in a master/slave environment where the master devices initiate the communication. A slave device is

FIGURE 26-9: MSSP BLOCK DIAGRAM (I²C MASTER MODE)





TABLE 27-1: SUMMARY OF REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BAUDxCON	ABDOVF	RCIDL	_	SCKP	BRG16	—	WUE	ABDEN	395
INTCON	GIE/GIEH	PEIE/GIEL	IPEN	—	—	INT2EDG	INT1EDG	INT0EDG	170
PIE3	RC2IE	TX2IE	RC1IE	TX1IE	BCL2IE	SSP2IE	BCL1IE	SSP1IE	174
PIR3	RC2IF	TX2IF	RC1IF	TX1IF	BCL2IF	SSP2IF	BCL1IF	SSP1IF	174
IPR3	RC2IP	TX2IP	RC1IP	TX1IP	BCL2IP	SSP2IP	BCL1IP	SSP1IP	190
RCxSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	394
RxyPPS	—	_	_			RxyPPS<4:0	>		218
TXxPPS	—	—	_			TXPPS<4:0	>		216
SPxBRGH			EUSARTx	Baud Rate	Generator, H	ligh Byte			404*
SPxBRGL		EUSARTx Baud Rate Generator, Low Byte							
TXxREG	EUSARTx Transmit Register							396*	
TXxSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	393

Legend: — = unimplemented location, read as '0'. Shaded cells are not used for asynchronous transmission. * Page provides register information.

BAUD RATE	Fosc = 8.000 MHz			Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 1.000 MHz		
	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 27-5: SAMPLE BAUD RATES FOR ASYNCHRONOUS MODES (CONTINUED)

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 = 0, BRGH = 0, BRG16 = 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	—	_	_

FIGURE 27-7: AUTO-WAKE-UP BIT (WUE) TIMING DURING NORMAL OPERATION

64636466 2022-023) + 03895 1886-1886 	204.00 90290 204.00 9070 297.2867			2909.0308 U.U.U.U.U. 		andre nan VANTAAN			union and an anna anna anna anna anna anna
						· · · · · · · · · · · · · · · · · · ·			
	: : !//////////////////////////////////	: : 21 230 2630	: ////////////////////////////////////			Stevenis Millinninninnin Millinninninninninni	ise is ipeer ie 11111111111111111111	996 (S. Ş. 1111111111	CAREAP IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
FIGURE 27-8:	AUT	O-WAKE	-UP BIT	(WUE) T	IMINGS I		SLEEP		



© 2015-2017 Microchip Technology Inc.

31.3 ADC 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 31-4. 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 31-4. 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 ADC acquisition must be completed before the conversion can be started. To calculate the minimum acquisition time, Equation 31-1 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 31-1: 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:

$$TC = -CHOLD(RIC + RSS + RS) \ln(1/2047)$$

= -10pF(1k\Omega + 7k\Omega + 10k\Omega) \ln(0.0004885)
= 1.37\mus

Therefore:

$$TACQ = 2\mu s + 892ns + [(50^{\circ}C - 25^{\circ}C)(0.05\mu s/^{\circ}C)]$$

= 4.62\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.

PIC18(L)F26/45/46K40

REGISTER 31-2: ADCON1: ADC CONTROL REGISTER 1

R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0	R/W-0/0
ADPPOL	ADIPEN	ADGPOL	-	-	-	_	ADDSEN
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7 ADDPOL: Precharge Polarity bit If ADPRE>0x00:

	Action During 1st Precharge Stage							
ADPFUL	External (selected analog I/O pin)	Internal (AD sampling capacitor)						
1	Shorted to AVDD	C _{HOLD} shorted to Vss						
0	Shorted to Vss	C _{HOLD} shorted to AVDD						

Otherwise:

The bit is ignored

bit 6 ADIPEN: A/D Inverted Precharge Enable bit

If ADDSEN = 1

- 1 = The precharge and guard signals in the second conversion cycle are the opposite polarity of the first cycle
- 0 = Both Conversion cycles use the precharge and guards specified by ADPPOL and ADGPOL

Otherwise:

The bit is ignored

bit 5 ADGPOL: Guard Ring Polarity Selection bit

- 1 = ADC guard Ring outputs start as digital high during Precharge stage
- 0 = ADC guard Ring outputs start as digital low during Precharge stage

bit 4-1 Unimplemented: Read as '0'

bit 0 ADDSEN: Double-sample enable bit

- 1 = Two conversions are performed on each trigger. Data from the first conversion appears in ADPREV
- 0 = One conversion is performed for each trigger





32.11 CWG1 Auto-Shutdown Source

The output of the comparator module can be used as an auto-shutdown source for the CWG1 module. When the output of the comparator is active and the corresponding WGASxE is enabled, the CWG operation will be suspended immediately (see Section 24.10.1.2 "External Input Source").

32.12 ADC Auto-Trigger Source

The output of the comparator module can be used to trigger an ADC conversion. When the ADACT register is set to trigger on a comparator output, an ADC conversion will trigger when the Comparator output goes high.

32.13 TMR2/4/6 Reset

The output of the comparator module can be used to reset Timer2. When the TxERS register is appropriately set, the timer will reset when the Comparator output goes high.

32.14 Operation in Sleep Mode

The comparator module can operate during Sleep. The comparator clock source is based on the Timer1 clock source. If the Timer1 clock source is either the system clock (FOSC) or the instruction clock (FOSC/4), Timer1 will not operate during Sleep, and synchronized comparator outputs will not operate.

A comparator interrupt will wake the device from Sleep. The CxIE bits of the PIE2 register must be set to enable comparator interrupts.

35.2 Extended Instruction Set

In addition to the standard 75 instructions of the PIC18 instruction set, PIC18(L)F2x/4xK40 devices also provide an optional extension to the core CPU functionality. The added features include eight additional instructions that augment indirect and indexed addressing operations and the implementation of Indexed Literal Offset Addressing mode for many of the standard PIC18 instructions.

The additional features of the extended instruction set are disabled by default. To enable them, users must set the XINST Configuration bit.

The instructions in the extended set can all be classified as literal operations, which either manipulate the File Select Registers, or use them for indexed addressing. Two of the instructions, ADDFSR and SUBFSR, each have an additional special instantiation for using FSR2. These versions (ADDULNK and SUBULNK) allow for automatic return after execution.

The extended instructions are specifically implemented to optimize re-entrant program code (that is, code that is recursive or that uses a software stack) written in high-level languages, particularly C. Among other things, they allow users working in high-level languages to perform certain operations on data structures more efficiently. These include:

- dynamic allocation and deallocation of software stack space when entering and leaving subroutines
- function pointer invocation
- software Stack Pointer manipulation
- manipulation of variables located in a software stack

A summary of the instructions in the extended instruction set is provided in Table 35-3. Detailed descriptions are provided in **Section 35.2.2 "Extended Instruction Set"**. The opcode field descriptions in Table 35-1 apply to both the standard and extended PIC18 instruction sets.

Note: The instruction set extension and the Indexed Literal Offset Addressing mode were designed for optimizing applications written in C; the user may likely never use these instructions directly in assembler. The syntax for these commands is provided as a reference for users who may be reviewing code that has been generated by a compiler.

35.2.1 EXTENDED INSTRUCTION SYNTAX

Most of the extended instructions use indexed arguments, using one of the File Select Registers and some offset to specify a source or destination register. When an argument for an instruction serves as part of indexed addressing, it is enclosed in square brackets ("[]"). This is done to indicate that the argument is used as an index or offset. MPASM™ Assembler will flag an error if it determines that an index or offset value is not bracketed.

When the extended instruction set is enabled, brackets are also used to indicate index arguments in byteoriented and bit-oriented instructions. This is in addition to other changes in their syntax. For more details, see Section 35.2.3.1 "Extended Instruction Syntax with Standard PIC18 Commands".

Note: In the past, square brackets have been used to denote optional arguments in the PIC18 and earlier instruction sets. In this text and going forward, optional arguments are denoted by braces ("{ }").

Mnemonic, Operands		Description	Cyclos	16-E	Bit Instru	Status		
		Description	Cycles	MSb			LSb	Affected
ADDFSR	f, k	Add literal to FSR	1	1110	1000	ffkk	kkkk	None
ADDULNK	k	Add literal to FSR2 and return	2	1110	1000	11kk	kkkk	None
CALLW		Call subroutine using WREG	2	0000	0000	0001	0100	None
MOVSF	z _s , f _d	Move z _s (source) to 1st word	2	1110	1011	0zzz	ZZZZ	None
		f _d (destination) 2nd word		1111	ffff	ffff	ffff	
MOVSS	z _s , z _d	Move z _s (source) to 1st word	2	1110	1011	lzzz	ZZZZ	None
		z _d (destination) 2nd word		1111	xxxx	XZZZ	ZZZZ	
PUSHL	k	Store literal at FSR2,	1	1110	1010	kkkk	kkkk	None
		decrement FSR2						
SUBFSR	f, k	Subtract literal from FSR	1	1110	1001	ffkk	kkkk	None
SUBULNK	k	Subtract literal from FSR2 and	2	1110	1001	11kk	kkkk	None
		return						

TABLE 35-3: EXTENSIONS TO THE PIC18 INSTRUCTION SET

44-Lead Plastic Thin Quad Flatpack (PT) - 10x10x1.0 mm Body [TQFP]



Microchip Technology Drawing C04-076C Sheet 1 of 2