E·XFL



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	20MHz
Connectivity	I ² C, LINbus, 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	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 12x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	20-DIP (0.300", 7.62mm)
Supplier Device Package	20-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f1508-i-p

Email: info@E-XFL.COM

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

TABLE 3-9: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
Bank 3	1										
F8Ch — FE3h	_	Unimplemen	ted							_	-
FE4h	STATUS_ SHAD	_		_	_	_	Z_SHAD	DC_SHAD	C_SHAD	xxx	:uuu
FE5h	WREG_ SHAD	Working Reg	Norking Register Shadow							XXXX XXX	uuuu uuuu
FE6h	BSR_ SHAD	_	-	— — Bank Select Register Shadow					x xxxx	:u uuuu	
FE7h	PCLATH_ SHAD	_	Program Counter Latch High Register Shadow						-xxx xxx	uuuu uuuu	
FE8h	FSR0L_ SHAD	Indirect Data	Memory Add	ress 0 Low F	Pointer Shado	W				XXXX XXX	uuuu uuuu
FE9h	FSR0H_ SHAD	Indirect Data	Memory Add	ress 0 High I	Pointer Shade	w				XXXX XXX	uuuu uuuu
FEAh	FSR1L_ SHAD	Indirect Data	ndirect Data Memory Address 1 Low Pointer Shadow							XXXX XXX	uuuu uuuu
FEBh	FSR1H_ SHAD	Indirect Data	ndirect Data Memory Address 1 High Pointer Shadow						XXXX XXX	uuuu uuuu	
FECh		Unimplemen	ted								_
FEDh	STKPTR	-	—	—	Current Star	ck Pointer				1 1111	1 1111
FEEh	TOSL	Top-of-Stack	Low byte							XXXX XXXX	uuuu uuuu
FEFh	TOSH	—	Top-of-Stack	High byte						-xxx xxxx	-uuu uuuu

 Legend:
 x = unknown, u = unchanged, g = value depends on condition, - = unimplemented, r = reserved. Shaded locations are unimplemented, read as '0'.

 Note
 1:
 PIC16F1508/9 only.

 2:
 Unimplemented, read as '1'.

5.5 Fail-Safe Clock Monitor

The Fail-Safe Clock Monitor (FSCM) allows the device to continue operating should the external oscillator or external clock fail. If an oscillator mode is selected, the FSCM can detect oscillator failure any time after the Oscillator Start-up Timer (OST) has expired. When an external clock mode is selected, the FSCM can detect failure as soon as the device is released from Reset.

FSCM is enabled by setting the FCMEN bit in the Configuration Words. The FSCM is applicable to external oscillator modes (LP, XT, HS) and external clock modes (ECH, ECM, ECL, EXTRC) and the Secondary Oscillator (SOSC).





5.5.1 FAIL-SAFE DETECTION

The FSCM module detects a failed oscillator by monitoring falling clock edges and using LFINTOSC as a time base. See Figure 5-9. Detection of a failed oscillator will take 32 to 96 cycles of the LFINTOSC. Figure 5-10 shows a timing diagram of the FSCM module.

5.5.2 FAIL-SAFE OPERATION

When the external clock fails, the FSCM switches the CPU clock to an internal clock source and sets the OSFIF bit of the PIR2 register. The internal clock source is determined by the IRCF<3:0> bits in the OSCCON register.

When the OSFIF bit is set, an interrupt will be generated, if the OSFIE bit in the PIE2 register is enabled. The user's firmware in the Interrupt Service Routine (ISR) can then take steps to mitigate the problems that may arise from the failed clock.

The system clock will continue to be sourced from the internal clock source until the fail-safe condition has been cleared, see Section 5.5.3 "Fail-Safe Condition Clearing".

5.5.3 FAIL-SAFE CONDITION CLEARING

When a Fail-Safe condition exists, the user must take the following actions to clear the condition before returning to normal operation with the external source.

The next sections describe how to clear the Fail-Safe condition for specific clock selections (FOSC bits) and clock switching modes (SCS bit settings).

When a Fail-Safe condition occurs with the FOSC bits selecting external oscillator (FOSC<2:0> = HS, XT, LP) and the clock switch has been selected to run from the FOSC selection (SCS<1:0> = 00), the condition is cleared by performing the following procedure.

When SCS<1:0> = 00 (Running from FOSC selection)

SCS<1:0> = 1x:

Change the SCS bits in the OSCCON register to select the internal oscillator block. This resets the OST timer and allows it to operate again.

OSFIF = 0:

Clear the OSFIF bit in the PIR2 register.

SCS<1:0> = 00:

Change the SCS bits in the OSCCON register to select the FOSC Configuration Word clock selection. This will start the OST. The CPU will continue to operate from the internal oscillator until the OST count is reached. When OST expires, the clock module will switch to the external oscillator and the Fail-Safe condition will be cleared.

If the Fail-Safe condition still exists, the OSFIF bit will again be set by hardware.

5.5.3.2 External Clock with SCS<1:0> = 00

When a Fail-Safe condition occurs with the FOSC bits selecting external clock (FOSC<2:0> = ECH, ECM, ECL, EXTRC) and the clock switch has selected to run from the FOSC selection (SCS<1:0> = 00), the condition is cleared by performing the following procedure.

When SCS<1:0> = 00 (Running from FOSC selection)

SCS<1:0> = 1x:

Change the SCS bits in the OSCCON register to select the internal oscillator block. This resets the OST timer and allows it to operate again.

OSFIF = 0:

Clear the OSFIF bit in the PIR2 register.

5.6 Register Definitions: Oscillator Control

U-0	R/W-0/0	R/W-1/1	R/W-1/1	R/W-1/1	U-0	R/W-0/0	R/W-0/0
		IRCF	<3:0>			SCS	<1:0>
bit 7							bit 0
Legend:							
R = Readable	bit	W = Writable I	U = Unimpler	nented bit, rea	d as '0'		
u = Bit is uncha	anged	x = Bit is unkn	own	-n/n = Value a	at POR and BO	DR/Value at all o	other Resets
'1' = Bit is set		'0' = Bit is clea	ared				
bit 7	Unimplemen	ted: Read as '	כי				
bit 6-3	IRCF<3:0>: I	nternal Oscillate	or Frequency	Select bits			
	1111 = 16 M	Hz					
	1110 = 8 MH	lz					
	1101 = 4 MH	lz					
	1100 = 2 MF	lz					
	1011 = 1 MF	1Z ∠⊔–(1)					
	1010 = 3001	<п∠`′ ∠⊔ ₇ (1)					
	1001 = 2301	(12 ⁽¹⁾					
	0111 = 500	≺Hz (default up	on Reset)				
	0110 = 250	κHz					
	0101 = 125	кНz					
	0100 = 62.5	kHz					
	001x = 31.2	5 kHz					
	000x = 31 kl	Hz LF					
bit 2	Unimplemen	ted: Read as '	כ'				
bit 1-0	SCS<1:0>: S	ystem Clock Se	elect bits				
	1x = Internal	oscillator block					
	01 = Seconda	ary oscillator					
	00 = Clock de	etermined by F	OSC<2:0> in	Configuration V	Vords.		
Note 1: Dup	licate frequend	cy derived from	HFINTOSC.				

REGISTER 5-1: OSCCON: OSCILLATOR CONTROL REGISTER



8.2 Low-Power Sleep Mode

This device contains an internal Low Dropout (LDO) voltage regulator, which allows the device I/O pins to operate at voltages up to 5.5V while the internal device logic operates at a lower voltage. The LDO and its associated reference circuitry must remain active when the device is in Sleep mode.

Low-Power Sleep mode allows the user to optimize the operating current in Sleep. Low-Power Sleep mode can be selected by setting the VREGPM bit of the VREGCON register, putting the LDO and reference circuitry in a low-power state whenever the device is in Sleep.

8.2.1 SLEEP CURRENT VS. WAKE-UP TIME

In the Default Operating mode, the LDO and reference circuitry remain in the normal configuration while in Sleep. The device is able to exit Sleep mode quickly since all circuits remain active. In Low-Power Sleep mode, when waking up from Sleep, an extra delay time is required for these circuits to return to the normal configuration and stabilize.

The Low-Power Sleep mode is beneficial for applications that stay in Sleep mode for long periods of time. The Normal mode is beneficial for applications that need to wake from Sleep quickly and frequently.

8.2.2 PERIPHERAL USAGE IN SLEEP

Some peripherals that can operate in Sleep mode will not operate properly with the Low-Power Sleep mode selected. The LDO will remain in the Normal Power mode when those peripherals are enabled. The Low-Power Sleep mode is intended for use with these peripherals:

- Brown-out Reset (BOR)
- Watchdog Timer (WDT)
- External interrupt pin/Interrupt-on-change pins
- Timer1 (with external clock source)

The Complementary Waveform Generator (CWG), the Numerically Controlled Oscillator (NCO) and the Configurable Logic Cell (CLC) modules can utilize the HFINTOSC oscillator as either a clock source or as an input source. Under certain conditions, when the HFINTOSC is selected for use with the CWG, NCO or CLC modules, the HFINTOSC will remain active during Sleep. This will have a direct effect on the Sleep mode current.

Please refer to sections Section 24.5 "Operation During Sleep", 25.7 "Operation In Sleep" and 26.10 "Operation During Sleep" for more information.

The PIC16LF1508/9 does not have a con-Note: figurable Low-Power Sleep mode. PIC16LF1508/9 is an unregulated device and is always in the lowest power state when in Sleep, with no wake-up time penalty. This device has a lower maximum Vdd and I/O voltage than the PIC16F1508/9. See Section 29.0 "Electrical Specifications" for more information.

8.3 Register Definitions: Voltage Regulator Control

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-1/1
—	—	—	—	—	—	VREGPM	Reserved
bit 7							bit 0
Legend:							

REGISTER 8-1: VREGCON: VOLTAGE REGULATOR CONTROL REGISTER⁽¹⁾

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 7-2	Unimplemented: Read as '0'
bit 1	VREGPM: Voltage Regulator Powe

VREGPM: Voltage Regulator Power Mode Selection bit

- 1 = Low-Power Sleep mode enabled in Sleep⁽²⁾
 Draws lowest current in Sleep, slower wake-up
- 0 = Normal Power mode enabled in Sleep⁽²⁾
 Draws higher current in Sleep, faster wake-up
- bit 0 **Reserved:** Read as '1'. Maintain this bit set.

Note 1: PIC16F1508/9 only.

2: See Section 29.0 "Electrical Specifications".

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	75
IOCAF	—	—	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	121
IOCAN	—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	121
IOCAP	—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	121
IOCBF	IOCBF7	IOCBF6	IOCBF5	IOCBF4	—	—	—	—	122
IOCBN	IOCBN7	IOCBN6	IOCBN5	IOCBN4	_	_	—	—	122
IOCBP	IOCBP7	IOCBP6	IOCBP5	IOCBP4	_	_	—	—	122
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	—	TMR2IE	TMR1IE	76
PIE2	OSFIE	C2IE	C1IE	—	BCL1IE	NCO1IE	—	—	77
PIE3	—	—	—	—	CLC4IE	CLC3IE	CLC2IE	CLC1IE	78
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	_	TMR2IF	TMR1IF	78
PIR2	OSFIF	C2IF	C1IF	—	BCL1IF	NCO1IF	—	—	78
PIR3	—	—	—	—	CLC4IF	CLC3IF	CLC2IF	CLC1IF	81
STATUS	—	—	—	TO	PD	Z	DC	С	19
WDTCON	_	_		WDTPS<4:0>					88

TABLE 8-1: SUMMARY OF REGISTERS ASSOCIATED WITH POWER-DOWN MODE

Legend: — = unimplemented, read as '0'. Shaded cells are not used in Power-Down mode.

TABLE 14-2: SUMMARY OF REGISTERS ASSOCIATED WITH THE TEMPERATURE INDICATOR

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR>1:0>		ADFVF	२<1:0>	125

Legend: Shaded cells are unused by the temperature indicator module.



21.5.3.3 7-bit Transmission with Address Hold Enabled

Setting the AHEN bit of the SSPxCON3 register enables additional clock stretching and interrupt generation after the eighth falling edge of a received matching address. Once a matching address has been clocked in, CKP is cleared and the SSPxIF interrupt is set.

Figure 21-19 displays a standard waveform of a 7-bit Address Slave Transmission with AHEN enabled.

- 1. Bus starts idle.
- Master sends Start condition; the S bit of SSPxSTAT is set; SSPxIF is set if interrupt on Start detect is enabled.
- Master sends matching address with R/W bit set. After the eighth falling edge of the SCLx line the CKP bit is cleared and SSPxIF interrupt is generated.
- 4. Slave software clears SSPxIF.
- Slave software reads ACKTIM bit of SSPxCON3 register, and R/W and D/A of the SSPxSTAT register to determine the source of the interrupt.
- 6. Slave reads the address value from the SSPxBUF register clearing the BF bit.
- Slave software decides from this information if it wishes to ACK or not ACK and sets the ACKDT bit of the SSPxCON2 register accordingly.
- 8. Slave sets the CKP bit releasing SCLx.
- 9. Master clocks in the \overline{ACK} value from the slave.
- 10. Slave hardware automatically clears the CKP bit and sets SSPxIF after the ACK if the R/W bit is set.
- 11. Slave software clears SSPxIF.
- 12. Slave loads value to transmit to the master into SSPxBUF setting the BF bit.

Note: SSPxBUF cannot be loaded until after the ACK.

- 13. Slave sets the CKP bit, releasing the clock.
- 14. Master clocks out the data from the slave and sends an ACK value on the ninth SCLx pulse.
- 15. Slave hardware copies the ACK value into the ACKSTAT bit of the SSPxCON2 register.
- 16. Steps 10-15 are repeated for each byte transmitted to the master from the slave.
- 17. If the master sends a not ACK the slave releases the bus allowing the master to send a Stop and end the communication.

Note: Master must send a not ACK on the last byte to ensure that the slave releases the SCLx line to receive a Stop.





21.6.8 ACKNOWLEDGE SEQUENCE TIMING

An Acknowledge sequence is enabled by setting the Acknowledge Sequence Enable bit, ACKEN bit of the SSPxCON2 register. When this bit is set, the SCLx pin is pulled low and the contents of the Acknowledge data bit are presented on the SDAx pin. If the user wishes to generate an Acknowledge, then the ACKDT bit should be cleared. If not, the user should set the ACKDT bit before starting an Acknowledge sequence. The Baud Rate Generator then counts for one rollover period (TBRG) and the SCLx pin is deasserted (pulled high). When the SCLx pin is sampled high (clock arbitration), the Baud Rate Generator counts for TBRG. The SCLx pin is then pulled low. Following this, the ACKEN bit is automatically cleared, the Baud Rate Generator is turned off and the MSSP module then goes into Idle mode (Figure 21-30).

21.6.8.1 WCOL Status Flag

If the user writes the SSPxBUF when an Acknowledge sequence is in progress, then the WCOL bit is set and the contents of the buffer are unchanged (the write does not occur).

21.6.9 STOP CONDITION TIMING

A Stop bit is asserted on the SDAx pin at the end of a receive/transmit by setting the Stop Sequence Enable bit, PEN bit of the SSPxCON2 register. At the end of a receive/transmit, the SCLx line is held low after the falling edge of the ninth clock. When the PEN bit is set, the master will assert the SDAx line low. When the SDAx line is sampled low, the Baud Rate Generator is reloaded and counts down to '0'. When the Baud Rate Generator times out, the SCLx pin will be brought high and one TBRG (Baud Rate Generator rollover count) later, the SDAx pin will be deasserted. When the SDAx pin is sampled high while SCLx is high, the P bit of the SSPxSTAT register is set. A TBRG later, the PEN bit is cleared and the SSPxIF bit is set (Figure 21-31).

21.6.9.1 WCOL Status Flag

If the user writes the SSPxBUF when a Stop sequence is in progress, then the WCOL bit is set and the contents of the buffer are unchanged (the write does not occur).

FIGURE 21-30: ACKNOWLEDGE SEQUENCE WAVEFORM



REGISTER 24-9: CLCDATA: CLC DATA OUTPUT

U-0	U-0	U-0	U-0	R-0	R-0	R-0	R-0		
—	—	—	—	MLC4OUT	MLC3OUT	MLC2OUT	MLC1OUT		
bit 7							bit 0		
Legend:									
R = Readable bit		W = Writable bit		U = Unimpler	nented bit, read	as '0'			
u = Bit is uncha	anged	x = Bit is unknown		-n/n = Value at POR and BOR/Value at all other Resets					
'1' = Bit is set		'0' = Bit is clea	ared						
bit 7-4	Unimplemente	ed: Read as '0'							
bit 3	MLC4OUT: M	irror copy of LC	C4OUT bit						
bit 2	MLC3OUT: Mirror copy of LC3OUT bit								
bit 1	MLC2OUT: Mirror copy of LC2OUT bit								
bit 0	MLC1OUT: M	irror copy of LC	C1OUT bit						

LSLF	Logical Left Shift	MOVF	Move f		
Syntax:	[<i>label</i>]LSLF f{,d}	Syntax:	[<i>label</i>] MOVF f,d		
Operands:	$0 \le f \le 127$ d $\in [0,1]$	Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ d \in [0,1] \end{array}$		
Operation:	$(f < 7 >) \rightarrow C$	Operation:	$(f) \rightarrow (dest)$		
	$(f < 6:0 >) \rightarrow dest < 7:1 >$	Status Affected:	Z		
Status Affected:	C, Z	Description:	The contents of register f is moved to a destination dependent upon the		
Description:	The contents of register 'f' are shifted one bit to the left through the Carry flag. A '0' is shifted into the LSb. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f'.		status of d. If $d = 0$, destination is W register. If $d = 1$, the destination is file register f itself. $d = 1$ is useful to test a file register since status flag Z is affected.		
	C register f	Words:	1		
		Cycles:	1		
		Example:	MOVF FSR, 0		
LSRF	Logical Right Shift		After Instruction W = value in FSR register		
Syntax:	[label]LSRF f{,d}		Z = 1		

Syntax:	[<i>label</i>] LSRF f {,d}
Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ d \in [0,1] \end{array}$
Operation:	$\begin{array}{l} 0 \rightarrow dest < 7 > \\ (f < 7:1 >) \rightarrow dest < 6:0 >, \\ (f < 0 >) \rightarrow C, \end{array}$
Status Affected:	C, Z
Description:	The contents of register 'f' are shifted one bit to the right through the Carry flag. A '0' is shifted into the MSb. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f'.
	0 → register f → C

ΜΟΥΨΙ	Move W to INDFn
Syntax:	[<i>label</i>] MOVWI ++FSRn [<i>label</i>] MOVWIFSRn [<i>label</i>] MOVWI FSRn++ [<i>label</i>] MOVWI FSRn [<i>label</i>] MOVWI k[FSRn]
Operands:	n ∈ [0,1] mm ∈ [00,01,10,11] -32 ≤ k ≤ 31
Operation:	$\label{eq:W} \begin{split} W &\rightarrow INDFn \\ \text{Effective address is determined by} \\ \bullet \ FSR + 1 \ (\text{preincrement}) \\ \bullet \ FSR + 1 \ (\text{predecrement}) \\ \bullet \ FSR + k \ (\text{relative offset}) \\ \text{After the Move, the FSR value will be either:} \\ \bullet \ FSR + 1 \ (\text{all increments}) \\ \bullet \ FSR + 1 \ (\text{all increments}) \\ \text{Unchanged} \end{split}$
Status Affected:	None

Mode	Syntax	mm
Preincrement	++FSRn	00
Predecrement	FSRn	01
Postincrement	FSRn++	10
Postdecrement	FSRn	11

Description:

This instruction is used to move data between W and one of the indirect registers (INDFn). Before/after this move, the pointer (FSRn) is updated by pre/post incrementing/decrementing it.

Note: The INDFn registers are not physical registers. Any instruction that accesses an INDFn register actually accesses the register at the address specified by the FSRn.

FSRn is limited to the range 0000h -FFFFh. Incrementing/decrementing it beyond these bounds will cause it to wrap-around.

The increment/decrement operation on FSRn WILL NOT affect any Status bits.

NOP	No Operation
Syntax:	[label] NOP
Operands:	None
Operation:	No operation
Status Affected:	None
Description:	No operation.
Words:	1
Cycles:	1
Example:	NOP

OPTION	Load OPTION_REG Register with W		
Syntax:	[label] OPTION		
Operands:	None		
Operation:	$(W) \to OPTION_REG$		
Status Affected:	None		
Description:	Move data from W register to OPTION_REG register.		

RESET	Software Reset			
Syntax:	[label] RESET			
Operands:	None			
Operation:	Execute a device Reset. Resets the nRI flag of the PCON register.			
Status Affected:	None			
Description:	This instruction provides a way to execute a hardware Reset by software.			

RRF	Rotate Right f through Carry						
Syntax:	[<i>label</i>] RRF f,d						
Operands:	$0 \le f \le 127$ d $\in [0,1]$						
Operation:	See description below						
Status Affected:	С						
Description:	The contents of register 'f' are rotated one bit to the right through the Carry flag. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.						



SUBLW	Subtract W from literal						
Syntax:	[label] SU	JBLW k					
Operands:	$0 \leq k \leq 255$						
Operation:	$k - (W) \to (W$	/)					
Status Affected:	C, DC, Z	C, DC, Z					
Description:	The W register is subtracted (2's com- plement method) from the 8-bit literal 'k'. The result is placed in the W regis- ter.						
	C = 0	W > k					
	C = 1	$W \le k$					
	DC = 0 W<3:0> k<3:0>						

DC = 1

 $W<3:0> \le k<3:0>$

SLEEP	Enter Sleep mode			
Syntax:	[label] SLEEP			
Operands:	None			
Operation:	$\begin{array}{l} \text{O0h} \rightarrow \text{WDT,} \\ 0 \rightarrow \underline{\text{WDT}} \text{ prescaler,} \\ 1 \rightarrow \overline{\text{TO}}, \\ 0 \rightarrow \overline{\text{PD}} \end{array}$			
Status Affected:	TO, PD			
Description:	The power-down Status bit, PD is cleared. Time-out Status bit, TO is set. Watchdog Timer and its pres- caler are cleared. The processor is put into Sleep mode with the oscillator stopped.			

SUBWF	Subtract W from f					
Syntax:	[label] SL	JBWF f,d				
Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ d \in [0,1] \end{array}$					
Operation:	(f) - (W) \rightarrow (destination)					
Status Affected:	C, DC, Z					
Description:	Subtract (2's complement method) W register from register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f.					
	C = 0	W > f				
	C = 1	$W \leq f$				
	DC = 0	W<3:0> > f<3:0>				
	DC = 1	W<3:0> ≤ f<3:0>				

SUBWFB	Subtract W from f with Borrow					
Syntax:	SUBWFB f {,d}					
Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ d \in [0,1] \end{array}$					
Operation:	$(f) - (W) - (\overline{B}) \rightarrow dest$					
Status Affected:	C, DC, Z					
Description:	Subtract W and the BORROW flag (CARRY) from register 'f' (2's comple- ment method). If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f'.					

TABLE 29-2: SUPPLY CURRENT (IDD)^(1,2)

PIC16LF	F1508/9 Standard Operating Conditions (unle					Inless ot	herwise stated)			
PIC16F1	508/9									
Param.	Device		T 4		Unite		Conditions			
No.	Characteristics	MIN.	турт	wax.	Units	VDD	Note			
D010		-	8	20	μA	1.8	Fosc = 32 kHz,			
		—	10	25	μA	3.0	LP Oscillator, -40°C \leq TA \leq +85°C			
D010		_	15	31	μA	2.3	Fosc = 32 kHz,			
			17	33	μA	3.0	LP Oscillator, $40^{\circ}C < T_{0} < \pm 85^{\circ}C$			
		—	21	39	μA	5.0	-40 C \leq TA \leq $+85$ C			
D011		—	60	100	μA	1.8	Fosc = 1 MHz,			
		—	100	180	μA	3.0	XT Oscillator			
D011		_	100	180	μA	2.3	Fosc = 1 MHz,			
		—	130	220	μA	3.0	XT Oscillator			
		—	170	280	μA	5.0				
D012		_	140	240	μA	1.8	Fosc = 4 MHz,			
		—	250	360	μA	3.0	XT Oscillator			
D012		—	210	320	μA	2.3	Fosc = 4 MHz,			
		—	280	410	μA	3.0	XT Oscillator			
		—	340	500	μA	5.0				
D013		—	30	65	μA	1.8	Fosc = 1 MHz,			
		—	55	100	μA	3.0	External Clock (ECM), Medium Power mode			
D013		_	65	110	μA	2.3	Fosc = 1 MHz,			
		—	85	140	μA	3.0	External Clock (ECM),			
		—	115	190	μA	5.0	Medium Power mode			
D014		—	115	190	μA	1.8	Fosc = 4 MHz,			
		—	210	310	μA	3.0	External Clock (ECM), Medium Power mode			
D014		_	180	270	μA	2.3	Fosc = 4 MHz,			
		—	240	365	μA	3.0	External Clock (ECM),			
		—	295	460	μA	5.0	Medium Power mode			
D015		—	3.2	12	μA	1.8	Fosc = 31 kHz,			
		—	5.4	20	μA	3.0	LFINTOSC, -40°C \leq Ta \leq +85°C			
D015		_	13	28	μA	2.3	Fosc = 31 kHz,			
		_	15	30	μA	3.0	LFINTOSC,			
		_	17	36	μA	5.0	-40°C ≤ IA ≤ +85°C			
	T I				1		· · · · · · · · · · · · · · · · · · ·			

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: The test conditions for all IDD measurements in active operation mode are: OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to Vss; MCLR = VDD; WDT disabled.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors, such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature, also have an impact on the current consumption.

3: For RC oscillator configurations, current through REXT is not included. The current through the resistor can be extended by the formula IR = VDD/2REXT (mA) with REXT in $k\Omega$.

TABLE 29-21: I²C BUS DATA REQUIREMENTS

Standard Operating Conditions (unless otherwise stated)							
Param. No.	Symbol	Characte	Min.	Max.	Units	Conditions	
SP100*	Тнідн	Clock high time	100 kHz mode	4.0	_	μS	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	0.6		μS	Device must operate at a minimum of 10 MHz
			SSP module	1.5Tcy	_		
SP101*	TLOW	Clock low time	100 kHz mode	4.7		μS	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	1.3		μS	Device must operate at a minimum of 10 MHz
			SSP module	1.5Tcy	_		
SP102*	TR	SDA and SCL rise	100 kHz mode	—	1000	ns	
	time	400 kHz mode	20 + 0.1Св	300	ns	CB is specified to be from 10-400 pF	
SP103* TF	SDA and SCL fall time	100 kHz mode	—	250	ns		
		400 kHz mode	20 + 0.1Св	250	ns	CB is specified to be from 10-400 pF	
SP106*	THD:DAT	Data input hold time	100 kHz mode	0		ns	
			400 kHz mode	0	0.9	μS	
SP107*	TSU:DAT	Data input setup time	100 kHz mode	250	_	ns	(Note 2)
			400 kHz mode	100		ns	
SP109*	ΤΑΑ	Output valid from clock	100 kHz mode	—	3500	ns	(Note 1)
			400 kHz mode	—	_	ns	
SP110*	TBUF	Bus free time	100 kHz mode	4.7	_	μS	Time the bus must be free
			400 kHz mode	1.3	_	μS	before a new transmission can start
SP111	Св	Bus capacitive loading		_	400	pF	

* These parameters are characterized but not tested.

Note 1: As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of Start or Stop conditions.

2: A Fast mode (400 kHz) I²C bus device can be used in a Standard mode (100 kHz) I²C bus system, but the requirement TSU:DAT \geq 250 ns must then be met. This will automatically be the case if the device does not stretch the low period of the SCL signal. If such a device does stretch the low period of the SCL signal, it must output the next data bit to the SDA line TR max. + TSU:DAT = 1000 + 250 = 1250 ns (according to the Standard mode I²C bus specification), before the SCL line is released.

FIGURE 30-15: IDD TYPICAL, EXTERNAL CLOCK (ECH), HIGH-POWER MODE, PIC16LF1508/9 ONLY



FIGURE 30-16: IDD MAXIMUM, EXTERNAL CLOCK (ECH), HIGH-POWER MODE, PIC16LF1508/9 ONLY













FIGURE 30-74: SLEEP MODE, WAKE PERIOD WITH HFINTOSC SOURCE, PIC16LF1508/9 ONLY

31.2 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

31.3 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

31.4 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

31.5 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