



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	32MHz
Connectivity	I ² C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	15
Program Memory Size	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	1K 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	28-VQFN Exposed Pad
Supplier Device Package	28-QFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf1847-e-ml

Email: info@E-XFL.COM

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

Name	Function	Input Type	Output Type	Description
RA0/AN0/CPS0/C12IN0-/SDO2	RA0	TTL	CMOS	General purpose I/O.
	AN0	AN	—	ADC Channel 0 input.
	CPS0	AN	—	Capacitive sensing input 0.
	C12IN0-	AN	_	Comparator C1 or C2 negative input.
	SDO2		CMOS	SPI data output.
RA1/AN1/CPS1/C12IN1-/SS2	RA1	TTL	CMOS	General purpose I/O.
	AN1	AN	_	ADC Channel 1 input.
	CPS1	AN	_	Capacitive sensing input 1.
	C12IN1-	AN	—	Comparator C1 or C2 negative input.
	SS2	ST	_	Slave Select input 2.
RA2/AN2/CPS2/C12IN2-/	RA2	TTL	CMOS	General purpose I/O.
C12IN+/VREF-/DACOUT	AN2	AN	_	ADC Channel 2 input.
	CPS2	AN	—	Capacitive sensing input 2.
	C12IN2-	AN	—	Comparator C1 or C2 negative input.
	C12IN+	AN	—	Comparator C1 or C2 positive input.
	VREF-	AN	—	ADC Negative Voltage Reference input.
	DACOUT	—	AN	Voltage Reference output.
RA3/AN3/CPS3/C12IN3-/C1IN+/	RA3	TTL	CMOS	General purpose I/O.
VREF+/C1OUT/CCP3/SRQ	AN3	AN	—	ADC Channel 3 input.
	CPS3	AN	—	Capacitive sensing input 3.
	C12IN3-	AN	—	Comparator C1 or C2 negative input.
	C1IN+	AN	—	Comparator C1 positive input.
	VREF+	AN	—	ADC Voltage Reference input.
	C10UT	—	CMOS	Comparator C1 output.
	CCP3	ST	CMOS	Capture/Compare/PWM3.
	SRQ	—	CMOS	SR latch non-inverting output.
RA4/AN4/CPS4/C2OUT/T0CKI/C	RA4	TTL	CMOS	General purpose I/O.
CP4/SRNQ	AN4	AN	—	ADC Channel 4 input.
	CPS4	AN	—	Capacitive sensing input 4.
	C2OUT	_	CMOS	Comparator C2 output.
	T0CKI	ST	—	Timer0 clock input.
	CCP4	ST	CMOS	Capture/Compare/PWM4.
	SRNQ	—	CMOS	SR latch inverting output.
RA5/MCLR/Vpp/SS1	RA5	TTL	CMOS	General purpose I/O.
	MCLR	ST	—	Master Clear with internal pull-up.
	Vpp	HV	_	Programming voltage.
	SS1	ST	—	Slave Select input 1.
Legend: AN = Analog input or o TTL = TTL compatible i HV = High Voltage	output CMC nput ST XTAI	OS= CMO = Schi _ = Crys	DS compa mitt Trigg stal	atible input or output OD = Open Drain er input with CMOS levels I^2C^{TM} = Schmitt Trigger input with I^2C levels

TABLE 1-2: PIC16(L)F1847 PINOUT DESCRIPTION

Note 1: Pin functions can be moved using the APFCON0 or APFCON1 register.

2: Default function location.





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 0											
000h ⁽¹⁾	INDF0	Addressing th (not a physical	nis location use al register)	es contents of	FSR0H/FSR0	L to address	data memory	,		XXXX XXXX	XXXX XXXX
001h ⁽¹⁾	INDF1	Addressing th (not a physica	nis location use al register)	es contents of	FSR1H/FSR1	L to address	data memory	,		XXXX XXXX	XXXX XXXX
002h ⁽¹⁾	PCL	Program Cou	inter (PC) Lea	st Significant B	Byte					0000 0000	0000 0000
003h ⁽¹⁾	STATUS	_	_	_	TO	PD	Z	DC	С	1 1000	q quuu
004h ⁽¹⁾	FSR0L	Indirect Data	Memory Addr	ess 0 Low Poir	nter					0000 0000	uuuu uuuu
005h ⁽¹⁾	FSR0H	Indirect Data	Memory Addr	ess 0 High Poi	nter					0000 0000	0000 0000
006h ⁽¹⁾	FSR1L	Indirect Data	Memory Addr	ess 1 Low Poir	nter					0000 0000	uuuu uuuu
007h ⁽¹⁾	FSR1H	Indirect Data	Memory Addr	ess 1 High Poi	nter					0000 0000	0000 0000
008h ⁽¹⁾	BSR	_	_	_	BSR4	BSR3	BSR2	BSR1	BSR0	0 0000	0 0000
009h ⁽¹⁾	WREG	Working Reg	ister							0000 0000	uuuu uuuu
00Ah ⁽¹⁾	PCLATH	_	Write Buffer f	or the upper 7	bits of the Pro	ogram Counte	r			-000 0000	-000 0000
00Bh ⁽¹⁾	INTCON	GIE	PEIE	TMR0IE	INTE	IOCE	TMR0IF	INTF	IOCF	0000 000x	0000 000u
00Ch	PORTA	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	XXXX XXXX	XXXX XXXX
00Dh	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	XXXX XXXX	XXXX XXXX
00Eh	_	Unimplement	ed							_	_
00Fh	_	Unimplement	ed							_	_
010h	—	Unimplement	ed							_	_
011h	PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
012h	PIR2	OSFIF	C2IF	C1IF	EEIF	BCL1IF	_	_	CCP2IF	0000 00	0000 00
013h	PIR3	_	_	CCP4IF	CCP3IF	TMR6IF	_	TMR4IF	_	00 0-0-	00 0-0-
014h	PIR4	_	_	_	_	_	_	BCL2IF	SSP2IF	00	00
015h	TMR0	Timer0 Modu	le Register							XXXX XXXX	uuuu uuuu
016h	TMR1L	Holding Regi	ster for the Lea	ast Significant	Byte of the 16	5-bit TMR1 Re	gister			XXXX XXXX	uuuu uuuu
017h	TMR1H	Holding Regi	ster for the Mo	st Significant E	Byte of the 16	-bit TMR1 Re	gister			XXXX XXXX	uuuu uuuu
018h	T1CON	TMR1C	S<1:0>	T1CKP	S<1:0>	T1OSCEN	T1SYNC	_	TMR10N	0000 00-0	uuuu uu-u
019h	T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T <u>1GGO</u> / DONE	T1GVAL	T1GSS	S<1:0>	0000 0x00	uuuu uxuu
01Ah	TMR2	Timer2 Modu	le Register				•			0000 0000	0000 0000
01Bh	PR2	Timer2 Perio	d Register							1111 1111	1111 1111
01Ch	T2CON	—		T2OUTP	°S<3:0>		TMR2ON	T2CKP	S<1:0>	-000 0000	-000 0000
01Dh	—	Unimplement	ed							—	—
01Eh	CPSCON0	CPSON	CPSRM	_	—	CPSRN	G<1:0>	CPSOUT	TOXCS	00 0000	00 0000
01Fh	CPSCON1	—	—		_		CPSCH	1<3:0>	•	0000	0000

TABLE 3-8:	SPECIAL	FUNCTION REGISTER	SUMMARY

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

Note 1: These registers can be addressed from any bank.

2: Unimplemented, read as '1'.

3.6.1 TRADITIONAL DATA MEMORY

The traditional data memory is a region from FSR address 0x000 to FSR address 0x1FFF. The addresses correspond to the absolute addresses of all SFR, GPR and common registers.





5.5 Fail-Safe Clock Monitor

The Fail-Safe Clock Monitor (FSCM) allows the device to continue operating should the external oscillator fail. The FSCM can detect oscillator failure any time after the Oscillator Start-up Timer (OST) has expired. The FSCM is enabled by setting the FCMEN bit in the Configuration Words. The FSCM is applicable to all external Oscillator modes (LP, XT, HS, EC, Timer1 Oscillator and RC).

FIGURE 5-9: FSCM BLOCK DIAGRAM



5.5.1 FAIL-SAFE DETECTION

The FSCM module detects a failed oscillator by comparing the external oscillator to the FSCM sample clock. The sample clock is generated by dividing the LFINTOSC by 64. See Figure 5-8. Inside the fail detector block is a latch. The external clock sets the latch on each falling edge of the external clock. The sample clock clears the latch on each rising edge of the sample clock. A failure is detected when an entire half-cycle of the sample clock elapses before the external clock goes low.

5.5.2 FAIL-SAFE OPERATION

When the external clock fails, the FSCM switches the device clock to an internal clock source and sets the bit flag OSFIF of the PIR2 register. Setting this flag will generate an interrupt if the OSFIE bit of the PIE2 register is also set. The device firmware can then take steps to mitigate the problems that may arise from a failed clock. The system clock will continue to be sourced from the internal clock source until the device firmware successfully restarts the external oscillator and switches back to external operation.

The internal clock source chosen by the FSCM is determined by the IRCF<3:0> bits of the OSCCON register. This allows the internal oscillator to be configured before a failure occurs.

5.5.3 FAIL-SAFE CONDITION CLEARING

The Fail-Safe condition is cleared after a Reset, executing a SLEEP instruction or changing the SCS bits of the OSCCON register. When the SCS bits are changed, the OST is restarted. While the OST is running, the device continues to operate from the INTOSC selected in OSCCON. When the OST times out, the Fail-Safe condition is cleared after successfully switching to the external clock source. The OSFIF bit should be cleared prior to switching to the external clock source. If the Fail-Safe condition still exists, the OSFIF flag will again become set by hardware.

5.5.4 RESET OR WAKE-UP FROM SLEEP

The FSCM is designed to detect an oscillator failure after the Oscillator Start-up Timer (OST) has expired. The OST is used after waking up from Sleep and after any type of Reset. The OST is not used with the EC or RC Clock modes so that the FSCM will be active as soon as the Reset or wake-up has completed. When the FSCM is enabled, the Two-Speed Start-up is also enabled. Therefore, the device will always be executing code while the OST is operating.

Note:	Due to the wide range of oscillator start-up times, the Fail-Safe circuit is not active during oscillator start-up (i.e., after exiting Reset or Sleep). After an appropriate amount of time, the user should check the Status bits in the OSCSTAT register to verify the oscillator start-up and that the
	verify the oscillator start-up and that the system clock switchover has successfully completed.

R/W-x/u	R/W-x/u	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
LATA7	LATA6	—	LATA4	LATA3	LATA2	LATA1	LATA0
bit 7				•		•	bit 0
Legend:							
R = Readable	bit	W = Writable	bit	U = Unimpler	nented bit, read	as '0'	
u = Bit is unch	anged	x = Bit is unkr	iown	-n/n = Value a	at POR and BO	R/Value at all c	ther Resets

REGISTER 12-5: LATA: PORTA DATA LATCH REGISTER

bit 7-6	LATA<7:6>: RA<7:6> Output Latch Value bits ⁽¹⁾
bit 5	Unimplemented: Read as '0'

'1' = Bit is set

bit 4-0 LATA<4:0>: RA<4:0> Output Latch Value bits⁽¹⁾

Note 1: Writes to PORTA are actually written to corresponding LATA register. Reads from PORTA register is return of actual I/O pin values.

REGISTER 12-6: WPUA: WEAK PULL-UP PORTA REGISTER

'0' = Bit is cleared

U-0	U-0	R/W-1/1	U-0	U-0	U-0	U-0	U-0
—	—	WPUA5	—	—	—	—	—
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 7-6	Unimplemented: Read as '0'
bit 5	WPUA5: Weak Pull-up RA5 Control bit
	If MCLRE in Configuration Words = 0, MCLR is disabled):
	1 = Weak Pull-up enabled ⁽¹⁾
	0 = Weak Pull-up disabled
	If MCLRE in Configuration Words = 1, MCLR is enabled):
	Weak Pull-up is always enabled.
bit 4-0	Unimplemented: Read as '0'

Note 1: Global WPUEN bit of the OPTION_REG register must be cleared for individual pull-ups to be enabled.

2: The weak pull-up device is automatically disabled if the pin is in configured as an output.









23.0 DATA SIGNAL MODULATOR

The Data Signal Modulator (DSM) is a peripheral which allows the user to mix a data stream, also known as a modulator signal, with a carrier signal to produce a modulated output.

Both the carrier and the modulator signals are supplied to the DSM module either internally, from the output of a peripheral, or externally through an input pin.

The modulated output signal is generated by performing a logical "AND" operation of both the carrier and modulator signals and then provided to the MDOUT pin.

The carrier signal is comprised of two distinct and separate signals. A carrier high (CARH) signal and a carrier low (CARL) signal. During the time in which the modulator (MOD) signal is in a logic high state, the DSM mixes the carrier high signal with the modulator signal. When the modulator signal is in a logic low state, the DSM mixes the carrier low signal with the modulator signal. Using this method, the DSM can generate the following types of Key Modulation schemes:

- Frequency-Shift Keying (FSK)
- Phase-Shift Keying (PSK)
- On-Off Keying (OOK)

Additionally, the following features are provided within the DSM module:

- Carrier Synchronization
- · Carrier Source Polarity Select
- Carrier Source Pin Disable
- Programmable Modulator Data
- Modulator Source Pin Disable
- Modulated Output Polarity Select
- Slew Rate Control

Figure 23-1 shows a Simplified Block Diagram of the Data Signal Modulator peripheral.





R/W-x/u	R/W-x/u	R/W-x/u	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
MDCLODIS	MDCLPOL	MDCLSYNC			MDCL	<3:0>	
bit 7							bit 0
Legend:							
R = Readable	bit	W = Writable b	oit	U = Unimplen	nented bit, read	as '0'	
u = Bit is uncha	anged	x = Bit is unkn	own	-n/n = Value a	t POR and BO	R/Value at all c	other Resets
'1' = Bit is set		'0' = Bit is clea	ared				
bit 7	MDCLODIS: 1 = Output s is disable 0 = Output s is enable	Modulator Low ignal driving the ed ignal driving the ed	Carrier Outp peripheral o peripheral o	ut Disable bit utput pin (select utput pin (select	ed by MDCL<3 ed by MDCL<3	:0> of the MD0 :0> of the MD0	CARL register) CARL register)
bit 6	MDCLPOL:	Modulator Low (Carrier Polari	ty Select bit			
	1 = Selected 0 = Selected	l low carrier sigr l low carrier sigr	nal is inverted nal is not inve	erted			
bit 5	MDCLSYNC	: Modulator Low	Carrier Syn	chronization En	able bit		
	1 = Modulato	or waits for a falli rier	ing edge on t	he low time carr	ier signal before	e allowing a swi	itch to the high
	0 = Modulate	or Output is not	synchronize	d to the low time	carrier signal ⁽¹	1)	
bit 4	Unimplemen	nted: Read as 'o)'				
bit 3-0	MDCL<3:0>	Modulator Data	High Carrier	Selection bits (1)		
	1111 = Res	erved. No chan	inel connecte	ed.			
	•						
	•						
	1000 = Res 0111 = CCF 0110 = CCF 0101 = CCF 0100 = CCF 0011 = Refe 0010 = MDC 0001 = MDC 0000 = Vss	erved. No char 24 output (PWM 23 output (PWM 22 output (PWM 21 output (PWM erence Clock mo CIN2 port pin CIN1 port pin	Inel connecte Output moc Output moc Output moc Output moc odule signal	ed. le only) le only) le only) le only)			

REGISTER 23-4: MDCARL: MODULATION LOW CARRIER CONTROL REGISTER

Note 1: Narrowed carrier pulse widths or spurs may occur in the signal stream if the carrier is not synchronized.

|--|

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
MDCARH	MDCHODIS	MDCHPOL	MDCHSYNC	—		MDCF	<3:0>		200
MDCARL	MDCLODIS	MDCLPOL	MDCLSYNC	—	MDCL<3:0>			201	
MDCON	MDEN	MDOE	MDSLR	MDOPOL	MDOUT MDBIT		198		
MDSRC	MDMSODIS	_	—	—	MDMS<3:0>			199	

Legend: — = unimplemented, read as '0'. Shaded cells are not used in the Data Signal Modulator mode.

^{© 2011-2013} Microchip Technology Inc.

24.4.3 ENHANCED PWM AUTO-SHUTDOWN MODE

The PWM mode supports an Auto-Shutdown mode that will disable the PWM outputs when an external shutdown event occurs. Auto-Shutdown mode places the PWM output pins into a predetermined state. This mode is used to help prevent the PWM from damaging the application.

The auto-shutdown sources are selected using the CCPxAS<2:0> bits of the CCPxAS register. A shutdown event may be generated by:

- A logic '0' on the FLT0 pin
- A logic '1' on a Comparator (Cx) output

A shutdown condition is indicated by the CCPxASE (Auto-Shutdown Event Status) bit of the CCPxAS register. If the bit is a '0', the PWM pins are operating normally. If the bit is a '1', the PWM outputs are in the shutdown state.

When a shutdown event occurs, two things happen:

The CCPxASE bit is set to '1'. The CCPxASE will remain set until cleared in firmware or an auto-restart occurs (see Section 24.4.4 "Auto-restart Mode").

The enabled PWM pins are asynchronously placed in their shutdown states. The PWM output pins are grouped into pairs [PxA/PxC] and [PxB/PxD]. The state of each pin pair is determined by the PSSxAC and PSSxBD bits of the CCPxAS register. Each pin pair may be placed into one of three states:

- Drive logic '1'
- Drive logic '0'
- Tri-state (high-impedance)

- Note 1: The auto-shutdown condition is a level-based signal, not an edge-based signal. As long as the level is present, the auto-shutdown will persist.
 - Writing to the CCPxASE bit of the CCPxAS register is disabled while an auto-shutdown condition persists.
 - 3: Once the auto-shutdown condition has been removed and the PWM restarted (either through firmware or auto-restart) the PWM signal will always restart at the beginning of the next PWM period.
 - 4: Prior to an auto-shutdown event caused by a comparator output or FLT0 pin event, a software shutdown can be triggered in firmware by setting the CCPxASE bit of the CCPxAS register to '1'. The Auto-Restart feature tracks the active status of a shutdown caused by a comparator output or FLT0 pin event only. If it is enabled at this time, it will immediately clear this bit and restart the ECCP module at the beginning of the next PWM period.



FIGURE 24-14: PWM AUTO-SHUTDOWN WITH FIRMWARE RESTART (PXRSEN = 0)

When one device is transmitting a logical one, or letting the line float, and a second device is transmitting a logical zero, or holding the line low, the first device can detect that the line is not a logical one. This detection, when used on the SCLx line, is called clock stretching. Clock stretching gives slave devices a mechanism to control the flow of data. When this detection is used on the SDAx line, it is called arbitration. Arbitration ensures that there is only one master device communicating at any single time.

25.3.1 CLOCK STRETCHING

When a slave device has not completed processing data, it can delay the transfer of more data through the process of Clock Stretching. An addressed slave device may hold the SCLx clock line low after receiving or sending a bit, indicating that it is not yet ready to continue. The master that is communicating with the slave will attempt to raise the SCLx line in order to transfer the next bit, but will detect that the clock line has not yet been released. Because the SCLx connection is open-drain, the slave has the ability to hold that line low until it is ready to continue communicating.

Clock stretching allows receivers that cannot keep up with a transmitter to control the flow of incoming data.

25.3.2 ARBITRATION

Each master device must monitor the bus for Start and Stop bits. If the device detects that the bus is busy, it cannot begin a new message until the bus returns to an Idle state.

However, two master devices may try to initiate a transmission on or about the same time. When this occurs, the process of arbitration begins. Each transmitter checks the level of the SDAx data line and compares it to the level that it expects to find. The first transmitter to observe that the two levels do not match, loses arbitration, and must stop transmitting on the SDAx line.

For example, if one transmitter holds the SDAx line to a logical one (lets it float) and a second transmitter holds it to a logical zero (pulls it low), the result is that the SDAx line will be low. The first transmitter then observes that the level of the line is different than expected and concludes that another transmitter is communicating.

The first transmitter to notice this difference is the one that loses arbitration and must stop driving the SDAx line. If this transmitter is also a master device, it also must stop driving the SCLx line. It then can monitor the lines for a Stop condition before trying to reissue its transmission. In the meantime, the other device that has not noticed any difference between the expected and actual levels on the SDAx line continues with its original transmission. It can do so without any complications, because so far, the transmission appears exactly as expected with no other transmitter disturbing the message.

Slave Transmit mode can also be arbitrated, when a master addresses multiple slaves, but this is less common.

If two master devices are sending a message to two different slave devices at the address stage, the master sending the lower slave address always wins arbitration. When two master devices send messages to the same slave address, and addresses can sometimes refer to multiple slaves, the arbitration process must continue into the data stage.

Arbitration usually occurs very rarely, but it is a necessary process for proper multi-master support.

TABLE 26-3: BAUD RATE FORMULAS

Configuration Bits				Reud Rete Formula		
SYNC	BRG16	BRGH	BRG/EUSART Mode	Baud Rate Formula		
0	0	0	8-bit/Asynchronous	Fosc/[64 (n+1)]		
0	0	1	8-bit/Asynchronous			
0	1	0	16-bit/Asynchronous	FOSC/[16 (n+1)]		
0	1	1	16-bit/Asynchronous			
1	0	х	8-bit/Synchronous	Fosc/[4 (n+1)]		
1	1	x	16-bit/Synchronous			

Legend: x = Don't care, n = value of SPBRGH, SPBRGL register pair

TABLE 26-4: SUMMARY OF REGISTERS ASSOCIATED WITH THE BAUD RATE GENERATOR

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	_	WUE	ABDEN	298
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	297
SPBRGL	BRG<7:0>								299*
SPBRGH	BRG<15:8>								299*
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	296

Legend: — = unimplemented location, read as '0'. Shaded cells are not used for the Baud Rate Generator.

* Page provides register information.

26.4.2.3 EUSART Synchronous Slave Reception

The operation of the Synchronous Master and Slave modes is identical (Section 26.4.1.5 "Synchronous Master Reception"), with the following exceptions:

- Sleep
- CREN bit is always set, therefore the receiver is never Idle
- SREN bit, which is a "don't care" in Slave mode

A character may be received while in Sleep mode by setting the CREN bit prior to entering Sleep. Once the word is received, the RSR register will transfer the data to the RCREG register. If the RCIE enable bit is set, the interrupt generated will wake the device from Sleep and execute the next instruction. If the GIE bit is also set, the program will branch to the interrupt vector.

- 26.4.2.4 Synchronous Slave Reception Setup:
- 1. Set the SYNC and SPEN bits and clear the CSRC bit.
- 2. Clear the ANSEL bit for both the CK and DT pins (if applicable).
- 3. If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
- 4. If 9-bit reception is desired, set the RX9 bit.
- 5. Set the CREN bit to enable reception.
- The RCIF bit will be set when reception is complete. An interrupt will be generated if the RCIE bit was set.
- 7. If 9-bit mode is enabled, retrieve the Most Significant bit from the RX9D bit of the RCSTA register.
- 8. Retrieve the eight Least Significant bits from the receive FIFO by reading the RCREG register.
- 9. If an overrun error occurs, clear the error by either clearing the CREN bit of the RCSTA register or by clearing the SPEN bit which resets the EUSART.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
APFCON0	RXDTSEL	SDO1SEL	SS1SEL	P2BSEL	CCP2SEL	P1DSEL	P1CSEL	CCP1SEL	118
APFCON1	—	_		—	—		_	TXCKSEL	118
BAUDCON	ABDOVF	RCIDL		SCKP	BRG16	-	WUE	ABDEN	298
INTCON	GIE	PEIE	TMR0IE	INTE	IOCE	TMR0IF	INTF	IOCF	88
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	89
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	93
RCREG	EUSART Receive Data Register							292*	
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	297
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	126
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	296

TABLE 26-10: SUMMARY OF REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION

Legend: — = unimplemented location, read as '0'. Shaded cells are not used for Synchronous Slave Reception.

* Page provides register information.

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:	$\label{eq:W} \begin{split} W &\rightarrow \text{INDFn} \\ \text{Effective address is determined by} \\ \bullet \ \text{FSR} + 1 \ (\text{preincrement}) \\ \bullet \ \text{FSR} - 1 \ (\text{predecrement}) \\ \bullet \ \text{FSR} + k \ (\text{relative offset}) \\ \text{After the Move, the FSR value will be} \\ \text{either:} \\ \bullet \ \text{FSR} + 1 \ (\text{all increments}) \\ \bullet \ \text{FSR} - 1 \ (\text{all decrements}) \\ \text{Unchanged} \end{split}$					
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

NOP	No Operation
Syntax:	[<i>label</i>] 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.				
Words:	1				
Cycles:	1				
Example:	OPTION				
	Before Instruction OPTION_REG = 0xFF W = 0x4F After Instruction OPTION_REG = 0x4F W = 0x4F				

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 soft- ware

TABLE 30-2:	SUPPLY CURRENT	(IDD) ^(1,2)
-------------	----------------	------------------------

PIC16LF	PIC16LF1847		Standard Operating Conditions (unless otherwise stated)							
PIC16F1	847									
Param.	Device	Min	Trank			Conditions				
No.	Characteristics	win.	турт	max.	Units	VDD	Note			
D010		—	9.5	14	μA	1.8	Fosc = 32 kHz			
		—	12.5	17	μA	3.0	LP Oscillator -40°C \leq TA \leq +85°C			
D010		_	22	29	μA	1.8	Fosc = 32 kHz			
			27	35	μA	3.0	LP Oscillator $40^{\circ}C < T_A < +85^{\circ}C$			
		_	30	38	μA	5.0	-40 C \leq IA \leq $+00$ C			
D010A			9.5	14	μA	1.8	Fosc = 32 kHz			
		_	12.5	17	μA	3.0	LP Oscillator -40°C \leq TA \leq +125°C			
D010A		_	22	29	μA	1.8	Fosc = 32 kHz			
		_	27	35	μA	3.0 LP Oscillator	LP Oscillator			
		_	30	38	μA	5.0	-40 C \leq IA \leq \pm 123 C			
D011		_	105	110	μA	1.8	Fosc = 1 MHz			
		_	160	190	μA	3.0	XT Oscillator			
D011		_	132	154	μA	1.8	Fosc = 1 MHz			
			186	220	μA	3.0	X I Oscillator			
		_	216	290	μA	5.0				
D012		_	264	370	μA	1.8	Fosc = 4 MHz			
		_	491	620	μA	3.0	XT Oscillator			
D012			285	300	μA	1.8	Fosc = 4 MHz			
			408	600	μA	3.0	X I Oscillator			
			490	700	μA	5.0				
D013		_	55	160	μA	1.8	Fosc = 1 MHz			
		_	90	230	μA	3.0	EC Oscillator Medium-Power mode			
D013			75	95	μA	1.8	Fosc = 1 MHz			
			116	130	μA	3.0	EC Oscillator Medium-Power mode			
			145	185	μA	5.0				
*	Th		a si a al la s							

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 VDD; 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: 8 MHz internal oscillator with 4x PLL enabled.
- 4: 8 MHz crystal oscillator with 4x PLL enabled.

5: 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 Ω .













Note 1: If the ADC clock source is selected as FRC, a time of TCY is added before the ADC clock starts. This allows the SLEEP instruction to be executed.



FIGURE 31-4: IDD MAXIMUM, XT AND EXTRC OSCILLATOR, PIC16LF1847 ONLY















