



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	-
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	17
Program Memory Size	3.5KB (2K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°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/pic16lf1507-e-ss

Email: info@E-XFL.COM

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

FIGURE 3-1:

PROGRAM MEMORY MAP AND STACK FOR PIC16(L)F1507

Rev. 10-000040C 7/30/2013 PC<14:0> CALL, CALLW 15 RETURN, RETLW Interrupt, RETFIE Stack Level 0 Stack Level 1 Stack Level 15 0000h **Reset Vector** ÷ Interrupt Vector 0004h On-chip 0005h Page 0 Program 07FFh Memory 0800h Rollover to Page 0 Rollover to Page 0 7FFFh

3.2.1 READING PROGRAM MEMORY AS DATA

There are two methods of accessing constants in program memory. The first method is to use tables of RETLW instructions. The second method is to set an FSR to point to the program memory.

3.2.1.1 RETLW Instruction

EXAMPLE 3-1

The RETLW instruction can be used to provide access to tables of constants. The recommended way to create such a table is shown in Example 3-1.

RETLW INSTRUCTION

constants	
BRW	;Add Index in W to
	;program counter to
	;select data
RETLW DATA0	;Index0 data
RETLW DATA1	;Index1 data
RETLW DATA2	
RETLW DATA3	
my_function	
; LOTS OF CODE	
MOVLW DATA_	INDEX
call constants	
; THE CONSTANT	IS IN W

The BRW instruction makes this type of table very simple to implement. If your code must remain portable with previous generations of microcontrollers, then the BRW instruction is not available so the older table read method must be used.

4.2 Register Definitions: Configuration Words

R/P-1 R/P-1 U-1 U-1 **R/P-1** U-1 BOREN<1:0>(1) CLKOUTEN ____ _ _____ bit 13 bit 8 R/P-1 R/P-1 R/P-1 **R/P-1** R/P-1 U-1 R/P-1 R/P-1 CP(2) PWRTE MCLRE WDTE<1:0> FOSC<1:0> bit 7 bit 0 Legend: R = Readable bit P = Programmable bit U = Unimplemented bit, read as '1' -n = Value when blank or after Bulk Erase 0' = Bit is cleared '1' = Bit is set bit 13-12 Unimplemented: Read as '1' bit 11 **CLKOUTEN:** Clock Out Enable bit 1 = CLKOUT function is disabled. I/O function on the CLKOUT pin 0 = CLKOUT function is enabled on the CLKOUT pin BOREN<1:0>: Brown-Out Reset Enable bits⁽¹⁾ bit 10-9 11 = BOR enabled 10 = BOR enabled during operation and disabled in Sleep 01 = BOR controlled by SBOREN bit of the BORCON register 00 = BOR disabled Unimplemented: Read as '1' bit 8 CP: Code Protection bit⁽²⁾ bit 7 1 = Program memory code protection is disabled 0 = Program memory code protection is enabled MCLRE: MCLR/VPP Pin Function Select bit bit 6 If LVP bit = 1: This bit is ignored. If LVP bit = 0: $1 = \overline{MCLR}/VPP$ pin function is \overline{MCLR} ; Weak pull-up enabled. 0 = MCLR/VPP pin function is digital input; MCLR internally disabled; Weak pull-up under control of WPUA3 bit. **PWRTE:** Power-Up Timer Enable bit bit 5 1 = PWRT disabled 0 = PWRT enabled WDTE<1:0>: Watchdog Timer Enable bits bit 4-3 11 = WDT enabled 10 = WDT enabled while running and disabled in Sleep 01 = WDT controlled by the SWDTEN bit in the WDTCON register 00 = WDT disabled bit 2 Unimplemented: Read as '1' bit 1-0 FOSC<1:0>: Oscillator Selection bits 11 = ECH: External Clock, High-Power mode: on CLKIN pin 10 = ECM: External Clock, Medium Power mode: on CLKIN pin 01 = ECL: External Clock, Low-Power mode: on CLKIN pin 00 = INTOSC oscillator: I/O function on CLKIN pin

REGISTER 4-1: CONFIG1: CONFIGURATION WORD 1

Note 1: Enabling Brown-out Reset does not automatically enable Power-up Timer.

2: Once enabled, code-protect can only be disabled by bulk erasing the device.

		R/P-1	U-1	R/P-1	R/P-1	R/P-1	U-1
		LVP ⁽¹⁾	_	LPBOR	BORV ⁽²⁾	STVREN	_
		bit 13		·			bit 8
U-1	U-1	U-1	U-1	U-1	U-1	R/P-1	R/P-1
—	—	_	—	—	—	WRT<	:1:0>
bit 7							bit 0
Legend:							
R = Read	lable bit	P = Programr	nable bit	U = Unimplen	nented bit, read	l as '1'	
'0' = Bit is	cleared	'1' = Bit is set		-n = Value wh	en blank or aft	er Bulk Erase	
bit 12 bit 11 bit 10 bit 9	1 = Low-volta 0 = High-volta Unimplemen LPBOR: Low 1 = Low-Powe 0 = Low-Powe BORV: Brown 1 = Brown-ou 0 = Brown-ou STVREN: State	 Lvr: Low-voltage Programming Enable bit¹⁷ 1 = Low-voltage programming enabled 0 = High-voltage on MCLR must be used for programming Unimplemented: Read as '1' LPBOR: Low-Power BOR Enable bit 1 = Low-Power Brown-out Reset is disabled 0 = Low-Power Brown-out Reset is enabled BORV: Brown-Out Reset Voltage Selection bit⁽²⁾ 1 = Brown-out Reset voltage (VBOR), low trip point selected 0 = Brown-out Reset voltage (VBOR), high trip point selected STVREN: Stack Overflow/Underflow Reset Enable bit 					
	1 = Stack Ove 0 = Stack Ove	erflow or Under erflow or Under	flow will cause	a Reset ause a Reset			
bit 8-2	Unimplemen	ted: Read as '	1'				
bit 1-0	WRT<1:0>: F 2 kW Flash m 11 = Wr 10 = 000 01 = 000 00 = 000	Elash Memory S <u>eemory (PIC16)</u> ite protection o Dh to 1FFh writ Dh to 3FFh writ Dh to 7FFh writ	Self-Write Prote (<u>L)F1507 only</u>) ff e-protected, 2 e-protected, 4 e-protected, n	ection bits : 00h to 7FFh ma 00h to 7FFh ma o addresses ma	ay be modified ay be modified ay be modified		
Note 1:	The LVP bit canne	ot be programr	ned to '0' whe	n Programming	mode is enter	ed via LVP.	

REGISTER 4-2: CONFIG2: CONFIGURATION WORD 2

2: See VBOR parameter for specific trip point voltages.

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0
_		_	_	_		CLC2IE	CLC1IE
bit 7							bit 0
Legend:							
R = Readable	e bit	W = Writable	bit	U = Unimpler	nented bit, read	as '0'	
u = Bit is unc	hanged	x = Bit is unkr	nown	-n/n = Value a	at POR and BO	R/Value at all o	ther Resets
'1' = Bit is set		'0' = Bit is clea	ared				
bit 7-2	Unimplemen	ted: Read as '	כ'				
bit 1	CLC2IE: Con	figurable Logic	Block 2 Interr	upt Enable bit			
	1 = Enables f	the CLC 2 inter	rupt				
	0 = Disables	the CLC 2 inte	rrupt				
bit 0 CLC1IE: Configurable Logic Block 1 Interrupt Enable bit							
1 = Enables the CLC 1 interrupt							
	0 = Disables	the CLC 1 inte	rrupt				
Note: Bi	t PEIE of the IN	TCON register	must be				

REGISTER 7-4:	PIE3: PERIPHERAL INTERRUPT ENABLE REGISTER 3
---------------	--

set to enable any peripheral interrupt.

8.0 POWER-DOWN MODE (SLEEP)

The Power-down mode is entered by executing a **SLEEP** instruction.

Upon entering Sleep mode, the following conditions exist:

- 1. WDT will be cleared but keeps running, if enabled for operation during Sleep.
- 2. PD bit of the STATUS register is cleared.
- 3. $\overline{\text{TO}}$ bit of the STATUS register is set.
- 4. CPU clock is disabled.
- 5. 31 kHz LFINTOSC is unaffected and peripherals that operate from it may continue operation in Sleep.
- 6. Timer1 and peripherals that operate from Timer1 continue operation in Sleep when the Timer1 clock source selected is:
 - LFINTOSC
 - T1CKI
- 7. ADC is unaffected, if the dedicated FRC oscillator is selected.
- 8. I/O ports maintain the status they had before SLEEP was executed (driving high, low or highimpedance).
- 9. Resets other than WDT are not affected by Sleep mode.

Refer to individual chapters for more details on peripheral operation during Sleep.

To minimize current consumption, the following conditions should be considered:

- · I/O pins should not be floating
- External circuitry sinking current from I/O pins
- Internal circuitry sourcing current from I/O pins
- Current draw from pins with internal weak pull-ups
- Modules using 31 kHz LFINTOSC
- CWG, NCO and CLC modules using HFINTOSC

I/O pins that are high-impedance inputs should be pulled to VDD or Vss externally to avoid switching currents caused by floating inputs.

Examples of internal circuitry that might be sourcing current include the FVR module. See **Section 13.0 "Fixed Voltage Reference (FVR)"** for more information on this module.

8.1 Wake-up from Sleep

The device can wake-up from Sleep through one of the following events:

- 1. External Reset input on MCLR pin, if enabled
- 2. BOR Reset, if enabled
- 3. POR Reset
- 4. Watchdog Timer, if enabled
- 5. Any external interrupt
- 6. Interrupts by peripherals capable of running during Sleep (see individual peripheral for more information)

The first three events will cause a device Reset. The last three events are considered a continuation of program execution. To determine whether a device Reset or wake-up event occurred, refer to Section 6.12 "Determining the Cause of a Reset".

When the SLEEP instruction is being executed, the next instruction (PC + 1) is prefetched. For the device to wake-up through an interrupt event, the corresponding interrupt enable bit must be enabled. Wake-up will occur regardless of the state of the GIE bit. If the GIE bit is disabled, the device continues execution at the instruction after the SLEEP instruction. If the GIE bit is enabled, the device executes the instruction after the SLEEP instruction, the device will then call the Interrupt Service Routine. In cases where the execution of the instruction following SLEEP is not desirable, the user should have a NOP after the SLEEP instruction.

The WDT is cleared when the device wakes up from Sleep, regardless of the source of wake-up.

8.1.1 WAKE-UP USING INTERRUPTS

When global interrupts are disabled (GIE cleared) and any interrupt source has both its interrupt enable bit and interrupt flag bit set, one of the following will occur:

- If the interrupt occurs **before** the execution of a SLEEP instruction
 - SLEEP instruction will execute as a NOP.
 - WDT and WDT prescaler will not be cleared
 - TO bit of the STATUS register will not be set
 - PD bit of the STATUS register will not be cleared.
- If the interrupt occurs **during or after** the execution of a SLEEP instruction
 - SLEEP instruction will be completely executed
 - Device will immediately wake-up from Sleep
 - WDT and WDT prescaler will be cleared
 - TO bit of the STATUS register will be set
 - PD bit of the STATUS register will be cleared

Even if the flag bits were checked before executing a SLEEP instruction, it may be possible for flag bits to become set before the SLEEP instruction completes. To determine whether a SLEEP instruction executed, test the PD bit. If the PD bit is set, the SLEEP instruction was executed as a NOP.

9.0 WATCHDOG TIMER (WDT)

The Watchdog Timer is a system timer that generates a Reset if the firmware does not issue a CLRWDT instruction within the time-out period. The Watchdog Timer is typically used to recover the system from unexpected events.

The WDT has the following features:

- Independent clock source
- Multiple operating modes
 - WDT is always on
 - WDT is off when in Sleep
 - WDT is controlled by software
 - WDT is always off
- Configurable time-out period is from 1 ms to 256 seconds (nominal)
- Multiple Reset conditions
- Operation during Sleep





10.2.4 WRITING TO FLASH PROGRAM MEMORY

Program memory is programmed using the following steps:

- 1. Load the address in PMADRH:PMADRL of the row to be programmed.
- 2. Load each write latch with data.
- 3. Initiate a programming operation.
- 4. Repeat steps 1 through 3 until all data is written.

Before writing to program memory, the word(s) to be written must be erased or previously unwritten. Program memory can only be erased one row at a time. No automatic erase occurs upon the initiation of the write.

Program memory can be written one or more words at a time. The maximum number of words written at one time is equal to the number of write latches. See Figure 10-5 (row writes to program memory with 16 write latches) for more details.

The write latches are aligned to the Flash row address boundary defined by the upper 10-bits of PMADRH:PMADRL, (PMADRH<6:0>:PMADRL<7:5>) with the lower five bits of PMADRL, (PMADRL<7:0>) determining the write latch being loaded. Write operations do not cross these boundaries. At the completion of a program memory write operation, the data in the write latches is reset to contain 0x3FFF. The following steps should be completed to load the write latches and program a row of program memory. These steps are divided into two parts. First, each write latch is loaded with data from the PMDATH:PMDATL using the unlock sequence with LWLO = 1. When the last word to be loaded into the write latch is ready, the LWLO bit is cleared and the unlock sequence executed. This initiates the programming operation, writing all the latches into Flash program memory.

Note:	The special unlock sequence is required
	to load a write latch with data or initiate a
	Flash programming operation. If the
	unlock sequence is interrupted, writing to
	the latches or program memory will not be
	initiated.

- 1. Set the WREN bit of the PMCON1 register.
- 2. Clear the CFGS bit of the PMCON1 register.
- 3. Set the LWLO bit of the PMCON1 register. When the LWLO bit of the PMCON1 register is '1', the write sequence will only load the write latches and will not initiate the write to Flash program memory.
- 4. Load the PMADRH:PMADRL register pair with the address of the location to be written.
- 5. Load the PMDATH:PMDATL register pair with the program memory data to be written.
- Execute the unlock sequence (Section 10.2.2 "Flash Memory Unlock Sequence"). The write latch is now loaded.
- 7. Increment the PMADRH:PMADRL register pair to point to the next location.
- 8. Repeat steps 5 through 7 until all but the last write latch has been loaded.
- Clear the LWLO bit of the PMCON1 register. When the LWLO bit of the PMCON1 register is '0', the write sequence will initiate the write to Flash program memory.
- 10. Load the PMDATH:PMDATL register pair with the program memory data to be written.
- Execute the unlock sequence (Section 10.2.2 "Flash Memory Unlock Sequence"). The entire program memory latch content is now written to Flash program memory.
- Note: The program memory write latches are reset to the blank state (0x3FFF) at the completion of every write or erase operation. As a result, it is not necessary to load all the program memory write latches. Unloaded latches will remain in the blank state.

An example of the complete write sequence is shown in Example 10-3. The initial address is loaded into the PMADRH:PMADRL register pair; the data is loaded using indirect addressing.

REGISTER 12-4: IOCBP: INTERRUPT-ON-CHANGE PORTB POSITIVE EDGE REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
IOCBP7	IOCBP6	IOCBP5	IOCBP4	_	—	_	—
bit 7							bit 0
Legend:							
R = Readable bi	t	W = Writable bi	it	U = Unimplem	ented 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 clear	ed				

IOCBP<7:4>: Interrupt-on-Change PORTB Positive Edge Enable bits
 1 = Interrupt-on-Change enabled on the pin for a positive going edge. IOCBFx bit and IOCIF flag will be set upon detecting an edge.

0 = Interrupt-on-Change disabled for the associated pin.

bit 3-0 Unimplemented: Read as '0'

bit 7-4

REGISTER 12-5: IOCBN: INTERRUPT-ON-CHANGE PORTB NEGATIVE EDGE REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
IOCBN7	IOCBN6	IOCBN5	IOCBN4	—	—	—	—
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 7-4
 IOCBN<7:4>: Interrupt-on-Change PORTB Negative Edge Enable bits
 1 = Interrupt-on-Change enabled on the pin for a negative going edge. IOCBFx bit and IOCIF flag will be set upon detecting an edge.
 0 = Interrupt-on-Change disabled for the associated pin.
 bit 3-0
 Unimplemented: Read as '0'

REGISTER 12-6: IOCBF: INTERRUPT-ON-CHANGE PORTB FLAG REGISTER

R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	U-0	U-0	U-0	U-0
IOCBF7	IOCBF6	IOCBF5	IOCBF4	—	—	—	—
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	HS - Bit is set in hardware

bit 7-4	 IOCBF<7:4>: Interrupt-on-Change PORTB Flag bits 1 = An enabled change was detected on the associated pin. Set when IOCBPx = 1 and a rising edge was detected on RBx, or when IOCBNx = 1 and a falling edge was detected on RBx. 0 = No change was detected, or the user cleared the detected change.
bit 3-0	Unimplemented: Read as '0'

16.1.3 SOFTWARE PROGRAMMABLE PRESCALER

A software programmable prescaler is available for exclusive use with Timer0. The prescaler is enabled by clearing the PSA bit of the OPTION_REG register.

Note:	The Watchdog Timer (WDT) uses its own
	independent prescaler.

There are eight prescaler options for the Timer0 module ranging from 1:2 to 1:256. The prescale values are selectable via the PS<2:0> bits of the OPTION_REG register. In order to have a 1:1 prescaler value for the Timer0 module, the prescaler must be disabled by setting the PSA bit of the OPTION_REG register.

The prescaler is not readable or writable. All instructions writing to the TMR0 register will clear the prescaler.

16.1.4 TIMER0 INTERRUPT

Timer0 will generate an interrupt when the TMR0 register overflows from FFh to 00h. The TMR0IF interrupt flag bit of the INTCON register is set every time the TMR0 register overflows, regardless of whether or not the Timer0 interrupt is enabled. The TMR0IF bit can only be cleared in software. The Timer0 interrupt enable is the TMR0IE bit of the INTCON register.

Note:	The Timer0 interrupt cannot wake the
	processor from Sleep since the timer is
	frozen during Sleep.

16.1.5 8-BIT COUNTER MODE SYNCHRONIZATION

When in 8-Bit Counter mode, the incrementing edge on the T0CKI pin must be synchronized to the instruction clock. Synchronization can be accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the instruction clock. The high and low periods of the external clocking source must meet the timing requirements as shown in **Section 25.0 "Electrical Specifications"**.

16.1.6 OPERATION DURING SLEEP

Timer0 cannot operate while the processor is in Sleep mode. The contents of the TMR0 register will remain unchanged while the processor is in Sleep mode.

17.3 Timer1 Prescaler

Timer1 has four prescaler options allowing 1, 2, 4 or 8 divisions of the clock input. The T1CKPS bits of the T1CON register control the prescale counter. The prescale counter is not directly readable or writable; however, the prescaler counter is cleared upon a write to TMR1H or TMR1L.

17.4 Timer1 Operation in Asynchronous Counter Mode

If control bit T1SYNC of the T1CON register is set, the external clock input is not synchronized. The timer increments asynchronously to the internal phase clocks. If the external clock source is selected then the timer will continue to run during Sleep and can generate an interrupt on overflow, which will wake-up the processor. However, special precautions in software are needed to read/write the timer (see Section 17.4.1 "Reading and Writing Timer1 in Asynchronous Counter Mode").

Note: When switching from synchronous to asynchronous operation, it is possible to skip an increment. When switching from asynchronous to synchronous operation, it is possible to produce an additional increment.

17.4.1 READING AND WRITING TIMER1 IN ASYNCHRONOUS COUNTER MODE

Reading TMR1H or TMR1L while the timer is running from an external asynchronous clock will ensure a valid read (taken care of in hardware). However, the user should keep in mind that reading the 16-bit timer in two 8-bit values itself, poses certain problems, since the timer may overflow between the reads.

For writes, it is recommended that the user simply stop the timer and write the desired values. A write contention may occur by writing to the timer registers, while the register is incrementing. This may produce an unpredictable value in the TMR1H:TMR1L register pair.

17.5 Timer1 Gate

Timer1 can be configured to count freely or the count can be enabled and disabled using Timer1 gate circuitry. This is also referred to as Timer1 Gate Enable.

Timer1 gate can also be driven by multiple selectable sources.

17.5.1 TIMER1 GATE ENABLE

The Timer1 Gate Enable mode is enabled by setting the TMR1GE bit of the T1GCON register. The polarity of the Timer1 Gate Enable mode is configured using the T1GPOL bit of the T1GCON register. When Timer1 Gate Enable mode is enabled, Timer1 will increment on the rising edge of the Timer1 clock source. When Timer1 Gate Enable mode is disabled, no incrementing will occur and Timer1 will hold the current count. See Figure 17-3 for timing details.

TABLE 17-3: TIMER1 GATE ENABLE SELECTIONS

T1CLK	T1GPOL	T1G Timer1 Operation		
\uparrow	0	0	Counts	
\uparrow	0	1	Holds Count	
\uparrow	1	0	Holds Count	
\uparrow	1	1	Counts	

17.5.2 TIMER1 GATE SOURCE SELECTION

Timer1 gate source selections are shown in Table 17-4. Source selection is controlled by the T1GSS<1:0> bits of the T1GCON register. The polarity for each available source is also selectable. Polarity selection is controlled by the T1GPOL bit of the T1GCON register.

TABLE 17-4: TIMER1 GATE SOURCES

T1GSS	Timer1 Gate Source
0	Timer1 Gate pin (T1G)
1	Overflow of Timer0 (T0_overflow) (TMR0 increments from FFh to 00h)

Note 1: Optionally synchronized comparator output.

REGISTER 21-6: NCOxINCL: NCOx INCREMENT REGISTER – LOW BYTE⁽¹⁾

R/W-0/0	R/W-1/1						
			NCOxIN	C<7:0>			
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 NCOxINC<7:0>: NCOx Increment, Low Byte

Note 1: Write the NCOxINCH register first, then the NCOxINCL register. See 21.1.4 "Increment Registers" for more information.

REGISTER 21-7: NCOxINCH: NCOx INCREMENT REGISTER – HIGH BYTE⁽¹⁾

| R/W-0/0 |
|---------|---------|---------|---------|---------|---------|---------|---------|
| | | | NCOxIN | C<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 NCOxINC<15:8>: NCOx Increment, High Byte

Note 1: Write the NCOxINCH register first, then the NCOxINCL register. See 21.1.4 "Increment Registers" for more information.

TABLE 21-1: SUMMARY OF REGISTERS ASSOCIATED WITH NCOX

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
APFCON	-	-	-	—	-	—	CLC1SEL	NCO1SEL	96
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	64
NCO1ACCH				NCO1AC	C<15:8>				175
NCO1ACCL				NCO1A	CC<7:0>				175
NCO1ACCU	_					NCO1AC	C<19:16>		175
NCO1CLK	N1PWS<2:0> —			—	—	_	- N1CKS<1:0>		174
NCO1CON	N1EN	N10E	N1OUT	N1POL	—	_	_	N1PFM	174
NCO1INCH		NCO1INC<15:8>						176	
NCO1INCL	NCO1IN			NC<7:0>				176	
PIE2	_	_	_	—	—	NCO1IE	—	—	66
PIR2	_	_	_		—	NCO1IF	—	—	69
TRISA	_	_	TRISA5	TRISA4	(1)	TRISA2	TRISA1	TRISA0	98
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	106
Legend: x = unknown, u = unchanged, - = unimplemented read as '0', q = value depends on condition. Shaded cells are not used for NCOx									

module. Note 1: Unimplemented, read as '1'.

22.10 Operation During Sleep

The CWG module operates independently from the system clock and will continue to run during Sleep, provided that the clock and input sources selected remain active.

The HFINTOSC remains active during Sleep, provided that the CWG module is enabled, the input source is active, and the HFINTOSC is selected as the clock source, regardless of the system clock source selected.

In other words, if the HFINTOSC is simultaneously selected as the system clock and the CWG clock source, when the CWG is enabled and the input source is active, the CPU will go idle during Sleep, but the CWG will continue to operate and the HFINTOSC will remain active.

This will have a direct effect on the Sleep mode current.

22.11 Configuring the CWG

The following steps illustrate how to properly configure the CWG to ensure a synchronous start:

- Ensure that the TRIS control bits corresponding to CWGxA and CWGxB are set so that both are configured as inputs.
- 2. Clear the GxEN bit, if not already cleared.
- 3. Set desired dead-band times with the CWGxDBR and CWGxDBF registers.
- 4. Setup the following controls in CWGxCON2 auto-shutdown register:
 - · Select desired shutdown source.
 - Select both output overrides to the desired levels (this is necessary even if not using auto-shutdown because start-up will be from a shutdown state).
 - Set the GxASE bit and clear the GxARSEN bit.
- 5. Select the desired input source using the CWGxCON1 register.
- 6. Configure the following controls in CWGxCON0 register:
 - · Select desired clock source.
 - Select the desired output polarities.
 - Set the output enables for the outputs to be used.
- 7. Set the GxEN bit.
- Clear TRIS control bits corresponding to CWGxA and CWGxB to be used to configure those pins as outputs.
- If auto-restart is to be used, set the GxARSEN bit and the GxASE bit will be cleared automatically. Otherwise, clear the GxASE bit to start the CWG.

22.11.1 PIN OVERRIDE LEVELS

The levels driven to the output pins, while the shutdown input is true, are controlled by the GxASDLA and GxASDLB bits of the CWGxCON1 register (Register 22-3). GxASDLA controls the CWG1A override level and GxASDLB controls the CWG1B override level. The control bit logic level corresponds to the output logic drive level while in the shutdown state. The polarity control does not apply to the override level.

22.11.2 AUTO-SHUTDOWN RESTART

After an auto-shutdown event has occurred, there are two ways to have resume operation:

- · Software controlled
- Auto-restart

The restart method is selected with the GxARSEN bit of the CWGxCON2 register. Waveforms of software controlled and automatic restarts are shown in Figure 22-5 and Figure 22-6.

22.11.2.1 Software Controlled Restart

When the GxARSEN bit of the CWGxCON2 register is cleared, the CWG must be restarted after an auto-shut-down event by software.

Clearing the shutdown state requires all selected shutdown inputs to be low, otherwise the GxASE bit will remain set. The overrides will remain in effect until the first rising edge event after the GxASE bit is cleared. The CWG will then resume operation.

22.11.2.2 Auto-Restart

When the GxARSEN bit of the CWGxCON2 register is set, the CWG will restart from the auto-shutdown state automatically.

The GxASE bit will clear automatically when all shutdown sources go low. The overrides will remain in effect until the first rising edge event after the GxASE bit is cleared. The CWG will then resume operation.

SWAPF	Swap Nibbles in f				
Syntax:	[label] SWAPF f,d				
Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ d \in [0,1] \end{array}$				
Operation:	$(f<3:0>) \rightarrow (destination<7:4>),$ $(f<7:4>) \rightarrow (destination<3:0>)$				
Status Affected:	None				
Description:	The upper and lower nibbles of regis- ter 'f' are exchanged. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed in register 'f'.				

XORLW	Exclusive OR literal with W				
Syntax:	[<i>label</i>] XORLW k				
Operands:	$0 \leq k \leq 255$				
Operation:	(W) .XOR. $k \rightarrow (W)$				
Status Affected:	Z				
Description:	The contents of the W register are XOR'ed with the 8-bit literal 'k'. The result is placed in the W register.				

TRIS	Load TRIS Register with W			
Syntax:	[label] TRIS f			
Operands:	$5 \le f \le 7$			
Operation:	(W) \rightarrow TRIS register 'f'			
Status Affected:	None			
Description:	Move data from W register to TRIS register. When 'f' = 5, TRISA is loaded. When 'f' = 6, TRISB is loaded. When 'f' = 7, TRISC is loaded.			

XORWF	Exclusive OR W with f					
Syntax:	[<i>label</i>] XORWF f,d					
Operands:	$0 \le f \le 127$ $d \in [0,1]$					
Operation:	(W) .XOR. (f) \rightarrow (destination)					
Status Affected:	Z					
Description:	Exclusive OR the contents of the W register with 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'.					

TABLE 25-2: SUPPLY CURRENT (IDD)^(1,2)

PIC16LF1507		Standard Operating Conditions (unless otherwise stated)						
PIC16F1	507							
Param. No.	Device Characteristics	Min.	Тур†	Max.	Units	Conditions		
						VDD	Note	
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), Modium Rower mode	
		—	115	190	μA	5.0		
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, LFINTOSC, -40°C ≤ TA ≤ +85°C	
		—	5.4	20	μA	3.0		
D015		_	13	28	μA	2.3	Fosc = 31 kHz,	
			15	30	μA	3.0	LFINTOSC,	
		—	17	36	μA	5.0	-40 C \sec 1A \sec +85 C	
D016		—	215	360	μA	1.8	Fosc = 500 kHz,	
		—	275	480	μA	3.0	HFINTOSC	
D016			270	450	μA	2.3	Fosc = 500 kHz,	
			300	500	μA	3.0	HFINTOSC	
		—	350	620	μA	5.0		
D017*		_	410	660	μA	1.8	Fosc = 8 MHz,	
		—	630	970	μA	3.0	HFINTOSC	
D017*			530	750	μA	2.3	Fosc = 8 MHz,	
			660	1100	μA	3.0	HFINTOSC	
		—	730	1200	μA	5.0		
D018			600	940	μA	1.8	Fosc = 16 MHz, HFINTOSC	
		_	970	1400	μA	3.0		
D018			780	1200	μA	2.3	Fosc = 16 MHz,	
		_	1000	1550	μA	3.0	HFINTOSC	
		_	1090	1700	μA	5.0		

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: 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.









26.0 DC AND AC CHARACTERISTICS GRAPHS AND CHARTS

The graphs and tables provided in this section are for design guidance and are not tested.

In some graphs or tables, the data presented are **outside specified operating range** (i.e., outside specified VDD range). This is for **information only** and devices are ensured to operate properly only within the specified range.

Note: The graphs and tables provided following this note are a statistical summary based on a limited number of samples and are provided for informational purposes only. The performance characteristics listed herein are not tested or guaranteed. In some graphs or tables, the data presented may be outside the specified operating range (e.g., outside specified power supply range) and therefore, outside the warranted range.

"Typical" represents the mean of the distribution at 25°C. "MAXIMUM", "Max.", "MINIMUM" or "Min." represents (mean + 3σ) or (mean - 3σ) respectively, where σ is a standard deviation, over each temperature range.

FIGURE 26-3: IDD, EXTERNAL CLOCK (ECL), LOW-POWER MODE, Fosc = 500 kHz, PIC16LF1507 ONLY



FIGURE 26-4: IDD, EXTERNAL CLOCK (ECL), LOW-POWER MODE, Fosc = 500 kHz, PIC16F1507 ONLY



















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